



(12) 发明专利

(10) 授权公告号 CN 101366080 B

(45) 授权公告日 2011. 10. 19

(21) 申请号 200780002049. 9

G10L 21/04 (2006. 01)

(22) 申请日 2007. 08. 15

(56) 对比文件

(30) 优先权数据

- 60/837, 627 2006. 08. 15 US
- 60/848, 049 2006. 09. 29 US
- 60/848, 051 2006. 09. 29 US
- 60/853, 461 2006. 10. 23 US

- US 20060045138 A1, 2006. 03. 02, 全文.
- EP 1288916 A2, 2002. 08. 16, 全文.
- US 7047190 B1, 2006. 05. 16, 全文.
- SERIZAWA M ET AL. A packet loss concealment method using pitch waveform repetition and internal state update on the decoded speech for the sub-band adpcm wideband speech codec. 《SPEECH CODING, 2002, IEEE WORKSHOP PROCEEDINGS》. 2002, 68-70.

(85) PCT申请进入国家阶段日 2008. 07. 07

(86) PCT申请的申请数据 PCT/US2007/076009 2007. 08. 15

(87) PCT申请的公布数据 W02008/022200 EN 2008. 02. 21

(73) 专利权人 美国博通公司
地址 美国加州

- CHIBANI M ET AL. Resynchronization of the Adaptive Codebook in a Constrained celp Codec After a Frame Erasure. 《ACOUSTIC, SPEECH AND SIGNAL PRECESSING, 2006》. 2006, 1-13.

(72) 发明人 罗伯塔·W·措普夫 杰斯·赛森
朱因韦·陈

审查员 张鑫

(74) 专利代理机构 深圳市顺天达专利商标代理有限公司 44217
代理人 蔡晓红 李琴

(51) Int. Cl. G10L 19/00 (2006. 01)

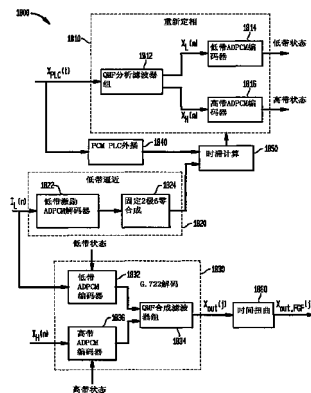
权利要求书 2 页 说明书 49 页 附图 23 页

(54) 发明名称

一种更新解码器的状态的方法和系统

(57) 摘要

本发明描述了一种更新用于解码表示编码音频信号的一系列帧的解码器状态的技术。根据所述技术，将解码器状态设定为在帧边界上与合成的输出音频信号对齐。基于所述合成的输出音频信号产生外插信号。计算所述外插信号和与所述一系列帧中丢失帧后的第一接收帧关联的解码音频信号之间的时滞，其中所述时滞表示所述外插信号和所述解码音频信号之间的相位差。然后基于所述时滞重设所述解码器状态。



CN 101366080 B

1. 一种更新用于解码表示编码音频信号的一系列帧的解码器的状态的方法,其特征在于,所述方法包括:

合成与所述一系列帧中的丢失帧关联的输出音频信号;

设定所述解码器状态为在帧边界上与所述合成的输出音频信号对齐;

基于所述合成的输出音频信号产生外插信号;

计算所述外插信号和与所述一系列帧中丢失帧后的第一接收帧关联的解码音频信号之间的时滞,其中所述时滞表示所述外插信号和所述解码音频信号之间的相位差;及

基于所述时滞重设所述解码器状态。

2. 根据权利要求1所述的方法,其特征在于,设定所述解码器状态为在帧边界上与所述合成的输出音频信号对齐包括对表示所述合成的输出音频信号一系列采样重新编码直到所述帧边界;以及

基于所述时滞重设所述解码器状态包括对表示所述合成的输出音频信号的一系列采样重新编码直到所述帧边界加上或减去与所述时滞关联的多个采样。

3. 根据权利要求1所述的方法,其特征在于,计算所述外插信号和所述解码音频信号之间的时滞包括最大化所述外插信号和所述解码音频信号之间的相关性。

4. 根据权利要求1所述的方法,其特征在于,计算所述外插信号和所述解码音频信号之间的时滞包括:

使用第一滞后搜索范围和第一滞后搜索窗口搜索所述外插信号与所述解码音频信号之间的标准化交叉相关性函数的第一峰值以确定粗时滞,其中所述第一滞后搜索范围指定在搜索过程中所述外插信号的起始点移位的范围,所述第一滞后搜索窗口指定计算所述标准化交叉相关性函数的采样数量;及

使用第二滞后搜索范围和第二滞后搜索窗口搜索所述外插信号与所述解码音频信号之间的标准化交叉相关性函数的第二峰值以确定细化时滞,其中第二滞后搜索范围小于第一滞后搜索范围。

5. 根据权利要求4所述的方法,其特征在于,搜索所述外插信号与所述解码音频信号之间的标准化交叉相关性函数的第一峰值包括搜索所述外插信号与所述解码音频信号的下采样样本之间的标准化交叉相关性函数的峰值。

6. 根据权利要求4所述的方法,其特征在于,所述第二滞后搜索窗口小于第一滞后搜索窗口。

7. 一种更新用于解码表示编码音频信号的一系列帧的解码器的状态的系统,其特征在于,包括:

用于对表示编码音频信号的一系列帧中的接收帧进行解码的解码器;

用于合成与所述一系列帧中的丢失帧关联的输出音频信号的音频信号合成器;

解码器状态更新逻辑,用于在产生所述合成的输出音频信号后设定所述解码器状态为在帧边界上与所述合成的输出音频信号对齐,基于所述合成的输出音频信号产生外插信号,计算所述外插信号和与所述一系列帧中丢失帧后的第一接收帧关联的解码音频信号之间的时滞,以及基于所述时滞重设所述解码器状态,其中所述时滞表示所述外插信号和所述解码音频信号之间的相位差。

8. 根据权利要求7所述的系统,其特征在于,所述解码器状态更新逻辑用于通过对表

示所述合成的输出音频信号的一系列采样重新编码直到所述帧边界来设定所述解码器状态为在帧边界上与所述合成的输出音频信号对齐;以及

所述解码器状态更新逻辑用于通过对表示所述合成的输出音频信号的一系列采样重新编码直到的所述帧边界加上或减去与所述时滞关联的多个采样来基于所述时滞重设所述解码器状态。

9. 根据权利要求 7 所述的系统,其特征在于,所述解码器状态更新逻辑用于通过最大化所述外插信号和所述解码音频信号之间的相关性来来计算所述外插信号和所述解码音频信号之间的时滞。

一种更新解码器的状态的方法和系统

技术领域

[0001] 本发明涉及一种在语音或音频编码器中隐藏丢包导致的质量降低效果的系统和方法。

[0002] 背景技术

[0003] 在声音或音频信号通过分组网络进行数字传输的过程中,编码的声音 / 音频信号通常被分割成帧,然后封装成包,其中每个包可包含一个或多个编码声音 / 音频数据的帧。然后通过分组网络传输这些包。有时候一些包会丢失,而且一些有用的包会太晚到达,从而被认定为丢失。这种包丢失会引起音频质量的显著降低,除非使用特殊的技术来隐藏丢包所导致的效果。

[0004] 当前存在用于独立块编码器或全频带预测编码器的基于外插音频信号的包丢失隐藏 (packet loss concealment, 缩写为 PLC) 方法。这种 PLC 方法包括以下美国专利申请中所公开的技术:申请号为 11/234, 291、发明人为陈、名称为“用于独立块语音编解码器的丢包隐藏技术”的美国专利申请和申请号为 10/183, 608、发明人为陈、名称为“用于预测语音编码的基于外插语音波形的帧删除隐藏方法和系统”的美国专利申请。然而,在这些申请中描述的技术不能直接用于子带预测编码器,如 ITU-T 推荐 G. 722 宽带语音编码器,这是因为存在这些技术未能解决的子带特定结构问题。此外,对于每个子带来说,G. 722 编码器使用了自适应差分脉冲编码调制 (ADPCM) 预测编码器,该 ADPCM 预测编码器使用了基于梯度方法的量化器步长和预测器系数的逐个采样 (sample-by-sample) 后向自适应,这造成了现有 PLC 技术未能解决的特殊难题。因此,需要一种为子带预测编码器 (如 G. 722) 特别设计的合适的 PLC 方法。

[0005] 发明内容

[0006] 本发明用于在子带预测编码器中隐藏丢包导致的质量降低效果。本发明具体解决了在对子带预测编码器应用音频波形外插技术时的一些子带特定的结构问题,本发明还解决了一般的后向自适应 ADPCM 编码器及特殊的 G. 722 子带 ADPCM 编码器的特定 PLC 难题。

[0007] 具体地说,本发明在此描述了一种更新用于解码表示编码音频信号的一系列帧的解码器的状态的方法。根据所述方法,合成与所述一系列帧中的丢失帧关联的输出音频信号。将所述解码器状态设定为在帧边界上与所述合成的输出音频信号对齐。基于所述合成的输出音频信号产生外插信号。计算出所述外插信号和与所述一系列帧中丢失帧后的第一接收帧关联的解码音频信号之间的时滞,其中所述时滞表示所述外插信号和所述解码音频信号之间的相位差。然后基于所述时滞重设所述解码器状态。

[0008] 本发明在此还描述了一种系统。所述系统包括解码器、音频信号合成器及解码器更新逻辑。所述解码器用于对表示编码音频信号的一系列帧中的接收帧进行解码。所述音频信号合成器用于合成与所述一系列帧中的丢失帧关联的输出音频信号。所述解码器状态更新逻辑用于在产生所述合成输出音频信号后设定所述解码器状态为在帧边界上与所述合成的输出音频信号对齐,基于所述合成的输出音频信号产生外插信号,计算所述外插信号和与所述一系列帧中丢失帧后的第一接收帧关联的解码音频信号之间的时滞,以及基于

所述时滞重设所述解码器状态。所述时滞表示所述外插信号和所述解码音频信号之间的相位差。

[0009] 本发明还描述一种计算机程序产品。所述计算机程序产品包括记录有计算机程序逻辑的计算机可读媒介,所述计算机程序逻辑用于使处理器更新用于解码表示编码音频信号的一系列帧的解码器的状态。所述计算机程序逻辑包括第一模块、第二模块、第三模块、第四模块及第五模块。第一模块用于使所述处理器合成与所述一系列帧中的丢失帧关联的输出音频信号。第二模块用于使所述处理器在产生所述合成输出音频信号后设定所述解码器状态为在帧边界上与所述合成的输出音频信号对齐。第三模块用于使所述处理器基于所述合成的输出音频信号产生外插信号。第四模块用于使所述处理器计算所述外插信号和与所述一系列帧中丢失帧后的第一接收帧关联的解码音频信号之间的时滞。第五模块用于使所述处理器基于所述时滞重设所述解码器状态,其中所述时滞表示所述外插信号和所述解码音频信号之间的相位差。

[0010] 本发明的更多特征和优点以及本发明各种实施例的结构和操作将参考附图作进一步的细节描述。注意,本发明并不局限于在此所述的特殊实施例。在此提出的实施例仅作为示例性的目的。基于包含在此的教导,更多的实施例对于所属领域的技术人员来说是显而易见的。

附图说明

[0011] 在此结合的附图是说明书的一部分,附图与文字描述一起阐明了本发明一个或多个实施例,并进一步用于解释本发明的用途、优点和原理,以及使所属领域的技术人员实施和使用本发明。

[0012] 图 1 是传统的 ITU-T G. 722 子带预测编码器的编码器结构的示意图;

[0013] 图 2 是传统的 ITU-T G. 722 子带预测编码器的解码器结构的示意图;

[0014] 图 3 是根据本发明实施例的解码器 /PLC 系统的模块图;

[0015] 图 4 是根据本发明实施例在解码器 /PLC 系统中处理帧以产生输出语音信号的方法流程图;

[0016] 图 5 是根据本发明实施例可由解码器 /PLC 系统处理的不同类型帧的时序图;

[0017] 图 6 是原始语音信号和外插语音信号的振幅的时间线示意图;

[0018] 图 7 是根据本发明实施例在解码语音信号和外插语音信号之间计算时滞 (time lag) 的方法流程图;

[0019] 图 8 是根据本发明实施例在解码语音信号和外插语音信号之间计算时滞的两个阶段的方法流程图;

[0020] 图 9 是根据本发明实施例在时滞计算执行过程中可相对于解码语音信号平移外插语音信号的方式的示意图;

[0021] 图 10A 是根据本发明实施例表示超前于外插语音信号的解码语音信号和重新编码操作的相关效果的时间线示意图;

[0022] 图 10B 是根据本发明实施例表示滞后于外插语音信号的解码语音信号和重新编码操作的相关效果的时间线示意图;

[0023] 图 10C 是根据本发明实施例表示在帧边界上同步的外插语音信号和解码语音信

号以及重新编码操作的相关效果的时间线示意图；

[0024] 图 11 是根据本发明实施例在丢包后执行子带 ADPCM 解码器内部状态的重新定相 (re-phasing) 的方法流程图；

[0025] 图 12A 是根据本发明实施例对超前于外插语音信号的解码语音信号应用时间扭曲 (time-warping) 的示意图；

[0026] 图 12B 和 12C 均是根据本发明实施例对滞后于外插语音信号的解码语音信号应用时间扭曲的示意图；

[0027] 图 13 是根据本发明实施例执行时间扭曲以沿时间轴收缩信号的一种方法的流程图；

[0028] 图 14 是根据本发明实施例执行时间扭曲以沿时间轴拉伸信号的一种方法的流程图；

[0029] 图 15 是根据本发明实施例的用于在解码器 /PLC 系统内发生丢包后对预定数量的接收帧之后的接收帧进行处理的逻辑的模块图；

[0030] 图 16 是根据本发明实施例的用于在解码器 /PLC 系统中执行波形外插来生成与丢失的帧相关联的输出语音信号的逻辑的模块图；

[0031] 图 17 是根据本发明实施例的用于更新解码器 /PLC 系统中的子带 ADPCM 解码器状态的逻辑的模块图；

[0032] 图 18 是根据本发明实施例的用于在解码器 /PLC 系统中执行重新定相和时间扭曲的逻辑的模块图；

[0033] 图 19 是根据本发明实施例的用于在解码器 /PLC 系统中对丢包后接收到的好帧执行约束和受控解码的逻辑的模块图；

[0034] 图 20 是根据本发明实施例的用于在丢包过程中更新低带 ADPCM 解码器的内部状态的简化低带 ADPCM 编码器的模块图；

[0035] 图 21 是根据本发明实施例的用于在丢包过程中更新高带 ADPCM 解码器的内部状态的简化高带 ADPCM 编码器的模块图；

[0036] 图 22A、22B 和 22C 均是根据本发明实施例对解码语音信号应用时间扭曲的时间线示意图；

[0037] 图 23 是根据本发明实施例的另一解码器 /PLC 系统的模块图；

[0038] 图 24 是实现本发明的实施例的计算机系统的模块图。

[0039] 通过结合附图做出的详细描述,本发明特征和优点将变得更加显而易见。附图中第一次出现的组件由对应的附图标记中最左边的阿拉伯数字表示。

具体实施方式

[0040] A、引言

[0041] 以下将参考附图对本发明的典型实施例作了详细描述。其它实施例也是可行的,而且可以在本发明精神和范围之内对示例性实施例做出修改。因此,以下的详细描述并不用于限制本发明。相反,本发明的范围是由权利要求定义的。

[0042] 所属领域的技术人员应容易理解,如以下所描述的,本发明可以在硬件、软件、软硬件和 / 或附图所示的实体中实现。任何以特定控制硬件实现本发明的实际软件代码并不

是本发明的限制。因而，以下对本发明的操作和行为的描述是基于以下理解给出的，即根据本申请中给出的详细描述的过程，可以对本申请中的实施例进行各种修改和变化。

[0043] 应理解，尽管在此提出的本发明的细节描述是针对的语音信号的处理，本发明还可用于涉及其它类型音频信号的处理。因此，在此使用的术语“语音”和“语音信号”仅仅是为了描述方便，并不作为限制。所属领域的技术人员应知悉，该术语可以用更常用的术语“音频”和“音频信号”来代替。此外，虽然语音和音频信号在此描述为被分割成多个帧，所属领域的技术人员应知悉，这种信号也可以被分割成其它离散的信号段，包括但不限于子帧。因而，在此描述的在帧上执行的操作也包括在语音或音频信号的其它段上执行的类似操作，如子帧。

[0044] 另外，虽然以下描述讨论了通过分组网络传输的音频信号的帧丢失（被称为丢包），但是本发明并不限于丢包隐藏（PLC）。例如，在无线网络中，由于信道损坏，也可能丢失或删除音频信号帧。这种情形称之为“帧删除”。当这种情形发生时，为了避免输出语音质量上的实质下降，无线系统中的解码器需要执行“帧删除隐藏”（FEC）来尝试隐藏因丢帧而导致的质量下降。对于 PLC 或 FEC 算法，丢包和帧删除都遇到相同的问题：某些传输的帧不能再用于解码，所以 PLC 或 FEC 算法需要产生波形来填补对应于丢失的帧的波形间隙（gap），从而隐藏帧因丢帧导致的质量下降。因为术语 FEC 和 PLC 通常指相同种类的技术，所以可以交替地使用。因而，为了方便起见，在此所用的术语“丢包隐藏”或 PLC 用于指代两者。

[0045] B、子带预测编码的回顾

[0046] 为了有利于更好地理解在后面章节中描述的本发明的各实施例，在此回顾一下子带预测编码的基本原理。通常，子带预测编码器可将输入语音信号分离成 N 个子带，其中 $N \geq 2$ 。在不失一般性的情况下，在此将 ITU-T G. 722 编码器的双带预测编码系统作为示例进行描述。所属领域的技术人员能够容易地将该描述概括到其它 N 带子带预测编码器。

[0047] 图 1 是 G. 722 子带预测编码器的简化编码器结构 100。编码器结构 100 包括正交镜像滤波器（QMF）分析滤波器组 110、低带自适应差分脉码调制（ADPCM）编码器 120、高带 ADPCM 编码器 130 和比特流复用器 140。QMF 分析滤波器组 110 将输入语音信号分离成低带语音信号和高带语音信号。低带 ADPCM 编码器 120 将低带语音信号编码成低带比特流。高带 ADPCM 编码器 130 将高带语音信号编码成高带比特流。比特流复用器 140 将低带比特流和高带比特流复用成单个输出比特流。在此讨论的分组传输应用中，该输出比特流被封装成包，然后被传输到子带预测解码器 200，如图 2 所示。

[0048] 如图 2 所示，解码器 200 包括比特流解复用器 210、低带 ADPCM 解码器 220、高带 ADPCM 解码器 230 和 QMF 合成滤波器组 240。比特流解复用器 210 将输入比特流分离成低带比特流和高带比特流。低带 ADPCM 解码器 220 将低带比特流解码成解码低带语音信号。高带 ADPCM 解码器 230 将高带比特流解码成解码高带语音信号。然后 QMF 合成滤波器组 240 将解码低带语音信号和解码高带语音信号合并成全带输出语音信号。

[0049] 有关编码器 100 和解码器 200 的结构和操作的更多细节可以在 ITU-T 推荐 G. 722 中找到，在此将其整体作为参考引入本文。

[0050] C、基于全带语音波形外插（extrapolation）的子带预测编码器的丢包隐藏技术

[0051] 现在对根据本发明的一个实施例的高质量 PLC 系统和方法进行描述。在本节中提

供了对该系统和方法的概括介绍,涉及该系统和方法具体实现的更多细节将在以下 D 节中进行描述。该示例系统和方法用于 ITU-T 推荐 G. 722 语音编码器。然而,所属领域的技术人员应知悉,在此参考该特殊实施例描述的许多概念可以用于在其它类型的子带预测语音编码器以及其它类型的语音和音频编码器中执行 PLC。

[0052] 如在此更多的细节中描述的,该实施例在 G. 722 语音解码器的 16kHz 输出域中执行 PLC。该方法使用周期性波形外插来填充与语音信号的丢帧相关联的波形,其中根据发生丢帧前的信号特征将外插波形与经滤波的噪声混合。为了更新子带 ADPCM 解码器的状态,外插的 16kHz 信号通过 QMF 分析滤波器组来生成子带信号,然后该子带信号由简化的子带 ADPCM 编码器处理。为了提供从与丢失的帧关联的外插波形到与丢包后接收的好帧关联的正常解码波形的平滑过渡,每次丢包后会执行额外的处理。其中,子带 ADPCM 解码器的状态与丢包后接收的第一个好帧进行相位对齐,并将与第一个好帧关联的正常解码波形进行时间扭曲,以在其与插值波形叠加之间将该正常解码波形与插值波形对齐,从而实现平滑过渡。对于长期的丢包情况,该系统和方法将逐渐将输出信号减弱。

[0053] 图 3 是实现这种 PLC 功能的 G. 722 语音解码器 300 的高级模块图。虽然在此描述的解码器 /PLC 系统 300 包括 G. 722 解码器,但是所属领域的技术人员应知悉,在此描述的许多概念通常可用于任何 N 带子带预测编码系统。类似地,针对每个子带的预测编码器不必是图 3 所示的 ADPCM 编码器,也可以是任何普通的预测编码器,并且可以是前向自适应或者后向自适应的。

[0054] 如图 3 所示,解码器 /PLC 系统 300 包括比特流解复用器 310、低带 ADPCM 解码器 320、高带 ADPCM 解码器 330、切换器 336、QMF 合成滤波器组 340、全带语音信号合成器 350、子带 ADPCM 解码器状态更新模块 360 和解码约束和控制模块 370。

[0055] 在此使用的术语“丢失帧”或者“坏帧”指的是没有在解码器 /PLC300 接收到的或者被认为不适合正常解码操作的语音信号帧。“接收帧”或“好帧”是在解码器 /PLC 系统 300 正常接收的语音信号帧。“当前帧”是当前正由解码器 /PLC300 处理以生成输出语音信号的帧,而“前帧”是之前由解码器 /PLC 系统 300 处理来生成输出语音信号的帧。术语“当前帧”和“前帧”均可用来指接收的帧以及正执行 PLC 操作和丢失帧。

[0056] 解码器 /PLC 系统 300 操作的方式将参考图 4 的流程图 400 来进行描述。如图 4 所示,流程图 400 的方法在步骤 402 开始,解码器 /PLC 系统 300 确定当前帧的帧类型。解码器 /PLC 系统 300 辨别六种不同类型的帧,分别用类型 1 到 6 表示。图 5 提供了不同帧类型的时间线 500。类型 1 的帧是丢包后的第八个接收帧后的任何接收帧。类型 2 的帧是与丢包相关的第一和第二丢失帧。类型 3 的帧是与丢包相关的第三到第六丢失帧中的任何一个。类型 4 的帧是与丢包相关的第六个丢失帧后的任何一个丢失帧。类型 5 的帧是紧随丢包后接收的任何接收帧。最后,类型 6 的帧是在丢包后接收的第二到第八个接收帧中的任何一个。所属领域的技术人员应容易知悉,根据本发明的替代实施例也可使用分类帧类型的其它方案。例如,在具有不同帧大小的系统中,每个帧类型中的帧数都与上述不同。同样,对于不同的编解码器(即非 G. 722 编解码器),每个帧类型中的帧数可以是不同的。

[0057] 解码器 /PLC 系统 300 处理当前帧来产生输出语音信号的方式是由当前帧的帧类型确定的。这在图 4 中通过一系列判断步骤 404、406、408 和 410 表现出来。具体地说,如果在步骤 402 确定当前帧是类型 1 的帧,那么执行第一序列的处理步骤来产生输出语音信号,

如判断步骤 404 所示。如果在步骤 402 确定当前帧是类型 2、类型 3 或类型 4 的帧,那么执行第二序列的处理步骤来产生输出语音信号,如判断步骤 406 所示。如果在步骤 402 确定当前帧是类型 5 的帧,那么执行第三序列的处理步骤来产生输出语音信号,如判断步骤 408 所示。最后,如果在步骤 402 确定当前帧是类型 6 的帧,那么执行第四序列的处理步骤来产生输出语音信号,如判断步骤 410 所示。以下将描述与每种不同帧类型关联的处理步骤。

[0058] 在执行完每个序列的处理步骤之后,在判断步骤 430 中确定是否有额外的帧要处理。如果有额外的帧要处理,那么处理返回到步骤 402。然而,如果没有额外的帧要处理,那么处理在所示步骤 432 结束。

[0059] 1、处理类型 1 的帧

[0060] 如在流程图 400 的步骤 412 所示,如果当前帧是类型 1 的帧,那么解码器 /PLC 系统 300 执行当前帧的正常 G. 722 解码。因此,解码器 /PLC 系统 300 的模块 310、320、330 和 340 正确地执行分别与传统 G. 722 解码器 200 相对的模块 210、220、230 和 240 相同的功能。具体地说,比特流解复用器 310 将输入比特流分离成低带比特流和高带比特流。低带 ADPCM 解码器 320 将低带比特流解码成解码低带语音信号。高带 ADPCM 解码器 330 将高带比特流解码成解码高带语音信号。然后 QMF 合成滤波组 340 将解码低带语音信号和解码高带语音信号重新合并成全带语音信号。在处理类型 1 的帧的过程中,切换器 336 连接到标记为“类型 1”的上面位置,从而将 QMF 合成滤波组 340 的输出信号作为针对类型 1 的帧的解码器 /PLC 系统 300 的最终输出语音信号。

[0061] 在完成步骤 412 之后,解码器 /PLC 系统 300 更新各种状态存储器,并执行有利于为后续丢帧而执行的 PLC 操作的一些处理,如步骤 414 所示。状态存储器包括 PLC 相关的低带 ADPCM 解码器状态存储器、PLC 相关的高带 ADPCM 解码器状态存储器和全带 PLC 相关的状态存储器。作为该步骤的一部分,全带语音信号合成器 350 将 QMF 合成滤波器组 340 的输出信号存储在内部信号缓存中,以为后续丢失帧处理过程中可能的语音波形外插做准备。子带 ADPCM 解码器状态更新模块 360 和解码约束和控制模块 370 在处理类型 1 的帧的过程中是非活跃的。以下将参考 D 节中描述的解码器 /PLC 系统 300 的具体实现提供了有关类型 1 的帧处理的更多细节。

[0062] 2、处理类型 2、类型 3 和类型 4 的帧

[0063] 在处理类型 2、类型 3 和类型 4 的帧的过程中,与丢失帧相关的输入比特流是不可用的。因此,模块 310、320、330 和 340 不能执行它们通常的功能,并且是非活跃的。相反,切换器 336 被连接到标记为“类型 -6”的下部位置,全带语音信号合成器 350 变为活跃,合成解码器 /PLC 系统 300 的输出语音信号。全带语音信号合成器 350 通过插入之前存储的与丢包前最后几个接收帧相关的输出语音信号来合成解码器 /PLC 系统 300 的输出语音信号。这在流程图 400 的步骤 416 中体现出来。

[0064] 在全带语音信号合成器 350 完成波形合成的任务后,子带 ADPCM 解码器状态更新模块 360 适当地更新低带 ADPCM 解码器 320 和高带 ADPCM 解码器 330 的内部状态,为下一帧中可能存在的好帧做准备,如步骤 418 所示。现在对执行步骤 416 和 418 的方式进行更多的细节描述。

[0065] a、波形外插

[0066] 执行步骤 416 的波形外插功能存在许多现有技术。以下在 D 节中描述的解码器 /

PLC 系统 300 的实现所使用的技术是在申请号为 11/234, 291、申请人为陈、提交日期为 2005 年 9 月 26 日、标题为“用于块独立语音编解码器的丢包隐藏技术”的美国专利申请中描述的技术的修改版本。在此将提供该技术的高级描述,而在 D 节中将提出更多的细节。

[0067] 为了实现波形外插功能,全带语音信号合成器 350 在接收帧的处理过程中分析来自 QMF 合成滤波器组 340 的存储的输出语音信号以提取基音周期 (pitchperiod)、短期预测因子和长期预测因子。然后将这些参数存储起来以便后续使用。

[0068] 全带语音信号合成器 350 通过执行两阶段搜索来提取基音周期。在第一阶段,通过对输入语音信号或其滤波版本的抽样版本 (decimated version) 执行搜索来确定低解析度的基音周期 (或粗基音)。在第二阶段,通过使用未抽样信号搜索粗基音的邻域来将粗基音提炼成正常解析度。这种两阶段搜索方法明显比在未抽样区间中的单阶段完整搜索需要明显较低的计算复杂度。在对语音信号或其滤波版本进行抽样之前,通常未抽样信号需要通过防混叠 (anti-aliasing) 低通滤波器。为了减少复杂度,普通的现有技术是使用低阶无限脉冲响应 (IIR) 滤波器,如椭圆滤波器。然而,好的低阶 IIR 滤波器的极点通常很靠近单位圆,因而当执行 16 位定点算法中与滤波器的全极点部分相对应的滤波运算时,需要双重精度的算法运算。

[0069] 与现有技术相比,全带语音信号合成器 350 使用有限脉冲响应 (FIR) 滤波器作为防混叠低通滤波器。通过以这种方式使用 FIR 滤波器,仅需要单精度的 16 位定点算法运算,FIR 滤波器能以较低的抽样信号的采样率进行运算。所以,这种方法可以显著地减少防混叠低通滤波器的计算复杂度。例如,在 D 节中描述的解码器 /PLC 系统 300 的实现中,未抽样信号具有 16kHz 的采样率,但是用于基音提取的抽样信号仅具有 2kHz 的采样率。在现有技术的基础上,可以使用 4 阶椭圆滤波器。椭圆滤波器的全极点部分需要双重精度的定点算法,并且需要以 16kHz 采样率运算。正因如此,尽管全零部分能以 2kHz 采样率运算,整个 4 阶椭圆滤波器和下采样操作运算需要 0.66WMOPS (加权百万运算每秒) 的计算复杂度。相反,即便使用相对的高阶 FIR 滤波器来代替 4 阶椭圆滤波器,由于 60 阶 FIR 滤波器是以非常低的 2kHz 采样率操作,所以整个 60 阶 FIR 滤波器和下采样操作仅需要 0.18WMOPS 的复杂度,相比 4 阶椭圆滤波器减少了 73%。

[0070] 在丢包的第一个丢失帧开始点,当到级联合成滤波器的输入设为零的时候,全带语音信号合成器 350 使用级联长期合成滤波器和短期合成滤波器来产生一个信号,称之为“呼叫信号 (ringing signal)”。然后全带语音信号合成器 350 分析某些信号参数 (如基音预测增益和标准化自相关) 来确定存储的输出语音信号中“发音 (voicing)”的度。如果之前的输出语音信号发音很高,那么以周期性方式外插该语音信号以产生当前坏帧的替代波形。周期性波形外插是使用在最近接收的帧上提取的基音周期的提炼版来执行的。如果之前的输出语音信号是不发声的或类似噪声的,那么成比例的 (scaled) 随机噪声通过短期合成滤波器来产生当前坏帧的替代信号。如果发音的度是在两个极端之间,那么将两个成分按发音度成比例的混合在一起。然后将该外插信号与呼叫信号叠加,以确保在丢包的第一个坏帧开始时不会有波形中断。此外,将波形外插扩展到超出当前坏帧末端一段至少等于叠加周期的时间,使得下一帧开始时该外插信号的额外采样可以用作下一帧开始时叠加的呼叫信号。

[0071] 在不是丢包的第一个坏帧的坏帧中 (即在类型 3 或类型 4 的帧中),全带语音信号

合成器 350 的操作本质上与上一段描述的操作是相同的,除了全带语音信号合成器 350 不需要计算呼叫信号,并且可以使用在超出上一帧末端后在上一帧内计算出的外插信号的额外采样作为叠加运算的呼叫信号,以此确保在该帧开始时没有波形中断。

[0072] 对于长期丢包的情况,全带语音信号合成器 350 逐渐将解码器 /PLC 系统 300 的输出语音信号减弱。例如,在 D 节中描述的解码器 /PLC 系统的实现中,丢包过程中产生的输出语音信号以线性方式衰减或“减弱”到零,从 20ms 开始,并在 60ms 结束。执行该功能是因为关于“实际”波形的形状和形式的不确定性随时间增加。实际上,当外插片段远超出约 60ms 的范围时,许多 PLC 方案开始产生嗡嗡响 (buzzy) 的输出。

[0073] 在本发明的替代实施例中,对于背景噪声中的 PLC, (通常) 本发明的实施例追踪背景噪声 (周围噪声) 的级别,并针对长时间的帧删除衰减到该级别,而不是零。这消除了因 PLC 系统对输出的消声而在背景噪声中产生的丢包中断效果。

[0074] 本发明的进一步替代实施例通过执行实现舒适噪声产生 (CNG) 功能解决了背景噪声中 PLC 的前述问题。当本发明的该实施例开始针对长期丢包衰减解码器 /PLC 系统 300 的输出语音信号时,其也开始混入由 CNG 产生的舒适噪声。通过混入舒适噪声和以舒适噪声替换,当解码器 /PLC 系统 300 的输出语音信号被衰弱并最终消声时,上述的中断效果将被消除并且提供出了该信号周围环境的可靠再现。该方法已经被证明并且普遍地在其它应用中被接受。例如,在子带回声消除器 (SBAEC) 中,或者一般的回声消除器 (AEC) 中,当检测到残余回声时,减弱该信号并且以舒适噪声替换。这通常是指非线性处理 (NLP)。本发明的该实施例的前提在于,PLC 呈现出非常类似的方案。与 AEC 相似,这种方法针对 PLC 的使用将提供一种更加增强的体验,这远没有中断效果令人讨厌。

[0075] b、低带和高带 ADPCM 解码器的内部状态的更新

[0076] 全带语音信号合成器 350 完成了在步骤 416 中执行的波形合成任务之后,子带 ADPCM 解码器状态更新模块 360 然后在步骤 418 中适当地更新低带 ADPCM 解码器 320 和高带 ADPCM 解码器 330 的内部状态,为下一帧中可能的好帧做准备。执行低带 ADPCM 解码器 320 和高带 ADPCM 解码器 330 的内部状态更新存在许多方法。由于图 1 中的 G. 722 编码器和图 2 中的 G. 722 解码器具有相同类型的内部状态,更新解码器 320 和 330 的内部状态的一种直接的方法是反馈全带语音信号合成器 350 的输出信号通过图 1 所示的标准 G. 722 编码器,以上一帧的最后一个采样留下的内部状态开始。然后,在对外插语音信号的当前坏帧编码后,当前坏帧的最后一个采样留下的内部状态被用来更新低带 ADPCM 解码器 320 和高带 ADPCM 解码器 330 的内部状态。

[0077] 然而,前述方法具有两个子带编码器的复杂度。为了节省复杂度,在 D 节中描述的解码器 /PLC 系统 300 对以上方法进行了近似实现。对于高带 ADPCM 编码器,公认的是在处理丢包后的第一个接收帧的时候,并不需要高带自适应量化步长 $\Delta_H(n)$ 。相反,量化步长被复位为丢包前的滑动平均值 (如在本申请中其它部分描述的)。因此,使用非量化的差分信号 (或预测误差信号) $e_H(n)$ 进行高带 ADPCM 编码器中的自适应预测更新,并且对 $e_H(n)$ 的量化操作被完全避免了。

[0078] 对于低带 ADPCM 编码器,方案稍微有点不同。由于维持低带自适应量化步长 $\Delta_L(n)$ 的基音调制的重要性,以下在 D 节中描述的解码器 /PLC 系统 300 的实现有效地在丢失帧中更新了这个参数。标准 G. 722 低带 ADPCM 编码器采用差分信号 (或预测误差信号) $e_L(n)$ 的

6 位量化。然而,根据 G. 722 标准,仅使用了 8 个幅度量化指数的子集来更新低带自适应量化步长 $\Delta_L(n)$ 。通过使用非量化差分信号 $e_L(n)$ 代替量化差分信号用于低带 ADPCM 编码器中的自适应预测更新,在维持低带自适应量化步长 $\Delta_L(n)$ 的同样的更新的情况下,在 D 节中描述的实施例能够使用不是很复杂的差分信号量化。

[0079] 所属领域的技术人员应容易知悉,在本申请中涉及高带自适应量化步长 $\Delta_H(n)$ 的描述中,高带自适应量化步长可由高带对数比例因子 $\nabla_H(n)$ 替换。同样地,在本申请中涉及低带自适应量化步长 $\Delta_L(n)$ 的描述中,低带自适应量化步长可由低带对数比例因子 $\nabla_L(n)$ 替换。

[0080] 与标准 G. 722 子带 ADPCM 编码器比较,在 D 节的实施例中所用的低带和高带 ADPCM 编码器的另一区别是编码器的基于信号属性和丢包持续时间的自适应复位。现在开始描述该功能。

[0081] 如以上介绍的,对于长时间的丢包,全带语音信号合成器 350 在预定时间后将输出语音波形消声。在以下 D 节中描述的解码器 /PLC 系统 300 的实现中,来自全带语音信号合成器 350 的输出信号馈通给 G. 722 QMF 分析滤波器组,以获得用于在丢失帧期间更新低带 ADPCM 解码器 320 和高带 ADPCM 解码器 330 内部状态的子带信号。因此,一旦来自全带语音信号合成器 350 的输出信号被消减到零,用于更新子带 ADPCM 解码器内部状态的子带信号也会变成零。恒零能使每个解码器中的自适应预测器与编码器中的自适应预测器区分开,因为恒零会不自然地使预测器部分在相同方向不停地适应性调整。这在传统的高带 ADPCM 解码器中是显而易见的,这在长时间丢包后处理好帧时通常会产生产生高频啁啾声(chirping)。对于传统的低带 ADPCM 解码器,由于预测器具有太高的滤波器增益,这个问题偶尔会引起不自然的能量增长。

[0082] 基于前述的论述,一旦 PLC 输出波形已经被消减到零,以下在 D 节中描述的解码器 /PLC 系统 300 的实现复位 ADPCM 子带解码器。该方法在长时间帧删除后几乎完全消除了高频啁啾声。全带语音信号合成器 350 产生的合成波形的不确定性随丢包时间的增加而增加,这种论述结果表明,在某个点上,使用该方法来更新子带 ADPCM 解码器 320 和 330 是不明显的。

[0083] 然而,即使在全带语音信号合成器 350 的输出被完全消声时复位子带 ADPCM 解码器 320 和 330,仍然存在以不常见的啁啾声(来自高带 ADPCM 解码器 330)和不常见且不自然能量增长(来自低带 ADPCM 解码器 320)的形式出现的一些问题。通过产生各子带 ADPCM 解码器自适应的复位深度,在 D 节中描述的实现中已经解决了这些问题。复位在波形消声时仍会发生,但是也可以提前复位子带 ADPCM 解码器 320 和 330 的一个或多个。

[0084] 如将在 D 节中描述的,提前复位的决定是基于在坏帧期间(即在基于来自全带语音信号合成器的输出信号更新子带 ADPCM 解码器 320 和 330 的期间)对控制子带 ADPCM 解码器 320 和 330 的自适应预测器的极点部分的自适应的信号某些性能的监控做出的。对于低带 ADPCM 解码器 320,部分重构的信号 $P_L(n)$ 驱动全极点滤波部分的自适应,而部分重构的信号 $P_H(n)$ 驱动高带 ADPCM 解码器 330 的全极点滤波部分的自适应。本质上,在 10ms 的丢失帧期间内监控到每个参数都很大程度上是恒定的,或者在当前丢失帧期间内主要为正或负。应注意,在 D 节描述的实现中,自适应复位被限制在包丢失 30ms 之后。

[0085] 3、处理类型 5 和类型 6 的帧

[0086] 在处理类型 5 和类型 6 的帧时,与当前帧相关的输入比特流再一次可用,因而模块 310、320、330 和 340 再次处于活跃状态。然而,低带 ADPCM 解码器 320 和高带 ADPCM 解码器 330 执行的解码操作被解码约束和控制模块 370 所约束和控制,以减少从丢失帧到接收帧的过渡处的赝像 (artifact) 和失真,从而提高丢包后解码器 /PLC 系统 300 的性能。对于类型 5 的帧,这体现在流程图 400 的步骤 420 中,对于类型 6 的帧则体现在步骤 426 中。

[0087] 对于类型 5 的帧,将对输出语音信号执行附加修改来确保由全带语音信号合成器 350 产生的合成信号和由 QMF 合成滤波器组 340 产生的输出信号之间的平滑过渡。因而,QMF 合成滤波器组 340 的输出信号并不直接用作解码器 /PLC 系统 300 的输出语音信号。相反,全带语音合成器 350 修改 QMF 合成滤波器组 340 的输出,并使用修改后的版本作为解码器 /PLC 系统 300 的输出语音信号。因而,在处理类型 5 或类型 6 的帧时,切换器 336 保持连接到标记为“类型 2-6”的下部位置,以接收来自全带语音信号合成器 350 的输出语音信号。

[0088] 在这点上,如果全带语音信号合成器 350 产生的合成信号和 QMF 合成滤波器组 340 产生的输出信号之间存在未对齐,全带语音信号合成器 350 执行的操作包括时间扭曲和重新定相。这些操作的执行在流程 400 的步骤 422 示出,并将在以下做更多的细节描述。

[0089] 同样,对于类型 5 的帧,全带语音信号合成器 350 产生的输出语音信号与来自先前处理的丢失帧的呼叫信号叠加。这样做是为了确保从与前帧关联的合成波形到与当前类型 5 的帧关联的输出波形的平滑过渡。该步骤的执行在流程图 400 的步骤 424 示出。

[0090] 在已经为类型 5 或类型 6 的帧产生输出语音信号之后,解码器 /PLC 系统 300 更新各种状态存储器,并执行一些处理来利于针对后续丢失帧以步骤 414 的类似方式所执行的 PLC 运算,如步骤 428 所示。

[0091] a、子带 ADPCM 解码的约束和控制

[0092] 如以上介绍的,解码约束和控制模块 370 对在处理类型 5 和类型 6 的帧过程中由低带 ADPCM 解码器 320 和高带 ADPCM 解码器 330 执行的解码操作进行约束和控制,以提高丢包后解码器 /PLC 系统 300 的性能。现在对解码约束和控制模块 370 采用的各种约束和控制进行描述。有关这些约束和控制的更多细节将在以下 D 节中参考解码器 /PLC 系统 300 的特殊实现作进一步描述。

[0093] i、用于高带 ADPCM 解码器的自适应量化步长的设置

[0094] 对于类型 5 的帧,解码约束和控制模块 370 将用于高带 ADPCM 解码器 330 的自适应量化步长 $\Delta_H(n)$ 设定为与丢包前接收的好帧关联的值的滑动平均值。通过减少背景噪声的片段中可见的丢包导致的能量下降,提高了背景噪声中解码器 /PLC 系统 300 的性能。

[0095] ii、用于低带 ADPCM 解码器的自适应量化步长的设置

[0096] 对于类型 5 的帧,解码约束和控制模块 370 执行自适应策略以为低带 ADPCM 解码器 320 设置自适应量化步长 $\Delta_L(n)$ 。在替代的实施例中,该方法也可用于高带 ADPCM 解码器 330。如在前面部分介绍的,对于高带 ADPCM 解码器 330,将自适应量化步长 $\Delta_H(n)$ 设定为丢包前第一个好帧上的值的滑动平均值,对解码器 /PLC 系统 300 在背景噪声内的性能是有益的。然而,相同方法应用到低带 ADPCM 解码器 320 偶尔会在有声语音上产生很大的不自然能量增长。这是因为在有声语音中使用基音周期对 $\Delta_L(n)$ 进行调制,并因此将 $\Delta_L(n)$ 设定为丢包前的滑动平均值会在丢包后的第一个好帧上引起 $\Delta_L(n)$ 非常大的反常增加。

[0097] 因此,在由基音周期调制 $\Delta_L(n)$ 的情况下,优选地是使用来自 ADPCM 解码器状态更新模块 360 的 $\Delta_L(n)$,而不是丢包前的 $\Delta_L(n)$ 的滑动平均值。回顾一下,通过将全带语音信号合成器 350 的输出信号传递通过 G. 722 QMF 分析滤波器组来获得低带信号,子带 ADPCM 解码器状态更新模块 360 更新低带 ADPCM 解码器 320。如果全带语音信号合成器 350 正在执行任务,可能是针对有声语音的,那么用于更新低带 ADPCM 解码器 320 的信号很可能非常匹配在编码器上使用的信号,因此参数 $\Delta_L(n)$ 也很可能非常接近该编码器的步长。对于有声语音,该方法优选地是将 $\Delta_L(n)$ 设定为丢包前的 $\Delta_L(n)$ 的滑动平均值。

[0098] 考虑到之前描述的,解码约束和控制模块 370 采用自适应策略以设定用于丢包后第一个好帧的 $\Delta_L(n)$ 。如果丢包前的语音信号相当稳定的,例如稳定的背景噪声,那么将 $\Delta_L(n)$ 设定为丢包前的 $\Delta_L(n)$ 的滑动平均值。然而,如果丢包前的语音信号呈现出 $\Delta_L(n)$ 上的变化,例如被认为是针对有声语音的,那么将 $\Delta_L(n)$ 设定为通过基于全带语音信号合成器 350 的输出的低带 ADPCM 解码器更新而获得的值。对于中间的情况, $\Delta_L(n)$ 被设定为基于丢包前 $\Delta_L(n)$ 变化在这两个值之间进行的线性加权。

[0099] iii、用于高带 ADPCM 解码器的自适应量化步长的自适应低通滤波

[0100] 在处理包丢失后最初几个好帧(类型 5 和类型 6 的帧)的过程中,为了减少产生太强的高频内容的局部波动(由于 G. 722 编码器和 G. 722 解码器之间同步的暂时丢失)的风险,解码约束和控制模块 370 会有效控制高带 ADPCM 解码器的自适应量化步长 $\Delta_H(n)$ 。这样会产生高频振荡效应,这正好是实际啁啾声的影响。因此,在最初几个好帧内,对高带量化步长 $\Delta_H(n)$ 应用自适应低通滤波器。经过自适应的时间周期以二次形式减少了平滑过渡。对于丢包前语音信号非常稳定的信号片段,持续时间更长(以下在 D 节中描述的解码器/PLC 系统 300 的实现中为 80ms)。对于丢包前不是很稳定的语音信号的情况,持续时间稍短(以下在 D 节中描述的解码器/PLC 系统 300 的实现中为 40ms),而对于不稳定的片段,则不采用低通滤波。

[0101] iv、最初几个好帧内全极点滤波器部分上的自适应安全裕度(adaptivesafety margin)

[0102] 由于在包丢失期间和之后 G. 722 解码器和编码器之间必然的偏差,解码约束和控制模块 370 在丢包后最初几个好帧(类型 5 和类型 6 的帧)期间对低带 ADPCM 解码器 720 的自适应预测器执行某些约束。根据 G. 722 标准,默认的编码器和解码器在子带预测器的极点部分上执行 1/16 的最小“安全”裕度。然而,已经发现的是,低带 ADPCM 解码器的两极六个零预测滤波器的全极点部分在丢包后经常会引起反常的能量增加。这通常以波形弹跳(pop)的形式感测到。显然,丢包引起较低的安全裕度,其与产生极高能量波形的具有较高增益的全极点滤波部分对应。

[0103] 通过自适应地在低带 ADPCM 解码器 320 的自适应预测器的全极点滤波部分上执行更多的严格约束,解码约束和控制模块 370 极大地减少了这种丢包后的反常能量增加。在包丢失后最初几个好帧上获得了增加的最小安全裕度。增加的最小安全裕度被逐渐降低至标准的 G. 722 最小安全裕度。此外,还对丢包前的安全裕度的滑动平均值进行监控,并对包丢失后最初几个好帧期间增加的最小安全裕度进行控制,以致不会超过该滑动平均值。

[0104] v、高带 ADPCM 解码器的内部信号上的 DC 移除

[0105] 在包丢失后最初几个好帧(类型 5 和类型 6 的帧)期间,据观察,G. 722 解码器经

常产生非常令人讨厌的明显的高频啁啾失真。该失真来自于因丢包而与高带 ADPCM 编码器失去同步并因而产生有偏差的预测的高带 ADPCM 解码器。导致啁啾失真的同步丢失表明了其自身在输入信号中对极点预测器 $P_H(n)$ 的自适应的控制和对在长时间内具有恒定正负号的重构高带信号 $r_H(n)$ 的控制。这使得预测器的极点部分发生漂移,因为自适应是基于符号的 (sign-based),因而在相同方向上保持更新。

[0106] 为了避免这个问题,通过在包丢失后最初几个好帧期间分别以高通滤波版本 $P_{H,HP}(n)$ 和 $r_{H,HP}(n)$ 来代替信号 $P_H(n)$ 和 $r_H(n)$,解码约束和控制模块 370 对这些信号增加了 DC 移除操作。这用来完全消除啁啾声。DC 移除被实现为 $P_H(n)$ 和 $r_H(n)$ 各自的滑动平均值的减法。这些滑动平均值针对好帧和坏帧持续地更新。在以下 D 节中描述的解码器 /PLC 系统 300 的实现中,这种替换针对包丢失后最初 40ms 发生。

[0107] b、重新定相和时间扭曲

[0108] 如以上介绍的,在流程图 400 的步骤 422 中,如果全带语音信号合成器 350 在丢包期间产生的合成语音信号和 QMF 合成滤波器组 340 在丢包后最初接收的帧期间产生的语音信号之间存在未对齐,全带语音信号合成器 350 执行 被称为“重新定相”和“时间扭曲”的技术。

[0109] 如上所述的,在处理丢失帧时,如果与丢包前的接收帧相关的解码语音信号几乎是周期性的,如语音元音信号,那么全带语音信号合成器 350 基于基音周期外插语音波形。仍如上所述的,在丢失帧结尾以外继续这种波形外插,以获得用于与下一帧的相关语音信号叠加的更多采样,从而确保了平滑过渡以及避免了任何中断。然而,解码语音信号的实际基音周期一般不跟随在丢失帧中进行波形外插期间所用的基音轨迹。所以,一般外插语音信号不会与第一个好帧关联的解码语音信号完全对齐。

[0110] 这在图 6 中示出,图 6 是示出了丢包前和丢包后第一个接收帧期间解码语音信号 602 的振幅(为了方便,还示出了丢失帧时的解码语音信号,但是应了解,解码器 /PLC 系统 300 不能解码原始信号的这个部分)以及在丢失帧期间和丢包后第一个接收帧期间产生的外插语音信号 604 的振幅的时间线 600。如图 6 所示,两个信号在第一个接收帧中是异相的。

[0111] 这种异相现象在解码器 /PLC 系统 300 中引起两个问题。第一,从图 6 中可以看到,在丢包后的第一个接收帧中,叠加区域中的解码语音信号 602 和外插语音信号 604 是异相的,并将部分对消,引起可听膺像。第二,与子带 ADPCM 解码器 320 和 330 相关的状态存储器呈现了某种程度的基音调制,并因此对语音信号的相位敏感。如果语音信号接近基音周期,即是靠近信号电平急剧上升和下降的基音脉冲的语音信号部分,这一问题尤其明显。因为子带 ADPCM 解码器 320 和 330 对语音信号的相位敏感,并因为外插语音信号 604 被用来在包丢失时更新这些解码器的状态存储器(如上所述),外插语音信号 604 和解码语音信号 602 之间的相位差将在丢包后的接收帧中产生明显的膺像,这是因为子带 ADPCM 编码器和解码器的内部状态失配。

[0112] 以下将对此作更多的细节描述,时间扭曲用于解决叠加区域中的相消干扰的第一个问题。具体地说,时间扭曲用于拉伸和收缩与丢包后第一个接收帧相关的解码语音信号的时间轴,以将它与用于隐藏前一丢失帧的外插语音信号对齐。虽然在此参考具有存储器的子带预测编码器对时间扭曲进行描述,但是这项普通技术也可应用到其它编码器,包括

但不限于具有和不具有存储器的编码器、预测和非预测编码器以及子带和全带编码器。

[0113] 同样在此作更多的细节描述,重新定相用于解决由于丢失帧和包丢失后第一帧的未对齐引起的子带 ADPCM 编码器和解码器的内部状态失陪的第二个问题。重新定相是将子带 ADPCM 解码器 320 和 330 的内部状态设定为外插语音波形与包丢失后第一个接收帧前的最后一个输入信号采样同相时的时间点的状态的处理过程。虽然在后向自适应系统的环境中描述了重新定相,但是它也可以用于在前向自适应预测编码器或者具有存储器的任何编码器中执行 PLC。

[0114] i、时滞计算

[0115] 重新定相和时间扭曲技术都需要计算外插语音信号和与包丢失后第一个接收帧相关的解码语音信号之间未对齐的采样的数量。这一未对齐被称为“滞后”,如图 6 中所标记的,其可认为是解码语音信号滞后外插语音信号的采样数量。在图 6 的情形中,滞后是负的。

[0116] 实行时滞计算的一种一般的方法在图 7 的流程图 700 中示出,但是也可以使用其它的方法。执行该方法的一种特定方式在以下 D 节中给出了描述。

[0117] 如图 7 所示,流程图 700 的方法在步骤 702 开始,在前一丢失帧期间由全带语音信号合成器 350 产生的语音波形被外插到丢包后第一个接收帧中。

[0118] 在步骤 704,计算时滞。在概念级上,通过最大化外插语音信号和与丢包后第一个接收帧关联的解码语音信号之间的相关性来计算时滞。如图 9 所示,相对于与第一接收帧(表示为 902)关联的解码语音信号,外插语音信号(表示为 904)在 $-MAXOS$ 到 $+MAXOS$ 的范围内漂移,其中 $MAXOS$ 表示最大偏移量,最大化相关性的漂移量被用作时滞。这可以通过在零附近的 $\pm MAXOS$ 的时滞范围内的信号间搜索标准化交叉相关性函数 $R(k)$ 的峰值来完成:

$$[0119] \quad R(k) = \frac{\sum_{i=0}^{LSW-1} es(i-k) \cdot x(i)}{\sqrt{\sum_{i=0}^{LSW-1} es^2(i-k) \sum_{i=0}^{LSW-1} x^2(i)}}, \quad k = -MAXOS, K, MAXOS \quad (1)$$

[0120] 其中 es 是外插语音信号, x 是与包丢失后第一个接收帧关联的解码语音信号, $MAXOS$ 是允许的最大偏移, LSW 是滞后搜索窗口长度, $i = 0$ 表示在滞后搜索窗口中的第一采样。最大化该函数的时滞将对应于两个波形之间的相对时间偏差。

[0121] 在一个实施例中,以基于基音周期自适应的方式确定出在其上计算相关性的采样的数量(又称为滞后搜索窗口)。例如,在以下 D 节中描述的实施例中,用于粗滞后搜索的采样(16kHz 采样率)数量上的窗口大小如下:

[0122]

$$LSW = \begin{cases} 80 & \lfloor ppfe \cdot 1.5 + 0.5 \rfloor < 80 \\ 160 & \lfloor ppfe \cdot 1.5 + 0.5 \rfloor > 160, \\ \lfloor ppfe \cdot 1.5 + 0.5 \rfloor & \text{否则} \end{cases} \quad (2)$$

[0123] 其中 $ppfe$ 是基音周期。该等式使用了地板函数(floor function)。实数 x 的地板函数 $\lfloor x \rfloor$ 是返回小于或等于 x 的最大整数的函数。

[0124] 如果在步骤 704 中计算的时滞是零,那么这表示外插语音信号和第一个接收帧关

联的解码语音信号是同相的,而正值表示与第一个接收帧关联的解码语音信号滞后于(相比较而言被延迟)外插语音信号,负值表示与第一个接收帧关联的解码语音信号超前于外插语音信号。如果时滞等于零,那么不需要执行重新定相和时间扭曲。在以下 D 节中提出的示例实现中,如果包丢失前的最后一个接收帧是无声的(通过针对该帧计算出的发声的度数来表示,如以上关于类型 2、类型 3 和类型 4 的处理所描述的),或者如果包丢失后的第一个接收帧是无声的,时滞也被设为零。

[0125] 为了最小化相关性计算的复杂度,可以使用多阶段处理来执行滞后搜索。图 8 的流程图 800 示出了这种方法,其中首先在步骤 802 使用信号的下采样表示来执行粗时滞搜索,然后在步骤 804 使用信号的较高采样率表示来执行细化时滞搜索。例如,将信号下采样到 4kHz 后执行粗时滞搜索,以 8kHz 上的信号执行细化时滞搜索。为了进一步减小复杂度,可以仅通过对信号进行二次采样而忽略任何混叠效应来执行向下采样。

[0126] 一个问题在于使用什么信号来与第一个接收帧内的外插语音信号进行相关。“强力(brute force)”方法是完整地解码第一个接收帧以获得解码语音信号然后在 16kHz 上计算相关性的方法。为解码第一个接收帧,可以使用从重新编码外插语音信号(如上所述)一直到帧边界而获得的子带 ADPCM 解码器 320 和 330 的内部状态。然而,由于以下所述的重新定相算法将为子带 ADPCM 解码器 320 和 330 提供一组的更佳状态,这需要重新运行 G. 722 解码。因为该方法执行两次完整的解码操作,所以在计算复杂度方面是非常浪费的。为了解决此问题,本发明的实施例实现了一种较低复杂度的方法。

[0127] 根据较低复杂度的方法,在第一个接收帧中接收的 G. 722 比特流仅被部分解码以获得低带量化差分信号 $d_{lt}(n)$ 。在正常 G. 722 解码过程中,从比特流解复用器 310 接收的比特由子带 ADPCM 解码器 320 和 330 转换成差分信号 $d_{lt}(n)$ 和 $d_{ht}(n)$,这两个信号由向后自适应比例因子进行缩放,并通过向后自适应零极点(pole-zero)预测器来获得子带语音信号,这些信号然后由 QMF 合成滤波器组 340 合成来产生输出语音信号。在该处理过程内的每个采样上,将更新子带 ADPCM 解码器 320 和 330 中的自适应预测器的系数(coefficients)。这个更新了解码器复杂度的重要部分。由于仅需要用于时滞计算的信号,那么在较低复杂度方法中,两极、六零预测滤波器系数仍保持不变(它们并不基于逐个采样被更新)。另外,因为滞后是由基音决定的,而人的语音的基音基本频率小于 4kHz,所以仅能获得低带近似信号 $r_l(n)$ 。关于此方法的更多细节将在以下 D 节中提供。

[0128] 在以下 D 节中描述的实施例中,两极、六零预测滤波器的固定滤波系数是从重新解码丢包期间的外插波形一直到最后一个丢失帧的末端来获得。在可选的实现中,固定滤波系数可以是在包丢失前的最后一个接收帧的结尾使用的那些滤波器系数。在另一可选的实现中,可以根据语音信号的特性或者其它标准,以自适应方式来选择这些系数组中的一个或者其它。

[0129] ii、重新定相

[0130] 在重新定相过程中,调整子带 ADPCM 解码器 320 和 330 的内部状态来考虑外插语音波形和包丢失后的第一个接收帧关联的解码语音波形之间的时滞。如之前所述,在处理第一个接收帧之前,通过对在前一丢失帧期间由全带语音信号合成器 350 合成的输出语音信号进行重新编码来估计子带 ADPCM 解码器 320 和 330 的内部状态。这些解码器的内部状态呈现出某种基音调制。因而,如果在与前一丢失帧关联的波形外插期间使用的基音周期

正好紧跟解码语音信号的基音轨迹,那么可在最后一个丢失帧和第一个接收帧之间的边界停止重编码处理,而子带 ADPCM 解码器 320 和 330 的状态与原始信号是同相的。然而,如上所述,在外插时使用的基音一般不匹配解码语音信号的基音轨迹,且在包丢失后的第一个接收帧开始时,外插语音信号和解码语音信号是不对齐的。

[0131] 为了克服此问题,重新定相使用时滞来控制何处停止重新定相处理。在图 6 的例子中,外插语音信号 604 和解码语音信号 602 之间的时滞是负的。假设该时滞由 lag 表示。那么,可以看出,如果对外插语音信号重编码超出帧边界的 $-lag$ 个采样,那么重编码将在外插语音信号 604 内的与帧边界上解码语音信号 602 的相位一致的相位上停止。产生的子带 ADPCM 解码器 320 和 330 的状态存储器将与第一个好帧中的接收数据同相,从而提供了更好的解码信号。因而,重新解码子带重构信号的采样数量如下:

$$[0132] \quad N = FS - lag \quad (3)$$

[0133] 其中 FS 是帧大小,所有参数都是以子带采样率 (8kHz) 为单位。

[0134] 图 10A、图 10B 和图 10C 分别给出了三种重新定相的方案。在图 10A 的时间线 1000 上,解码语音信号 1002 超前于外插语音信号 1004,所以重新解码超出帧边界 $-lag$ 个采样。在图 10B 的时间线 1010 上,解码语音信号 102 滞后于外插语音信号 1014,重新解码在帧边界前 lag 个采样处中止。在图 10C 的时间线 1020 上,外插语音信号 1024 和解码语音信号 1022 在帧边界上同相(尽管丢失帧期间的基音轨迹是不同的),且重新编码在帧边界上停止。注意,为了方便,在图 10A、10B 和 10C 中,都示出了在丢失帧期间的解码语音信号,但是应清楚解码器 300 并不能解码原始信号的这个部分。

[0135] 如果没有执行子带 ADPCM 解码器 320 和 330 内部状态的重新定相,那么可以在丢失帧的整个处理过程中执行用于更新这些内部状态的重新编码。然而,因为一直要到包丢失后的第一个接收帧才知道滞后,所以并不能在丢失帧的整个过程中执行重新编码。解决此问题的简单方法是存储用于替换前一丢失帧的整个外插波形,然后在第一个接收帧期间执行重新编码。然而,这需要存储器来存储 $FS + MAXOS$ 个采样。重新编码的复杂度也全部落入第一个接收帧。

[0136] 图 11 是以重新分配对在前丢失帧的大量计算的方式执行重新编码的方法流程图 110。因为 $MAXOS \ll FS$,所以从计算的负载平衡角度来看,这是合理和可行的。

[0137] 如图 11 所示,流程图 1100 的方法开始于步骤 1102,在丢失帧内执行重新编码一直到帧边界,然后存储帧边界上子带 ADPCM 解码器 320 和 330 的内部状态。另外,还要存储重新编码 $FS - MAXOS$ 个采样后的中间内部状态,如步骤 1104 所示。在步骤 1106,在存储器中保存用于重新编码 $FS - MAXOS + 1$ 到 $FS + MAXOS$ 所生成的波形外插采样。在步骤 1108,在包丢失后的第一个接收帧内,使用帧边界上存储的内部状态作为初始状态以执行子带近似解码(用于确定上述的 lag)。然后,在判决步骤 1110,确定 lag 是正的还是负的。如果 lag 是正的,那么重新存储在 $FS - MAXOS$ 采样上的内部状态,并开始重新解码 $MAXOS - lag$ 采样,如步骤 1112 所示。相反,如果 lag 是负的,那么使用帧边界上的内部状态,并重新编码附加的 $|lag|$ 采样。根据本方法,在第一个接收帧中重新编码至多 $MAXOS$ 采样。

[0138] 所属领域的技术人员应知悉,可以通过在丢失帧中的重新编码过程时沿途存储更多的 G. 722 状态来减少第一个好帧中的重新编码的量。在极端情况下,可以存储 $FRAME SIZE - MAXOS$ 和 $FRAME SIZE + MAXOS$ 之间每个采样的 G. 722 状态,且并不需要在第一个接

收帧中进行重新编码。

[0139] 相比流程图 1100 的方法,一种在第一个接收帧中需要更多的重新编码的替代方法中,重新编码在丢失帧期间针对 FS-MAXOS 个采样而执行。子带 ADPCM 解码器 320 和 330 的内部状态和剩余 $2 * \text{MAXOS}$ 个采样被保存在存储器中以便在第一个接收帧中使用。在第一个接收帧中,计算出滞后,并基于该滞后对合适量的采样从存储的 G. 722 状态开始执行重新编码。该方法需要存储 $2 * \text{MAXOS}$ 个重构的采样、G. 722 状态的一份副本以及第一个接收帧中的至多 $2 * \text{MAXOS}$ 个采样的重新编码。该替代方法的缺点是不能存储用于上述较低复杂度解码和时滞计算的帧边界上的子带 ADPCM 解码器 320 和 330 的内部状态。

[0140] 理想地说,滞后应该与外插语音信号和第一个接收帧关联的解码语音信号 之间的帧边界上的相移一致。根据本发明的一个实施例,通过相对长的滞后搜索窗口来计算出粗滞后估计值,该窗口的中心并不与帧边界一致。例如,滞后搜索窗口可以是基音周期的 1.5 倍。滞后搜索范围(即相对于原始语音信号偏移外插语音信号的采样数量)也相对比较宽(即 ± 28 个采样)。为了改善对齐度,于是执行滞后细化搜索。作为滞后细化搜索的一部分,移动搜索窗口以从第一个接收帧的第一个采样开始。这可以通过由粗滞后估计来偏置外插语音信号来完成。滞后细化搜索中的滞后搜索窗口的大小可以比较小,且滞后搜索范围也可以比较小(即 ± 4 个采样)。搜索方法可以与上述 3. b. i 节中的方法相同。

[0141] 本发明在以上 G. 722 后向自适应预测编译码器的环境中已经提出了重新定相的概念。我们可以很容易的将此概念扩展到其它后向自适应预测编译码器,如 G. 726。然而,重新定相的使用并不局限于后向自适应预测编译码器。相反,多数基于存储器的编码器在状态存储器中表现出相位相关性,并因此受益于重新定相。

[0142] iii、时间扭曲

[0143] 如在此使用的,术语时间扭曲指的是沿时间轴拉伸或收缩信号的处理。如在此别处讨论的,为了维持连续的信号,本发明的实施例将用于代替丢失帧的外插语音信号和包丢失后第一个接收帧关联的解码语音信号合并以避免中断。这是通过执行两个信号之间的叠加来完成的。然而,如果信号间是相互异相的,那么可能发生波形对消(cancellation),并产生可听膺像,例如图 6 中的叠加区域。在此区域中执行叠加将引起解码语音信号 602 的负的部分和外插语音信号 604 之间明显的波形对消。

[0144] 根据本发明的实施例,与包丢失后第一个接收帧关联的解码语音信号被执行时间扭曲,以在第一个接收帧中的某个时间点上使解码语音信号与外插语音信号相位对齐。时间扭曲的量是由时滞的值控制的。因而,在一个实施例中,如果时滞是正的,那么将拉伸与第一个接收帧关联的解码语音信号,并且叠加区域可以设置在第一个接收帧的开始处。然而,如果时滞是负的,解码语音信号将被压缩。因此,叠加区域被设置到进入第一个接收帧 $|\text{lag}|$ 个采样。

[0145] 在 G. 722 的情况下,丢包后第一个接收帧开始的有些采样并不是可靠的,这是因为帧开始时子带 ADPCM 解码器 320 和 330 的内部状态是不正确的。因此,在本发明的实施例中,根据对与第一个接收帧关联的解码语音信号应用的时间扭曲,第一个接收帧内的头 MIN_UNSTBL 个采样可不包含在叠加区域中。例如,在以下 D 节描述的实施例中,将 MIN_UNSTBL 设定为 16,或者是 160 个采样的 10ms 帧内的第一个 1ms。在该区域中,外插语音信号可用作解码器 /PLC 系统 300 的输出语音信号。这种实施例有效地解决了第一个接收帧

中语音信号的重新收敛时间。

[0146] 图 12A、图 12B 和图 12C 示出了这一概念的几个例子。在图 12A 的例子中，时间线 1200 示出了在第一个接收帧中解码语音信号超前于外插信号。因此，解码语音信号经过了 $-lag$ 个采样的时间扭曲收缩（时滞 lag 是负的）。应用了时间扭曲后的结果如时间线 1210 所示。如时间线 1210 所示，这些信号在叠加区的中心或中心附近是同相的。在这种情况下，叠加区域的中心位于 $MIN_UNSTBL-lag+OLA/2$ 处，其中 OLA 是叠加区域中采样的数量。在图 12B 的例子中，时间线 1220 示出了在第一个接收帧中解码语音信号滞后于外插信号。因此，将解码语音信号作 lag 个采样的时间扭曲拉伸来完成对齐。采用时间扭曲的结果如图 1230 所示。在这种情况下， $MIN_UNSTBL > lag$ ，并且在第一个接收帧中仍存在不稳定的区域。在图 12C 的例子中，时间线 1240 示出了解码信号再一次滞后于外插信号，所以解码语音信号被时间扭曲拉伸来提供时间线 1250 中的结果。然而，如时间线 1250 所示，因为 $MIN_UNSTBL \leq lag$ ，所以叠加区域可以从第一个接收帧中的第一个采样开始。

[0147] 解码语音信号和外插信号之间的“同相点”需要在叠加区域的中间，叠加区域尽可能设置在靠近第一个接收帧开始的地方。这减少了必须将前一丢失帧关联的合成语音信号外插到第一个接收帧的时间。在本发明的一个实施例中，这是通过执行两个阶段的时滞估计来完成的。在第一阶段，通过相对长的滞后搜索窗口来计算粗滞后估计值，窗口的中心可不与叠加区域的中心一致。例如，滞后搜索窗口可以是基音周期的 1.5 倍。滞后搜索范围（即相对于原始语音信号偏移外插语音信号的采样数量）也相对比较宽（即 ± 28 个采样）。为了改善对齐度，于是执行滞后细化搜索。作为滞后细化搜索的一部分，滞后搜索窗口与依照粗滞后估计得到的预计叠加设置同中心。这可以通过由粗滞后估计来偏置外插语音信号来完成。滞后细化搜索中的滞后搜索窗口的大小可以较小（例如叠加区域的大小），而滞后搜索范围也可以较小（即 ± 4 个采样）。搜索方法可以与上述 3. b. i 节中的方法相同。

[0148] 执行时间扭曲存在许多技术，一种技术包括分段式 (piece-wise) 单采样平移和叠加。图 13 的流程图 1300 描述了一种使用该技术进行收缩的方法。根据该方法，如步骤 1302 所示，周期性降低采样。从采样降低的这个点开始，叠加原始信号和向左平移的信号（由于降低），如步骤 1304 所示。图 14 的流程图 1400 描述了一种使用该技术进行拉伸的方法。根据该方法，周期性重复采样，如步骤 1402 所示。从采样重复的那个点开始，叠加原始信号和向右平移的信号（由于采样重复），如步骤 1404 所示。这些操作的叠加窗口的长度取决于采样增加 / 降低的周期性。为了避免太多的信号平滑，可以定义一个最大的叠加周期（即 8 个采样）。发生采样增加 / 降低的周期取决于各种因素，如帧大小、采样增加 / 降低的数量以及是否正在执行增加或降低。

[0149] 时间扭曲的量可以被限制。例如，在以下 D 节描述的 G. 722 系统中，可以将时间扭曲的量针对 10ms 的帧限制为 $\pm 1.75ms$ （或者 160 个采样的 10ms 帧中的 28 个采样）。大于此范围的扭曲可以消除上述的破坏性干扰，但是通常会带来一些其它的可听失真。因而，在这种实施例中，在时滞超出此范围的情况下，将不执行时间扭曲。

[0150] 本发明将以下在 D 节中描述的系统设置来确保在丢包后第一个接收帧之后的零采样延迟。为此，该系统不对超过第一个接收帧的解码语音信号执行时间扭曲。这进而限制了在上一段中所述的没有可听失真发生的时间扭曲的量。然而，所属领域的技术人员应知悉，在容纳某些采样延迟（在包丢失后的第一个接收帧之后）的系统中，可以对超过第

一个好帧的解码语音信号应用时间扭曲,从而能在没有可听失真的情况下调整到较大的时滞。当然,在这种系统中,如果第一个接收帧后的帧丢失,那么时间扭曲仅可以应用于与第一个好帧关联的解码语音信号。这种可选实施例也在本发明的范围和精神之内。

[0151] 在本发明的可选实施例中,对解码语音信号和外插语音信号都可以执行时间扭曲。由于多种原因,这种方法可以提供更好的性能。

[0152] 例如,如果时滞是-20,那么根据上述方法将解码语音信号作20个采样的收缩。这就是说需要产生外插语音信号的20个采样以用于第一个接收帧内。这个数量也可以通过收缩外插语音信号来减少。例如,可以将外插语音信号收缩4个采样,留下16个采样用于解码语音信号。这减少了必须用于第一个接收帧内的外插信号的采样数量,也减少了必须在解码语音信号上执行的扭曲的量。如上记录的,在D节的实施例中,时间扭曲需要限制到28个采样。用于对齐信号所需的时间扭曲量的减少意味着在时间扭曲过程中会引入更少的失真,而且也增加了可以改善的情形数量。

[0153] 通过将解码语音信号和外插语音信号作时间扭曲,也应该获得叠加区域中更好的波形匹配。解释如下:如果滞后是之前例子中的-20个采样,这就是说解码语音信号超前于外插信号20个采样。这种情况最有可能的原因是用于外插的基音周期大于实际基音。通过同样收缩外插语音信号,叠加区域中该信号的有效基音变得更小,更接近于实际基音周期。同样,通过收缩原始信号,该信号的有效基音周期大于仅用于收缩的情况。因此,叠加区域中的两个波形会具有更加匹配的基音周期,因而波形会更匹配。

[0154] 如果滞后是正的,那么拉伸解码语音信号。在这种情况下,虽然拉伸外插信号会增加用于第一个接收帧中的外插采样的数量,而是否可以获得改善却并不清楚。然而,如果存在长期的丢包,并且两个波形明显异相,那么该方法可以提供改善的性能。例如,如果滞后是30个采样,因为大于28个采样的限制,在前述的方法中并没有执行扭曲。30个采样的扭曲很可能引起本身的失真。然而,如果该30个采样分布在两个信号之间,如拉伸外插语音信号10个采样并拉伸解码语音信号20个采样,那么可以在没使用过多时间扭曲的情况下将它们对齐。

[0155] D、G. 722 解码器中的示例实现的细节

[0156] 这部分提供了涉及本发明在ITU-T推荐G. 722语音解码器中特殊实现的具体细节。该示例实现在固有的10毫秒(ms)帧大小上执行,也可以在任何10ms的倍数的包或帧大小上执行。较长的输入帧作为超帧(super frame)进行处理,对此,PLC逻辑以其固有的10ms帧大小调用合适数量的次数。与使用相同帧大小的常规G. 722解码相比,其并不会引起附加的延迟。本发明仅通过示例的方式提供了这些实现细节及以下给出的内容,并不能用来限制本发明。

[0157] 本节中描述的实施例满足与G. 722附录IV中描述的PLC算法相同的复杂度需求,但是提供了比那个附录中描述的PLC算法明显更好的语音质量。由于它的高质量,本节中描述的实施例适用于发生帧删除或丢包的G. 722的一般应用。这种应用包括,例如,互联网协议语音技术(VoIP)、无线保真语音技术(WiFi)和下一代数字增强无线通信(DECT)。本节中描述的实施例易于适用,除了在执行无PLC的基本G. 722解码器之后实际没有留下复杂度净空(headroom)的应用之外。

[0158] 1、缩写和约定

[0159] 表 1 中列出了在本节中使用的一些缩写。

[0160]

缩写	描述
ADPCM	自适应差分 PCM
ANSI	美国国家标准研究所
dB	分贝
DECT	数字增强无线通信
DC	直流
FIR	有限脉冲响应
Hz	赫兹
LPC	线性预测编码
OLA	叠加
PCM	脉冲编码调制
PLC	丢包隐藏
PWE	周期性波形外插
STL2005	软件工具库 2005
QMF	镜像二次滤波器
VoIP	互联网协议语音技术
WB	宽带
WiFi	无线保真

[0161] 表 1 :缩写

[0162] 本发明的描述也使用了一些约定,并将对其中一部分作出解释。PLC 算法以 10ms 的固有帧大小进行运算,因此该算法的描述仅针对 10ms 的帧。对于更大的包(10ms 的倍数),以 10ms 分段对接收包进行解码。在 16kHz 采样率级别上信号的离散时间一般使用“j”或“i”表示。在 8kHz 采样级别上信号的离散时间索引一般使用“n”表示。低带信号(0-4kHz)以下标“L”标识,高带信号(4-8kHz)以下标“H”标识。如果可能,该描述将重新使用 ITU-T G.722 的规范。

[0163] 以下表 2 中列出了最常用的符号及它们的描述。

[0164]

符号	描述
$x_{out}(j)$	16kHz G.722 解码器输出
$x_{PLC}(i)$	16kHz G.722 PLC 输出
$w(j)$	LPC 窗口
$x_w(j)$	窗口语音
$r(i)$	自相关
$\bar{r}(i)$	频谱平滑和白噪声纠正后的自相关
\hat{a}_i	中间 LPC 预测器系数
a_i	LPC 预测器系数
$d(j)$	16kHz 短期预测误差信号
avm	平均幅度
\hat{a}_i'	加权短期合成滤波器系数
$xw(j)$	16kHz 加权语音
$xwd(n)$	向下采样的加权语音(2kHz)
b_i	用于向下采样的 60 阶低通滤波器
$c(k)$	粗基音分析的自相关(2kHz)
$E(k)$	粗基音分析的能量(2kHz)

[0165]

$c2(k)$	粗基音分析的带符号平方自相关(2kHz)
cpp	粗基音周期
$cpplast$	最近一帧的粗基音周期
$E_i(j)$	内插的 $E(k)$ (到 16kHz)
$c2i(j)$	内插的 $c2(k)$ (到 16kHz)
$\tilde{E}(k)$	基音细化的能量(16kHz)
$\tilde{c}(k)$	基音细化的自相关(16kHz)
$ppfe$	用于帧删除的基音周期
$ptfe$	用于帧删除的基音抽头(tap)
ppt	基音预测器抽头
$merit$	周期性指标的数值
Gr	随机分量的比例因子
Gp	周期分量的比例因子
$ltring(j)$	长期(基音)呼叫
$ring(j)$	最终呼叫(包括短期)
$w_i(j)$	淡入窗口
$w_o(j)$	淡出窗口
$wn(j)$	噪声生成器的输出
$wgn(j)$	噪声生成器的经缩放输出
$fn(j)$	经滤波和缩放的噪声
$cfecount$	连续 10ms 帧删除的计数器
$w_i(j)$	叠加窗口
$w_o(j)$	叠加窗口
h_i	QMF 滤波器系数
$x_L(n)$	低带子带信号(8kHz)
$x_H(n)$	高带子带信号(8kHz)
$I_L(n)$	低带 ADPCM 编码器的索引(8kHz)

[0166]

$I_H(n)$	高带 ADPCM 编码器的索引(8kHz)
$s_{Lz}(n)$	低带预测信号, 零部分贡献
$s_{Lp}(n)$	低带预测信号, 极点部分贡献
$s_L(n)$	低带预测信号
$e_L(n)$	低带预测误差信号
$r_L(n)$	低带重构信号
$p_{Lt}(n)$	低带部分重构截断信号
$\nabla_L(n)$	低带对数比例因子
$\Delta_L(n)$	低带比例因子
$\nabla_{L,m1}(n)$	低带对数比例因子, 第一平均
$\nabla_{L,m2}(n)$	低带对数比例因子, 第二平均
$\nabla_{L,track}(n)$	低带对数比例因子, 追踪
$\nabla_{L,chg}(n)$	低带对数比例因子, 变化度
$\beta_L(n)$	低带极点部分的稳定性裕度
$\beta_{L,MA}(n)$	低带极点部分的稳定性裕度的移动平均
$\beta_{L,min}$	低带极点部分的最小稳定性裕度
$s_{Hz}(n)$	高带预测信号, 零部分贡献
$s_{Pz}(n)$	高带预测信号, 极点部分贡献
$s_H(n)$	高带预测信号
$e_H(n)$	高带预测误差信号
$r_H(n)$	高带重构信号
$r_{H,HP}(n)$	高带高通滤波重构信号
$p_H(n)$	高带部分重构信号
$p_{H,HP}(n)$	高带高通滤波部分重构信号
$\nabla_H(n)$	高带对数比例因子
$\nabla_{H,m}(n)$	高带对数比例因子, 平均
$\nabla_{H,track}(n)$	高带对数比例因子, 追踪

[0167]

$\nabla_{H,chg}(n)$	高带对数比例因子, 变化度
$a_{LP}(n)$	高带对数比例因子的高通滤波系数
$\nabla_{H,LP}(n)$	低通滤波高带对数比例因子
$r_{Le}(n)$	估计的低带重构误差信号
$es(n)$	用于重新定相的时滞计算的外插信号
$R_{SUB}(k)$	子采样的标准化交叉相关性
$R(k)$	标准化交叉相关性
T_{LSUB}	子采样时滞
T_L	用于重新定相的时滞
$es_{tw}(n)$	用于时间扭曲的时滞细化的外插信号
T_{Lwarp}	用于时间扭曲的时滞
$x_{warp}(j)$	经时间扭曲的信号(16kHz)
$es_{ola}(j)$	用于叠加的外插信号(16kHz)

[0168] 表 2 :常用符号及其描述

[0169] 2、PLC 算法的一般描述

[0170] 如以上参考图 5 描述的, 解码器 /PLC 系统 300 处理的帧有六种类型: 类型 1、类型 2、类型 3、类型 4、类型 5 和类型 6。类型 1 的帧是任何在包丢失之后的第八个接收帧以外的接收帧。类型 2 的帧是与包丢失关联的第一个和第二个丢失帧的其中之一。类型 3 的帧是与包丢失关联的第三个到第六个丢失帧的其中任何一个。类型 4 的帧是与包丢失关联的第六帧以外的任何丢失帧。类型 5 的帧是紧随包丢失后的一个接收帧。最后, 类型 6 的帧是紧随包丢失后的第二个到第八个接收帧的其中任何一个。本节中描述的 PLC 算法是在持续时间为 10ms 的固定帧大小上进行运算的。

[0171] 本发明根据标准的 G. 722 操作来解码类型 1 的帧, 增加了某些状态存储器的维护和处理以利于 PLC 及相关的处理。图 15 是根据本发明的实施例执行这些操作的逻辑的模块图 1500。具体地说, 如图 15 所示, 在处理类型 1 的帧时, 从比特解复用器 (图 15 中未示出) 中接收低带 ADPCM 编码器的索引 (index) $I_L(n)$, 并由低带 ADPCM 解码器 1510 解码来产生子带语音信号。类似地, 从比特解复用器中接收高带 ADPCM 编码器的索引数 $I_H(n)$, 并由高带 ADPCM 解码器 1520 解码来产生子带语音信号。QMF 合成滤波器组 1530 合成低带语音信号和高带语音信号来产生解码输出信号 $x_{out}(j)$ 。这些操作与标准 G. 722 解码是一致的。

[0172] 除这些标准 G. 722 解码操作以外, 在处理类型 1 的帧时, 逻辑模块 1540 用于更新 PLC 相关的低带 ADPCM 状态存储器, 逻辑模块 1550 用于更新 PLC 相关的高带 ADPCM 状态存储器, 逻辑模块 1560 用于更新 WB PCM PLC 相关的状态存储器。这些状态存储器更新用于加快与其它帧类型相关的 PLC 处理。

[0173] 对于类型 2、类型 3 和类型 4 的帧,宽带 (WB)PCM PLC 是在 16kHz 输出语音域内执行的。图 16 中提供了用于执行 WB PCM PLC 的逻辑的模块图 1600。G. 722 解码器的之前的输出语音 $x_{out}(j)$ 被缓存起来,并被传送到 WB PCMLC 逻辑。WB PCM PLC 算法是基于周期性波形外插 (PWE) 的,基音估计是 WB PCM PLC 逻辑的重要组成部分。最初,基于加权语音域中的向下采样 (到 2kHz) 信号来估计粗基音。随后,使用原始的 16kHz 采样以全解析度细化该估计值。WB PCM PLC 逻辑的输出 $x_{PLC}(i)$ 是周期性外插波形和由 PLC 定形的噪声的线性组合。对于持续的帧删除,输出波形 $x_{PLC}(i)$ 被逐渐减弱。减弱从帧丢失后的 20ms 后开始,并在帧丢失后的 60ms 后完成。

[0174] 如图 17 的模块图 1700 所示,对于类型 2、类型 3 和类型 4 的帧,WB PCMLC 逻辑的输出 $x_{PLC}(i)$ 被传送通过 G. 722QMF 分析滤波器组 1702 以获得相应的子带信号,这些子带信号随后被分别传送给改进的低带 ADPCM 编码器 1704 和改进的高带 ADPCM 编码器 1706 以更新解码器的状态和存储器。只有部分简化的子带 ADPCM 编码器用于该更新。

[0175] 图 16 和图 17 所示的逻辑执行的处理发生在丢失帧期间。改进的低带 ADPCM 编码器 1704 和改进的高带 ADPCM 编码器均被简化以减少复杂度。在本申请的其它地方将对它们作细节描述。在编码器 1704 和 1706 中出现的一个特征 (在常规的 G. 722 子带 ADPCM 编码器中并不存在) 是基于信号属性和包丢失持续时间的编码器自适应复位。

[0176] 与 PLC 算法关联的最复杂的处理是对于类型 5 的帧的处理,类型 5 的帧是紧随包丢失的第一个接收帧。在这个帧期间发生外插波形到标准解码波形的过渡。在处理类型 5 的帧时使用的技术包括重新定相和时间扭曲,这些将在此作更多的细节描述。图 18 提供了用于执行这些技术的逻辑的模块图 1800。另外,在处理类型 5 的帧时,以在此描述了更多细节的方式来更新解码器中的 QMF 合成滤波器组。与处理类型 5 的帧关联的另一功能包括在包丢失后第一个接收帧开始时的低带和高带对数比例因子的设定。

[0177] 在此都是使用改进的和约束的子带 ADPCM 解码器来解码类型 5 和类型 6 的帧。图 19 描述了用于处理类型 5 和类型 6 的帧的逻辑的模块图 1900。如图 19 所示,逻辑 1970 在处理类型 5 和 / 或类型 6 的帧时对子带 ADPCM 解码器 1910 和 1920 施加约束和控制。子带 ADPCM 解码器的约束和控制是在包丢失后的头 80ms 期间施加的。其中有些没有扩展到 40ms 以外,而其它的约束和控制持续时间或度上是自适应的。约束和控制机制将在本申请中作更多的细节描述。如图 19 所示,逻辑模块 1940、1950 和 1960 用于在处理类型 5 或类型 6 的帧之后更新状态存储器。

[0178] 在无错信道的条件下,本节中描述的 PLC 算法与 G. 722 是比特精准的 (bit-exact)。此外,在错误条件下,该算法在包丢失后第八帧以外与 G. 722 是相同的,如果没有比特错误,应该能得到向 G. 722 无错输出的收敛。

[0179] 本节中描述的 PLC 算法支持任何是 10ms 的倍数的帧大小。对于大于 10ms 的包大小,仅需针对每个包将 PLC 算法以 10ms 间隔调用多次。因此,在本节的后续部分中,将按照 10ms 的固定帧大小来描述 PLC 算法。

[0180] 3、G. 722 输出的波形外插

[0181] 对于与包丢失相应的丢失帧 (类型 2、类型 3 和类型 4 的帧),在图 16 中描述的 WB PCM PLC 逻辑外插与前一帧关联的 G. 722 输出波形来产生当前帧的替换波形。然后在处理类型 2、类型 3 和类型 4 的帧时将该外插宽带信号波形 $x_{PLC}(i)$ 用作 G. 722PLC 逻辑的输出波

形。为了方便地描述图 16 中各种模块,当 WB PCM PLC 逻辑为丢失帧计算出信号 $x_{\text{PLC}}(i)$ 以后,信号 $x_{\text{PLC}}(i)$ 被写入存储了 $x_{\text{out}}(j)$ 的缓存,其中 $x_{\text{out}}(j)$ 是整个 G.722 解码器 /PLC 系统的最终输出。现在对图 16 的每个处理模块作更多的细节描述。

[0182] a、八阶 LPC 分析

[0183] 模块 1604 用于在计算出当前帧关联的信号 $x_{\text{out}}(j)$ 并将其存储在缓存之后,在接近帧处理循环的结尾执行 8 阶 LPC 分析。这种 8 阶 LPC 分析是一种自相关 LPC 分析,具有应用到与当前帧关联的 $x_{\text{out}}(j)$ 信号的 10ms 不对称分析窗口。这种不对称窗口定义如下:

$$[0184] \quad w(j) = \begin{cases} \frac{1}{2} \left[1 - \cos\left(\frac{(j+1)\pi}{121}\right) \right], & \text{for } j = 0, 1, 2, \dots, 119 \\ \cos\left(\frac{(j-120)\pi}{80}\right), & \text{for } j = 120, 121, \dots, 159 \end{cases} \quad (4)$$

[0185] 假设 $x_{\text{out}}(0)$ 、 $x_{\text{out}}(1)$ 、 x_{out} 、 $x_{\text{out}}(159)$ 表示与当前帧关联的 G.722 解码器 /PLC 系统输出宽带信号采样。执行窗口运算如下:

$$[0186] \quad x_w(j) = x_{\text{out}}(j) w(j), \quad j = 0, 1, 2, \dots, 159. \quad (5)$$

[0187] 下一步,计算自相关系数如下:

$$[0188] \quad r(i) = \sum_{j=i}^{159} x_w(j) x_w(j-i), \quad i = 0, 1, 2, \dots, 8. \quad (6)$$

[0189] 然后将频谱平滑和白噪声纠正操作应用到自相关系数,如下:

$$[0190] \quad \hat{r}(i) = \begin{cases} 1.0001 \times r(0), & i = 0 \\ \frac{-(-2\pi i \sigma / f_s)^2}{r(i) e^2}, & i = 1, 2, \dots, 8 \end{cases}, \quad (7)$$

[0191] 其中 $f_s = 16000$ 是输入信号的采样率, $\sigma = 40$ 。

[0192] 下一步,使用列文森-杜宾 (Levinson-Durbin) 递归来将自相关系数 $\hat{r}(i)$ 转换为 LPC 预测器系数 \hat{a}_i , $i = 0, 1, \dots, 8$ 。如果列文森-杜宾递归在完成递归之前过早的退出(例如,因为预测误差能量 $E(i)$ 小于零),那么在当前帧中使用与最近一帧关联的短期预测器系数。为了解决这种方式的例外,需要 \hat{a}_i 数组的初始值。 \hat{a}_i 数组的初始值被设定为 $\hat{a}_0 = 1$ 和 $\hat{a}_i = 0$, $i = 1, 2, \dots, 8$ 。列文森-杜宾递归算法具体规定如下:

[0193] 1、如果 $\hat{r}(0) \leq 0$, 使用最近一帧的 \hat{a}_i 数组, 并退出列文森-杜宾递归

[0194] 2. $E(0) = \hat{r}(0)$

[0195] 3. $k_1 = -\hat{r}(1) / \hat{r}(0)$

[0196] 4. $\hat{a}_1^{(1)} = k_1$

[0197] 5. $E(1) = (1 - k_1^2) E(0)$

[0198] 6、如果 $E(1) \leq 0$, 使用最近一帧的 \hat{a}_i 数组, 并退出列文森-杜宾递归

[0199] 7、对于 $i = 2, 3, 4, \dots, 8$, 执行如下运算:

$$[0200] \quad \text{a. } k_i = \frac{-\hat{r}(i) - \sum_{j=1}^{i-1} \hat{a}_j^{(i-1)} \hat{r}(i-j)}{E(i-1)}$$

[0201] b. $\hat{a}_i^{(i)} = k_i$

[0202] c. $\hat{a}_j^{(i)} = \hat{a}_j^{(i-1)} + k_i \hat{a}_{i-j}^{(i-1)}$, for $i = 1, 2, \dots, i-1$

[0203] d. $E(i) = (1-k_i^2)E(i-1)$

[0204] e、如果 $E(i) \leq 0$, 使用最近一帧的 \hat{a}_j 数组, 并退出列文森 - 杜宾递归

[0205] 如果过早退出递归, 使用之前处理的帧的 \hat{a}_j 数组。如果顺利完成递归 (正常情况下), LPC 预测器系数如下:

[0206] $\hat{a}_0 = 1$ (8)

[0207] 及

[0208] $\hat{a}_i = \hat{a}_i^{(8)}$, for $i = 1, 2, \dots, 8$. (9)

[0209] 通过对以上获得的系数应用宽带扩展运算, 最终获得的 LPC 预测器系数组如下:

[0210] $a_i = (0.96852)^i \hat{a}_i$, $i = 0, 1, \dots, 8$. (10)

[0211] b、短期预测残余信号的计算

[0212] 图 16 的模块 1602 (标记为“ $A(z)$ ”) 表示短期线性预测误差滤波器, 滤波器系数为以上计算的 a_i , $i = 0, 1, \dots, 8$ 。模块 1602 用于在执行 8 阶 LPC 分析后进行操作。模块 1602 计算短期预测残余信号 $d(j)$ 如下:

[0213] $d(j) = x_{out}(j) + \sum_{i=1}^8 a_i \cdot x_{out}(j-i)$ for $j = \dots, 2, \dots, 159$. (11)

[0214] 传统上, 当前帧的时间索引 n 从之前处理的帧的时间索引继续。换言之, 如果时间索引范围 $0, 1, 2, \dots, 159$ 表示当前帧, 那么时间索引范围 $-160, -159, \dots, -1$ 就表示之前处理的帧。因而, 在以上的等式中, 如果索引 $(j-i)$ 是负的, 那么该索引指向在之前处理的帧的结尾附近的信号采样。

[0215] c、比例因子的计算

[0216] 图 16 中的模块 1606 用于计算与当前帧关联的短期预测残余信号的平均幅度。这一操作在模块 1602 以上述方式计算出短期预测残余信号 $d(j)$ 之后才执行。平均幅度 avm 的计算如下:

[0217] $avm = \frac{1}{160} \sum_{j=0}^{159} |d(j)|$. (12)

[0218] 如果下一待处理的帧是丢失帧 (换言之, 与包丢失相应的帧), 可以使用该平均幅度作为比例因子来调整白高斯 (Gaussian) 噪声序列 (如果当前帧是无声的)。

[0219] d、加权语音信号的计算

[0220] 图 16 的模块 1608 (标记为“ $1/A(z/y)$ ”) 表示加权短期合成滤波器。模块 1608 用于在以上述方式 (参考模块 1602) 计算出当前帧的短期预测残余信号 $d(j)$ 之后进行操作。这个加权短期合成滤波器的系数 a'_i ($i = 0, 1, \dots, 8$) 计算如下 (其中 $y_1 = 0.75$):

[0221] $a'_i = y_1^i a_i$, $i = 1, 2, \dots, 8$. (13)

[0222] 短期预测残余信号 $d(j)$ 通过该加权合成滤波器。相应的输出加权语音信号 $xw(j)$ 的计算如下:

[0223] $xw(j) = d(j) - \sum_{i=1}^8 a'_i \cdot xw(j-i)$, $j = \dots, 2, \dots, 159$. (14)

[0224] e、八比一抽样 (eight-to-one decimation)

[0225] 图 16 的模块 1616 将模块 1608 输出的加权语音信号传送通过 60 阶最小相位有限脉冲响应 (FIR) 滤波器, 然后执行 8 : 1 抽样来将获得的 16kHz 低通滤波加权语音信号向下采样为 2kHz 向下采样的加权语音信号 $xwd(n)$ 。这一抽样操作是在计算出加权语音信号后才执行的。为了减少复杂度, 仅当需要 $xwd(n)$ 的新采样时才执行 FIR 低通滤波操作。因而, 向下采样的加权语音信号 $xwd(n)$ 的计算如下:

$$[0226] \quad xwd(n) = \sum_{i=0}^{59} b_i \cdot xw(8n+7-i), \quad n = 1, 2, \dots, 19, \quad (15)$$

[0227] 其中 b_i ($i = 0, 1, 2, \dots, 59$) 是 60 阶 FIR 低通滤波器的滤波系数, 如表 3 所示。

[0228]

滞后, i	Q15 格式的 b_i	滞后, i	Q15 格式的 b_i	滞后, i	Q15 格式的 b_i
0	1209	20	-618	40	313
1	728	21	-941	41	143
2	1120	22	-1168	42	-6
3	1460	23	-1289	43	-126
4	1845	24	-1298	44	-211
5	2202	25	-1199	45	-259
6	2533	26	-995	46	-273
7	2809	27	-701	47	-254
8	3030	28	-348	48	-210
9	3169	29	20	49	-152
10	3207	30	165	50	-89
11	3124	31	365	51	-30
12	2927	32	607	52	21
13	2631	33	782	53	58
14	2257	34	885	54	81
15	1814	35	916	55	89
16	1317	36	881	56	84
17	789	37	790	57	66
18	267	38	654	58	41
19	-211	39	490	59	17

[0229] 表 3 :60 阶 FIR 滤波器的系数

[0230] f、粗基音周期提取

[0231] 为了减小计算的复杂度, WB PCM PLC 逻辑执行基音提取分为两个阶段: 首先使用 2kHz 抽样信号的时间解析度来确定粗基音周期, 然后使用 16kHz 非抽样信号的时间解析度来执行基音周期细化。仅当计算出向下采样的加权语音信号 $xwd(n)$ 之后才执行这种基音提取。本子部分描述了图 16 的模块 1620 所执行的第一阶段的粗基音周期提取算法。该算法基于使用某种附加判定逻辑来最大化标准的交叉相关性。

[0232] 在粗基音周期提取时可使用 15ms 的基音分析窗口。基音分析窗口的尾部与当前帧的尾部对齐。在 2kHz 的采样率下, 15ms 对应 30 个采样。在不失一般性的情况下, 假设索引范围 $n = 0$ 到 $n = 29$ 对应于用于 $xwd(n)$ 的基音分析窗口。粗基音周期提取算法通过计算以下值开始:

$$[0233] \quad c(k) = \sum_{n=0}^{29} xwd(n)xwd(n-k), \quad (16)$$

$$[0234] \quad E(k) = \sum_{n=0}^{29} [xwd(n-k)]^2, \quad (17)$$

[0235] 及

[0236]

$$c2(k) = \begin{cases} c^2(k), & \text{如果 } c(k) \geq 0 \\ -c^2(k), & \text{如果 } c(k) < 0 \end{cases} \quad (18)$$

[0237] 以上计算针对从 $k = \text{MINPPD}-1$ 到 $k = \text{MAXPPD}+1$ 范围内的所有整数进行, 其中 $\text{MINPPD} = 5$ 和 $\text{MAXPPD} = 33$ 分别是抽样范围中的最小和最大基音周期。然后粗基音周期提取算法在 $k = \text{MINPPD}, \text{MINPPD}+1, \text{MINPPD}+2, \dots, \text{MAXPPD}$ 的范围内进行搜索, 以找到满足 $c(k) > 0$ 的数组 $\{c2(k)/E(k)\}$ 的所有局部峰值。(如果一个值相邻的两个值都比它小, 将这个值定义为局部峰值)。假设 N_p 表示正的局部峰值的数量。假设 $k_p(j)$ ($j = 1, 2, \dots, N_p$) 为索引, 其中 $c2(k_p(j))/E(k_p(j))$ 是局部峰值且 $c(k_p(j)) > 0$, 并假设 $k_p(1) < k_p(2) < \dots < k_p(N_p)$ 。为了方便, $c2(k)/E(k)$ 将被称为“标准化相关性平方 (normalized correlation square)”。

[0238] 如果 $N_p = 0$, 即如果函数 $c2(k)/E(k)$ 不存在正的局部峰值, 那么该算法将搜索具有最大幅值 $|c2(k)/E(k)|$ 的最大的负局部峰值。如果找到这个最大的负局部峰值, 那么将相应的索引 k 用作输出粗基音周期 c_{pp} , 并终止模块 1620 的处理。如果标准化相关性平方函数 $c2(k)/E(k)$ 既没有正局部峰值, 也没有负局部峰值, 那么将输出粗基音周期设定为 $c_{pp} = \text{MIPPD}$, 并终止模块 1620 的处理。如果 $N_p = 1$, 那么将输出粗基音周期设定为 $c_{pp} = k_p(1)$, 并终止模块 1620 的处理。

[0239] 如果存在至少两个局部峰值 ($N_p \geq 2$), 那么这个模块使用算法 A、B、C 和 D (将在下面描述), 按照这个顺序来确定输出粗基音周期 c_{pp} 。在这四个算法中 较前的算法中计算出的变量将传递给后一算法继续使用。

[0240] 以下的算法 A 用于识别标准化相关性平方 $c2(k_p)/E(k_p)$ 的局部峰值周围的最大二次内插峰值。针对 $c(k_p)$ 执行二次内插, 而针对 $E(k_p)$ 执行线性内插。这种内插是使用 16kHz 非抽样语音信号的时间解析度来执行的。在以下的算法中, D 表示将 $xw(n)$ 抽样为到 $xwd(n)$ 时使用的抽样因子。因而, 在此 $D = 8$ 。

[0241] 算法 A- 在 $c2(k_p)/E(k_p)$ 周围查找最大二次内插峰值。

[0242] A、设定 $c2_{\max} = -1, E_{\max} = 1, j_{\max} = 0$ 。

[0243] B、对于 $j = 1, 2, \dots, N_p$, 进行如下 12 个步骤:

[0244] 1、设定 $a = 0.5[c(k_p(j)+1) + c(k_p(j)-1)] - c(k_p(j))$

[0245] 2、设定 $b = 0.5[c(k_p(j)+1) + c(k_p(j)-1)]$

[0246] 3、设定 $j_i = 0$

[0247] 4、设定 $e_i = E(k_p(j))$

[0248] 5、设定 $c2_m = c2(k_p(j))$

[0249] 6、设定 $E_m = E(k_p(j))$

[0250] 7、如果 $c2(k_p(j)+1)E(k_p(j)-1) > c2(k_p(j)-1)E(k_p(j)+1)$, 进行步骤 7 的余下部分:

[0251] a、 $\Delta = [E(k_p(j)+1) - e_i]/D$

- [0252] b、对于 $k = 1, 2, \dots, D/2$, 进行步骤 7 的以下部分 :
- [0253] i. $c_i = a(k/D)^2 + b(k/D) + c(k_p(j))$
- [0254] ii. $e_i \leftarrow e_i + \Delta$
- [0255] iii. 如果 $(c_i)^2 E_m > (c_{2m}) e_i$, 进行下面三行 :
- [0256] a. $j_i = k$
- [0257] b. $c_{2m} = (c_i)^2$
- [0258] c. $E_m = e_i$
- [0259] 8、如果 $c_2(k_p(j)+1)E(k_p(j)-1) \leq c_2(k_p(j)-1)E(k_p(j)+1)$, 进行步骤 8 的余下部分 :
- [0260] a. $\Delta = [E(k_p(j)-1) - e_i]/D$
- [0261] b、对于 $k = -1, -2, \dots, -D/2$, 进行步骤 8 的以下部分 :
- [0262] i. $c_i = a(k/D)^2 + b(k/D) + c(k_p(j))$
- [0263] ii. $e_i \leftarrow e_i + \Delta$
- [0264] iii. 如果 $(c_i)^2 E_m > (c_{2m}) e_i$, 进行下面三行 :
- [0265] a. $j_i = k$
- [0266] b. $c_{2m} = (c_i)^2$
- [0267] c. $E_m = e_i$
- [0268] 9、设定 $\text{lag}(j) = k_p(j) + j_i/D$
- [0269] 10、设定 $c_{2i}(j) = c_{2m}$
- [0270] 11、设定 $E_i(j) = E_m$
- [0271] 12、如果 $c_{2m} \times E_{\max} > c_{2\max} \times E_m$, 进行以下三行 :
- [0272] a. $j_{\max} = j$
- [0273] b. $c_{2\max} = c_{2m}$
- [0274] c. $E_{\max} = E_m$
- [0275] 符号 \leftarrow 表示使用右侧的值更新左侧的参数。
- [0276] 为了避免选择大约是实际粗基音周期的整数倍的粗基音周期, 对与 $c_2(k_p)/E(k_p)$ 的局部峰值相对应的各时滞进行搜索, 以确定是否有时滞足够接近之前处理的帧的输出粗基音周期 (表示为 cp_{plast} , 对于每个第一帧, cp_{plast} 被初始化为 12)。如果有时滞位于 cp_{plast} 的 25% 以内, 那么就认为足够接近了。对于所有在 cp_{plast} 的 25% 以内的时滞, 将相应的标准化相关性平方 $c_2(k_p)/E(k_p)$ 的二次内插峰值进行比较, 并选择其中对应于最大标准化相关性平方的内插时滞用于后续处理。以下的算法 B 执行了上述的任务。在该算法中使用了在上述算法 A 中计算出的内插数组 $c_{2i}(j)$ 和 $E_i(j)$ 。
- [0277] 算法 B- 在所有接近最近一帧的输出粗基音周期的时滞之中查找最大化内插 $c_2(k_p)/E(k_p)$ 的时滞 :
- [0278] A、设定索引 $im = -1$
- [0279] B、设定 $c_{2m} = -1$
- [0280] C、设定 $E_m = 1$
- [0281] D、对于 $j = 1, 2, \dots, N_p$, 进行如下运算 :
- [0282] 1. 如果 $|k_p(j) - cp_{\text{plast}}| \leq 0.25 \times cp_{\text{plast}}$, 进行如下 :

[0283] a. 如果 $c_{2i}(j) \times E_m > c_{2m} \times E_i(j)$, 进行如下三行 :

[0284] i. $im = j$

[0285] ii. $c_{2m} = c_{2i}(j)$

[0286] iii. $E_m = E_i(j)$

[0287] 注意, 如果没有位于 $cpplast$ 的 25% 以内的时滞 $k_p(j)$, 那么索引 im 的值将在执行算法 B 后保持为 -1。如果存在一个或多个位于 $cpplast$ 的 25% 以内的时滞, 那么索引 im 在这些时滞中对应于最大标准化相关性平方。

[0288] 接下来, 算法 C 确定是否在基音范围的前半周选择另一个时滞作为输出粗基音周期。该算法搜索小于 16 个的所有内插时滞 $lag(j)$, 并检查它们中是否有接近其每个整数倍 (一直到 32, 包括其本身) 的足够大的标准化相关性平方的局部峰值。如果存在满足这个条件的一个或多个时滞, 那么选择这些满足条件的时滞中最小的一个时滞作为输出粗基音周期。

[0289] 此外, 在以上算法 A 和算法 B 中计算出的各个变量也传递其最终值给以下的算法 C。如下所述, 参数 $MPDTH$ 为 0.06, 给出阈数组 $MPTH(k)$ 为 $MPTH(2) = 0.7$, $MPTH(3) = 0.55$, $MPTH(4) = 0.48$, $MPTH(5) = 0.37$, $MPTH(k) = 0.30$, $k > 5$ 。

[0290] 算法 C- 检查是否应该选择粗基音周期范围内的前半周中的另一个时滞作为输出粗基音周期 :

[0291] A、对于 $j = 1, 2, 3, \dots, N_p$, 当 $lag(j) < 16$ 时, 按该顺序进行如下操作 :

[0292] 1、如果 $j \neq im$, 设定 $threshold = 0.73$; 否则, 设定 $threshold = 0.4$ 。

[0293] 2、如果 $c_{2i}(j) \times E_{max} \leq threshold \times c_{2max} \times E_i(j)$, 取消这个 j , 并跳过对应这个 j 的步骤 (3), 将 j 增 1 并返回步骤 (1)。

[0294] 3、如果 $c_{2i}(j) \times E_{max} > threshold \times c_{2max} \times E_i(j)$, 进行如下操作 :

[0295] a、对于 $k = 2, 3, 4, \dots$, 当 $k \times lag(j) < 32$ 时, 进行如下 :

[0296] i、 $s = k \times lag(j)$

[0297] ii、 $a = (1 - MPDTH) s$

[0298] iii、 $b = (1 + MPDTH) s$

[0299] iv、按照顺序经过 $m = j+1, j+2, j+3, \dots, N_p$, 看是否有时滞 $lag(m)$ 位于 a 和 b 之间。如果没有时滞处于 a 和 b 之间, 跳过这个 j , 停止步骤 3, 将 j 增 1 并返回步骤 1。如果存在满足 $a < lag(m) < b$ 且 $c_{2i}(m) \times E_{max} > MPTH(k) \times c_{2max} \times E_i(m)$ 的至少一个 m , 那么认为在 $lag(j)$ 的第 k 个整数倍的附近找到了标准化相关性平方的足够大的峰值; 在这种情况下, 停止步骤 3. a. iv, 将 k 增 1 并返回步骤 3. a. i. b、如果是在没有过早停止的情况下完成步骤 3. a 的, 也就是, 如果在小于 32 的 $lag(j)$ 的每个整数倍的 $\pm 100 \times MPDTH\%$ 中存在标准化相关性平方的足够大的内插峰值, 那么停止该算法, 跳过算法 D 并将 $c_{pp} = lag(j)$ 作为最终的输出粗基音周期。

[0300] 如果执行完以上的算法 C 而没有找到满足条件的输出粗基音周期 c_{pp} , 算法 D 将检查最近一帧的粗基音周期周围的标准化相关性平方的最大局部峰值 (在以上的算法 B 中得到), 并最终确定输出粗基音周期 c_{pp} 。同样, 在以上算法 A 和算法 B 中计算出的变量将其最终值传递给以下的算法 D。如下所述, 参数为 $SMDTH = 0.095$, $LPTHI = 0.78$ 。

[0301] 算法 D- 输出粗基音周期的最终确定 :

[0302] A、如果 $im = -1$ ，即如果在最近一帧的粗基音周期周围没有足够大的标准化相关性平方的局部峰值，那么将在算法 A 最后计算出的 c_{pp} 作为最终的输出粗基音周期，并退出该算法。

[0303] B、如果 $im = j_{max}$ ，即如果在最近一帧的粗基音周期周围的标准化相关性平方的最大局部峰值也是该帧中标准化相关性平方的所有内插峰值中的全局最大值，那么将在算法 A 最后计算出的 c_{pp} 作为最终的输出粗基音周期，并退出该算法。

[0304] C、如果 $im < j_{max}$ ，进行如下部分：

[0305] 1、如果 $c_{2m} \times E_{max} > 0.43 \times c_{2max} \times E_m$ ，进行步骤 C 的以下部分：

[0306] a、如果 $lag(im) > MAXPPD/2$ ，设定输出 $c_{pp} = lag(im)$ ，并退出该算法。

[0307] b、否则，对于 $k = 2, 3, 4, 5$ ，进行以下部分：

[0308] i、 $s = lag(j_{max})/k$

[0309] ii、 $a = (1-SMDTH)s$

[0310] iii、 $b = (1+SMDTH)s$

[0311] iv、如果 $lag(im) > a$ 且 $lag(im) > b$ ，设定输出 $c_{pp} = lag(im)$ ，并退出该算法。

[0312] D、如果 $im > j_{max}$ ，进行以下部分：

[0313] 1、如果 $c_{2m} \times E_{max} > LPTH1 \times c_{2max} \times E_m$ ，设定输出 $c_{pp} = lag(im)$ ，并退出该算法。

[0314] E、如果算法执行到此，那么上述步骤都不没有选择出最终的输出粗基音周期。在这种情况下，仅接受在算法 A 的结尾计算出的 c_{pp} 作为最终的输出粗基音周期。

[0315] g、基音周期细化

[0316] 图 16 中的模块 1622 用于通过使用 G. 722 解码输出语音信号以完整的 16kHz 时间解析度搜索粗基音周期的附近区域来执行基音周期提取算法的第二阶段处理。该模块首先通过用抽样因子 D 乘以粗基音周期 c_{pp} 来将粗基音周期 c_{pp} 转换到非抽样信号域，其中 $D = 8$ 。基音细化分析窗口大小 WSZ 选定为 $c_{pp} \times D$ 个采样和 160 个采样（对应 10ms）中较小的一个窗口大小： $WSZ = \min(c_{pp} \times D, 160)$ 。

[0317] 接下来，计算搜索范围的下边界为 $lb = \max(MINPP, c_{pp} \times D - 4)$ ，其中 $MINPP = 40$ 个采样是最小基音周期。计算搜索范围的上边界为 $ub = \max(MAXPP, c_{pp} \times D + 4)$ ，其中 $MAXPP = 265$ 个采样是最大基音周期。

[0318] 模块 1622 维持了总共是 $XQOFF = MAXPP + 1 + FRSZ$ 个采样的 16kHz G. 722 解码语音信号 $x_{out}(j)$ 的缓存，其中 $FRSZ = 160$ 是帧大小。这个缓存的最后 $FRSZ$ 个采样包含当前帧的 G. 722 解码语音信号。前 $MAXPP + 1$ 个采样是当前帧之前被处理的上一帧中的 G. 722 解码器 / PLC 系统输出信号。分析窗口的最后一个采样与当前帧的最后一个采样对齐。设从 $j = 0$ 到 $j = WSZ - 1$ 的索引范围对应于该分析窗口（该窗口是 $x_{out}(j)$ 缓存中的最后 WSZ 个采样），并且设负索引表示分析窗口之前的采样。在搜索范围 $[lb, ub]$ 内针对时滞 k 计算非抽样信号域中的以下相关性函数和能量项（energy term）如下：

$$[0319] \quad \tilde{c}(k) = \sum_{j=0}^{WSZ-1} x_{out}(j)x_{out}(j-k) \quad (19)$$

$$[0320] \quad \tilde{E}(k) = \sum_{j=0}^{WSZ-1} x_{out}(j-k)^2 \quad (20)$$

[0321] 然后将最大化比率 $\tilde{c}^2(k)/\tilde{E}(k)$ 的时滞 $k \in [lb, ub]$ 选定为帧删除的最终细化基音周期, 或 ppfe。即

$$[0322] \quad ppfe = \arg \max_{k \in [lb, ub]} \left[\frac{\tilde{c}^2(k)}{\tilde{E}(k)} \right]. \quad (21)$$

[0323] 接下来, 模块 1622 还计算了两个更加与基音相关的比例因子。第一个被称为 ptfe 或帧删除的基音抽头, 其是用于周期性波形外插的比例因子, 并被计算为分析窗口中 $x_{out}(j)$ 信号的平均幅度和 ppfe 个采样之前的 $x_{out}(j)$ 信号部分的平均幅度的比, 具有与这两个信号部分之间的相关性函数相同的符号, 如下:

$$[0324] \quad ptfe = \text{sign}(\tilde{c}(ppfe)) \left[\frac{\sum_{j=0}^{WSZ-1} |x_{out}(j)|}{\sum_{j=0}^{WSZ-1} |x_{out}(j - ppfe)|} \right]. \quad (22)$$

[0325] 在 $\sum_{j=0}^{WSZ-1} |x_{out}(j - ppfe)| = 0$ 的退化情况下, 将 ptfe 设定为 0。当 ptfe 的这种计算完成之后, ptfe 值的边界范围为 $[-1, 1]$ 。

[0326] 第二个与基音相关的比例因子被称为 ppt 或基音预测抽头, 用于计算长期滤波呼叫信号 (后面将对此进行叙述), 其被计算为 $ppt = 0.75 \times ptfe$ 。

[0327] h、计算混合比 (Mixing Ratio)

[0328] 图 16 中的模块 1618 计算优值系数 (figure of merit) 来确定丢失帧期间的周期性外插波形和经滤波噪声波形之间的混合比。这一计算仅在每次发生包丢失时的第一个丢失帧期间执行。优值系数是三个信号特性的加权和: 对数增益、第一标准化自相关和基音预测增益, 它们每个的计算如下。

[0329] 使用与之前的子节描述中相同的用于 $x_{out}(j)$ 的索引约定, 基音细化分析窗口中的 $x_{out}(j)$ 信号的能量为

$$[0330] \quad \text{sige} = \sum_{j=0}^{WSZ-1} x_{out}^2(j), \quad (23)$$

[0331] 且以 2 为底的对数增益 lg 的计算如下

$$[0332] \quad lg = \begin{cases} \log_2(\text{sige}), & \text{若 } \text{sige} \neq 0 \\ 0, & \text{若 } \text{sige} = 0 \end{cases}. \quad (24)$$

[0333] 如果 $\tilde{E}(ppfe) \neq 0$, 基音预测余能的计算如下

$$[0334] \quad \text{rese} = \text{sige} - \tilde{c}^2(ppfe)/\tilde{E}(ppfe), \quad (25)$$

[0335] 而基音预测增益 pg 的计算如下

$$[0336] \quad pg = \begin{cases} 10 \log_{10} \left(\frac{\text{sige}}{\text{rese}} \right), & \text{若 } \text{rese} \neq 0 \\ 20, & \text{若 } \text{rese} = 0 \end{cases}. \quad (26)$$

[0337] 如果 $\tilde{E}(ppfe) = 0$, 设定 $pg = 0$ 。如果 $\text{sige} = 0$, 同样设定 $pg = 0$ 。

[0338] 第一标准化自相关 ρ_1 的计算如下

[0339]

$$\rho_1 = \begin{cases} \frac{\sum_{j=0}^{WSZ-2} x_{out}(j)x_{out}(j+1)}{sige}, & \text{若 } sige \neq 0 \\ 0, & \text{若 } sige = 0 \end{cases} \quad (27)$$

[0340] 当获得这三个信号特性之后, 优值系数的计算如下

$$[0341] \quad \text{merit} = \lg + \text{pg} + 12\rho_1 \quad (28)$$

[0342] 以上计算出的 merit 确定了两个比例因子 G_p 和 G_r , 这两个比例因子有效地确定了周期性外插波形和经滤波噪声波形之间的混合比。这里有两个用于 merit 的阈值: 优值高阈值 MHI 和优值低阈值 MLO。这些阈值被设定为 $MHI = 28$ 和 $MLO = 20$ 。随机 (滤波噪声) 成分的比例因子 G_r 的计算如下

$$[0343] \quad G_r = \frac{MHI - \text{merit}}{MHI - MLO}, \quad (29)$$

[0344] 而周期性成分的比例因子 G_p 的计算如下

$$[0345] \quad G_p = 1 - G_r \quad (30)$$

[0346] i、周期性波形外插

[0347] 图 16 中的模块 1624 用于在丢失帧期间周期性地外插前一输出语音波形 (如果 $\text{merit} > MLO$)。现在将对模块 1624 执行该功能的方式进行描述。

[0348] 对于每次包丢失的第一个丢失帧, 计算出每帧的平均基音周期增量。基音周期历史缓存 $\text{pph}(m)$ ($m = 1, 2, \dots, 5$) 保存了前 5 帧的基音周期 ppfe 。平均基音周期增量是按照以下过程获得的。以即时的最近一帧开始, 计算从其前一帧到该帧的基音周期增量 (负值表示基音周期减量)。如果基音周期增量为零, 该算法会检查前一帧的基音周期增量。该处理过程持续直到检测到第一个具有非零基音周期增量的帧为止, 或者直到检测到第四个先前帧为止。如果所有在前的五帧具有相同的基音周期, 那么将平均基音周期增量设为零。否则, 如果在第 m 个先前帧发现第一个非零基音周期增量, 并且如果该基音周期增量的幅度小于在该帧上的基音周期的 5%, 则平均基音周期增量 ppinc 计算为该帧上的基音周期增量除以 m , 并且将结果值限制在 $[-1, 2]$ 的范围内。

[0349] 在包丢失的第二个连续丢失帧中, 将平均基音周期增量与基音周期 ppfe 相加, 并将结果数四舍五入成最接近的整数, 然后将其限制到 $[MIPP, \text{MAXPP}]$ 的范围内。

[0350] 如果当前帧是包丢失的第一个丢失帧, 那么计算出在叠加中使用的所谓的“呼叫信号”, 以保证在帧起始时的平滑波形过渡。呼叫信号和周期性外插波形的叠加长度是第一个丢失帧的 20 个采样。假设 $j = 0, 1, 2, \dots, 19$ 的索引范围对应当前第一个丢失帧的头 20 个采样, 即为叠加周期, 并且假设负索引对应先前的帧。便可获得长期呼叫信号为短期预测残余信号 (其比叠加周期早一个基音周期) 的缩放版本:

[0351]

$$ltring(j) = x_{out}(j - \text{ppfe}) + \sum_{i=1}^8 a_i \cdot x_{out}(j - \text{ppfe} - i), \quad j = 0, 1, 2, \dots, 19. \quad (31)$$

[0352] 在计算出 $ltring(j)$ 的这些 20 个采样之后,用模块 622 计算出的比例因子 ppt 进一步对它们进行调整:

$$[0353] \quad ltring(j) \leftarrow ppt \cdot ltring(j), j = 0, 1, 2, \dots, 19. \quad (32)$$

[0354] 使用在最近一帧中被初始化为 $x_{out}(j)$ 信号的最后 8 个采样的滤波存储器 $ring(j)$ ($j = -8, -7, \dots, -1$), 获得的最终呼叫信号如下:

$$[0355] \quad ring(j) = ltring(j) - \sum_{i=1}^8 a_i \cdot ring(j-i), \quad j = 0, 1, 2, \dots, 19. \quad (33)$$

[0356] 假设 $j = 0, 1, 2, \dots, 159$ 的索引范围对应于当前第一个丢失帧, 而 $j = 160, 161, 162, \dots, 209$ 的索引范围对应于下一帧的头 50 个采样。此外, 假设 $wi(j)$ 和 $wo(j)$ ($j = 0, 1, \dots, 19$) 分别是三角形淡入和淡出窗口, 所以 $wi(j) + wo(j) = 1$ 。那么, 周期性波形外插是分以下两个步骤执行的:

[0357] 步骤 1:

$$[0358] \quad x_{out}(j) = wi(j) \cdot ptfe \cdot x_{out}(n-ppfe) + wo(j) \cdot ring(j), j = 0, 1, 2, \dots, 19. \quad (34)$$

[0359] 步骤 2:

$$[0360] \quad x_{out}(j) = ptfe \cdot x_{out}(j-ppfe), j = 20, 21, 22, \dots, 209. \quad (35)$$

[0361] j、标准化噪声生成器

[0362] 如果 $merit < MHI$, 图 16 中的模块 1610 会产生具有单一平均幅度的白高斯随机噪声序列。为了降低计算复杂度, 预先计算白高斯随机噪声并存储在表中。为了避免使用过长的表并避免由于过短的表而重复相同的噪声模式, 本发明将使用一种特殊的索引方案。在该方案中, 白高斯噪声表 $wn(j)$ 具有 127 个条目, 噪声生成器模块的输出的调整版本为

$$[0363] \quad wgn(j) = avm \times wn(\text{mod}(cfecount \times j, 127)), j = 0, 1, 2, \dots, 209, \quad (36)$$

[0364] 其中 $cfecount$ 是帧计数器, 对于当前包丢失中的第 k 个连续丢失帧, $cfecount = k$, $\text{mod}(m, 127) = m - 127 \times \lfloor m/127 \rfloor$ 是模运算。

[0365] k、噪声序列的滤波

[0366] 图 16 中的模块 1614 表示短期合成滤波器。如果 $merit < MHI$, 模块 1614 对经调整的白高斯噪声进行滤波以对其给出与最近一帧内的 $x_{out}(j)$ 信号的频谱包络相同的频谱包络。获得的经滤波噪声 $fn(j)$ 如下

$$[0367] \quad fn(j) = wgn(j) - \sum_{i=1}^8 a_i \cdot fn(j-i), \quad j = 0, 1, 2, \dots, 209. \quad (37)$$

[0368] 1、周期性和随机成分的混合

[0369] 如果 $merit > MHI$, 那么只有模块 1624 计算出的周期性外插波形 $x_{out}(j)$ 用作 WB PCM PLC 逻辑的输出。如果 $merit < MLO$, 只有模块 1614 产生的经滤波噪声信号 $fn(j)$ 用作 WB PCM PLC 逻辑的输出。如果 $MLO \leq merit \leq MHI$, 那么将两个成分混合为

$$[0370] \quad x_{out}(j) \leftarrow Gp \cdot x_{out}(j) + Gr \cdot fn(j), j = 0, 1, 2, \dots, 209. \quad (38)$$

[0371] 外插的 $x_{out}(j)$ 信号 ($j = 160, 161, 162, \dots, 199$) 的前 40 个额外采样将变为下一帧的呼叫信号 $ring(j)$, $j = 0, 1, 2, \dots, 39$ 。如果下一帧还是丢失帧, 那么只有该呼叫信号的前 20 个采样用于叠加。如果下一帧是接收帧, 那么该呼叫信号的所有 40 个采样都将用于叠加。

[0372] m、有条件的斜降 (conditional ramp down)

[0373] 如果包丢失持续 20ms 或更短,那么通过混合周期性和随机成分而产生的 $x_{out}(j)$ 信号将用作 WB PCM PLC 输出信号。如果包丢失持续大于 60ms,那么 WB PCM PLC 输出信号被完全消声。如果包丢失持续大于 20ms 但小于 60ms,那么通过混合周期性和随机成分而产生的 $x_{out}(j)$ 信号将产生线性斜降 (以线性方式衰减到零)。如以下特定的算法中规定的,这个有条件的斜降是在 $cfecount > 2$ 时的丢失帧期间执行的。在此给出了以 Q15 形式的数组 $gawd()$ 为 $\{-52, -69, -104, -207\}$ 。同样, $j = 0, 1, 2, \dots, 159$ 的索引范围对应于 $x_{out}(j)$ 的当前帧。有条件的斜降算法:

[0374] A、如果 $cfecount \leq 6$,进行以下 9 行:

[0375] 1、 $\delta = gawd(cfecount-3)$

[0376] 2、 $gaw = 1$

[0377] 3、对于 $j = 0, 1, 2, \dots, 159$,进行以下两行:

[0378] a. $x_{out}(j) = gaw \cdot x_{out}(j)$

[0379] b. $gaw = gaw + \delta$

[0380] 4、如果 $cfecount < 6$,进行以下三行:

[0381] a、对于 $j = 160, 161, 162, \dots, 209$,进行以下两行:

[0382] i. $x_{out}(j) = gaw \cdot x_{out}(j)$

[0383] ii. $gaw = gaw + \delta$

[0384] B、否则 (如果 $cfecount > 6$),设定 $x_{out}(j) = 0, j = 0, 1, 2, \dots, 209$ 。

[0385] n、第一个接收帧中的叠加

[0386] 对于类型 5 的帧,将来自 G. 722 解码器的输出 $x_{out}(j)$ 与来自最后一个丢失帧的呼叫信号 $ring(j)$ (由模块 1624 以如上描述的方式计算出) 进行叠加:

[0387] $x_{out}(j) = w_i(j) \cdot x_{out}(j) + w_o(j) \cdot ring(j) \quad j = 0 \dots L_{OLA}-1, (39)$

[0388] 其中

[0389]
$$L_{OLA} = \begin{cases} 8 & \text{if } G_p = 0 \\ 40 & \text{otherwise} \end{cases} \quad (40)$$

[0390] 4、PLC 输出的重新编码

[0391] 为了在丢失帧期间 (类型 2、类型 3 和类型 4 的帧) 更新 G. 722ADPCM 解码器的存储器和参数,PLC 输出本质上要经过 G. 722 编码器。图 17 是用于执行该重新编码处理的逻辑的模块图 1700。如图 17 所示,PLC 输出 $x_{out}(j)$ 通过 QMF 分析滤波器组 1702 以产生低带子带信号 $x_L(n)$ 和高带子带信号 $x_H(n)$ 。低带子带信号 $x_L(n)$ 由低带 ADPCM 编码器 1704 进行编码,高带子带信号 $x_H(n)$ 由高带 ADPCM 编码器 1706 进行编码。为了降低复杂度,与传统的 ADPCM 子带编码器相比,本发明将 ADPCM 子带编码器 1704 和 1706 进行简化。现在对前述的操作进行更多的细节描述。

[0392] a、传送 PLC 输出经过 QMF 分析滤波器组

[0393] QMF 分析滤波器组 1702 的存储器被初始化来提供与解码子带信号连续的子带信号。WB PCM PLC 输出的最初 22 个采样构成了滤波存储器,子带信号根据如下等式计算得到:

$$[0394] \quad x_L(n) = \sum_{i=0}^{11} h_{2i} \cdot x_{PLC}(23+j-2i) + \sum_{i=0}^{11} h_{2i+1} \cdot x_{PLC}(22+j-2i), \text{ and} \quad (41)$$

$$[0395] \quad x_H(n) = \sum_{i=0}^{11} h_{2i} \cdot x_{PLC}(23+j-2i) - \sum_{i=0}^{11} h_{2i+1} \cdot x_{PLC}(22+j-2i), \quad (42)$$

[0396] 其中 $x_{PLC}(0)$ 对应当前帧的 16kHz WB PCM PLC 输出的第一个采样, $x_L(n=0)$ 和 $x_H(n=0)$ 分别对应当前帧的 8kHz 低带和高带子带信号的第一个采样。除了偏移额外的 22 个采样外,所述滤波与 G. 722 编码器的传送 QMF 是相同的,并且 WB PCM PLC 输出(与输入相对)被传送到滤波器组。此外,为了产生子带信号的完整帧(80 个采样~10ms),WB PCM PLC 需要在当前帧之外扩展 22 个采样,并产生 182 个采样~11.375ms。子带信号 $x_L(n)$ ($n=0,1,\dots,79$) 和 $x_H(n)$ ($n=0,1,\dots,79$) 分别是根据等式 41 和 42 产生的。

[0397] b、低带信号的重新编码

[0398] 低带信号 $x_L(n)$ 是使用简化的低带 ADPCM 编码器来进行编码的。简化的低带 ADPCM 编码器 2000 的模块图如图 20 所示。在图 20 中可以看到,删除了标准的低带 ADPCM 编码器的反向量化器,且用非量化预测误差代替了量化预测误差。此外,因为自适应量化器的更新仅基于由 6 比特低带编码器索引 $I_L(n)$ 表示的 64 元素集合中的 8 元素子集进行,所以预测误差仅被量化为 8 元素集。这提供了自适应量化器的相同更新,也简化了量化。表 4 列出了基于 $e_L(n)$ 的绝对值的 8 级简化量化器的判定级别、输出代码和乘数。

[0399]

m_L	低阈值	高阈值	I_L	乘数, W_L
1	0.00000	0.14103	3c	-0.02930
2	0.14103	0.45482	38	-0.01465
3	0.45482	0.82335	34	0.02832
4	0.82335	1.26989	30	0.08398
5	1.26989	1.83683	2c	0.16309
6	1.83683	2.61482	28	0.26270
7	2.61482	3.86796	24	0.58496
8	3.86796	∞	20	1.48535

[0400] 表 4 :8 级简化量化器的判定级别、输出代码和乘数

[0401] 图 20 的实体是根据它们的 G. 722 低带 ADPCM 子带编码器的等效形式来计算的 :

$$[0402] \quad s_{Lz}(n) = \sum_{i=1}^6 b_{L,i}(n-1) \cdot e_L(n-i), \quad (43)$$

$$[0403] \quad s_{Lp}(n) = \sum_{i=1}^2 a_{L,i}(n-1) \cdot x_L(n-i), \quad (44)$$

$$[0404] \quad s_L(n) = s_{Lp}(n) + s_{Lz}(n), \quad (45)$$

$$[0405] \quad e_L(n) = x_L(n) - s_L(n), \text{ 及} \quad (46)$$

$$[0406] \quad p_{Lt}(n) = s_{Lz}(n) + e_L(n). \quad (47)$$

[0407] 自适应量化器准确地根据 G. 722 编码器的规定进行更新。零和极点部分的自适应与在 G. 722 编码器中一样发生,如 G. 722 规范的条款 3.6.3 和 3.6.4 中所描述的。

[0408] 低带 ADPCM 解码器 1910 在帧丢失的 60ms 后自动复位,但是其可以在帧丢失期间早 30ms 进行自适应复位。在低带信号的重新编码过程中,对部分重构信号 $p_{Lt}(n)$ 的属性进行监控,并控制低带 ADPCM 解码器 1910 的自适应复位。 $p_{Lt}(n)$ 信号在整个丢失过程中被监

控,因此其在第一个丢失帧时被设为零:

$$[0409] \quad \text{sgn}[p_{L_t}(n)] = \begin{cases} \text{sgn}[p_{L_t}(n-1)]+1 & p_{L_t}(n) > 0 \\ \text{sgn}[p_{L_t}(n-1)] & p_{L_t}(n) = 0 \\ \text{sgn}[p_{L_t}(n-1)]-1 & p_{L_t}(n) < 0 \end{cases} \quad (48)$$

[0410] 对于丢失帧,在每帧的基础上监控与恒定信号对比的 $p_{L_t}(n)$ 的属性,因此在每个丢失帧的起初将属性 ($\text{cnst}[\]$) 设为零。其被更新为

$$[0411] \quad \text{cnst}[p_{L_t}(n)] = \begin{cases} \text{cnst}[p_{L_t}(n-1)]+1 & p_{L_t}(n) = p_{L_t}(n-1) \\ \text{cnst}[p_{L_t}(n-1)] & p_{L_t}(n) \neq p_{L_t}(n-1) \end{cases} \quad (49)$$

[0412] 如果满足以下条件,在丢失帧 3 到 5 的结尾复位子带解码器:

$$[0413] \quad \left| \frac{\text{sgn}[p_{L_t}(n)]}{N_{\text{lost}}} \right| > 36 \text{ 或 } \text{cnst}[p_{L_t}(n)] > 40, \quad (50)$$

[0414] 其中 N_{lost} 是丢失帧的数量,即 3、4 或 5。

[0415] c、高带信号的重新编码

[0416] 高带信号 $x_H(n)$ 是使用简化的高带 ADPCM 编码器来进行编码的。简化的高带 ADPCM 编码器 2100 的模块图如图 21 所示。在图 21 中可以看到,标准高带 ADPCM 编码器的自适应量化器已经被删除,因为该算法使用丢包前的移动平均值重写第一个接收帧上的对数比例因子,因而并不需要高带重编码的对数比例因子。高带 ADPCM 编码器 2100 的量化预测误差被非量化预测误差替代了。

[0417] 图 21 的实体是根据它们的 G. 722 高带 ADPCM 子带编码器的等效形式来计算的:

$$[0418] \quad s_{Hz}(n) = \sum_{i=1}^6 b_{H,i}(n-1) \cdot e_H(n-i), \quad (51)$$

$$[0419] \quad s_{Hp}(n) = \sum_{i=1}^2 a_{H,i}(n-1) \cdot x_H(n-i), \quad (52)$$

$$[0420] \quad s_H(n) = s_{Hp}(n) + s_{Hz}(n), \quad (53)$$

$$[0421] \quad e_H(n) = x_H(n) - s_H(n), \text{ 及 } (54)$$

$$[0422] \quad p_H(n) = s_{Hz}(n) + e_H(n). \quad (55)$$

[0423] 零和极点部分的自适应与在 G. 722 编码器中一样发生,如 G. 722 规范的条款 3.6.3 和 3.6.4 中所描述的。

[0424] 与低带重新编码类似,高带 ADPCM 解码器 1920 在帧丢失的 60ms 后自动复位,但是其可以在帧丢失期间早 30ms 进行自适应复位。在高带信号的重新编码过程中,对部分重构信号 $p_H(n)$ 的属性进行监控,并控制高带 ADPCM 解码器 1910 的自适应复位。 $p_H(n)$ 信号在整个丢失过程中被监控,因此其在第一个丢失帧时被设为零:

$$[0425] \quad \text{sgn}[p_H(n)] = \begin{cases} \text{sgn}[p_H(n-1)]+1 & p_H(n) > 0 \\ \text{sgn}[p_H(n-1)] & p_H(n) = 0 \\ \text{sgn}[p_H(n-1)]-1 & p_H(n) < 0 \end{cases} \quad (56)$$

[0426] 对于丢失帧,在每帧基础上监控与恒定信号对比的 $p_H(n)$ 的属性,因此在每个丢失帧的开始将属性 ($\text{cnst}[\]$) 设为零。其被更新为

$$[0427] \quad \text{cnst}[p_H(n)] = \begin{cases} \text{cnst}[p_H(n-1)]+1 & p_H(n) = p_H(n-1) \\ \text{cnst}[p_H(n-1)] & p_H(n) \neq p_H(n-1) \end{cases} \quad (57)$$

[0428] 如果满足以下条件,在丢失帧 3 到 5 的结尾复位子带解码器:

$$[0429] \quad \left| \frac{\text{sgn}[p_H(n)]}{N_{\text{lost}}} \right| > 36 \text{ 或 } \text{cnst}[p_H(n)] > 40. \quad (58)$$

[0430] 5、监控信号特性及其 PLC 的使用

[0431] 以下描述了图 19 的约束和控制逻辑 1970 的功能,用于减少从丢失帧到接收帧的过渡上的残像和失真,从而提高丢包后解码器/PLC 系统 300 的性能。

[0432] a、低带对数比例因子

[0433] 在接收帧期间更新低带对数比例因子 $\nabla_L(n)$ 的特性,并在帧丢失后的第一个接收帧上使用这些特性来自适应地设定针对比例因子的自适应量化器的状态。因而得到一种低带对数比例因子平稳性的度量 (measure),用于确定状态的正确复位。

[0434] i、低带对数比例因子的平稳性

[0435] 在接收帧期间计算和更新低带对数比例因子 $\nabla_L(n)$ 的平稳性,这是基于具恒定泄漏 (leakage) 的 $\nabla_L(n)$ 的一阶移动平均值 $\nabla_{L,m1}(n)$ 的:

$$[0436] \quad \nabla_{L,m1}(n) = 7/8 \cdot \nabla_{L,m1}(n-1) + 1/8 \cdot \nabla_L(n). \quad (59)$$

[0437] 一阶移动平均值的追踪 $\nabla_{L,trck}(n)$ 的度量计算如下

[0438]

$$\nabla_{L,trck}(n) = 127/128 \cdot \nabla_{L,trck}(n-1) + 1/128 \cdot \left| \nabla_{L,m1}(n) - \nabla_{L,m1}(n-1) \right|. \quad (60)$$

[0439] 具有自适应泄漏的二阶移动平均值 $\nabla_{L,m2}(n)$ 根据如下等式 61 计算:

[0440]

$$\nabla_{L,m2}(n) = \begin{cases} 7/8 \cdot \nabla_{L,m2}(n-1) + 1/8 \cdot \nabla_{L,m1}(n) & \nabla_{L,trck}(n) < 3277 \\ 3/4 \cdot \nabla_{L,m2}(n-1) + 1/4 \cdot \nabla_{L,m1}(n) & 3277 \leq \nabla_{L,trck}(n) < 6554 \\ 1/2 \cdot \nabla_{L,m2}(n-1) + 1/2 \cdot \nabla_{L,m1}(n) & 6554 \leq \nabla_{L,trck}(n) < 9830 \\ \nabla_{L,m2}(n) = \nabla_{L,m1}(n) & 9830 \leq \nabla_{L,trck}(n) \end{cases} \quad (61)$$

[0441] 低带对数比例因子的平稳性根据如下等式被衡量为变化度:

[0442]

$$\nabla_{L,chg}(n) = 127/128 \cdot \nabla_{L,chg}(n-1) + 1/128 \cdot 256 \cdot \left| \nabla_{L,m2}(n) - \nabla_{L,m2}(n-1) \right|. \quad (62)$$

[0443] 在丢失帧期间没有更新,换言之:

$$\nabla_{L,m1}(n) = \nabla_{L,m1}(n-1)$$

$$\nabla_{L,trck}(n) = \nabla_{L,trck}(n-1)$$

$$[0444] \quad \nabla_{L,m2}(n) = \nabla_{L,m2}(n-1) \quad (63)$$

$$\nabla_{L,chg}(n) = \nabla_{L,chg}(n-1)$$

[0445] ii、低带自适应量化器的对数比例因子的复位

[0446] 在帧丢失后的第一个接收帧处,根据帧丢失前的平稳性自适应地复位(重写)低带对数比例因子:

[0447]

$$\nabla_L(n-1) \leftarrow \begin{cases} \nabla_{L,m2}(n-1) & \nabla_{L,chg}(n-1) \leq 6554 \\ \frac{\nabla_L(n-1)}{3276} [\nabla_{L,chg}(n-1) - 6554] + \frac{\nabla_{L,m2}(n-1)}{3276} [9830 - \nabla_{L,chg}(n-1)] & 6554 \leq \nabla_{L,chg}(n-1) \leq 9830 \\ \nabla_L(n-1) & 9830 < \nabla_{L,chg}(n-1) \end{cases} \quad (64)$$

[0448] b、高带对数比例因子

[0449] 在接收帧期间更新高带对数比例因子 $\nabla_H(n)$ 的特性,并在帧丢失后的接收帧上使用这些特性来设定自适应量化比例因子的状态。此外,所述特性自适应地控制帧丢失后高带对数比例因子的收敛。

[0450] i、高带对数比例因子的移动平均值和平稳性

[0451] 根据下式计算 $\nabla_H(n)$ 的追踪:

[0452]

$$\nabla_{H,trck}(n) = 0.97 \cdot \nabla_{H,trck}(n-1) + 0.03 \cdot [\nabla_{H,m}(n-1) - \nabla_H(n)]. \quad (65)$$

[0453] 基于追踪,计算具有自适应泄漏的移动平均值如下

[0454]

$$\nabla_{H,m}(n) = \begin{cases} 255/256 \cdot \nabla_{H,m}(n-1) + 1/256 \cdot \nabla_H(n) & |\nabla_{H,trck}(n)| < 1638 \\ 127/128 \cdot \nabla_{H,m}(n-1) + 1/128 \cdot \nabla_H(n) & 1638 \leq |\nabla_{H,trck}(n)| < 3277 \\ 63/64 \cdot \nabla_{H,m}(n-1) + 1/64 \cdot \nabla_H(n) & 3277 \leq |\nabla_{H,trck}(n)| < 4915 \\ 31/32 \cdot \nabla_{H,m}(n-1) + 1/32 \cdot \nabla_H(n) & 4915 \leq |\nabla_{H,trck}(n)| \end{cases} \quad (66)$$

[0455] 该移动平均值用于在第一个接收帧上复位高带对数比例因子,这将在稍后的子节中进行描述。

[0456] 从平均值中计算高带对数比例因子的平稳性的度如下:

$$\nabla_{H,chg}(n) = 127/128 \cdot \nabla_{H,chg}(n-1) + 1/128 \cdot 256 \cdot |\nabla_{H,m}(n) - \nabla_{H,m}(n-1)|. \quad (67)$$

[0458] 该平稳性的度量用于控制帧丢失后 $\nabla_H(n)$ 的重收敛,这将在稍后的子节中进行描述。

[0459] 在丢失帧期间没有更新,换言之:

$$\begin{aligned} \nabla_{H,trck}(n) &= \nabla_{H,trck}(n-1) \\ \nabla_{H,m}(n) &= \nabla_{H,m}(n-1) \\ \nabla_{H,chg}(n) &= \nabla_{H,chg}(n-1) \end{aligned} \quad (68)$$

[0461] ii、高带自适应量化器的对数比例因子的复位

[0462] 在第一个接收帧上将高带对数比例因子复位为丢包前的接收帧的移动平均值:

$$\nabla_H(n-1) \leftarrow \nabla_{H,m}(n-1). \quad (69)$$

[0464] iii、高带自适应量化器的对数比例因子的收敛帧丢失后的高带对数比例因子的收敛是由丢帧前的平稳性的度量 $\nabla_{H,chg}(n)$ 来控制的。对于平稳的情况,在丢包后对 $\nabla_H(n)$ 应用自适应低通滤波器。该低通滤波器应用到 0ms、40ms 或 80ms 上,在此期间低通

滤波的程度逐渐减少。采样的持续时间 N_{LP,∇_H} 根据如下来确定：

$$[0465] \quad N_{LP,\nabla_H} = \begin{cases} 640 & \nabla_{H,chg} < 819 \\ 320 & \nabla_{H,chg} < 1311. \\ 0 & \nabla_{H,chg} \geq 1311 \end{cases} \quad (70)$$

[0466] 低通滤波如下给出：

$$[0467] \quad \nabla_{H,LP}(n) = \alpha_{LP}(n)\nabla_{H,LP}(n-1) + (1 - \alpha_{LP}(n))\nabla_H(n), \quad (71)$$

[0468] 其中系数按如下给出：

$$[0469] \quad \alpha_{LP}(n) = 1 - \left(\frac{n+1}{N_{LP,\nabla_H} + 1} \right)^2, \quad n = 0, 1, \dots, N_{LP,\nabla_H} - 1. \quad (72)$$

[0470] 因此，低通滤波逐个采样的减少了时间 n 。经低通滤波的对数比例因子在 N_{LP,∇_H} 采样期间简单地替换常规的对数比例因子。

[0471] c、低带极点部分

[0472] 在接收帧期间为子带 ADPCM 解码器更新称之为（极点部分的）稳定性裕度 的实体 (entity)，以用于约束帧丢失后的极点部分。

[0473] i、低带极点部分的稳定性裕度

[0474] 低带极点部分的稳定性裕度被定义为

$$[0475] \quad \beta_L(n) = 1 - |a_{L,1}(n)| - |a_{L,2}(n)|, \quad (73)$$

[0476] 其中 $a_{L,1}(n)$ 和 $a_{L,2}(n)$ 是两个极点系数。接收帧期间的稳定性裕度的移动平均值根据下式进行更新：

$$[0477] \quad \beta_{L,MA}(n) = 15/16 \cdot \beta_{L,MA}(n-1) + 1/16 \cdot \beta_L(n) \quad (74)$$

[0478] 在丢失帧期间，不更新移动平均值：

$$[0479] \quad \beta_{L,MA}(n) = \beta_{L,MA}(n-1). \quad (75)$$

[0480] ii、低带极点部分的约束

[0481] 在常规 G. 722 低带（和高带）ADPCM 编码和解码中，要维持 $\beta_{L,\min} = 1/16$ 的最小稳定性裕度。在帧丢失后的最初 40ms 内，为低带 ADPCM 解码器维持增加的最小稳定性裕度，其是帧丢失以来的时间及稳定性裕度的移动平均值的函数。

[0482] 对于最初的三个 10ms 帧，最小稳定性裕度

$$[0483] \quad \beta_{L,\min} = \min \{ 3/16, \beta_{L,MA}(n-1) \} \quad (76)$$

[0484] 设置在帧边界上并在整个帧上执行该最小稳定裕度。在进入到第四个 10ms 帧的帧边界上，执行最小稳定性裕度

$$[0485] \quad \beta_{L,\min} = \min \left\{ 2/16, \frac{1/16 + \beta_{L,MA}(n-1)}{2} \right\} \quad (77)$$

[0486] 而对其它帧执行 $\beta_{L,\min} = 1/16$ 的常规最小稳定性裕度。

[0487] d、高带部分重构信号和高带重构信号

[0488] 在所有帧期间，包括丢失帧和接收帧，保存持高带部分重构信号 $p_H(n)$ 和高带重构信号 $r_H(n)$ 的高通滤波版本：

$$[0489] \quad p_{H,HP}(n) = 0.97 [p_H(n) - p_H(n-1) + p_{H,HP}(n-1)], \text{ 及 } \quad (78)$$

[0490]

$$r_{H,HP}(n) = 0.97[r_H(n) - r_H(n-1) + r_{H,HP}(n-1)]. \quad (79)$$

[0491] 这对应约 40Hz 的 3dB 截断,主要是去除 DC。

[0492] 在帧丢失后的最初 40ms 内,常规的部分重构信号和常规的重构信号被它们各自的高通滤波版本所代替,这分别是为了高带极点部分自适应和高带重构输出。

[0493] 6、时滞计算

[0494] 在此讨论的重新定相和时间扭曲技术需要有丢帧隐藏波形 $x_{PLC}(j)$ 和第一个接收帧中的信号未对齐的采样数量。

[0495] a、低子带重构信号的低复杂度估计

[0496] 在第一个接收帧内用于计算时滞的信号是通过使用极零滤波系数 ($a_{Lpwe,i}(159)$, $b_{Lpwe,i}(159)$) 和从 STATE₁₅₉ 获得的其它的必要状态信息对低子带截断差分信号 $d_{Lt}(n)$ 进行滤波来获得的:

$$r_{Le}(n) = \sum_{i=1}^2 a_{Lpwe,i}(159) \cdot r_{Le}(n-i) + \sum_{i=1}^6 b_{Lpwe,i}(159) \cdot d_{Lt}(n-i) + d_{Lt}(n),$$

[0498] $n = 0, 1, \dots, 79$. (80)

[0499] 这一功能由图 18 的模块 1820 来执行。

[0500] b、重新定相和时间扭曲需求的确定

[0501] 如果最后一个接收帧是无声的,如优值系数所表示的,时滞 T_L 被设定为:

[0502] 如果 $\text{merit} \leq \text{MLO}$, $T_L = 0$. (81)

[0503] 另外,如果第一个接收帧是无声的,如标准化第一自相关系数所表示:

$$r(1) = \frac{\sum_{n=0}^{78} r_{Le}(n) \cdot r_{Le}(n)}{\sum_{n=0}^{78} r_{Le}(n) \cdot r_{Le}(n+1)}, \quad (82)$$

[0505] 时滞被设定为零:

[0506] 如果 $r(1) < 0.125$, $T_L = 0$. (83)

[0507] 否则,时滞的计算如以下部分所解释。所述时滞的计算是由图 18 的模块 1850 来执行的。

[0508] c、时滞的计算

[0509] 时滞的计算包括以下步骤:(1) 生成外插信号;(2) 粗时滞搜索;及(3) 细化时滞搜索。这些将在以下的子节中进行描述。

[0510] i、外插信号的生成

[0511] 时滞表示 $x_{PLC}(j)$ 和 $r_{Le}(n)$ 之间的偏差。为了计算这个偏差,将 $x_{PLC}(j)$ 延伸到第一个接收帧,并将标准化交叉相关性函数最大化。这个子节描述了如何外插 $x_{PLC}(j)$,并详细说明了所需信号的长度。假设 $x_{PLC}(j)$ 被复制到 $x_{out}(j)$ 缓存中。因为这是一个类型 5 的帧(第一个接收帧),所以假设相应的是:

[0512] $x_{out}(j-160) = x_{PLC}(j)$, $j = 0, 1, \dots, 159$ (84)

[0513] 搜索相关性(correlation)的范围如下:

[0514]

$$\Delta_{TL} = \min(\lfloor ppfe \cdot 0.5 + 0.5 \rfloor + 3, \Delta_{TLMAX}), \quad (85)$$

[0515] 其中 $\Delta_{TLMAX} = 28$, $ppfe$ 是在产生 $x_{PLC}(j)$ 时使用的周期性波形外插的基音周期。滞后搜索的窗口大小 (16kHz 采样率下) 如下:

[0516]

$$LSW_{16k} = \begin{cases} 80 & \lfloor ppfe \cdot 1.5 + 0.5 \rfloor < 80 \\ 160 & \lfloor ppfe \cdot 1.5 + 0.5 \rfloor > 160 \\ \lfloor ppfe \cdot 1.5 + 0.5 \rfloor & \text{其它} \end{cases} \quad (86)$$

[0517] 指定 8kHz 采样率下的滞后搜索窗口 LSW 是很用的, 如下:

[0518]

$$LSW = \lfloor LSW_{16k} \cdot 0.5 \rfloor \quad (87)$$

[0519] 如上给出的, 需要从 $x_{PLC}(j)$ 得到的外插信号的总长度为:

[0520] $L = 2 \cdot (LSW + \Delta_{TL})$. (88)

[0521] 外插信号相对于接收帧中第一个采样的起始位置为:

[0522] $D = 12 - \Delta_{TL}$. (89)

[0523] 外插信号 $es(j)$ 是根据如下方法重构的:

[0524] 如果 $D < 0$

[0525] $es(j) = x_{out}(D+j)$ $j = 0, 1, \dots, -D-1$

[0526] 如果 $(L+D \leq ppfe)$

[0527] $es(j) = x_{out}(-ppfe+D+j)$ $j = -D, -D+1, \dots, L-1$

[0528] 否则

[0529] $es(j) = x_{out}(-ppfe+D+j)$ $j = -D, -D+1, \dots, ppfe-D-1$

[0530] $es(j) = es(j-ppfe)$ $j = ppfe-D, ppfe-D+1, \dots, L-1$

[0531] 否则

[0532]

$$ovs = ppfe \cdot \lceil D / ppfe \rceil - D$$

[0533] 如果 $(ovs \geq L)$

[0534] $es(j) = x_{out}(-ovs+j)$ $j = 0, 1, \dots, L-1$

[0535] 否则

[0536] 如果 $(ovs > 0)$

[0537] $es(j) = x_{out}(-ovs+j)$ $j = 0, 1, \dots, ovs-1$

[0538] 如果 $(L-ovs \leq ppfe)$

[0539] $es(j) = x_{out}(-ovs-ppfe+j)$ $j = ovs, ovs+1, \dots, L-1$

[0540] 否则

[0541] $es(j) = x_{out}(-ovs-ppfe+j)$ $j = ovs, ovs+1, \dots, ovs+ppfe-1$

[0542] $es(j) = es(j-ppfe)$ $j = ovs+ppfe, ovs+ppfe+1, \dots, L-1$.

[0543] ii、粗时滞搜索

[0544] 粗略估计的时滞 T_{LSUB} 首先通过搜索子采样标准化交叉相关性函数 $R_{SUB}(k)$ 的峰值来计算得到:

[0545]

$$R_{SUB}(k) = \frac{\sum_{i=0}^{LSW/2-1} es(4i-k+\Delta_{TL}) \cdot r_{Le}(2i)}{\sqrt{\sum_{i=0}^{LSW/2-1} es^2(4i-k+\Delta_{TL}) \sum_{i=0}^{LSW-1} r_{Le}^2(2i)}}, \quad k = -\Delta_{TL}, -\Delta_{TL} + 4, -\Delta_{TL} + 8, \dots, \Delta_{TL}$$
(90)

[0546] 为了在细化时避免搜索超出边界,调整 T_{LSUB} 如下:[0547] 如果 $(T_{LSUB} > \Delta_{TLMAX}-4)$ $T_{LSUB} = \Delta_{TLMAX}-4$ (91)[0548] 如果 $(T_{LSUB} < -\Delta_{TLMAX}+4)$ $T_{LSUB} = -\Delta_{TLMAX}+4$. (92)

[0549] iii、细化时滞搜索

[0550] 然后通过搜索 $R(k)$ 的峰值细化搜索以给出时滞 T_L , $R(k)$ 如下:

[0551]

$$R(k) = \frac{\sum_{i=0}^{LSW-1} es(2i-k+\Delta_{TL}) \cdot r_{Le}(i)}{\sqrt{\sum_{i=0}^{LSW-1} es^2(2i-k+\Delta_{TL}) \sum_{i=0}^{LSW-1} r_{Le}^2(i)}}, \quad k = -4 + T_{LSUB}, -2 + T_{LSUB}, \dots, 4 + T_{LSUB}$$
(93)

[0552] 最后,检查以下条件:

[0553] 如

果

$$\text{如果 } \sum_{i=0}^{LSW-1} r_{Le}^2(i) = 0 \quad (94)$$

$$\text{Or } \sum_{i=0}^{LSW-1} es(2i-T_L+\Delta_{TL}) \cdot r_{Le}(i) \leq 0.25 \cdot \sqrt{\sum_{i=0}^{LSW-1} r_{Le}^2(i)} \quad (95)$$

[0555] Or $(T_L > \Delta_{TLMAX}-2) \parallel (T_L < -\Delta_{TLMAX}+2)$ (96)[0556] 那么 $T_L = 0$.

[0557] 7、重新定相

[0558] 重新定相是将内部状态设定为丢帧隐藏波形 $x_{PLC}(j)$ 与第一个接收帧前的最后一个输入信号采样同相时的状态的处理过程。重新定相可以分成以下几个步骤:(1) 在丢帧的重新编码过程中存储中间 G. 722 状态;(2) 根据时滞调整重新编码;及(3) 更新 QMF 合成滤波存储器。以下的子部分将描述这些步骤的更多细节。重新定相是由图 18 的模块 1810 来执行的。

[0559] a、在重新编码过程中存储中间 G. 722 状态

[0560] 如在本申请其它地方所描述的,在丢失帧期间重新编码重构信号 $x_{PLC}(j)$ 以更新 G. 722 解码器状态存储器。假设 $STATE_j$ 是对 $x_{PLC}(j)$ 的第 j 个采样重新编码后的 G. 722 状态和 PLC 状态。那么除在帧边界上的 G. 722 状态(将被正常维持,即 $STATE_{159}$)之外,还存储了 $STATE_{159-\Delta_{TLMAX}}$ 。为了促进重新定相,还存储了子带信号:

[0561] $x_L(n), x_H(n) \quad n = 69 - \Delta_{TLMAX}/2 \dots 79 + \Delta_{TLMAX}/2$

[0562] b、根据时滞调整重新编码

[0563] 依据时滞的符号,调整重新编码的过程如下:

[0564] 如果 $\Delta_{TL} > 0$

[0565] 1、恢复 G. 722 状态和 PLC 状态到 $STATE_{159-\Delta_{TLMAX}}$

[0566] 2、以前述方式重新编码 $x_L(n)$ 、 $x_H(n)$ ($n = 80 - \Delta_{TLMAX}/2 \dots 79 - \Delta_{TL}/2$)

[0567] 如果 $\Delta_{TL} < 0$

[0568] 1、恢复 G. 722 状态和 PLC 状态到 $STATE_{159}$

[0569] 2、以前述方式重新编码 $x_L(n)$ 、 $x_H(n)$ ($n = 80 \dots 79 + |\Delta_{TL}/2|$)

[0570] 注意,为了促进 $x_L(n)$ 和 $x_H(n)$ 的重新编码一直到 $n = 79 + |\Delta_{TL}/2|$,需要 $x_{PLC}(j)$ 的一直到 $\Delta_{TLMAX} + 182$ 个采样。

[0571] c、更新 QMF 合成滤波器存储器

[0572] 在第一个接收帧上,由于在丢失帧期间因为 PLC 发生在 16kHz 输出语音域内, QMF 合成滤波器组是不活跃的,所以需要计算 QMF 合成滤波器存储器。时间上来说,该存储器一般对应最后一个丢失帧的最后一个采样。然而,需要将重新定相考虑进去。根据 G. 722, QMF 合成滤波器存储器给定如下:

[0573] $x_d(i) = r_L(n-i) - r_H(n-i)$, $i = 1, 2, \dots, 11$, 及 (97)

[0574] $x_s(i) = r_L(n-i) + r_H(n-i)$, $i = 1, 2, \dots, 11$ (98)

[0575] 第一个接收帧的最初两个输出采样计算如下:

[0576]
$$x_{out}(j) = 2 \sum_{i=0}^{11} h_{2i} \cdot x_d(i), \text{ 及 (99)}$$

[0577]
$$x_{out}(j+1) = 2 \sum_{i=0}^{11} h_{2i+1} \cdot x_s(i). \quad (100)$$

[0578] 滤波器存储器 (即 $x_d(i)$ 和 $x_s(i)$ ($i = 1, 2, \dots, 11$)) 是根据在重新编码 $x_L(n)$ 和 $x_H(n)$ ($n = 69 - \Delta_{TL}/2, 69 - \Delta_{TL}/2 + 1, \dots, 79 - \Delta_{TL}/2$) (即一直到重新定相点的最后一些采样) 时给简化子带 ADPCM 编码器的经重新定相的输入中的最后 11 个采样计算出的:

[0579] $x_d(i) = x_L(80 - \Delta_{TL}/2 - i) - x_H(80 - \Delta_{TL}/2 - i)$, $i = 1, 2, \dots, 11$, 及 (101)

[0580] $x_s(i) = x_L(80 - \Delta_{TL}/2 - i) + x_H(80 - \Delta_{TL}/2 - i)$, $i = 1, 2, \dots, 11$, (102)

[0581] 其中 $x_L(n)$ 和 $x_H(n)$ 在丢失帧期间已经被存储在状态存储器中。

[0582] 8、时间扭曲

[0583] 时间扭曲是沿时间轴拉伸或收缩信号的处理。以下描述了如何对 $x_{out}(j)$ 进行时间扭曲以改善与周期性波形外插信号 $x_{PLC}(j)$ 的对齐。仅当 $T_L \neq 0$ 时执行该算法。时间扭曲由图 18 的模块 1860 来执行。

[0584] a、时滞细化

[0585] 通过取叠加窗口中的交叉相关性的最大值来细化用于时间扭曲的时滞 T_L 。基于 T_L 估计出的第一个接收帧中的叠加窗口的起始位置如下:

[0586] $SP_{OLA} = \max(0, MIN_UNSTBL - T_L)$, (103)

[0587] 其中 $MIN_UNSTBL = 16$ 。

[0588] 相对于 SP_{OLA} 的外插信号的起始位置如下:

[0589] $D_{ref} = SP_{OLA} - T_L - RSR$, (104)

[0590] 其中 $RSR = 4$ 是细化搜索范围。

[0591] 外插信号的所需长度如下：

$$[0592] \quad L_{\text{ref}} = \text{OLALG} + \text{RSR}. \quad (105)$$

[0593] 外插信号 $es_{\text{tw}}(j)$ 是使用如 D. 6. c. i 节所述的相同过程来获得的, 除了 $\text{LSW} = \text{OLALG}$, $L = L_{\text{ref}}$ 及 $D = D_{\text{ref}}$ 。

[0594] 细化滞后 T_{ref} 是通过搜索以下的峰值来获得的：

[0595]

$$R(k) = \frac{\sum_{i=0}^{\text{OLALG}-1} es_{\text{tw}}(i-k+\text{RSR}) \cdot x_{\text{out}}(i+SP_{\text{OLA}})}{\sqrt{\sum_{i=0}^{\text{OLALG}-1} es_{\text{tw}}^2(i-k+\text{RSR}) \sum_{i=0}^{\text{OLALG}-1} x_{\text{out}}^2(i+SP_{\text{OLA}})}}, \quad k = -\text{RSR}, -\text{RSR}+1, \dots, \text{RSR}. \quad (106)$$

[0596] 然后获得用于时间扭曲的最终时滞：

$$[0597] \quad T_{\text{Lwarp}} = T_L + T_{\text{ref}}. \quad (107)$$

[0598] b、计算经时间扭曲的 $x_{\text{out}}(j)$ 信号

[0599] 信号 $x_{\text{out}}(j)$ 经过 T_{Lwarp} 个采样的时间扭曲, 形成随后与波形外插信号 $es_{\text{old}}(j)$ 叠加的信号 $x_{\text{warp}}(j)$ 。图 22A、图 22B 和图 22C 的时间线 2200、2220 和 2240 分别示出了依据 T_{Lwarp} 值的三种情况。在图 22A 中, $T_{\text{Lwarp}} < 0$, $x_{\text{out}}(j)$ 经历收缩或压缩。 $x_{\text{out}}(j)$ 的第一个 MIN_UNSTBL 采样在扭曲过程中不用于创建 $x_{\text{warp}}(j)$ 且 $x_{\text{start}} = \text{MIN_UNSTBL}$ 。在图 22B 中, $0 \leq T_{\text{Lwarp}} \leq \text{MIN_UNSTBL}$, 对 $x_{\text{out}}(j)$ 进行 T_{Lwarp} 个采样的拉伸。同样, $x_{\text{out}}(j)$ 的第一个 MIN_UNSTBL 未被使用且 $x_{\text{start}} = \text{MIN_UNSTBL}$ 。在图 22C 中, $T_{\text{Lwarp}} \geq \text{MIN_UNSTBL}$, 再一次对 $x_{\text{out}}(j)$ 进行 T_{Lwarp} 个采样的拉伸。然而, 因为在扭曲过程中会创建额外的 T_{Lwarp} 个采样, 所以在这种情况下不需要 $x_{\text{out}}(j)$ 的头 T_{Lwarp} 个采样; 因而, $x_{\text{start}} = T_{\text{Lwarp}}$ 。

[0600] 在每种情况下, 每次叠加 / 下降的采样数量如下：

$$[0601] \quad \text{spad} = \frac{(160 - x_{\text{start}})}{|T_{\text{Lwarp}}|}. \quad (108)$$

[0602] 扭曲是通过分段式 (piece-wise) 单采样平移和三角叠加来实现的, 从 $x_{\text{out}}[x_{\text{start}}]$ 开始。为了执行收缩, 周期性地降低采样。从采样降低的点开始, 将原始信号和向左平移的信号 (由于降低) 进行叠加。为了执行拉伸, 周期性地重复采样。从采样重复的点开始, 将原始信号和向右平移的信号 (由于采样重复) 进行叠加。叠加窗口的长度 L_{olawarp} (注意: 这不同于图 22A、22B 和 22C 中所示的 OLA 区域) 取决于采样增加 / 降低的周期性, 如下：

$$[0603] \quad \text{如果 } T_{\text{Lwarp}} < 0, L_{\text{olawarp}} = \frac{(160 - x_{\text{start}} - |T_{\text{Lwarp}}|)}{|T_{\text{Lwarp}}|}$$

$$[0604] \quad \text{否则 } L_{\text{olawarp}} = \lceil \text{spad} \rceil \quad (109)$$

$$[0605] \quad L_{\text{olawarp}} = \min(8, L_{\text{olawarp}}).$$

[0606] 经扭曲的输入信号 x_{warp} 的长度如下：

$$[0607] \quad L_{\text{xwarp}} = \min(160, 160 - \text{MIN_UNSTBL} + T_{\text{Lwarp}}). \quad (110)$$

[0608] c、计算波形外插信号

[0609] 如图 22A、22B 和 22C 所示,在第一个接收帧中将扭曲信号 x_{warp} 和外插信号 $es_{\text{ola}}(j)$ 进行叠加。按照如下两个步骤的处理可在 $x_{\text{out}}(j)$ 信号缓存中直接产生外插信号 $es_{\text{ola}}(j)$:

[0610] 步骤 1

$$[0611] \quad es_{\text{ola}}(j) = x_{\text{out}}(j) = \text{ptfe} \cdot x_{\text{out}}(j - \text{ppfe}) \quad j = 0, 1, \dots, 160 - L_{\text{xwarp}} + 39 \quad (111)$$

[0612] 步骤 2

$$[0613] \quad x_{\text{out}}(j) = x_{\text{out}}(j) \cdot w_i(j) + \text{ring}(j) \cdot w_o(j) \quad j = 0, 1, \dots, 39, \quad (112)$$

[0614] 其中 $w_i(j)$ 和 $w_o(j)$ 是长度为 40 的三角形斜升和斜降叠加窗口, $\text{ring}(j)$ 是以本申请中其它地方所述的方式计算出的呼叫信号。

[0615] d、时间扭曲信号与波形外插信号的叠加

[0616] 在前一段中计算出的外插信号与扭曲信号 $x_{\text{warp}}(j)$ 进行叠加,如下:

$$[0617] \quad x_{\text{out}}(160 - L_{\text{xwarp}} + j) = x_{\text{out}}(160 - L_{\text{xwarp}} + j) \cdot w_o(j) + x_{\text{warp}}(j) \cdot w_i(j), \quad j = 0, 1, \dots, 39.$$

[0618] (113)

[0619] 然后 $x_{\text{warp}}(j)$ 的剩余部分被简单复制到信号缓存中:

$$[0620] \quad x_{\text{out}}(160 - L_{\text{xwarp}} + j) = x_{\text{warp}}(j), \quad j = 40, 41, \dots, L_{\text{xwarp}} - 1. \quad (114)$$

[0621] E、基于子带语音波形外插的子带预测编码器的丢包隐藏

[0622] 如图 23 中的解码器 /PLC 系统 2300 所示,为本发明的一个可选实施例。上述为解码器 /PLC 系统 300 研究的大部分技术也可用于这第二个示例性实施例。解码器 PLC 系统 2300 和解码器 /PLC 系统 300 的主要区别是在子带语音信号域内执行语音信号波形外插,而不是在全带语音信号域。

[0623] 如图 23 所示,解码器 /PLC 系统 2300 包括比特流解复用器 2310、低带 ADPCM 解码器 2320、低带语音信号合成器 2322、切换器 2336 和 QMF 合成滤波器组 2340。比特流解复用器 2310 本质上与图 2 的比特流解复用器 210 相同,QMF 合成滤波器组 2340 本质上与图 2 的 QMF 合成滤波器组 240 相同。

[0624] 与图 3 的解码器 /PLC 系统 300 一样,解码器 /PLC 系统 2300 以依据帧类型的方式处理帧,并且使用与上述图 5 中相同的帧类型。

[0625] 在处理类型 1 的帧时,解码器 /PLC 系统 2300 执行标准 G. 722 解码。在这种操作模式下,解码器 /PLC 系统 2300 的模块 2310、2320、2330 和 2340 分别执行与传统的 G. 722 解码器 200 的对应模块 210、220、230 和 240 相同的功能。具体地说,比特流解复用器 2310 将输入比特流分离成低带比特流和高带比特流。低带 ADPCM 解码器 2320 将低带比特流解码成解码低带语音信号。切换器 2326 连接到标记为“类型 1”的上面位置,从而将解码低带语音信号连接到 QMF 合成滤波器组 2340。高带 ADPCM 解码器 2330 将高带比特流解码成解码高带语音信号。切换器 2336 也连接到标记为“类型 1”的上面位置,从而将解码高带语音信号连接到 QMF 合成滤波器组 2340。然后 QMF 合成滤波器组 2340 将解码低带语音信号和解码高带语音信号重新组合成全带输出语音信号。

[0626] 因此,在处理类型 1 的帧时,解码器 /PLC 系统相当于图 2 的解码器 200,区别是为了后续丢失帧中可能的使用,解码低带语音信号被存储在低带语音信号合成器 2322 中,同样为了后续丢失帧中可能的使用,解码高带语音信号被存储在高带语音信号合成器 2332 中。预计执行 PLC 操作的其它状态更新和处理也可以被执行。

[0627] 在处理类型 2、类型 3 和类型 4 的帧（丢失帧）时，从与先前帧关联的存储的子带语音信号中对每个子带的解码语音信号进行外插，以填补与当前丢失帧关联的波形间隙。这种波形外插是由低带语音信号合成器 2322 和高带语音信号合成器 2332 来执行的。执行模块 2322 和 2332 的波形外插功能存在许多现有技术。例如，可使用在申请号为 11/234, 291、发明人为陈、申请时间为 2005 年 9 月 26 日、名称为“独立块语音编解码器的丢包隐藏技术”的美国专利中描述的技术，或者使用那些技术的修改版本，如涉及上述图 3 的解码器 /PLC 系统 300 的技术。

[0628] 在处理类型 2、类型 3 或类型 4 的帧时，切换器 2326 和 2336 位于标记为“类型 2-6”的下面位置。因而，它们将合成低带音频信号和合成高带音频信号连接到 QMF 合成滤波器组 2340，进而将它们重新组合成针对当前丢失帧的合成输出语音信号。

[0629] 与解码器 /PLC 系统 300 类似，紧随坏帧（类型 5 和类型 6 的帧）的最初几个接收帧需要特殊的处理来最小化由于不匹配 G. 722 状态导致的语音质量下降，以及保证从最后一个丢失帧中的外插语音信号波形到紧随最后一个坏帧的最初几个好帧中的解码语音信号波形的平滑过渡。因而，在处理这些帧时，切换器 2326 和 2336 保持在标为“类型 2-6”的下面位置，以便来自低带 ADPCM 解码器 2320 的解码低带语音信号可以在提供到 QMF 合成滤波器组 2340 之前由低带语音信号合成器 2322 进行修改，以及来自高带 ADPCM 解码器 2330 的解码高带语音信号可以在提供到 QMF 合成滤波器组 2340 之前由高带语音信号合成器 2332 进行修改。

[0630] 所属领域的技术人员应知悉，在以上子节 C 和 D 中针对包丢失后最初几个帧描述的大部分技术也可以很容易地用于包丢失后的最初几个帧的特殊处理的这种示例性实施例中。例如，解码约束和控制逻辑（图 23 中未示出）也包括在解码器 /PLC 系统 2300 中，用于以上述参照解码器 /PLC 系统 300 的类似方式来约束和控制由低带 ADPCM 解码器 2320 和高带 ADPCM 解码器 2330 在处理类型 5 和类型 6 的帧时执行的解码操作。同样，每个子带语音信号合成器 2322 和 2332 用于执行重新定相和时间扭曲技术，如上述参照解码器 /PLC 系统 300 描述的那些技术。因为在前面部分提供了这些技术的完整描述，所以在此不需要重复在解码器 /PLC 系统 2300 的环境中使用的这些技术的描述。

[0631] 与解码器 /PLC 系统 300 相比，解码器 /PLC 系统 2300 的主要优点是其具有更低的复杂度。这是因为在子带域中外插语音信号排除了采用 QMF 合成滤波器组将全带外插语音信号分离成子带语音信号的需要，如在第一个示例性实施例中完成的。然而，在全带域中外插语音信号也具有其优点，以下将对此进行阐述。

[0632] 当图 23 中的系统 2300 外插高带语音信号时，存在一些潜在的问题。首先，如果其不对高带语音信号执行周期性波形外插，那么输出语音信号将不保留可能在一些高周期性有声信号中出现的高带语音信号的周期性特性。换言之，如果其对高带语音信号执行周期性波形外插，即使其使用用于外插低带语音信号的相同基音周期来减少计算并保证两个子带语音信号正使用相同的基音周期用于外插，仍然存在另一问题。当周期性外插高带语音信号时，外插高带语音信号将是周期性的，并在频谱上具有谐波结构。换言之，高带语音信号的频谱中的频谱峰值的频率将与整数倍相关。然而，一旦合成滤波器组 2340 将高带语音信号与低带语音信号重新组合，高带语音信号的频谱将被“译成”或移位为较高的频率，同时可能发生镜像，这取决于所用的 QMF 合成滤波器组。因而，在这种镜像和频率偏移之后，

并不能保证全带输出语音信号的高带部分中的频谱峰值仍具有低带语音信号中的基音频率的整数倍的频率。这可能潜在地导致高周期性有声信号的输出音频质量的下降。相反,图 3 中的系统 300 并不存在这个问题。因为系统 300 在全带域中执行音频信号外插,所以保证了高带中的谐波峰值的频率是基音频率的整数倍。

[0633] 总之,解码器 /PLC 系统 300 的优点是对于有声信号,外插全带语音信号将在整个语音带上保留频谱峰值的谐波结构。换言之,解码器 /PLC 系统 2300 具有低复杂度的优点,但是其不可以在较高的子带中保留这种谐波结构。

[0634] F、硬件和软件实现

[0635] 为了保证完整,本发明提供了通用的计算机系统的以下描述。本发明可以在硬件或软硬件的组合中实现。因此,本发明可在计算机系统或其它处理系统的环境中实现。图 24 示出了这种计算机系统 2400 的一个示例。在本发明中,以上在 C、D 和 E 节中描述的所有解码和 PLC 操作可在一个或多个不同的计算机系统 2400 上执行,以实现本发明的各种方法。

[0636] 计算机系统 2400 包括一个或多个处理器,如处理器 2404。处理器 2404 可以是特殊用途或通用的数字信号处理器。处理器 2404 连接到通信架构 2402(例如,总线或网络)。各种软件实现是按照这个示例性计算机系统来描述的。在阅读完这个叙述之后,所属领域的技术人员将很容易理解如何使用其它计算机系统和 / 或计算机结构实现本发明。

[0637] 计算机系统 2400 还包括主存储器 2406,优选是随机存储器 (RAM),还可包括第二存储器 2420。第二存储器 2420 可包括,例如,硬盘驱动器 2422 和 / 或移动存储驱动器 2424,代表性的有软盘驱动器、磁带驱动器、光盘驱动器等等。移动存储驱动器 2424 以众所周知的方式读和 / 或写移动存储单元 2428。移动存储单元 2428 有软盘、磁带、光盘等等,其由移动存储驱动器 2424 进行读和写。应知悉,移动存储单元 2428 包括已将计算机软件和 / 数据存储在其中的计算机使用存储媒介。

[0638] 在可选的实现中,第二存储器 2420 可包括其它类似的装置,用于将计算机程序或其它指令载入计算机系统 2400。这种装置也包括,例如,移动存储单元 2430 和接口 2426。这种装置的例子包括程序盒 (cartridge) 和盒式接口 (如在视频游戏设置中使用的)、移动存储芯片 (如 EPROM 或 PROM) 和相关插槽、其它使数据从移动存储单元 2430 传输到计算机系统 2400 的移动存储单元 2430 和接口 2426。

[0639] 计算机系统 2400 还包括通信接口 2440。通信接口 2440 使软件和数据在计算机系统 2400 和外部设置之间传输。通信接口 2440 的示例包括调制解调器、网络接口 (如以太网卡)、通信端口、PCMCIA 槽和卡等。通过通信接口 2440 传输的软件和数据的形式可以是电、电磁、光或其它能够由通信接口 2440 接收的信号。通过通信通道 2442 可将这些信号提供到通信接口 2440。通信通道 2442 传送信号,其可使用电线或电缆、光纤、电话线、移动电话连接、RF 连接和其它通信信道来实现。

[0640] 在此所用的术语“计算机程序媒介”和“计算机可使用媒介”通常用于指的是媒介,如移动存储单元 2428、安装在硬盘驱动器 2422 上的硬盘及由通信接口 2440 接收的信号。这些计算机程序产品是用于提供软件给计算机系统 2400 的手段。

[0641] 计算机程序 (也称为计算机控制逻辑) 被存储在主存储器 2406 和 / 或第二存储器 2420 中。计算机程序也可通过通信接口 2440 进行接收。这种计算机程序在被执行时,

使计算机系统 2400 实现了本发明,如在此讨论的。具体地说,计算机程序被执行时,使处理器 2400 实现了本发明的处理,如在此讨论的任何方法。因此,这种计算机程序表示计算机系统 2400 的控制器。在使用软件实现本发明时,可以将软件存储在计算机程序产品中并使用移动存储驱动器 2424、接口 2426 或通信接口 2440 将软件载入到计算机系统 2400 中。

[0642] 在另一实施例中,本发明的特征主要是在硬件中实现,例如,使用硬件器件,如专用集成电路 (ASIC) 和门矩阵。使用硬件状态机执行在此所述的功能的实现对于所属领域的技术人员来说也是显而易见的。

[0643] G、结论

[0644] 虽然以上描述了本发明的各种实施例,应当理解,其目的仅在于举例说明,而没有限制性。本领域的技术人员知悉,在不离开本发明的精神和范围情况下,在形式上和细节上还可做各种的改变。因此,本发明的保护范围不当仅局限于以上描述的任一实施例,而应该依照权利要求及其等效替换来限定。

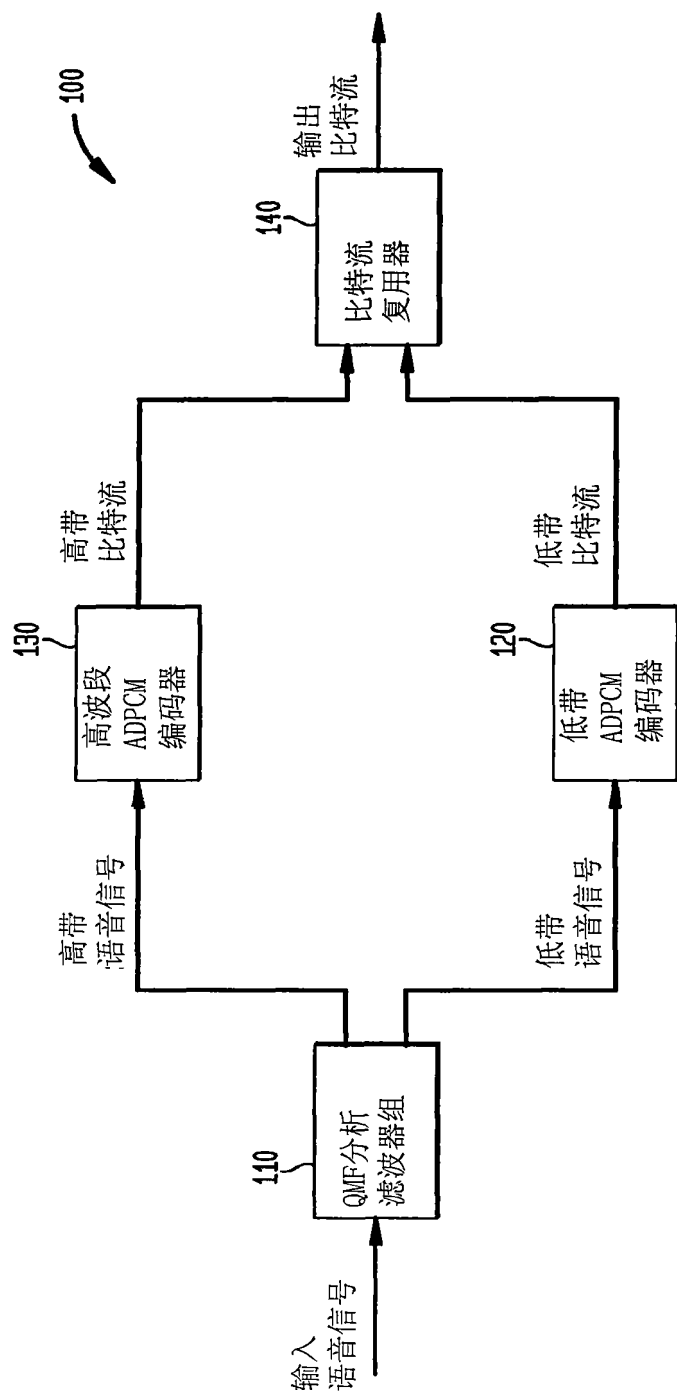


图 1
(现有技术)

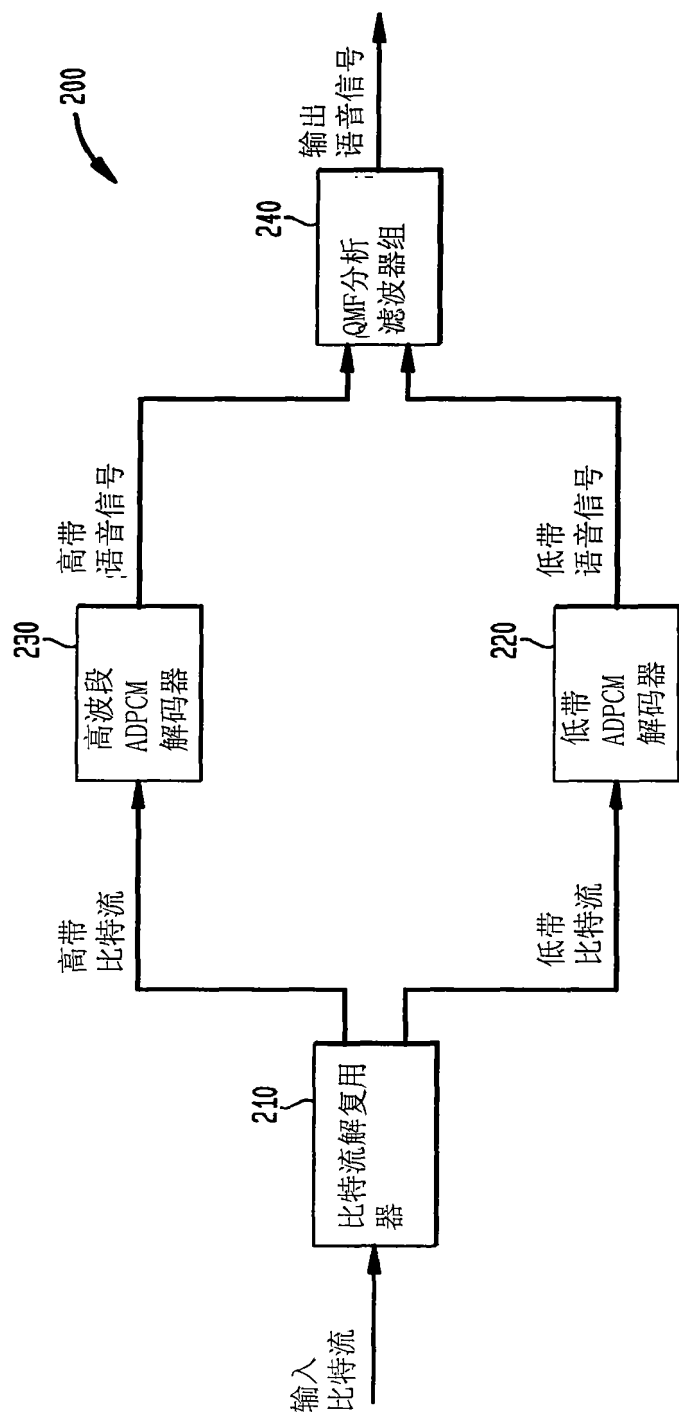


图 2
(现有技术)

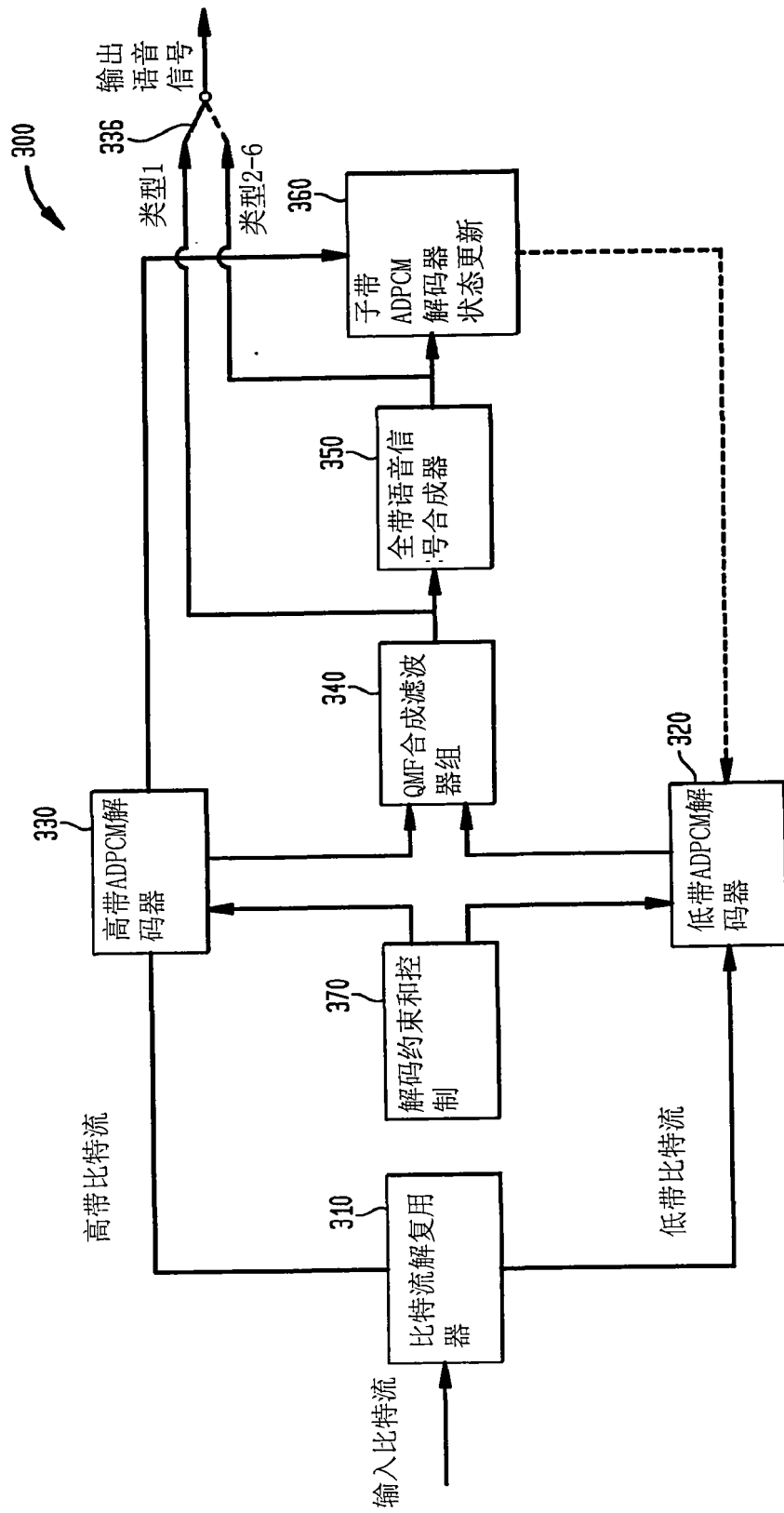


图 3

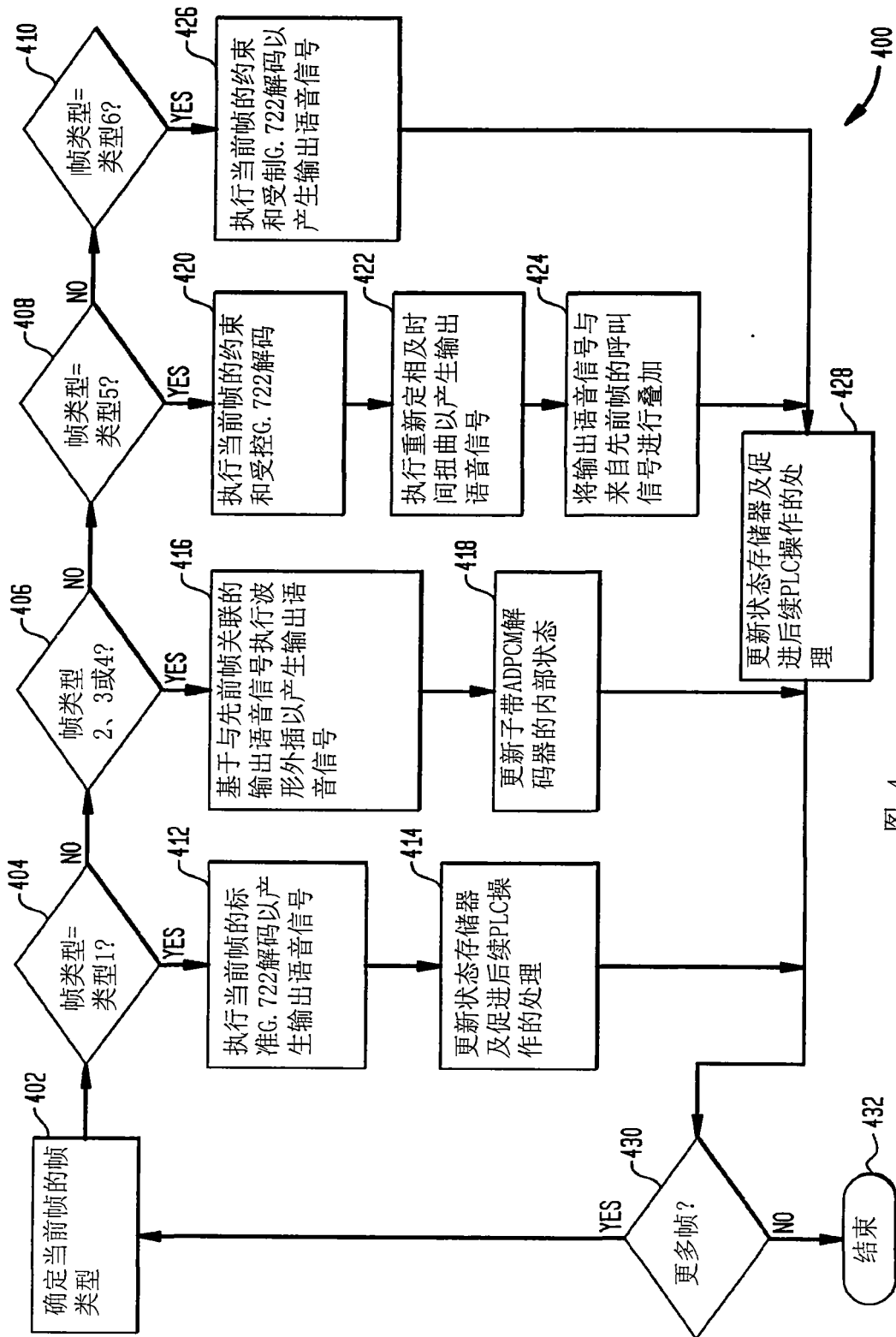


图 4

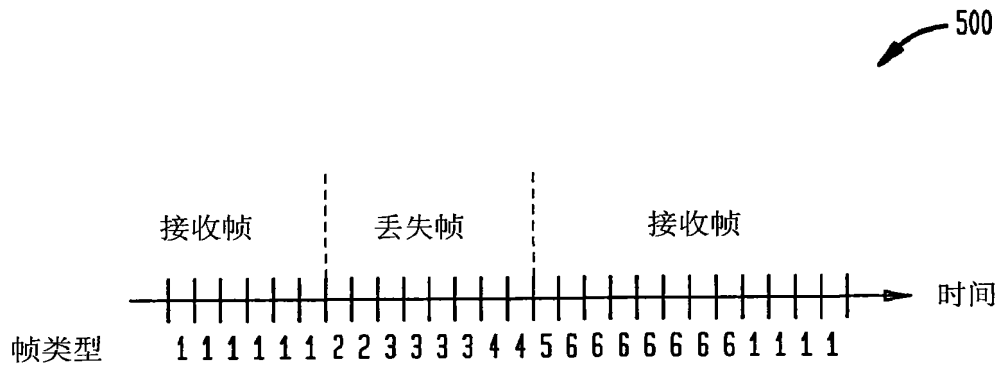


图 5

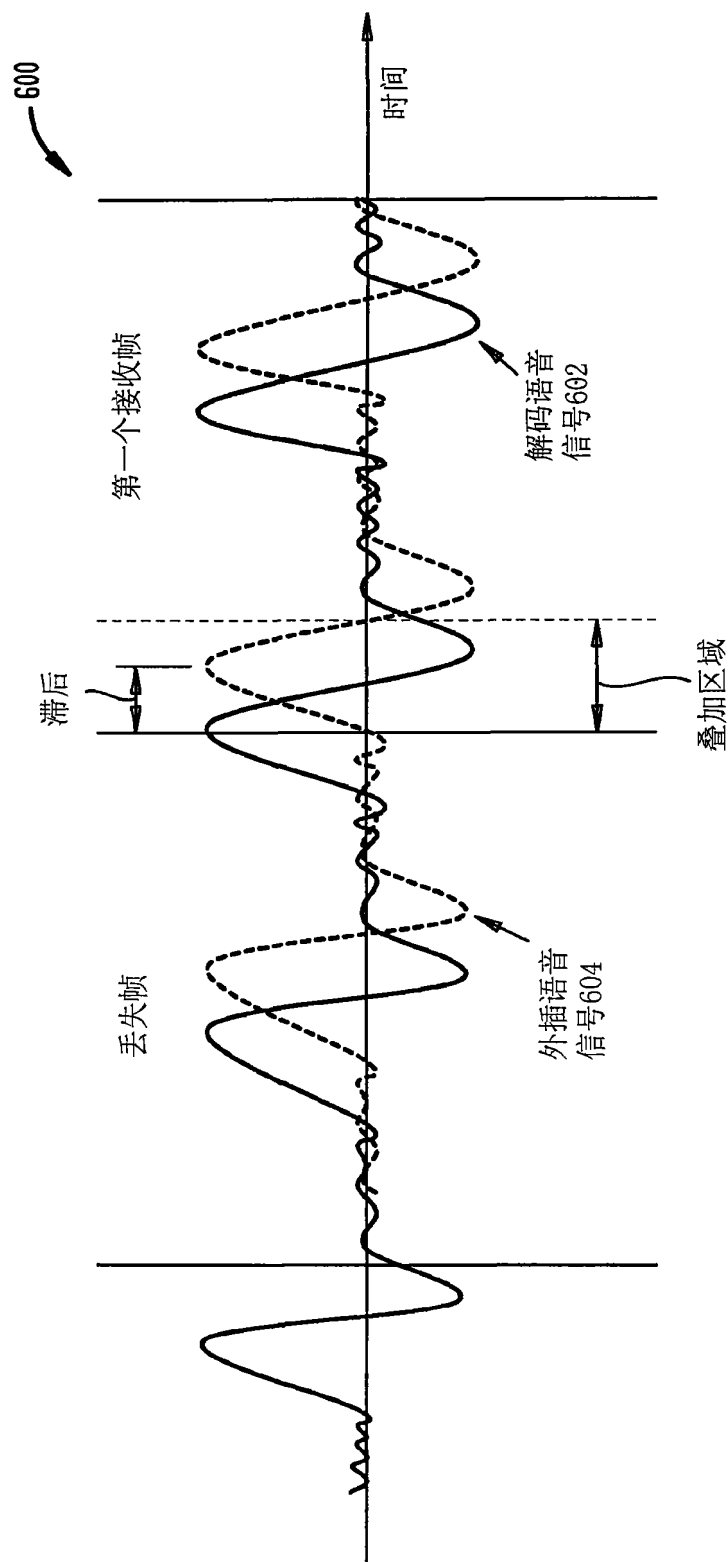


图 6

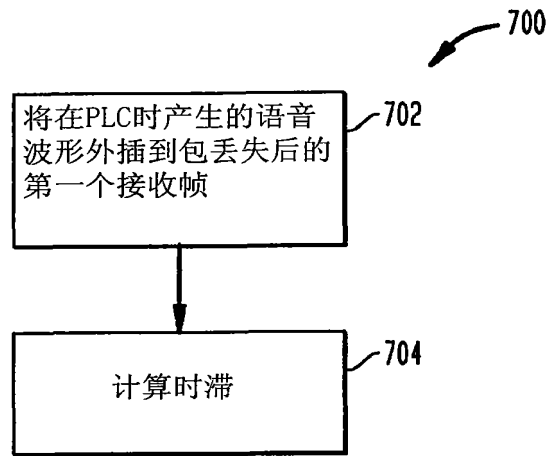


图 7

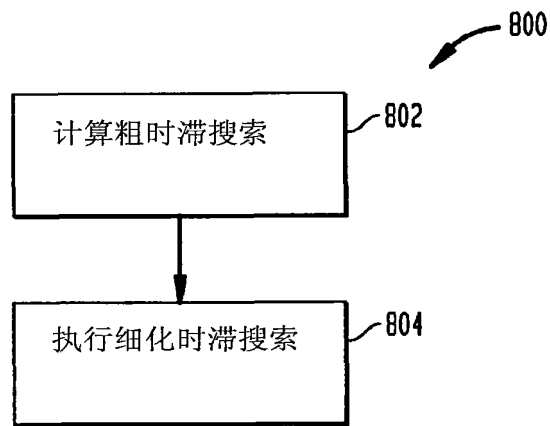


图 8

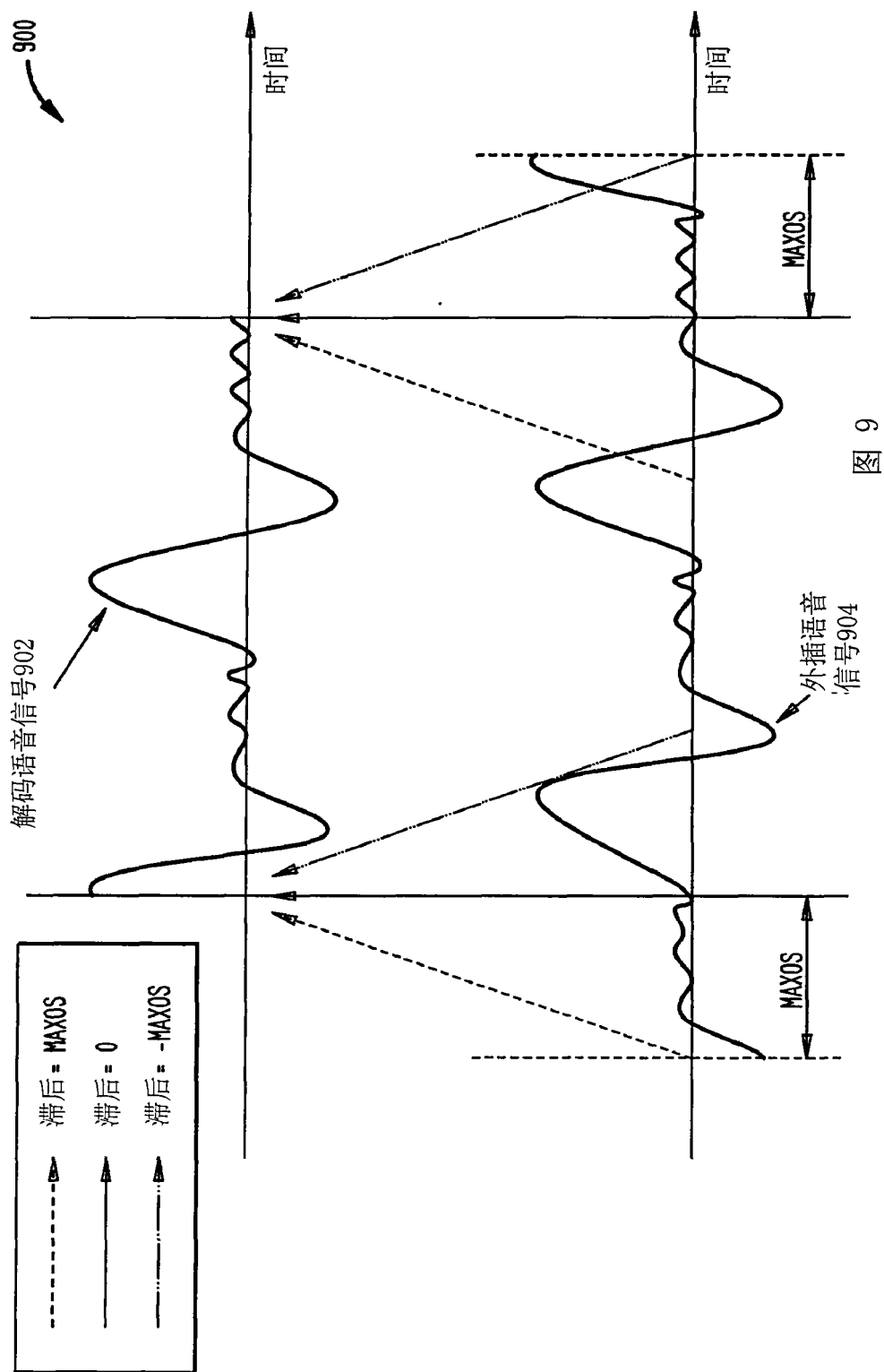


图 9

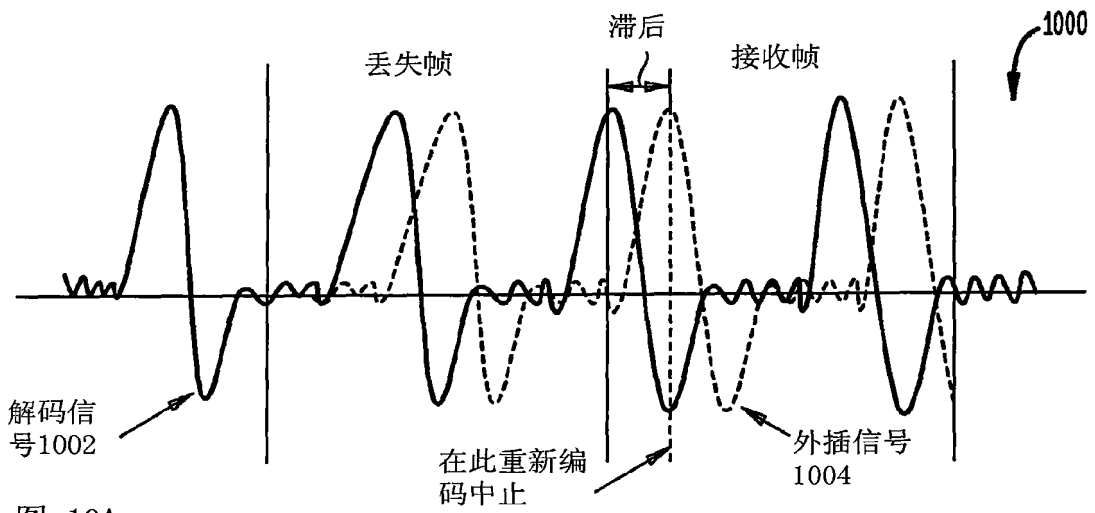


图 10A

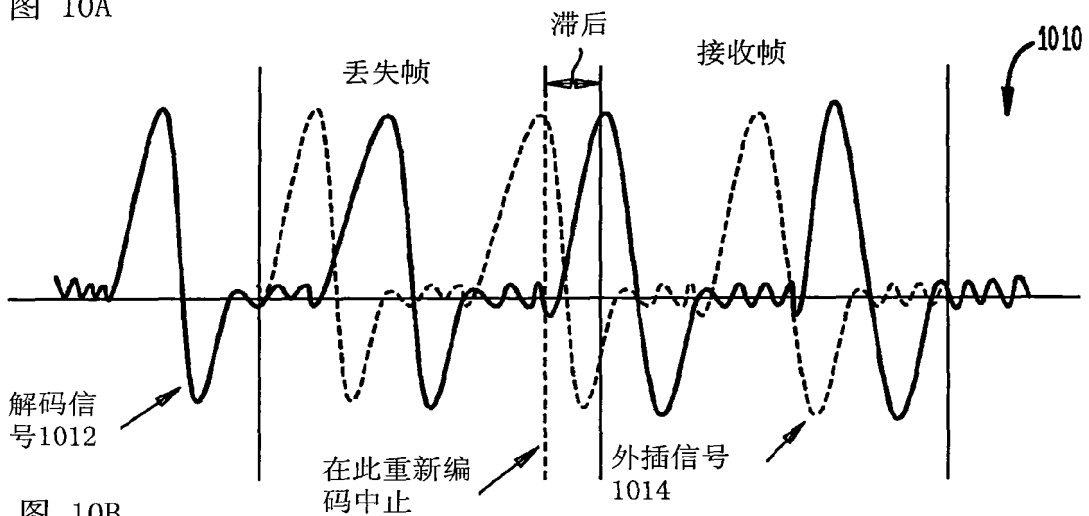


图 10B

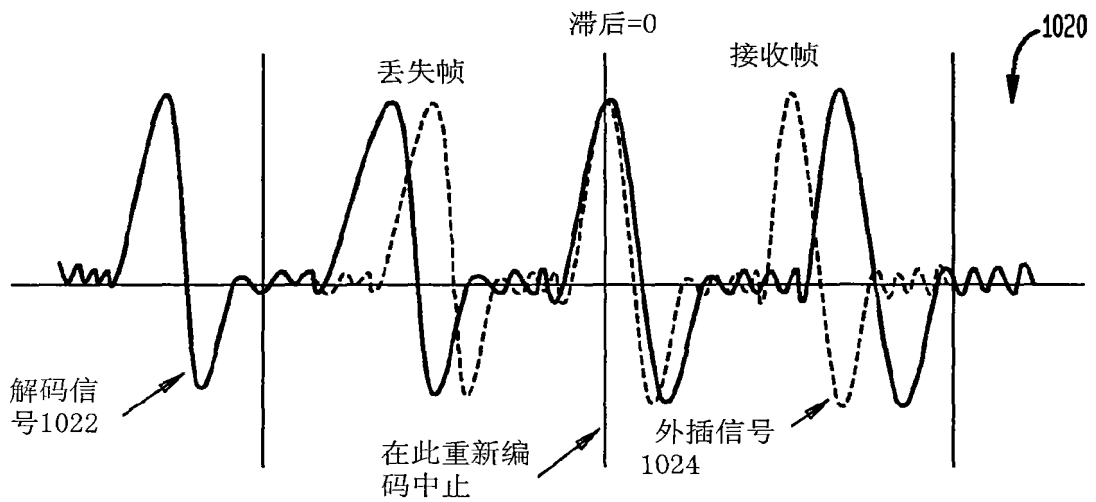


图 10C

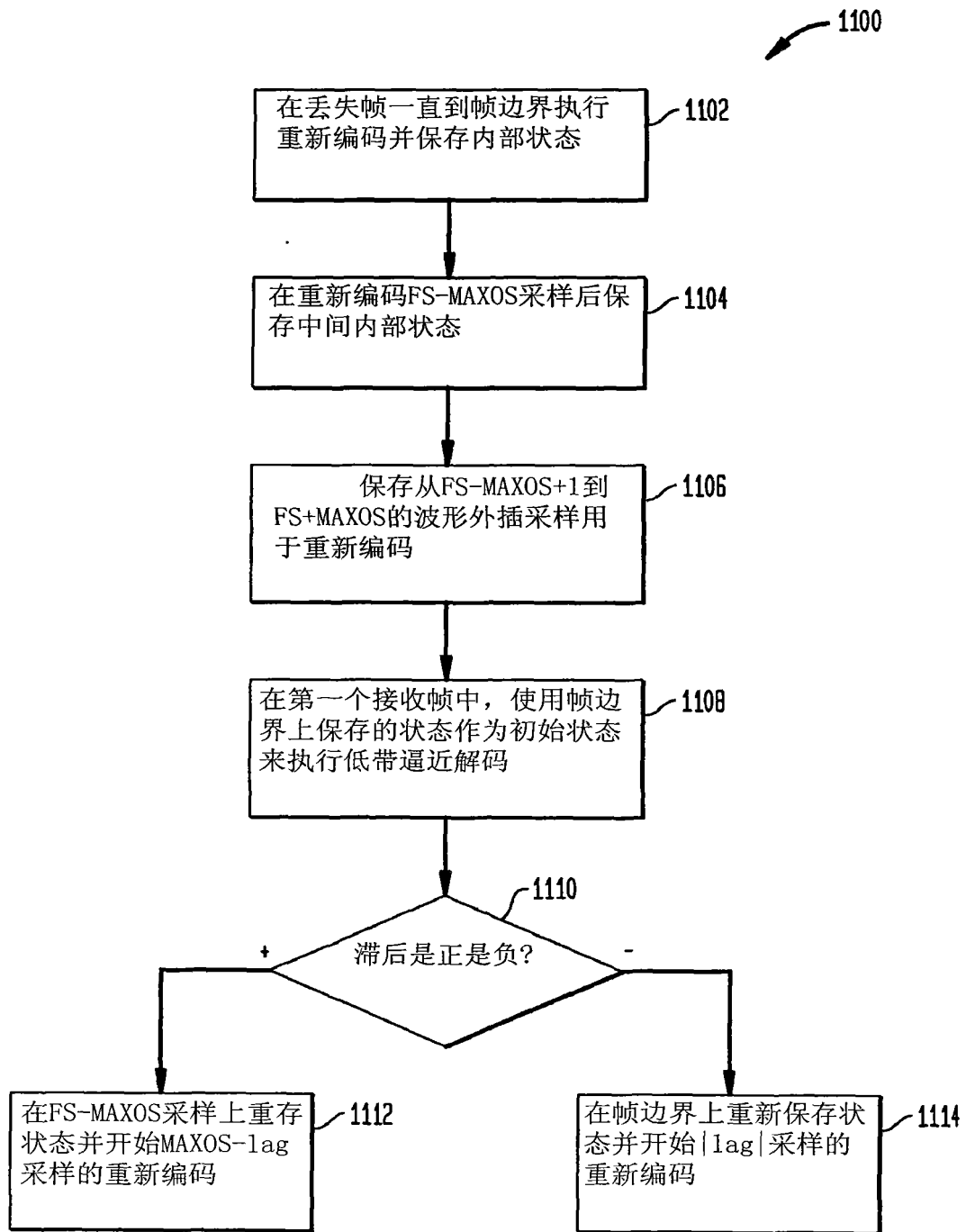


图 11

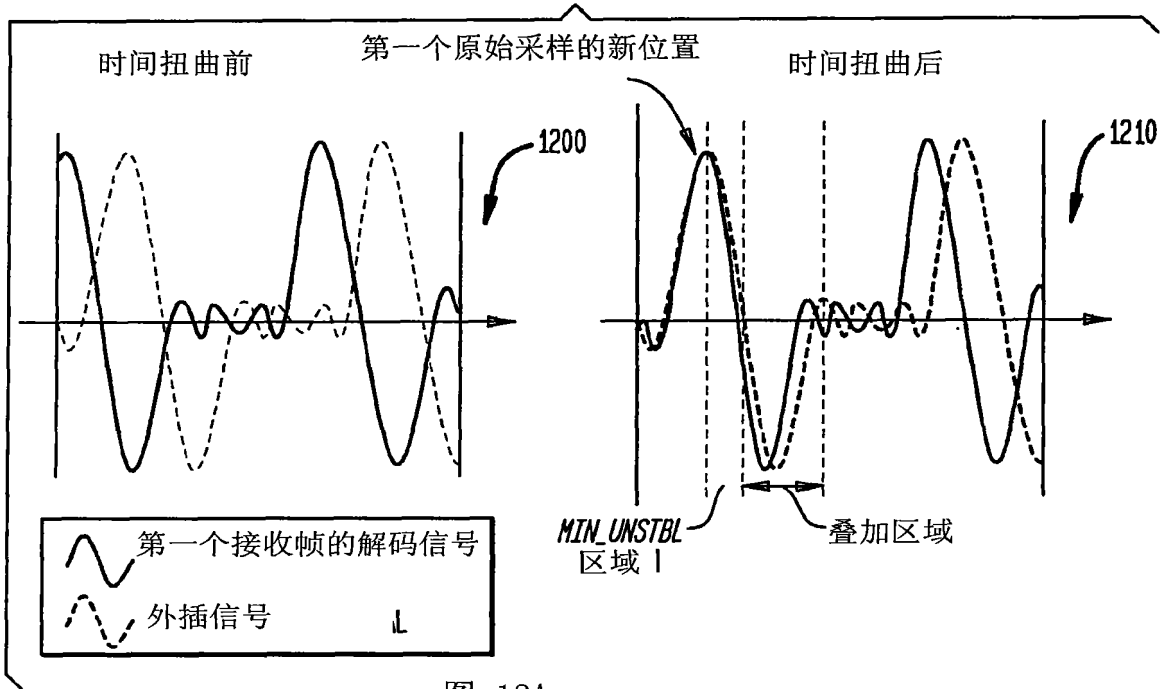


图 12A

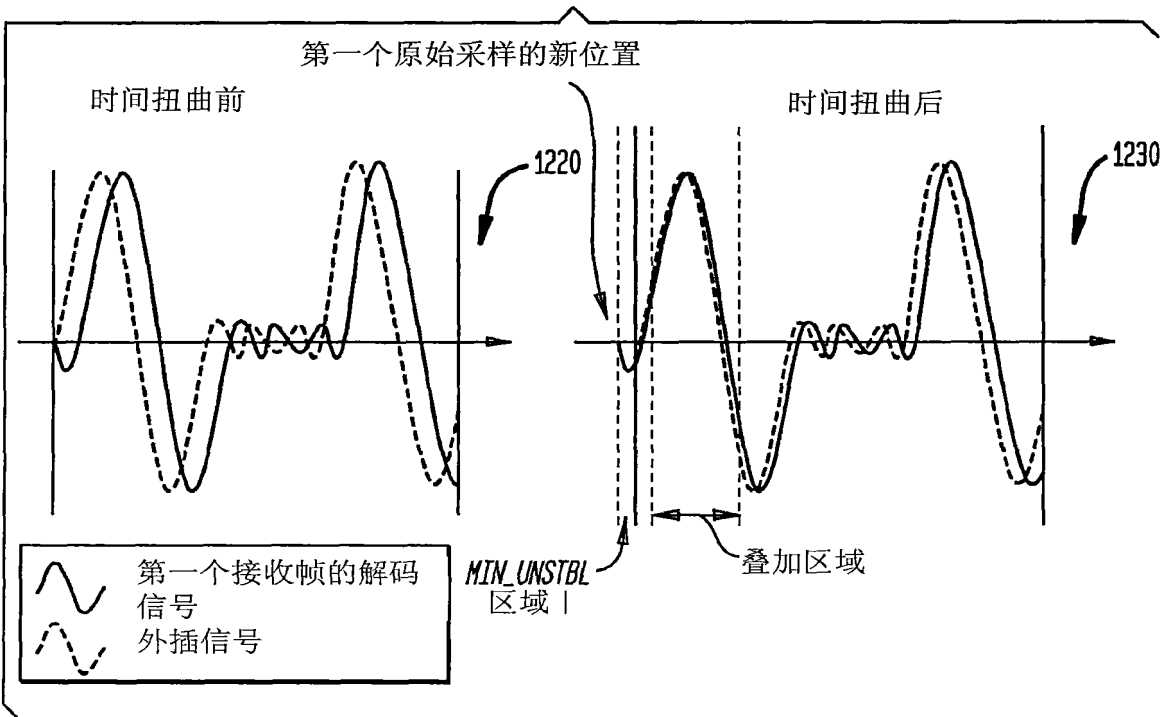


图 12B

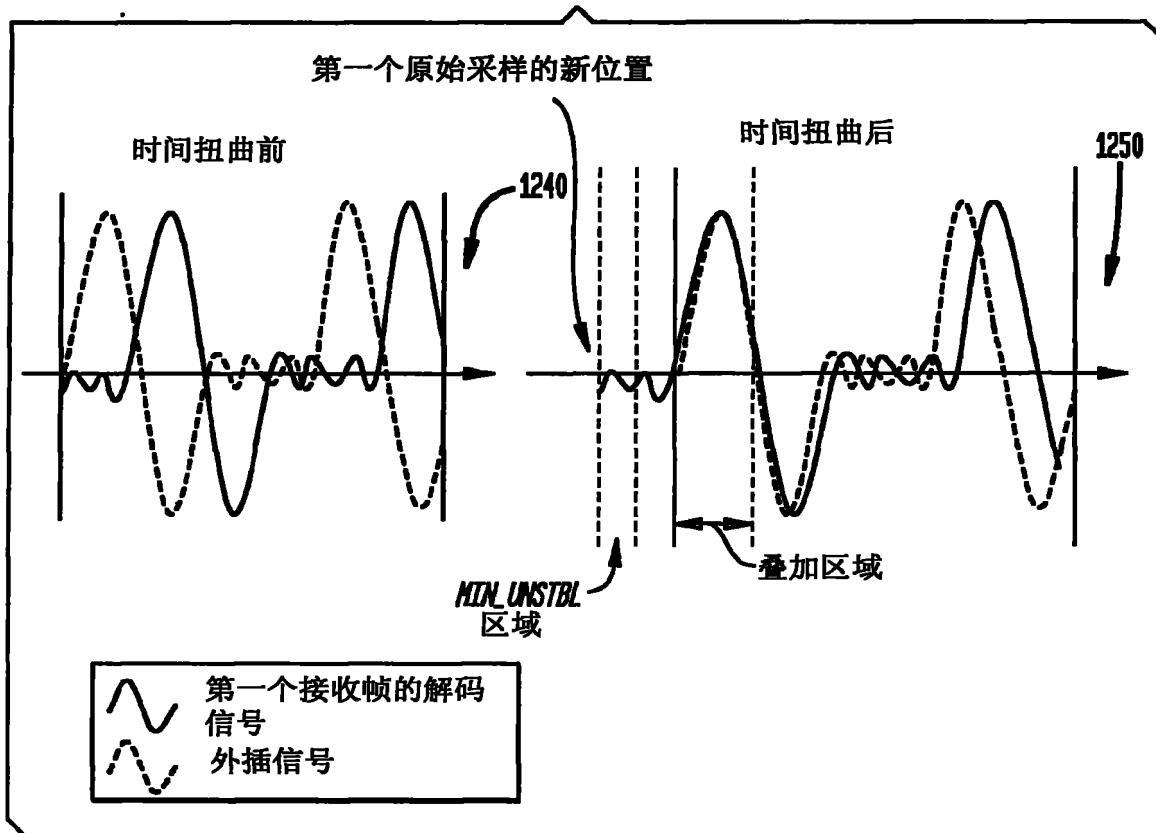


图 12C

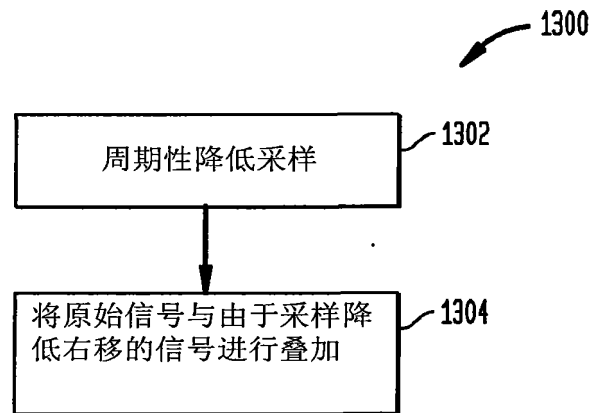


图 13

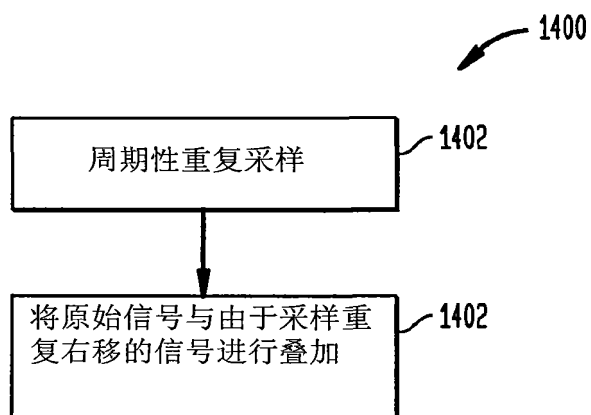


图 14

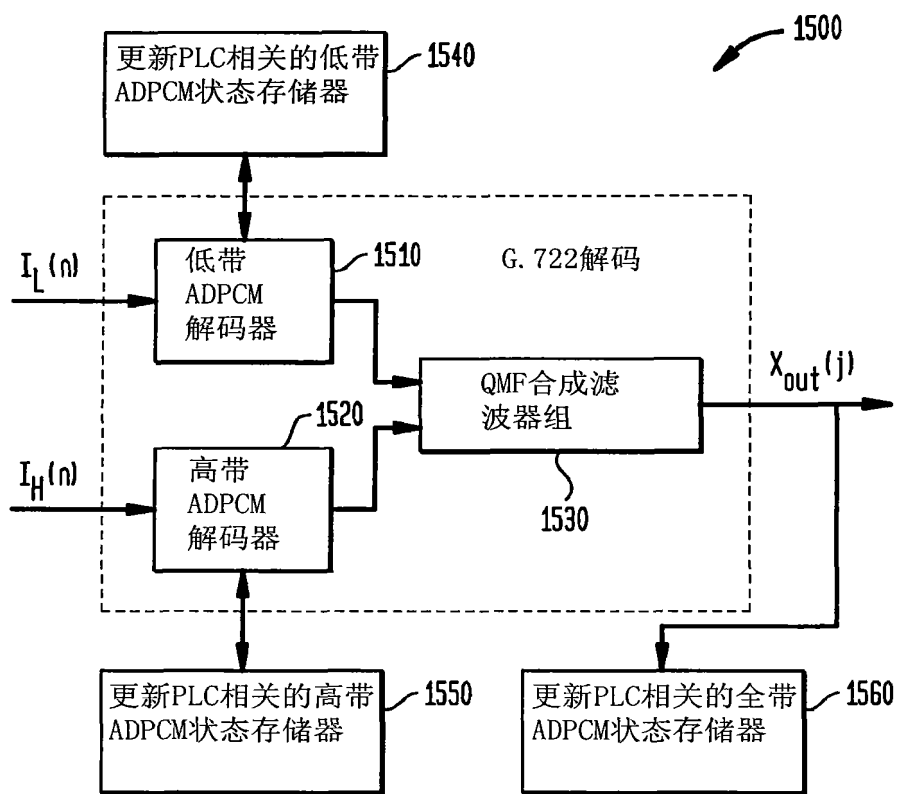


图 15

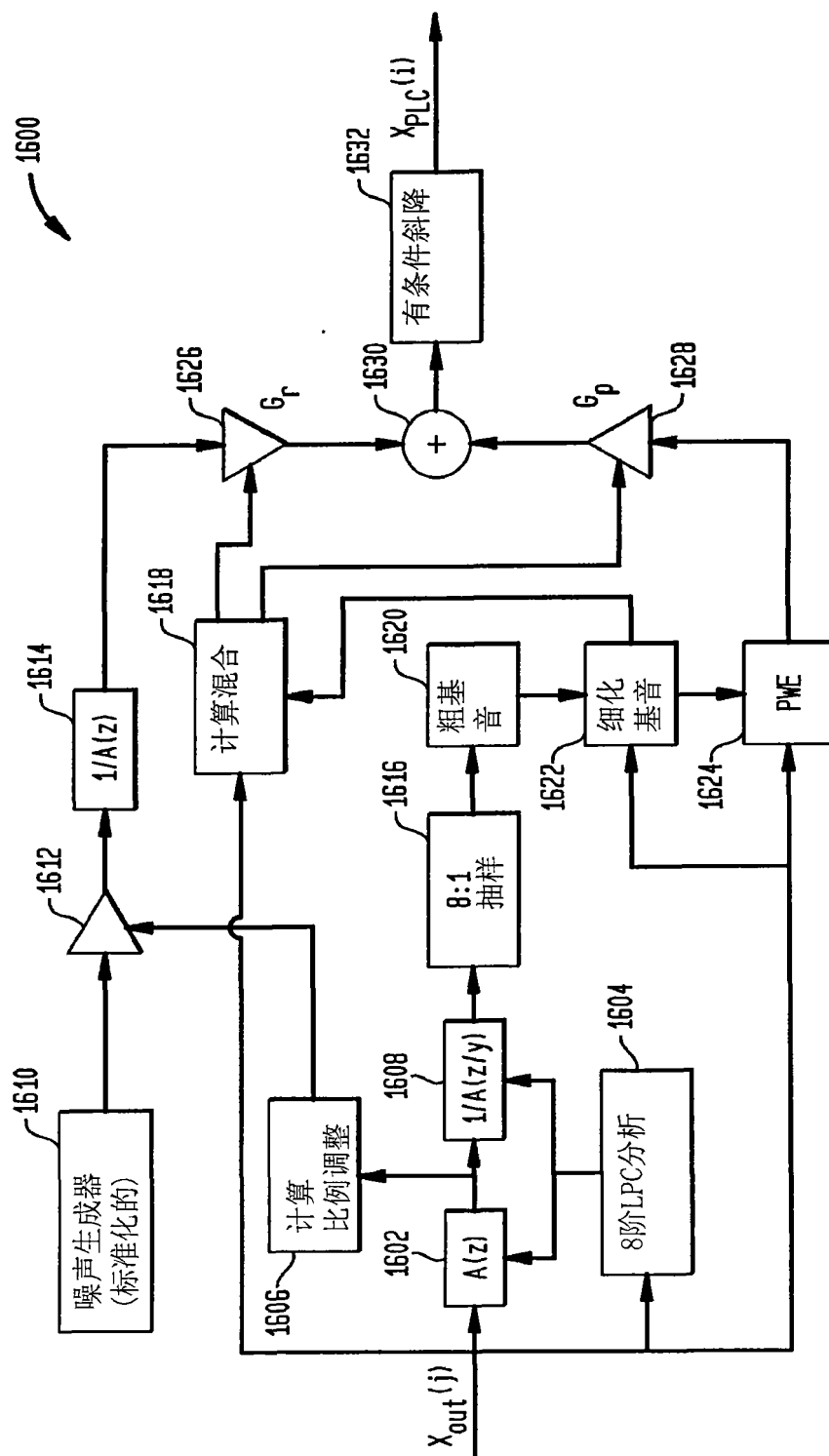


图 16

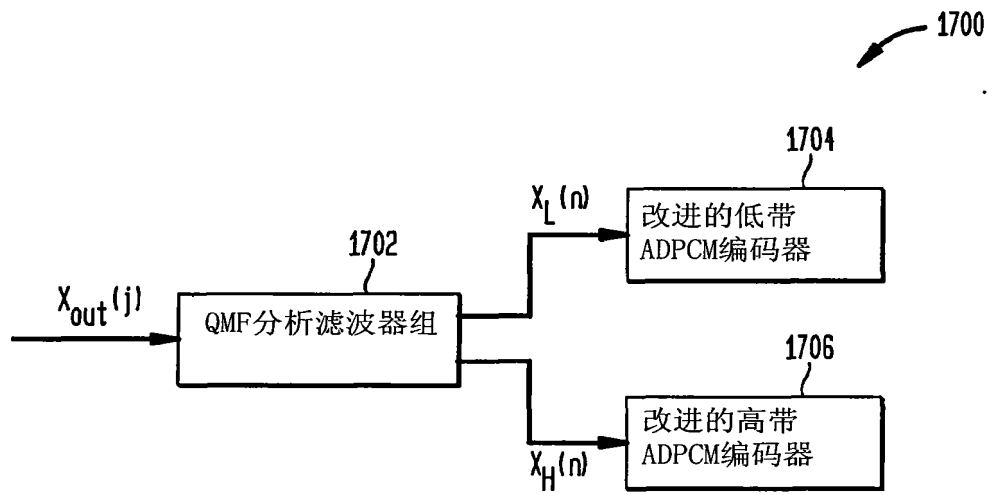


图 17

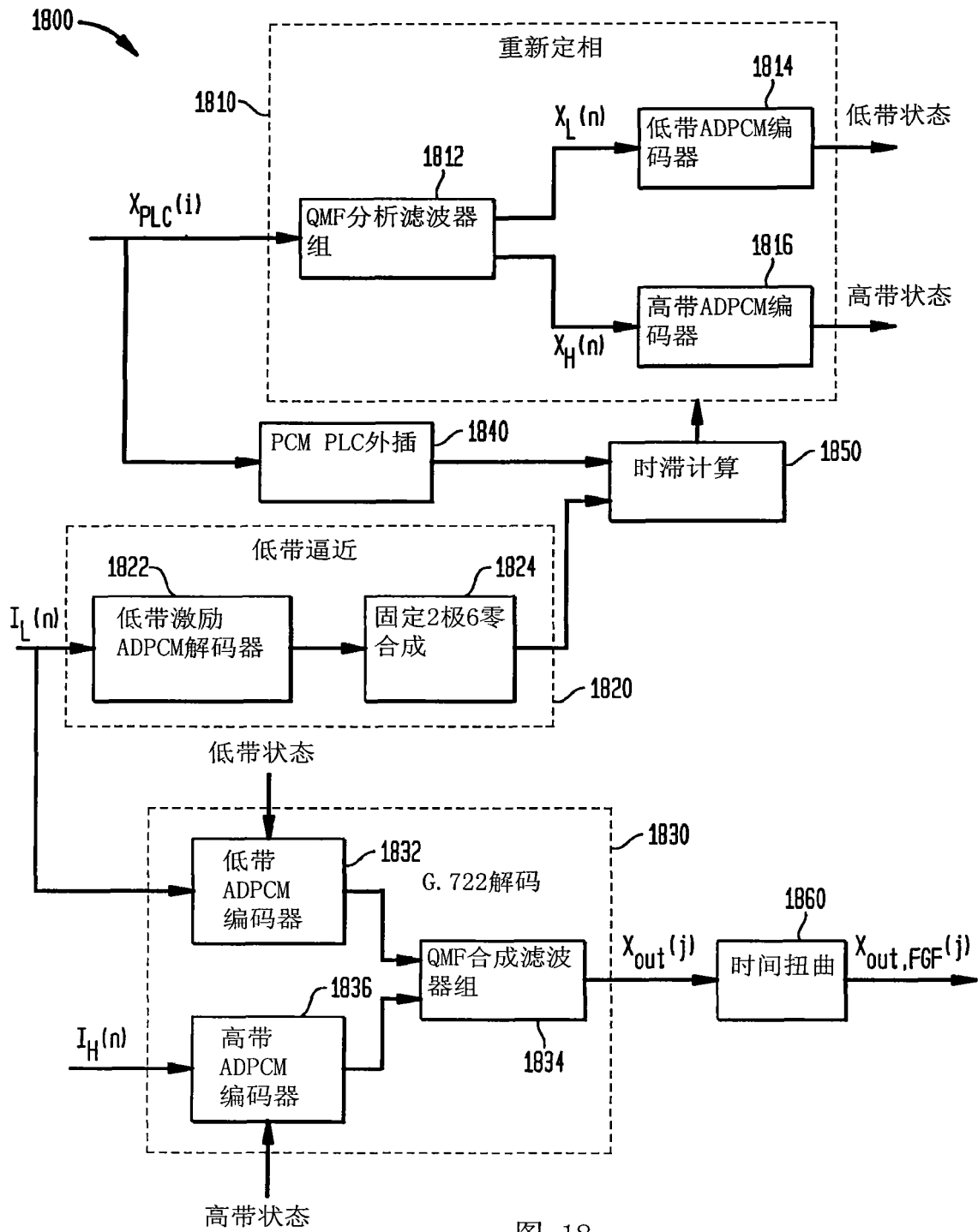


图 18

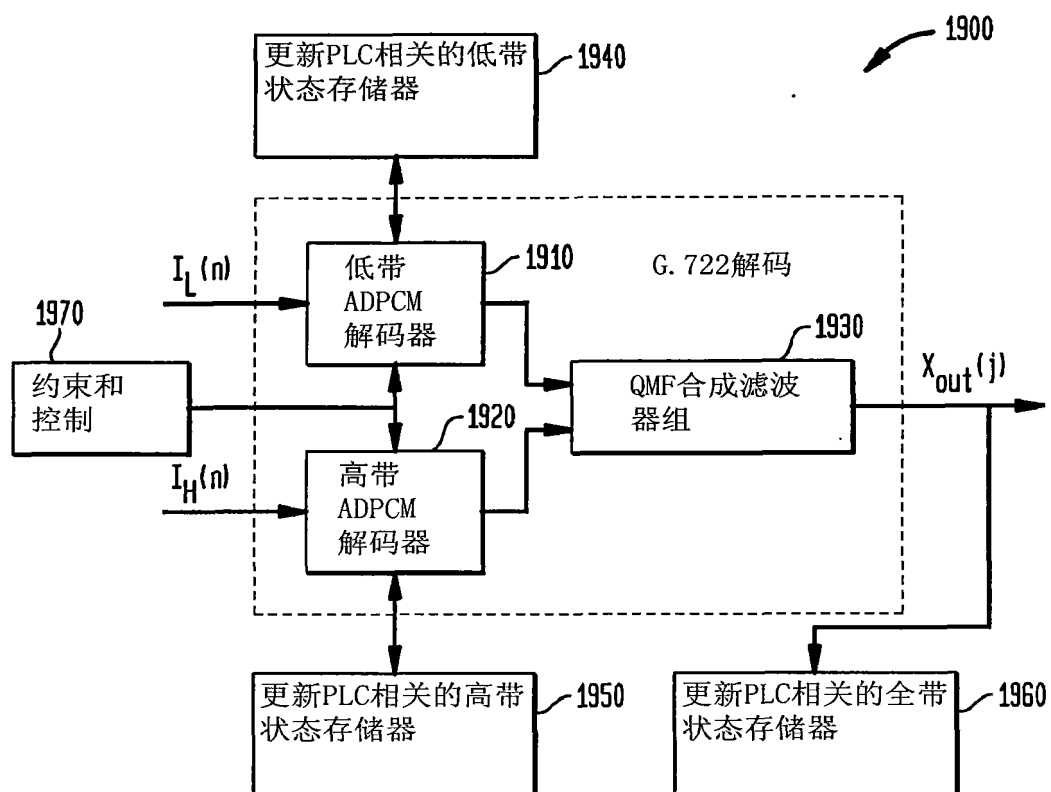


图 19

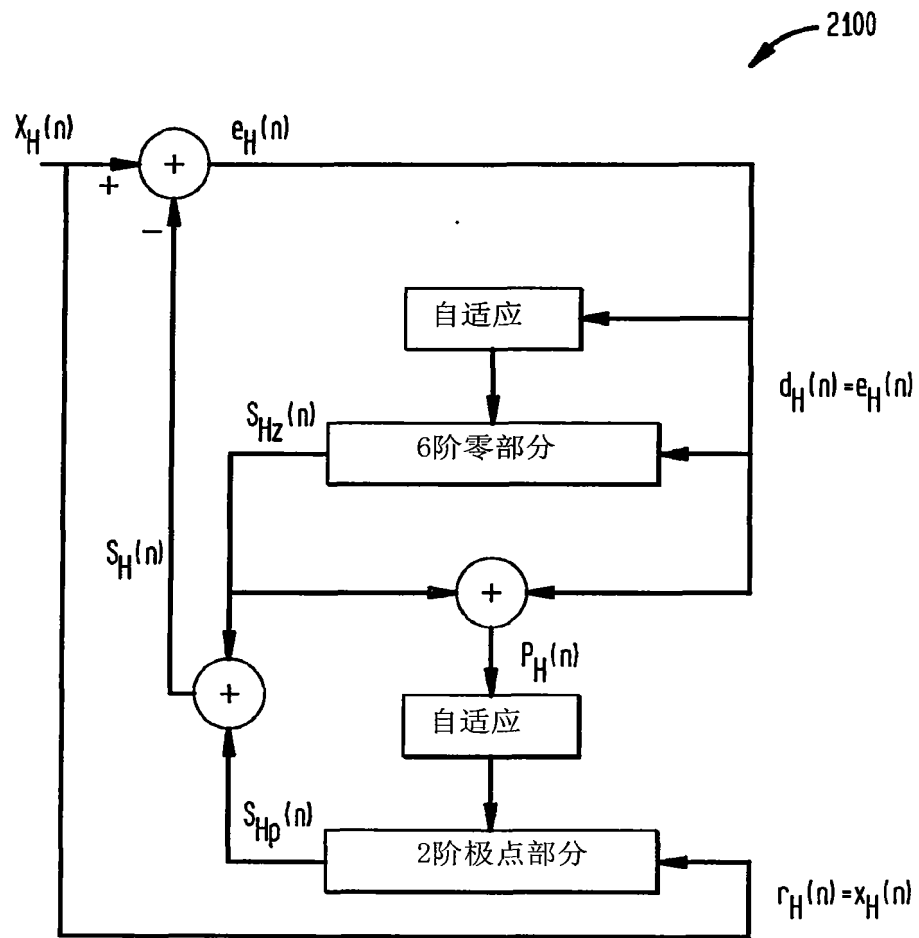


图 21

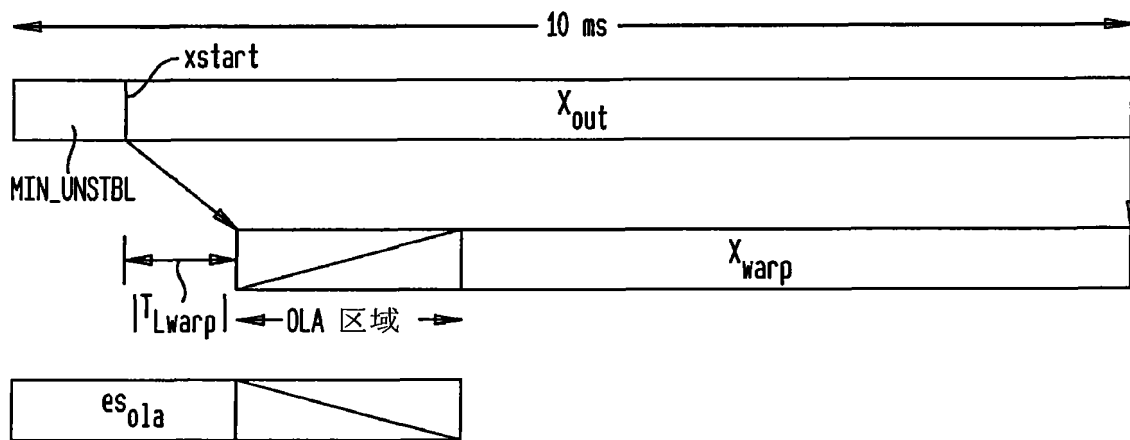


图 22A

2200

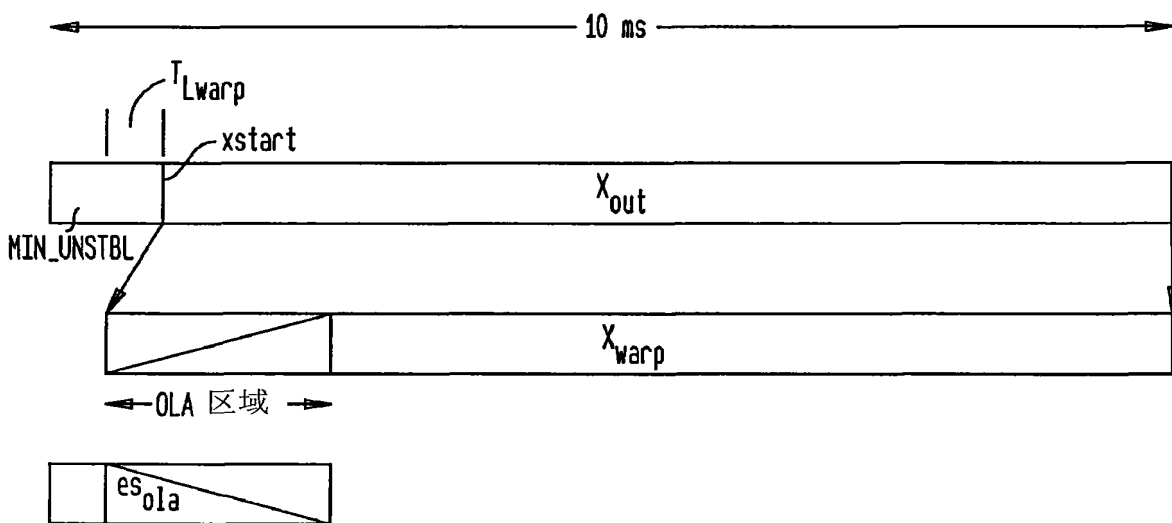


图 22B

2220

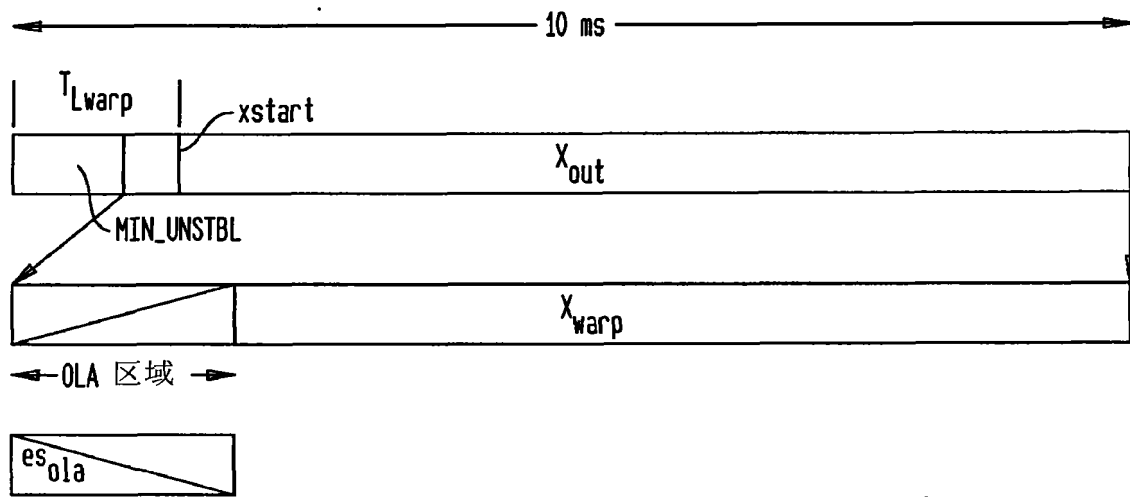


图 22C

2240

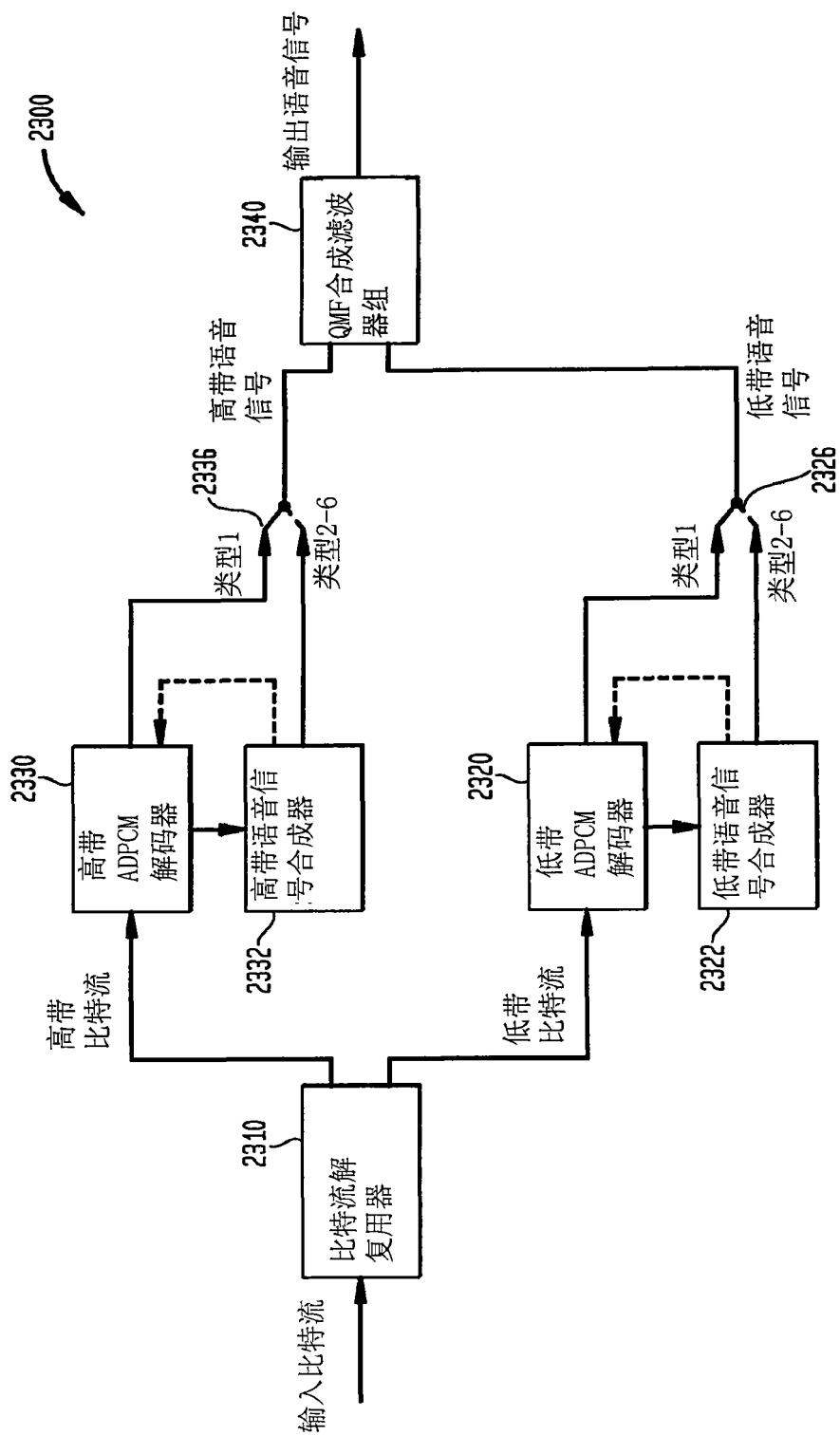


图 23

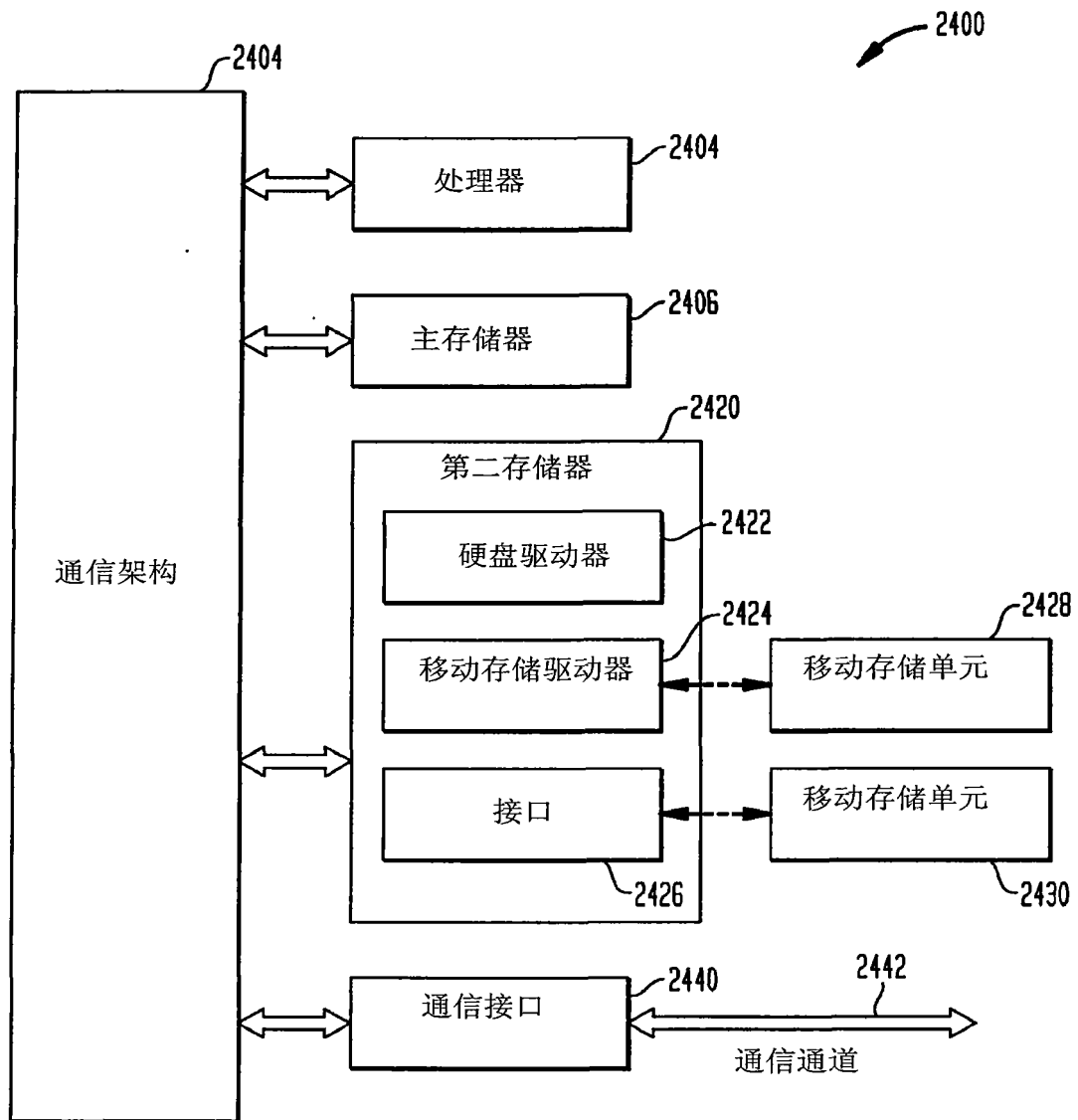


图 24