



(12) 发明专利申请

(10) 申请公布号 CN 103947147 A

(43) 申请公布日 2014. 07. 23

(21) 申请号 201280057064. 4

(74) 专利代理机构 北京康信知识产权代理有限
责任公司 11240

(22) 申请日 2012. 11. 21

代理人 余刚 吴孟秋

(30) 优先权数据

61/562, 093 2011. 11. 21 US

(51) Int. Cl.

H04L 1/00 (2006. 01)

(85) PCT国际申请进入国家阶段日

2014. 05. 20

(86) PCT国际申请的申请数据

PCT/EP2012/073261 2012. 11. 21

(87) PCT国际申请的公布数据

W02013/076156 EN 2013. 05. 30

(71) 申请人 弗兰霍菲尔运输应用研究公司

地址 德国慕尼黑

(72) 发明人 科尔内留斯·黑尔格 托马斯·席尔

托马斯·维甘德

大卫·戈麦斯-巴尔克罗

权利要求书4页 说明书20页 附图26页

(54) 发明名称

用于层获知前向纠错的交错

(57) 摘要

一种纠错编码器包括纠错数据发生器,用于接收属于第一类型的有效载荷数据(第一类型有效载荷数据),用于接收属于第二类型的有效载荷数据,用于为第一类型有效载荷数据确定第一纠错数据,并且用于为第二类型有效载荷数据确定第二纠错数据。所述纠错编码器进一步包括交错器,用于至少使所述第二纠错数据与所述第二类型有效载荷数据交错。关于第一纠错数据和第一类型有效载荷数据的交错的第一交错长度与关于第二纠错数据和第二类型有效载荷数据的交错的第二交错长度不同。还公开了相应的纠错解码器以及用于纠错编码/解码的方法。根据可选的实施方式,有效载荷交错长度与纠错数据交错长度不同。

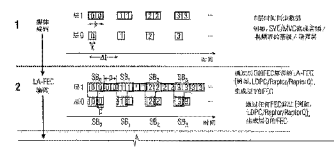


图 18 图 18A
图 18B

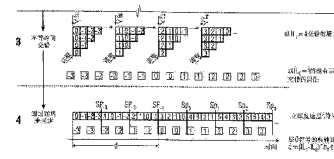


图 18 图 18A
图 18B

1. 一种纠错编码器,包括:

纠错数据发生器,用于接收属于第一类型的有效载荷数据(第一类型有效载荷数据),用于接收属于第二类型的有效载荷数据,用于为所述第一类型有效载荷数据确定第一纠错数据,并且用于为所述第二类型有效载荷数据确定第二纠错数据;以及

交错器,用于至少使所述第二纠错数据与所述第二类型有效载荷数据彼此交错,其中,关于所述第一纠错数据和所述第一类型有效载荷数据的交错的第一交错长度与关于所述第二纠错数据和所述第二类型有效载荷数据的交错的第二交错长度不同。

2. 根据权利要求1所述的纠错编码器,其中,所述第一交错长度与所述第二交错长度不同,以使所述第一类型有效载荷数据能够由包括所述第一类型有效载荷数据和所述第一纠错数据的交错数据流的相对更短部分重构,并且所述第二类型有效载荷数据能够由所述交错数据流或包括所述第二类型有效载荷数据和所述第二纠错数据的另一个交错数据流的相对更长部分重构,以对于影响包括所述第二类型有效载荷数据和所述第二纠错数据的交错数据流的相对更短部分的一个或多个数据差错,所述第二类型有效载荷数据和所述第二纠错数据相对稳健。

3. 根据权利要求1所述的纠错编码器,其中,所述第一类型有效载荷数据和所述第二类型有效载荷数据通过依赖关系相互关联。

4. 根据权利要求1所述的纠错编码器,其中,所述第一类型有效载荷数据是有效载荷内容的多描述编码(MDC)的第一描述的一部分,并且其中,所述第二类型有效载荷数据是所述有效载荷内容的多描述编码的第二描述的一部分。

5. 根据权利要求1所述的纠错编码器,其中,所述第一类型有效载荷数据对应于可扩展视频编码(SVC)或多视点视频编码(MVC)方案的基级层,并且所述第二类型有效载荷数据对应于所述可扩展视频编码或多视点视频编码方案的增强层。

6. 根据权利要求1所述的纠错编码器,其中,所述第一类型有效载荷数据包括音频数据,并且所述第二类型有效载荷数据包括视频数据。

7. 根据权利要求1所述的纠错编码器,其中,所述纠错数据发生器被配置为进一步接收所述第一类型有效载荷数据并且根据所述第一类型有效载荷数据和所述第二类型有效载荷数据确定所述第二纠错数据,以使所述第一类型有效载荷数据的解码器侧纠错能采用所述第一纠错数据和所述第二纠错数据,从而与仅仅使用所述第一纠错数据相比,提高了对所述第一类型有效载荷数据的纠错性能。

8. 根据权利要求1所述的纠错编码器,其中,所述纠错数据发生器被配置为执行层获知前向纠错(LA-FEC)。

9. 根据权利要求1所述的纠错编码器,进一步包括多路复用器,用于至少多路复用所述第一类型有效载荷数据、所述第一纠错数据、所述第二类型有效载荷数据以及所述第二纠错数据,以形成多路复用数据流。

10. 根据权利要求1所述的纠错编码器,进一步包括延迟元件,用于延迟所述第一类型有效载荷数据和所述第一纠错数据。

11. 根据权利要求1所述的纠错编码器,进一步包括信令元件,用于将有关所述第一交错长度和所述第二交错长度中的至少一个的交错长度信息传输给解码器侧。

12. 一种纠错解码器,包括:

解交错器,用于将包括第二类型有效载荷数据和第二纠错数据的交错数据流解交错;以及

差错校正器,用于从所述解交错器中通过解交错形式接收第一类型有效载荷数据、第一纠错数据、所述第二类型有效载荷数据以及所述第二纠错数据,用于至少使用所述第一纠错数据来校正在所述第一类型有效载荷数据中可能具有的差错,并且用于至少使用所述第二纠错数据来校正在所述第二类型有效载荷数据中可能具有的差错;

其中,关于所述第一类型有效载荷数据和所述第一纠错数据的交错的第一交错长度与关于所述第二类型有效载荷数据和所述第二纠错数据的交错的第二交错长度不同。

13. 根据权利要求 12 所述的纠错解码器,其中,所述第一交错长度与所述第二交错长度不同,以使所述第一类型有效载荷数据可由包括所述第一类型有效载荷数据和所述第一纠错数据的交错数据流的相对更短部分重构,并且所述第二类型有效载荷数据可由所述交错数据流或包括所述第二类型有效载荷数据和所述第二纠错数据的交错数据流的相对更长部分重构,以对于影响包括所述第二类型有效载荷数据和所述第二纠错数据的交错数据流的相对更短部分的差错,所述第二类型有效载荷数据和所述第二纠错数据比较稳健。

14. 根据权利要求 12 所述的纠错解码器,其中,所述第一类型有效载荷数据和所述第二类型有效载荷数据通过依赖关系相互关联。

15. 根据权利要求 14 所述的纠错解码器,其中,在编码器侧纠错编码已经根据所述第一类型有效载荷数据和所述第二类型有效载荷数据确定所述第二纠错数据时,所述差错校正器被配置为使用所述第二纠错数据来校正在所述第一类型有效载荷数据中可能具有的一个或多个差错。

16. 根据权利要求 12 所述的纠错解码器,其中,所述第一类型有效载荷数据是有效载荷内容的多描述编码(MDC)的第一描述的一部分,并且其中,所述第二类型有效载荷数据是所述有效载荷内容的多描述编码的第二描述的一部分。

17. 根据权利要求 12 所述的纠错解码器,其中,所述第一类型有效载荷数据与可扩展视频编码(SVC)或多视点视频编码(MVC)方案的基级层对应,并且所述第二类型有效载荷数据与所述可扩展视频编码或多视点视频编码方案的增强层对应。

18. 根据权利要求 12 所述的纠错解码器,其中,所述第一类型有效载荷数据包括音频数据,并且其中,所述第二类型有效载荷数据包括视频数据。

19. 根据权利要求 12 所述的纠错解码器,其中,所述纠错数据发生器被配置为执行层获知前向纠错(LA-FEC)。

20. 根据权利要求 12 所述的纠错解码器,进一步包括:多路分用器,用于至少多路分用包括所述第一类型有效载荷数据、所述第一纠错数据以及所交错的所述第二类型有效载荷数据和第二纠错数据的多路复用的数据流。

21. 根据权利要求 12 所述的纠错解码器,进一步包括:信令元件,用于从编码器侧中接收与所述第一交错长度和所述第二交错长度中的至少一个有关的交错长度信息。

22. 一种用于进行纠错编码的方法,包括:

接收属于第一类型的有效载荷数据;

接收属于第二类型的有效载荷数据;

为所述第一类型有效载荷数据确定第一纠错数据;

为所述第二类型有效载荷数据确定第二纠错数据；以及

至少使所述第二纠错数据与所述第二类型有效载荷数据彼此交错，其中，关于所述第一纠错数据和所述第一类型有效载荷数据的交错的第一交错长度与关于所述第二纠错数据和所述第二类型有效载荷数据的交错的第二交错不同。

23. 根据权利要求 22 所述的方法，其中，所述第一交错长度与所述第二交错长度不同，以使所述第一类型有效载荷数据能够由包括所述第一类型有效载荷数据和所述第一纠错数据的交错数据流的相对更短部分重构，并且所述第二类型有效载荷数据能够由所述交错数据流或包括所述第二类型有效载荷数据和所述第二纠错数据的另一个交错数据流的相对更长部分重构，以对于影响包括所述第二类型有效载荷数据和所述第二纠错数据的交错数据流的相对更短部分的一个或多个数据差错，所述第二类型有效载荷数据和所述第二纠错数据相对稳健。

24. 根据权利要求 22 所述的方法，其中，所述第一类型有效载荷数据和所述第二类型有效载荷数据通过依赖关系相互关联。

25. 根据权利要求 22 所述的方法，其中，所述第一类型有效载荷数据是有效载荷内容的多描述编码 (MDC) 的第一描述的一部分，并且其中，所述第二类型有效载荷数据是所述有效载荷内容的多描述编码的第二描述的一部分。

26. 根据权利要求 22 所述的方法，其中，所述第一类型有效载荷数据与可扩展视频编码 (SVC) 或多视点视频编码 (MVC) 方案的基级层对应，并且所述第二类型有效载荷数据与所述可扩展视频编码或多视点视频编码方案的增强层对应。

27. 根据权利要求 22 所述的方法，其中，所述第一类型有效载荷数据包括音频数据，并且所述第二类型有效载荷数据包括视频数据。

28. 根据权利要求 22 所述的方法，其中，所述纠错数据发生器被配置为进一步接收所述第一类型有效载荷数据并且根据所述第一类型有效载荷数据和所述第二类型有效载荷数据确定所述第二纠错数据，以使所述第一类型有效载荷数据的解码器侧纠错能使用所述第一纠错数据和所述第二纠错数据，从而与仅仅采用所述第一纠错数据相比，提高了对所述第一类型有效载荷数据的纠错性能。

29. 根据权利要求 22 所述的方法，其中，所述纠错数据发生器被配置为执行层获知前向纠错 (LA-FEC)。

30. 一种用于进行纠错解码的方法，包括：

接收包括第二类型有效载荷数据和第二纠错数据的交错数据流；

将所述交错数据流解交错；

接收或获得第一类型有效载荷数据和第一纠错数据；

至少使用所述第一纠错数据来校正所述第一类型有效载荷数据中可能具有的差错；

以及

至少使用所述第二纠错数据来校正所述第二类型有效载荷数据中可能具有的差错；

其中，关于所述第一类型有效载荷数据和所述第一纠错数据的交错的第一交错长度与关于所述第二类型有效载荷数据和所述第二纠错数据的交错的第二交错长度不同。

31. 一种计算机可读数字储存介质，在所述介质上储存了具有程序代码的计算机程序，

该程序代码用于在计算机上运行时执行用于进行纠错编码的方法,所述方法包括:

接收属于第一类型的有效载荷数据;

接收属于第二类型的有效载荷数据;

为所述第一类型有效载荷数据确定第一纠错数据;

为所述第二类型有效载荷数据确定第二纠错数据;以及

至少使所述第二纠错数据与所述第二类型有效载荷数据彼此交错,其中,关于所述第一纠错数据和所述第一类型有效载荷数据的交错的第一交错长度与关于所述第二纠错数据和所述第二类型有效载荷数据的交错的第二交错不同。

32. 一种计算机可读数字储存介质,在所述介质上储存了具有程序代码的计算机程序,该程序代码用于在计算机上运行时执行用于进行纠错解码的方法,所述方法包括:

接收包括第二类型有效载荷数据和第二纠错数据的交错数据流;

将所述交错数据流解交错;

至少使用所述第一纠错数据来校正所述第一类型有效载荷数据中可能具有的差错;以及

至少使用所述第二纠错数据来校正所述第二类型有效载荷数据中可能具有的差错;

其中,关于所述第一纠错数据和所述第一类型有效载荷数据的交错的第一交错长度与关于所述第二纠错数据和所述第二类型有效载荷数据的交错的第二交错长度不同。

33. 一种纠错编码器,包括:

纠错数据发生器,用于接收有效载荷数据并且为所述有效载荷数据确定纠错数据;以及

交错器,用于使所述纠错数据与所述有效载荷数据交错,其中,在交错数据流中,所述纠错数据在所述有效载荷数据之前,并且其中,有效载荷交错长度与纠错数据的交错长度不同。

34. 根据权利要求 33 所述的纠错编码器,其中,来自特定有效载荷数据块的所有有效载荷数据项目插入相应的交错数据块内,并且其中,所述相应的纠错数据项目插入直接或间接位于所述相应的交错数据块之前的至少两个交错的数据块内。

35. 一种纠错解码器,包括:

解交错器,用于为将包括有效载荷数据和用于所述有效载荷数据的纠错数据的交错数据流解交错,其中,在所述交错数据流中,所述纠错数据在所述有效载荷数据之前,并且其中,有效载荷交错长度与纠错数据交错长度不同;以及

差错校正器,用于接收所述有效载荷数据和所述纠错数据,并且用于使用所述纠错数据来校正所述有效载荷数据中可能具有的差错,其中,在仅仅接收到所述有效载荷数据并且错过所述纠错数据时,所述有效载荷是可利用的,但是可能是错误的。

36. 根据权利要求 35 所述的纠错解码器,其中,特定有效载荷数据块的所有有效载荷数据项目被插入相应的交错数据块内,并且其中,所述相应的纠错数据项目被插入直接或间接位于所述相应的交错数据块之前的至少两个交错的数据块内。

用于层获知前向纠错的交错

技术领域

[0001] 本发明涉及一种纠错数据发生器、一种纠错解码器、一种用于进行纠错编码的方法、一种用于进行纠错解码的方法以及相应的计算机可读数字储存介质。

背景技术

[0002] 首先,概述所谓的 LA-FEC(层获知(layer-aware)前向纠错)方法。在【5】和【18】中已经公布了内容。SVC(可扩展视频编码)利用不同的时间和层间预测,用于实现在 SVC 视频流的部分之间引入依赖性的编码效率。图 1 显示了一个示例性编码结构,该结构具有一个基级层和一个增强层,该增强层增大了基级层的时间和空间分辨率。而且,图 1 示意性示出了由分层预测结构和层间预测造成的在时间-空间可扩展性内的依赖性。在图 1 中的箭头表示在不同的访问单元之间的依赖性。

[0003] 在 SVC 中,基级层比增强层更重要。在错过基级层信息的情况下,由于错过预测信息,所以增强层信息通常变得无用。

[0004] 因此,稳健性的区别通常有利于传输 SVC,其中,基级层获得比增强层更强的保护。

[0005] 根据权利要求 1 或权利要求 33 所述的纠错编码器、根据权利要求 12 或 35 所述的纠错解码器、根据权利要求 22 所述的用于进行纠错编码的方法、根据权利要求 30 所述的用于进行纠错解码的方法、根据权利要求 31 所述的计算机可读数字储存介质、根据权利要求 32 所述的计算机可读数字储存介质相对于上面发表的评论提供改进的纠错编码/解码。

发明内容

[0006] 实施方式提供了一种纠错编码器,包括:纠错数据发生器,用于接收属于第一类型的有效载荷数据(第一类型有效载荷数据),用于接收属于第二类型的有效载荷数据,用于为第一类型有效载荷数据确定第一纠错数据,并且用于为第二类型有效载荷数据确定第二纠错数据。该纠错编码器进一步包括交错器,用于至少使所述第二纠错数据与所述第二类型有效载荷数据彼此交错,其中,关于第一纠错数据和第一类型有效载荷数据的交错的第一交错长度与关于第二纠错数据和第二类型有效载荷数据的交错的第二交错长度不同。

[0007] 进一步的实施方式提供了一种纠错解码器,包括:解交错器,用于将包括第二类型有效载荷数据和第二纠错数据的交错数据流解交错。该纠错解码器进一步包括差错校正器,用于从所述解交错器中通过解交错形式接收第一类型有效载荷数据、第一纠错数据、第二类型有效载荷数据以及第二纠错数据,用于至少使用第一纠错数据来校正第一类型有效载荷数据中可能具有的差错,并且用于至少使用第二纠错数据来校正第二类型有效载荷数据中可能具有的差错。关于第一纠错数据和第一类型有效载荷数据的交错的第一交错长度与关于第二纠错数据和第二类型有效载荷数据的交错的第二交错长度不同。

[0008] 根据进一步的实施方式,一种用于进行纠错编码的方法包括以下步骤:接收属于第一类型的有效载荷数据以及接收属于第二类型的有效载荷数据。然后,为第一类型有效载荷数据确定第一纠错数据。为第二类型有效载荷数据确定第二纠错数据。该方法进一步

包括至少使所述第二纠错数据与所述第二类型有效载荷数据彼此交错,其中,关于第一纠错数据和第一类型有效载荷数据的交错的第一交错长度与关于第二纠错数据和第二类型有效载荷数据的交错的第二交错不同。

[0009] 进一步的实施方式提供了一种用于进行纠错解码的方法。该方法包括接收包括第二类型有效载荷数据和第二纠错数据的交错数据流。然后,将交错的数据流解交错。该方法还包括接收或获得第一类型有效载荷数据和第一纠错数据。至少使用第一纠错数据来校正第一类型有效载荷数据中可能具有的差错,并且至少使用第二纠错数据来校正第二类型有效载荷数据中可能具有的差错。关于第一纠错数据和第一类型有效载荷数据的交错的第一交错长度与关于第二纠错数据和第二类型有效载荷数据的交错的第二交错长度不同。

[0010] 进一步的实施方式提供了一种计算机可读数字储存介质,在所述介质上储存了具有程序代码的计算机程序,该程序代码用于在计算机上运行时,执行用于进行纠错编码或解码的方法或者用于进行纠错编码的方法和用于进行纠错解码的方法,如在本文中所述。

[0011] 根据进一步的实施方式,提供一种纠错编码器包括:纠错数据发生器,用于接收有效载荷数据并且为有效载荷数据确定纠错数据;以及交错器,用于使纠错数据与有效载荷数据交错。在交错的数据流中,所述纠错数据在所述有效载荷数据之前,并且其中,有效载荷交错长度与纠错数据的交错长度不同。

[0012] 进一步的实施方式提供了一种纠错解码器,包括解交错器,用于将包括有效载荷数据和用于有效载荷数据的纠错数据的交错数据流解交错。在交错的数据流中,所述纠错数据在所述有效载荷数据之前。有效载荷交错长度与纠错数据的交错长度不同。该纠错解码器进一步包括差错校正器,用于接收有效载荷数据和纠错数据,并且用于使用纠错数据来校正有效载荷数据中可能具有的差错。在仅仅接收有效载荷数据并且错过纠错数据时,所述有效载荷是可利用的,但是可能是错误的。

附图说明

[0013] 在下面,参照示图,更详细地描述本发明的优选实施方式,并且强调在几个设计方面提供优点的本发明的各种优选实施方式。在图中,

[0014] 图 1 示意性示出了由分层预测结构和层间预测造成的在时间-空间可扩展性内的依赖性;

[0015] 图 2 示出了层获知 FEC 生成的一种示意性通用方法;

[0016] 图 3 示意性示出了标准和 LA-FEC 的编码;

[0017] 图 4 示意性示出了标准和 LA-FEC 的解码;

[0018] 图 5 示意性示出了由在【3】中的规定的 Raptor 预编码工序从源符号中生成预编码符号;

[0019] 图 6 示意性示出了由在【3】中的规定的 Raptor 代码的 LT 编码工序生成编码符号;

[0020] 图 7 示意性示出了遵循在【1】中的规格的用于两层的 LA-FEC Raptor 预编码矩阵;

[0021] 图 8 示意性示出了具有两层的 LA-FEC Raptor LT 编码工序;

[0022] 图 9A 和 9B 示意性示出了以固定的业务比特率 1300kbps 通过 SVC 层 CR(基层层/

增强层)使用具有不同的编码率分布的标准(ST)-FEC和LA-FEC的用于VGA接收器的250帧的冻结帧的平均数(图9A)和平均PSNR值(图9B);

[0023] 图10示意性示出了标准LDPC矩阵(顶部)和扩展的LDPC矩阵(底部);

[0024] 图11示意性示出了扩展的LA-FEC解码矩阵;

[0025] 图12显示了在AWGN通道中的16k LA-FEC对16k St-FEC的模拟结果;

[0026] 图13示意性示出了具有 k_0 源和 p_0 奇偶符号的FEC源块 $0(SB_0)$ 的不同交错长度(IL)和增加的时间分集($IL = 1, 2, 3$);

[0027] 图14示意性示出了与在【2】中规定的那样用于层0和单层的RaptorQ的预编码矩阵,要注意的是,在解码和编码工序中使用相同的预编码矩阵;

[0028] 图15示意性示出了RaptorQ的LT编码工序;

[0029] 图16示意性示出了具有两层的LA-FEC RaptorQ的预编码工序;

[0030] 图17示意性示出了具有两层的LA-FEC RaptorQ编码工序的第二步骤以及扩展为层0的预编码符号的矩阵;

[0031] 图18示意性示出了发送器侧工序;

[0032] 图19示意性示出了在传输内包括示例性突发脉冲差错的实例接收器工序;

[0033] 图20示意性示出了FEC源块交错的工序;

[0034] 图21示意性示出了对于每个单层的情况通过某个纠删概率接收的在PSNR方面的视频质量、单独用于每层(ST-FEC)的FEC生成的标准FEC以及具有不同交错长度的LA-FEC UI;

[0035] 图22A和22B示意性示出了基底层(图22A)和增强层(图22B)的层解码概率;

[0036] 图23示意性示出了在称为单层(SL)(交错长度为1和8)的一个FEC源块内具有音频和视频的媒体解码概率方面进行的比较;

[0037] 图24示意性示出了具有源符号的数量 k 的交错长度(IL)、源符号尺寸 T 、唯一符号识别符(FEC ID)以及实际源块的交错长度(IL)的示例性带内信号。

具体实施方式

[0038] 要注意的是,在描述示图期间,在这些示图的几幅图中具有的元件在这些示图的每幅图中由相同的参考符号表示,并且就功能而言时,不重复描述这些元件,以免不必要的重复。然而,为每幅图提供的功能和描述也适用于其他示图,除非明确规定相反。

[0039] 层获知FEC-通用方法:

[0040] 该部分概述了LA-FEC方法。在图2中显示了LA-FEC的基本方法。图2显示了层获知FEC生成的可能的通用方法。生成层0的冗余,遵循基底的FEC算法(Raptor)。在所有依赖层上生成层1到层N的冗余。层获知FEC(LA-FEC)方法的基本概念在于在单独的视频层上扩展FEC算法的编码工序。依然不接触基底层的FEC工序,因此,依然允许单独解码基底层,并且保留原始FEC算法的校正功能。由于在FEC算法内从不太重要的媒体层中引入连接,所以更重要的媒体层由额外的修复数据保护。这增大了更重要的层的纠错功能,而不增加额外的修复数据。在图2中的方案显示了交叉层FEC生成。虽然基底层(“层0”)FEC生成工序未改变,但是在“层1”和“层0”的源符号上,生成“层1”的FEC数据,在“层2”、“层1”以及“层0”等上生成“层2”的FEC数据,直到“层N”的FEC数据,在“层N”和

所有依赖的媒体层的源符号上生成“层 N”的 FEC 数据。作为一种通用的 FEC 方法,通过仅仅在所有依赖的媒体层上扩展媒体增强层的编码工序,LA-FEC 可整合在任何 OSI 层(物理链路或应用层)上,并且应用于 FEC 代码中,例如,LDPC、Raptor 以及 RaptorQ。

[0041] 为了说明 LA-FEC 方法的原理,图 3 和图 4 为一个源块使用一个简单的奇偶校验码来比较标准 FEC 和 LA-FEC(修改由虚线标出)的编码和解码工序。在该实例中,采用纠删信道。两个质量层进行 FEC 编码,其中,由于在媒体流(例如,SVC)内的层间预测,所以层 1 取决于层 0。对于每层 $l = 0, 1$,具有三个源比特 $k_0/l = 3$ 和两个奇偶性 $p_0/l = 2$ 比特。

[0042] 为了进行编码(图 3),通过源比特的简单 XORing 工序,计算奇偶位。使用标准 FEC,仅仅在当前层内应用 XORing 工序,然而,使用 LA-FEC, XORing 工序在遵循现有依赖性的层上扩展。因此,在层 0 和层 1 这两层的源比特上生成层 1 的奇偶位,并且该奇偶位可与层 0 的奇偶位一起进一步用于对这两层进行纠错。源和奇偶位组合成码字并且通过容易发生错误的信道传输。

[0043] 在所述解码实例(图 4)中,在标有“?”的层 0 的码字内具有三个传输差错,并且在层 1 内没有差错。要注意的是,所述实例采用纠删(erasure)信道,例如,二进制纠删信道(BEC)或 Gilbert-Elliott 信道,其中,正确地或者在未知的状态中接收比特或数据包(在图中标为“?”)。这种通道模型用于在链路或应用层上模拟传输差错,其中,在下部 ISO/OSI 层上(例如,在物理层上)实现的前述方法构成纠错或错误检测(例如,通过 LDPC 代码和/或循环冗余检验(CRC))。在物理层上的 FEC 代码通常与源自解调制工序的 LLR 值一起运行,然而,在链路或应用层 FEC 上的 FEC 代码与又称为大量数据包的纠删一起运行。在物理层上不能校正的数据包通常标记为错误并且转发给更高层(例如,链路或应用层)。这就是在链路或应用层上实现的 FEC 代码不需要实现错误检测的方式的原因。然而,重要的是,在此处注意,为了简单起见,仅仅在应用或链路层上显示规定的实例,并且在该说明书中提出的方法还可应用于物理层上,与在“具有 LDPC 代码的物理信道上的 LA-FEC”部分内在 LDPC 描述中所显示的一样,该物理层在错误信道(例如,AWGN)上运行并且需要错误检测的方式。使用标准 FEC,在层 0 内没有充足的奇偶位。因此,不能校正。虽然成功地接收了层 1,但是由于在层 0 中缺少依赖性,所以不能使用层 1。使用 LA-FEC,层 1 的奇偶位可与层 0 的奇偶位一起用于校正层 0。在规定的实例中,仅仅通过 LA-FEC,可校正这两层。总之,图 4 示意性显示了标准和 LA-FEC 的解码。使用 LA-FEC,这两层的奇偶位可用于组合式解码。

[0044] 在减少保护增强层的情况下,得以改进基级层保护。通过 LA-FEC,不能独立于基级层校正增强层。然而,由于在 SVC 视频流内现有的依赖性,所以在丢失基级层的这种情况下,通常也不能使用增强层数据。因此,在视频质量方面,LA-FEC 绝不比标准 FEC 差。

[0045] 在【1】中的讨论之后,由等式(1)提供通过 ST-FEC 和最佳 FEC 算法将层 1 解码的条件:

$$[0046] \quad r_0 > k_0 \text{ und } r_1 > k_1 \quad (1)$$

[0047] 由于 LA-FEC 的增强层符号还保护基级层符号,所以将基级层解码的条件【1】通过 LA-FEC 变成等式(2):

$$[0048] \quad (r_0 \geq k_0) \cup (r_0 + r_1 \geq k_0 + k_1) \quad (2)$$

[0049] 由于通过 LA-FEC,增强层 FEC 校正也取决于基级层恢复,所以恢复增强层的条件通过 LA-FEC 变成等式(3):

[0050] $(r_1 \geq k_1) \cap (r_0+r_1 \geq k_0+k_1)$ (3)

[0051] 然而,由于现有媒体依赖性,所以增强层取决于基级层,在没有基底层时,无论如何都不能使用增强层,从而还是通过 ST-FEC,用于对增强层进行成功的媒体解码的条件与等式 (3) 的条件一样。

[0052] 层获知 FEC 整合到 Raptor FEC 内:

[0053] LA-FEC 方案可应用于这两层上, FEC 与 LDPC 代码一样应用于物理层上或者与 Raptor 【1】、【3】一样用于上层 FEC 上。在此处考虑的示例性上层 FEC 是 Raptor FEC。要注意的是,通过与更有效的 RaptorQ FEC 【2】相似的方式,可应用所显示的扩展,如在“层获知 FEC 整合到 RaptorQ FEC 内”部分内所描述的。

[0054] Raptor 代码通常是具有线性时间编码和解码的第一类已知喷泉码中的一个。在制备编码时,在源块内收集某个数量的数据。源块的数据进一步分成固定符号尺寸的 k_0 源符号。图 5 和图 6 显示了 Raptor 编码和解码工序的两个步骤,与在【3】中为单个媒体“层 0”所规定的那样,该工序构成了两个编码步骤:预编码生成和 LT 代码生成。图 5 示意性显示了由在【3】中的规定的 Raptor 预编码工序从源符号中生成预编码符号。图 6 示意性示出了由在【3】中的规定的 Raptor 代码的 LT 编码工序生成编码符号。要注意的是,在【3】中的说明描述了用于进行编码和解码的相同两步工序。在第一步骤中,在源符号上可应用固定速率的‘预编码’步骤,在此处通常是任何纠删码,例如,LDPC,以生成层 $l = 0$ 的所谓的 L_0 预编码符号。预编码符号的值由在图 5 中所示的矩阵决定,该矩阵由预编码矩阵 G_{LDPC_0} 、单位矩阵 I_{S_0} 、零矩阵 $0_{S \times H_0}$ 以及 LT 矩阵 $G_{LT_0}[0:k_0-1]$ 构成,其中,在第二编码工序中,后面的矩阵与最初的 k_0 行 $G_{LT_0}[0:n_0-1]$ 相同。在括号里面的值表示行数。矩阵 G_{LT_0} 的整合确保了在 LT 编码之后的最初的 k_0 编码符号与源符号相同并且 Raptor 代码提供了系统码字。在完成第一步骤之后,将预编码符号转发给第二步骤,LT 编码。

[0055] n_0 编码符号的来源由预编码符号的 XORing 运算计算,遵循由 LT 代码规定的并且由在 Fehler! Verweisquelle konnte nicht gefunden werden.. 中描述的 $G_{LT_0}[0:n_0-1]$ 矩阵说明的连接。要注意的是,也通过 LA-FEC,基级层编码符号的生成在最初的 Raptor 工序之后。

[0056] 通过标准的 FEC 方法,通过相同的方法生成层 1 (增强层) 的编码符号,在【3】中的工序之后。为了通过 LA-FEC 生成增强“层 1”的编码符号,LA-FEC 方法需要整合在 Raptor 编码工序内。这一方面需要扩展基级层的预编码符号的 LT 编码步骤的 G_{LT} 矩阵,另一方面需要扩展预编码工序的 G_{LT} 矩阵,以保留代码的系统行为。图 7 和图 8 显示了为了生成层 $l = 1$ 的编码符号所需要的扩展。图 7 示意性示出了遵循在【1】中的规格的用于两层的 LA-FEC Raptor 预编码矩阵。

[0057] 在图 7 中为两层显示了 LA-FEC Raptor 的预编码矩阵。该矩阵由这两层的两个预编码矩阵 (左上和右下)、零矩阵 (右上) 以及从层 1 到层 0 的 LA-FEC 扩展 (左下) 的串联构成。要求 LA-FEC 扩展保持系统代码。要注意的是, G_{LT_1} 和相关的扩展 $G_{LT_0}[n_0:n_0+k_1-1]$ 行在位置 n_0 处开始。一方面是因为层 0 的线性独立行的数量增大,另一方面是因为允许编码符号的唯一识别符复制在接收器上的矩阵。在图 8 中的工序之后,生成层 0 的编码符号。为了生成增强层编码符号,将这两层的预编码符号转发给在图 8 中描述的 Raptor 编码工序的下一个步骤。

[0058] 在图 8 中显示了 LA-FEC Raptor 工序的第二步骤（具有两层的 LA-FEC LT 编码工序）。通过 $G_{LT_0}[n_0:n_0+n_1-1]$ 扩展 $G_{LT_1}[n_0:n_0+n_1-1]$ 矩阵，在第二步骤中的编码符号的生成扩展为层 0 的预编码符号。通过层 0 预编码符号的额外整合，层 1 的编码符号可与层 0 的编码符号一起使用，用于共同解码，如上面在“层获知 FEC-通用方法”部分中的实例中示例性所示。

[0059] LA-FEC 扩展重新使用算法，用于在【3】中所述的预编码生成和 LT 编码。在成功解码的“层 0”的情况下，LA-FEC 扩展不再需要在层上引入连接，并且通过在“层 1”的 FEC 工序中 XORing 层 0 的预编码符号，可去除这些连接。在这种情况下，在标准的 Raptor 编码工序之后，可校正“层 1”，启用其完全校正性能。

[0060] LA-FEC 的信令：

[0061] 在传输系统中使用 LA-FEC，需要特定的信令和输送技术来支持与 LA-FEC 编码相结合的多层方法。假设在链路或应用层上整合 LA-FEC Raptor 扩展，以用于通过 RTP [6] 进行实时传输。对于实时应用，由于其允许最小输送延迟具有无连接的以及不可靠的性质，所以通常通过 UDP [7] 使用 RTP。RTP 提供了基本特征，例如，媒体同步、传输顺序恢复、多路复用、源识别以及接收反馈信息。对于 SVC，媒体有效载荷分组需要可扩展视频编码【8】的 RTP 有效载荷格式，并且对于 MVC，需要多视点视频编码【9】的 RTP 有效载荷格式。尤其地，SVC 和 MVC 的这些有效载荷格式在多个 RTP 会话中限定分层的 SVC 和 MVC 数据的传输，这允许使用 LA-FEC 编码工序的传输系统根据输送地址（例如，IP 地址、UDP 端口或在 RTP 数据包报头内的同步源标识符（SSRC））在 SVC 层和 MVC 视点之间简单地进行区分。在会话描述协议【10】中限定会话相关信息的信令。为了用信号通知相同编码解码器的包含 RTP 会话的层或视点的依赖性，需要在【11】中的 SDP 扩展。

[0062] 为了输送 FEC 编码的数据，IETF 创建通用的 FECFRAME 框架，该框架为基于 FEC 的内容传输协议限定基本方式，这还可用于 RTP 中。除了其他特征，该框架限定处理多个媒体和修复流程的方式并且进一步为源符号提供识别机构，作为有效载荷分组信息的一部分。为了使用具有 Raptor 代码的该框架，打算使用【12】。

[0063] 为了使该框架适用于 LA-FEC，可使用 Raptor FEC 方案【13】和 Raptor RTP 有效载荷格式【14】，无需为了将修复流程分组而进行修改。在【12】中限定 Raptor FEC 方案的信令，其中，已经在 RFC6364 [15] 中限定了依赖修复流程的指示，与 LA-FEC 基级层保护和 LA-FEC 增强层保护所要求的那样。FEC 框架还规定添加的 FEC 流程，这些流程可共同用于进行纠错，与在 LA-FEC 增强层和基级层 FEC 流程的情况中一样。在媒体元件的 RFC5888 [17] 一般分组中，并且更具体而言，为在 RFC5956 [16] 中的 FEC 框架，描述为在 SDP 中的附加流程进行 FEC 分组的语义学。

[0064] RFC6364 规定使用 SDP 并且描述用信号通知在发送器与接收器之间的 FEC 框架所需要的参数。对于 UDP 格式化的输入流，协议标识符是‘FEC/UDP’。FEC 框架在源流和修复流程之间进行区分。在属性‘源-滤波器’中描述源 IP 地址。

[0065] 多个源和修复流程的分组

[0066] FEC 框架允许多个源和修复流程分组。这种组可由单个或多个 FEC 框架实例保护。在源和修复流程之间的相关性可由‘组’属性表示。在 RFC5956 中描述有关附加的修复流程的‘组’属性的用途的详情。在 RFC5956 之后，包含在一个‘组’属性中的所有流程称为 FEC

组。在单个‘组’中具有一个以上的修复流程的情况下，必须添加这些修复流程。RFC5888 限定‘FEC-FE’，作为使用 FEC 框架的分组语义学。因此，‘a = group:FEC-FR’语义学必须用于使多个流程相关联。

[0067] LA-FEC 的典型场景具有两个源流程 (S1 (基级层) 和 S2 (增强层)) 和两个修复流程 (R1 (基级层 FEC) 和 R2 (层获知的基级和增强层 FEC))。为了进行描述，这种场景需要在 SDP 内规定两个 FEC 组，如在以下实例中所示：

[0068] a = group:FEC-FR S1 R1

[0069] a = group:FEC-FR S1 S2 R1 R2

[0070] 下面提供使所有讨论的 RFC 相结合的一个示例性 SDP 描述 (水平线表示在 SDP 描述内的不同部分)。

[0071] v = 0

[0072] o = Cornelius11223344551122334466IN IP41a-fec.example.com

[0073] s = SVC LA-FEC Raptor Example

[0074] t = 00

[0075] a = group:FEC-FR S1 S2 R1 R2

[0076] a = group:FEC-FR S1 R1

[0077] c = IN IP4233.252.0.1/127

[0078] a = group:DDP S1 S2

[0079] -----

[0080] m = video20000RTP/AVP96

[0081] b = AS:90

[0082] a = framerate:15

[0083] a = rtpmap:96H264/90000

[0084] a = fec-source-flow:id = 0

[0085] a = mid:S1

[0086] -----

[0087] m = video20002RTP/AVP97

[0088] b = AS:64

[0089] a = framerate:15

[0090] a = rtpmap:97H264-SVC/90000

[0091] a = fec-source-flow:id = 1

[0092] a = mid:S2

[0093] a = depend:97lay S1:97

[0094] -----

[0095] m = application30000RTP/AVP110

[0096] c = IN IP4233.252.0.3/127

[0097] a = fec-repair-flow:encoding-id = 6 ;fssi = Kmax:8192,T:128,P:A

[0098] a = mid:R1

[0099] -----

[0100] $m = \text{application30000RTP/AVP111}$

[0101] $c = \text{IN IP4233.252.0.4/127}$

[0102] $a = \text{fec-repair-flow:encoding-id} = 6 ; \text{fssi} = \text{Kmax:8192, T:128, P:A}$

[0103] $a = \text{mid:R2}$

[0104] 在移动电视频道内的示例性模拟结果：

[0105] 图 9A 和 9B 显示了在 DVB-H 中比较标准 Raptor 代码 (ST-FEC) 和层获知 Raptor (LA-FEC) 的 AL-FEC 实现方式的在移动电视 (DVB-H) 频道中模拟的示例性结果, 代替现有链路层 -FEC MPE-FEC。尤其地, 显示了以固定的业务比特率 1300kbps 通过 SVC 层 CR (基级层 / 增强层) 使用具有不同的编码率分布的标准 (ST)-FEC 和 LA-FEC 的用于 VGA 接收器的 250 帧的冻结帧的平均数 (图 9A) 和平均 PSNR 值 (图 9B)。通过 ST-FEC, 为每层生成 FEC 数据, 与其他层无关。通过 LA-FEC, 在基级层的所有源符号上生成增强层 FEC 数据。以不同的码率分布和规定的总业务比特率 (包括音频、视频以及 FEC) 1300kbps, 模拟这些设置。模拟包含总 FEC 码率 $CR = 0.68$, 由层 1 的源符号 k_1 与编码符号 n_1 的比率来计算 CR, $CR = k/m$ 。在示图的文字说明中, 显示在层上的 CR 分布, 基级层码率是第一值, 并且增强层码率是第二值。上部分图显示了在 250 个帧的传输中冻结帧的数量, 并且下部分图显示了在 PSNR 方面所产生的平均视频质量。在【1】中可以找出有关基底模拟条件的更多详情。

[0106] 从图中可以看出, 在服务连续性 (较少数量的冻结帧) 和平均视频质量 (更高的 PSNR) 方面, LA-Raptor 胜过标准 Raptor 方案。通过在 FEC 整合中将增强层的额外连接加入基级层中, 这并未增大服务比特流。

[0107] 具有 LDPC 代码的在物理信道上的 LA-FEC：

[0108] 该部分显示了 LA-FEC 可应用于物理层 LDPC 代码中的方式。作为一个示例性 FEC 代码, 我们应用 DVB-S2 的物理层的 LDPC 代码 16k 矩阵。图 10 比较了用于基级层 (BL) 的标准 LDPC 矩阵与用于增强层 (EL) 编码的扩展的 LA-FEC LDPC 矩阵。标准 LDPC 矩阵是顶部矩阵, 并且扩展的 LDPC 矩阵是底部矩阵。黑点表示在矩阵中 1s 的位置, 而白色区域包含 0。

[0109] 对于增强层 (EL), 标准 LDPC 矩阵由简单的单位矩阵扩展成基级层的源符号。这仅仅表示一个示例性扩展。人们可以考虑更复杂的矩阵扩展。在图 11 中描述了连接的解码矩阵 (扩展的 LA-FEC 解码矩阵)。在一个解码矩阵中, 使 BL 和 EL 的源和奇偶符号相结合。用于将 LDPC 代码解码的任何解码算法还可用于将该扩展矩阵解码。

[0110] 在图 12 中显示了用于物理层 LDPC 代码的一些示例性结果, 该图显示了在 AWGN 通道中的 16k LA-FEC 对 16k St-FEC 的模拟结果, 因此, 在 AWGN 通道中比较 16k LDPC 代码的标准 FEC (stFEC) 和在不同的 E_b/N_0 上模拟的 16k LDPC 代码 (LA-FEC) 的 LA-FEC 扩展。使用 QPSK, 并且应用具有等错误保护的不同码率。标绘图显示了 ST-FEC 编码 (与 BL+EL 相同的比特率) 的比特误码率 (BER) 与基级层和增强层 BER 曲线相比较。可以看出, BL 性能增大, 而 EL 性能显示了相似的性能, 与 ST-FEC 一样。通过传统的不等错误保护 (UEP) 方案, 与基级层相比, 增强层不太受保护, 并且 BER 曲线会显示增强层性能减小。这并非 LA-FEC 方案和等错误保护的情况。然而, UEP 也可用于 LA-FEC 中。

[0111] 时间交错：

[0112] 时间交错是增大信号的时间分集的一个重要方式,从而增大其抵抗突发脉冲差错的稳健性。通过充足的交错长度,服务可克服长差错突发脉冲。典型的时间交错的主要障碍是在时间或切换时间中的服务优化的增大。长时间交错要求接收器等待,直到已经接收该交错周期的所有数据包并且将其装入解交错缓冲器内。即,虽然更长的交错长度会大幅提高服务稳健性,但是如今的传输系统试图尽可能减小时间交错长度。

[0113] FEC 源块的时间分集随着某个交错长度 (IL) 增大,与在图 13 中为 FEC 源块 0 (SB_0) (n_0 符号由 k_0 源符号和 p_0 奇偶符号构成) 以及 IL(1, 2, 3) 所描述的那样。图 13 示意性示出了具有 k_0 源和 p_0 奇偶符号的 FEC 源块 0 (SB_0) 的不同交错长度 (IL) 和增加的时间分集 (IL = 1, 2, 3)。

[0114] 恢复源块 SB_0 的所有符号 $n_0 = 4$ 所需要的时间在符号方面称为交错延迟 d_{interl} 。

[0115] 层获知 FEC 整合到 RaptorQ FEC 内:

[0116] 该部分显示了 LA-FEC 可整合到 RaptorQ[2]FEC 代码内的方式。RaptorQ 是 Raptor 代码的进化版本 (比较“层获知 FEC 整合到 Raptor FEC 内”部分),其通过某些进一步的优化提高了性能。

[0117] 在【2】中规定 RaptorQ 工序,并且该工序由用于进行编码和解码的相同两个步骤构成,与 Raptor 代码工序一样 (比较“层获知 FEC 整合到 Raptor FEC 内”部分)。要注意的是,通过具有相同的编码和解码工序的方式,限定编码工序。第一步骤是预编码工序,该工序输入固定尺寸的 k_0 源符号并且生成 L_0 预编码符号。未规定 RaptorQ 用于所有 k_0 值。因此,通过填充到下一个可用的 k_0' 中,源符号 k_0 的数量增大。在图 14 中显示了在【2】中规定的预编码矩阵。图 14 示意性示出了与在【2】中规定的那样用于层 0 和单层的 RaptorQ 的预编码矩阵。要注意的是,在解码和编码工序中使用相同的预编码矩阵。

[0118] 预编码矩阵由两个 LDPC 子矩阵 ($G_LDPC, 1_0$ 和 $G_LDPC, 2_0$) (所谓的 G_HDPC_0 矩阵)、两个单位矩阵 (I_{S_0} 和 I_{H_0}) 以及包含 k_0' 行的 G_ENC_0 矩阵构成。虽然要求 G_ENC_0 保持系统代码,但是 $G_LDPC, 1_0$ 、 $G_LDPC, 2_0$ 、 G_HDPC_0 、 I_{S_0} 、 I_{H_0} 构成 RaptorQ 工序的实际预编码。在【2】中可找出关于生成子矩阵的更多详情。要注意的是,整个预编码矩阵由源符号 k_0 的数量、源符号本身以及符号尺寸 T 生成。

[0119] 第二步骤构成在图 15 中所示的 RaptorQ 的 LT 编码工序。图 15 示意性示出了 RaptorQ 的 LT 编码工序。要注意的是,在矩阵 G_ENC_0 内的密度分布与 RaptorQ 工序的 G_LT_0 矩阵不同 (比较“层获知 FEC 整合到 RaptorFEC 内”部分)。第二步骤从第一步骤中采用 L_0 预编码符号,并且通过 XORing 不同组的预编码符号,生成 n_0 编码符号。由 G_ENC_0 矩阵规定这些组,该矩阵是稀疏矩阵并且由 ‘0’ 和 ‘1’ 构成。‘1’ 表示相关的预编码符号包含在相关编码符号的 XORing 工序内。

[0120] 在图 16 中为两层显示了 LA-FEC RaptorQ 的预编码矩阵,即,图 16 示意性示出了具有两层的 LA-FEC RaptorQ 的预编码工序。该矩阵由这两层的两个预编码矩阵 (左上和右下)、零矩阵 (右上) 以及从层 1 到层 0 的 LA-FEC 扩展 (左下) 的串联构成。要求 LA-FEC 扩展保持系统代码。要注意的是, G_ENC_1 和相关的扩展 $G_ENC_0[n_0:n_0+k_1-1]$ 行在位置 n_0 处开始。一方面是因为层 0 的线性独立行的数量增大,另一方面是因为允许编码符号的唯一识别符复制在接收器上的矩阵。在图 15 中的工序之后,生成层 0 的编码符号。为了生成增强层编码符号,将这两层的预编码符号转发给在图 17 中描述的 RaptorQ 编码工序的下一个

步骤。图 17 示意性示出了具有两层的 LA-FEC RaptorQ 编码工序的第二步骤以及扩展为层 0 的预编码符号的矩阵。

[0121] 考虑层 0 的预编码符号和相关的编码矩阵 G_ENC_0 ，由扩展的编码工序生成层 1 的编码符号 n_1 。由于在预编码步骤中包含最初的 k 行扩展的 $G_ENC = [G_ENC_0 G_ENC_1]$ 矩阵，所以所产生的编码符号具有系统性。应注意的是，解码工序包括相同的步骤，但是用所接收的符号代替源符号并且用源符号代替编码的符号。

[0122] 具有不等时间交错的 LA-FEC (LA-FEC UI) :

[0123] 层获知 FEC 和分层的媒体 (例如, SVC 和 MVC) 的组合能够具有一种新的业务提供方式, 该方式与快速切换和长期交错匹配。实施方式尤其涉及在该部分以及以下部分中描述的具有不等时间交错的 LA-FEC。

[0124] 快速切换 (zapping) 具有质量更低的基级层, 并且短时间交错具有抵抗突发脉冲差错的低稳健性, 但是具有低延迟。由 SVC 增强层提供长期交错, 具有更强的抵抗突发脉冲丢失的稳健性, 但是具有更长的延迟。然而, 由于在基级层上通过层获知 FEC 将增强层进行 FEC 编码, 所以基级层还受益于增强层的改进的时间分集。

[0125] 该方案在下文中称为具有 LA-FEC 或 LA-FEC UI 的不等时间交错。重要的是, 注意 LA-FEC UI 不仅可应用于 SVC 或任何其他类型的分层媒体中, 而且还应用于任何类型的时间同步数据中。例如, 一个可能的应用程序还具有短时间交错的音频流以及长时间交错的视频流, 该音频流和视频流由 LA-FEC 连接。图 18 示出了 LA-FEC UI 的工序 (即, 根据在本文中公开的教导内容, 用于进行纠错编码的方法) 并且提供了一个说明性实例。换言之, 图 18 示意性示出了发送器侧工序: 具有通过 LA-FEC 编码的不等时间交错的 LA-FEC、不等时间交错以及延迟用于进行同步的更短交错的数据包的工序。

[0126] 图 18 示出了用于在发送器侧上生成 LA-FEC UI 流的第四步。

[0127] LA-FEC UI 工序的步骤 1(1) 是媒体编码本身。编码流可为分层的媒体流, 例如, 具有 SVC 或 MVC, n 媒体层具有在层 0 中的基级层 (属于第一类的有效载荷数据) 和在更高层内的相关增强层 (属于第二类 (并且可能属于第三类、第四类...) 的有效载荷数据)。另一个有意义的设置可在层 0 内具有音频流并且在层 1 中具有相关的视频流。在图 18 中所描述的实例显示了两层, 每层具有一个不同的比特率。每层的时间周期 Δt 的比特流分成 k 个源符号, 其中, 由于在这两层中的不同的比特流, 所以 $k_{layer0} = 1$ 并且 $k_{layer1} = 2$ 。源符号包含实际信息并且是用于进一步 FEC 处理的输入。步骤 1(1) 可被视为接收属于第一类的有效载荷数据的步骤以及接收属于第二类的有效载荷数据的步骤。

[0128] 该工序的步骤 2(2) 是 LA-FEC 工序本身, 例如, 基于示例性 Raptor 代码, 如在“层获知 FEC”部分中所述。在图 18 中的图示显示了这两层的 FEC 源块 (SB) (有效载荷数据) 和奇偶符号 p (纠错数据) (即, 第一和第二类有效载荷数据和相关纠错数据)。在该实例中, 相同的码率适用于每层, $CR = k/(k+p) = 0.5$ 。要注意的是, 也可应用不等码率分布。在 LA-FEC 工序之后, 在包含在层 0 的源块 SB_0, \dots, SB_3 内的层 0 的源符号之上, 通过标准的 RaptorQ 算法, 生成在层 0 中的单个奇偶符号 p (第一类有效载荷数据的纠错数据)。通过层 0 的源符号, 在扩展的源块之上, 生成层 1 的 $p = 2$ 奇偶符号 (第二类有效载荷数据的纠错数据)。步骤 2(2) 可被视为确定第一类有效载荷数据的第一纠错数据的步骤以及确定第二类有效载荷数据的第二纠错数据的步骤。

[0129] LA-FEC UI 工序的步骤 3(3) 是不等交错,在图 18 的实例中由卷积交错器表示,层 1 的交错长度 $IL_1 = 4$ (在该实例中,交错长度与 FEC 源块的数量对应) 并且层 0 没有交错, $IL_0 = 1$ 。要注意的是,任何其他类型的交错器(例如,块交错器)可用于 LA-FEC UI 工序。在所述实例中,交错工序在固定尺寸的 FEC 符号上运行。重要的是,在此处注意,交错工序还可根据数据包级别来执行,例如,包含编码的 FEC 源和奇偶符号的 UDP 数据包。步骤 3(3) 可被视为至少使第二纠错数据(层 1 的奇偶符号)和第二类有效载荷数据(层 1 的符号)彼此交错的步骤,其中,相对于第一纠错数据(层 0 的奇偶符号)和第一类有效载荷数据(层 0 的符号)的交错的第一交错长度与相对于第二纠错数据和第二类有效载荷数据的交错的第二交错长度不同。在所显示的实例中,第一交错长度是 1,即, $IL_0 = 1$,这表示在该实例中,第一纠错数据和第一类有效载荷数据未进行活动交错。然而,通常,在至少使第一纠错数据和第一类有效载荷数据交错的额外步骤中,第一纠错数据可与第一类有效载荷数据交错。而且,第一纠错数据、第一类有效载荷数据、第二纠错数据以及第二类有效载荷数据可在组合的交错步骤中交错,其中,使用不同的第一和第二交错长度(公式“至少使第二纠错数据和第二类有效载荷数据交错”也包括这种情况)。

[0130] LA-FEC UI 的步骤 4(4) 表示具有更短交错的层的延迟传输,用于使这两层同步。因此,纠错编码的方法可包括延迟传输交错的第一纠错数据和第一类有效载荷数据的额外步骤。必须通过一种方式设置延迟长度,通过这种方式,在也已经接收在层 0(第一类)中的某个源块的所有符号时,已经接收相应源块的层 1(第二类)的所有符号,如在图 18 中所示。符号的延迟长度取决于这两层的实际交错长度以及在具有最高比特率的层(层 1,即,第二类)中的符号的数量。考虑在图 18 中的示例性交错,为了进行同步所需要的延迟 d 可由 $d = (IL_1 - IL_0) * (k_1 + p_1)$ 计算。

[0131] 而且,通过任何类型的传输信道,分别传输或多路复用这两个比特流。要注意的是,在该实例中,我们考虑实时传输(其中,需要在某个时间帧内在接收器中具有这两层)以及具有突发脉冲删除的信道的传输。

[0132] 相应的纠错编码器可包括纠错数据发生器,用于接收属于第一类型的有效载荷数据(第一类型有效载荷数据),用于接收属于第二类型的有效载荷数据,用于为第一类型有效载荷数据确定第一纠错数据,并且用于为第二类型有效载荷数据确定第二纠错数据。该纠错编码器可进一步包括第二纠错数据发生器,用于接收属于第二类型的有效载荷数据并且用于为第二类型有效载荷数据确定第二纠错数据。而且,交错器可为纠错编码器的一部分,用于至少使所述第二纠错数据与所述第二类型有效载荷数据彼此交错,其中,相对于第一纠错数据和第一类型有效载荷数据的交错的第一交错长度与相对于第二纠错数据和第二类型有效载荷数据的交错的第二交错长度不同。

[0133] 在本文中公开的教导内容的实施方式中,纠错数据发生器可包括第一单元(即,第一纠错数据发生器)和第二单元(即,第二纠错数据发生器)。

[0134] 图 19 显示了在传输内包括一个示例性突发脉冲差错的一个实例接收器工序。换言之,图 19 示意性示出了接收器侧:通过解交错、LA-FEC 解码的不等时间交错的 LA-FEC 的工序。

[0135] 步骤 5(5) 显示了通过容易发生错误的信道传输在步骤(4)中生成的媒体流。该传输受到影响这两层的突发脉冲差错的影响。仅仅为了简单起见,在纠错信道中概述了所

规定的实例,这通常用于在链路或应用层上模拟传输差错(见在“层获知 FEC”部分中的注释)。然而,在差错信道中还可应用相同的工序,其中,接收器并不了解差错的位置,例如,使用在“具有 LDPC 代码的在物理信道上的 LA-FEC”部分内描述的 LA-FECLDPC 代码。突发脉冲差错造成在层 0 中丢失 4 个数据包(例如,4 比特/符号)并且在层 1 中丢失 8 个数据包(例如,8 比特/符号),其中,丢失的数据包的数量取决于层的实际比特率。丢失的数据包标有十字形。假设示例性接收器在时间实例 t_0 中调谐成媒体流。要注意的是,接收器可仅仅利用完全接收的符号或数据包,如果在数据包的中间进行调谐,那么接收器由于缺少报头从而不能了解数据包,并且会丢弃该数据包。如果完全接收报头,那么可利用关于数据包的所有需要的信息。在纠错解码的方法中,步骤 5(5)可被视为接收第一类型有效载荷数据和第一纠错数据的步骤。第一类型有效载荷数据可在交错的数据流中与第一纠错数据交错,并且接收第一类型有效载荷数据和第一纠错数据的步骤可包括接收交错的数据流。

[0136] 步骤 6(6)对所接收的媒体流进行解交错(即,将包括(至少)第二类型有效载荷数据和第二纠错数据的交错的数据流解交错)。这要求接收器意识到每层 IL_0 和 IL_1 的实际交错时间(交错长度)。可由带内或带外信号通知交错长度。

[0137] 步骤 7(7)显示了 FEC 解码。由于在传输期间丢失所有数据包,所以层 0 的标准 FEC 不能校正 SB_1 和 SB_2 。由于 LA-FEC 概念,增强层(第二类)保护基级层(第一类),并且由于交错时间更长,所以在增强层中仅仅丢失一个数据包。即,在增强层内接收充足的数据包,以满足等式(2)和(3),这允许在 SB_1 和 SB_2 中校正这两层。换言之,步骤 7(7)与至少使用第一纠错数据(即,第一纠错数据和第二纠错数据)来校正第一类型有效载荷数据中可能具有的差错的步骤对应。而且,至少使用第二纠错数据,来校正第二类型有效载荷数据。第二类型有效载荷数据可进一步由额外的纠错数据(例如,属于第三类型有效载荷数据的第三纠错数据)(例如,层 2、层 3...)保护。与基级层的交错相比,增强层的交错时间更长,这表示相对于第一类型有效载荷数据和第一纠错数据的交错的第一交错长度与相对于第二类型有效载荷数据和第二纠错数据的交错的第二交错长度不同。

[0138] 步骤 8(8)显示了由 LA-FEC 和不等时间交错相结合造成的没有任何差错的恢复的视频数据。

[0139] 纠错解码器可包括解交错器和差错校正器。解交错器可被配置为执行上述步骤 6(6)的一些或所有行为。差错校正器可被配置为执行上述步骤 7(7)的一些或所有行为。尤其地,解交错器可被配置为将包括属于第二类型的有效载荷数据和相应的第二纠错数据的交错数据流解交错。要注意的是,交错的数据流可进一步包括属于第一类型的有效载荷数据以及相应的第一纠错数据。

[0140] 第一类型有效载荷数据和第一纠错数据可与第二类型有效载荷数据和第二纠错数据交错。通过交替的方式,第一类型有效载荷数据和/或相应的第一纠错数据可与包括第二类型有效载荷数据和第二纠错数据的交错的数据流分开提供。而且,第一类型有效载荷数据和第一纠错数据需要实际交错/解交错(交错长度 $\neq 1$)或者不需要实际交错/解交错(交错长度 $= 1$)。换言之,在不执行实际(活动的)交错/解交错(通过交错器/解交错器)时,所产生的交错长度为 1。第一交错长度相对于第一类型有效载荷数据和第一纠错数据。第二交错长度相对于第二类型有效载荷数据和第二纠错数据。第一交错长度与第二交错长度不同。

[0141] 差错校正器可被配置为接收第一类型有效载荷数据和第一纠错数据。而且,差错校正器可被配置从解交错器中通过解交错的形式接收第二类型有效载荷数据和第二纠错数据。差错校正器可至少使用第一纠错数据校正第一类型有效载荷数据中可能具有的差错。差错校正器可至少使用第二纠错数据校正第二类型有效载荷数据中可能具有的差错。

[0142] 在实现层获知前向纠错(LA-FEC)时,校正在第一类型有效载荷数据(例如,基级层数据)内可能具有的差错,可使用第一纠错数据以及第二纠错数据(例如,增强层的纠错数据)。如上所述,可通过这种方式更有效地对第一有效载荷数据进行纠错,通常在减少第二类型有效载荷数据的纠错能力的情况下。尤其在第二类型有效载荷数据(增强层)取决于第一类型有效载荷数据(基级层)的情况下,可使用这种纠错方案。

[0143] 快速切换、与单层媒体的长期交错以及 FEC 源块交错:

[0144] 一种相似的机构可应用于单层编码中。图 20 显示了 FEC 源块交错的工序。换言之,图 20 示意性示出了用于快速调谐的 FEC 源块交错以及长期交错的工序。

[0145] 步骤 1(1) 显示了单层(层 0)媒体流。每个时间单位 Δt 具有 k 个源符号。在该实例中, $k = 2$ 。

[0146] 步骤 2(2) 显示了具有 FEC 源块(SB_x)长度 Δt 和 p 奇偶符号的 FEC 编码。在该实例中, $p = 2$ 。FEC 算法可例如为 LDPC、Raptor 或 RaptorQ 代码。

[0147] 步骤 3(3) 显示了 FEC 源块的交错,其中,交错长度 IL_0 表示交错的 FEC 源块的数量。在规定的实例中, $IL_0 = 3$ 。

[0148] 步骤 4(4) 显示了在每个 FEC 源块(SB_x)的时间 t_0 以及相关的同步点(SP_x)中通过示例性调谐进行的最终传输调度。在时间实例 t_0 处调谐的接收器可在 SP_0 处开始视频播出。此时,未接收 SB_0 的其他 FEC 符号,以便在此时限制纠错功能。在 SP_1 处,接收器可使用一个额外的奇偶符号将 SB_1 解码。然而,在 SP_2 处,已经接收了完全校正性能以及时间交错长度。

[0149] 一种相应的纠错编码器可包括:纠错数据发生器,用于接收有效载荷数据并且为有效载荷数据确定纠错数据;以及交错器,用于使纠错数据与有效载荷数据交错。在交错的数据流中,所述纠错数据在所述有效载荷数据之前。有效载荷交错长度与纠错数据的交错长度不同。有效载荷交错长度的一个可能的定义是:有效载荷交错长度描述了一个有效载荷数据块的多个有效载荷数据项目插入交错的数据流的一个相应的交错数据块内。同样,纠错数据的交错长度描述了源自所述有效载荷数据块的多个纠错数据项目插入前述交错的数据块内。在图 20 的实例中,有效载荷交错长度是 1(即,一个规定的源块的所有有效载荷符号包含在一个相应的传输块内)。纠错数据交错长度是 3($IL_{0, \text{error correcting}} = 3$),这是因为属于某个源块的至少一个纠错符号是远离包含相应的有效载荷符号的传输块的三个传输块。例如,在 t_0 处开始的传输块包含两个有效载荷符号“0”(白色方块)。相应的纠错符号(灰色方块)包含在两个前述传输块内,这两个前述传输块在 t_{-1} 和 t_{-2} 处开始(在图 20 中未明确表示)。

[0150] 可选地,所述有效载荷数据块的所有有效载荷数据项目可插入所述交错的数据块内,并且相应的纠错数据项目可插入直接或间接位于所述交错的数据块之前的至少两个交错的数据块内。

[0151] 一种用于进行纠错编码的相应方法可包括：

[0152] - 接收有效载荷数据并且为有效载荷数据确定纠错数据；

[0153] - 使纠错数据与有效载荷数据交错，其中，在交错的数据流中，纠错数据位于有效载荷数据之前，并且其中，有效载荷的交错长度与纠错数据的交错长度不同。

[0154] 就解码侧（例如，接收器侧）而言，一种相应的纠错解码器可包括解交错器和差错校正器。解交错器被配置为用于为有效载荷数据将包括有效载荷数据和纠错数据的交错数据流解交错，其中，在交错的数据流中，纠错数据在有效载荷数据之前，并且其中，有效载荷交错长度与纠错数据的交错长度不同。差错校正器被配置为用于接收有效载荷数据和纠错数据，并且用于使用纠错数据来校正有效载荷数据中可能具有的差错，其中，在仅仅接收有效载荷数据并且错过纠错数据时，有效载荷可利用，但是可能错误（例如，由于后期调谐，以便不能再捕获纠错数据）。

[0155] 一种用于进行纠错解码的相应方法可包括：

[0156] - 为有效载荷数据将包括有效载荷数据和纠错数据的交错数据流解交错，其中，在交错的数据流中，所述纠错数据在所述有效载荷数据之前，并且其中，有效载荷交错长度与纠错数据的交错长度不同；

[0157] - 接收有效载荷数据和纠错数据；以及

[0158] - 使用纠错数据来校正有效载荷数据中可能具有的差错，其中，在仅仅接收有效载荷数据并且错过纠错数据时，所述有效载荷可利用，但是可能错误。

[0159] 通过这种传输调度，可能进行快速调谐，同时在交错周期 IL_0 之后，实现相同的校正性能和时间分集。在没有 LA-FEC 时，也可实现这种传输调度，并且该调度需要在接收器侧上具有复杂的解交错机构。

[0160] LA-FEC UI 的性能实例：

[0161] 根据模拟固定的突发脉冲长度的 Gilbert-Elliott 模型，执行模拟结果。我们假设固定的平均突发脉冲长度为 9.57 个符号并且变化的平均丢失概率由纠错概率 P_{er} 表示。

[0162] 类 SVC 模拟（空间可扩展性，例如，720p 到 1080p）；

[0163] 表 1 概述了所选择的配置参数。要注意的是，该说明性实例在层 1:2 上采用了某个符号分布，因为对于每层 0.5，其通常用于 SVC 编码的视频流以及固定码率。与单层编码相比，不等码率分布可在层上实现进一步改进。

[0164] 表 1：媒体流和 FEC 编码参数：

[0165]

	单层	分层媒体
每个 FEC 源块 k 的源符号	60	层 0:20 层 1:40 (在 SVC 中的空间可扩展性的典型分布)
FEC 码率 $CR=k/n$	0.5	层 0:0.5 层 1:0.5
编码符号的数量 (源+奇偶) n	120	层 0:40 层 1:80 总数: 120
交错长度	1;9	层 0:1 层 1: 1;9
PSNR	40dB 15dB (不可解码)	层 0:30dB 层 1:40dB 15dB (不可解码)

[0166] 在图 21 中显示了对于每个单层情况在 PSNR 方面在某个纠错概率之上所接收的视频质量、独立于每层生成 FEC 的标准 FEC (ST-FEC) 以及具有不同的交错长度的 LA-FEC UI, 并且在图 22 中显示了基级层的层解码概率, 以强调 ST-FEC UI 和 LA-FEC UI 的基级层解码概率的差别。要注意的是, 交错长度为 $IL = 9$ 的单层情况不提供与其他方案相同的调谐时间, 并且仅仅包括该单层情况, 以表示单层 FEC 源块交错的潜在性能。

[0167] 图 21 显示了比较 LA-FEC、LA-FEC UI、ST-FEC、ST-FEC UI 以及单层 (SL) 编码在 PSNR 方面的视频质量。使用两个不同的交错长度 $IL = 1$ (未交错) 和 $IL = 9$ 。要注意的是, 仅仅 $SL-IL = 1$ 显示了与所有 LA-FEC UI 方案相同的调谐时间。在图 21 中的结果显示没有 UI 的 $LA-FEC IL_0 = 1; IL_1 = 1$ 胜过 ST-FEC 方案并且接近 SL 方案的 $IL = 1$ 性能。LA-FEC UI $IL_0 = 1; IL_1 = 9$ 明显胜过单层 FEC 编码, 该编码提供相同的调谐延迟 $SL-IL = 1$ 。LA-FEC UI 未达到单层 $IL = 9$ 性能。即, LA-FEC UI 方案提供在 $SL-IL_1$ 与 $SL-IL_9$ 之间的性能, 同时允许快速调谐到服务内。

[0168] 图 22A 和 22B 显示了比较基级层和增强层解码 LA-FEC、LA-FEC UI、ST-FEC、ST-FEC UI 以及单层 (SL) 编码。使用两个不同的交错长度 $IL = 1$ (未交错) 和 $IL = 9$ 。要注意的是, 仅仅 $SL-IL = 1$ 显示了与所有 LA-FEC UI 方案相同的调谐时间。在 Fehler! Verweisquelle konnte nicht gefunden werden. 22 (顶部) 中, 可以看出 LA-FEC UI 方案对基级层性能的影响。与 ST-FEC UI 相反, 其中, 通过 LA-FEC, 增强层更长的交错对基级层没有任何影响; 基级层性能增大。总体上, 结果显示了在与具有相同的调谐 (切换) 时间的方案相比较时的 LA-FEC UI 的性能增益。通过优化在层 (UEP) 上的码率分布并且通过在层上的源符号分布的另一个比率, 预期进一步的增益。

[0169] 具有作为层 0 的音频流和作为层 1 的视频流的 LA-FEC UI :

[0170] 重要的是,注意所提出的LA-FEC UI 方案也可适用于在例如音频流上的单层视频。这些部分提供了性能结果,在层 0 中具有音频流和在层 1 中具有视频流。

[0171] 表 2 概述了所选择的配置参数。要注意的是,该说明性实例在这两层 1:8 上采用了某个符号分布,因为与视频流相比,这对于视频编码流可较为现实。而且,每层采用 0.5 的固定码率。与单层编码相比,不等码率分布可在层上实现进一步改进。

[0172] 表 2:媒体流和 FEC 编码参数:

[0173]

	单层	分层媒体
每个 FEC 源块 k 的源符号	60	层 0:7 层 1:54 (采用音频(层 0)和音频(层 1)的比特率分布)
FEC 码率 $CR=k/n$	0.5	层 0:0.5 层 1:0.5
编码符号的数量(源+奇偶) n	120	层 0:14 层 1:106 总数: 120
交错长度	1;9	层 0:1 层 1: 1;9

[0174] 在图 23 中的结果表示在层 0 中的音频流和在层 1 中的视频流的设置,在层上的比特率为 1:8。该图显示了在媒体解码概率方面进行的比较,在称为单层 (SL) 的一个 FEC 源块内具有音频和视频,交错长度为 1 和 8。换言之,图 23 显示了比较 LA-FEC UI 的音频和视频层解码和单层 (SL) 编码。使用两个不同的交错长度 $IL = 1$ (未交错) 和 $IL = 9$ 。要注意的是,仅仅 $SL-IL = 1$ 显示了与所有 LA-FEC UI 方案相同的调谐时间。

[0175] 在图 23 中的结果显示通过在这两层的比特率之间更大的差值,LA-FEC UI 的增益也增大。虽然 LA-FEC UI 视频编码层几乎达到 SL 设置的性能,但是由于音频流的小源块在差错突发脉冲通道的更高差错区域内有利,所以 LA-FEC UI 音频流甚至胜过 SL 流。

[0176] Raptor 和 RaptorQ 以及 LA-FEC UI 的信令:

[0177] Raptor 和 RaptorQ 要求在带外或带内用信号传递源符号 k (带内) 和 K_{max} (SDP) 的量、符号尺寸 T、每个编码符号 X (在 IETF FEC 框架中包含在 Raptor 有效载荷 ID 内) 的唯一标识符以及在该实例中由编码 ID (SDP) 和 FEC ID (带内) 限定的所使用的 FEC 算法,用于在接收器侧上复制预编码和编码矩阵 (比较“层获知 FEC 整合到 Raptor FEC 内”部分以及“层获知 FEC 整合到 RaptorQ FEC 内”部分)。

[0178] LA-FEC 要求在共同的解码工序中可使用增强层符号和基级层符号的信令 (比较“LA-FEC 的信令”部分)。

[0179] 为了在接收器侧上对解交错工序进行有效的内存管理,LA-FEC UI Raptor 或 RaptorQ 另外要求所有涉及层的 FEC 编码符号的交错长度 (IL) 的信令 (比较“时间交错”

部分)。

[0180] 下面由潜在的新参数《a:fec.interleaving:200ms》显示使用 SDP 的一个示例性带外信令(比较“LA-FEC 的信令”部分)。

[0181] v = 0

[0182] o = Cornelius11223344551122334466IN IP41a-fec.example.com

[0183] s = SVC LA-FEC Raptor Example

[0184] t = 0 0

[0185] a = group:FEC-FR S1 S2 R1 R2

[0186] a = group:FEC-FR S1 R1

[0187] c = IN IP4233.252.0.1/127

[0188] a = group:DDP S1 S2

[0189] m = video20000RTP/AVP96

[0190] b = AS:90

[0191] a = framerate:15

[0192] a = rtpmap:96H264/90000

[0193] a = fec-source-flow:id = 0

[0194] a = mid:S1

[0195] m = video20002RTP/AVP97

[0196] b = AS:64

[0197] a = framerate:15

[0198] a = rtpmap:97H264-SVC/90000

[0199] a = fec-source-flow:id = 1

[0200] a = mid:S2

[0201] a = depend:97lay S1:97

[0202] m = application30000RTP/AVP110

[0203] c = IN IP4233.252.0.3/127

[0204] a = fec-repair-flow:encoding-id = 6 ;fssi = Kmax:8192, T:128, P:A ;

[0205] a = fec-interleaving:200ms

[0206] a = mid:R1

[0207] m = application30000RTP/AVP111

[0208] c = IN IP4233.252.0.4/127

[0209] a = fec-repair-flow:encoding-id = 6 ;fssi = Kmax:8192, T:128, P:A

[0210] a = fec-interleaving:1000ms

[0211] a = mid:R2

[0212] 另一种可能性是通过数据包报头进行带内信令,如在图 24 中所述,该图示意性示出了具有源符号的数量k的交错长度(IL)、源符号尺寸T、唯一符号识别符(FEC ID)以及实际源块的交错长度(IL)的示例性带内信号。在SDP实例中,IL值与a = fec-interleaving参数对应。

[0213] 超过先有技术水平的进步:

[0214] - 不等时间交错和 LA-FEC 共同造成了快速切换（业务调谐）和长期交错。短时间交错的基级层允许快速调谐。由于 LA-FEC，所以增强层具有保护基级层的长期交错。

[0215] - 调谐成服务的接收器首先开始将基级层解码。在过渡周期之后，可将增强层解码。由于 LA-FEC，通过更长时间地交错增强层，整个服务稳健性增大。

[0216] - 通过延迟基级层传输以便使具有不等时间交错长度的基级和增强层保持同步的方式，调度这两层的传输调度。

[0217] - 交错具有单层和分层媒体编码的 FEC 源块。

[0218] - 在接收器侧上用信号传递每层的 FEC 源块的交错长度。

[0219] - LA-FEC 还可通过任何同步数据应用于单层中，例如，单层及其相关的音频流。在现有 LA-FEC 专利申请 US2010/017686A1 中不考虑这个。

[0220] 虽然在设备的背景下已经描述了一些方面，但是这些方面显然还表示相应方法的描述，其中，一个模块或装置与一个方法步骤或方法步骤的一个特征对应。同样，在方法步骤的背景下描述的方面还表示一个相应设备的相应模块或项目或特征的描述。一些或所有方法步骤可由（使用）硬件设备（例如，微处理器、可编程计算机或电子电路）执行。在一些实施方式中，某个或某些最重要的方法步骤可由这种设备执行。

[0221] 发明的编码 / 压缩信号可储存在数字储存介质上或者可在传输介质（例如，无线传输介质或有线传输介质，例如，互联网）上传输。

[0222] 根据某些实现要求，本发明的实施方式可在硬件内或软件内实现。使用数字储存介质，例如，在其上储存电子可读控制信号的软盘、DVD、蓝光、CD、ROM、PROM、EPROM、EEPROM 或 FLASH 内存，可进行该实施，这些控制信号与可编程计算机系统配合（或者能够与其配合），以便执行各个方法。因此，数字储存介质可具有计算机可读性。

[0223] 根据本发明的一些实施方式包括数据载体，该载体具有电子可读控制信号，这些控制信号能够与可编程计算机系统配合，以便执行在本文中描述的一种方法。

[0224] 通常，本发明的实施方式可作为具有程序代码的计算机程序产品来实现，在计算机程序产品在计算机上运行时，该程序代码进行操作，用于执行一种方法。程序代码可例如储存在机器可读载体上。

[0225] 其他实施方式包括用于执行在本文中描述的一种方法的计算机程序，该计算机程序储存在机器可读载体上。

[0226] 换言之，因此，发明方法的一个实施方式是具有程序代码的计算机程序，用于在计算机程序在计算机上运行时执行在本文中描述的一种方法。

[0227] 因此，发明方法的另一个实施方式是数据载体（或数字储存介质或计算机可读介质），该数据载体包括在其上记录的计算机程序，用于执行在本文中描述的一种方法。

[0228] 因此，发明方法的另一个实施方式是表示计算机程序的数据流或一系列信号，用于执行在本文中描述的一种方法。该数据流或这系列信号可例如被配置为通过数据通信连接（例如，通过互联网）来传输。

[0229] 另一个实施方式包括处理装置，例如，计算机或可编程逻辑装置，该处理装置被配置为或者适合于执行在本文中描述的一种方法。

[0230] 另一个实施方式包括在其上安装计算机程序的计算机，用于执行在本文中描述的一种方法。

[0231] 在一些实施方式中,可编程逻辑装置(例如,现场可编程门阵列)可用于执行在本文中描述的方法的一些或所有功能。在一些实施方式中,现场可编程门阵列可与微处理器配合,以便执行在本文中描述的一种方法。通常,这些方法优选地由任何硬件设备执行。

[0232] 上述实施方式仅仅用于说明本发明的原理。要理解的是,对于本领域的技术人员,在本文中描述的设置和细节的修改和变更显而易见。因此,其目的在于,仅仅由未决专利权利要求的范围限制,而不被通过描述和解释在本文中的实施方式提出的具体细节限制。

[0233] 参考文献:

[0234] [1]A. Shokrollahi, "Raptor Codes," IEEE Transactions on Information Theory, Vol. 52, Issue6, pp. 2551-2567, 2006.

[0235] [2]M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery," IETF RMT, draft-ietf-rmt-bb-fec-raptorq-04, August2010, <http://tools.ietf.org/html/draft-ietf-rmt-bb-fec-raptorq-04>.

[0236] [3]M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery," IETF RFC5053, Internet Engineering Task Force (IETF), Network Working Group, September2007, <http://tools.ietf.org/html/rfc5053>.

[0237] [4]C. Hellge, T. Schierl, and T. Wiegand, "Mobile TV using scalable video coding and layer-aware forward error correction," IEEE International Conference on Multimedia and Expo (ICME'08), Hanover, Germany, June2008.

[0238] [5]Cornelius Hellge, David Gomez-Barquero, Thomas Schierl, and Thomas Wiegand: Layer-Aware Forward Error Correction for Mobile Broadcast of Layered Media,

IEEE Transactions on Multimedia, vol. 13, pp. 551-562, 2011.

[0240] [6]H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-time Applications", IETF STD0064, RFC3550, July2003, <http://tools.ietf.org/html/rfc3550>.

[0241] [7]J. Postel, "User Datagram Protocol", IETF STD6, RFC768, September1981, <http://tools.ietf.org/html/rfc768>.

[0242] [8]S. Wenger, Y. -K. Wang, T. Schierl, and A. Eleftheriadis, "RTP payload format for SVC video", IETF AVT, RFC6190, May2011, <http://tools.ietf.org/html/rfc6190>.

[0243] [9]Y. -K. Wang, T. Schierl, "RTP Payload Format for MVC Video," work in progress, IETF AVT, draft-ietf-payload-rtp-mvc-01.txt, September2011, <http://tools.ietf.org/html/draft-ietf-payload-rtp-mvc-01>.

[0244] [10]M. Handly, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol", IETF RFC4566, July2006, <http://tools.ietf.org/html/rfc4566>.

[0245] [11]T. Schierl and S. Wenger, "Signaling media decoding dependency in Session Description Protocol (SDP)," IETF MMUSIC, April2009, <http://tools.ietf.org/html/rfc4566>.

org/html/rfc5583.

[0246] [12] M. Watson, A. Begen, V. Roca, "Forward Error Correction (FEC) Framework," IETF FECFRAME, RFC6363, October 2011.

[0247] [13] M. Watson, T. Stockhammer, M. Luby, "Raptor FEC Schemes for FECFRAME," IETF FECFRAME, draft-ietf-fecframe-raptor-05, September 2011, <http://tools.ietf.org/html/draft-ietf-fecframe-raptor-05>.

[0248] [14] M. Watson, T. Stockhammer, "RTP Payload Format for Raptor FEC", IETF FECFRAME, draft-ietf-fecframe-rtp-raptor-05, October 2011, <http://tools.ietf.org/html/draft-ietf-fecframe-rtp-raptor-05>.

[0249] [15] A. Begen, "Session Description Protocol Elements for the Forward Error Correction (FEC) Framework", IETF FECFRAME, RFC6364, October 2011, <http://tools.ietf.org/html/rfc6364>.

[0250] [16] A. Begen, "Forward Error Correction Grouping Semantics in the Session Description Protocol", IETF FECFRAME, RFC5956, September 2010, <http://tools.ietf.org/html/rfc5956>.

[0251] [17] G. Camarillo, H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework," IETF MMUSIC, RFC5888, June 2010, <http://tools.ietf.org/html/rfc5888>.

[0252] [18] Patent Application LA-FEC (Publication No. US2010/017686A1)

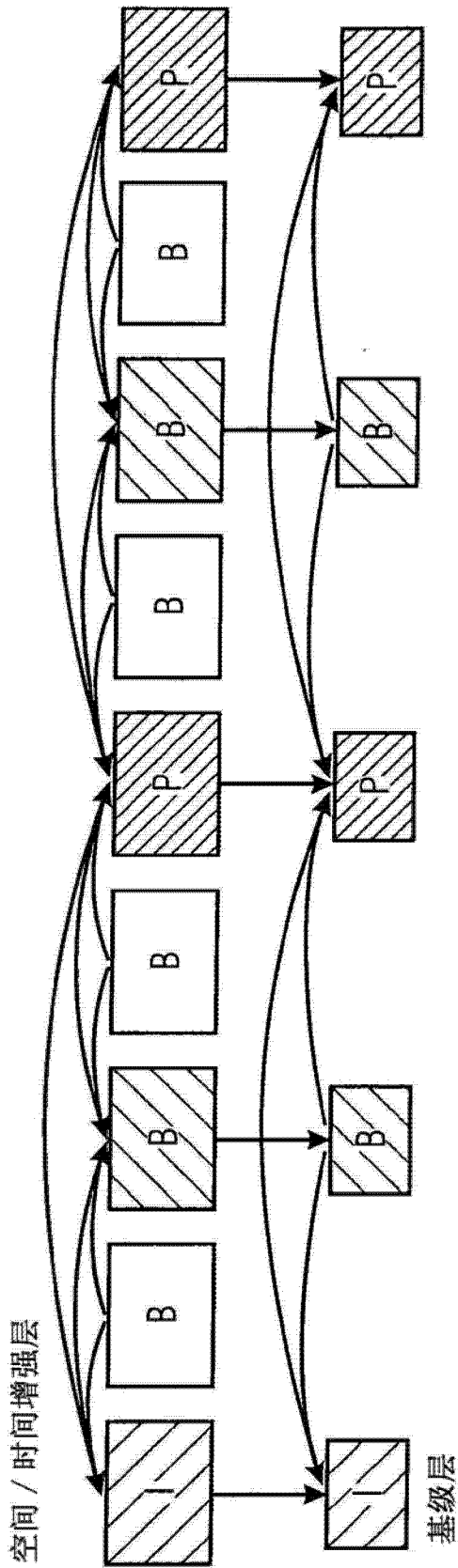


图 1

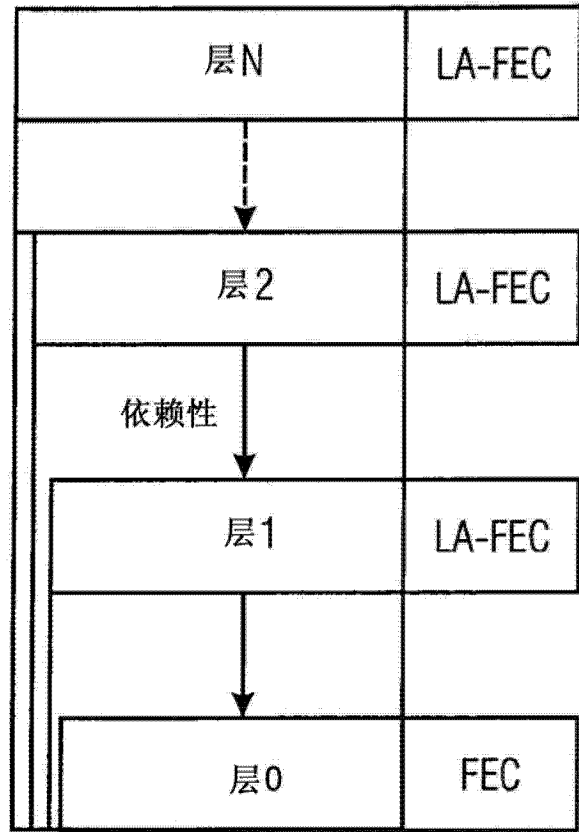


图 2

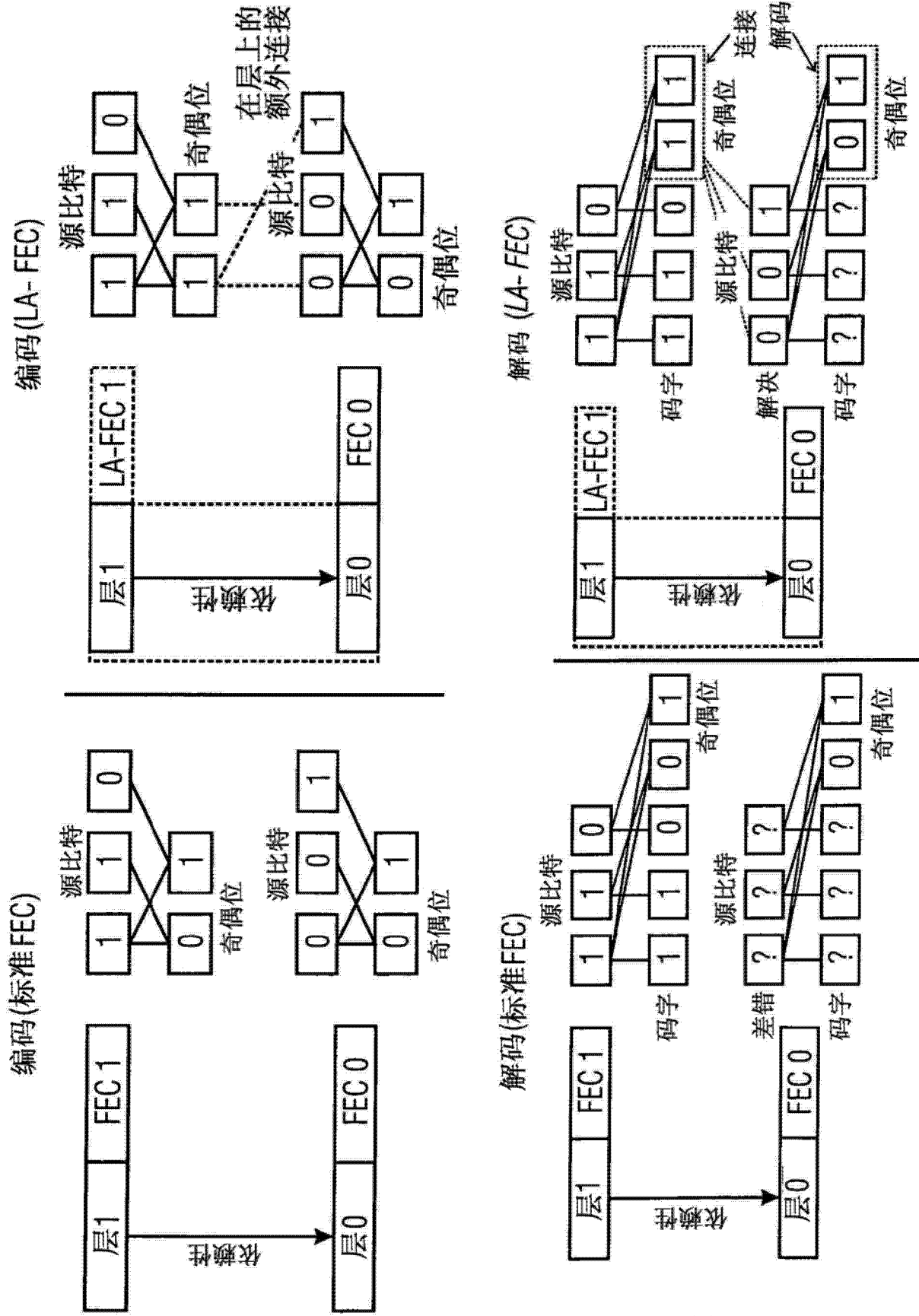


图 3

图 4

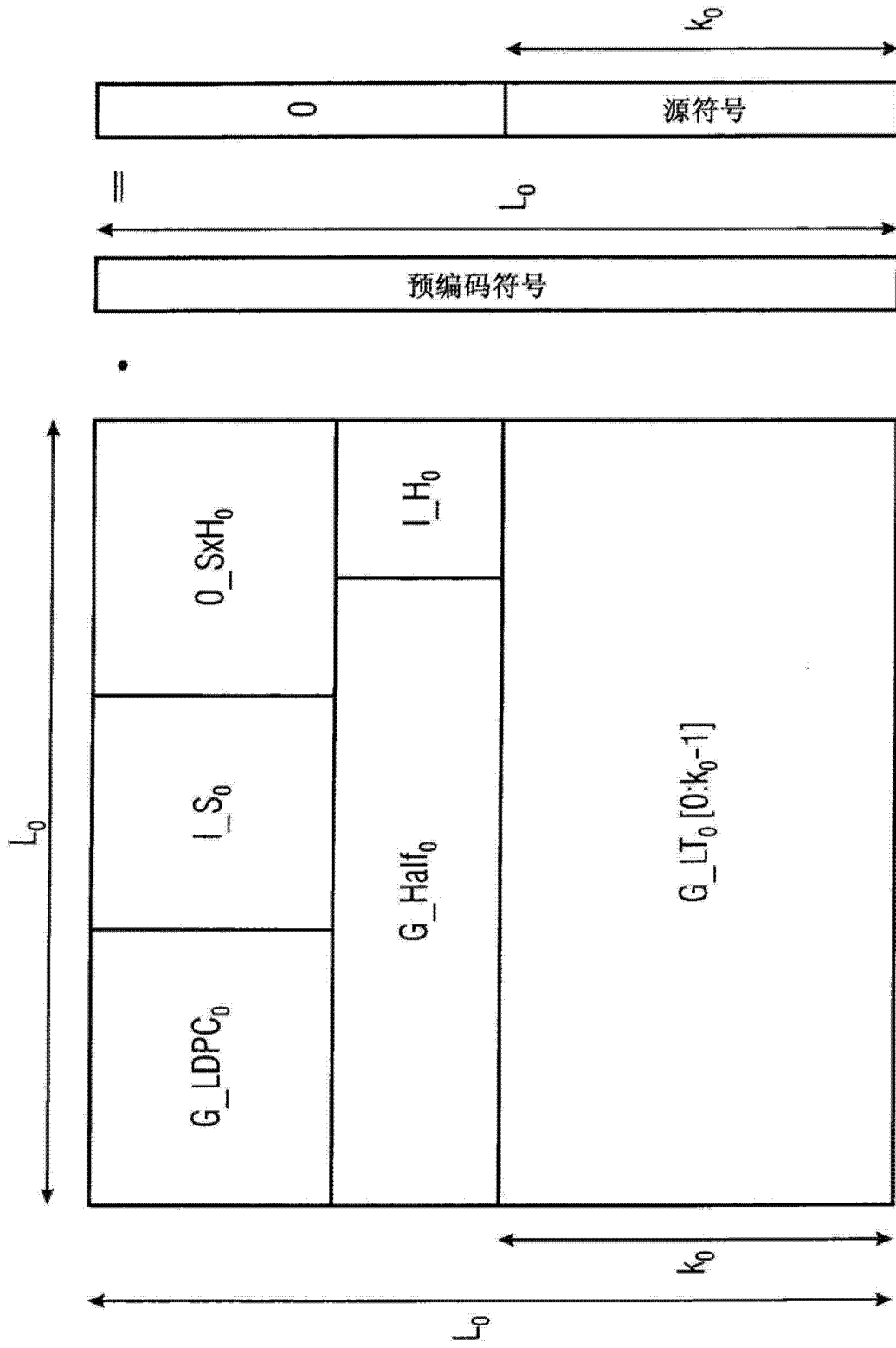


图 5

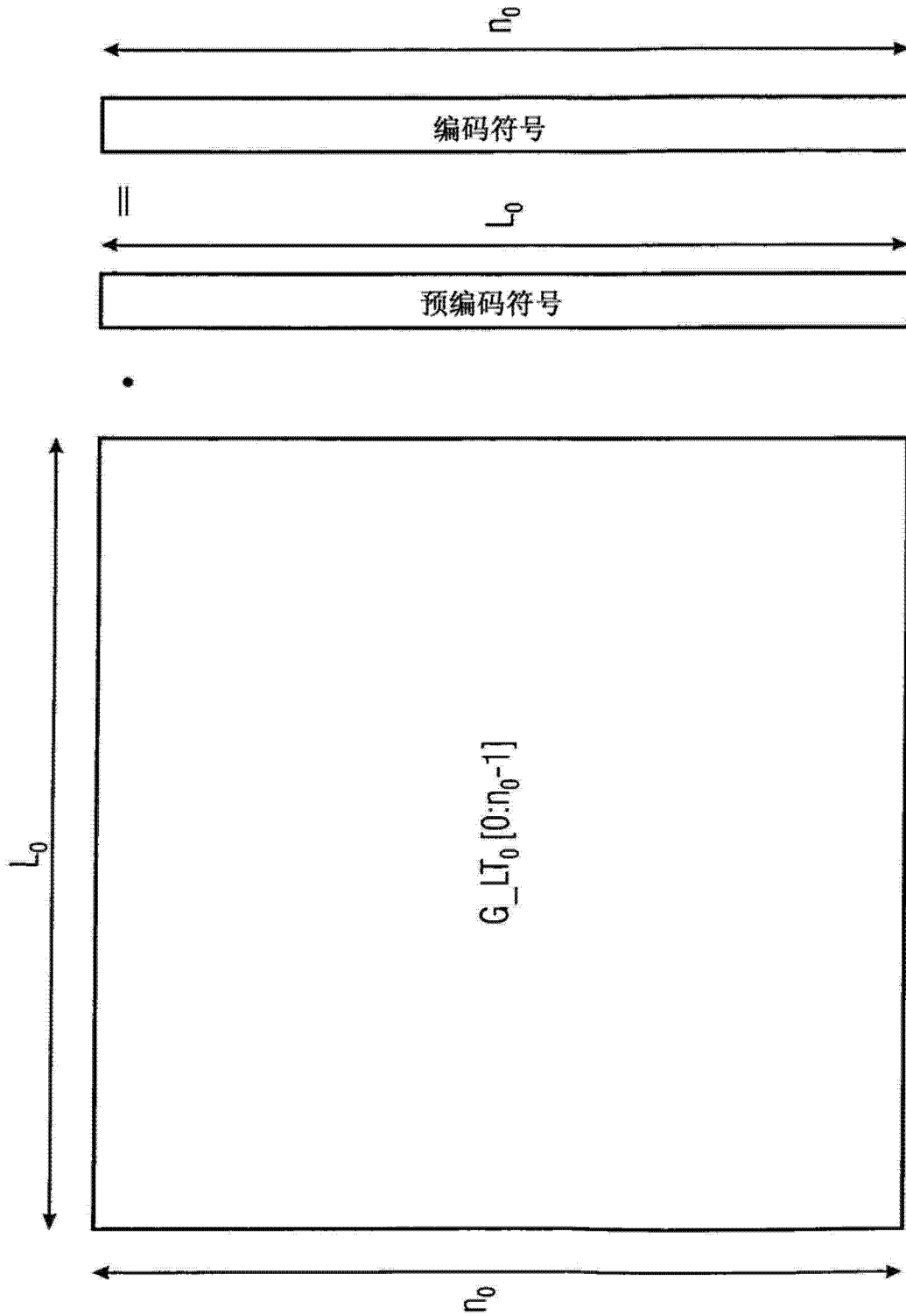


图 6

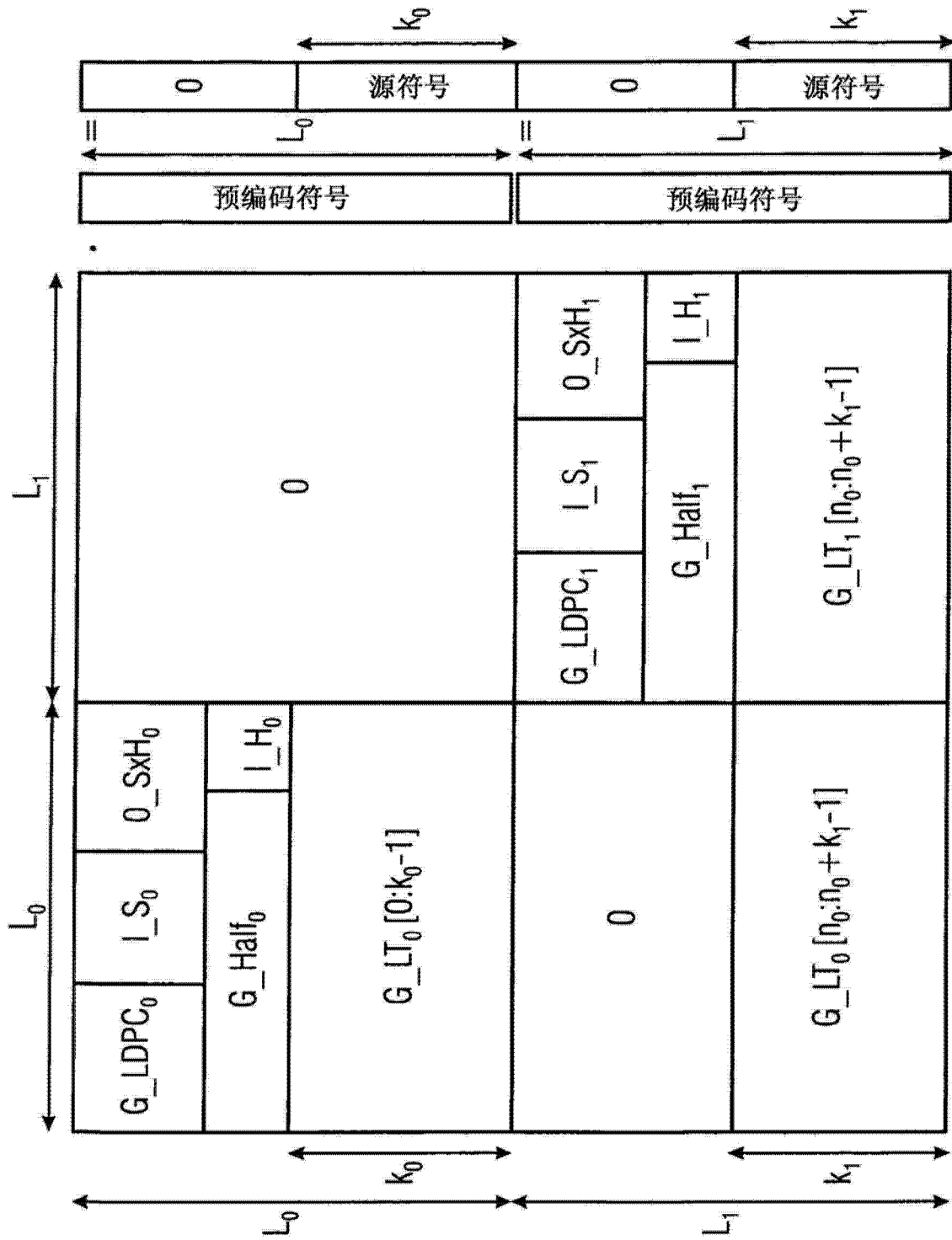


图 7

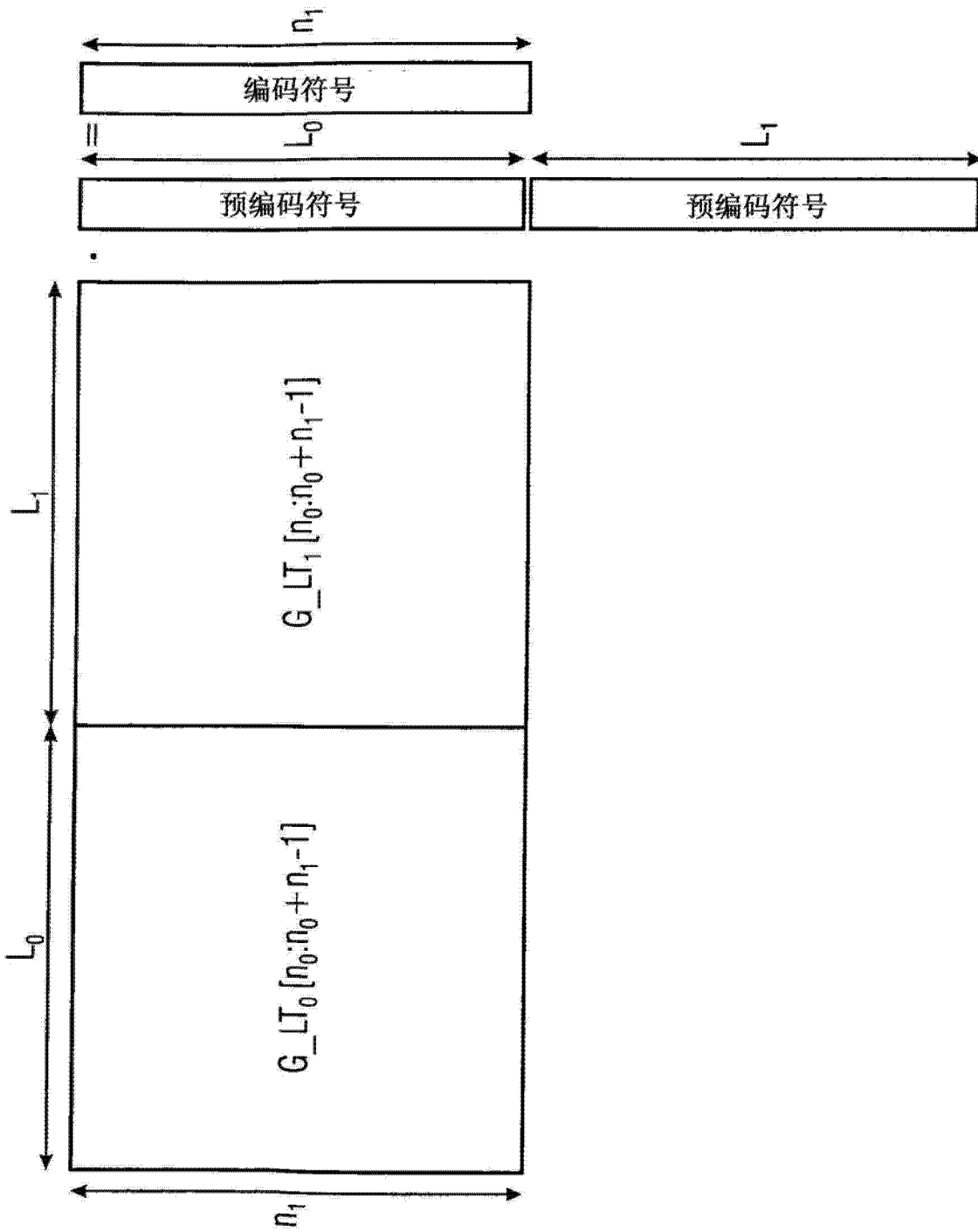


图 8

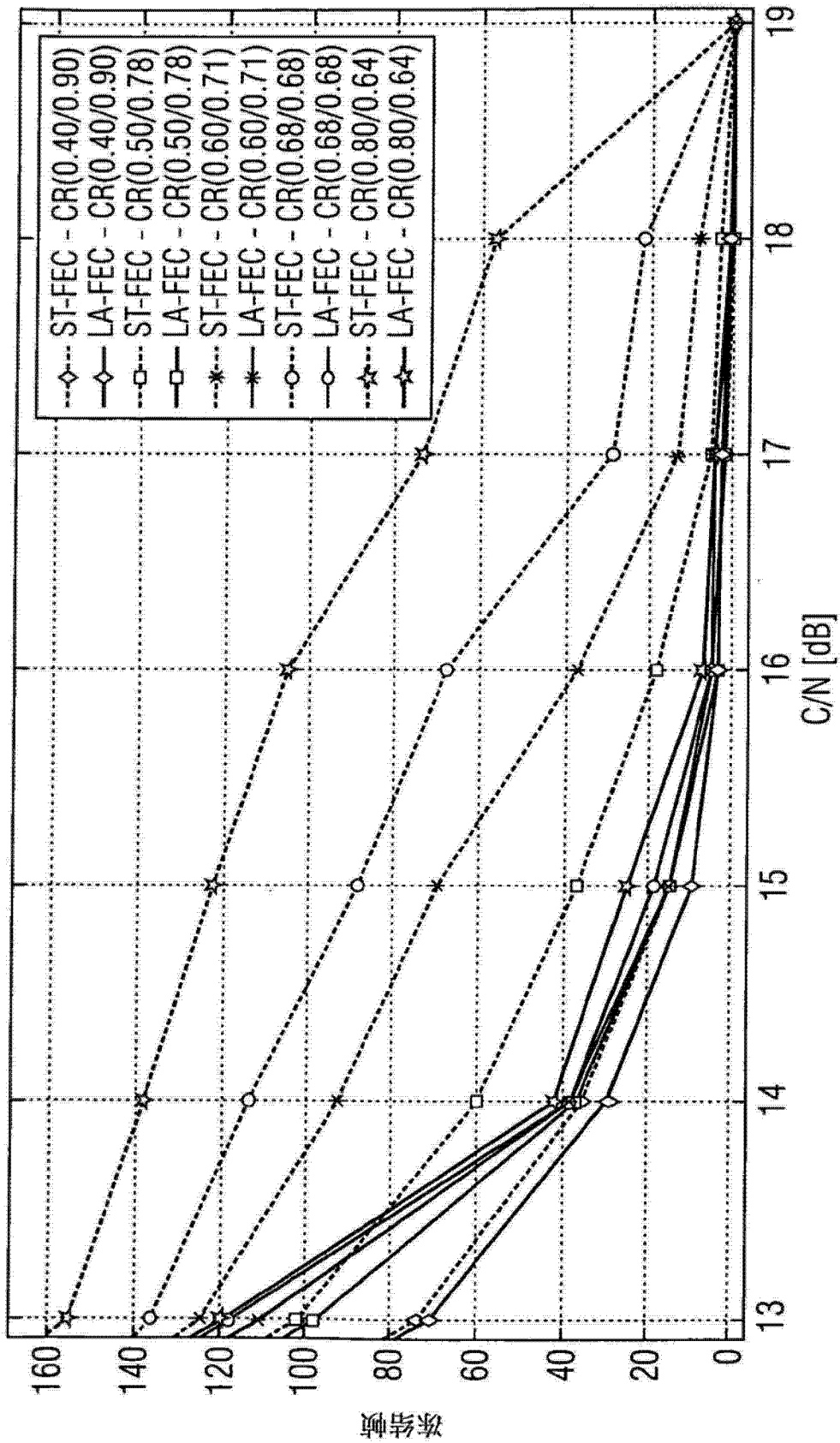


图 9A

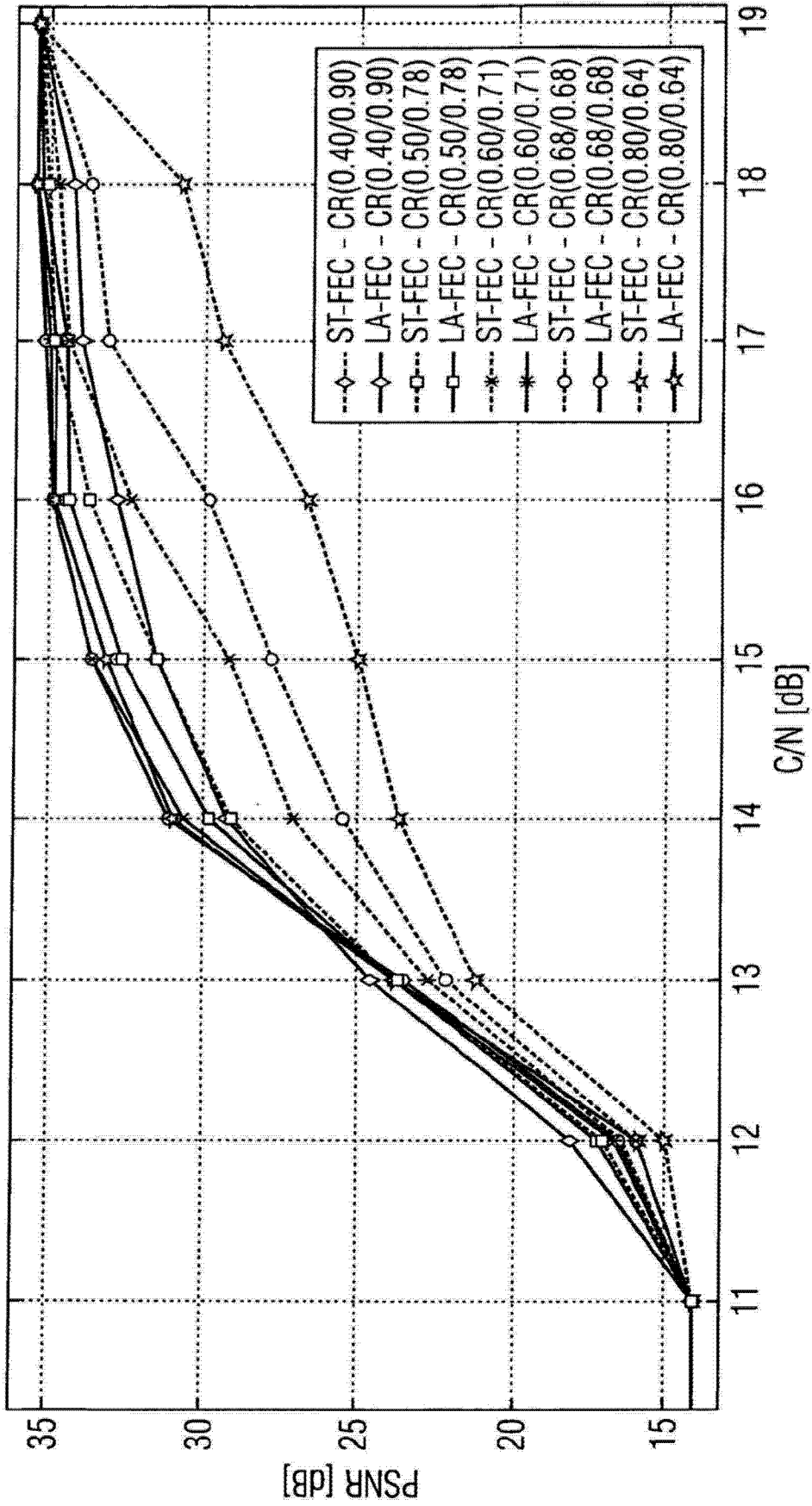


图 9B

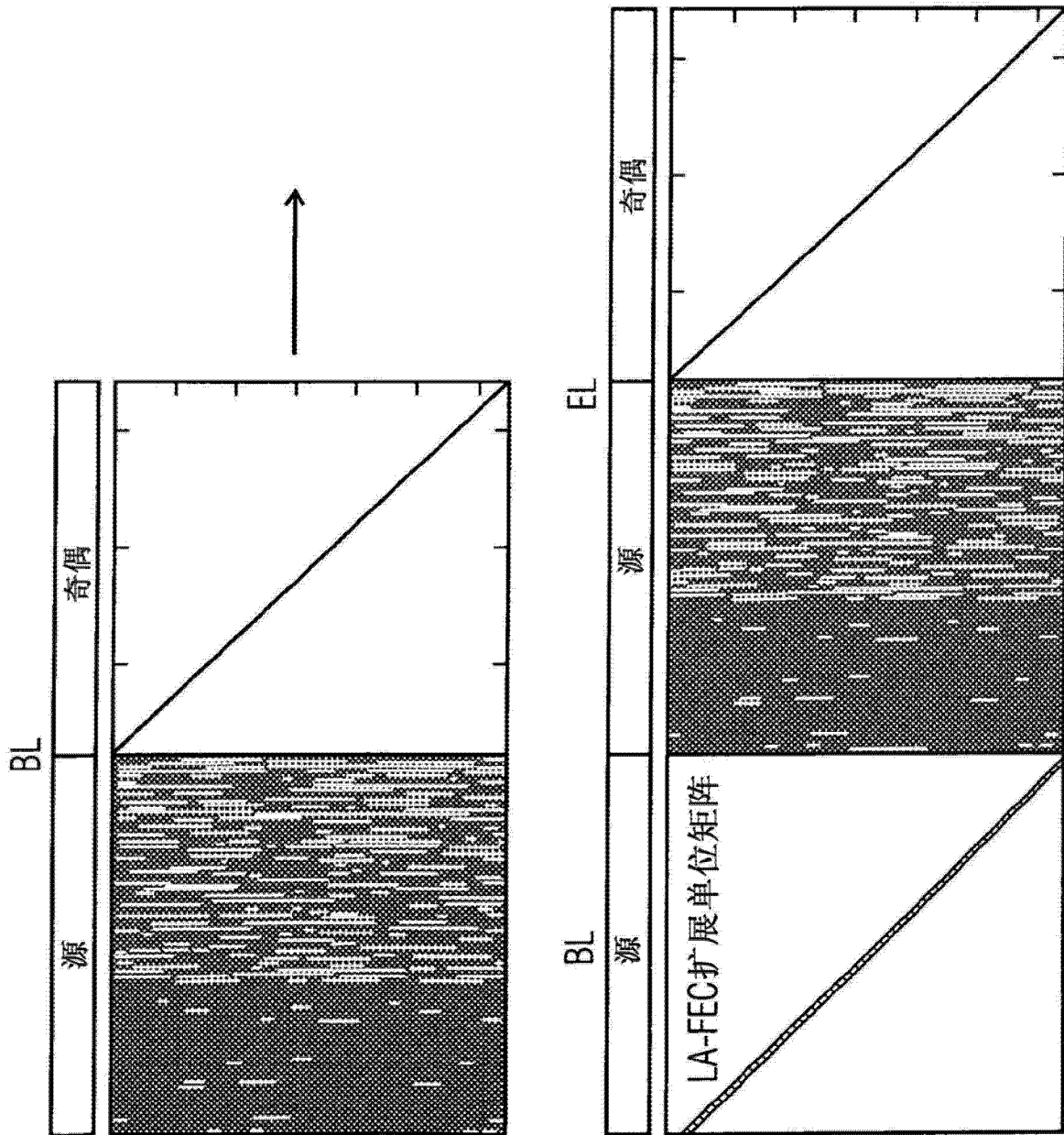


图 10

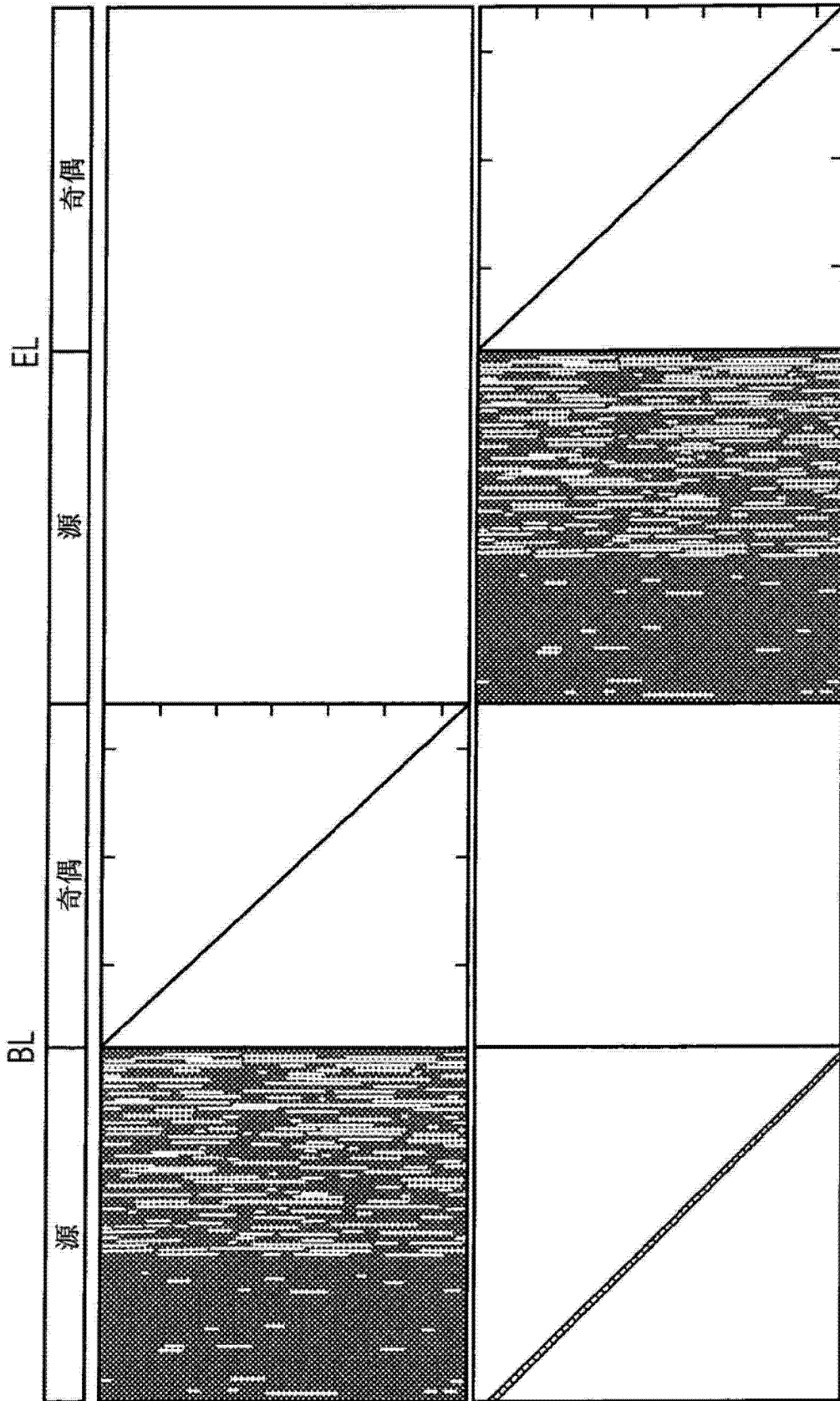


图 11

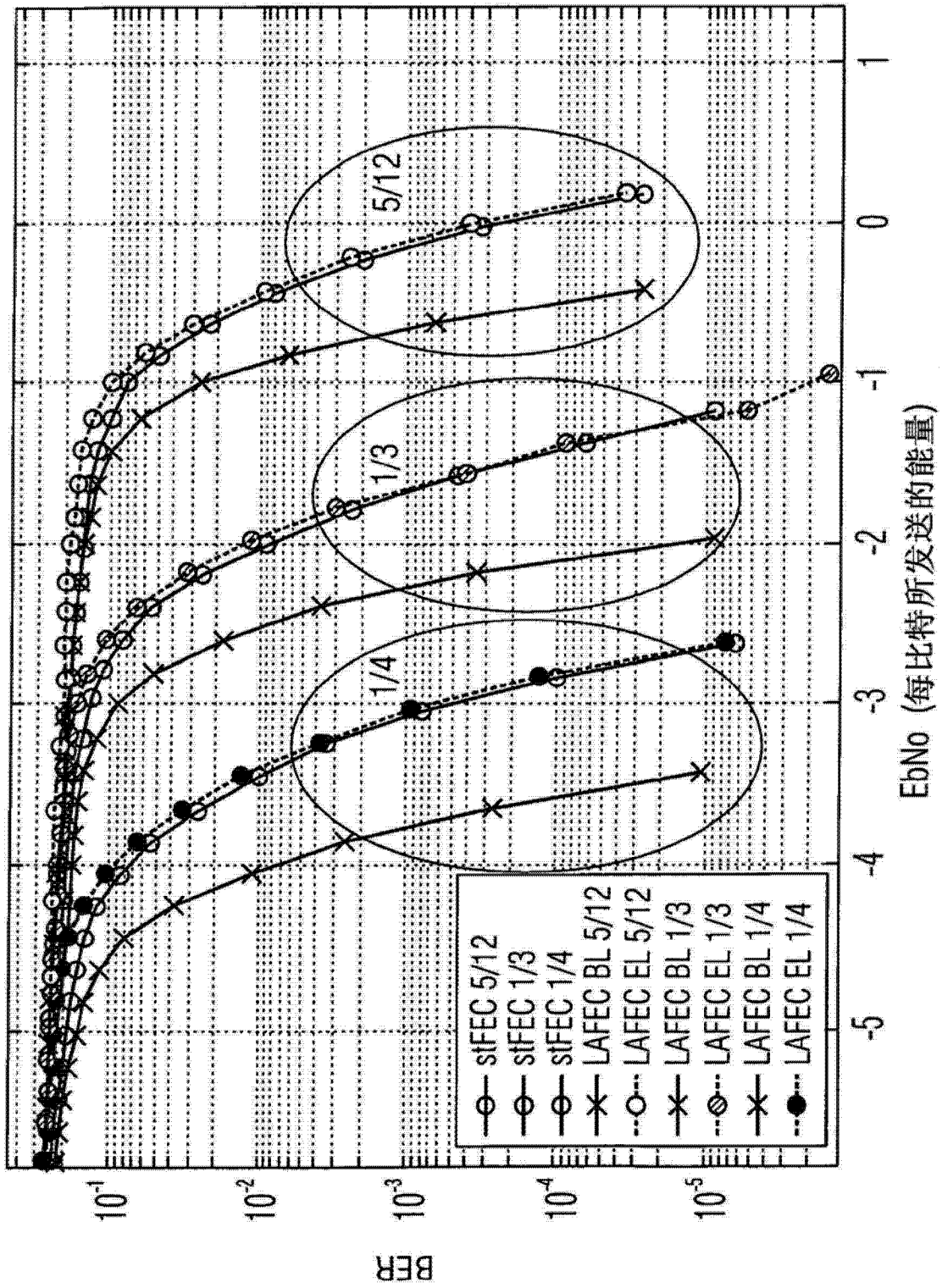


图 12

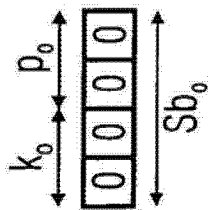
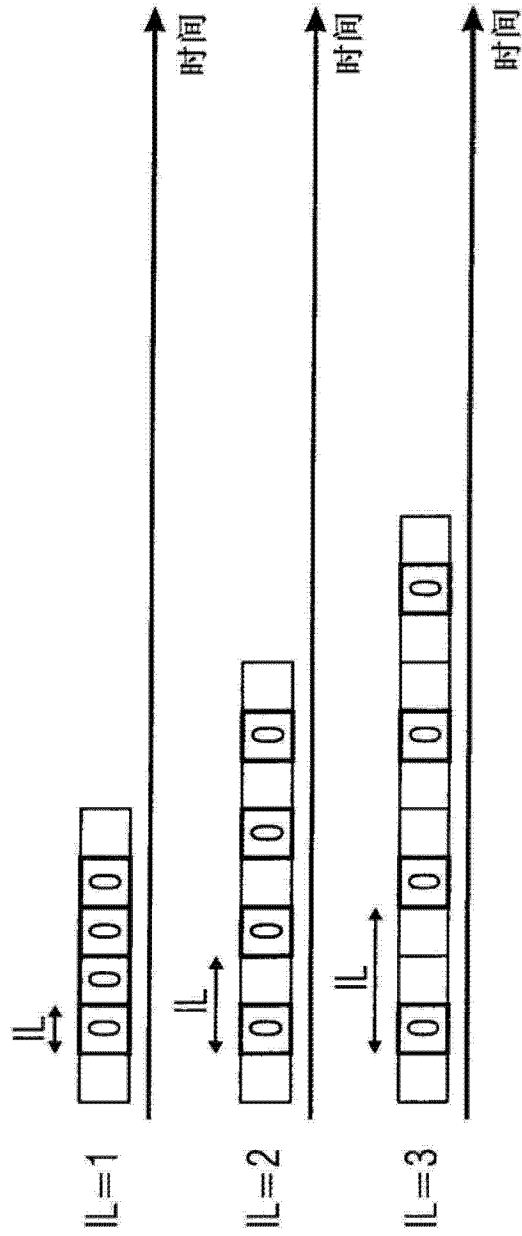


图 13

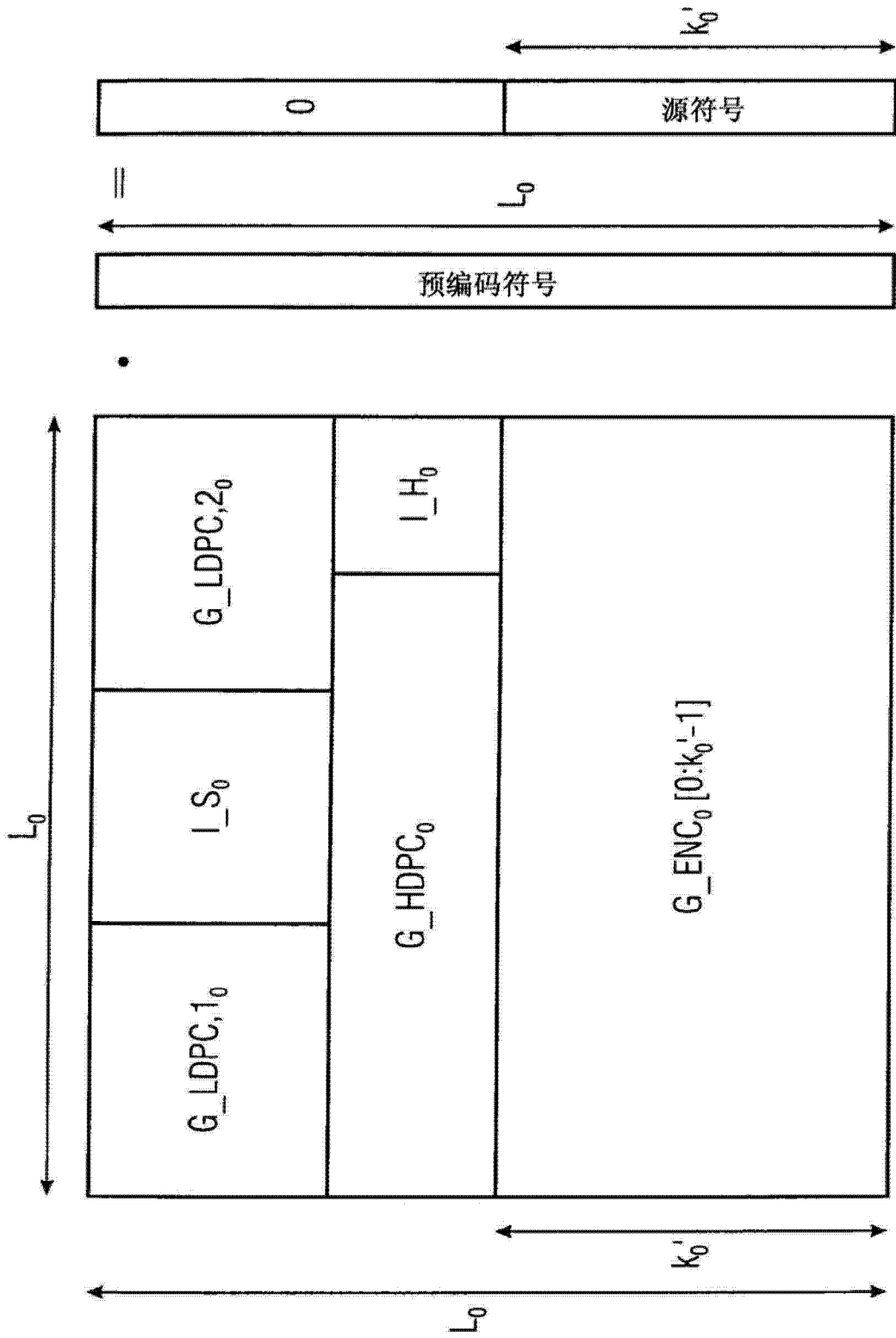


图 14

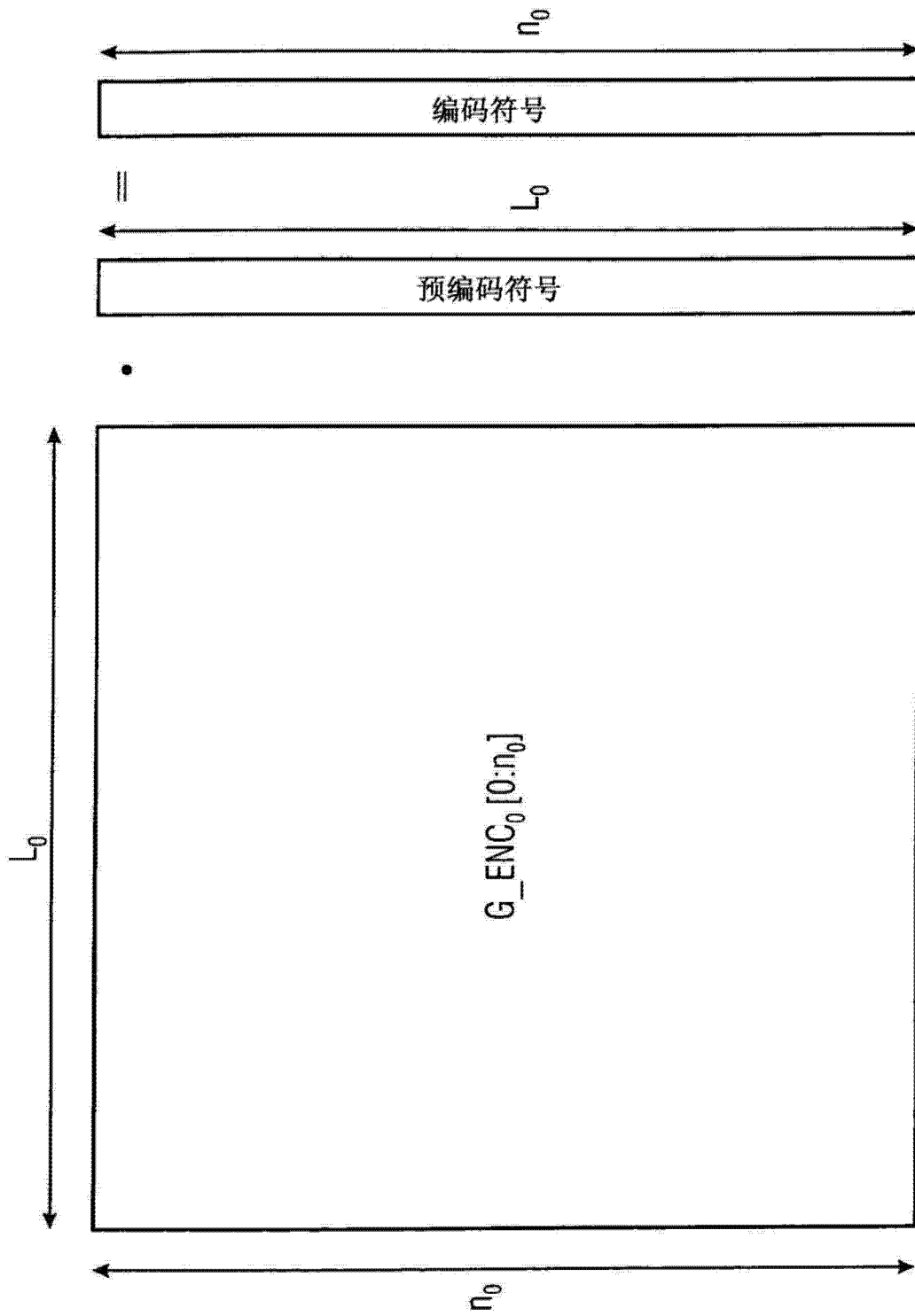


图 15

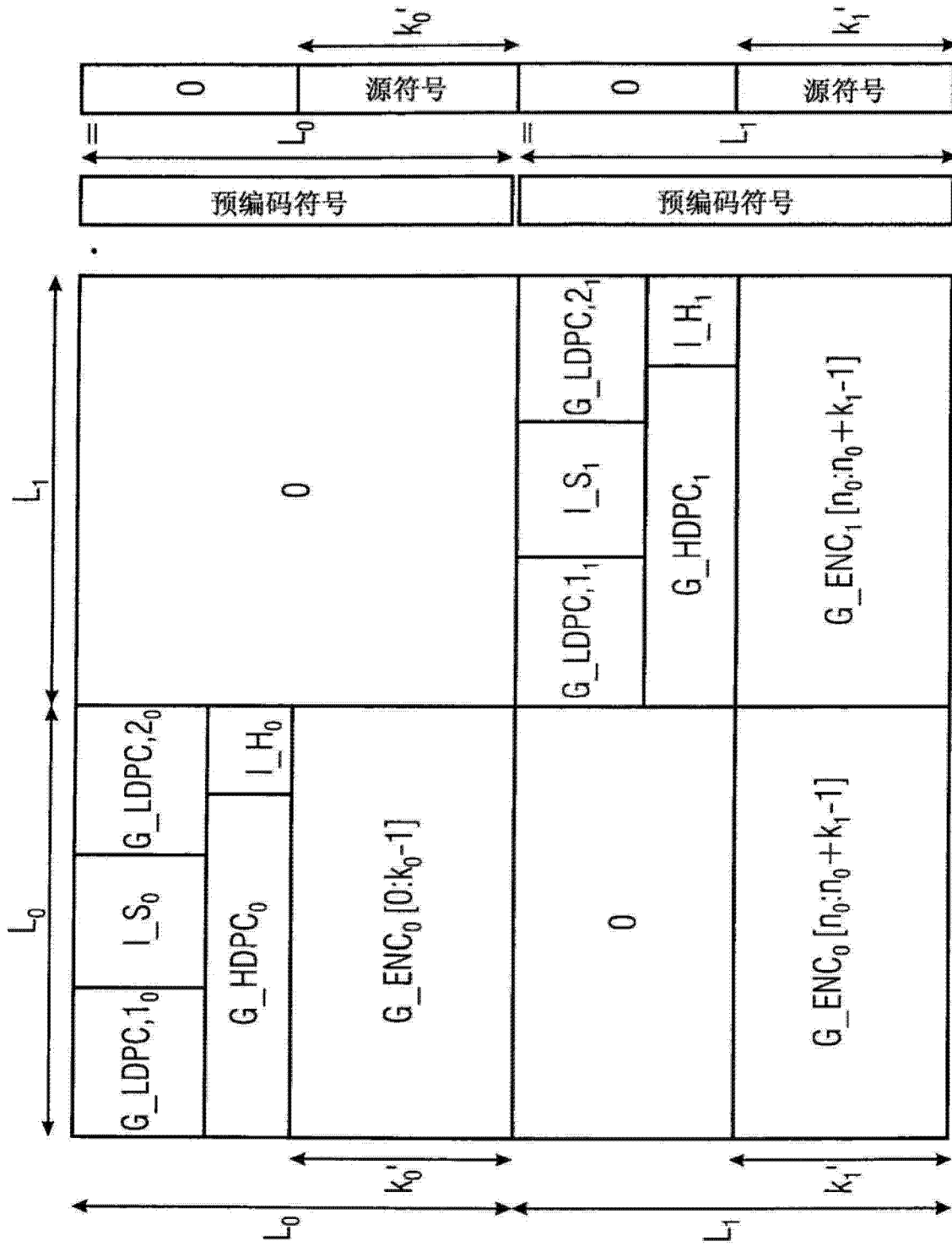


图 16

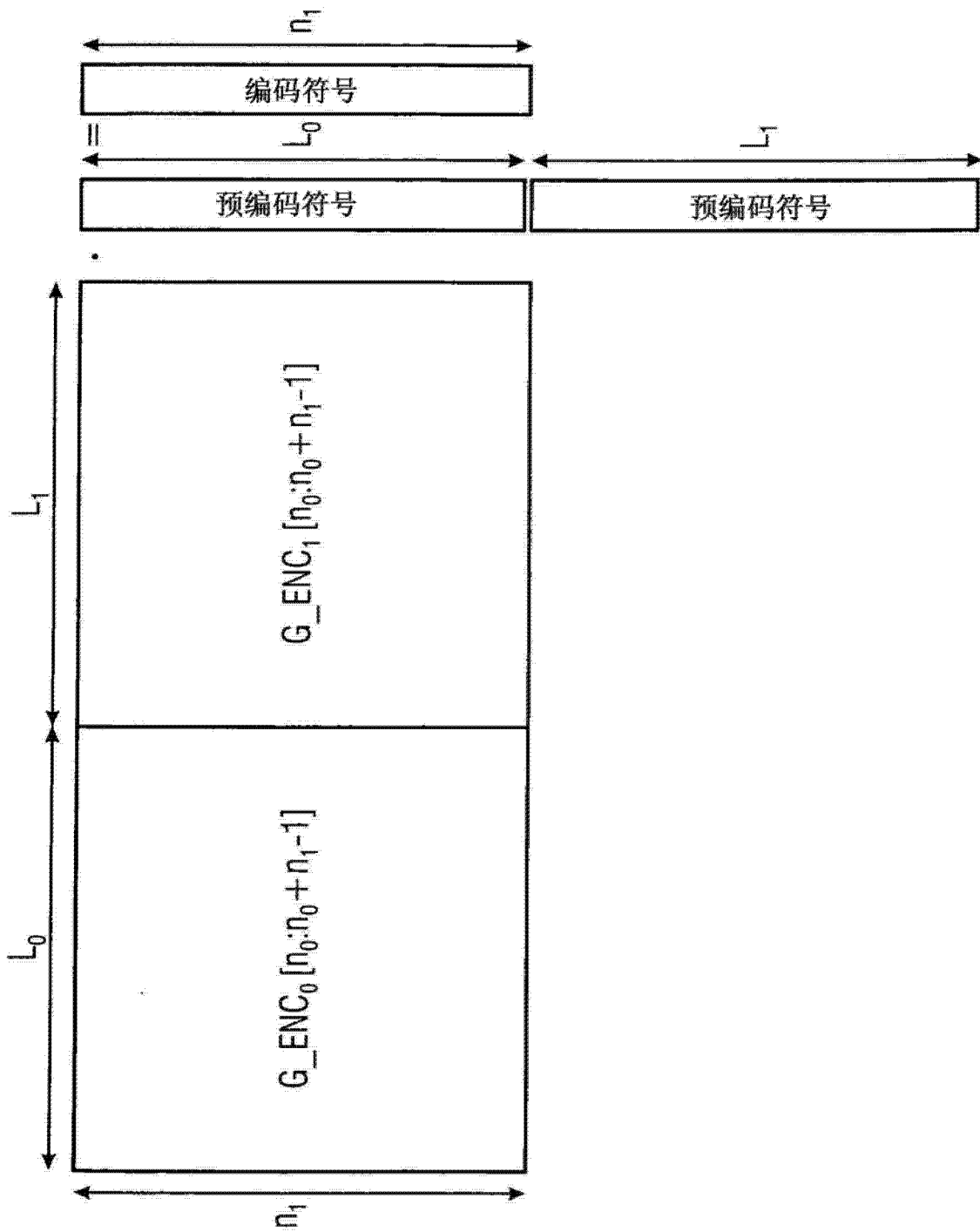


图 17

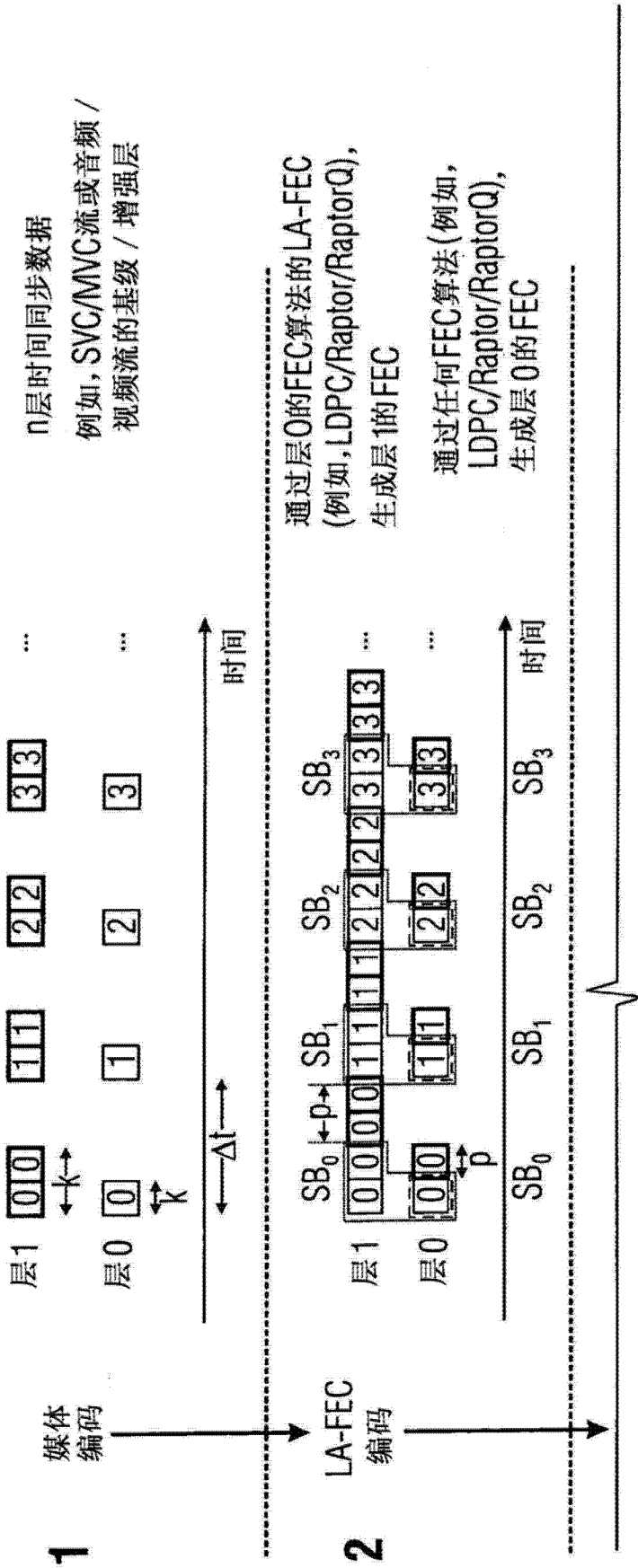


图 18	图 18A
	图 18B

图 18A

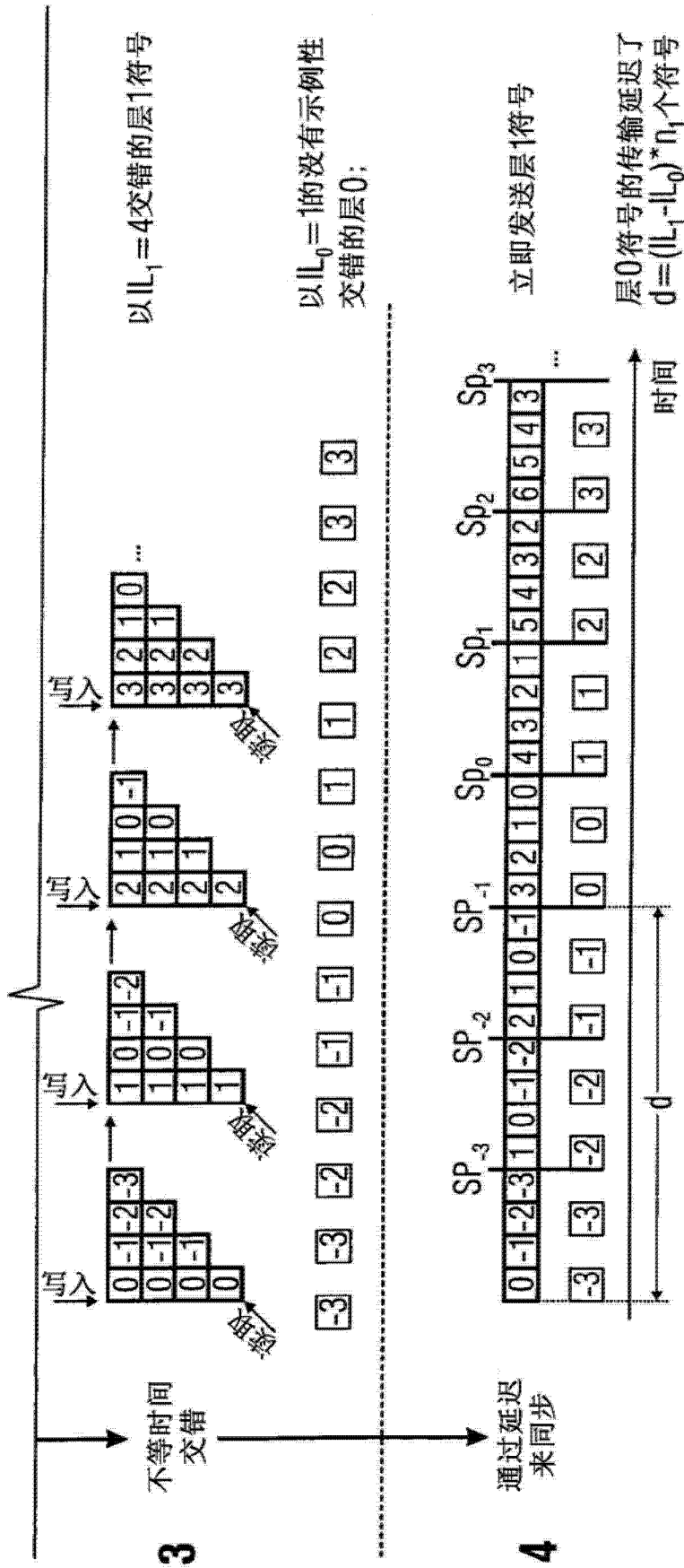


图 18	图 18A
	图 18B

图 18B

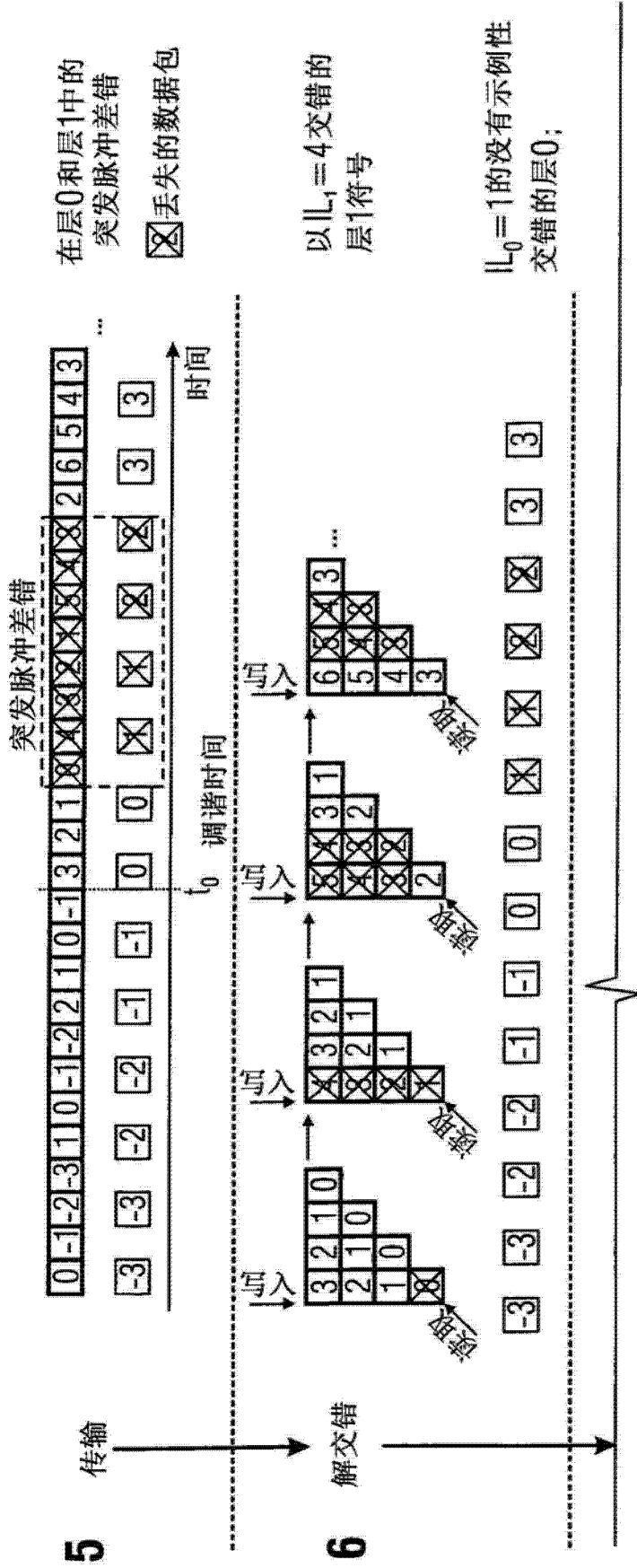


图 19	图 19A
	图 19B

图 19A

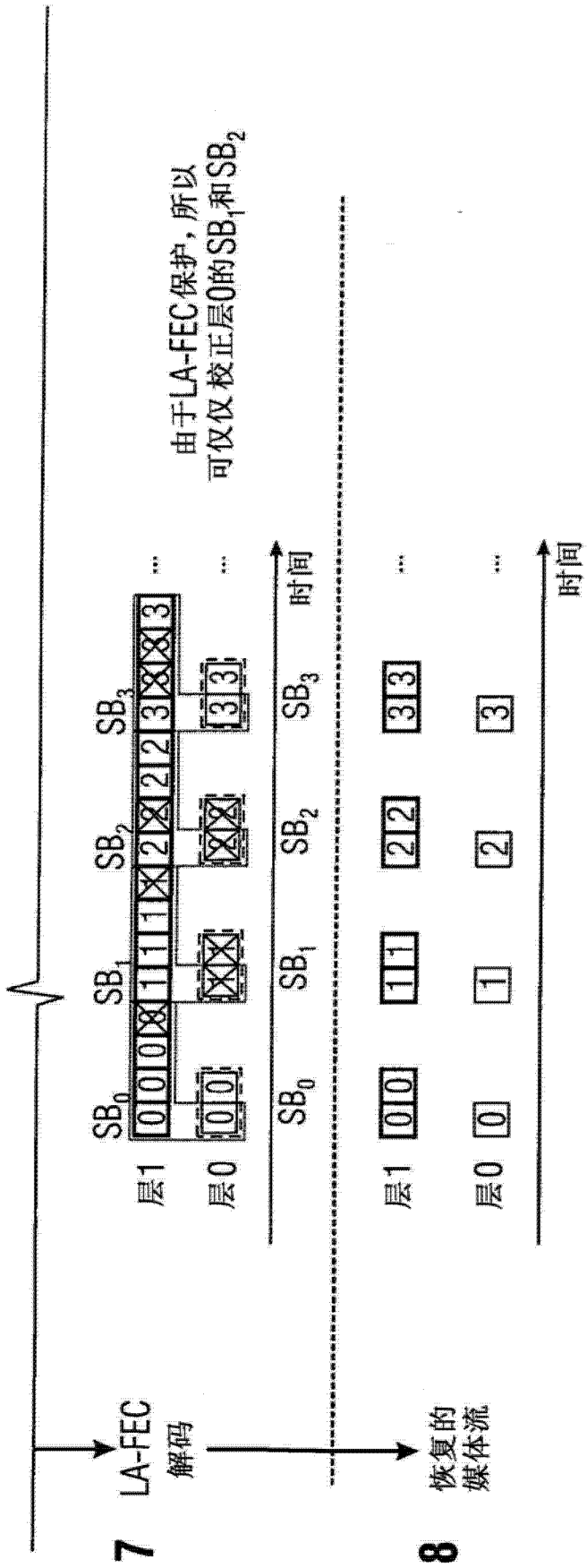
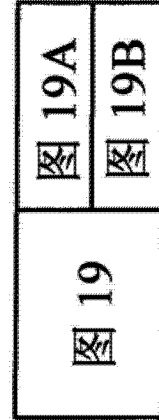


图 19B



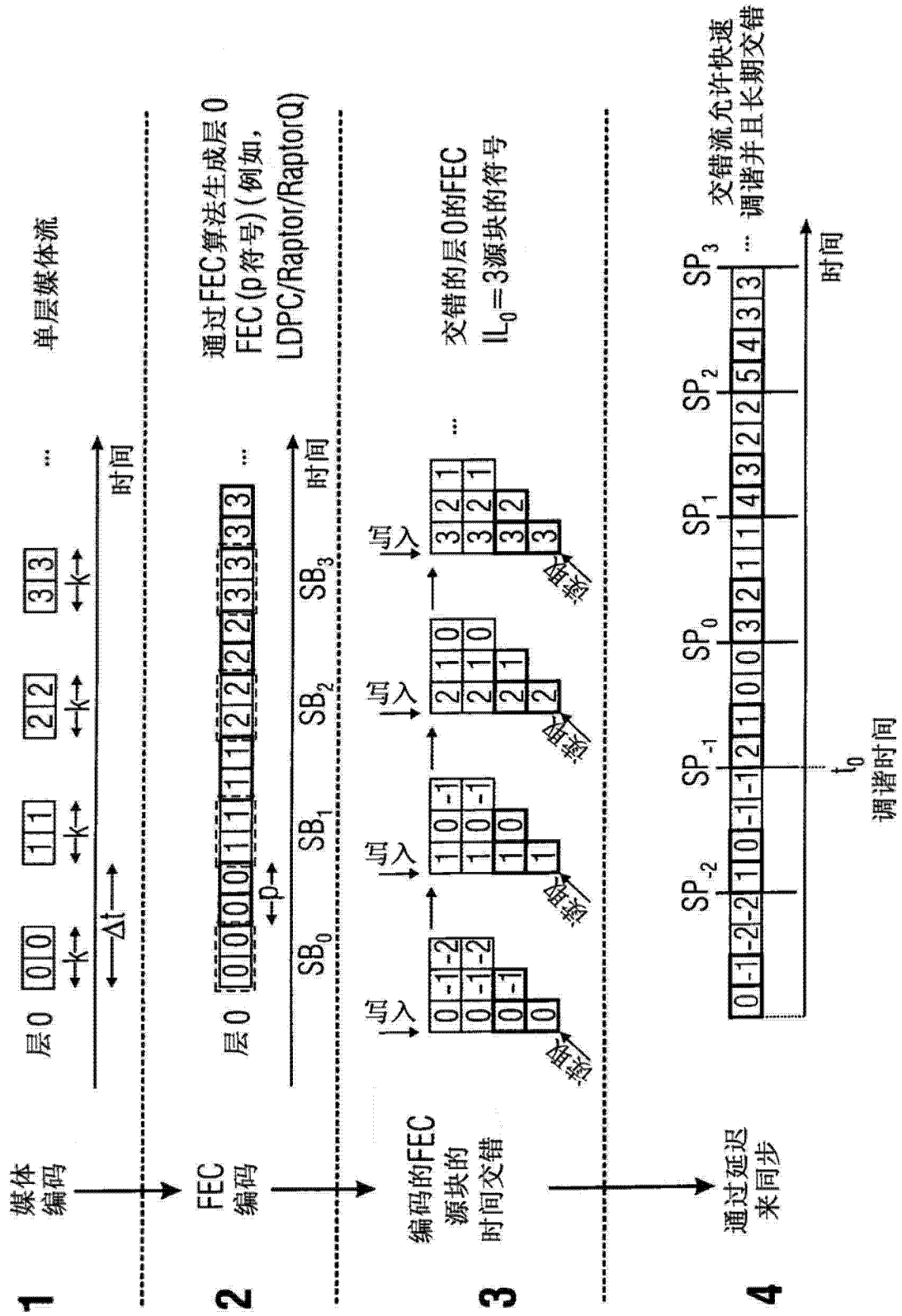


图 20

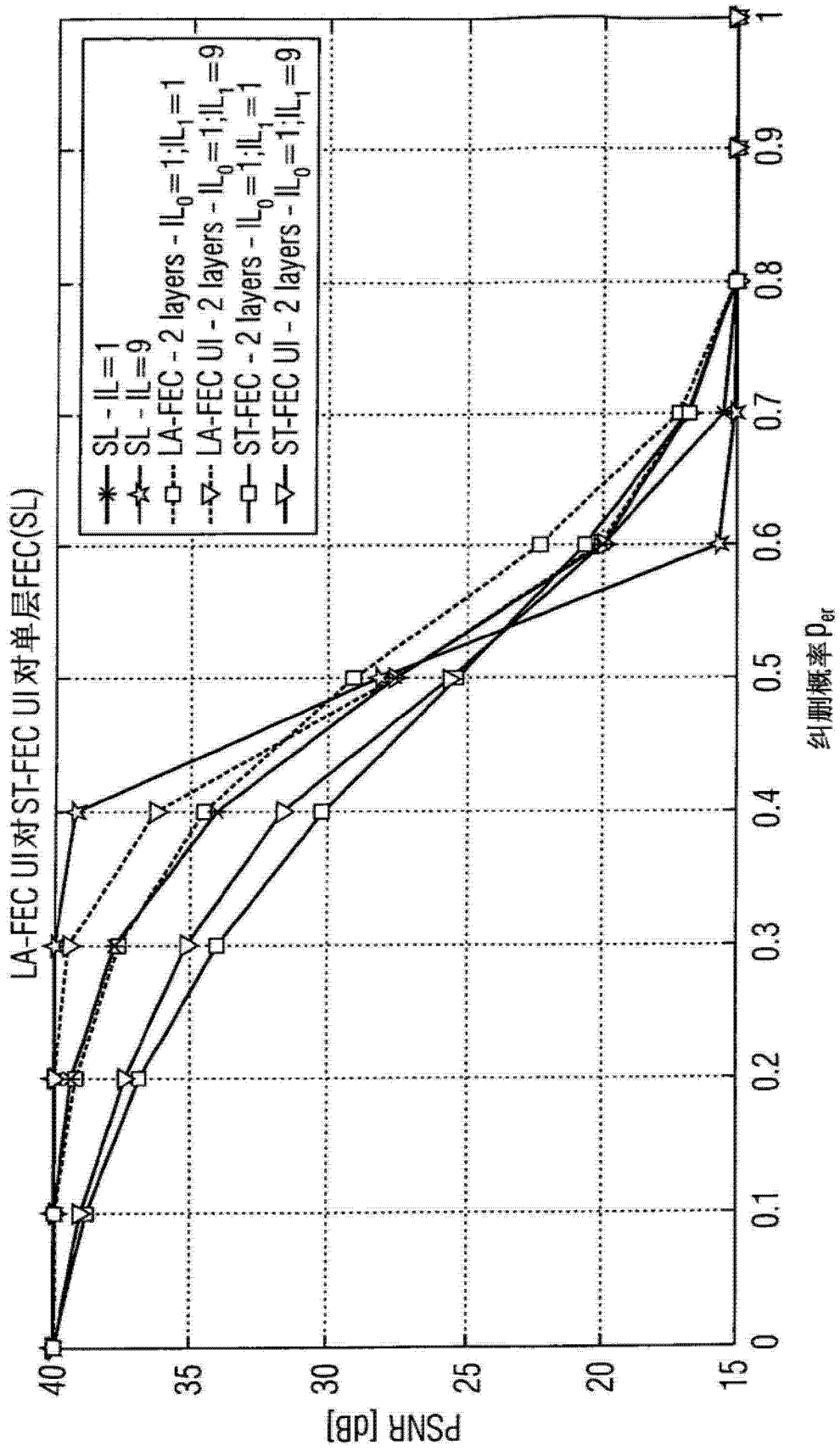


图 21

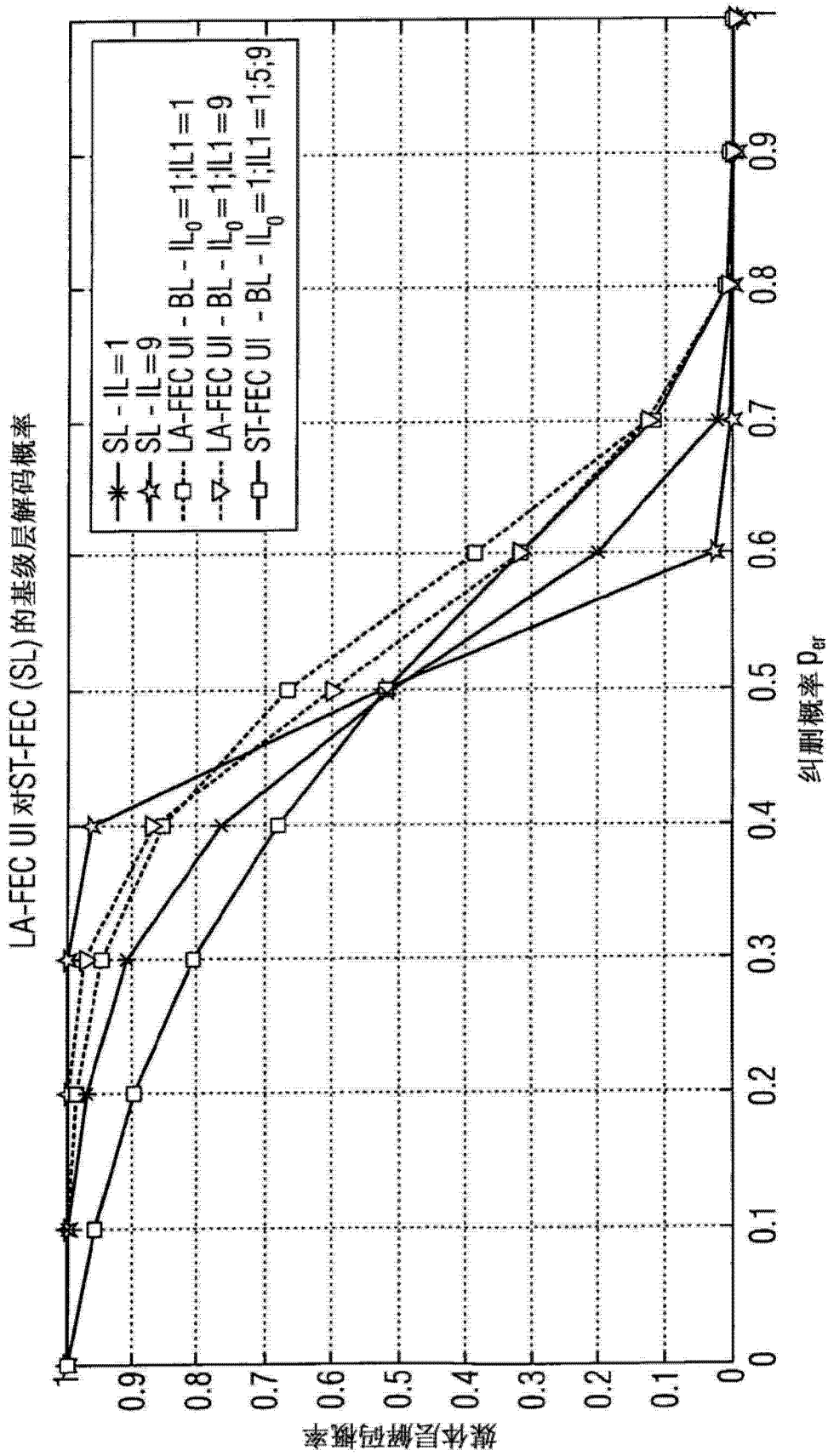


图 22A

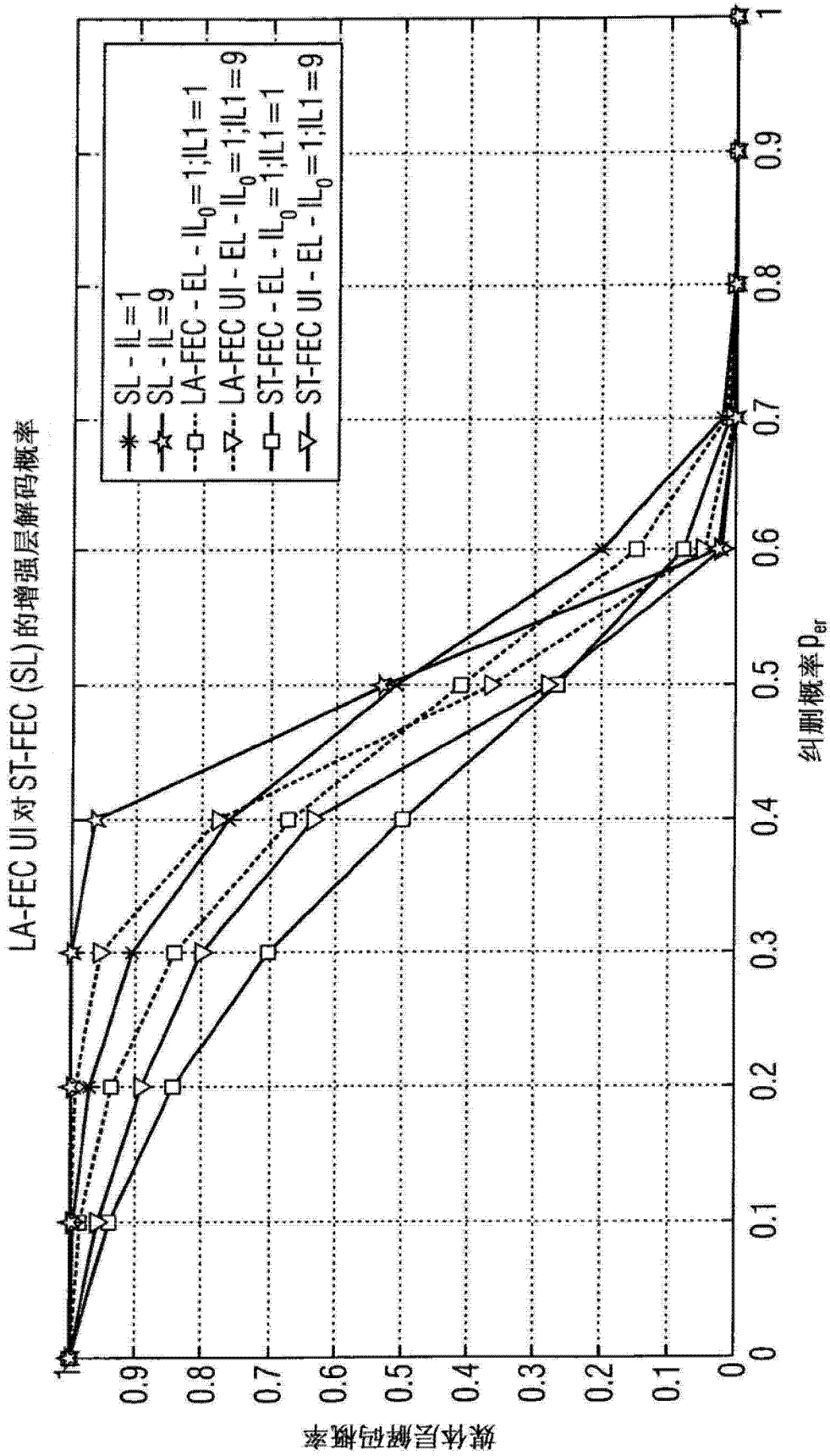


图 22B

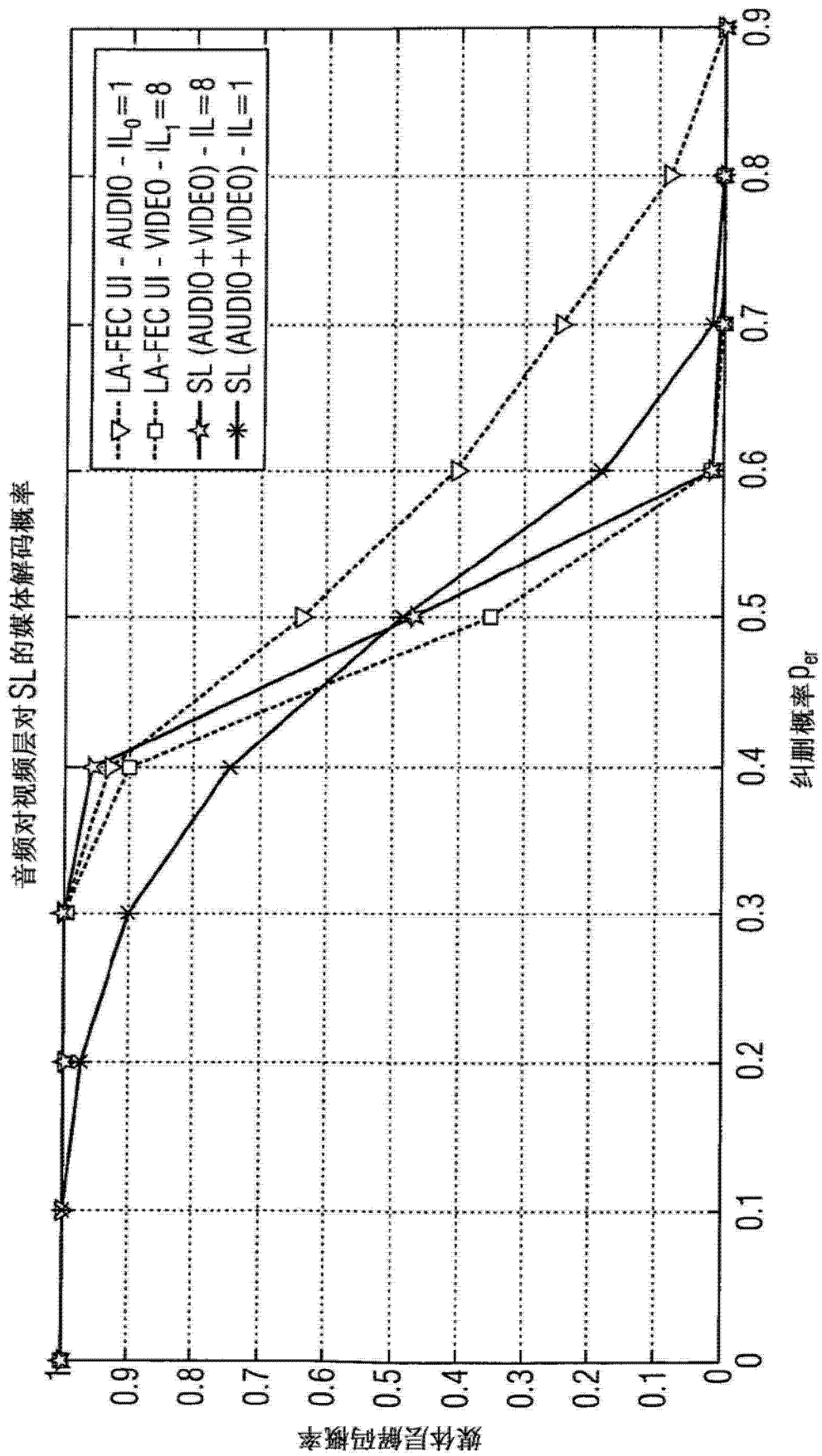


图 23

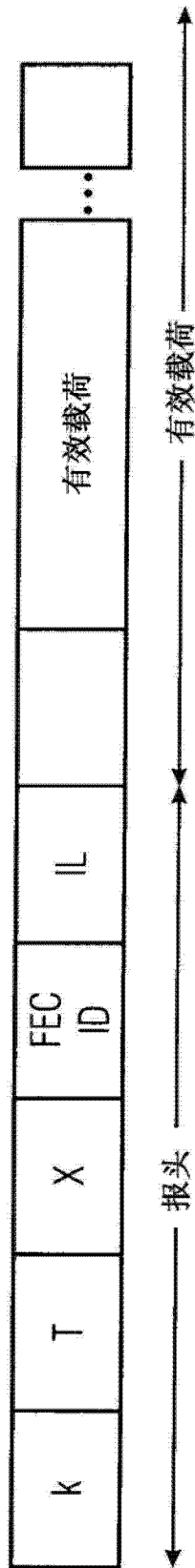


图 24