



# [12] 发明专利申请公开说明书

[21] 申请号 97190399.9

[43]公开日 1998年8月5日

[11] 公开号 CN 1189935A

[22]申请日 97.4.14

[30]优先权

[32]96.4.19 [33]US[31]08 / 636,732

[86]国际申请 PCT / US97 / 06129 97.4.14

[87]国际公布 WO97 / 40582 英 97.10.30

[85]进入国家阶段日期 97.12.19

[71]申请人 通用电气公司

地址 美国纽约州

[72]发明人 S·M·拉迪克

J·B·安德尔森

[74]专利代理机构 中国专利代理(香港)有限公司

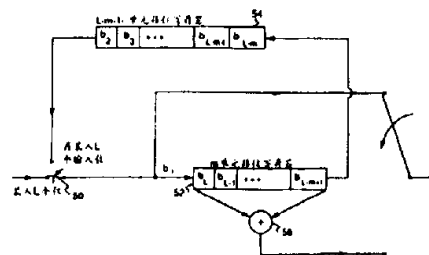
代理人 邹光新 傅康

权利要求书 7 页 说明书 16 页 附图页数 5 页

[54]发明名称 并行链接截尾卷积码及其译码器

[57]摘要

并行链接卷积编码方案采用截尾非递归系统卷积码。相关的译码器迭代地利用循环最大后验译码来产生硬及软判决输出。这种编码/译码系统改善了对于短信息的纠错性能。



## 权 利 要 求 书

1.进行并行链接卷积编码的方法,包括以下步骤:

把数据位块提供给包括并行链接的  $N$  个分量编码器和  $N - 1$  个交错器的并行链接编码器;

5 通过把截尾非递归系统卷积码施加给这些分量编码器中的第一个而在该第一个分量编码器内编码该数据位块,由此产生包括数据位和奇偶校验位的相应第一分量码字;

交错该数据位块来产生被置换数据位块;

10 通过把截尾非递归系统卷积码应用于后续的分量编码器而在该后续分量编码器内编码所得到的被置换数据位块,由此产生包括数据位和奇偶校验位的相应第二分量码字;

利用其余  $N - 2$  个交错器和其余  $N - 2$  个分量编码器重复交错和编码所得到的被置换数据位块的步骤,由此产生包括数据位和奇偶校验位的分量码字; 以及

15 格式化这些分量码字的位来产生一合成码字。

2.权利要求 1 的方法,其中格式化步骤的执行使得该合成码字只包括数据位块中的每一个位的一次出现。

3.权利要求 1 的方法,其中格式化步骤的执行使得该合成码字只包括按照预定模式组成分量码字的位中的被选定的一些位。

20 4.译码并行链接卷积码的方法,包括以下步骤:

从信道接收合成码字,该合成码字包括多个 ( $N$  个)分量码字的被格式化的位集合,该多个 ( $N$  个)分量码字是在并行链接编码器内通过把截尾非递归系统卷积码应用于数据位块而被产生的,根据该接收的合成码字产生接收的分量码字,每一个接收分量码字被一合成译码器的  $N$   
25 个分量译码器相应的一个接收,每一个分量译码器还接收数据位值的一组先验软判决信息;

利用  $N$  个分量译码器和  $N - 1$  个交错器通过迭代处理来译码接收的分量码字,产生合成译码器的软判决输出,这  $N$  个分量译码器的每一个按照相应分量编码器进行编码的顺序产生数据块中的每个数据位的  
30 软判决信息,这  $N - 1$  个交错器的每一个交错来自上一分量译码器的软判决信息,把被置换软信息块提供给后续分量译码器,对于第一次迭

代, 假定数据位的值是等概率来计算供  $N$  个分量译码器的第一个使用的一组先验软判决信息, 此后的一组先验软判决信息包括软判决信息的第一函数, 该软判决信息的第一函数从第  $N$  个分量译码器通过包括相应于  $N - 1$  个交错器的  $N - 1$  个解交错器的第一解交错器进行反馈, 该第一解交错器的  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺序进行设置, 提供给随后每一个分量译码器的一组先验软判决信息包括来自上一个分量译码器的软判决信息的第一函数; 以及

利用相应于  $N - 1$  个交错器的  $N - 1$  个解交错器在第二解交错器中进行解交错, 以便产生第  $N$  个分量译码器的软判决输出的第二函数作为合成译码器的软判决输出, 该第二解交错器的  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺序进行设置。

5. 权利要求 4 的方法, 其中利用分量译码器、交错器和解交错器进行的迭代的次数是一预定数值。

6. 权利要求 4 的方法, 在该方法中, 如果迭代次数小于最大次数, 则利用分量译码器、交错器和解交错器进行的迭代就继续进行到检测到译码器收敛为止; 否则在最大次数的迭代之后终止译码, 合成译码器利用第二解交错器提供第  $N$  个分量译码器的软判决输出的第二函数作为其软判决输出。

7. 权利要求 4 的方法, 还包括执行判决规则来产生作为合成译码器软判决输出的函数的硬判决输出。

8. 权利要求 4 的方法, 其中被格式化的位集合被按照预定模式进行收缩, 该译码方法还包括在产生接收的分量码字时插入中性值代替全部被收缩位的步骤。

9. 权利要求 4 的方法, 其中的译码步骤由包括循环 MAP 译码器的  $N$  个分量译码器来执行, 该译码步骤包括解本征矢量问题。

10. 权利要求 4 的方法, 其中的译码步骤由包括循环 MAP 译码器的  $N$  个分量译码器来执行, 该译码步骤包括递归方法。

11. 编码和译码并行链接卷积码的方法, 包括以下步骤:

把数据位块提供给包括并行链接的  $N$  个分量编码器和  $N - 1$  个交错器的并行链接编码器;

通过把截尾非递归系统卷积码施加给这些分量编码器中的第一个而在该第一个分量编码器内编码该数据位块, 由此产生包括数据位和奇

偶校验位的相应第一分量码字;

交错该数据位块来产生被置换数据位块;

通过把截尾非递归系统卷积码施加给一后续的分量编码器而在该后续分量编码器内编码所得到的被置换数据位块, 由此产生包括数据位和奇偶校验位的相应第二分量码字;

5 利用其余  $N - 2$  个交错器和其余  $N - 2$  个分量编码器重复交错和编码所得到的被置换数据位块的步骤, 由此产生包括数据位和奇偶校验位的分量码字;

格式化分量码字的位来产生合成码字;

10 从信道接收一合成码字;

根据该接收的合成码字产生接收的分量码字;

把每一个接收的分量码字提供给合成译码器的  $N$  个分量译码器相应的一个, 每一个分量译码器还接收数据位值的一组先验概率;

15 利用  $N$  个分量译码器和  $N - 1$  个交错器通过迭代处理来译码接收的分量码字, 产生合成译码器的软判决输出, 这  $N$  个分量译码器的每一个按照相应分量编码器进行编码的顺序产生数据块中的每个数据位的软判决信息, 这  $N - 1$  个交错器的每一个交错来自上一分量译码器的软判决信息, 把被置换软信息块提供给后续分量译码器, 对于第一次迭代, 假定数据位的值是等概率来计算供  $N$  个分量译码器的第一个使用的一组先验判决信息, 此后的迭代就包括了软判决信息的第一函数, 该软判决信息的第一函数从第  $N$  个分量译码器通过包括相应于  $N - 1$  个交错器的  $N - 1$  个解交错器的第一解交错器进行反馈, 该第一解交错器的  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺序进行设置, 提供给随后每一个分量译码器的一组先验软判决信息包括来自上一个分量译码器的软判决信息的第一函数; 以及

25 利用相应于  $N - 1$  个交错器的  $N - 1$  个解交错器在第二解交错器中进行解交错, 以便产生第  $N$  个分量译码器的软判决输出的第二函数作为合成译码器的软判决输出, 该第二解交错器的  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺序进行设置。

30 12. 权利要求 11 的方法, 其中格式化步骤的执行使得该合成码字只包括数据位块中的每一个位的一次出现。

13. 权利要求 11 的方法, 其中格式化步骤的执行使得该合成码字只

包括由符合预定模式的一些分量码字组成的位中的被选定的一些位。

14.权利要求 11 的方法，其中利用分量译码器、交错器和解交错器进行的迭代的次数是一预定数值。

15.权利要求 11 的方法，如果迭代次数小于最大次数，则利用分量译码器、交错器和解交错器进行的迭代继续进行到检测到译码器收敛为止；否则在最大次数的迭代之后终止译码，合成译码器利用第二解交错器提供第 N 个分量译码器的软判决输出的第二函数作为其软判决输出。

16.权利要求 11 的方法，还包括执行判决规则来产生作为合成译码器软判决输出的函数的硬判决输出。

17.权利要求 11 的方法，其中的译码步骤由包括循环 MAP 译码器的 N 个分量译码器来执行；该译码步骤包括解本征矢量问题。

18.权利要求 11 的方法，其中的译码步骤由包括循环 MAP 译码器的 N 个分量译码器来执行，该译码步骤包括递归方法。

19.权利要求 11 的方法，其中的格式化步骤还包括按照预定模式收缩组成该合成码字的分量码字中被选定的一些位，该译码方法还包括在产生接收的分量码字时插入中性值代替全部被收缩位的步骤。

20.并行链接编码器，包括：

并行链接的多个（N 个）分量编码器和多个（N 个）交错器，有次序地把截尾非递归系统卷积码应用于数据位块和对该数据位块进行各种置换，由此产生包括数据位和奇偶校验位的分量码字；以及

合成码字格式化器，格式化分量码字的位集合来产生合成码字。

21.权利要求 20 的编码器，其中的合成码字格式化器如此地产生合成码字，使得该合成码字只包括数据位块中的每一个位的一次出现。

22.权利要求 20 的编码器，其中的合成码字格式化器如此地产生合成码字，使得该合成码字只包括按照预定模式组成分量码字的位中的被选定的一些位。

23.译码并行链接卷积码的合成译码器，包括：

合成码字 - 分量码字变换器，从信道接收合成码字，该合成码字包括在并行链接编码器内通过把截尾非递归卷积码应用于数据位块而产生的 N 个分量码字的被选定的位，并根据该合成码字产生 N 个相应的接收分量码字；

多个 ( $N$  个) 分量译码器, 每一个译码器从该合成码字 - 分量码字变换器接收相应的接收分量码字, 每一个译码器还接收数据位值的一组先验软判决信息, 这  $N$  个分量译码器的每一个按照在并行链接编码器内的相应分量编码器进行编码的顺序产生数据块中的每个数据位的软判决信息;

多个 ( $N$  个) 交错器, 每一个交错器交错相应分量译码器的软判决信息, 把被置换软信息块提供给后续分量译码器, 接收码字被  $N$  个分量译码器和  $N - 1$  个交错器利用迭代处理来译码, 产生合成译码器的软判决输出;

包括相应于  $N - 1$  个交错器的  $N - 1$  个解交错器的第一解交错器, 该第一解交错器的这  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺序进行设置, 对于第一次迭代, 假定数据位的值是等概率来计算供  $N$  个分量译码器的第一个使用的一组先验软判决信息, 此后的一组先验软判决信息包括软判决信息的第一函数, 该软判决信息的第一函数由第  $N$  个译码器输出并通过该第一解交错器进行反馈, 提供给随后每一个分量译码器的一组先验软判决信息包括来自上一个分量译码器的软判决信息的第一函数; 以及

包括相应于  $N - 1$  个交错器的  $N - 1$  个解交错器的第二解交错器, 该第二解交错器的这  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺序进行设置, 该第二解交错器解交错第  $N$  个分量译码器的软判决输出的第二函数来产生合成译码器的软判决输出。

24. 权利要求 23 的译码器, 其中利用分量译码器、交错器和解交错器进行的迭代的次数是一预定数值。

25. 权利要求 23 的译码器, 在该译码器中, 如果迭代次数小于最大次数, 则利用分量译码器、交错器和解交错器进行的迭代就继续进行到检测到译码器收敛为止; 否则在最大次数的迭代之后终止译码, 合成译码器利用第二解交错器提供第  $N$  个分量译码器的软判决输出的第二函数作为其软判决输出。

26. 权利要求 23 的译码器, 还包括执行判决规则来产生作为合成译码器软判决输出的函数的硬判决输出的判定装置。

27. 权利要求 23 的译码器, 其中的  $N$  个分量译码器包括通过解本征矢量问题进行译码的循环 MAP 译码器。

28.权利要求 23 的译码器, 其中的  $N$  个分量译码器包括利用递归方法进行译码的循环 MAP 译码器。

29.编码和译码并行链接卷积码的编码器和译码器系统, 包括:

并行链接编码器, 包括并行链接的多个 ( $N$  个) 分量编码器和多个  
5 ( $N$  个) 交错器, 有次序地把截尾非递归系统卷积码应用于数据位块和对该数据位块进行各种置换, 由此产生包括数据位和奇偶校验位的分量码字;

合成码字格式化器, 格式化分量码字的位集合来产生合成码字;

合成码字 - 分量码字变换器, 从信道接收合成码字, 并根据该合成  
10 码字产生  $N$  个相应的接收分量码字;

多个 ( $N$  个) 分量译码器, 每一个译码器从该合成码字 - 分量码变  
换器接收相应的接收分量码字, 每一个译码器还接收数据位值的一组先  
验软判决信息, 这  $N$  个分量译码器的每一个按照在该并行链接编码器内  
15 的相应分量编码度进行编码的顺序产生数据块中的每个数据位的软判  
决信息;

多个 ( $N$  个) 交错器, 每一个交错器交错相应分量译码器的软判决  
信息, 把被置换软信息块提供给后续分量译码器, 接收码字被  $N$  个分量  
译码器和  $N - 1$  个交错器利用迭代处理来译码, 产生合成译码器的软判  
20 决输出;

包括相应于  $N - 1$  个交错器的  $N - 1$  个解交错器的第一解交错器,  
该第一解交错器的这  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺  
序进行设置, 对于第一次迭代, 假定数据位的值是等概率来计算供  $N$  个  
分量译码器的第一个使用的一组先验软判决信息, 此后的一组先验软判  
25 决信息包括软判决信息的第一函数, 该软判决信息的第一函数由第  $N$  个  
译码器输出并通过该第一解交错器进行反馈, 提供给随后每一个分量译  
码器的一组先验软判决信息包括来自上一个分量译码器的软判决信息的  
第一函数; 以及

包括相应于  $N - 1$  个交错器的  $N - 1$  个解交错器的第二解交错器,  
该第二解交错器的这  $N - 1$  个解交错器按照与  $N - 1$  个交错器相反的顺  
30 序进行设置, 该第二解交错器解交错第  $N$  个分量译码器的软判决输出的  
第二函数来产生合成译码器的软判决输出。

30.权利要求 29 的编码器和译码器系统, 其中的合成码字格式化器

如此地产生合成码字，使得该合成码字只包括数据位块中的每一个位的一次出现。

5 31.权利要求 29 的编码器和译码器系统，其中的合成码字如此地产生合成码字，使得该合成码字只包括按照预定模式组成分量码字的位中的被选定的一些位。

32.权利要求 29 的编码器和译码器系统，其中利用分量译码器、交错器和解交错器进行的迭代的次数是一预定数值。

10 33.权利要求 29 的编码器和译码器系统，在该系统中，如果迭代次数小于最大次数，则利用分量译码器、交错器和解交错器进行的迭代就继续进行到检测到译码器收敛为止；否则在最大次数的迭代之后终止译码，合成译码器利用第二解交错器提供第 N 个分量译码器的软判决输出的第二函数作为其软判决输出。

34.权利要求 29 的编码器和译码器系统，还包括执行判决规则来产生作为合成译码器软判输出的函数的硬判决输出的判定装置。

15 35.权利要求 29 的编码器和译码器系统，其中的 N 个分量译码器包括通过解本征矢量问题进行译码的循环 MAP 译码器。

36.权利要求 29 的编码器和译码器系统，其中的 N 个分量译码器包括利用递归方法进行译码的循环 MAP 译码器。



# 说明书

## 并行链接截尾卷积码及其译码器

5 本发明总的来说涉及在不良信道中传送短信息的纠错编码，尤其涉及并行链接编码技术及其译码器。

由于被称为并行链接卷积编码 (PCCC) 或“增强 (turbo) 编码”的一种并行链接编码在应用于 10,000 或 10,000 以上个位的块时所表现出来的给人深刻印象的编码增益，它已成为近来编码研究的主题。(见 C. Bernu、A. Glavieux 和 P. Thitimajshima 发表在 Proceedings of IEEE International Conference on Communications (1993 年, 1064 - 1070 10 页) 上的论文“接近仙农极限的纠错编码和译码: Turbo 码”; J. D. Anderson 发表于丹麦技术大学的电信研究所的 Report IT-146 ISSN 0105-854 上的论文“Turbo 编码方案”; 以及 P. Robertson 发表在 1994 IEEE Globecom Conference 第 1298 - 1303 页上的论文“阐明代码结构和并行链接递归系统 (Turbo) 码的译码器”。) 15

但是，已发现 turbo 码的性能随着编码数据块长度的减小而显著地降低。这是由于其分量递归系统卷积码的权重结构对于块长度的强依赖性造成的。第二个问题是施加给 turbo 编码器的信息块的正确终止。正如 O. Joeresson 和 H. Meyr 在“终止 turbo 码的格构” (IEE Electronics 20 Letters, 30 卷, 16 期, 1994 年 8 月 14 日, 1285 - 1286 页) 中所描述的那样, turbo 编码器中采用的交错会造成无法用一组尾位来终止交错和非交错编码器输入序列的后果。虽然可以用嵌入到信息结构中去的第二尾序列来正确地终止对交错数据序列进行操作的编码器, 但这样做加倍了与编码器终止有关的额外开销, 减小了有效码率。另一种做法是不终止编码器序列之一, 但这样做会降低编码器/译码器系统的性能, 尤 25 其在作用于短信息时更是如此。在“终止处于同一状态的 turbo 码的格构” (JEE Electronics Letters, 31 卷, 1 期, 1995 年 1 月 5 日, 22 - 23 页) 中, A. S. Barbulescu 和 S. S. Pietrobon 描述了对交错器的设计施加约束以使用一个终止位序列来终止两个分量递归系统卷积 (RSC) 编码器的方法。它们的性能结果与在使用最佳交错器时通过 30 终止两个编码器而获得的性能相比有一些下降。此外, 公开的相对于每位能量与噪声功率谱密度之比 ( $E_b/N_0$ ) 的位差错率 (BER) 显示了

在 turbo 编码器中采用 RSC 时在  $E_b/N_0$  值的一定范围内 BER 的平坦性。

因此，需要提供用于短信息块的改进的并行链接编码技术。

根据本发明，并行链接卷积编码方案采用截尾非递归系统卷积 (NSC) 码。相关的译码器反复利用循环最大后验 (MAP) 译码来产生硬和软判决输出。截尾码的应用解决了终止 turbo 编码中的输入数据序列的问题，由此避免了对于短信息相关译码器性能的降低。虽然随着数据块长度的逐渐增大，NSC 码一般来说比相同记忆的递归系统卷积 (RSC) 码弱，但 NSC 码的自由距离对数据块长度不那么敏感。因此，对于比一定的门限数据块长度短的信息而言，利用 NSC 码的并行链接编码将优于相同记忆的 RSC 码的并行链接编码。

参看以下结合附图对本发明的详细描述将明了本发明的特点和优点，附图中：

图 1 是表示并行链接编码器的简化方框图；

图 2 是表示并行链接码的译码器的简化方框图；

图 3 是表示供本发明的编码方案使用的截尾非递归系统卷积编码器的简化方框图；

图 4 是表示本发明的可用作并行链接卷积编码方案的译码器中的分量译码器的循环 MAP 译码器的简化方框图；

图 5 是表示本发明的可用作并行链接卷积编码方案的译码器中的分量译码器的循环 MAP 译码器的一替代实施例的简化方框图。

图 1 是并行链接编码方案的编码器信号处理 10 的简略方框图。它包括对来自数据源的数据位块进行操作的  $N$  个分量编码器 12。这些数据块被交错器 14 利用交错算法进行置换。最后，分量编码器的输出被合成码字格式化器 16 组合成为单个合成码字。选择该合成码字格式化器来适应信道特性，其后可跟有被选择来适应信道和通信系统的信道接入技术的帧格式化器。该帧格式化器还可以插入例如控制位和同步符号这样的其它必要的额外开销。

如果分量码是系统码，就能够在并行链接编码中获得有效码率的优点。系统编码器产生的码字 (输出) 包括被作为该编码器的输入而提供的原始数据位和附加的奇偶校验位。(这些奇偶校验位引入的冗余增强代码的纠错能力。) 因此，当在图 1 所示并行链接编码器中使用系统编

码器时，全部分量编码器 12 产生的码字都包含输入数据位。如果格式化器 16 产生只包含每一分量编码器 12 产生的奇偶校验位和待编码的信息位块的数据包或合成码字，则通过消除被发送合成码字中的信息位的重复就实现了合成并行链接码码率的显著改善。例如，如果包括两个分量码的并行链接卷积码 (PCCC) 编码器的分量编码器 1 和分量编码器 2 都是比率 1/2 码，则合成并行链接码码率将从非系统分量码的 1/4 提高到使用系统分量码时的 1/3。

采用递归系统卷积 (RSC) 码的并行链接编码方案已成为许多研究近来的课题。这些并行链接卷积码 (PCCC) 还在文献中被统称为 “turbo” 码。如上所述，已发现这些 PCCC 在信息相对较大、即一万或一万以上个位的情况下，能够在相对于每位能量与噪声功率谱密度之比 ( $E_b/N_0$ ) 的位差错率 (BER) 方面获得给人深刻印象的性能。但是，还已发现因为递归系统卷积分量码的强度对数据块长度相当敏感，所以利用 turbo 码获得的编码增益随着数据块尺寸的减小而显著地降低。相反地，对于大多数实际应用来说，非递归系统截尾卷积码的性能与数据块长度无关；仅当编码的数据位块小于由 NSC 的判决深度特性确定的最小尺寸时，可获得的性能才降低。

图 2 以方框图的形式表示并行链接码的一普通译码器 20。译码器 20 包括：合成码字 - 分量码字变换器 22，把从信道接收的合成码字变换成为每个分量译码器 24 的单个接收码字；相应于图 1 的 N 个分量编码器的 N 个分量译码器 24；与在并行链接编码器 (图 1) 中使用的交错器同类型 (或相同) 的交错器 14；以及第一和第二解交错器 28 和 29，它们各具有等同于与编码所用的 N - 1 个交错器相应的串联的 N - 1 个解交错器 30 的序列重排特性。这些解交错器所需的排序如图 2 所示，与交错器的排序相反。分量译码器 24 的输出是关于接收码字中的每一数据位的估算值的某种软判决信息。例如，分量译码器的输出可以是译码位是 0 或 1 的概率的第一函数，译码位是 0 或 1 的概率受来自信道的接收符号序列的制约。这种第一函数的一个例子从分量译码器的软判决输出中消除条件概率  $P\{d_t^j = 0/Y_t^j\}$  的影响，该软判决输出在被恰当置换之后被输入给下一个顺序的分量译码器，这里的  $P\{d_t^j = 0/Y_t^j\}$  是第 j 个信息位在时刻 t 等于 0 的概率，该概率受接收的信道输出符号  $Y_t$  的第 j 个 (系统) 位的制约。或者，分量译码器 24 输出的软判决信息可以是似然比

的函数:

$$\Lambda(d_i^j) = \frac{P(d_i^j = 1 | Y_i^j)}{P(d_i^j = 0 | Y_i^j)} = \frac{1 \cdot P(d_i^j = 0 | Y_i^j)}{P(d_i^j = 0 | Y_i^j)}$$

5

或作为  $\log$  似然比的函数  $\log[\Lambda(d_i^j)]$ 。如图所示, 第  $N$  个分量译码器具有第二输出, 即译码位值或上述似然比的条件概率的第二函数。这种第二函数的一个例子是  $P\{d_i^j = 0 | Y_i^j\}$  和接收自第一分量译码器的先验概率  $d_i^j = 0$  的乘积。

并行链接码的译码器按照以下方式迭代地进行操作。第一分量译码器(译码器 1)根据接收码字和关于被发送信息位的任何先验信息计算被第一分量编码器编码的信息位序列的一组软判决值。在第一次迭代中, 如果没有关于源统计资料的先验信息, 就假定位等于 0 或 1 的概率是相等的(即  $P\{\text{位} = 0\} = P\{\text{位} = 1\} = 1/2$ )。译码器 1 计算的软判决值然后被利用与在编码器中用来为第二编码器置换数据位块的交错器同类型(或相同)的交错器进行交错。这些被置换判决值和相应的接收码字包括了至下一个分量译码器(译码器 2)的输入。从上一分量译码器和交错器接收的被置换软判决值被下一分量译码器用作关于待译码数据位的先验信息。分量译码器如此地顺序操作, 直到第  $N$  个译码器计算了被编码器编码的数据位块的一组软判决输出为止。接下来的步骤是如上所述地解交错该第  $N$  个译码器的软判决值。第一译码器然后把该第  $N$  个译码器的新软判决值作为接收码字的先验信息, 再对该接收码字进行译码。译码器如此地进行所希望次数的迭代操作。在最后迭代结束时, 是第  $N$  个译码器计算的软判决输出的第二函数的值序列被解交错, 以便使数据回复到被 PCCC 编码器编码时的顺序。迭代次数可以是一预定数值, 或者可以通过检测译码器的收敛动态地被确定。

译码器提供软判决信息, 它是概率  $P\{d_i^j = 0 | Y_i^j\}$  的函数, 该概率  $P\{d_i^j = 0 | Y_i^j\}$  就是在接收到信道输出集合  $Y_i^j = \{y_1, \dots, y_L\}$  的情况下在时刻  $t$  输入给编码器的  $k$  个位的符号中的第  $j$  个数据位是 0 的条件概率。此外, 译码器还可利用判定装置来提供作为其软判决输出的函数的硬判决信息, 该判定装置执行例如这样的判定规则:

30

$$\begin{aligned}
 & \hat{d}_i^j = 0 \\
 P\{d_i^j = 0 / Y_1^L\} & > \frac{1}{2}; \\
 & \hat{d}_i^j = 1
 \end{aligned}$$

5

就是说，如果  $P\{d_i^j = 0 / Y_1^L\} > \frac{1}{2}$ ，则  $\hat{d}_i^j = 0$ ；如果  $P\{d_i^j = 0 / Y_1^L\} < \frac{1}{2}$ ，则  $\hat{d}_i^j = 1$ ；

否则就随机地给  $d_i^j$  分配值 0 或 1。

一般的 turbo 译码器使用例如 L. R. Bahl、J. Cocke、F. Jelinek 和 J. Raviv 在“使符号差错率最小的线性码的最佳译码”（IEEE Transactions of Information Theory, 1974 年 3 月, 284 ~ 287 页）中描述的后验（MAP）译码器，或者使用如 J. Hagenauer 和 P. Hoehner 在“带有软判决输出的维特比算法及其应用”（1989 IEEE Globecom Conference, 1680 ~ 1686 页）中所描述的软输出维特比算法（SOVA）译码器。MAP 译码器产生译码位值是 0 或 1 的概率。相反地，SOVA 通常计算每一译码位的似然比：

15

$$\frac{P\{\text{译码位是1}\}}{P\{\text{译码位是0}\}}$$

显然，可根据  $P\{\text{译码位是 0}\}$  和反之亦然地利用  $P\{\text{译码位是 0}\} = 1 - P\{\text{译码位是 1}\}$  来获得这一似然比。已发现 MAP 或 SOVA 译码器以似然比的对数、即

20

$$\log \left( \frac{P\{\text{译码位是1}\}}{P\{\text{译码位是0}\}} \right)$$

进行工作可得到某些计算方面的优点。

已证明利用 turbo 码获得的编码增益（纠错能力）随着数据块尺寸的减小而降低。有几位作者已把这种编码增益的降低主要归因于 RSC 码的性质。已发现 RSC 码的距离性质随数据块长度的增大而增大。相反地，RSC 码的最小距离随数据块长度的减小而减小。第二个问题是因交错造成的终止包括 turbo 编码方案的全部 RSC 码方面的困难。不利

25

的是，无序列终止或对交错器设计所加上的约束造成的不利效果是显著的，并随着数据块长度的减小而越发突出。

根据本发明，并行链接卷积编码方案中的分量码包括截尾非递归系统卷积码。这种截尾码的使用解决了在进行 turbo 编码时终止输入数据序列的问题，由此避免了对于短信息译码器性能的降低。虽然 NSC 码一般来说比相同记忆的 RSC 码弱，但 NSC 码的自由距离对数据块长度不那么敏感。因此，对于比预定门限数据块长度短的信息而言，利用 NSC 码的并行链接编码将优于相同记忆的利用 RSC 码的并行链接编码。性能交叉点是所希望的位差错率、码率和码记忆的函数。

图 3 表示在本发明的并行链接卷积编码 (PCCC) 方案中使用的比率 =  $1/2$ 、记忆 =  $m$  的截尾非递归系统卷积编码器的一个例子。为描述起见， $(n, k, m)$  编码器表示这样一种编码器，其中的输入符号包括  $k$  个位，输出符号包括  $n$  个位， $m = k$  个位符号中的编码器记忆。为说明起见，在图 3 中示出二进制输入符号，即  $k = 1$ 。但是，本发明适用于  $k$ 、 $n$  和  $m$  的任何值。

一开始，开关 50 处于向下位置， $L$  个输入位每次  $k$  个（本实例每次一个输入符号）地被移入移位寄存器 52。在第  $L$  个位被装入编码器之后，开关移到向上位置，编码开始，第二移位寄存器 54 的第一个位被移入该非递归系统编码器，该编码器此时的状态是  $\{b_L, b_{L-1}, \dots, b_{L-(km-1)}\}$ 。在本实例中，编码器的输出包括作为该编码器状态和当前输入符号的函数而在方框 56（本实例中为一模 2 加法器）中被形成的输入位和奇偶校验位。当第  $L$  个位被编码时，编码就结束。

在本发明的另一个方面中，上述并行链接编码器的相关译码器包括由本发明人在共同转让的待审查美国专利申请第 (RD - 24,923) 号中描述的循环 MAP 译码器，该美国专利申请援引于此作参考。美国专利申请第 (RD - 24,923) 号特别描述了可用来译码截尾卷积码的循环 MAP 译码器。该循环 MAP 译码器能够把编码数据块的估算值和可信度信息作为在重发请求判断中使用的码组差错率的量度传送给数据接收器，例如在传输差错隐蔽中使用的语音合成信号处理器或用于分组数据的协议处理器。

特别是，如在美国专利申请第 (RD - 24923) 号中所描述的，采用了截尾的纠错格子码的循环 MAP 译码器产生软判决输出。该循环

MAP 译码器提供格构第一级的状态的概率的估算, 这些概率代替普通 MAP 译码器的开始状态的先验知识. 该循环 MAP 译码器以两种方式的任一种提供初始状态概率分布. 第一种方式涉及到本征值问题的解, 对于该本征值问题, 得到的本征矢量就是所需的初始状态概率分布; 利用该开始状态的知识, 该循环 MAP 译码器按照普通 MAP 译码算法执行剩余的译码. 第二种方式以递归为基础, 对于该递归, 迭代收敛于开始状态分布. 在足够多次迭代后, 就以大的概率得知圆形状态序列上的一状态, 循环 MAP 译码器按照普通 MAP 译码算法执行剩余的译码.

普通 MAP 译码算法的目的是确定条件概率:

$$10 \quad P\{\text{时刻 } t \text{ 的状态 } m / \text{接收信道输出 } y_1, \dots, y_L\}$$

公式中的项  $L$  表示以编码器符号的数目为单位的一组数据的长度.

( $(n, k)$  码的编码器对  $k$  个位的输入符号进行操作, 产生  $n$  个位的输出符号.) 项  $y_t$  是时刻  $t$  的信道输出 (符号).

MAP 译码算法实际上首先确定概率:

15

$$\lambda_t(m) = P\{S_t = m; Y_t^L\}; \quad (1)$$

即时刻  $t$  的编码器状态  $S_t$  是  $m$  和接收了信道输出集合  $Y_t^L = \{y_1, \dots, y_L\}$  的联合概率. 这些就是被乘以常数 ( $P\{Y_t^L\}$ , 接收信道输出集合  $\{y_1, \dots, y_L\}$  的概率) 的所需概率.

20 现在用下式定义矩阵  $\Gamma_t$  的元素:

$\Gamma_t(i, j) = P\{\text{时刻 } t \text{ 的状态 } j; y_t / \text{时刻 } t-1 \text{ 的状态 } i\}$  把矩阵  $\Gamma_t$  作为信道转换概率  $R(Y_t, X)$ 、编码器在时刻  $t$  从状态  $m'$  转换到状态  $m$  的概率  $P_t(m/m')$  和在假定先前编码器状态  $i$  是  $m'$  和当前编码器状态是  $m$  的情况下编码器的输出符号是  $X$  的概率  $q_t(X/m', m)$  的函数来进行计算. 特别是对全部可能的编码器输出  $X$  进行以下求和来计算  $\Gamma_t$  的每一元素:

$$25 \quad \gamma_t(m', m) = \sum_X p_t(m/m') q_t(X/m', m) R(Y_t, X). \quad (2)$$

30 MAP 译码器计算这些矩阵的  $L$ , 每一格构级一个矩阵. 它们由接收的信道输出符号和给定代码的格构分支的性质组成.

行矢量  $\alpha_t$  的  $M$  个联合概率元素定义如下:

$$\alpha_t(j) = P(\text{state } j \text{ at time } t; y_1, \dots, y_t) \quad (3)$$

列矢量  $\beta_t$  的  $M$  个条件概率元素定义如下:

$$\beta_t(j) = P\{y_{t+1}, \dots, y_L / \text{时刻 } t \text{ 的状态 } j\} \quad (4)$$

5  $j = 0, 1, \dots, (M - 1)$ ,  $M$  是编码器状态的数目。(注意在此用黑体字来表示矩阵和矢量。)

MAP 译码算法的步骤如下:

(i) 利用正向递归计算  $\alpha_1, \dots, \alpha_L$ :

$$10 \quad \alpha_t = \alpha_{t-1} \Gamma_t, \quad t=1, \dots, L \quad (5)$$

(ii) 利用反向递归计算  $\beta_1, \dots, \beta_{L-1}$ :

$$\beta_t = \Gamma_{t+1} \beta_{t+1}, \quad t=L-1, \dots, 1 \quad (6)$$

(iii) 用下式计算  $\lambda_t$  的分量:

$$15 \quad \lambda_t(i) = \alpha_t(i) \beta_t(i), \quad \text{对于全部 } i, \quad t=1, \dots, L \quad (7)$$

(iv) 根据需要确定相关量。例如, 设  $A_t^j$  为状态  $S_t$  的集合 =  $\{S_t^1, S_t^2, \dots, S_t^{km}\}$ , 使得  $S_t$  的第  $j$  个元素  $S_t^j$  等于零。对于普通非递归格子码,  $S_t^j = d_t^j, d_t^j$  为时刻  $t$  的第  $j$  个数据位。

因此, 译码器的软判决输出是

$$20 \quad P\{d_t^j = 0 / Y_1^L\} = \frac{1}{P\{Y_1^L\}} \sum_{S_t \in A_t^j} \lambda_t(m)$$

其中  $P\{Y_1^L\} = \sum_m \lambda_L(m)$ ,  $m$  是相应于状态  $S_t$  的指数。

25 把  $P\{d_t^j = 0 / Y_1^L\}$  应用到以下判决规则来获得译码器的硬判决或译码位输出:

$$30 \quad \hat{d}_t^j = \begin{cases} 0 & P\{d_t^j = 0 / Y_1^L\} > \frac{1}{2} \\ 1 & P\{d_t^j = 0 / Y_1^L\} < \frac{1}{2} \end{cases}$$



就是说，如果  $P\{d_i^j = 0/Y_1^L\} > \frac{1}{2}$ ，则  $\hat{d}_i^j = 0$ ；如果  $P\{d_i^j = 0/Y_1^L\} < \frac{1}{2}$ ，则  $\hat{d}_i^j = 1$ ；

否则，随机地指定  $d_i^j$  的值为 0 或 1。

作为上述步骤 (iv) 的相关量的另一个例子，概率  $\sigma_t$  的矩阵包括如下定义的元素：

5

$$\sigma_t(i, j) = P(S_{t-1} = i; S_t = j; Y_1^L) = \alpha_{t-1}(i) \gamma_t(i, j) \beta_t(j)$$

这些概率在需要确定编码器输出位的后验概率时有用。

在 MAP 译码算法的标准应用中，利用矢量  $\alpha_0 = (1, 0, 0, \dots, 0)$  来初始化正向递归，利用矢量  $\beta_L = (1, 0, \dots, 0)^T$  来初始化反向递归。这些初始条件是基于这样的假定的，即编码器的初始状态  $S_0 = 0$ ，其结束状态  $S_L = 0$ 。

10

根据本发明的一个实施例，循环 MAP 译码器通过如下地求解本征值问题来确定初始状态概率分布。设  $\alpha_t$ 、 $\beta_t$ 、 $\Gamma_t$  和  $\lambda_t$  同前，但初始  $\alpha_0$  和  $\beta_L$  如下：

15

把  $\beta_L$  定为列矢量  $(111 \dots 1)^T$ 。

设  $\alpha_0$  为未知 (矢量) 变量。

然后，

(i) 根据公式 (2) 计算  $\Gamma_t$ ， $t = 1, 2, \dots, L$ 。

20

(ii) 确定矩阵积  $\Gamma_1 \Gamma_2 \dots \Gamma_L$  的最大本征值。归一化相应的本征值，以便其分量和为单位值。这一矢量就是  $\alpha_0$  的解。本征值是  $P\{Y_1^L\}$ 。

(iii) 利用公式 (5) 规定的正向递归求出后续  $\alpha_t$ 。

(iv) 从  $\beta_L$  开始，如上所述初始化，利用公式 (6) 规定的反向递归求出  $\beta_t$ 。

25

(v) 按公式 (7) 计算  $\lambda_t$ ，以及上述的其它所需变量，例如软判决输出  $P\{d_i^j = 0/Y_1^L\}$  或概率  $\sigma_t$  的矩阵。

发明人已证明未知变量  $\alpha_0$  满足矩阵方程

$$\alpha_0 = \frac{\alpha_0 \Gamma_1 \Gamma_2 \dots \Gamma_L}{P\{Y_1^L\}}$$

30

根据该公式表示了概率间的关系这一事实，我们得知右侧  $\Gamma_t$  矩阵的积具

有等于  $P\{Y_t^L\}$  的最大本征值, 相应的本征矢量必定是概率矢量。

利用初始  $\beta_L = (111 \cdots 1)^T$ , 由公式 (6) 可求出  $\beta_{L-1}$ 。因此, 重复执行这一反向递归将求出全部  $\beta_L$ 。一旦  $\alpha_0$  已知和确定了  $\beta_L$ , 在本发明的循环 MAP 译码器中的全部计算就按照普通 MAP 译码算法。

5 图 4 是表示按照上述本征矢量方法译码纠错截尾格子码的循环 MAP 译码器 110 的简化方框图。译码器 110 包括一  $\Gamma_t$  计算器 112, 把  $\Gamma_t$  作为信道输出  $y_t$  的函数进行计算。该  $\Gamma_t$  计算器接收存储器 130 的以下内容作为输入: 信道转换概率  $R(Y_t, X)$ , 编码器在时刻  $t$  从状态  $m'$  转换到状态  $m$  的概率  $P_t(m/m')$ , 以及在假定先前编码器状态是  $m'$  而当前编码器状态是  $m$  的情况下编码器的输出符号是  $X$  的概率  $q_t(X/m', m)$ 。该  $\Gamma_t$  计算器根据公式 (2) 求和全部可能的编码器输出  $X$  来计算  $\Gamma_t$  的每一元素。

$\Gamma_t$  的计算值被提供给矩阵积计算器 114, 以便利用例如接收自存储器、开关 118 和延迟电路 120 的单位矩阵 116 来产生矩阵积  $\Gamma_1 \Gamma_2 \cdots \Gamma_L$ 。在时刻  $t = 1$ , 该单位矩阵被作为一个输入提供给该矩阵积计算器。对于从  $t = 2$  至  $t = L$  的每一后续时刻, 矩阵积  $\prod_{i=1}^{t-1} \Gamma_i$  通过延迟电路反馈给该矩阵积计算器。然后, 在时刻  $t = L$ , 得到的矩阵积通过开关 121 提供给归一化本征矢量计算机 122, 该归一化本征矢量计算机 122 计算相应于给其输入的矩阵积的最大本征值的归一化本征矢量。在  $\alpha_0$  如此被初始化的条件下, 即在这一归一化本征矢量的条件下, 如图所示, 利用延迟电路 126 和开关电路 128 在矩阵积计算器 124 中按照公式 (5) 递归地确定后续  $\alpha_t$  矢量。从存储器 130 检索  $\Gamma_t$  的恰当值, 然后把得到的  $\alpha_t$  存储在存储器 130 内。

利用开关电路 134 和延迟电路 136 在矩阵积计算器 132 中按照公式 25 (6) 确定  $\beta_t$  的值。然后在逐个分量积计算器 140 中按照公式 (7) 根据  $\alpha_t$  和  $\beta_t$  的值计算概率  $\lambda_t$ 。  $\lambda_t$  的值被提供给译码位值概率计算器 150, 该译码位值概率计算器 150 确定时刻  $t$  的第  $j$  个译码位  $d_t^j$  等于零的概率。该概率被提供给执行以下判定规则的阈值判定装置 152: 如果计算器 150 的概率大于  $1/2$ , 则认为译码位是零; 如果该概率小于  $1/2$ , 则认为译码位是 1; 如果等于  $1/2$ , 则译码位被随机地指定值 0 或 1。阈值判定装置的输出是时刻  $t$  的译码器输出。

译码位等于零的概率 ( $P\{d_t^j = 0\}$ ) 如图 4 所示还被提供给软输出函

数方框 54，以便作为译码器的软判决输出提供概率函数、即  $f$  ( $P\{d_i^j=0\}$ )，这种概率函数例如可以是

$$\text{似然比} = \frac{1 - P\{d_i^j=0\}}{P\{d_i^j=0\}}.$$

5  $P\{d_i^j=0\}$  的另一有用函数是

$$\log \text{似然比} = \log \left\{ \frac{1 - P\{d_i^j=0\}}{P\{d_i^j=0\}} \right\}.$$

或者，方框 154 的有用函数可仅仅是单位函数，以便软输出就是  $P\{d_i^j=0\}$ 。

本发明一替代实施例的循环 MAP 译码器利用递推方法确定状态概  
10 率分布。特别是，在一实施例（动态收敛方法）中，在检测到译码器收  
敛之前继续进行递推。在这种递推（即动态收敛）方法中，上述本征矢  
量方法的步骤（ii）和（iii）被以下步骤代替：

（ii.a）从等于  $(1/M, \dots, 1/M)$  的初始  $\alpha_0$  开始，其中  $M$  是格构中  
的状态数目，计算  $L$  次正向递推。归一化结果，以便每一新  $\alpha_t$  的分量的  
15 和等于单位值。保留全部  $L$  个  $\alpha_t$  矢量。

（ii.b）设  $\alpha_0$  等于先前步骤的  $\alpha_L$ ，并从  $t=1$  开始，再计算前  $L_{wmin}$  个  $\alpha_t$   
概率矢量。就是说，计算  $\alpha_t(m) = \sum_{i=0}^{M-1} \alpha_{t-1}(i) r_t(i, m)$ ， $m=0,$   
1,  $\dots, M-1$  而  $t=1, 2, \dots, L_{wmin}$ ，其中  $L_{wmin}$  是合适的最少格  
构级数。同前地进行归一化。只保留在步骤（ii.a）和（ii.b）中利用  
20 递推确定的最新  $L$  个  $\alpha$  的集合和在步骤（ii.a）中先前确定的  $\alpha_{L_{wmin}}$ 。

（ii.c）将步骤（ii.b）的  $\alpha_{L_{wmin}}$  与步骤（ii.a）的先前确定的集合作

比较。如果新和者的 $\alpha L_{w_{\min}}$ 的  $M$  个相应分量都在容限范围内，就到达上述步骤 ( iv )。否则，到达步骤 ( ii.d )。

( ii.d ) 设  $t=t+1$  并计算  $\alpha_t = \alpha_{t-1}\Gamma_t$ 。同前地进行归一化。只保留所计算的最新  $L$  个  $\alpha$  的集合和先前在步骤 ( ii.a ) 中确定的  $\alpha_t$ 。

- 5 ( ii.e ) 将新的  $\alpha_t$  与先前确定的集合作比较。如果  $M$  个新和老的  $\alpha_t$  在容限范围内，就到达步骤 ( iv )。否则，如果两个最新矢量在容限范围内不一致，并且如果递推次数没有超过规定的最大值 ( 一般是  $2L$  )，就到达步骤 ( ii.d )；如果不是这样，就到达步骤 ( iv )。

10 本方法然后继续到以上对于本征矢量方法所给出的步骤 ( iv ) 和 ( v )，以便产生循环 MAP 译码器的软判决输出和译码输出位。

在如美国专利申请第 ( RD - 24923 ) 号中所描述的循环 MAP 译码器的另一替代实施例中，修改采用上述递推方法的循环 MAP 译码器，使得译码器第二次只需处理预定的固定数目的格构级，即只需处理预定的回绕深度。因为译码所需的计算次数对于每一组编码信息都是相同的，所以对于实施是有利的。这样一来就减轻了硬件和软件的复杂性。

20 估算截尾卷积码 MAP 译码的所需回绕深度的一种方法是利用硬件或软件实验来对其进行确定，要求设计出具有可变回绕深度的循环 MAP 译码器，作实验对于逐渐增大的回绕深度测量相对于  $E_b/N_0$  的译码位差错率。在回绕深度的进一步增大不减小差错概率时，就确定了给出译码位差错的最小概率的最小译码器回绕深度。

25 如果大于在规定的  $E_b/E_0$  情况下可实现的最小值的译码位差错率是可容许的，就可以减少循环 MAP 译码器处理的格构级的所需数目。特别是，一旦获得所希望的位差错平均概率，就可简单地终止上述回绕深度搜索。

确定已知代码的回绕深度的另一种方法是利用该代码的距离特性。为此目的，需要定义两种不同的译码器判决深度。在此所使用的术语“正确路径”是指贯穿因编码一组数据位而得到的格构的状态序列或

路径。术语“错误的节点子集”是指正确路径节点外的全部错误的（格构）分支及它们的子分支的集合。以下定义两种判决深度都依赖于卷积编码器。（为说明起见，此处相对于卷积编码器描述本发明的这一实施例，但是，应懂得本发明不受卷积码的限制。）

5 判决深度定义如下：

（i）把纠  $e$  个错的正向判决深度  $LF(e)$  定义为一格构中的第一深度，在这一深度下，正确路径初始节点的错误子集中的全部路径——不管以后是否汇合至正确路径——离开正确路径的汉明距离都大于  $2e$ 。 $LF(e)$  的意义是：如果初始节点的差错是  $e$  个或小于  $e$  个并已知在该初始节点处开始编码，则译码器必然正确地译码。J. B. Anderson 和 K. Balachandran 在他们发表在 IEEE Transactions on Information Theory（1989 年 3 月 IT - 35 卷 455 - 459 页）上的论文“卷积码的判决深度”中提供了卷积码的正向判决深度的正式造表。在这一论文中以及还由 J. B. Anderson 和 S. Mohan 在《信源和信道编码——一种算法近似》（Kluwer Academic 出版社，Norwell，麻萨诸塞州，1991 年）中公开了  $LF(e)$  的一些性质。这些性质中主要的一种是在  $LF$  和  $e$  之间存在简单的线性关系；例如，对于比率  $1/2$  码， $LF$  约是  $9.08e$ 。

15 （ii）接着把纠  $e$  个错的不汇合判决深度  $LU(e)$  定义为一格构中的第一深度，在这一深度下，该格构中的从未与正确路径接触的全部路径离开正确路径的汉明距离大于  $2e$ 。

软判决循环 MAP 译码的  $LU(e)$  的意义是：在译码器处理了  $LU(e)$  个格构级之后，区分在实际发送路径上的一状态的概率将较大。因此，循环 MAP 译码的最小回绕深度是  $LU(e)$ 。深度  $LU(e)$  的计算表明其总是大于  $LF(e)$ ，但遵循相同的近似规律。这意味着如果代码的不汇合判决深度不是已知的，就可把最小回绕深度估算为正向判决深度  $LF(e)$ 。

25 通过确定一给定编码器的最小不汇合判决深度，我们就确定了必需被产生软判决输出的实际循环译码器处理的最少格构级数。以上引用的 J. B. Anderson 和 K. Balachandran 的论文“卷积码的判决深度”描述了确定正向判决深度  $LF(e)$  的算法。为了确定  $LU(e)$ ：

（i）从左至右延伸代码格构，同时从全部格构节点开始，零状态除外。

( ii ) 在每一级中, 删除汇合至正确 ( 全零 ) 路径的任何路径; 不延伸从正确 ( 零 ) 状态节点出发的任何路径。

( iii ) 在级  $k$  中, 确定在这一级中终止于各节点的路径的最小汉明距离, 即权重。

5 ( iv ) 如果该最小距离大于  $2e$ , 就停止。然后,  $LU ( e ) = k$ 。

如在美国专利申请第 ( RD - 24923 ) 号中所描述的, 利用计算机仿真的实验给出了两个出乎意料的结果: ( 1 )  $\beta_t$  的回绕处理改善译码器性能; ( 2 )  $LU ( e ) + LF ( e ) = 2LF ( e )$  的回绕深度的应用显著改善性能。这两个出乎意料的结果促进了基于递推的截尾格子码循环 MAP 译码器的改进。因此, 基于递推的循环 MAP 译码器算法的一最佳实施例包括以下步骤:

( i ) 按照公式 ( 2 ) 计算  $\Gamma_t$ ,  $t=1, 2, \dots, L$ 。

15 ( ii ) 以初始  $\alpha_0$  等于  $( 1/M, \dots, 1/M )$  开始, 其中  $M$  是格构中的状态的数目, 计算公式 ( 5 ) 的正向递推  $( L + L_w )$  次,  $u = 1, 2, \dots, ( L + L_w )$ , 其中  $L_w$  是译码器的回绕深度。格构级指数  $t$  取值  $( ( u-1 ) \text{ 模 } L ) + 1$ 。一旦译码器回绕了来自信道的接收的符号序列, 把  $\alpha_L$  看作是  $\alpha_0$ 。归一化结果, 以使每一新  $\alpha_t$  的分量之和等于单位值。保留通过递推确定的  $L$  个最新的  $\alpha$  矢量。

20 ( iii ) 以初始  $\beta_L$  等于  $( 1, \dots, 1 )^T$  开始, 计算公式 ( 6 ) 的反向递推  $( L + L_w )$  次,  $u = 1, 2, \dots, ( L + L_w )$ 。格构级指数  $t$  取值  $L - ( u \text{ 模 } L )$  一旦译码器回绕了接收序列, 就在计算新的  $\beta_L$  时把  $\beta_1$  作为  $\beta_{L+1}$  和把  $\Gamma_1$  作为  $\Gamma_{L+1}$ 。归一化结果, 以使每一新  $\beta_t$  的分量之和等于单位值。再次保留通过这一递推确定的  $L$  个最新的  $\beta$  矢量。

25 本最佳递推方法的下一个步骤与以上对于本征矢量方法给出的步骤 ( V ) 相同, 利用循环 MAP 译码器产生软判决和译码位输出。

图 5 是表示本发明的本最佳实施例的循环 MAP 译码器 180 的简化方框图。该译码器 180 包括把  $\Gamma_t$  作为信道输出  $y_t$  的函数进行计算的一  $\Gamma_t$  计算器 182。信道输出  $y_1, \dots, y_L$  通过开关 184 提供给该  $\Gamma_t$  计算器。在该开关处于向下位置时,  $L$  个信道输出符号每次一个地被装入  $\Gamma_t$  计算器 182 和移位寄存器 186。然后开关 184 被移至向上位置, 以便使该移位寄存器能够把前  $L_w$  个接收符号再移入该  $\Gamma_t$  计算器, 即提供循环处理。该  $\Gamma_t$  计算器从存储器 196 接收信道转换概率  $R ( Y_t, X )$ 、编码

器在时刻  $t$  从状态  $m'$  转换到状态  $m$  的概率  $P_t(m/m')$  以及在假定先前编码器状态是  $m'$  而当前编码器状态是  $m$  的情况下编码器的输出符号是  $X$  的概率  $q_t(X/m', m)$  作为输入。该  $\Gamma_t$  计算器根据公式 (2) 求和全部可能的编码器输出  $X$  来计算  $\Gamma_t$  的每一分量。

5  $\Gamma_t$  的计算值被提供给矩阵积计算器 190，该矩阵积计算器 190 用经延迟电路 192 和多路分解器电路 194 循环地提供的  $\alpha_{t-1}$  矩阵乘  $\Gamma_t$  矩阵。在  $t=1$  时，控制信号 CNTRL1 使多路分解器 194 从存储器 196 选择  $\alpha_0$  作为矩阵积计算器 190 的一个输入。当  $2 \leq t \leq L$  时，控制信号 CNTRL1 使多路分解器 194 从延迟电路 192 选择  $\alpha_{t-1}$  作为矩阵积计算器 190 的一个输入。在需要时把  $\Gamma_t$  和  $\alpha_t$  的值存储在存储器 196 内。

10 利用延迟电路 202 和多路分解器电路 204 在矩阵积计算器 200 中递归地计算  $\beta_t$  矢量。在  $t=L-1$  时，控制信号 CNTRL2 使多路分解器 204 从存储器 196 中选择  $\beta_L$  作为矩阵积计算器 200 的一个输入。当  $L-2 \geq t \geq 1$  时，控制信号 CNTRL2 使多路分解器 204 从延迟电路 202 中选择  $\beta_{t+1}$  作为矩阵积计算器 200 的一个输入。如上所述，得到的  $\beta_t$  的值在逐个象素积计算器 206 内被乘以  $\alpha_t$  的值，以便提供概率  $\lambda_t$ 。按照与以上参看图 4 描述的方式相同的方式， $\lambda_t$  的值被提供给译码位值概率计算器 150，该译码位值概率计算器 150 的输出被提供给阈值判定装置 152，得到译码器的译码输出位。

20 译码位等于零的概率 ( $P\{d_t^j = 0/y_t^j\}$ ) 如图 5 所示还被提供给软输出函数方框 154，以便作为译码器的软判决输出提供概率函数、即  $f(P\{d_t^j = 0/y_t^j\})$ ，这种概率函数例如可以是

$$\text{似然比} = \frac{1 - P\{d_t^j = 0/y_t^j\}}{P\{d_t^j = 0/y_t^j\}}$$

25

$P\{d_t^j = 0/y_t^j\}$  的另一有用函数是

$$\log \text{似然比} = \log \left\{ \frac{1 - P\{d_t^j = 0/y_t^j\}}{P\{d_t^j = 0/y_t^j\}} \right\}.$$

或者，方框 154 的有用函数可仅仅是单位函数，以便软输出就是

$P\{d_i^j = 0\}$ .

根据本发明，可以在经信道发送合成码字的位之前，通过按照有利地选定的模式删除由合成码字格式化器产生的合成码字中的选定的位来增大包括截尾非递归系统码的并行链接编码方案的比率。这一技术被称为收缩。这一收缩模式也为译码器所得知。由接收的合成码字-分量码字变换器执行的以下简单的附加步骤提供所需要的译码器操作：接收的合成码字-分量码字变换器在接收的分量码字产生期间仅仅插入一中性值来代替每一已知的收缩位。例如，该中性值是为了改善在加性白高斯噪声信道中进行对映传信的情况。译码器的其余操作如上所述。

10 如 S. Benedetto 和 G. Montorsi 在“并行链接卷积码的设计”（即将出版的 IEEE Transactions on Communications）中所描述的，因为 RSC 码对于相对较大的数据块长度的优越的距离性质，所以迄今已普遍认为非递归系统卷积码不适合作为并行链接编码方案中的分量码。但是，如上所述，发明人已发现 NSC 码的最小距离对数据块长度不那么敏感，  
15 因此这种码可被有益地应用于通过噪声很大的信道传送短的数据位块的通信系统。此外，发明人还已发现截尾码的应用解决了终止 turbo 码的输入数据序列的问题。此前没有提出过把截尾卷积码用作并行链接编码方案中的分量码。因此，本发明提供了并行链接非递归截尾系统卷积编码方案，译码器是循环 MAP 译码器，它译码分量截尾卷积码，以相  
20 对于信噪比的位差错率所量度的那样，对于短的数据块长度提供了优于普通 turbo 编码方案的性能。

虽然在此已描述了本发明的最佳实施例，但显然这些实施例只是说明性的。本领域的普通技术人员不脱离本发明可以作出许多改进和替换。因此，本发明将只受所附权利要求书的范围和精神的限制。

25



说明书附图

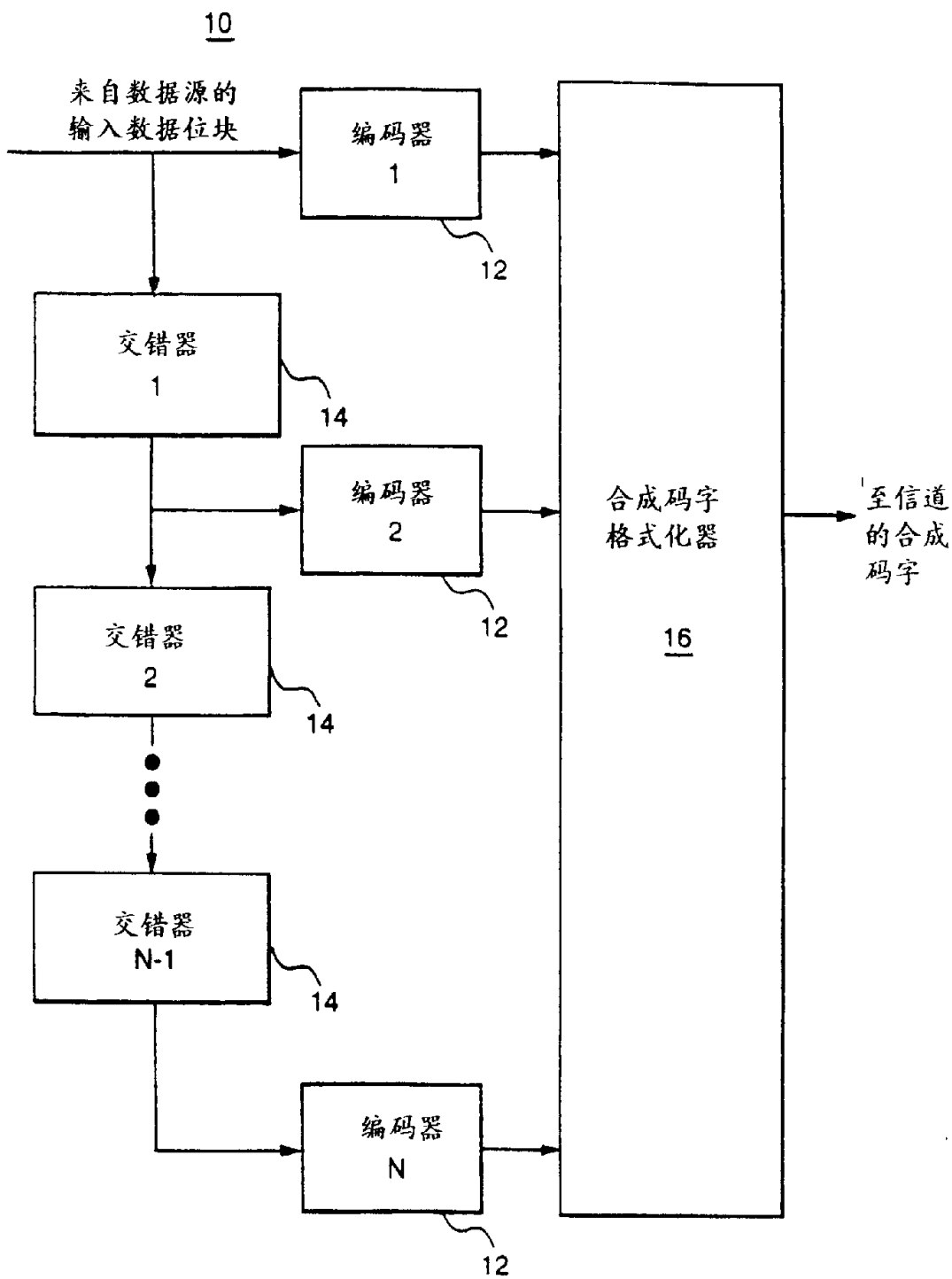


图 1

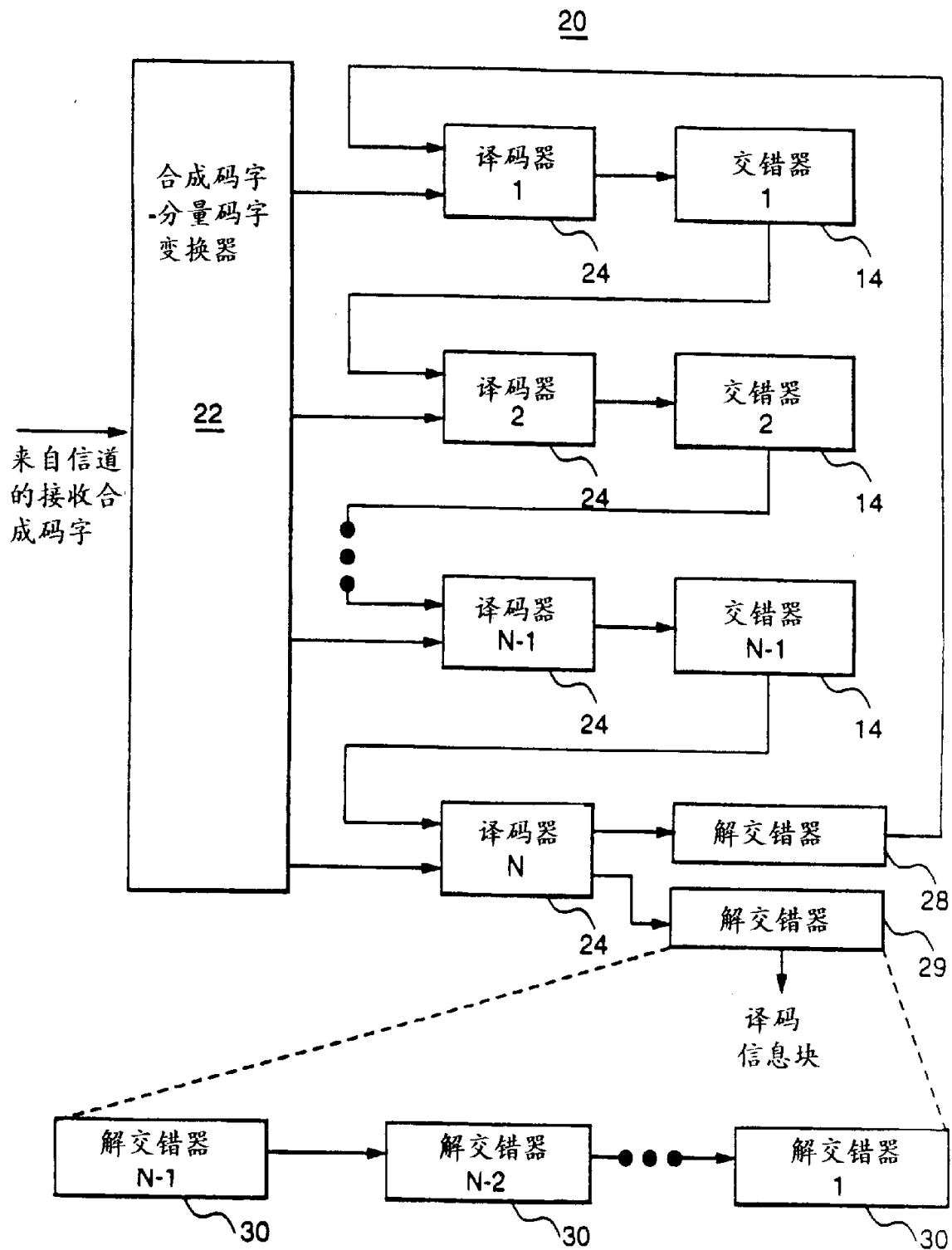


图 2

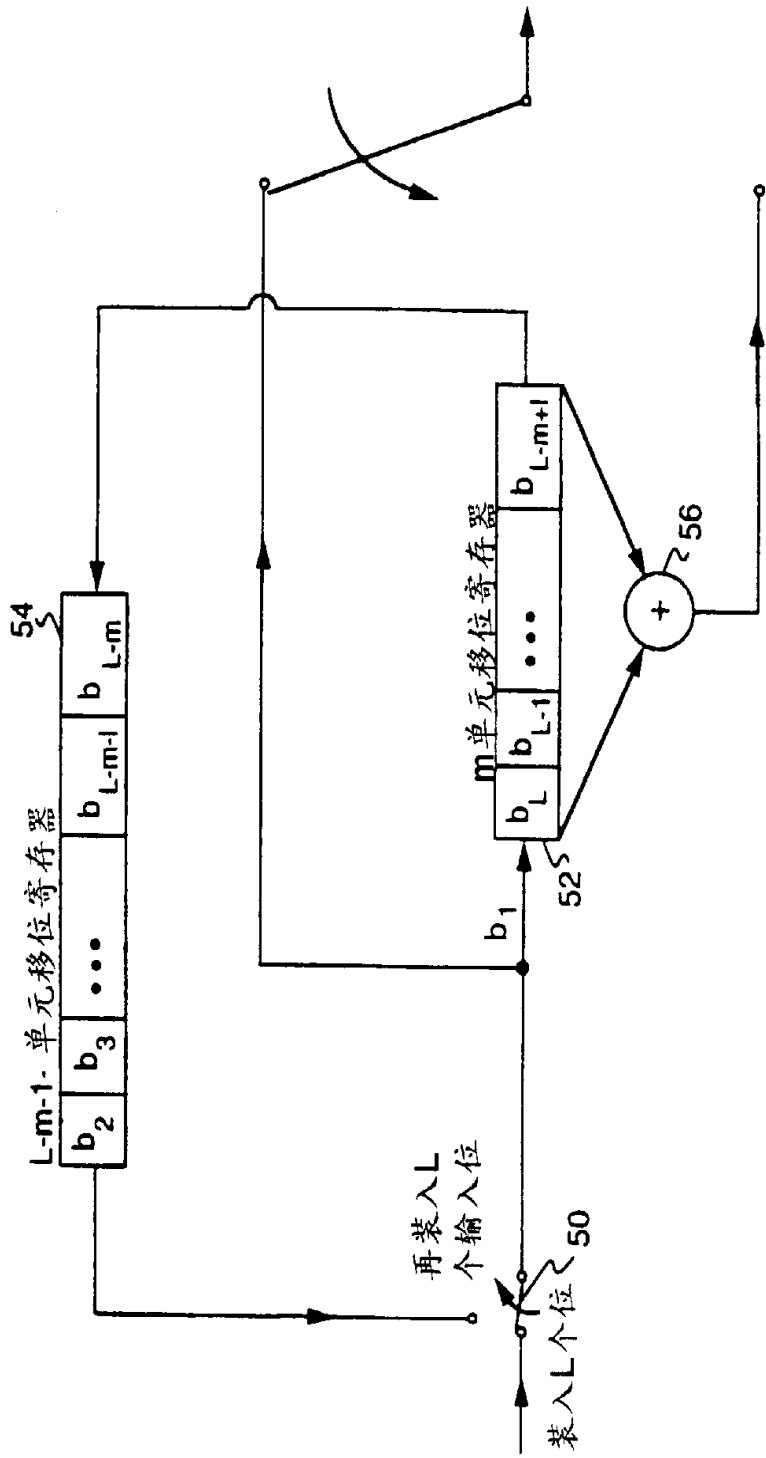


图 3

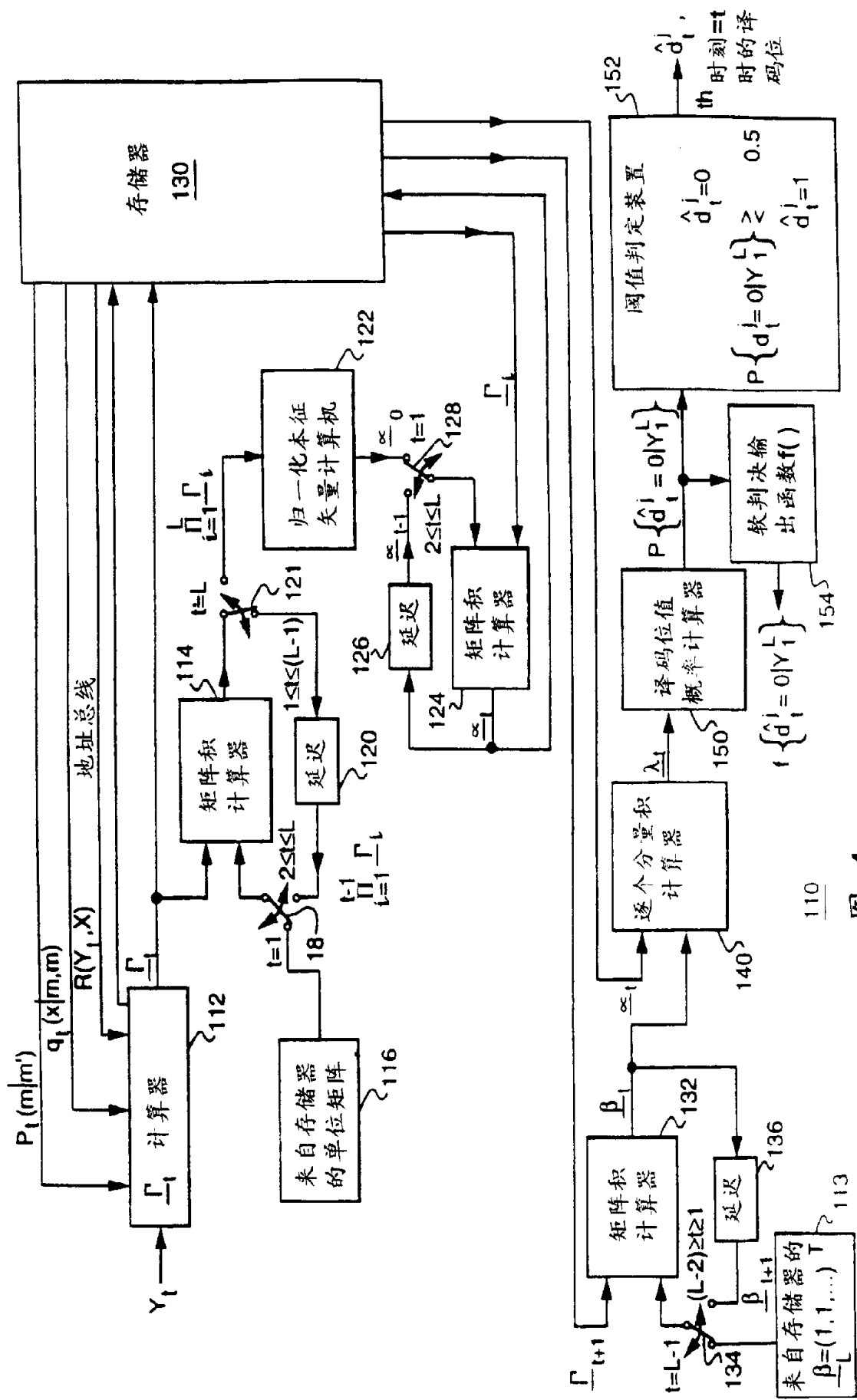


图 4

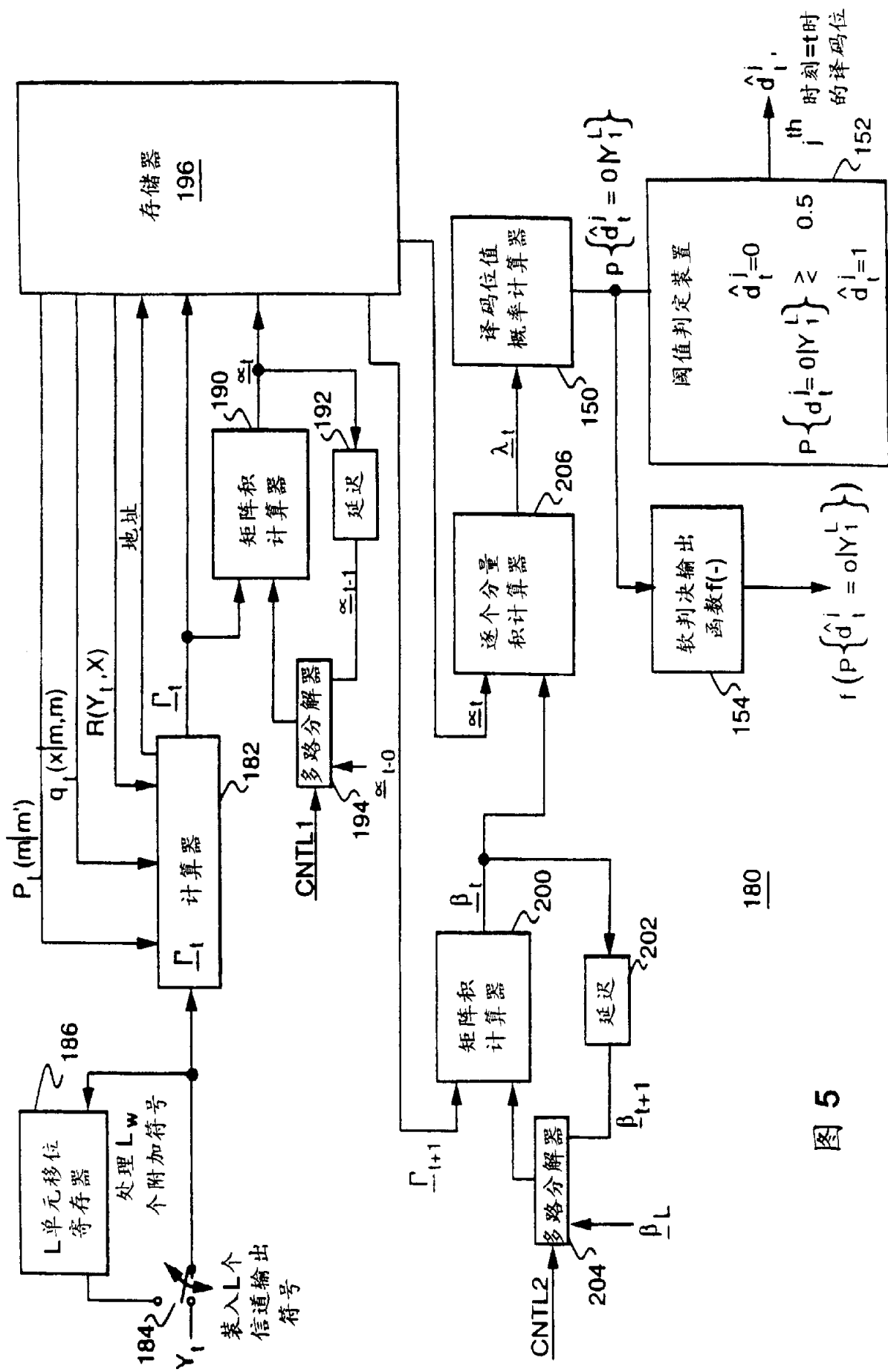


图 5