第 37 卷第 6 期　　电 子 与 信 息 学 报　　Vol.37 No.6
2015 年 6 月　　Journal of Electronics & Information Technology　　Jun. 2015

一种用于 RaptorQ 码的降维快速译码算法

郭　晓*　　张更新　　徐任晖　　牛大伟
(解放军理工大学通信工程学院　南京　210007)

摘　要：针对新型高效数字喷泉码—— RaptorQ 码译码复杂度高的问题，利用它是系统码的特性，该文提出一种降维快速译码算法。该算法利用预先计算的逆矩阵，将译码过程中对接收编码约束矩阵的求逆转化为对更小维数矩阵的求逆，以降低译码复杂度。算法译码效果与现有译码算法等价。仿真结果表明，在信道符号删除概率较低(小于 0.2)时，该算法的译码速度显著高于现有算法。

关键词：译码算法；数字喷泉；RaptorQ 码；降维译码
中图分类号：TN911.22　　**文献标识码**：A　　**文章编号**：1009-5896(2015)06-1310-07
DOI: 10.11999/JEIT141037

Fast Decoding Algorithm for RaptorQ Code Using Matrix Dimensionality Reduction

Guo Xiao　　Zhang Geng-xin　　Xu Ren-hui　　Niu Da-wei
(*Institute of Communications Engineering*, *PLA University of Science and Technology*, *Nanjing* 210007, *China*)

Abstract: RaptorQ code is a novel and efficient digital fountain code and its decoder is known to be too complicated. Considering the characteristic of the systematic code, a very fast decoding algorithm can be performed using matrix dimensionality reduction. The algorithm exploits a pre-calculated inverse matrix to achieve dimensionality reduction for the received code constraint matrix. As a result, the decoding complexity is reduced significantly while the failure-overhead curve is still identical to that of the conventional approaches. The simulations show that the decoding speed of the proposed algorithm outperforms the state-of-the-art algorithms, when the erasure probability of the channel is relatively low (less than 0.2).

Key words: Decoding algorithm; Digital fountain; RaptorQ code; Dimensionality reduction decoding

1　引言

RaptorQ 码[1]是数字喷泉技术的最新研究成果，目前被广泛应用于无线实时多媒体传输、文件分发、卫星通信等诸多领域[2−6]。文献[7]的研究结果表明，在分组长度小于10^4量级时，仅仅通过引入两个冗余符号，RaptorQ 码即可将译码失败概率降至10^{-6}量级；与 LT(Luby-Transform)码和 Raptor 码相比，RaptorQ 码为实现成功译码所需的冗余分组数大为降低。

然而，RaptorQ 码的性能提升是以编译码复杂度的提高为代价的。理论上，使用置信传播(Belief Propagation, BP)译码算法可在线性时间复杂度内对 Raptor 类码(包括 RaptorQ 码)进行译码[8]。但在实践中，受码长度的限制，单纯使用 BP 译码算法会使成功译码概率大大降低。为了实现在较少编码冗余的情况下提高成功译码的概率，当前实用的译码算法主要依赖于对接收编码约束矩阵 $\boldsymbol{A}'$ 的求逆运算[9]。文献[10]的研究表明，在符号长度 $T=4$ 的情况下，求逆运算所耗费的时间约占整个编译码时间的 99%；$T=1024$ 时，这一时间约占整个编译码时间的 95%。可见，RaptorQ 码的译码复杂度由矩阵求逆运算的复杂度决定。

目前，针对 RaptorQ 码编译码过程中求逆运算复杂度较高的问题，IETF RFC 6330[1]利用码约束矩阵具有稀疏性的特点，采用失活译码(Inactivation Decoding, ID)技术和高斯消元(Gaussian Elimination, GE)法，给出了一种有效的 RaptorQ 码译码算法，称为失活译码高斯消元(Inactivation Decoding Gaussian Elimination, IDGE)算法。文献[10]在此基础上提出了优化失活译码高斯消元(Optimized Inactivation Decoding Gaussian Elimination, OIDGE)算法，该算法通过简化译码步骤和对行选择方法进行优化，提高了译码计算的效率。文献[11]等利用 GPU 的并行结构，给出了

2014-08-04 收到，2014-10-31 改回
国家自然科学基金(91338201, 61032004)资助课题
*通信作者：郭晓　gosiuua@163.com

RaptorQ 码的并行译码算法。文献[12]利用 Sherman-Morrison 公式和预先计算的逆矩阵，给出了一种递归译码算法，该算法在信道符号删除概率较低时，相对于前述算法性能有较大提升。本文称该算法为递归逆矩阵译码(Recursive Matrix Inversion Decoding, RMID)算法。但该算法需要获得信道的先验符号删除概率，计算效率的提升依赖于对信道符号删除概率估计的准确性，有一定的应用局限性。

为了进一步降低译码复杂度，本文利用 RaptorQ 是系统码的特性，提出一种降维快速译码(Dimensionality Reduced Fast Decoding, DRFD)算法。该算法利用预先计算的逆矩阵，将译码过程中对接收编码约束矩阵 $\boldsymbol{A}'$ 的求逆转化为对更小维数矩阵 $\boldsymbol{A}''$ 的求逆，以降低译码复杂度。算法使用的降维变换方法未改变接收端编码约束矩阵的符号间约束关系，译码效果与现有译码算法等价。

2　RaptorQ 码编译码原理与译码复杂性分析

RaptorQ 可以看作一种无限码长的线性分组码，其编译码过程可由生成矩阵来表示，如图 1 所示。

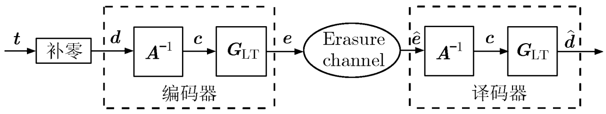


图 1　RaptorQ 码编译码原理框图

编码过程分为预编码和 LT(Luby-Transform)编码[13]两个步骤，预编码器首先将 K 个源符号编码成 L 个中间符号，LT 编码器再将 L 个中间符号编码成无限长码序列。RaptorQ 码的编译码运算是在 GF(256)上实现的，基本的运算单位为 Byte(8 bit)，为了提高编译码效率，通常将同时参与运算的若干个 Byte 视为一个符号。符号大小(即每个符号包含的字节个数)为 T , IETF RFC6330 推荐的取值范围为 4 至 1024。具体编码过程如下。在预编码阶段，K 个源符号构成的向量 $\boldsymbol{t}=(t_0,t_1,\cdots,t_{K-1})^{\mathrm{T}}$ 经过尾部补零操作后形成长度为 K' 的向量 $\boldsymbol{t}'$, K' 为 RaptorQ 码预置的源符号向量长度，$K'\ge K$ 。$\boldsymbol{t}'$ 再经过首部补足 $S+H$ 个零后形成输入向量 $\boldsymbol{d}=(d_0,d_1,\cdots,d_{L-1})^{\mathrm{T}}$，$S$ 为预编码矩阵 $\boldsymbol{A}$ 中 LDPC(Low Density Check Codes) 约束的个数，H 为 HDPC(High Density Check Codes) 约束的个数，$L=K'+S+H$ 。经过预编码器后生成中间符号向量 $\boldsymbol{c}=(c_0,c_1,\cdots,c_{L-1})^{\mathrm{T}}$，输入向量和中间符号向量的关系为

$$\boldsymbol{c}=\boldsymbol{A}^{-1}\cdot\boldsymbol{d} \tag{1}$$

预编码矩阵 $\boldsymbol{A}$ 由一系列子矩阵构成。

$$\boldsymbol{A}=\begin{pmatrix} \boldsymbol{G}_{\mathrm{LDPC1}} & \boldsymbol{I}_S & \boldsymbol{G}_{\mathrm{LDPC2}} \\ \hline \multicolumn{2}{c}{\boldsymbol{G}_{\mathrm{HDPC}}} & \boldsymbol{I}_H \\ \hline \multicolumn{3}{c}{\boldsymbol{G}_{\mathrm{LT}(i),1\le i<K'}} \end{pmatrix}_{L\times L} \tag{2}$$

其中，$\boldsymbol{I}_S$ 为 $S\times S$ 维单位阵，$\boldsymbol{I}_H$ 为 $H\times H$ 维单位阵，$\boldsymbol{G}_{\mathrm{LDPC1}}$ 和 $\boldsymbol{G}_{\mathrm{LDPC2}}$ 为行数 S 的 LDPC 矩阵，$\boldsymbol{G}_{\mathrm{HDPC}}$ 为 $H\times(L-H)$ 维 HDPC 矩阵，$\boldsymbol{G}_{\mathrm{LT}(i),1\le i<K'}$ 为 $K'\times L$ 维的 LT 约束矩阵，i 为内部符号标识(Internal Symbol Identifier, ISI)，RaptorQ 码可通过 ISI 唯一确定编码符号的 LT 约束关系，详见文献[1]。

在 LT 编码阶段，中间符号向量 $\boldsymbol{c}$ 经过 LT 编码器生成无限长编码符号向量 $\boldsymbol{e}=(e_0,e_1,\cdots,e_{K-1},\cdots)^{\mathrm{T}}$ 。其中由 $K'-K$ 个补零符号所生成的编码符号恒为零，接收端译码器可根据编码器参数进行重现，不需要进行传输。其余的符号用非负整数编号，即

$$\boldsymbol{e}=\boldsymbol{G}_{\mathrm{LT}(i'),i\in\mathbb{Z}^+}\cdot\boldsymbol{c} \tag{3}$$

$\mathbb{Z}^+$ 为非负整数集，i' 为编码符号标识(Encoding Symbol Identifier, ESI)。

在编码过程中，为了保证 RaptorQ 的系统码特性，预编码和 LT 编码使用了 K 个相同的 LT 约束关系，由

$$\begin{aligned}(e_0,e_1,\cdots,e_{K-1})^{\mathrm{T}}&=\boldsymbol{G}_{\mathrm{LT}(1\cdots K)}\cdot\boldsymbol{A}^{-1}\cdot\boldsymbol{d}\\&=(t_0,t_1,\cdots,t_{K-1})^{\mathrm{T}}\end{aligned} \tag{4}$$

可知，LT 编码器生成的前 K 个编码符号即为 K 个源符号。在生成的无限长码编码序列中，除去 K 个源符号的编码符号称为修复符号。

译码过程也分为两个步骤。第 1 步，通过接收符号向量 $\hat{\boldsymbol{e}}=(\hat{e}_{i_1},\hat{e}_{i_2},\cdots,e_{i_N})^{\mathrm{T}}$ 恢复出中间符号向量 $\boldsymbol{c}$, N 为编码符号序列经过删除信道后正确接收到编码符号的个数，$N\ge K$ 。由编码过程可知，该 N 个编码符号代表中间符号的 N 个 LT 约束关系，加上已知的 $K'-K$ 个补零 LT 约束关系、S 个 LDPC 约束关系和 H 个 HDPC 约束关系后，共有 $M=N+K'-K+S+H$ 个约束关系，构成接收编码约束矩阵 $\boldsymbol{A}'$，$\boldsymbol{A}'$ 为 $M\times L$ 维矩阵，其结构跟编码约束矩阵 $\boldsymbol{A}$ 类似，不同之处是其中的 K 个编码 LT 约束关系被 N 个接收符号的 LT 约束关系所替代。若 $\boldsymbol{A}'$ 可逆，中间符号 $\boldsymbol{c}$ 即可通过解方程组

$$\boldsymbol{A}'_{M\times L}\cdot\boldsymbol{c}=\begin{pmatrix}\hat{\boldsymbol{e}}_{N\times 1}\\ \boldsymbol{0}_{(M-N)\times 1}\end{pmatrix} \tag{5}$$

得到。若 $\boldsymbol{A}'$ 不可逆，方程组不具有唯一解，译码失败，需要接收更多的编码符号 e' ，直至 $\boldsymbol{A}'$ 可逆。第 2 步，由式(3)可得向量 $\boldsymbol{t}'$，去掉补足的零后即可得

源符号向量$\boldsymbol{t}$，从而完成译码过程。需要指出的是，本文所指的矩阵可逆是指矩阵的列满秩，即可以通过初等矩阵行变换将其上三角化。

从上述编译码过程可以看出，RaptorQ 码的编译码算法都依赖于对约束矩阵的求逆运算。在发送端，对于给定的编码长度K，预编码约束矩阵$\boldsymbol{A}$是固定的，其逆矩阵$\boldsymbol{A}^{-1}$可以通过预先计算的方式得到。在接收端，由于传输过程中符号丢失具有随机性，每一次参与译码运算的$\boldsymbol{A}'$是不同的，译码过程不可避免地需要对$\boldsymbol{A}'$进行求逆运算。一般的矩阵求逆算法，如 GE 算法，具有$O(N^3)$的计算复杂度。为了提高译码的效率，现有的译码算法[1,10,12]利用$\boldsymbol{A}'$具有稀疏性的特征，主要采用失活译码技术加快译码速度。在失活译码的过程中，每个迭代译码步骤都需要优先选择矩阵中非零元最少的行优先进行译码。该操作需要对矩阵元素进行扫描，占用大量的译码时间，在矩阵维数较高时，算法相当低效。综上所述，对高维矩阵的操作是现有译码算法效率不高的主要原因。

3 降维快速译码(DRFD)算法设计

3.1 DRFD 算法设计的出发点

在第 2 节介绍的两步译码过程中，首先需要求解式(5)恢复出L个中间符号，也就是求解一个包含M个线性约束关系和L个未知变量的线性方程组，该计算过程等价于对一个$M\times L$维矩阵进行求逆操作。观察进入 RaptorQ 码译码器输入端的接收符号向量$\hat{\boldsymbol{e}}=(\hat{e}_{i_1},\hat{e}_{i_2},\cdots,\hat{e}_{i_N})^{\mathrm{T}}$，$K$个源符号经过删除信道后，接收端收到其中的$s$个，其余为$r$个修复符号，$s+r=N$，译码器仅需要译出另外$K-s$个源符号即可得到源符号向量$\boldsymbol{t}$。RaptorQ 码的编译码过程都是线性运算，理论上，对一个包含r ($r\ge K-s$)个线性约束关系和$K-s$个未知变量的线性方程组求解，即有可能得到$K-s$个未知变量，等价于对$r\times(K-s)$维矩阵进行求逆操作。

RaptorQ 码具有极高的码率特性，其译码失败的概率为[14]

$$p_{\text{fail}}=\begin{cases}1, & N<K\\ 0.01\times 0.01^{N-K}, & N\ge K\end{cases}\tag{6}$$

由此可以看出，成功译码所需的编码符号数N以很高的概率等于参与编码符号个数K，即当修复符号的个数r约等于丢失符号的个数$K-s$时，译码即可以很高的概率成功译码。编码符号经过符号删除概率为p的删除信道后，丢失的源符号数量$K-s\approx K\cdot p$。当信道符号删除率$p\ll 1$时，$K\cdot p\ll K$，可得$r\approx K-s\approx K\cdot p\ll K<L\le M$。从这个数量关系看，在信道的符号删除概率较低时，第 2 节所述的两步译码方法是低效的。如果存在一种有效的方法将译码矩阵从$M\times L$维降为$r\times(K-s)$维，RaptorQ 码译码即可转化为对较小维数矩阵求逆的过程，从而提高译码速度。

3.2 降维变换

DRFD 算法利用预先计算的编码矩阵的逆$\boldsymbol{A}^{-1}$来实现接收端编码约束矩阵的降维变换。采用 DRFD 算法的译码器结构如图 2 所示。

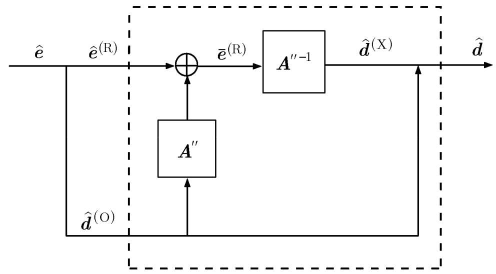


图 2 使用 DRFD 算法的 RaptorQ 码译码器结构

降维变换的原理可以用向量分解的方法进行说明。将含补零符号的待求的译码向量$\hat{\boldsymbol{d}}$分解成两个向量之和：

$$\hat{\boldsymbol{d}}=\hat{\boldsymbol{d}}^{(\mathrm{O})}+\hat{\boldsymbol{d}}^{(\mathrm{X})}\tag{7}$$

其中，$\hat{\boldsymbol{d}}^{(\mathrm{O})}$为已知符号向量，$s$个接收到的源符号被放置在相应的位置，其余元素为零。$\hat{\boldsymbol{d}}^{(\mathrm{X})}$为未知符号向量，$K-s$个未知符号作为未知变量被放置在相应的位置，其余元素为零。$\hat{\boldsymbol{d}}^{(\mathrm{O})}$和$\hat{\boldsymbol{d}}^{(\mathrm{X})}$均包含$L$个元素。

令包含r个修复符号的向量为$\hat{\boldsymbol{e}}^{(\mathrm{R})}=(\hat{e}_{i_s},\hat{e}_{i_{s+1}},\cdots,\hat{e}_{i_{s+r-1}})^{\mathrm{T}}$。在 RaptorQ 码的编码符号的生成过程中，修复符号的生成仅仅依赖于中间符号向量$\boldsymbol{c}$和该符号对应的 LT 约束关系，即

$$\begin{aligned}\hat{\boldsymbol{e}}^{(\mathrm{R})}&=\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}\cdot\boldsymbol{c}=\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}\cdot\boldsymbol{A}^{-1}\cdot\hat{\boldsymbol{d}}^{(\mathrm{O})}\\&\quad+\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}\cdot\boldsymbol{A}^{-1}\cdot\hat{\boldsymbol{d}}^{(\mathrm{X})}\end{aligned}\tag{8}$$

其中$\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}$为修复符号对应的LT约束关系矩阵，RI 为修复符号对应的 ESI 集合。对于给定的编码长度K，依据文献[1]给出的 RaptorQ 码编码构造方法，编码约束矩阵的逆矩阵$\boldsymbol{A}^{-1}$可以通过预先计算得到。

式(8)可以化简为

$$\bar{\boldsymbol{e}}^{(\mathrm{R})}=\boldsymbol{A}''\cdot\hat{\boldsymbol{d}}^{(\mathrm{X})}\tag{9}$$

其中$\boldsymbol{A}''=\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}\cdot\boldsymbol{A}^{-1}$，$\bar{\boldsymbol{e}}^{(\mathrm{R})}=\hat{\boldsymbol{e}}^{(\mathrm{R})}+\boldsymbol{A}''\cdot\hat{\boldsymbol{d}}^{(\mathrm{O})}$。$\hat{\boldsymbol{d}}^{(\mathrm{X})}$为$L$维向量，但仅包含$K-s$个未知源符号，其余元素为零，因此$\boldsymbol{A}''$中仅有未知符号对应的$K-s$个列参与$\bar{\boldsymbol{e}}^{(\mathrm{R})}$的生成，$\boldsymbol{A}''$可以看作$r\times(K-s)$维矩

阵，从而完成降维变换。

给出下面的定理，该定理为降维变换提供理论依据。

定理 1　对于 RaptorQ 码的译码，式(9)有唯一解，当且仅当式(5)有唯一解。

证明　RaptorQ 码的设计保证了预编码矩阵 $\boldsymbol{A}_{L\times L}$ 是可逆的，即其所有 L 个行是线性无关的。在接收编码约束矩阵 $\boldsymbol{A}'_{M\times L}$ 中，s 个源符号、S 个 LDPC 约束关系、H 个 HDPC 约束关系、$K'-K$ 个补零约束关系所对应的行与 $\boldsymbol{A}_{L\times L}$ 相同，因此这些行也构成线性无关组。

当式(5)有唯一解时，$\mathrm{rank}(\boldsymbol{A}'_{M\times L})=L$。去除矩阵 $\boldsymbol{A}'_{M\times L}$ 中的 $s+S+H+(K'-K)$ 个线性无关的行，剩余矩阵的秩为 $L-(s+S+H+(K'-K))=K-s$，即 $\mathrm{rank}(\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}})=K-s$。$\boldsymbol{A}^{-1}$ 为满秩矩阵，由 $\boldsymbol{A}''=\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}\cdot\boldsymbol{A}^{-1}$ 可知 $\mathrm{rank}(\boldsymbol{A}'')=\mathrm{rank}(\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}})=K-s$，故式(9)有唯一解。

当式(5)无唯一解时，$\mathrm{rank}(\boldsymbol{A}'_{M\times L})=L-\delta$，$0<\delta<L$。同理，$\mathrm{rank}(\boldsymbol{A}'')=K-s-\delta$，$\boldsymbol{A}''$ 不可逆，式(9)没有唯一解。　证毕

定理 1 说明，降维变换并没有改变接收编码符号的可译特性。式(9)使用了与式(5)相同的编码约束关系，即降维译码方法与传统的两步译码方法译码效果等价。

3.3 DRFD 算法

如第 2 节所述，传统 RaptorQ 码的译码算法依赖于对接收编码约束矩阵 $\boldsymbol{A}'$ 进行求逆，在能够进行成功译码的前提下，$\boldsymbol{A}'$ 的维数仅取决于编码器的参数 K，不受信道删除率的影响。而 DRFD 算法则依赖于对降维矩阵 $\boldsymbol{A}''$ 的求逆，$\boldsymbol{A}''$ 的维数受信道删除率的影响。当信道符号删除率 $p\ll 1$ 时，$\boldsymbol{A}''$ 相对于式(5)中的 $\boldsymbol{A}'$，维数大大降低。

以 $K=10$ 为例，根据文献[1]给出的 RaptorQ 码编码构造方法，编码器向信道发送 11 个编码符号。假设信道分组删除率约为 0.1，接收端接收到 9 个源符号和 1 个修复符号，加入编码约束关系后，接收端编码约束矩阵为 $\boldsymbol{A}'_{27\times 27}$。现有的 GE 算法、IDGE 算法和 OIDGE 算法都需要对这个 27×27 维矩阵执行求逆运算，以恢复出 27 个中间符号。RMID 算法虽然不直接对 $\boldsymbol{A}'$ 进行求逆，但在递归求逆的过程中，使用到的中间矩阵维数仍然是 27×27 维的。利用本文所提算法，经过降维变换后，$\boldsymbol{A}''$ 为 1×1 维矩阵，直接可以计算出丢失的源符号。

经过降维变换后，解式(9)即可得到丢失的源符号。丢失的源符号与接收到的源符号合并，即可得到完整的译码向量 $\hat{\boldsymbol{d}}$。

对式(9)通常使用高斯消元法求解，若高斯消元成功，即可完成译码过程；若高斯消元法失败，则需要接收更多的修复符号以完成译码过程。在需要多次译码情况下，可以采用文献[15]提出的渐增译码算法，以减少重复执行高斯消元法需要的计算开销。

表 1 给出 DRFD 算法的步骤。

表 1　DRFD 算法

输入：接收到的源符号 ESI 集合 OI，修复符号 ESI 集合 RI，接收符号向量 $\hat{\boldsymbol{e}}$，预计算的编码矩阵的逆 $\boldsymbol{A}^{-1}$。

初始化：$r=|\mathrm{RI}|$；$\hat{\boldsymbol{d}}^{(\mathrm{O})}=(\cdots,\hat{\boldsymbol{e}}_i,\cdots)^{\mathrm{T}}, i\subset\mathrm{OI}$；$\hat{\boldsymbol{e}}^{(\mathrm{R})}=(\cdots,\hat{\boldsymbol{e}}_i,\cdots)^{\mathrm{T}}, i\subset\mathrm{RI}$。

步骤 1　利用修复符号的 LT 约束关系生成矩阵 $\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}=(g_{ij})^{r\times L}$；

步骤 2　利用下面算法计算 $\boldsymbol{A}''=\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}\cdot\boldsymbol{A}^{-1}$

for $i=1:r$ do

for $k=1:L$ do

if $g_{ik}\neq 0$ then

$\boldsymbol{a}''_i=\boldsymbol{a}''_i+\boldsymbol{a}^{-1}_k$

步骤 3　计算 $\bar{\boldsymbol{e}}^{(\mathrm{R})}=\hat{\boldsymbol{e}}^{(\mathrm{R})}+\boldsymbol{A}''\cdot\hat{\boldsymbol{d}}^{(\mathrm{O})}$

步骤 4　使用 GE 算法解方程式 $\bar{\boldsymbol{e}}^{(\mathrm{R})}=\boldsymbol{A}''\cdot\hat{\boldsymbol{d}}^{(\mathrm{X})}$。若 GE 算法成功，即可完成译码过程；否则，译码失败，等待接收更多的编码分组，转到步骤 1。

从表 1 给出的 DRFD 算法可译看出，该算法改变了传统的两步译码结构，不再需要先译出中间符号而后译出源符号，而是直接通过接收的编码符号译出丢失的源符号。

在执行 DRFD 算法之前，译码需要等待接收 N 个编码符号。由于 RaptorQ 码的高可译特性，通常取 $N=K+2$。采用此取值可将译码失败的概率降到 10^{-6} 量级，同时引入的解码时延和计算开销都比较小。译码器在收到编码符号的同时，可以通过某种同步方式获得每个符号的编码符号标识 ESI(如将 ESI 同编码符号一同发送至接收端)。译码器通过 ESI 可识别接收到的源符号和修复符号，这些作为算法参数输入至 DRFD 算法。

修复符号的 LT 约束关系由修复符号的 ESI 唯一确定，算法步骤 1 的实现方法详见文献[1]。矩阵 $\boldsymbol{G}_{\mathrm{LT}(i)}$ 为二进制稀疏矩阵，算法步骤 2 利用的该特征将矩阵乘法运算转化成矩阵的行累加运算，其中 $\boldsymbol{a}''_i$ 为 $\boldsymbol{A}''$ 的第 i 行，$\boldsymbol{a}''^{-1}_i$ 为 $\boldsymbol{A}^{-1}$ 的第 k 行。算法步骤 3 中，$\hat{\boldsymbol{d}}^{(\mathrm{O})}$ 为 L 维向量，但部分元素为零，在计算矩阵乘法时，$\boldsymbol{A}''$ 中部分列不参与运算，可以节约一些计算开销。算法步骤 4 中，$\hat{\boldsymbol{d}}^{(\mathrm{X})}$ 为 L 维向量，但

仅包含 $K-s$ 个未知源符号，其余元素为零，因此 $\boldsymbol{A}''$ 中仅有未知符号对应的 $K-s$ 个列参与 $\bar{\boldsymbol{e}}^{(\mathrm{R})}$ 的生成，$\boldsymbol{A}''$ 可以看作 $r\times(K-s)$ 维矩阵。

算法中所有的加法和乘法运算都是 GF(256)上的运算。算法步骤 2 中使用到预编码约束矩阵 $\boldsymbol{A}$ 的逆矩阵 $\boldsymbol{A}^{-1}$，该矩阵预先计算后存储在译码器端，在译码过程中，其计算开销可以忽略不计，但增加了译码器端的存储开销。

3.4 计算复杂度分析

DRFD 算法使用降维变换降低执行 GE 算法的矩阵的维数，以达到降低译码计算复杂度的目标。作为降维的代价，计算 $\boldsymbol{A}''$ 和 $\bar{\boldsymbol{e}}^{(\mathrm{R})}$ 的过程需要引入额外的矩阵乘法和加法，在一定程度上增加了计算量。但 $\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}$ 为二进制稀疏矩阵，利用算法中步骤 2 给出计算方法，对 $\boldsymbol{A}^{-1}$ 中的行进行异或操作即可计算出 $\boldsymbol{A}''$，不需要进行 GF(256)上的矩阵乘法。若信道的符号删除率为 p，则矩阵 $\boldsymbol{G}_{\mathrm{LT}(i),i\in\mathrm{RI}}$ 的行数 r 约为 pK，DRFD 算法中步骤 2 计算 $\boldsymbol{A}''$ 的译码计算复杂度为 $O(pKL)$，L 为中间符号的个数，略大于 K。$\hat{\boldsymbol{d}}^{(\mathrm{O})}$ 中约有 $(1-p)K$ 个非零元，矩阵 $\boldsymbol{A}''$ 的行数约为 pK，DRFD 算法中步骤 3 计算 $\bar{\boldsymbol{e}}^{(\mathrm{R})}$ 的译码计算复杂度为 $O(p(1-p)K^2)$。算法步骤 4 在矩阵 $\boldsymbol{A}''$ 上执行 GE 算法，计算复杂度 $O(p^3K^3)$。因此 DRFD 算法整体的计算复杂度为 $O(p^3K^3)$。

另外，求解式(9)可直接得到未知符号，不再需要利用矩阵乘法操作从中间符号 $\boldsymbol{c}$ 来恢复未知符号。在信道的符号删除率为 p 较小的情况下，相对于具有 $O(N^3)$ 计算复杂度的 GE 消元法，DRFD 算法可以大大减小计算开销。

4 仿真结果及分析

为了验证 DRFD 算法的性能，本节利用仿真实验将其与 GE, IDGE, OIDGE 和 RMID 算法进行比较。IDGE 算法参照文献[1]实现，OIDGE 算法参照文献[10]实现，GF(256)上的乘法运算采用查表法实现。仿真中所采用的信道为符号删除信道，且具有稳定的符号删除概率 p。参与 RaptorQ 编码的源符号个数 K 的取值范围为 10 到 2000，符号大小 T 取值分别为 4 和 128，符号删除概率 p 取值为 0.01, 0.10 和 0.20，对于每个(K , T , p)三元组进行 500 次实验，每次实验使用相同的编码数据，分别使用 5 种不同的译码算法进行译码。仿真算法采用 C 语言实现，仿真程序在同一台 PC 机(Intel i3 CPU@2.4G, DDR2@400MHz)上运行。

由于 RaptorQ 码的可译特性由码结构本身决定，上述 5 种译码算法具有相同的可译性能，本文主要对译码算法的计算复杂度进行比较，不对译码失败概率进行比较。各译码算法在实现过程中使用微秒精度的时间计数器，译码计算复杂度取译码时间的平均值作为仿真结果。

在给定分组个数 K 和符号大小 T 的条件下，由于 RMID 算法需要预先对信道符号删除概率进行估计，且算法性能受信道符号删除率和估计准确性两个因素的影响，本文所提的 DRFD 算法的性能仅受信道符号删除率影响，GE, IDGE, OIDGE 算法的性能则跟这两个因素无关。为了清楚说明各算法性能之间的关系，下文分两部分进行分析。

4.1 与 GE, IDGE 和 OIDGE 算法的比较

图 3(a)给出了在符号长度 $T=4$ 时，译码时间随源分组个数 K 的变化曲线。GE, IDGE, OIDGE 算法的译码时间仅受分组个数 K 和符号大小 T 影响，在图中分别给出一条曲线。

由图 3(a)可以看出，RFC6330 给出的 IDGE 算法译码速度较 GE 算法和 OIDGE 算法低，原因是 IDGE 算法采用了较为复杂的行选择算法，该算法虽然可以最大可能地利用接收编码约束矩阵 $\boldsymbol{A}'$ 的稀疏特性，但需要对 $\boldsymbol{A}'$ 的元素进行多次扫描，扫描矩阵元素需要消耗大量的计算时间；在 K 较小时，GE 算法的译码速度略优于 OIDGE 算法，随着 K 的不断增加，OIDGE 算法的译码速度显著优于 GE 算法。上述结果与文献[10]的结论一致。DRFD 算法译码速度在符号删除概率 p 较低时，译码性能显著优于上述 3 种算法，例如在 $p=0.1$ 时，对应于 K 为 100, 500 和 1500, DRFD 算法的译码速度是 OIDGE 算法的 17.4, 12.7 和 7.3 倍。随着 p 的增长，DRFD 算法的译码速度随之降低。这是由于随着 p 的增长，参与 DRFD 算法译码的修复分组个数随之增加，导致译码矩阵 $\boldsymbol{A}''$ 的规模随之增长，在 $\boldsymbol{A}''$ 上执行高斯消元算法的算法复杂度为 $O(N^3)$，从而增加了算法中步骤 4 的译码时间。

图 3(b)给出了在 $T=128$ 时的仿真结果。与图 3(a)的结果类似，在符号删除概率 p 较小时，DRFD 算法的译码速度同样优于现有 3 种译码算法，例如在 $p=0.1$ 时，对应于 K 为 100, 500 和 1500, DRFD 算法的译码速度是 OIDGE 算法的 5.3, 2.7 和 2.1 倍，但相对于 $T=4$ 时效率的提高有所减小。T 增大导致 DRFD 算法译码速度降低的主要原因是，在算法的步骤 3 中，T 增大将导致矩阵乘法的计算量增加，从而使步骤 3 占用大量的计算时间。

从图 3(a)和图 3(b)可以看出，随着 p 和 K 的增加，DRFD 算法的译码速度的优势会逐步降低。在 $p=0.2$，$K=2000$ 时，DRFD 算法的性能接近

OIDGE 算法。表明 DRFD 算法的应用有一定的局限性，即 DRFD 算法不适用于高误码率和分组块较大的应用场景。$K=2000$，$T=128$ 时，单个传输的数据分段大小为 256k Byte。单个传输分段大小小于 256k Byte、信道符号删除概率 p 小于 0.20，这些限制对于当前大多数的多媒体应用来说是可满足的[16]。

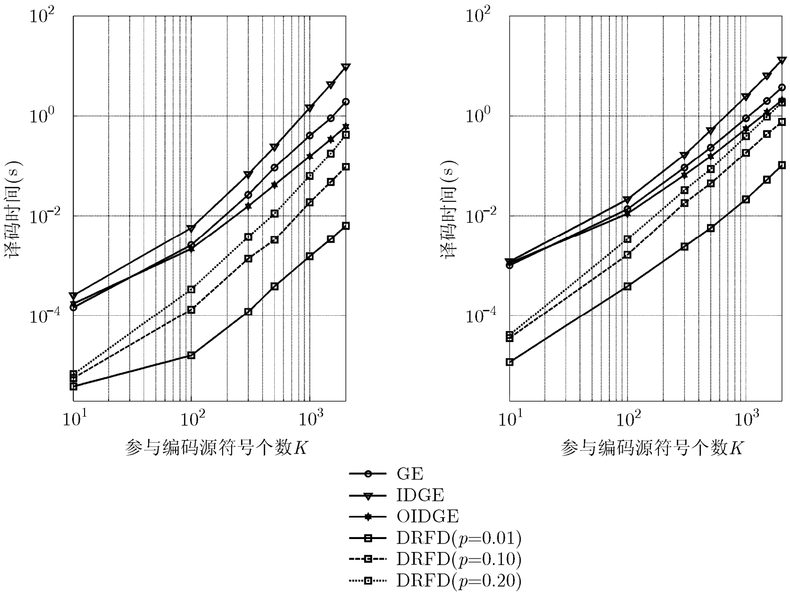
4.2 与 RMID 算法的比较

图 4 给出了 DRFD 算法与 RMID 算法在 $T=4$ 时，$p=0.01$ 和 $p=0.20$ 的仿真结果。由于 RMID 算法需要预先估计信道符号丢失率，故给出两条曲线，分别表示能够准确预先估计信道质量和估计的信道质量有误差时的情况。$\hat{p}$ 表示预先估计的信道符号丢失率。从图上可以看出，在给定的信道符号丢失率的情况下，对信道质量的估计值影响 RMID 算法的性能，估计值的误差会降低 RMID 算法的译码速度，这与文献[12]的结论一致。DRFD 算法不需要预先估计信道符号丢失率，故给出一条曲线。在两种信道符号丢失率的情况下，DRFD 算法的译码速度均优于 RMID 算法。

在 $T=128$ 时，DRFD 算法与 RMID 算法的对比仿真结论与上述仿真结论类似，本文不再赘述。

5 结束语

RaptorQ 码是一种高效的数字喷泉码，但由于译码运算需要对较高维数的接收端编码约束矩阵进行求逆运算，其译码的计算复杂度较高。本文利用预先计算的逆矩阵，提出了一种降维快速译码(DRFD)算法，该算法译码结果和现有译码算法等价，不改变现有算法对 RaptorQ 码的可译性，在信道的符号删除概率 p 较低(小于 0.2)时，译码速度显著优于现有算法。信道的符号删除率越低，DRFD 算法的性能提升约明显。该算法的性能虽然受信道符号删除率的影响，但不需要预先得到信道符号删除率的先验信息，相对于 RMID 算法，在提高译码速度的同时，提升了算法的可用性。



(a)符号大小 T=4时　　(b)符号大小 T=128时

图 3　DRFD 算法在不同符号删除率下与 GE, IDGE 和 OIDGE 算法性能比较

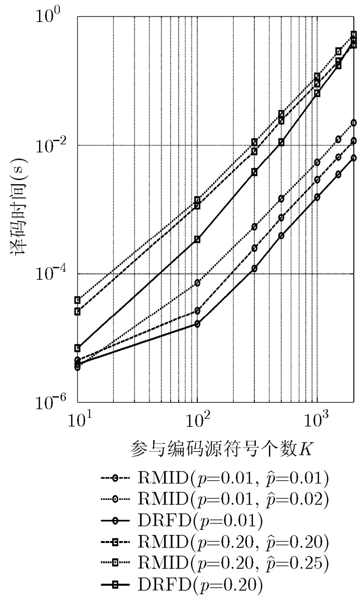


图 4　$T=4$ 时 DRFD 算法在不同符号删除率下与 RMID 算法性能比较

参 考 文 献

[1] IETF RFC 6330. RaptorQ forward error correction scheme for object delivery[S]. IETF Proposed Standard, 2011.

[2] Calabuig J, Monserrat J F, Gozálvez D, *et al*.. AL-FEC for streaming services in LTE E-MBMS[J]. *EURASIP Journal on Wireless Communications and Networking*, 2013, 2013(1): 1–12.

[3] Bouras C, Kanakis N, Kokkinos V, *et al*.. Embracing RaptorQ FEC in 3GPP multicast services[J]. *Wireless Networks*, 2013, 19(5): 1023–1035.

[4] Bouras C, Kanakis N, Kokkinos V, *et al*.. Application layer forward error correction for multicast streaming over LTE networks[J]. *International Journal of Communication Systems*, 2013, 26(11): 1459–1474.

[5] Pandya M A U, Trapasiya S D, and Chinnam S S. Implementation of AL-FEC RaptorQ code over 3GPP E-MBMS network[J]. *International Journal of Engineering*

Research and Technology, 2013, 2(5): 170–177.

[6] 黄晓可, 刘洛琨, 张剑, 等. RaptorQ 码级联方案在卫星通信中的应用[J]. 信息工程大学学报, 2013, 14(3): 306–311.
Huang Xiao-ke, Liu Luo-kun, Zhang Jian, *et al*.. Application of the RaptorQ codes concatenation in satellite communications[J]. *Journal of Information Engineering University*, 2013, 14(3): 306–311.

[7] Shokrollahi A and Luby M. Raptor codes[J]. *Foundations and Trends in Communications and Information Theory*, 2011, 6(3/4): 213–322.

[8] Shokrollahi A. Raptor codes[J]. *IEEE Transactions on Information Theory*, 2006, 52(6): 2551–2567.

[9] Kim S, Lee S, and Chung S Y. An efficient algorithm for ML decoding of Raptor codes over the binary erasure channel[J]. *IEEE Communications Letters*, 2008, 12(8): 578–580.

[10] Mladenov T, Nooshabadi S, Kim K. Efficient GF (256) raptor code decoding for multimedia broadcast/multicast services and consumer terminals[J]. *IEEE Transactions on Consumer Electronics*, 2012, 58(2): 356–363.

[11] Hu L, Nooshabadi S, and Mladenov T. Forward error correction with Raptor GF(2) and GF(256) codes on GPU[J]. *IEEE Transactions on Consumer Electronics*, 2013, 59(1): 273–280.

[12] Lu Y, Lai I, Lee C, *et al*.. Low-complexity decoding for RaptorQ codes using a recursive matrix inversion formula[J]. *IEEE Wireless Communications Letters*, 2014, 3(2): 217–220.

[13] Luby M. LT codes[C]. Proceeding of the 43rd Annual IEEE Symposium on the Foundations of Computer Science, Vancouver, Canada, 2002: 271–280.

[14] 3GPP Tdoc S4-110449. Rationale for MBMS AL-FEC Enhancements[R]. 3rd Generation Partnership Project (3GPP), 2011.

[15] Kim S, Ko K, and Chung S Y. Incremental Gaussian elimination decoding of raptor codes over BEC[J]. *IEEE Communications Letters*, 2008, 12(4): 307–309.

[16] Mladenov T, Nooshabadi S, and Kim K. MBMS raptor codes design trade-offs for IPTV[J]. *IEEE Transactions on Consumer Electronics*, 2010, 56(3): 1264–1269.

郭 晓： 男，1981 年生，讲师，博士生，研究方向为信道编码、深空通信、无线网络.

张更新： 男，1967 年生，教授，博士，博士生导师，研究方向为卫星通信、深空通信.

徐任晖： 男，1978 年生，讲师，博士，研究方向为认知无线电.

牛大伟： 男，1978 年生，讲师，博士，研究方向为无线网络、光交换网络.