

## 一种新型的卷积码混合译码算法

杨帆<sup>①</sup> 罗振东<sup>②</sup> 田宝玉<sup>①</sup>

<sup>①</sup>(北京邮电大学信息工程学院 北京 100876)

<sup>②</sup>(信息产业部电信研究院通信标准研究所 北京 100045)

**摘要:** 该文提出了一种级联的卷积码混合译码算法。该算法由两级译码实现,第 1 级采用置信传播(Belief-Propagation, BP)算法,而第 2 级采用修改的维特比译码(Modified Viterbi Decoding, MVD)算法。BP 首先对接收序列进行预译码,并利用伴随式将译码输出的对数似然比值分为可靠的和不可靠的两类。不可靠的对数似然比值用接收符号取代,可靠的部分硬判决为编码符号,它们共同组成混合序列。随后, MVD 对该混合序列作进一步纠错译码。仿真表明,与传统的维特比算法相比,所提出的混合译码算法的误码性能只有很小的损失,其译码平均复杂度在中高信噪比条件下有明显降低。

**关键词:** 混合译码; 置信传播; 维特比译码

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2009)05-1237-04

## A Novel Hybrid Decoding Algorithm for Convolutional Codes

Yang Fan<sup>①</sup> Luo Zhen-dong<sup>②</sup> Tian Bao-yu<sup>①</sup>

<sup>①</sup>(School of Information Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

<sup>②</sup>(Institute of Communication Standards Research China Academy of Telecommunication Research of Ministry of Information Industry (MII), Beijing 100045, China)

**Abstract:** A concatenated hybrid decoding algorithm for convolutional codes is presented. The algorithm is complemented by using two-stage decoding, where the first stage uses the Belief-Propagation (BP) algorithm, while the second stage uses the Modified Viterbi Decoding (MVD) algorithm. Firstly, the received sequence is pre-decoded by BP, and its outputs are divided into two groups which are reliable Log-Likelihood Ratios (LLRs) and unreliable LLRs. The hard decision symbols corresponding to reliable LLRs and the parts of received symbols corresponding to unreliable LLRs are formed a hybrid sequence, which is further corrected by MVD. Simulation shows that compared with the conventional Viterbi decoding algorithm, the proposed algorithm has a little performance deterioration with much lower average complexity at moderate-to-high signal to noise ratio.

**Key words:** Hybrid Decoding (HD); Belief-Propagation (BP); Viterbi Decoding (VD)

### 1 引言

众所周知,维特比译码(Viterbi Decoding, VD)<sup>[1]</sup>是一种性能优良的卷积码译码算法,其译码复杂度随着编码寄存器约束长度的增加而指数增加,当约束长度较大时,很难实际运用。序列译码中的堆栈(stack algorithm)算法<sup>[2]</sup>也是一种低复杂度译码算法,可用于较大的约束长度,由于它利用缓存来存储输入数据,对于有些编码序列需要很长的搜索过程,会导致缓存器溢出,使得信息丢失或删除,且其译码时间为一随机变量。文献[3]提出一种基于错误选择的算法,在保持与 VD 性能相当的前提下,一定程度上降低了复杂度。对于固定时间译码的算法而言, M 算法<sup>[4]</sup>不失为一种较为直接的实现方式,它在每个译码时间段,采用减少状态数的方法来

降低运算量。随着 LDPC 码的重新发现,其中具有代表性的一类,置信传播(Belief-Propagation, BP)算法<sup>[5]</sup>,能在二分图结构无环的情况下逼近最大似然译码的性能,并且由于它具有多项式的复杂度,受到人们的广泛重视。

本文提出了一种混合译码(Hybrid Decoding, HD)算法,它使用两个级联的译码器,分别应用 BP 串行译码<sup>[6]</sup>和修改的维特比译码(Modified Viterbi Decoding, MVD)。首先,第 1 级 BP 译码器利用伴随式将输出的对数似然比值(Log Likelihood Ratios, LLRs)分为可靠的和不可靠(质疑)的两类。可靠的部分硬判决为码符号(每个码符号包含 1bit 信息位和 1bit 校验位),不可靠的部分用接收符号取代,从而形成了一条由硬判决码符号和接收符号组成的混合序列。由于编码后的码字在网格图中表现为一条连续路径, MVD 利用该性质对混合序列进一步纠错。(1)对于混合序列中的码符号, MVD 只需判断当前码符号对应的网格图中的分支是否

与之前的译码路径相连接。若连接则继续判断下一个符号，否则从该时刻回溯一个约束长度，采用传统的VD对这一时间段内的接收符号重新译码。(2)对于混合序列的接收符号部分则直接使用传统VD进行处理。需要指出的是，MVD中调用的VD只是在局部范围内对相应的接收符号译码，我们称这个过程为“局部维特比译码(Partial Viterbi Decoding, PVD)”。HD算法可应用于编码寄存器阶数相对较大的情况，从原理上讲对所有的卷积码(系统与非系统码)都适用，且码率也不受限制。这是因为所有的卷积码都有其对应的校验矩阵，BP只需根据此校验矩阵译码即可。不失一般性，本文只考虑(2,1,L)递归系统卷积(Recursive Systematic Convolutional, RSC)码，其中L为寄存器长度。

粗略一看，HD由于加入了BP预译码，其复杂度应该比传统的只用单个维特比译码算法的复杂度高，但注意到此处BP只做少量迭代，且每次译码复杂度是线性的，当编码器阶数较大时，相对于VD的指数复杂度可以忽略不计。HD由于利用了BP预译码，在适中信噪比时，接收符号中的大部分错误信息能在第1级BP译码中得到纠正，使得混合算法译码的平均复杂度低于传统的VD算法。最差的情况下，亦即混合序列全由接收符号组成，此时MVD的复杂度与VD复杂度几乎相同，HD的复杂度只比VD算法略高，且它的复杂度不会超过BP与VD复杂度之和。其次由于在BP预译码后涉及到对LLRs的硬判决，貌似损失了部分信息，然而在后面将看到，由于MVD在对码符号重新纠错的过程中采用了回溯译码和扩展质疑码符号集的策略，使得最终的译码错误率得到有效的控制，并在很大程度上补偿了损失的信息，使得最终性能与传统VD性能相差很小。

本文结构安排如下。在第2节中，将首先给出系统模型，并阐明伴随式，断点等一些概念，然后详细介绍HD算法，第3节分析了HD算法与VD算法的复杂度，并给出了仿真数据，最后得出本文的结论。

## 2 混合译码算法HD

前面提到，HD算法主要是通过两级纠错的方法来降低译码复杂度，并取得近似最优的性能，下面将分别介绍这两级译码算法，同时阐明其中涉及到的一些概念。

### 2.1 系统模型

一长为N的信息序列 $\mathbf{u} = [u_1, u_2, \dots, u_N]^T$ 通过1/2码率的RSC编码器编码后得到码字 $\mathbf{c} = [c_1^s, c_1^p, c_2^s, c_2^p, \dots, c_{N+L}^s, c_{N+L}^p]$ ，上标s和p分别表示编码符号的系统位和校验位。将 $\mathbf{c}$ 作BPSK调制后的发送序列 $\mathbf{x} = [x_1^s, x_1^p, x_2^s, x_2^p, \dots, x_{N+L}^s, x_{N+L}^p]^T$ 送入噪声为 $\mathbf{n} = [n_1^s, n_1^p, n_2^s, n_2^p, \dots, n_{N+L}^s, n_{N+L}^p]^T$ 的AWGN信道，得到接收符号 $\mathbf{y} = [y_1^s, y_1^p, y_2^s, y_2^p, \dots, y_{N+L}^s, y_{N+L}^p]^T$ ，其中 $\mathbf{n}$ 为均值为0，方差为 $\sigma^2$ 的高斯噪声，满足 $\mathbf{y} = \mathbf{c} + \mathbf{n}$ 。在接收端，BP对 $\mathbf{y}$ 预译码后得到混合序列 $\hat{\mathbf{x}}$ ，它将通过第2级译码器MVD进一步纠错后得到最终译码序

列 $\hat{\mathbf{u}}$ 。

### 2.2 伴随式与混合序列

卷积码也可以用校验矩阵 $\mathbf{H}^{[7]}$ 来描述，该矩阵由少量的1和大量的0元素组成，是一个稀疏矩阵。设 $\mathbf{e} = [e_1^s, e_1^p, e_2^s, e_2^p, \dots, e_{N+L}^s, e_{N+L}^p]^T$ 为在传输中产生的错误向量， $\hat{\mathbf{c}} = [\hat{c}_1^s, \hat{c}_1^p, \hat{c}_2^s, \hat{c}_2^p, \dots, \hat{c}_{N+L}^s, \hat{c}_{N+L}^p]^T$ 为BP译码得出的LLRs硬判决出的比特序列，t时刻对应的码符号为 $\hat{c}_t = (\hat{c}_t^s, \hat{c}_t^p)$ ，且有如下关系式成立

$$\hat{\mathbf{c}} = \mathbf{c} \oplus \mathbf{e} \quad (1)$$

$$\mathbf{s} = \mathbf{H}\mathbf{c} = \mathbf{H}(\mathbf{c} \oplus \mathbf{e}) = (\mathbf{H}\mathbf{c}) \oplus (\mathbf{H}\mathbf{e}) \quad (2)$$

而 $\mathbf{H}\mathbf{c} = \mathbf{0}$ ，所以

$$\mathbf{s} = \mathbf{H}\mathbf{e} \quad (3)$$

由比特翻转算法<sup>[8]</sup>的思想

$$\mathbf{f} = \mathbf{H}^T \mathbf{s} \quad (4)$$

其中符号 $\oplus$ 表示模二加， $\mathbf{s}$ 被称作伴随式，只与差错 $\mathbf{e}$ 有关， $\mathbf{f} = [f_1, f_2, \dots, f_N]^T$ 是一个指示器， $\mathbf{f}$ 中第n个元素的值为信息结点n参加校验检测失败的数目，因此 $\mathbf{f}$ 中不为0的元素所在的位置表示序列 $\hat{\mathbf{c}}$ 在相应位置上可能出现误码，这些位置对应的LLRs称为质疑的LLRs，将其它位置对应的LLRs称为可靠的LLRs。需要说明的是，在低信噪比条件下，利用伴随式和比特翻转的思想来区分LLRs可能会将一些错误的码符号误认为正确。然而，即便是认为可靠的码符号，仍需经过网格图的检验，路径不连续时则会回溯译码，使得被认为可靠的错误码符号仍有可能被纠正。

以(1,5/7)码为例(5和7分别表示前馈多项式 $D^2 + 1$ 和反馈多项式 $D^2 + D + 1$ )，设原始的信息序列为 $\hat{\mathbf{u}} = [0 \ 0 \ 1 \ 1 \ 0]^T$ ，加上拖尾的2个比特，编码序列为 $\mathbf{c} = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1]^T$ 。该编码器对应的校验矩阵为

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

若硬判出 $\hat{\mathbf{c}} = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1]^T$ ，由式(1)知 $\mathbf{e} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0]^T$ ，根据式(3)，式(4)分别可得 $\mathbf{s} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]^T$ ， $\mathbf{f} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1]^T$ 。注意到此处 $\mathbf{f}$ 中不为0的位置上并未包含所有的错误位置，下面将介绍扩展质疑符号集的方法。

将对应于 $\mathbf{f}$ 中为1和为0元素的位置的集合分别称为质疑点集合与可靠点集合，记为 $A_s$ 和 $A_r$ 。由于突发错误的影

响，亦即在寄存器限制长度内，同时可能有几个码符号都出错，导致  $\mathbf{f}$  在这些位置上的值为 0。从伴随式的观点来看，得出的  $\mathbf{s}$  实际上只是陪集首，还存在其它错误图样，所以  $A_s$  不能完全反映所有误码的位置，在  $\mathbf{f}$  不为 0 的元素附近，往往有较高概率出现误码。比较直接的做法是扩展集合  $A_s$ ，使得它包含更多可能出错的位置，进一步减小与最优译码的差距，令

$$L_f = D_f L \quad (5)$$

其中  $L_f$  为扩展长度， $D_f$  是一个非负整数，称为扩展系数。设  $A_s$  中的每个元素向周围扩展  $L_f$  个时间段后，组成集合  $B_s$ ，相应的  $A_r$  缩减为集合  $B_r$  ( $A_r$  与  $A_s$  元素与  $B_r$  跟  $B_s$  元素个数的总和均为  $2N$ )。

### 2.3 断点与译码路径

设  $s_{t-1}$  为寄存器在  $t-1$  时刻的状态，则  $s_{t-1}$  与  $\hat{c}_t$  中的信息位  $\hat{c}_i$  就唯一确定了一条分支  $b_t$ 。同时， $\hat{c}_i$  对应于一个分支集合  $B_t$ 。BP 预判出的码符号不一定能构成一条连续的路径，MVD 根据该性质对码符号进行检测。在 MVD 过程中，若  $b_t \notin B_t$ ，则时刻  $t-1$  称为“断点”，否则，将分支  $b_1, b_2, \dots, b_t$  形成的一条连续路径称为“译码路径”。译码路径是在译码过程中逐渐形成的，当出现断点时，它可以通过回溯译码得到更新。

### 2.4 PPVD 和 CPVD

MVD 在对  $\hat{\mathbf{x}}$  的译码过程中，可能会多次出现断点，从而需要多次调用 PVD，为方便后文叙述，需要在调用的先后顺序上区分这些 PVD。称包含当前译码时刻的 PVD 为“当前的 PVD(Current PVD, CPVD)”；CPVD 之前的最近一次 PVD 的过程称为“先前的 PVD(Previous PVD, PPVD)”。此外，若当前时刻出现断点，回溯作 CPVD 可能会与 PPVD 的译码范围产生重合，若它们分别译码，则重叠部分会译码两次，增加了译码复杂度，MVD 通过 CPVD 去掉和 PPVD 译码范围重复的部分，用剩下的部分接着 PPVD 继续译码。

### 2.5 MVD

设  $p_s$  表示译码路径上分支连接的前一时刻所处的状态， $n_s$  表示译码路径末时刻对应的状态。 $P_s$  和  $N_s$  分别表示 PVD 的初始和终止时刻所有可能的状态。图 1 中，长划线表示译码路径，当前译码时刻为  $t_2$  时， $p_s$  和  $n_s$  分别为状态 00 和 10；当前译码时刻为  $t_9$  时， $P_s$  和  $N_s$  分别为 {01} 和 {00,10,01,11}， $n_s$  为状态 01。

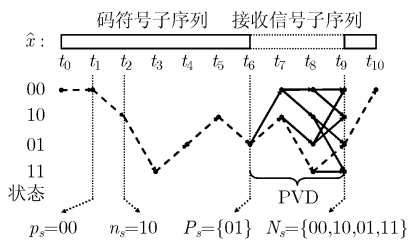


图 1 码符号和接收信号对应于网格图中的初末状态

设  $t_s$  为 CPVD 的初始时刻， $t_n$  和  $t_e$  分别为 PVD(PPVD 或 CPVD)的起始和终止时刻。若当前译码时刻为  $t$ ，且  $t$  为断点，则  $t_s = t - L$ 。此时，在 CPVD 更新译码路径之前， $t_s$  为 CPVD 的初始时刻， $t_e$  为 PPVD 的末时刻，若  $t_s \leq t_e$ ，说明 PPVD 与 CPVD 所涉及的范围有部分重合，用  $\psi = 1$  表示， $t_n$  为 PPVD 的初始时刻。若不重合，则  $t_n = t_s$ ， $\psi = 0$ 。显而易见，在时间顺序上  $t_n \leq t_s$ 。设  $\mathbf{y}_p = \mathbf{y}(t_s \rightarrow t_e)$  为时间段  $t_s \rightarrow t_e$  上的接收符号， $\mathbf{u}_p$  为时间段  $t_n \rightarrow t_e$  上的译码输出。MVD 可以归纳如下：

第 1 步 从零状态出发，即  $p_s$  为零状态，初始化  $\psi = 0$ ， $t = 1$ ，由  $p_s$  和  $c_t^s$  确定出  $n_s$ 。

第 2 步 若  $t > (N + L)$ ，则 MVD 结束。

第 3 步 若  $\hat{\mathbf{x}}$  在  $t$  时刻为码符号，转到第 4 步，否则  $\hat{\mathbf{x}}$  在  $t$  时刻为接收符号，转到第 6 步。

第 4 步 若  $t$  时刻为断点，转到第 5 步，否则  $\hat{u}_t = c_t^s$ ， $p_s = n_s$ ， $t = t + 1$  返回第 2 步。

第 5 步  $t_s = \max(t - L, 1)$ ，如果  $t_s \leq t_e$ ，则  $\psi = 1$ ， $P_s = N_s$ ， $t_s = t_e + 1$ ，将路径寄存器和度量寄存器的初始值设定为  $t_e$  时刻对应的值。否则  $t_n = t_s$ ， $t_e = t$ ， $\mathbf{y}_p = \mathbf{y}(t_s \rightarrow t_e)$ ，将路径寄存器和度量寄存器初始化为零，转到第 7 步。

第 6 步 若  $t_e = t - 1$ ，则  $\psi = 1$ ， $P_s = N_s$ 。否则， $t_n = t$ ，从  $t$  往后搜索  $\hat{\mathbf{x}}$  到  $t_l$ ， $t_e = t_l$ ，使得  $t \rightarrow t_l$  时间段上  $\hat{\mathbf{x}}$  全为接收符号， $\hat{\mathbf{x}}$  在  $t_l + 1$  为码符号，且  $t_l \leq (N + L)$ ， $\mathbf{y}_p = \mathbf{y}(t \rightarrow t_l)$ 。

第 7 步 PVD 对  $\mathbf{y}_p$  译码，通过网格图中每个分支符号与接收符号作相关，并利用加-比-选运算更新度量值，最后将最大度量值对应的路径寄存器中的路径  $\mathbf{u}_p$  存入  $\hat{\mathbf{u}}$  的相应位置上，并令  $p_s = n_s$ ， $\psi = 0$ ， $t = t + 1$  返回第 2 步。

## 3 仿真及复杂度分析

本节根据 2.1 节的仿真环境，分别使用 (1,753/561)，(1,1545/1167) 两种常见的卷积码来比较 HD 与 VD 的性能和复杂度。对于规则  $(N', J, K)$  LDPC 码，采用文献[6]提出的简化 BP 算法译码，只需  $(K - 1)M + 4JN'\eta$  次加法运算。其中， $N'$ ， $J$ ， $K$  分别对应编码后的码长，校验矩阵的列重量和行重量。另外， $\eta$  为 BP 最大迭代次数， $M$  为校验矩阵的行数。此处  $N' = 2(N + L)$ ， $M = N + 2(L - 1)$ ， $K$  为  $\mathbf{H}$  矩阵的行重量，BP 预译码所需的加法次数为  $(N + L) \cdot (4K\eta + (K - 1))$ 。对 VD 而言，网格图中的每 1 时间段内有  $2^{L+1}$  条分支，每个分支作相关运算需 1 次加法，再与前一时刻幸存路径的度量值相加，所以  $2^{L+1}$  条分支总共需  $2^{L+2}$  次加法。VD 所需的总加法数为  $2^{L+2}(N + L)$ 。类似地，HD 所需的总的加法次数为  $(N + L)(4K\eta + (K - 1)) + 2^{L+2}\varphi$ ，其中  $\varphi$  为 MVD 中 PVD 占用的总时间段数，且满足  $\varphi \leq (N + L)$ 。BP 查表以及 VD 比较并选择幸存路径的运算可忽略不计。令  $\rho$  为 VD 所需的复杂度与 HD 所需的平均复杂度之比，它

反映了 HD 相对 VD 运算量的节省

$$\rho = \frac{2^{L+2}(N+L)}{(N+L)(4K\eta + (K-1)) + 2^{L+2}\varphi} \quad (6)$$

此处需要注意到不同  $D_f$  与  $\eta$  对 HD 性能和复杂度的影响。当  $L$  一定时,  $D_f$  越大, 即质疑的码符号集较大, HD 与 VD 的逼近程度越好, 但运算量的节省相对较少。仿真发现, 随着信噪比的提高, 当  $D_f > 3$  时, HD 与最优性能的差距已经很微小, 所以通常可以选择  $D_f \leq 3$  来获得性能与复杂度之间的折衷。同时,  $\eta$  也不能太大, 这会导致 BP 的译码复杂度增加, 使得总的运算量节省减少。该文给出了一组较为合理的参数配置如表 1 所示。在确定了  $D_f$  和  $\eta$  的情况下, 表 1 还比较了两种常用卷积码分别采用 VD 与 HD 的复杂度, 相应的误比特率曲线如图 2 所示, 其中  $N = 500$ 。可以看出当编码寄存器长度一定时, 在中高信噪比条件下, HD 相对 VD 的平均译码复杂度有明显降低, 这是由于 BP 通过少量迭代后, 就已经具有较好的误码性能, 使得需要重新作 PVD 的符号大幅减少。在低信噪比条件下, HD 复杂度略高于 VD 算法, 这可以通过开关措施来解决, 亦即在低信噪比时, 不选择第 1 级 BP 译码, 而直接做 MVD, 输入到 MVD 的混合序列全由接收符号组成。

#### 4 结论

本文提出了一种混合译码算法, 它利用 BP 预译码和 MVD 双重纠错能达到近似最优的性能。对常用的卷积码,

表 1 两种卷积码分别采用 VD 与 HD 所需的译码复杂度之比

$E_b/N_0$	(1,753/561)			(1,1545/1167)		
	$D_f$	$\eta$	$\rho$	$D_f$	$\eta$	$\rho$
1.0	1	1	0.953	1	1	0.974
1.5	1	1	0.967	1	1	0.985
2.0	1	2	1.001	1	5	1.031
2.5	2	3	1.011	2	5	1.073
3.0	2	4	1.200	2	6	1.291
3.5	2	5	1.545	2	7	1.673
4.0	3	5	1.792	—	—	—

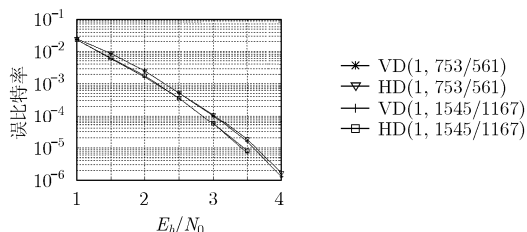


图 2 HD 与 VD 的性能比较

混合算法在中高信噪比条件下, 保证近似最优性能的同时, 复杂度的降低是明显的, (1,753/561) 采用 HD 算法在 4dB 时, 相对 VD 而言至少可节省 44.2% 的运算量。该算法适用于各种卷积码在不同码率下的编译码, 其关键在于确定卷积码的校验矩阵。HD 算法可以灵活的调整参数  $D_f$  和  $\eta$  来取得不同性能和复杂度的折衷。

#### 参考文献

- [1] Forney G D J. The Viterbi algorithm. *Proc[J]. IEEE*, 1973, 61(3): 268-278.
- [2] Benaissa M and Zhu Yi-qun. Reconfigurable hardware architectures for sequential and hybrid decoding[J]. *IEEE Trans. on Circuits and Systems I: Regular Papers*, 2007, 54(3): 555-565.
- [3] Sudhakar R, Mukhtar A, and Gu Zhen-zhong. Low-complexity error selective Viterbi decoder [J]. *Electronics Letters*, 2000, 36(2): 147-148.
- [4] Zadeh S A G and Soleymani M R. An iterative M-algorithm based decoder for convolutional codes [C]. Canadian Conference on Electrical and Computer Engineering, Saskatoon, 2005: 1618-1621.
- [5] Felzenszwalb P F and Huttenlocher D R. Efficient belief propagation for early vision[J]. *Computer Vision and Pattern Recognition*, 2006, 70(1): 261-268.
- [6] 杨帆, 罗振东, 田宝玉. 改进的 LDPC 串行译码[J]. *北京邮电大学学报*, 2008, 31(4): 130-134.
- [7] Yang Fan, Luo Zhen-dong, and Tian Bao-yu. A modified serial decoding algorithm for low-density parity-check codes [J]. *Journal of Beijing University of Posts and Telecommunications*, 2008, 31(4): 130-134.
- [8] Zhu L X, Wang J F, and Yang S Z. Iterative decoding of turbo codes based on factor graphs[C]. IEEE 2002 international conference on Communications, Circuits and Systems and West Sino Expositions (ICCCAS&WeSino Exppo'02), Chengdu, China, 2002, vol.1: 46-50.
- [9] Wu Xiao-fu, Ling Cong, and Jiang Ming, et al.. Towards understanding weighted bit-flipping decoding[C]. IEEE International Symposium on Information Theory' 07, Nice, France, 2007: 1666-1670.

杨帆: 男, 1982 年生, 博士生, 研究方向为移动通信的理论和实现。

罗振东: 男, 1975 年生, 博士, 研究方向为移动通信的理论和实现。

田宝玉: 男, 1946 年生, 教授, 研究方向为信息处理、移动通信。