

基于可信位置排序的咬尾卷积码译码算法

王晓涛* 刘振华

(中国电子科技集团公司第三十八研究所 合肥 230088)

摘要: 咬尾卷积码的传统译码算法没有考虑咬尾格形图的循环性,译码起始位置固定,译码效率相对较低。该文首次证明了咬尾卷积码基于格形图的译码算法与译码起始位置无关,即从任意位置开始译码得到的最优咬尾路径即为全局最优咬尾路径。基于此提出一种基于可信位置排序的咬尾卷积码译码算法。新算法利用咬尾格形图的循环性,根据接收到的信道输出序列估算每个译码起始位置的可靠性,从而选择一个可靠性最高的译码起始位置。和传统译码算法相比,所提算法具有更快的收敛速度。

关键词: 咬尾码; 咬尾格形图; 循环性; 最大似然译码

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2015)07-1575-05

DOI: 10.11999/JEIT141459

Belief Ranking Based Low-complexity Maximum Likelihood Decoding Algorithm for Tail-biting Convolutional Codes

Wang Xiao-tao Liu Zhen-hua

(China Electronics Technology Group Corporation No.38 Research Institute, Hefei 230088, China)

Abstract: For a long time, the circularity of the tail-biting trellis is ignored in conventional decoding algorithms of Tail-Biting Convolutional Codes (TBCC). This kind of algorithm starts decoding from the fixed location, and consequently exhibits relatively lower decoding efficiency. For the first time, this paper proves that the decoding result of the tail-biting convolutional codes is independent on the decoding starting location. It means that the Maximum Likelihood (ML) tail-biting path, which starts from any location of the tail-biting trellises, is the global ML tail-biting path. Based on this observation, a new ML decoding algorithm is proposed. The new algorithm ranks the belief-value of each location on the trellis at first, and then selects the location with the highest belief-value as the decoding starting location. Compared with other existing ML decoders, the new decoder exhibits higher convergence speed.

Key words: Tail-biting codes; Tail-biting trellis; Circularity; Maximum Likelihood (ML) decoding

1 引言

卷积码在编码时其编码器可以用已知比特来初始化,也可以用信息比特来初始化。对于寄存器长度为 v 的卷积码编码器,当采用信息比特的最后 v 位来初始化编码器时,编码结束时编码器的状态和其初始状态是相同的,同时码字对应的格形图呈咬尾状态^[1,2]。我们称这种方式得到的码字为咬尾卷积码(Tail-Biting Convolutional Codes, TBCC),其对应的格形图为咬尾格形图。

采用咬尾方式编码可以消除用已知比特初始化编码器带来的码率损失^[3-5],这种损失对于短码来说显得更为严重,比如对于生成多项式为{354, 237}(8进制),信息序列长度为32的卷积码来说,如果

不采用咬尾编码其有效码率损失会达到20%。因此咬尾卷积码作为一种高效的短码编码方式被广泛用于各种通信系统中作为控制信道和广播信道的编码方案,如在增强型数据速率 GSM 演进技术(Enhanced Data rate for GSM Evolution, EDGE)^[6]和 3GPP 长期演进项目(Long Term Evolution, LTE)^[7]等通信系统。

由于咬尾卷积码的编码器采用信息序列进行初始化,因此其对应的咬尾格形图在起始位置有 2^v 个起始状态,这种结构给译码器的设计带来了困难^[8-12]。目前咬尾卷积码的译码算法主要有3大类:基于循环 Viterbi 算法(Circular Viterbi Algorithm, CVA)的译码算法,如 WAVA 和 TD-CVA 等^[1,13];基于 Viterbi 算法和启发式搜索的混合式译码(Viterbi-Heuristic, VH)^[14];和基于双向搜索的有效译码算法(Bidirectional Efficient Algorithm for

Searching code Trees, BEAST)^[15]。BEAST 算法工作时需要将咬尾格形图转化成传统格形图，且需要保存整个格形图的结构信息，这将消耗大量的存储空间。比如对于(24,12)的 Golay 码来说，其传统格形图的中间位置处的状态数高达 512 个^[15]，而其具有最低状态复杂度的咬尾格形图在每个位置处的状态数只有 16 个^[11]。VH 算法分为两个译码阶段，在第 1 阶段 Viterbi 算法将咬尾格形图上所有位置处到达每个状态的幸存路径的度量值保存下来，利用这些信息进行启发式搜索。VH 算法的两个译码步骤基于两种完全不同的搜索算法，不仅需要保存大量的中间信息，同时译码器实现复杂度高，不利于实际应用。

CVA 是一种完全基于 Viterbi 的译码算法，通过在咬尾格形图多次迭代来寻找最优咬尾路径及其对应的码字。传统的基于 CVA 的译码算法没有考虑译码过程中存在的循环陷阱，这导致译码器在译码结束时可能无法收敛到最优咬尾路径，因此是一种次优译码算法，如 WAVA 和 ET-CVA 等。TD-CVA 从循环陷阱出发，通过在迭代过程中对非最大似然状态的排除，使得译码器能最终收敛到全局最优咬尾路径。

本文利用咬尾格形图的循环性，提出了一种基于 CVA 的最大似然译码算法。新算法首先根据接收到的似然比信息来确定可靠性最高的起始位置，然后从该起始位置处开始译码。这样经过几个分支的搜索以后，幸存路径的起始状态会收敛到更少的幸存状态上。相对于已有的算法，本文算法具有更快的收敛速度，译码效率得到进一步提高。

本文结构如下，第 2 节介绍新算法的设计原理，并通过例子给出算法说明，同时对算法的最优性进行证明；第 3 节给出算法性能仿真验证；第 4 节总结全文。

2 算法描述

2.1 变量定义

如图 1 所示的咬尾格形图 T 是生成多项式为 $\{7,5\}$ 、信息序列长度为 $L=8$ 的 TBCC 对应的咬尾格形图。格形图 T 中位置 l 处的状态集合记为 S_l ， $0 \leq l \leq L$ ($S_0 = S_L$)。对于咬尾卷积码来说其咬尾格形图是规则的^[12]，即有 $S_i = S_j$ ， $0 \leq i, j \leq L$ 。格形图 T 中任意位置 l 处的状图 s 都有其对应的咬尾格形子图，记为 $T_{\text{sub}}(s)$ 。图 1 中实线所示为位置 $l=2$ 处的状态 $s=01$ 的咬尾格形子图。 $T_{\text{sub}}(s)$ 上所有完整路径都是咬尾路径，整个咬尾格形图 T 是由位置 l 处所有状态的咬尾格形图的并集组成^[12]，即有

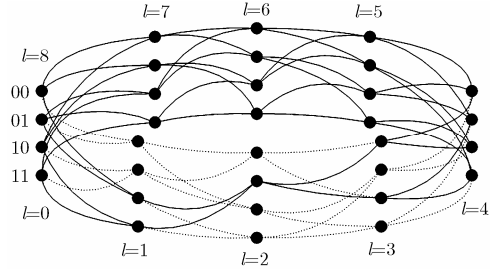


图 1 生成多项式为 $\{7,5\}$ 的咬尾卷积码格形图

$$T = \bigcup_{\forall s \in S_l} T_{\text{sub}}(s) \quad (1)$$

在译码开始的时候，对接收到的软信息进行处理以选择可靠性最高的位置 l_{opt} 作为译码的起始位置。设信息序列长度为 L 的 TBCC 的码率为 b/c ，编码以后的码字为 $v_l^{(j)} \in \{0,1\}$ ，其中 $0 \leq l \leq L$ ， $0 \leq j \leq c-1$ 。设码字经过 BPSK 调制得到映射后的符号为： $x_l^{(j)} = (1 - 2v_l^{(j)})\sqrt{E_s}$ ，不失一般性令 E_s 为 1。经过双边噪声功率谱密度为 $N_0/2$ 的高斯白噪声 (Additive White Gaussian Noise, AWGN) 信道后，对应接收到的符号为 $r_l^{(j)}$ ，计算得到似然比信息 $\lambda_l^{(j)} = 4r_l^{(j)}/N_0$ 。定义 l_{opt} 的计算方式为

$$\begin{aligned} l_{\text{opt}} &= \arg \max_l \sum_{q=l}^{(l+Q)_L} \sum_{j=0}^{c-1} \text{abs}(\lambda_l^{(j)}) \\ &= \arg \max_l \sum_{q=l}^{(l+Q)_L} \sum_{j=0}^{c-1} \text{abs}(4r_l^{(j)}/N_0) \\ &= \arg \max_l \sum_{q=l}^{(l+Q)_L} \sum_{j=0}^{c-1} \text{abs}(r_l^{(j)}), \quad 0 \leq l < L \end{aligned} \quad (2)$$

式中 $(l+Q)_L = (l+Q) \bmod L$ ，其中 Q 是待确定的量，在具体应用中将根据不同的码字选择合适的值。

2.2 算法说明

由于 CVA 的译码过程是迭代的，即在第 i 次迭代中每个状态的初始度量值是用第 $i-1$ 次迭代得到的幸存路径的度量值来初始化的，因此每次迭代都会得到不同的幸存路径及其路径度量值^[1]。第 i 次迭代中，起始于状态 s ，结束于状态 s' 的幸存路径记为 $P^i(s, s')$ ，其中 $s, s' \in S_{l_{\text{opt}}}$ ， $P^i(s, s')$ 对应的路径累积度量值记为 $M^i(s, s')$ 。如果咬尾格形图 T 上的幸存路径的起始状态和终止状态相同，即 $s = s' (\in S_{l_{\text{opt}}})$ ，则路径 $P^i(s, s)$ 为咬尾路径。 $T_{\text{sub}}(s)$ 是由所有从状态 s 起始并结束到状态 s 的咬尾路径的集合构成，记 $T_{\text{sub}}(s)$ 上对应于接收序列度量值最大的咬尾路径为 $P_{l_{\text{opt}}}^o(s, s)$ ：

$$P_{l_{\text{opt}}}^o(s, s) = \max_{M^i(s, s)} P^i(s, s) \quad (3)$$

其中 $s \in S_{l_{\text{opt}}}$ ， $i \geq 1$ 。

对于从位置 l_{opt} 开始的译码过程，每个 $s \in S_{l_{\text{opt}}}$ 的 $T_{\text{sub}}(s)$ 上都有一条 $P_{l_{\text{opt}}}^o(s, s)$ ，所有 $P_{l_{\text{opt}}}^o(s, s)$ 中具有最大路径度量值的记为 $P_{l_{\text{opt}}}^o$ ， $P_{l_{\text{opt}}}^o = \max_{M_{l_{\text{opt}}}^o(s, s)} P_{l_{\text{opt}}}^o(s, s)$ ， $s \in S_{l_{\text{opt}}}$ 。最大似然译码算法需要寻找到咬尾格形图上的全局最优咬尾路径，记为 P^o ，在后面将证明 P^o 和 $P_{l_{\text{opt}}}^o$ 为同一条路径。每次迭代结束以后，将 $S_{l_{\text{opt}}}$ 中满足 $P_{l_{\text{opt}}}^o(s, s) \leq P_{l_{\text{opt}}}^o$ 的状态删除掉，可以得到幸存状态集合 $S_{l_{\text{opt}}}^{\dagger}$ ，最终 $S_{l_{\text{opt}}}^{\dagger}$ 会变成空集，算法收敛到 P^o 。

算法的步骤描述如下：

步骤 1 根据式(2)计算最佳译码起始位置 l_{opt} ，初始化迭代次数 $i = 0$ ，令 $S_{l_{\text{opt}}}^{\dagger} = S_{l_{\text{opt}}}$ ；

步骤 2 $i = i + 1$ ，若 $S_{l_{\text{opt}}}^{\dagger} = \emptyset$ ，则执行步骤 4，否则以集合 $S_{l_{\text{opt}}}^{\dagger}$ 中的状态为起始状态开始执行 CVA 算法的第 i 次迭代；

步骤 3 第 i 次迭代结束以后，若搜寻到比 $P_{l_{\text{opt}}}^o$ 度量值更大的咬尾路径，则更新 $P_{l_{\text{opt}}}^o$ ，并判断是否有循环陷阱产生：

(a) 有循环陷阱：找到具有最大路径度量值 $M^i(s, s')$ 的幸存路径 $P^i(s, s')$ ，在 $T_{\text{sub}}(s)$ 上执行一次 Viterbi 算法得到 $P_{l_{\text{opt}}}^o(s, s)$ ，将 s 从 $S_{l_{\text{opt}}}^{\dagger}$ 中删除；同时若 $P_{l_{\text{opt}}}^o(s, s) > P_{l_{\text{opt}}}^o$ ，更新 $P_{l_{\text{opt}}}^o$ ；返回步骤 2。

(b) 无循环陷阱：返回步骤 2。

步骤 4 令 $P^o = P_{l_{\text{opt}}}^o$ ，输出 P^o 作为最终译码结果，译码结束。

2.3 最优性证明

由文献[1]可知，通过对循环陷阱的控制，译码算法最终会收敛到 $P_{l_{\text{opt}}}^o$ ，而 $P_{l_{\text{opt}}}^o$ 是译码器从位置 l_{opt} 处开始译码得到的最大似然咬尾路径，因此证明算法的最优性就是证明译码结果与起始位置无关。

定理 1 译码算法在咬尾格形图上得到的最大似然咬尾路径与译码起始位置相独立，即有 $P^o = P_{l_{\text{opt}}}^o$ 。

证明 利用反证法来证明这个结论。设译码器从位置 l_{opt} 开始译码，得到最优咬尾路径为 $P_{l_{\text{opt}}}^o$ ，其路径度量值为 $M_{l_{\text{opt}}}^o$ ，若 $P_{l_{\text{opt}}}^o$ 不是全局最优咬尾路径，则在咬尾格形图上必有一条咬尾路径 $P(s, s)$ ，其路径度量值为 $M(s, s)$ ，其中 $s \in S_{l'}$ ， $l' \neq l_{\text{opt}}$ ，满足

$$M(s, s) > M_{l_{\text{opt}}}^o \quad (4)$$

因为 $P(s, s)$ 是咬尾路径，由咬尾路径定义可知

该路径在咬尾格形图上起始和结束于同一个状态，且任意位置 l 处的状态集合 S_l 中有且仅有一个状态在咬尾路径上(非咬尾路径在起始位置处不满足此条件)。设 $P(s, s)$ 经过 $S_{l_{\text{opt}}}$ 中的状态 s' ，则有

$$P(s, s) \in T_{\text{sub}}(s') \quad (5)$$

可见 $P(s, s)$ 是咬尾格形子图 $T_{\text{sub}}(s')$ 上的咬尾路径，设 $T_{\text{sub}}(s')$ 上的最优咬尾路径为 $P_{l_{\text{opt}}}^o(s', s')$ ，且其路径度量值为 $M_{l_{\text{opt}}}^o(s', s')$ ，从而有

$$M(s, s) \leq M_{l_{\text{opt}}}^o(s', s') \quad (6)$$

由译码算法步骤 3 可知，译码结束时 $P_{l_{\text{opt}}}^o$ 中保存的是所有咬尾格形子图 $T_{\text{sub}}(s)$ 中度量值最大的咬尾路径，其中 $\forall s \in S_{l_{\text{opt}}}$ ，因此有

$$M_{l_{\text{opt}}}^o(s', s') \leq M_{l_{\text{opt}}}^o \quad (7)$$

由式(6)，式(7)两式可知：

$$M(s, s) \leq M_{l_{\text{opt}}}^o \quad (8)$$

可见式(8)与假设条件式(4)相矛盾，即译码结束时咬尾格形图上不存在任何咬尾路径其度量值大于 $M_{l_{\text{opt}}}^o$ ，译码过程与译码起始位置无关， $P_{l_{\text{opt}}}^o$ 中保存的即为全局最优咬尾路径，所以有 $P^o = P_{l_{\text{opt}}}^o$ 。证毕

3 仿真实验

本节通过两个实验来验证本文算法的有效性，为引用方便，将本文算法记为 NCVA(New CVA)。此处选用两种经典算法进行性能比较，WAVA 算法和 VH 算法，这是因为 WAVA 是咬尾格形图上基于 CVA 的经典译码算法，而 VH 算法是咬尾格形图上基于启发式搜索的经典算法。编码以后的比特通过 QPSK 映射后经过 AWGN 信道，接收端接收到信息序列以后分别采用上述 3 种译码算法进行译码，译码过程中每个译码块的平均访问状态数作为衡量译码复杂度的标准。

在第 1 个实验中，我们选用 LTE 中用于控制信道编码的咬尾卷积码码字(120, 40)进行实验，该咬尾卷积码生成多项式为(133, 171, 165)(8 进制)，编码器寄存器长度为 $v = 6$ 。仿真结果如图 2 所示。将 WAVA 性能和 NCVA 性能比较可见：由于循环陷阱控制的添加，以及选择可信度更高的译码起始位置，能够极大的提高译码效率；VH 算法的性能和 NCVA 的性能比较可以发现，在信噪比从低到高变化的范围内 NCVA 都可以达到更高的译码效率。随着信噪比的升高，NCVA 从任何地方开始译码都可以得到相同的译码复杂度，即执行一次迭代即可得到最优解；同时 VH 算法也只需要执行一次 Viterbi

算法即可得到最优解。因此, NCVA 和 VH 算法的译码复杂度最终收敛到一起, 为 $2^v L = 2560$, 这和仿真结果相吻合。

表 1 中给出了几种不同译码算法的误码率性能。由于 VH 算法, TD-CVA, NCVA 是最优译码算法, 因此它们有相同的误码率性能, 而 WAVA 是次优译码算法, 因此 WAVA 算法的性能要略差于 NCVA 算法。

表 1 不同译码算法对(120,40)咬尾卷积码译码的误码率性能

算法	$E_b/N_0(\text{dB})$				
	1.0	2.0	3.0	4.0	5.0
WAVA	1.39e-1	5.19e-2	1.33e-2	2.04e-3	1.63e-4
TD-CVA/ VH/NCVA	1.34e-1	5.00e-2	1.28e-2	1.96e-3	1.60e-4

和 TD-CVA^[1]相比, 本文算法利用了可信位置排序来改进算法的收敛速度。为了验证可信位置排序对算法收敛速度的改进, 表 2 中给出 NCVA 在不同 Q 值下的译码复杂度和 TD-CVA 算法的译码复杂度详细对比列表。通过对比可以发现以下两点规律:

(1) 由于可信位置的选择, NCVA 可以比 TD-CVA 获得更快的收敛速度;

表 2 NCVA 与 TD-CVA 算法复杂度对比

算法	$E_b/N_0(\text{dB})$				
	1.0	2.0	3.0	4.0	5.0
NCVA ($Q=1$)	3835	3108	2767	2646	2607
NCVA ($Q=2$)	3834	3086	2770	2653	2592
NCVA ($Q=3$)	3812	3086	2772	2654	2593
NCVA ($Q=4$)	3839	3096	2796	2660	2598
NCVA ($Q=5$)	3856	3123	2801	2671	2598
TD-CVA	4025	3246	2861	2701	2612

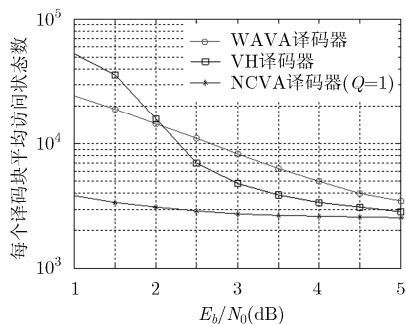


图 2 不同译码算法对(120,40)咬尾卷积码的译码性能比较

(2) 当 Q 值越小时, 可靠度的作用越明显; 由于接收序列服从正态分布, 因此当 Q 值越大时, 由式 (2) 计算得到的每个位置的可靠度越接近; 极限情况是 Q 为接收序列的长度, 此时每个位置的可靠度完全一样, 那么译码从头开始, 复杂度与 TD-CVA 相同。

综合以上考虑, 实际中选取 $Q \leq 2$ 。

在第 2 个实验中, 我们选用(24,12)Golay 码为例。Golay 码的 16 状态咬尾格形图状态复杂度低, 但是该咬尾格形图是周期为 4 的咬尾格形图, 整个格形图分为 3 段, 长度为 12。这种格形图在每个时刻的路径输出不同, 不利于译码器的实现, 本文选用其 64 状态咬尾格形图表示, 该格形图可以由生成多项式 $\{103, 166\}$ 得到^[1]。图 3 中给出了不同译码算法对(24,12)Golay 码译码复杂度比较, 可以得到和第 1 个实验相同的结论。

表 3 中给出了几种算法的误块率性能, 可见在信噪比较高, 即 $E_b/N_0 \geq 3$ dB 时, 次优译码算法 WAVA 的性能和 NCVA 性能接近。

表 3 不同译码算法对(24,12)Golay 码译码的误码率性能

算法	$E_b/N_0(\text{dB})$				
	1.0	2.0	3.0	4.0	5.0
WAVA	8.54e-2	1.37e-2	1.35e-3	7.28e-5	1.96e-6
TD-CVA/ VH/NCVA	8.47e-2	1.36e-2	1.35e-3	7.28e-5	1.96e-6

4 结束语

咬尾格形图具有循环性, 在咬尾格形图上的译码过程可以从任意位置开始, 最终得到的最大似然译码结果都是相同的。文中以定理的形式证明了这个结论, 从任何位置开始译码得到的最优咬尾路径都是全局最优咬尾路径。本文中提出的译码器利用咬尾格形图的循环性, 用接收到的信道输出序列先

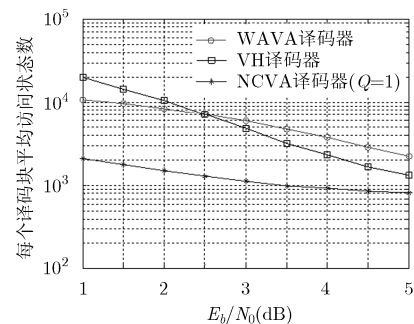


图 3 不同译码算法对(24,12)Golay 码译码性能比较

计算一个可信度最高的译码起始位置。从该位置开始译码可以发现译码器能够更快地收敛到最优解，最后的仿真结果也证明了算法有效性。

参考文献

- [1] Wang Xiao-tao, Qian Hua, Xiang Wei-dong, *et al.* An efficient ML decoder for tail-biting codes based on circular trap detection[J]. *IEEE Transactions on Communications*, 2013, 61(4): 1212-1221.
 - [2] Gluesing-Luerssen H and Forney G D. Local irreducibility of tail-biting trellises[J]. *IEEE Transactions on Information Theory*, 2013, 59(10): 6597-6610.
 - [3] 王晓涛, 钱骅, 徐景, 等. 基于陷阱检测的咬尾卷积码译码算法[J]. 电子与信息学报, 2011, 33(10): 2300-2305.
Wang Xiao-tao, Qian Hua, Xu Jing, *et al.* Trap detection based decoding algorithm for tail-biting convolutional codes [J]. *Journal of Electronics & Information Technology*, 2011, 33(10): 2300-2305.
 - [4] 王晓涛, 钱骅, 康凯. 基于 Viterbi-双向搜索的咬尾码最大似然译码算法[J]. 电子与信息学报, 2013, 35(5): 1017-1022.
Wang X T, Qian H, and Kang K. Viterbi-bidirectional searching based ML decoding algorithm for tail-biting codes [J]. *Journal of Electronics & Information Technology*, 2013, 35(5): 1017-1022.
 - [5] Wu T Y, Chen P N, Pai H T, *et al.* Reliability-based decoding for convolutional tail-biting codes[C]. IEEE Vehicular Technology Conference, Taipei, 2010: 1-4.
 - [6] 3GPP TS. 45.003-3rd generation partnership project; technical specification group GSM/EDGE radio access network; channel coding (release 9)[S]. 2009.
 - [7] 3GPP TS. 36.212-3rd generation partnership project; technical specification group radio access network; evolved universal terrestrial radio access (E-UTRA); multiplexing and channel coding (release 8)[S]. 2009.
 - [8] Williamson A R, Marshall M J, and Wesel R D. Reliability-output decoding of tail-biting convolutional codes [J]. *IEEE Transactions on Communications*, 2014, 62(6): 1768-1778.
 - [9] Bin Khalid F, Masud S, and Uppal M. Design and implementation of an ML decoder for tail-biting convolutional codes[C]. IEEE International Symposium on Circuits and Systems, Beijing, 2013: 285-288.
 - [10] Zhu L, Jiang M, and Wu C. An improved decoding of tail-biting convolutional codes for LTE systems[C]. 2013 International Conference on Wireless Communications & Signal Processing, Hangzhou, 2013: 1-4.
 - [11] Calderbank A, Forney G Jr, and Vardy A. Minimal tail-biting trellises: the Golay code and more[J]. *IEEE Transactions on Information Theory*, 1999, 45(5): 1435-1455.
 - [12] Wang X T, Qian H, Kang K, *et al.* A low-complexity maximum likelihood decoder for tail-biting trellis[J]. *EURASIP Journal on Wireless Communications and Networking*, 2013, 130(1): 1-11.
 - [13] Shao R Y, Lin S, and Fossorier M P C. Two decoding algorithms for tailbiting codes[J]. *IEEE Transactions on Communications*, 2003, 51(10): 1658-1665.
 - [14] Pai H T, Han Y, Wu T, *et al.* Low-complexity ML decoding for convolutional tail-biting codes[J]. *IEEE Communications Letters*, 2008, 12(12): 883-885.
 - [15] Bocharova I, Johannesson R, Kudryashov B, *et al.* BEAST decoding for block codes[J]. *European Transactions on Telecommunications*, 2004, 15(1): 297-305.
- 王晓涛：男，1985年生，博士，工程师，研究方向为卫星数据传输关键技术研究、信道编译码理论研究。
- 刘振华：男，1983年生，博士，工程师，研究方向为卫星通信自适应信号处理理论。