A Class of Binary Rate One-Half Convolutional Codes that Allows an Improved Stack Decoder

ADRIANUS J. VINCK, MEMBER, IEEE, A. J. P. DE PAEPE, AND J. PIETER M. SCHALKWIJK, SENIOR MEMBER, IEEE

***Abstract*—A class of binary rate one-half convolutional codes that allows a modified stack decoder are introduced. The distribution of the number of computations per decoded frame is greatly improved. Simulations indicate that for a binary symmetric channel with transition probability $p=2^{-5}$ the modified stack decoder requires less than one-fifth the stack size of the classical implementation.**

I. Introduction

THIS INTRODUCTORY section applies to general binary rate k/n convolutional codes. In Section II we specialize to rate one-half to describe a class of codes that allows an interesting modification of the stack decoding algorithm [1]–[2]. Section III presents our simulation results.

Let G be a conventional convolutional encoder [3] whose rows are the first k rows of a polynomial $n\times n$ matrix B with polynomial inverse B^{-1}. The generator matrix G has an instantaneous polynomial inverse G^{-1} and a syndrome former H^T consisting of the first k columns and the last $n-k$ columns of B^{-1}, respectively. The last $n-k$ rows of B are the transpose of the inverse of H. Summarizing,

$$B=\begin{bmatrix} G \\ H^{-1^T}\end{bmatrix}, \qquad B^{-1}=\left[G^{-1},H^T\right].$$

Let $\boldsymbol{m}$, $\boldsymbol{n}$, and $\boldsymbol{y}$ be respectively the message vector, the channel noise vector, and the received data vector. The components of these vectors are a formal power series in the delay operator D. Given $\boldsymbol{y}$ and given an estimate $\hat{\boldsymbol{n}}$ of the channel noise one can form an estimate $\hat{\boldsymbol{m}}$ of the message vector as follows:

$$\hat{\boldsymbol{m}}=(\boldsymbol{y}+\hat{\boldsymbol{n}})G^{-1}=(\boldsymbol{m}G+\boldsymbol{n})G^{-1}+\hat{\boldsymbol{n}}G^{-1}=\boldsymbol{m}+\boldsymbol{e}+\hat{\boldsymbol{e}}.$$

As in [4], given the syndrome vector

$$\boldsymbol{z}=\boldsymbol{y}H^T=(\boldsymbol{m}G+\boldsymbol{n})H^T=\boldsymbol{n}H^T,$$

we directly form an estimate $\hat{\boldsymbol{e}}$ of the message error vector $\boldsymbol{e}=\boldsymbol{n}G^{-1}$. Note that

$$\hat{\boldsymbol{n}}=\hat{\boldsymbol{n}}B^{-1}B=(\hat{\boldsymbol{n}}G^{-1},\hat{\boldsymbol{n}}H^T)B=(\hat{\boldsymbol{e}},\boldsymbol{z})\begin{bmatrix} G \\ H^{-1^T}\end{bmatrix}=\hat{\boldsymbol{e}}G+\boldsymbol{z}H^{-1^T}. \tag{1}$$

Therefore with maximum likelihood (ML) decoding the estimator should find that codeword $\hat{\boldsymbol{e}}G$ closest to $\boldsymbol{z}H^{-1^T}$, since the resulting message error estimate $\hat{\boldsymbol{e}}$ corresponds to a noise estimate $\hat{\boldsymbol{n}}$ of minimum Hamming weight.

II. Special $R=1/2$ Codes—Computational Savings

In [4] we introduced a class $\Gamma_{n,\nu,l}$ of binary rate $(n-1)/n$ convolutional codes that allowed a Viterbi decoder [5] of reduced complexity. Similarly one can [6] define a class $L_{n,\nu,l}$ of rate $1/n$ convolutional codes that allows a stack decoder [1], [2] of reduced complexity. In this paper we study the most important of these codes, the class $L_{2,\nu,l}$ of binary rate one-half codes $G=[g_1,g_2]$ where $g_1\neq g_2$ and

$$g_{1,\nu}=1, \tag{2a}$$

$$g_{1,j}=g_{2,j}, \qquad \text{for } 0\leqslant j\leqslant l-1, \tag{2b}$$

$$\gcd(g_1,g_2)=1. \tag{2c}$$

Note that because of condition (2b) these codes have a bad distance profile [7] and hence it is somewhat surprising that they perform so well in conjunction with sequential decoding. If condition (2c) is satisfied it follows from the invariant factor theorem [3] that the code $G=[g_1,g_2]$ is noncatastrophic.

We now explain how the symmetries of $L_{2,\nu,l}$ can be used to advantage in stack decoding. Let

$$\hat{\boldsymbol{e}}(t-l+1;t)=\sum_{i=t-l+1}^{t}\hat{\boldsymbol{e}}_iD^i, \qquad t=\cdots,-1,0,+1,\cdots,$$

represent the last l elements, up to and including the present, of a message error sequence estimate $\hat{\boldsymbol{e}}(-\infty;t)$. Given the syndrome sequence $\boldsymbol{z}(-\infty;t)$ let $\hat{\boldsymbol{n}}(-\infty;t)$ be the corresponding estimate of the noise vector sequence (see (1)). Now consider a vector sequence

$$\tilde{\boldsymbol{n}}(t-l+1;t)=\sum_{i=t-l+1}^{t}\tilde{\boldsymbol{n}}_iD^i, \qquad t=\cdots,-1,0,+1,\cdots,$$

where

$$\tilde{\boldsymbol{n}}_i\in\{(0,0),(1,1)\}, \qquad \text{for all } i.$$

Manuscript received November 28, 1978. This work was supported by the Netherlands Organization for the Advancement of Pure Research (Z.W.O.). This paper was presented at the International Symposium on Information Theory, Grignano, Italy, June 25–29, 1979.

A. J. Vinck and J. P. M. Schalkwijk are with the Department of Electrical Engineering, Eindhoven University of Technology, Den Dolech 2, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

A. J. P. de Paepe was with the Information and Communication Theory Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands. He is now with the Data Processing Division, Rijkswaterstaat, Rijswijk, The Netherlands.

0018-9448/80/0700-0389$00.75

TABLE I
STACK CONTENTS AFTER REORDERING

Information : 1 0 1 0 0		Encoder : $G = (1+D+D^2, 1+D^2)$	
Transmitted : 11 10 00 10 11		$z(H^{-1})^T$: 10 11 00 10 01	
Received : 01 10 01 10 11		nG^{-1} : 0 1 1 1 0	
time:	Classical stack algorithm Stack content after reordening	time:	Modified stack algorithm Stack content after reordening
1	0,-9 ; 1,-9 ;	1	1(1),-9
2	1,-9 ; 00,-18 ; 01,-18	2	01(0),-7 ; 10(1),-18
3	10,-7 ; 00,-18 ; 01,-18 ; 11,-29	3	011(1),-16 ; 10(1),-18 ; 000(0),-27 ;
4	100,-16 ; 101,-16 ; 00,-18 ; 01,-18 11,-29	4	0111(0),-14 ; 10(1),-18 ; 0100(1),-25 ; 000(0),-27
5	101,-16 ; 00,-18 ; 01,-18 ; 1000,-25 1001,-25 ; 11,-29	5	01110(0),-12 ; 10(1),-18 ; 0100(1),-25 ; 000(0),-27 ; 01101(1),-45 ;
6	1010,-14 ; 00,-18 ; 01,-18 ; 1000,-25 1001,-25 ; 11,-29 ; 1011,-36		
7	10100,-12 ; 00,-18 ; 01,-18 ; 1000,-25 1001,-25 ; 11,-29 ; 10101,-34 ; 1011,-36		

There are 2^l such vector sequences $\tilde{n}(t-l+1;t)$. Further note that because of (2b) one can always find an $\tilde{e}(t-l+1;t)$ such that $\tilde{n}(t-l+1;t)=\tilde{e}(t-l+1;t)G$, i.e., $\tilde{e}(t-l+1;t)=\tilde{n}(t-l+1;t)G^{-1}$. Now, given our original message error sequence estimate $\hat{e}(-\infty;t)$ one can find 2^l-1 new estimates $\hat{e}'(-\infty;t)=\hat{e}(-\infty;t)+\tilde{e}(t-l+1;t)$, one for each nonzero value of $\tilde{n}(t-l+1;t)$. If we define the norm $f[\hat{e}(-\infty;t)]$ to equal the Fano norm [8] of the corresponding noise sequence $\hat{n}(-\infty;t)=\hat{e}(-\infty;t)G+z(-\infty;t)H^{-1^T}$ (this norm is finite since we assume that all sequences start at some finite time in the past), then the norm $f[\hat{e}'(-\infty;t)]$ of $\hat{e}'(-\infty;t)=\hat{e}(-\infty;t)+\tilde{e}(t-l+1;t)$ is given by

$$f[\hat{e}'(-\infty;t)]=f[\hat{e}(-\infty;t)]+\sum_{i=t-l+1}^{t}\Delta[\tilde{n}_i;\hat{n}_i], \quad (3)$$

where

$$\Delta[x;y]=\begin{cases}-2\log\dfrac{1-p}{p}, & \text{if } x=(1,1), \text{ and } y=(0,0)\\ +2\log\dfrac{1-p}{p}, & \text{if } x=(1,1), \text{ and } y=(1,1)\\ 0, & \text{otherwise.}\end{cases}$$

If $g_{1,l}\neq g_{2,l}$, a class of 2^l message error sequence estimates $\hat{e}(-\infty;t)$ upon extension gives rise to two new classes of estimates $\hat{e}(-\infty;t+1)$, each of size 2^l. If $g_{1,l}=1$, then one new class contains the extensions $\hat{e}(-\infty;t+1)$ of those $\hat{e}(-\infty;t)$ for which $\hat{n}_{t-l+1}=(0,\cdot)$, and the other new class contains the extensions of those estimates that have $\hat{n}_{t-l+1}=(1,\cdot)$. If $g_{2,l}=1$, the old class of estimates $\hat{e}(-\infty;t)$ splits into two parts of 2^{l-1} estimates each, according to whether $\hat{n}_{t-l+1}=(\cdot,0)$ or $(\cdot,1)$. On extension, each of these sets of 2^{l-1} estimates $\hat{e}(-\infty;t)$ gives rise to a complete image class of 2^l estimates $\hat{e}(-\infty;t+1)$. With each class of 2^l message error sequence estimates we can associate a representative member. A whole parent class of 2^l estimates can now be extended into two image classes by finding the representative and its associated metric for each image class given the representative and its metric for the parent class. We arbitrarily select as the class representative that estimate $\hat{e}(-\infty;t)$ for which the corresponding $\hat{n}=\hat{e}G+zH^{-1^T}$ satisfies $\hat{n}_i\in\{(0,0),(0,1)\}$, $t-l+1\leqslant i\leqslant t$. Note that the class representative $\hat{e}(-\infty;t)$ has the maximum norm within the class. Let $\hat{n}(-\infty;t)=\hat{e}(-\infty;t)G+z(-\infty;t)H^{-1^T}$ be the noise vector sequence estimate that corresponds to the representative $\hat{e}(-\infty;t)$ of the parent class. Then for one of the image classes the representative $\hat{e}'(-\infty;t+1)$ has an associated noise sequence $\hat{n}'(-\infty;t+1)$ that coincides over $(-\infty,t]$ with $\hat{n}(-\infty;t)$. The other images class, however, has a representative $\hat{e}''(-\infty;t+1)$ for which the associated noise sequence $\hat{n}''(-\infty;t+1)$ coincides over $(-\infty,t]$ with $\hat{n}(-\infty;t)+\tilde{n}$, where $\tilde{n}=(1,1)D^{t-l+1}$. By (3), addition of $\tilde{n}$ decreases the norm (because the representative had the highest norm within the parent class) by $-2\log(1-p)/p$ if and only if the representative $\hat{e}(-\infty;t)$ of the parent class had an associated noise vector sequence estimate $\hat{n}(-\infty;t)$ such that $\hat{n}_{t-l+1}=(0,0)$. Note that for a representative $\hat{e}(-\infty;t)$, the associated noise vector sequence estimate $\hat{n}(-\infty;t)$ has $\hat{n}_i\in\{(0,0),(0,1)\}, t-l+1\leqslant i\leqslant t$. Hence in the stack decoder we need an indicator register $I[0:l-1]$, with content $C(I[i:i])$ equal to zero or one according to whether n_{t-i} equals $(0,0)$ or $(0,1)$.

Table I traces the steps of the competing stack decoders when decoding a specific data vector sequence y. The left part of Table I that applies to the classical stack decoder is taken from [8]. The right part applies to our modified stack decoder. $C(I[l-1:l-1])$ is given in parentheses following the stack contents. As described above, this number is used in computing the norm $f[\hat{e}''(-\infty;t+1)]$ of one of the two new representatives. From Table I we see that for this specific decoding job the modified stack

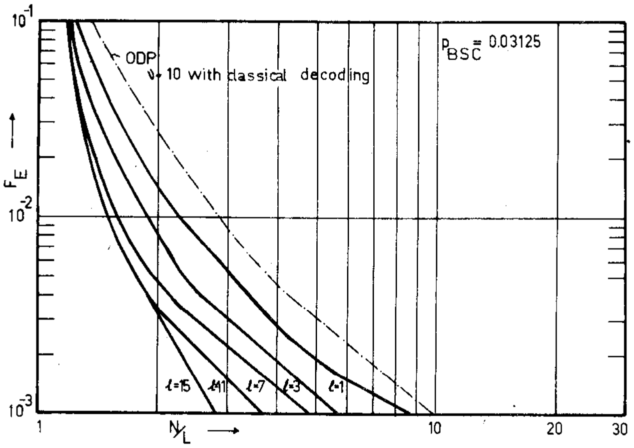

Fig. 1. Distribution of number of computations per frame $p=2^{-5}$.

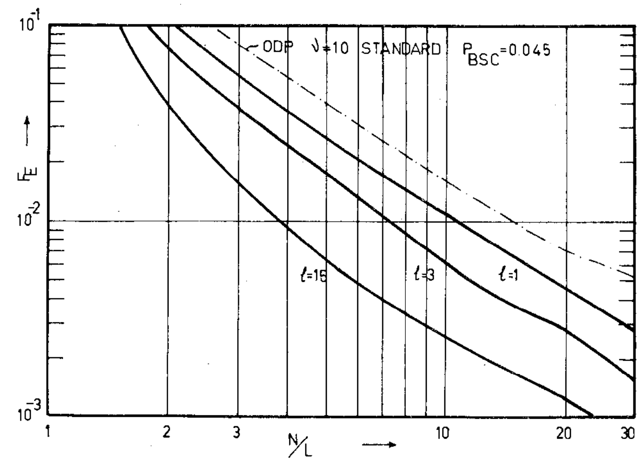

Fig. 2. Distribution of number of computations per frame at R_{comp}.

decoder requires fewer computations and less storage. In the next section we describe some simulations of the classical and modified stack decoding algorithms.

III. Simulations

A simulation run corresponds to the transmission of 10000 frames of $L=256$ binary message digits each, where each frame is followed by a tail of ν (the constraint length of the code) consecutive zeros. The simulations were carried out for both a constraint length $\nu=10$ optimum distance profile (ODP) code [7] using the classical stack decoder and for various $L_{2,\nu,l}$ codes using the modified stack decoder.

Fig. 1 is a plot of the fraction of frames (out of 10000) requiring more than N computations to decode versus N/L. Note that the minimum number of computations per frame equals the frame length $L+\nu$. Hence the minimum value of N/L equals $1+\nu/L$. The dashed curve is for the constraint length $\nu=10$ ODP code with classical stack decoding. The other curves are for $L_{2,10,1}$, $L_{2,12,3}$, $L_{2,14,7}$, $L_{2,22,11}$, and $L_{2,30,15}$ codes, respectively. Increasing the constraint length for the ODP codes beyond $\nu=10$ does not have an appreciable effect on the dashed curve in Fig. 1. The same holds true for the $l=15$ curve and the $L_{2,\nu,l}$ codes. Comparing the $l=15$ curve for modified stack decoding with the dashed curve for classical stack decoding we observe a significant reduction in the number of computations per frame. Thus the reduction in the number of computations observed in Table I of the previous section holds true in our simulations.

Fig. 2 is similar to Fig. 1, except that for $p=0.045$ the stack decoder is operating at the computational cutoff rate R_{comp} [8].

In Table I we also observed that the modified stack decoder needs less storage. Tables II and III compare the constraint length $\nu=10$ ODP code with classical stack decoding to an $L_{2,30,15}$ code with modified stack decoding. The distance profiles for the ODP $L_{2,30,15}$ codes are 2 3 3 4 4 5 5 6 6 6 7 and 2^{15} 3 4 5 5 6 7 7 8 8 9 9$\cdots$, respectively. Note that for a stack depth of 500 the classical decoder has four errors and five erasures (in 10000 frames). The modified stack decoder only requires a stack depth of 100 for the comparable performance of five errors and two erasures. Hence for $p=2^{-5}$ the modified stack decoder requires one-fifth the storage of the classical stack decoder for comparable performance.

TABLE II
Erasures and Errors for Different Stack Sizes
$p_{BSC}=0.03125$

stack depth	number of frames in error		number of frames with N/L > 30	
	classical	modified	classical	modified
25	515	39	0	0
50	170	13	30	2
75	49	7	64	3
100	20	5	57	2
200	3	5	23	1
500	4	5	5	1
1000	3	5	2	0

TABLE III
Erasures and Errors for Different Stack Sizes $p_{BSC}=0.045$

stack depth	number of frames in error		number of frames with N/L > 30	
	classical	modified	classical	modified
25	2120	375	0	0
50	1019	131	101	30
75	433	43	315	68
100	204	29	367	59
200	55	31	220	32
500	31	39	100	11
1000	32	42	46	8

IV. Conclusion

This paper describes a new class $L_{2,\nu,l}$ of binary rate one-half convolutional codes and a modified stack decoding algorithm. For a representative value $p=2^{-5}$ of the transition probability of the binary symmetric channel the new stack decoder requires less than one-fifth the storage of the classical stack decoder for comparable performance. The distribution of the number of computations per decoded frame for $L_{2,\nu,l}$ codes with modified stack decoding compares favorably with a similar distribution for ODP codes decoded in the classical manner. The $L_{2,\nu,l}$ codes can also be used to advantage in soft-decision decoding.

For the same class $L_{2,\nu,l}$ a reduced state decoder has been implemented. At each instant a small stack (of about 32 entries) contains the most probable paths, all of the same length. Preliminary results indicate that this simple decoder yields a number of errors roughly equal to the combined number of errors and erasures of the modified stack decoder.

We are presently investigating how a Fano decoder can make use of the symmetries of the class $L_{2,\nu,l}$ of binary rate one-half convolutional codes.

Acknowledgment

The authors wish to thank W. J. H. M. Lippmann and A. P. C. van Schendel for their contributions to this research and Mrs. G. H. Driever-van Hulsen for the accurate typing of the manuscript.

References

[1] K. Zigangirov, "Some sequential decoding procedures," *Probl. Peredach. Inform.*, vol. 2, no. 4, pp. 13–15, 1966.

[2] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, pp. 675–685, Nov. 1969.

[3] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970, and vol. IT-17, p. 360, May 1971.

[4] J. P. M. Schalkwijk, A. J. Vinck, and K. A. Post, "Syndrome decoding of binary rate k/n convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 553–562, Sept. 1978.

[5] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.* (*Special Issue on Error Correcting Codes—Part II*), vol. COM-19, pp. 751–772, Oct. 1971.

[6] A. J. Vinck and A. J. P. de Paepe, "Reducing the number of computations in stack decoding of convolutional codes by exploiting symmetries of the encoder," Univ. of Technol., Eindhoven, The Netherlands, TH rep. 78-E-90, ISBN 90-6144-090-4, Sept. 1978.

[7] R. Johannesson, "Robustly optimal rate one-half binary convolutional codes," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-21, pp. 464–468, July 1975.

[8] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: Wiley, 1978.