Transactions Letters

Rate-1/2 Component Codes for Nonbinary Turbo Codes

Andrew C. Reid, T. Aaron Gulliver, *Senior Member, IEEE*, and Desmond P. Taylor, *Fellow, IEEE*

***Abstract*—The iterative decoding structure and component maximum *a posteriori* decoders used for decoding binary concatenated codes can be extended to the nonbinary domain. This paper considers turbo codes over nonbinary rings, specifically ternary, quaternary, penternary, hexernary, and octernary codes. The best rate-1/2 component codes are determined using a practical search algorithm. The performance of the resulting rate-1/3 turbo codes on an additive white Gaussian noise channel using q-ary phase-shift keying modulation is given.**

***Index Terms*—Nonbinary turbo codes, ring codes.**

I. Introduction

It has been demonstrated that turbo codes, a class of parallel concatenated codes [1], provide near-capacity performance at low signal-to-noise ratios (SNRs). Practical decoding is possible because the concatenated code structure allows for near-maximum *a posteriori* (MAP) decoding via an iterative decoder [1], [2].

The vast majority of turbo-code research considers binary component codes concatenated via a random bit interleaver (e.g., [1] and [3]–[5]). In this letter, we consider turbo codes constructed from nonbinary component codes concatenated via a random symbol interleaver and mapped onto the appropriate phase-shift keying (PSK) constellation for transmission through the *additive white Gaussian noise* (AWGN) channel. The use of q-ary PSK such that the constellation size is matched to the size of the ring has two useful features. First, it results in coded signals that achieve high code diversity on fading channels [6], while retaining good bandwidth efficiency. Second, particularly in the special cases of $q = 3, 5, 6$, it leads to codes that are inherently rotationally invariant, as can be easily seen from the state diagrams of the codes. In these cases, the mapping of bits to symbols is somewhat more complicated, but that appears to be a minor penalty.

Paper approved by A. H. Banihashemi, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received October 7, 2003; revised December 28, 2004. This paper was presented in part at the IEEE International Symposium on Information Theory, June 30–July 5, 2002, Lausanne, Switzerland.

A. C. Reid and D. P. Taylor are with the Communications Research Centre, Department of Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand (e-mail: ac_reid@ieee.org; desmond.taylor@canterbury.ac.nz).

T. A. Gulliver is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: agullive@ece.uvic.ca).

Digital Object Identifier 10.1109/TCOMM.2005.855019

Let $\mathbb{Z}_q$ be the ring of integers mod q. It is well known that for $q = p^n$, p prime, n an integer, a field with q elements exists, $\mathbb{F}_q$. If q is prime, $\mathbb{F}_q = \mathbb{Z}_q$. For $q = 4$, there are two additional commutative rings other than $\mathbb{F}_q$ and $\mathbb{Z}_q$, namely, $\mathbb{F}_2 + u\mathbb{F}_2$ and $\mathbb{F}_2 + v\mathbb{F}_2$ [7]. Ring convolutional codes with nonbinary modulation have been considered by Rimoldi and Li [8], Yang and Taylor [9], and Karam *et al.* [10]. In addition, Chen *et al.* constructed ring codes for use with 6-PSK [11]. However, only White and Costello [17] consider the design of component codes for nonbinary turbo codes, and then just for $\mathbb{F}_4$ and $\mathbb{F}_8$. In addition, only results for noncoherent M-ary frequency-shift keying (M-FSK) modulation are presented. Ghrayeb and Abualrub [18] use the $\mathbb{F}_4$ component codes from [17] to compare parallel and serial concatenated turbo codes with quaternary phase-shift keying (QPSK) modulation. Results are given for punctured codes of rates 1/2 or higher. In this letter, we consider the construction of turbo codes for $q = 3, 4, 5, 6, 8$, and present the first systematic construction of component codes for M-PSK modulation.

In the next section, we discuss the construction and decoding of nonbinary turbo codes. We then investigate the code parameters that good component codes possess, find sets of such codes, and present the performance of some nonbinary turbo codes.

II. Nonbinary Turbo Codes

The encoder structure considered is identical to that used for binary turbo codes, except that all operations are on symbols from a ring of of size q. The source provides a random stream of q-ary symbols which are divided up into frames of n symbols. These symbols could be derived from a binary source in a number of ways. The symbols in each frame are encoded using a rate-1/2 *recursive systematic convolutional* (RSC) encoder to produce one set of parity symbols. The systematic symbols are then randomly permuted or interleaved, and passed through a second RSC encoder to provide a second set of parity symbols. If the systematic symbols and both sets of parity symbols are transmitted, the overall code rate is 1/3. We do not consider punctured codes here, as our aim is a performance comparison with different component codes.

The transmission scheme considered is q-ary PSK, where q is the size of the ring considered. For example, penternary codes constructed over the field $\mathbb{F}_5$ are transmitted using 5-PSK. At the receiver, the demodulator calculates soft probabilities $\mathbb{P}(y|x_i)$,

$i = 0, \ldots, q-1$, for each received symbol, where y is the received signal point, and the x_i are each of the possible transmitted constellation points. These probabilities are then passed to the iterative decoder.

In [12], Berkmann describes nonbinary iterative decoding. The block structure of the decoder used is identical to that of a standard binary iterative decoder. In binary iterative decoding, probability information is passed between the constituent decoders in the form of a time-indexed vector of log-likelihood ratios (LLRs). For nonbinary codes, a set of ratios is required to represent the probabilities associated with each symbol in the frame. Apart from the necessity to consider sets of ratios for each received symbol, the iterative process proceeds in an identical fashion to the binary case.

There is nothing inherently binary in the Bahl–Cocke–Jelinek–Raviv (BJCR) algorithm [14], so it can be used without modification for nonbinary codes. Each component decoder takes as its input the *a priori* probabilities for each of the information symbols $\mathbb{P}(u_t = u)$, and the transmission probabilities for both the systematic and parity code symbols $\mathbb{P}(y_{tS}|x_S)$ and $\mathbb{P}(y_{tP}|x_P)$, where x_S and x_P are systematic and parity points from the q-ary PSK transmission constellation, and y_{tS} and y_{tP} are the received systematic and parity signal points. In return, the decoder calculates the MAP probabilities $\mathbb{P}(u_t = u|\boldsymbol{y}_1^\tau)$.

III. Choosing Component Codes

To obtain good performance, it is necessary to choose good component codes from the set of all possible RSC codes with a particular rate and memory order. Benedetto and Montorsi [3], and later Benedetto *et al.* [4], address this problem in the binary context. They demonstrate that the component codes must be recursive for the interleaver to provide significant gain. Specifically, no interleaver gain is possible for the weight-one error events present in all nonrecursive codes, whereas for weight-two error events, the interleaver provides a gain of $\sim O(1/N)$, where N is the interleaver length. Weight-two error events are the smallest possible if a recursive component code is used. A consequence of this is that the lowest output parity weight $z_{\min}$ possible from a weight-two input to the component encoder is the dominant parameter determining turbo code error-floor performance. This is because the interleaver provides more gain for higher weight sequences. In fact, this is why codes with less than the maximum possible minimum distance can give the best performance in the waterfall or low-SNR region, provided that the minimum-distance codewords correspond to input sequences of weight greater than two (in the waterfall region, the dynamics of iterative decoding are more important than minimum distance).

It is also argued in [3] and [4] that the denominator polynomial of the function specifying the nonsystematic output of a component code should be primitive (the polynomial should have the largest possible order). This maximizes the length of the shortest error event possible from a weight-two input, thus making it easier to choose a denominator polynomial that maximizes $z_{\min}$. Also by maximizing this length, the number of weight-two events that can be present in a single frame is minimized.

In [3], the best codes are determined using a metric that first maximizes $z_{\min}$ and then has the best bit-error rate (BER) performance bound. In [4], a simpler, and therefore, more easily computed, metric is employed that considers the weight and multiplicity of codewords generated by low-weight inputs. We adapt the latter metric (described below) and associated search algorithm to find good nonbinary rate-1/2 RSC component codes for turbo codes using M-PSK modulation.

Considering the *input–output weight enumerating function* (IOWEF) for all possible rate-1/2 RSC component codes with the desired parameters, we choose the codes that first maximize the minimum output weight possible from a weight-two input, and second, minimize the multiplicity of these low-weight outputs. Next, we consider weight-three and higher inputs, and, if necessary, the second-lowest output weight due to weight-two inputs, etc., until the best codes have been identified. Depending on the situation, this metric may be computed using either Hamming or Euclidean weights.

A. The Search Algorithm

Practical considerations prevent the complete IOWEF of candidate component codes from being determined for any reasonable frame length. However, as we are only interested in the portion of the IOWEF resulting from low-weight inputs, the following method can be used to approximately apply the metric.

1) Cycle through each input weight i, starting with $i = 2$ and continuing until enough information is gained to separate all the possible component codes into sets with identical IOWEFs.
2) For each i, generate all possible input sequences of this weight that have a "1" in the first position, and length less than some reasonable length l.
3) Pass each of these sequences through the component encoder. If the encoder terminates in the zero state, and if the zero state was not reentered and left again during the sequence, then increment the appropriate term in the IOWEF.

Essentially, what these steps do is enumerate all the simple error events of length $\leq l$ that are generated from low-weight inputs. This approximate IOWEF is then used to determine the best codes.

The justification for considering only simple error events which diverge immediately is made in Benedetto *et al.* [4], and stems from the bounding method in [3], which only assumes knowledge of the WEF of the associated convolutional code, and estimates the additional multiplicities from this.

IV. A Lower Bound on $z_{\min}$

The dominant component of our chosen metric is $z_{\min}$, the minimum output parity weight due to a weight-two input. Benedetto and Montorsi [3] prove that in the binary case,

TABLE I
BEST TERNARY RATE-1/2 COMPONENT CODES

codes	states	d_2, N_2	d_3, N_3	d_4, N_4	d_5, N_5	d_6, N_6	d_{free}
$\langle 12\|11 \rangle$	3	**2,2** 3,2	**2,2** 3,4	**2,2** 3,6	**2,2** 3,8	**2,2** 3,10	4
$\langle 112\|211 \rangle$	9	**5,2** 8,2	**3,6** 6,18	**4,26** 7,134	**2,2** 5,144	**3,18** 6,802	6
$\langle 1111\|1121 \rangle$	27	**11,2** 20,2	**5,6** 6,4	**2,2** 4,6	**4,10** 5,22	**2,2** 4,22	6
$\langle 10202\|21121 \rangle$	81	**29,2** 56,2	**6,2** 9,2	**6,10** 7,2	**2,2** 3,4	**3,4** 4,6	7

considering Hamming distance, there exist rate-1/2 RSC codes with memory order ν that achieve

$$z_{\min} = 2^{\nu-1} + 2 \tag{1}$$

and furthermore, they state that this is the maximum $z_{\min}$ possible. The generalization of this bound to nonbinary codes is given in the following theorem.

Theorem 1: Consider the rate-1/2 q-ary convolutional encoders defined over $\mathbb{F}_q$ *with memory* ν *and generator matrix* $G = [1\ (k(D)/d(D))]$, *where* $d(D)$ *is a primitive polynomial. Then there exists at least one encoder such that*

$$z_{\min} = q^{\nu-1} + 2 \tag{2}$$

where ν *is the degree of* $d(D)$. *In this case,* $k(D)$ *has the form* $k(D) = \gamma_\nu D^\nu + \cdots + \gamma_0$, $\gamma_\nu \neq 0$, $\gamma_0 \neq 0$.

Proof: Let $n = (q^\nu - 1)/(q-1)$. Assuming $k(D)$ has degree $\deg[k(d)] \leq \nu$, we first prove that all polynomials $k(D)$ with $\deg[k(D)] < \nu$ yield a value of $z_{\min}$ strictly less than the right-hand side of (2).

Since $d(D)$ is primitive, it is the generator polynomial of an $(n, n-\nu)$ constacyclic Hamming code C over $\mathbb{F}_q$ [15]. In addition, $\phi(D^n - \alpha^n)$, $\phi \neq 0$, $\phi \in \mathbb{F}_q$, α a primitive element in $\mathrm{GF}(q^\nu)$, must be the shortest weight-two inputs that produce finite weight-error events, and these error events must be the lowest weight possible from a weight-two input. Moreover, the quotient $h(D)$ obtained from the division of $(D^n - \alpha^n)$ by $d(D)$ is the generator polynomial of the dual (n, ν) constacyclic code $C^\perp$. The products $h(D)k(D) = (D^n - \alpha^n)(k(D)/d(D))$ are codewords of this code, in which all codewords (except the all-zero codeword) have the same weight $z = q^{\nu-1}$, which is strictly less than (2). This completes the first part of the proof.

To increase the value of $z_{\min}$, we must increase the degree of $k(D)$ to ν. We now prove that there exist polynomials $k(D)$ of the form $k(D) = \gamma_\nu D^\nu + \cdots + \gamma_0$ that can achieve (2). Split $k(D)$ as $k(D) = D \cdot \gamma_\nu D^{\nu-1} + k_2(D) \triangleq D \cdot k_1(D) + k_2(D)$, and consider that the products $c_1(D) = h(D)k_1(D)$ and $c_2(D) = h(D)k_2(D)$ are codewords of $C^\perp$. Furthermore, as $h(D)$ has the form $(D^{n-\nu} + \cdots + \eta_0)$, $c_1(D)$ must have the form $(\gamma_\nu D^{n-1} + \cdots + \gamma_\nu \eta_0 D^{\nu-1})$, and $c_2(D)$ the form $(\beta_x D^x + \cdots + \gamma_0 \eta_0)$, $x \leq D^{n-1}$.

The product $Dc_1(D)$ represents a (nonconstacyclic) shift of one position to the left of $c_1(D)$. Since the constant weight code $C^\perp$ is constacyclic, and $c_1(D)$ has degree $n-1$, the word represented by $Dc_1(D)$ coincides with a codeword of $C^\perp$ except for the most significant symbol (γ_ν corresponding to the power D^n) and the least significant symbol (a "0" instead of γ_ν/α^n, which would follow from a constacyclic shift of the codeword $c_1(D)$). Thus, summing the words represented by $Dc_1(D)$ and $c_2(D)$ yields, for the powers from D up to D^{n-1}, part of a codeword of $C^\perp$. This portion of the codeword will have weight $q^{\nu-1}$, provided that $\gamma_\nu/\alpha^n + \gamma_0\eta_0 = 0$. As to the remaining powers, D^n contributes one to the weight, and D^0 adds another one, since $c_2(D)$ has a least significant bit equal to $\gamma_0\eta_0$, whereas this bit in $Dc_1(D)$ is 0. ■

This theorem means that we only need examine codes which achieve this bound. In fact, $z_{\min} = q^{\nu-1} + 2$ was the maximum achievable by all codes presented here. Note that this theorem applies only to Hamming distance.

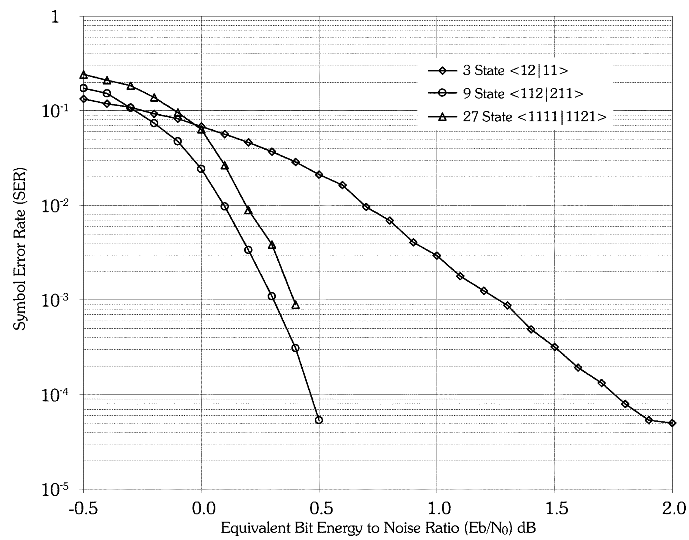

Fig. 1. Performance results for ternary turbo codes.

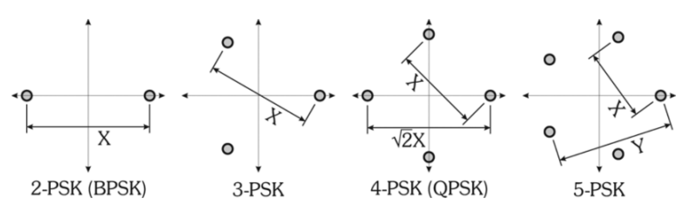

Fig. 2. Symbol-to-symbol distances for various n-PSK constellations.

TABLE II
BEST $\mathbb{F}_4$ RATE-1/2 RSC CODES FOUND USING THE PAIRWISE OPTIMIZATION CRITERION AND HAMMING OR EUCLIDEAN DISTANCES WITH NATURAL MAPPING

Hamming Distance						
codes	states	d_2, N_2	d_3, N_3	d_4, N_4	d_5, N_5	d_{free}
⟨12|21⟩	4	**2,3** 3,3	**2,3** 3,9	**2,3** 3,15	**2,3** 3,21	4
⟨111|121⟩	16	**6,3** 10,3	**3,3** 4,12	**2,3** 4,15	**3,9** 4,48	6
⟨1203|2111⟩	64	**18,3** 34,3	**5,3** 6,3	**3,3** 4,6	**4,6** 5,54	7
Euclidean Distance						
codes	states	d_2, N_2	d_3, N_3	d_4, N_4	d_5, N_5	d_{free}
⟨12|21⟩	4	**2.000, 1** 2.414, 2 3.000, 2	**2.000, 1** 2.414, 2 3.000, 2	**2.000, 2** 2.828, 1 3.000, 4	**2.000, 1** 2.414, 2 3.000, 10	4.000
⟨101|121⟩	16	**6.828, 3** 10.828, 1 11.657, 2	**2.000, 2** 2.828, 1 4.000, 3	**2.000, 2** 2.828, 1 4.000, 8	**2.000, 2** 2.828, 1 3.414, 6	5.000
⟨1203|2131⟩	64	**20.485, 3** 38.142, 1 38.971, 2	**5.414, 2** 6.000, 1 6.243, 1	**3.414, 3** 4.000, 3 4.828, 6	**4.000, 7** 4.828, 10 5.414, 28	7.414

V. RESULTS

A. Ternary Codes

Table I presents the best ternary component codes found using the method presented in Section III. The notation used to describe a particular encoder is $\langle a_n \dots a_0 | b_n \dots b_0 \rangle$, where a_i and b_i are coefficients of the numerator and denominator of the nonsystematic portion of the encoder matrix $G(D)$, giving

$$\langle a_n \dots a_0 | b_n \dots b_0 \rangle \triangleq \left[1 \;\; \frac{a_n D^n + a_{n-1} D^{n-1} + \cdots + a_1 D + a_0}{b_n D^n + b_{n-1} D^{n-1} + \cdots + b_1 D + b_0} \right]. \tag{3}$$

The bold d_i, N_i pairs in the table are the Hamming weight and multiplicity of the lowest weight parity sequences possible from a weight-i information sequence. The second (indented) d_i, N_i pair are the weight and multiplicity of the next lowest weight parity sequences possible.

For example, consider row two of the table, which lists the best nine-state ternary component code. The minimum parity weight possible from a weight-two input is five, and there are two different weight-two inputs that will give this output weight. The next lowest parity weight possible from a weight-two input is eight, and again, there are two different inputs that will give this parity weight.

As expected, all codes in the table achieve $z_{\min} = 3^{\nu-1} + 2$.

The performance of rate-1/3 turbo codes based on the component codes in Table I is presented in Fig. 1. The frames contained 1000 data symbols, and the parity sequences were terminated using the method described in [5]. An S-random interleaver was used, with $S = 21$ [16]. Transmission was over an AWGN channel using 3-PSK modulation. 10 decoder iterations were performed, and the simulation was stopped when 50 frames containing errors had been received.

These results show that as one would expect, a turbo code based on the single memory element component code performs poorly. However, the nine-state code provides a turbo code with excellent performance. Note that the 27-state code has worse performance than the nine-state code.

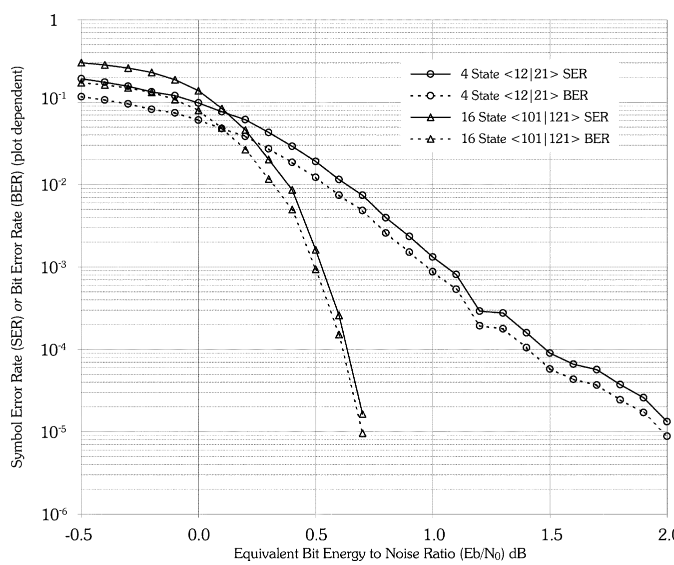

Fig. 3. Performance results for quaternary turbo codes.

B. Quaternary Codes

For quaternary codes, we consider $\mathbb{F}_4, \mathbb{Z}_4, \mathbb{F}_2 + u\mathbb{F}_2$, and $\mathbb{F}_2 + v\mathbb{F}_2$. With 4-PSK modulation, the distance between any two constellation points is not a constant, and so, Euclidean distance is no longer equivalent to Hamming distance. The distances in the constellations for 2-PSK, 3-PSK, 4-PSK, and 5-PSK are illustrated in Fig. 2. X denotes the minimum Euclidean distance between constellation points. For convenience, we define $X = 1$.

Table II presents the best codes found when considering Hamming distance with natural mapping (symbols in numerical order 0, 1, 2, 3), over $\mathbb{F}_4$. As in the previous tables, the d_i, N_i are the minimum Hamming parity weight and associated multiplicity for inputs of weight i.

TABLE III
BEST $\mathbb{Z}_5$ AND $\mathbb{Z}_6$ RATE-1/2 RSC CODES FOUND USING THE PAIRWISE OPTIMIZATION CRITERION AND EUCLIDEAN DISTANCES WITH NATURAL MAPPING

$\mathbb{Z}_5$						
codes	states	d_2, N_2	d_3, N_3	d_4, N_4	d_5, N_5	d_{free}
$\langle 12 \vert 11 \rangle$	5	**2.618, 4** 3.618, 2 4.236, 2	**2.000, 2** 3.000, 4 3.236, 2	**2.618, 4** 3.000, 2 3.618, 12	**2.000, 2** 3.000, 8 3.236, 2	4.618
$\langle 113 \vert 211 \rangle$	25	**8.236, 2** 10.090, 2 15.090, 2	**3.618, 6** 4.236, 6 5.618, 2	**4.000, 4** 4.618, 10 5.236, 24	**2.000, 2** 3.236, 2 4.000, 2	6.618
$\mathbb{Z}_6$						
codes	states	d_2, N_2	d_3, N_3	d_4, N_4	d_5, N_5	d_{free}
$\langle 21 \vert 11 \rangle$	6	**2.000, 1** 2.732, 2 3.464, 2	**2.000, 2** 2.732, 2 3.000, 2	**2.000, 2** 2.732, 2 3.000, 2	**2.000, 2** 2.732, 2 3.000, 4	4.000
$\langle 145 \vert 511 \rangle$	36	**8.000, 1** 8.660, 2 12.000, 1	**3.732, 2** 4.000, 1 5.000, 4	**3.732, 4** 4.000, 5 5.000, 8	**2.000, 2** 3.464, 2 4.000, 9	6.732

The 4- and 16-state codes identified by White and Costello [17] have the same parameters as the codes listed in Table II. In this case, the aim was to find good component codes for turbo codes using noncoherent FSK modulation. Their 64-state code is not included, as it was eliminated by the greater depth of our search.

Note that each of the codes listed represents a set of three multiplicatively equivalent codes, as shown by the following lemma.

Lemma 1: Over any field $\mathbb{F}_q$, the set of codes $\langle \alpha a_n \ldots \alpha a_0 | b_n \ldots b_0 \rangle$ $\forall \alpha \in \mathbb{F}_q$ have identical Hamming weight spectra.

Proof: Over $\mathbb{F}_q$ division is defined, and hence, we can see from the code-transfer function that if an input $\boldsymbol{\gamma}$ applied to the code $\langle a_n \ldots a_0 | b_n \ldots b_0 \rangle$ generates the output $\boldsymbol{\phi}$, then when the input $\boldsymbol{\gamma}$ is applied to the codes $\langle \alpha a_n \ldots \alpha a_0 | b_n \ldots b_0 \rangle$, the output will be $\boldsymbol{\phi}/\alpha$. Clearly, as we are only considering Hamming weight, the outputs $\boldsymbol{\phi}$ and $\boldsymbol{\phi}/\alpha$ will have identical weight. This is true for all inputs $\boldsymbol{\gamma}$, thus the codes $\alpha\langle a_n \ldots a_0 | b_n \ldots b_0 \rangle$ will have the same weight spectrum. ■

The use of a 4-PSK constellation means that the errors between different symbols are not equally likely. Because of this, codes obtained using a Euclidean distance metric will provide component codes more suited to the modulation. Table II presents the best Euclidean distance codes over $\mathbb{F}_4$ for the natural symbol to constellation mapping. In this case, the d_i are the lowest, second lowest, and third lowest Euclidean parity weights possible from inputs with Hamming weight i.

Codes were also found over the three rings $\mathbb{Z}_4$, $\mathbb{F}_2 + u\mathbb{F}_2$, and $\mathbb{F}_2 + u\mathbb{F}_2$. In all of these cases, the codes identified had lower distances than those found over $\mathbb{F}_4$. One reason for this is the fact that the maximum order of a polynomial of degree n over $\mathbb{F}_4$ is larger than the corresponding maximum order over the other rings. In addition, because of the multiplicative equivalence of the best codes identified, identical codes were found using the other symbol to constellation mappings (0, 1, 3, 2 and 0, 2, 1, 3).

The performance of the rate-1/3 turbo codes based on the best 4- and 16–state codes is presented in Fig. 3. The channel and other parameters are the same as for the ternary codes. Because binary information can easily be represented using quaternary symbols (two bits per symbol), the BERs are plotted in addition to the symbol-error rates. For each point, all possible digit-to-symbol (constellation) mappings were tried. For all but two high-SNR points, the standard Gray mapping gave the best performance.

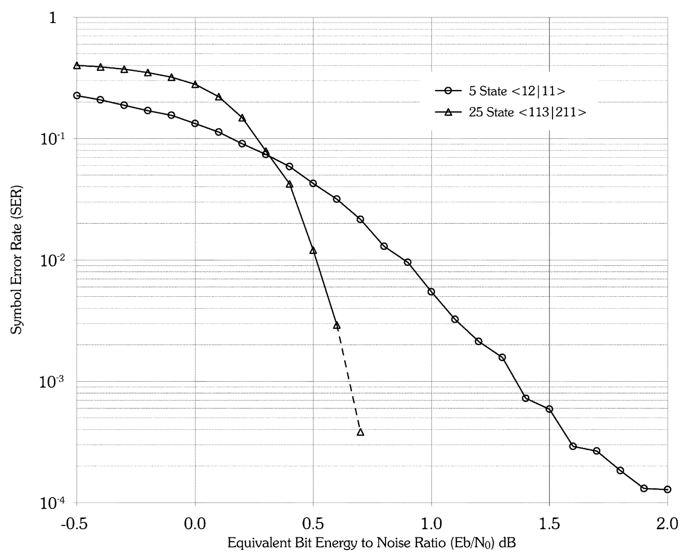

Fig. 4. Performance results for penternary turbo codes.

Compared with the symbol-based turbo code presented in [19], the 4-state code provides slightly better performance, while the 16-state code is 1.0 dB better at a BER of 10^{-4} (note that the blocklength in [19] is smaller).

C. Penternary Codes

Table III presents the best Euclidean distance codes over $\mathbb{Z}_5$ for the natural symbol-to-constellation mapping. The performance of the rate-1/3 turbo codes based on the best 5- and 25-state codes is presented in Fig. 4. The channel and other parameters are the same as for the ternary codes, except for the final point on the 25-state curve, which represents only 13 frames in error.

TABLE IV
BEST $\mathbb{F}_8$ RATE-1/2 RSC CODES FOUND USING THE PAIRWISE OPTIMIZATION CRITERION AND HAMMING OR EUCLIDEAN DISTANCES WITH NATURAL MAPPING

Hamming Distance						
codes	states	d_2, N_2	d_3, N_3	d_4, N_4	d_5, N_5	d_{free}
$\langle 12 \vert 11 \rangle$	8	**2, 7** 3, 7 4, 7	**2, 7** 3, 49 4, 91	**2, 7** 3, 91 4, 427	**2, 7** 3, 133 4, 1015	4
$\langle 171 \vert 131 \rangle$	64	**10, 7** 18, 7 26, 7	**3, 7** 4, 14 5, 21	**3, 14** 4, 70 5, 196	**3, 35** 4, 126 5, 770	6
Euclidean Distance						
codes	states	d_2, N_2	d_3, N_3	d_4, N_4	d_5, N_5	d_{free}
$\langle 12 \vert 51 \rangle$	8	**2.848, 2** 3.414, 2 4.262, 1	**2.848, 1** 3.000, 1 3.414, 1	**2.000, 1** 2.848, 1 3.000, 5	**2.000, 1** 2.848, 1 3.000, 1	4.848
$\langle 112 \vert 621 \rangle$	64	**17.551, 1** 18.399, 2 18.598, 2	**3.848, 1** 4.414, 1 4.696, 1	**3.848, 1** 4.613, 1 4.696, 1	**3.848, 2** 4.000, 2 4.414, 1	6.848

D. Hexernary and Octernary Codes

The last sets of codes we present are for $\mathbb{Z}_6$ and $\mathbb{F}_8$. Table III lists the best 6- and 36–state codes found over $\mathbb{Z}_6$ when considering Euclidean distance and the natural symbol-to-constellation mapping for 6-PSK. Table IV lists the best Hamming and Euclidean distance codes over $\mathbb{F}_8$. The codes obtained for the Hamming distance have the same parameters as those given by White and Costello [17], and, as expected, they achieve the bound calculated in Section IV.

VI. CONCLUSIONS

We have identified good rate-1/2 component codes for use in nonbinary codes, based on the response of these codes to low-weight inputs. Additionally, we have obtained a bound on an important component code parameter, $z_{\min}$, for codes over any field $\mathbb{F}_q$. The performance results show that, in particular, a turbo code based on 9-state ternary component codes performs very well.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near-Shannon limit error-correcting coding and decoding: Turbo codes ," in *Proc. IEEE Int. Conf. Commun.*, 1993, pp. 1064–1070.

[2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[3] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591–600, May 1996.

[4] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 9, pp. 1101–1105, Sep. 1998.

[5] D. Divsalar and F. Pollara, "Turbo codes for deep-space communications,", JPL TDA Prog. Rep. 42–120, 1995.

[6] N. A. Van Stralen, "Channel coding for time-selective fading channels," Ph.D. dissertation, McMaster Univ., Hamilton, ON, Canada, 1994.

[7] C. Bachoc, "Application of coding theory to the construction of modular lattices," *J. Combin. Theory Ser. A*, vol. 78, pp. 92–119, 1997.

[8] B. Rimoldi and Q. Li, "Coded continuous phase modulation using ring convolutional codes," *IEEE Trans. Commun.*, vol. 43, no. 11, pp. 2714–2720, Nov. 1995.

[9] R. H.-H. Yang and D. P. Taylor, "Trellis-coded continuous-phase frequency-shift keying with ring convolutional codes," *IEEE Trans. Inf. Theory*, vol. 40, no. 7, pp. 1057–1067, Jul. 1994.

[10] G. Karam, K. Gosse, and K. Maalej, "Trellis coded CPFSK over rings ," in *Proc. IEEE Int. Conf. Commun.*, 1995, pp. 673–677.

[11] C.-J. Chen, T.-Y. Chen, and H.-A. Loeliger, "Construction of linear ring codes for 6-PSK," *IEEE Trans. Inf. Theory*, vol. 40, no. 3, pp. 563–566, Mar. 1994.

[12] J. Berkmann, "On turbo decoding of nonbinary codes," *IEEE Commun. Lett.*, vol. 2, no. 1, pp. 94–96, Jan. 1998.

[13] A. C. Reid, T. A. Gulliver, and D. P. Taylor, "Classification of errors in turbo decoders," *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2045–2051, Dec. 2001.

[14] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.

[15] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[16] D. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations,", JPL TDA Prog. Rep. 42–122, vol. , 1995.

[17] G. S. White and D. J. Costello, Jr, "Construction and performance of q-ary turbo codes for use with M-ary modulation techniques," in *Proc. Conf. Inf. Sci. Syst.*, Mar. 1999, [CD-ROM].

[18] A. Ghrayeb and T. Abualrub, "Asympyoyic performance comparison of concatenated (turbo) codes over GF(4)," *Int. J. Commun. Syst.*, vol. 17, pp. 479–490, 2004.

[19] M. Bingeman and A. K. Khandani, "Symbol-based turbo codes," *IEEE Commun. Lett.*, vol. 3, no. 10, pp. 285–287, Oct. 1999.