Coherent Continuous-Phase Frequency-Shift Keying: Parameter Optimization and Code Design

Shi Cheng, *Member, IEEE,* Matthew C. Valenti, *Senior Member, IEEE,* and Don Torrieri, *Senior Member, IEEE*

***Abstract*—The symmetric information rate of a modulation-constrained transmission system is the information-theoretic limit on performance under the assumption that the inputs are independent and uniformly distributed. The symmetric information rate for continuous-phase frequency-shift keying (CPFSK) over an AWGN channel may be estimated by considering the system to be a finite-state Markov channel and executing a BCJR-like algorithm. In this paper, the estimated symmetric information rate is used along with the exact expression for the 99% power bandwidth to determine the information-theoretic tradeoff between energy and spectral efficiency for CPFSK modulation. Using this tradeoff, the code rate and modulation index are jointly optimized for a particular spectral efficiency and alphabet size. Codes are then designed for the optimized system. The codes are comprised of variable nodes (which represent irregular repetition codes), check nodes (which represent single parity-check codes), and an interleaver connecting the variable and check nodes. The degree distributions of the code are optimized from the system's EXIT chart by using linear programming. Additional details of the code design, including labeling and interleaver design, are also discussed. Simulation results show that the optimized coded systems achieve bit error rates within 0.4 dB of the information-theoretic limits at BER $= 10^{-5}$.**

***Index Terms*—Continuous-phase modulation, CPM, Capacity, Channel coding.**

I. Introduction

CONTINUOUS-PHASE modulation (CPM) is a general class of constant-envelope modulation that achieves high spectral efficiency with low spectral sidelobes by requiring a smooth phase transition between adjacent symbols. CPM is said to be *full-response* if the symbols at the input to the frequency modulator are represented by pulses that are entirely contained within one symbol interval. A simple form of full-response CPM is *continuous-phase frequency-shift keying* (CPFSK), which has a rectangular pulse shaping function that spans the entire symbol, i.e. the 1REC pulse shape [1].

Coherent detection of CPFSK is discussed in [1]. Unlike memoryless FSK, the phase of CPFSK is accumulated from symbol to symbol to maintain a smooth phase transition. When the modulation index h is a rational number, the accumulated phases take values from a finite set $\Phi \subset [0, 2\pi)$. In such a case, the phase trajectory can be viewed as a finite-state Markov random process, so that the modulator and additive white Gaussian noise (AWGN) channel can together be considered as a finite-state Markov channel (FSMC). This allows coherent detection to be performed on a trellis.

It is difficult to compute the capacity of an FSMC in general, and CPFSK in particular, since the computation requires a maximization over the probability density function (pdf) of a long input sequence. Shamai et al. [2] discuss bounds on capacity for the discrete-time intersymbol interference (ISI) channel, one type of FSMC. Fortunately in practice, the input to the FSMC is usually preceded by an outer channel encoder, which typically produces uniformly distributed outputs. Pfister et al. [3] and Arnold et al. [4] use the forward recursion of the BCJR algorithm [5] to compute the *symmetric information rate* of the FSMC, which is the mutual information when the inputs are independent and uniformly distributed. A similar technique based on the BCJR algorithm was used by Ganesan [6] and Padmanabham et al. [7] to estimate the symmetric information rate of CPM.

The symmetric information rate of CPFSK depends on the modulation index h and the alphabet size M. The bandwidth of the system depends on both of these parameters as well as the code rate r. For a given definition of bandwidth (e.g. 99% power bandwidth), there will be a tradeoff between the energy and spectral efficiency embodied in the expressions for bandwidth and symmetric information rate. As a result of the tradeoff, there will be an optimal choice for the combination of h and r which minimizes the $\mathcal{E}_b/N_0$ required for a particular M and spectral efficiency. In this paper, we discuss a methodology for optimizing h and r with respect to the symmetric information rate, required spectral efficiency, and constraints on M and coherent demodulator complexity (number of states in the trellis demodulator).

Once the system parameters are optimized, the next step in the design process is to develop coding schemes capable of achieving the information-theoretic performance limits. Iterative demodulation and decoding of CPM with binary convolutional codes is considered in [8], [9] and with non-binary convolutional codes in [10]. Narayanan et al. [11] consider a low-density parity-check (LDPC) coded minimum-shift keying (MSK) system. Ganesan [6] considers matched bit-interleaved coded-modulation (BICM) systems consisting of

Manuscript received November 22, 2007; revised July 25, 2008 and November 3, 2008; accepted November 3, 2008. The associate editor coordinating the review of this paper and approving it for publication was J. Andrews.

S. Cheng is with ArrayComm LLC, San Jose, CA (e-mail: shi.cheng@gmail.com).

M. C. Valenti is with West Virginia University, Morgantown, WV (e-mail: mvalenti@wvu.edu).

D. Torrieri is with the US Army Research Laboratory, Adelphi, MD (e-mail: dtorr@arl.army.mil).

Digital Object Identifier 10.1109/TWC.2009.071311

1536-1276/09$25.00 © 2009 IEEE

an outer LDPC code that is matched to the CPM modulation. Xiao and Aulin [12], [13] consider systems that cascade a set of irregular repetition codes (represented by variable nodes), an interleaver, a set of single parity-check codes (represented by check nodes), and a CPM modulator. From one perspective [12], the system is like an irregular repeat-accumulate (IRA) code [14] except that the accumulator of the IRA code is replaced by the continuous-phase encoder, which is recursive. This perspective was also used by Guillén i Fàbregas et al. [15] to design codes for noncoherent orthogonal FSK. From another perspective [13], the system is a serial concatenation of an outer low-density generator matrix (LDGM) code [16] with an inner CPM modulator. Regardless of the perspective, the code may be optimized by using a curve-matching technique [17], [18] that matches the EXIT curve of the variable nodes with the EXIT curve of the combination of check nodes and CPM modulator. For a given check-node degree distribution, the variable-node degree distribution is optimized to minimize the area between the two curves, which is a linear-programming problem previously applied to the design of LDPC codes [19], [20].

In this paper, we design codes with a structure similar to that of [12], [13] for the proposed CPFSK systems with parameters optimized using information-theoretic considerations. A discussion is provided regarding the choice of symbol mapping when there are parallel transitions in the trellis and certain aspects of interleaver design. The degrees of the variable and check nodes are optimized using an EXIT curve-matching technique, and results are given for spectral efficiencies of 0.5 bits per second per Hz (bps/Hz) and $M = \{2, 4, 8\}$. For all three alphabet sizes, the coded system performs within 0.4 dB of the limit predicted by the symmetric information rate at BER $= 10^{-5}$.

The remainder of this paper is organized as follows. First, the system model for CPFSK in AWGN is introduced in Section II. Next, Section III provides a brief review of the symmetric information rate of CPFSK in AWGN. Section IV uses the symmetric information rate and the expression for the 99% power bandwidth to determine the information-theoretic tradeoff between energy and spectral efficiencies and jointly optimizes h and r with respect to this tradeoff. Section V presents the code structure, including the transmitter and receiver decoding algorithm. The code optimization is discussed in Section VI. Simulation results are shown in Section VII, and finally the paper concludes in Section VIII.

II. System Model

In the following discussion, bold lowercase letters will be used to denote (column) vectors, e.g. $\mathbf{x}$, and bold uppercase letters will be used for matrices, e.g. $\mathbf{X}$. The scalar value $x_{i,j}$ is used to denote the $(i,j)^{th}$ entry of the matrix $\mathbf{X}$, while the scalar value x_i is used to denote the i^{th} element of the vector $\mathbf{x}$. All matrices and vectors are indexed starting at zero, $\mathbf{x} = [x_0, x_1, ..., x_{M-1}]^T$. The notation $\mathbf{x}_i^j$ represents the set $\{\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_j\}$.

The input to the CPFSK modulator is a sequence $\mathbf{b}$ of bits. During the i^{th} signaling interval, a group of $\log_2(M)$ bits is used to select a symbol q_i from the integers from 0 to $M-1$. A *symbol-labeling rule* is used to determine which symbol is selected. Two rules are considered: *natural labeling* and *Gray labeling*. With a natural labeling, q_i is merely the decimal representation of the group of bits. With a Gray labeling, two consecutive integers will be associated with groups of bits that differ in just one bit position. The symbols are placed into the sequence $\mathbf{q}$ and it is assumed that the elements are independent and uniformly distributed over the integers from 0 to $M-1$.

The CPFSK modulator may be decomposed into a continuous-phase encoder (CPE) and a memoryless modulator [21]. The CPE ensures that the continuous-phase constraint is satisfied by accumulating the phase of each modulated symbol according to

$$\phi_{i+1} = \phi_i + 2q_i h\pi, \tag{1}$$

where ϕ_i is the accumulated phase at the start of the i^{th} symbol [1].

For every entry of $\mathbf{q}$, the memoryless modulator chooses a signal $x_i(t)$ as the q_i^{th} signal of the set $\mathcal{S} = \{s_k(t), k = 0, 1, \cdots, M-1\}$, where

$$s_k(t) = \frac{1}{\sqrt{T_s}} \exp\left\{\frac{j2\pi kht}{T_s}\right\}, \quad t \in [0, T_s), \tag{2}$$

and h is the *modulation index*.

The complex-baseband representation of the transmitted continuous-time waveform is $\sqrt{\mathcal{E}_s}e^{j\phi_i}x_i(t)$, and the corresponding complex-baseband received signal is

$$y_i(t) = \sqrt{\mathcal{E}_s}e^{j\phi_i}x_i(t) + n_i(t), \tag{3}$$

where $n_i(t)$ is a circularly symmetric complex AWGN process with noise-spectral density N_0, and $\mathcal{E}_s$ is the energy per symbol [22].

Given the initial phase ϕ_i at the start of the i^{th} interval, the front end of the coherent receiver determines the likelihoods of receiving $y_i(t)$ conditioned on each signal in $\mathcal{S}$. Since this process is the same for every received symbol, we drop the index i for the remainder of this section. The received signal $y(t), 0 \leq t \leq T_s$, is first passed through a bank of M pairs of matched filters, with one pair matched to the in-phase and quadrature components of each signal in $\mathcal{S}$, and then sampled at the symbol epoch. The sampled signal can be written in vector form as

$$\mathbf{y} = e^{j\phi}\sqrt{\mathcal{E}_s}\mathbf{x} + \mathbf{n}, \tag{4}$$

where the complex elements of $\mathbf{x}$ and $\mathbf{n}$ are

$$x_k = \int_0^{T_s} x(t)s_k^*(t)dt \tag{5}$$

$$n_k = \int_0^{T_s} n(t)s_k^*(t)dt, \tag{6}$$

and $k = \{0, 1, ..., M-1\}$. The noise vector $\mathbf{n}$ is complex Gaussian with a covariance matrix $\mathbf{R} = \mathbf{E}(\mathbf{n}\mathbf{n}^H)$ with $(k,i)^{th}$ element

$$\begin{aligned} r_{k,i} &= N_0 \int_0^{T_s} s_k^*(t)s_i(t)dt \\ &= N_0 \frac{\sin(\pi(i-k)h)}{\pi(i-k)h} e^{j\pi(i-k)h}. \end{aligned} \tag{7}$$

When conditioned on both $\mathbf{x}$ and ϕ, the vector $\mathbf{y}$ is complex Gaussian with mean $\mathbf{x}$ and covariance $\mathbf{R}$, and has conditional pdf

$$p(\mathbf{y}|\mathbf{x},\phi) \quad = \quad \frac{1}{\pi^M \det(\mathbf{R})} e^{-(\mathbf{y}-e^{j\phi}\sqrt{\mathcal{E}_s}\mathbf{x})^H \mathbf{R}^{-1}(\mathbf{y}-e^{j\phi}\sqrt{\mathcal{E}_s}\mathbf{x})}. \tag{8}$$

The exponent can be simplified as

$$\begin{aligned} & -(\mathbf{y}-e^{j\phi}\sqrt{\mathcal{E}_s}\mathbf{x})^H\mathbf{R}^{-1}(\mathbf{y}-e^{j\phi}\sqrt{\mathcal{E}_s}\mathbf{x}) \\ = \quad & -\mathbf{y}^H\mathbf{R}^{-1}\mathbf{y} - \mathcal{E}_s\mathbf{x}^H\mathbf{R}^{-1}\mathbf{x} + 2\mathrm{Re}(e^{-j\phi}\sqrt{\mathcal{E}_s}\mathbf{x}^H\mathbf{R}^{-1}\mathbf{y}). \end{aligned} \tag{9}$$

Define $\mathbf{K} \triangleq \frac{1}{N_0}\mathbf{R}$, i.e., a normalized version of $\mathbf{R}$. Note that when $x(t) = s_\nu(t)$, $\mathbf{x}$ is the ν^{th} column of $\mathbf{K}$. Therefore, given $x(t) = s_\nu(t)$, the exponent becomes

$$-\frac{\mathbf{y}^H\mathbf{K}^{-1}\mathbf{y} + \mathcal{E}_s}{N_0} + 2\frac{\sqrt{\mathcal{E}_s}}{N_0}\mathrm{Re}(e^{-j\phi}y_\nu). \tag{10}$$

Taking the logarithm of (8) and discarding terms that are common to all hypothesis, the log-likelihood for coherent reception can be expressed for each postulated $\nu = \{0, ..., M-1\}$ as

$$\log f(\mathbf{y}|\mathbf{x} = \mathbf{k}_\nu, \phi) \quad = \quad 2\frac{\sqrt{\mathcal{E}_s}}{N_0}\mathrm{Re}(e^{-j\phi}y_\nu), \tag{11}$$

where $\mathbf{k}_\nu$ represents the νth column of $\mathbf{K}$ and $f(\mathbf{y}|\mathbf{x},\phi) \propto p(\mathbf{y}|\mathbf{x},\phi)$.

Trellis-based detection of CPFSK requires that the modulation index h be a rational number so that the accumulated phase ϕ takes values from a finite set. Suppose $h = P/Q$, where P and Q are relatively-prime positive integers. The total number of unambiguous values that ϕ can assume is Q. Thus, demodulation can be performed over a trellis with Q states and QM branches per trellis section.

III. Symmetric Information Rate

The *mutual information rate* between the input process $X = (\mathbf{x}_1, \mathbf{x}_2, ...)$ and output process $Y = (\mathbf{y}_1, \mathbf{y}_2, ...)$ is [4]

$$I(X;Y) \quad = \quad \lim_{N\to\infty}\frac{1}{N}I(\mathbf{x}_0^{N-1};\mathbf{y}_0^{N-1}) \tag{12}$$

where $I(\mathbf{x}_0^{N-1};\mathbf{y}_0^{N-1})$ is the mutual information between $\mathbf{x}_0^{N-1}$ and $\mathbf{y}_0^{N-1}$. The *symmetric information rate* is the mutual information rate under the constraint that the x_i are independent and uniformly distributed. Since the $\{x_i\}$ are independent, the chain rule of entropy gives

$$\begin{aligned} I(\mathbf{x}_0^{N-1};\mathbf{y}_0^{N-1}) \quad &= \quad H(\mathbf{x}_0^{N-1}) - H(\mathbf{x}_0^{N-1}|\mathbf{y}_0^{N-1}) \\ &= \quad \sum_{i=0}^{N-1}H(\mathbf{x}_i) - \sum_{i=0}^{N-1}H(\mathbf{x}_i|\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1}). \end{aligned} \tag{13}$$

Because $\mathbf{x}_i$ is uniformly distributed over M constellation points, $H(\mathbf{x}_i) = \log_2 M$, and all that remains to be calculated is $H(\mathbf{x}_i|\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1})$ [7]. From the definition of conditional entropy,

$$H(\mathbf{x}_i|\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1}) \quad = \quad -E\left[\log_2 p(\mathbf{x}_i|\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1})\right]. \tag{14}$$

The above expectation can be found using Monte Carlo integration [23], as described below.

To compute the probability $p(\mathbf{x}_i|\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1})$, first apply Bayes' rule to obtain

$$p(\mathbf{x}_i|\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1}) \quad = \quad \frac{p(\mathbf{x}_i,\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1})}{p(\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1})}. \tag{15}$$

Rather than explicitly calculating the denominator in (15), its value is found to ensure that

$$\sum_{\mathbf{x}_i} p(\mathbf{x}_i|\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1}) \quad = \quad 1. \tag{16}$$

Similar to [4], [6], [7], a BCJR-like method can be used to compute $p(\mathbf{x}_i,\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1})$, which is described as follows. Assume ϕ takes on values from the set Φ, whose cardinality is Q. Define α, β, γ as

$$\alpha_i(\phi_i) \quad \triangleq \quad p(\phi_i,\mathbf{y}_0^{i-1},\mathbf{x}_0^{i-1}) \tag{17}$$

$$\beta_{i+1}(\phi_{i+1}) \quad \triangleq \quad p(\mathbf{y}_{i+1}^{N-1}|\phi_{i+1}) \tag{18}$$

$$\gamma(\phi_i \to \phi_{i+1},\mathbf{y}_i,\mathbf{x}_i) \quad \triangleq \quad p(\mathbf{y}_i,\phi_{i+1}|\phi_i,\mathbf{x}_i). \tag{19}$$

Note that $\gamma(\phi_i \to \phi_{i+1},\mathbf{y}_i,\mathbf{x}_i)$ is nonzero only when $\mathbf{x}_i$ causes the state transition from ϕ_i to ϕ_{i+1}. Therefore, it may be written as

$$\begin{aligned} & \gamma(\phi_i \to \phi_{i+1},\mathbf{y}_i,\mathbf{x}_i = \mathbf{k}_\nu) \\ = \quad & p(\phi_{i+1}|\phi_i,\mathbf{x}_i = \mathbf{k}_\nu)p(\mathbf{y}_i|\phi_{i+1},\phi_i,\mathbf{x}_i = \mathbf{k}_\nu) \\ = \quad & \begin{cases} p(\mathbf{y}_i|\phi_i,\mathbf{x}_i = \mathbf{k}_\nu) & \phi_{i+1} = \phi_i + 2\nu h\pi \\ 0 & \phi_{i+1} \neq \phi_i + 2\nu h\pi. \end{cases} \end{aligned} \tag{20}$$

As with the BCJR algorithm, α can be calculated in a forward recursion as

$$\alpha_{i+1}(\phi_{i+1}) \quad = \quad \frac{1}{M}\sum_{\phi_i\in\Phi}\alpha_i(\phi_i)\gamma(\phi_i \to \phi_{i+1},\mathbf{y}_i,\mathbf{x}_i). \tag{21}$$

Similarly, β can be calculated in a backward recursion as

$$\beta_i(\phi_i) \quad = \quad \frac{1}{M}\sum_{\mathbf{x}_i}\sum_{\phi_{i+1}\in\Phi}\beta_{i+1}(\phi_{i+1})\gamma(\phi_i \to \phi_{i+1},\mathbf{y}_i,\mathbf{x}_i). \tag{22}$$

Note that $\mathbf{x}_i$ is marginalized out of the summand since $\beta_i(\phi_i)$ does not depend on it.

In the absence of knowing the starting and ending states, both α_0 and β_N can be initialized assuming equally likely states, i.e. $\alpha_0(\phi) = \beta_N(\phi) = 1/Q$, $\forall\phi \in \Phi$. Alternatively, if the initial phase ϕ_0 is known to the detector, α_0 can be set to all zeros except a one at the corresponding entry. Obviously, the effect of the initial states of α_0 and β_N diminish as N approaches infinity.

Given the above definitions, $p(\mathbf{x}_i,\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1})$ is found from

$$\begin{aligned} & p(\mathbf{x}_i,\mathbf{x}_0^{i-1},\mathbf{y}_0^{N-1}) \\ = \quad & \frac{1}{M}\sum_{\phi_i\in\Phi}\sum_{\phi_{i+1}\in\Phi}\alpha_i(\phi_i)\beta_{i+1}(\phi_{i+1})\gamma(\phi_i \to \phi_{i+1},\mathbf{y}_i,\mathbf{x}_i). \end{aligned} \tag{23}$$

Fig. 1 shows the symmetric information rate of MSK ($M = 2, h = 1/2$) computed using the method highlighted

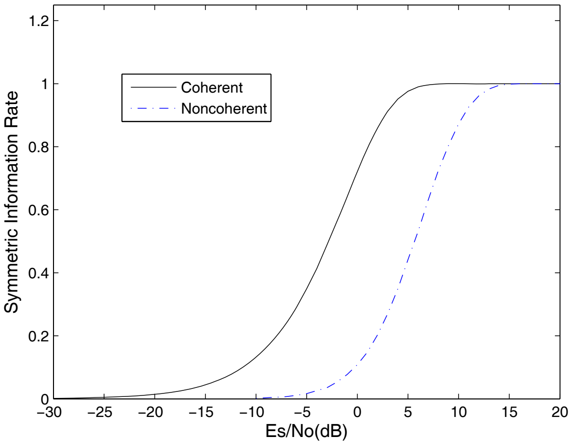

Fig. 1. The symmetric information rate of MSK with coherent reception and symbol-wise noncoherent reception.

above with $N = 100 \times 10^6$. In addition to the coherent capacity, the symmetric information rate of symbol-wise noncoherent reception [24] is shown. From this curve, it is seen that coherent reception offers a 7 dB gain over symbol-wise noncoherent reception.

There are several ways to approach the symmetric information rate in practice. One strategy is to use an iterative receiver structure and a simple code optimized for the structure, which is the topic of Section V. An alternative strategy is to use BICM without any iteration between demodulator and decoder [6], [25]. However, the loss in symmetric information rate due to using BICM is rather severe when the modulator is left in the recursive form given in Section II [6], [26]. One approach to recover this loss is to use a nonrecursive modulator instead of a recursive one [6]. Another approach is to use an adaptive symbol-labeling rule such that the labeling depends on the current state of the modulator [25]. While using BICM with either of these modifications has the benefit of not requiring iteration between demodulator and decoder, they require a strong outer error correcting code, such as an LDPC code. Thus there is a tradeoff between using a strong code in a non-iterative receiver structure, as in [6], [25], versus using a simple code in an iterative receiver structure, as we advocate in Section V.

IV. Energy-Spectral Efficiency Tradeoff

The symmetric information rate of CPFSK is a function of the per-symbol SNR $\mathcal{E}_s/N_0$ and depends on the choice of modulation index h and alphabet size M. If one fixes the value of h, M, and the information rate, then the corresponding value of $\mathcal{E}_s/N_0$ can be read from the symmetric information rate curve. This value of $\mathcal{E}_s/N_0$ represents the minimum per-symbol SNR required to achieve an arbitrarily low BER under the assumption of independent and uniformly distributed inputs. The information rate is the code rate r required to achieve the minimum per-symbol SNR. As can be seen from Fig. 1, the minimum $\mathcal{E}_s/N_0$ can be made arbitrarily small by letting the rate approach zero. However, one does not normally want to minimize the per-symbol SNR $\mathcal{E}_s/N_0$, but rather seeks to minimize the *per-bit* SNR $\mathcal{E}_b/N_0$, which is related to the per-symbol SNR by $\mathcal{E}_b = \mathcal{E}_s/(r \log_2 M)$. While one can reduce the required $\mathcal{E}_b/N_0$ by lowering the code rate, this comes at the expense of requiring a larger transmitted bandwidth to accommodate the lower code rate. As the precise relationship between code rate and required $\mathcal{E}_b/N_0$ depends on the choice of h and M, one might ask if it is perhaps better to adjust the values of h and M rather than arbitrarily lowering the value of r.

The values of h, M, and r also contribute to the bandwidth of the signal. In particular, the 99% power bandwidth B_{99} of uncoded CPFSK can be found by integrating the power-spectral density of CPFSK given in Section 4.4.2 of [27]. This bandwidth is a function of M, h, and the symbol rate $R_s = 1/T_s$. Given that $s(t)$ with parameters M and h is transmitted at a rate of R_s baud, we can define the *normalized bandwidth* to be $B(M,h) = B_{99}T_s$ Hz/baud. We can then define the *spectral efficiency* $\eta = r \log_2 M/B(M,h)$, which has units of bits-per-second-per-Hz (bps/Hz). If one constrains the spectral efficiency η, then there is a tradeoff among h, M, and r embodied by the expression for bandwidth. Once the values of η, M, and h are fixed, there is a corresponding minimum value of r; reducing the code rate below this value would violate the bandwidth constraint.

The relationship among h, M, r, and η suggests a method for optimizing these parameters for CPFSK modulation. One would first constrain η based on the system requirements for spectral efficiency. Next, one would constrain M based on complexity and implementation considerations. Then, one would constrain $h = P/Q$ to be chosen from a set of possibilities with denominator Q kept reasonably small in order to limit the demodulator complexity. For each value of h in the set of allowable values, the symmetric information rate would be computed using the techniques given in the last section. For that particular combination of h, M, and η, there will be a minimum r. The required value of $\mathcal{E}_s/N_0$ for that particular r would be read back from the symmetric information rate curve and then used to find the required value of $\mathcal{E}_b/N_0$. This would be repeated for each h in the set, and the value of h that results in the smallest $\mathcal{E}_b/N_0$ would be selected along with the corresponding rate r.

As an example, Fig. 2 shows the $\mathcal{E}_b/N_0$ required for several different spectral efficiencies as a function of h for $M = 2$. For comparison, the unconstrained AWGN capacity is also shown for each spectral efficiency. To constrain complexity, the denominator of h satisfies $Q \leq 5$. For each value of h, the value of r is set to its minimum allowable value. For example, when $\eta = 0.5$ bps/Hz, the minimum values of r are $0.39, 0.55, 0.64$, and 0.96 for $h = \frac{1}{5}, \frac{2}{5}, \frac{3}{5}$, and $\frac{4}{5}$, respectively. For the loosest constraint ($\eta = 0.02$), the required $\mathcal{E}_b/N_0$ approaches -1.6 dB for every choice of h. As the bandwidth constraint gets tighter, the required $\mathcal{E}_b/N_0$ becomes larger. When $\eta = 0.25$, the minimum required $\mathcal{E}_b/N_0$ is about -1 dB, and it is achieved at $h = \frac{3}{5}$ and code rate $r = 0.32$. When $\eta = 0.5$, the minimum $\mathcal{E}_b/N_0 = -0.1$ dB is still achieved at $h = \frac{3}{5}$, and the code rate is doubled to about 0.64. When $\eta = 0.75$, the minimum $\mathcal{E}_b/N_0 = 1.9$ dB is achieved at $h = \frac{2}{5}$, and the optimal code rate is $r = 0.83$. Note that the curves are not smooth functions of h because the amount of memory in

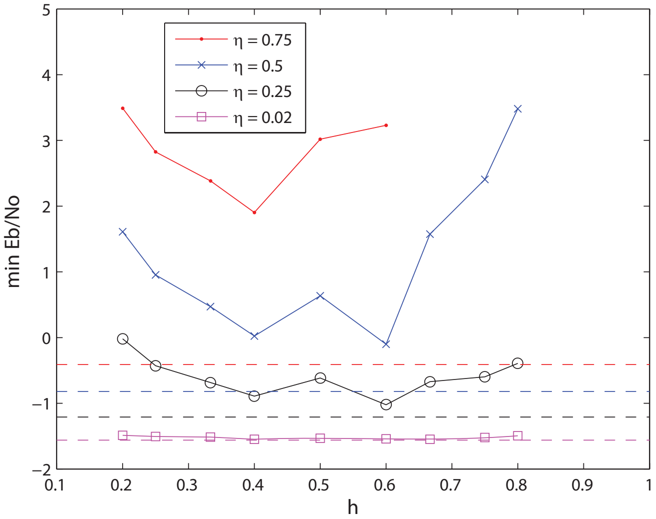

Fig. 2. $\mathcal{E}_b/N_0$ required for binary CPFSK at different spectral efficiencies. The modulation index $h = P/Q$ is constrained to be rational with denominator $Q \leq 5$. For comparison, the unconstrained AWGN capacity is also shown for each spectral efficiency (indicated by dashed lines).

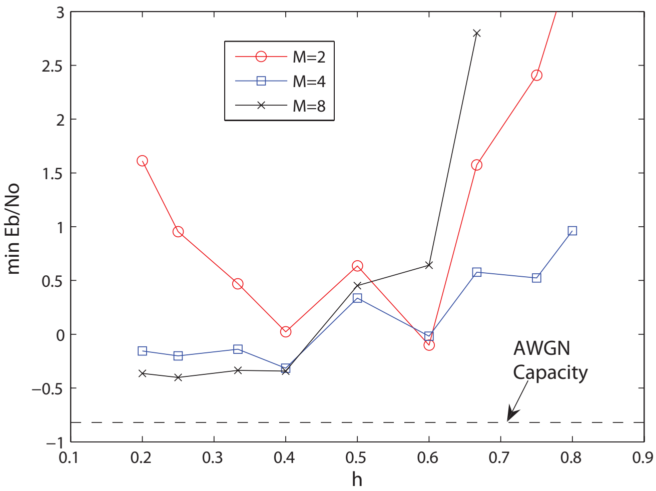

Fig. 3. $\mathcal{E}_b/N_0$ required for CPFSK with different alphabet sizes M at spectral efficiency $\eta = 0.5$ bps/Hz. The modulation index $h = P/Q$ is constrained to be rational with denominator $Q \leq 5$. For comparison, the unconstrained AWGN capacity is also shown.

the modulation varies significantly from one value of h to the next. For instance, performance of $h = 1/2$ tends to be rather poor because the modulation has only $Q = 2$ states. However, because the modulation has $Q = 5$ states, performance for $h = 2/5$ and $h = 3/5$ is better than it is for $h = 1/2$.

Fig. 3 shows the required $\mathcal{E}_b/N_0$ as a function of h for several M at spectral efficiency $\eta = 0.5$, along with the unconstrained AWGN capacity. Again, the denominator of h is chosen to satisfy $Q \leq 5$. The optimal choice of h for each M can be easily found when $\mathcal{E}_b/N_0$ achieves its lowest value. Table I lists the optimal choices of h and corresponding code rate r for each M at this spectral efficiency. Notice that performance is highly dependent on the number of states in the modulator. For instance, when $h = 1/2$, $Q = 2$ and performance is worse than for other values of h that have a higher Q. Also note the relationship between Q and M. When $Q < M$, there are fewer states than symbols and performance tends to suffer. For instance, when $h = 3/5$, $Q = 5$ and performance is much better with $M = 4$ than it is with $M = 8$. However, if Q were allowed to be larger, then $M = 8$ may outperform $M = 4$.

TABLE I

Modulation and Code Optimization Results for Spectral Efficiency $\eta = 0.5$ bps/Hz and Rational $h = P/Q$ Chosen Such That $Q \leq 5$. "Bound" Is the Minimum $\mathcal{E}_b/N_0$ Required to Achieve the Symmetric Information Rate. The i^{th} Element of the Labeling Vector Is the Octal Value of the Bit Pattern Labeling Symbol q_i. "Simulation" Is the Value of $\mathcal{E}_b/N_0$ for Which a System With $N_u = 100,000$ Message Bits and 200 Decoder Iterations Reaches a Simulated BER of 10^{-5}.

M	2	4	8
h	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{1}{4}$
r	0.6428	0.5410	0.4458
Bound ($\mathcal{E}_b/N_0$)	−0.1 dB	−0.31 dB	−0.4 dB
Labeling	Natural [0,1]	Natural [0,1,2,3]	Gray [0,1,3,2,6,7,5,4]
Variable node distribution $\{\lambda_d\}$	$\lambda_2 = 0.3$ $\lambda_3 = 0.3461$ $\lambda_6 = 0.2435$ $\lambda_7 = 0.1104$	$\lambda_2 = 0.2056$ $\lambda_3 = 0.3937$ $\lambda_9 = 0.0152$ $\lambda_{11} = 0.3855$	$\lambda_3 = 0.4947$ $\lambda_4 = 0.0577$ $\lambda_{10} = 0.3417$ $\lambda_{11} = 0.1059$
Check node distribution $\{\rho_d\}$	$\rho_1 = 0.001$ $\rho_2 = 0.999$	$\rho_1 = 0.001$ $\rho_2 = 0.999$	$\rho_1 = 0.001$ $\rho_2 = 0.999$
Threshold ($\mathcal{E}_b/N_0$)	0.03 dB	−0.22 dB	−0.19 dB
Simulation ($\mathcal{E}_b/N_0$)	0.31 dB	0.07 dB	0.02 dB

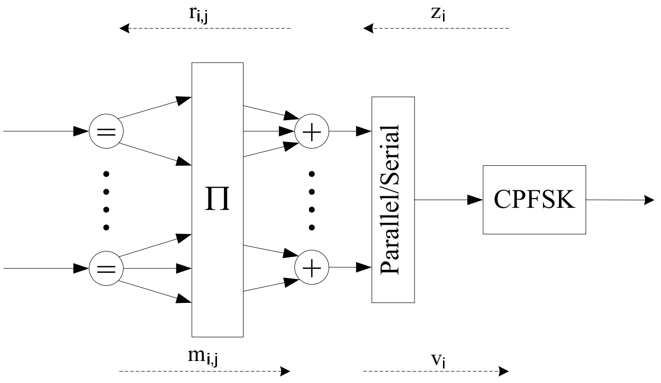

Fig. 4. Coding structure with variable nodes indicated by '=' and check nodes indicated by "+". The dashed lines above and below indicate the flow of extrinsic information through the corresponding decoder.

V. Coded System Implementation

The structure of a coded system capable of approaching the CPFSK capacity limits is shown in Fig. 4. The system is a serial concatenation of two codes separated by an interleaver Π. The outer code is a mixture of repetition codes represented by variable nodes "=". The *degree* d of a variable node is the number of times that the corresponding message bit is repeated. Since the code is irregular, the variable nodes do not all have the same degree. The entire set of repeated bits is interleaved and sent to the check nodes, represented by "+". Each check node forms a single party-check (SPC) on a distinct subset of interleaved bits. The degree of a check node is the number of edges that connect the check node to variable nodes (i.e. the number of edges that enter the check node from the left). The outputs of the SPC nodes are grouped together and passed to the modulator. As in [15], the code is nonsystematic, and therefore unlike [18], the message bits are not modulated.

At the transmitter, a binary-message vector $\mathbf{u} \in \{0,1\}^{N_u}$ is encoded in parallel by N_u variable nodes, each generating a repetition code. The *degree distribution* of the variable nodes can be described by either the node-perspective degree distribution $\tilde{\lambda}_d$ or the edge-perspective degree distribution λ_d. In particular, $\tilde{\lambda}_d$ is the fraction of nodes that have degree d, while the λ_d is the fraction of edges that touch degree d nodes. The two perspectives are related by

$$\lambda_i \;=\; \frac{\tilde{\lambda}_i i}{\sum_{d=1}^{d_v} \tilde{\lambda}_d d} \qquad (24)$$

where d_v is the maximum variable-node degree.

The outputs from the N_u variable nodes form a vector $\mathbf{c}'$ of length $N_c = \sum_{d=1}^{d_v} \tilde{\lambda}_d d N_u$. $\mathbf{c}'$ is then interleaved into $\mathbf{c}$ and forwarded to N_b check nodes. The check-node degree distributions may be represented in either node-perspective $\tilde{\rho}_d$ or edge-perspective ρ_d. The variables N_b and N_c are related by $N_b = N_c/(\sum_{d=1}^{d_c} \tilde{\rho}_d d)$, where d_c is the maximum check-node degree. Each check node computes the single parity-check of its inputs, and the check nodes form the vector $\mathbf{b}$ for modulation. Therefore, the code rate r satisfies

$$r \;=\; \frac{\sum_{d=1}^{d_c} \tilde{\rho}_d d}{\sum_{d=1}^{d_v} \tilde{\lambda}_d d} = \frac{\sum_{d=1}^{d_v} \frac{\lambda_d}{d}}{\sum_{d=1}^{d_c} \frac{\rho_d}{d}}. \qquad (25)$$

Before it is modulated, the binary vector of bits $\mathbf{b}$ must be transformed into the M-ary symbol vector $\mathbf{q}$ by an appropriate symbol-labeling function $g(\cdot)$, which can be expressed as

$$q_i \;=\; g\left(\sum_{j=0}^{\mu-1} b_{i\mu+j} 2^{\mu-1-j}\right), \qquad (26)$$

where $\mu = \log_2 M$. In this paper, we simply assume N_b is divisible by μ. Otherwise, $\mathbf{b}$ can be padded to meet this requirement. Therefore, $\mathbf{q}$ is an M-ary vector of length $N = N_b/\mu$. For the binary case, labeling is not important, since the two frequency tones are interchangeable. However, when M is greater than 2, the labeling can be very important for certain values of h. This point will be discussed in the next section.

At the receiver, decoders for each of the inner and outer codes exchange extrinsic information using a turbo-like schedule [18], or equivalently using the sum-product algorithm [28]. Fig. 4 also shows the extrinsic information flow using dashed arrows. During the first stage, based on the channel observation $\mathbf{y}_0^{N-1}$, the log-likelihood ratio $\mathbf{z}$ is found by using the BCJR [5] or SISO [29] decoding algorithm. The j^{th} element of $\mathbf{z}$ is

$$z_j \;=\; \log \frac{p(b_j = 0|\mathbf{y}_0^{N-1}, \mathbf{v}\backslash v_j)}{p(b_j = 1|\mathbf{y}_0^{N-1}, \mathbf{v}\backslash v_j)}, \qquad (27)$$

where $\mathbf{v}$ is the extrinsic information from the check nodes, and $\backslash v_j$ indicates that v_j is excluded, so that z_j carries extrinsic information only. In the first decoding half-iteration, $\mathbf{v}$ is all zeros.

The check nodes then update the messages sent to the variable nodes. The output information $r_{i,j}$ from j^{th} check node to i^{th} variable node, assuming they are connected, can be calculated as [28]

$$\begin{aligned} r_{i,j} \;&=\; \mathrm{sign}(z_j) \prod_{i':\mathcal{R}_j\backslash i} \mathrm{sign}(m_{i',j}) \\ &\quad \times \psi\left(\psi(|z_j|) + \sum_{i':\mathcal{R}_j\backslash i} \psi(|m_{i',j}|)\right) \end{aligned} \qquad (28)$$

where

$$\mathrm{sign}(z) \;=\; \begin{cases} -1 & z < 0 \\ 1 & z \geq 0 \end{cases} \qquad (29)$$

$$\psi(|z|) \;=\; \log \frac{e^{|z|}+1}{e^{|z|}-1}, \qquad (30)$$

$\mathcal{R}_j$ is the set of indices of the variable nodes connected to the j^{th} check node, and $\backslash i$ means excluding the element i. Here, $m_{i,j}$ is the *a priori* information from the i^{th} variable node to the j^{th} check node, which is zero for the first half-iteration.

The second half-iteration begins with every variable node updating its output, which is forwarded to the check nodes. When the i^{th} variable node and j^{th} check node are connected, the output is

$$m_{i,j} \;=\; \sum_{j':\mathcal{M}_i} r_{i,j'} - r_{i,j}, \qquad (31)$$

where $\mathcal{M}_i$ is the set of indices of the check nodes connected to the i^{th} variable node. Here, the first term $\sum_{j':\mathcal{M}_i} r_{i,j'}$ is used for the hard decision of the decoding output.

After all variable nodes update their outputs, the check nodes calculate the extrinsic information forwarded to the CPFSK SISO. Every check node processes the messages from all the connected variable nodes, and the information on the j^{th} check node can be computed as

$$v_j \;=\; \prod_{i':\mathcal{R}_j} \mathrm{sign}(m_{i',j}) \psi\left(\sum_{i':\mathcal{R}_j} \psi(|m_{i',j}|)\right) \qquad (32)$$

The second half of the first iteration is finished once all check nodes update their outputs $\mathbf{v}$. The second iteration then starts to compute (27) with the nonzero sequence $\mathbf{v}$, and performs (28), (31) and (32) accordingly. It is feasible that the variable nodes and check nodes can perform several local iterations within a single global iteration. That is, (28) and (31) are evaluated more than once before (27) is executed in the next global iteration. But in this paper, in order to exploit the most information from the CPFSK trellis, we only perform (28) and (31) once per global iteration.

VI. Code Optimization

EXIT charts are often used to analyze the convergence behavior of iterative decoding systems. In [17], a curve-matching technique was applied that allows EXIT charts to be directly used as a code design methodology. This technique was later applied to the design of IRA codes in [18] and systems using orthogonal FSK with symbol-by-symbol noncoherent detection in [15]. Here, we apply the EXIT curve-matching technique to design codes for CPFSK with coherent detection.

An EXIT chart is created for a particular SNR by drawing the *information-transfer* functions for the inner and outer codes on the same plot. The information-transfer function for an outer repetition code of degree d is [30]

$$I_{E,d}^{(o)}(I_A^{(o)}) \quad = \quad J\left(\sqrt{d-1}J^{-1}(I_A^{(o)})\right), \qquad (33)$$

where the superscript (o) denotes the outer code, and the subscripts A and E represent the *a priori* input information and the extrinsic output information. The function $J(\cdot)$ is defined in [30] as

$$J(\sigma) \quad = \quad \int \frac{1}{2\pi\sigma} e^{-\frac{\left(x-\sigma^2\right)^2}{2\sigma^2}} \log_2\left(1+e^{-x}\right) dx, \quad (34)$$

and can be predetermined by numerical or Monte Carlo integration.

The overall information-transfer function for the irregular outer code can be approximated by using the edge-perspective degree distribution to linearly combine the component information-transfer functions according to [30]

$$I_E^{(o)}(I_A^{(o)}) \quad = \quad \sum_{d=2}^{d_v} \lambda_d I_{E,d}^{(o)}(I_A^{(o)}). \qquad (35)$$

Note that $d = 1$ does not appear in the above summation because $I_{E,1}^{(o)}(I_A^{(o)}) = J(0) = 0$. This implies that degree-one variable nodes do not help the iterative decoding, and so in our code design we always set $\lambda_1 = 0$.

As with the outer code, the overall information-transfer function of the inner code can be approximated by using the corresponding edge-perspective degree distribution to linearly combine the component information-transfer functions $I_{E,d}^{(i)}$, resulting in

$$I_E^{(i)}(I_A^{(i)}) \quad = \quad \sum_{d=1}^{d_c} \rho_d I_{E,d}^{(i)}(I_A^{(i)}). \qquad (36)$$

What remains is the calculation of the function $I_{E,d}^{(i)}(\cdot)$ for each d. Unlike the outer code, the component information-transfer functions $I_{E,d}^{(i)}(\cdot)$ cannot be easily expressed in integral form like (33)-(34), and therefore must be found via Monte Carlo simulation for each d as follows. A length-N_c vector $\mathbf{c}$ of independent and uniformly distributed binary symbols is randomly generated and encoded into the length-N_b vector $\mathbf{b}$ by the check nodes. Typically, N_c is chosen to be large in order to reduce the influence of the initial and final states of the CPFSK modulator trellis. The symbol-labeling function (26) transforms $\mathbf{b}$ into the length-N symbol vector $\mathbf{q}$ which is then passed into the CPFSK modulator to produce the modulated waveform $x(t)$. The modulated signal is passed through an AWGN channel and a bank of matched filters to produce the sequence $\mathbf{y}_0^{N-1}$. The actual received sequence $\mathbf{y}_0^{N-1}$ and a simulated *a priori* input sequence $\mathbf{v}$ are input to the trellis-based CPFSK decoder, which produces the extrinsic output $\mathbf{z}$ given by (27). The sequence $\mathbf{v}$ is created using (32), where each $m_{i,j}$ corresponds to the simulated message received by the j^{th} check node from the i^{th} variable node. The $m_{i,j}$'s are assumed to be conditionally Gaussian and consistent, with variance σ^2 and mean $(-1)^{c_k}\sigma^2/2$, where c_k is the corresponding simulated bit of $\mathbf{c}$. The variance σ^2 is found from the information-transfer function's argument $I_A^{(i)}$ by inverting (34). Once $\mathbf{z}$ is generated, (28) is used to generate the messages $r_{i,j}$ sent from the check nodes to the variable nodes. Finally, an estimate of $I_{E,d}^{(i)}(\cdot)$ is found for the given codeword and channel realization by measuring the mutual information between $\mathbf{c}$ and the corresponding $r_{i,j}$'s. The process is repeated for many simulated codewords and channel realizations, and the sample mean is computed.

Once the information-transfer functions for the inner and outer codes have been found, they are drawn on the same plot. The inner-code's information-transfer function is drawn with $I_A^{(i)}$ as its horizontal axis and $I_E^{(i)}$ as its vertical axis, while the outer-code's information-transfer function is drawn with $I_E^{(o)}$ as its horizontal axis and $I_A^{(o)}$ as its vertical axis. The plot showing both of these curves constitutes the system's EXIT chart. The code is said to *converge* if there is a gap between the two curves, and the *convergence threshold* is the minimum SNR for which the two curves just barely touch. The design objective is to minimize this threshold through the proper selection of the degree distributions.

A. Degree Distribution Optimization

The convergence threshold can generally be minimized by minimizing the area between the inner and outer EXIT curves. This property was used in [15] and [31] to design capacity-approaching codes. We apply the same principle by first fixing the degree distribution of the inner code $\{\rho_d\}$ and the channel SNR $\mathcal{E}_s/N_0$, and then finding the degree distribution of the outer code $\{\lambda_d\}$ that minimizes the area between the curves. This can be done by using linear programming. We sample the outer-code's EXIT curve $I_E^{(o)}(\cdot)$ along the $I_A^{(o)}$ axis and the inner-code's inverse EXIT curve $I_A^{(i)}(\cdot) = I_E^{(i)-1}(\cdot)$ along the $I_A^{(o)}$ axis. Let $I_i \in (0,1)$ denote the i^{th} sampling point and $\mathcal{I}$ denote the indices of the sampling points. Convergence requires the two curves do not intersect, which implies that $I_A^{(i)}(I_i) < I_E^{(o)}(I_i)$ for all $i \in \mathcal{I}$. When there are a large number of uniformly spaced sampling points, the area between the two curves can be approximated as

$$A \quad \propto \quad \sum_{i\in\mathcal{I}} \left(I_E^{(o)}(I_i) - I_A^{(i)}(I_i)\right). \qquad (37)$$

Given that the maximum variable-node degree is d_v, the area A in (37) can be minimized subject to the following constraints: (1) $I_A^{(i)}(I_i) < I_E^{(o)}(I_i)$ for all $i \in \mathcal{I}$; (2) $\sum_{d=2}^{d_v} \lambda_d = 1$; and (3) The desired code rate r in (25) is attained. If a solution to the linear-programming problem is found for a particular channel SNR $\mathcal{E}_s/N_o$, then the SNR is lowered and the process repeated until a solution can no longer be found. The final design and the convergence threshold is found from the last successful solution to the linear-programming problem. If desired, the check-node and variable-node distributions could be iteratively optimized by fixing one while solving the corresponding linear-programming problem required to optimize the other. However, the convergence threshold is much more strongly influenced by the variable-node distribution than the check-node distribution.

B. Symbol Labeling Issues

The way that edges in the demodulator trellis are labeled with symbols has a significant impact on performance. If one

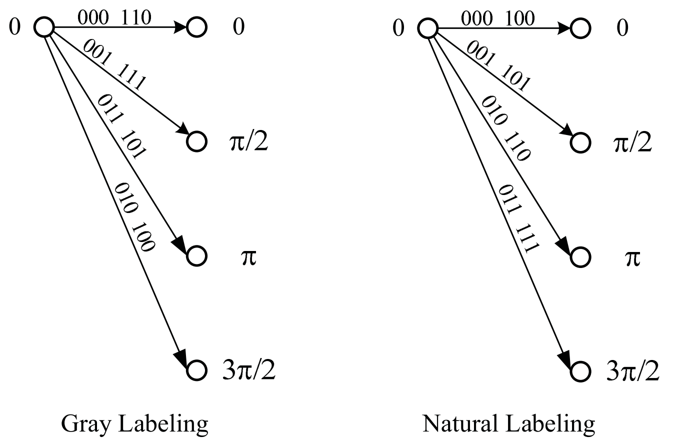

Fig. 5. Two symbol labelings for $M = 8$, $h = 1/4$, and $\phi_i = 0$. In this example, there are two parallel edges connecting the starting state with each of the four possible ending states. This trellis is labeled by labeling the first edge from top to bottom followed by labeling the second edge from top to bottom.

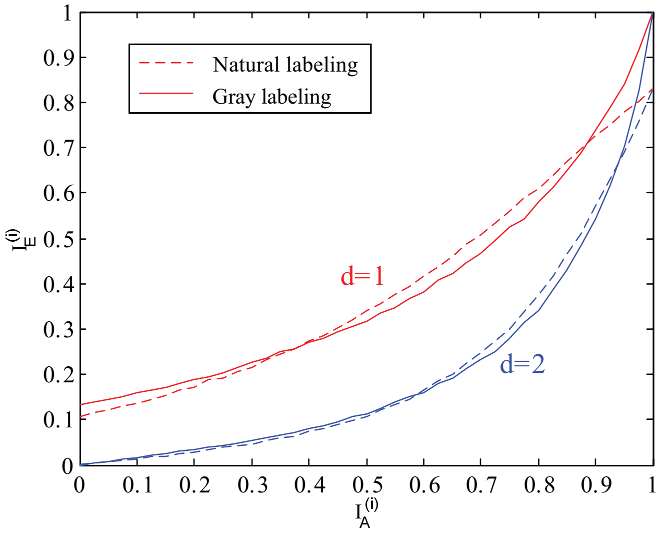

Fig. 6. Inner-code EXIT curves for $M = 8$ and $h = 1/4$ with Gray and natural symbol labelings. $\mathcal{E}_s/N_0 = 0$ dB.

were designing an uncoded or non-iterative system, the labels of all edges would need to be jointly optimized in order to minimize the BER. However, for an iterative system, the extrinsic information from the decoder helps to resolve the starting and ending states. As the corresponding EXIT chart is driven to the upper-right region of operation, the starting and ending states become essentially known. For such iterative systems, the most important symbol-labeling consideration is how *parallel edges* are labeled.

Parallel edges occur when $M > Q$, in which case there are more edges leaving each starting state than there are possible ending states. As an example, Fig. 5 shows the trellis for $M = 8$, $h = 1/4$, and starting state $\phi_i = 0$. In this case, there are two parallel edges connecting the starting state with each of the four possible ending states. When there are parallel edges, several alternative labelings are possible. For instance, the right side of Fig. 5 shows a natural labeling while the left side shows a Gray labeling. The symbol-labeling function has a profound effect on the shape of the inner-code's EXIT curve. Fig. 6 shows the EXIT curves for $M = 8$, $h = 1/4$ at $\mathcal{E}_s/N_0 = 0$ dB with Gray and natural labelings. The effect of the symbol labeling is most pronounced on the right side of the EXIT curves. With Gray labeling, the EXIT curves for codes with degree one and two terminate in the upper-right corner, i.e. the $(1, 1)$ point. However, with natural labeling, when the input *a priori* information is perfect, the output extrinsic information is only about $7/8$. This means that even when all other bits are perfectly known, there is still some uncertainty about the current bit. The reason for this behavior can be seen in the example trellis of Fig. 5. Suppose that the decoder wants to make a decision on the first bit and it knows the identity of the last two bits and that the starting and ending states are zero. With the natural labeling, the last two bits are both labeled by "00" and thus knowing their identity does not uniquely determine the value of the first bit. However, with the Gray labeling, the last two bits have different labels ("00' and "10") and so knowing their values is enough to discriminate the first bit.

When the inner-code's EXIT curve does not terminate in the upper-right corner, the optimization technique described in the previous section will not work for the class of codes considered because $I_A^{(i)}(I_i) < I_E^{(o)}(I_i)$ will not be satisfied as I_i approaches unity. In addition to violating the linear-programming constraint, the code will have a high error floor due to the early crossing of the inner-code and outer-code EXIT curves. To prevent these issues, the symbol labeling should be chosen to ensure that the inner-code EXIT curve extends to the $(1, 1)$ point. Thus, for the example of $M = 8$ and $h = 1/4$, Gray labeling is more desirable than natural labeling. However, Gray labeling is not universally preferred. For instance, we found that when $M = 4$ and $h = 1/3$ that natural labeling is better than Gray lableing. As a general rule, a suitable labeling is one that ensures that parallel transition pairs are labeled with at least two distinct bits. This rule is sufficient and necessary to force the inner-code EXIT curve to the $(1, 1)$ point.

C. Interleaver Design Issues

Because the system may be viewed from the perspective of having an outer nonsystematic LDGM code [13], the interleaver must be designed to avoid a many-to-one mapping of message sequences $\mathbf{u}$ to codewords $\mathbf{b}$. Such a mapping could occur, for instance, when a variable node is connected to a single check node by an even number of parallel edges. In such a case, the bit associated with the variable node will be added an even number of times, always resulting in a value of zero being passed to the check node. A similar problem occurs when a pair of degree-2 variable nodes is connected to a pair of check nodes in a "butterfly" fashion. If the two variable nodes are associated with the same bit value, then a value of zero will always be passed to both check nodes regardless of whether both bits were zeros or both were ones. Similarly, if the two variable nodes are associated with different bit values, then a value of one will be passed to both check nodes regardless of which variable node was set to one and which was set to zero.

In light of these examples, the likelihood of randomly drawing an interleaver with a many-to-one mapping can be greatly diminished by not allowing degree-2 variable nodes [17]. While higher-degree variable nodes could result in a many-to-one mapping, the probability of randomly drawing a many-to-one design with degree-3 nodes or higher decreases with order $N_c^{-\alpha}, \alpha \geq 1$, and is therefore inconsequential for sufficiently large N_c. However, degree-2 variable nodes can be allowed as long as the interleaver is designed to avoid many-to-one mappings. For instance, one good design procedure is to associate all the degree-two variable nodes with degree-one check nodes, similar to the doping method of system I in [13]. However, this procedure requires a large ρ_1, which leads to high SNR convergence threshold. In our design methodology, degree-two variable nodes are permitted, and ρ_1 is set to a very small value just for the purpose of decoding initialization. The interleaver is designed by first randomly linking the $2N_u\tilde{\lambda}_2$ outputs of the degree-two variable nodes to distinct check nodes. Afterwards, the edges of the variable nodes of degree higher than two are placed at random. Since there must be at least $2N_u\tilde{\lambda}_2$ check nodes, this requires $2N_u\tilde{\lambda}_2 \leq N_b$ which implies that $\tilde{\lambda}_2 \leq 1/(2r)$. Generally, this constraint is not restrictive except at very high code rates and was easily satisfied for all the designs presented in this paper.

A guideline given in [12] is that when the check nodes have degrees of either one or d_c, then the doping rate $\tilde{\rho}_1$ should satisfy

$$\tilde{\rho}_1 \quad \geq \quad \frac{d_c - 1}{d_v + d_c - 1}. \tag{38}$$

This bound is suitable if only one iteration is performed. However, if multiple iterations are permitted, $\tilde{\rho}_1$ does not have to satisfy this criteria. This is illustrated by the interleaver shown in Fig. 7 which serves as a counter-example. In the counter-example, all variable nodes have degree two and the check nodes may have degree one or two. The bound (38) would imply that at least $1/3$ of the check nodes should have degree one in order to permit successful decoding. However, in Fig. 7, there are only two check nodes of degree one, and therefore $\tilde{\rho}_1$ could be quite small. During each iteration of decoding, the information from the degree-one nodes at the top and bottom will propagate towards the center of the graph. Thus, the system is decodable given a sufficient number of iterations, despite not adhering to to (38).

This result also implies that with a large maximum number of iterations allowed, $\tilde{\rho}_1$ could be made to a very small number. In this paper, our code designs are all based on the check nodes with degree distribution $\rho_1 = 0.001$ and $\rho_2 = 0.999$, because a fairly early waterfall can be achieved using this distribution, while the decoding is still manageable in 200 iterations.

VII. Optimization and Simulation Results

Code optimizations were performed for the three systems listed in Table I which achieve spectral efficiency $\eta = 0.5$ bps/Hz using $M = \{2, 4, 8\}$. In each case, the inner-code's degree distribution was set to $\rho_1 = 0.001$ and $\rho_2 = 0.999$. A very small number of degree-one check nodes are needed to allow the iterative decoding process to start properly. Otherwise, the decoding process always stays at the origin of the EXIT chart [12]. Setting a smaller ρ_1 could help the decoder converge in fewer iterations, but the required $\mathcal{E}_b/N_0$ tends to be higher.

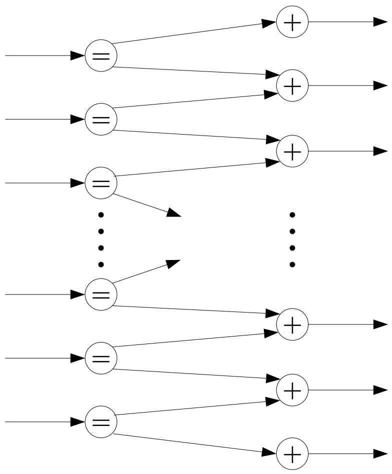

Fig. 7. Counter-example showing that the bound given by (38) is not necessary when there is more than one decoding iteration.

Having fixed the inner-code's degree distribution, the outer-code's degree distribution was found using linear programming under the constraint that the maximum outer-code degree is $d_v = 20$. Gray labeling was used for the $M = 8$ system, and natural labeling for the $M = 4$ system. EXIT curves for the optimized system with parameters $M = 8$, $h = 1/4$, and $r = 0.4458$ are shown in Fig. 8. From the EXIT curves, the convergence threshold is found to be $\mathcal{E}_b/N_0 = -0.19$ dB. The convergence thresholds and optimized degree distributions are as shown in the table for each of the three systems. Fig. 9 shows BER curves for all three systems after 100 and 200 decoder iterations with $N_u = 100,000$ message bits. Using the results from the figure, Table I shows the $\mathcal{E}_b/N_0$ required for the actual coded system to achieve a simulated bit error probability of 10^{-5} after 200 decoding iterations are performed. For all three systems, the simulation results are about 0.4 dB from the capacity limit.

VIII. Conclusion

Before engaging in the design of capacity-approaching codes for CPFSK modulation, it is useful to compute the information-theoretic limits for a given alphabet size M and modulation index h. This computation is facilitated by treating CPFSK over AWGN channels as a finite-state Markov channel and then computing the symmetric information rate using a BCJR-like algorithm. In addition to serving as a benchmark to measure the effectiveness of actual coded systems, the information-theoretic analysis provides useful insight into the optimal selection of the parameters M and h. This selection is especially important when bandwidth is constrained, for there will be a lower limit on the allowable code rate r that depends on the choice of M and h. Usually, complexity concerns require that $h = P/Q$ be rational with a small denominator Q. Thus, for any particular spectral efficiency, alphabet size M, and complexity limit, there will be an optimal combination

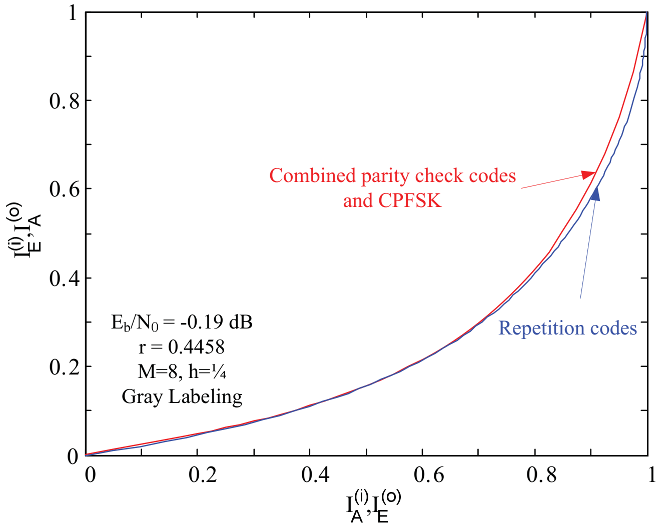

Fig. 8. EXIT chart for the optimized system with $M = 8$, $h = 1/4$, and Gray labeling.

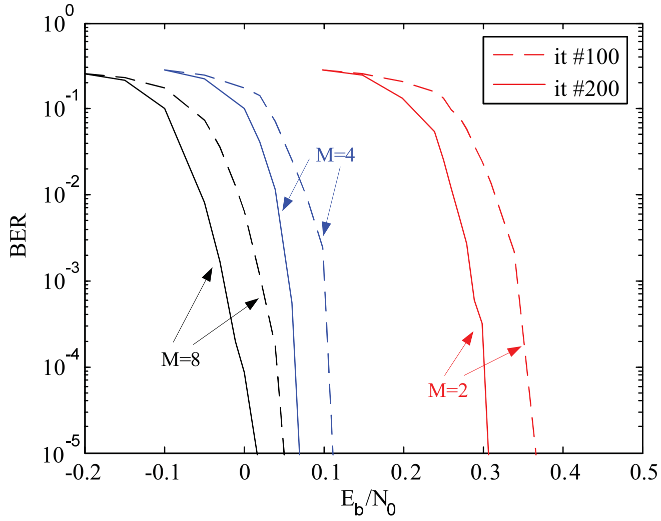

Fig. 9. BER of the three optimized systems listed in Table I. The message length is $N_u = 100,000$ bits. For each system, performance after 100 and 200 decoder iterations is shown.

of r and h that can be found through information-theoretic analysis.

Once the system parameters and information-theoretic limits are determined, the next step in the system design is to optimize the code, which may be done with the aid of the EXIT chart. First the EXIT curve for the inner code is drawn for a particular target-channel SNR, where the inner code is implemented by combining check nodes with the CPFSK modulator. The outer-code degree distribution is determined through linear programming with the objective of minimizing the area between the inner-code and outer-code EXIT curves. The optimal design is the one that minimizes this area at the lowest channel SNR without allowing the two curves to cross. Results show that this threshold SNR is between 0.1 and 0.2 dB from the value predicted by the corresponding capacity limit. Certain care must be taken to avoid bad interleavers and symbol labelings. Simulation results using the actual coded system achieve a BER of 10^{-5} at only about 0.4 dB from the capacity with a message length of 10^5 and 200 decoder iterations. Because the system performed remarkably close to capacity, we made no particular attempt to optimize the inner code. The whole process could be repeated for different inner code designs, which could result in a design that is even closer to the corresponding capacity limits.

References

[1] J. B. Anderson, T. Aulin, and C. E. Sundberg, *Digital Phase Modulation (Applications of Communications Theory)*. Springer, 1986.

[2] S. Shamai, L. H. Ozarow, and A. D. Wyner, “Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs,” *IEEE Trans. Inform. Theory*, vol. 37, pp. 1527–1539, Nov. 1991.

[3] H. Pfister, J. Soriaga, and P. Siegel, “On the achievable information rates of finite state ISI channels,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, San Anotonio, TX, Nov. 2001.

[4] D. Arnold, H.-A. Loeliger, P. Vontobel, A. Kavcic, and W. Zeng, “Simulation-based computation of information rates for channels with memory,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 3498–3508, Aug. 2006.

[5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[6] A. Ganesan, “Capacity estimation and code design principles for continuous phase modulation (CPM),” Master’s thesis, Texas A&M University, College Station, TX, May 2003.

[7] K. Padmanabhan, S. Ranganathan, S. P. Sundaravaradham, and O. M. Collins, “General CPM and its capacity,” in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Adelaide, Australia, pp. 750–754, Sept. 2005.

[8] P. Moqvist and T. Aulin, “Serially concatenated continuous phase modulation with iterative decoding,” *IEEE Trans. Commun.*, vol. 49, pp. 1901–1915, Nov. 2001.

[9] K. R. Narayanan and G. L. Stüber, “Performance of trellis-coded CPM with iterative demodulation and decoding,” *IEEE Trans. Commun.*, vol. 49, pp. 676–687, Apr. 2001.

[10] M. Xiao and T. Aulin, “Serially concatenated continuous phase modulation with convolutional codes over rings,” *IEEE Trans. Commun.*, vol. 54, pp. 1387–1396, Aug. 2006.

[11] K. Narayanan, I. Altunbaş, and R. Narayanaswami, “Design of serial concatenated MSK schemes based on density evolution,” *IEEE Trans. Commun.*, vol. 51, pp. 1283–1295, Aug. 2003.

[12] M. Xiao and T. Aulin, “Irregular repeat continuous phase modulation,” *IEEE Commun. Lett.*, vol. 9, pp. 723–725, Aug. 2005.

[13] M. Xiao and T. Aulin, “On analysis and design of low density generator matrix codes for continuous phase modulation,” *IEEE Trans. Wireless Commun.*, vol. 6, pp. 3440–3449, Sept. 2007.

[14] H. Jin, A. Khandekar, and R. McEliece, “Irregular repeat-accumulate codes,” in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 1–8, Sept. 2000.

[15] A. Guillén i Fàbregas and A. Grant, “Capacity approaching codes for non-coherent orthogonal modulation,” *IEEE Trans. Wireless Commun.*, vol. 6, pp. 4004–4013, Nov. 2007.

[16] J. Garcia-Frias and W. Zhong, “Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix,” *IEEE Commun. Lett.*, vol. 7, pp. 266–268, June 2003.

[17] S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.

[18] A. Roumy, S. Guemghar, G. Caire, and S. Verdu, “Design methods for irregular repeat-accumulate codes,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 1711–1727, Aug. 2004.

[19] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the shannon limit,” *IEEE Commun. Lett.*, vol. 5, pp. 58–60, Feb. 2001.

[20] M. Ardakani and F. Kschischang, “A more accurate one-dimensional analysis and design of irregular LDPC codes,” *IEEE Trans. Commun.*, vol. 52, pp. 2106–2114, Dec. 2004.

[21] B. Rimoldi, “A decomposition approach to CPM,” *IEEE Trans. Inform. Theory*, vol. 34, pp. 260–270, Mar. 1988.

[22] D. Torrieri, *Principles of Spread-Spectrum Communication Systems*. Springer, 2005.

[23] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2004.

[24] S. Cheng, R. I. Seshadri, M. Valenti, and D. Torrieri, "The capacity of noncoherent continuous-phase frequency shift keying," in *Proc. Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, Mar. 2007.

[25] S. Benedetto, G. Montorsi, A. Perotti, and A. Tarable, "A pragmatic approach to coded continuous-phase modulation," in *Proc. Info. Theory and Applications Workshop*, San Diego, CA, pp. 36–40, Feb. 2007.

[26] R. I. Seshadri, S. Cheng, and M. Valenti, "The BICM capacity of coherent continuous-phase frequency shift keying," in *Proc. IEEE Vehicular Tech. Conf. (VTC)*, Baltimore, MD, Oct. 2007.

[27] J. Proakis, *Digital Communications*. New York: McGraw-Hill, Inc., 4th ed., 2001.

[28] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[29] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, Jan. 1997.

[30] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Processing*, vol. 51, pp. 2764–2772, Nov. 2003.

[31] M. Tuechler, "Design of serially concatenated systems depending on the block length," *IEEE Trans. Commun.*, vol. 52, pp. 209–218, Feb. 2004.

Shi Cheng received the B.E. and M.S. degrees in electrical engineering from Southeast University, Nanjing, China in 2000 and 2003 respectively, and the Ph.D. degree in electrical engineering from West Virginia Unversity, Morgantown, WV in 2007. He is currently a system research engineer in ArrayComm LLC, San Jose, CA. His research interests lie in the areas of information theory, coding theory, and communications signal processing.

Matthew C. Valenti has been with West Virginia University since 1999, where he is currently an Associate Professor in the Lane Department of Computer Science and Electrical Engineering. He holds BS and Ph.D. degrees in Electrical Engineering from Virginia Tech and a MS in Electrical Engineering from the Johns Hopkins University. From 1992 to 1995 he was an electronics engineer at the US Naval Research Laboratory. He serves as an associate editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and has served as a track co-chair for the Fall 2007 Vehicular Technology Conference (Baltimore, Maryland) and the 2009 International Conference on Communications (Dresden, Germany). His research interests are in the areas of communication theory, error correction coding, applied information theory, wireless networks, simulation, and grid computing. His research is been funded by the National Research Foundation and the Department of Defense.

Don Torrieri is a research engineer and Fellow of the US Army Research Laboratory. His primary research interests are communication systems, adaptive arrays, and signal processing. He received the Ph. D. degree from the University of Maryland. He is the author of many articles and several books including Principles of Spread-Spectrum Communication Systems (Springer, 2005). He teaches graduate courses at Johns Hopkins University and has taught many short courses. In 2004, he received the Military Communications Conference achievement award for sustained contributions to the field.