

Fast Performance Estimation of Block Codes

Rajan Srinivasan, *Fellow, IEEE*, and Nuo Wang

Abstract—Importance sampling is used in this paper to address the classical yet important problem of performance estimation of block codes. Simulation distributions that comprise discrete- and continuous-mixture probability densities are motivated and used for this application. These mixtures are employed in concert with the so-called *g-method*, which is a conditional importance sampling technique that more effectively exploits knowledge of underlying input distributions. For performance estimation, the emphasis is on bit by bit maximum *a-posteriori* probability decoding, but message passing algorithms for certain codes have also been investigated. Considered here are single parity check codes, multidimensional product codes, and briefly, low-density parity-check codes. Several error rate results are presented for these various codes, together with performances of the simulation techniques.

Index Terms—Block codes, error rates, Monte Carlo simulation, importance sampling, mixture densities.

I. INTRODUCTION

THIS paper is yet another attempt at estimating error rates of block coded communication systems using importance sampling. The problem has been addressed by several authors (see [1], [2], and references therein) with varying success. Importance sampling (IS) has been employed in various fields, in many cases with remarkable results. It has not yielded quite the same benefits in the study of coded systems. Part of the reason presumably lies in the fact that error rate estimation involves evaluation (by simulation) of high-dimensional integrals over regions that are very complicated. It has rendered the task of constructing good simulation distributions difficult.

The approach to IS taken here is motivated by bitwise maximum *a-posteriori* probability (MAP) decoding algorithms, which are based on distance properties and are optimum in terms of error rate performance. Bit decoding algorithms (such as the BCJR algorithm of [3]) have received less attention in the past (compared to maximum-likelihood word decoding, [4]) as they have had a relatively high level of decoding complexity. They have attracted more attention recently, mainly due to the fact that suboptimal decoding algorithms have been developed which have low implementation complexity, [5]. Examples are the class of iterative decoding algorithms such as message passing which are based on the concept of

belief propagation. It is noted that message passing algorithms provide exact MAP decoding performance for codes that have Tanner graphs which are cycle-free, [9]. The codes considered here are examples of single parity check (SPC), Hamming, and low density parity check (LDPC) codes. Of these, only the SPC codes have graphs that are cycle-free. Although bit decoding algorithms can have error-rate performances close to ML decoding, it is clear that the collection of decoded bits may not constitute a valid codeword. This paper concentrates on estimating error rates for bitwise decoding.

An examination of the geometry of signal spaces generated by transmission of codes indicates that simulation distributions that comprise mixtures of different distributions could be ‘good’ for error rate estimation. They indeed are, as borne out by results reported herein. Though subjective, goodness here means achieving simulation gains (over straight Monte Carlo (MC) procedures) that are close to or more than $1/P_e$, where P_e is the error rate being estimated. In essence, this implies that fewer than a hundred simulation trials are sufficient to provide absolute accuracies within 20% with 95% confidence, for any P_e . Mixture distributions for fast simulation were (first) suggested in [8] and have been studied more recently in greater detail in [2], for the general problem of estimating small probabilities. They were suggested from the perspective of performing IS by exploiting certain large deviation properties of rare-event probabilities. In the latter reference it was shown that certain mixtures are *efficient*, in the sense of maximizing the asymptotic rate (as noise becomes smaller) of decay of variance. The approach taken here is to try to devise procedures that attempt to minimize error variances of IS estimators. Mixture-density simulation estimators have the appealing intuition that they can, with a single distribution, ‘scan’ (boundaries of) error regions effectively with economy of simulation effort. It is shown here that they emerge quite simply from a (virtual) partitioning of rare-event regions that are natural in an application. Some properties of mixture simulation distributions are given.

The principal results of this paper comprise: a proof that mixture simulation IS densities outperform the usual procedure of partitioning the codeword error region into a set of sub-error events and estimating each separately, a demonstration that combining the so called *g-method* ([11] and [6]) with mixtures can provide gains over conventional MC simulation exceeding $1/P_e$ for bitwise MAP decoding error rates when the codebook is known, and the realization that further research needs to be carried out to develop equally effective IS techniques for evaluating iterative decoding algorithms for long codes. The next section provides a straightforward account of bit by bit MAP decoding leading to formulation of the *g-method* technique of estimation. Section III motivates

Paper approved by S. G. Wilson, the Editor for Coding Theory and Applications of the IEEE Communications Society. Manuscript received December 2, 2004; revised January 15, 2006. This paper is an augmented and enlarged version of results presented (see [10]) at ICCAS05.

R. Srinivasan is with the Institute of Communications Engineering and the Electrical Engineering Department, National Sun Yat-sen University, 70 Lien-hai Road, Kaohsiung 80424, Taiwan (ROC) (e-mail: srinivasan@faculty.nsysu.edu.tw).

N. Wang is with the Systems Group, Neocific Inc., Chengkuan Electronics (Shanghai) Co., Ltd., Zuchongzhi Road No. 887, Building 84-402, Pudong district, Shanghai, P.R. China 201203 (e-mail: nwang@neocific.com).

Digital Object Identifier 10.1109/TCOMM.2008.040674.

mixture distributions for IS and describes their construction for uniform signal constellations. Application to block codes and results are in Section IV, followed by a concluding section.

II. ERROR PROBABILITY FOR BITWISE MAP DECODING

Consider a binary linear block code $C(n, k)$ with a codeword $\mathbf{c} = (c_1, c_2, \dots, c_i, \dots, c_n)$ of length n . Using binary phase-shift keying, the mapping to channel symbols is

$$\begin{aligned} \mathbf{y} &= (y_1, y_2, \dots, y_i, \dots, y_n) \\ &= (m(c_1), m(c_2), \dots, m(c_i), \dots, m(c_n)) \\ &\triangleq \mathbf{m}(\mathbf{c}) \end{aligned}$$

where $m(0) = -1$ and $m(1) = 1$. The vector \mathbf{r} received through an additive white Gaussian noise channel is

$$\begin{aligned} \mathbf{r} &= (r_1, r_2, \dots, r_i, \dots, r_n) \\ &= \mathbf{y} + \mathbf{n} \end{aligned}$$

where

$$r_i = y_i + n_i = m(c_i) + n_i, \quad i = 1, \dots, n$$

and $n_i \sim \mathcal{N}(0, \sigma^2)$ where $\sigma^2 = N_0/2$ and N_0 W/Hz is the one-sided noise spectral height. This represents a geometrically uniform signal constellation in n -dimensional Euclidean space \mathbf{R}^n . The noise samples $\{n_i\}$ are assumed independent. Using a MAP rule, the j^{th} bit is to be decoded based on observing all n received samples. The code is partitioned into a subcode C_j^0 and its coset C_j^1 , according to values of the bits in the j^{th} position, as

$$C_j^0 = \{\mathbf{c} \in C : c_j = 0\} \quad \& \quad C_j^1 = \{\mathbf{c} \in C : c_j = 1\}$$

The likelihood function of the received vector can be written as

$$p(\mathbf{r}|\mathbf{c}) = (2\pi\sigma^2)^{-n/2} e^{-\|\mathbf{r}-\mathbf{m}(\mathbf{c})\|^2/2\sigma^2}$$

Assuming transmitted sequences of independent and equally probable bits, the ratio of posterior probabilities for bit c_j is equal to its corresponding likelihood ratio LR . This is given by

$$\begin{aligned} LR(c_j) &= \frac{p(\mathbf{r}|c_j = 1)}{p(\mathbf{r}|c_j = 0)} \\ &= \frac{\sum_{\mathbf{c} \in C_j^1} e^{-\|\mathbf{r}-\mathbf{m}(\mathbf{c})\|^2/2\sigma^2}}{\sum_{\mathbf{c} \in C_j^0} e^{-\|\mathbf{r}-\mathbf{m}(\mathbf{c})\|^2/2\sigma^2}} \end{aligned}$$

For linear block codes in symmetric memoryless channels, the pairwise error probability is independent of the transmitted codeword. It is assumed the all-zero codeword ($c_i = 0$ for all i) is transmitted and thus C_j^1 is the set of all error codewords. The probability of decoding error for bit c_j can then be written as

$$\begin{aligned} P_e(j) &= P(LR(c_j) \geq 1 | y_j = -1) \\ &= P(r_j \geq \eta(\mathbf{r}_j)) \end{aligned} \quad (1)$$

after some algebra, where

$$\eta(\mathbf{r}_j) = \frac{\sigma^2}{2} \log \frac{\sum_{\mathbf{c} \in C_j^0} e^{\sum_{i=1, i \neq j}^n r_i m(c_i)/\sigma^2}}{\sum_{\mathbf{c} \in C_j^1} e^{\sum_{i=1, i \neq j}^n r_i m(c_i)/\sigma^2}}$$

with $\mathbf{r}_j \equiv (r_1, \dots, r_{j-1}, r_{j+1}, \dots, r_n)$, and each channel output r_i in the above equation is $-1 + n_i$. Further, we can write (1) as

$$\begin{aligned} P_e(j) &= E\{P(r_j \geq \eta(\mathbf{r}_j) | \mathbf{r}_j)\} \\ &\triangleq E\{g(\mathbf{r}_j)\} \end{aligned} \quad (2)$$

where the expectation is over the distribution of \mathbf{r}_j (or equivalently \mathbf{n}_j , which is similarly defined). This is an $(n-1)$ -fold integral where n can be large for long codes. Clearly, the conditional probability g -function is

$$g(\mathbf{r}_j) = \frac{1}{2} \text{erfc}[1 + \eta(\mathbf{r}_j)/(\sigma\sqrt{2})] \quad (3)$$

where erfc is the standard complement of error function. Calculation of bit error probability is then equivalent to evaluating the expectation of this g -function, which we proceed to estimate using fast simulation based on IS.

III. FAST ESTIMATION USING IS

An unbiased and consistent estimator of P_e in (2) is described by

$$\widehat{P}_e(j) = \frac{1}{K} \sum_{i=1}^K [g(\mathbf{r}_j) W(\mathbf{r}_j)]^{(i)}; \quad \mathbf{r}_j \sim f_{\star n-1} \quad (4)$$

where $W = f/f_{\star n-1}$ is a likelihood ratio between the density f of \mathbf{r}_j and a simulation density f_{\star} for it. We have to therefore bias the $(n-1)$ -vector \mathbf{r}_j . A good simulation distribution is one which provides the estimator with low variance and thus proceeds with a small simulation length K . The reason why this conditional IS procedure, initially introduced as the *g-method* in [11], is used is that it can sometimes provide enormous simulation gains compared to the usual IS estimator (given in (5) below). A proof, in the context of this paper, that the method yields lower variance estimates is given in Appendix A for convenience of the reader. It is a slight modification and generalization of the original proof given in [11]. A more complete generalization for estimators in an arbitrary rare-event setting can be found in [6].

A. On mixture distributions

In the form for P_e and its estimate adopted in (2) and (4), the error region in \mathbf{R}^n (referred to as \mathcal{E} hereafter with $P(\mathcal{E}) = P_e(j)$) contained in (1) is no longer explicit. Nevertheless its knowledge is helpful in designing good simulation distributions for the estimator of (4). As well known to analysts of coded systems, error sets have complicated shapes. This is the principal reason why evaluating $P(\mathcal{E})$ by analytical means is hard, even for low-length codes. Consider now the

usual IS estimator of any rare-event probability $P(\mathcal{E})$ which, in general, can be expressed as

$$\widehat{P}(\mathcal{E}) = \frac{1}{K} \sum_1^K 1(\mathcal{E}) W(\mathbf{r}); \quad \mathbf{r} \sim f_{*n} \quad (5)$$

A basic and well known fact of IS is that the perfect (or *clairvoyant*) simulation density for this estimator is, in present notation, given by

$$f_{*n}^{\text{opt}}(\mathbf{r}) = \frac{1}{P(\mathcal{E})} 1(\mathcal{E}) f(\mathbf{r}) \quad (6)$$

Let $\mathcal{E} = \bigcup_1^M \mathcal{E}_i$ denote a partition of the rare-event region into M disjoint regions. The optimal density then takes the form

$$\begin{aligned} f_{*}^{\text{opt}}(\mathbf{r}) &= \frac{1}{P(\mathcal{E})} 1\left(\bigcup_1^M \mathcal{E}_i\right) f(\mathbf{r}) \\ &= \frac{1}{P(\mathcal{E})} \sum_1^M 1(\mathcal{E}_i) f(\mathbf{r}) \\ &= \sum_1^M q_i f_{*i}^{\text{opt}}(\mathbf{r}) \end{aligned} \quad (7)$$

where

$$f_{*i}^{\text{opt}}(\mathbf{r}) \equiv 1(\mathcal{E}_i) f(\mathbf{r})/p_i$$

and $q_i \equiv p_i/P(\mathcal{E})$ are weighting probabilities with $p_i = P(\mathcal{E}_i)$. That is, f_{*}^{opt} is a convex combination of densities and represents a (discrete-) mixture distribution using optimum mixing probabilities $\{q_i\}$ as defined above. From the result stated in Appendix A, it follows that marginalizing (7) by removing the variable r_j yields an M -component mixture which will be the perfect simulation density, denoted by f_{*g}^{opt} , for the estimator in (4). That is

$$\begin{aligned} \int f_{*}^{\text{opt}}(\mathbf{r}) dr_j &= \frac{1}{P(\mathcal{E})} \int \sum_{i=1}^M 1(\mathcal{E}_i) f(\mathbf{r}) dr_j \\ &= \frac{1}{P(\mathcal{E})} \int 1(\mathcal{E}) f(r_j | \mathbf{r}_j) dr_j \cdot f(\mathbf{r}_j) \\ &= \frac{1}{P(\mathcal{E})} P(r_j \geq \eta(\mathbf{r}_j) | \mathbf{r}_j) f(\mathbf{r}_j) \\ &= \frac{1}{P(\mathcal{E})} g(\mathbf{r}_j) f(\mathbf{r}_j) \quad \text{from (2)} \\ &= f_{*g}^{\text{opt}}(\mathbf{r}_j) \end{aligned}$$

the last line being a result similar to (6) (see [12], page 5).

Although the perfect density is an abstraction and the above construction somewhat artificial, its form suggests that one could seek mixture distributions for use in simulation. This is in fact better, in terms of estimator variance and therefore IS gain, than performing M independent simulations (of different lengths) and adding up the weighted results. The latter is a procedure that has been adopted in the literature, [1], and is illustration of a ‘divide and conquer’ approach. That mixtures are better is quite easily shown, in Appendix B. Another advantage of the mixture-density estimator lies in its implementation. A partition of \mathcal{E} need not be simulated, as would be the case when using multiple estimators. In the ideal case above, the size and shape of the partition of \mathcal{E} can be

arbitrary without affecting optimality. In a real application, one can only conjecture that the number of error codewords (or at least those that that dominate the error probability) should determine the number of component densities used. This and the choice of mixing probabilities will influence estimator performance.

B. Error regions in signal space

The received signal space generated by a transmitted code will contain $2^k - 1$ points corresponding to codewords different from the all-zeros codeword. Only a subset of these, belonging to C_j^1 , are the error codewords for MAP decoding of the j th bit. The weights q_i of the mixture in our construction decrease with the Hamming distances of the error codewords from the all-zero codeword. Hence regions of the partition containing signal points corresponding to codewords with large Hamming weight would be simulated rarely if the mixture distribution is obtained by a randomized combination. The surface of the region \mathcal{E} has bumps (or distortions) in the vicinity of points in the signal space corresponding to the error codewords, and is difficult to characterize. A brief review of the geometry of error codewords, however, is helpful to understand how simulation distributions can be constructed.

In uniform signal constellations, all codewords in signal space lie on the surface of a Euclidean ball or hypersphere in \mathbf{R}^n having radius \sqrt{n} and centered at the origin. Those codewords that have minimum Euclidean (and Hamming) distance to the all-zero codeword are also located on an n -dimensional hypersphere of radius $2\sqrt{d_{\min}}$ with the latter as centre. Here, d_{\min} denotes minimum Hamming distance. The intersection of these two hyperspheres given by

$$r_1^2 + r_2^2 \cdots + r_n^2 = n$$

and

$$(r_1 + 1)^2 + (r_2 + 1)^2 \cdots + (r_n + 1)^2 = 4d_{\min}$$

is an $(n - 1)$ -dimensional hypersphere (in \mathbf{R}^n) that can be written as the intersection

$$\{r_1^2 + \cdots + r_n^2 = n\} \cap \{r_1 + \cdots + r_n = 2d_{\min} - n\}$$

that is, with a hyperplane. All error codewords with minimum Hamming distance, including those with a 1 in the j th bit position, lie on this hypersphere. Sketch of a possible situation is shown in Figure 1, to be interpreted as the n -dimensional signal space. Assuming the first bit is to be MAP-decoded, the error codewords lie on the $(n - 2)$ -dimensional hypersphere in \mathbf{R}^n given by

$$\begin{cases} r_2^2 + \cdots + r_n^2 = n - 1 \\ r_2 + \cdots + r_n = 2d_{\min} - n - 1 \\ r_1 = 1 \end{cases} \quad (8)$$

This hypersphere has radius $2\sqrt{d_{\min}(1 - d_{\min}/(n - 1))}$ and centre at $r_i = 2d_{\min}/(n - 1) - 1$, $i = 1, \dots, n$. Error codewords with the next smallest Hamming distance also lie on a corresponding hypersphere, and so on.

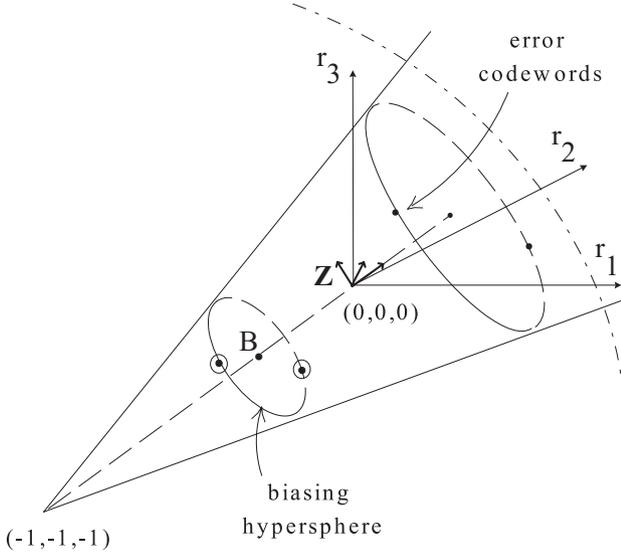


Fig. 1. Uniform signal constellation geometry and biasing.

1) *Discrete mixtures*: Suppose that the codebook is available and the dominating error codewords (i.e. those that have minimum and low Hamming distances and thus principally contribute to the error probability) can be identified in the signal space. The IS simulation mixture distribution is then constructed by moving the noise distribution (essentially halfway) towards the locations of the dominating codewords. In Figure 1 these are shown as circled dots on the ellipse labelled ‘biasing hypersphere’. Biasing by translation is an often used method in digital communication system simulation. Together with this, scaling of the noise variance was carried out in the applications below. The amounts of biasing (both translation and scaling) were optimized using adaptive algorithms. It turned out that in all cases only small increases in IS gain could be achieved with optimum biasing parameters lying close to the halfway translation and zero variance-scaling cases. Equal variance scaling was used for all noise variables. Mixing probabilities were estimated based on Hamming distance considerations, and uniform mixing carried out where symmetries exist in certain codes.

2) *Continuous mixture*: An alternative to discrete mixtures is the use of continuous mixtures for simulation. The advantage is that minimum and low Hamming distance codewords need not be determined. A continuous mixture (see Appendix B) is set up by projecting the noise distribution onto points on the surface of the biasing hypersphere whose radius and location are determined as described below. It is assumed that d_{\min} or an estimate of it is available. The procedure for obtaining the simulation distribution is described here in relation to the g -method estimator.

To remove the dependence of random variables imposed by (8), a Gram-Schmidt orthogonalization, $\mathbf{z}_1^T = \mathbf{G} \cdot \mathbf{r}_1^T$, is used to rotate the original coordinates to new ones, where $\mathbf{z}_1 \equiv (z_2, \dots, z_n)$ and \mathbf{G} is a normalized orthogonal matrix. In Figure 1 the rotated system is labelled as ‘Z’. The all-zero codeword in signal space becomes the point $\mathbf{z}_1 = (-\sqrt{n-1}, 0, \dots, 0)$ in the new system, assuming that we are talking in relation to the $(n-2)$ -dimensional hypersphere in

(8). The unbiased density of \mathbf{z}_1 is therefore

$$f(\mathbf{z}_1) = (\sqrt{2\pi}\sigma)^{1-n} e^{-[(z_2 + \sqrt{n-1})^2 - \sum_3^n z_i^2]/2\sigma^2} \quad (9)$$

The biasing hypersphere, in this rotated system, will then be (approximately) placed at center $(b, 0, \dots, 0)$ denoted by B in Figure 1, with radius c , where

$$\begin{aligned} b &\equiv \sqrt{n-1} (d_{\min}/(n-1) - 1) \\ c &\equiv \sqrt{d_{\min}[1 - d_{\min}/(n-1)]} \end{aligned}$$

Biased random variables are obtained by first selecting points randomly from (for simplicity) a uniform distribution on a hypersphere with radius c and centered at the origin. This is for the variables $\{z_i\}_3^n$ whereas z_2 is simply translated to B. Denoting this bias vector by $\mathbf{b}_1 = (d_{\min}/\sqrt{n-1}, b_3, \dots, b_n)$ and rotating back to the original transmitted signal coordinates yields

$$\mathbf{r}_1^T = -\mathbf{1}^T + \mathbf{G}^{-1} \cdot \mathbf{b}_1^T + \mathbf{n}_1^T$$

for the biased received vector. Using \mathbf{r}_1 , the g -function in (3) is calculated. Rotating again as $\mathbf{z}_1^T = \mathbf{G} \cdot \mathbf{r}_1^T$ yields the biased vector

$$\begin{aligned} \mathbf{z}_1^T &= -\mathbf{G} \cdot \mathbf{1}^T + \mathbf{b}_1^T + \mathbf{G} \cdot \mathbf{n}_1^T \\ &= (b + \nu_2, b_3 + \nu_3, \dots, b_n + \nu_n)^T \end{aligned}$$

in the rotated system and where $\{\nu_i\}_2^n$ are i. i. d. zero-mean Gaussian with variance σ^2 . The (biased) density of \mathbf{z}_1 is given by

$$\begin{aligned} f_*(\mathbf{z}_1) &= \frac{e^{-[(z_2-b)^2 + \sum_3^n z_i^2 + c^2]/2\sigma^2}}{(\sqrt{2\pi}\sigma)^{n-1}} \\ &\cdot \frac{\Gamma((n-2)/2)2^{(n-4)/2}}{c^{n-4}} \cdot \frac{I_{(n-4)/2}(c(\sum_3^n z_i^2)^{1/2}/\sigma^2)}{[(\sum_3^n z_i^2)^{1/2}/\sigma^2]^{(n-4)/2}} \end{aligned} \quad (10)$$

which follows in a straightforward manner by multiplying the density of z_2 with the formula given in [2] (with appropriate notation changes) for a Gaussian density translated randomly and uniformly onto the surface of a hypersphere. Ratio of the densities in (9) and (10) is the weighting function to be used in the estimator of (4). In the above, Γ and $I_{(\cdot)}$ denote the standard Gamma and modified Bessel functions respectively.

IV. APPLICATIONS

The codebook must be known (or computed) to use the estimator in (4) for estimating bit error probability of MAP decoding. If all or almost all of the dominating codewords are known for a specific code, then discrete mixture IS distributions can be easily constructed. Continuous mixtures can be used if only an estimate of d_{\min} is available. For long codes for which the codebook is not explicitly available or difficult to compute, the performance of message passing decoders can still be estimated using mixture distributions.

Effectiveness of IS simulations (and therefore, estimator precision) is measured by estimating sample-size gains over straight Monte Carlo to achieve the same variance. In the applications below, the simulation gain Γ is estimated as

$$\hat{\Gamma} = \frac{\widehat{P}_e(j) - \widehat{P}_e(j)^2}{\hat{I} - \widehat{P}_e(j)^2}$$

where $\widehat{P}_e(j)$ is given by (4), $I = E\{g^2(\mathbf{r}_j)W(\mathbf{r}_j)\}$, and

$$\hat{I} = \frac{1}{K} \sum_{i=1}^K [g^2(\mathbf{r}_j)W^2(\mathbf{r}_j)]^{(i)}; \quad \mathbf{r}_j \sim f_{*n-1}$$

A little algebra reveals that the estimated IS variance is given by

$$\begin{aligned} \widehat{\text{var}} \widehat{P}_e(j) &= \frac{\hat{I} - \widehat{P}_e(j)^2}{K} \\ &= \frac{1}{2K^3} \sum_{i=1}^K \sum_{\substack{m=1 \\ i \neq m}}^K [y_i(j) - y_m(j)]^2 \\ &> 0 \text{ with probability 1} \end{aligned} \quad (11)$$

where $y_i(j) \equiv [g(\mathbf{r}_j)W(\mathbf{r}_j)]^{(i)}$. To implement the error probability and gain estimators, an appropriate value of K is required. We have used the following empirical procedure (consisting of a few steps) to make this choice. When nothing is known about the target P_e , then an arbitrary initial value of K is used to generate \hat{I} for the chosen biasing scheme as a function of the biasing parameter. The aim is to estimate an initial parameter value for adaptive biasing algorithms that minimize \hat{I} . In cases where the computation load in calculating \hat{I} is intensive (as often happens in radar detection algorithms), then a rough optimum biasing parameter estimate can be obtained by using the same underlying random variables to make the calculations. Using this initial parameter estimate, $\hat{\Gamma}$ is estimated, the value of K is compared with $100/(\widehat{P}_e(j)\hat{\Gamma})$, and then K is altered suitably. This procedure needs to be usually carried out not more than 2 or 3 times to obtain reasonable choices of K and biasing parameter to adaptively fine-tune the IS simulation.

If the estimator in (5) is used instead of the g -estimator of (4), then the inequality in (11) is not strict. In such cases, as pointed out in [13], it is safer to overbias the simulation to ensure that the indicator function in the estimator is not always zero. In our simulations below with message passing decoders for long codes, estimates of d_{\min} have been used to construct mixture distributions.

These IS simulation ideas are used below for certain block codes. Both bitwise MAP and message passing decoding have been investigated. All the LDPC codes used in this paper can be found on the website of MacKay (<http://www.inference.phy.cam.ac.uk/mackay/>).

A. Locating dominating codewords

Codewords with weight d_{\min} lie on a hypersphere. Suppose an estimate of the minimum Hamming distance d_{\min} (possibly based on bounds) for a code is available. Assume the all-zero codeword is input to a message-passing decoder. Then, moving the noise distribution randomly onto a hypersphere that uses the estimated d_{\min} would, presumably with high probability, cause the decoder to find minimum Hamming distance codewords satisfying the parity check condition for that code. All such codewords (and d_{\min}) could be found if this is done a sufficiently larger number of times. To increase the chances of locating desired codewords, the samples on the hypersphere are perturbed with some Gaussian noise and the

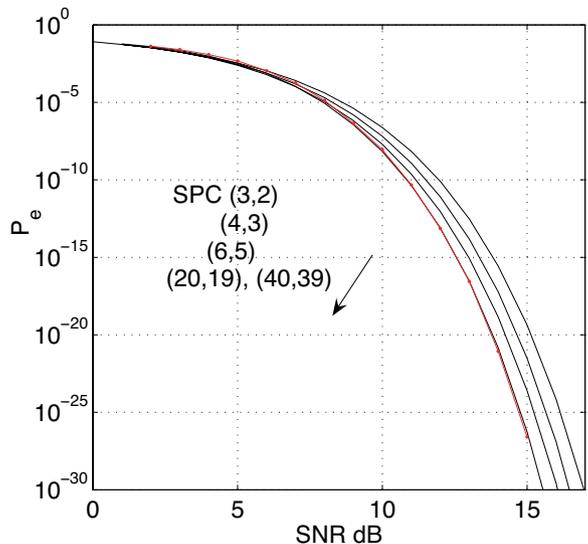


Fig. 2. Performance of bitwise MAP decoding of SPC codes.

TABLE I
ESTIMATED MINIMUM HAMMING DISTANCE AND MULTIPLICITY.

| Codes | Hamming distance | Multiplicity |
|----------------|------------------|--------------|
| LDPC(96,48) | 6 | 3 |
| LDPC(204,102) | 8 | 1 |
| LDPC(504,252) | 20 | 2 |
| LDPC(1008,504) | 34 | 1 |

variance is controlled. Carrying out this procedure with the next larger Hamming weight would locate codewords with that weight. In this manner, distance properties of a code can be determined. Some results of such experiments are given in Table I. To locate all the minimum distance codewords for the LDPC (96,48) code required 10^5 trials, whereas 10^8 trials were required for the LDPC (1008,504) code to locate one codeword. The numbers presented in this table are estimates, and should not be taken as guaranteed, especially in the last two rows. The smallest distances listed are probably d_{\min} .

B. Single parity-check codes

Single parity-check codes, denoted as $\text{SPC}(n, n-1)$, are simple codes with $d_{\min} = 2$, [7]. The parity-check condition is $c_1 + \dots + c_n = 0$. Message-passing decoding for this family of codes performs as well as MAP decoding. Both discrete- and continuous-mixture IS distributions have been used to simulate error rate performances. In Figure 2 is shown the bit error rate for these codes. The signal-to-noise ratio SNR in all figures is E_b/N_0 where E_b denotes transmitted energy per bit. Simulation performances of the two mixture distributions are compared in Figure 3 for various SPC codes. It is clear that discrete-mixture simulation distributions are superior to continuous-mixtures ones, and this is generally representative of all our investigations. The difference in performance can be appreciable. At error rates below 10^{-10} , discrete mixtures require IS simulation lengths that are at least a few hundred times smaller. It is evident that estimated simulation gains clearly exceed $1/P_e$. For example, all the results in Figure 2

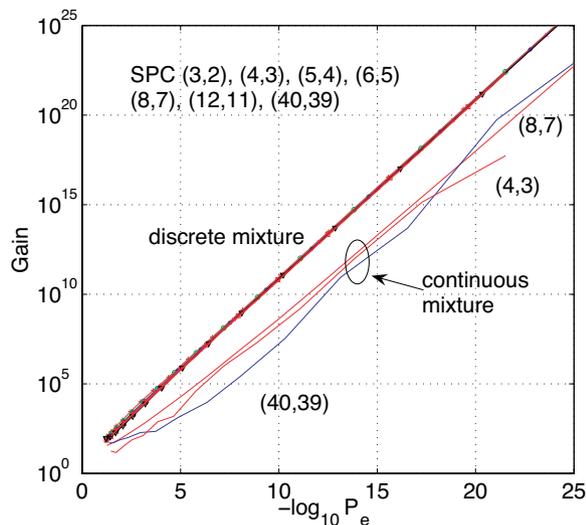


Fig. 3. Estimated simulation gains for SPC codes.

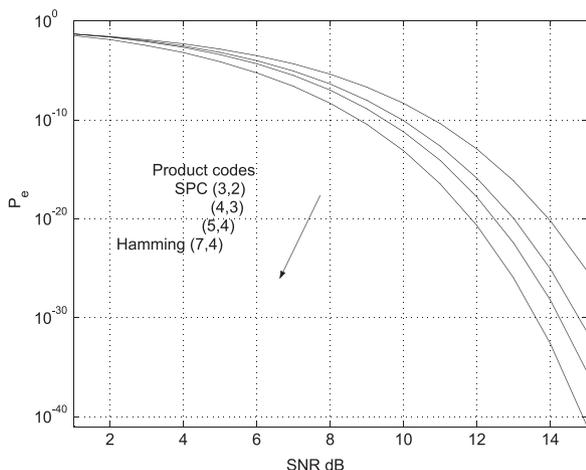


Fig. 4. Performance of (exact) MAP decoding for 2-dimensional product codes.

were generated using $K = 200$ IS trials or less, with discrete-mixture simulation.

C. Multidimensional product codes

The (multidimensional) product codes studied here consist of Hamming as well as SPC component codes. Results for exact MAP decoding are in Figures 4 and 5 for two-dimensional codes. For these codes, simulation gains are close to $1/P_e$.

Complexity of MAP decoding increases rapidly with code size. A lower complexity algorithm which is suboptimal is discussed in [7] and has been used here, with comparable results for low dimensions. When the dimension of the product code increases, the block length increases rapidly. This makes it difficult to employ discrete mixtures with benefit, especially at low SNR's. At higher SNR's, simulating with continuous mixtures gives some gains but discrete mixtures perform better and have been used, although gains are low. Results for an SPC(8, 7) 5-dimensional product code are shown in Figure 6 for each iteration of the decoding algorithm. For a bit error rate of 3×10^{-18} at 5dB, the gain is only 1.25×10^{15} at the 5th iteration.

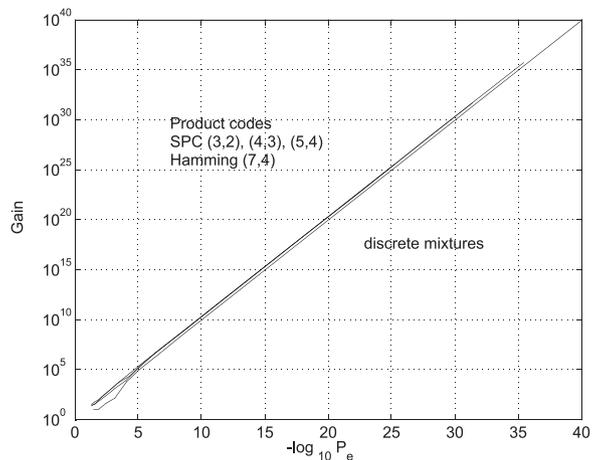


Fig. 5. Estimated simulation gains for 2-dimensional product codes.

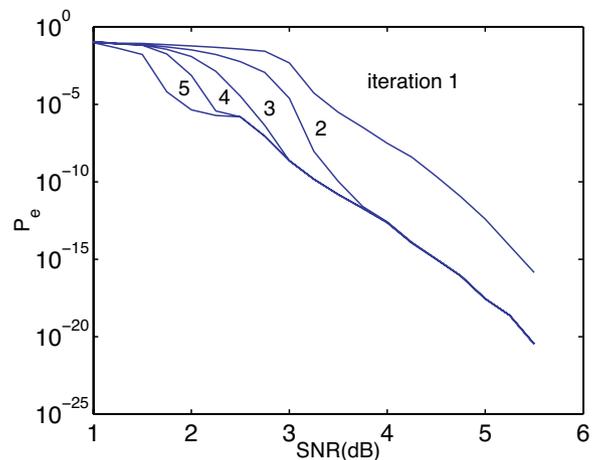


Fig. 6. Iteration-wise performance for 5-dimensional SPC(8, 7) product code.

D. Low-density parity-check codes

These codes employ message-passing decoding to reduce complexity. Their asymptotic (in code length) performance has been investigated thoroughly, [9]. The IS techniques described here have been applied to a MacKay(96,48) rate 1/2 regular LDPC code with 5 decoding iterations. Results are shown in Figures 7 and 8. At low SNR's, the continuous mixture simulation distribution has been used but gains are lower than obtained with discrete mixtures. At SNR's higher than 9dB, discrete mixtures have been used to speed up the simulations. Although the 5-dimensional SPC(8, 7) code has a rate of 0.513, close to that of LDPC (96, 48), the former has codeword length 32768. Hence it outperforms the LDPC code. It is clear that estimated simulation gains for the LDPC code fall far short of $1/P_e$.

V. CONCLUSIONS

Some headway has been made in this paper in devising fast simulation procedures for error rate analysis of block codes. From a theoretical standpoint it has been shown that mixture-density simulation has attractive properties. Although continuous mixtures are easier to implement for long codes than discrete mixtures, their IS performances are not as good.

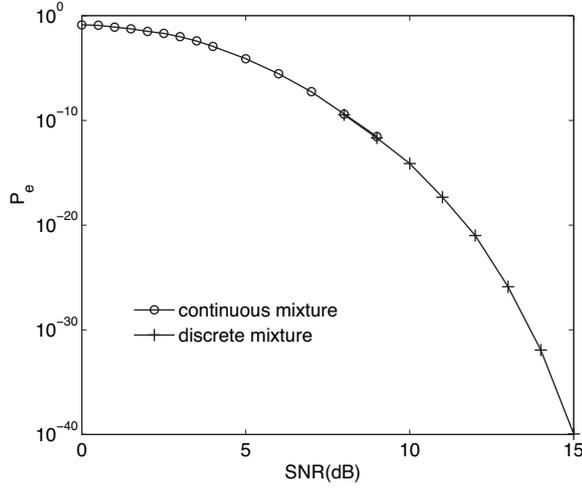


Fig. 7. Performance of LDPC(96,48) code.

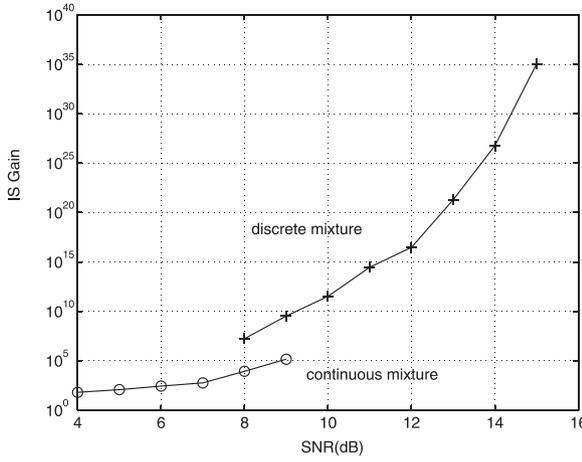


Fig. 8. Estimated simulation gains for LDPC(96,48) code.

Excellent results have been obtained for bitwise MAP decoding with discrete mixtures but the techniques are cumbersome to apply for very long codes. The main reason lies not as much in the IS techniques themselves as in the computational intensity involved in identifying dominating codewords and performing associated calculations required in the estimation algorithms. The latter difficulties are ameliorated to some extent for message passing decoding, but our IS techniques reveal shortcomings in terms of simulation gains achievable. It appears that better fast simulation techniques need to be developed to handle decoding algorithms that have low implementation complexity but involve operations that are relatively obdurate to mathematical analysis.

APPENDIX A

VARIANCE OF THE g -METHOD ESTIMATOR

The probability to be estimated is $p = P(X \geq Y)$ where X and Y have density $f(x, y)$. Two estimators are to be compared for variance. The usual one employs a simulation density f_* (other than f) with weighting function $W = f(x, y)/f_*(x, y)$ in the estimator

$$\frac{1}{K} \sum_1^K \mathbf{1}(X \geq Y) W(X, Y); \quad (X, Y) \sim f_*(x, y)$$

that has variance $(I - p^2)/K$, where $I = E\{\mathbf{1}(X \geq Y) W(X, Y)\}$. We show that, using the (marginal) simulation density

$$f_*(y) = \int f_*(x, y) dx$$

with weighting function $V = f(y)/f_*(y)$, the g -method estimator

$$\frac{1}{K} \sum_1^K g(Y) V(Y); \quad Y \sim f_*(y)$$

with $g(y) = P(X \geq Y | Y = y)$ and variance $(I_g - p^2)/K$ where $I_g = E\{g^2(Y) V(Y)\}$, has the property that $I_g \leq I$.

Define $W(x | y) = f(x | y)/f_*(x | y)$. We have

$$\begin{aligned} g(y) &= \int \mathbf{1}(x \geq y) f(x | y) dx \\ &= \int \mathbf{1}(x \geq y) W(x | y) f_*(x | y) dx \end{aligned}$$

Then

$$\begin{aligned} g^2(y) &\leq \int \mathbf{1}(x \geq y) W^2(x | y) f_*(x | y) dx \\ &= \int \mathbf{1}(x \geq y) W(x | y) f(x | y) dx \end{aligned}$$

by Jensen's inequality. Therefore

$$\begin{aligned} I_g &= \int g^2(y) V(y) f(y) dy \\ &\leq \iint \mathbf{1}(x \geq y) W(x | y) f(x | y) V(y) f(y) dx dy \\ &= \iint \mathbf{1}(x \geq y) W(x, y) f(x, y) dx dy \\ &= I \end{aligned}$$

APPENDIX B

MIXTURE SIMULATION DISTRIBUTIONS

The results given here are applicable to IS estimation of rare-event probabilities in general.

A. Discrete mixtures

For notational comfort, random vectors are suppressed in the following as manipulations on these are not needed. All expectations below are with respect to f . The probability $p = \sum_1^M p_i$, each $p_i > 0$, is to be estimated using M unbiased estimators given by

$$\hat{p}_i = \frac{1}{K_i} \sum_1^{K_i} \mathbf{1}_i W_i; \quad i = 1, \dots, M$$

where $\mathbf{1}_i = \mathbf{1}(\mathcal{E}_i)$ and $W_i = f/f_i$, as $\hat{p} = \sum \hat{p}_i$. The simulation lengths K_i satisfy $\sum K_i = K$, the total length. The variance of this estimator is

$$\text{var} \hat{p} = \sum_1^M \frac{1}{K_i} [E\{\mathbf{1}_i W_i\} - p_i^2]$$

This is to be compared with a single (unbiased) estimator, using the M -component mixture $f_{\text{mx}} = \sum q_i f_i$ with the same component densities as the multiple estimator above, given by

$$\hat{p}_{\text{mx}} = \frac{1}{K} \sum_1^K \mathbf{1}(\mathcal{E}) W_{\text{mx}}$$

where $W_{\text{mx}} = f/f_{\text{mx}}$, with variance

$$\begin{aligned} \text{var } \widehat{p}_{\text{mx}} &= \frac{1}{K} [E\{1(\mathcal{E})W_{\text{mx}}\} - p^2] \\ &= \sum_1^M E\{1_i W_{\text{mx}}/K\} - p^2/K \end{aligned}$$

It is easy to show that the variances of both estimators are strictly convex, respectively, in $\{K_i\}$ and $\{q_i\}$. In the case of the mixture estimator, for instance, with $V(q) \equiv E\{1(\mathcal{E})W_{\text{mx}}\}$ where $q \in \mathbf{R}^M$, it is sufficient to show that $h(t) \equiv V(r+ts)$ is strictly convex in t where $r, s \in \mathbf{R}^M$. Minimization of $V(q)$ over the probability simplex $\sum q_i = 1$ is not possible without specifying the density functions involved. For the multiple estimator, however, minimization of $\text{var } \widehat{p}$ leads to

$$K_i = \frac{a_i}{\sum_1^M a_i} K$$

where $a_i^2 = E\{1_i W_i\} - p_i^2$. This results in the minimum variance $(\sum a_i)^2/K$. When no biasing is used ($W_i = 1$), we have $a_i^2 = p_i - p_i^2$. The minimized variance can then be written as

$$\begin{aligned} \min \text{var } \widehat{p} &= \frac{1}{K} \left(\sum_{i=1}^M (p_i - p_i^2)^{1/2} \right)^2 \\ &\geq \frac{1}{K} \sum_{i=1}^M (p_i - p_i^2) \\ &= \frac{1}{K} \left(p - \sum_{i=1}^M p_i^2 \right) \\ &\geq \frac{1}{K} \left[p - \left(\sum_{i=1}^M p_i \right)^2 \right] \\ &= \frac{1}{K} (p - p^2) \end{aligned}$$

the last quantity being the variance of a straight K -length MC estimator of p . Multiple estimation (even with optimized simulation lengths) is therefore inferior to MC simulation. This is not surprising from an intuitive point of view.

With IS, a_i can always be written as $a_i^2 = c_i^2 p_i^2$ where each c_i depends on p_i and f_i such that $0 \leq c_i^2 \leq (1/p_i) - 1$. If each biasing density is good (in terms of reducing the associated IS variance), then c_i will be small. Suppose now that we degrade the estimator performance by setting each c_i equal to some constant c . Then we have

$$K_i = \frac{p_i}{p} K$$

and

$$\min \text{var } \widehat{p} \approx \frac{c^2 p^2}{K}$$

That is, simulation lengths are chosen in proportion to the probabilities being estimated. A similar result was obtained in [1]. Note that knowledge of the $\{p_i\}$ is still required to determine simulation lengths. In actual applications, it may be possible to roughly estimate these.

For comparing with mixture estimation when IS is used, we have the following.

Proposition: There exists a mixture estimator, using mixing

probabilities q_i in the same proportions as the lengths K_i of the multiple estimator above, with a variance that is always smaller.

Proof: Let $q_i = K_i/K = p_i/p$. Then $W_{\text{mx}} = Kf/\sum K_j f_j$ and

$$\begin{aligned} \frac{W_{\text{mx}}}{K} - \frac{W_i}{K_i} &= \frac{f}{\sum_{j=1}^M K_j f_j} - \frac{f}{K_i f_i} \\ &= \frac{f}{K_i f_i + \sum_{\substack{j=1 \\ j \neq i}}^M K_j f_j} - \frac{f}{K_i f_i} \\ &< 0 \end{aligned}$$

As $\sum p_i^2/K_i = p^2/K$, it follows that

$$\begin{aligned} \text{var } \widehat{p}_{\text{mx}} - \text{var } \widehat{p} &= \sum_1^M E\{1_i [W_{\text{mx}}/K - W_i/K_i]\} \\ &< 0 \end{aligned} \quad \square$$

B. Continuous mixtures

Mixing probabilities as given in the Proposition above can be taken to be nearly optimal but are impossible to determine exactly, as knowledge of the unknown p is needed. In some cases symmetry suggests using uniform probabilities. Another possibility is to use continuous mixtures for simulation, [2]. The advantage is that multiple component densities and mixing probabilities need not be set up. This is partially offset by poorer simulation performance that is obtained in some cases. The issue of determining optimal mixing probabilities is replaced by one of finding an appropriate mixing density. The optimal mixing density can be obtained as described below.

Let $f_*(x|\alpha)$ denote a conditional simulation density given a mixing random vector α with density $\nu(\alpha)$, where it is assumed for convenience that $x, \alpha \in \mathbf{R}^n$. Then

$$f_*(x) = \int f_*(x|\alpha) \nu(\alpha) d\alpha \quad (12)$$

In analogy with the discrete-mixture case, it follows that the probabilities $\nu(\alpha) d\alpha$ should be chosen in proportion to probabilities of an elemental partition of the error event region \mathcal{E} . Let $J(x|\alpha)$ denote the Jacobian of some one-to-one transformation. Then ν must satisfy

$$\nu(\alpha) = \frac{1(\mathcal{E})}{p} f(x) J(x|\alpha)$$

and is a nearly optimal mixing density in terms of minimizing IS variance for the choice of conditional simulation density $f_*(x|\alpha)$. There are thus, as for discrete mixtures, two optimization aspects to be addressed. Consider the following simple example.

Example. For $n = 2$, let $X = (X_1, X_2)$ where X_1 and X_2 are independent, zero-mean Gaussian with variances σ_1^2 and σ_2^2 . The region \mathcal{E} is $\{x_1^2 + x_2^2 \geq a^2\}$, whose probability p is to be estimated. The conditional simulation density is chosen as a random translation of $f(x)$ to points on the circle $x_1^2 + x_2^2 = b^2$. The density is therefore

$$f_*(x|b, \theta) = f(x_1 - b \cos \theta, x_2 - b \sin \theta)$$

An optimum choice for the radius b exists that minimizes IS variance, but this is not addressed here. Let $\alpha = (r, \theta)$ be the mixing random vector in polar coordinates. The mixing density can then be written as

$$\begin{aligned} \nu(r, \theta) &= \frac{1(\mathcal{E})}{p} r f(x_1, x_2) \\ &= \frac{1(r \geq a)}{p} \cdot \frac{r e^{-r^2 A(\theta)}}{2\pi\sigma_1\sigma_2} \end{aligned}$$

where $A(\theta) = \cos^2 \theta / 2\sigma_1^2 + \sin^2 \theta / 2\sigma_2^2$. It follows that

$$\nu(\theta) = \frac{e^{-a^2 A(\theta)}}{4\pi\sigma_1\sigma_2 p A(\theta)}, \quad 0 \leq \theta \leq 2\pi$$

For the special case of $\sigma_1 = \sigma_2$, this reduces to a uniform density and the corresponding simulation density is derived in [2]. From symmetry, this is exactly optimal in terms of minimizing IS variance. In general, the mixing density given above covers with higher probability those parts of the error region which dominate the probability p .

ACKNOWLEDGMENT

The authors wish to thank Harm Cronie for interesting discussions on LDPC codes. They are also grateful for suggestions from the anonymous reviewers, one of whom carried out a painstaking review.

REFERENCES

- [1] M. Ferrari and S. Bellini, "Importance sampling simulation of concatenated block codes," *IEE Proc. Commun.*, vol. 147, no. 5, pp. 245-251, Oct. 2000.
- [2] J. A. Bucklew and R. Radeke, "On the Monte Carlo simulation of digital communication systems in Gaussian noise," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 267-274, Feb. 2003.
- [3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, pp. 284-287, Mar. 1974.
- [4] S. G. Wilson, *Digital Modulation and Coding*. Upper Saddle River, NJ: Prentice Hall, 1996.
- [5] A. Abedi and A. K. Khandani, "Some properties of bit decoding algorithms over a generalised channel model," in *Proc. Conference on Information Sciences and Systems (CISS 2002)*, Princeton, NJ, pp. 112-117, Mar. 2002.
- [6] J. A. Bucklew, "Conditional importance sampling estimators," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 143-153, Jan. 2005.

- [7] D. M. Rankin and T. A. Gulliver, "Single parity check product codes," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1354-1362, Aug. 2001.
- [8] J. S. Sadowsky and J. A. Bucklew, "On large deviations theory and asymptotically efficient Monte Carlo estimation," *IEEE Trans. Inf. Theory*, vol. 36, no. 3, pp. 579-588, May 1990.
- [9] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599-618, Feb. 2001.
- [10] N. Wang and R. Srinivasan, "On importance sampling for iteratively decoded linear block codes," in *Proc. International Conference on Communications, Circuits and Systems (ICCCAS 2005)*, vol. 1, Hong Kong, China, May 27-30, 2005.
- [11] R. Srinivasan, "Some results in importance sampling and an application to detection," *Signal Processing*, vol. 65, no. 1, pp. 73-88, Feb. 1998.
- [12] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*. Berlin: Springer-Verlag, 2002.
- [13] J. A. Bucklew, *Introduction to Rare Event Simulation*. Springer, 2004.



Rajan Srinivasan received two degrees from the Indian Institute of Technology, Delhi, India, and a doctorate from Aston University, Birmingham, UK, all in electrical engineering. He is an educator and researcher, having worked in various academic institutions worldwide including some years spent as a senior research scientist in the Ministry of Defence of the Government of India. His principal interests are in statistical signal processing covering the fields of communications and radar. During his career he received a Young Scientist award from the

International Union of Radio Science, in Florence, Italy, for his work on distributed detection which resulted in seminal publications. More recently he wrote *Importance Sampling - Applications in Communications and Detection* the first monograph on the subject in the world, published by Springer-Verlag in 2002. A widely published author, Srinivasan is a Fellow of both the IEE and IEEE and till recently taught at the University of Twente in the Netherlands, carrying out research mainly focused on the development of fast simulation techniques for design and analysis of highly reliable processing systems. He is currently a professor in the Institute of Communications Engineering at the National Sun Yat-sen University in Taiwan.



Nuo Wang obtained a B.S. and a Ph.D. degree in electronic engineering from University of Science and Technology of China, Hefei, China, in 1999 and 2004. During 2003-2004 he spent a year on a Huygens Fellowship at the University of Twente, Enschede, The Netherlands. From 2005 to 2007 he was with the Semiconductor Division of Samsung Electronics, Suwon, Republic of Korea and is presently with the Systems Group at Neocific Inc., in Shanghai, China. His research interests in communication theory and applications include OFDM,

channel coding, and MIMO technologies.