Minimum-Variance Importance-Sampling Bernoulli Estimator for Fast Simulation of Linear Block Codes over Binary Symmetric Channels

Gianmarco Romano, Member, IEEE, and Domenico Ciuonzo, Student Member, IEEE

Abstract—In this paper the choice of the Bernoulli distribution as biased distribution for importance sampling (IS) Monte-Carlo (MC) simulation of linear block codes over binary symmetric channels (BSCs) is studied. Based on the analytical derivation of the optimal IS Bernoulli distribution, with explicit calculation of the variance of the corresponding IS estimator, two novel algorithms for fast-simulation of linear block codes are proposed. For sufficiently high signal-to-noise ratios (SNRs) one of the proposed algorithm is SNR-invariant, i.e. the IS estimator does not depend on the cross-over probability of the channel. Also, the proposed algorithms are shown to be suitable for the estimation of the error-correcting capability of the code and the decoder. Finally, the effectiveness of the algorithms is confirmed through simulation results in comparison to standard Monte Carlo method.

Index Terms—Binary symmetric channel (BSC), importance sampling (IS), linear block codes, Monte-Carlo simulation.

I. INTRODUCTION

T HE Monte-Carlo (MC) simulation is a general method to estimate performances of complex systems for which analytical solutions are not available or mathematically tractable and it is extensively used in the analysis and design of communications systems [1], [2]. The MC method has also been extensively employed to evaluate the performances of forward-error-correcting (FEC) codes with different decoding algorithms, in terms of probability of bit error (BER) or word error (WER), for which, in many cases, is not possible to obtain exact closed-form expressions [3]–[5]. In general an upper bound is available for any linear block code, however the error correcting capability of the code is required [4], [5]. The MC method is also used as verification tool in the design, development and implementation of decoding algorithms.

The computational complexity of the MC method is given by the number of generated random samples that are needed to obtain a reliable estimate of the parameters of interest. In the case of FEC codes, estimation of low BER or WER requires a high number of generated codewords to obtain results of acceptable or given accuracy, thus leading to prohibitive computational complexity. Furthermore, for very long codes the computational complexity is high even for small number of generated words, since the decoding complexity increases the simulation time considerably. A practical case is represented by low-density parity-check (LDPC) codes [6]– [8], for which it is crucial to examine the performances at very low probability of error in order to avoid error floors, i.e. the rate of decrease of the probability of error is not as high as at lower SNRs (i.e. in the waterfall region) [9], [10]. One of the impediments in the adoption of LDPC codes in fiberoptics communications, where the order of magnitude of the probability of error of interest is 10^{-12} and below, has been the inability to rule out the existence of such floors via analysis or simulations [11]. While for some LDPC codes it is possible to predict such floors, in many other cases the MC method is the only tool available. LDPC codes are also employed, for example, in nanoscale memories [12], where a majoritylogic decoder is chosen instead of soft iterative decoders as these may not be fast enough for error correction; therefore an efficient method to estimate the performances of hard-decision decoding at very low WERs is extremely desirable.

1

Several mathematical techniques have been proposed in the literature in order to reduce the computational complexity of the MC method and estimate low WERs with the same accuracy¹ [13]. Importance sampling (IS) is regarded as one of the most effective variance-reduction techniques and it is widely adopted to speed up simulation of rare events, i.e. events that occur with very low probability [14]. The idea is to increase the frequency of occurrence of rare events, by means of a biased distribution. The optimal biased IS distribution is known, but it cannot be used in practice since it depends on the parameter to be estimated itself. Therefore, a number of sub-optimal alternatives have been developed in the literature [13], [15]. Some of them are obtained by restricting the search of the biased distribution to a parametric family of simulation distributions; then the parameters are derived as minimizers of the estimator variance or other related metrics, such as the cross-entropy [14], [16]. The choice of the family of biased distribution is somewhat arbitrary and may depend on the specific application of the IS method [14]. In the case of FEC, the rare event corresponds to the decoding error and the IS method, in order to be effective, needs to generate more frequently the codewords that are likely to be erroneously decoded. The mathematical structure of the code, or some performance parameter of the code, such as the minimum distance and/or the number of correctable errors or, in the

The authors are with the Department of Industrial and Information Engineering, Second University of Naples, via Roma, 29, 81031 Aversa (CE), Italy. Email: {gianmarco.romano, domenico.ciuonzo}@unina2.it

¹One possibility to cope with the computational complexity of the MC method is to adopt more powerful hardware in order to reduce the generation and processing time of each codeword; this might constitute a practical solution to reduce the overall simulation time. Nevertheless, the increased system complexity requires more time per sample and compensates the reduction of execution time, thus limiting the achievable gain.

case of LDPCs, the minimum size of the absorbing sets in their Tanner graphs, may be taken into account to choose a good family [17], [18]. In [19], an SNR-invariant IS method is proposed, which, though independent of the minimum distance of the code, provides better estimates when the error-correcting capability of the decoder is available. In this paper we consider generic linear block codes and we do not make any assumption on specific parameter or structure of the code.

In this paper a specific problem is considered: (i) which is the best joint independent Bernoulli distribution that can be used as biased distribution for IS estimation of block linear code performances and (ii) what are the strengths and limitations of this solution. The choice of such family of distributions is arbitrary and it is motivated by the fact that the random generator required for the IS method is of the same type of that required in the standard MC method and hence is made because of its simplicity rather than taking into account the specific structure or properties of codes. On the other hand, since the study is restricted to the parametric family of the joint independent Bernoulli distributions, the gain in computational complexity that is obtained is limited by this choice, as sub-optimal IS distributions that lead to smaller IS estimator variance may exist.

Another performance measure for FEC codes is the minimum distance of the code and/or the error correcting capability of the code or decoder, i.e. the maximum number of errors that a specific couple (code, decoder) are able to correct. The minimum distance of codes can be estimated to overcome the computational complexity required by the exhaustive search, which increases exponentially with the length of the information word. In [20] the error impulse method is proposed for linear codes and is based on the properties of the error impulse response of the soft-in soft-out decoder and its error-correcting capability. Due to the sub-optimality of the iterative decoder employed with LDPC codes, the error impulse method can lead to wrong estimates of minimum distance. For this class of codes the method has been improved in [21] and [22]. More recently, integer programming methods have been used to calculate either the true minimum distance or an upper bound [23]. Alternatively, a branch-and-cut algorithm for finding the minimum distance of linear block codes has been proposed in [24]. In this paper a novel MC method to estimate the error correcting capability of the code/decoder is derived.

Summarizing, the main contributions are the following: (i) analytical derivation of the optimal importance sampling distribution among the family of Bernoulli distributions, with explicit calculation of the variance of the corresponding IS estimator and proof of convexity; (ii) derivation of two algorithms for fast-simulation, one to estimate numerically the optimal parameter of the importance sampling distribution and one that is invariant to SNR; (iii) derivation of one algorithm for efficiently estimate the number of correctable errors. Some illustrative numerical examples of application of the proposed algorithms, for BCH and LDPC codes, are also provided.

The proposed fast-simulation algorithms achieve large gains over standard MC simulation for a vast variety of communication systems where linear block codes are employed over binary symmetric channels (BSC). They are simple to im-



Figure 1. Illustrative block scheme of a communication system.

plement because they require only small modifications to the standard MC method, as the same random sample generator can be maintained and only the parameter of the Bernoulli generator is changed. Furthermore, in most practical situations the SNR-invariant version of the algorithm allows to efficiently obtain entire curves of performance, e.g. WERs corresponding to various SNRs, *by just running one IS simulation at one sufficiently high SNR*. In such a case the gain with respect to (w.r.t.) the standard MC simulation is even higher, as the number of simulation runs is dramatically reduced to one.

The outline of the paper is the following: in Sec. II the system model is introduced and some preliminaries on MC and IS method are given; the main results of the paper are presented in Sec. III; in Sec. IV fast-simulation algorithms are formulated and some examples are shown in Sec. V; finally, in Sec. VI some concluding remarks are given; proofs are confined to the appendices.

Notation - Lower-case bold letters denote vectors; the function $wt(\mathbf{z})$ returns the number of 1's in the binary vector \mathbf{z} ; $\mathbb{E}[\cdot]$ and var $[\cdot]$ denote expectation and variance operators, respectively; $\lceil \cdot \rceil$ denotes the ceiling operator; $P(\cdot)$ and $f(\cdot)$ are used to denote probabilities and probability mass function (pmf); $\mathcal{B}(i,p)$ denotes the pmf of the *n*-dimensional multivariate independent Bernoulli variable \mathbf{z} , with parameter p, i.e. $f(\mathbf{z};p) = p^i (1-p)^{n-i}$, where $i = wt(\mathbf{z})$; $\mathbb{E}_p[\cdot]$ and $\operatorname{var}_p[\cdot]$ denote expectation and variance operators with respect to the joint Bernoulli distribution of parameter p, respectively; finally, the symbols \sim and \oplus mean "distributed as" and "modulo-2 addition", respectively.

II. SYSTEM MODEL

A communication system where binary codewords are transmitted over a BSC with transition probability p is shown in Fig. 1. A codeword **c**, belonging to the block code $C \subset \mathcal{X}^n = \{0,1\}^n$ is obtained by encoding message word $\mathbf{m} \in \mathcal{X}^k$; at the output of the channel a word $\mathbf{z} \in \mathcal{X}^n$, corrupted by noise, is observed. The decoder's task is to possibly recover **m** given the observed **z**. The BSC may represent, for example, an additive white Gaussian noise (AWGN) channel with binary phase-shift keying (BPSK) modulation and hard-decision at the receiver, as shown in Fig. 1.

Performances of linear block codes over noisy channels are measured by the probability of decoding error, i.e. the probability that a decoded word is different from the transmitted message word, because the block code was not able to correct the errors due to the channel. This probability is also called probability of word error or WER. This event occurs when the error pattern is not a co-set leader (under the assumption that syndrome decoding is employed). Calculation of WER is often very complex and some upper bounds are available [4].

The WER, denoted as P(e) hereinafter, can be expressed in terms of an indicator function $I(\mathbf{z})$ that equals to 1 when the received word is erroneously decoded and 0 otherwise; its explicit form is given as

$$P(e) = \sum_{\mathbf{z} \in \mathcal{X}^n} I(\mathbf{z}) f(\mathbf{z}) = \mathbb{E}_p[I(\mathbf{z})].$$
(1)

Note that the indicator function hides the specific decoding algorithm employed. The effect of the BSC channel is to flip some bits, which can be mathematically expressed by $\mathbf{z} = \mathbf{c} \oplus \mathbf{e}$, with $\mathbf{e} \sim \mathcal{B}(wt(\mathbf{e}), p)$. Since the code is linear and the channel symmetric, without loss of generality (w.l.o.g.) the transmission of the codeword of all zeros is assumed, i.e. $\mathbf{c} = \mathbf{0}$, and hence the output of the channel $\mathbf{z} = \mathbf{e}$, i.e. equals the error pattern \mathbf{e} .

A. Monte-Carlo simulation

In the MC simulation method the WER is estimated as follows

$$\hat{P}_{MC}\left(e\right) = \frac{1}{N} \sum_{i=1}^{N} I\left(\mathbf{z}_{i}\right), \qquad (2)$$

where z_i are generated according the distribution of the random variable z. It is known that the MC estimator (2) is unbiased and its variance

$$\operatorname{var}\left[\hat{P}_{MC}\left(e\right)\right] = \frac{P\left(e\right)\left(1 - P\left(e\right)\right)}{N} \tag{3}$$

is inversely proportional to N (see, for example, [14]), then it can be made arbitrarily small as N grows, thus increasing the accuracy of the estimator. Rather than studying the variance it is often preferable to consider as accuracy of the estimator the relative error [14], defined as

$$\kappa \triangleq \frac{\sqrt{\operatorname{var}\left[\hat{P}_{MC}\left(e\right)\right]}}{P\left(e\right)}.$$
(4)

In standard MC simulation κ becomes

$$\kappa = \sqrt{\frac{1 - P(e)}{P(e)N}},\tag{5}$$

and, for small probabilities of error $(P(e) \ll 1)$, it is well approximated as

$$\kappa \simeq \frac{1}{\sqrt{P(e)N}}.$$
(6)

It follows that the number of generated samples needed to achieve a given κ is

$$N \simeq \frac{1}{\kappa^2 P\left(e\right)}.\tag{7}$$

Eq. (7) shows that the number of samples needed to obtain a given κ is inversely proportional to P(e) and becomes soon very high and often impractical as P(e) decreases. For example with a relative error of 10%, at least $N \simeq 10^2/P(e)$ samples are needed to obtain the desired accuracy.

Algorithm 1 Standard MC simulation algorithm

```
totWords = 0
WERre = 1
while
       (WERre > re) and (totWords <
maxNumWords) do
 z = rand(n, numWords) 
 output }
 \hat{\mathbf{m}} = \text{decode}(\mathbf{z}) \{ \text{decoder output} \}
  if
      wt(\hat{\mathbf{m}}) > 0 then
   totWErr = totWErr + 1
  end if
  totWords = totWords + numWords
  if totWords > minNumWords then
   update WERre { relative error }
  end if
end while
```

Algorithm 1 represents a generic implementation of MC simulation for estimation of P(e) in BSCs [1], [14]. The algorithm depends on three parameters: re, minNumWords, maxNumWords. The first parameter, re, is the relative error and it is computed according to (5) or its approximation (6), where P(e) is replaced by $P_{MC}(e)$, i.e. the current estimate. The second parameter, minNumWords, represents the minimum number of words needed to obtain a sufficiently accurate estimate of κ . Once a confident estimate of κ is obtained, a stop condition on the relative error can be employed. In practice in most cases a relative error of 10%, i.e. $\kappa = 0.1$, may suffice, as often only the order of magnitude of the estimate is of interest. Finally, maxNumWords represents the maximum number of generated words and it is used to implement a second stop condition that prevents the simulation to run too long.

Alternative stopping rules for MC simulations can also be considered. One common rule consists of fixing the number of generated word before running the simulation and the accuracy is estimated at the end of simulation [1]. Another rule, analyzed in [25], is based on the number of errors: when a given number has been reached, then the simulation stops. The advantage of this second rule is that it does not require to know the sample size and can achieve a given accuracy.

B. Importance sampling

In IS simulation the WER is expressed by the following equivalent of (1)

$$P(e) = \sum_{\mathbf{z} \in \mathcal{X}^{n}} I(\mathbf{z}) \frac{f(\mathbf{z})}{f^{*}(\mathbf{z})} f^{*}(\mathbf{z}), \qquad (8)$$

where $f^*(\mathbf{z})$ is a different pmf for which the sum in (8) exists. The corresponding estimator is

$$\hat{P}_{IS}(e) = \frac{1}{N} \sum_{i=1}^{N} I(\mathbf{z}_i) \frac{f(\mathbf{z}_i)}{f^*(\mathbf{z}_i)},$$
(9)

$$= \frac{1}{N} \sum_{i=1}^{N} I(\mathbf{z}_i) W(\mathbf{z}_i)$$
(10)

where $\mathbf{z}_{i} \sim f^{*}(\cdot)$ and the ratio

$$W(\mathbf{z}) \triangleq \frac{f(\mathbf{z})}{f^*(\mathbf{z})} \tag{11}$$

is referred to as the likelihood ratio or weighting function. The estimator in (10) is called the *IS estimator* and is a generalization of the simple MC estimator in (2), that can be obtained as special case (i.e. $f^*(\mathbf{z}) = f(\mathbf{z})$).

The distribution $f^*(\mathbf{z})$ is called the IS or biased distribution and as long as the random generation of samples is under our control, as in the case of MC simulation, it is possible to choose any distribution. However, it is crucial to choose the IS distribution such that the variance of the IS estimator is minimized. The optimal distribution is known from theory (see for example [13], [14]) and it is given by

$$f_{opt}^{*}\left(\mathbf{z}\right) = \frac{I\left(\mathbf{z}\right)f\left(\mathbf{z}\right)}{P\left(e\right)}.$$
(12)

This distribution leads to zero variance: this comes at no surprise since $f_{opt}^*(\mathbf{z})$ contains P(e) (which is the true value of the parameter being estimated). For this reason, the optimal solution cannot be used for MC simulation. Nonetheless, significant gains in simulation time can be achieved with sub-optimal biased distributions. Several methods to find sub-optimal biased distributions have been developed and the interested reader can refer to the comprehensive tutorial in [13]. One important goal in searching a sub-optimal IS distribution is to obtain a probability distribution from which samples can be easily generated and that, at the same time, provides a weighted estimator with as low variance as possible.

III. SUB-OPTIMAL IMPORTANCE SAMPLING

The main problem in the design of IS simulations is to find sub-optimal distributions that lead to low variance of the IS estimator. The problem can be simplified if the search is limited within a parametric family of distributions, since the problem can be recast into a standard optimization w.r.t. a finite number of parameters. Also, a proper choice of the parametric family can reduce the computational complexity due to the generation of random samples. In this paper the family of Bernoulli distributions with parameter q is considered, thus maintaining the simplicity of random generation of error patterns, since no change of the random generator is required. In practice the WER for a BSC with cross-over probability p is estimated by simulating the transmission over a different BSC with a different cross-over probability, denoted with q. Within this restriction the optimal q, denoted \hat{q} , is the cross-over probability that minimizes the IS estimator variance over all possible BSCs. Hereinafter, a general formula for \hat{q} is derived for any linear block code and for any decoding algorithm.

Consider the parametric family of joint Bernoulli distributions $\mathcal{B}(wt(\mathbf{z}), q)$ generated by varying q as IS distributions. The IS estimator for WER in (9) specializes to

$$\hat{P}_{IS}(e) = \frac{1}{N} \sum_{i=1}^{N} I(\mathbf{z}_i) \frac{f(\mathbf{z}_i; p)}{f(\mathbf{z}_i; q)}$$
(13)

$$= \frac{1}{N} \sum_{i=1}^{N} I(\mathbf{z}_{i}) \frac{p^{wt(\mathbf{z}_{i})} (1-p)^{n-wt(\mathbf{z}_{i})}}{q^{wt(\mathbf{z}_{i})} (1-q)^{n-wt(\mathbf{z}_{i})}}$$
(14)

$$= \frac{1}{N} \sum_{i=1}^{N} I(\mathbf{z}_i) W(wt(\mathbf{z}_i); p, q), \qquad (15)$$

where $\mathbf{z}_i \sim \mathcal{B}(wt(\mathbf{z}), q)$. Under the assumption $\mathbf{c} = \mathbf{0}$, the estimator can be equivalently expressed as

$$\hat{P}_{IS}(e) = \frac{1}{N} \sum_{i=1}^{N} I(\mathbf{e}_i) W(wt(\mathbf{e}_i); p, q), \qquad (16)$$

where $\mathbf{e}_{i} \sim \mathcal{B}(wt(\mathbf{e}), q)$. The general expression of the variance for the above estimator is

$$\operatorname{var}_{q}\left[\hat{P}_{IS}\left(e;q\right)\right] = \frac{\mathbb{E}_{q}\left[I\left(\mathbf{e}\right)W^{2}\left(wt\left(\mathbf{e}\right);p,q\right)\right] - P\left(e\right)^{2}}{N}$$
(17)

and clearly depends on q through the weighting function $W(\cdot)$ [13]. Therefore, the problem is to find the parameter q that minimizes (17), i.e.

$$\hat{q} = \arg\min_{q} \operatorname{var}_{q} \left[\hat{P}_{IS} \left(e; q \right) \right].$$
(18)

The expression of the IS estimator variance in the general case of linear block codes is given by the following lemma.

Lemma 1. The variance of $\hat{P}_{IS}(e;q)$ with importance sampling distribution in the parametric family $\mathcal{B}(i,q)$ is given by

$$\operatorname{var}_{q}\left[\hat{P}_{IS}\left(e;q\right)\right] = \frac{1}{N} \sum_{i=t+1}^{n} \left(W\left(i;p,q\right) P_{p}\left(e;i\right) - P_{p}\left(e;i\right)^{2}\right)$$
(19)

where W(i; p, q) is the weighting function of the IS estimator; $P_p(e; i)$ is the joint probability of decoding error with *i* errors over a BSC with cross-over probability p; *t* is the errorcorrecting capability of the decoder.

Proof: The proof is given in Appendix A. The above lemma provides a general expression of the variance of the IS estimator that depends on the specific de-

variance of the IS estimator that depends on the specific decoding algorithm employed *only through the error-correcting capability of the decoder t*. This parameter represents the maximum number of errors that the decoder is able to correct and depends on the structure of the linear block code and the decoding algorithm [4].

In order to solve the problem given by (18) we need to search for the equilibrium points of (19) w.r.t. q. The following lemma gives a closed-form expression of the variance derivative.

Lemma 2. The derivative of the variance of the IS estimator (16) is given by

$$\frac{\partial}{\partial q} \operatorname{var}_{q} \left[\hat{P}_{IS} \left(e \right) \right] = -\frac{1}{N} \sum_{i=t+1}^{n} \frac{i - nq}{q \left(1 - q \right)} W \left(i; p, q \right) P_{p} \left(e; i \right).$$
(20)

Proof: The proof is given in Appendix B. The solution of the minimization problem (18) can be obtained by equating to zero $\frac{\partial}{\partial q} \operatorname{var}_q \left[\hat{P}_{IS}(e) \right]$ if the IS variance is convex with respect to the variable q. The following lemma states that the second derivative of the IS estimator is always positive and then the variance of the IS estimator is convex.

Lemma 3. The IS estimator (16) is a convex function with respect to the variable q.

Proof: The proof is given in Appendix C. The following theorem gives the general expression for the value of q that minimizes the variance of the IS estimator and for which the estimation requires the minimum number of generated samples for a fixed relative error.

Theorem 4. The parameter q that minimizes the variance of the IS estimator given by (16) is

$$\hat{q} = \frac{1}{n} \frac{\sum_{i=t+1}^{n} iW(i; p, \hat{q}) P_p(e; i)}{\sum_{i=t+1}^{n} W(i; p, \hat{q}) P_p(e; i)}.$$
(21)

 $\begin{array}{cccc} \textit{Proof:} & \text{The proof is obtained by solving} \\ \frac{\partial}{\partial q} \mathrm{var}_q \left[\hat{P}_{IS} \left(e \right) \right] = 0 \text{ and exploiting Lemma 2.} \\ & \\ & \text{The result in (21) defines implicitly the optimal } q \text{ and} \end{array}$

The result in (21) defines implicitly the optimal q and therefore it is not possible to obtain a closed-form solution. In some cases, however, (21) assumes a simplified expression. When $np \ll 1$ the following approximation holds [4]

$$\operatorname{var}_{q}\left[\hat{P}_{IS}\left(e\right)\right] \simeq \frac{1}{N}W\left(t+1;p,q\right)P_{p}\left(e;t+1\right) - \frac{1}{N}P_{p}\left(e;t+1\right)^{2}$$
 (22)

and \hat{q} can be expressed explicitly, as stated by the following theorem.

Theorem 5. Under the approximation $np \ll 1$, the parameter *q* that minimizes the variance of the IS estimator (16)is

$$\hat{q} \simeq \frac{t+1}{n}.\tag{23}$$

Proof: The proof is given in Appendix D.

A notable consequence of Theorem 5 is the independence of \hat{q} from the cross-over probability p (which in turn depends on the SNR), therefore leading to an SNR-invariant IS-MC simulation. In this case estimation of WERs for a whole range of SNRs can be obtained by running one IS-MC simulation with a BSC with parameter \hat{q} given by (23), in the place of one simulation for each SNR. Thus the whole performance curve WER versus SNR can be obtained with a dramatic reduction of the number of samples to be generated. It is also interesting to note that for the Hamming code (7, 4) Eq. (23) gives $\hat{q} =$ 2/7 = 0.2857 which confirms the value of \hat{q} that Sadowsky found empirically in [17]. Furthermore, Sadowsky noted also the SNR invariance of \hat{q} with respect to p, without giving, unfortunately, any explanation.

Note also that for short codes the assumption $np \ll 1$ holds for a large range of SNRs and then (23) is valid for values of p of interest, while for long codes the same assumption holds only for high SNRs and (23) may not be useful in practice.

IV. Algorithms

The results presented in the previous section are exploited here to formulate two different IS-MC simulation algorithms to obtain performance curves in terms of WER vs SNR and an algorithm to estimate the error correcting capability of the decoder. The two fast-simulation algorithms compute the the WER estimate by means of the same IS estimator (16) and they differ only in the choice of the Bernoulli IS distribution parameter q. The first algorithm, called basic fastsimulation algorithm (IS-MC basic), estimates the optimal value \hat{q} and then proceeds with WER estimation. It is the most general algorithm since no specific assumption is required. The second algorithm assumes q = (t+1)/n, a choice based on Th. 5, and since q is independent on the current SNR, the algorithm is called SNR-invariant IS-MC algorithm. Under the assumption $np \ll 1$ the SNR-invariant IS-MC algorithm is computationally more efficient with respect to the IS-MC basic, as the same generated samples can be used to estimate WERs at different SNRs. The choice between the two algorithms depends on the code length n and cross-over probability p (or, equivalently, the range of SNRs of interest) and therefore on whether the assumption $np \ll 1$ holds or not.

Finally, the third algorithm is also based on the result of Th. 5 and does not estimate the WER, but rather the error correcting capability of the code.

A. Basic fast-simulation algorithm (IS-MC basic)

The basic version of the algorithm computes an estimate of the parameter \hat{q} iteratively, i.e. by updating q at iteration j from the q at iteration j - 1. In fact, from (21) the following update rule can be derived

$$\hat{q}_{j} = \frac{1}{n} \frac{\sum_{i=t+1}^{n} iW(i; p, \hat{q}_{j-1}) P_{p}(e; i)}{\sum_{i=t+1}^{n} W(i; p, \hat{q}_{j-1}) P_{p}(e; i)},$$
(24)

that can also be written as

$$\hat{q}_{j} = \frac{1}{n} \frac{\sum_{i=t+1}^{n} iW^{2}\left(i; p, \hat{q}_{j-1}\right) P_{q}\left(e; i\right)}{\sum_{i=t+1}^{n} W^{2}\left(i; p, \hat{q}_{j-1}\right) P_{q}\left(e; i\right)},$$
(25)

since $P_p(e;i) = W(i;p,q) P_q(e;i)$. Finally, the stochastic counterpart approximating (25) can be written in terms of the indicator function $I(\cdot)$

$$\hat{q}_{j} = \frac{1}{n} \frac{\sum_{i=1}^{N_{q}} I(\mathbf{z}_{i}) wt(\mathbf{z}_{i}) W^{2}(\mathbf{z}_{i}; p, \hat{q}_{j-1})}{\sum_{i=1}^{N_{q}} I(\mathbf{z}_{i}) W^{2}(\mathbf{z}_{i}; p, \hat{q}_{j-1})}, \quad (26)$$

where $\mathbf{z}_i \sim \mathcal{B}(wt(\mathbf{z}_i), \hat{q}_{j-1}).$

In practice the IS simulation consists of two major steps. During the first step an estimate of \hat{q} is derived through (26) with a fixed number of iterations and in the second step the WER estimation is performed by running the simulation with **Algorithm 2** IS simulation with embedded estimation of \hat{q}

```
totWords = 0
WERre = 1
\hat{q} = \hat{q}_0
while
         (WERre > re) and (totWords <
maxNumWords) do
 z = rand(n, numWords) < \hat{q}
                                  { BSC output
  }
  \hat{\mathbf{m}} = \text{decode}(\mathbf{z})
                     { decoder output }
  if
      \hat{q} has been estimated then
    compute running estimate of the WER
    according to (15)
    update totWords
    if
        totWords > minNumWords
                                      then
      update relative error WERre
                                         {
      relative error }
    end if
  end if
  if totWords < l*minNumWordsIS</pre>
                                         then
    update \hat{q} according (26)
  end if
end while
```

the IS Bernoulli distribution with parameter \hat{q} estimated in the first step. Even though an additional step is required to derive \hat{q} , it is expected that the total number of generated words will be reduced dramatically w.r.t. the standard MC simulation given κ .

Algorithm 2 implements the basic algorithm. The while loop implements the main part of the simulation that stops when either the relative error WERre is less than the given relative error re or the total number of generated words is greater than maxNumWords. First iterations of the algorithm compute the estimate of \hat{q} , with parameter l controlling the number of iterations required. The number of words N is represented by the variable minNumWordsIS. After \hat{q} has been estimated then the algorithm starts estimating the WER.

The search for \hat{q} depends on the starting probability \hat{q}_0 . A bad choice of \hat{q}_0 may slow down the rate of convergence of the estimation of \hat{q} and after 1 iterations \hat{q}_l might not be close to the optimal solution at all. It is important to choose \hat{q}_0 such a way that important events can be generated and a sufficiently number of errors are obtained to get an accurate estimate of \hat{q} . Obviously, if \hat{q}_0 is close to the optimal solution then a small number of iterations is required. If the number of correctable errors t is known then a possible choice could be the \hat{q} given by (23), even though for $np \ll 1$ a more computationally efficient simulation algorithm is possible, as it will be shown in the next section. An alternative choice can be made by observing that in a typical scenario the algorithm is run to draw a performance curve as function of the SNRs or the cross-over probability p. One can use the \hat{q} estimated with the simulation at the previous SNR as starting probability for the current SNR, i.e. \hat{q}_0 (SNR_i + Δ SNR) = \hat{q}_l (SNR_i), since for relatively small Δ SNR the new optimal \hat{q} is expected to be in the neighborhood of the previous \hat{q} . Furthermore, at low

SNRs the WER is usually high enough to require a limited and acceptable number of generated samples even with standard Monte-Carlo simulation and therefore at low SNRs the choice of \hat{q}_0 is less critical and can be chosen equal to the cross-over probability p.

The structure of the algorithm is very similar in its formulation to that presented in the context of cross-entropy method for simulation of rare events in [16]. An application of the cross-entropy method to the estimation of very low WERs of linear block codes has been proposed in [26]. The main difference with respect to the algorithm proposed in this paper is that the WER estimator in [26] has been proven to minimize the cross-entropy between the optimal IS solution and the parametric family of the joint Bernoulli distributions. Differently, the estimator proposed in this paper has minimum variance, thus leading to a different stochastic update rule for \hat{q} . The aforementioned update rule is proven to converge since the IS estimator variance (within the Bernoulli family) is convex and therefore one (global) minimum exists. Finally, it is worth noticing that the two approaches lead to the same SNR-invariant algorithm, as the result of Th. 4 holds in both cases.

B. SNR-invariant fast-simulation algorithm

The result of the Theorem 5 suggests a more computationally efficient IS-MC simulation algorithm that improves the basic algorithm derived in the previous sub-section. In fact, under the assumptions of the Theorem 5, the \hat{q} given by (23) does not depend on the current specific cross-over probability p of the channel being simulated. Then, the same set of generated samples with \hat{q} can be used to calculate the estimate of the WER at different SNRs. More specifically, in (15) the only term that depends on p is the weight function $W(wt(\mathbf{z}_i); p, q)$, which is a deterministic function. Therefore given one set of N realization of $\mathbf{z}_i \sim \mathcal{B}(wt(\mathbf{z}_i); q)$ it is possible to compute the estimated WER for any p for which the approximation $np \ll 1$ holds. In other words, with just one IS simulation WERs for any SNR in the range of application of Theorem 5 can be estimated.

On the other hand, the estimated κ that controls the number of words to be generated depends on the current SNR. A conservative rule for the choice of the relative error to be used in the stop condition is to select the relative error corresponding to the highest SNR in the given range, since, due to the monotonic decrease of the WER curve, this guarantees that all the other relative errors will be smaller.

C. Error-correcting capability estimation algorithm

The first step of the basic algorithm can be used to estimate the error correcting capability of the code and/or decoder, under the assumption of relatively high SNR, as stated by Theorem 4. In fact, Eq. (23) can be inverted to derive t from \hat{q} , that can be estimated. Note that, since the solution must be an integer, the estimate of \hat{q} may not need to have the same accuracy as that required for fast-simulation. Note also, that especially for long codes, the number of generated words to obtain t is far less than the number of codewords.



Figure 2. Estimated WER vs signal-to-noise ratio per uncoded bit (in dB) with IS-MC basic fast-simulation algorithm for a set of BCH codes with $R \simeq 0.9$.

V. EXAMPLES

In this section some examples of applications of the proposed fast-simulation algorithms are shown. The first example considers the application of the IS-MC basic, by simulating performances of a set of BCH codes [4] with code rate $R = k/n \simeq 0.9$, decoded with the Berlekamp-Massey algorithm [27], [28]. In Fig. 2 the WER vs signal-to-noise ratio per uncoded bit in dB, $(\mathcal{E}_b/N_0)_{dB}$, is reported, along with the parameters of the code that have been simulated. Each curve is obtained by running the basic algorithm at different SNRs with a stop condition on the relative error $\kappa = 0.1$. For reliable estimation of the parameter \hat{q} , the simulation of minNumWords= 10^2 has been assured and only one iteration has been performed, i.e. 1=1. The results of each simulation run are plotted with points on the interpolated curves, and correspond to the performances predicted by the theoretical upper bound for linear block codes [4]. On the same set of BCH codes the error correcting capability estimation algorithm has been applied with 100 generated words, and returns the correct number of correctable errors.

In Fig. 3 it is shown the number of generated words required by a standard MC simulation with $\kappa = 0.1$ for BCH code (2047, 1849). The number, that includes also the number of words required to estimate \hat{q} , increases with the SNR, but at some point, in the IS case (blue curve), it reaches a steady value. This corresponds to the region where the IS distribution does not depend on the cross-over probability of the channel (cf. Th. 5).

A second example is shown in Fig. 4 where the performances in term of WER vs SNR for the SNR-invariant IS-MC fast-simulation algorithm are plotted. In this case a different set of BCH codes is considered, with a code rate $R \simeq 0.5$. This



Figure 3. Number of generated words vs signal-to-noise ratio per uncoded bit (in dB) for IS-MC basic and MC (estimated with (7)), BCH code (2047, 1849).



Figure 4. IS-MC SNR-Invariant fast-simulation algorithm for BCH code with $R\simeq 0.5.$

set presents a greater number of correctable errors, and thus the decoding algorithm requires an increased computational complexity. The stop condition has been set on the relative error estimated at the highest SNR and only points with $\kappa < 0.1$ has been plotted. The performances in terms of WER confirm the theoretical results for BCH codes. More interestingly, it is important to note that each curve has been obtained with a single simulation run with a total number of generated words reported in Tab. I: with approximately 2×10^3 words it is possible to obtain the *entire curve* of performance.

The IS-MC method can be also employed to estimate the performances of LDPC codes. Fig. 5 shows the results of IS simulations of a set of LDPC codes taken from [29], [30], in terms of WER vs SNR per uncoded bit, for $\kappa = 0.1$. All codes

(n,k)	# of generated words		
(255, 231)	980		
(511, 259)	1150		
(1023, 513)	1560		
(2047, 1024)	2030		
(4095, 2057)	2640		
(8191, 7372)	1710		
(16383, 8200)	2410		
(32767, 29497)	2040		
(65535, 58991)	2100		

Table I Number of generated words with IS-MC SNR-invariant Algorithm. For each BCH codes the total number required to draw an entire performance curve is reported.



Figure 5. IS-MC basic fast-simulation algorithm for a set of LDPC codes. The code (273, 191) is taken from [3], the others from [29].

are decoded with the bit-flip iterative algorithm described in [19], with a number of iterations equal to 20. Estimation of \hat{q} has been performed with $N = 10^3$ generated words in one iteration. The same number is the minimum enforced to obtain a reliable estimate of the relative error.

The total number of generated words as function of the SNR is shown in Fig. 6. It is interesting to note that, as for BCH codes, at some point the number of generated words required to achieve the prescribed relative error (i.e. $\kappa = 0.1$) reaches a steady value. The flat region reflects the independence of the IS estimator variance on the SNR and identifies the SNR range over which the SNR-invariant algorithm can be effectively applied. However, the range of SNRs for which the curve is flat is different for each linear block code as it depends on the IS estimator variance which in turn depends on the structure of the code and the decoding algorithm. Numerical results show also that the assumption $np \ll 1$ is too strict, as it would have as consequence a flat region starting at higher SNR that those shown in Fig. 6. Furthermore, the number of generated words in the flat region varies with the codes. Results confirm the



Figure 6. Total number of generated words to obtain results in Fig. 5.

theoretical results obtained in Sec. III.

The error correcting estimation algorithm gives the number of correctable errors shown in Tab. II. Based on these estimates, the IS-MC SNR-invariant method is employed to draw the performance curves corresponding to the codes of Fig. 5. Results are reported in Fig. 7, that, as expected, shows the same performance results as shown in Fig. 5. The algorithm sets a stop condition on the relative error corresponding to the WER estimate at the higher SNR (in this case $\mathcal{E}_b/N_0 = 15 dB$) and only WER estimates with relative error less than the given $\kappa = 0.1$ are plotted. Results show also that the SNR-invariant algorithm correctly estimates WER for a large range of SNRs. On the other hand, at very low SNRs, the approximation (21) becomes sensibly different from the optimal solution. In Fig. 8, the relative error κ vs \mathcal{E}_b/N_0 is plotted, where becomes evident that (21) at low SNRs is not a good choice as the IS estimator variance increases up to a level that makes the computational complexity of IS simulation even higher that standard MC method or, equivalently, the relative error much higher than the one obtained with the same number of generated words with the standard MC method. Furthermore, it is interesting to note that the range of SNRs for which the relative error is below $\kappa = 0.1$ is larger than it was expected, suggesting that the assumption in Th. 4 is too strict.

VI. CONCLUSIONS

In this paper an IS estimator for fast-simulation of linear block codes with hard-decision decoding was presented. The estimator is optimal, i.e. it has minimum variance, within the restriction of the parametric family of IS distributions. It is possible to obtain huge gains w.r.t. the standard MC in terms of generated words. Although limited to the family of Bernoulli distributions, numerical examples have shown that in most practical cases the gains obtained are significant. However,

code	(96, 48)	(273, 191)	(495, 433)	(999, 888)	(1908, 1696)	(4376, 4094)
t	2	8	1	1	2	2

Table II

ESTIMATED NUMBER OF CORRECTABLE ERRORS FOR LDPC CODES OF FIG. (5) WITH BIT-FLIP DECODING [19].



Figure 7. IS-MC SNR-invariant fast-simulation algorithm for a set of LDPC codes with $\hat{q} = (t + 1) / n$ and t given by Table II. The code (273, 191) is taken from [3], the others from [29].



Figure 8. Relative error as function of $(\mathcal{E}_b/N_0)_{dB}$ corresponding to WER estimations obtained by application of the SNR-invariant algorithm and reported in Fig. 7.

the effective gain depends on the code and/or decoder performances in terms of WER. The advantage of the proposed methods is the low computational complexity and simplicity, since little modification w.r.t. the standard MC simulation is required. Finally, higher gains are achievable when the IS estimator does not depend on the cross-over probability of the channel being simulated, typically at high SNR.

VII. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Associate Editor and the anonymous reviewers for taking their time into reviewing this manuscript and providing comments that contributed to improve the quality and the readability of the manuscript.

APPENDIX A Proof of Lemma 1

The IS estimator can be rewritten as weighted sum indexed by the weights of the error patterns

$$\hat{P}_{IS}(e) = \frac{1}{N} \sum_{i=t+1}^{n} N_i W_i$$
(27)

where, for ease of notation, we denote W(i; p, q) as W_i ; N_i is the number of words with *i* errors; *t* is the maximum number of errors that the decoder can correct; *N* is the total number of generated samples. Therefore the variance can be written as

$$\operatorname{var}_{q}\left[\hat{P}_{IS}\left(e\right)\right] = \operatorname{var}_{q}\left[\frac{1}{N}\sum_{i=t+1}^{n}N_{i}W_{i}\right]$$
(28)

$$= \frac{1}{N^2} \sum_{i=t+1}^{n} \operatorname{var}_q \left[N_i W_i \right], \quad (29)$$

since generated samples constitute a realization of an i.i.d sequence of random variables. The variance under the summation can be also expressed as

$$\operatorname{var}_{q}[N_{i}W_{i}] = \operatorname{var}_{q}\left[\sum_{j=1}^{N} I_{i}\left(\mathbf{z}_{j}\right) W\left(wt\left(\mathbf{z}_{j}\right); p, q\right)\right] (30)$$
$$= \operatorname{var}_{q}\left[\sum_{i=1}^{N} I_{i}\left(\mathbf{z}_{j}\right) W_{i}\right] \qquad (31)$$

$$= W_i^2 \operatorname{var}_q \left[\sum_{j=1}^N I_i \left(\mathbf{z}_j \right) \right]$$
(32)

$$= W_i^2 \sum_{j=1}^N \operatorname{var}_q \left[I_i \left(\mathbf{z}_j \right) \right]$$
(33)

where $I_i(\cdot)$ is the indicator function that returns 1 when the event " \mathbf{z}_j contains *i* errors" occurs. Note that the term W_i is deterministic as it does not depend on the random variable \mathbf{z}_j , j = 1, ..., N. Now define

$$P_q(e;i) \triangleq \sum_{\mathbf{z}} I_i(\mathbf{z}) f(\mathbf{z};q)$$
(34)

as the joint probability that a decoding error occurs with an error pattern of weight i, when the IS distribution is a Bernoulli with parameter q. The variance of the estimator can be written as

$$\operatorname{var}_{q}[N_{i}W_{i}] = W_{i}^{2}\sum_{i=1}^{N}P_{q}(e;i)\left(1-P_{q}(e;i)\right)$$
 (35)

$$= NW_{i}^{2}P_{q}(e;i)(1-P_{q}(e;i))$$
(36)

The probability $P_q(e;i)$ can also be expressed in terms of $P_p(e;i)$. By definition

$$P_p(e;i) \triangleq \sum_{\mathbf{z}} I_i(\mathbf{z}) f(\mathbf{z};p)$$
 (37)

$$= \sum_{\mathbf{z}} I_i(\mathbf{z}) \frac{f(\mathbf{z};p)}{f(\mathbf{z};q)} f(\mathbf{z};q)$$
(38)

$$= W_i P_q(e;i) \tag{39}$$

Finally, the variance of the IS estimator is

$$\operatorname{var}\left[\hat{P}_{IS}\left(e\right)\right] = \frac{1}{N^{2}} \sum_{i=t+1}^{n} W_{i}^{2} N P_{q}\left(e;i\right) \left(1 - P_{q}\left(e;i\right)\right)$$
$$= \frac{1}{N} \sum_{i=t+1}^{n} W_{i}^{2} P_{q}\left(e;i\right) \left(1 - P_{q}\left(e;i\right)\right)$$
$$= \frac{1}{N} \sum_{i=t+1}^{n} W_{i}^{2} P_{q}\left(e;i\right) - \frac{1}{N} \sum_{i=t+1}^{n} W_{i}^{2} P_{q}\left(e;i\right)^{2}$$
$$= \frac{1}{N} \sum_{i=t+1}^{n} W_{i} P_{p}\left(e;i\right) - \frac{1}{N} \sum_{i=t+1}^{n} P_{p}\left(e;i\right)^{2}. \quad (40)$$

APPENDIX B PROOF OF LEMMA 2

The derivative of $\operatorname{var}_{q}\left[\hat{P}_{IS}\left(e\right)\right]$ can be written as

$$\frac{\partial}{\partial q} \operatorname{var}_{q} \left[\hat{P}_{IS}(e) \right] = \frac{\partial}{\partial q} \left(\frac{1}{N} \sum_{i=t+1}^{n} \left(W(i; p, q) P_{p}(e; i) - P_{p}(e; i)^{2} \right) \right) \\
= \frac{1}{N} \sum_{i=t+1}^{n} \frac{\partial W(i; p, q)}{\partial q} P_{p}(e; i), \quad (41)$$

where

$$P_{p}(e;i) \triangleq \sum_{\mathbf{z}} I_{i}(\mathbf{z}) f(\mathbf{z};p)$$
(42)

does not depend on q. After some manipulations the derivative of W(i; p, q) w.r.t. q can be written as

$$\frac{\partial W\left(i;p,q\right)}{\partial q} = \frac{\partial}{\partial q} \left(\frac{p^{i} \left(1-p\right)^{n-i}}{q^{i} \left(1-q\right)^{n-i}} \right)$$
$$= -W\left(i;p,q\right) \left(\frac{i}{q} - \frac{n-i}{1-q}\right). \quad (43)$$

By substituting (43) into (41) we obtain

$$\frac{\partial}{\partial q} \operatorname{var}_{q} \left[\hat{P}_{IS} \left(e \right) \right] = \\
= -\frac{1}{N} \sum_{i=t+1}^{n} W\left(i; p, q \right) \left(\frac{i}{q} - \frac{n-i}{1-q} \right) P_{p}\left(e; i \right) \\
= -\frac{1}{N} \sum_{k=t+1}^{n} \frac{i-nq}{q\left(1-q\right)} W\left(i; p, q \right) P_{p}\left(e; i \right). \quad (44)$$

APPENDIX C Proof of Lemma 3

The convexity is proven by showing that $\frac{\partial^2}{\partial q^2} \operatorname{var}_q \left[\hat{P}_{IS}(e) \right] > 0$. The second derivative of the variance is evaluated as follows (starting from Eq. (19)) :

$$\frac{\partial^2}{\partial q^2} \operatorname{var}_q \left[\hat{P}_{IS} \left(e \right) \right] =$$

$$= \frac{\partial}{\partial q} \left\{ -\frac{1}{N} \sum_{i=t+1}^n \frac{i - nq}{q \left(1 - q \right)} W \left(i; p, q \right) P_p \left(e; i \right) \right\}$$

$$= -\frac{1}{N} \sum_{i=t+1}^n P_p \left(e; i \right)$$

$$\left[\frac{\partial}{\partial q} \left(\frac{i - nq}{q \left(1 - q \right)} \right) \cdot W \left(i; p, q \right) + \frac{i - nq}{q \left(1 - q \right)} \cdot \frac{\partial W \left(i; p, q \right)}{\partial q} \right].$$
(45)

After some manipulations, derivatives in (45) can be written as

$$\frac{\partial}{\partial q} \left(\frac{i - nq}{q \left(1 - q \right)} \right) = \frac{i \cdot \left(2q - 1 \right) - nq^2}{\left[q \left(1 - q \right) \right]^2}$$
(46)

$$\frac{\partial W(i;p,q)}{\partial q} = -W(i;p,q)\left(\frac{i}{q} - \frac{n-i}{1-q}\right)$$
(47)

$$= -W(i; p, q) \left(\frac{i - nq}{q(1 - q)}\right) \quad (48)$$

After plugging (46) and (48) into (45) the following expression is obtained

$$\frac{\partial^{2}}{\partial q^{2}} \operatorname{var}_{q} \left[\hat{P}_{IS} \left(e \right) \right] = = -\frac{1}{N} \sum_{i=t+1}^{n} P_{p} \left(e; i \right) \\
\left[\frac{i \cdot (2q-1) - nq^{2}}{\left[q(1-q) \right]^{2}} \cdot W \left(i; p, q \right) - \left(\frac{i - nq}{q \left(1 - q \right)} \right)^{2} \cdot W \left(i; p, q \right) \right] \\
= \frac{1}{N} \sum_{i=t+1}^{n} P_{p} \left(e; i \right) W \left(i; p, q \right) \left[\frac{\xi(q, i)}{q^{2}(1-q)^{2}} \right] \quad (49)$$

where $\xi(q,i) \triangleq [(i-nq)^2 + nq^2 - i(2q-1)]$. The sign of the second derivative depends only on the term $\xi(q,i)$ that can be rewritten as

$$\xi(q,i) = i^2 - 2inq + n^2q^2 + nq^2 - 2iq + i$$
 (50)

$$= n (1+n) q^{2} - 2i (n+1) q + i (1+i)$$
(51)

The discriminant of the quadratic inequality $\xi(q, i) > 0$ is given by

$$\Delta = (-2i(n+1))^2 - 4n(1+n)i(1+i)$$
 (52)

$$= 4i^{2} (n+1)^{2} - 4ni (n+1) (i+1)$$
(53)

$$= 4i(n+1)[i(n+1) - n(i+1)]$$
(54)

$$= 4i(n+1)(ni+i-ni-n)$$
(55)

$$= 4i(n+1)(i-n).$$
(56)

For i < n we have $\Delta < 0$, therefore the corresponding terms in the sum that defines the second derivative are all positive. For i = n the term $\xi(q, n)$ is given by

$$(q,n) = (n-nq)^{2} + nq^{2} - n(2q-1)$$
(57)
(1)² + (²/₂ - 2) + 1) (57)

$$= n(1-q) + n(q^2 - 2q + 1)$$
 (58)

$$= (n+1)(1-q)^{2}$$
(59)

which implies $\xi(q, n) \ge 0$. The property $\xi(q, n) \ge 0$ readily implies convexity of $\operatorname{var}_q \left[\hat{P}_{IS}(e) \right]$.

Appendix D

PROOF OF THEOREM 5

The best IS distribution in the parametric family of Bernoulli distributions can be obtained by searching the parameter q that minimizes the variance of the IS estimator (16). From (22) we have that the only term that depends on q is W(t+1; p, q), denoted for convenience as W_{t+1} . In order to minimize the variance of the IS estimator the term W_{t+1} has to be to minimized, hence

$$\arg\min_{q} \operatorname{var}\left[\hat{P}_{IS}\left(e\right)\right] = \arg\min_{q} W_{t+1} \tag{60}$$

or equivalently

ξ

$$\arg\min_{q} \operatorname{var}\left[\hat{P}_{IS}\left(e\right)\right] = \arg\min_{q} \ln W_{t+1}$$
$$= \arg\max_{q} \ln \left[q^{t+1} \left(1-q\right)^{n-t-1}\right]. \quad (61)$$

The solution is obtained by equating the derivative of $\ln \left[q^{t+1} \left(1-q\right)^{n-t-1}\right]$ to zero and, after some manipulations, results to be

$$q = \frac{t+1}{n} \tag{62}$$

The choice of q according to the above equation minimizes the variance of the IS estimator. Note that q = 0 and q = 1 cannot be solutions of the minimization problem, as t is always non negative and upper bounded by $\lceil (d_{min} - 1)/2 \rceil$, where d_{min} is the minimum distance of the code that is always less than n. From (20) it is immediate to see that for q = 0 and q = 1 the variance of the IS estimator presents vertical asymptotes.

REFERENCES

- M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, Simulation of communication systems: modeling, methodology, and techniques. Kluwer Academic, 2nd ed., 2002.
- [2] W. Tranter, Principles of communication systems simulation with wireless applications. Prentice Hall, 2004.
- [3] R. Morelos-Zaragoza, *The art of error correcting coding*. John Wiley, 2006.
- [4] S. Benedetto and E. Biglieri, Principles of digital transmission: with wireless applications. Kluwer Academic, 1999.
- [5] J. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill International Edition, McGraw-Hill Higher Education, 2008.
- [6] R. Gallager, "Low-Density Parity-Check Codes," IRE Transactions on Information Theory, vol. 8, pp. 21–28, Jan. 1962.
- [7] D. J. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399–431, Mar. 1999.
- [8] S. Lin and D. Costello, *Error control coding: fundamentals and applications*. Pearson-Prentice Hall, 2004.
- [9] T. Richardson, "Error floors of LDPC codes," in *Proc. of the 41st Annual Allerton Conference on Communication, Control, and Computing*, vol. 41, pp. 1426–1435, 2003.
- [10] S. Chilappagari, S. Sankaranarayanan, and B. Vasic, "Error Floors of LDPC Codes on the Binary Symmetric Channel," in *Proc. of IEEE International Conference on Communications 2006 (ICC 2006)*, vol. 3, pp. 1089–1094, June 11–15, 2006.
- [11] B. P. Smith and F. R. Kschischang, "Future Prospects for FEC in Fiber-Optic Communications," *IEEE J. Sel. Topics Quantum Electron.*, vol. 16, pp. 1245–1257, Sept. 2010.
- [12] S. Ghosh and P. D. Lincoln, "Dynamic LDPC Codes for Nanoscale Memory with Varying Fault Arrival Rates," in Proc. of the 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), Apr. 2011.
- [13] P. Smith, M. Shafi, and H. Gao, "Quick Simulation: A Review of Importance Sampling Techniques in Communications Systems," *IEEE J. Sel. Areas Commun.*, vol. 15, pp. 597–613, May 1997.
- [14] R. Y. Rubinstein and D. P. Kroese, Simulation and the Monte Carlo Method. Wiley, 2nd ed., 2008.
- [15] R. Srinivasan, Importance Sampling: Applications in Communications and Detection. Springer-Verlag, 2002.
- [16] R. Y. Rubinstein and D. P. Kroese, The Cross-Entropy Method: A Unified Approach to Monte Carlo Simulation, Randomized Optimization and Machine Learning. Springer Verlag, 2004.
- [17] J. S. Sadowsky, "A New Method for Viterbi Decoder Simulation Using Importance Sampling," *IEEE Trans. Commun.*, vol. 38, no. 9, pp. 1341– 1351, 1990.
- [18] B. Xia and W. E. Ryan, "On Importance Sampling for Linear Block Codes," in *Proc. of IEEE International Conference on Communications* 2003 (ICC 2003), pp. 2904–2908, May 11–15, 2003.
- [19] A. Mahadevan and J. M. Morris, "SNR-Invariant Importance Sampling for Hard-Decision Decoding Performance of Linear Block Codes," *IEEE Trans. Commun.*, vol. 55, pp. 100–111, Jan. 2007.
- [20] C. Berrou, S. Vaton, M. Jezequel, and C. Douillard, "Computing the Minimum Distance of Linear Codes by the Error Impulse Method," in *Proc. of the Global Telecommunications Conference 2002 (GLOBECOM* 2002), vol. 2, pp. 1017–1020, Nov. 17–21, 2002.
- [21] X.-H. Hu, M. P. Fossorier, and E. Eleftheriou, "On the Computation of the Minimum Distance of Low-Density Parity-Check Codes," in *Proc.* of *IEEE International Conference on Communications 2004 (ICC 2004)*, vol. 2, pp. 767–771, June 20–24, 2004.
- [22] F. Daneshgaran, M. Laddomada, and M. Mondin, "An algorithm for the computation of the minimum distance of LDPC codes," *European Transactions on Telecommunications*, vol. 17, no. 1, pp. 57–62, 2006.
- [23] M. Punekar, F. Kienle, N. Wehn, A. Tanatmis, S. Ruzika, and H. W. Hamacher, "Calculating the minimum distance of linear block codes via Integer Programming," in *Proc. of the 6th International Symposium on Turbo Codes & Iterative Information Processing*, pp. 329–333, IEEE, Sept. 2010.
- [24] A. Keha and T. Duman, "Minimum distance computation of LDPC codes using a branch and cut algorithm," *IEEE Trans. Commun.*, vol. 58, pp. 1072–1079, Apr. 2010.
- [25] L. Mendo and J. M. Hernando, "A simple sequential stopping rule for Monte Carlo Simulation," *IEEE Trans. Commun.*, vol. 54, pp. 231–241, Feb. 2006.

- [26] G. Romano, A. Drago, and D. Ciuonzo, "Sub-optimal importance sampling for fast simulation of linear block codes over BSC channels," in *Proc. of the 8th International Symposium on Wireless Communication Systems (ISWCS 2011)*, pp. 141–145, Nov. 2011.
- [27] S. Wicker, *Error control systems for digital communication and storage*. Prentice Hall, 1995.
- [28] E. Berlekamp, Algebraic Coding Theory. No. M-6, Aegean Park Press, 1984.
- [29] D. J. MacKay, "Encyclopedia of sparse graph codes." http://www.inference.phy.cam.ac.uk/mackay/codes/data.html.
- [30] R. H. Morelos-Zaragoza, "The art of error correcting coding." http://theart-of-ecc.com.



Domenico Ciuonzo (S'11) was born in Aversa (CE), Italy, on June 29th, 1985. He received the B.Sc. (*summa cum laude*), the M.Sc. (*summa cum laude*) degrees in computer engineering and the Ph.D. in electronic engineering, respectively in 2007, 2009 and 2013, from the Second University of Naples, Aversa (CE), Italy. In 2011 he was involved in the Visiting Researcher Programme of the former NATO Underwater Research Center (now Centre for Maritime Research and Experimentation), La Spezia, Italy; he worked in the "Maritime Situation

Awareness" project. In 2012 he was a visiting scholar at the Electrical and Computer Engineering Department of University of Connecticut (UConn), Storrs, US. He is currently a postdoc researcher at Dept. of Industrial and Information Engineering of Second University of Naples. His research interests are mainly in the areas of Data and Decision Fusion, Statistical Signal Processing, Target Tracking and Probabilistic Graphical Models. Dr. Ciuonzo is a reviewer for several IEEE, Elsevier and Wiley journals in the areas of communications, defense and signal processing. He has also served as reviewer and TPC member for several IEEE conferences.



Gianmarco Romano (M'11) is currently Assistant Professor at the Department of Information Engineering, Second University of Naples, Aversa (CE), Italy. He received the "Laurea" degree in Electronic Engineering from the University of Naples "Federico II" and the Ph.D. degree from the Second University of Naples, in 2000 and 2004, respectively. From 2000 to 2002 he has been Researcher at the National Laboratory for Multimedia Communications (C.N.I.T.) in Naples, Italy. In 2003 he was Visiting Scholar at the Department of Electrical and Elec-

tronic Engineering, University of Conncticut, Storrs, USA. Since 2005 he has been with the Department of Information Engineering, Second University of Naples and in 2006 has been appointed Assistant Professor. His research interests fall within the areas of communications and signal processing.