ASYNCHRONOUS ERROR EVENT SAMPLING: AN ADAPTIVE IMPORTANCE SAMPLING TECHNIQUE FOR FAST MEASUREMENT OF LOW ERROR RATES IN DIGITAL COMMUNICATION SYSTEMS*

İrfan Acar
Steen A. Parl
Signatronics Technology Corporation
Concord, Massachusetts

ABSTRACT

We present an importance sampling technique for fast measurement of low error rates in digital communication systems via hardware simulation. This technique does not require detailed information about the system being tested and can be implemented in the absence of perfect synchronization of the noise samples with the transmitted symbols and the receiver decisions.

INTRODUCTION

It is well known that the low-error-rate performance of a digital communication system can be estimated much faster by employing the Importance Sampling (IS) simulation technique instead of the standard Monte Carlo (MC) simulation approach. The basic principle behind IS is that an unbiased estimate of the true error rate can be obtained by biasing the noise distribution to increase the frequency of error occurrences and properly weighting each error to compensate for this artificial increase [2].

A recent survey of various IS biasing schemes can be found in [5]. However, these IS schemes are designed primarily for software simulation and are not directly applicable to hardware testing of error-rate performances. In particular, they assume detailed knowledge of the system and/or perfect synchronization with the receiver, but these may not be available in a practical error-rate measurement in hardware.

The method we propose, originally developed in [4], is called Asynchronous Error Event Sampling (AEES), and employs an excision biasing scheme [1], [3]-[7] which tries to constrain the support of the noise magnitude distribution to cover the error region as tightly as possible while leaving the noise phase distribution intact. The actual amount of excision does not depend on the transmitted symbol sequence and is determined adaptively in a side experiment conducted prior to the actual measurement. Since the biased distribution is a scaled version of the original distribution outside the excised region, we only need to scale the total error count by the volume of the support of the biased distribution instead of weighting each error occurrence by a factor which depends on the realization of the noise, as is the case in other biasing schemes. Because of this property and a cyclic symmetry constraint imposed on the excised region, the AEES technique can be implemented even when perfect synchronization with the receiver is not available.

The reason for biasing only the noise magnitude distribution is that this allows us to perform the excision in a very short time relative to the consequent reduction in the total measurement time by utilizing the assumption that the effect of the noise magnitudes is monotonic. Under this assumption, we can identify a hyperrectangle in the noise magnitude space as either inside or outside the error region by empirically identifying only one of its corner points. We present an efficient adaptive algorithm for successive selection of the corner points to be identified, so that a large part of the error-free region can be excised quickly by iteratively removing error-free hyperrectangles from the support of the noise magnitude distribution. The overall result is a much faster error rate measurement, because the reduction in the number of trials will greatly outweigh the additional time required for the side experiment when the error rate is low.

To implement the AEES technique sequentially, the excision algorithm is supplemented by a new method for the generation of random noise magnitude samples from the biased distribution. The sequential simulation unfolds in time, emulating the actual system, except that a window of length $M$, where $M$ is the system memory length, slide along the noise magnitude sequence, is constrained to always lie outside the excised region. The generation of a new sample is accomplished at a rate which is independent of $M$, thus ensuring that the highest sampling rate that can be supported by AEES is not limited by $M$.

PROBLEM STATEMENT

Consider the problem of measuring the error rate of the baseband communication system depicted in Figure 1, where the transmitted signal is sampled and noise is added digitally. Here, $X_i$ denote the complex signal samples and $N_i$ denote independent and identically distributed complex Gaussian random noise samples. The transmitted bits and the corresponding binary decisions are denoted by $b_i$ by $b_{in}$, respectively. Note that the dashed box can be viewed as a generic test set for verifying the error-rate performance of a transmitter-receiver pair. We assume that there are $L$ input

---

*This material is based upon work supported by the Naval Surface Warfare Center, Dahlgren Division under Contract No. N00178-96-C-3061
samples per symbol and that each symbol decision depends on \( M = Lm \) input samples, where \( m \) is the receiver memory in number of symbols. We also assume that the receiver does not make any erroneous decisions when \( N_i \equiv 0 \).

Our aim is to accelerate the measurement of the low error rates of this system using a strategy as simple and as general as the MC simulation. In particular, we would like to be able to implement this strategy even when perfect synchronization with the receiver is not possible. This means that our implementation should not rely on the knowledge of the symbol arrival times or the receiver decision times, as we might not be able to associate an erroneous decision (indicated by the AEES processor that compares the transmitted bits to the bit decisions) with the responsible noise vector.

Since each symbol decision depends on \( M \) input noise samples, identifying error-free subregions in a noise magnitude space of the same dimension is sufficient to describe the realizations of the noise process which contains the noise magnitude sequences that cause errors.

Let us denote the vectors obtained by sliding a window of width \( M \) along the noise sequence by \( N_i \equiv [N_{i-M+1}, \ldots, N_{i-1}, N_i] \). The corresponding noise magnitude vector is then defined as \( R_i \equiv [R_{i-M+1}, \ldots, R_{i-1}, R_i] \), where \( R_j = |N_j| \).

In a simulation, the random noise magnitudes are usually generated as nonlinear transformations of uniform random variables in the interval \( U = [0, 1) \). For example, assuming that the noise process is Gaussian with real and imaginary components having variance \( \sigma^2 \), the noise magnitude vector \( R \) corresponds to \( U \in U^M \), with the components given by \( U_j = 1 - e^{-R_j^2/2\sigma^2} \), for \( j = 0, 1, \ldots, M - 1 \). Since it is more convenient to work in \( U^M \), we will do so in the sequel.

**ASYNCHRONOUS ERROR EVENT SAMPLING**

We define the region \( \mathcal{G} \) in the noise magnitude space \( U^M \) as the collection of all noise magnitude vectors \( p \in U^M \) with the property that some cyclic shift of \( p \) can cause a decision error for some signal vector and some noise phase vector.

In AEES, we would like to generate the noise magnitude samples with a biased density \( f_{U^*}(u) \) the support of which is constrained to \( \mathcal{G} \). Namely,

\[
f_{U^*}(u) = \begin{cases} \frac{1}{P_e} f_U(u), & u \in \mathcal{G}, \\ 0, & \text{otherwise}, \end{cases}
\]

where \( f_U(u) \) is the original density of the noise magnitude samples, and \( P_e = \int_{\mathcal{G}} f_U(u) \, du \). The AEES estimation is performed by generating noise samples with a distribution proportional to the original; thus, no weighting of the errors is involved during the simulation. Namely, the AEES estimate of \( P_e \) is given by

\[
P_e^* = \frac{P_e}{J^*} \sum_{n=1}^{J^*} 1(\hat{b}_n \neq b_n),
\]

where the error indicator function \( 1(\cdot) \) is equal to 1 when its argument is true and 0 otherwise. This estimator is unbiased and its variance is equal to \( \left( P_e P_e - P_e^2 \right) / J^* \), where \( J^* \) is the number of simulation trials and \( P_e \) is the probability of error. When we compare the time-reliability product \( [2] \) of this estimator to the MC estimator, we note that the same precision in the error-rate estimate can be obtained with \( J^* \approx P_e J \), the approximation being valid for very small error probabilities.

However, the region \( \mathcal{G} \) is not available to us and must be determined by conducting a side experiment to obtain an estimate of it. The most important concern here is to excise as large a part of \( \mathcal{G} \) with as little effort as possible to reduce the measurement time, while at the same time being able to support a high sampling rate. These requirements, combined with the monotonicity assumptions,\(^1\) lead us to

\begin{itemize}
  \item The monotonicity assumptions are: (i) if a certain noise magnitude vector causes errors for some signal vector and some noise phase vector, then any amount of increase in one or more of its components will give us another noise magnitude vector that can also cause errors for some
\end{itemize}
employ the following approach. First, we determine whether or not a certain noise magnitude vector belongs to $G$. This is accomplished by combining a long noise magnitude sequence obtained by repeating the noise magnitude vector in question with a random noise phase sequence, and feeding it to the system. The occurrence of an error during this test implies that the noise magnitude vector and its cyclic shifts are in $G$. If an error is not encountered, we conclude that this noise magnitude vector and its cyclic shifts are not in $G$. However, since the reliability of this conclusion depends on the length of the test sequence, it must be chosen carefully. For, if it is too long, the excision time will increase and, as a result, the measurement time will suffer. On the other hand, if it is too short, we will increase the probability that some part of $G$ will be excised by mistake and result in a biased estimate.

Once we determine a single vector $p$ in $G$, the use of the monotonicity assumptions allows us to conclude that the hyperrectangle with its lower corner given by $p$ and its cyclic shifts are in $G$. We will denote an arbitrary hyperrectangle extending from the tip of the vector $a$ to the tip of the vector $b$ by $H(a,b)$. Formally,

$$H(a,b) = \{ u : a_i \leq u_i \leq b_i, i = 0,1,\ldots,M-1 \}.$$ 

The $j$-th cyclic shift of $H(a,b)$ is given by $H_j(a,b) = \{ u : a_i \leq u_{(i+j)\mod M} \leq b_i, i = 0,1,\ldots,M-1 \}$, where $j = 0,1,\ldots,M-1$. We adopt the convention that $H_0(a,b) = H(a,b)$.

Similarly, determining a vector $p$ outside $G$ tells us that the hyperrectangle with its upper corner equal to $p$ and its cyclic shifts are outside $G$. Then, by selecting other suitable points and determining whether they belong to $G$ or not, it is possible to approximate these regions to any desired accuracy as unions of hyperrectangles. Although it is desirable to excise as much as possible from the noise magnitude space, the reduction in the simulation runtime is inversely proportional to the volume outside the excised region, and after a certain point this reduction may not justify the extra time required to perform these excisions.

**CYCLOSYMMETRIC HYPERRECTANGLE EXCISION**

It can be shown that, in the absence of any information about the the error region boundary, the minimax optimum strategy for excising an error-free hyperrectangle from a hyperrectangular subregion in the noise magnitude space is to determine the intersection of the main diagonal of this subregion with the boundary of the error region. Before we start testing, the noise magnitude space is an indeterminate hyperrectangle, except for its corners. Therefore, our first step is to search along the main diagonal of the noise magnitude space for its intersection with the error region boundary, denoted by $p^* = [p^*,p^*,\ldots,p^*]$. We start by generating a long noise sequence so that every sample has a random phase but the magnitude is kept constant at a large value $p_0$. The reason for this is that an error would be encountered much earlier on the average, thereby enabling us to complete the side experiment faster. Note that, compared to error-free region points, determining an error region point takes less time on the average. While this test sequence is being fed to the system, we observe whether an error occurs. Normally, we should encounter an error with such a large noise magnitude. We repeat this test by generating the next sequence with the constant magnitude given by $p_k = 1 - \sqrt{1 - p_{k-1}}$, until an error is not encountered for a sufficiently long sequence and we conclude that the corresponding magnitude belongs to the error-free region. Let us assume that this occurs for the $K$-th magnitude $p_K$. Then, we have that $p_K < p^* < p_{K-1}$. At this point, we generate the next magnitude to be tested as $p_{K+1} = 1 - \sqrt{1 - p_K(1 - p_{K-1})}$. Note that this corresponds to a geometric mean search. Depending on the outcome of this test, one of the bounds on $p^*$ is replaced. We then repeat the same procedure until the lower and upper bounds are sufficiently close to one another. The decision to stop is made by comparing the gain to be obtained by bringing the bounds closer to the time it will take to do so. Then, the point corresponding to the lower bound at the stopping time, $p^* = [p^*,p^*,\ldots,p^*]$, is used to excise the error-free hyperrectangle extending from the origin, 0, to $p^*$. The lower bound, rather than the upper bound, is chosen to avoid the excision of part of the error region as this would lead to a biased estimate.

The point corresponding to the upper bound at the stopping time, $p^* = [p^*,p^*,\ldots,p^*]$, defines the error region hyperrectangle extending from $p^*$ to 1, the upper corner of the noise magnitude space with all coordinates equal to 1. We call the part of the noise space $U^M$ that lies outside of both the error region hyperrectangle and the error-free hyperrectangle the indeterminate region. We could partition this indeterminate region into disjoint hyperrectangles and search for the intersection of the diagonal of each one with the boundary of the error region. The intersection points would then give us a new error-free hyperrectangle, a new error region hyperrectangle, and a new indeterminate subregion in each one of the disjoint hyperrectangles. This procedure could then be repeated to further excise more error-free hyperrectangles from the noise space. However, this approach is not practical as the complexity of partitioning the initial indeterminate region grows exponentially with $M$. To simplify the excision process, we use the fact that $p^*$ will be very close to 1 for low error probabilities to limit further excursion to some subset of the indeterminate region which can be partitioned into $M$ disjoint hyperrectangles that are cyclic shifts of one another and whose volume is very close to that of the whole indeterminate region. The rest of the indeterminate region, whose volume is negligible compared to the subset considered for further excision, is assigned directly to
the final simulation region without affecting the performance significantly. After the excision of error-free hyperrectangles, the simulation region consists of disjoint cyclosymmetric sub-regions, which will be defined shortly.

For some \( c \) on the diagonal of \( \mathcal{H}(a, b) \), i.e. \( c = (1 - \alpha) a + \alpha b \), \( 0 < \alpha < 1 \), we define the following subregions of \( \mathcal{H}(a, b) \):

\[
\mathcal{F}_c(\mathcal{H}(a, b)) = \{ u : a_i \leq u_i \leq c_i, i = 0, 1, \ldots, M - 1 \},
\]

\[
\mathcal{E}_c(\mathcal{H}(a, b)) = \{ u : c_i < u_i \leq b_i, i = 0, 1, \ldots, M - 1 \},
\]

\[
\mathcal{I}_c^{(j)}(\mathcal{H}(a, b)) = \{ u : c_j < u_j \leq b_j, a_i \leq u_i \leq c_i, i \neq j \},
\]

\[
\mathcal{I}_c(\mathcal{H}(a, b)) = \bigcup_{j=0}^{M-1} \mathcal{I}_c^{(j)}(\mathcal{H}(a, b)),
\]

\[
\mathcal{S}_c(\mathcal{H}(a, b)) = \mathcal{H}(a, b) - \mathcal{F}_c(\mathcal{H}(a, b)) - \mathcal{E}_c(\mathcal{H}(a, b)) - \mathcal{I}_c(\mathcal{H}(a, b)),
\]

and

\[
\hat{\mathcal{S}}_c(\mathcal{H}(a, b)) = \mathcal{H}(a, b) - \mathcal{F}_c(\mathcal{H}(a, b)).
\]

If \( c \) is the intersection of the error region boundary with the diagonal of \( \mathcal{H}(a, b) \), then \( \mathcal{F}_c \) is the error-free hyperrectangle that will be excised, \( \mathcal{E}_c \) is the hyperrectangle that is known to be in the error region. If the excision is continued, \( \mathcal{I}_c \) is the part that will be considered for further excision in the next step, and \( \mathcal{S}_c \) is what we call the carving of type II that will be added to the simulation region. If the excision is stopped, \( \hat{\mathcal{S}}_c \) is what we call the carving of type I that will be added to the simulation region.\(^3\)

In order to help in visualizing these regions, we provide a schematic in Figure 2 for the three-dimensional case.

Returning back to our algorithm, we note that the noise magnitude space is itself a hyperrectangle described by \( \mathcal{H}(0, 1) \). Therefore, after the determination of \( p^* \), we excise \( \mathcal{F}_{p^*}(\mathcal{H}(0, 1)) \) as the error-free hyperrectangle, and add the excised hyperrectangle \( \hat{\mathcal{S}}_{p^*}(\mathcal{H}(0, 1)) \) to the simulation region. Note that the indeterminate region \( \mathcal{I}_{p^*}(\mathcal{H}(0, 1)) \)

\(^3\)The terms carving of type I and carving of type II will be used as generic names for subregions having the same form as \( \hat{\mathcal{S}}_c \) and \( \mathcal{S}_c \).
tangles are considered for excision. In practice, however, we have to select the hyperrectangles that will be considered for further excision according to their volumes, as the contribution to the gain in simulation runtime is marginal for an excision from a very small hyperrectangle. Furthermore, preliminary computer simulations performed for linear systems in Gaussian noise suggest that the improvement in the gain will be insignificant after the second or third stage.

Once we decide that any further excision will not bring substantial improvements in the simulation runtime, the excision algorithm stops. At this point, the final indeterminate hyperrectangles are combined with the corresponding subregion $S$ (carving of type II) to form a single subregion $\hat{S}$ (carving of type I) to be added to the simulation region. Thus, the final simulation region consists of only excised hyperrectangles of type I and II. In the next section, we introduce a new method for generating a realization of a noise process with uniform distribution inside a carving of type I or II.

**SEQUENTIAL SIMULATION**

In the following, we will present alternative representations for the regions which will be used for random sequence generation. First, we define the following counting function for the vectors $u \in \mathcal{H}(a, b)$:

$$C_{a, b, c}(u) = \sum_{i=0}^{M-1} C_{a, b, c}^{(i)}(u_i),$$

where

$$C_{a, b, c}^{(i)}(x) = \begin{cases} 1, & c_i < x \leq b_i, \\ 0, & a_i \leq x \leq c_i. \end{cases}$$

Note that, $C_{a, b, c}(u)$ counts the number of coordinates of $u$ that are larger than the corresponding coordinates of $c$.

Then,

$$F_C(\mathcal{H}(a, b)) = \{u : C_{a, b, c}(u) = 0\},$$

$$E_C(\mathcal{H}(a, b)) = \{u : C_{a, b, c}(u) = M\},$$

$$I_C(\mathcal{H}(a, b)) = \{u : C_{a, b, c}(u) = 1\},$$

$$S_C(\mathcal{H}(a, b)) = \{u : C_{a, b, c}(u) \geq 2\},$$

$$\hat{S}_C(\mathcal{H}(a, b)) = \{u : C_{a, b, c}(u) \geq 1\}.$$

The important point here is that the final simulation region consists of subregions of the form $S_C$ and $\hat{S}_C$, and that the membership to these subregions is determined only by the count $C_{a, b, c}(u)$ irrespective of $M$.

Now, consider the generation of a random sequence so that a sliding window of width $M$ will be always in $S_C(\mathcal{H}(a, b))$ or in one of its cyclic shifts, namely, in the region described by

$$\bigcup_{j=0}^{M-1} S_{c_j}(\mathcal{H}_j(a, b)).$$

where the vector $u_i$ is assumed to be in $S_{C_i}(\mathcal{H}_j(a, b))$. Then, $U_{i+1}$ has to be generated with a uniform distribution in an interval that will ensure that $u_{i+1}$ is in $S_{C_{i+1}}(\mathcal{H}_{j+1}(a, b))$. We note that the counting function for $u_{i+1}$ is related to the counting function for $u_i$ by

$$C_{a_{i+1}, b_{i+1}, c_{i+1}}(u_{i+1}) = C_{a_i, b_i, c_i}(u_i) - C_{a_i, b_i, c_i}^{(0)}(u_{i-M+1}) + C_{a_i, b_i, c_i}^{(0)}(u_{i+1}).$$

Thus, as we slide the window by one sample, there are only two samples, the one that falls out of the window and the one that we are about to generate, which can effect the count that will determine whether $u_{i+1}$ is in $S_{C_{i+1}}(\mathcal{H}_{j+1}(a, b))$ or not. Since this count uniquely determines whether the new window is in the desired subregion, we only have to check the value of the sample falling out of the window in order to determine the interval in which the new sample is to be generated. If the sample left out decreases the count below the lower bound count of the constraint (2 for a carving of type II and 1 for a carving of type I) then the new sample has to be generated with a uniform distribution in the interval $[c_{j+1}, b_{j+1}]$ to increase the count back up. Otherwise, the interval is not constrained, i.e. $[a_{j+1}, b_{j+1}]$. With this method, only the support of the original joint distribution is modified. This can be shown using the fact that each noise sample has a uniform conditional distribution.

Note that the computational load of generating the sample $U_{i+1}$ is independent of $M$. The implication is of course that the highest sampling rate that can be supported by the AEES algorithm in real-time will not be constrained by the system memory.

Having developed a method for generating a random sequence inside the excised hyperrectangles, we can now describe the overall simulation part of the AEES algorithm. After the error-free hyperrectangles are excised in the side region trials for each one of the simulation subregions of type I ($\hat{S}$) or II ($S$) in proportion to their respective volumes. Then, for each of the one subregions, which together constitute the simulation region, we generate a separate random noise sequence using the method described above, and count the number of error occurrences. Afterwards, the total error count is simply scaled by the total volume of the simulation region to obtain an estimate of the error probability:

$$\hat{P}_e = \sum_{k \neq 0} \frac{P_{x_k}}{J_x^*} \sum_{i=1}^{J_x^*} I(\hat{b}_n \neq b_n),$$

where the index $k$ runs through the (disjoint) carvings that constitute the final simulation region $\mathcal{G}$. This estimator essentially consists of subestimators for the conditional error probability given that the noise magnitude sample is in the carving $\mathcal{G}_k$. If the number of simulation trials for the subestimator for $\mathcal{G}_k$ is chosen such that $J_x^* = \lceil P_{x_k} J^*/(1 - P_{y_k}) \rceil$, then it can be shown that the estimator above achieves a smaller variance than a direct approach without partitioning with $J^*$ simulation trials.
CONCLUSION

In this paper, we have presented an importance sampling technique that can be used for hardware measurement of error rates. This new technique employs an adaptive biasing scheme and excises cyclosymmetric error-free subregions from the noise magnitude space. The remainder is partitioned into cyclosymmetric carvings allowing us to implement a sequential simulation algorithm. As a result, the data rates that can be supported are not limited by the system memory.

REFERENCES


