IPN Progress Report 42-201 • May 15, 2015

# Confidence Intervals for Error Rates Observed in Coded Communications Systems

Jon Hamkins*

ABSTRACT. — We present methods to compute confidence intervals for the codeword error rate (CWER) and bit error rate (BER) of a coded communications link. We review several methods to compute exact and approximate confidence intervals for the CWER, and specifically consider the situation in which the true CWER is so low that only a handful, if any, codeword errors are able to be simulated. In doing so, we answer the question of how long an error-free simulation must be run in order to certify that a given CWER requirement is met with a given level of confidence, and discuss the bias introduced by aborting a simulation after observing the first codeword error.

Next, we turn to the lesser studied problem of determining confidence intervals for the BER of coded systems. Since bit errors in systems that use coding or higher-order modulation do not occur independently, blind application of a method that assumes independence leads to inappropriately narrow confidence intervals. We present a new method to compute the confidence interval properly, using the first and second sample moments of the number of bit errors per codeword. This is the first method we know of to compute a confidence interval for the BER of a coded or higher-order modulation system.

## I. Introduction

The simulated performance of a communications system should always come with the caveat that error bars are associated with the numerical results. Longtime JPLer Dick Mathison was notorious for asking why error bars were not being shown on plots presented at reviews. Care should be taken to verify that the simulations are long enough so that the error bars are

*Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. 

acceptably small, even if they are not shown.

However, at very low error rates it may be impractical to simulate many error events. What can we conclude about performance when only a handful, or even zero, events are simulated? A related question that a flight project might ask is, how long must a laboratory test be run error-free before concluding that a given error-rate requirement is met? How do these answers change when the error events are not independent? This article presents methods to compute the statistical confidence intervals that help answer these questions.

First, we address the computation of confidence intervals for the codeword error rate (CWER), or frame error rate, observed in a simulation or test of transmitted codewords. The approach is quite general and applies to any independently occuring events including, for example, the BER of *uncoded* binary channels, or the modulation-symbol error rate on an uncoded memoryless channel, but for the ease of presentation, we henceforth discuss only codeword errors. We develop guidelines that can aid the practicing engineer in certifying that their simulation or laboratory test of codeword errors is sufficient to prove that an error rate requirement has been met. Even though these guidelines are mathematically rigorous, we still refer to them as:

*Rules of Thumb*

1. An error-free simulation or test of $w$ codewords certifies that the true CWER is less than $4/w$, with 95% confidence.
2. At least 4 codeword errors must be observed in order to determine the CWER to within an order of magnitude, with 95% confidence.
3. A simulation will be accurate to within plus or minus $100\gamma\%$ of its observed CWER, with 95% confidence, if $4/\gamma^2$ codeword errors are observed. This means 16 codeword errors are sufficient for an error margin of plus or minus 50%, 400 are enough for 10%, and 40000 are enough for 1%, with 95% confidence.
4. A simulation or test designed to immediately abort after observing the first codeword error produces a biased estimate of the CWER, and this bias can be more than an order of magnitude when CWER $< 10^{-6}$.

A second, and as far as we know novel, contribution of this article is the development of confidence intervals for the bit error rate (BER) of coded systems. Unlike codeword errors, bit errors in coded systems do not occur independently. We show that blindly computing a confidence interval for the BER under the false assumption that bit errors are independent leads to an inappropriately narrow interval. We present a method to compute the confidence interval properly, using the first and second sample moments of the number of bit errors per codeword. This method connects the rules of thumb above for CWER to corresponding rules of thumb for BER of coded systems.

Section II discusses the history of the confidence interval problem in communications, Section III presents confidence intervals for the CWER, and Section IV presents confidence intervals for the BER. Numerical examples are given in Section V.

## II. History of the Confidence Interval Problem in Communications

In 1976, Massey had a contract with NASA [1] to help find a channel code appropriate for the International Ultraviolet Explorer (IUE), a joint space mission between NASA, the European Space Agency, and the UK Space Research Council. The mission did in fact use a convolutional code and went on to great success, returning over 100,000 images that formed a heavily used astronomy database and spawned nearly 4,000 peer-reviewed astronomy papers.

IUE decided it wanted a constraint-length 24, rate 1/2 convolutional code with sequential decoding. See, e.g., [2] for a discussion of sequential decoding of convolutional codes. This mission was just prior to the era in which much shorter constraint-length codes decoded with the Viterbi algorithm began to become the norm. Indeed, it stands out that at the time Massey referred to constraint-length 24 as "rather short" [1]!

Massey proceeded to analyze and simulate virtually every binary (24,1/2) convolutional code that had been proposed up to that time. He considered ten codes in all, including ones designed by himself, Costello, Johannesson, Bahl, Jelinek, Bussgang, Lin, and Lyne — see the references of [1] for full details on the codes. With the large constraint length of 24 and short codeword length of 256 bits, terminating the trellis reduced the code rate noticeably — unfortunately, this was before the invention of the tail-biting method, which would have neatly avoided the problem (see, e.g., [3]) — and so Massey also considered various partial termination schemes.

Given the computers of the day, Massey's simulations were limited to decoding, for each code, 10,000 codewords at one signal-to-noise ratio (SNR). Four top candidate codes were each simulated for an additional 40,000 codewords. His simulations produced 0 to 5 codeword errors for each code. With this hard-fought-for but meager amount of data, Massey used confidence intervals to make conclusions about the relative merits of the various codes.

## III. Confidence Interval for the CWER

Suppose that $X = x$ codeword errors are observed in a simulation or test of $w$ transmitted codewords. We assume that the channel is memoryless, so that codeword errors are statistically independent. The observed CWER is $\hat{p} \triangleq x/w$. Let $p$ denote the true CWER. Then $X$ is a binomially distributed random variable with success probability $p$ and number of trials $w$, which we denote by $X \sim \mathrm{Bin}(w; p)$. Let $a(X, w)$ and $b(X, w)$ be two deterministic functions of the observed data such that, with probability $\beta$, the interval $(a(X, w), b(X, w))$ contains the true value $p$:

$$P[a(X,w) \leq p \leq b(X,w)] = \beta. \tag{1}$$

Here, $X$ is the only random parameter ($w$ is a known constant, $p$ is an unknown non-random parameter, and the functions $a()$ and $b()$ are known). Any interval defined by a function pair that satisfies (1) is said to be a $\beta$-confidence interval.

Care should be given in speaking correctly of confidence intervals. When one runs a simulation and the random variable $X$ is observed to be $x$ and the 95% confidence interval is computed to be $(a(x,w), b(x,w))$, it is tempting but not appropriate to say that the probability that $a(x,w) \leq p \leq b(x,w)$ is 0.95. The reason this is incorrect is that there is no randomness associated with $x$, $w$, $p$, or the deterministic functions $a()$ and $b()$, so it makes no sense to speak of probabilities relating only to them. For the particular value of $X$ observed, $x$, the resulting interval $(a(x,w), b(x,w))$ either contains $p$ or it doesn't.

Instead, one should speak of the probability relating to $(a(X,w), b(X,w))$. Thus, we can properly say that a priori a $w$-trial simulation has a 95% probability of producing a confidence interval that contains $p$ (even though any particular confidence interval from a $w$-trial simulation either definitively does or or does not contain it); or, equivalently, that if the $w$-trial simulation is repeated an unbounded number of times, 95% of the computed confidence intervals will contain the parameter $p$.

#### A. Exact Confidence Interval Using the Binomial Distribution

A robust confidence interval may be derived using the actual binomial distribution being observed, with no approximation. In this section, we review the approach presented in [4]. The probability that the true value $p$ is outside of a $\beta$-confidence interval $(a(X,w), b(X,w))$ is $1-\beta$. If we constrain the interval to have the property that the probability that the true value $p$ is below the interval is the same as the probability that $p$ is above the interval, i.e., $\alpha \triangleq (1-\beta)/2$ in each case, the result is the Clopper-Pearson confidence interval $(a, b)$, where $a$ is defined so that a binomial random variable with success-probability $a$ will have probability $\alpha$ of having at least $x$ successes in $w$ trials,

$$P[\text{Bin}(w; a) \geq x] = \alpha \tag{2}$$

and where $b$ is defined so that a binomial random variable with success-probability $b$ will have probability $\alpha$ of having at most $x$ successes in $w$ trials,

$$P[\text{Bin}(w; b) \leq x] = \alpha. \tag{3}$$

A direct computation of this using the binomial distribution can be unwieldy for large $w$. To compute these probabilities in a numerically robust way we can use the relationship between a binomial cumulative distribution and the incomplete beta function:

$$P[\text{Bin}(w; b) \leq x] = I_{1-b}(w-x, x+1) \tag{4}$$

where $I_y(c,d)$ is given by

$$I_y(c,d) \triangleq \frac{1}{B(c,d)} \int_0^y t^{c-1}(1-t)^{d-1} dt \tag{5}$$

and where $B(c,d)$ is the beta function given by

$$B(c,d) \triangleq \int_0^1 t^{c-1}(1-t)^{d-1} dt = \frac{(c-1)!(d-1)!}{(c+d-1)!} \tag{6}$$

when $c$ and $d$ are integers.

Thus, the $\beta$-confidence interval $(a,b)$ is given by the solution to:

$$1 - I_{1-a}(w-x+1, x) = \frac{1-\beta}{2} = I_{1-b}(w-x, x+1). \tag{7}$$

Evaluating this numerically is straightforward, if cumbersome. Fortunately, the intervals so calculated are insentive to $w$, in the sense that the normalized interval

$$(\bar{a}, \bar{b}) \triangleq \begin{cases} \left(\frac{a}{\hat{p}}, \frac{b}{\hat{p}}\right) & \text{if } \hat{p} > 0 \\ (aw, bw) & \text{if } \hat{p} = 0 \end{cases} \tag{8}$$

to three significant digits does not depend on $w$ when $w$ is greater than about 2000. Table 1 lists the normalized confidence intervals $(\bar{a}, \bar{b})$, for small values of $x$ and $w > 2000$, as computed by a C program. The table can be used to compute the 90%, 95%, or 99% confidence interval $(a,b)$, by

$$(a,b) \triangleq \begin{cases} (\bar{a}\hat{p}, \bar{b}\hat{p}) & \text{if } \hat{p} > 0 \\ \left(\frac{\bar{a}}{w}, \frac{\bar{b}}{w}\right) & \text{if } \hat{p} = 0 \end{cases} \tag{9}$$

where, recall, $\hat{p} = x/w$ is the observed CWER.

From the first row in Table 1, we see that the upper end of the 95% confidence in an error-free simulation of $w$ codewords is $3.69/w$, which establishes Rule of Thumb 1. From the second column in Table 1, we have $b/a = \bar{b}/\bar{a} < 10$ when $x \geq 4$, which establishes Rule of Thumb 2.

This confidence interval can also be computed by the BERCONFINT function of MATLAB[1], subject to a caveat: the BERCONFINT function seems to be numerically inaccurate in some low-probability cases. For example, Figure 1 shows the 95% confidence intervals when observing exactly $x = 1$ error in $w$ trials, for varying $w$, as computed with BERCONFINT(1,$w$,0.95) in MATLAB 2013b, and with a C program which numerically solves (7). In MATLAB, the lower endpoint of the interval appears to be inaccurate below about $10^{-9}$.

From Figure 1, it is clear that no matter how long a simulation is run, if only $x = 1$ event is observed, the upper and lower ends of the 95% confidence interval will remain about two and a half decades apart, consistent with the $x = 1$ entry in Table 1.

---

[1]MATLAB, version 8.2.0 (R2013b), The MathWorks Inc., Natick, Massachusetts, 2013.

**Table 1. Normalized confidence interval $(\bar{a}, \bar{b})$ for CWER when $x$ codeword errors are observed.**

| | $(\bar{a}, \bar{b})$ at confidence level: | | |
|---|---|---|---|
| $x$ | 90% | 95% | 99% |
| 0 | (0,3.00) | (0,3.69) | (0,5.30) |
| 1 | (0.0513,4.74) | (0.0253,5.57) | (0.00501,7.43) |
| 2 | (0.178,3.15) | (0.121,3.61) | (0.052,4.64) |
| 3 | (0.273,2.58) | (0.206,2.92) | (0.113,3.66) |
| 4 | (0.342,2.29) | (0.272,2.56) | (0.168,3.15) |
| 5 | (0.394,2.10) | (0.325,2.33) | (0.216,2.83) |
| 6 | (0.436,1.97) | (0.367,2.18) | (0.256,2.61) |
| 7 | (0.469,1.88) | (0.402,2.06) | (0.291,2.45) |
| 8 | (0.498,1.80) | (0.432,1.97) | (0.321,2.32) |
| 9 | (0.522,1.75) | (0.457,1.90) | (0.348,2.22) |
| 10 | (0.543,1.70) | (0.480,1.84) | (0.372,2.14) |
| 15 | (0.616,1.54) | (0.560,1.65) | (0.460,1.88) |
| 20 | (0.663,1.45) | (0.611,1.54) | (0.518,1.73) |
| 25 | (0.695,1.40) | (0.647,1.48) | (0.560,1.64) |
| 30 | (0.720,1.36) | (0.675,1.43) | (0.592,1.57) |
| 35 | (0.739,1.33) | (0.697,1.39) | (0.618,1.52) |
| 40 | (0.755,1.30) | (0.714,1.36) | (0.640,1.48) |
| 45 | (0.768,1.28) | (0.729,1.34) | (0.658,1.45) |
| 50 | (0.779,1.27) | (0.742,1.32) | (0.673,1.43) |
| 75 | (0.818,1.21) | (0.787,1.25) | (0.728,1.34) |
| 100 | (0.841,1.18) | (0.814,1.22) | (0.761,1.29) |
| 200 | (0.887,1.12) | (0.866,1.15) | (0.827,1.20) |
| 400 | (0.919,1.09) | (0.904,1.10) | (0.876,1.14) |
| 500 | (0.928,1.08) | (0.914,1.09) | (0.889,1.12) |
| 750 | (0.941,1.06) | (0.930,1.07) | (0.908,1.10) |
| 1000 | (0.949,1.05) | (0.939,1.06) | (0.920,1.08) |

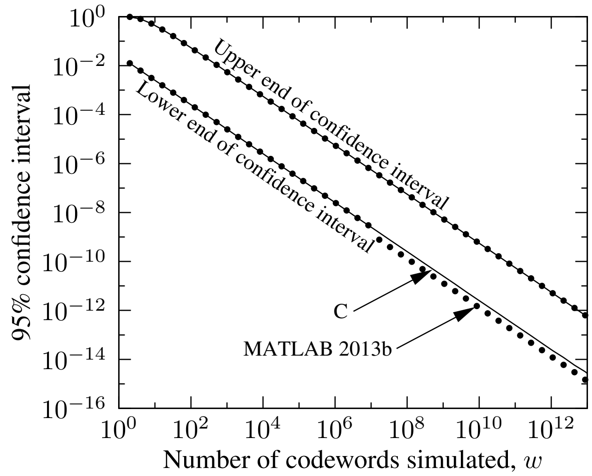


**Figure 1. The confidence interval after observing 1 codeword error in $w$ codewords, for various $w$. The BERCONFINT function in MATLAB is not always accurate at low probabilities.**

### B. Gaussian Approximation

Generally, the confidence interval of the previous section, which involves no approximation, is the best approach and computers can easily compute the confidence intervals. Nevertheless, approximations remain conceptionally and computationally easier, and are still popular.

One simple approach to confidence intervals is to note that by the Central Limit Theorem (CLT), as $w \to \infty$, the probability density function of $\hat{p}$ approaches that of a Gaussian random variable with mean $p$ and variance $p(1-p)/w$ [5]. The mean and variance may be estimated by $\hat{p}$ and $\hat{p}(1-\hat{p})/w$, respectively, which leads to the $\beta$-confidence interval

$$(\hat{p} - a, \hat{p} + a) \tag{10}$$

where

$$a = \sqrt{\frac{\hat{p}(1-\hat{p})}{w}} \cdot Q^{-1}(\alpha) \tag{11}$$

where $Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$ is the tail probability of a zero-mean, unit-variance Gaussian random variable, and as before, $\alpha \triangleq (1-\beta)/2$. For a 95% confidence interval, $Q^{-1}(0.025) \approx 1.96$.

Massey points out that when $X$ is small, $\hat{p}(1-\hat{p})/w$ is not a good estimate of the variance of $X$ [1]. In addition, the Gaussian distribution itself is not a good approximation when $pw$ is small. In fact, no matter how large $w$ is, if $X < 4$ the confidence interval includes a negative range!

A simulation that collects $X \geq 385$ codeword errors will be able to estimate the CWER to within an error of 10%, with 95% confidence, because from (11),

$$\frac{a}{\hat{p}} = \frac{Q^{-1}(0.025)}{\hat{p}}\sqrt{\frac{\hat{p}(1-\hat{p})}{w}} < \frac{Q^{-1}(0.025)}{\sqrt{\hat{p}w}} < \frac{1.96}{\sqrt{385}} < 0.1 \tag{12}$$

where we have used $1-\hat{p} < 1$, $Q^{-1}(0.025) < 1.96$ and $\hat{p} \geq 385/w$. In general, when the length of the $\beta$-confidence interval for the CWER is desired to be shorter than plus or minus $100\gamma\%$ from the estimate, the simulation should be run until $X \geq (Q^{-1}(\alpha)/\gamma)^2$. This establishes Rule of Thumb 3, which is useful for the communications engineer to know how long to run a simulation to get a desired accuracy for the CWER.

#### C. Poisson Approximation

Massey introduced what is now a standard method to overcome some of the limitations of the Gaussian approximation above, by noting that when $w \gg 1$ and $p \ll 1$, $X$ is approximately Poisson distributed, with mean $\lambda$ and variance $\lambda$, where $\lambda \triangleq wp$. Using the Poisson probability mass function $f_X(i) = \frac{\lambda^i}{i!}e^{-\lambda}$ leads to the $\beta$-confidence interval $(\lambda_L, \lambda_H)$ for $\hat{p}w$, given by the solution to

$$\sum_{i=x}^{w} \frac{\lambda_L^i}{i!} e^{-\lambda_L} = \alpha \tag{13}$$

$$\sum_{i=0}^{x-1} \frac{\lambda_H^i}{i!} e^{-\lambda_H} = \alpha \tag{14}$$

where again, $\alpha \triangleq (1-\beta)/2$. The confidence interval for $\hat{p}$, then, is $(\lambda_L/w, \lambda_H/w)$. This solution works adequately for $x < 5$ [1], and for $x > 10$ it is nearly identical to the Gaussian approximation discussed above [4].

#### D. Aborted Simulations

Imagine that a simulation of a channel code is set up to abort after the first codeword error is observed, and let the random variable $W$ denote the number of codewords simulated. How does the estimate $\hat{p} = 1/W$ relate to the true probability of error $p$?

It turns out that $\hat{p}$ is a biased estimate of $p$:

$$E[\hat{p}] = \sum_{w=1}^{\infty}(1-p)^{w-1}p \cdot \frac{1}{w} = \frac{p}{1-p}\sum_{w=1}^{\infty}\frac{(1-p)^w}{w} = p \cdot \frac{-\ln(p)}{1-p}. \tag{15}$$

The bias factor, $[-\ln(p)]/(1-p)$, is greater than 10 when $p < 10^{-6}$, as is seen in Table 2. This establishes Rule of Thumb 4.

On the other hand, the expected value of $W$ is

$$E[W] = \sum_{i=0}^{\infty}(1-p)^i p(i+1) = \frac{1}{p}. \tag{16}$$

**Table 2. The bias in a simulation that aborts after first codeword error.**

| $p$ | Bias factor |
|---|---|
| $10^{-1}$ | 2.56 |
| $10^{-3}$ | 6.91 |
| $10^{-6}$ | 13.82 |
| $10^{-9}$ | 20.72 |

That is, if there is a one in a million chance of simulating a codeword error, one should expect to simulate about a million codewords to see the first codeword error.

Given $W = w$, what is the maximum likelihood (ML) estimate of $p$? This is given by finding the $p$ which maximizes the probability mass function of $W$ at $W = w$:

$$\hat{p} = \underset{p\in[0,1]}{\operatorname{argmax}}(1-p)^{w-1}p. \tag{17}$$

When $w = 1$, $\hat{p} = 1$. When $w > 1$, we may solve

$$\frac{d}{dp}(1-p)^{w-1}p = 0 \tag{18}$$

to see that $\hat{p} = 1/w$. So the ML estimate of $p$ is the observed value, $1/w$, even though this is a biased estimate.

## IV. Confidence Intervals for the BER of a Coded System

We turn now to the novel contribution of this article. We desire to determine the confidence interval for the BER of a coded communications system, based on simulations of the decoder. As we mentioned earlier, if the system is uncoded and a binary-input memoryless channel is used, the bit errors would be independent and the methods in Section III could be directly applied. The analysis in this section applies to any situation in which block errors occur and bit errors occur (only) within block errors. As such, it could also be used to analyze the BER of a link using higher-order modulation, whether uncoded or coded.

### A. Early Identification of the Problem

Massey recognized in 1976 that while the procedure in Section III-C is useful for computing CWER confidence intervals, it cannot be used in the same way to compute BER confidence intervals [1]: "It probably should be pointed out that, although 256 information bits are decoded in each frame so that there are 256 times as many bit decoding decisions as frame decoding decisions, one cannot assert greater statistical confidence in the observed decoding bit error probability than in the observed frame error probability. The reason of course is that the decodings of bits within a frame are highly dependent so that one has no more independent bit decoding decisions from which to infer probabilities than one has independent frame decoding decisions."

In other words, counting $x$ bit errors in $w$ total simulated bits and applying the formulas (13) and (14) would result in an inappropriately narrow confidence interval for the BER.

Much of the literature on confidence intervals for BER treats the case in which bit errors are independent events (e.g., [6, 7]), and MATLAB's function BERCONFINT() to compute the confidence interval for BER also assumes this[2]. While there has been some work on BER confidence intervals on channels with memory [8], to our knowledge the computation of confidence intervals for a block-coded system has not been presented.

#### B. A BER Confidence Interval Based on a Gaussian Approximation

Suppose $w$ codewords of a binary $(n, k)$ code are simulated, and for the sake of analysis, suppose that the decoder is required to output an estimate of the information bits in each codeword, whether it successfully completes decoding the codeword or not. Let $B_i$ be a random variable representing the number of bit errors, $1 \leq B_i \leq k$, in the $i^{\text{th}}$ information block at the output of the decoder. Then $B_1, \ldots, B_w$ is a set of i.i.d. random variables. Let

$$\mu \triangleq E[B_i] \tag{19}$$

$$\sigma^2 \triangleq \text{var}[B_i] = E[B_i^2] - \mu^2. \tag{20}$$

Typically, the distribution of $B_i$ is unknown. For modern iteratively decoded channel codes such as low-density parity-check codes, the distribution of $B_i$ may depend on the SNR of the simulation, or details of the decoder, *even when conditioned on the event that a codeword error has occurred*. For example, at low SNR when codeword errors are dominated by the decoder's failure to converge, many bit errors may occur in each codeword error, while at high SNR where the decoder performance is limited by the code's minimum distance or trapping sets, only a handful of bit errors might typically occur in each codeword error (and of course, the CWER itself is lower).

Let $p_b$ denote the true BER. The number of bits simulated is $kw$, so the observed BER is given by

$$\hat{p}_b = \frac{1}{kw} \sum_{i=1}^{w} B_i.$$

As $w \to \infty$, by the CLT we have

$$\hat{p}_b \sim N\left(\frac{\mu}{k}, \frac{\sigma^2}{k^2 w}\right).$$

---

[2]MATLAB, version 8.2.0 (R2013b), The MathWorks Inc., Natick, Massachusetts, 2013.

For large $w$ we may estimate the first and second moments of $B_i$ by their sample first and second moments:

$$\mu \approx \hat{\mu} \triangleq \frac{1}{w}\sum_{i=1}^{w} B_i = k\hat{p}_b \tag{21}$$

$$\sigma^2 \approx \hat{\sigma}^2 \triangleq \left(\frac{1}{w}\sum_{i=1}^{w} B_i^2\right) - \hat{\mu}^2 \tag{22}$$

where, as before, for convenience we use the biased sample variance instead of the unbiased sample variance. Thus, a $\beta$-confidence interval for the BER can be given by $(\hat{p}_b - a', \hat{p}_b + a')$, with

$$a' = \frac{\hat{\sigma}}{k\sqrt{w}} \cdot Q^{-1}(\alpha) \tag{23}$$

$$= \frac{1}{k\sqrt{w}}\sqrt{\left(\frac{1}{w}\sum_{i=1}^{w} B_i^2\right) - \left(\frac{1}{w}\sum_{i=1}^{w} B_i\right)^2} \cdot Q^{-1}(\alpha) \tag{24}$$

where $\alpha \triangleq (1-\beta)/2$. A simulation would normally record only $\sum_i B_i$; by also recording one extra quantity, $\sum_i B_i^2$, the confidence interval in (24) may be computed. These two partial sums may be augmented with each new simulated codeword, so that the entire sequence $B_1, B_2, \ldots$ need not be stored. Thus, the confidence interval remains easy to compute.

#### C. BER Confidence Interval When Few Codeword Errors Are Simulated

As with the Gaussian-approximation for the CWER confidence interval, the accuracy of the interval in (24) depends on the accuracy of the approximation in (22). Even with $w$ very large — ensuring the accuracy of the CLT approximation for $\hat{p}_b$ — if only a few $B_i$ are greater than zero, we won't have an accurate estimate of the variance of $\hat{p}_b$. For the CWER, $X > 10$ is sufficient; for the BER, even more are needed.

What can be said when only a few codeword errors have been collected? Since bit errors occur in bunches, not singly, neither the individual bit errors nor the bit errors per codeword, $B_i$, are binomial or Poisson distributed, and Massey's approach [9] cannot be directly applied. When codeword $i$ is in error, $1 \leq B_i \leq k$, so that

$$\frac{\hat{p}}{k} \leq \hat{p}_b \leq \hat{p}, \tag{25}$$

which loosely bounds the confidence interval for BER as $(\lambda_L/(kw), \lambda_H/w)$, where $\lambda_L$ and $\lambda_H$ are given in (13) and (14).

### D. Guideline for Simulation Length

We present now an analogous guideline for how long a simulation of a coded system should be run to get a good estimate of the BER. Let

$$X' \triangleq \left(\sum_{i=1}^{w} B_i\right)^2 \Big/ \sum_{i=1}^{w} B_i^2 \, . \tag{26}$$

**Theorem 1.** *The error for the BER estimate, with $\beta$-confidence, is less than $100\gamma\%$ of the observed BER, $\hat{p}_b$, if $X' > (Q^{-1}(\alpha)/\gamma)^2$, where $\alpha \triangleq (1-\beta)/2$.*

*Proof.* From (24), we have

$$a' < \frac{1}{k\sqrt{w}} \sqrt{\left(\frac{1}{w}\sum_{i=1}^{w} B_i^2\right)} \cdot Q^{-1}(\alpha) \tag{27}$$

$$\leq \frac{\gamma}{kwQ^{-1}(\alpha)} \sum_{i=1}^{w} B_i \cdot Q^{-1}(\alpha) \tag{28}$$

$$= \gamma \hat{p}_b. \tag{29}$$

□

This connects the accuracy of the Gaussian-approximation confidence interval for BER, based on $X'$, to the accuracy of the Gaussian-approximation confidence interval for CWER, based on $X$. Thus, with 95% confidence the BER is within plus or minus 10% of of the simulated BER when $X' \geq 385$. Table 3 summarizes the minimum $X$ or $X'$ a simulation must reach in order to achieve a given accuracy at a given confidence level, for the CWER or BER, respectively, using the Gaussian approximation to the interval discussed in the preceding sections.

## V. Examples

### A. Constant Number of Bit Errors Per Codeword

Suppose the coded system is such that whenever a codeword error is made, the decoded codeword contains exactly $b$ bit errors, where $b$ is a constant. In this case,

$$\hat{p}_b = \frac{1}{kw} \sum_{i=1}^{w} bI\{\text{codeword } i \text{ in error}\} = \frac{b\hat{p}}{k} \tag{30}$$

$$\frac{1}{w}\sum_{i=1}^{w} B_i = \frac{1}{w}\sum_{i=1}^{w} bI\{\text{codeword } i \text{ in error}\} = b\hat{p} \tag{31}$$

$$\frac{1}{w}\sum_{i=1}^{w} B_i^2 = \frac{1}{w}\sum_{i=1}^{w} b^2 I\{\text{codeword } i \text{ in error}\} = b^2\hat{p} \tag{32}$$

**Table 3. Minimum $X$ (CWER) or $X'$ (BER) for a simulation to achieve a given level of accuracy at a given level of confidence.**

| Error in | $X$ or $X'$, at Confidence: | | |
|---|---|---|---|
| CWER or BER | 90% | 95% | 99% |
| 1% | 27056 | 38415 | 66349 |
| 2.5% | 4329 | 6147 | 10616 |
| 5% | 1083 | 1537 | 2654 |
| 10% | 271 | 385 | 664 |
| 25% | 44 | 62 | 107 |
| 50% | 11 | 16 | 27 |

$X$ = Number of codeword errors
$X'$ = Given by (26)

where $I$ is the indicator function and, as before, $\hat{p}$ is the observed CWER. Plugging into (24), we have

$$a' = \frac{1}{k\sqrt{w}}\sqrt{b^2\hat{p} - b^2\hat{p}^2} \cdot Q^{-1}(\alpha) \tag{33}$$

$$= \frac{b}{k}\sqrt{\frac{\hat{p}(1-\hat{p})}{w}} \cdot Q^{-1}(\alpha) \tag{34}$$

$$= \frac{b}{k}a \tag{35}$$

where $a$ is given in (11), and so the BER confidence interval is

$$(\hat{p}_b - a', \hat{p}_b + a') = \frac{b}{k}(\hat{p} - a, \hat{p} + a). \tag{36}$$

That is, the BER confidence interval is exactly $b/k$ times the CWER confidence interval in (11), as expected. This means that on a log plot of BER and CWER, the length of the confidence intervals will be the same.

This is an extreme case. Typically, there is some variation in the number of bits-in-error in the simulated codeword-in-error. In those cases, the length of the BER confidence interval is *strictly greater* than that of the CWER confidence interval, reflecting the uncertainty both in the number of codeword errors and in the number of bits in error within codeword containing errors.

### B. A Turbo Code

Among the turbo codes that have been standardized for space communications [10], we consider the one with input length $k = 1784$ and code rate $r = 1/2$. The performance of the

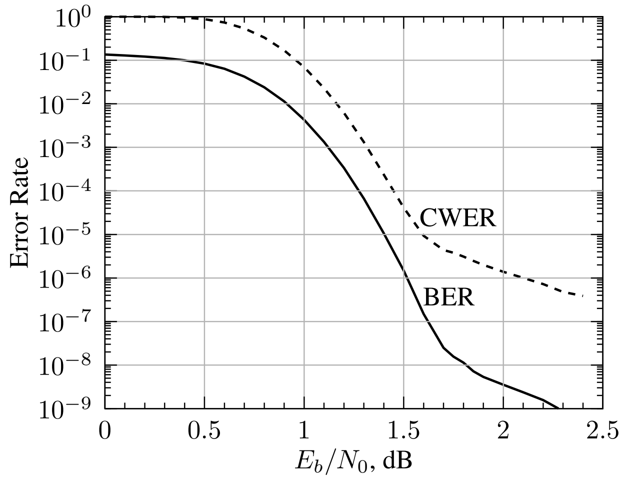


**Figure 2. Performance of the CCSDS $k = 1784$, $r = 1/2$ turbo code.**

code is shown in Figure 2, where it can be seen that an error floor begins at just below CWER$= 10^{-5}$.

A decoder was simulated at $E_b/N_0 = 0$ dB, 1 dB, and 1.7 dB, and the observed distribution of $B_i > 0$ is shown in dark gray, light gray, and black, respectively, in Figure 3. At $E_b/N_0 = 0$ dB and 1 dB, the code is operating in the waterfall region, where codewords that fail to decode correctly have a relatively large number of bit errors. The distribution of these errors is often indistinguishable from random errors that would occur in uncoded transmission at that SNR.

At $E_b/N_0 = 1.7$ dB, however, the code is operating just inside the error floor region. The code has two codewords with input weight 3 and output weight 17, which is the minimum distance of the code. This explains why $B_i = 3$ was observed for about 1/4 of the codewords in error. The code has a total of a few dozen codewords of weight 18, 19, . . . , 28, and at least 836 codewords with input weight 9 and output weight 29 — consistent with $B_i = 9$ being observed in more than 10% of the codewords in error. Despite the fact that $B_i \leq 9$ in more than 82% of the codewords in error, the *average* value of $B_i$ is higher, approximately 10.0. Thus, most codeword errors do not contribute a representative amount to the BER, making it necessary to simulate longer, and check that the guideline in Table 3 holds.

A simulation of about $10^{10}$ codewords was run at $E_b/N_0 = 1.7$ dB. Figure 4 illustrates the 95% confidence intervals for the CWER and BER as the simulation progressed. The confidence intervals for the CWER and BER were computed from (11) and (24), respectively. Since the CWER at this SNR is less than $10^{-5}$, more than $10^6$ simulated codewords were necessary to collect even ten codewords in error, when the Gaussian-approximation confidence intervals begin to be appropriate.

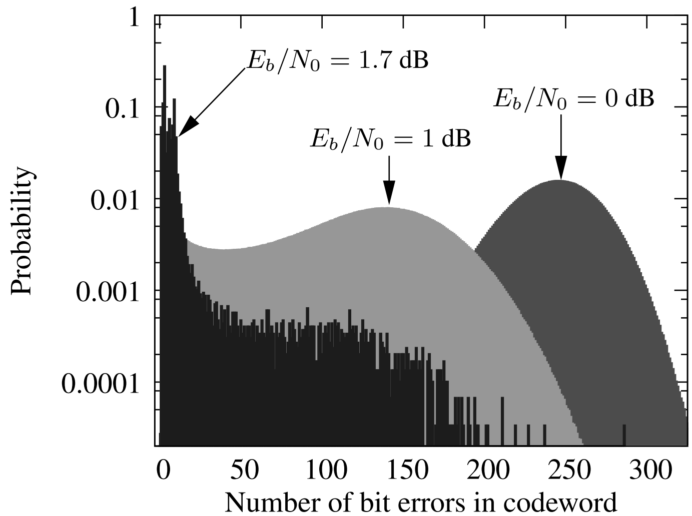


**Figure 3. Observed distribution of $B_i > 0$ at various $E_b/N_0$.**

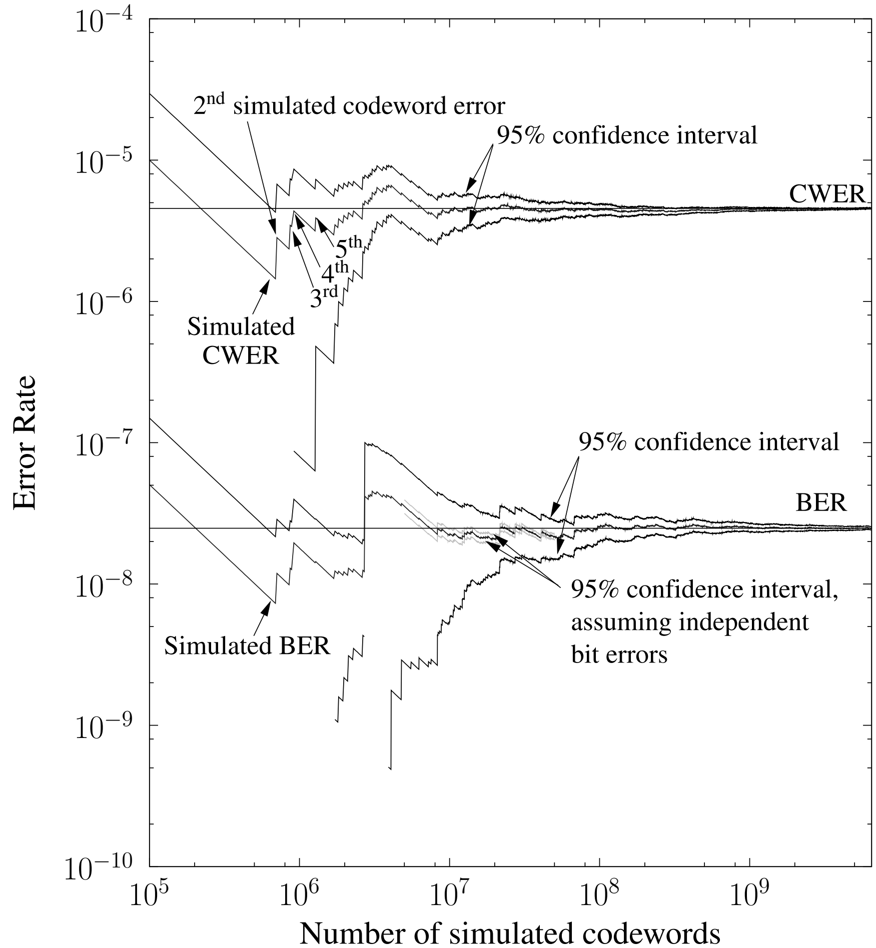


**Figure 4. 95% confidence intervals for CWER and BER at $E_b/N_0 = 1.7$ dB, as a simulation progressed.**

The first few codeword errors are identified in Figure 4. After $5 \times 10^6$ codewords were simulated, the CWER confidence interval is less than one decade thick, while the BER confidence interval is about 1.5 decades. At every point in the simulation, the confidence interval for BER is wider than that of the CWER, as expected. As the simulation of independent codewords progresses, the CWER confidence interval continually shrinks; the BER confidence interval also usually shrinks, except that occasionally codewords with large numbers of bit errors are observed, which can temporarily increase the uncertainty. For example, the first 13 codeword errors observed contained a total of 59 bit errors, but the 14th codeword error alone contained 144 bit errors. This had a large impact on the average BER to that point, and an enormous impact on the BER confidence interval, which became vacuous at the lower end.

In this example simulation, the condition of $X' > 385$ was met when $w \approx 460$ million codewords and $X = 2048$ codeword errors had been simulated. This was more than five times the simulation length required to achieve the same 10% uncertainty in the CWER. This is consistent with the higher observed variation in the simulated BER, compared to the simulated CWER, as the simulation progressed.

Also shown in Figure 4 is a portion of the 95% confidence interval that would be computed if we incorrectly assumed that bit errors were independent. It can be immediately seen that the interval is inappropriately narrow, because the true BER is outside of the confidence interval over wide ranges of the number of simulated codewords.

## VI. Conclusions

Four Rules of Thumb were presented relating to the confidence intervals of the CWER of communications systems. We recommend that the Rules of Thumb be incorporated into test procedures for flight missions, to assist in properly certifying that error rate requirements are met with the desired level of confidence.

We caution against computing a BER confidence interval from a simulation or tests of coded systems, if the method for doing so assumes bit errors are statistically independent. We provide a proper method to compute the confidence interval for the BER, and show that because of the variation in the number of bits in error per codeword, verifying the BER to a given level of fidelity requires a longer simulation or test than is needed verify the CWER to the same level of fidelity. In the example shown, which is typical, the BER simulation needed to be run five times as long as the CWER simulation.

This suggests that for coded systems, tests of CWER are generally to be preferred over BER tests, because they can be shorter and more accurate. This conclusion is strenghtened by noting that during mission operations, codewords in error are detected and discarded, regardless of the number of bit errors they contain, which makes CWER a more relevant metric than

BER. The JPL Design Principles[3] reinforce this approach, by stating the telecommunications error rate requirements in terms of CWER, and not BER.

## References


[1] J. L. Massey, "Comparison of Rate One-Half, Equivalent Constraint Length 24, Binary Convolutional Codes for Use with Sequential Decoding on the Deep-Space Channel," Tech. Rep., University of Notre Dame, Department of Electrical Engineering, Apr. 1976.

[2] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, New Jersey, 1983.

[3] Howard Ma and Jack Wolf, "On Tail Biting Convolutional Codes," *Communications, IEEE Transactions on*, vol. 34, no. 2, pp. 104–111, 1986.

[4] Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan, *Simulation of Communication Systems*, Kluwer Academic/Plenum, New York, second edition, 2000.

[5] H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*, Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1994.

[6] C. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," *Selected Areas in Communications, IEEE Journal on*, vol. 2, no. 1, pp. 153–170, 1984.

[7] S. Berber, "Bit Error Rate Measurement with Predetermined Confidence," *Electronics Letters*, vol. 27, no. 13, pp. 1205–1206, 1991.

[8] M.D. Knowles and A.I. Drukarev, "Bit Error Rate Estimation for Channels with Memory," *Communications, IEEE Transactions on*, vol. 36, no. 6, pp. 767–769, 1988.

[9] James L. Massey, "All Signal Sets Centered About the Origin are Optimal at Low Energy-to-Noise Ratios on the AWGN Channel," in *Proceedings of the International Symposium on Information Theory*, Ronneby, Sweden, June 1976, pp. 80–81.

[10] "CCSDS 131.0-B-2. TM Synchronization and Channel Coding. Blue Book. Issue 2," Aug. 2011.


[3] Brian Muirhead, "JPL Design, Verification/Validation & Ops Principles for Flight Systems (Design Principles), Rev. 6," (internal document), Jet Propulsion Laboratory, Pasadena, California, October 4, 2012.

JPL CL#15-1635