誤差確率推定の統計的信頼レベル

ディジタル通信システムの多くの部品は、ビットエラー確率(P(ε))の最小仕様を満たしている必要があります。所与のシステムにおけるP(ε)の推定は、入力に印加された所定のパターンと出力ビットパターンを比較することによって行われます。入力と出力ビットストリームの間に不一致があるとエラーとしてフラグが立ち、検出されたビットエラー(ε)と伝送された全ビット数(n)との比がP′(ε)となります。プライム符号は、これが実際のP(ε)の推定値であることを示します。この推定値の質は、伝送されるビットの総数が増えるほどよく高くなります。この関係は次式で表されます。

$$P'(\varepsilon) = \frac{\varepsilon}{n} \xrightarrow[n \to \infty]{} P(\varepsilon) \quad \textbf{[式1]}$$

P′(ε)が実際のP(ε)の妥当な近似値であることを保証するには、システムを通じて十分な数のビットが伝送されることが重要です(即ち、P(ε)はテストを無限に続けた場合に得られる値となります)。このため、テスト時間の妥当なリミットを設けるには、統計的に有効なテストにするための最小ビット数を知る必要があります。

多くの場合、P(ε)が所定の基準と同等あるいはそれよりも良好であることを確かめれば十分です。つまり、P(ε)がある上限以下であることを証明します。例えば、多くのテレコムシステムで必要とされるP(ε)は10^{-10}以下(上限が10^{-10})となっていますが、信頼レベルを上限と結びつける統計的な考え方を使うことにより、数量化可能な確度で、実際のP(ε)が仕様のリミットより小さいと結論することができます。この方法の主な利点は、テスト時間と測定精度の間の妥協点をみつけることが可能であることです。

統計的信頼レベルの定義

統計的信頼レベルは、ある事象の実際の確率が指定されたレベルよりも良好である確率として定義されます。この確率は一連の測定値に基づいています。(この定義においては、実際の確率とは、試行回数が無限に向かった時にリミット内で測定される確率を意味します。) P(ε)の推定の際、統計的信頼レベルは(伝送されたnビットのうち、εのエラーが検出されたということに基づき)実際のP(ε)が指定されたレベルγ(例えば10^{-10})よりも良好な確率であると結論できます。数学的には、これは次式で表されます。

$$CL = P\left[P(\varepsilon) < \gamma \,\middle|\, \varepsilon, n\right] \quad \textbf{[式2]}$$

ここで、P[ ]は確率を意味し、CLは信頼レベルです。信頼レベルは定義上確率であるため、可能な値は0%～100%の範囲です。

信頼レベルを計算すると、P(ε)がγよりも良好であるという信頼レベルがCLパーセントであるといえます。別の言い方をすると、ビットエラーテストを多数回繰り返して各テスト期間でP′(ε) = ε/nを計算した場合、測定のうちのCLパーセントにおいてP′(ε)がγより良好であることが期待できるということです。

信頼レベルの計算

信頼レベルの計算は、多くの統計学の教科書で説明されている二項分布関数に基づいています[1、2]。二項分布関数は一般に次式で表現されます。

$$P_n(k) = \binom{n}{k} P^k q^{n-k}, \text{ where } \binom{n}{k} \text{ is defined as } \frac{n!}{k!(n-k)!}$$

[式3]

式3は、n回の試行(即ちnビットの伝送)においてk回のイベント(即ちビットエラー)が起こる確率を与えます。ここで、pは1回の試行でイベントが発生する確率(即ち1つのビットエラー)、qは1回の試行でイベントが発生しない確率(即ちビットエラーなし)を表わしています。二項分布は2つの可能な結果(例えば成功/失敗、表/裏、エラー/ノーエラー等)を持つイベントのモデルであることに注意して下さい。つまり、p + q = 1です。

N回の試行でイベントが起こる回数がN回以下である確率(あるいは逆にN回より多く起こる確率)を知りたい場合、累積二項分布関数(式4)が有用です。

$$P(\varepsilon \le N) = \sum_{k=0}^{N} P_n(k) = \sum_{k=0}^{N} \left(\frac{n!}{k!(n-k)!}\right) p^k q^{n-k}$$

$$P(\varepsilon > N) = 1 - P(\varepsilon \le N) = \sum_{k=N+1}^{n} \left(\frac{n!}{k!(n-k)!}\right) p^k q^{n-k}$$

[式4]

式3及び式4をグラフで表現して、特性を説明したのが図1です。

二項分布関数

標準的な信頼レベル測定においては、十分な信頼レベルを選択し、p(1つのビットを伝送する時のビットエラーの確率)を仮定します。選ばれたp値はp_hで表されます。一般に、これらの値は仕様が要求するリミットに基づいて選びます(例えば、仕様が$P(\varepsilon) \leqq 10^{-10}$であれば、$p_h = 10^{-10}$を選び、信頼レベルとして、例えば99%を選びます)。

次に、(全部でnビットが伝送された時にN回より多くビットエラーが発生する確率p_hに基づいて)式4を使用し、$P(\varepsilon > N | p_h)$を求めます。実際のテスト中に($P(\varepsilon > N | p_h)$が高いにもかかわらず)N回より少ないビットエラーが起こった場合、次の2つの結論のうちの1つが妥当とされます：つまり、(a)単なる幸運、あるいは(b)実際のpの値がp_hよりも小さい。テストを何回も行った後で引き続きビットエラーがN回よりも少ない場合は、結論(b)の信頼性が増してきます。

$P(\varepsilon > N | p_h)$は、結論(b)に対する我々の信頼レベルを定義する数値です。$p_h = p$である場合、N回よりも多いビットエラーを検出する確率は高いといえます。エラーがN回よりも少ない場合は、pがおそらくp_hよりも小さいと結論し、信頼レベルはこの結論が正しい確率として定義されます。別の言い方をすると、$P(\varepsilon)$(ビットエラーの実際の確率)がp_hよりも小さいことによって、CL%の信頼度を持っていることになります。

累積二項分布関数を使用すると、信頼レベルは次式で定義されます。

$$CL = P\left(\varepsilon > N \mid p_h\right) =$$

$$1 - \sum_{k=0}^{N} \left(\frac{n!}{k!\left(n-k\right)!}\right)\left(p_h\right)^k \left(1 - p_h\right)^{n-k} \quad \text{[式5]}$$

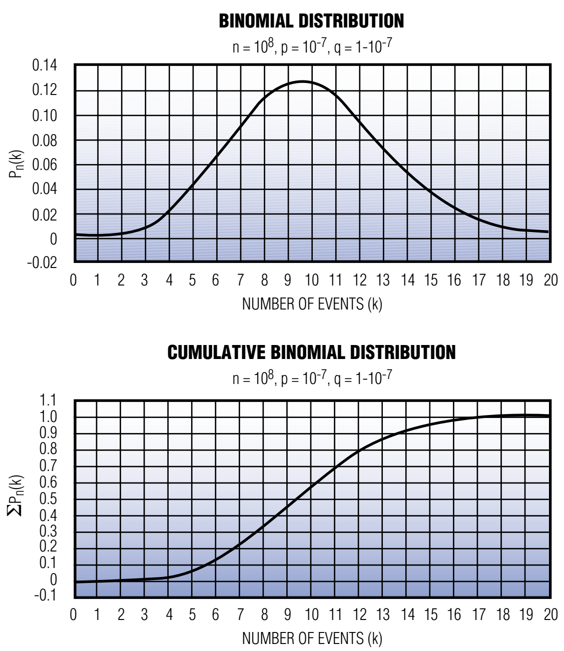
ここで、CLはパーセントで表された信頼レベルです。

前述のように、信頼レベル法を使用する場合、一般に希望する信頼レベル(CL)と共にpの仮想値(p_h)を選び、次に式5を解いて仮定を証明するためにシステムを通してどれだけの数のビット(n)を(N以下のエラーで)伝送しなければならないかを求めます。近似を行わないと、nとNについて解くことは困難です。

np > 1(即ちビットエラーレートの数学的逆数だけのビットを伝送)で、kがnpと同じ桁の大きさである場合、ポアッソンの定理[1](式6)によって二項分布関数の推定値が得られます。

$$P_n\left(k\right) = \left(\frac{n!}{k!\left(n-k\right)!}\right) p^k q^{n-k} \xrightarrow[n \to \infty]{} \frac{\left(np\right)^k}{k!} e^{-np}$$

[式6]



$$P_n\left(k\right) = \left(\frac{n!}{k!\left(n-k\right)!}\right) p^k q^{n-}$$

n = Total number of trials (i.e., total bits transmitted)

k = Number of events occurring in n trials (i.e., bit errors)

p = Probability that an event occurs in one trial (i.e., probability of bit error)

q = Probability that an event does not occur in one trial (i.e., probability of no bit error)

$p + q = 1$
Mean (μ) = nq
Variance (σ^2) = npq

$$CL = 1 - \sum_{k=0}^{N} P_n\left(k\right)$$

図1. 二項及び累積二項分布は試行回数と実測エラーをエラーが起こる(あるいは起こらない)確率に結びつけます。

式7は、式6を使用して累積二項分布の近似値を得る方法を示しています

$$\sum_{k=0}^{N} P_n(k) \approx \sum_{k=0}^{N} \frac{(np)^k}{k!} e^{-np} \quad \textbf{[式7]}$$

式5と式7を組み合わせてnについて解くと、以下が得られます。

$$\sum_{k=0}^{N} P_n(k) = 1 - CL \quad (\textbf{式5を移項})$$

$$\sum_{k=0}^{N} \frac{(np)^k}{k!} e^{-np} = 1 - CL \quad (\textbf{式7を使用})$$

$$-np = 1n\left[\frac{1-CL}{\sum_{k=0}^{N} \frac{(np)^k}{k!}}\right] \quad n = -\frac{1n(1-CL)}{p} + \frac{1n\left(\sum_{k=0}^{N} \frac{(np)^k}{k!}\right)}{p}$$

[式8]

式8の第2項はN = 0の時にゼロであり、その場合は簡単に解けることに注意して下さい。N > 0の場合、式8は解くことがより難しくなりますが、コンピュータを利用して実験的に解くことができます。これで、希望の信頼レベルを達成するためにシステムを通して伝送しなければならないビットの総数を求めることができます。この手順の例を以下に示します。

1) pの仮想値であるp_hを選択します。この値は、我々が確かめたいビットエラーの確率です。例えば、P(ε)≦10^{-10}であることを示す場合、式8の中のpをp_h = 10^{-10}に設定します。

2) 希望の信頼レベルを選択します。ここでは、信頼性とテスト時間の間の妥協が必要です。テスト時間を短くするには、妥当な最低の信頼レベルを選んで下さい。テスト時間と信頼レベルの妥協点は-1n (1 - CL)に比例します。図2を参照して下さい。

3) 式8をnについて解きます。殆どの場合、この作業はテスト中にビットエラーが発生しない(即ちN = 0)と仮定することによって単純になります。

4) テスト時間を計算します。テストを完了するために必要な時間はn/Rです(Rはデータ速度)。

CLを使用してP(ε)を推定

多くのテレコムシステムにおいては、P(ε)が10^{-10}よりも良好であることが仕様で指定されています。2つのクロック/データリカバリチップMAX3675(622Mbps)とMAX3875(2.5Gbps)をテストして、この仕様に適合しているかどうか確かめなければならないと仮定します。

まず、p_h = 10^{-10}と設定します。希望の仕様について100%の信頼性を与えるテストを望むところですが、そうなるとテスト時間が無限大になってしまうため、信頼レベル99%で妥協します。次に、様々なNの値(0、1、2、3等)を使用して、式8をnについて解きます。結果は下記の表1に示されています。

表1から、(2.5Gbpsシステムにおいて)18.5sの間ビットエラーが検出されなければP(ε)≦10^{-10}であると99%の信頼レベルで結論を出すことができます。26.7sで1ビットエラーが起こった場合、あるいは33.7sの間に2ビットエラーが起こった場合にも結果は同じで、P(ε)≦10^{-10}であると99%の信頼レベルで結論を出すことができます。

MAX3675/MAX3875のための標準P(ε)テストを開発する場合は、例えば表1のN = 3に対応するテスト時間を選択できます。ビットエラーレートテスター(BERT)を使用して、2つのチップのそれぞれを通して10^{11}ビットを伝送します。10^{11}ビットのテスト時間は622Mbpsで2分41秒、2.5Gbpsで40.2秒です。テスト時間の最後に、検出されたビットエラーの数(ε)をチェックします。ε≦3であればその素子は合格で、P(ε)≦10^{-10}であると99%の信頼レベルで結論を出すことができます。

テスト時間を減らすために システムにストレスをかける方法

Dan Wolaverは、システムにストレスをかけることによってテスト時間を短縮する方法を説明しています(3)。これは、ビットエラーの主要な原因がレシーバの入力におけるサーマル(ガウス)ノイズであるという前提に基づいています。(この前提は、ジッタ等、その他のエラーの原因になりうる要素を無視していることに注意して下さい。)この前提が有効であるようなシステムにおいては、伝送経路に固定減衰を挿入することによって信号対雑音比

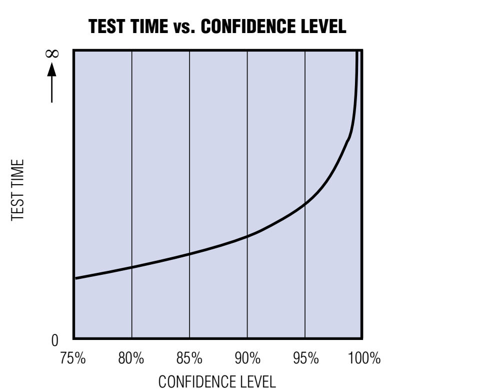


図2. (指定のエラーレートにおける)信頼レベルはテスト時間の増加と共に高くなります。

表1．ビットエラー確率の推定(例：CL = 99%及びp_h = 10^{-10})

ビットエラー数≦N N =	伝送する必要のある ビット数(n)	ビットレートが622Mbpsの時の テスト時間(秒)	ビットレートが2.5Gbpsの時の テスト時間(秒)
0	$4.61 \cdot 10^{10}$	74.1	18.5
1	$6.64 \cdot 10^{10}$	106	26.7
2	$8.40 \cdot 10^{10}$	135	33.7
3	$1.00 \cdot 10^{11}$	161	40.2
4	$1.16 \cdot 10^{11}$	186	46.6

(SNR)を低減できます(即ち、減衰は信号のみにかかり、主要なノイズ源にはかかりません)。前述の例(MAX3675及びMAX3875)においては、維持リミティングアンプ非直線利得及びジッタ効果がこの方法の基本的な前提に反すると結論されたため、この方法が採用されていません。

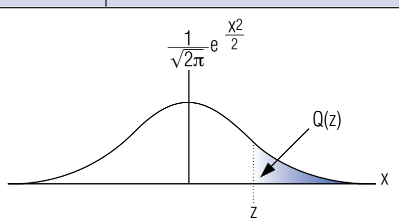
この前提が有効なシステムにおいては、ビットエラーの確率は一般に次式で計算できます[4、5]。

$$P(\varepsilon) = Q\left(\frac{\sqrt{SNR_{electrical}}}{2}\right) = Q\left(\frac{\sqrt{SNR_{optional}}}{2}\right) \quad \text{[式9]}$$

ここで、Q(x)はコンプリメンタリエラー(多くの通信学教科書[6]でQ関数と呼ばれているもの)です。このデータは、他の様々なソース(Microsoft ExcelのNORMDIST関数等)からも得られます。表2に、コンプリメンタリエラー関数の主な値を示します。

表2．コンプリメンタリエラー(Q)関数の値の表

$z = \frac{x - \mu}{\sigma}$	$Q(z) = \frac{1}{\sqrt{2p}} \int_z^{\infty} e^{\frac{x^2}{2}} dx$
3.71	10^{-4}
4.26	10^{-5}
4.75	10^{-6}
5.19	10^{-7}
5.61	10^{-8}
5.99	10^{-9}
6.36	10^{-10}
6.70	10^{-11}
7.03	10^{-12}



式9は、ビットエラーの確率がSNRの減少と共に増加することを示しています。伝送経路に固定減衰(α)が挿入された場合、信号電力(P_S)はファクタαだけ減少しますが、ノイズ電力(P_N)は不変です。このため、SNRはSNR = P_S/P_Nから SNR = $P_S/\alpha P_N$に減少します。対応するP(ε)は、式9及び表2を使用することによって計算できるファクタだけ増加します。

ここで、p_hの修正された値を使用して以前のテスト方法を再び実施することができます。この計算は式9を使用することにより、他の任意のSNRに外挿できます。結果は同じですが、テスト時間は著しく短縮できます。

システムにストレスをかける方法の短所は、測定及び計算をより高精度で行う必要があることです。これは、ストレスのないレベルまで結果を外挿すると、丸め誤差、測定公差等に起因するエラーが何倍にも拡大されるためです。

参考文献

1. Papoulis著, Probability, Random Variables, and Stochastic Processes [確率、ランダム変数及び確率過程]. New York: McGraw-Hill, 1984.
2. K.S. Shanmugan and A.M. Breipohl著, Random Signals: Detection, Estimation, and Data Analysis [ランダム信号：検出、推定及びデータ解析]. New York: John Wiley and Sons, 1988.
3. D.H. Wolaver著, "Measure Error Rates Quickly and Accurately" [エラーレートの迅速、正確な測定], *Electronic Design*, pp. 89–98, May 30, 1995.
4. J.G. Proakis著, Digital Communications [ディジタル通信]. New York: McGraw-Hill, 1995.
5. J.M. Senior著, Optical Fiber Communications: [光ファイバ通信] Principles and Practice (second edition). Englewood Cliffs, New Jersey: Prentice Hall, 1992.
6. B. Sklar著, Digital Communications: Fundamentals and Applications [ディジタル通信：基本と応用]. Englewood Cliffs, New Jersey: Prentice Hall, 1988.