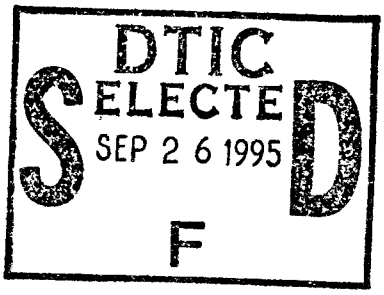
RL-TR-95-113
In-House Report
June 1995

MATHEMATICAL MODELLING OF ERRORS ON A SATELLITE CHANNEL: THEORY AND PRACTICE

J. David Rosen (Carnegie Mellon University)



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

19950922 038

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

DTIC QUALITY INSPECTED 1

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-95-113 has been reviewed and is approved for publication.

APPROVED: Daniel J. McAuliffe

DANIEL J. McAULIFFE
Chief, Communications Division
Command, Control & Communications Directorate

FOR THE COMMANDER: John A. Graniero

JOHN A. GRANIERO
Chief Scientist
Command, Control & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (C3BB) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1995	In-House Jan 92 - Dec 94

4. TITLE AND SUBTITLE
MATHEMATICAL MODELLING OF ERRORS ON A SATELLITE CHANNEL: THEORY AND PRACTICE

5. FUNDING NUMBERS
PE - 62702F
PR - 4519
TA - 22
WU - 24

6. AUTHOR(S)
*J. David Rosen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3BB)
525 Brooks Road
Griffiss AFB NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
RL-TR-95-113

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3BB)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
N/A

11. SUPPLEMENTARY NOTES
*AFOSR Fellow: J. David Rosen (Carnegie Mellon University)
Rome Lab Program Manager: N. P. Robinson/C3BB/(315) 330-1876

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report details a cooperative RL/AFOSR project to provide a laboratory environment for test and evaluation of equipments/protocols that are or will be used in DOD/Federal/Commercial networks utilizing "disadvantaged" communications links with higher error rates than copper or fiber. One way is to utilize Error Injector Units (EIU) previously developed under another Rome Lab effort: "Markov Models for Data Communications." The EIUs are inserted between communication networking equipments to emulate the errors seen over these links, giving the user an idea of actual equipment/protocol performance. Accurate, reliable, transition matrices must be generated for the Markov models used in the EIUs, based on bit error sequences collected by observing live, over-the-air channels during operation. The report focusses on identifying metrics to evaluate the fidelity of a suggested model and proposing methods of generating models which attempt to optimize one or more of these metrics. A procedure is outlined for the use of these models and metrics in a testing environment.

14. SUBJECT TERMS
Asynchronous Transfer Mode (ATM) Error Gap Distribution
Markov Models Maximum Likelihood Estimation (see reverse

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Block 14. Subject Terms

Transition Matrix Autocorrelation Function

Mathematical Modelling of Errors on a Digital Communications Channel: Theory and Practice

J. David Rosen

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Work Performed at:
Rome Laboratory
Griffiss Air Force Base
Rome, NY

N.P. Robinson/C3BB
Program Manager

Accesion For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification .....
By .....
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

Abstract

In order to understand the impact of channel errors on protocols such as ATM, understanding of the error process occurring on a channel is important. Not just the rate of errors, but also their distribution, will have significant impact on how well the protocol runs over the channel, and on what sort of error correction/detection protocol is required to best overcome the disadvantages of the link.

This paper describes the special purpose Error Injection Units (EIUs) at Rome Lab and summarizes methods for generating models for the EIUs which were previously in use at Rome Lab. It then goes on to describe a set of metrics for determining the quality of an EIU model, and proposes other methods of building models for the EIU. Finally, it describes the process to take in order to use and evaluate these models, in support of the Joint Advanced Demonstration Environment's (JADE) project at Rome Lab and other sites around the nation.

1 Introduction

A tactical environment necessitates the use of "disadvantaged" links -- links with relatively low bandwidth and a relatively large number of errors. Furthermore, in order to provide seamless integration in a tactical environment, and thus provide the best possible information services to the warrior, some sort of standard protocol, such as Asynchronous Transfer Mode (ATM) is desirable. In order to understand the impact of channel errors on protocols such as ATM, understanding of the error process occurring on a channel is important. Not just the rate of errors, but also their distribution, will have significant impact on how well the protocol runs over the channel, and on what sort of error correction/detection protocol is required to best overcome the disadvantages of the link.

This paper describes the special purpose Error Injection Units (EIUs) at Rome Lab and summarizes previously developed methods of generating models for the EIUs from actual data. It then goes on to describe a set of metrics for determining the quality of an EIU model. It proposes other methods of building models for the EIU. Finally, it describes procedures for evaluating these methods, both by use of the metrics and through comparison of results with the actual protocols under test.

The remainder of this paper is organized as follows. Section 2 is the background material, including an overview of the operation of the EIUs, and early work done in deriving models based on data. Section 3 proposes a set of metrics, based both on the strengths of the earlier methods and on its perceived weaknesses. Section 4 presents two alternate methods of model building. The first is an adaptation of a model from the literature. The second, which is still under development, is designed to satisfy one of the proposed metrics. Section 5 describes an ideal use of these models, from data collection through model selection. Section 6 con-

cludes the paper, and Section 7 is a bibliography. Finally, Section 8 acknowledges those who had a part in this project.

2 Background

2.1 The Error Injection Units

The error injection units (EIUs) were designed and built for Rome Lab in the early 90's. Their purpose is to emulate a disadvantaged link so that system level tests could be conducted without committing the resources of an actual link of the desired type. The EIUs take as input a digital signal at up to T1 speed, and introduce (or inject) errors into the stream according to a configurable, 8 state Markov chain. The EIUs are configured with a state transition matrix; additionally, each state can be configured to be either an error state or an error free state.

Each EIU is equipped with a pseudo random number generator, and maintains state information. During every bit time, the EIU:

- Generates a random number
- Determines what state to transition into, and makes that transition
- Outputs a bit based on the incoming bit, and the error status of the current state

With appropriate use of the states, it is easy to create many of the 'standard' error models, including Bernoulli errors (independent, identical), 'bursty' errors, and much more general types of error processes.

2.2 Error Gap Modelling

In this section, we present the method of model building previously practiced at Rome Lab, as well as earlier models which are presented here for completeness.

2.2.1 The Bernoulli Model

The simplest error model is that which is implicit made by a variety of analytical equipment, notably, the standard Bit Error Rate Tester (BERT). This model has one parameter -- the average rate at which errors occur. The error status of each bit is assumed to be independent of the error status of each other bit. While this model is attractive in its simplicity, it tends to have poor predictive power over satellite and other radio links, due to time-persistent effects such as channel fading. These effects tend to cause errors to occur in groups, commonly called bursts. This leads us to investigate more complicated error models which take this into account.

2.2.2 Gilbert's Model

The Gilbert error model was first described in [Gilbert60], as a description of errors on a telephone channel. Gilbert assumed that errors occurred in bursts of exponentially distributed length, and that error-free runs were also exponentially distributed in length. Furthermore, he assumed that during an error burst, each bit was in error independently with some probability $1-h$. This model is described more carefully in Section 4.

2.2.3 Tsai's Model

The previous work done in generating models (i.e., transition matrices) for the EIU was primarily done by Dr. Wayne Smith of Mississippi State University. He assumed that the errors arrive in a distribution which is nearly, but not exactly, Poisson. This leads to the approximation of error gap distribution as the sum of exponentials. The resulting model has exactly one error state, with transitions to and from each of the non-error states, which do not interact with one another. This model corresponds closely to those discussed in [Tsai69] and [McManamon70].

The procedure which results for building a model from data is relatively straightforward. Since, given the assumption of one error state, the model is completely determined by the error gap distribution, one should simply build a histogram of error-free-run lengths; fit the histogram as well as possible as a sum of exponentials; and then compute the appropriate transition probabilities, so that the error-free-run distribution will match the fitted curve. Remarks on effective curve fitting can be found in [Smith93].

To restate the result, if we have an approximation to the error gap distribution (that is, f such that $P[0^m 1|1] = f(m) = \sum_{i=1}^{n-1} A_i e^{a_i m}$), and we assume that states 1 through $n-1$ are error-free states and state n is the error state, then the non-zero transitions can be given as

$$p_{ii} = e^{a_i}, \text{ for } i < n \qquad \text{(EQ 1)}$$

$$p_{ni} = A_i e^{a_i}, \text{ for } i < n \qquad \text{(EQ 2)}$$

$$p_{in} = 1 - p_{ii}, \text{ for } i < n \qquad \text{(EQ 3)}$$

$$p_{nn} = 1 - \sum_{i=1}^{n-1} p_{ni} \qquad \text{(EQ 4)}$$

It is relatively straightforward to verify that this set of transitions produces the desired distribution of error free runs.

3 Metrics

There are two primary methods to evaluate the quality of a model. The best method is to compare the performance of the protocol or application of interest when run over the actual channel, with the performance that is predicted by the model. However, this is not always possible or practical. Therefore, metrics must be available to compare the distribution of errors on the actual channel to the distribution of errors produced by the model. In this section, several such metrics will be introduced.

The first is suggested quite directly from the model above. If we believe that the burst-length distribution and/or the error-free-gap distribution are important characteristics of the error process (as is no doubt the case), then these suggest the first of our measures: goodness of fit of error burst distribution, and goodness of fit of error gap distribution. There are a number of ways we could evaluate the "goodness" of fit between these two functions described in [Smith93]. Briefly, Smith proposes use of the sample standard deviation, the correlation coefficient, or the plot of the residuals.

The error burst distribution length can be generalized by using the parameter Δ, as described in [Tsai69]. Here, a burst does not necessarily consist of all errors, but begins and ends with an error, and has a density of errors exceeding Δ. We see

that the previous definition of an error burst is just a special case of the generalization with $\Delta = 1$. Clearly, however, we can compare the gap distributions for different values of Δ, and thereby obtain additional metrics.

The third metric is motivated by considering the error sequence as a random process. Then it is natural to ask about the second order statistics of this random process; in particular, the autocorrelation function. Assuming that the process is wide-sense stationary (WSS), and that we represent the sequence of errors as $e_0, e_1, \ldots$, the autocovariance function is given by

$$R_k = E[e_i e_{i+k}] \qquad \text{(EQ 5)}$$

Since the random process is binary, it is clear that R_k, is exactly the probability that two bits at distance k are simultaneously 1 (in error). In particular, note that $R_0 = E[e_i e_i]$, the correlation of one term, is equal to the Bit Error Rate (BER) (again, because the rp is binary). Also, $\lim_{k \to \infty} R_k = (R_0)^2$, since, in the extreme long term, bits are in error independently.

In any case, the goodness of fit of the autocorrelation function is also a natural metric to use. Once again, we can use any of the methods proposed by [Smith93] to measure the goodness of fit.

Finally, a survey of recent literature provides another model and, with it, another metric. It is natural to consider the probability that a given model produced the observed error sequence. A model which has a relatively higher probability of producing a given sequence is, intuitively, more likely to be the "Hidden Markov Model" producing that sequence. (The Markov Model is hidden because we observe the error sequence, and not the sequence of states through which the

model passes. Since there are, in general, multiple error-producing states and multiple error-free states, we may never be certain what state the model is in.) Note that direct computation of this probability is computationally prohibitive; fortunately, there is a well known method for computing it that is much more efficient. This will be discussed in Section 4.3; the method for optimizing this method is also described in [Poritz88],[Turin93], and [Morgera91].

4 Derivation of Models

In this section, several methods of deriving models from data are presented. Note that in addition to those presented in this section, Tsai's model was presented in Section 2.2.3.

4.1 The Gilbert Model

The Gilbert Model [Gilbert60] is a two state model, where the states represent 'good' and 'bad' line conditions. (See Figure 1). In the good state G, the model produces no errors. In the bad state B, the model produces an error with probability $1-h$.

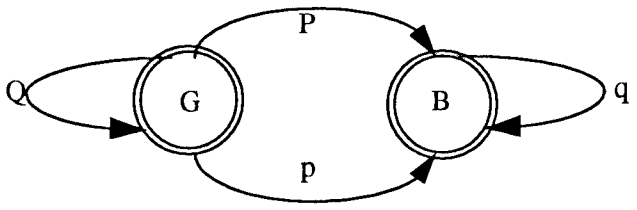


Figure 1. Gilbert's model state diagram

Now, it should be mentioned that this model cannot be directly encoded in an EIU model, because, due to the hardware design, the EIUs states must either always produce an error, or never produce an error; they cannot, like state B, produce an error with some fixed probability. However, it is relatively straightforward to split B into two states, one producing an error, and the other error free. This is shown in [Turin93]; see Figure 2.

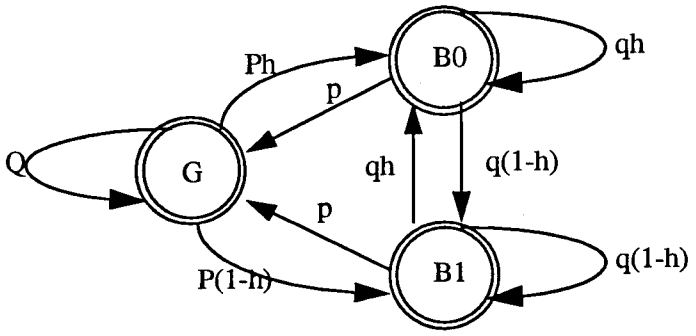


Figure 2. Three state EIU equivalent to Gilbert's Model. B1 is the only error producing state

Gilbert then observes that there are three independent parameters, since $P = 1 - Q$, and $p = 1 - q$. Therefore, three measured quantities are needed to estimate them. He chooses the observations

$$a = P(1) \tag{EQ 6}$$

$$b = P(1|1) \tag{EQ 7}$$

$$c = \frac{P(111)}{P(101) + P(111)} \tag{EQ 8}$$

Clearly, a can be measured simply by dividing the number of errors by the number of bits, b can be measured by dividing the number of occurrences of the bit

string '11' by the number of errors, and c can be measured by counting the number of occurrences of the bit strings '101' and '111', and performing the appropriate operations. Note that c is the probability of a 1 occurring in the center position when two ones occur one space apart.

It is easy to find equations for a, b, and c in terms of P, p, and h. Then, solving these equations for the variables on the right, we get

$$1-p = q = \frac{ac-b^2}{2ac-b(a+c)} \quad \text{(EQ 9)}$$

$$h = 1-\frac{b}{q} \quad \text{(EQ 10)}$$

$$P = \frac{ap}{1-h-a} \quad \text{(EQ 11)}$$

It is interesting to note that, as Gilbert observed, these equations tend to produce ridiculous values for the parameters if not enough data is presented. Our experience indicates that they may also produce such out of range values if the observed error process does not fit the Gilbert model.

4.2 The Autocorrelation Matching Model

The weakness of Tsai's method is that it assumes that adjacent error gaps are distributed independently of one another. It is easy to see this from the way the model is constructed -- after each error burst, the model selects a non-error state independently of its last non-error state, and then tarries in that state until the next error burst. Also, notice that it is built entirely from the probability mass function of the error gaps. No information about the placement or ordering of the gap sizes is included in the models derivation.

In order to try to include this sort of information, we can try to fit the autocorrelation function of the data, instead of just the gap distribution. The autocorrelation function gives the joint probability of two bits being in error, regardless of whether the intervening bits are in error or not, while the error gap characterization gives the joint probability of two bits being in error, assuming that the intervening bits are error-free.

In this section, we will derive expressions for the autocorrelation function, and describe briefly the difficulties with building a probability transition function with the desired autocorrelation function.

Let P be the probability transition matrix for a Markov chain, such that if π_t is a vector of probabilities of being in each state at time t, then $\pi_{t+1} = P\pi_t$ is a vector of the probabilities of being in each state at time $t+1$. There exists a vector π_0 such that $\pi_0 = P\pi_0$; this is the long-term steady state probability of being in each state. Now, if we let E be a diagonal matrix with 1 in the diagonal entries corresponding to error states and 0 in the other diagonal entries, then $E\pi_0$ is the long-term steady state probability vector, with all non-error entries zeroed. Clearly, the ith entry of $P^n E\pi_0$ is the probability of seeing an error string $1x^n$, ending in state i. Then $EP^n E\pi_0$ is the same vector with all non-error ending states zeroed. Finally, if we let $\bar{1}$ be a column vector of 1's, then $\bar{1}^T EP^n E\pi_0$ is the probability of seeing the string $1x^{n-1}1$, or R_n; thus

$$R_n = \bar{1}^T EP^n E\pi_0 \qquad \text{(EQ 12)}$$

However, this form of the autocorrelation function is not entirely satisfying. Therefore, we continue to simplify. Recall that any matrix A can be rewritten $A = S\Lambda S^{-1}$, where Λ is a diagonal matrix whose entries are the eigenvalues of A, and S is a matrix containing the eigenvectors of A, one in each column; that is, if $\lambda_1, \lambda_2, \ldots, \lambda_m$ are the eigenvalues of A and $v_1, v_2, \ldots, v_m$ are the corresponding eigenvectors, then $\Lambda = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \lambda_m \end{bmatrix}$ and $S = \begin{bmatrix} v_1 & v_2 & \ldots & v_m \end{bmatrix}$. Using this decomposition, let $P = S\Lambda S^{-1}$; then

$$R_n = \bar{1}^T E (S\Lambda S^{-1})^n E\underline{\pi}_0 = \bar{1}^T E S\Lambda^n S^{-1} E\pi_0 = \bar{1}^T E S \begin{bmatrix} \lambda_1^n & 0 & \ldots & 0 \\ 0 & \lambda_2^n & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \lambda_m^n \end{bmatrix} S^{-1} E\pi_0 \quad \text{(EQ 13)}$$

It is clear that the form of this equation will be

$$R_n = \sum_{i=1}^{m} k_i \lambda_i^n = \sum k_i e^{n \log \lambda_i} \quad \text{(EQ 14)}$$

This form is familiar; it is the same as that we dealt with in fitting the gap distributions. Thus, we have only to measure the autocorrelation function, fit it with m exponentials, and then determine matrices E, S, and Λ to give us the desired function R_n. Note that Λ is determined immediately from the coefficients in the exponents. This leaves only the problem of determining E, which indicates which states are error states, and S, the eigenvector matrix of P.

In fact, it would appear that the solution is not unique; further, no closed form for a solution has been located. Exploring the method further, however, let us suppose that we set the number of error states, N_e, a priori. Then E can be set so that the first elements of the diagonal are one, and later entries are zero, depending on the number of error states. (It does not matter which state is which, so there is no loss of generality in designating the first N_e states to be error states.) Using this further information, we can develop some information about the necessary form of the eigenvector matrix.

Note that any probability transition matrix has columns whose entries sum to 1. As a result, when the transition matrix multiplies any vector (whose entries sum to some value x), the resulting vector also has entries summing to x. Furthermore, the long-term state occupancy probability vector (π_0) is clearly a fixed point with respect to multiplication by the transition matrix. This is the same as saying that it is an eigenvector corresponding to the eigenvalue 1. It is relatively easy to see that all the other eigenvectors must have elements which sum to 0; if they added to one, there would be two long-term state occupancy probability vectors (which, of course, can happen, but only if the Markov chain is disconnected, which we do not wish to consider). If the sum of elements was neither zero nor one, then there would be some vectors whose sum was not conserved during a multiplication by the transition matrix.

Now, let us first consider the term $S^{-1}E\pi_0$. Note that $SS^{-1}E\pi_0 = E\pi_0$, so we can interpret the vector $S^{-1}E\pi_0$ as a set of coefficients such that $\sum_{i=1}^{m} d_i v_i = E\pi_0$. Here, we take advantage of the fact that the eigenvectors v_i can be scaled however we

like. We scale them in such a way that $d_i = 1$, for $i > 1$. Extending this to the vector $\Lambda^n S^{-1} E\pi_0$, we see that $\Lambda^n S^{-1} E\pi_0 = \begin{bmatrix} d_1\lambda_1^n \\ d_2\lambda_2^n \\ \cdots \\ d_m\lambda_m^n \end{bmatrix} = \begin{bmatrix} d_1 \\ \lambda_1^n \\ \cdots \\ \lambda_m^n \end{bmatrix}$.

We can now turn our attention to the first half of the equation for R_n, the term $\bar{1}^T ES$. Since $S = \begin{bmatrix} v_1 & v_2 & \cdots & v_m \end{bmatrix} = \begin{bmatrix} v_{1,1} & v_{2,1} & \cdots & v_{m,1} \\ v_{1,2} & v_{2,2} & \cdots & v_{m,2} \\ \cdots & \cdots & \cdots & \cdots \\ v_{1,m} & v_{2,m} & \cdots & v_{m,m} \end{bmatrix}$, it is clear that $\bar{1}^T ES = \begin{bmatrix} \sum_{j=1}^{N_e} v_{1,i} & \sum_{i=1}^{N_e} v_{2,i} & \cdots & \sum_{i=1}^{N_e} v_{m,i} \end{bmatrix}$. Further, since $v_1 = \pi_0$, $\sum_{i=1}^{N_e} v_{1,i} = BER$. Therefore, combining this description with the one above, we see that

$$R_n = d_1 BER + \sum_{i=2}^{m} \lambda_i^n \sum_{j=1}^{N_e} v_{i,j} \qquad \text{(EQ 15)}$$

From the initial description of the autocorrelation function, then, we know that $d_1 = BER$. To summarize our knowledge about S, then, we observe that $v_1 = \pi_0$, $(BER)\, v_1 + \sum_{i=2}^{m} v_i = Ev_1$, $\sum_{i=1}^{N_e} v_{1,i} = BER$, and $\sum_{j=1}^{N_e} v_{i,j} = k_i$ (for $i \geq 2$), where the values of k_i come from the curvefit. Clearly, this leaves open many degrees of freedom on the matrix S. These degrees are needed to insure that the entries in P are positive. The problem of finding a matrix S which meets this requirement has not been solved; we have attempted to use an "annealing" algorithm of sorts to

stochastically search for such a matrix. However, it has proved resistant to these techniques and other methods are currently under investigation. The details are not presented in this paper.

4.3 The Maximum Likelihood Model (Baum-Welch Algorithm)

Clearly, the improvement gained in moving to the autocorrelation method is somewhat counteracted by the fact that its solution is not well understood. Therefore, a more refined method is desirable. The maximum likelihood model is such a method. The following presentation is a mixture of those found in [Turin93], [Morgera91], and [Poritz88].

First, let us suppose that we have a binary observation vector (that is, for, example, the error bit vector from a measurement of the channel), call it $E = e_1 e_2 \ldots e_T$. Then we wish to find a Markov model λ which maximizes the expression $Pr[E|\lambda]$. In other words, what we seek is the Markov model which has the highest probability of producing the observed error sequence. Formally, we consider $\lambda = (A, B, \pi)$, where $A = [a_{ij}]$, $B = [b_{ij}]$ are matrices which describe the Markov model and π is probability vector describing the likelihood of having started in a particular state; a_{ij} is the probability of a transition from state i to state j, and b_{ij} is the probability of outputting symbol j, given that the system is in state i. We have already seen this type of state model in the Gilbert model; however, for the EIU, we will require that all entries b_{ij} are either 0 or 1.

The general approach to solving this method in the literature is simple; guess at λ, compute $Pr[E|\lambda]$, and then use a gradient ascent method to maximize that quan-

tity. While this is quite straightforward, there are hidden difficulties. For example, the direct method for computation of $Pr[E|\lambda]$ is exponentially complex in the length of the error observation. Fortunately, there is a simpler way of computing this quantity, namely the forward-backward recursion. Before presenting this, however, we will need a little notation.

Let $B(v_k) = \begin{bmatrix} b_{1k} & 0 & \dots & 0 \\ 0 & b_{2k} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & b_{Nk} \end{bmatrix}$. Let $P(v_k) = AB(v_k)$. Then we can define the forward variable $\alpha_t(i) = Pr[e_1, e_2, \dots, e_t, i_t \equiv q_i|\lambda]$. We usually speak of the vector $\alpha_t = \left[\alpha_t(1) \dots \alpha_t(N)\right]$. Similarly, we can define the backward variable $\beta_t(i) = Pr[e_{t+1}, e_{t+2}, \dots, e_T|i_t \equiv q_i, \lambda]$, with β_t defined as a column vector in the obvious manner. Then it is fairly easy to show that

$$\alpha_0 = \pi \qquad \text{(EQ 16)}$$

$$\alpha_{t+1} = \alpha_t P(e_{t+1}) \qquad \text{(EQ 17)}$$

$$\beta_T = \bar{1} \qquad \text{(EQ 18)}$$

$$\beta_t = P(e_{t+1})\beta_{t+1} \qquad \text{(EQ 19)}$$

Furthermore, it is clear that, for any t,

$$Pr[E|\lambda] = \alpha_t\beta_t \qquad \text{(EQ 20)}$$

Now, we can present one formulation of the Baum-Welch reestimation equations. Define $U_{ij} = \frac{\partial A}{\partial a_{ij}}$ (that is, the matrix that is 0 everywhere except in position ij, where it is 1). Then given $\lambda = (A, B, \pi)$, we reestimate using:

$$\overline{a_{ij}} = \frac{a_{ij}\sum_{t=0}^{T-1}\alpha_t U_{ij} b_j(e_{t+1})\beta_{t+1}}{\sum_{t=0}^{T-1}\alpha_t U_{ii}\beta_t} \qquad \text{(EQ 21)}$$

$$\overline{b_{ij}} = \frac{\sum_{t \ni e_t = v_j}^{T}\alpha_t U_{ij}\beta_t}{\sum_{t=1}^{T}\alpha_t U_{ii}\beta_t} \qquad \text{(EQ 22)}$$

$$\overline{\pi_i} = \frac{\alpha_0(i)\beta_0(i)}{Pr[E|\lambda]} \qquad \text{(EQ 23)}$$

Notice the computational complexity -- linear in the length of the observation sequence. Since we expect to be working with very long observation sequences, it is worth our while to investigate the suggestions made in [Turin93] to speed up the computation by taking advantage of long sequences of repeated symbols. Let us suppose that we can break up our error observation string into segments, each of which consist of a single symbol, repeated one or more times; $E = e_1 e_2 \ldots e_T = X_1^{l_1} X_2^{l_2} \ldots X_\tau^{l_\tau}$. For convenience, we also define $t_i = \sum_{j=1}^{i} l_j$ to be the boundary times between the runs of symbols.

Now, using this extra information, we can rewrite the forward-backward recursions in a more efficient way, remembering that matrix powers can be computed efficiently by any number of methods:

$$\alpha_{t_0} = \pi \qquad \text{(EQ 24)}$$

$$\alpha_{t_{i+1}} = \alpha_{t_i} P^{l_{i+1}}(e_{t_{i+1}}) \qquad \text{(EQ 25)}$$

$$\beta_{t_\tau} = \bar{1} \tag{EQ 26}$$

$$\beta_{t_i} = P^{l_{i+1}}(e_{r_{i+1}})\,\beta_{t_{i+1}} \tag{EQ 27}$$

Notice that the same K matrices (K is the number of symbols in the alphabet; in the case of error sequences, it is always 2) are being raised to various powers. For this reason, it is probably preferable to compute the matrix powers using an eigen decomposition, because the expense of computing the eigenvalues and eigenvectors can be easily payed for by the vastly simplified computations that result.

However, notice that, in the reestimation formula, we take the sums over t of α and β. Unless we can somehow find a faster way to compute these sums, we cannot take advantage of the above computations, which skip intermediate values. Fortunately, [Turin93] continues on to solve this problem as well. Consider, for example, the numerator of the reestimation equation for a_{ij}:

$$\sigma_{ij} = \sum_{t=0}^{T-1} \alpha_t U_{ij} b_j(e_t)\,\beta_{t+1} = \sum_{k=1}^{\tau} \sigma_{ijk} \tag{EQ 28}$$

$$\sigma_{ijk} = \sum_{t=t_k}^{t_{k+1}-1} \alpha_{t_k} C_{ijX_k}(l_k)\,\beta_{t_{k+1}} \tag{EQ 29}$$

$$C_{ijX_k}(l_k) = \sum_{\gamma=0}^{l_k-1} P^{\gamma}(X_k)\,U_{ij} b_j(X_k)\,P^{l_k-\gamma-1}(X_k) \tag{EQ 30}$$

We notice that $C_{ijX_k}(l_k)$ is in the form of a matrix convolution, and thus take the z-transform:

$$\Phi_{ijX_k}(z) = Z(C_{ijX_k}(l)) = \sum_{l=1}^{\infty} z^{-l} C_{ijX_k}(l) \tag{EQ 31}$$

$$\Phi_{ijX_k}(z) = \sum_{l=1}^{\infty} z^{-l} \sum_{\gamma=0}^{l-1} P^{\gamma}(X_k)\, U_{ij} b_j(X_k)\, P^{l-\gamma-1}(X_k) \qquad \text{(EQ 32)}$$

$$\Phi_{ijX_k}(z) = \sum_{\gamma=0}^{\infty} \sum_{l=\gamma+1}^{\infty} [z^{-1}P(X_k)]^{\gamma} U_{ij} b_j(X_k)\, z^{-1} [z^{-1}P(X_k)]^{l-\gamma-1} \qquad \text{(EQ 33)}$$

Now, if we let $\delta = l-\gamma-1$, then we can say:

$$\Phi_{ijX_k}(z) = \left(\sum_{\gamma=0}^{\infty} [z^{-1}P(X_k)]^{\gamma} \right) U_{ij} b_j(X_k)\, z^{-1} \left(\sum_{\delta=0}^{\infty} [z^{-1}P(X_k)]^{\delta} \right) \qquad \text{(EQ 34)}$$

$$\Phi_{ijX_k}(z) = [zI - P(X_k)]^{-1} U_{ij} b_j(X_k)\, z\, [zI - P(X_k)]^{-1} \qquad \text{(EQ 35)}$$

Here, we must pause to describe a decomposition that allows us to better describe this z-transform -- in particular, a way of rewriting $[zI-P(X_k)]^{-1}$. Let us first focus on the matrix $P(X_k)$. If we let $\lambda_{k1}, \lambda_{k2}, \ldots, \lambda_{kN}$ be the eigenvalues of $P(X_k)$, with $\Lambda_k = \begin{bmatrix} \lambda_{k1} & 0 & \ldots & 0 \\ 0 & \lambda_{k2} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \lambda_{kN} \end{bmatrix}$, and $S_k = \begin{bmatrix} v_{k1} & v_{k2} & \ldots & v_{kN} \end{bmatrix}$ be the matrix whose columns are the eigenvectors of $P(X_k)$, then it is well known that

$$P(X_k) = S_k \Lambda_k S_k^{-1} \qquad \text{(EQ 36)}$$

If we then rewrite $\Lambda_k = \sum_{i=1}^{N} \lambda_{ki} U_{ii}$, we see that

$$P(X_k) = \sum_{i=1}^{N} \lambda_{ki} D_{ki}, \text{ where} \qquad \text{(EQ 37)}$$

$$D_{ki} = S_k U_{ii} S_k^{-1} \qquad \text{(EQ 38)}$$

Also, since $(P(X_k))^{\alpha} = S_k \Lambda_k^{\alpha} S_k^{-1}$, it is easy to see that $(P(X_k))^{\alpha} = \sum_{i=1}^{N} \lambda_{ki}^{\alpha} D_{ki}$.

However, we now turn our attention to the matrix $zI - P(X_k)$. Notice that $zI = zSIS^{-1} = \sum_{i=1}^{N} zSU_{ii}S^{-1}$, while $P(X_k) = \sum_{i=1}^{N} \lambda_{ki} D_{ki} = \sum_{i=1}^{N} \lambda_{ki} SU_{ii} S^{-1}$; thus

$$zI - P(X_k) = \sum_{i=1}^{N} (z - \lambda_{ki}) SU_{ii}S^{-1} = \sum_{i=1}^{N} (z - \lambda_{ki}) D_{ki} \qquad \text{(EQ 39)}$$

In other words, we can treat $zI - P(X_k)$ just as $P(X_k)$, except with eigenvalues $z - \lambda_{k1}, z - \lambda_{k2}, \ldots, z - \lambda_{kN}$. Hence, it is clear that

$$[zI - P(X_k)]^{-1} = \sum_{i=1}^{N} (z - \lambda_{ki})^{-1} D_{ki} = \sum_{i=1}^{N} \frac{D_{ki}}{z - \lambda_{ki}} \qquad \text{(EQ 40)}$$

Now, returning to our characterization of $\Phi_{ijX_k}(z)$, and substituting in the above expression, we see that

$$\Phi_{ijX_k}(z) = \sum_{\varepsilon=1}^{N} \sum_{\zeta=1}^{N} D_{k\varepsilon} U_{ij} b_j(X_k) D_{k\zeta} \left(\frac{z}{(z - \lambda_{k\varepsilon})(z - \lambda_{k\zeta})} \right) \qquad \text{(EQ 41)}$$

Now, since the z-transform is a linear operator, we need only find the inverse z-transforms of each element of the sum in order to recover $C_{ijX_k}(l_k)$ from $\Phi_{ijX_k}(z)$.

$$\frac{z}{(z - \lambda)(z - \mu)} \Leftrightarrow \frac{\lambda^l - \mu^l}{\lambda - \mu}, \text{ if } \lambda \neq \mu \qquad \text{(EQ 42)}$$

$$\frac{z}{(z - \lambda)^2} \Leftrightarrow l\lambda^{l-1} \qquad \text{(EQ 43)}$$

Therefore, it is easy to see that

$$C_{ijX_k}(l_k) = \sum_{\varepsilon=1}^{N}\sum_{\zeta=1}^{N} D_{k\varepsilon}U_{ij}b_j(X_k)D_{k\zeta}Z^{-1}\left(\frac{z}{(z-\lambda_{k\varepsilon})(z-\lambda_{k\zeta})}\right)\Bigg|_{l_k}, \text{ where} \quad \text{(EQ 44)}$$

$$Z^{-1}\left(\frac{z}{(z-\lambda_{k\varepsilon})(z-\lambda_{k\zeta})}\right)\Bigg|_{l_k} = \frac{\lambda_{k\varepsilon}^{l_k}-\lambda_{k\zeta}^{l_k}}{\lambda_{k\varepsilon}-\lambda_{k\zeta}}, \text{ if } \lambda_{k\varepsilon}\neq\lambda_{k\zeta}, \text{ or} \quad \text{(EQ 45)}$$

$$Z^{-1}\left(\frac{z}{(z-\lambda_{k\varepsilon})(z-\lambda_{k\zeta})}\right)\Bigg|_{l_k} = l_k\lambda_{k\varepsilon}^{l_k-1}, \text{ if } \lambda_{k\varepsilon}=\lambda_{k\zeta} \quad \text{(EQ 46)}$$

This allows us to calculate the desired quantities in a much more efficient manner.

Finally, one remaining implementation issue needs to be dealt with. Because the computations involve the exponentiation of stable matrices, we must take care to avoid numerical underflow. During the computations of α and β, we should store vectors which are normalized so that the sum of their elements is one. The scaling factors used should also be stored, as they will be needed to successfully evaluate the other equations. A description of this process can be found in [Morgera91] or [Turin93].

5 A Test Scenario

In order to explain how these methods fit together, this section will describe an idealized testing scenario, from beginning to end.

Company C wishes to characterize the performance of Application A over a Satellite Channel SC. Therefore, Company C arranges time on the channel SC, and sets up and runs application A over the channel when they get it. In addition, they take

measurements of the channel at a bit level, either at the same time as the test is running, or separately, perhaps at night.

Now, as was expected, the application A ran fairly poorly. Therefore, the application folks begin to work on its ability to work under such adverse circumstances. Meanwhile, channel characterization can take place. First, each of the available models are used on the available data to generate channel models. The faster models are built first; much of the next step can be done while the more complicated ones are being computed. Now, once the models are available, the distribution of error bursts and error free bursts and their autocorrelation functions can be compared with the measured distributions and autocorrelations, taken from the data itself. Also, the probability that each model generated the given error strings can be computed. (Note that these probabilities decrease exponentially fast with the length of the error string. It is not the absolute size of these probabilities, but the relative size which is of interest.) Now, using these techniques, it should be possible to identify a few models which seem most promising. These models (and perhaps some others, as well) should reach the next stage of testing.

The selected models should then be loaded into an EIU (or other similar device). The test of application A which was conducted earlier should be repeated over each of the modelled links. Those whose performance most closely match the performance observed during the actual test are to be regarded as best for this application.

Now, by this time, the applications folks should have some improvements in place, creating a new version of application A, call it A'. But by now they're wondering if they improved it enough to cover for the channel errors, or if they have sandbagged the performance by providing too much redundancy. The channel

model(s) which have been selected can be used to evaluate these question. When the revision performs well on the modelled channel, company C can again arrange time on channel SC, and can test and tune the parameters of the new model. In addition, of course, it may very well be that a new application, B, needs to be tested. And so here the cycle swings full circle and the process begins again.

6 Conclusions

This work has presented newer and more sophisticated methods for the development of error channel models than those which were previously in use at Rome Laboratory. These methods have been implemented on available computer systems, and have been placed firmly within a development cycle which includes data collection, model building, and evaluation of the fitness of the various models. While further work is still required on the autocorrelation matching method, the other methods provide an array of models varying in complexity and fidelity. Further study is required to predict the quality of each method, and to understand the trade-off between model quality and computation time.

7 Reference

[1] Fritchman, Bruce D. "A Binary Channel Characterization Using Partitioned markov Chains." IEEE Transactions on Information Theory, Vol IT-13, No 2, April, 1967.

[2] Gilbert, E. N. "Capacity of a Burst-Noise Channel." The Bell System Technical Journal, September 1960.

[3] Jury, E. I. "Theory and Application of the z-Transform Method." John Wiley and Sons, Inc., New York 1964.

[4] Maple, S. Lawrence, Jr. "Digital Spectral Analysis with Applications." Prentice Hall Inc., Englewood Cliffs, NJ 1987.

[5] McManamon, Peter. "HF Markov Chain Models and Measured Error Averages." IEEE Transactions on Communication Technology, Vol COM-18, No 3, June, 1970.

[6] Morgera, Salvatore D., and Frederic Simard. "Parameter Estimation for a Burst-Noise Channel," ICASSP '91, Vol 3, pp. 1701-4.

[7] Poritz, Alan B. "Hidden Markov Models: A Guided Tour," ICASSP '88, Vol 1, pp. 7-13.

[8] Smith, Wayne D. "Markov Models for simulating Error Patterns on Data Communications Links," Final Report of 1991 USAF-UES Research Initiation Program (UES PO #5-210-11ME-012) June 15, 1991.

[9] Smith, Wayne D., Karen D. Burns, and Jane N. Moorhead. "Using Markov Chains to Model the Error Behavior of Data Communications Channels." Final Project Report Draft dated Feb 9, 1993. (Contract Number F30602-92-C-0023)

[10] Swarts, F., H. C. Ferreira, and D. R. Oosthuizen. "Renewal Channel Models for PSK on Slowly Fading Rayleigh Channel." Electronics Letters, Vol 25, No 22, Oct 26, 1989.

[11] Tsai, Stephen. "Markov Characterization of the HF Channel." IEEE Transactions on Communication Technology, Vol COM-17, No 1, February, 1969.

[12] Turin, William, and M. Mohan Sondhi. "Modelling Error Sources in Digital Channels," IEEE Journal on Selected Areas in Communications, Vol 11, No 3, April 1993. pp. 340-7.

8 Acknowledgments

This work could not have been done without the support of the staff of Rome Lab's Command, Control, and Communications Directorate, Communications Division, especially the Networks Branch. My mentor, Dr. John Evanowsky, was instrumental in setting the direction of this work, and my supervisor, Joe Zdanowicz, was always supportive. Thanks to Priscilla Cassidy, Sean Lanigan, Nick Kowalchuk, and Anne White for their infinite patience and invaluable help. I owe my understanding of stochastic processes to Professor Jose Moura's class at Carnegie Mellon. Finally, thanks to my advisor, Hyong S. Kim, the Air Force Office of Scientific Research, and the Southeastern Center for Electrical Engineering Education (SCEEE) for the impetus to complete this research period, and to Pete Robinson for taking the initiative to see its publication.

Rome Laboratory

Customer Satisfaction Survey

RL-TR-_____

Please complete this survey, and mail to RL/IMPS, 26 Electronic Pky, Griffiss AFB NY 13441-4514. Your assessment and feedback regarding this technical report will allow Rome Laboratory to have a vehicle to continuously improve our methods of research, publication, and customer satisfaction. Your assistance is greatly appreciated.
Thank You

_____

_____

Organization Name:_____(Optional)

Organization POC: _____(Optional)

Address:_____

1. On a scale of 1 to 5 how would you rate the technology developed under this research?

5-Extremely Useful 1-Not Useful/Wasteful

Rating_____

Please use the space below to comment on your rating. Please suggest improvements. Use the back of this sheet if necessary.

2. Do any specific areas of the report stand out as exceptional?

Yes___ No_____

If yes, please identify the area(s), and comment on what aspects make them "stand out."

3. Do any specific areas of the report stand out as inferior?

Yes___ No___

If yes, please identify the area(s), and comment on what aspects make them "stand out."

4. Please utilize the space below to comment on any other aspects of the report. Comments on both technical content and reporting format are desired.

☆U.S. GOVERNMENT PRINTING OFFICE: 1995-610-126-20003

MISSION

OF

ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.