INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or “target” for pages apparently lacking from the document photographed is “Missing Page(s)”. If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of “sectioning” the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.
RAMASWAMY, PREM

EVALUATION OF A BINARY COMMUNICATION CHANNEL SIMULATOR

THE UNIVERSITY OF ARIZONA M.S. 1983

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106

Copyright 1984 by

RAMASWAMY, PREM
All Rights Reserved
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark \( \checkmark \).

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background \( \checkmark \)
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages \( \checkmark \)
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) __________ lacking when material received, and not available from school or author.
12. Page(s) __________ seem to be missing in numbering only as text follows.
13. Two pages numbered __________. Text follows.
14. Curling and wrinkled pages _____
15. Other ___________________________________________________________
EVALUATION OF A BINARY COMMUNICATION
CHANNEL SIMULATOR

by

Prem Ramaswamy

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
In the Graduate College
THE UNIVERSITY OF ARIZONA

1983
STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Prem Ramaswamy

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

L. C. Schooley
Associate Professor
of Electrical Engineering

August 17, 1985
Date
ACKNOWLEDGEMENT

I wish to express my deep appreciation to Professor L.C. Schooley for suggesting the problem and for his continuous guidance. I am also grateful to him for his assistance in the preparation of the final manuscript. I am thankful to Mr. Elmer Grubbs for making the experimental results available to me. Appreciation is also expressed to Professor R. Martinez, Professor R. Strickland and Mr. G.R. Davis for reading the first draft and making many useful comments.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xi</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. TEST STATISTICS AND CHANNEL MODELS</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Some Test Statistics</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Bit Error Rate</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2 Throughput</td>
<td>6</td>
</tr>
<tr>
<td>2.1.3 Burst and Gap Statistics</td>
<td>7</td>
</tr>
<tr>
<td>2.1.4 Other Test Statistics</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Channel Models</td>
<td>11</td>
</tr>
<tr>
<td>3. THE PSEUDO RANDOM ERROR SEQUENCE GENERATOR</td>
<td>15</td>
</tr>
<tr>
<td>3.1 Theory of Operation</td>
<td>15</td>
</tr>
<tr>
<td>3.1.1 Shift Register Sequences</td>
<td>16</td>
</tr>
<tr>
<td>3.1.2 Error Generator Using a Single Shift Register</td>
<td>18</td>
</tr>
<tr>
<td>3.1.3 Error Generations Using Several Shift Registers</td>
<td>18</td>
</tr>
<tr>
<td>3.2 Construction Techniques and Operation Mechanism</td>
<td>19</td>
</tr>
<tr>
<td>3.2.1 The Shift Registers</td>
<td>19</td>
</tr>
<tr>
<td>3.2.2 State Select Logic</td>
<td>20</td>
</tr>
<tr>
<td>3.2.3 The Output Logic</td>
<td>22</td>
</tr>
<tr>
<td>4. THE DIGITAL CHANNEL PERFORMANCE EVALUATOR</td>
<td>31</td>
</tr>
<tr>
<td>4.1 General</td>
<td>31</td>
</tr>
<tr>
<td>4.2 Long Term Error Rates</td>
<td>32</td>
</tr>
<tr>
<td>4.3 Short Term Error Rates</td>
<td>32</td>
</tr>
<tr>
<td>4.4 Gap and Burst Statistics</td>
<td>33</td>
</tr>
<tr>
<td>4.5 Interruption Statistics</td>
<td>33</td>
</tr>
<tr>
<td>4.6 DCOPE Architecture</td>
<td>34</td>
</tr>
<tr>
<td>4.6.1 Error Counting and Interface Subsystem</td>
<td>34</td>
</tr>
<tr>
<td>4.6.2 Control and Arithmetic Subsystem</td>
<td>35</td>
</tr>
<tr>
<td>4.6.3 Housekeeping Subsystem</td>
<td>36</td>
</tr>
<tr>
<td>4.6.4 The GPIB Subsystem</td>
<td>36</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS—Continued

## 5. DERIVATION AND CALCULATION OF ERROR STATISTICS  43

5.1 Derivation of Statistics  43
   5.1.1 Gap Distribution  43
   5.1.2 Burst Length Distribution  45
   5.1.3 Block and Character Error Rate  50
   5.1.4 Throughput  51
   5.1.5 Transition Matrix  51
   5.1.6 Mean Time in State  53
   5.1.7 Statistics for the Two and Three State Case  54

5.2 Calculations  58
   5.2.1 Gap Distribution  59
   5.2.2 Burst Length Distribution  59
   5.2.3 Block and Character Error Rate  60
   5.2.4 Throughput  60
   5.2.5 Mean Time in State  62

## 6. EXPERIMENTAL RESULTS AND COMPARISON WITH THEORETICAL CALCULATIONS  81

6.1 One State Case  81
6.2 Two and Three State Case  83

## 7. SUMMARY AND CONCLUSIONS  97

APPENDIX A: PROGRAM FOR COMPUTING GAP LENGTH DISTRIBUTION  101
APPENDIX B: PROGRAM FOR COMPUTING BURST LENGTH DISTRIBUTION  102
REFERENCES  104
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Block diagram of a typical digital communication system</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>$P(m, 31)$ distribution for telephone channel (T) and binary symmetric channel model with the same average bit error rate ($5 \times 10^{-5}$)</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Transition diagram for the Markov chain</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>A three state Gilbert model for a typical telephone channel</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>a) Random binary sequence b) its autocorrelation function c) its power spectrum</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Generalized shift register sequence generator</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>A maximum-length shift register sequence generator of degree 4</td>
<td>26</td>
</tr>
<tr>
<td>3.4</td>
<td>a) Normalized autocorrelation function of a PN binary waveform of bit duration T. b) Power spectrum of a PN binary waveform of period $P$ having the autocorrelation shown in a</td>
<td>26</td>
</tr>
<tr>
<td>3.5</td>
<td>Error Generation using a single shift register</td>
<td>27</td>
</tr>
<tr>
<td>3.6</td>
<td>Error generation using several shift registers</td>
<td>27</td>
</tr>
<tr>
<td>3.7</td>
<td>Block diagram of the PRESG</td>
<td>28</td>
</tr>
<tr>
<td>3.8</td>
<td>Decision logic for implementing state transitions</td>
<td>28</td>
</tr>
<tr>
<td>3.9</td>
<td>Comparators for comparing the ten bit vector with the two programmable levels</td>
<td>29</td>
</tr>
<tr>
<td>3.10</td>
<td>a) Typical transition matrix, b) Levels that must be selected in order to implement this matrix</td>
<td>29</td>
</tr>
<tr>
<td>3.11</td>
<td>Actions of the PRESG during each clock cycle</td>
<td>30</td>
</tr>
<tr>
<td>4.1</td>
<td>Block diagram of the DCE 200</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td>Error counting and interface subsystem</td>
<td>39</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.3</td>
<td>Control and arithmetic subsystem</td>
<td>40</td>
</tr>
<tr>
<td>4.4</td>
<td>Housekeeping subsystem</td>
<td>41</td>
</tr>
<tr>
<td>4.5</td>
<td>GPIB (IEEE-488) subsystem</td>
<td>42</td>
</tr>
<tr>
<td>5.1</td>
<td>State transition diagram of two state Markov chain process</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>Cumulative distribution of gap length. Gap constant = 1.0, Block length = 4 bits</td>
<td>63</td>
</tr>
<tr>
<td>5.3</td>
<td>Cumulative distribution of gap length. Gap constant = 1.0, Block length = 8 bits</td>
<td>64</td>
</tr>
<tr>
<td>5.4</td>
<td>Cumulative distribution of gap length. Gap constant = 1.0, Block length = 4 bits</td>
<td>65</td>
</tr>
<tr>
<td>5.5</td>
<td>Cumulative distribution of gap length. Gap constant = 2.0, Block length = 8 bits</td>
<td>66</td>
</tr>
<tr>
<td>5.6</td>
<td>Cumulative distribution of gap length. Gap constant = 4.0, Block length = 4 bits</td>
<td>67</td>
</tr>
<tr>
<td>5.7</td>
<td>Cumulative distribution of gap length. Gap constant = 4.0, Block length = 8 bits</td>
<td>68</td>
</tr>
<tr>
<td>5.8</td>
<td>Cumulative distribution of gap length. Gap constant = 8.0, Block length = 4 bits</td>
<td>69</td>
</tr>
<tr>
<td>5.9</td>
<td>Cumulative distribution of gap length. Gap constant = 8.0, Block length = 8 bits</td>
<td>70</td>
</tr>
<tr>
<td>5.10</td>
<td>Cumulative distribution of burst length. Gap constant = 1.0, Block length = 4 bits</td>
<td>71</td>
</tr>
<tr>
<td>5.11</td>
<td>Cumulative distribution of burst length. Gap constant = 1.0, Block length = 8 bits</td>
<td>72</td>
</tr>
<tr>
<td>5.12</td>
<td>Cumulative distribution of burst length. Gap constant = 2.0, Block length = 4 bits</td>
<td>73</td>
</tr>
<tr>
<td>5.13</td>
<td>Cumulative distribution of burst length. Gap constant = 2.0, Block length = 8 bits</td>
<td>74</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.14.</td>
<td>Cumulative distribution of burst length. Gap constant = 4.0, Block length = 4 bits</td>
<td>75</td>
</tr>
<tr>
<td>5.15.</td>
<td>Cumulative distribution of burst length. Gap constant = 4.0, Block length = 8 bits</td>
<td>76</td>
</tr>
<tr>
<td>5.16.</td>
<td>Cumulative distribution of burst length. Gap constant = 8.0, Block length = 4 bits</td>
<td>77</td>
</tr>
<tr>
<td>5.17.</td>
<td>Cumulative distribution of burst length. Gap constant = 8.0, Block length = 8 bits</td>
<td>79</td>
</tr>
<tr>
<td>5.18.</td>
<td>Cumulative distribution of burst length. Gap constant = 16.0, Block length = 4 bits</td>
<td>79</td>
</tr>
<tr>
<td>5.19.</td>
<td>Cumulative distribution of burst length. Gap constant = 16.0, Block length = 8 bits</td>
<td>80</td>
</tr>
<tr>
<td>6.1.</td>
<td>a) Transition matrices used for obtaining statistics for the two state case</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>b) three state case</td>
<td></td>
</tr>
<tr>
<td>6.2.</td>
<td>Cumulative distribution of burst lengths for the two state case (AB). Gap constant = 1.0, Block length = 8 bits</td>
<td>89</td>
</tr>
<tr>
<td>6.3.</td>
<td>Cumulative distribution of burst length for the two state case (BC). Gap constant = 1.0, Block length = 8 bits</td>
<td>90</td>
</tr>
<tr>
<td>6.4.</td>
<td>Cumulative distribution of burst length for the two state case (AC). Gap constant = 1.0, Block length = 8 bits</td>
<td>91</td>
</tr>
<tr>
<td>6.5.</td>
<td>Cumulative distribution of burst length for the three state case (ABC). Gap constant = 1.0, Block length = 8 bits</td>
<td>92</td>
</tr>
<tr>
<td>6.6.</td>
<td>Cumulative distribution of gap length for the two state case (AB). Gap constant = 1.0, Block length = 8 bits</td>
<td>93</td>
</tr>
<tr>
<td>6.7.</td>
<td>Cumulative distribution of gap length for the two state case (BC). Gap constant = 1.0, Block length = 8 bits</td>
<td>94</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.8</td>
<td>Cumulative distribution of gap length for the two state case (AC). Gap constant = 1.0, Block length = 8 bits.</td>
<td>95</td>
</tr>
<tr>
<td>6.9</td>
<td>Cumulative distribution of gap length for the three state case (ABC). Gap constant = 1.0, Block length = 8 bits.</td>
<td>96</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.</td>
<td>List of feedback polynomials used</td>
<td>23</td>
</tr>
<tr>
<td>3.2.</td>
<td>Prime factors of $2^n-1$</td>
<td>24</td>
</tr>
<tr>
<td>5.1.</td>
<td>BER of the channel</td>
<td>59</td>
</tr>
<tr>
<td>5.2.</td>
<td>Block and character error rate</td>
<td>60</td>
</tr>
<tr>
<td>5.3.</td>
<td>Throughput for block size of 4 bits</td>
<td>61</td>
</tr>
<tr>
<td>5.4.</td>
<td>Throughput for block size of 8 bits</td>
<td>61</td>
</tr>
<tr>
<td>5.5.</td>
<td>Mean time in state</td>
<td>62</td>
</tr>
<tr>
<td>6.1.</td>
<td>Table of calculated and measured values for one state case</td>
<td>84</td>
</tr>
<tr>
<td>6.2.</td>
<td>Table of measured and calculated BERs for the two state case</td>
<td>85</td>
</tr>
<tr>
<td>6.3.</td>
<td>Table of measurement and calculated BERs for the three state case</td>
<td>86</td>
</tr>
</tbody>
</table>
ABSTRACT

A Pseudo Random Error Sequence Generator (PRESG) that simulates the Gilbert model for a binary channel is available. The object of this thesis is to theoretically evaluate the statistics of the generator and to compare it with the statistics obtained experimentally using the Digital Channel Performance Evaluator (DCPE) - an instrument that is capable of acquiring error data and performing on-line computation of all the error statistics.

This report provides definitions of all the statistics as well as detailed derivation of all the formulae used to perform the theoretical calculations. A functional description of the two instruments—the PRESG and the DCPE is given. A comparison between the theoretical calculations and the experimental results has also been made and the agreement is found to be very close. On the basis of the close agreement obtained it is concluded that the PRESG is functioning as desired and that it should prove useful for evaluating the performance of a digital communication system.
CHAPTER 1

INTRODUCTION

In most present digital communication systems, information is handled in binary form, more specifically information is coded into binary digits "0" or "1". A block diagram of a typical digital communication system is shown in Figure 1.1.

![Block diagram of a typical digital communication system.](image)

Figure 1.1. Block diagram of a typical digital communication system.

The first element is a digital voice terminal, which typically consists of several channels transmitting pulse code modulated (PCM) voice or voice band data. This is followed by a time division multiplexer for multiplexing the associated bit streams, a bulk encryption device, a second multiplexer and an encoding device. The information is then sent over the channel by a radio or optical radio transmitter. The
channel is invariably subject to various types of noise disturbances, natural or man made. At the receiving end, based upon the signal received during a certain observation period, the receiver tries to decide whether a '0' or '1' was transmitted and regenerates a '1' or a '0' accordingly. The output from the receiver is passed to a decoding device which decodes the data according to the rules of decoding, a demultiplexer, a decryption device, another demultiplexer and finally to the user terminal.

The key point is that if no transmission errors are made due to the noise, there will be pairs of bit streams in the system which will be identical. For the model in Fig. 1.1 such pairs are represented by c and r, m and m*, u and u*, etc. We can represent any such pair (for example c and r) as

\[ r = c + e \]

where the addition is done modulo two on a bit by bit basis and where e is a sequence of bits consisting of a '1' in each position in which c and r differ and a '0' in each position where they do not.

Due to the properties of modulo two addition it is true that

\[ e = r + c \]

If the statistics of the noise source are known, a measurement of system performance can thus be made by comparing one or more pairs of particular interest on a bit-by-bit basis, computing the error statistics and comparing them with the statistics of the noise source. In order to perform such a measurement, a method for making bit-by-bit comparison of
the transmitted and received data streams is required. If the transmitter and receiver are colocated, this is not a problem. Usually, however, the transmitter and receiver will be separately located, and often several hundreds of miles away from each other. In such a case, a suitable test sequence has not only to be transmitted, but also somehow saved or recreated so that it may be added bit-by-bit to the received sequence to form the error sequence. There are several techniques for doing this, but most of them are generally complicated since they usually rely on the use of a feedback channel or require complex synchronization equipment. There is however an easy way to obviate this difficulty. This is to replace the channel by a noise source simulator with known statistics (as shown in dotted lines in Fig. 1.1) and to perform a 'bench' type measurement. The noise source simulator must however faithfully emulate the errors occurring in the channel. This is the function performed by the Pseudo Random Error Sequence Generator, which is designed to simulate the Gilbert model for an error causing channel.

The purpose of this study is to theoretically compute the statistics of the PRESG and to compare it with the experimental results obtained using the DCPE. In Chapter 2, the definitions of all the error statistics used for performance evaluation are given. The Gilbert model on which the design of the PRESG is based is also discussed. In Chapter 3 a functional description of the PRESG is given. Chapter 4 gives a description of the architecture and capabilities of the DCPE. Chapter 5 provides detailed derivation of all the formulae used to calculate the statistics as well as the results of the calculations. A comparison between the theoretical calculations and the experimental results is
presented in Chapter 6. The computer programs written for the purpose of performing the calculations are listed in the Appendix.
CHAPTER 2

TEST STATISTICS AND CHANNEL MODELS

In the previous chapter it was mentioned that the performance of a system could be evaluated using the PRESG by measuring certain test statistics at particular points in the system and by comparing them with the statistics of the PRESG. It was also mentioned that the PRESG is designed to simulate the Gilbert model for a binary channel. In this chapter we will study some of the test statistics which are of interest; we will also discuss channel models in general and examine the Gilbert model in this context.

2.1 Some Test Statistics

2.1.1 Bit Error Rate

The most simple method of characterizing the performance of a digital communications system is to measure the bit error rate (BER) which is defined [1] as follows:

\[
\text{BER} = \frac{\text{number of bit errors detected}}{\text{Total number of bits received}}
\]  \hspace{1cm} (2.1)

Referring to the earlier notation

\[
\text{BER} = \frac{\text{Number of ones in } e}{\text{Total number of digits in } e}
\]  \hspace{1cm} (2.2)

where \( e \) is the error sequence previously defined. It should be noted that BER is a random variable in the sense that even if system
characteristics remain fixed, the measured value of BER will generally be different for different test periods.

It is usually assumed that 1's occur in the error pattern with probability \( p \) for any particular bit and that this probability is independent of whether or not other bits have been in error. This is not an accurate representation of most real channels, but it greatly simplifies calculations. Under the above assumption it can be shown [1] that the measured BER is on the average equal to the probability of bit error \( p \), if the measuring time is sufficiently long.

Although easy to measure, BER lacks much of the detail to accurately determine system performance [1]. Nevertheless it is a statistic of interest since it is the most widely used performance measure and also since it is useful in defining some other statistics.

2.1.2 Throughput

In many systems the binary information has been formatted by the user into blocks of specified length, and only block errors are of interest.

In any such system throughput describes the efficiency of block transmission as follows:

\[
\text{Throughput} = \frac{\text{number of correct blocks received}}{\text{Total number of blocks transmitted}}
\]  

Under many circumstances throughput is a better measure of channel capabilities than is bit error rate [1]. This result is true because throughput groups all bit errors into block errors; if a block contains one or all errors only one block error is detected. In many
cases, independence of block error is a more valid assumption than independence of bit errors. For this reason, throughput can often be measured with more confidence than bit error rate.

If bit error rate is measured on a particular channel and each error occurs independently with probability $p$, then it can be shown [1] that the throughput for a block size of $n$ bits is governed by the following equation:

$$\text{Throughput (block size of } n \text{ bits)} = (1 - p)^n$$

(2.4)

If, however, the errors do not occur independently but the above calculation is used for predicting throughput, it can be shown [1] that the calculation of throughput based on BER yields a worst case estimate of the actual throughput. The reason for this is that clusters or bursts of errors tend to cause fewer block errors than randomly distributed independent errors. In addition, because the bit error rate measurement cannot necessarily be made with the same degree of confidence as throughput (the errors are not independent), both BER and throughput need to be considered in establishing a meaningful figure of merit.

2.1.3 Burst and Gap Statistics

Even the use of throughput in conjunction with BER measurement is not sufficient to accurately assess the performance of most digital communication systems [1]. There are several reasons for this, a few of which will be discussed.

The primary reason, of course, is that the overwhelming majority of real channels are bursty channels. This implies that the impact of
errors on the user will be quite different than if errors at the same average rate had occurred in a truly random fashion. The previous section illustrated the situation in terms of throughput referenced to a specific block size. What complicates the issue is that the impact on the user now becomes very dependent on the user’s format. Using throughput as an example, and considering extremes, if the block size is one bit then clearly throughput = 1-BER no matter how the errors might be grouped. On the other hand as block size becomes very large throughput approaches zero, again irrespective of the grouping. For intermediate block size, however, the details of grouping become important.

The difficulty with the adequacy of throughput as a performance measure is that there is no well-defined block size on which to base the calculation. It is therefore necessary to define some more fundamental statistics, which can be measured and serve as a basis for estimating performance under a wide variety of different criteria [1].

It is surprisingly difficult to define precisely what one means by a "burst of errors," which is the basis for all such calculations. The nature of the difficulty can be seen by consideration of the following example. Suppose a segment of the error sequence is as follows:

\[ e = \ldots 101001100001011000000110111\ldots \]

where '1' represents a bit (block) in error, and '0' represents a correct bit (block) and the ellipses indicate that this particular segment is preceded and followed by a large number of zeros.
Now the immediate question is: How many errors bursts are there and how long are they? The answer is clearly ambiguous and depending on the observer's viewpoint there might be one long burst or two or three short bursts. In other words it is not possible to define "burst" in a universal fashion but only as a function of one or more analysis parameters. The classic definition of a burst is as follows [2]:

Let a parameter $G$, called guard space, be fixed at some integer value. Then a burst of length $b$, relative to the guard space $G$, is a segment of the error sequence which is $b$ bits (blocks) in length and:

1. Begins and ends with a bit (block) in error.
2. Is immediately preceded and followed by at least $G$ error free bits (blocks).
3. Contains no segment consisting of $G$ or more adjacent error free bits (blocks).

Let us illustrate this definition with reference to the previous example. Suppose that $G=1$, which is the smallest possible value. Then there are seven bursts - three of length $b=1$, three of length $b=2$, and one of length $b=3$. If the value of the guard space parameter is different, however, the situation changes, as indicated by the following table. In this table the right hand column indicates how many bursts of various length exist in accordance with the above definition.

<table>
<thead>
<tr>
<th>$G$</th>
<th>number of bursts of length $b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$3(b=1), 3(b=2), 1(b=3)$</td>
</tr>
<tr>
<td>2</td>
<td>$1(b=2), 1(b=3), 1(b=4), 1(b=6)$</td>
</tr>
<tr>
<td>3</td>
<td>$1(b=4), 1(b=6), 1(b=7)$</td>
</tr>
</tbody>
</table>
Figure 2.6. Example of definition of burst of length b relative to guard space of length G.

As can be seen from the table, higher values of gap parameter permit bursts of higher length. The choice of the gap parameter is therefore critical. There is no single method for choosing the gap parameter and in fact several approaches are possible [1]. One approach is to fit the value of G to the system being tested. To say this another way, it often happens that any tested value of G in a fairly broad range (typically a few tens to a few hundreds) will yield essentially identical burst and gap distributions. It is then appropriate to choose G to be any convenient value in this range as it "naturally" separates the bursts which occur in the error sequence. Another possibility is to choose a value of G appropriate to the user's block format; that is, choose a value large enough that at least one block is sure to be transmitted successfully during the gap.

There is however a doubt regarding the usefulness of the burst definition based on the gap parameter when applied to real channels. Most such channels exhibit a superposition of two effects; the first is truly random behavior, usually at a low rate and due to receiver thermal and antenna noise; the second is truly bursty behavior due to
transmission media effects. Several authors have therefore based their definition of burst not on error free gaps but on low error density intervals.

If the classic burst definition is used, then the gap distribution is a useful statistic. This distribution is the conditional probability that, given an error burst has occurred, it will be followed by at least $J$ consecutive error free bits (blocks) where, $J > G$ the gap parameter.

2.1.4 Other Test Statistics

Other test statistics of interest include the block error rate and the character error rate. These statistics will be defined later in Chapter 5.

2.2 Channel Models

Generally speaking, it is convenient to classify channels as being either memoryless, in which case demodulation errors are random events, or having memory, in which case demodulation errors exhibit statistical dependence [3].

If the noise added to the transmitted signal is Gaussian and white, the channel is memoryless. A deep space probe is, for example, approximately memoryless. In communication theory the binary symmetric channel is the classical model of a memoryless binary channel. This channel generates a sequence of binary noise digits $z_n$ which it adds (module 2) to input digits $x_n$ to produce $y_n = z_n + x_n$. Since the binary symmetric channel is memoryless, a sequence of independent trials produces the noise digits $z_n$. Each trial has the same probability $p$ of
producing an error and 1-p of producing no error. The channel is therefore completely characterized by the crossover probability p.

Unfortunately, all real channels exhibit some form of time dispersion such as fading, and impulse noise as well as Gaussian noise is added to the signal. In such channels errors occur not randomly but in isolated bursts. Independent trials fail to simulate such a burst noise. Channels in which demodulation errors are not independent random events are said to have memory. Telephone channels, afflicted as they are by switching transients, dropouts, lightning strokes, etc., and high frequency radio channels, which are subject to fading, are examples of channels of this type. Perhaps because of the myriad of underlying physical phenomena involved, the characterization of channels with memory is more difficult than that of memoryless channels. Ideally, such a characterization should be mathematically tractable and should represent the pertinent aspects of the error process accurately. Unfortunately, no completely satisfactory characterization of channels with memory has been found.

The simplest characterization of an error causing channel is a BSC. However, the BSC is a very poor channel model if impulse noise dominates Gaussian noise and most errors occur in infrequent long bursts. To illustrate the degree of inaccuracy, Fig. 2.1 shows the distribution of the average probability that m errors will occur in a block of 31 bits, p(m, 31) for an ensemble of switched voiceband telephone channels with the corresponding distribution for a BSC with the same bit error rate: 5 \times 10^{-5}. It can be seen that the two curves differ substantially.
It seems clear that this simple model is inadequate to characterize channels with memory.

A more complex model in which the error-generating mechanism occupies one of a given number of states with a fixed probability has been successful in characterizing channels on which errors occur in bursts. This is the Gilbert model [4] which considers the channel to be composed of a small number of binary symmetric channels. The state of the channel, i.e., the BSC through which a given bit is transmitted, is controlled by a Markov process. Besides being mathematically tractable, this model is intuitively satisfying in that error bursts are often the result of a temporary change in the macroscopic physical state of the channel. The model as originally suggested uses a Markov chain with two states G and B to generate bursts as shown in Figure 2.2. In state G, the noise digit is always \( z_n = 0 \). In state B, \( z_n \) is a 0 or 1 with probability one half. The probability of one half is included because actual bursts contain good digits interspersed with errors. After producing the noise digit \( z_n \), the Markov chain makes a transition to prepare for \( z_{n+1} \). To simulate burst noise, the states B and G must tend to persist, therefore typically, the transition probabilities \( P = \text{Prob}(G \rightarrow B) \) and \( p = \text{Prob}(B \rightarrow G) \) will be small, and the probabilities \( Q \) and \( q \) of remaining in G and B will be large. Although two states are sufficient to simulate bursts, ordinarily three states are needed to define a model whose \( P(m, n) \) distribution is similar to that of a given channel. Figure 2.3 shows a three-state Gilbert model for a typical telephone channel [3].
Figure 2.1. $P(m, 31)$ distribution for telephone channel (T) and binary symmetric channel model with the same average bit error rate ($5 \times 10^{-5}$)

Figure 2.2. Transition diagram for the Markov chain

Figure 2.3. A three state Gilbert model for a typical telephone channel
CHAPTER 3

THE PSEUDO RANDOM ERROR SEQUENCE GENERATOR

The Pseudo Random Error Sequence Generator (PRESG) is designed to simulate the Gilbert model for a binary channel. The Gilbert model as discussed earlier, considers the channel to be composed of a small number of binary symmetric channels, where the state of the channel is controlled by a Markov process. The PRESG is capable of assuming three states; in each state it behaves like a binary symmetric channel, generating random errors at a predesigned bit error rate. Later in this chapter a functional description of the PRESG will be given. It is instructive to first study the theory underlying the operation of the PRESG. This will be done in the following section.

3.1 Theory of Operation

In each state the PRESG must be capable of generating error sequences that are completely random. The autocorrelation and power spectrum of such a sequence is shown in Figure 3.1. In practice, however, such a sequence would be virtually impossible to generate. What is required therefore is a sequence that has sufficient "randomness" and yet is deterministic, making it relatively easy to generate. Pseudo noise sequences, or PN sequences as they are called, have precisely such characteristics. These are periodic binary sequences which possess the
following properties associated with randomness [5, 6].

1. The Balance Property:
   In each complete period of the sequence the number of ONES differs from the number of ZEROS by at most one.

2. The Run Property:
   Among the runs of consecutive ONES and ZEROS in one complete period of the sequence, a half of the runs of each kind are of length 1, a quarter of each kind are of length 2, an eighth are of length 3, and so on.

3. The Correlation Property:
   If an exact period of the sequence is compared bit by bit with any shift of itself, the number of agreements differs from the number of disagreements by at most one.

3.1.1. Shift Register Sequences
   By far the most important method for generating PN sequences is by means of a shift register [5]. A shift register of degree n is a device consisting of n consecutive binary storage elements. The contents (ONE or ZERO) of each element can be shifted to the next position on receipt of a regular clock pulse. Clearly, if no action is taken the initial "fill" of the register would be shifted by the clock pulse until after n clock pulses the register would be empty. In order to prevent this from occurring "feedback" may be applied to the first stage of the register. The feedback must be a ONE or a ZERO and may be computed as a logical (i.e., Boolean) function of the contents of the n elements of the
shift register. A block diagram of such a device is shown in Figure 3.2. This general method of generating sequences is extremely powerful as any binary sequence may be generated by a suitable choice of the number of stages \((n)\) in the shift register, the feedback function \(f(x_1, \ldots, x_n)\) and the initial condition of the register. For any shift register of degree \(n\) the output sequence will be periodic, the maximum period being \(2^n - 1\). Any output sequence achieving \(p = 2^n - 1\) is called a "maximal length linear shift register sequence" or "linear m-sequence" for short. It is well known that such a sequence can be obtained if the feedback connections implement a primitive polynomial of degree \(n\)[5,6]. An example of a linear m sequence generator is shown in Figure 3.3.

It is easily verified [5] that shift register sequences satisfy the properties of balance, run and correlation. Referring to the example of a linear m sequence generator shown in Figure 3.3, it can be shown that the complete sequence produced from the last stage of the shift register with an initial fill of ONES will be:

\[
111100010011010
\]

This sequence repeats itself with a periodicity of 15. The balance property is satisfied as there are 8 ONES and 7 ZEROS. The run property is satisfied as among the eight runs half have unit length (two single ONES and two single ZEROS), a quarter have length two (one pair of ZEROS and one pair of ONES), and an eighth have length three (a single run of three zeros). Smaller fractions become rather academic as the maximum possible run length is four. Finally, the correlation property is as follows.
There are, therefore, seven agreements and eight disagreements, a difference of one. In fact any of the other possible shifts will give exactly the same result.

The normalized auto correlation and power spectrum for a linear m sequence of period \( p \) is shown in Figure 3.4. Whereas the random binary waveform produces a continuous spectrum, the PN waveform, due to its repetitiveness nature, produces a line spectrum.

3.1.2 Error Generation using a single shift register

Since the output of a shift register satisfies the properties of randomness, it can be used to generate errors as shown in Figure 3.5. As shown, the output from the shift register is EXORed with the incoming bit streams so that whenever a '1' occurs in the output of the shift register, an error is generated. This method of generating errors using a single shift register has a drawback however—it can only achieve one BER (one state), which is one-half. Since in order to simulate the Gilbert model more than one state is required, this method is inadequate.

3.1.3 Error Generation using several shift registers

The above mentioned problem can be solved by using the output of several shift registers as shown in Figure 3.6. Here, the outputs are ANDed together to produce the error sequence. An error is generated only if a '1' occurs simultaneously in the output of all the shift registers. The error sequence thus generated will be periodic, the period being
equal to the L.C.M. of the period of each individual sequence. If \( n \) shift registers are ANDed together, the BER of the sequence will be \( 1/2^n \). The sequences must however be chosen so that the crosscorrelation between the sequences is small and invariant for all shifts except the zero shift. This condition has to be met if the output sequence is to be random.

It is clear that the Gilbert model can be simulated using such a parallel composition of shift registers. Any desired frequency of ones (BER) can be approximated in each state by selectively ANDing together a certain number of shift registers. By using the present state to decide the state to be assumed next, the state transitions can be made to implement a Markov chain process.

3.2 Construction Techniques and Operation Mechanism

(In this section a description of how the PRESG implements the Gilbert model will be given. This includes construction details as well as a functional description of all the components of the PRESG.)

A block diagram of the PRESG is shown in Figure 3.7. It consists of a set of shift registers for generating the error sequence, state select logic for implementing the state transitions and output logic for controlling the BER in each of the three states.

3.2.1 The Shift Registers

The PRESG makes use of ten shift registers, this number being chosen both for few common prime factors and for availability of maximal length polynomials. The outputs of the shift registers are ANDed together to produce the error sequence, the number to be ANDed being
determined by the output logic. The length of the shift registers along with the feedback polynomials used is shown in Table 3.1. In order to achieve maximal length sequences only primitive polynomials have been selected. The lengths of the sequences generated by the polynomials shown in the table have few common prime factors (the sequence generated by ANDing them is therefore sufficiently long) and the sequences themselves have been found to possess good cross correlation properties. The shift register lengths chosen are 14, 15, 17, 19, 22, 26, 27, 29 and 31. Lengths of 37 and 39, although preferable, have not been used because of the non-existence of good polynomials. Table 3.2 gives a list of the prime factors of \(2^n - 1\) for \(n=14, 15, \ldots, 31\). The repetition length of the error sequence generated when all the ten shift registers are used can be calculated using the table. The repetition length of the error sequence is equal to the LCM of the repetition lengths of each individual sequence and is approximately equal to \(2.139 \times 10^{65}\). At a clock rate of 32 kb/s this sequence will repeat itself almost every \(2 \times 10^{50}\) years, which is more than adequately long for simulation purposes.

3.2.2 State Select Logic

As mentioned earlier, the transition between the states is controlled by the state select logic. The transition is done in such a way that it implements a Markov chain process. The FRESG is capable of assuming three states. At the end of each clock cycle, a ten-bit vector \(s\) is compared to two programmable levels \(L_I\) and \(L_{II}\) to decide which of the three states to assume at the beginning of the next clock cycle, the two levels being determined by the state the machine is
presently in. Thus if the machine is presently in state A, the vector is compared to the two programmable levels $L_I$ and $L_{II}$ for state A ($L_{IA}, L_{IIA}$) while if the machine is presently in states B or C, the vector is compared to the levels $L_I$ and $L_{II}$ for those states.

The ten bit vector is formed by taking one bit from each of the ten shift registers, (the machine was originally designed so that the bit was taken from the output of each of the shift registers, but this was found to produce a correlation between the error occurrence and the state transitions) with the least significant bit being taken from the shift register of length 14 and the most significant bit taken from the shift register with length 31. The magnitude of this vector is a ten bit random number that is equally likely to have one of the $2^{10}$ possible values.

Figure 3.8 illustrates how the decision is made. If the magnitude of the vector $s$ is greater than or equal to $L_{II}$ the machine goes into state C. If $s$ is less than $L_I$ the machine assumes state B while if $s$ falls in between the two levels the machine assumes state A.

The comparison between the vector and the two levels is performed by two comparators as shown in Figure 3.9. The two output bits of the comparator form the current state (CS) line which indicates the current state of the machine or the state to be assumed at the beginning of the next clock cycle.

The two levels $L_I$ and $L_{II}$ for each state are determined by the state transition matrix of the Markov process. Alternatively, given any transition matrix, it can be implemented by a proper choice of these levels. Thus, for example, if a high self transition probability
is required in state A, this can be achieved by choosing a large value for $L_{IA}$. By choosing $L_{IB}$ to be small and $L_{IIB}$ to be large, a high self transition probability can be achieved in state B for instance. It is also possible to eliminate one or more states by a proper choice of these levels. By eliminating two states, a random channel can be simulated. Figure 3.10 shows a typical transition matrix and the levels that must be selected in order to implement this matrix. The levels $L_I$ and $L_{II}$ can be chosen to have any value ranging from 0 to 1023 (as long as $L_{II} > L_I$). Because the range is limited to these values, the smallest non-zero transition probability that can be achieved is $1/2^{10}$. Once the levels are selected, they can be entered into the machine using two sets of switches available for each state.

### 3.2.3 The Output Logic

The BER in each of the three states is controlled by the output logic. The output logic accomplishes this by ANDing together a certain number of shift registers in each state. The information regarding the state of the machine is obtained by examining the state of the CS line at the end of each cycle. The number of shift registers ANDed together in each state and therefore the BER has been fixed at a preselected value. Thus, three shift registers are ANDed in state A, seven in state B and ten in state C, giving bit error rates of $1/2^3$, $1/2^7$, and $1/2^{10}$ respectively. Table 3.3 shows the state of the comparator and the BER when the machine is in each of the three states, A, B, and C.

A flowchart showing the actions of the machine during each clock cycle is given in Figure 3.11.
<table>
<thead>
<tr>
<th>Shift Register Length</th>
<th>Feedback polynomial</th>
<th>Octal Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>$x^{14} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$</td>
<td>4333</td>
</tr>
<tr>
<td>15</td>
<td>$x^{15} + x^{12} + x^{11} + x^8 + x^7 + x^6 + x^2 + 1$</td>
<td>114725</td>
</tr>
<tr>
<td>17</td>
<td>$x^{17} + x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^4 + x^2 + 1$</td>
<td>432625</td>
</tr>
<tr>
<td>19</td>
<td>$x^{19} + x^{17} + x^{15} + x^{14} + x^{13} + x^{12} + x^6 + x + 1$</td>
<td>2570103</td>
</tr>
<tr>
<td>22</td>
<td>$x^{22} + x^{19} + x^{16} + x^{13} + x^{10} + x^7 + x^4 + x + 1$</td>
<td>22222223</td>
</tr>
<tr>
<td>23</td>
<td>$x^{23} + x^{17} + x^{11} + x^9 + x^8 + x^5 + x^4 + x + 1$</td>
<td>40405463</td>
</tr>
<tr>
<td>26</td>
<td>$x^{26} + x^6 + x^2 + 1$</td>
<td>400000107</td>
</tr>
<tr>
<td>27</td>
<td>$x^{27} + x^{18} + x^{11} + x^{10} + x^9 + x^5 + x^4 + x^3 + 1$</td>
<td>1001007071</td>
</tr>
<tr>
<td>29</td>
<td>$x^{29} + x^{20} + x^{16} + x^{11} + x^8 + x^4 + x^3 + x^2 + 1$</td>
<td>4004204435</td>
</tr>
<tr>
<td>31</td>
<td>$x^{31} + x^{27} + x^{23} + x^{19} + x^{15} + x^{11} + x^7 + x^3 + 1$</td>
<td>2104210421</td>
</tr>
</tbody>
</table>

Table 3.1. List of Feedback polynomials used
Table 3.2

PRIME FACTORS OF $2^n - 1$

<table>
<thead>
<tr>
<th>n</th>
<th>$2^n - 1$</th>
<th>Prime Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>3, 5</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>63</td>
<td>3, 17</td>
</tr>
<tr>
<td>7</td>
<td>127</td>
<td>127</td>
</tr>
<tr>
<td>8</td>
<td>255</td>
<td>3, 5</td>
</tr>
<tr>
<td>9</td>
<td>511</td>
<td>511</td>
</tr>
<tr>
<td>10</td>
<td>1023</td>
<td>3, 17</td>
</tr>
<tr>
<td>11</td>
<td>2047</td>
<td>11, 19</td>
</tr>
<tr>
<td>12</td>
<td>4095</td>
<td>3, 5, 17</td>
</tr>
<tr>
<td>13</td>
<td>8191</td>
<td>8191</td>
</tr>
<tr>
<td>14</td>
<td>16383</td>
<td>7, 239</td>
</tr>
<tr>
<td>15</td>
<td>32767</td>
<td>3, 10867</td>
</tr>
<tr>
<td>16</td>
<td>65535</td>
<td>3, 5, 17</td>
</tr>
<tr>
<td>17</td>
<td>131071</td>
<td>131071</td>
</tr>
<tr>
<td>18</td>
<td>265521</td>
<td>265521</td>
</tr>
<tr>
<td>19</td>
<td>524287</td>
<td>524287</td>
</tr>
<tr>
<td>20</td>
<td>1048575</td>
<td>5, 209715</td>
</tr>
<tr>
<td>21</td>
<td>2097151</td>
<td>2097151</td>
</tr>
<tr>
<td>22</td>
<td>4194303</td>
<td>3, 193, 2771</td>
</tr>
<tr>
<td>23</td>
<td>8388607</td>
<td>3, 193, 2771</td>
</tr>
<tr>
<td>24</td>
<td>16777215</td>
<td>3, 193, 2771</td>
</tr>
<tr>
<td>25</td>
<td>33554431</td>
<td>33554431</td>
</tr>
<tr>
<td>26</td>
<td>67108863</td>
<td>3, 223, 36667</td>
</tr>
<tr>
<td>27</td>
<td>134217727</td>
<td>134217727</td>
</tr>
<tr>
<td>28</td>
<td>268435455</td>
<td>268435455</td>
</tr>
<tr>
<td>29</td>
<td>536870911</td>
<td>536870911</td>
</tr>
<tr>
<td>30</td>
<td>1073741823</td>
<td>1073741823</td>
</tr>
<tr>
<td>31</td>
<td>2147483647</td>
<td>2147483647</td>
</tr>
</tbody>
</table>

The repetition length of the sequence is equal to the product of the numbers across the top of the table. This product is $\approx 2.139 \times 10^{65}$. At a clock rate of 32kb/s, this sequence will repeat itself almost every $2 \times 10^{50}$ years.
Figure 3.1. a) Random binary sequence b) its autocorrelation function c) its power spectrum

Figure 3.2. Generalized shift register sequence generator
Figure 3.3. A maximum-length shift register sequence generator of degree 4

Figure 3.4. a) Normalized autocorrelation function of a PN binary waveform of bit duration $T$. b) Power spectrum of a PN binary waveform of period $P$ having the autocorrelation shown in a
Figure 3.5. Error Generation using a single shift register

INCOMING SIGNAL

Figure 3.6. Error generation using several shift registers

INCOMING SIGNAL
Figure 3.7. Block diagram of the PRESG

Figure 3.8. Decision logic for implementing state transitions
Figure 3.9. Comparators for comparing the ten-bit vector with the two programmable levels.

<table>
<thead>
<tr>
<th>Current State/P(next state)</th>
<th>(P(A))</th>
<th>(P(B))</th>
<th>(P(C))</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.7</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>B</td>
<td>0.05</td>
<td>0.8</td>
<td>0.15</td>
</tr>
<tr>
<td>(a)</td>
<td>C</td>
<td>0.03</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Figure 3.10 a) Typical transition matrix b) Levels that must be selected in order to implement this matrix.
Figure 3.11. Actions of the PRESG during each clock cycle
CHAPTER 4

THE DIGITAL CHANNEL PERFORMANCE EVALUATOR

4.1 General

The Digital Channel Performance Evaluator (DCPE) is a device built by industry to specifications suggested by reference [A]. It features multiprocessor architecture and is capable of acquiring, processing and storing error data, and can be programmed to perform real time calculation of error statistics such as gap and burst length distribution, burst density, bit error rate, character error rate and block error rate. These statistics which are continuously updated are available for display to the user at any time through the front panel display switches.

The DCPE is designed to operate at a nominal data rate of 32 kbps. Single bit fineness at this rate requires that all calculations be completed within a 31.25 μsec period, which is not feasible using currently available microprocessors. A compromise between data fineness and time availability for computation is therefore necessary. An eight bit measurement block size gives a computation time of 250 μ secs which is sufficient for all calculations to be completed while providing adequate resolution at the same time. This is the nominal block size adopted for the DCPE. The block size can, however, be altered by changing a program constant—a facility that is required when the DCPE is operating in a second mode where data rates up to 20 Mbps are possible.
The constant must, however, be chosen to define a block size at least 250
seconds long, which permits approximately 500 machine functions of the
microprocessor. During this time the microprocessor executes a program
which computes and stores a running value of each of the statistics given
below [7].

4.2 Long Term Error Rates

The DCPE counts the number of blocks that are in error as well as
the total number of errors occurred. This provides both long-term BER
and character error rate or block error rate.

When operating in Mode I, the DCPE actually counts the total
number of data bits for calculating the error rate. The high clock rates
in Mode II preclude actual counting and error rate is calculated using
the data rate and total block time. As a result, accurate calculation of
error rates requires an accurate knowledge of the data rate.

Character error rate is determined by calculating the block error
rate with the data format set for a basic block length of 8 bits. The
DCPE can evaluate block error rate for any block length up to 4800 bits.

4.3 Short-Term Error Rates

Short-term error rates at data rates up to 20 Mbps can be
obtained when the DCPE is operating in Mode II. The DCPE counts the
number of errors in each data block of a specified block length for the
purpose of computing short-term bit error rates. Short-term character
error rate is determined by subdividing the block into an 8-bit
character.

The maximum block length (interval) at 20 Mbps is 20 seconds with
a sustained average error rate of $10^{-2}$ for a worst case of 4 million errors. The block lengths (intervals) are selectable from 1, 2, 5, 10, or 20 seconds. With a large number of intervals the BERs can then be plotted as a cumulative distribution to indicate channel availability as the percentage of time the BER is less than a given value.

4.4 Gap and Burst Statistics

The DCPE considers an error-free gap as being a sequence of consecutive data blocks which contain no errors. A variable program constant is used to specify the least number of error-free blocks necessary to constitute a gap. This constant is variable from 1 to $2^{14}$ blocks. A burst is then the interval preceded and followed by an error-free gap. The DCPE calculates and maintains cumulative distribution of gap and burst length and burst densities. These are quantized in terms of $\log_2$ and stored in 24 locations. These distributions can be used to compute throughput and availability information.

4.5 Interruption Statistics

In order to prevent incorrect recording of error statistics caused by equipment failure, an error which persists for more than a certain period of time is termed an interruption. The interruption is termed a short interruption if the burst length exceeds the short interruption parameter but not the long interruption parameter. The short interruption parameter may be set to any value between 4 to 1600 blocks. The long interruption parameter can be set to any value between 8 and $2^{16} - 1$ blocks.

The calculated values that are stored and available for display.
are the total number of short and long interruptions, total number of errors in short and long interruptions, duration of interruptions and start and end time of interruptions.

4.6 DCPE Architecture

The DCPE is a multiple microprocessor system with interfacing capability to a variety of inputs and outputs. The DCPE includes three 8-bit microprocessors, one 16-bit special purpose arithmetic processor, 9 k words of random access memory and 9 k words of internal fixed program memory.

The DCPE consists of the following subsystems:

1. Error Counting and Interface Subsystem
2. Control and Arithmetic Subsystem
3. Housekeeping Subsystem
4. GPIB subsystem

These subsystems are functionally connected as shown in Figure 4.1. A concise description of each of these subsystems follows.

4.6.1 Error Counting and Interface Subsystem

The Error Counting and Interface Subsystem is shown in more detail in Figure 4.2. This subsystem includes the Pseudo Random Sequence Generator, the Pseudo Random Error Detector, the Error Counters, the block counters, the interface cards and the system timing generator.

The DCPE can operate by accepting clock and error data from an external error detection system. Alternatively, it can generate test bit streams and detect received errors by the use of the Pseudo Random Sequence Generator and Error Correlator Subsystem. The Pseudo Random
Sequence Generator generates a nonrepeating, binary sequence of length $2^{24}$ bits to serve as a test message.

The error counters consist of two $2^{24}$ bit capability counters which count errors from the Pseudo Random Error Detector. The contents of the error counters are read by the Control and Arithmetic Subsystem through the Control Data Bus.

The block counter is used to count block periods. In Modes I and III the block counter counts data blocks to determine blocks. In Mode II operation, the block counter counts 2000 one MHz system clocks to establish a two millisecond block period.

4.6.2 Control and Arithmetic Subsystem

The Control and Arithmetic Subsystem is responsible for all data acquisition and storage during a test. It also performs special arithmetic functions in response to demands from the Housekeeping Subsystems.

The Control and Arithmetic Subsystem consists of a control microprocessor with associated random access memory, program memory and decoding. It includes the system joint memory and arithmetic processing unit, the control interrupt vectoring unit and the high speed dual-port control/housekeeper memory.

The control microprocessor includes a Type 6502 processor with 1 k words of RAM and provisions for $4^k$ words of ROM, of which 3 k words are used. The program memory includes all routines needed to acquire, interpret and store data and statistics obtained by the DCPE.

System arithmetic calculations are accomplished by a high speed
arithmetic unit capable of 16 bit fixed point, 32 bit fixed point and 32 bit floating point calculations. This unit performs all on line arithmetic and supports off line computations required for data output and display.

4.6.3 Housekeeping Subsystem

The Housekeeping Subsystem is shown in Figure 4.4. It includes a 6502 microprocessor with associated random access/fixed memory, the system keyboard, the LED display, the recorder, the video display generator and the GPIB to Housekeeper two-port memory.

The Housekeeper performs a variety of functions, some of which are as follows:

a) Initiation and setting up of all parameters
b) Entering of data to the LED display
c) Transferring data to the magnetic tape through its 256 word buffers
d) Writing data to the video screen
e) Transferring data to external devices through the GPIB Subsystem.

4.6.4 The GPIB Subsystem

A block diagram of the GPIB Subsystem is shown in Figure 4.5. This subsystem provides access via a standard interface (IEEE 488, 1975) to any external instrumentation which is also equipped with the interface. A separate microprocessor is employed to manage this function so that it need not be synchronous with any internal functions.

The GPIB Subsystem is capable of answering status requests.
without the support of the Housekeeper microprocessor. The GPIB Subsystem also decodes secondary commands from the GPIB Bus and passes them to the Housekeeper if the command requests its support.

As shown, the subsystem includes a 6502 microprocessor, 1 k of fixed word (EPROM) memory, a special purpose GPIB interface and drivers and receivers meeting the GPIB specification.
Figure 4.1. Block diagram of the DCPG 200
Figure 4.2. Error counting and interface subsystem
Figure 4.3. Control and arithmetic subsystem
Figure 4.4. Housekeeping subsystem
Figure 4.5. GPIB (IEEE-488) subsystem
CHAPTER 5

DERIVATION AND CALCULATION OF ERROR STATISTICS

In Chapter 2, the definition of all the test statistics as well as the justification for their use was given. These statistics have been calculated for the PRESG. In this chapter the derivation of all the formulae used to perform the calculations as well as the results of the calculations are presented.

5.1 Derivation of Statistics

The statistics have first been derived for the one-state case. This is the case when the PRESG behaves like a binary symmetric channel. (The usual assumptions regarding binary symmetric channels have therefore been made in the derivations). Later in this chapter, we will see how the statistics of the two and three state case can be obtained from the statistics for the one state case.

5.1.1 Gap Distribution

As mentioned before, a region of error free blocks between two blocks in error is defined as a gap provided the number of error free blocks > G, where G = the gap constant. The gap distribution is the plot of the cumulative relative frequency of the gap length m versus the length m.
Let

\[ P_{\text{bit}} = \text{probability of bit in error} \]
\[ p = \text{probability of block being in error} \]
\[ q = \text{probability of correct block} \]
\[ n = \text{no. of bits in a block} \]

Then,

\[ q = (1 - P_{\text{bit}})^n \] \hspace{1cm} \ldots \ldots (5.1)

Let us assume that the test has just been terminated with a block in error. The gap distribution \( P_g(m) \) is then the probability of finding \( m \) or less error free blocks before the next incorrect block.

For \( m < G \)

\( P_g(m) \) is evidently zero

For \( m > G \)

\[ P_g(m) = \frac{p[q^G + q^{G+1} + \ldots + q^m]}{1 - [p + pq + pq^2 + \ldots + pq^{G-1}]} \]

\[ = \frac{pq^G (1 - q^{m-G+1})}{1 - q} \]

\[ = 1 - q^{m-G+1} \] \hspace{1cm} \ldots \ldots \hspace{1cm} (5.2)

For \( G = 1 \), \( P_g(m) = 1 - q^m \) \hspace{1cm} \ldots \ldots \hspace{1cm} (5.3)
5.1.2 Burst Length Distribution

Even if a channel is random, there are more errors in certain portions of the data stream. These are commonly referred to as bursts. In order to describe the burst quantitatively, it is necessary to know where the burst begins and where it ends. This can be done if the burst is required to satisfy the following criterion (as explained in Chapter 2). First, a burst must begin and end with a block in error. Secondly, it must be immediately preceded and followed by at least \( G \) successive error free blocks, where \( G \) is the gap constant. Finally, the segment must not contain \( G \) or more successive error free blocks.

With the burst well defined, the burst distribution is the plot of the cumulative relative frequency of the burst of length \( m \) versus the length \( m \).

Let us consider the following two cases.

**Case I**

This is the case when \( G = 1 \). Let '1' denote a block in error and '0' a correct block. Assume that a gap has just been terminated. Then a burst of length 1 could be any one of the following:

\[
/10 \quad /100 \quad /1000 \quad \text{and so on.}
\]

The probability of occurrence

\[
= [q + q^2 + \ldots] \, p
\]

\[
= \frac{q[1 - q^n]}{1 - q} \, p
\]

\[
\lim_{n \to \infty}
\]
\[ \frac{pq}{1-q} \] \hspace{1cm} \ldots \ldots (5.4)\]

Using the same argument, the probability distribution of a burst of length \( m \), \( P(m) \)

\[ = \frac{1}{1 - q} \left[ p + p^2 + \ldots + p^m \right] q \]

\[ = 1 - p^m \] \hspace{1cm} \ldots \ldots (5.5)\]

**Case II**

This is the case when \( G > 1 \). Let us first determine the density function \( p_b(m) \) for this more general case.

A burst of length \( m \) is one of the following:

1. \( \underbrace{1 \ldots \ldots 1}_{m \text{ G zeroes}} 0 0 0 \ldots 0 \)
   - sequence of length \( m \) \( \text{G zeroes} \)
   - where no more than \( G - 1 \) zeroes occur in succession

or

1. \( \underbrace{1 \ldots \ldots 1}_{m \text{ G + 1 zeroes}} 0 0 0 \ldots 0 \)
   - \( m \text{ G + 1 zeroes} \)

or
1 \ldots \ldots \ldots 1 0 0 \ldots \ldots \ldots 0

\begin{align*}
\text{m} & \quad \text{G + 2 zeroes}
\end{align*}

and so on.

The density function \( p_b(m) \) is then

\[
p_b(m) = [q^G + q^{G+1} + \ldots ] \times \quad \text{[Probability of a sequence of length m where no more than G-1 zeroes occur in succession]}
\]

For the purpose of illustration assume \( G = 3, m = 6 \). The sequence could then be

100101 or 101001 or 111111 etc.

The number of ways this can occur can be solved using the theory of partitions. Consider the same example where \( G = 3 \) and \( m = 6 \). One possible sequence of length 6 is 1 0 1 0 1 1. \( \\psi \\psi \psi \)

In this case there are 3 partitions (spaces between the ones). Note that partitions of length zero are allowed.

If \( m = \) length of the sequence

\( k = \) number of partitions

Then, \( k + 1 = \) number of ones and the number of zeroes = \( m - (k + 1) \)

= \( N \) say
The problem is then to find the number of ordered partitions of \( m - (k + 1) \) into \( k \) parts without any part exceeding \( G - 1 \) but partitions of zero being allowed. The partition of zero can be eliminated by adding 1 to each partition. This then reduces to the number of ordered partitions of \( N + k \) into \( k \) positive parts without any part exceeding \( G \).

Call this \( Y(N + k, k, G) \).

Then, the probability of a sequence of length \( m \) is simply
\[
\sum_{k} Y(N + k, k, G) \, p^{k+1} q^{N-k} \tag{5-6}
\]
and the probability density of a burst of length \( m \)
\[
= \frac{q^{G}}{1-q} \sum_{k} Y(N + k, k, G) \, p^{k+1} q^{N-k} \tag{5.7}
\]

We are now left with finding an expression for \( Y(N+k, k, G) \) and determining the upper and lower limits for \( k \).

Let us first determine
\[
Y(N + k, k, G)
\]
This is a standard problem in the theory of ordered partitions (compositions) and is given by [8] the coefficient of \( x^{N+k} \) in the expansion
\[
(x + x^2 + \ldots + x^G)^k
\]
Writing the above expression as
\[
\left[ \frac{x(1-x^G)}{1-x} \right]^k = x^k (1-x^G)^k (1-x)^{-k}
\]
it is easily seen that the coefficient of \( x^{N+k} \) is the coefficient of \( x^N \) in the expansion
\[(1 - x^G)^k (1 - x)^{-k}\]

The expression on the right is an infinite series

\[
1 + \frac{kx}{1} + \frac{k(k+1)x^2}{2-1} + \ldots + \frac{k(k+1)\ldots(k+r-1)x^r}{r(r-1)\ldots1} + \ldots
\]

where the coefficient of \(x^r = \binom{k + r - 1}{k - 1}\)

The expression on the left is the series

\[
1 - \binom{k}{1}x^G + \binom{k}{2}x^{2G} + \ldots (-1)^r \binom{k}{r}x^{Gk}
\]

The coefficient of \(x^N\) in the product of the two series is therefore

\[
\sum_{j=0}^{\text{INT}(N/G)} (-1)^j \binom{k}{j} \binom{k+N-jG-1}{k-1}
\]

\[
\sum_{j=0}^{\text{INT}(N/G)} (-1)^j \binom{k}{j} \binom{m-2-jG}{k-1}
\]

since, \(k + N = m - 1\).

Let us now determine the upper and lower limits of \(k\).

The maximum value of \(k\) (number of partitions) is evidently \(m - 1\).

The minimum value of \(k\)

\[
= m - (\text{MAXZ} + 1) \quad \text{where MAXZ = Maximum number of zeroes that can occur in a sequence of length m.}
\]

The maximum number of zeroes can be shown to be

\[
= m - 2 - \text{INT} \left[ \frac{m-2}{G} \right] \quad \text{if m is even}
\]
Substituting the expression for \( Y(N + k, k, G) \) in equation [5.7] gives the following double summation

\[
p_b(m) = \frac{q^G}{1-q} \sum_k \frac{\text{INT}[N/G]}{k^{k+1}} q^k \sum_j (-1)^j \binom{k}{j} \binom{m-2-jG}{k-1}
\]

(5.9)

The burst length distribution \( p_b(m) \) is then

\[
= \frac{q^G}{1-q} \sum_{i=1}^m \sum_k \frac{\text{INT}[N/G]}{k^{k+1}} q^k \sum_j (-1)^j \binom{k}{j} \binom{m-2-jG}{k-1}
\]

(5.10)

5.1.3 Block and Character Error Rate

Long term block error rate can be shown on the average to equal the probability of a block in error [1].

If,

\[
P_{\text{bit}} = \text{probability of bit in error}
\]

\[
n = \text{number of bits in a block}
\]

\[
q = \text{probability of a correct block}
\]

\[
= (1 - P_{\text{bit}})^n
\]

(5.11)

Then,

\[
\text{Long term Block Error Rate} = 1 - q = 1 - (1 - P_{\text{bit}})^n
\]

(5.12)
Similarly, long term character error rate can be shown [1] on the average to be equal to the probability of a character being incorrect. If,

\[ n_c = \text{character size in bits} \]
\[ q_c = \text{probability of character being correct} \]
\[ = (1 - p_{\text{bit}})^{n_c} \]

Then,

Long term character error rate \[= 1 - q_c \]
\[= 1 - (1 - p_{\text{bit}})^{n_c} \]

\[ \ldots \ldots (5.13) \]

5.1.4 Throughput

As previously defined

\[ \text{Throughput} = \frac{\text{No. of correct blocks received}}{\text{No. of blocks sent}} \]

If errors occur independently with probability \( p_{\text{bit}} \), then the long term throughput for a block size of \( n \) bits can be shown [1] to be governed by the following equation:

Throughput (block size of \( n \) bit) \[= (1 - p_{\text{bit}})^n \]

\[ \ldots \ldots (5.14) \]

5.1.5 Transition Matrix

The Gilbert model considers the channel to be composed of a small number of BSC where the state of the channel is controlled by a Markov process.

Since the transition matrix \( P \) of a Markov process is a stochastic matrix, it satisfies the following properties of a stochastic matrix [9,10].
1. P has a unique fixed probability vector $t$, and the components of $t$ are all positive.

2. The sequence $P$, $P^2$, $P^3$... of powers of $P$ approach the matrix $T$ whose rows are each the fixed point $t$.

3. If $p$ is any probability vector, then the sequence of vectors $pP$, $pP^2$, $pP^3$... approach the fixed point $t$.

Since $t$ is a probability vector, properties (ii) and (iii) imply that

$$tP = t \quad \ldots \ldots \quad (5.15)$$

The probability of any state or the mean time in that state is therefore controlled by the transition matrix. Alternatively the transition matrix can be chosen to satisfy certain probability of state or mean time requirements.

Assuming a two state model and suppose the requirement is that

$$P(B) = 10P(A)$$

Since,

$$P(A) + P(B) = 1$$

$$P(A) = \frac{1}{11}, \; P(B) = \frac{10}{11}$$

Let $P$ the the transition matrix of the Markov chain process be the following:

$$P = \begin{pmatrix} A & B \\ A & (1-p) & p \\ B & q & 1-q \end{pmatrix}$$

Applying equation (5.15),
\[
\begin{pmatrix}
1 & 0 \\
11 & 11
\end{pmatrix}
\begin{pmatrix}
1-p & p \\
q & 1-q
\end{pmatrix}
= \begin{pmatrix}
1 & 0 \\
11 & 11
\end{pmatrix}
\]
\[
\frac{1}{11} (1-p) + q \frac{10}{11} = \frac{1}{11}
\]

\[q = \frac{p}{10}\]

Therefore, the probability of going to state B in the next step given state A has occurred has to be ten times the probability of going to state A given state B has occurred.

5.1.6 Mean time in state

The state transition diagram of a two state Markov chain process is shown below.

![State transition diagram of a two state Markov chain process](image)

**Figure 5.1** State transition diagram of a two state Markov chain process

The state transition matrix \( P \) of the chain is

\[
P =
\begin{pmatrix}
A & B \\
A & r & p \\
B & q & s
\end{pmatrix}
\]

The conditional probability that given a state has occurred, the process will remain in that state for \( m \) steps is a random variable \( x \).
The expected value of this random variable is then the mean time the
process spends in that state.

Assuming that the process is presently in state A, the
probability that it will go to state B
in one step = \( p \)
in two steps = \( rp \)
in three steps = \( r^2p \) and so on.
The mean time in state A is therefore
\[
= p + 2rp + 3r^2p + \ldots .
\]
\[
= p \left[ 1 + 2r + 3r^2 + \ldots \right]
\]
This infinite series converges if \( r < 1 \) and is
\[
= \frac{p}{(1-r)^2} = \frac{1}{p} \quad \ldots \ldots (5.16)
\]
Similarly the mean time in state B
\[
= \frac{q}{(1-s)^2} = \frac{1}{q} \quad \ldots \ldots (5.17)
\]

5.1.7 Statistics for the two and three state case

The BER, throughput, block and character error rate for the two
and three state case can be easily evaluated using the statistics derived
from the one state case in the preceding sections. Considering a three
state case in particular, where the three states A, B, and C have
probabilities of occurrences \( P(A) \), \( P(B) \), and \( P(C) \) respectively and where
\( P_{EA} \), \( P_{EB} \) and \( P_{EC} \) are the statistics in each of the states,
the combined statistic is simply given by
\[ P_E = P_{EA} P(A) + P_{EB} P(B) + P_{EC} P(C) \]

This simple expression is, however, inaccurate for calculating burst and gap length distributions. There are two reasons for this, both of which can be illustrated by considering the burst length distribution. The first reason is that since the statistics of the one-state case have been derived assuming that the process continues indefinitely in that state, they are representative of the statistics of the state only if the process spends sufficient time in that state. This can be clarified by an example. Consider a process with two states A and B where state A produces errors and where state B produces no errors. If for a particular transition matrix for this process, the probability of the process staying in state A for more than \( x \) blocks is zero then the probability of a burst of length \( x \) will be zero regardless of what \( P_{BA} \), the one-state statistic for state A predicts. It is clear that the statistics of the one-state case are not valid in the multiple state case. It is therefore necessary to calculate new statistics for each state \( (P_{BA}', P_{BB}', P_{BC}') \) which are based on the statistics of the one-state case \( (P_{BA}, P_{BB}, P_{BC}) \) and the particular transition matrix for the process.

In order to do this, we need to consider two different cases. The first case is when the mean time in the state is much larger than the average length of a burst in that state. In this case the statistics of the one state case are valid and \( P_{BA}' \) (considering state A for the
moment) can be taken to be equal to $P_{BA}$. The second case is when the mean time is smaller than the average length of an error burst. In this case $P_{BA}'$ is calculated using the equation

$$P_{BA}' = \sum_{i} \left( P_{BA}' / a_i \right) P(a_i)$$

Here, $a_i$ denotes the event [Process stays in state A for $i$ blocks] and $P(a_i)$ the probability of that event - this probability can be easily calculated from the transition matrix (section 5.16). The statistic $P_{BA}' / a_i$ represents the distribution of burst length given the occurrence of event $a_i$. This statistic can be obtained from the one-state statistic $P_{BA}'$.

The other reason why expression 5.19 is inaccurate is that since state A has a much higher BER than states B or C, the number of bursts occurring in state A is much higher than the number of bursts occurring in states B or C. Consequently, the combined statistic will be biased towards the statistic of state A. In order to take this bias into account, expression 5.19 needs to be modified so that the statistic of each state is weighted not only according to the probability of occurrence of that state but also according to the BER in that state. For the case when the mean time is much larger than the average length of an error burst, this can be done in the following manner (we can proceed in a similar way in the case where the mean time is smaller than the average length of an error burst).

Assuming that the burst length distribution has been measured for the one-state case for each of the states A, B, and C for time T. Let the total number of bursts occurring in states A, B, and C be $n$, $m$, and $y$.
respectively and let the number of burst of length \( b \) be \( n_b \), \( m_b \) and \( x_b \) respectively. Now, assume that the burst length distribution is measured for the three state case for the same time \( T \) with the states A, B and C being made to occur with relative frequencies \( p \), \( q \) and \( r \) respectively. Then the relative frequency of occurrence of a burst of length \( b \) will be

\[
\frac{p_n b + q_m b + r x_b}{p_n + q_m + r y}
\]

The probability of a burst of length \( b \) \( P_b \) is then

\[
= \lim_{n \to \infty} \frac{p_n b + q_m b + r x_b}{p_n + q_m + r y}
\]

\[
= \lim_{n \to \infty} \lim_{m \to \infty} \lim_{y \to \infty} \frac{p_n b}{p_n + q_m + r y} + \frac{q_m b}{p_n + q_m + r y} + \frac{r x_b}{p_n + q_m + r y}
\]

\[
= \frac{P_{BA}}{1 + \frac{q_m + r y}{p_n}} + \frac{P_{BB}}{1 + \frac{p_n + q_m}{q_m}} + \frac{P_{BC}}{1 + \frac{p_n + q_m}{r y}}
\]

since,

\[
\lim_{n \to \infty} \frac{n_b}{n} = P_{BA}, \quad \lim_{m \to \infty} \frac{m_b}{m} = P_{BB}, \quad \lim_{y \to \infty} \frac{x_b}{y} = P_{BC}
\]

If we now assume that the total number of bursts occurring in each state is proportional to the probability of bit error in that state, then
\[ P_b = \frac{P_{BA}}{1 + q \frac{P_B}{P_A} + r \frac{P_C}{P_A}} + \frac{P_{BB}}{1 + \frac{P_B}{P_A} + r \frac{P_C}{P_A}} + \frac{P_{BC}}{1 + \frac{P_A}{P_B} + q \frac{P_B}{P_C}} \]  

(5.23)

where, \( P_A \), \( P_B \), and \( P_C \) are the probabilities of bit error in states A, B, and C respectively.

In the above formulation we have ignored 'end effects'. That is we have assumed that a burst starting in state A for instance terminates when the process goes into states B or C. This assumption is justified if either all the states or if states B and C have low error rates. If this is not so, 'end effects' cannot be ignored. Calculation of burst (or gap) length distribution in such a case is fairly complicated. This area has been left for future study.

5.2 Calculations

The statistics of interest are

1. BER, character error rate and block error rate
2. Cumulative distribution of burst and gap length
3. Throughput

Each of these statistics was calculated for two cases

Case I

Measurement block size = 4 bits
Character size = 8 bits
Throughput block = 16, 32, 64, 128 measurement blocks
Gap Length parameter = 1, 2, 4, 8, 16 measurement blocks
Case II

Measurement block & character size = 8 bits

Throughput block = 16, 32, 64, 128 measurement blocks

Gap Length parameter = 1, 2, 4, 8 measurement blocks

The calculations were performed assuming that the channel is in one of the four states A, B, C, or D. The BER of the channel in each of the four states is given below.

Table 5.1. BER of the Channel

<table>
<thead>
<tr>
<th>State</th>
<th>State</th>
<th>State</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BER</td>
<td>$1/2^3$</td>
<td>$1/2^5$</td>
<td>$1/2^7$</td>
</tr>
</tbody>
</table>

5.2.1 Gap Distribution:

The gap distribution was calculated for gap constant values 1, 2, 4, and 8 with BER and block length as parameters. The calculations were done using the program GAPLEN (See Appendix A). The calculated values have been plotted in Figures 5.2 - 5.9.

5.2.2 Burst Length Distribution:

The burst length distribution was similarly calculated using the program BURLEN (See Appendix B) written for this purpose. The effect of varying the gap constant and block length can be observed from Figures 5.10 - 5.19. For long burst lengths the binomial coefficient in the expression for the burst length distribution (5.10) is very time consuming to evaluate and therefore the curves of the burst length
distribution is these cases have been extrapolated.

5.2.3 Block and Character Error Rate

The block and error rate were calculated for block length values of 4 and 8 bits. The calculations were done in each case for the four different bit error rates. The calculated values are shown tabulated below.

Table 5.2 Block and Character Error Rate

<table>
<thead>
<tr>
<th>BER</th>
<th>Block Error Rate</th>
<th>Character Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2^3</td>
<td>0.41381</td>
<td>0.65639</td>
</tr>
<tr>
<td>1/2^5</td>
<td>0.11926</td>
<td>0.22430</td>
</tr>
<tr>
<td>1/2^7</td>
<td>0.03088</td>
<td>0.06081</td>
</tr>
<tr>
<td>1/2^10</td>
<td>0.0039</td>
<td>0.00778</td>
</tr>
</tbody>
</table>

For block size of 8 bits, the Block Error Rate = Character Error Rate and is shown in the last column of the Table above.

5.2.4 Throughput

The calculation of throughput was done for throughput block sizes of 16, 32, 64 and 128 measurement blocks. The calculated values of throughput along with the BERS and block length used in the calculations are calculated below.
Table 5.3 Throughput for block size of 4 bits.

<table>
<thead>
<tr>
<th>BER</th>
<th>Throughput Block Size</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2³</td>
<td>16</td>
<td>0.00019</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>3.78 x 10⁻¹⁷</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1.4258 x 10⁻¹⁵</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>2.03293 x 10⁻³⁰</td>
</tr>
<tr>
<td>1/2⁵</td>
<td>16</td>
<td>0.13108</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.01718</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.00295</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>8.72 x 10⁻⁸</td>
</tr>
<tr>
<td>1/2⁷</td>
<td>16</td>
<td>0.60534</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.36643</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.13427</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.01803</td>
</tr>
<tr>
<td>1/2¹⁰</td>
<td>16</td>
<td>0.93938</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.88244</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.77870</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.60638</td>
</tr>
</tbody>
</table>

Table 5.4 Throughput for block size of 8 bits

<table>
<thead>
<tr>
<th>BER</th>
<th>Throughput Block Size</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2³</td>
<td>16</td>
<td>3.78 x 10⁻¹⁷</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>1.4258 x 10⁻¹⁵</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>2.0329 x 10⁻³⁰</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>4.1328 x 10⁻⁶⁰</td>
</tr>
<tr>
<td>1/2⁵</td>
<td>16</td>
<td>0.01718</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.00295</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>8.72 x 10⁻⁸</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>7.5997 x 10⁻¹⁵</td>
</tr>
<tr>
<td>1/2⁷</td>
<td>16</td>
<td>0.36643</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.13427</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.18030</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.00032</td>
</tr>
<tr>
<td>1/2¹⁰</td>
<td>16</td>
<td>0.88244</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>0.77870</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.60638</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.36769</td>
</tr>
</tbody>
</table>
5.2.5 Mean Time in State

The mean time in states A and B (refer to Figure 5.1) was calculated for different values of the transition probability \( p \) (with \( q = p/10 \)) and is shown below. A clock frequency of 10 MHz was assumed.

Table 5.5 Mean time in state

<table>
<thead>
<tr>
<th>q</th>
<th>Mean Time (steps) A</th>
<th>Mean Time (steps) B</th>
<th>Mean Time (( \mu ) secs) A</th>
<th>Mean Time (( \mu ) secs) B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0009</td>
<td>111.11</td>
<td>1111.11</td>
<td>11.11</td>
<td>111.11</td>
</tr>
<tr>
<td>0.001</td>
<td>100</td>
<td>1000</td>
<td>10.0</td>
<td>100.0</td>
</tr>
<tr>
<td>0.002</td>
<td>50</td>
<td>500</td>
<td>5.0</td>
<td>50.0</td>
</tr>
<tr>
<td>0.004</td>
<td>25</td>
<td>250</td>
<td>2.5</td>
<td>25.0</td>
</tr>
<tr>
<td>0.006</td>
<td>16.66</td>
<td>166.6</td>
<td>1.66</td>
<td>16.6</td>
</tr>
<tr>
<td>0.008</td>
<td>12.5</td>
<td>125</td>
<td>1.25</td>
<td>12.5</td>
</tr>
<tr>
<td>0.01</td>
<td>10</td>
<td>100</td>
<td>1.0</td>
<td>10.0</td>
</tr>
<tr>
<td>0.02</td>
<td>5</td>
<td>50</td>
<td>0.5</td>
<td>5.0</td>
</tr>
<tr>
<td>0.04</td>
<td>2.5</td>
<td>25</td>
<td>0.25</td>
<td>2.5</td>
</tr>
<tr>
<td>0.06</td>
<td>1.66</td>
<td>16.6</td>
<td>0.166</td>
<td>1.66</td>
</tr>
<tr>
<td>0.08</td>
<td>1.25</td>
<td>12.5</td>
<td>0.125</td>
<td>1.25</td>
</tr>
<tr>
<td>0.1</td>
<td>1.0</td>
<td>10</td>
<td>0.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Figure 5.2. Cumulative distribution of gap length. Gap constant = 1.0, Block length = 4 bits
Figure 5.3. Cumulative distribution of gap length. Gap constant = 1.0, Block length = 8 bits
Figure 5.4. Cumulative distribution of gap length. Gap constant = 1.0, Block length = 4 bits
Figure 5.5. Cumulative distribution of gap length. Gap constant = 2.0, Block length = 8 bits
Figure 5.6. Cumulative distribution of gap length. Gap constant = 4.0, Block length = 4 bits.
Figure 5.7. Cumulative distribution of gap length. Gap constant = 4.0, Block length = 8 bits
Figure 5.8. Cumulative distribution of gap length. Gap constant = 8.0, Block length = 4 bits
Figure 5.9. Cumulative distribution of gap length. Gap constant = 8.0, Block length = 8 bits
Figure 5.10 Cumulative distribution of burst length. Gap constant = 1.0, Block length = 4 bits
Figure 5.11. Cumulative distribution of burst length. Gap constant = 1.0, Block length = 8 bits
Figure 5.12. Cumulative distribution of burst length. Gap constant = 2.0, Block length = 4 bits
Figure 5.13. Cumulative distribution of burst length. Gap constant = 2.0, Block length = 8 bits
Figure 5.14. Cumulative distribution of burst length. Gap constant = 4.0, Block length = 4 bits
Figure 5.15. Cumulative distribution of burst length. Gap constant = 4.0, Block length = 8 bits
Figure 5.16. Cumulative distribution of burst length. Gap constant = 8.0, Block length = 4 bits
Figure 5.17. Cumulative distribution of burst length. Gap constant = 8.0, Block length = 8 bits
Figure 5.18. Cumulative distribution of burst length. Gap constant = 16.0, Block length = 4 bits
Figure 5.19. Cumulative distribution of burst length. Gap constant = 16.0, Block length = 8 bits
CHAPTER 6

EXPERIMENTAL RESULTS AND COMPARISON WITH THEORETICAL CALCULATIONS

In the previous chapter the derivation and calculation of the error statistics of the PRESG, was presented. These statistics have been measured for the one, two and three state case using the Digital Channel Performance Evaluator. The measurements were made in each case with the block length set to 8 bits, as this was found to be a practically useful block length. The results of the measurement, as well as a comparison with the theoretical calculations is presented below.

6.1 One State Case

Fifteen minute tests were run with the PRESG in the single error state mode. In this mode the PRESG remains in one error state for the entire fifteen minutes. The states are A (BER = 0.125), B (BER = 0.00781), and C (BER = 0.00977). For each state, runs were taken with the gap parameter on the DCPE set at 1, 2 and 4.

Table 6.1 compares the theoretical and experimental BER, block error rate, and throughput for the single state tests. As can be seen the two exhibit very good agreement.

Figure 5.2 is a graph of the theoretical curves for the cumulative distributions of gap length for gap parameter = 1, block length = 8 bits, and single states A, B and C. The actual measured values are shown as dots on the graphs. Figure 5.11 is a graph of the theoretical curve for
the cumulative distribution of burst length for gap parameter = 1, 
block length = 8 bits and single states A, B and C. The actual measured 
values are shown as dots on the graphs. In all cases the theoretical 
and measured results are identical. Figures 5.5, 5.7, 5.13 and 5.15 
show similarly drawn curves for the gap and burst length distributions 
for the 3 single states for gap parameters of 2 and 4. As is readily 
evident from the curves, the theoretical and measured values are identical 
for gap length distributions in all cases. However, the measured 
burst distribution curves for gap constants 2 and 4 have the same 
shape as the theoretical curves but are shifted to the right. The 
theoretical calculations were independently checked by three researchers 
and were found to be correct. The calculations were also found to 
agree with exhaustive enumeration of all possible patterns for the 
lower values of the burst length m. Furthermore, it was discovered 
that the experimental results for the burst length distribution for 
gap parameters 2 and 4 were identical with the theoretically calculated 
results for gap parameters 3 and 5 respectively. Based on these 
findings, it was concluded that the theoretical calculations were 
correct and the experimental results were in error. This conclusion 
was also supported by the fact that all the other statistics calculated 
including the gap length distribution agree with experimental results 
suggesting that the PRESG was operating as desired. The nature of 
the disagreement led us to suspect that the error was caused by a 
programming problem in the DCPE. Careful checking of the DCPE revealed 
a programming error related to the gap parameter that would cause 
the experimental curves to be shifted to the right, as is the case.
The problem has since been corrected but unfortunately time constraints associated with the experimental measurements prevented the gathering of new data for verification. However, it seems certain that the disagreement between the theoretical calculations and the experimental results was caused by this programming error.

6.2 Two and Three State Case

The PRESG was first set up in the two state mode using the transition matrices shown in Figure 6.1 (a). In this mode the PRESG remains in one of two error states. Measurements of BER, gap and burst length distribution were taken for the 3 two state cases. The PRESG was then set up in the three state mode using the transition matrix shown in Figure 6.2 (b). Measurement of BER, gap and burst length distribution were again taken. In each case, fifteen minute tests were run for three different mean times in state. This was done by setting the step size (of the Markov process) at 1, 8 and 128 bit periods.

Figures 6.2 - 6.5 show the theoretically calculated values for the burst length distribution for the two and three state case for gap parameter = 1. These values, which are shown in dotted lines, were obtained from the single state test results using equation 5.20. The measured values of the burst length distribution for the three different mean times in state are shown in solid lines. It was indicated in the previous chapter that equation 5.20 is accurate only if the mean time is larger than the average length of an error burst. This can be seen to be true by observing the graphs. In all cases, as
the mean time is increased the theoretical and measured values are identical. Figures 6.6 - 6.9 show similarly plotted curves for the burst length distribution for gap parameter = 1. Again, as the mean time is increased, the theoretical and experimental curves can be seen to coincide.

Table 6.2 and 6.3 compare the theoretical and experimental BER for the two and three state case. The theoretical values, shown in the rightmost column were obtained from the single state test results using equation 5.19. As can be seen, the agreement between the theoretical and experimental values is very close. It is also evident that the measured BER remains almost unaltered as the mean time in state is increased which verifies the observations made in the previous chapter.
Table 6.1. Table of calculated and measured values for the one state case

<table>
<thead>
<tr>
<th>State</th>
<th>Gap Clock</th>
<th>Gap Parm</th>
<th>Measured BER</th>
<th>Block Error + Throughput</th>
<th>Calculated BER</th>
<th>Block Error + Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>.124</td>
<td>.648 5.56x10^{-8}</td>
<td>.125</td>
<td>.65639 3.78x10^{-8}</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
<td>.124</td>
<td>.903</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>4</td>
<td>.123</td>
<td>.660</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td>.0078</td>
<td>.0608 .366</td>
<td>.0078</td>
<td>.06081 .36643</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>2</td>
<td>.0078</td>
<td>.0704</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>4</td>
<td>.0068</td>
<td>.0913</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>.00098</td>
<td>.0078 .882</td>
<td>.00098</td>
<td>.00778 .88244</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>2</td>
<td>.00100</td>
<td>.0080</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>4</td>
<td>.00099</td>
<td>.0085</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
</tbody>
</table>
Table 6.2. Table of measured and calculated BERs for the state case

<table>
<thead>
<tr>
<th>STATE</th>
<th>CLOCK</th>
<th>GAP PARM</th>
<th>MEASURED BER</th>
<th>CALCULATED BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>1</td>
<td>1</td>
<td>.067</td>
<td>.066</td>
</tr>
<tr>
<td>AB</td>
<td>8</td>
<td>1</td>
<td>.066</td>
<td>&quot;</td>
</tr>
<tr>
<td>AB</td>
<td>128</td>
<td>1</td>
<td>.067</td>
<td>&quot;</td>
</tr>
<tr>
<td>AC</td>
<td>1</td>
<td>1</td>
<td>.052</td>
<td>.051</td>
</tr>
<tr>
<td>AC</td>
<td>8</td>
<td>1</td>
<td>.051</td>
<td>&quot;</td>
</tr>
<tr>
<td>AC</td>
<td>128</td>
<td>1</td>
<td>.052</td>
<td>&quot;</td>
</tr>
<tr>
<td>BC</td>
<td>1</td>
<td>1</td>
<td>.0074</td>
<td>.0075</td>
</tr>
<tr>
<td>BC</td>
<td>8</td>
<td>1</td>
<td>.0074</td>
<td>&quot;</td>
</tr>
<tr>
<td>BC</td>
<td>128</td>
<td>1</td>
<td>.0074</td>
<td>&quot;</td>
</tr>
</tbody>
</table>
Table 6.3. Table of measured and calculated values for the three state case

<table>
<thead>
<tr>
<th>STATE</th>
<th>CLOCK</th>
<th>GAP PARM</th>
<th>MEASURED BER</th>
<th>CALCULATED BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>1</td>
<td>1</td>
<td>.045</td>
<td>.047</td>
</tr>
<tr>
<td>ABC</td>
<td>8</td>
<td>1</td>
<td>.046</td>
<td>&quot;</td>
</tr>
<tr>
<td>ABC</td>
<td>128</td>
<td>1</td>
<td>.046</td>
<td>&quot;</td>
</tr>
</tbody>
</table>
### Figure 6.1

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.86</td>
<td>0.14</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0.06</td>
<td>0.94</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.87</td>
<td>0</td>
<td>0.13</td>
</tr>
<tr>
<td>B</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>C</td>
<td>0.04</td>
<td>0</td>
<td>0.96</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0.91</td>
<td>0.09</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0.04</td>
<td>0.96</td>
</tr>
</tbody>
</table>

![Transition Matrices](image)

Figure 6.1.  

a) Transition matrices used for obtaining statistics for the two state case.  

b) for the three state case.
Figure 6.2. Cumulative distribution of burst lengths for the two state case (AB). Gap constant = 1.0, Block length = 8 bits
Figure 6.3: Cumulative distribution of burst length for the two-state case (BC). Gap constant = 1.0, Block length = 8 bits.
Figure 6.4. Cumulative distribution of burst length for the two state case (AC). Gap constant = 1.0, Block length = 8 bits.
Figure 6.5. Cumulative distribution of burst length for the three state case (ABC). Gap constant = 1.0, Block length = 8 bits
Figure 6.6. Cumulative distribution of gap length for the two state case (AB). Gap constant = 1.0, Block length = 8 bits
Figure 6.7. Cumulative distribution of gap length for the two state case (BC). Gap constant = 1.0, Block length = bits
Figure 6.8. Cumulative distribution of gap length for the two state case (AC). Gap constant = 1.0, Block length = 8 bits
Figure 6.9. Cumulative distribution of gap length for the three state case (ABC). Gap constant = 1.0, Block length = 8 bits
CHAPTER 7

SUMMARY AND CONCLUSIONS

In Chapter 1 we described how 'bench' type testing of a digital communication system can be performed by replacing the channel by a noise source simulator and measuring certain test statistics at particular points in the system. In Chapter 2 the definitions of all the test statistics as well as the justification for their use was given. This was followed by a general discussion of channel models. It was noted that the binary symmetric channel is a very poor characterization of channels with memory. A more accurate model for channels with memory was presented. This is the so called Gilbert model which considers the channel to be composed of a small number of BSCs, where the state of the channel is controlled by a Markov process.

In Chapter 3 we studied the properties of shift register sequences and illustrated how they could be used for error generation. The operational principles of the Pseudo Random Error Sequence Generator (PRESG) was described. This included details of the method used to achieve the three different bit error rates and to implement the state transitions. In Chapter 4 the architecture and capabilities of the Digital Channel Performance Evaluator (DCPE) - the instrument used for measuring the statistics of the PRESG was given.

In Chapter 5, formulae used to theoretically compute the
statistics of the PRESG were derived. Expressions were obtained for
the gap and burst length distribution, block and character error rate
and throughput for the single state case. We also indicated how the
statistics for the two and three state case could be obtained from
the single state results. It was pointed out that in order to calculate
the gap and burst length distributions it is necessary to distinguish
between two cases – the case when the mean time in each state is larger
than the average length of an error burst and the case when it is not.
In the case when the mean time is much larger, the calculations are
fairly straightforward. The calculations are relatively more difficult
in the other case. The above mentioned statistics – gap and burst
length distribution, block and character error rate and throughput
were calculated for different values of gap parameters and block length
for the single state case. The results of these calculations were
presented. Plots were made of gap and burst length distribution in
each case to facilitate comparison with the experimental results.

The comparison between the theoretical calculations and the
experimental results was performed in Chapter 6. The error statistics
were measured for the three single states for gap parameters 1, 2
and 4 and block length = 8 bits. The measured values were found to
agree extremely well with the theoretical calculations except in the
case of the burst length distribution for gap parameters 2 and 4.
The measured curves in this case were found to have the same shape
as the theoretical curves but were shifted to the right. After consider-
able investigation, recalculation and rechecking of data, it was concluded
that the theoretical calculations were in fact correct and the experimental
results were in error. It was suspected that the discrepancy was caused by a programming problem in the DCPE. The DCPE was checked by other researchers and was found to contain a program error that could indeed account for the discrepancy. The gap and burst length distribution for the 2 and 3 state case were calculated theoretically from the single state results. Calculations were made assuming that the mean time in each state is larger than the average length of a burst (or gap). As mentioned earlier, the calculations in this case are fairly simple. The experimental measurements of gap and burst length distribution were taken for the two and three state case for three different mean times in state. It was found in all cases that as the mean time was increased, the theoretical and experimental curves were identical. The BER for the 2 and 3 state case was similarly calculated from the single state results and compared with the experimental BER for three different mean times in state. Again, it was found in all cases that the experimental results were in excellent agreement with the theoretical predictions. It was also found that the experimental BER remains unaltered as the mean time is increased. This substantiates the postulation made in Chapter 5 that the mean time in state has an appreciable effect on the gap and burst length distributions only.

To sum up, the statistics of the PRESG which were obtained theoretically were found to be in very close agreement with statistics measured experimentally using the DCPE. On the basis of the close agreement obtained it is concluded that the PRESG is operating as
desired and that it should prove to be a useful instrument for evaluating the performance of a digital communication system.
APPENDIX A

PROGRAM FOR COMPUTING GAP LENGTH DISTRIBUTION.

PROGRAM GAPLEN (INPUT, OUTPUT, TAPE=INPUT, TAPE=OUTPUT)
C FORTRAN PROGRAM FOR CALCULATING CUMULATIVE DISTRIBUTION OF GAP LENGTH
DIMENSION PGAP1(50), PGAP2(50), LEN(50)
INTEGER G, BL1, BL2
READ (5, 10) PBIT
READ (5, 20) G, BL1, BL2
WRITE (6, 10) PBIT
WRITE (6, 20) G
10 FORMAT (F9.7)
20 FORMAT (3I3)
PGAP1 = (1.-PBIT)**BL1
PGAP2 = (1.-PBIT)**BL2
DO 30 M = 1, 50
LEN(M) = M
IF (M .LT. G) GOTO 60
PGAP1(M) = 1. - (PGAP1**(M-G+1))
PGAP2(M) = 1. - (PGAP2**(M-G+1))
GOTO 30
60 PGAP1(M) = 0.0E+00
PGAP2(M) = 0.0E+00
30 CONTINUE
WRITE (6, 40) (PGAP1(I), PGAP2(I), LEN(I), I = 1, 50)
40 FORMAT (3E17.10, 5X, I4/)
50 FORMAT (13)
STOP
END
APPENDIX B

PROGRAM FOR COMPUTING BURST LENGTH DISTRIBUTION.
PROGRAM BURLEN (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)
C FORTRAN PROGRAM FOR CALCULATING CUMULATIVE DISTRIBUTION
C OF BURST LENGTH

INTEGER G, BL1, BL2
READ (5,10) G, BL1, BL2
READ (5,20) PB
WRITE (6,20) PB
WRITE (6,50) G

10 FORMAT (3I3)
20 FORMAT (F9.7)
50 FORMAT (I3)

Q1 = (1.0-PB)**BL1
Q2 = (1.0-PB)**BL2
PB1 = 1.0 - Q1
PB2 = 1.0 - Q2
PBUR1(1) = PB1**G
PBUR2(1) = PB2**G
DO 60 M=2,100
NO = M/2
SUM1 = 0.
SUM2 = 0.
IF (2*NO.EQ.M) GOTO 30
NUM = (M-1)/G
MAXZ = M-2-NUM
IF (NUM**G.EQ.M-1) MAXZ = M-1-NUM
GOTO 40
30 MAXZ = M-2-((M-2)/G)
40 MPART = M-(MAXZ+1)

MAX = -1
DO 70 K=MPART,MAX
N=M-(K+1)
LU=(N/6)+1
PROD=1.
DO 60 J=1,LU
MR=G*(J-1)
PROD=PROD*FACT(K,J-1)*FACT(N+K-MR-1,N-MR)*((-1)**(J-1))
60 CONTINUE
SUM1 = SUM1 + PROD*(Q1**N)*(PB1**(K+1))
SUM2 = SUM2 + PROD*(Q2**N)*(PB2**(K+1))
70 CONTINUE
PBUR1(M) = PBUR1(M-1) + SUM1*(Q1**G)/PB1
PBUR2(M) = PBUR2(M-1) + SUM2*(Q2**G)/PB2
LEN(M) = M
80 CONTINUE
WRITE (6,90) (PBUR1(I),PBUR2(I),LEN(I),I=1,100)
90 FORMAT (2E17.10,5X,14)
STOP
END

FUNCTION FACT(N,K)
C THIS PROGRAM CALCULATES BINOMIAL COEFFICIENTS
FACT=1.0
IF (K.EQ.N) GOTO 130
IF (K.EQ.0) GOTO 130
IF (N.EQ.0) GOTO 130
M=N-K
IF (K.LE.M) GOTO 110
DO 100 T=1,M
FACT=FACT*FLOAT(N-I+1)/FLOAT(M-I+1)
100 CONTINUE
GOTO 130
110 DO 120 J=1,K
FACT=FACT*FLOAT(N-J+1)/FLOAT(K-J+1)
120 CONTINUE
RETURN
END
REFERENCES


