AFRL-SN-WP-TR-1999-1115

ELECTRONIC WARFARE (EW) RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval 2)

DR. W. THOMAS BASS

MERCER ENGINEERING RESEARCH CENTER
A UNIT OF MERCER UNIVERSITY
135 OSIGIAN BOULEVARD
WARNER ROBINS, GA 31088

JULY 1999

FINAL REPORT FOR 12/02/1997 – 03/31/1999

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-7318

NOTICE

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Nicholas A. Pequignot

NICHOLAS A. PEQUIGNOT
Technical Advisor
Applied Aperture & Receiver Branch
RF Sensor Technology Division
Sensors Directorate

Anthony W White

ANTHONY W. WHITE
Chief
Applied Aperture & Receiver Branch
RF Sensor Technology Division
Sensors Directorate

EMIL R. MARTINSEK
Chief
RF Sensor Technology Division
Sensors Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
JULY 1999

3. REPORT TYPE AND DATES COVERED
FINAL REPORT FOR 12/02/1997 - 03/31/1999

4. TITLE AND SUBTITLE
ELECTRONIC WARFARE (EW) RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval 2)

5. FUNDING NUMBERS
C F09603-93-G-0012
PE 62204
PR 7633
TA 11
WU AM

6. AUTHOR(S)
DR. W. THOMAS BASS

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MERCER ENGINEERING RESEARCH CENTER
A UNIT OF MERCER UNIVERSITY
135 OSIGIAN BOULEVARD
WARNER ROBINS, GA 31088

8. PERFORMING ORGANIZATION REPORT NUMBER
MR990426.01-RV-

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB, OH 45433-7318
POC: NICHOLAS PEQUIGNOT, AFRL/SNRP, 937-255-6127 EXT. 4235

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-SN-WP-TR-1999-1115

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

EW Receiver and Processing Concepts Evaluation Program (RAPCEval 2) tasks have provided analytical support for current research at the ESM group at Air Force Research Laboratory. Tasks initiated under OPTION 1 of the contract of this program provided analysis for inputs and countermeasures for electronic receivers of radar, electro-optic, infrared, and ultraviolet systems. Research has been performed under the direction of the Joint Program Research Standards Committee, composed of members from Wright-Patterson AFB, Warner Robins AFB, Mercer University, and Mercer Engineering Research Center. The report includes research presentations and reports from graduate students. Topics included are represented in the keyword list of this form.

14. SUBJECT TERMS
RAPCEval, imaging, RAD, Reed-Solomon codes, parallel processing, signal processing

15. NUMBER OF PAGES
340

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
SAR

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

This report describes work accomplished under contract option 1 of the EW Receiver and Processing Concepts Evaluation Program (RAPCEval 2). The Air Force Research Laboratory (AFRL) awarded this contract. This branch of AFRL is located at Wright Patterson Air Force Base, in Dayton, Ohio. The Option 1 task of this program was awarded to Mercer Engineering Research Center (MERC) on October 15, 1997. Work was completed on the RAPCEval 1 contract in October, 1996, and this contract was initiated in November 1996. This report describes work accomplished on the contract from December 2, 1997 through March 31, 1999. The project includes five options and has maximum duration of 60 months from the date of award. In support of the contract, a number of activities and projects have been initiated. Some of these are complete, and many more are continuing. A number of activities have spawned new directions for investigative effort.

The Program Research Standards Committee (PRSComm), established at the outset of the Basic task, has continued to meet regularly and has provided valuable guidance and suggestions both as to the direction of students involved in the effort, and the scope and emphasis of the research efforts in general. Current membership of this committee represents the Air Force Research Laboratory, Robins Air Force Base, Mercer University, and Mercer Engineering Research Center. The specific members are listed in another section of this report (section 4, Research Support).

Three meetings of the PRSComm were held during contract Option 1. Eight student research reviews were presented and were approved by the PRSComm in the course of these meetings (see section 5, Project Activity). Valuable discussion and suggestions were provided for directing and focusing the students' work. In addition, suggestions were received for topics' reduction in scope or better focus. Frequently, unanticipated or unknown resources and techniques were pointed out for the benefit of the students. One meeting involved a presentation by AFRL researcher James Stephens, who delivered a visiting lecture with an emphasis on communication countermeasures, with focus on advanced digital signal processing for electronic warfare.

Students engaged in the RAPCEval program have continually enjoyed fruitful contact with knowledgeable personnel at the Air Force Research Laboratory in their respective areas of interest. They have also interacted with experienced colleagues at Robins Air Force Base, employees of MERC, Mercer University School of Engineering faculty, and various representatives of industry.

It should be noted that software generated under this contract is not government-owned.

The RAPCEval contract has stimulated gratifying communication and collaborative research effort among students, university faculty, MERC personnel, personnel at the Air Force Research Laboratory and Warner Robins AFB and industry. All parties have expressed satisfaction with the contract results.

2. INTRODUCTION

These tasks specify requirements for analytical and research support of in-house research at the Air Force Research Laboratory, AFRL/SNR. There is increasing sophistication, quantity, and mobility of hostile radars, AAM, SAM, and AAA fire control systems. Electronic warfare receivers for radar, electro-optic, infrared, ultraviolet missile warning and electronic countermeasures need operational upgrades to allow penetrating aircraft acceptable survivability. This encourages maintenance of in-house labs to support development, to evaluate concepts, and to test new receivers, processors, and software.

2.1 EW Receiver Effort

Complex EW environments have caused employment of numerous receiving systems. Augmentation of in-house capability for evaluation, novel concept development, and exploitation of new technology is needed. Computer-aided simulation of new systems and concepts can save resources. New high speed analog-to-digital (A/D) converter technology may allow input frequencies to be digitized in baseband before the crystal video detector, possibly allowing real time digitized frequency, pulse width, and pulse amplitude. Advancing materials technology for infrared (IR) / ultraviolet (UV) / radio frequency (RF) energy offers the possibility of augmented and combined sensors. Investigation of these materials is needed to reduce the kind and number of avionics needed in combat. Filtering and discrimination advances, both in hardware and software, may allow enhancement of fielded EW systems.

2.2 EW Processing Effort

A modern EW system must face an increasing number of hostile threats that are multimode. Sensors to intercept such threats now include radar warning and electronic intelligence systems. Information from these sensors must be processed, the threat identified, and appropriate countermeasures initiated to counter these threats. An augmentation of in-house capability is required to evaluate processor hardware and software, to exploit novel ideas, and to investigate advanced concepts such as artificial intelligence to determine the nature of the threat, and what countermeasures, if any, to employ.

2.3 EW Exciter Effort

Digital exciters are being developed to provide a flexible active ECM asset against a wide variety of modern threats. The need exists to evaluate the various exciter architectures, advance and develop unique concepts, and advance the digital exciter technology base. The novel concepts and technologies must be evaluated for effectiveness against the proposed application.

2.4 EW Antenna Effort

The role of antennas as the "eyes and ears" of the sensor suites continues to make RF antenna technology development vital to the Air Force mission. Airborne antenna apertures of the future will be low cost, broadband, low radar cross section (RCS), and multifunction in nature to earn their way onto platforms where space is at a premium.

3. SCOPE

The overall program consists of a basic task and four options that are the conglomerate of different work efforts and technologies within the Electronic Warfare arena. Detailed descriptions are given as follows:

- Basic Task - The basic task will provide the tasks necessary to analyze software and hardware approaches to perform the exploratory development of EW technology in these technology areas: radar hardware, laser hardware, infrared hardware, and ultraviolet hardware. The task will analyze receiver and exciter technology to generate ECM signals to improve ECM system performance. In addition, the scope of the basic task will include signal processing technology related to the hardware.

- Option 1 - These tasks will be those necessary to analyze receiver technology for application to modern digital spectrum estimation techniques in order to improve EW / SIGINT / ELINT / IR / EO receiver performance.

- Option 2 - This option consists of those tasks necessary to identify high risk design areas for an EW / SIGINT / ELINT / IR / EO hardware approach, to perform exploratory design assessments for selected functions, and to determine the degree of parallel processing achievable.

- Option 3 – This option is "reserved."

- Option 4 - These tasks are those essential to EW / SIGINT / ELINT / IR / EO hardware and signal processing including but not limited to pulse-deinterleaving, parametric extraction, and threat identification.

4. RESEARCH SUPPORT

For support of the overall contract, a "Program Research Standards Committee" has been established. Membership for this committee was most recently updated March, 1997. Current members are:

- from the Air Force Research Laboratory in Dayton,
    - Mr. Nicholas Pequignot (the program manager for AFRL)
    - Mr. Emil R. Martinsek
    - Mr. Norman A. Toto
    - Dr. Duane A. Warner
    - Mr. Paul J. Westcott

- from Warner Robins Air Logistics Center,
    - Mr. Steve Strawn (the program manager for WR-ALC)
    - Mr. John LaVecchia
    - Mr. Phil Oliver
    - Mr. Ches Rehburg
    - Mr. Larry Sheets

- from Mercer University and MERC,
    - Dr. Tom Bass (the program manager for MERC)
    - Dr. David Barwick, (chairman of the standards committee)
    - Dr. Aaron Collins (Mercer U)
    - Dr. Behnam Kamali (Mercer U)
    - Dr. Paul MacNeil (Mercer U)

The EW Receiver and Processing Concepts Evaluation Program was awarded to the Mercer Engineering Research Center (MERC) by WPAFB/WL under contract F09603-93-G-0012-0017. This contract is administered through WR-ALC. The overall program has a funding ceiling of $499,940. Incremental funding will be accomplished via a series of contract options. The basic contract is $99,998.00, option (1) is $99,998.00, option (2) is $99,998.00, option (3) is “reserved”, and option (4) is $50,000.00.

Contracts have been awarded for the Basic program, Option 1, Option 2, and Option 4.

5. PROJECT ACTIVITY

5.1 Steering Committee, April 1998

5.1.1 Meeting Minutes

EW Receiver and Processing Concepts Evaluation Program
Program Research Standards Committee Meeting
Minutes 2 April 1998

A meeting of the Program Research Standards Committee (PRSC) for the EW Receiver and Processing Concepts Evaluation Program (RAPCEval2) was hosted by Mercer Engineering Research Center at 8:30 a.m. in the new MERC auditorium. Committee members present were Dave Barwick, Tom Bass, Aaron Collins, Benham Kamali, Nick Pequignot, Norman Toto, Duane Warner, Paul MacNeil, Phil Oliver, and Ches Rehberg. Also present were several representatives from Mercer University, students scheduled to speak, and other students considering participation in the RAPCEval program. Five student presentations were given. Four were reports on research in progress, while one was a research proposal.

After welcoming remarks by Dr. Barwick, Dr. Tom Bass gave a brief description of the RAPCEval program and introduced the students who were scheduled to speak. The students and their topics are listed in the table.

Ron Brinkley	"Burst Error Correction with Reed-Solomon Codes"
Dennis Ludwig	"Identification of IR Images Using Neural Networks"
Henderson Benjamin	"Selection of Reed Solomon Codes Using Neural Networks"
Randy Ford	"Passive Location via Evolutionary Genetic Algorithms"
Dennis Ludwig	"Identification of IR Images Using Neural Networks"
Tracy Tillman	"Hardware Exploitation of RAD Filtration"

Ron Brinkley showed overheads introducing the concept of Reed-Soloman error correction methods, pointing out that RS codes are block codes. He also gave a summary of the types of communication errors, together with interleaving techniques designed to combat the errors. Ron discussed his analysis and tabulation of the Reed-Soloman error-correcting codes for specific classes and burst error categories. Announcement was made of Ron's presentation on this subject to the IEEE Pacific Rim Conference (PACRIM '97).

Henderson Benjamin reviewed his topic on the application of neural networks to the selection of Reed-Solomon (RS) codes. Henderson showed many of the details of the manner in which RS codes are selected by a neural net (NN). Included were training techniques for the NN and performance statistics for the trained network. Henderson acknowledged dependence on the work of Ron Brinkley for availability of code selection tables that were required for the NN application. This work is nearly complete and the thesis is under final review. An abbreviated version of the thesis is available in the RAPCEval annual report, dated 3 February 1998.

Dennis Ludwig reviewed the pose estimation problem for an object identified by an infrared (or other) image. He had planned to extend a two-dimensional solution developed at the Wright Laboratory to a general solution in three dimensions. The chief difficulty encountered is the necessity of taking available two-dimensional images, and somehow recreating the third dimension. A number of questions were evoked by this presentation, ranging from whether three-dimensional data is available, to the specifics of how the images are brought into alignment.

Tracy Tillman presented additional details on the development of a unique method of table generation and table lookup for use in RAD parameter production. He has discovered some limitations for the technique, and plans to finish a report on his discoveries, calculations, and algorithmic development in the next quarter or two.

Dr. Barwick chaired the committee discussions that followed the last presentation. Suggestions were gathered regarding the student presentations. The group agreed that the communication research projects directed by Dr. Kamali were very pertinent to current AF needs and should be continued. Several sensed that Dennis Ludwig seemed to have lost focus of the original goals, probably due to the departure of his thesis advisor recently. Dr. Bass, in conjunction with Dr. Collins, will be giving increased direction and steer him back on track. A question arose as to whether the RAD lookup tables were in use in the PLAID program. Currently, they are not, but further evaluation is planned during the next phase of R&D. The comment was made that Randy Ford's schedule seemed optimistic. Another question concerned the potential to test the differential evolution solution technique on actual PLAID data. This will be investigated.

During final discussions, Norm Toto said that topics formerly being investigated at AFIT were "going away." He thought MERC might be able to continue some of these via RAPCEval. Norm also suggested potential RAPCEval research investigations into GPS anti-jamming techniques and related technologies.

At this point, the meeting was adjourned.

5.1.2 Meeting Agenda

Agenda

RAPCEval program

STEERING COMMITTEE MEETING

98 April 2 --- 8:30 AM to 2:30 PM
Main Auditorium, Mercer Engineering Research Center
135 Osigian Boulevard, Warner Robins, GA

Meeting called by:	Nicholas Pequignot, AFRL/SNRP, Air Force Research Laboratory Program Manager	Facilitator: Dr. Tom Bass

Committee Members:	**AF Rsch Laboratory**	**Robins AFB**	**Mercer University**
	Mr Nicholas Pequignot	Mr Steve Strawn	Dr David Barwick
	Mr Emil R Martinsek	Mr John LaVecchia	Dr Tom Bass
	Mr Norman A Toto	Mr Phil Oliver	Dr Aaron Collins
	Dr Duane A Warner	Mr Ches Rehberg	Dr Benham Kamali
	Mr Paul J Westcott	Mr. Larry Sheets	Dr Paul MacNeil
Email addresses:	pequigna@sensors.wpafb.af.mil	lnerr@ewir-wr.robins.af.mil	dbarwick@merc.mercer.edu
	martinser@sensors.wpafb.af.mil	lavecchia@ewir-wr.robins.af.mil	tbass@merc.mercer.edu
	totona@sensors.wpafb.af.mil	oliver@ewir-wr.robins.af.mil	collins_as@mercer.edu
	warnerda@sensors.wpafb.af.mil	rehberg@ewir-wr.robins.af.mil	macneil_pe@mercer.peachnet.edu
	westcopj@sensors.wpafb.af.mil	sheets@ewir-wr.robins.af.mil	kamali_b@mercer.peachnet.edu

Meeting Agenda, continued…

Schedule

Continental Breakfast / Greetings	Dr David Barwick	8:30 AM - 9:00 AM
Meeting Overview	Dr Tom Bass	9:00 AM - 9:20 AM
Student Proposals	Dr Tom Bass / Students	
Burst Error Correction with Reed-Solomon Codes	Ron Brinkley	9:20 AM - 9:55 AM
Selection of Reed Solomon Codes Using Neural Networks	Henderson Benjamin	9:55 AM - 10:30 AM
Break		10:30 AM - 10:50 AM
Student Proposals (continued)	Dr Tom Bass / Students	
Passive Location via Evolutionary Genetic Algorithms	Randy Ford	10:50 AM - 11:25 AM
Identification of IR Images Using Neural Networks	Mr Dennis Ludwig	11:25 AM - NOON
Lunch		NOON - 1:00 PM
Student Proposals (continued)	Dr Tom Bass / Students	
Hardware Exploitation of RAD Filtration	Mr. Tracy Tillman	1:00 PM - 1:35 PM
Discussions & New Business	Dr. Tom Bass	1:35 PM - 2:30 PM
Adjourn		2:30 PM

5.1.3 Attendance Roster

The attendees at this meeting are listed here:

RAPCEVAL MEETING - Attendance Roster - April, 2, 1998

NAME	WORKPLACE	PHONE #	E-MAIL ADDRESS
Barnett, Kevin	Mercer University	(912) 752-2112	barnett_kd@mercer.edu
Barwick, Dave	MERC	(912) 953-6800	dbarwick@merc.mercer.edu
Bass, Tom	MERC	(912) 953-6800	tbass@merc.mercer.edu
Benjamin,	WR-ALC/LKSE	(912) 327-2864	benjaminh@jslks.robins.af.mil
Brinkley, Ron	NAWC/AD	(301) 342-6416	brinkleyrl%AM7@MR.NAWCAD.NAVY.M
Collins, Aaron	Mercer University	(912) 752-2097	collins@mercer.edu
Ford, Randy	WR-ALC/LYSBL	(912) 926-0423	fordra@cssa.robins.af.mil
Holmes, Kerwin	WR-ALC/LYSKS	(912) 327-2880	holmeskr@jslks.robins.af.mil
Hughes, Theresa	WR-ALC/LFESR	(912) 922-7862	
Kamali, Behnam	Mercer University	(912) 752-2415	kamali_b@mercer.edu
Ludwig, Dennis	WR-ALC/LNERB	(912) 926-5453	d.ludwig@ieee.org
MacNeil, Paul	Mercer University	(912) 752-2185	macneil_pe@mercer.edu
Oliver, (Phil) R.P.	WR-ALC/LNERT	(912) 926-2588	oliver@ec.robins.af.mil
Pequignot, Nick	AFRL/SNRP	(937) 255-6127	pequigna@sensors.wpafb.af.mil
Rehberg, Ches	WR-ALC/LNEX	(912) 926-4525	rehberg@ec.robins.af.mil
Shaw, Rudy	MERC (North)	(937) 256-3708	r.shaw@worldnet.att.net
Stinehelfer, Wes	WR-ALC/LYKS	(912) 327-2911	stinehelferc@jslks.robins.af.mil
Taylor, Bill	MERC AFRL (North)	(937) 255-4174	taylorwr@sensors.waafb.af.mil
Tehan, Jack	MERC (North)	(803) 795-7030	jtehan@aol.com
Toto, Norman A.	AFRL/SNRW	(937) 255-4933	toto@sensors.wpafb.af.mil
Warner, Duane	AFRL/SUJO	(937) 255-4174	warnerda@aa.wpafb.af.mil

5.1.4 Overview of the Program (Dr. Bass)

The Overview Briefing of the RAPCEval Program as presented at this meeting is reproduced on the next 7 pages.

RAPCEval STEERING COMMITTEE MEETING

April 2, 1998

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

April
1998

PROJECT INFORMATION

- Contract: F09603-93-G-0012-0017
- Customer: Wright Laboratory
- Contract Value: $299,964
- Program Status:
    - Graduate Research Jointly Supported by Mercer, Wright Lab, WR-ALC, and Industry
    - Successful research projects (9) have been completed; several (5) research projects are ongoing
    - All research has been determined by the committee to be useful to the Air Force, as well as having merit for graduate research

PROGRAM RESEARCH STANDARDS COMMITTEE

- ★ *MERC:* Dr. David Barwick (Chairman), Dr. Tom Bass (PMgr)
- ★ *AIR FORCE LABORATORY:* Mr. Nick Pequignot (PMgr), Mr. Emil R. Martinsek, Mr. Norman A. Toto, Dr. Duane A. Warner, Mr. Paul J. Westcott
- ★ *WR-ALC:* Mr. Steve Strawn (Pmgr), Mr. John LaVecchia, Mr. Phil Oliver, Mr. Ches Rehburg, Mr. Larry Sheets
- ★ *MERCER UNIVERSITY:* Dr. Aaron Collins, Dr. Benham Kamali, Dr. Paul MacNeil

RESEARCH IN PROGRESS

- ★ *Henderson Benjamin,* "Selection of Reed Solomon Codes Using Neural Networks"
- ★ *Ron Brinkley,* "Burst Error Correction with Reed-Solomon Codes"
- ★ *Randy Ford,* "Passive Location via Evolutionary Genetic Algorithms"
- ★ *Dennis Ludwig*, "Identification of IR Images Using Neural Networks"
- ★ *Tracy Tillman,* "Hardware Exploitation of RAD Filtration"

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

April
1998

GRADUATES

- ★ *Mark Astin,* "Application of Parallel Computing Techniques to the RAD Algorithm", classified, publication in process
- ★ *Mark Campbell,* "Auto-Regressive Spectral Analysis - EW Applications" , unclassified report available from MERC
- ★ *Claus Franzkowiak,* "Four-Pulse Primary RAD Filter Development", classified, publication in process
- ★ *Neal Garner*, "Error Correction and Prediction for Improved Communication of Time and Time Measurements", unclassified report available from MERC

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

April
1998

GRADUATES

- *Joseph Kelley* , "A Parameter Determination Alternative for RAD Analysis", WL-TR-95-1005
- *Joseph Kelley,* "MultiGroup Simultaneous RAD Parameter Selection", WL-TR-97-1094 *(final edit)*
- *Max Roesel*, "Agile RF/PRI Radar Analysis via RAD", WL-TR-95-1020
- *Dave Schuler,* "Comparison of Algorithms for Geolocation of Radar Signals", available from MERC on need to know
- *Kirk Wright*, "Object Oriented Modeling of the AN/ALQ-172", publication in process

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

April 1998

TODAY'S STUDENT PRESENTATIONS

- ★ *Ron Brinkley,* "Burst Error Correction with Reed-Solomon Codes" *Dennis Ludwig,* "Identification of IR Images Using Neural Networks"
- ★ *Henderson Benjamin,* "Selection of Reed Solomon Codes Using Neural Networks"
- ★ *Randy Ford,* "Passive Location via Evolutionary Genetic Algorithms"
- ★ *Dennis Ludwig,* "Identification of IR Images Using Neural Networks
- ★ *Tracy Tillman,* "Hardware Exploitation of RAD Filtration

5.1.5 Presentation By Ron Brinkley

The student briefing presented by Ron Brinkley at this meeting is reproduced on the next 32 pages.

RESEARCH FINAL BRIEFING / THESIS DEFENSE

R. Brinkley
NAWC-AD Pax River, MD
Electrical Engineer

Date Approved: February 1996

Completion Date: April 1998

Research Topic: *BURST ERROR CORRECTION WITH REED-SOLOMON CODES*

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Ronald L. Brinkley
MSEE Program
Final Briefing

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

INTRODUCTION

- **Digital Communication System**
- **What are Reed-Solomon Codes**
- **Block Code Format**
- **Types of Communication Errors**
- **Types of Interleaving Techniques**

BURST ERROR CORRECTION CAPABILITIES

BURST ERROR CORRECTION EFFICIENCY

INTERLEAVED BURST ERROR CORRECTION

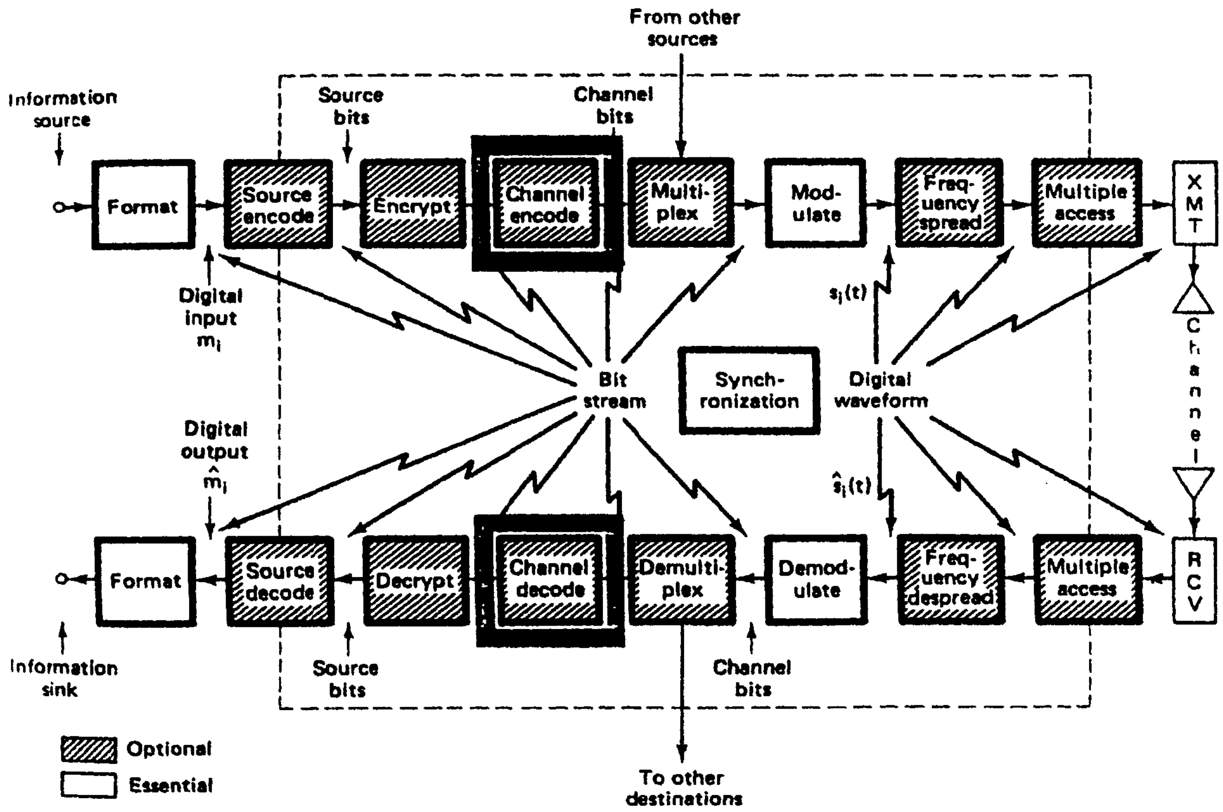
EXAMPLE PROBLEM

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

ORIGINAL PROBLEM STATEMENT

The purpose of this thesis is to introduce and explore the burst error correcting capabilities of various Reed-Solomon codes. Also, the parameter titled burst error correction efficiency will be introduced. Burst error correction efficiency is a measured value that will allow a code designer additional information for evaluating possible Reed-Solomon codes. The concepts of burst error correcting capability and burst error correction efficiency will be expanded to include block and convolutional interleaving techniques.

BURST ERROR CORRECTION WITH REED-SOLOMON CODES



[Skl] B. Sklar, Digital Communication, Fundamentals and Applications, Englewood Cliffs, NJ: Prentice Hall, 1988.

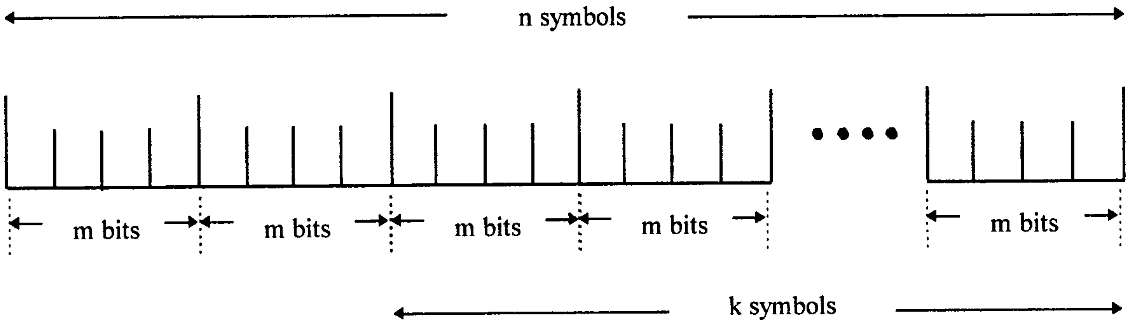
BURST ERROR CORRECTION WITH REED-SOLOMON CODES

WHAT ARE REED-SOLOMON CODES?

- Linear Block Codes
- Non-Binary
- Error Detection/Error Correction
- Random Errors/Burst Errors
- Maximum Distance Separable (MDS) Codes

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

Word Oriented Block Code Format



(n,k) Defines the Reed - Solomon code
n - k Number of parity symbols
t = n - k / 2 Random error correcting capability

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

TYPES OF ERRORS

Random Error
- Individual Symbol Errors Occurring In No Particular Manner

Burst Error
- Cluster of Individual Errors
- Channel Memory
- Long Single Burst
- Multiple Short Burst

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

TWO POPULAR INTERLEAVING TECHNIQUES

Block Interleaving
- Most Widely Used
- Easy to Implement

Convolutional Interleaving
- More Efficient than Block Interleaving
- CIRC

BLOCK INTERLEAVER

Interleaver

Input Stream: A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, B1, B2, B3, B4, B5, B6, ...F9, F10, F11, F12

← n symbols →

d

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	F2	F3	F4	F5	F6	F6	F8	F9	F10	F11	F12

Output Stream: A1,B1, C1, D1, E1, F1, A2, B2, C2, D2, E2, F2, A3, B3, C3, D3, E3, F3, A4, ..., C12, D12, E12, F12

← burst →

De-interleaver

Input Stream: A1,B1, C1, D1, E1, F1, A2, B2, C2, D2, E2, F2, A3, B3, C3, D3, E3, F3, A4, ..., C12, D12, E12, F12

← n symbols →

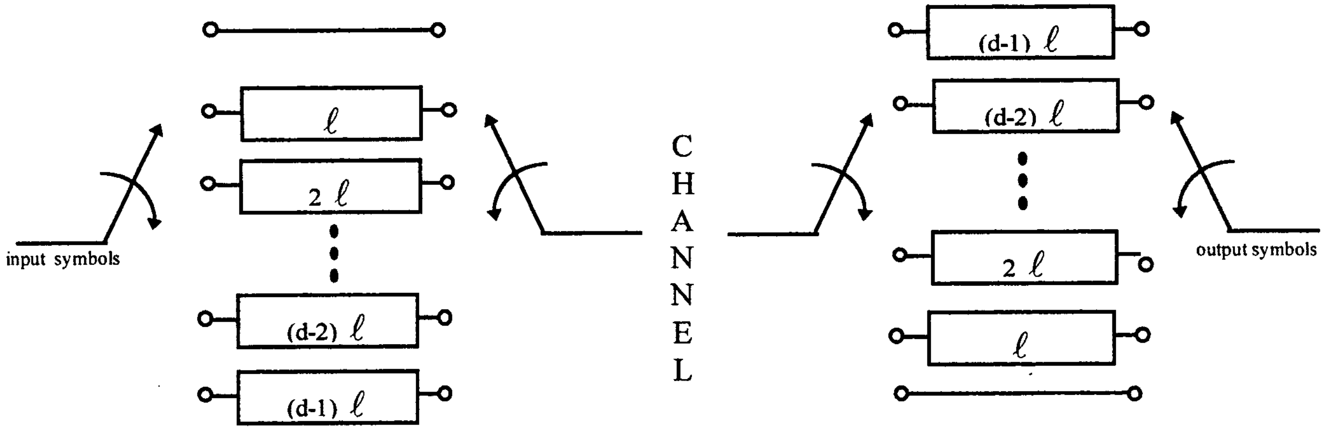
d

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	F2	F3	F4	F5	F6	F6	F8	F9	F10	F11	F12

Output Stream: A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, B1, B2, B3, B4, B5, B6, ...F9, F10, F11, F12

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

CONVOLUTIONAL INTERLEAVER



[For1] G. D. Forney, Jr., "Burst Correcting Codes for the Classic Bursty Channel," IEEE Transactions on Communication, Volume COM-19, pp. 772-781, October 1971.

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

INTRODUCTION

BURST ERROR CORRECTION CAPABILITIES

- **Random Error Correction Capability**
- **Maximum Single Burst Error Correction Capability**
- **Maximum Multiple Burst Error Correction Capability**
- **Guaranteed Single Burst Error Correction Capability**
- **Guaranteed Multiple Burst Error Correction Capability**

BURST ERROR CORRECTION EFFICIENCY

INTERLEAVED BURST ERROR CORRECTION

EXAMPLE PROBLEM

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

RANDOM ERROR CORRECTING CAPABILITY

t symbols

MAXIMUM SINGLE BURST ERROR CORRECTING CAPABILITY

t m bits

MAXIMUM MULTIPLE BURST ERROR CORRECTING CAPABILITY

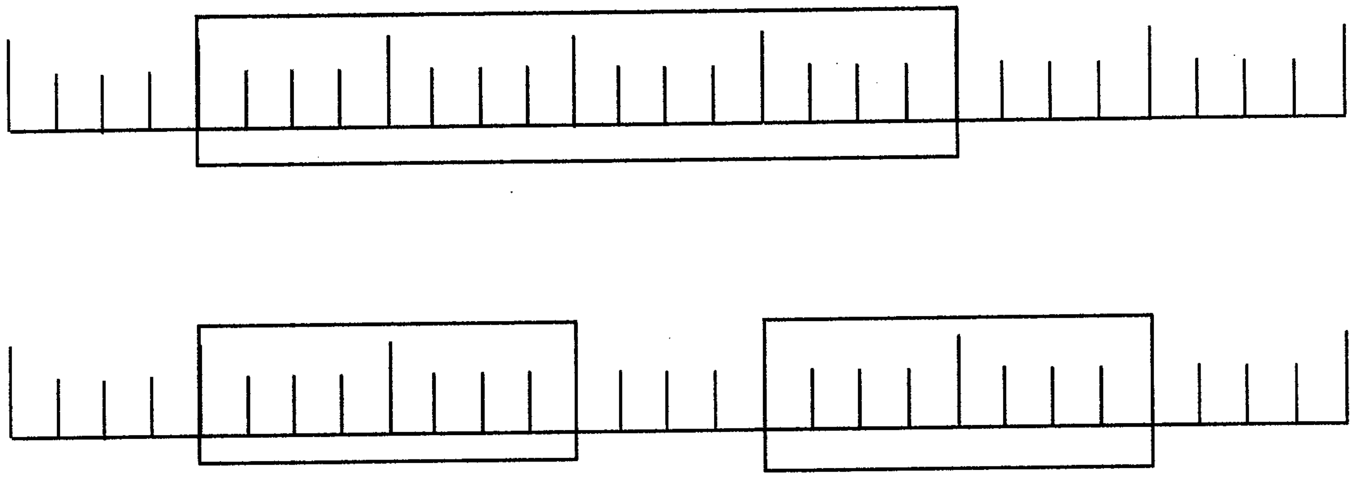
$$b_{2\,max} = \left\lfloor \frac{t}{2} \right\rfloor m$$

$$b_{3\,max} = \left\lfloor \frac{t}{3} \right\rfloor m$$

$$b_{s\,max} = \left\lfloor \frac{t}{s} \right\rfloor m$$

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

MAXIMUM SINGLE AND MULTIPLE BURST ERRORS (t = 4)



BURST ERROR CORRECTION WITH REED-SOLOMON CODES

GUARANTEED SINGLE BURST ERROR CORRECTING CAPABILITY

$$b_1 = m(t-1)+1$$

GUARANTEED MULTIPLE BURST ERROR CORRECTING CAPABILITY

$$b_2 = m\left(\left\lfloor \frac{t}{2} \right\rfloor - 1\right)+1$$

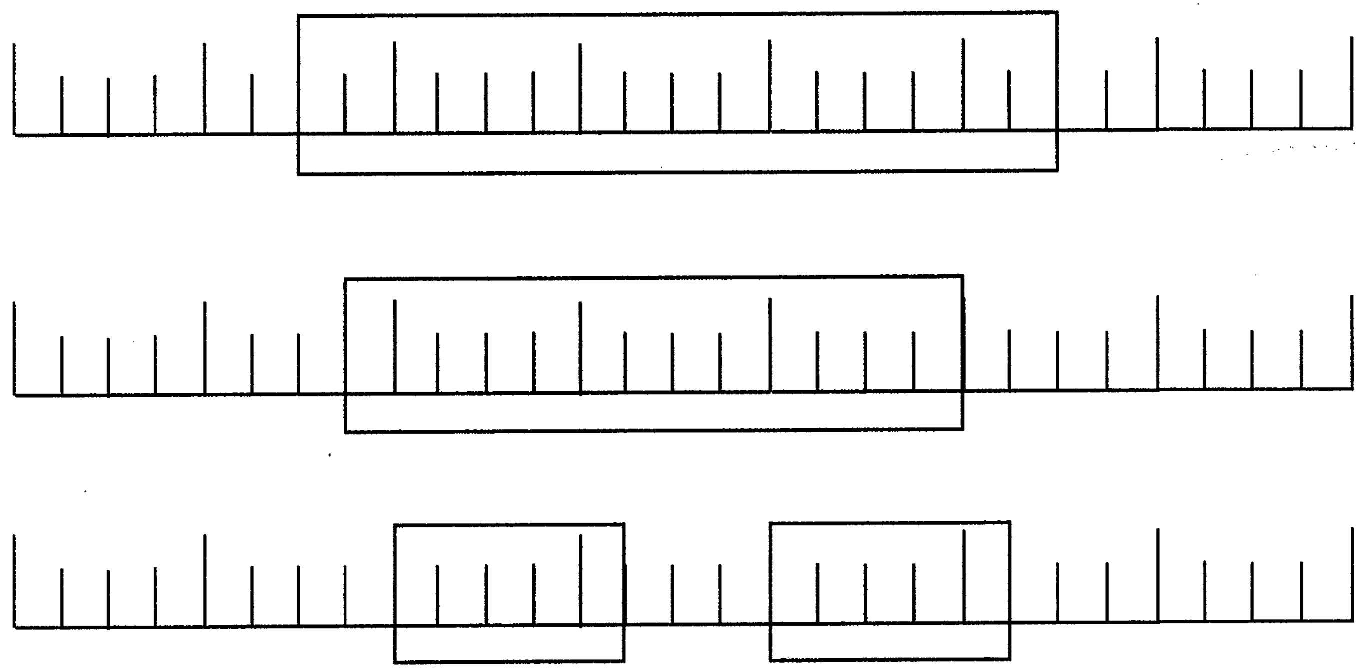
$$b_3 = m\left(\left\lfloor \frac{t}{3} \right\rfloor - 1\right)+1$$

$$b_s = m\left(\left\lfloor \frac{t}{s} \right\rfloor - 1\right)+1$$

[Kam] B. Kamali, Lecture notes on Reed-Solomon Codes, EEE 691.80 Spring Qrt. 1995, Mercer University.

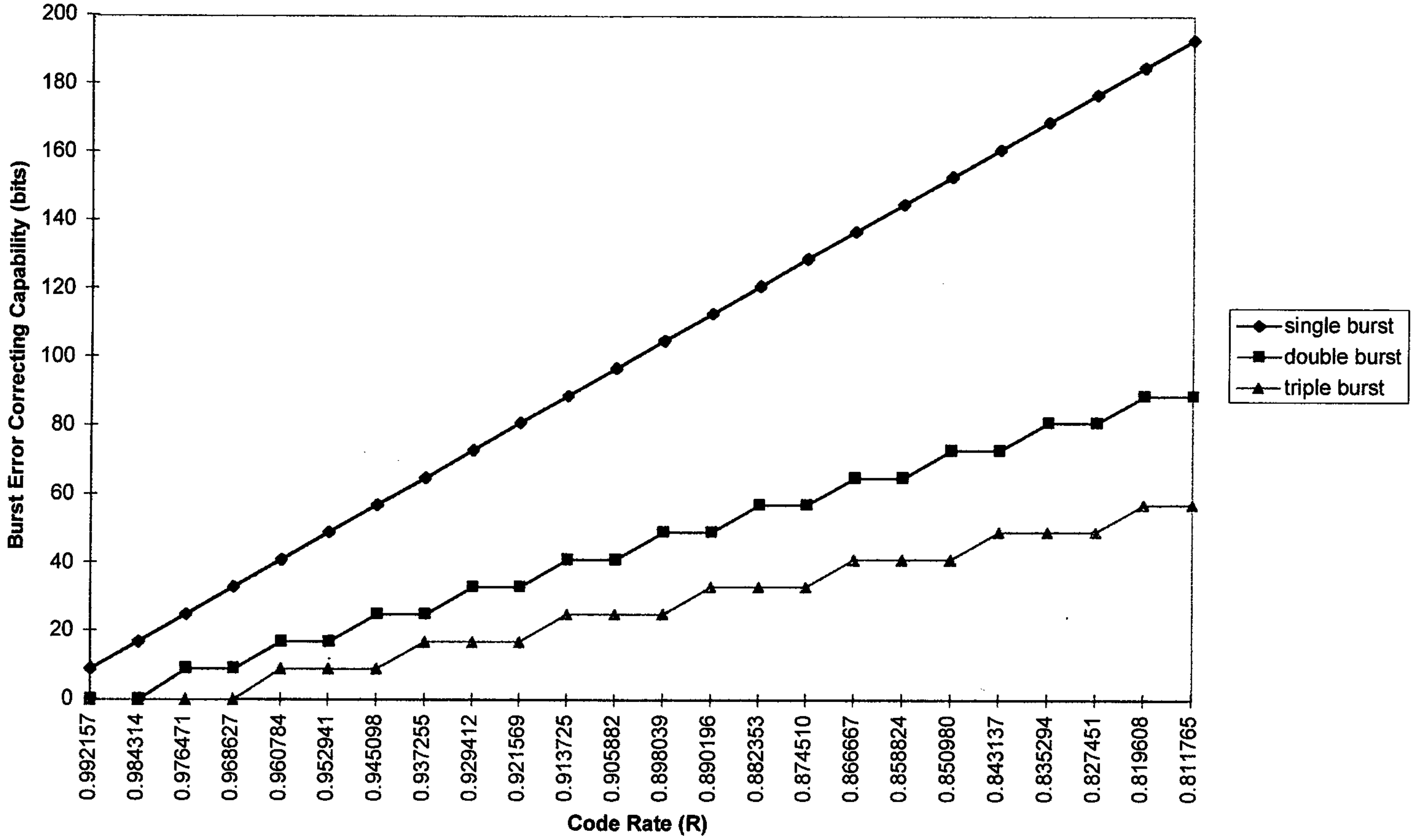
BURST ERROR CORRECTION WITH REED-SOLOMON CODES

GUARANTEED SINGLE AND MULTIPLE BURST ERRORS (t = 4)



BURST ERROR CORRECTION WITH REED-SOLOMON CODES

GUARANTEED BURST ERROR CORRECTING CAPABILITY
(Reed-Solomon Codes of Length n = 255)



BURST ERROR CORRECTION WITH REED-SOLOMON CODES

INTRODUCTION

BURST ERROR CORRECTION CAPABILITIES

BURST ERROR CORRECTION EFFICIENCY
- **Burst Error Efficiency**
- **Maximum Single Burst Error Correction Efficiency**
- **Maximum Multiple Burst Error Correction Efficiency**
- **Guaranteed Burst Error Correction Efficiency**

INTERLEAVED BURST ERROR CORRECTION

EXAMPLE PROBLEM

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

FROM THE GALLAGER BOUND AND REIGER BOUND [CAP]:

$$n-k \geq 2b \qquad n-k \geq b+d$$

BURST ERROR CORRECTION EFFICIENCY OF LINEAR BLOCK CODES [GAL1]

$$z = {}^{2b}\!\!\diagup\!_{(n-k)}$$

*Linear block codes for which z = 1 are considered to be optimal codes

MAXIMUM SINGLE BURST ERROR CORRECTION EFFICIENCY OF RS CODES

$$z_1 = \frac{2b_1}{(n-k)} = \frac{2tm}{(n-k)m} = \frac{2\left(\frac{(n-k)}{2}\right)m}{(n-k)m} = \frac{(n-k)m}{(n-k)m} = 1$$

[Cap] V. Cappellini, Data compression and Error Control Techniques with Applications, Academic Press, 1985.

[Gal1] R. G. Gallager, Information Theory and Reliable Communication, New York: John Wiley & Sons, 1968

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

MAXIMUM MULTIPLE (S) BURST ERROR CORRECTION EFFICIENCY OF RS CODES

$$z_s = \frac{2b_{s\max}}{m(n-k)}$$

$$b_{s\max} = \left\lfloor \frac{t}{s} \right\rfloor m$$

where;

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

GUARANTEED SINGLE, DOUBLE, AND TRIPLE BURST ERROR CORRECTION EFFICIENCY OF REED SOLOMON CODES

$$\gamma_1 = \frac{2\left[m(t-1)+1\right]}{m(n-k)}$$

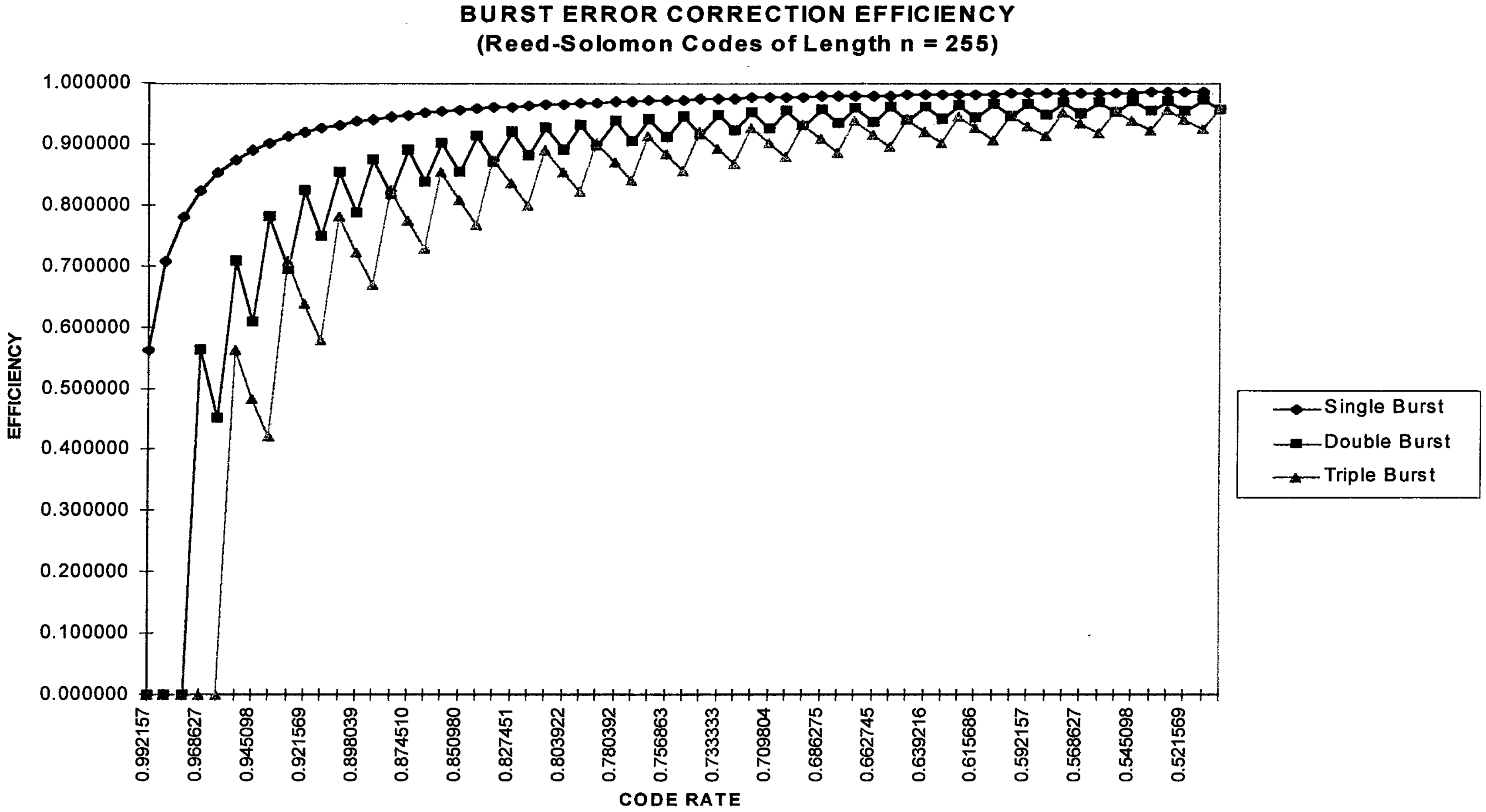
$$\gamma_s = \frac{2sb_s}{m(n-k)}$$

$$\gamma_2 = \frac{4\left[m\left(\left\lfloor \frac{t}{2} \right\rfloor - 1\right) + 1\right]}{m(n-k)}$$

$$b_s = m\left(\left\lfloor \frac{t}{s} \right\rfloor - 1\right) + 1$$

$$\gamma_3 = \frac{6\left[m\left(\left\lfloor \frac{t}{3} \right\rfloor - 1\right) + 1\right]}{m(n-k)}$$

BURST ERROR CORRECTION WITH REED-SOLOMON CODES



(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(255,253)	1	0.992157	-	-	-	-	-	-
(255,251)	2	0.984314	9	0.562500	-	-	-	-
(255,249)	3	0.976471	17	0.708333	-	-	-	-
(255,247)	4	0.968627	25	0.781250	9	0.562500	-	-
(255,245)	5	0.960784	33	0.825000	9	0.450000	-	-
(255,243)	6	0.952941	41	0.854167	17	0.708333	9	0.562500
(255,241)	7	0.945098	49	0.875000	17	0.607143	9	0.482143
(255,239)	8	0.937255	57	0.890625	25	0.781250	9	0.421875
(255,237)	9	0.929412	65	0.902778	25	0.694444	17	0.708333
(255,235)	10	0.921569	73	0.912500	33	0.825000	17	0.637500
(255,233)	11	0.913725	81	0.920455	33	0.750000	17	0.579545
(255,231)	12	0.905882	89	0.927083	41	0.854167	25	0.781250
(255,229)	13	0.898039	97	0.932692	41	0.788462	25	0.721154
(255,227)	14	0.890196	105	0.937500	49	0.875000	25	0.669643
(255,225)	15	0.882353	113	0.941667	49	0.816667	33	0.825000
(255,223)	16	0.874510	121	0.945313	57	0.890625	33	0.773438
(255,221)	17	0.866667	129	0.948529	57	0.838235	33	0.727941
(255,219)	18	0.858824	137	0.951389	65	0.902778	41	0.854167
(255,217)	19	0.850980	145	0.953947	65	0.855263	41	0.809211
(255,215)	20	0.843137	153	0.956250	73	0.912500	41	0.768750
(255,213)	21	0.835294	161	0.958333	73	0.869048	49	0.875000
(255,211)	22	0.827451	169	0.960227	81	0.920455	49	0.835227
(255,209)	23	0.819608	177	0.961957	81	0.880435	49	0.798913
(255,207)	24	0.811765	185	0.963542	89	0.927083	57	0.890625
(255,205)	25	0.803922	193	0.965000	89	0.890000	57	0.855000
(255,203)	26	0.796078	201	0.966346	97	0.932692	57	0.822115
(255,201)	27	0.788235	209	0.967593	97	0.898148	65	0.902778
(255,199)	28	0.780392	217	0.968750	105	0.937500	65	0.870536
(255,197)	29	0.772549	225	0.969828	105	0.905172	65	0.840517
(255,195)	30	0.764706	233	0.970833	113	0.941667	73	0.912500
(255,193)	31	0.756863	241	0.971774	113	0.911290	73	0.883065
(255,191)	32	0.749020	249	0.972656	121	0.945313	73	0.855469
(255,189)	33	0.741176	257	0.973485	121	0.916667	81	0.920455
(255,187)	34	0.733333	265	0.974265	129	0.948529	81	0.893382
(255,185)	35	0.725490	273	0.975000	129	0.921429	81	0.867857
(255,183)	36	0.717647	281	0.975694	137	0.951389	89	0.927083
(255,181)	37	0.709804	289	0.976351	137	0.925676	89	0.902027
(255,179)	38	0.701961	297	0.976974	145	0.953947	89	0.878289
(255,177)	39	0.694118	305	0.977564	145	0.929487	97	0.932692
(255,175)	40	0.686275	313	0.978125	153	0.956250	97	0.909375
(255,173)	41	0.678431	321	0.978659	153	0.932927	97	0.887195
(255,171)	42	0.670588	329	0.979167	161	0.958333	105	0.937500
(255,169)	43	0.662745	337	0.979651	161	0.936047	105	0.915698
(255,167)	44	0.654902	345	0.980114	169	0.960227	105	0.894886
(255,165)	45	0.647059	353	0.980556	169	0.938889	113	0.941667
(255,163)	46	0.639216	361	0.980978	177	0.961957	113	0.921196
(255,161)	47	0.631373	369	0.981383	177	0.941489	113	0.901596
(255,159)	48	0.623529	377	0.981771	185	0.963542	121	0.945313
(255,157)	49	0.615686	385	0.982143	185	0.943878	121	0.926020
(255,155)	50	0.607843	393	0.982500	193	0.965000	121	0.907500
(255,153)	51	0.600000	401	0.982843	193	0.946078	129	0.948529
(255,151)	52	0.592157	409	0.983173	201	0.966346	129	0.930288
(255,149)	53	0.584314	417	0.983491	201	0.948113	129	0.912736
(255,147)	54	0.576471	425	0.983796	209	0.967593	137	0.951389
(255,145)	55	0.568627	433	0.984091	209	0.950000	137	0.934091
(255,143)	56	0.560784	441	0.984375	217	0.968750	137	0.917411
(255,141)	57	0.552941	449	0.984649	217	0.951754	145	0.953947
(255,139)	58	0.545098	457	0.984914	225	0.969828	145	0.937500
(255,137)	59	0.537255	465	0.985169	225	0.953390	145	0.921610
(255,135)	60	0.529412	473	0.985417	233	0.970833	153	0.956250
(255,133)	61	0.521569	481	0.985656	233	0.954918	153	0.940574
(255,131)	62	0.513725	489	0.985887	241	0.971774	153	0.925403
(255,129)	63	0.505882	497	0.986111	241	0.956349	161	0.958333

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Ronald L. Brinkley
MSEE Program
Final Briefing

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

INTRODUCTION

BURST ERROR CORRECTION CAPABILITIES

BURST ERROR CORRECTION EFFICIENCY

INTERLEAVED BURST ERROR CORRECTION

- **Block Interleaved, Guaranteed Burst Error Correction Capability**
- **Block Interleaved, Guaranteed Burst Error Correction Efficiency**
- **Convolutional Interleaved, Guaranteed Burst Error Correction Capability**
- **Convolutional Interleaved, Guaranteed Burst Error Correction Efficiency**

EXAMPLE PROBLEM

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

BLOCK INTERLEAVED, GUARANTEED BURST ERROR CORRECTION CAPABILITIES OF REED-SOLOMON CODES

$$\psi_1 = m(td - 1) + 1$$

$$\psi_2 = m\left[\left\lfloor \frac{t}{2} \right\rfloor d - 1\right] + 1$$

$$\psi_3 = m\left[\left\lfloor \frac{t}{3} \right\rfloor d - 1\right] + 1$$

$$\psi_s = m\left[\left\lfloor \frac{t}{s} \right\rfloor d - 1\right] + 1$$

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Ronald L. Brinkley
MSEE Program
Final Briefing

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

BLOCK INTERLEAVED, GUARANTEED BURST ERROR CORRECTION EFFECEINCY OF REED-SOLOMON CODES

$$\Psi_1 = \frac{2\left[m(td-1)+1\right]}{m(n-k)d}$$

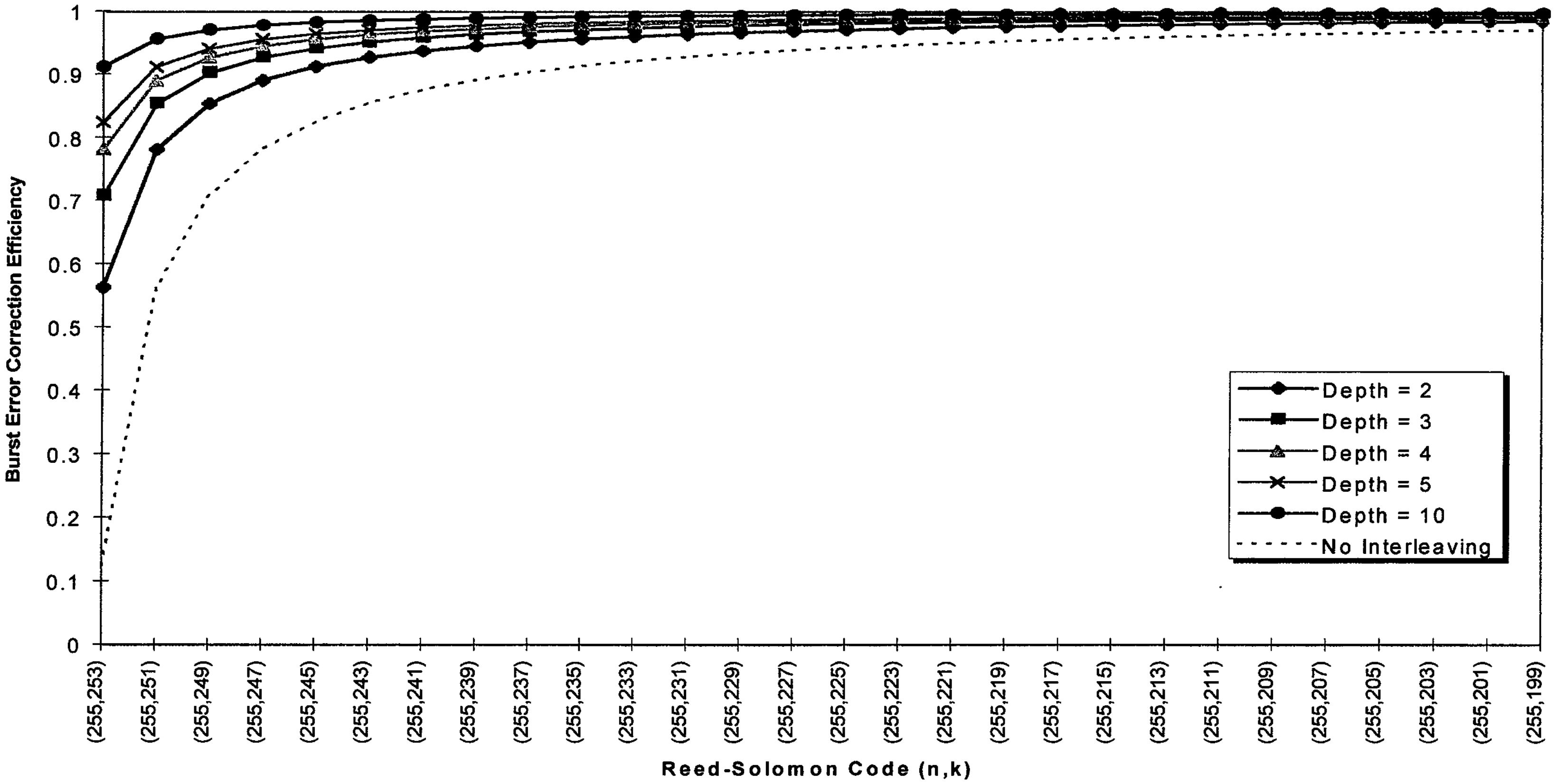
$$\Psi_2 = \frac{4\left[m\left(\left\lfloor \frac{t}{2} \right\rfloor d-1\right)+1\right]}{m(n-k)d}$$

$$\Psi_3 = \frac{6\left[m\left(\left\lfloor \frac{t}{3} \right\rfloor d-1\right)+1\right]}{m(n-k)d}$$

$$\Psi_s = \frac{2s\left[m\left(\left\lfloor \frac{t}{s} \right\rfloor d-1\right)+1\right]}{m(n-k)d}$$

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

BLOCK INTERLEAVED REED-SOLOMON CODE
Guaranteed Single Burst Error Correction Efficiency
(Code Length n = 255)



BURST ERROR CORRECTION WITH REED-SOLOMON CODES

CONVOLUTIONAL INTERLEAVED, GUARANTEED BURST ERROR CORRECTION CAPABILITIES OF REED-SOLOMON CODES.

$$\phi_1 = m[t(d\ell + 1) - 1] + 1$$

$$\phi_2 = m\left[\left\lfloor \frac{t}{2} \right\rfloor (d\ell + 1) - 1\right] + 1$$

$$\phi_3 = m\left[\left\lfloor \frac{t}{3} \right\rfloor (d\ell + 1) - 1\right] + 1$$

$$\phi_s = m\left[\left\lfloor \frac{t}{s} \right\rfloor (d\ell + 1) - 1\right] + 1$$

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

CONVOLUTIONAL INTERLEAVED, GUARANTEED BURST ERROR CORRECTION EFFECEINCY OF REED-SOLOMON CODES.

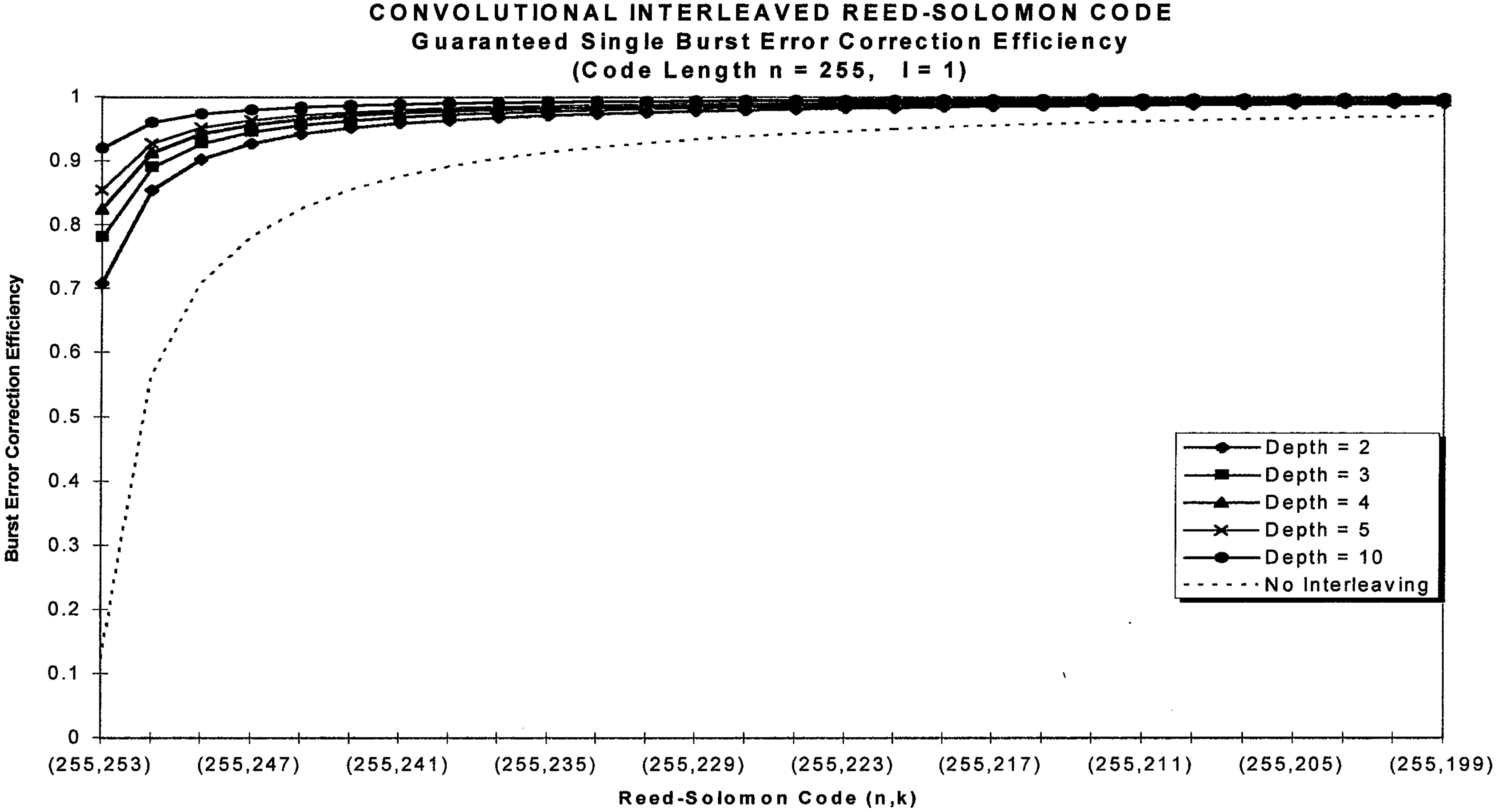
$$\Phi_1 = \frac{2m\left[t(d\ell+1)-1\right]+1}{m(n-k)(d\ell+1)}$$

$$\Phi_2 = \frac{4m\left[\left\lfloor \frac{t}{2} \right\rfloor (d\ell+1)-1\right]+1}{m(n-k)(d\ell+1)}$$

$$\Phi_3 = \frac{6m\left[\left\lfloor \frac{t}{3} \right\rfloor (d\ell+1)-1\right]+1}{m(n-k)(d\ell+1)}$$

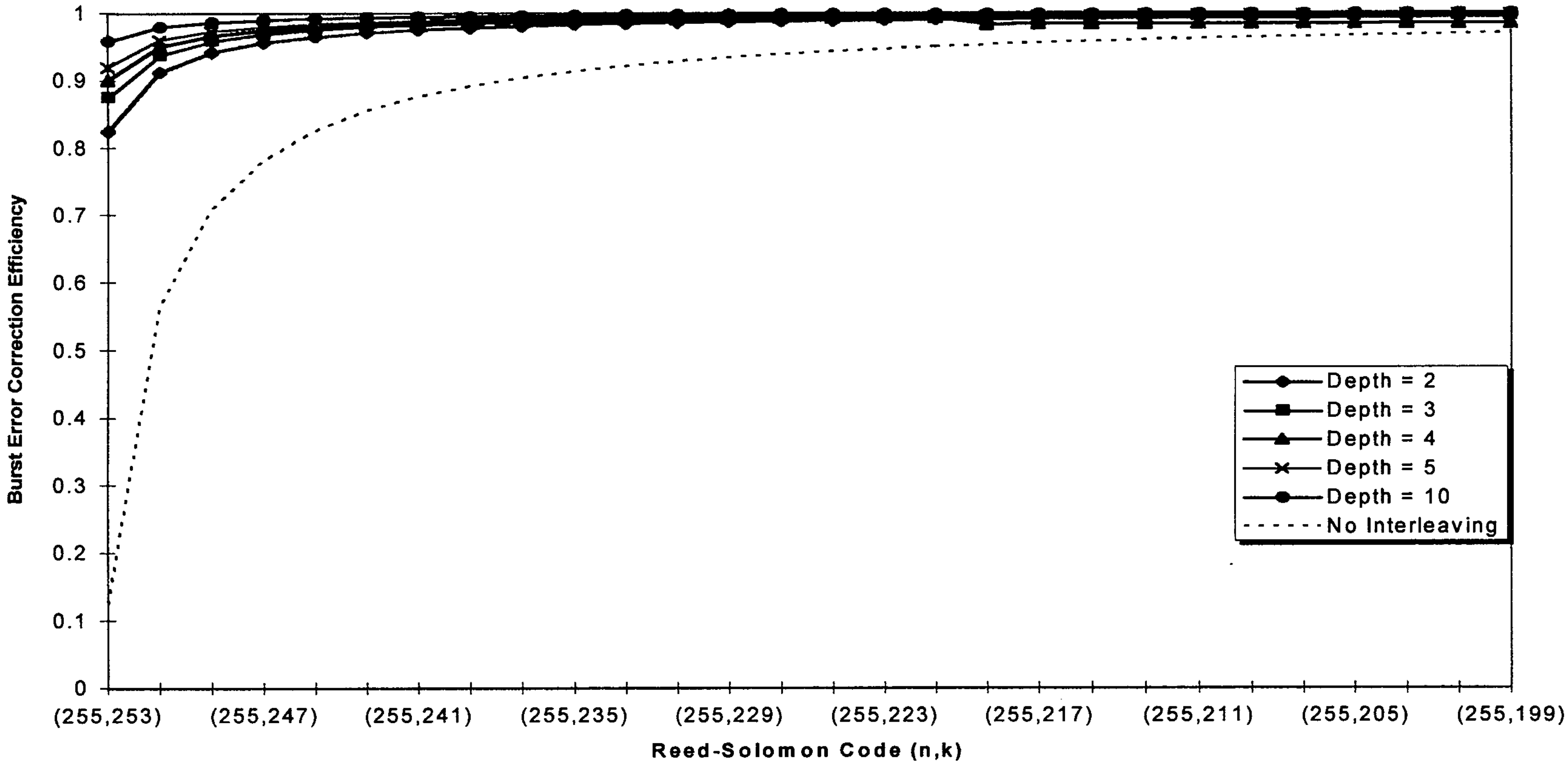
$$\Phi_s = \frac{2sm\left[\left\lfloor \frac{t}{s} \right\rfloor (d\ell+1)-1\right]+1}{m(n-k)(d\ell+1)}$$

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

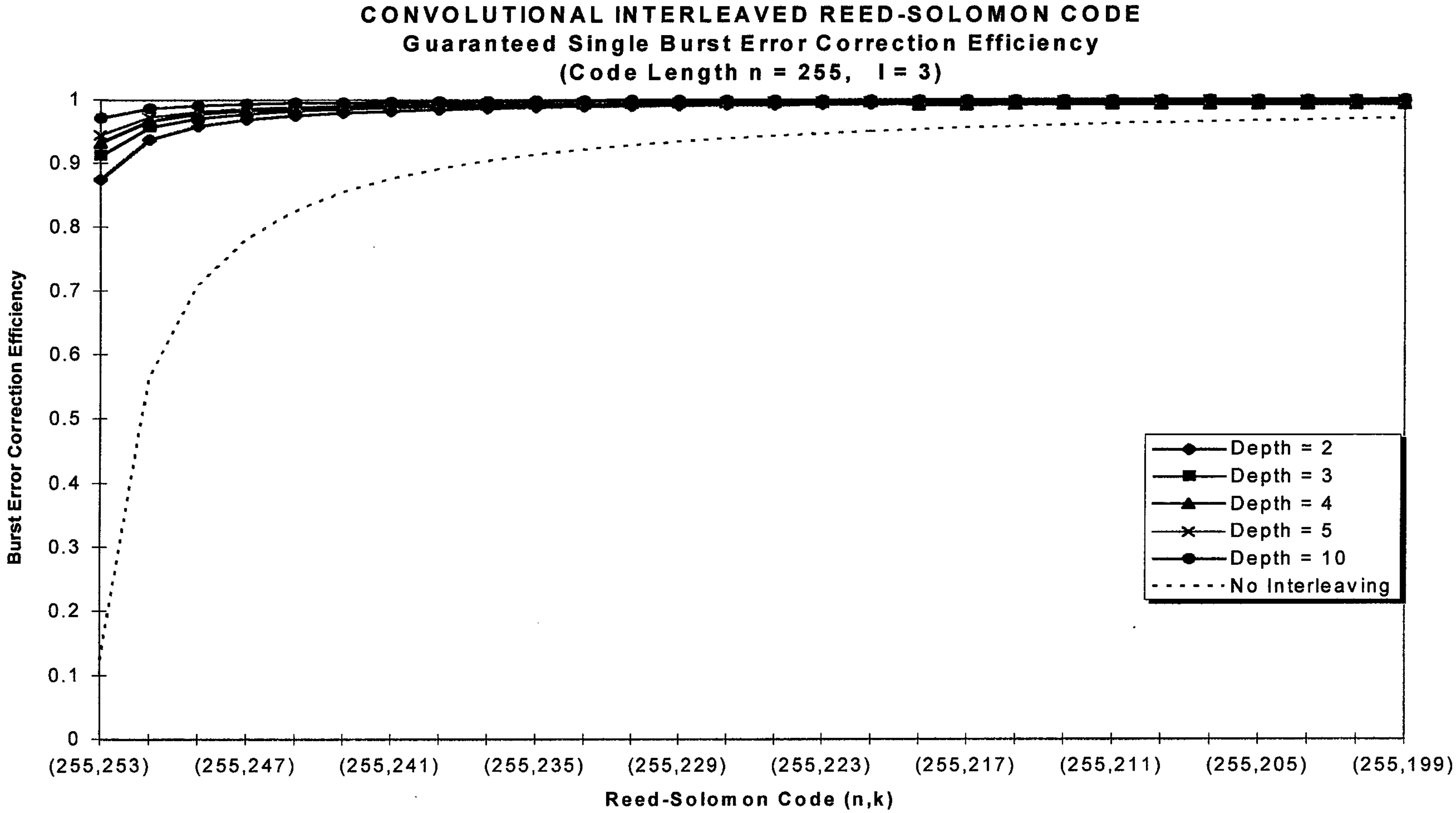


BURST ERROR CORRECTION WITH REED-SOLOMON CODES

CONVOLUTIONAL INTERLEAVED REED-SOLOMON CODE
Guaranteed Single Burst Error Correction Efficiency
(Code Length n = 255, I = 2)

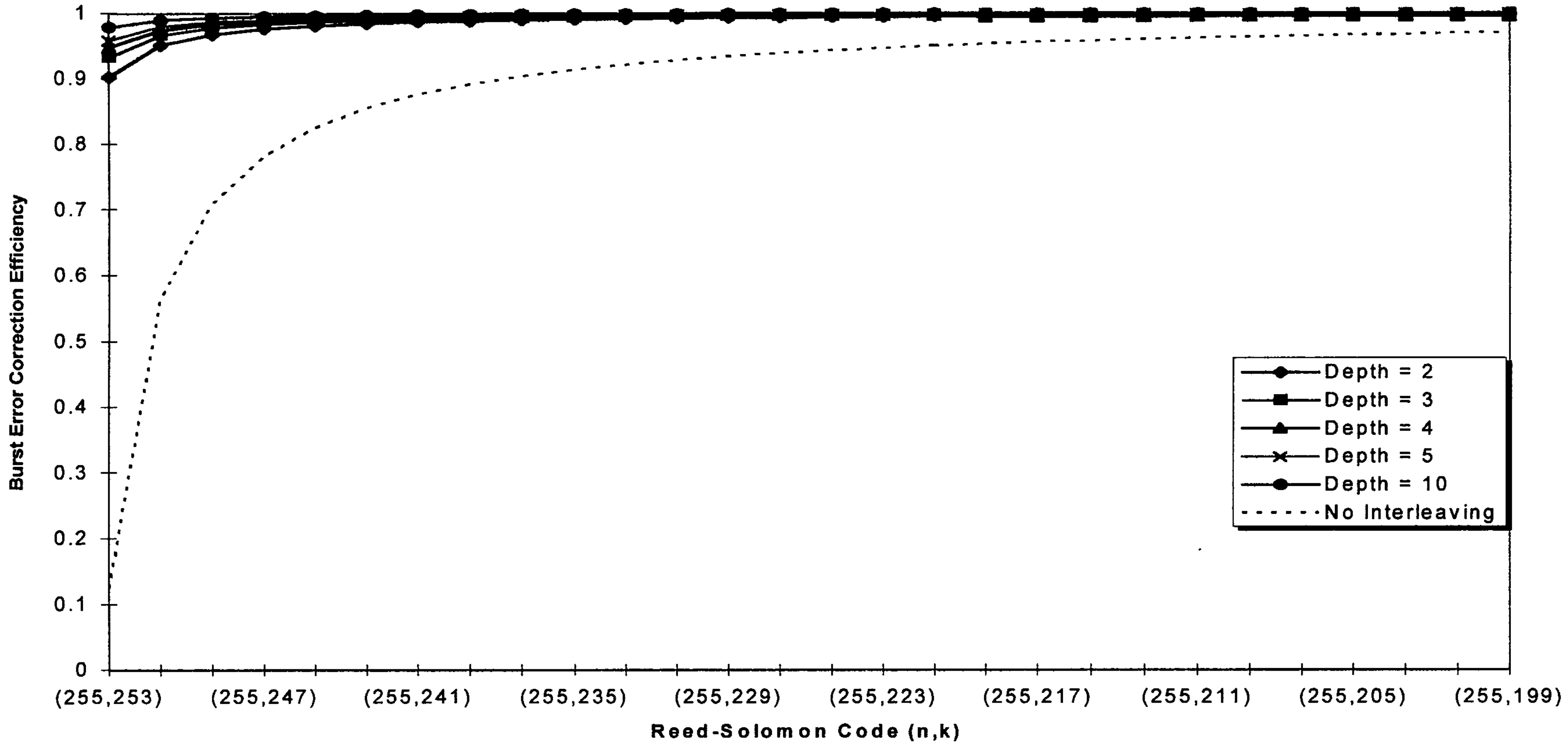


BURST ERROR CORRECTION WITH REED-SOLOMON CODES



BURST ERROR CORRECTION WITH REED-SOLOMON CODES

CONVOLUTIONAL INTERLEAVED REED-SOLOMON CODE
Guaranteed Single Burst Error Correction Efficiency
(Code Length n = 255, I = 4)



5.1.6 Presentation by Henderson Benjamin

The student briefing presented by Henderson Benjamin at this meeting is reproduced on the next 24 pages.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

RESEARCH FINAL REPORT

H. Benjamin
WR-ALC/LKSC
JSTARS System Engineer

Date Approved: May 7, 1996

Projected Completion Date: April 2, 1998

Research Topic: *NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION*

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

PROBLEM STATEMENT

Reed-Solomon codes are powerful error control coding due to their ability to detect and correct random and burst errors. Reed-Solomon codes are maximum distance separable. Because of the large number of Reed-Solomon codes to select from, it was determined that some means of Artificial Intelligence would be useful for selecting a Reed-Solomon code for a given application. Data was collected and analyzed for Reed-Solomon codes and shortened Reed-Solomon codes. The goal of this thesis work is to develop a Neural Network which will select the best possible code for a particular application.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

RESEARCH OBJECTIVES

- Determine the method for selecting Reed-Solomon codes

- Collect/Analyze Reed-Solomon Code Data
    - Collect/Analyze data for RS codes of length 7, 15, 31, 63, 127, and 255
    - Collect/Analyze additional data deemed useful to a designer
    - Develop Neural Network

- Collect/Analyze shortened Reed-Solomon Code Data
    - Collect/Analyze data for shortened RS codes of length 255
    - Collect/Analyze additional data deemed useful to a designer
    - Develop Neural Network

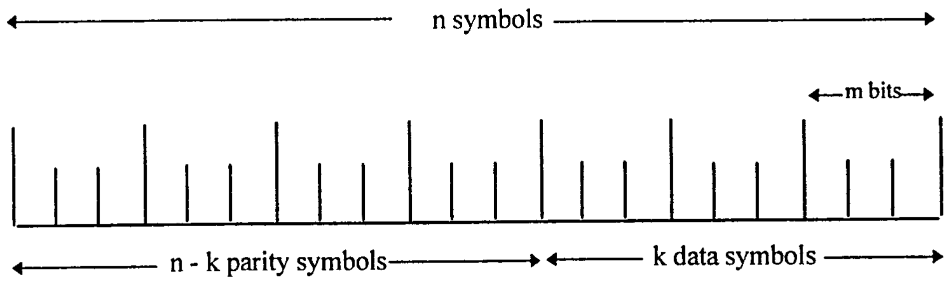
NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

REED-SOLOMON CODES

- Reed-Solomon codes are word oriented which means every codeword is composed of m bit symbols.
- The nonbinary alphabet is the preferred method because it provides large burst error correcting capability, and the encoder/decoder speed is much better than an equivalent binary code.
- It is a powerful tool against burst error which occur in multipath fading channels, Jamming, and recording systems.

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

(7,3) REED-SOLOMON CODE



RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

Guaranteed RS Code Burst Error Correction Equations:

$$\gamma_1 = m[t-1]+1$$

$$\gamma_2 = m\left[\left\lfloor \frac{t}{2} \right\rfloor - 1\right]+1$$

$$\gamma_3 = m\left[\left\lfloor \frac{t}{3} \right\rfloor - 1\right]+1$$

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

MODIFIED REED-SOLOMON CODES

- RS codes are modifiable which means that they can be punctured, extended, or shortened.
- The most efficient modifications of Reed-Solomon generated codes are through shortening and puncturing.
- The shortened codes are preferred because they require a smaller memory matrix in interleaved cases

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

SHORTENED REED-SOLOMON CODE

- CD's currently use an interleaved shortened RS code.
- The original length was (255, 251). The shortened lengths used are (32,28) and (28,24).
- This results in 87% and 89% reduction of codewords and 87% and 90% of the code dimensions.
- The trade off of shortened codes is the efficiency of the coderate 10 and 12% respectively in the above case.

*NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON
CODES FOR A SPECIFIC APPLICATION*

NEURAL NETWORKS

- A Neural Network is an approach to computing that involves developing mathematical structures with the ability to learn.

- The human brain has many features that are desirable in AI (fault tolerant and robust, very flexible; deals with fuzzy, probabilistic, noisy or inconsistent data; parallel, small, compact, and dissipates very little power).

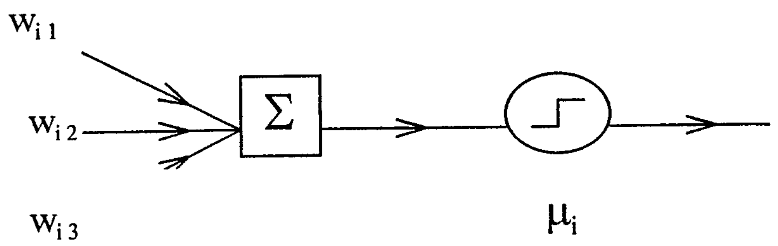
- The current cycle time of a neuron is approximately a millisecond which is a million times slower than semiconductor devices.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

MODELING NEURONS



$$a = w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

$$a = \sum_{n=1}^{n} w_nx_n$$

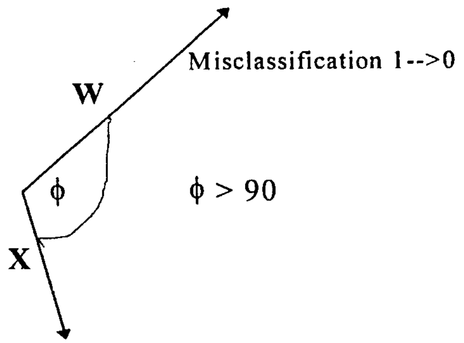
$$y = \begin{cases} 1 & \text{if } a \geq \theta \\ 0 & \text{if } a < \theta \end{cases}$$

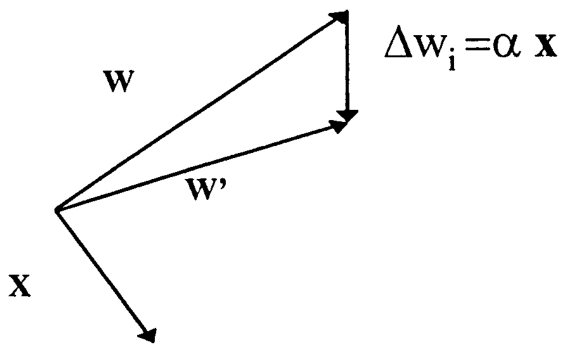
RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

ADJUSTING WEIGHTS FOR LEARNING

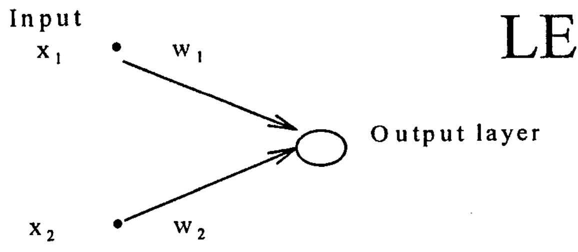




$$\mathbf{w'} = \Delta w_i + w_i$$

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

LEARNING THE LOGICAL AND



$$a = w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

$$a = \sum_{n=1}^{n} w_nx_n$$

$$y = \begin{cases} 1 & \text{if } a \geq \theta \\ 0 & \text{if } a < \theta \end{cases}$$

$$w_i' = \alpha\, x_i + w_i$$

step	w_1	w_2	x_1	x_2	a=Activation	y=Output	Comment
1	0	0	0	0	0	0	Output is correct
2	0	0	0	1	0	0	Output is correct
3	0	0	1	0	0	0	Output is correct
4	0	0	1	1	0	0	Output is incorrect Adjust the weights
5	0.5	0.5	0	0	0	0	Output is correct
6	0.5	0.5	0	1	0.5	0	Output is correct
7	0.5	0.5	1	0	0.5	0	Output is correct
8	0.5	0.5	1	1	1	1	Output is correct
9	0.5	0.5	0	0	0	0	Output is correct
10	0.5	0.5	0	1	0.5	0	Output is correct
11	0.5	0.5	1	0	0.5	0	Output is correct

RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON
CODES FOR A SPECIFIC APPLICATION

NEURONS AS FUNCTIONS

- The sigmoid function introduces nonlinearity into a network.
- The sigmoid function is used to bound the input signal.

$$S(x) = \frac{1}{1 + e^{-\alpha}}$$

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

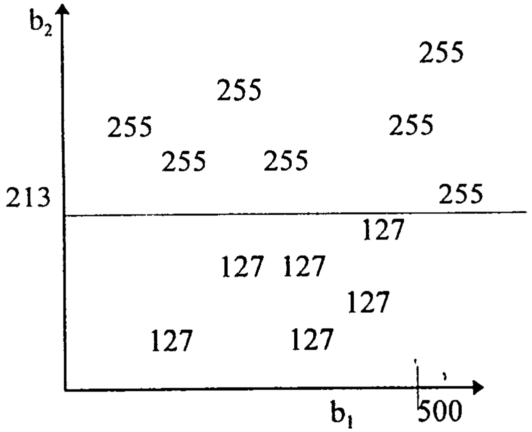
- The sigmoid function is strictly increasing for positive scaling constant $c > 0$. Strict monotonicity implies that the activation derivative of S is positive.

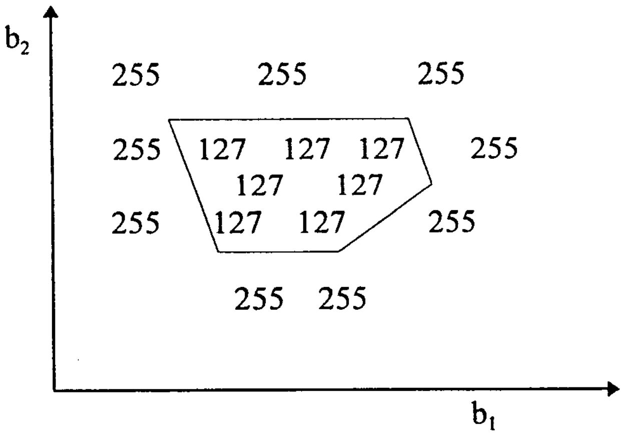
$$S' = \frac{dS}{dx} = cS(1-S) > 0$$

- Monotonic increasing implies that the output increases when the input increases.

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

DECISION SURFACE



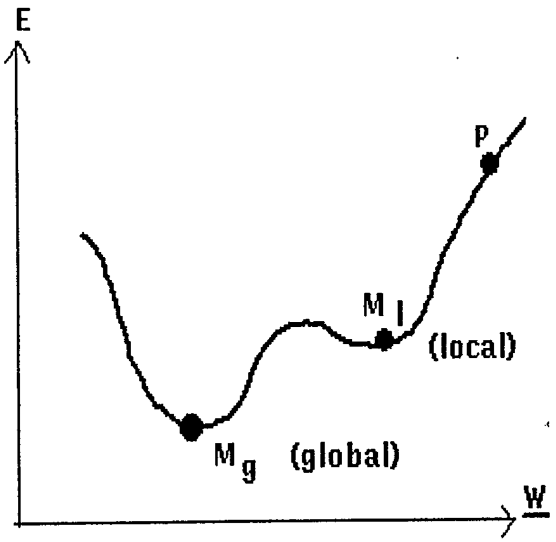


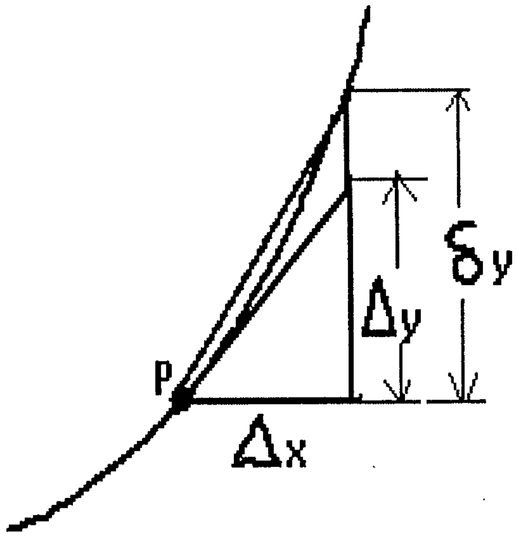
RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

MINIMIZING AN ERROR





$$E = 1/2\,(T - A)^2$$
$$\delta y \cong -\alpha\,(\Delta y/\Delta x)^2$$

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

BACKPROPOGATION ALGORITHM

- A Backpropogation algorithms a nonlinear extension of the LMS algorithm. The LMS is similar to the Kalman filter.

- It is derived with iterative applications of the chain rule of differential calculus and embedded in a stochastic approximation framework.

- Levenberg-Marquardt (LM) is a variation of backpropogation method. It also utilizes an approximation of Newton's method.

$$h_j^{\mu} = \sum_k w_{jk}\, \xi_k^{\mu}$$

$$V_j^{\mu} = g(h_j^{\mu}) = g(\sum_k w_{jk} \xi_k^{\mu})$$

$$h_i^{\mu} = \sum_j W_{ij} V_j^{\mu} = \sum_j W_{ij}\, g(\sum_k w_{jk} \xi_k^{\mu})$$

$$O_i^{\mu} = g(h_j^{\mu}) = g\,(\sum_j W_{ij} V_j^{\mu}) = g\,(\sum_j W_{ij}\, g(\sum_k w_{jk} \xi_k^{\mu}))$$

$$E[w] = ½ \sum_{\mu i} [\, \zeta_i^{\mu} - O_i^{\mu}]^2$$

$$E[w] = ½ \sum_{\mu i} [\, \zeta_i^{\mu} - g\,(\sum_j W_{ij}\, g(\sum_k w_{jk} \xi_k\,))\,]^2$$

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = \eta \sum_{\mu} [\, \zeta_i^{\mu} - O_i^{\mu}]\, g'\,(h_i^{\mu})\, V_j^{\mu} = \eta \sum_{\mu} \delta_i^{\mu} V_j^{\mu}$$

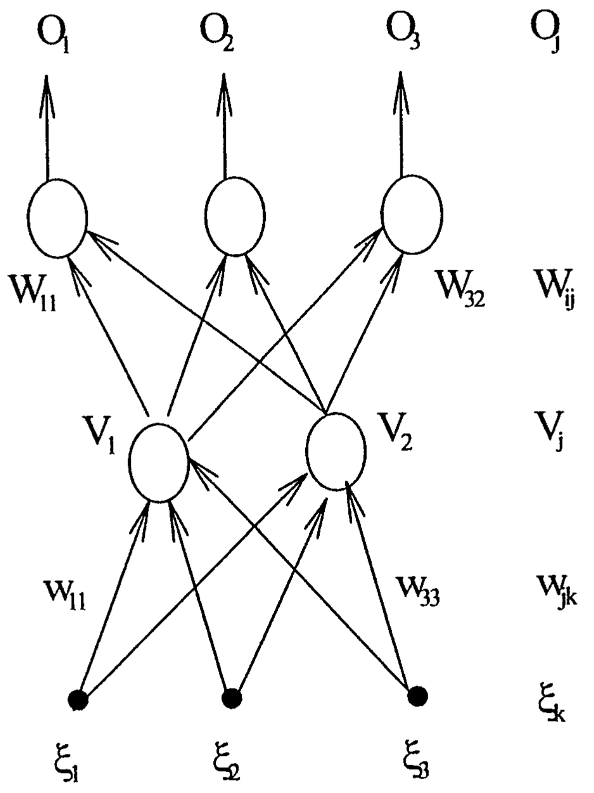
$$\delta_i^{\mu} = g'(h_i^{\mu})\, [\, \zeta_i^{\mu} - O_i^{\mu}]$$

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \sum \frac{\partial E}{\partial V_j^{\mu}} \frac{\partial V_j^{\mu}}{\partial w_{jk}} = \eta \sum_{\mu i} [\, \zeta_i^{\mu} - O_i^{\mu}]\, g'(h_i^{\mu})\, W_{ij}\, g'(h_i^{\mu})\, \xi_k^{\mu}$$

$$= \eta \sum_{\mu i} \delta_i^{\mu} W_{ij}\, g'(h_i^{\mu})\, \xi_k^{\mu} = \eta \sum_{\mu i} \delta_j^{\mu}\, \xi_k^{\mu}$$

$$\delta_j^{\mu} = g'(h_i^{\mu}) \sum W_{ij} \delta_i^{\mu}$$

$$\Delta W = (\, J^T J + \mu\, I\,)^{-1} J^T$$

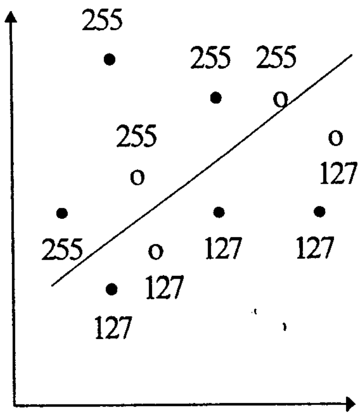


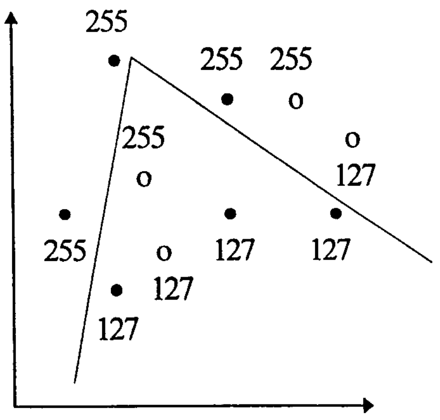
RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

GENERALIZING 127 AND 255 RS CODE LENGTHS





RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

NN # 1 PATTERNS

m	t	Rc	b1	b2	b3
5	X	X	X	X	X
5	1	X	X	X	X
5	X	61.29	X	X	X
5	X	X	26	X	X
5	X	X	X	11	X
5	X	X	X	X	6
5	1	61.29	X	X	X
5	1	X	22	X	X
5	1	X	X	11	X
5	2	X	X	X	6
5	2	25	23	X	X
5	2	50	X	8	X
5	3	55	X	X	6
5	3	25	24	7	X
5	4	45	25	X	6
5	5	60	X	4	6
5	6	20	26	1	6

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM

Henderson C. Benjamin
MSEE Program
Research Update

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

NN # 1 TRAINING DATA FOR "DON'T CARES"

m	t	Rc	b_1	b_2	b_3	n	k
5	6	1	0	0	0	31	19
5	6	2	0	0	0	31	19
5	6	:	0	0	0	31	19
5	6	61.29	0	0	0	31	19
5	6	1	1	0	0	31	19
5	6	1	2	0	0	31	19
5	6	1	:	0	0	31	19
5	6	1	26	0	0	31	19
5	6	2	1	0	0	31	19
5	6	2	2	0	0	31	19
5	6	2	:	0	0	31	19
5	6	2	26	0	0	31	19
5	6	:	1	0	0	31	19
5	6	61.29	:	0	0	31	19

m	t	Rc	b_1	b_2	b_3	n	k
5	6	1	0	0	0	31	19
5	6	61.29	0	11	0	31	19
5	6	1	26	0	0	31	19
5	6	1	0	11	0	31	19
5	6	1	0	0	6	31	19
5	6	61.29	26	0	0	31	19
5	6	61.29	0	11	0	31	19
5	6	61.29	0	0	6	31	19

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

NN # 2 PATTERNS/TRAINING DATA FOR “DON’T CARES”

%s	t	b_1	b_2	b_3
0	X	X	X	X
X	10	X	X	X
X	X	50	X	X
X	X	X	100	X
X	X	X	X	100
10	15	X	X	X
20	X	250	X	X
30	X	X	200	X
40	X	X	X	150
50	18	400	X	X
60	27	X	300	X
70	35	X	X	200
80	50	500	400	X
90	95	700	X	250
90	115	800	450	X
90	127	1000	500	300

% s	t	b_1	b_2	b_3	n	k
0	1	162	74	42	255	211
0	1	162	0	0	255	211
0	1	0	74	0	255	211
0	1	0	0	42	255	211
0	1	169	0	0	255	211
0	1	0	81	0	255	211
0	1	0	0	49	255	211
0	1	162	74	0	255	211
0	1	162	0	42	255	211
0	1	0	74	42	255	211
0	1	169	81	0	255	211
0	1	169	0	49	255	211
0	1	0	81	49	255	211

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

RESEARCH RESULTS

Neural Network Identification # 1: This NN is used when the designer specifies the hardware to be used and will result in the most efficient code dimension.

This is a six input system (m, t, Rc, γ_1, γ_2, γ_3)

- The desired outcome the code dimension (k).
- The test data results were : 99.2 % Correct

  0.8 % Incorrect by 1 data length

NEURAL NETWORK SYSTEM THAT SELECTS REED-SOLOMON CODES FOR A SPECIFIC APPLICATION

RESEARCH RESULTS

Neural Network Identification # 2: The second NN selects the most efficient shortened Reed-Solomon code. The designer's only requirement is to enter the percentage of the total number of words to be shortened.

This is a five input system (%s, t, γ_1, γ_2, γ_3)

- The desired outcome is to determine the code and data length (n, k)
- The test data results were : 98.36% Correct for the codewords
  84.40% Correct for the code dimension

5.1.7 Presentation by Randy Ford

The student briefing presented by Randy Ford at this meeting is reproduced on the next 26 pages.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

RESEARCH PROPOSAL PRESENTATION

James R. Ford

WRALC/LYSBC
JTIDS Software Engineer

Background and Experience:

Education:	BEE from Georgia Tech
	Pursuing MSE/SE with emphasis in AI - 33 hrs. completed
U.S. Army:	Signal Corps Platoon Leader, 2 years
WR/ALC:	Avionics Software Engineering, 14 years

Research Topic:

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROBLEM STATEMENT

In “An Efficient Method of Passive Emitter Location” Klaus Becker proposes using multiple frequency measurements by a moving sensor to generate a Cramer-Rao bound ellipsoid. This ellipsoid is then projected into the x-y plane to form an error surface which may be searched by generating a starting point and then minimizing the sum square error (SSE) using the simplex method. Once the emitter’s location is found, the estimated emitter frequency is determined by obtaining a maximum likelihood (ML) value at that point.

Differential Evolution, developed by Kenneth Price and Rainer Storn, is a minimization technique for real-valued problems which is based on genetic algorithms. This project will compare the performance of this method, with the performance of the Nelder-Mead simplex method. Both methods will be implemented in C++ and called from an existing MATLAB implementation.

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROPOSAL JUSTIFICATION

- ★ Applicability to Mercer University MS SSE Program
  - Practical application of genetic algorithms/AI
  - Practical application of Differential Evolution
- ★ Applicability to the USAF
  - Application of genetic algorithms/AI to increase effectiveness of existing technique
  - Possible use of Differential Evolution in other areas

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROPOSED RESEARCH OBJECTIVES

- ★ Implementation of Differential Evolution
  - In C++
  - Possibly in MATLAB
- ★ Primary Objective is:
  - To determine best minimization technique:
    Which one produces <u>fastest</u> solution?

MERCERIENSIS UNIVERSITATIS SIGILLUM
SCIENTIIS ARTIBUS RELIGIONI
MACON / ATLANTA
MDCCCXXXVIII

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROPOSED METHOD OF INVESTIGATION

- ★ Study existing MATLAB implementation
- ★ Implement minimization in C++
  - Implement Nelder-Mead simplex method
  - Implement DE minimization.
- ★ Test both implementations
  - Identical Platforms - Pentium PC
  - Determine <u>best</u> method based on speed of detection.

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROPOSED METHOD OF INVESTIGATION

- ★ Test both implementations (cont)
  - Consideration will also be given to the frequency with which a method produces an unacceptably slow solution or no solution.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROPOSED SCHEDULE

- ★ Semester 1 (Spring 98 - 3 hours)
  - Academic Approval
  - RAPCEval Approval
  - Study MATLAB Implementation of simulator
  - Begin coding C++ Nelder-Mead Implementation
- ★ Semester 2 (Summer 98 - 1 hour)
  - Finish coding C++ Nelder-Mead Implementation

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROPOSED SCHEDULE

- ★ Semester 2 (Summer 98 - 1 hour)
    - Begin coding C++ DE Implementation
- ★ Semester 3 (Fall 98 - 2 hours)
    - Finish coding C++ DE Implementation
    - Test and Compare Implementations
    - Final Project Report

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Passive Emitter Location

- Locate a stationary emitter by taking multiple frequency measurements from a moving platform.
- Obtain the Cramer-Rao (CR) bound ellipsoid
- Get x,y ellipse by projecting the CR ellipsoid onto the x,y plane.

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Passive Emitter Location (continued)
  - Run a Monte Carlo simulation to determine starting points for the search
  - Find minimum sum squared errors and estimated emitter frequency starting at each point.

★ Passive Emitter Location discussion is based on “An Efficient Method of Passive Emitter Location” by Klaus Becker in the October 1992 IEEE Transactions on Aerospace and Electronic Systems.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM

(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Nelder-Mead Simplex Method

- A **Simplex** is a geometrical figure consisting, in *N* dimensions, of *N* + 1 points and all their interconnecting line segments and polygonal faces.
- Initialization: Start with N + 1 initial points which give an initial simplex.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- Take steps - Either:
  - Reflect the simplex away from the high point
  - Reflect and expand the simplex away from the high point
  - Contract the simplex in one dimension from the high point
  - Contract the simplex along all dimensions toward a low point.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ The Nelder-Mead Simplex Method (cont)
  - Termination - Either:
    - Stop when the distance moved in a step is less than some tolerance
    - Stop when the decrease in function value during a step is less than some tolerance
  - Restart at minimum

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Nelder-Mead Simplex Method discussion is based on Chapter 10 of Numerical Recipes in C (2nd Edition) by William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution

- A **Genetic Algorithm** is a probabilistic search technique that makes use of the principles of genetics.
- An **Evolutionary Strategy** is a form of Genetic Algorithm that is especially useful in minimizing functions with real variables and many local minima.
- Differential Evolution is an **Evolutionary Strategy**

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ Differential Evolution (cont)
  - Evolutionary Strategies:
    - Encode values as floating point numbers instead of translating them into a special alphabet.
    - Usually use a deterministic selection scheme during crossover.
    - Use addition instead of symbol-swapping during mutation. This makes incremental search easier.

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- Initialization:
  - Form two arrays of NP, D-dimensional, real-valued, vectors - current population and next generation.
  - Set reasonable limits on the parameter values.
  - Generate the initial population and evaluate the ***cost*** of each member. Store these values in a ***cost*** array.
  - Each member of the current population takes its turn as the ***target*** vector for mutation, crossover, and selection

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- Initialization (cont):
  - Assign F, the scaling factor. ($0 < F \leq 1.2$)
  - Assign CR, the crossover constant. ($0 \leq CR \leq 1$)
- Mutation:
  - Determine the difference between two randomly chosen population members. $\mathbf{x}_a - \mathbf{x}_b$
  - Multiply by the scaling factor F. $F^*(\mathbf{x}_a - \mathbf{x}_b)$

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- Mutation (cont):
  - Add to a randomly chosen vector $\mathbf{x}_c$ to form the ‘mutated’ vector: $\mathbf{x}_c' = \mathbf{x}_c + F^*(\mathbf{x}_a - \mathbf{x}_b)$.
- Crossover:
  - Produce the trial vector $\mathbf{x}_t$ by combining the target with $\mathbf{x}_c'$.
  - Conduct D-1 experiments starting at a randomly selected parameter.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 APRIL 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ Differential Evolution (cont)
  - Crossover (cont):
    - For each parameter, compare CR to a uniformly distributed random number from 0 to 1.
    - If the number > CR then the trial vector's parameter comes from the target.

MERCERIENSIS UNIVERSITATIS SIGILLUM MDCCCXXXVIII

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 APRIL 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ Differential Evolution (cont)
  - Crossover (cont):
    - If the number ≤ CR then the trial vector's parameter comes from $\mathbf{x}_c'$.
    - The last parameter always comes from $\mathbf{x}_c'$ in order to make sure that the trial vector differs from the target.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 APRIL 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- Selection:
  - The cost of each trial vector is compared to the cost of the target.
  - The vector with the lowest cost is moved to the next generation array. If the trial vector wins, its cost replaces its parent's cost in the cost array.

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 APRIL 1998

COMPARISION OF DIFFERENTIAL EVOLUTION
TO THE SIMPLEX METHOD IN OPTIMIZATION
DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- The Next Generation:
  - After each vector in the current array has been targeted, the next generation array becomes the current array.
  - The old current array is used to hold the results of the next round of trials.
- Termination:
  - Trials continue for a pre-set number of iterations or until the cost of the best vector is less than or equal to a pre-set value.

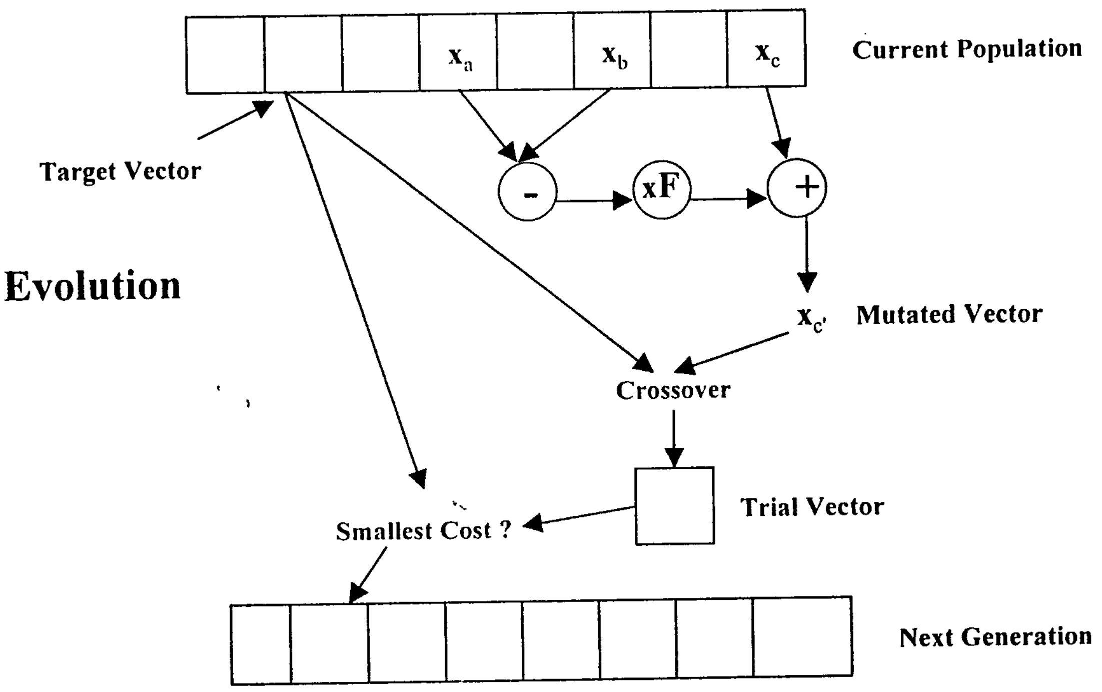
ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

Differential Evolution



ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

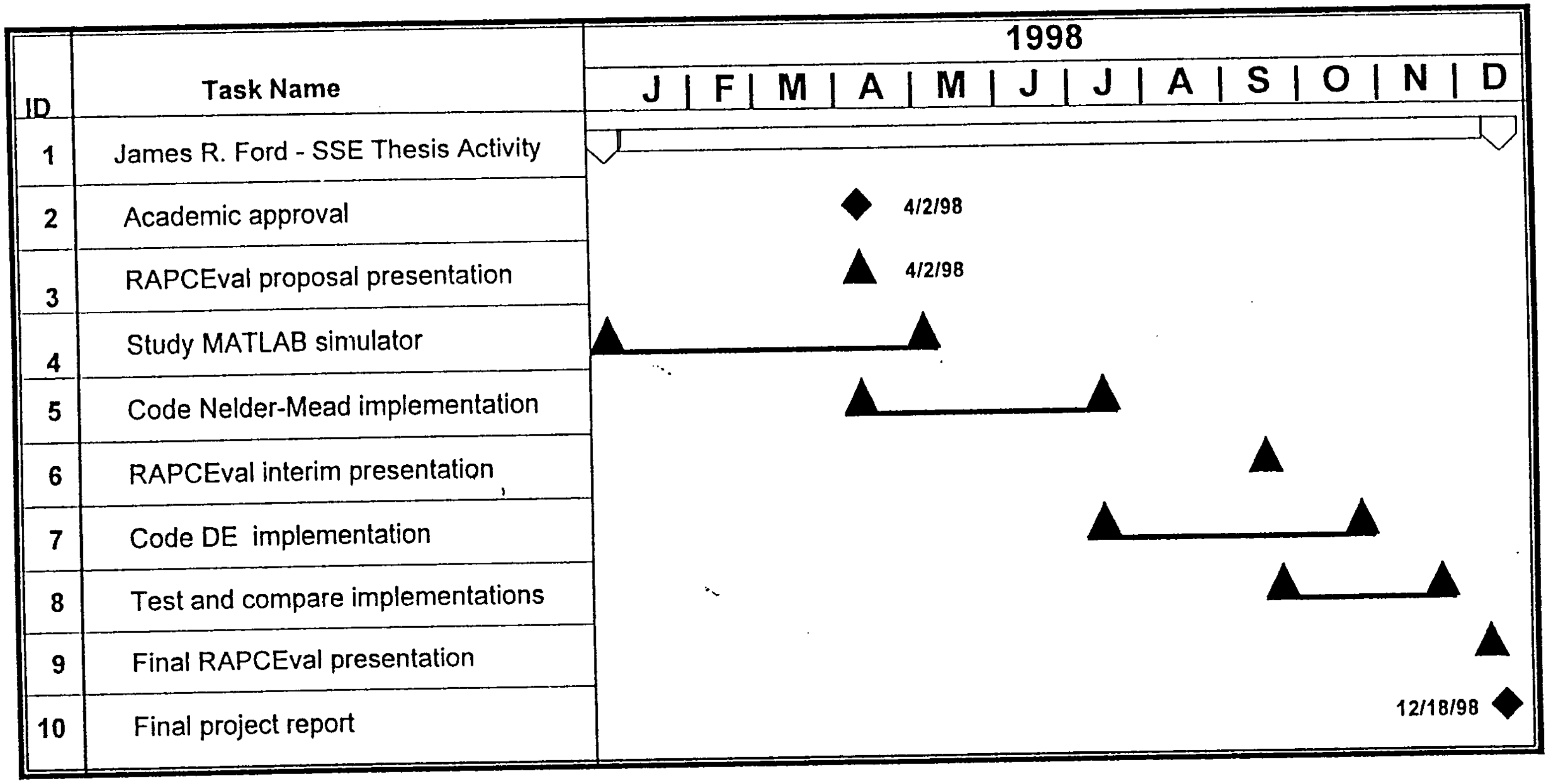
★ Differential Evolution discussion is based on "Differential Evolution" by Kenneth Price and Rainer Storn, Dr. Dobb's Journal, April 1997, Issue #264

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
2 April 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

Schedule Chart



5.1.8 Presentation by Dennis Ludwig

The student briefing presented by Dennis Ludwig at this meeting is reproduced on the next 12 pages.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-1

RESEARCH PROPOSAL PRESENTATION

Dennis A. Ludwig

WR-ALC/LNERB
ALQ-155 System Engineer

Background and Experience:

Education:	BS EE from Louisiana Tech, 1981
	MSA from Georgia College, 1988
	Pursuing MSEE from Mercer, 45 hrs. completed
WR/ALC:	Electronic/Software Engineer

Research Topic:
Pose estimation using a Neural Network

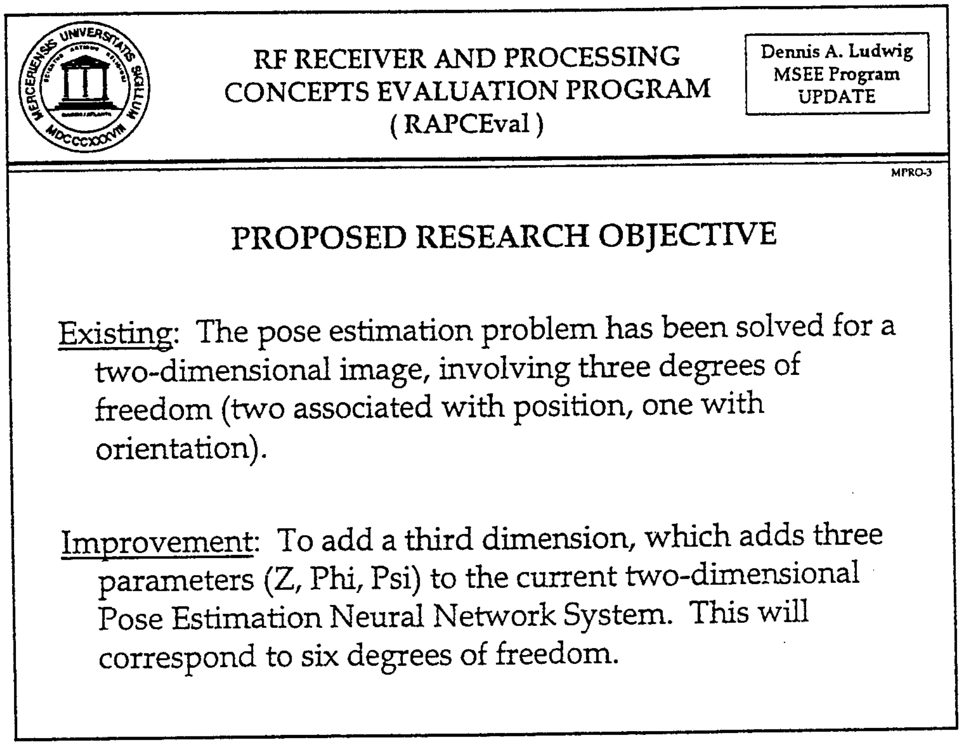
RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-2

PROBLEM STATEMENT

- ⋆ Design a Neural Network system to resolve the Pose Estimation Problem of a target/object in three dimensions. That is the estimation of location and orientation of a detected target in a Pose volume.
- ⋆ Given an image of an object in a background, and a model that we would like to match to the object, what is the best pose for the model? (Ghee and Doria)



RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

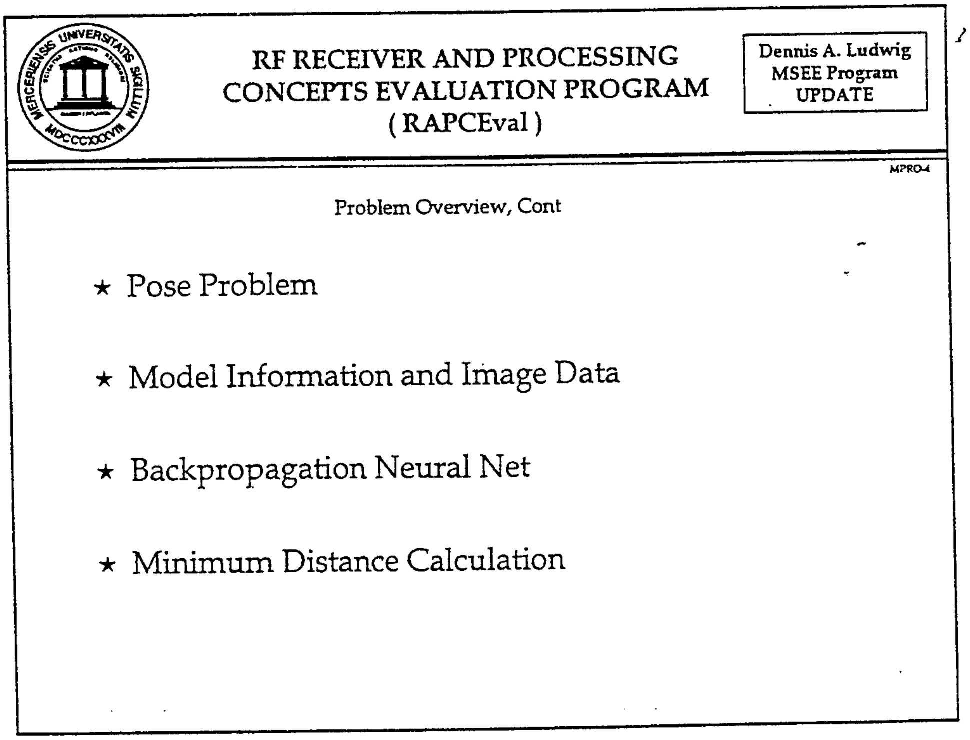
Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-3

PROPOSED RESEARCH OBJECTIVE

Existing: The pose estimation problem has been solved for a two-dimensional image, involving three degrees of freedom (two associated with position, one with orientation).

Improvement: To add a third dimension, which adds three parameters (Z, Phi, Psi) to the current two-dimensional Pose Estimation Neural Network System. This will correspond to six degrees of freedom.



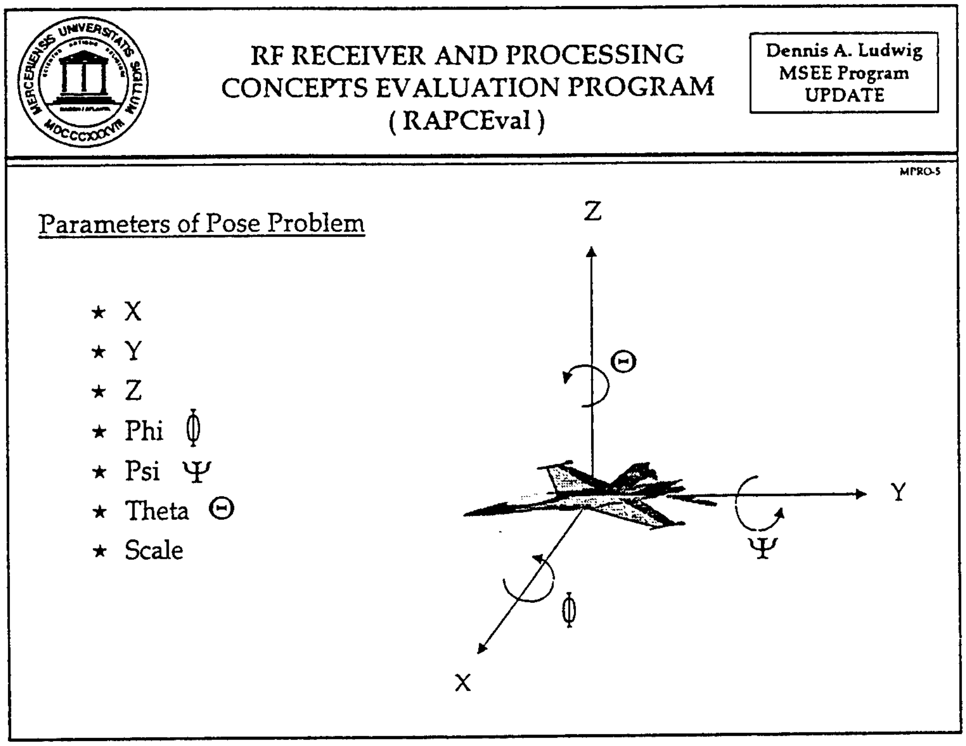
RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-4

Problem Overview, Cont

- ⋆ Pose Problem
- ⋆ Model Information and Image Data
- ⋆ Backpropagation Neural Net
- ⋆ Minimum Distance Calculation



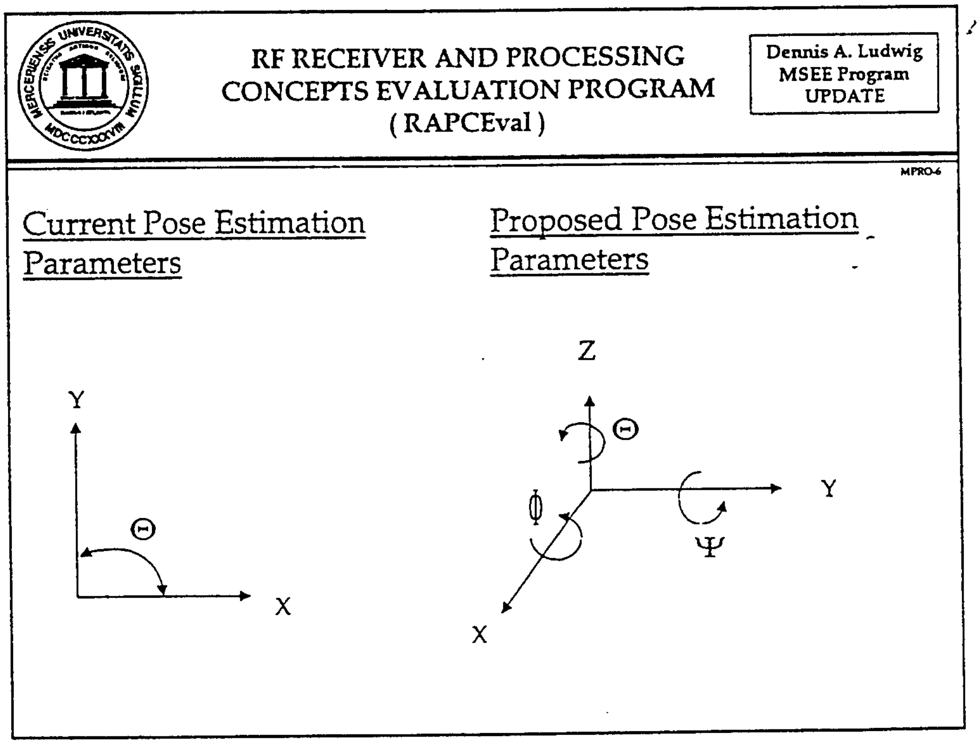
RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-5

Parameters of Pose Problem

- ★ X
- ★ Y
- ★ Z
- ★ Phi Φ
- ★ Psi Ψ
- ★ Theta Θ
- ★ Scale



RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-6

Current Pose Estimation Parameters

Proposed Pose Estimation Parameters

RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-7

TECHNICAL SECTION

3-D Image Representation

- Three degrees of freedom for translation in the X, Y and Z directions
- Three degrees of freedom for rotation or orientation about the X, Y, or Z axis (Φ , Ψ , Θ) respectively.
- Scale will be incorporated into a distance.

★ Note that 3-D Pose volume has 6 parameters describing it.

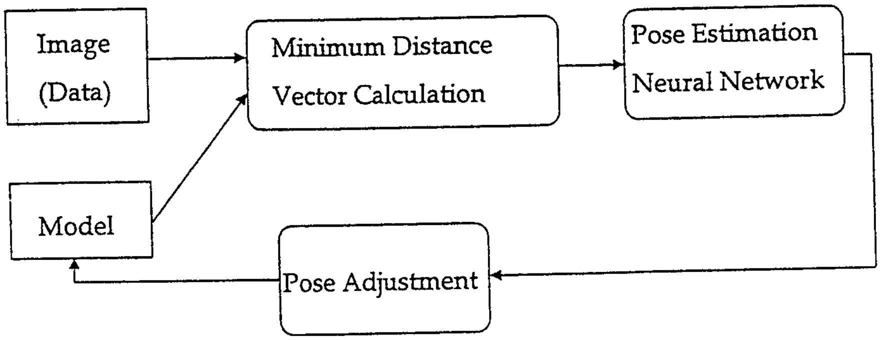
RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-8

TECHNICAL SECTION

System Overview



RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-9

TECHNICAL SECTION

Model

- An imitation of an existing object
- Contains a priori knowledge of the feature
- Is encoded in traditional data structure (two or three dimensional array)
- Referred to as a *synthetic image*

RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

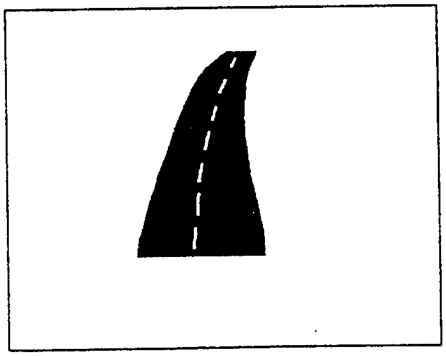
MPRO-10

TECHNICAL SECTION: MODEL CONT.

Example of a synthetic image

A synthetic image encoding the knowledge (perspective data) of a road.

Data structure is two dimensional array of pixels.

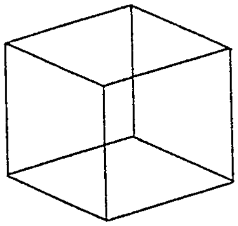


RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-11

TECHNICAL SECTION: MODEL CONT.



1, 1, 1
1, 1,-1
-1, 1,-1
-1, 1, 1
1,-1, 1
1,-1,-1
-1,-1,-1
-1,-1, 1

Perspective Inversion makes it difficult to see which of the opposite corners of the cube is the nearer.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-12

TECHNICAL SECTION

Neural Network

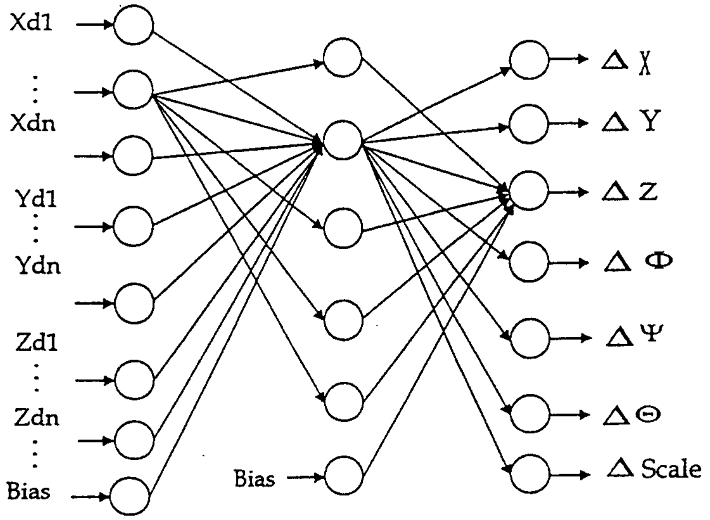
- ★ Architecture (MLFN) (MLP) (Backpropagation)
- ★ Type of Learning Procedure (Supervised)
- ★ A set of training patterns is used to teach a NN.
  - Input/Output vectors
  - Adjust weights based on output error
- ★ Each parameter variation will have an output node corresponding to it.
- ★ $(\Delta \chi, \Delta Y, \Delta Z, \Delta \Phi, \Delta \Psi, \Delta \Theta)$

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-13

Sample Architecture of BPN Network

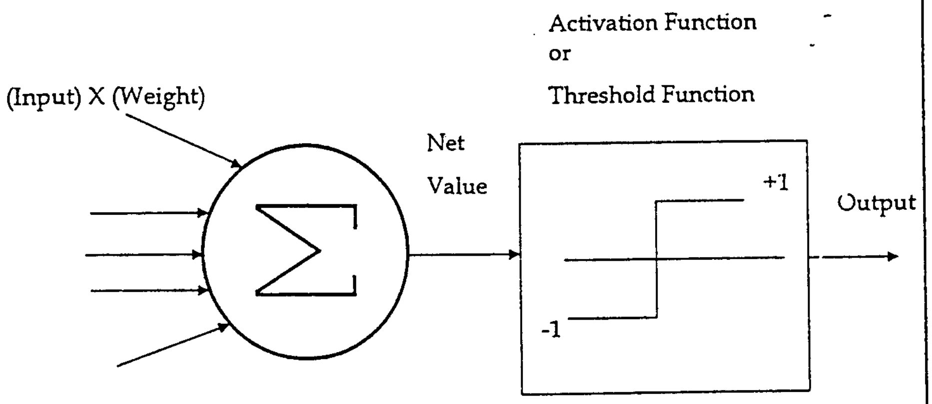


RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-14

TECHNICAL SECTION



A neural network node showing activation function

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-15

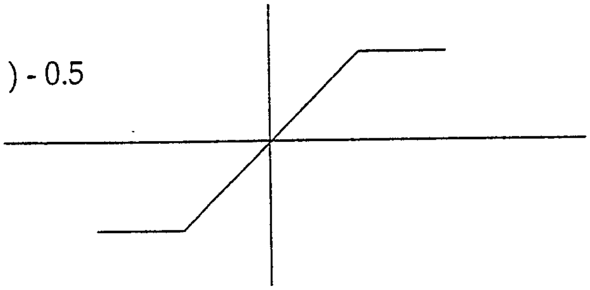
TECHNICAL SECTION

Activation Function

- ★ Sigmoid or Logistics function often used
- ★ Must be differentiable to use Backpropagation

$$f(x) = (1/1+\exp(-bx)) - 0.5$$

$$f'(x) = f(x)(1 - f(x))$$



RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-16

TECHNICAL SECTION: NEURAL NETWORK CONT.

A typical weight file for Five input nodes, Two hidden nodes, and Six output nodes:

O(1,0) O(1,1) O(1,2) O(1,3)
O(2,0) O(2,1) O(2,2) O(2,3)
O(3,0) O(3,1) O(3,2) O(3,3)
O(4,0) O(4,1) O(4,2) O(4,3)
O(5,0) O(5,1) O(5,2) O(5,3)
O(6,0) O(6,1) O(6,2) O(6,3)
H(0,0) H(0,1) H(0,2) H(0,3) H(0,4) H(0,5)
H(1,0) H(1,1) H(1,2) H(1,3) H(1,4) H(1,5)

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-17

TECHNICAL SECTION: NEURAL NETWORK CONT.

Parameters:

- ★ Learning Rate, can be different for each layer
- ★ Momentum, can be different value for each layer
- ★ Activation Function Temperature
- ★ Bias
- ★ Number of patterns in Epoch.

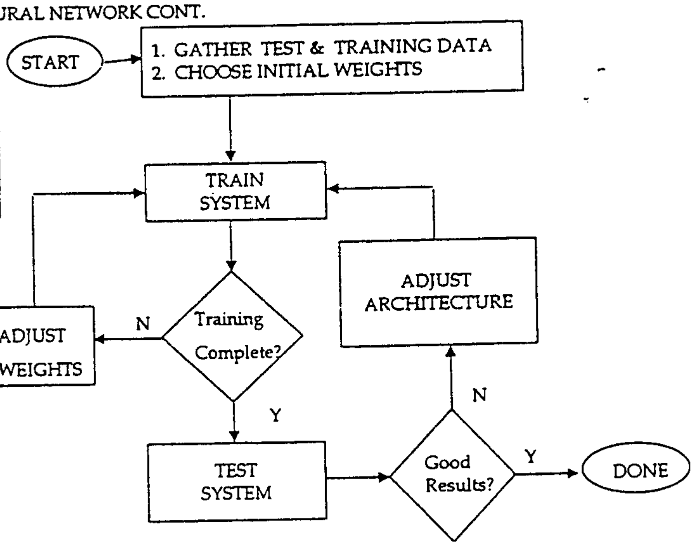
RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

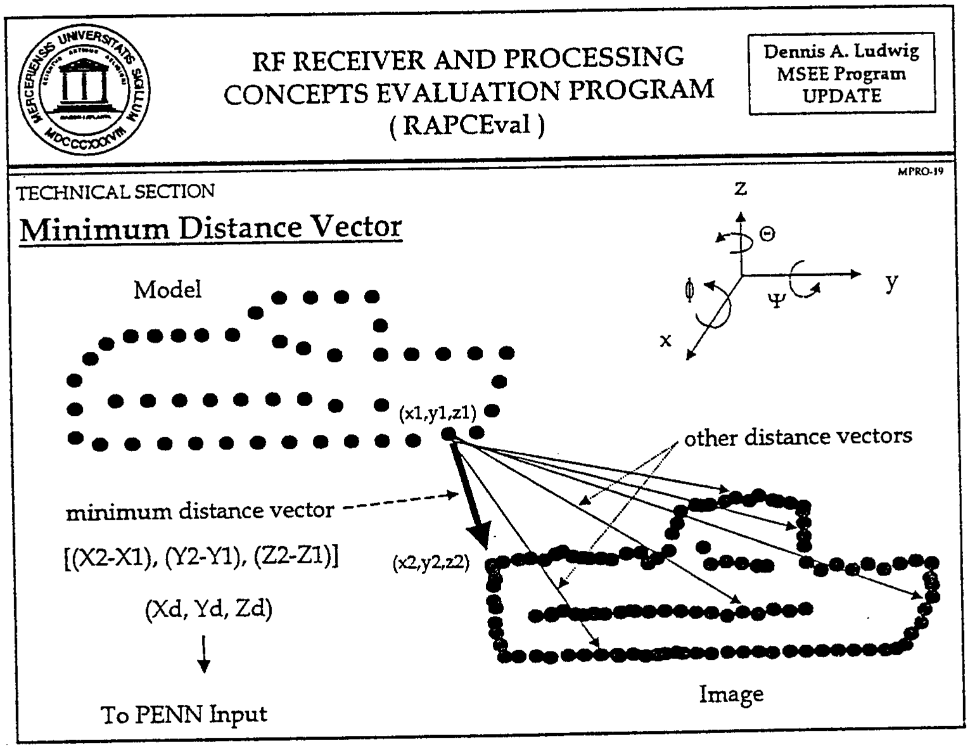
Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-18

TECHNICAL SECTION: NEURAL NETWORK CONT.

Designing a Backpropagation Network





RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-19

TECHNICAL SECTION

Minimum Distance Vector

minimum distance vector
[(X2-X1), (Y2-Y1), (Z2-Z1)]
(Xd, Yd, Zd)
↓
To PENN Input

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-20

TECHNICAL SECTION: NEURAL NETWORK CONT.

Data Collection for Training

- ⋆ Set model in pose volume and measure sampled points from focal point.
- ⋆ Re-orient the model and measure sampled points from focal point.
- ⋆ Determine the new pose given those Δ-parameters (outputs) and the resultant minimum distance vectors (inputs)

RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

TECHNICAL SECTION: NEURAL NETWORK CONT.

Training and Testing the Network

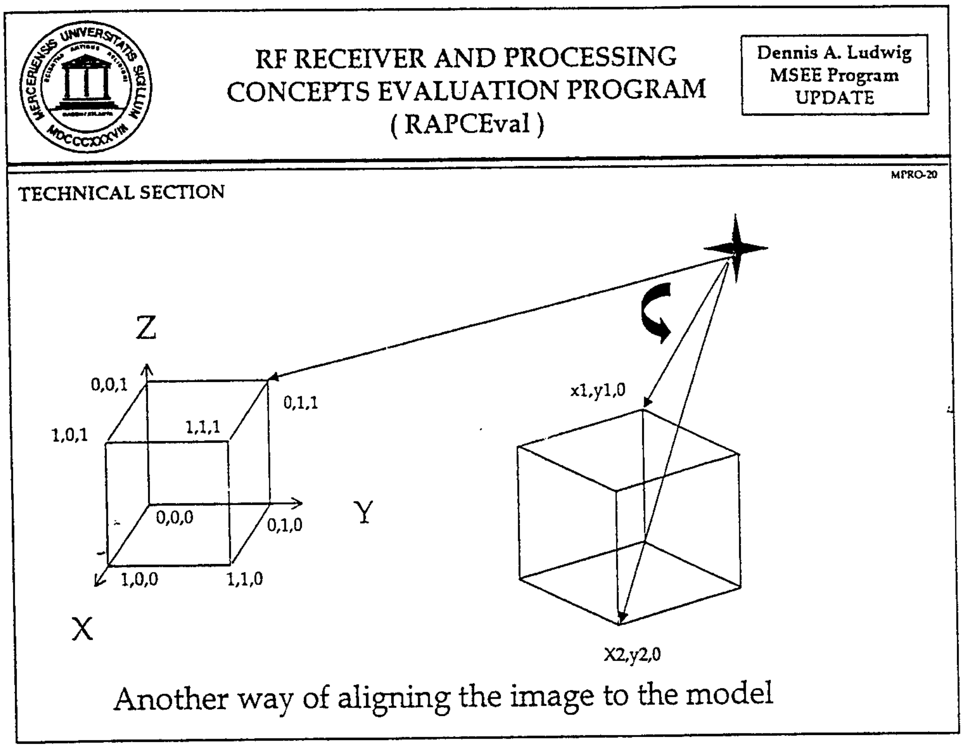
- ★ Data is divided into training and testing sets
- ★ Training occurs using collected data
- ★ Trained NN is then evaluated
- ★ Adjustments made if results are not acceptable

RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

PROPOSED SCHEDULE

- ★ Quarter 1
  - Develop expertise in data modeling and BRL-CAD software (Didn't use)
  - Develop expertise in neural network design package (Didn't use, wrote my own)
  - Develop initial solution strategy for proposed problem
- ★ Quarter 2 (Current Semester)
  - Develop and test backpropagation network
  - Document results



RF RECEIVER AND PROCESSING
CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Dennis A. Ludwig
MSEE Program
UPDATE

MPRO-20

TECHNICAL SECTION

Another way of aligning the image to the model

I could get the model to move to the image by minimizing an angle between the light source ray tracing the minimum and maximum points of the two images. The model point is well defined, but the Z coordinate will be missing from the target image.

Look at rotating an enclosed area to get Z information.

5.1.9 Presentation by Tracy Tillman

The student briefing presented by Tracy Tillman at this meeting is reproduced on the next 30 pages.

RECEIVER AND PROCESSOR CONCEPTS EVALUATION PROGRAM (RAPCEval)

RESEARCH PROJECT UPDATE

Tracy J. Tillman

Mercer Engineering Research Center (MERC)

Electronic Systems Division Manager

Date Approved:	March 1995
Projected Completion Date:	September 1998

Research Topic:

HARDWARE IMPLEMENTATION for ADVANCED PULSE PROCESSING

RECEIVER AND PROCESSOR CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

RESEARCH OBJECTIVES

- ★ Develop Hardware Implementation for Candidate PRP Selection
- ★ Develop/Optimize Lookup Table Generation for Candidate PRP Selection
- ★ Quantify Speed Achieved Through Hardware Implementation
- ★ Establish Hardware Requirements
- ★ Develop Specification for Hardware Implementation

UNIVERSITATIS MERCERIENSIS SIGILLUM MDCCCXXXVIII

RECEIVER AND PROCESSOR CONCEPTS EVALUATION PROGRAM (RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

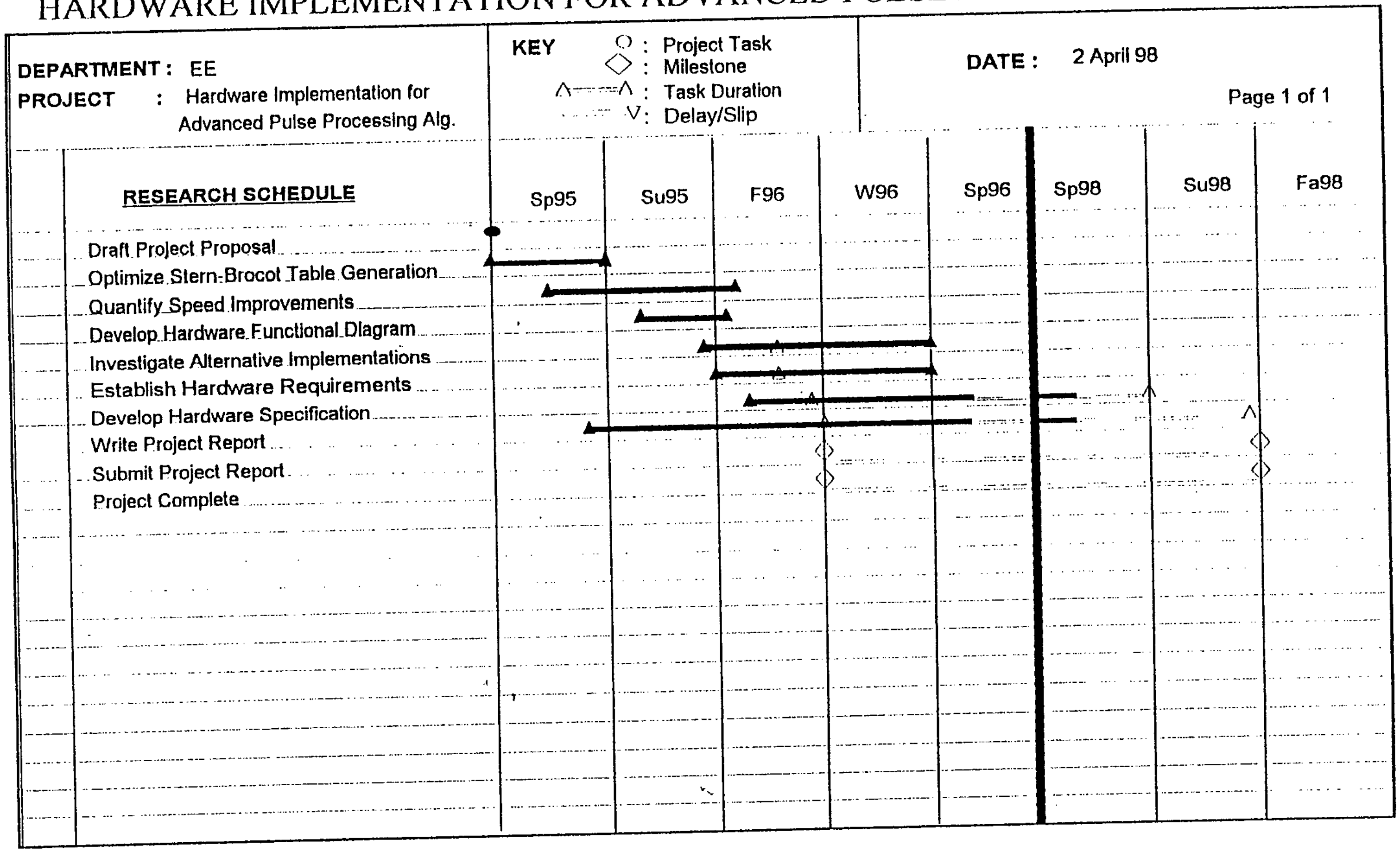
HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

DEPARTMENT : EE
PROJECT : Hardware Implementation for Advanced Pulse Processing Alg.

KEY
○ : Project Task
◇ : Milestone
Λ-----Λ : Task Duration
------V : Delay/Slip

DATE : 2 April 98

Page 1 of 1

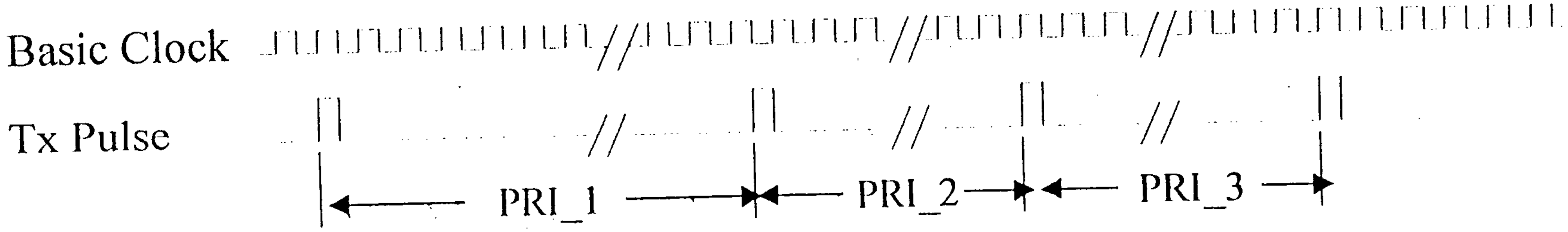


RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

TYPICAL TRANSMITTER TIMING



- •PRI_1/PRI_2 Less Than 2 (Typically)
- •PRI = Downcount X Clock Period
- •PRI_1 / DC_1 = PRI_2 / DC_2

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

★ "BRUTE FORCE" : PRI_X / N

N	PRI_1	PRI_2
	240.0	210.0
20	12.000	10.500
21	11.429	10.000
22	10.909	9.545
23	10.435	9.130
24	10.000	8.750
25	9.600	8.400

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

RECEIVER AND PROCESSOR CONCEPTS EVALUATION PROGRAM
(RAPCEval)

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

LOOKUP TABLE

- ★ RANGE OF INTEREST: $1 < \text{RATIO} < 2$
- ★ EACH ENTRY REPRESENTS A PRI RATIO
- ★ EACH ENTRY = RATIO OF INTEGERS
- ★ ENTRIES LINEARILY SPACED
- ★ RATIO USED TO CALCULATE TABLE INDEX
  - $11/10 = 1.1$
  - INDEX = 0.1 X TABLE LENGTH
- ★ DATA STORED: RATIO & SUM OF INTEGERS

RECEIVER AND PROCESSOR CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

STERN-BROCOT TREE

- ★ RELATIVELY PRIME ENTRIES
- ★ POSITIVE FRACTIONS FROM RATIONAL NUMBERS
- ★ NO ENTRIES REPEATED
- ★ STARTING ENTRIES

$$\frac{m}{n} = \frac{0}{1} \qquad \frac{m'}{n'} = \frac{1}{0}$$

- ★ NEW ENTRIES

$$\frac{m + m'}{n + n'}$$

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

STERN-BROCOT TREE SUBSET

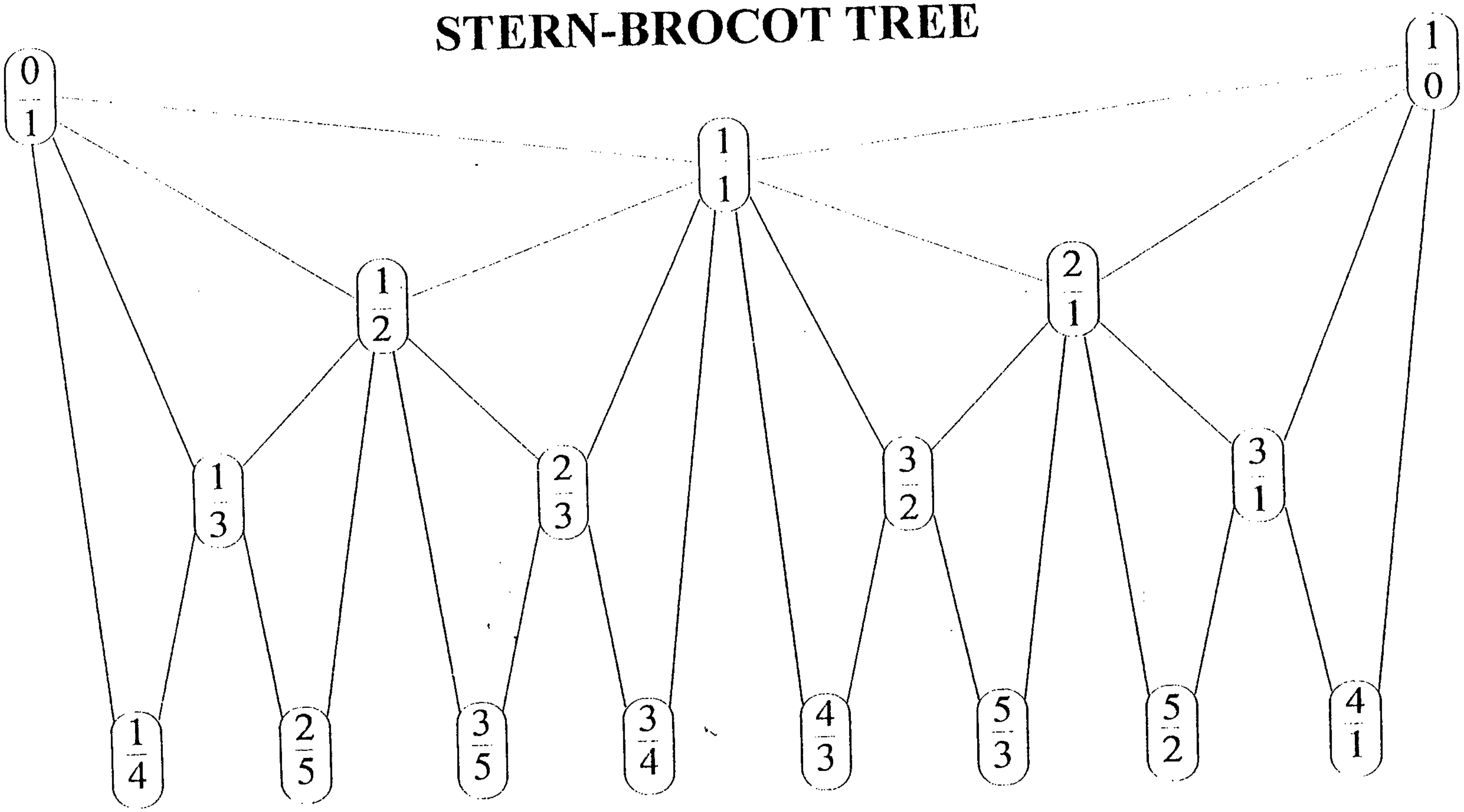
- ★ FAREY SERIES OF ORDER N (Largest Denominator in Series)
- ★ RANGE OF INTEREST: 1 < RATIO < 2
- ★ SYMMETRICAL ABOUT 1.5
- ★ ODD NUMBER OF ENTRIES IN BRANCH

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

STERN-BROCOT TREE



RECEIVER AND PROCESSOR CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

LOOK-UP TABLE GENERATION

- ★ RANGE OF ENTRIES: $1 < \text{RATIO} < 2$
  - N: LARGEST DENOMINATOR
  - M: 2N - 1
- ★ ARRANGE IN ORDER SMALLEST TO LARGEST
- ★ TABLE ENTRY
  - SUM OF INTEGERS
  - RATIO
- ★ N DETERMINES TABLE SIZE/NUMBER OF ENTRIES
  - No. Entries $= 3*N^2/\pi^2$ (approx.)
  - $N = 100$: 3043 ENTRIES; 18K
  - $N = 300$: 27,397 ENTRIES; 162K

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

SAMPLE LOOK-UP TABLE

N = 10 - 31 ENTRIES

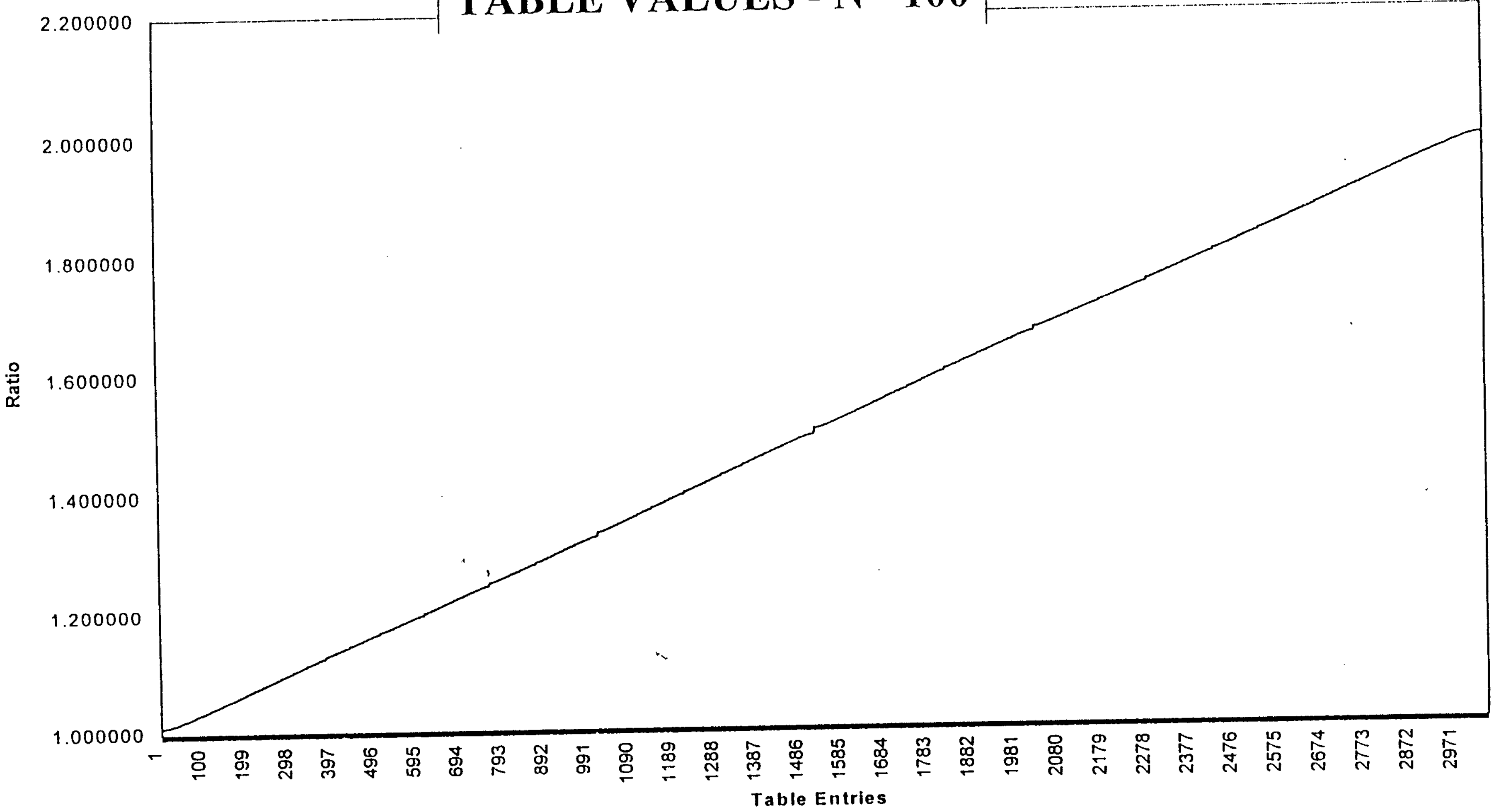
10	11	1.100000
9	10	1.111110
8	9	1.125000
7	8	1.142860
6	7	1.166670
5	6	1.200000
9	11	1.222220
4	5	1.250000
7	9	1.285710
10	13	1.300000
3	4	1.333330
8	11	1.375000
5	7	1.400000
7	10	1.428570
9	13	1.444440
2	3	1.500000
9	14	1.555560
7	11	1.571430
5	8	1.600000
8	13	1.625000
3	5	1.666670
10	17	1.700000
7	12	1.714290
4	7	1.750000
9	16	1.777780
5	9	1.800000
6	11	1.833330
7	13	1.857140
8	15	1.875000
9	17	1.888890
10	19	1.900000

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

TABLE VALUES - N =100



UNIVERSITATIS MERCERIENSIS SIGILLUM MDCCCXXXVIII

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

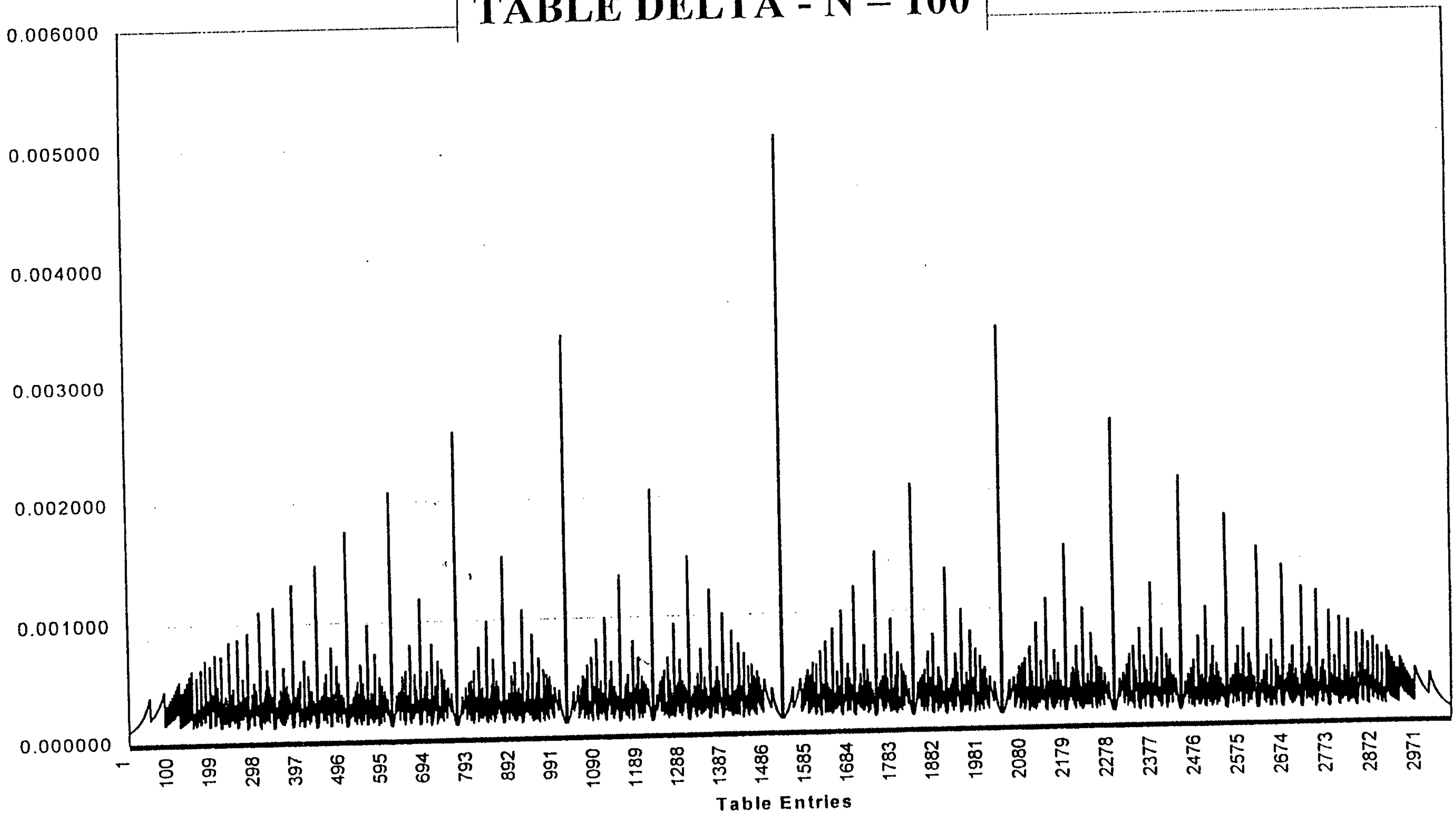


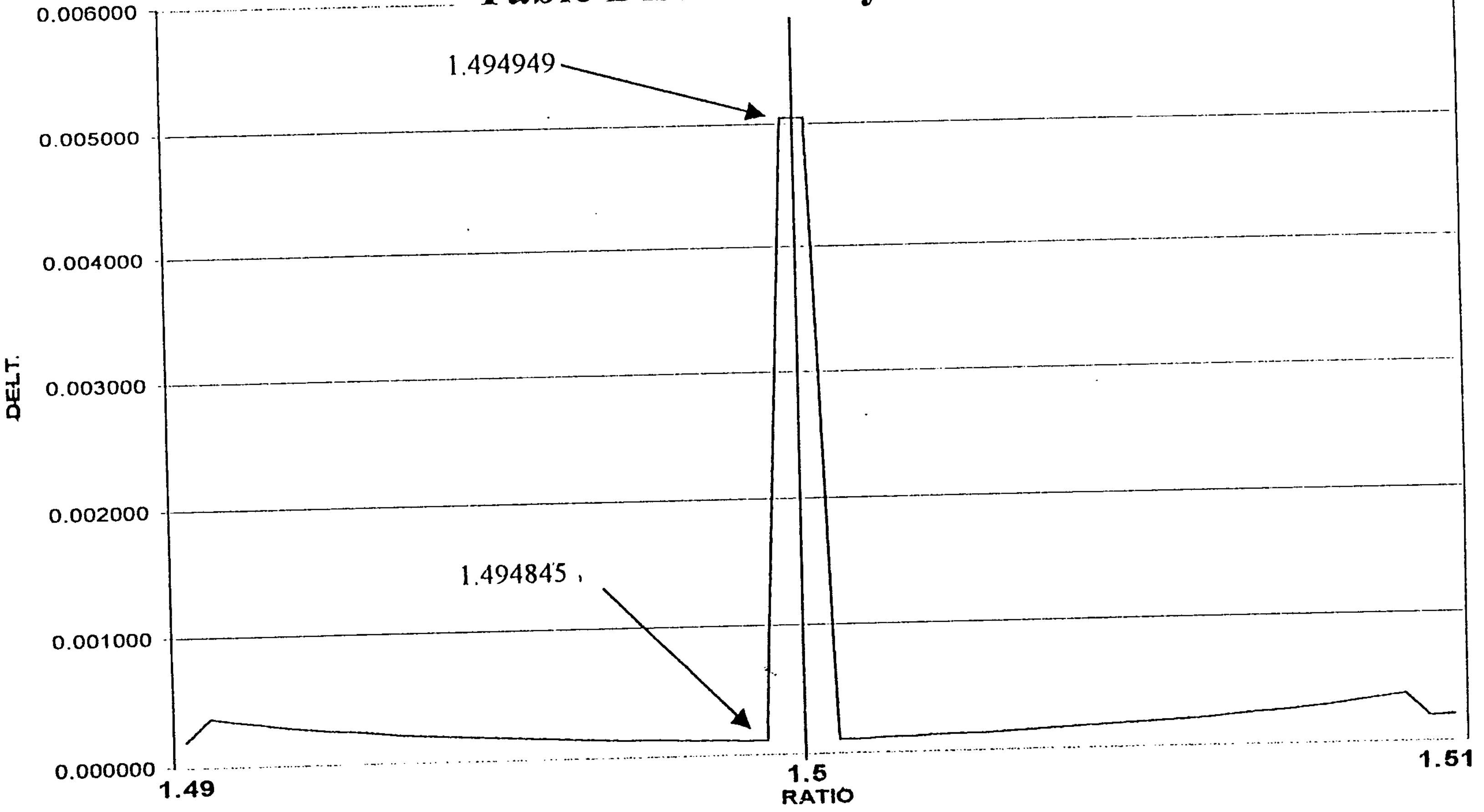
TABLE DELTA - N = 100

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

Table Discontinuity - N = 100



RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

TABLE SIZE COMPARISON

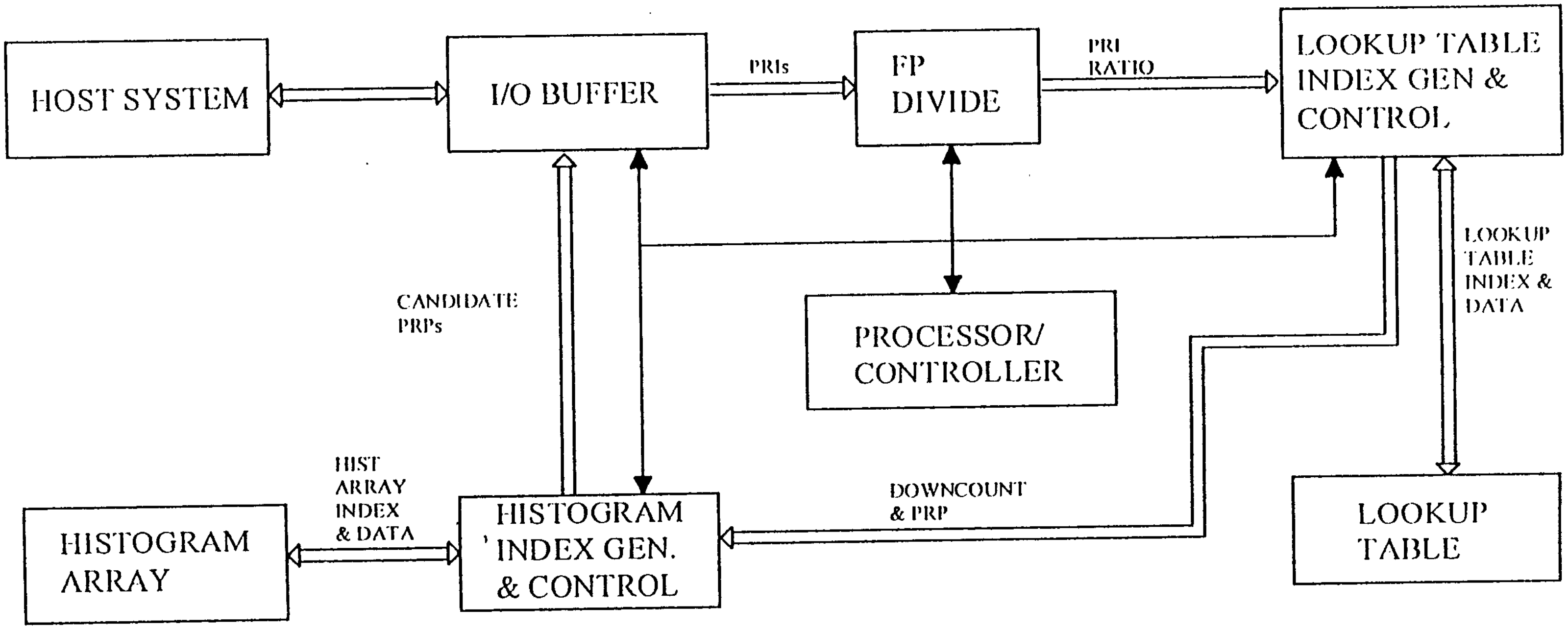
N	No ENTRIES	MEM SIZE (bytes)
100	3043	18258
200	12231	73386
300	27397	164382
400	48667	292002
500	76115	456690 ½ Meg

RECEIVER AND PROCESSOR CONCEPTS EVALUATION PROGRAM (RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

FUNCTIONAL DIAGRAM



RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PULSE PROCESSING

★ MIN / MAX RATIO

- $\text{ratio\_min} = \dfrac{\text{pri\_max} - \text{err}}{\text{pri\_min} + \text{err}}$
- $\text{ratio\_max} = \dfrac{\text{pri\_max} + \text{err}}{\text{pri\_min} - \text{err}}$

★ INITIAL TABLE INDICES

- index_min = int(ratio_min - 1 X table_length)
- index_max = int(ratio_max - 1 X table_length)

★ TABLE OUTPUT

- ratio_min <= ratio of table entry <= ratio_max
- candidate countdown pairs/PRPs

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

ABC PULSE ANALYSIS

★ INPUT: 90.0 & 110.0

★ OUTPUT:

INTEGERS		PRP
91	111	0.9890
50	61	1.8000
59	72	1.5254
68	83	1.3235
77	94	1.1688
86	105	1.0465
95	116	0.9474
9	11	10.0000 ⇦
94	115	0.9574
85	104	1.0588
76	93	1.1842
67	82	1.3433
58	71	1.5517
49	60	1.8367

★91 EQUIVALENT FP ADDITIONS

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

ABC PULSE ANALYSIS

- ★ INPUT: 173.3 & 193.3
- ★ OUTPUT:

INTEGERS		PRP
35	39	4.9514
96	107	1.8052
61	68	2.8410
87	97	1.9920
26	29	6.6654 ⇦
95	106	1.8242
69	77	2.5116
43	48	4.0302

- ★ 59 EQUIVALENT FP ADDITIONS

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

ABC PULSE ANALYSIS

★ INPUT: 173.3 & 200.0

★ OUTPUT:

INTEGERS		PRP	
85	98	2.0388	
98	113	1.7684	
13	15	13.3308	⇦
97	112	1.7866	
84	97	2.0631	
71	82	2.4408	
58	67	2.9879	

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

Down Counts Greater Than N

N = 100		
58	57	6.4922
57	56	6.6071
56	55	6.7261

N = 200		
176	173	2.1393
117	115	3.2181
175	172	2.1516
58	57	6.4922
173	170	2.1767
115	113	3.2746
172	169	2.1894
57	56	6.6071
170	176	2.2154
113	111	3.3330
169	166	2.2287
56	55	6.7261
167	164	2.2556
111	109	3.3936

Input Data:

PRI=1: 370, DC 111

PRI-2: 376.6, DC 113

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PERFORMANCE COMPARISON

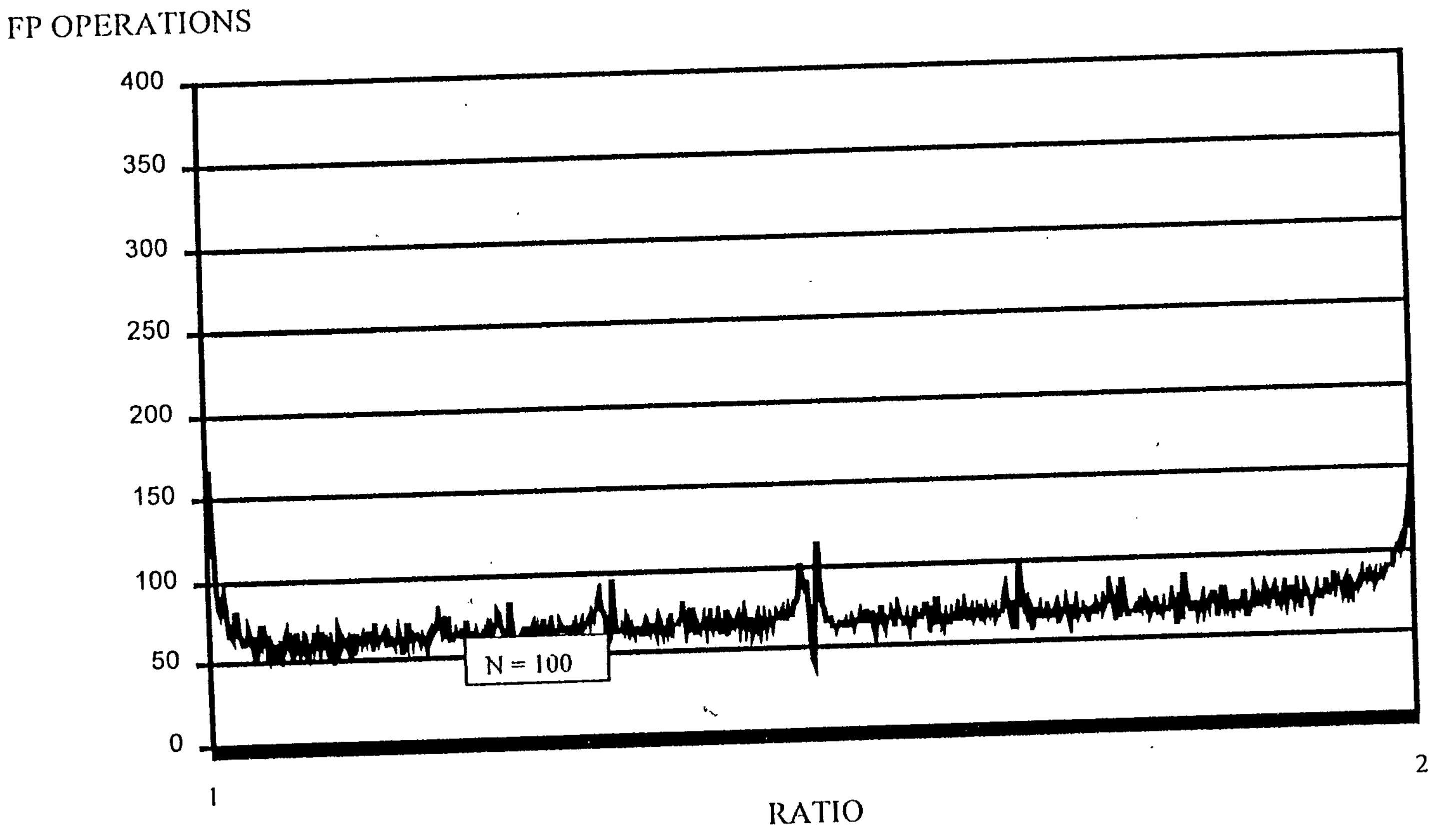
Table Base / PRIs	100		200		300	
	PRP	OPS	PRP	OPS	PRP	OPS
173.3 - 193.3	8	59	29	150	67	327
186.7 - 206.7	8	79	28	159	64	307
173.3 - 200	8	87	30	167	68	319
166.7 - 213.3	11	95	36	187	77	351
126.7 - 246.7	19	131	57	303	128	575

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PERFORMANCE COMPARISON (CONT)

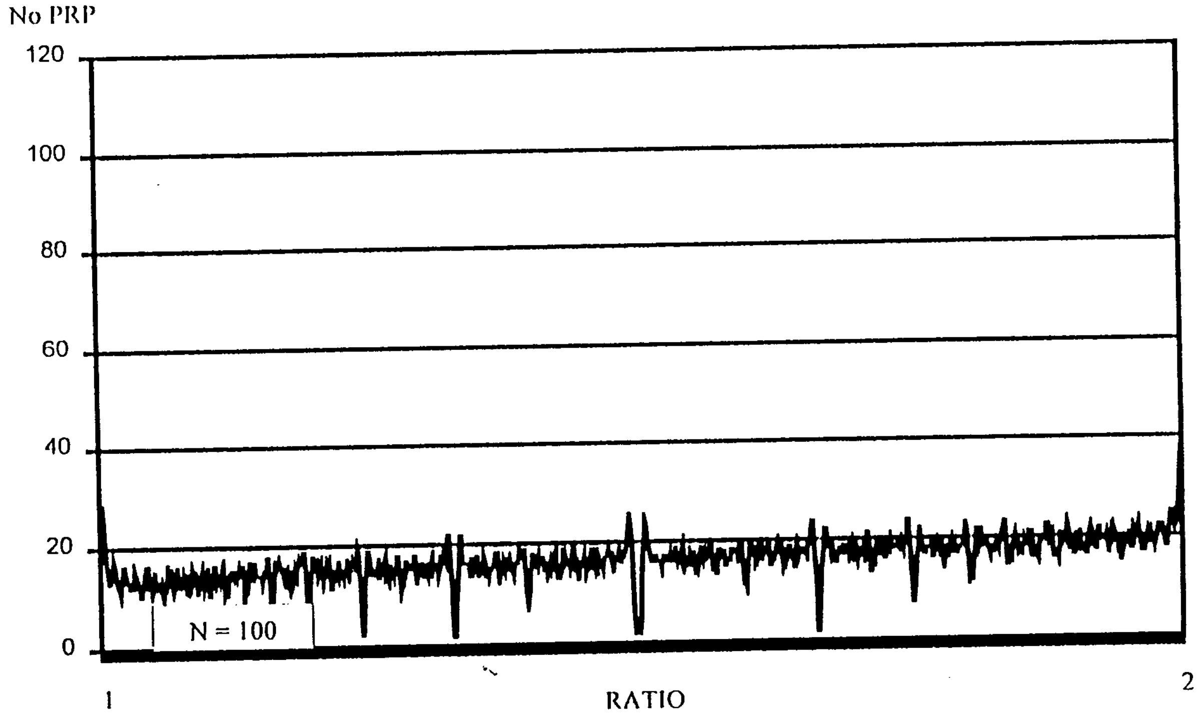


RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PERFORMANCE COMPARISON (CONT)



RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

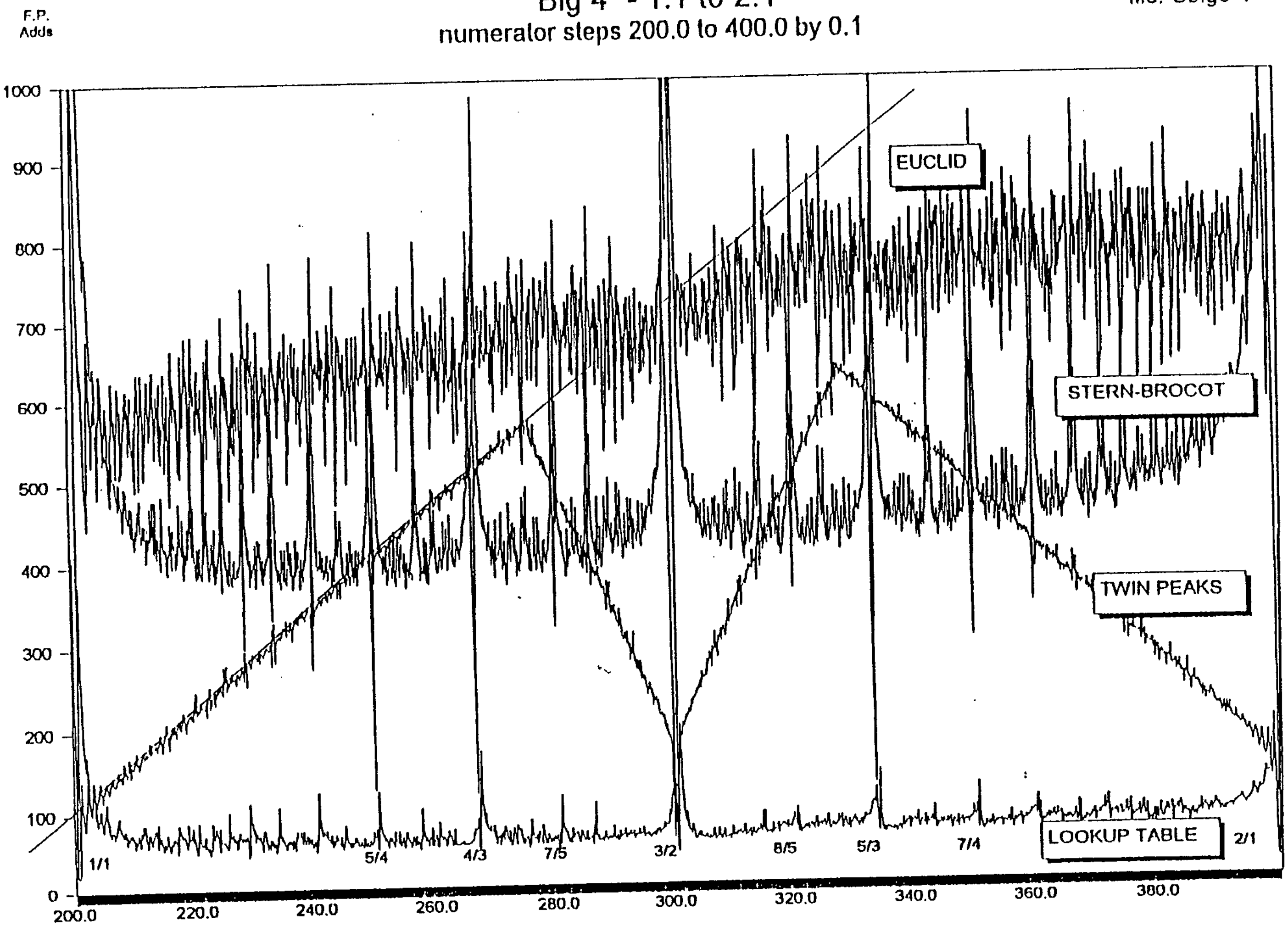
Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PERFORMANCE COMPARISON

"Big 4" - 1:1 to 2:1
numerator steps 200.0 to 400.0 by 0.1

file: Cbig3-1



RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PULSE TRAIN ANALYSIS

- ★ HISTOGRAM ABC PULSE DATA
- ★ HISTOGRAM TABLE
  - ZEROED FOR EACH PULSE TRAIN
  - INDEXED TO LOOK-UP TABLE
  - NUMBER OF OCCURRENCES OF EACH TABLE ENTRY
- ★ OUTPUT: VALUES ABOVE THRESHOLD

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PULSE TRAIN ANALYSIS

★ INPUT PULSE TRAINS

- NO. 1
    - 110 - 120
    - 110 - 130
    - 120 - 130
    - 90 - 110
- NO. 2
    - 173.3 - 193.3
    - 186.7 - 206.7
    - 173.3 - 200.0
    - 166.7 - 213.3
    - 126.7 - 246.7

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PULSE TRAIN ANALYSIS

- OUTPUT DATA - PULSE TRAIN NO. 1

PRP	COUNT
1.34	2
1.41	2
1.44	2
1.52	2
1.54	2
1.64	2
1.69	2
10.00	4

- 652 EQUIVALENT FP ADDITIONS

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Tracy Tillman
MSEE Program
Research Proj Update
2 April 98

HARDWARE IMPLEMENTATION FOR ADVANCED PULSE PROCESSING ALGORITHM

PULSE TRAIN ANALYSIS

- ★ OUTPUT DATA - PULSE TRAIN NO. 2

PRP	COUNT
1.27	2
1.29	2
1.30	2
1.31	2
1.34	2
1.56	2
1.60	2
1.64	2
1.68	2
2.03	2
2.14	2
3.24	2
6.33	2
6.66	4

- ★ 721 EQUIVALENT FP ADDITIONS

SUMMARY

- ★ OPTIMIZATION OF LOOK-UP TABLE COMPLETE
- ★ HARDWARE REQUIREMENS DEFINED
- ★ LOOK-UP TABLE REDUCES PROCESSING TIME
- ★ PROVIDES DATA ON PAIRS OF PULSES AND PULSE TRAINS
- ★ AREAS OF FURTHER RESEARCH
  - REAL TIME EXPANSION OF TABLE
  - LARGE COUNTDOWN/SMALL PRPs
  - PULSE TRAIN ANALYSIS WITH LARGE TABLES

5.2 Steering Committee Meeting, August 1998

5.2.1 Meeting Minutes

EW Receiver and Processing Concepts Evaluation Program
Program Research Standards Committee Meeting
Minutes August 12, 1998

A meeting of the RAPCEval Advisory Committee was held in the Mercer Engineering Research Center Auditorium at 1:00 p.m., August 12, 1998.

The meeting consisted primarily of an invited lecture by James P. Stephens, Sr.. Dr. Stevens works in the RF Sensor Technology Division, Electronic Warfare Technology Branch, of the Air Force Research Laboratory in Dayton, Ohio. The lecture focused on the topic of RF sensor technologies in the area of C3 Electronic Combat.

The lecture had a communication countermeasure emphasis, focusing on advanced digital signal processing for EW. It began with an organizational background and objectives, C2W trends, and vision from the perspective of SNRW.

A brief tutorial was provided to explain the motivation of the use of time-frequency distributions, wavelets, and cyclostationary signal processing. Current research activities of SNRW in all areas related to C3 Electronic Combat were introduced. These include such areas as featureless waveform countermeasures, time-frequency analysis, frequency-hopping prediction, GPS ECM and ECCM, and covert waveform detection and jamming. In-house activities include time-difference of arrival using cyclostationarity, frequency hopping synthesizer development, spectral correlation analysis, and transform domain waveform experimentation.

5.2.2 Attendance Roster

Name (last)	Name (First)	Organization	Phone	E-mail
Bahl	Erik	MERC	912-953-6800	ebahl@merc.mercer.edu
Bass	Charles	MERC	912-953-6800	cbass@merc.mercer.edu
Bass	Tom	MERC	912-953-6800	tbass@merc.mercer.edu
Black	Joseph K.	MERC	912-953-6800	jblack@merc.mercer.edu
Bryant	Peter	MERC	912-953-6800	pbryant@merc.mercer.edu
Campbell	Mark	WR-ALC/LYSFR	912-926-5673	mcampbel@avionics2.robins.af.mil
Carter	Glenn	MERC	912-953-6800	gcarter@merc.mercer.edu
DePuy	George	MERC	912-953-6800	gdepuy@merc.mercer.edu
Finnigan	Skip	MERC	912-953-6800	sfinnigan@merc.mercer.edu
Guest	Robert	MERC	912-953-6800	rguest@merc.mercer.edu
Harris	Rhondia	MERC	912-953-6800	rharris@merc.mercer.edu
Hillhouse	Ed	ITC	912-922-8106	hillhouse@itc-2.com
Jones	Houston	TASC	912-926-5458	jhjones@hom.net
Kelley	Joseph F.	WR-ALC/LNIE	912-926-4482	kelley.joe@ec.robins.af.mil
May	Mark	MERC	912-953-6800	mmay@merc.mercer.edu
McFarland	Chris	MERC	912-953-6800	cmcfarland@merc.mercer.edu
Messer	Bob	MERC	912-953-6800	bmesser@merc.mercer.edu
Miller	David	WR-ALC/LNEV	912-926-2994	miller@ec.robins.af.mil
Mitchell	Ray	MERC	912-953-6800	rmitchell@merc.mercer.edu
Moody	Douglas	SAIC	912-918-2911	moodyd@saic.com
Moody	Douglas	SAIC	912-918-2911	moddyd@saic.com
O'Dowd	Bill	Ball	912-922-4363	bodowd@f15aisf.robins.af.mil
Oliver	R.P	WR-ALC/LNERT	912-926-2588	oliver@ec.robins.af.m.il
Pequignot	Nick	AFRL/SNRD	937-255-6127	pequignot@sensors.af.mil
Potts	Marvin	AFRL/SNRW	937-255-5579	pottsmn@sensors.wpafb.af.mil
Smith	Jermaine	MERC	912-953-6800	jsmith@merc.mercer.edu
Smith	Jerome	WR-ALC/LNEV	912-926-2994	smithjm@ec.robins.af.mil
Thorstens	Gary	Ball	912-922-4363	gthorste@ball.com
Tillman	Tracy	MERC	912-953-6800	ttillman@merc.mercer.edu
Wimpy	Dale	WR-ALC/LYSFR	912-926-9529	dwimpy@f15aisf.robins.af.mil

5.2.3 Overview of the Program (Dr. Bass)

The Overview Briefing of the RAPCEval Program as presented at this meeting is reproduced on the next 11 pages.

RAPCEval MEETING

August 12, 1998

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

August
1998

PROJECT INFORMATION

- ★ *Contract:* F09603-93-G-0012-0017
- ★ *Customer:* Air Force Research Laboratory
- ★ *Program Status:*
  - ★ Graduate research jointly supported by Mercer, AFRL, WR-ALC, and Industry
  - ★ Successful research projects (10) have been completed; several (7) research projects are ongoing
  - ★ All research is determined by the steering Committee to be useful to the Air Force and is certified by the University as having academic merit

RESEARCH STANDARDS COMMITTEE

- ★ ***AFRL*** **:** Mr. Nick Pequignot (PMgr), Mr. Emil R. Martinsek, Mr. Norman A. Toto, Dr. Duane A. Warner, Mr. Paul J. Westcott
- ★ ***WR-ALC*** **:** Mr. Steve Strawn (PMgr), Mr. John LaVecchia, Mr. Phil Oliver, Mr. Ches Rehburg, Mr. Larry Sheets
- ★ ***MERC*** **:** Dr. David Barwick (Chairman), Dr. Tom Bass (PMgr)
- ★ ***MERCER U*** **:** Dr. Aaron Collins, Dr. Benham Kamali, Dr. Paul MacNeil

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

August 1998

NEW STUDENTS

- ★ *An Nguyen, advisor* **Dr. Tom Bass**
- ★ *Steve Boswell, advisor* **Dr. Paul MacNeil**
- ★ *Houston Jones, advisor* **Dr. Benham Kamali**

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

August 1998

RESEARCH IN PROGRESS

- ★ *Randy Ford,* “Passive Location via Evolutionary Genetic Algorithms”
- ★ *Dennis Ludwig,* “Identification of IR Images Using Neural Networks”
- ★ *Tracy Tillman,* “Hardware Exploitation of RAD Filtration”

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

August 1998

RESEARCH WAITING FOR FINAL APPROVAL OR PRINTING

- ★ *Henderson Benjamin,* “Selection of Reed Solomon Codes Using Neural Networks”
- ★ *Ron Brinkley,* “Burst Error Correction with Reed-Solomon Codes”

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

August
1998

GRADUATES

★ *Mark Astin,* “Application of Parallel Computing Techniques to the RAD Algorithm”, classified, draft in edit at AFRL

★ *Mark Campbell,* “Auto-Regressive Spectral Analysis - EW Applications” , unclassified, available from MERC

★ *Claus Franzkowiak,* “Four-Pulse Primary RAD Filter Development”, classified, draft in edit at AFRL

★ *Neal Garner,* “Error Correction and Prediction for Improved Communication of Time and Time Measurements”, unclassified, available from MERC

GRADUATES

- ★ *Joseph Kelley,* "A Parameter Determination Alternative for RAD Analysis", classified, WL-TR-95-1005
- ★ *Joseph Kelley,* "MultiGroup Simultaneous RAD Parameter Selection", classified, WL-TR-97-1094
- ★ *Max Roesel,* "Agile RF/PRI Radar Analysis via RAD", classified, WL-TR-95-1020
- ★ *Dave Schuler,* "Comparison of Algorithms for Geolocation of Radar Signals", available from MERC on need to know
- ★ *Kirk Wright,* "Object Oriented Modeling of the AN/ALQ-172", classified, draft in edit at AFRL

TODAY'S GUEST PRESENTATION

James P. Stephens, Sr.
Air Force Research Laboratory

"RF Sensor Technologies"

C3 Electronics Combat

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

August
1998

ABSTRACT

This briefing constitutes an overview of the research areas of the C3 Electronic Combat Sub-Team, one of three EW Sub-Thrust focus areas in the EW Technology Branch of the RF Sensor Technology Division. The briefing has a communication countermeasures emphasis, focusing on advanced digital signal processing for EW. It begins with organizational background and objectives, C2W trends, and vision from the perspective of SNRW. A brief tutorial is provided to explain the motivation of the use of time-frequency distributions, wavelets, and cyclostationary signal processing. The current research activities of SNRW in all areas related to C3 Electronic Combat are introduced – both contracted and in-house. This includes such areas as featureless waveform countermeasures, time-frequency analysis, frequency hopping prediction, GPS ECM and ECCM, and covert waveform detection and jamming. In-house activities include time-difference of arrival using cyclostationarity, frequency hopping synthesizer development, spectral correlation analysis, and transform domain waveform experimentation. This 90-minute briefing is intended as a technical overview of the current activities of SNRW and should be of interest to technical managers, researchers, faculty, and engineering students.

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

August
1998

BIOGRAPHY

James P. Stephens Sr. is currently employed as a senior research and development engineer in the specialized area of communication countermeasures at the Sensor Directorate, RF Sensor Technology Division, Electronic Warfare Technology Branch. Mr. Stephens is engaged in exploratory development of next generation airborne radar, communication, and navigation jamming systems. He is responsible for in-house programs in communication countermeasures that develop, evaluate, and validate advanced countermeasure approaches and directs contracted developments that relate specifically to advanced communication countermeasures. His current areas of interest are applying signal-processing techniques against anti-jam (AJ) and low probability-of-intercept (LPI) waveforms using cyclostationarity, wavelet analysis, and time-frequency distributions. From Aug 1982 to Jan 1991, Mr. Stephens was employed at National Aerospace and Intelligence Center (NAIC - formerly Foreign Technology Division) as an electronics engineering analyst responsible for the analysis of foreign communication systems with emphasis on digital communications and advanced modulation techniques. He performed reverse engineering of foreign equipment to assess reliability, capability, and effectiveness of foreign communication systems.

Prior to beginning his career with the Air Force, Mr. Stephens was employed as the Engineer-in-Charge of the Cincinnati, OH field office from Aug 1969 to Aug 1982. He was responsible for the administration of the FCC field office. His duties included RF measurements, electromagnetic interference (EMI), and radio direction finding. Mr. Stephens performed inspections, investigations, and monitoring of a wide variety of radio systems in a three-state area with prior assignments in Detroit and Norfolk. Since Jan 1982, Mr. Stephens has been employed part-time as an instructor in the Electrical Engineering Technology curriculum, Associate Degree program, at Miami University where he teaches a variety of courses in electronic and microprocessor technology. He has authored or co-authored more than 15 publications.

5.2.4 Presentation by Jim Stephens, AFRL/SNRW, Dayton

A guest lecture presented by James P. Stephens, Sr. at this meeting was described by a synopsis in the overview of the previous section. Copies of this presentation are available on request from the RAPCEval Program Manager at Mercer Engineering Research Center.

5.3 Steering Committee, November 1998

5.3.1 Meeting Minutes

EW Receiver and Processing Concepts Evaluation Program
Program Research Standards Committee Meeting
Minutes 9 November, 1998

The RAPCEval Steering Committee held a meeting at MERC on 9 November, 1998 at 1:30 p.m. Dr. David Barwick, MERC Executive Director, kicked off the meeting with a welcome.

Dr. Tom Bass, MERC RAPCEval Program Director, then presented a "State of RAPCEval" slide overview presentation.

Dr. Bass then introduced the three students who were to give presentations on their research topics. Those students and their topics were:

- Mr Houston Jones – "Evaluation of Reed-Solomon Codes for CDMA Systems"
- Mr Mark Napier – "Civil IFF Reed-Solomon Code Application"
- Mr. Randy Ford – "Passive Location via Evolutionary Genetic Algorithms"

The next agenda item was a period of discussion and suggestions about the projects. Ches Rehberg (during discussions and via a later e-mail) pointed out the need for Air Force personnel adept in the topics being presented to attend the meetings. For example, two of the topics were on communications (robustivity via Reed-Solomon code implementations). Ches suggested such participants for future meetings.

Phil Oliver stated that data links are a big topic in the F-15 community. He suggested that these communication robustivity research topics may have an application in that area.

Several attendees remarked that the student presentations appeared higher in quality on this occasion. Some questions were raised about the applicability of the communication topics to the original RAPCEval charge.

The meeting was then adjourned.

5.3.2 Agenda

Agenda

RAPCEval program

STEERING COMMITTEE MEETING

98 November 9 --- 1:30 PM to 4:15 PM
Conference Room, Mercer Engineering Research Center
135 Osigian Boulevard, Warner Robins, GA

Meeting called by: Nicholas Pequignot, AFRL/SNRP, Air Force Research Laboratory Program Manager

Facilitator: Dr. Tom Bass

	AF Research Laboratory	Robins AFB	Mercer University
Committee Members:	Mr Nicholas A Pequignot	Mr Steve Strawn	Dr David Barwick
	Mr Emil R Martinsek	Mr John LaVecchia	Dr Tom Bass
	Mr Norman A Toto	Mr Phil Oliver	Dr Aaron Collins
	Dr Duane A Warner	Mr Ches Rehberg	Dr Benham Kamali
	Mr Paul J Westcott	Mr. Larry Sheets	Dr Paul MacNeil
Email addresses:	pequigna@sensors.wpafb.af.mil martinser@sensors.wpafb.af.mil totona@sensors.wpafb.af.mil warnerda@sensors.wpafb.af.mil westcopj@sensors.wpafb.af.mil	lnerr@ewir-wr.robins.af.mil lavecchia@ewir-wr.robins.af.mil oliver@ewir-wr.robins.af.mil rehberg@ewir-wr.robins.af.mil sheets@ewir-wr.robins.af.mil	dbarwick@merc.mercer.edu tbass@merc.mercer.edu collins_as@mercer.edu macneil_pe@mercer.peachnet.edu kamali_b@mercer.peachnet.edu

Meeting Agenda, continued,,,

Schedule

Greetings	Dr David Barwick	1:30 PM – 1:35 PM
Meeting Overview	Dr Tom Bass	1:35 PM - 1:45 PM
Student Proposals	Students:	
Evaluation of Reed-Solomon Codes for CDMA Systems	Mr Houston Jones	1:45 PM – 2:15: PM
Civil IFF Reed-Solomon Code Application	Mr Mark Napier	2:15 PM – 2:45 PM
Passive Location via Evolutionary Genetic Algorithms	Mr. Randy Ford	2:45 PM – 3:15: PM
Discussions & New Business	Dr. Tom Bass	3:15 PM - 3:45 PM
Adjourn		4:15 PM

5.3.3 Attendance Roster

The attendees at this meeting are listed here:

	Name	Organization	Phone	Email Address
1	Tom Bass	MERC	912-953-6800	bass_wt@mercer.edu
2	Dave Barwick	MERC	912-953-6800	barwick_dt@mercer.edu
3	Ches Rehberg	WR-ALC /LNEX	912-936-4525	rehberg@ec.robins.af.mil
4	Charles Bass	MERC	912-953-6800	bass_cd@mercer.edu
5	RP (Phil) Oliver	WR-ALC /LNERT	912-926-2588	oliver@ec.robins.af.mil
6	Steve Strawn	WR-ALC /LNERR	912-926-6435	strawn@ec.robins.af.mil
7	Tony White	AFRL /SWRP	937-255-6127, Ext. 4236	whiteaw@sensors.wpafb.af.mil
8	Nicholas Pequignot	AFRL /SNRP	937-255-6127, Ext. 4235	pequigna@sensors.wpafb.af.mil
9	Mark Napier	Scientific Atlanta	770-903-6980	mark.napier@subasic.sciatl.com
10	Randy Ford	WR-ALC /LYSBC	912-926-0423	fordra@avionics.robins.af.mil
11	Houston Jones	WR-ALC /QLYM	912-926-5458	jhjones@hom.net
12	Behnam Kamali	Mercer	912-752-2415	kamali_b@mercer.edu
13	Paul McNeil	Mercer	912-752-2185	macneil_pe@mercer.edu
14	Joseph K. Black	MERC	912-953-6800	jblack@merc.mercer.edu
15	Ray Mitchell	MERC	912-953-6800	rmitchell@merc.mercer.edu
16	Skip Finnigan	MERC	912-953-6800	sfinnigan@merc.mercer.edu

5.3.4 Overview of the Program (Dr. Bass)

The Overview Briefing of the RAPCEval Program as presented at this meeting is reproduced on the next 7 pages.

RAPCEval STEERING COMMITTEE MEETING

November 9, 1998

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

November
1998

PROJECT INFORMATION

- ★ **Contract: F09603-93-G-0012-0017**
- ★ **Customer: Wright Laboratory**
- ★ **Contract Value: $299,964**
- ★ **Program Status:**
  - ★ **Graduate Research Jointly Supported by Mercer, Wright Lab, WR-ALC, and Industry**
  - ★ **Successful research projects (9) have been completed; several (5) research projects are ongoing**
  - ★ **All research has been determined by the committee to be useful to the Air Force, as well as having merit for graduate research**

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

November
1998

PROGRAM RESEARCH STANDARDS COMMITTEE

- ★ *MERC:* Dr. David Barwick (Chairman), Dr. Tom Bass (PMgr)
- ★ *AIR FORCE LABORATORY:* Mr. Nick Pequignot (PMgr), Mr. Emil R. Martinsek, Mr. Norman A. Toto, Dr. Duane A. Warner, Mr. Paul J. Westcott
- ★ *WR-ALC:* Mr. Steve Strawn (Pmgr), Mr. John LaVecchia, Mr. Phil Oliver, Mr. Ches Rehburg, Mr. Larry Sheets
- ★ *MERCER UNIVERSITY:* Dr. Aaron Collins, Dr. Benham Kamali, Dr. Paul MacNeil

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

November
1998

GRADUATES

- ★ *Mark Astin,* “Application of Parallel Computing Techniques to the RAD Algorithm”, AFRL-SN-WP-TR-1998-1088 (classified) final edit in progress
- ★ *Henderson Benjamin,* “Selection of Reed Solomon Codes Using Neural Networks” (unclassified) AFRL-SN-WP-TR-1998-1056 pg. 131
- ★ *Ron Brinkley,* “Burst Error Correction with Reed-Solomon Codes” to be included in forthcoming unclassified report
- ★ *Mark Campbell,* “Auto-Regressive Spectral Analysis - EW Applications”, unclassified report available from MERC

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

November
1998

GRADUATES

★ *Claus Franzkowiak,* “Four-Pulse Primary RAD Filter Development”, AFRL-SN-WP-TR-1998-1087 (classified) final edit in process

★ *Neal Garner*, “Error Correction and Prediction for Improved Communication of Time and Time Measurements”, unclassified report available from MERC

★ *Joseph Kelley* , “A Parameter Determination Alternative for RAD Analysis”, WL-TR-95-1005 (classified).

★ *Joseph Kelley,* “MultiGroup Simultaneous RAD Parameter Selection”, WL-TR-97-1094 (classified).

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

November
1998

GRADUATES

- ★ *Max Roesel*, “Agile RF/PRI Radar Analysis via RAD”, WL-TR-95-1020 (classified)
- ★ *Dave Schuler*, “Comparison of Algorithms for Geolocation of Radar Signals”, available from MERC on need to know
- ★ *Kirk Wright*, “Object Oriented Modeling of the AN/ALQ-172” AFRL-SN-WP-TR-1998-1086 (classified) final edit in process

EW RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM (RAPCEval) OVERVIEW

November
1998

TODAY'S STUDENT PRESENTATIONS

★ *Mark Napier, "Application of Reed-Solomon Codes to Improve Noise Resistance of Civil IFF Communication"*

★ *Houston Jones, "Evaluation of Reed-Solomon codes for CDMA systems"*

★ *Randy Ford,* "Passive Location via Evolutionary Genetic Algorithms"

5.3.5 Presentation By Mark Napier

The student briefing presented by Mark Napier at this meeting is reproduced on the next 15 pages.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

RESEARCH PROPOSAL PRESENTATION

David M. Napier

Scientific Atlanta
ASIC Engineer, Digital Subscriber Group

Background and Experience:

Education: BSCPE from North Carolina State University.
Pursuing MSEE with emphasis in Digital Communications - 26 Semester hrs. completed.

Scientific Atlanta:

Digital and Analog Electronics Design, ASIC design and test.

Research Topic:

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS

PROBLEM STATEMENT

A proposed aircraft collision avoidance scheme uses the existing civilian IFF system. The technique uses the Mode S signaling scheme but defines a new message. The GPS position and velocity along with barometric altitude are transmitted instead of aircraft ID.

The new message is 112 bits long, 40 of which have been reserved for FEC coding. The proposed FEC scheme is a 5 bit t=4 Reed-Solomon code, RS(31,23). The proposed work is to analyze this scheme to determine the reliability improvements to be realized from the RS code. A decoder will be designed, implemented and tested in Verilog.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS

PROPOSAL JUSTIFICATION

- ★ Applicability to Mercer University MSEE Program
  - Digital Communication Classes Provide Material
  - RS Decoder Engine useful for other projects
- ★ Applicability to the USAF
  - Similar encoding could be used in Military IFF
  - Military aircraft could participate in the civilian system without revealing aircraft ID

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS

PROPOSED RESEARCH OBJECTIVES

- ★ Model the Mode S Modulation Scheme with RS
  - Verify that the Model Agrees with Analytical Predictions
  - Show that System Performance is Increased
- ★ Design the RS Decoder
  - Design the Decoder in Verilog
  - Verify with Verilog Testbench

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS

PROPOSED METHOD OF INVESTIGATION

- ★ Implement RS Decoder in Verilog
- ★ Model the IFF Mode S Modulation Scheme
  - Literature Search and Analysis
  - Model Development and Verification
- ★ Model the System with RS FEC coding
  - Model without Erasure Information
  - Model With Erasure Information

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS

PROPOSED SCHEDULE

- ★ Spring Semester
    - Search literature for any previous work on PPM signaling characteristics.
    - Provide analysis as a basis for model.
    - Develop and verify PPM model.
    - Simulate in presence of fading with and without RS coding.
    - Develop and test decoder for RS(31,23) code.
    - Document results and submit project report.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS

TECHNICAL PRESENTATION

The civilian aircraft transponders are based on a WWII IFF (Information Friend or Foe) system. It is intended for ground based ATCRBS (Air Traffic Control Radar Beacon System) use and in general no information is available to aircraft not using air traffic control services. TCAS provides major air carriers with collision avoidance information but is an expensive system that has very limited capacity. A distributed collision avoidance system using GPS(Global Positioning System) would be inexpensive and highly reliable.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

The current system interrogates aircraft on 1030 MHz. The transponders respond on 1090MHz with mode 3A (squawk code), mode C (squawk altitude), or various mode S (squawk ID) messages depending on the interrogation sequence received and the transponder's capability. It transmits at a peak power output of 250 Watts. It uses pulse shaping such that the transmitted power at plus or minus 25MHz is down by 60 dB [TSO C74C]. The receiver circuit has a threshold sensitivity of -70 dBm.

A proposed scheme[1] would use current transponder technology to transmit at random intervals GPS position and velocity along with barometric altitude in addition to the normal mode 3A/C responses. If widely used, any aircraft with a compatible receiver could have a cockpit display showing other aircraft in the area. The new system has been named "Tail Light", analogous to the tail light in a car at night or in the fog.

The proposed system would use the mode S downlink format signaling which is a PPM(Pulse Position Modulation) scheme with a 1 Mbit/s rate. A "1" is defined to be a 0.5us burst followed by 0.5us of off time. A "0" is defined to be 0.5us of off time followed by a 0.5us burst. The message is proceeded by a 8us sync pulse. Either 56 (single length) or 112 (double length) bits of data follow.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

For the double length message, 40 bits have been assigned for Forward Error Correction (FEC) using Reed-Solomon encoding. Since this is a short message, the optimal burst error capability is obtained[2] with a 5 bit t=4 or RS(31,23) code. This code can correct a 16 bit worst case burst error.

Also, if erasure information can be provided by the receiver a burst error of 36 bits can be corrected effectively doubling the error correction capability[3]. Note that with PPM a simple system for obtaining erasure information is available. Since "00" and "11" are not defined, any bit received with these sequences should be flagged as an erasure. As these bits are arranged into 5 bit words for the decoder, the word would be marked as an erasure.

In conclusion, the proposed system would be a benefit for general aviation which lacks a cost effective solution for collision avoidance. The FEC scheme proposed would greatly enhance overall system reliability. Lastly, the RS(31,23) decoder would be useful for any mobile system that uses short (61-155 bits) bursts of data.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

References:

[1] Peshak, B. Keith; http://www.monarch-air.com/gaviation/

[2] B. Kamali, ”Some new Outlooks on Burst Error Correction Capabilities of Reed-Solomon Codes with Applications in Mobile-Communications”, Proceedings of IEEE VTC’98, Ottawa, Canada, May 1998, pp. 343-347.

[3] S. Lin & D. J. Costello, “Error Control Coding: Fundamentals & Applications”, Prentice Hall, 1983.

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

Mode A Packet - Four digit squawk code from front panel is encoded in octal form.

Bit	Description
F1	1st Framing Bit - 1
C1	3rd Digit 1's Value
A1	1st Digit 1's Value
C2	3rd Digit 2's Value
A2	1st Digit 2's Value
C4	3rd Digit 4's Value
A4	1st Digit 4's Value
X	No Transmit - 0
B1	2nd Digit 1's Value
D1	4th Digit 1's Value
B2	2nd Digit 2's Value
D2	4th Digit 2's Value
B4	2nd Digit 4's Value
D4	4th Digit 4's Value
F2	2nd Framing Bit - 1
X	No Transmit - 0
X	No Transmit - 0
SPIP	Special Purpose ID Pulse; Front Panel Ident. Button.

Mode C Packet - Identical to Mode A packet. Altitude encoded on 10 bits of the digit values.

Mode S Sync Pulse.

Time	Value
0 - 0.5 US	1
0.5 - 1.0 US	0
1.0 - 1.5 US	1
1.5 - 3.5 US	0
3.5 - 4.0 US	1
4.0 - 4.5 US	0
4.5 - 5.0 US	1

Tail Light Long Format Message, 112 bits.

Field	Length	Description
Preface	5 bits	TBD, possible DF26 or 11010 base 2.
Latitude	16 bits	Ones of degrees, minutes and tenths (DMM.M). Precision is 0.1 minutes = 600 feet. Period is 9 degrees, 59.9 minutes = 600 NM. The second byte, tens of minutes, only ranges from 0 through 5, thus doesn't use the MSB. Set that bit to 0 for north, and 1 for south. 4 numbers, 16 bits.
Longitude	16 bits	Similar to latitude. Set the MSB of the tens of minutes byte to 0 for west and 1 for east. From 70 through 80 degrees latitude send tens of degrees through whole minutes (DDMM). Above 80 degrees send whole degrees and tens of minutes (DDDM). 4 numbers, 16 bits.
Altitude	10 bits	From Altitude Encoder.
Speed	12 bits	000-999 knots. If the craft is traveling over 999 knots, send 999, don't blindly drop the leading byte and send 000. 3 numbers, 12 bits.
Course	12 bits	000-359 degrees true. Use the otherwise unused MSB of the hundreds of degrees to include the message validity flag. 3 numbers, 12 bits.
Stuff Bit	1 bit	TBD
FEC Parity	40 bits	RS(31,23) code. 5 bit symbols, t = 4.

Tail Light Short Format Message, 56 bits.

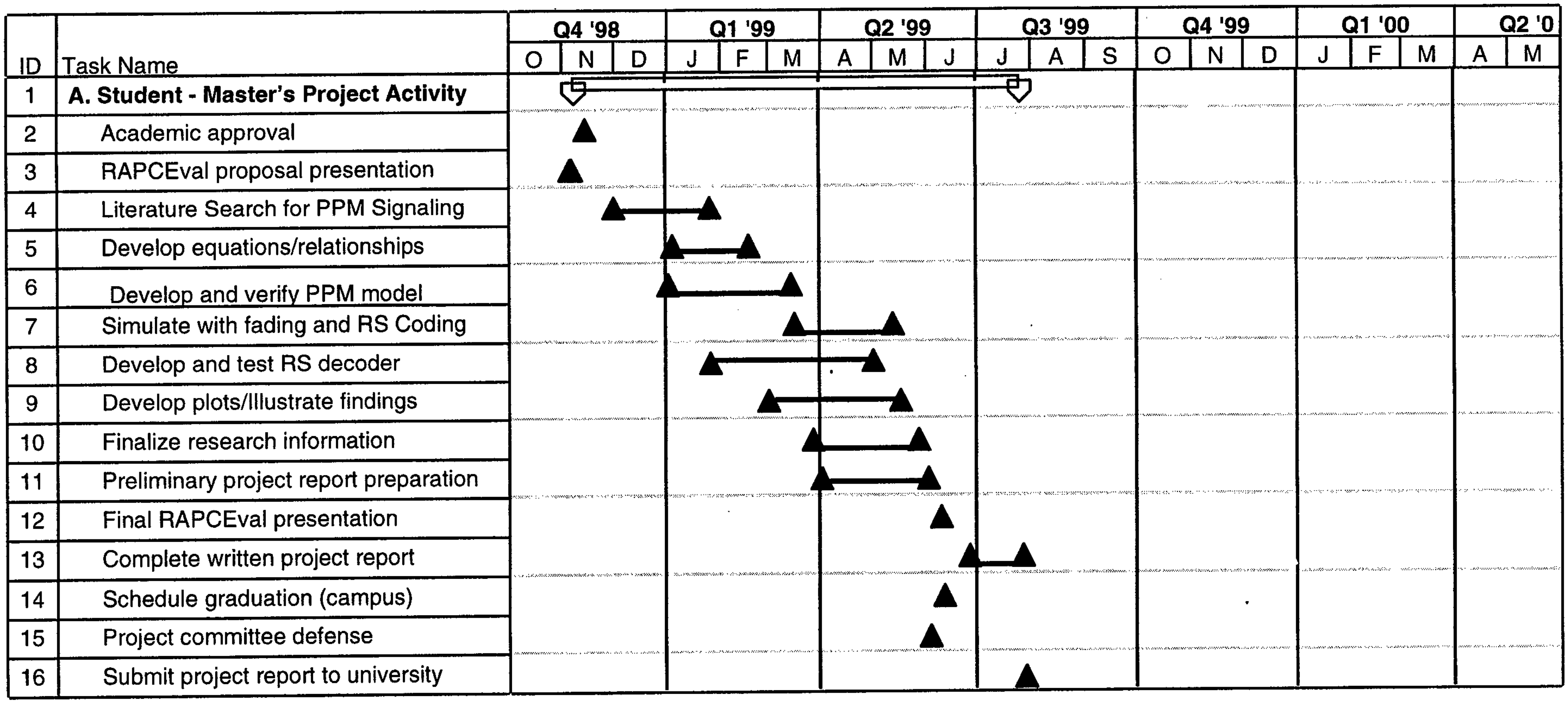
Field	Length	Description
Preface	5 bits	TBD, possible DF27 or 11010 base 2.
Latitude	12 bits	Minutes and tenths (MM.M). Precision is 0.1 minutes = 600 feet. Period is 59.9' = 60 NM. The first byte, tens of minutes, only ranges from 0 through 5, thus doesn't use the MSB. Set that bit to 0 for north, and 1 for south. 3 numbers, 12 bits.
Longitude	12 bits	Similar to latitude. Set the MSB of the tens of minutes byte to 0 for west and 1 for east. From 70 through 80 degrees latitude send ones of degrees through whole minutes (DMM). Above 80 degrees send whole degrees only (DDD) and put the E/W bit in the otherwise unused first bit of the hundreds of degrees. 3 numbers, 12 bits.
Altitude	10 bits	From Altitude Encoder.
Speed	8 bits	10 knots precision, up to 990 knots. 2 numbers, 8 bits.
Course	8 bits	10 degrees precision, 000-350 degrees true. Use the otherwise unused MSB of the hundreds of degrees to include the validity flag. 2 numbers, 8 bits.
Parity	1 bit	Single parity bit for message.

MERCERIENSIS UNIVERSITATIS SIGILLUM MDCCCXXXVIII

RF RECEIVER AND PROCESSING CONCEPTS EVALUATION PROGRAM
(RAPCEval)

Mark Napier
PROPOSAL
11/9/1998

APPLICATION OF REED-SOLOMON ENCODING TO IMPROVE PROPOSED COLLISION AVOIDANCE SYSTEM BASED ON CIVILIAN ATCRBS



ID	Task Name
1	**A. Student - Master's Project Activity**
2	Academic approval
3	RAPCEval proposal presentation
4	Literature Search for PPM Signaling
5	Develop equations/relationships
6	Develop and verify PPM model
7	Simulate with fading and RS Coding
8	Develop and test RS decoder
9	Develop plots/Illustrate findings
10	Finalize research information
11	Preliminary project report preparation
12	Final RAPCEval presentation
13	Complete written project report
14	Schedule graduation (campus)
15	Project committee defense
16	Submit project report to university

5.3.6 Presentation by Houston Jones

The student briefing presented by Houston Jones at this meeting is reproduced on the next 15 pages.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

RESEARCH PROPOSAL PRESENTATION

H. JONES

WRALC/QLYM
Special Projects System Engineer

Background and Experience:
Education: BS from Jacksonville State University (Mathematics)
BEE from Auburn University
MSA from Georgia College (Management)
Pursuing MSE with emphasis in Electrical Engineering. - 29 semester hrs. completed
WR/ALC: Specialized Management Directorate
Research Topic:

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROBLEM STATEMENT

Investigate the feasibility of the application of using Reed-Solomon (RS) codes to correct single burst errors in Rayleigh fading channel for Code Division Multiple Access (CDMA) Digital Cellular Phone systems

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROPOSAL JUSTIFICATION

★ Applicability to Mercer University MSE Program

- *Provides the results of research of rapidly growing area of applications of spread spectrum communications to digital wireless systems*
- *This specific area of research has not been addressed*

★ Applicability to the USAF

- *Digital cellular communications are important and valuable to military as well as civilian users. Increases the knowledge base in digital mobile communications*

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROPOSED RESEARCH OBJECTIVES

- ★ Determine the feasibility of using Reed-Solomon (RS) codes for performance improvement in Code Division Multiple Access (CDMA) mobile communications systems
- ★ Determine the feasibility of using Reed-Solomon (RS) codes to reduce system complexity in CDMA mobile communications systems

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROPOSED RESEARCH OBJECTIVES
Performance Evaluation

★ Minimum:

No performance degradation

Error control evaluation

Determination of frame error rate and bit error rate

★ Optimistic: Improved performance and reduced complexity

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROPOSED RESEARCH OBJECTIVES

★ Criteria for success/added value:

Evaluation of the performance of RS codes vs convolutional codes with block interleaver to correct single burst errors in a Rayleigh fading channel

Performance -- RS < CC+BI;

RS = CC + BI or RS > CC + BI

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROPOSED METHOD OF INVESTIGATION

- ★ Conduct a literature search on related subjects
- ★ Study the characteristics and statistics of fading in wireless mobile systems
- ★ Conduct a study to determine the expected value of the number of bits corrupted of by fades
- ★ Study to determine the feasibility of using RS code families for single burst error correction in Rayleigh fading channel

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROPOSED METHOD OF INVESTIGATION

- ★ Select the proper RS code for this application
- ★ Evaluate convolution codes (complexity, symbol size, and error correction capability)

  $r = 1/2\ ;\ K = 9$ $r = 1/3\ ;\ K = 9$

- ★ Simulate CDMA system with convolutional encoder and block interleaver

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

PROPOSED METHOD OF INVESTIGATION

★ Simulate CDMA system with the selected RS codes (RS codes replace the convolutional codes and block interleaver)

★ Analyze Results

  Error correction capability

  Frame error rate; bit error rate

  Delay

  System complexity

AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

First generation cellular systems in North America was the Advanced Mobile Phone System (AMPS) AMPS is an analog system introduced in 1983.

Second generation cellular in North America evolved to a time division multiple access (TDMA) system. This was the first digital cellular system in the US. This system was introduced in 1991.

In 1993 a cellular system based on code division multiple access (CDMA) was developed. This system uses direct sequence spread spectrum techniques requiring less power and allowing for more users than the other designs.

MERCERIENSIS UNIVERSITATIS SIGILLUM MDCCCXXXVIII

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

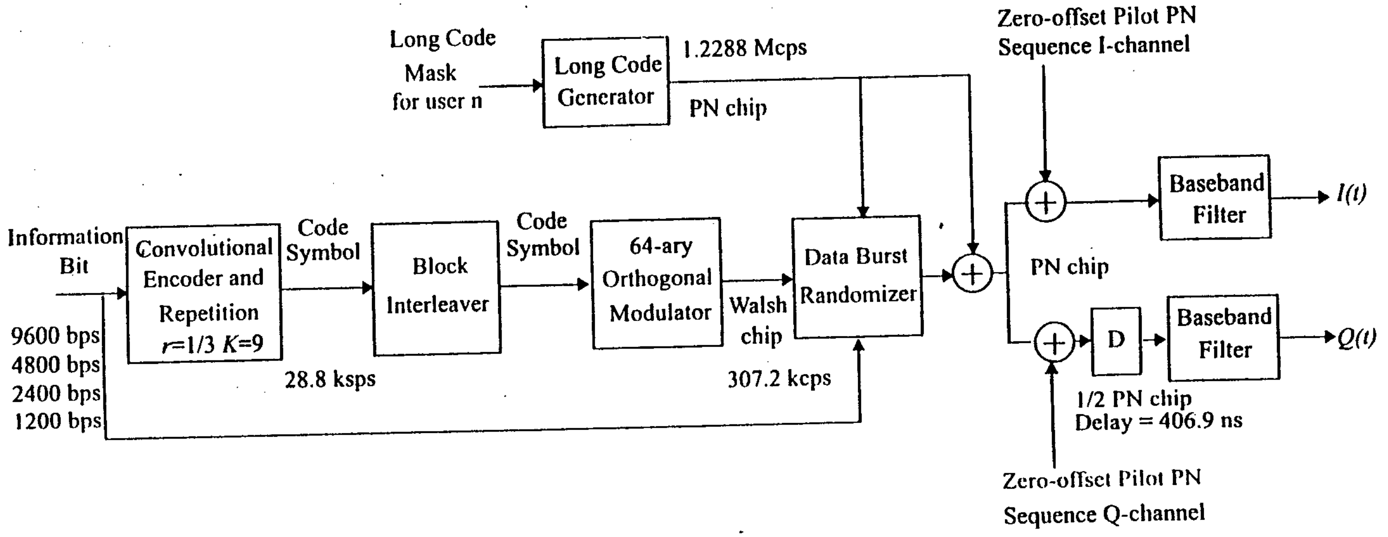
AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

My proposal is to study fading in mobile cellular systems and investigate the feasibility of using Reed-Solomon codes in place of convolutional codes and the block interleaver to correct single burst errors in CDMA cellular systems.

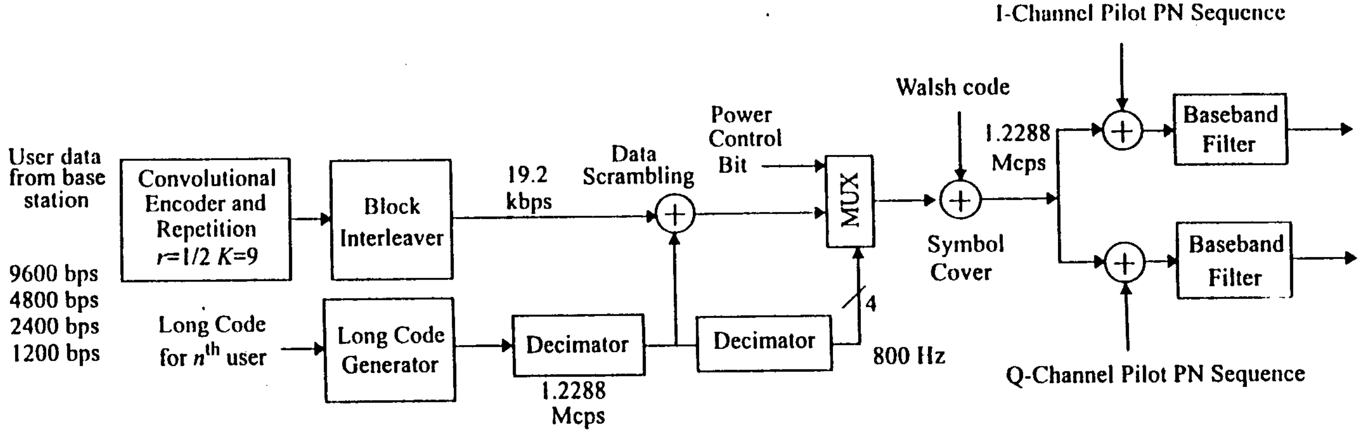
AN *INVESTIGATION ON THE FEASIBILITY OF THE APPLICATION OF REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS*

Tools

I will need a standard software package to model and simulate the CDMA systems.



Reverse IS-95 channel modulation process for a single user.

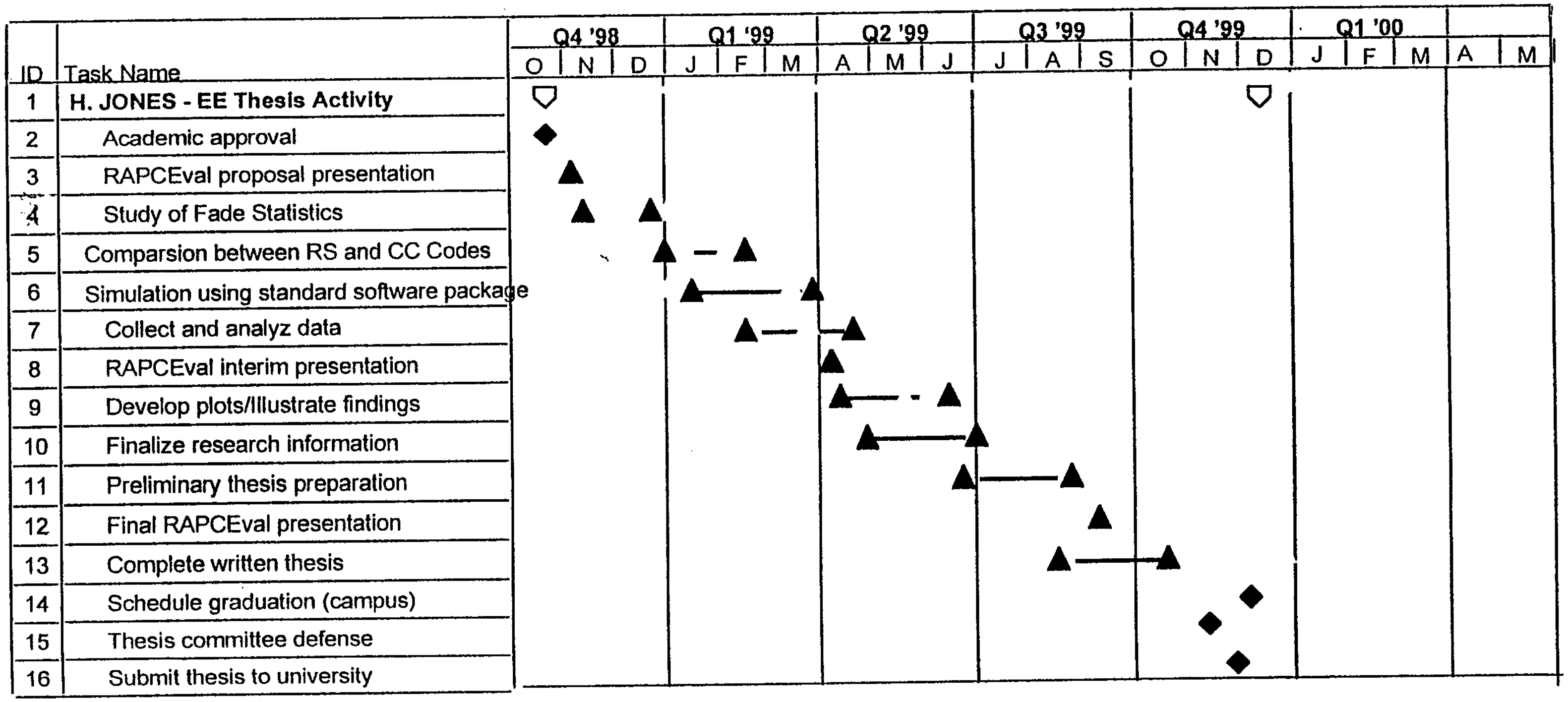


Forward CDMA channel modulation process.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

H. Jones
MSE/EE Program
Proposal

REED-SOLOMON CODES IN WIRELESS CODE DIVISION MULTIPLE ACCESS SYSTEMS - Schedule Chart



5.3.7 Presentation by Randy Ford

The student briefing presented by Randy Ford at this meeting is reproduced on the next 47 pages.

MERCERIENSIS UNIVERSITATIS SIGILLUM MDCCCXXXVIII

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

RESEARCH PROJECT FINAL REPORT

James R. Ford

WRALC/LYSBC
JTIDS Software Engineer

Background and Experience:

Education:	BEE from Georgia Tech
	Pursuing MSE/SE with emphasis in AI
U.S. Army:	Signal Corps Platoon Leader, 2 years
WR/ALC:	Avionics Software Engineering, 15 years

Research Topic:

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

PROBLEM STATEMENT

In "An Efficient Method of Passive Emitter Location" Klaus Becker proposes using multiple frequency measurements by a moving sensor to generate a Cramer-Rao bound ellipsoid. This ellipsoid is then projected into the x-y plane to form an error surface which may be searched by generating a starting point and then minimizing the sum square error (SSE) using the simplex method. Once the emitter's location is found, the estimated emitter frequency is determined by obtaining a maximum likelihood (ML) value at that point.

Differential Evolution, developed by Kenneth Price and Rainer Storn, is a minimization technique for real-valued problems which is based on genetic algorithms. This project will compare the performance of this method, with the performance of the Nelder-Mead simplex method. Both methods will be implemented in C++ and called from an existing MATLAB implementation.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION

RESEARCH OBJECTIVES

- ★ Implementation of Differential Evolution
  - In C++
- ★ Primary Objective is:
  - To determine best minimization technique: Which one produces <u>fastest</u> solution?

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

METHOD OF INVESTIGATION

- ★ Study existing MATLAB implementation
- ★ Implement minimization in C++
  - Implement Nelder-Mead simplex method
  - Implement DE minimization.
- ★ Test both implementations
  - Identical Platforms - Pentium PC
  - Determine <u>best</u> method based on speed of detection.

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

METHOD OF INVESTIGATION

- ★ Test both implementations (cont)
  - Compare accuracy of solutions obtained.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Passive Emitter Location

- Locate a stationary emitter by taking multiple frequency measurements from a moving platform.
- Obtain the Cramer-Rao (CR) bound ellipsoid
- Get x,y ellipse by projecting the CR ellipsoid onto the x,y plane.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Passive Emitter Location (continued)

- Run a Monte Carlo simulation from starting points for the search of the error ellipse.
- Find minimum sum squared errors and estimated emitter frequency for each run.

★ Passive Emitter Location discussion is based on “An Efficient Method of Passive Emitter Location” by Klaus Becker in the October 1992 IEEE Transactions on Aerospace and Electronic Systems.

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Nelder-Mead Simplex Method

- A **Simplex** is a geometrical figure consisting, in N dimensions, of $N + 1$ points and all their interconnecting line segments and polygonal faces.
- Initialization: Start with N + 1 initial points which give an initial simplex.

UNIVERSITATIS SIGILLUM MERCERIENSIS MDCCCXXXVIII
MACON / ATLANTA

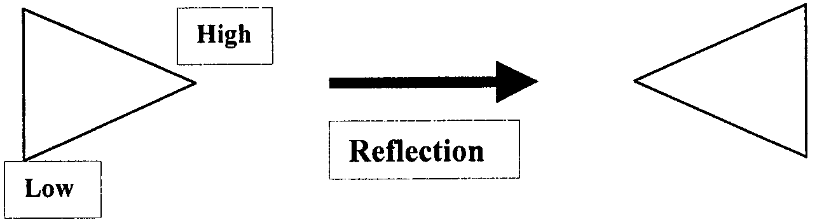
ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- Take steps - Either:
  - Reflect the simplex away from the high point



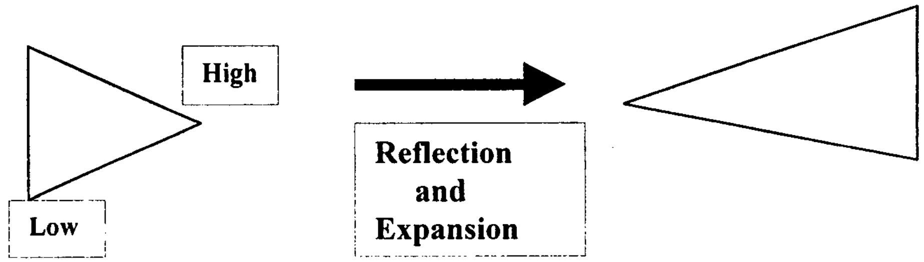
ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- Take steps (cont) - Either:
  - Reflect and expand the simplex away from the high point



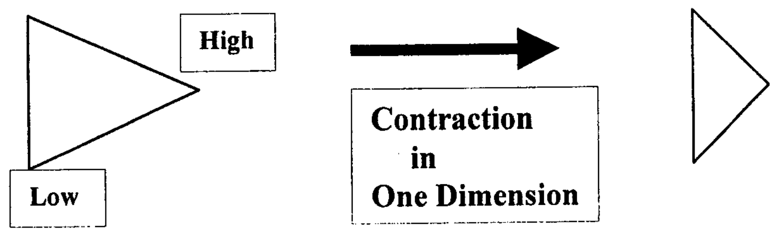
ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- Take steps (cont) - Either:
    - Contract the simplex in one dimension from the high point



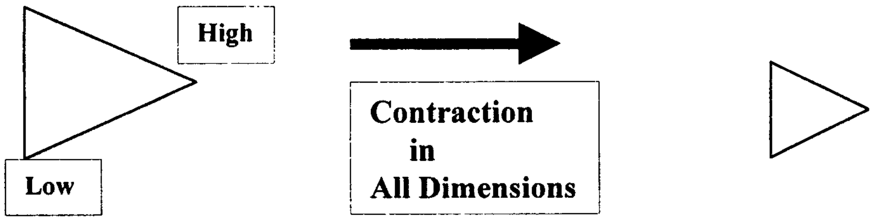
ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- Take steps (cont) - Either:
    - Contract the simplex along all dimensions toward a low point.



- ★ The Nelder-Mead Simplex Method (cont)
  - Termination - Either:
    - Stop when the distance moved in a step is less than some tolerance.
    - Stop when the decrease in function value during a step is less than some tolerance.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Nelder-Mead Simplex Method discussion is based on Chapter 10 of Numerical Recipes in C (2nd Edition) by William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution

- A **Genetic Algorithm** is a probabilistic search technique that makes use of the principles of genetics.
- An **Evolutionary Strategy** is a form of Genetic Algorithm that is especially useful in minimizing functions with real variables and many local minima.
- Differential Evolution is an **Evolutionary Strategy**

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ Differential Evolution (cont)
  - Evolutionary Strategies:
    - Encode values as floating point numbers instead of translating them into a special alphabet.
    - Usually use a deterministic selection scheme during crossover.
    - Use addition instead of symbol-swapping during mutation. This makes incremental search easier.

★ Differential Evolution (cont)

- Initialization:
  - Form two arrays of NP, D-dimensional, real-valued, vectors - current population and next generation.
  - Set reasonable limits on the parameter values.
  - Generate the initial population and evaluate the ***cost*** of each member. Store these values in a ***cost*** array.
  - Each member of the current population takes its turn as the ***target*** vector for mutation, crossover, and selection

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- Initialization (cont):
  - Assign F, the scaling factor. $(0 < F \leq 1.2)$
  - Assign CR, the crossover constant. $(0 \leq CR \leq 1)$
- Mutation:
  - Determine the difference between two randomly chosen population members. $\mathbf{x}_a - \mathbf{x}_b$
  - Multiply by the scaling factor F. $F*(\mathbf{x}_a - \mathbf{x}_b)$

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- Mutation (cont):
  - Add to a randomly chosen vector $\mathbf{x}_c$ to form the 'mutated' vector: $\mathbf{x}_c' = \mathbf{x}_c + F^*(\mathbf{x}_a - \mathbf{x}_b)$.
- Crossover:
  - Produce the trial vector $\mathbf{x}_t$ by combining the target with $\mathbf{x}_c'$.
  - Conduct D-1 experiments starting at a randomly selected parameter.

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- Crossover (cont):
  - For each parameter, compare CR to a uniformly distributed random number from 0 to 1.
  - If the number > CR then the trial vector's parameter comes from the target.
  - If the number ≤ CR then the trial vector's parameter comes from $\mathbf{x}_c$'.
  - The last parameter always comes from $\mathbf{x}_c$' in order to make sure that the trial vector differs from the target.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ Differential Evolution (cont)
  - Selection:
    - The cost of each trial vector is compared to the cost of the target.
    - The vector with the lowest cost is moved to the next generation array. If the trial vector wins, its cost replaces its parent's cost in the cost array.

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Differential Evolution (cont)

- The Next Generation:
  - After each vector in the current array has been targeted, the next generation array becomes the current array.
  - The old current array is used to hold the results of the next round of trials.
- Termination:
  - Trials continue for a pre-set number of iterations or until the cost of the best vector is less than or equal to a pre-set value.

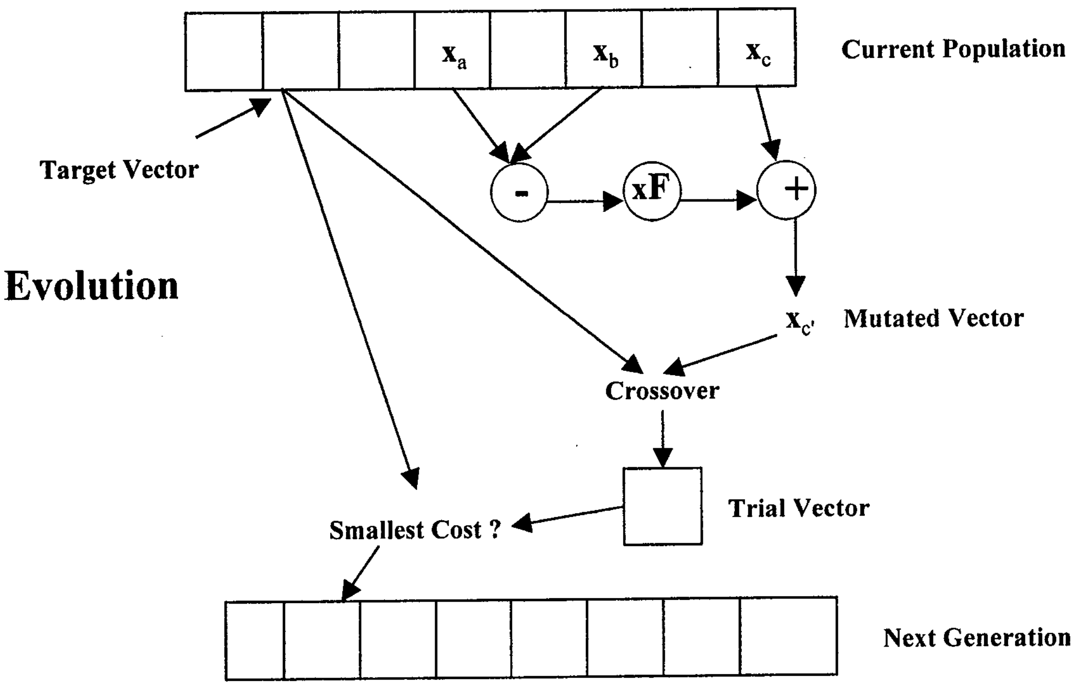
ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

Differential Evolution



ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ Differential Evolution discussion is based on “Differential Evolution” by Kenneth Price and Rainer Storn, Dr. Dobb’s Journal, April 1997, Issue #264

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The MATLAB Simulator:
- Based on program developed at MERC which used the built-in MATLAB fmins function for minimization.
- Calls external C++ dlls called MEX files for minimization
- Two versions - nmloc.m and deloc.m

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The MATLAB Simulator (cont) :

- Version nmloc.m calls C++ file nmc275.dll to minimize using the Nelder-Mead simplex method.
- Version deloc.m calls C++ file dec95.dll to minimize using DE.
- Both read in data from same external file data.m

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The MATLAB Simulator (cont) :

- Both receive a starting point and number of Monte Carlo runs from the user.
- The DE version also receives the population size, scaling factor, crossover constant, and number of iterations from the user.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The MATLAB Simulator (cont) :

- Both output emitter locations, error distances, and ML frequencies for all the Monte Carlo runs
- Both output the Average Error Distance (AED) over all of the Monte Carlo runs. This is a way to compare the accuracy of the two methods

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

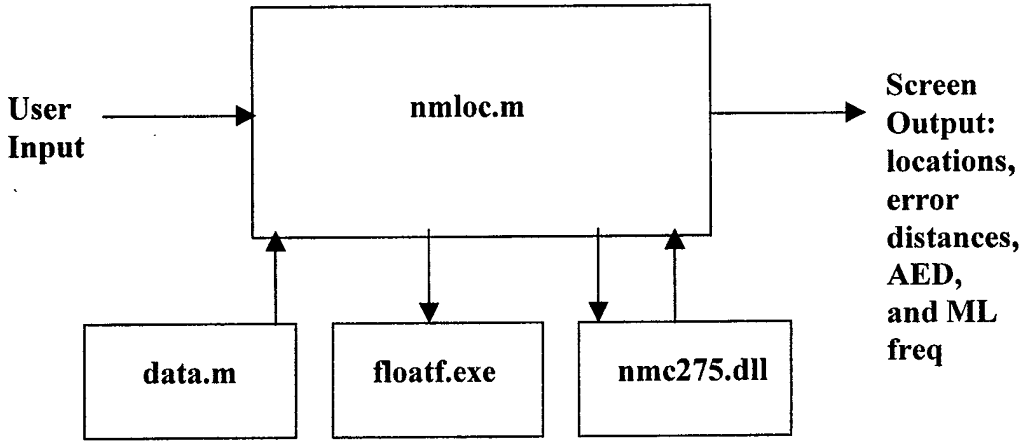
James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION

TECHNICAL PRESENTATION

★ The MATLAB Simulator (cont) :

- The simplex version:



User Input

nmloc.m

Screen Output: locations, error distances, AED, and ML freq

data.m

floatf.exe

nmc275.dll

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

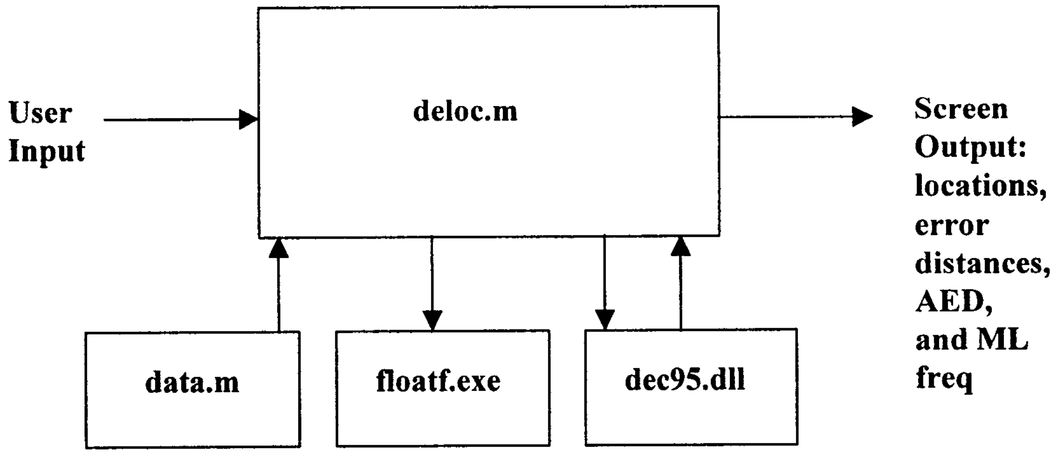
James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION

TECHNICAL PRESENTATION

★ The MATLAB Simulator (cont) :

- The DE version:



User Input

deloc.m

Screen Output: locations, error distances, AED, and ML freq

data.m

floatf.exe

dec95.dll

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The simplex dll :
  - Based on an algorithm in Numerical Recipes in C
  - Is sent the search starting point and other search data by MATLAB.
  - Generates an initial simplex by calling get_starting_simplex.
  - Searches the error space by calling amoeba.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

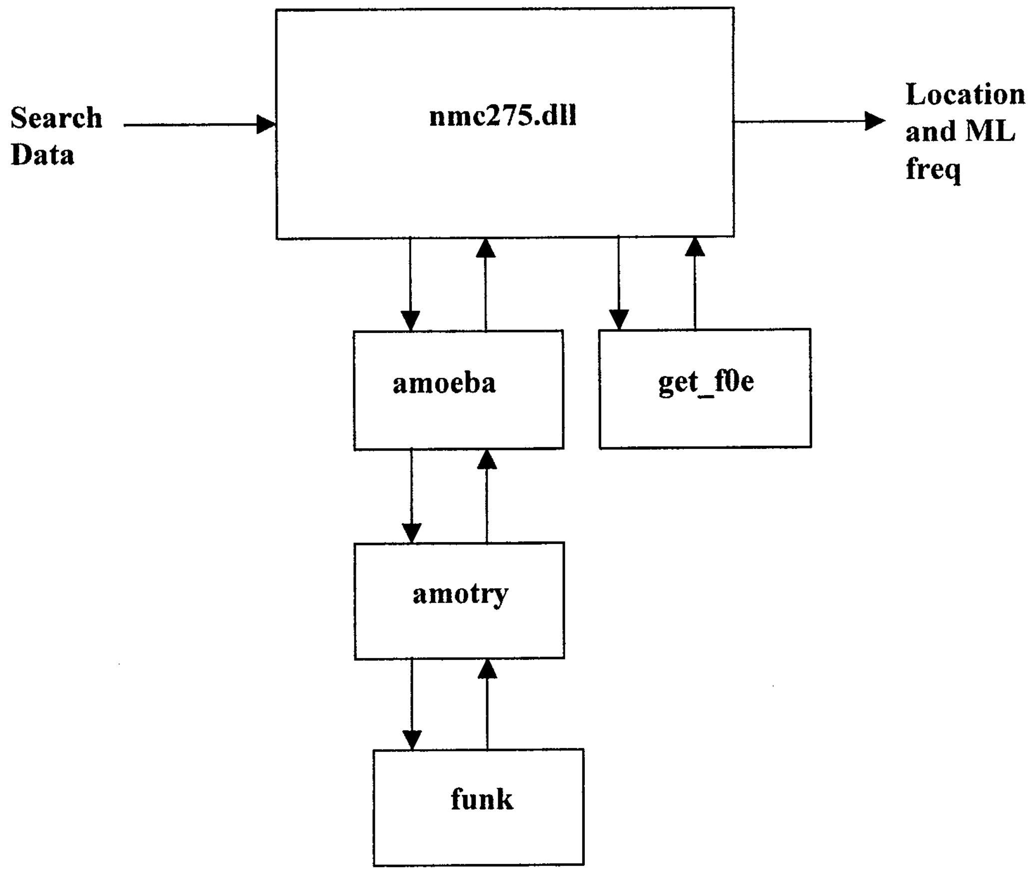
TECHNICAL PRESENTATION

★ The simplex dll (cont):
  - Function amotry determines the value of the function funk at the simplex points.
  - Function get_f0e calculates the ML frequency at the emitter's location.
  - Returns the low point and ML frequency to MATLAB.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.


Search Data

nmc275.dll

Location and ML freq

amoeba

get_f0e

amotry

funk

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- The DE dll :
  - Is sent the search starting point , the search data, the population size, scaling factor, crossover constant, and number of iterations by MATLAB.
  - Sets up the current generation array and the cost array and sends them to de_minimizer.
  - Searches the error space by calling de_minimizer.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The DE dll (cont) :

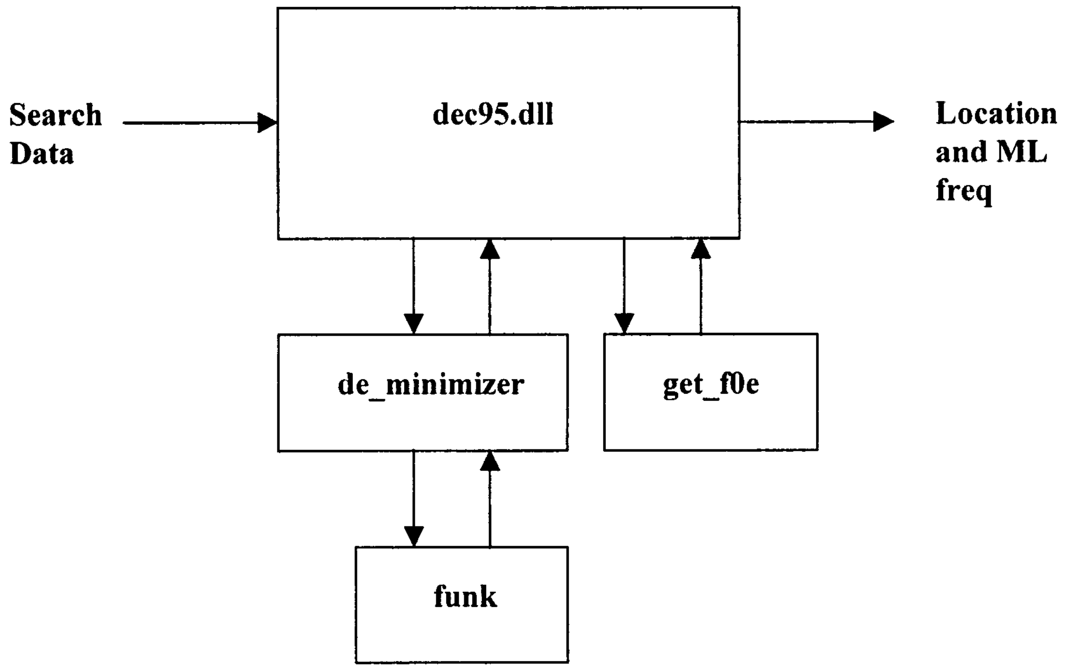
- Function get_f0e calculates the ML frequency at the emitter's location.
- Returns the low point and ML frequency to MATLAB.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION


Search Data

dec95.dll

Location and ML freq

de_minimizer

get_f0e

funk

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Testing:

- Test Sets 1-12 consist of the Comparison Tests.
- Test Set 13 consists of the Supplemental Parameter Tests.
- Test Set 14 consists of the Monte Carlo Tests.
- Test Set 15 consists of the AED Repeatability Tests.

ELECTRONIC WARFARE CONCEPTS
EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Comparison Tests :
  - The focus of the project.
  - Directly compare the performance of the two methods.
  - Start with a baseline run from a common starting point.
  - The starting points were chosen to illustrate various types of search behavior.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Comparison Tests (cont):
- Use identical noise in both MATLAB runs.
- Timed with the same stopwatch.
- Use the same data while adjusting the parameters of the DE simulator to try to improve its performance.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

Test #	MATLAB Program	C++ DLL	Pop Size	SF	CR	Iter	Time Elapsed (secs)	Average Error Distance (meters)
11	nmloc.m	nmc275	n/a	n/a	n/a	n/a	3.13	1.894432742614892 e +003
	deloc.m	dec95	10	.9	.5	10	2.82	1.624047131179735 e +005
12	nmloc.m	nmc275	n/a	n/a	n/a	n/a	3.05	1.894432742614892 e +003
	deloc.m	dec95	10	.5	.5	10	2.76	2.306490855796788 e +004
13	nmloc.m	nmc275	n/a	n/a	n/a	n/a	3.23	1.894432742614892 e +003
	deloc.m	dec95	10	.5	.7	10	2.68	1.627518825784596 e +005

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

Test #	MATLAB Program	C++ DLL	Pop Size	SF	CR	Iter	Time Elapsed (secs)	Average Error Distance (meters)
14	nmloc.m	nmc275	n/a	n/a	n/a	n/a	3.25	1.894432742614892 e +003
	deloc.m	dec95	10	.5	.3	10	2.63	1.57862488568820 e +003
15	nmloc.m	nmc275	n/a	n/a	n/a	n/a	3.20	1.894432742614892 e +003
	deloc.m	dec95	10	.3	.3	10	2.62	8.576546882938009 e +004
16	nmloc.m	nmc275	n/a	n/a	n/a	n/a	3.36	1.894432742614892 e +003
	deloc.m	dec95	10	1.1	.3	10	2.65	1.703117275665491 e +005

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Results:

- Mixed - DE method was almost always faster but not always as accurate.
- The DE method was faster and had a lower Average Error Distance (AED) in Comparison Test Sets 1,2,3,6,7,8,9, and 11.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

- ★ The Results (cont) :
  - The simplex method was slower but had a significantly lower AED in Test Sets 4, 5, 10, and 12.
  - No one set of parameters produced the best AED in all test sets.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Results (cont):

- Setting population size to 10, F to .5, CR to .3, and iterations to 10 results in the DE method being faster and more accurate in 7 of the 12 test sets.
- Setting population size to 10, F to .5, CR to .5, and iterations to 10 results in the DE method being faster and more accurate in only 5 of the 12 test sets.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ The Results (cont):

- Running greater or fewer numbers of Monte Carlo simulations has no real impact on the comparison of the two methods.
- Repeating tests with the same input data results in the same AED.

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION.

TECHNICAL PRESENTATION

★ Conclusions:

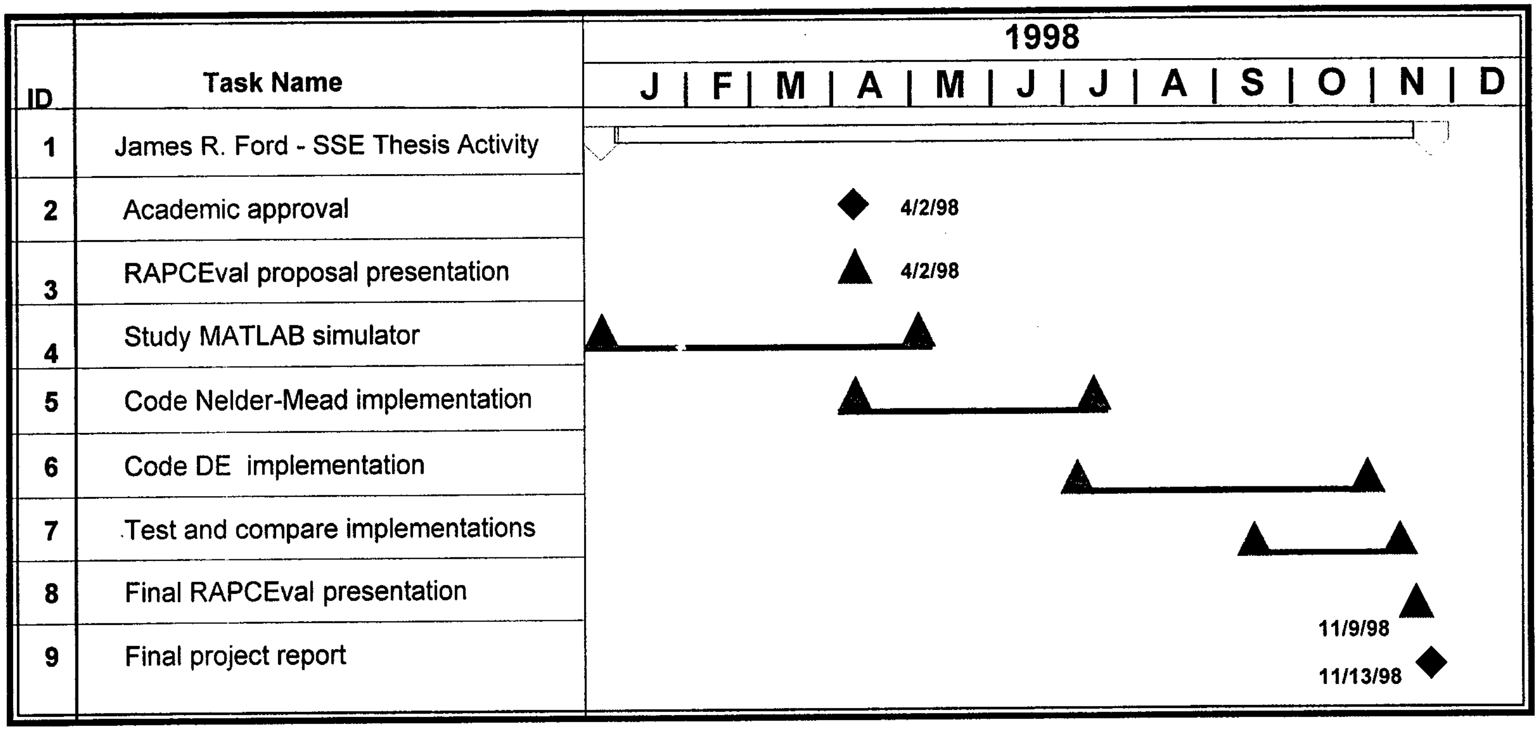
- DE was faster than the simplex method.
- In a majority of test sets, DE was more accurate than the simplex method.
- Further study of parameter combinations may make DE even more accurate.
- Further study of alternate crossover techniques may make DE faster and more accurate

ELECTRONIC WARFARE CONCEPTS EVALUATION PROGRAM
(RAPCEval)

James R. Ford
MSE/SE Program
9 November 1998

COMPARISION OF DIFFERENTIAL EVOLUTION TO THE SIMPLEX METHOD IN OPTIMIZATION DURING PASSIVE EMITTER LOCATION

Schedule Chart



5.4 *Thesis Report by Ron Brinkley*

A version with standard report style has been generated from the Master's thesis of Ron Brinkley. This is reproduced here on the next 83 pages.

BURST ERROR CORRECTION WITH REED-SOLOMON CODES

by

Ronald L. Brinkley Jr.

B.S.E.E, Old Dominion University, Norfolk, VA, 1992

A Report Generated from
A Thesis Submitted to the Graduate Faculty,
Mercer University, School of Engineering,
in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

Macon, Ga

1998

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Summary

Communication and storage systems, whether they are magnetic or optical recording devices, satellite or cellular mobile communication, or even military communication systems in hostile jamming environments, are faced with a need to correct burst errors. It is the intent of this research work to explore the efficiency of powerful burst error-correcting Reed-Solomon (RS) codes. First, an explanation as to the cause of burst errors, and a definition of burst errors is presented. This explanation is followed by an introduction to the finite field mathematics necessary in understanding RS codes. An introduction to RS codes and their ability to combat burst errors is given. Subsequently an overview of the most widely used interleaving techniques is presented to complete the preliminary portion of this thesis.

The guaranteed burst error-correcting capability of RS codes is first explored for single and multiple burst errors. This burst error-correcting capability is then used to define the burst error-correcting efficiency of selected RS codes. The idea of guaranteed burst error correction and burst error correction efficiency is then expanded to include block and convolutional interleaving techniques. Many useful tables and graphs of guaranteed burst error-correcting capabilities and efficiencies have been generated for the code designer.

2 Introduction

Why the concern for burst error correction? Other than deep space communications, an overwhelming majority of practical communication systems cannot be modeled as simple Additive White Gaussian Noise (AWGN) channels. In recording systems, whether they are magnetic or optical, defects in the recording material and recording mechanisms cause systematic errors. Wireless communication systems with fading effects are also faced with the occurrence of non-random errors. Jamming environments, especially intentional jamming as in military applications, are confronted with the effects of non-random errors. Communication channels that have non-random characteristics are considered to contain memory and tend to produce what is known as burst errors. In fact, most operational communication system applications exhibit a combination of random and non-random characteristics. These channels are termed diffuse or compound channels [Swe] and cannot be modeled as AWGN channels. Therefore, in answering the question why burst error correction, most real world communication systems need codes that can combat the common occurrence of burst errors.

2.1 Burst Errors

The purpose of this section is to provide the reader with a better understanding of the errors associated with digital communication systems. First a simple example and definition of errors that occur in binary codewords is given. A discussion of the two different types of possible errors follows. Then the definition of errors is expanded to include the word oriented RS codes. The section is concluded with brief examples of the classic error causing bursty channel.

A received encoded block of data can be interpreted as the linear combination of a valid codeword and an error pattern. This relationship is shown as R = C + E where R is equal to the received block of data, C is a valid codeword and E is an error pattern and the summation implies binary vector addition. In the binary case a '1' in the error pattern represents the occurrence of an error in that position. As an example, consider a communication system where the sixteen bit block of received data is R = 0010001110010000. Then suppose the original transmitted data was the valid codeword C = 0001010010001000. Therefore, because of errors in the channel, an error pattern E = 0011011100011000 is produced. It can be observed that the modulo two addition of R and E will correct the errors in R and return the intended block of data C.

Errors occur in communication systems in two different forms, random and burst errors. An example of a random error is given in Figure 2.0 where n bits of a received data stream are given and x represents a bit error. There are five random errors shown in Figure 2.0 and the error pattern would be E = 0001010100011000 with weight e equal to five. Note that for no known deterministic reason, two of the random errors occurred adjacent to each other. Additive white Gaussian noise channels are random error channels.

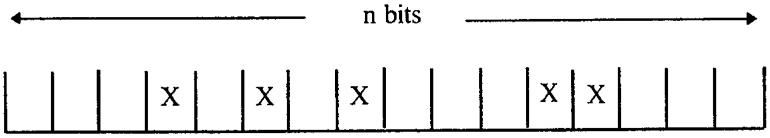


Figure 2.0 Random Errors

Burst errors are conventionally defined in terms of their length, and can be viewed as a cluster of individual errors. The length of a burst error is equal to the number of bits that span the error cluster. Therefore, a burst error of length b is defined as the number of bits in the error cluster starting from the first bit in error (first '1' in the error pattern) and ending with the last bit in error (last '1' in the error pattern) [Lin]. For example the burst error shown in Figure 2.1 is of length b = 11 bits and has an error pattern of E = 0011011100011000 with weight e equal to seven.

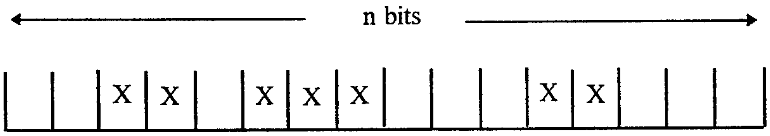


Figure 2.1 Conventional Burst Error

More than one burst error may be imposed onto a codeword. When this happens, the codeword is said to contain multiple burst errors. For example in Figure 2.1 above, the codeword may be considered to contain three burst errors of which two are of weight two and one burst error of weight three. In keeping with the notation used in [Ben], a codeword may contain s burst errors ($s \geq 1$) of length b bits ($b > 1$) and each burst of weight e (e > 1).

To resolve any ambiguities about defining burst errors, the idea of a guard space is now explained. Just as a channel with memory produces error clusters, the channel also produces spaces where no errors occur. These clusters of error free bits are indicated as sequences of all zeros in the error pattern E and are called guard spaces. Therefore, a burst error b is defined in relation to the surrounding guard space g [Gal1]. For example, in Figure 2.2 the knowledge of the guard spaces dictates the definition of the possible burst errors. The bits of information indicated by x's in Figure 2.2 can represent burst errors of length less than or equal to b.

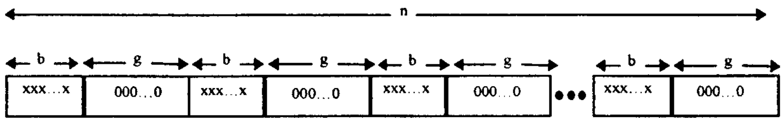


Figure 2.2 Burst Errors with Guard Spaces

The definition of burst errors will now be expanded for the case of RS codes that are examined in this report. Reed-Solomon codes are word-oriented codes, in which every codeword is composed of m bit symbols. We assume that the random error-correcting capability of a RS code is denoted by t. If any one of the m bits of a given symbol is in error, then the entire symbol is considered to be erroneous. Therefore, it is customary to measure random errors in a RS code according to the number of symbols in error and not the number of bits. Figure 2.3, a representation of a n symbol codeword, indicates two (t = 2) random errors.

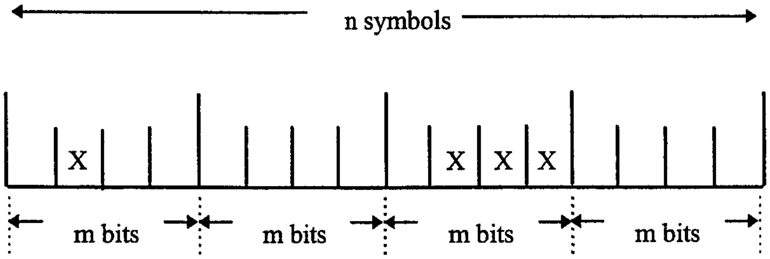


Figure 2.3 Word Oriented Random Error

Let's assume a burst error of length b bits corrupts an RS code and t consecutive symbol errors occur. To define the relation between b and t for burst errors in RS codes, assume all burst errors are of the worst case. Here the worst case burst error is defined as a burst error in which all b bits of the error cluster are in error. The worst case burst error would have an error pattern with a burst pattern of all '1's (e.g. E = 0000111111110000 has burst error length of b = 8 bits as shown in Figure 2.4). If the burst error occurs exactly over t symbols as shown in Figure 2.4, then the length of the burst is equal to the number of bits over t symbols (b = t·m bits). However, a burst error of the same length can corrupt up to t + 1 symbols as shown in Figure 2.5. The burst error-correcting capability of RS codes will be further examined in Section 4 of this document.

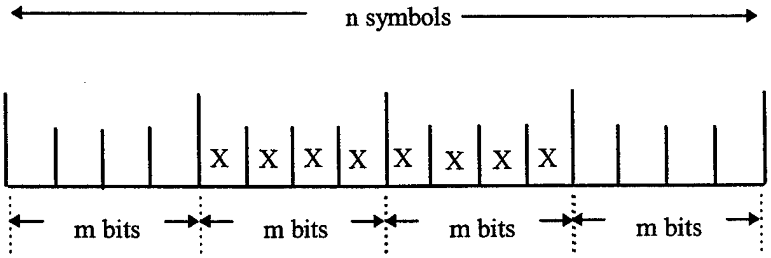


Figure 2.4 Word Oriented Burst Error Located Over Exactly (t) Symbols

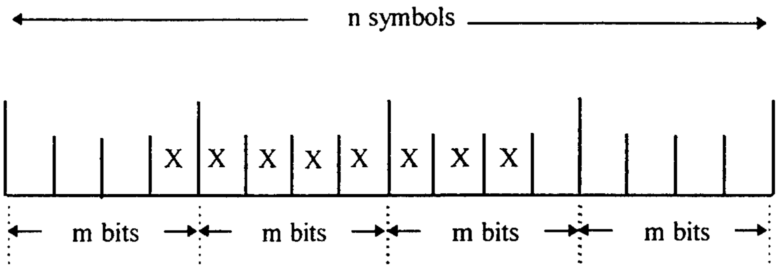


Figure 2.5 Word Oriented Burst Error Over (t + 1) Symbols

2.2 The Bursty Channel

There are several systems of interest that operate over the classic bursty channel which use burst error-correcting RS codes. Burst errors are found in both magnetic and optical recording systems. In digital mobile communication systems, such as cellular and satellite communication systems, burst errors are a major concern. And in military communications, intentional jamming may introduce burst errors into the channel. Burst errors can have serious adverse effects on signal reliability, and hence the need for burst error-correcting codes.

With increases in recording speeds and data storage densities for magnetic recording systems, burst errors have become common place. Dust and defects in the recording material and the read/write heads of a magnetic disk recording system are usually to blame for the occurrence of burst errors. These defects have been known to produce burst errors of hundreds of bits in length [Lin, chap 16]. One example of the use of RS codes in magnetic recording systems is the IBM 3370 disk storage system [Mee].

Optical recording systems have used RS codes extensively for the correction of burst errors. One digital recording system that owes its existence to RS codes is the compact disk (CD) recording system. With the reduction in size of the recording media (a compact disk has an outer diameter of 120mm [Car]), the density of the data has become enormous. The track width of a CD is roughly 0.6 μm and one bit of audio data covers 1.0 μm^2 of disk space [Car]. In comparison to the LP record track, the CD has a track ratio of approximately 60:1 [Bae]. That is, sixty tracks can be placed on a CD in the same amount of space once used by one LP track. With these data densities, it is clear that the slightest defect or scratch may cause extremely long burst. If it were not for the error control scheme known as Cross Interleaved Reed-Solomon Code (CIRC), the Compact Disk recording system could not deliver nearly the quality of music it now does. The CIRC error correction scheme has the capability to correct burst errors of approximately 4000 bits in length, this corresponds to roughly 2.5 mm of track length [Imm2]. For a complete description of the CIRC error control method as applied to the compact disk refer to [Vri] and [Imm1].

Since the onset of the digital communication era, mobile communication has become a fast developing area. Popular topics in the mobile communication arena include cellular telephone systems, personal communication systems (PCS), and satellite communication systems. All of these communication systems are faced with channel problems such as multipath effects, channel fading and Doppler shifts. In light of such adverse channel impairments, burst errors are very common. One example of the use of RS codes in mobile communications is in the cellular digital packet data (CDPD) system [Kam1] [Kam4]. For a more detailed review of the signal characteristics of a mobile communications channel, see [Vau].

Fading in a mobile communication channel can be statistically modeled as a Rayleigh distributed process [Wic3]. Fades are categorized into what are known as slow and fast fades. The size of the burst error produced is directly related to the length of the fade imposed on the system. Slow fades are longer in duration relative to the symbol rate of the communication signal being affected. Therefore, a slow fade will produce multiple symbol errors. It has been shown that mobile communication systems with the aid of error control coding and interleaving techniques have improved performance [Yue].

In today's military combat environment, the ability to communicate determines winners or losers. For this reason electronic warfare (EW) and information warfare (IW) are targets of many discussions and much research. Of particular interest is the survival of a communication system in the presence of intentional jamming. There are many different types of jamming techniques, however pulsed jamming has shown to be the most optimal for creating interference [Tor]. Pulsed jamming concentrates more of the total signal power into each pulse unlike a continuous wave jammer whose power is more spread out. If the pulse width of the jamming signal is of the same order as the symbol transmission rate, then approximately one symbol will be affected per jamming pulse. If the pulse width is greater than the symbol transmission rate, then multiple symbols can be affected per jamming pulse causing burst errors. As illustrated in Figure 2.6 below, jamming signal $J_1(t)$ has pulse width approximately equal to the symbol rate of the communication signal S(t). Jamming signal $J_2(t)$ has a pulse width approximately equal to three times the symbol rate of S(t) thus causing burst errors.

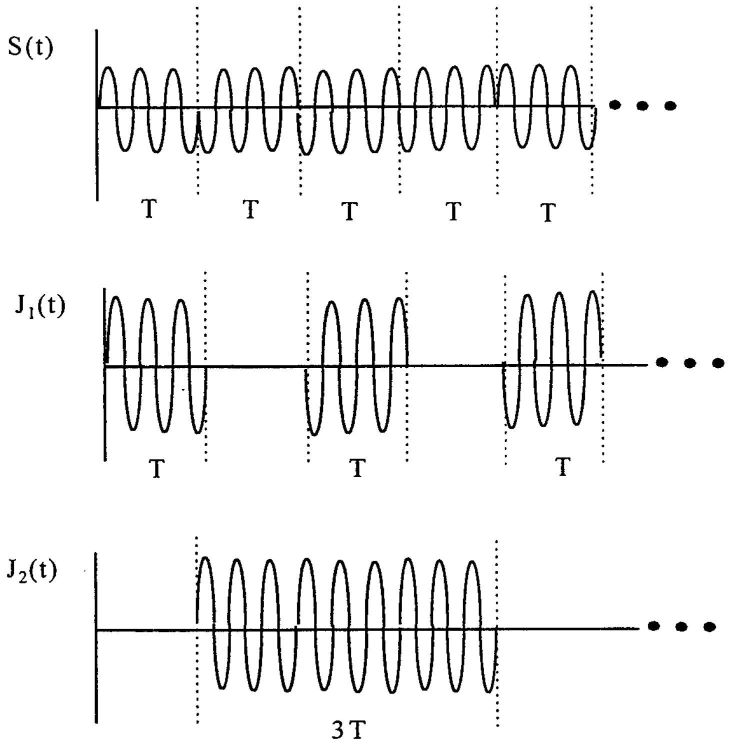


Figure 2.6 Pulsed Jamming

Techniques such as spread spectrum communication have been employed with good success in defending against intentional jamming. However, it can be shown that using spread spectrum with the addition of error control coding and interleaving techniques, the affect of pulsed noise jamming can almost be eliminated [Zie, chap 11&12] [Pur]. One example of a military communication system that uses RS codes is the Joint Tactical Information Distribution System (JTIDS) being supported by all branches of the military and NATO [Zie, chap 13].

With the ever increasing complexity of communication systems, the powerful burst error-correcting RS codes will no doubt see more applications in digital recording systems, mobile communication systems, and in communication environments with interference.

3 Finite Field Mathematics

To understand Reed-Solomon codes an introduction to Galois (pronounced 'gal-wa') field algebra is presented. To learn and understand Galois fields it is important to have a foundation in the laws of abstract algebra that govern Galois field theory. However, it is by no means the intention of this section to present an in-depth coverage of abstract algebra. The purpose of this section is to present an overview of basic ideas and laws needed in understanding Galois field algebra as applicable to RS coding. The reader is encouraged to review any of the following references for a more detailed analysis of abstract algebra and Galois field theory [Cla], [Lin], [McE1], [Pet1], and [Wic1].

3.1 Groups

A group (G) is a set in which a binary operation ($\cdot$) has been defined. Binary in the sense that the operation is performed on two elements. The resultant of the binary operation is also another element in the set. This unique property is called closure. Three other properties are required in order for a set to form a group. The law of associativity must be upheld. There must exist an identity element in the group. Every element in the group must have a unique inverse. Optionally, if a group satisfies as a fifth property the law of commutativity, the group is said to be commutative and called an Abelian group. In summary, the follow properties apply:

a) Closure: $\forall\ a,b \in G;\ \ a \cdot b = c;\ \text{ where } c\ \in G$

b) Associativity: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

c) Identity: $\forall\ a \in G$ there exist $e \in G$ such that $a \cdot e = e \cdot a = a$

d) Inverse: $\forall a \in G$ there exist $a^{-1} \in G$ such that $a \cdot a^{-1} = a^{-1} \cdot a = e$

e) Commutativity: $\forall a,b \in G$; $a \cdot b = b \cdot a$ (optional)

3.2 Rings

A ring (R) is a set of elements with two binary operations (+ and ·) defined. There are three properties that must hold for a set to form a ring. The ring must form an Abelian group under (+) with the additive identity element indicated by (0). The law of associativitity must be upheld for the binary operation (·). Also, the binary operation of (·) must be distributive over the binary operation of (+). Additionally, if the operation (·) is commutative, then the ring is called a commutative ring. If the operation (·) has an identity element labeled as (1), then the ring is said to be a ring with identity. If the ring meets both of the additional properties, then it is called a commutative ring with identity. The ring is an important building block for error control coding. It is the ring that allows for the development of the polynomial. Summarizing, the following properties apply:

a) Commutative group under addition (+) with identity element (0)

b) Associativity under multiplication (·): $\forall a,b,c \in R$; $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

c) Multiplication (·) distributes over addition (+): $\forall a,b,c \in R$;
$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

d) Commutativity under multiplication (·): $\forall a,b \in R$; $a \cdot b = b \cdot a$

e) Multiplication (·) has an identity element labeled as (1)

3.3 Fields

Now using the concept of the group, the algebraic structure of a field can be introduced. A field (F) is defined such that it can be thought of as a combination of a commutative group under addition (+) and a commutative group under multiplication ($\cdot$). Basically, this makes the field a set of elements in which the operations of addition (+), multiplication ($\cdot$), subtraction (-), and division ($\div$) are defined. The follow properties must be satisfied to define a field:

a) F is a commutative group under (+) with additive identity element (0).

b) The set of nonzero elements in F is a commutative group under ($\cdot$) with multiplicative identity element (1).

c) Multiplication ($\cdot$) distributes over addition (+), that is $\forall$ a,b,c $\in$ F;
$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

In block coding, fields of finite order are key algebraic structures of interest. A finite field is referred to as a Galois field that is named after Evariste Galois, the discoverer of finite fields. A Galois field is denoted by GF(q), where the order of the field is q.

It should be obvious from the definition that a field needs to contain at least two elements, the additive identity element (0) and the multiplicative identity element (1). The field that is formed from these two elements is known as the binary field GF(2). The binary field and its extensions are very important and have many applications in engineering. Before presenting the binary extension field, primitive elements and primitive polynomials must be introduced. But first the concept of a vector space will be introduced.

3.4 Vector Spaces

Let V represent a set of elements called vectors and F represent a field of elements called scalars. Two additional operations are introduced that can be applied to the elements of V and F. Let (+) be a binary additive operation that maps linear combinations of vectors v_1, v_2, v_3, ..., contained in V into a single vector v_i also contained in V. This binary additive operation is referred to as vector addition. And let ($\cdot$) represent a binary multiplicative operation performed on a scalar a_i contained in F and a vector v_i contained in V, such that ($\cdot$) maps a_i and v_i into a vector $u_i = a_i \cdot v_i$ also contain in V. A vector space is formed if the following rules are meet:

a) V forms a commutative group under (+).

b) $\forall$ a $\in$ F and $\forall$ v $\in$ V; $a \cdot v = u \in V$.

c) Associativity under multiplication ($\cdot$): $\forall$ a,b $\in$ F and $\forall$ v $\in$ V;
$(a \cdot b) \cdot v = a \cdot (b \cdot v)$

d) Multiplication ($\cdot$) distributes over addition (+): $\forall$ a,b $\in$ F and $\forall$ u,v $\in$ F;
$a \cdot (u + v) = (a \cdot u) + (a \cdot v)$, and $(a + b) \cdot u = (a \cdot u) + (b \cdot u)$.

e) $\forall$ v $\in$ V, $1 \cdot v = v \cdot 1 = v$; the scalar multiplication identity element is 1.

In a vector space, F is commonly referred to as the "ground field" or "base field" of the vector space. The properties of the vector space allow for the existence of what is known as a vector n-tuple. A n-tuple has the form $v = (v_0, v_1, v_2, ..., v_{n-1})$ where the elements (v_i) are from the ground field F. It is the n-tuple representation of a vector space that is so important in implementing error control codes due to the association of binary words with vector n-tuples. For example, if the ground field is taken to be binary

(GF(2)), then 00101 is a five bit binary word can be represented by v = (00101) a 5-tuple taken from the vector space V_5. Of course, here the vector space is finite and composed of 32 (2^5) vector 5-tuples.

Some other important ideas of vector spaces associated with error control coding are the dimension of the vector space, the basis of a vector space, and the spanning set of a vector space. A spanning set $S = (v_0, v_1, v_2, ..., v_{n-1})$, is a set of vectors from which the linear combination of the vectors ($a_0 \cdot v_0 + a_1 \cdot v_1 + a_2 \cdot v_2 + ... + a_{n-1} \cdot v_{n-1}$; where $a_i = 0$ or 1) forms all other possible vectors of the vector space. Going another step further, a basis is a set of vectors of minimum order that spans the vector space. Therefore, a basis is a spanning set that has no additional redundant vectors. A basis set is always a spanning set but a spanning set may not be a basis. Finally, the dimension of a vector space is equal to the number (k) of vectors in the basis set. This of course leads to the fact that each vector of a vector space is unique.

3.5 Primitive Element

Elements of a finite field have individual cardinalities. Defining $a^2 = a \cdot a$, $a^3 = a \cdot a \cdot a$, etc..., the order of an element (n = ord(a)) is the value that satisfies $a^n = 1$, the identity element. Of particular importance, are the elements in field GF(q) with order q-1, which are called primitive elements. It can be proven that, since a primitive element has order equal to one less than that of the size of the field, consecutive powers of a primitive element can represent all nonzero elements of a field. It also has been proven that there exist at least one element in GF(q) with order q-1, and hence there is always at least one primitive element associated with a finite field. The importance of primitive elements will be further illustrated in the section on the development of binary extension fields.

3.6 Primitive Polynomials

It has been shown thus far that a field GF(q) can be represented by 0 and q-1 consecutive powers of a primitive element (α). Remembering the additive and multiplicative nature of the field, a field can be composed of polynomials of m^th^ degree whose coefficients are taken from a defined finite field. The notation GF(q)[x] represents such a polynomial f(x) with coefficients taken from GF(q). As defined earlier, this collection of polynomials is called a commutative ring with identity.

A polynomial f(x) which cannot be factored into the product of lower degree polynomials in GF(q)[x], is an irreducible polynomial in GF(q). That is to say, the polynomial f(x) is in the simplest form over GF(q)[x]. For example, $f(x) = x^2 + x + 1$ is irreducible in GF(2)[x], however f(x) can be factored in GF(4)[x]. With this illustration, it is clear that when defining irreducibility the field must be specified.

An irreducible polynomial p(x) of arbitrary degree m, defined in GF(p)[x] where p is a prime, is called a primitive polynomial if the smallest value of n for which p(x) divides $x^n - 1$ is $n = p^m - 1$. For example, $p(x) = x^3 + x + 1$ is a primitive polynomial in GF(2)[x] and the smallest degree polynomial of the form $x^n - 1$ that p(x) divides is $x^7 - 1$. Note that all primitive polynomials in GF(q)[x] are irreducible, but not all irreducible polynomials are primitive. An extremely important property of primitive polynomials, which will be exploited in the development of RS codes, is that the roots (α^i) of a primitive polynomial p(x) in GF(p)[x] have order $p^m - 1$. Recall, an element of order $p^m - 1$

forms all nonzero elements of a field GF(q) of order $q = p^m - 1$. Summarizing, an m^{th} degree polynomial in GF(p)[x], is a function of x with coefficients in GF(p) and has roots (α^i) over the field $GF(p^m)$.

3.7 Binary Extension Fields

In the study of error control coding we are interested in polynomials in GF(p)[x] that have coefficients in GF(p) and roots (α^I) in $GF(p^m)$, where p is equal to the binary ground field. Simply stated, $p = 2$. By defining a field in this fashion, the elements of the field become binary n-tuples of length m over $GF(p^m)$. Essentially, binary words of length m are defined. Using a primitive polynomial p(x) of degree m, and the fact α is a primitive root of p(x), the construction of a binary extension field of order p^m will now be illustrated.

a) Step 1: Choose a primitive polynomial of degree m, where m is equal to the size of the binary word of interest. For instance, if working with 3-bit binary symbols, choose a primitive polynomial of degree three. Choose $p(x) = x^3 + x +1$.

b) Step 2: Let α be a primitive root of p(x). Therefore, $p(\alpha) = \alpha^3 + \alpha +1 = 0$ and

   $\alpha^3 = \alpha + 1$.

c) Step 3: Recall that p(x) must divide $f(x) = x^7 - 1$; where $n = 2^3 - 1$. If p(x) divides f(x) and has α as a root, then α must also be a root of f(x). Therefore,

   $f(\alpha) = \alpha^7 - 1 = 0$ and hence $\alpha^7 = 1$.

d) Step 4: Include the zero element. Since the consecutive powers of the primitive element α only form the nonzero elements of the field, the zero element must be included.

e) Step 5: Begin construction of the binary extension field using the properties of the above steps to keep all elements in their reduced format. The field elements are

$$0,\ \alpha^0 = 1,\ \alpha^1 = \alpha,\ \alpha^2,\ \alpha^3 = \alpha + 1,\ \alpha^4 = \alpha^3\alpha = (\alpha+1)\,\alpha = \alpha^2 + \alpha,$$

$$\alpha^5 = \alpha^4\alpha = (\alpha^2 + \alpha)\alpha = (\alpha^3 + \alpha^2) = (\alpha + 1 + \alpha^2) = \alpha^2 + \alpha + 1, \text{etc...}$$

Table 3.0 Construction of Binary Extension Field GF(8)

EXPONENTIAL REPRESENTATION	POLYNOMIAL REPRESENTATION	BINARY N-TUPLE REPRESENTATION
0	0	000
1	1	100
α	α	010
α^2	α^2	001
α^3	$\alpha + 1$	110
α^4	$\alpha^2 + \alpha$	011
α^5	$\alpha^2 + \alpha + 1$	111
α^6	$\alpha^2 + 1$	101

4 Reed-Solomon Codes

With the evolution of today's complex communication systems, it is apparent that a strong error-correcting scheme is needed to combat burst errors. Reed-Solomon codes, once thought of as only a great non-practical mathematical theory, have proven to be the answer to the burst error-correcting problem. Reed-Solomon codes received their glory from such applications as the deep space exploration missions of Galileo and Voyager [McE2], the compact disc optical recording system [Imm2], and many applications in magnetic recording systems [Her]. With the advances being made in communication systems, RS codes are sure to see many future applications. The remainder of this section will attempt to explain what RS codes are and how they are encoded and decoded.

Reed-Solomon codes are a subclass of linear block codes. Like any other block codes, the RS codes append parity information to a block of data so that at the receiver error detection and correction is possible. Reed-Solomon codes are typically specified by their length (n) and dimension (k). Therefore, we specify a particular RS code as a (n,k) RS code. Recall from Section 3 that k is the number of elements in the basis of the vector space and is an indication of the number of data symbols in the original information block. If n (where $n > k$) is equal to the number of symbols in the encoded block, then there is n-k parity symbols appended to the k original data symbols. Also, a RS code is said to have a code rate defined as $R_c = k / n$. Refer to Figure 4.0 below describing the structure of a (7,3) RS codeword.

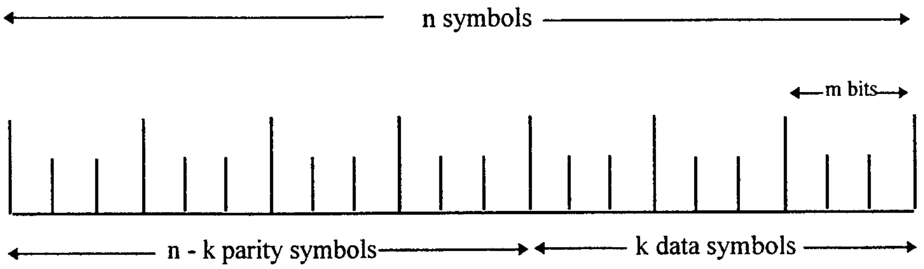


Figure 4.0 Structure of a (7,3) Reed-Solomon Code

4.1 Error-correcting Capability

Reed-Solomon codes are word oriented linear block codes capable of correcting t random symbol errors. Reed-Solomon codes have the ability to correct burst errors of length equal to t consecutive symbols. Since RS codes are word oriented, that is they are composed of m-bit symbols, they can correct bursts of maximum length equal to $t \cdot m$ bits. It can be shown that the guaranteed length for burst error correction of a RS code is $b = m(t - 1) + 1$ bits. This can be expanded to guaranteed double burst error correction ($b = m(\lfloor t/2 \rfloor - 1) + 1$) and guaranteed triple burst error correction ($b = m(\lfloor t/3 \rfloor - 1) + 1$) [Kam2]. The definitions given here for the guaranteed (not maximum) burst error correction will be utilized in the following sections to define the efficiencies of RS codes.

4.2 Maximum Distance Codes

Reed-Solomon codes have the important property of being Maximum Distance Separable (MDS) codes. It has been shown that RS codes satisfy the Singleton bound with equality and therefore are considered Maximum Distance codes [Wic1]. Maximum distance codes have the highest obtainable error correction capability of any block code of the same redundancy [Sin].

Hamming weight has been defined as the number of non-zero elements in a binary symbol. For example, the symbol (11010001) has a Hamming weight of four (denoted W = 4). Hamming distance is defined as the number of positions between two binary symbols that are in difference. For example, the Hamming distance between the two binary symbols (11010001) and (10001010) is equal to five (denoted D = 5). The Hamming distance is also shown to be the Hamming weight of the modulo-two addition of any combination of symbols. For example, (11010001) + (10001010) = (01011011) and the Hamming weight of the resultant (01011011) is five as before. The minimum distance (d_{min}) of a code is defined as the minimum Hamming distance between all possible valid pairs of codewords in a given vector space. Recall that a vector space defined over a binary extension field is composed 2^n possible code-vectors, where n equals the code length. For the same vector space, there are 2^k valid codewords, where k is the dimension of the vector space. Therefore, there are at least d_{min} bits separating valid codewords. Maximum distance codes, are codes that have the largest possible minimum distance of separation between valid codewords. In fact it has been shown by [Sin] that if a code has $d_{min} = n - k + 1$, it is a MDS code. This can be explained by visualization of any two codewords n symbols in length with k data symbols and n-k parity symbols. To be valid codewords, the k data symbols must at minimum differ in one position. Also, the most desired case would be when all n-k parity symbols differ between two valid codewords. Combining the above two statements, the largest minimum separation between valid codewords is equal to the maximum possible distance between all parity symbols

(D = n-k) and the single distance between data symbols (D = 1), yields $d_{min} = n - k + 1$.

4.3 Encoding

Since the introduction of RS codes in 1960 by Irving Reed and Gus Solomon [Ree], there have been many different approaches for encoding of RS codes. These differences are due to the fact that there existed no efficient methods of decoding RS codes, therefore new encoding methods where created that gave way to better decoding algorithms. The original method defining RS codes as polynomials over certain finite fields can be reviewed in Reed and Solomon's original 1960 paper [Ree]. There is also a summary in [Wic2]. The method of encoding RS codes that will follow is the extended non-binary BCH code approach. There is a reason for choosing the extended BCH approach. In most applications this approach leads to more efficient decoding algorithms.

In order to construct an RS code as an extension of a BCH code, one must use the generator polynomial approach. RS codes are represented by generator polynomials g(x). These polynomials are constructed over $GF(2^m)$ and therefore have as roots the primitive elements α^i. In order to be MDS, g(x) must meet the BCH bound. Therefore, the α^i primitive elements must be 2t = n-k consecutive primitive roots of g(x). Such a polynomial must be of the form $g(x) = (x - \alpha^b)(x - \alpha^{b+1})(x - \alpha^{b+2})\ldots(x - \alpha^{b+2t-1})$. For narrow-sense RS codes, g(x) can be represented by the following equation (in which b equals one):

$$\mathbf{g(x) = \prod_{i=1}^{2t} (x - \alpha^i)}$$

When all the factors of $g(x)$ are multiplied out to form the generator polynomial, it will be a polynomial of degree $2t = n - k$ defined over the field $GF(q)$ where $q = 2^m$ and will thus have q^k valid codewords.

By defining RS codes as extended BCH codes using the polynomial approach, the cyclic properties can be utilized. The generator polynomial has the form $g(x) = g_0 + g_1x + g_2x^2 + ... + g_{n-k}\, x^{n-k}$, and a block of information data can be represented as a polynomial of the form $d(x) = d_0 + d_1x + d_2x^2 + ... + d_{k-1}\, x^{k-1}$. Therefore, a codeword C is encoded as $c(x) = d(x)g(x)$ and is represented as $c(x) = c_0 + c_1x + c_2x^2 + ... + c_{n-1}x^{n-1}$. Typically RS codes are put into a systematic format in order to reduce the number of steps needed for encoding. A codeword is in systematic format if it has the form $C = (p_0, p_1, p_2, ... p_{2t-1}, d_0, d_1, d_2, ... d_{k-1})$, where the data symbols (d_i) are grouped together and the n-k parity symbols (p_i) are also grouped together. Using the cyclic properties and polynomial representations, the codeword has the form $c(x) = p(x) + x^{n-k}d(x)$, where multiplying by x^{n-k} simply performs a cyclic shift of the data symbols into their proper position. In the above polynomial representation the parity information is equivalent to the remainder of $x^{n-k}d(x)/g(x)$, that is $p(x) = x^{n-k}d(x) \bmod g(x)$ [Has]. The resultant code word polynomial $c(x)$, systematic or non-systematic, has as roots the same $2t$ consecutive powers of α as does the generator polynomial. This property allows for an easy way of determining if a valid codeword has been received and for efficient decoding algorithms of RS codes.

4.4 Decoding

Since the introduction of RS codes in 1960, there has been a rush by researchers to find an efficient decoding algorithm for these codes. The simple techniques used in linear block codes just do not work. For instance using the syndrome look-up table method is incomprehensible due to the number of syndrome computations needed. For instance, a (63,53) RS code requires approximately 10^{20} syndrome calculations and has approximately 10^{95} (64^{53}) valid codewords.

In the original paper by Reed and Solomon [Ree] they introduced a possible method for decoding. This decoding method involved the solution of a set of k equations in k unknowns. The method proved to only be useful for the simplest RS codes. Shortly after, Peterson [Pet2] presented a popular decoding method for BCH codes. Gorenstein and Zierler [Gor], and later Chien [Chi] and Forney [For2], expanded the Peterson algorithm for BCH codes to the non-binary case for RS codes. These codes, although much more efficient than any others, were only applicable to codes with a small number of errors.

It was not until Berlekamp [Ber2] presented his version of an efficient decoder for use with non-binary BCH and RS codes that the application of RS codes became more than a mathematical exercise. One year later Massey [Mas] showed that the Berlekamp algorithm could be realized as a technique of finding the shortest Linear Feedback Shift Register (LFSR) capable of generating a given sequence. The combination of the two ideas has now become what is known as the Berlekamp-Massey algorithm. Although the Berlekamp-Massey algorithm is not the most mathematically simple to understand, it produces an efficient technique for decoding RS codes. The explanation of the Berlekamp-

Massey algorithm that follows is a simple overview and the reader is referred to [Ber1], [Ber3], [Has], [Lin], [Nas], [Wic1].

The Berlekamp-Massey algorithm is a four-step process for error correction in RS codes. Reed-Solomon codes, which unlike BCH codes are word oriented, require an additional step of determining the error value. The first step in the Berlekamp-Massey algorithm is to calculate the syndromes. Next the error locator polynomial must be obtained. Once the error locator polynomial has been obtained, the error location can be found. Finally, the error values are calculated. These four steps are explained in the following paragraphs.

Recall that a received vector has the form $r(x) = r_0 + r_1x + r_2x^2 + \ldots + r_{n-1}x^{n-1}$. The received vector can be represented has the combination of a valid codeword and an error pattern, $r(x) = c(x) + e(x)$ where $c(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-1}x^{n-1}$ is a valid codeword and $e(x) = e_0 + e_1x + e_2x^2 + \ldots + e_{n-1}x^{n-1}$ is an error pattern. Since the codeword c(x) has as roots the 2t consecutive powers of α, the received vector will have the same roots if no errors are present. That is $r(x) = c(x)$ when e(x) is equal to all zeros. Calculating the syndromes is done by evaluating the received vector at all 2t consecutive roots of the codewords, $S_i = r(\alpha^i) = c(\alpha^i) + e(\alpha^i)$ for $i = 1,2,\ldots,2t$. Since S_i will equal zero except where $e(\alpha^i)$ is not zero, the syndrome is calculated from the equation below to find out if errors exist.

$$S_i = \sum_{k}^{n-1} e_k(\alpha^i)^k, \quad i = 1,2,\ldots,2t$$

The second step of the Berlekamp-Massey algorithm is to obtain the error locator polynomial $\Lambda(x)$. As the name implies, this polynomial gives information as to the locations of the errors in the received code-vector. It is actually the integer power of the inverse root of the locator polynomial that reveals the error location. Simply stated: find the roots of the error locator polynomial, find the inverse of the roots, and then the power of the resultant is the location of the error. For example if the root was α^2 defined over GF(8), then the inverse of α^2 is α^5 and the error is at the fifth location.

It is the method of finding the error locator polynomial that makes the Berlekamp-Massey algorithm so complex. The error locator polynomial is found through an iterative process, usually performed in tabular form, whereby the solution for $\Lambda(x)$ is checked for correctness after each iteration. If the error locator polynomial $\Lambda(x)$ is not correct, then a correction factor is added to $\Lambda(x)$ and the process continues for 2t iterations. The error locator polynomial $\Lambda(x)$ is given below for ν errors, where Λ_i is a coefficient of the form α^j.

$$\Lambda(\mathbf{x}) = (1+\Lambda_1 \mathbf{X}+\Lambda_2 \mathbf{X}^2+\ldots+\Lambda_\nu \mathbf{X}^\nu)$$

After the error locator polynomial $\Lambda(x)$ has been found, the third step is to find all of the roots of the locator polynomial. This can be performed by factoring $\Lambda(x)$ into the form $\Lambda(x) = (1+\Lambda_i X^i)(1+\Lambda_{i+1} X^{i+1})\ldots(1+\Lambda_\nu X^\nu)$ or just by evaluating $\Lambda(x)$ at all power of α. Once the roots of $\Lambda(x)$ are found, the inverses of the roots reveal the error locations.

The fourth step in the Berlekamp-Massey algorithm is that of calculating the actual error values at the now known error locations. In order to perform this task, the polynomial known as the error magnitude polynomial Z(x) is defined using the error locator polynomial coefficients and the syndrome values. The error magnitude polynomial is defined as $Z(x) = 1 + (S_1 + \Lambda_1)x + (S_1 + \Lambda_1 S_1 + \Lambda_2) x^2 + \ldots + (S_v + \Lambda_1 S_{v-1} + \ldots + \Lambda_v) x^v$ for v errors. This error magnitude polynomial is used in the following equation to find the error values. The numerator of the error value equation is the error

$$e_{ij} = \frac{Z(X^{-i}_j)}{\prod\limits_{\substack{i=1 \\ i \neq j}}^{v} (1 + X_i X^{-i}_j)}$$

magnitude polynomial evaluated at the root of the error locator polynomial. The denominator is equal to v factors where X_i represents all roots of the error locator polynomial that are not at the current location being calculated, and X^{-i}_j is the inverse root of the error locator polynomial for the current error value. Once all v values for e_{ij} are found the error polynomial can be constructed and added back to the received vector for error correction. The error polynomial would have the form $e(x) = e_i + e_{i+1}x^{i+1} + e_{i+2}x^{i+2} + \ldots + e_{n-1}x^{n-1}$ and the corrected codeword are obtained by $c(x) = r(x) + e(x)$.

5 Interleaving

Communication systems discussed up to this point have been shown to operate over channels with memory. Furthermore, it is the memory in the channel that is to blame for the majority of burst errors. Interleaving is a technique that can be applied in an attempt to randomize the transmission of encoded signals over the bursty channel. This randomizing of the data allows the channel to appear memoryless. The remainder of this section will discuss the two most widely applied interleaving techniques, block interleaving and convolutional interleaving.

5.1 Block Interleaving

Block interleaving attempts to scramble the codeword over the channel so that the channel appears to be random. Block interleaving, although appearing random to the channel, is truly a one-to-one deterministic ordering of the codeword symbols [Ram]. This one-to-one sequencing of the data provides for a very efficient process in that unscrambling of codewords is easily carried out. The unscrambling, or deinterleaving as the process will be called hereafter, is essentially an interleaver used in reverse order. All of the important properties will be brought forth in the following discussion of the block interleaving process.

A block interleaver is nothing more than a buffering mechanism inserted between the error coding circuit (channel encoder) and the modulator. Similarly, the de-interleaver is a buffer placed after the demodulator and before the channel decoder. Figure 5.0 is a simplified block diagram of a typical communication system. The interleaver and deinterleaver can be thought of as a d x n matrix where d is the depth of the interleaver indicating the number of codewords stored. Each cell in the matrix represents one symbol of length m bits. Depending on the size of the interleaver desired, it is normally implemented with memory devices. Therefore, in the case of RS codes, memory devices of $n \cdot m \cdot d$ bits are required.

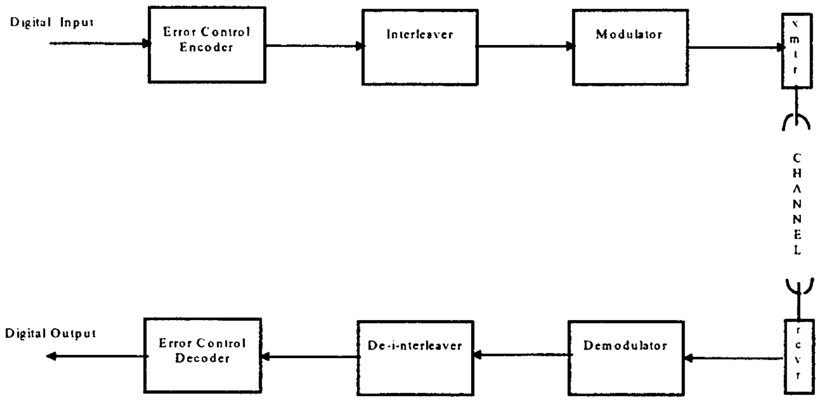
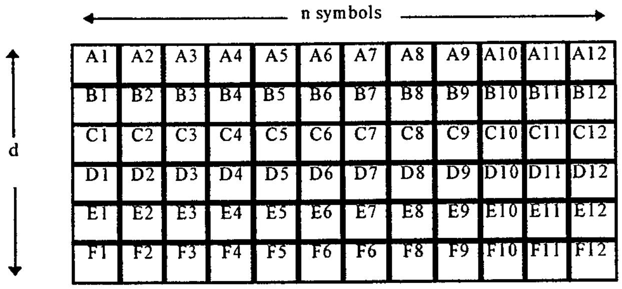


Figure 5.0 Simplified Communications System Block Diagram

The operation of the block interleaving process, as illustrated in Figure 5.1, is simply the inputting of codewords into the interleaver matrix in a row-by-row fashion, then outputting them to the modulator column-by-column. This technique acts to spread the codewords over the channel and therefore spreads the effects of a burst error over multiple codewords. With block interleaving, as shown in Figure 5.2, a burst error of length $b \leq d$ in the binary case or $b \leq m \cdot d$ in the non-binary case, will be reduced to a single error in each codeword [Ste]. Therefore, the length of the burst error to be corrected is dependent on the depth (d) of the block interleaver. The depth of the block interleaver can be increased but not without the tradeoff of increased delays. Also as shown in Figure 5.3, any burst error of length $b = r \cdot m \cdot d$ where $r > 1$, will produce r symbol errors in each codeword. Hence, a code with error-correcting ability $t \geq r$ would be needed. Therefore, by using block interleaving techniques, a less complex of the error correction code can be applied. Reed-Solomon codes with lower error-correcting capabilities can be used in conjunction with interleaving techniques to correct large burst errors.

Input Stream: A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, B1, B2, B3, B4, B5, B6, ...,F9, F10, F11, F12



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	F2	F3	F4	F5	F6	F6	F8	F9	F10	F11	F12

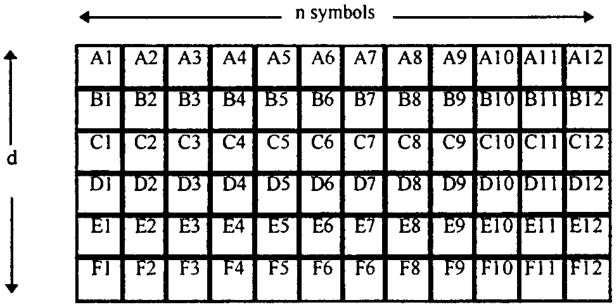
Output Stream: A1,B1, C1, D1, E1, F1, A2, B2, C2, D2, E2, F2, A3, B3, C3, D3, E3, F3, A4, ..., C12, D12, E12, F12

Figure 5.1 Block Interleaver

Conversely, using a RS code with a higher error correction capability and applying interleaving, further strengthens the code and allows for the correction of much longer burst. Note, all data in the following figures are read from left to right with the left most data element being the first element input to the interleaver/de-interleaver. The shaded rectangles represent the occurrence of symbol errors.

Interleaver

Input Stream: A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, B1, B2, B3, B4, B5, B6, ...,F9, F10, F11, F12



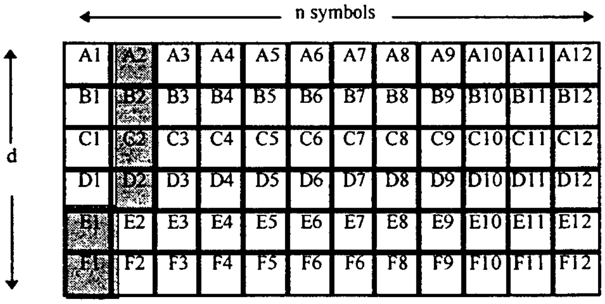
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	F2	F3	F4	F5	F6	F6	F8	F9	F10	F11	F12

Output Stream: A1, B1, C1, D1, E1, F1, A2, B2, C2, D2, E2, F2, A3, B3, C3, D3, E3, F3, A4, ...,C12, D12, E12, F12

← burst →

De-interleaver

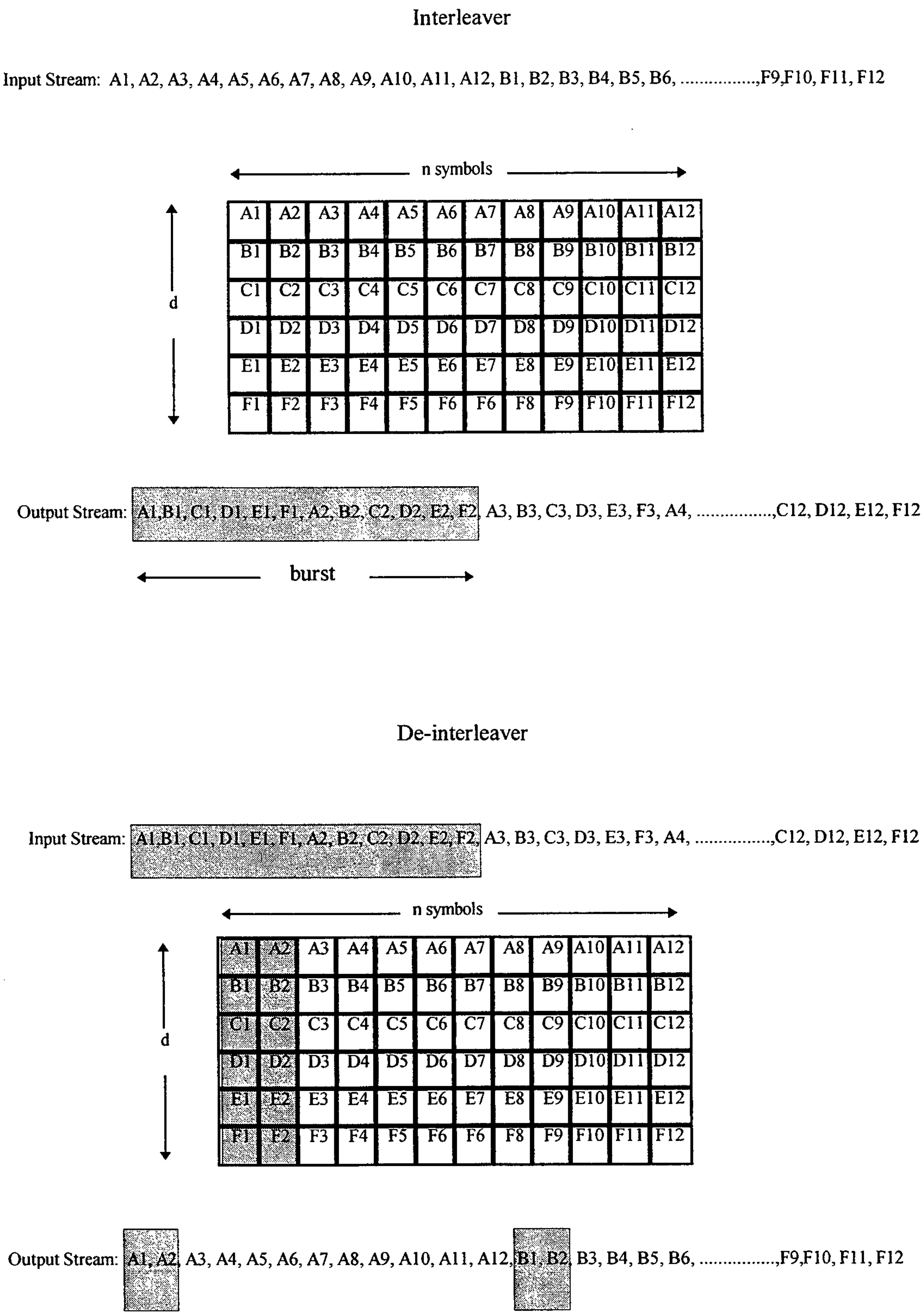
Input Stream: A1, B1, C1, D1, E1, F1, A2, B2, C2, D2, E2, F2, A3, B3, C3, D3, E3, F3, A4, ...,C12, D12, E12, F12



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	F2	F3	F4	F5	F6	F6	F8	F9	F10	F11	F12

Output Stream: A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, B1, B2, B3, B4, B5, B6, ...,F9, F10, F11, F12

Figure 5.2 Block Interleaver with (b ≤ d)



Interleaver

Input Stream: A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, B1, B2, B3, B4, B5, B6, ...,F9,F10, F11, F12

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	F2	F3	F4	F5	F6	F6	F8	F9	F10	F11	F12

Output Stream: A1,B1, C1, D1, E1, F1, A2, B2, C2, D2, E2, F2, A3, B3, C3, D3, E3, F3, A4, ...,C12, D12, E12, F12

De-interleaver

Input Stream: A1,B1, C1, D1, E1, F1, A2, B2, C2, D2, E2, F2, A3, B3, C3, D3, E3, F3, A4, ...,C12, D12, E12, F12

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	F2	F3	F4	F5	F6	F6	F8	F9	F10	F11	F12

Output Stream: A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, B1, B2, B3, B4, B5, B6, ...,F9,F10, F11, F12

Figure 5.3 Block Interleaver with (b = rd; r = 2)

5.2 Convolutional Interleaving

Convolutional interleavers are very much like block interleavers in that their function is to scramble the codewords before transmission. The scrambling of a convolutional interleaver is also a one-to-one deterministic process. It can be seen that for a block interleaver, after a column of d code symbols has been output from the interleaver, storage space is still allocated for the already transmitted column of data. The additional storage needed for the block interleaver produces unwanted delays and memory requirements in the communication system. Convolutional interleaving is a more optimal interleaving technique in that it reduces the amount of storage space required and therefore reduces delays. There are many excellent references available describing interleaving techniques, however the explanation of convolutional interleaving that follows is as proposed by [For1] and further explained by [Ste] and [Sim1].

Figure 5.4 is a representation of a convolutional interleaver and corresponding deinterleaver. The first input line of the interleaver is connected directly to the channel (modulation is assumed). All other input lines are composed of registers with each successive register containing ℓ additional storage cells. A serial data stream of codewords from the encoder circuit is input to the convolutional interleaver. Serial to parallel conversion maybe needed. The second input line is input to a shift register of length ℓ symbols. The third line is input into a shift register of length 2 ℓ symbols and the process continues for d-1 lines, for a total of d input lines. Remember that for RS codes each symbol is m bits in length and therefore, each shift register cell is considered to be m bits in length. At the same time that the input demultiplexer is inputting a codeword symbol, the output multiplexer is selecting the oldest symbol shifted out from each shift register. The output stream is then sent to the modulator.

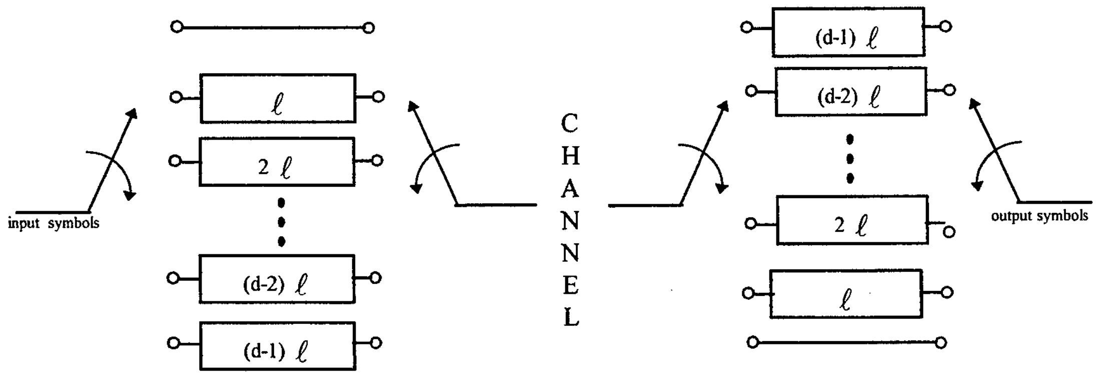


Figure 5.4 Convolutional Interleaver

For the case of using convolutional interleavers to scramble RS coded information, it is convenient to design the interleaver with the depth (d) equal to the length of a codeword (d = n). Therefore, a delay of d ℓ symbols or d ℓm bits is placed between adjacent input symbols. From this implementation, it is obvious that a burst of length (d ℓ+1)m bits will only introduce one symbol error in d ℓ consecutive codewords after de-interleaving takes place.

5.3 Pseudorandom Interleaving

Another important interleaving technique worth mentioning is called pseudorandom interleaving. Pseudorandom interleavers work on the principle of using synchronous pseudorandom number generators in both the interleaver and deinterleaver. The pseudorandom codes are used to vary the depth of interleaving in a manner only known to the communication system. The importance of this technique is in combating intentional jamming in hostile communication environments. If the depth of an interleaver is known, a pulsed jammer can be used with a pulse repetition interval (PRI) equal to the interleaving depth in order to produce burst errors at the output of the deinterleaver [Sim2]. By using pseudorandom interleaving depths, pulsed jammers lose their effectiveness. The information provided here on pseudorandom interleaving is only for completeness and will not be explored any further.

5.4 Interleaving Delay

Referring back to Figure 5.1, the block interleaving scheme uses $d \cdot n$ symbol storage cells or $d \cdot m \cdot n$ bit storage cells. This introduces a delay of $n(d-1)+1$ symbols because transmission can begin as soon as the first symbol in the last row is input. Obviously, this allows for the output of the first column before the input of the remaining symbols in the last row. If measuring the delay of the interleaver in bits, then there is a $m[n(d-1)+1]$ bit delay. The delay can be measured in time (seconds) using the known fact that the data rate (R) is equal to the inverse of the bit transmission rate ($R = 1/t_S$). Therefore, the time delay is ($t_S \cdot T_B$) seconds where $T_B = m[n(d-1)+1]$ is the interleaver bit delay measured by the number of bits. Therefore, the end-to-end delay time for block interleaving is $2(t_S \cdot T_B)$ + channel delays.

Convolutional interleavers are more efficient than block interleavers because of their smaller storage requirement and therefore smaller delay times. Referring to Figure 5.4, it can be seen that the total number of storage cells needed for a convolutional interleaver is $n \cdot \ell(d-1)/2$ symbols. For simplicity define ℓ to be equal to one and recall each symbol is m bits in length. Therefore, the storage requirement can be measured as $m[n(d-1)]/2$ bits. As it was done for the block interleaving, time delay can be measured as $t_S \cdot m \cdot [n(d-1)]/2$

seconds. Recalling that for the block interleaver the storage requirement is n(d-1)+1 symbols and n(d-1)/2 symbols for the convolutional interleaver, therefore it can be seen that the convolutional interleaver is nearly twice as efficient as the block interleaver. The interleaver efficiency is examined in more detail in [Wic1].

6 Interleaved Burst Error Correction Efficiency

Up to this point a preliminary review of RS codes and error control coding in general have been presented. This section, along with Section 7, constitutes the findings of the research performed on evaluating the burst error correction of RS codes. First in this section the burst error correction efficiency of linear block codes will be explained using the traditional definition of efficiency. The concept of phased burst error correction efficiency will also be explained. Then finally the burst error correction efficiency of RS codes will be defined.

Recall from Section 2 one that burst errors are defined according to the length of the burst b, and the guard space g separating the individual burst errors. Hence, the Gallager bound states that for a binary code of burst error-correcting capability b and guard space g, the code rate R must satisfy the inequality below [Gal2]. The Reiger bound, which is

$$\frac{g}{b} \geq \frac{(1 + R)}{(1 - R)} \rightarrow R \leq \left[\frac{\left(g \big/ (b - 1) \right)}{\left(g \big/ (b + 1) \right)} \right]$$

derived from the Gallager bound, states that a (n,k) linear block code with burst error-correcting capability b has to meet the following condition [Cap].

$$n - k \geq 2b$$

Also, a linear block code with burst error correction b and detection d must meet the condition below.

$$n - k + 1 \geq b + d$$

Based on the Reiger bound, the traditional method for defining burst error correction efficiency is z as shown below. This ratio is called the "phased burst error correction efficiency". Linear block codes for which z = 1 are considered optimal codes because

$$z = 2b/(n-k)$$

these give the most error correction achievable for the amount of parity used.

6.1 Maximum Burst Error Correction of Reed-Solomon Codes

The Gallager bound, Reiger Bound, and the phased burst error correction efficiency are defined for binary linear block codes correcting single burst errors. Reed-Solomon codes are non-binary multiple burst error-correcting linear block codes. Recall from Section 4 that RS codes have random error-correcting capability t = (n-k)/2 and maximum single burst error-correcting capability b_{1max}= tm, where the subscript (1) indicates single burst error correction. The term maximum is used because the burst errors must be located exactly over t symbols (which may not always be the case). This (b_{1max}) forms an upper bound on the single burst error-correcting capability of RS codes. Using the phased single burst error-correcting capability of RS codes b_{1max} and t = (n-k)/2, it can be seen that RS codes are always optimal phased single burst error-correcting codes.

$$z_1 = \frac{2b_1}{(n-k)} = \frac{2tm}{(n-k)m} = \frac{2\left(\frac{(n-k)}{2}\right)m}{(n-k)m} = \frac{(n-k)m}{(n-k)m} = 1$$

6.2 Multiple Burst Error Correction of Reed-Solomon Codes

In some situations frequent bursts occur in a given codeword and single burst error-correcting codes are of no use. This usually occurs when the guard spaces are not well defined and when a cluster of bursts are too long to be treated as a single large burst. As mentioned earlier RS codes are multiple burst error-correcting codes, which has made them very popular in combating these types of errors. With this in mind, the phased (maximum) double and triple burst error-correcting capability of RS codes are given below. Where the function $\lfloor x \rfloor$ is the floor function and is defined as the largest integer less than or equal to x. Because RS codes are word-oriented codes, the term phased burst

$$b_{2max} = \left\lfloor \frac{t}{2} \right\rfloor m$$

$$b_{3max} = \left\lfloor \frac{t}{3} \right\rfloor m$$

error correction applies only to the case that the burst errors occur over exactly t symbols. Therefore, the phased burst error correction of RS codes is the maximum burst error correction.

The maximum double-burst error correction capability is defined as the maximum size of two simultaneous burst errors that can be corrected. The code may be able to correct different combinations of double burst errors so long as t is not exceeded. However, no two burst greater than b_{2max} can be corrected at once. For instance if m = 8 and t = 6 ((255,243) RS code), the code will correct two burst of maximum length 24 bits. However, the same code may correct two bursts, one of length 16 bits and the other 32 bits. If one burst is of length greater than b_{2max}, then the other burst will be less than b_{2max}. The maximum burst-error correction capability also applies to triple and higher order burst error-correcting codes. The general form for phased burst error correction efficiency of a RS codes is given as z_s below. It can be easily seen that although the double, triple, and higher order phased burst error correction efficiencies of RS codes are not optimal, they have very high efficiencies.

$$z_s = \frac{2sb_s}{m(n-k)}$$

where;

$$b_{s\max} = \left\lfloor \frac{t}{s} \right\rfloor m$$

6.3 Guaranteed Burst Error Correction Of Reed-Solomon Codes

The maximum or phased burst error correction capabilities defined above are only applicable when the burst errors occur exactly over t symbols. In most practical applications the burst errors will not occur exactly over t symbols and therefore the need to find the guaranteed burst error-correcting capabilities of a code. Recall from Section 4 that the guaranteed single, double, and triple burst error correction capability of RS codes are b_1, b_2, and b_3 restated below. This is the burst error-correcting capability of a RS code no matter where the burst errors occur.

$$b_1 = m(t-1)+1$$

$$b_2 = m\left(\left\lfloor \frac{t}{2} \right\rfloor - 1\right) + 1$$

$$b_3 = m\left(\left\lfloor \frac{t}{3} \right\rfloor - 1\right) + 1$$

At this point the guaranteed burst error-correcting efficiency for a given RS code is defined. Using the traditional definition of efficiency and the guaranteed burst error-correcting capability equations, the single, double, and triple guaranteed burst error-correcting efficiency of RS codes are γ_1, γ_2, and γ_3 as follows.

$$\gamma_1 = \frac{2\left[m(t-1)+1\right]}{m(n-k)}$$

$$\gamma_2 = \frac{4\left[m\left(\left\lfloor \frac{t}{2} \right\rfloor - 1\right)+1\right]}{m(n-k)}$$

$$\gamma_3 = \frac{6\left[m\left(\left\lfloor \frac{t}{3} \right\rfloor - 1\right)+1\right]}{m(n-k)}$$

For the general case, the burst error correction efficiency of an s burst error-correcting RS code is given [Kam3].

$$\gamma_s = \frac{2sb_s}{m(n-k)}$$

where;

$$b_s = m\left(\left\lfloor \frac{t}{s} \right\rfloor - 1\right)+1$$

6.4 Analysis

Reed-Solomon codes of length n = 255, symbol length m = 8 bits, are very popular and find many applications due to their byte sized symbols. Also, codes with rate R less than fifty percent are not often used because of the large amount of redundancy. Therefore, Table 6.0 shows RS single, double, and triple guaranteed burst error correction capabilities and efficiencies for codes of length n = 255 and rate $R \geq 50\%$. For a complete table of RS codes of length n = 7, 15, 31, 63, 127, and 255 see appendix A.

As can be seen from Table 6.0 and as illustrated in Figure 6.0, the guaranteed single burst error correction efficiency approaches optimality very quickly as code rate decreases. Note that as can be observed from the equations for guaranteed burst error correction efficiency of RS codes, γ_1 , γ_2 , and γ_3 can never equal one, however they are remarkably close. The fluctuations in γ_2 and γ_3 are due to the fact that there are RS codes that have the same double and triple guaranteed burst error-correcting capabilities with increased redundancy as observed in the flat parts in the graphs of Figure 6.1. Figure 6.1 only shows RS codes up to a code rate of eighty percent for better resolution. For a complete set of graphs of burst error-correcting capabilities of RS codes see appendix A.

Using the guaranteed burst error-correcting capability and efficiency of RS codes, a code designer now has additional information at his/her disposal for selecting the best possible RS code for the error correction scheme being designed. For instance, if the designer is interested in double burst error-correcting codes, the codes on the flat areas of the guaranteed double burst error-correcting capability graph with the highest code rate are the best choice. The other codes on each flat region have decreased code rates and lower burst error-correcting efficiencies. The information presented here can help guide the code designer to get more error correction for the amount of redundancy (more bang for the buck).

Table 6.0 Burst Error Correction of Reed-Solomon Codes of Length n = 255

(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(255,253)	1	0.992157	-	-	-	-	-	-
(255,251)	2	0.984314	9	0.562500	-	-	-	-
(255,249)	3	0.976471	17	0.708333	-	-	-	-
(255,247)	4	0.968627	25	0.781250	9	0.562500	-	-
(255,245)	5	0.960784	33	0.825000	9	0.450000	-	-
(255,243)	6	0.952941	41	0.854167	17	0.708333	9	0.562500
(255,241)	7	0.945098	49	0.875000	17	0.607143	9	0.482143
(255,239)	8	0.937255	57	0.890625	25	0.781250	9	0.421875
(255,237)	9	0.929412	65	0.902778	25	0.694444	17	0.708333
(255,235)	10	0.921569	73	0.912500	33	0.825000	17	0.637500
(255,233)	11	0.913725	81	0.920455	33	0.750000	17	0.579545
(255,231)	12	0.905882	89	0.927083	41	0.854167	25	0.781250
(255,229)	13	0.898039	97	0.932692	41	0.788462	25	0.721154
(255,227)	14	0.890196	105	0.937500	49	0.875000	25	0.669643
(255,225)	15	0.882353	113	0.941667	49	0.816667	33	0.825000
(255,223)	16	0.874510	121	0.945313	57	0.890625	33	0.773438
(255,221)	17	0.866667	129	0.948529	57	0.838235	33	0.727941
(255,219)	18	0.858824	137	0.951389	65	0.902778	41	0.854167
(255,217)	19	0.850980	145	0.953947	65	0.855263	41	0.809211
(255,215)	20	0.843137	153	0.956250	73	0.912500	41	0.768750
(255,213)	21	0.835294	161	0.958333	73	0.869048	49	0.875000
(255,211)	22	0.827451	169	0.960227	81	0.920455	49	0.835227
(255,209)	23	0.819608	177	0.961957	81	0.880435	49	0.798913
(255,207)	24	0.811765	185	0.963542	89	0.927083	57	0.890625
(255,205)	25	0.803922	193	0.965000	89	0.890000	57	0.855000
(255,203)	26	0.796078	201	0.966346	97	0.932692	57	0.822115
(255,201)	27	0.788235	209	0.967593	97	0.898148	65	0.902778
(255,199)	28	0.780392	217	0.968750	105	0.937500	65	0.870536
(255,197)	29	0.772549	225	0.969828	105	0.905172	65	0.840517
(255,195)	30	0.764706	233	0.970833	113	0.941667	73	0.912500
(255,193)	31	0.756863	241	0.971774	113	0.911290	73	0.883065
(255,191)	32	0.749020	249	0.972656	121	0.945313	73	0.855469
(255,189)	33	0.741176	257	0.973485	121	0.916667	81	0.920455
(255,187)	34	0.733333	265	0.974265	129	0.948529	81	0.893382
(255,185)	35	0.725490	273	0.975000	129	0.921429	81	0.867857
(255,183)	36	0.717647	281	0.975694	137	0.951389	89	0.927083
(255,181)	37	0.709804	289	0.976351	137	0.925676	89	0.902027
(255,179)	38	0.701961	297	0.976974	145	0.953947	89	0.878289
(255,177)	39	0.694118	305	0.977564	145	0.929487	97	0.932692
(255,175)	40	0.686275	313	0.978125	153	0.956250	97	0.909375
(255,173)	41	0.678431	321	0.978659	153	0.932927	97	0.887195
(255,171)	42	0.670588	329	0.979167	161	0.958333	105	0.937500
(255,169)	43	0.662745	337	0.979651	161	0.936047	105	0.915698
(255,167)	44	0.654902	345	0.980114	169	0.960227	105	0.894886
(255,165)	45	0.647059	353	0.980556	169	0.938889	113	0.941667
(255,163)	46	0.639216	361	0.980978	177	0.961957	113	0.921196
(255,161)	47	0.631373	369	0.981383	177	0.941489	113	0.901596
(255,159)	48	0.623529	377	0.981771	185	0.963542	121	0.945313
(255,157)	49	0.615686	385	0.982143	185	0.943878	121	0.926020
(255,155)	50	0.607843	393	0.982500	193	0.965000	121	0.907500
(255,153)	51	0.600000	401	0.982843	193	0.946078	129	0.948529
(255,151)	52	0.592157	409	0.983173	201	0.966346	129	0.930288
(255,149)	53	0.584314	417	0.983491	201	0.948113	129	0.912736
(255,147)	54	0.576471	425	0.983796	209	0.967593	137	0.951389
(255,145)	55	0.568627	433	0.984091	209	0.950000	137	0.934091
(255,143)	56	0.560784	441	0.984375	217	0.968750	137	0.917411
(255,141)	57	0.552941	449	0.984649	217	0.951754	145	0.953947
(255,139)	58	0.545098	457	0.984914	225	0.969828	145	0.937500
(255,137)	59	0.537255	465	0.985169	225	0.953390	145	0.921610
(255,135)	60	0.529412	473	0.985417	233	0.970833	153	0.956250
(255,133)	61	0.521569	481	0.985656	233	0.954918	153	0.940574
(255,131)	62	0.513725	489	0.985887	241	0.971774	153	0.925403
(255,129)	63	0.505882	497	0.986111	241	0.956349	161	0.958333

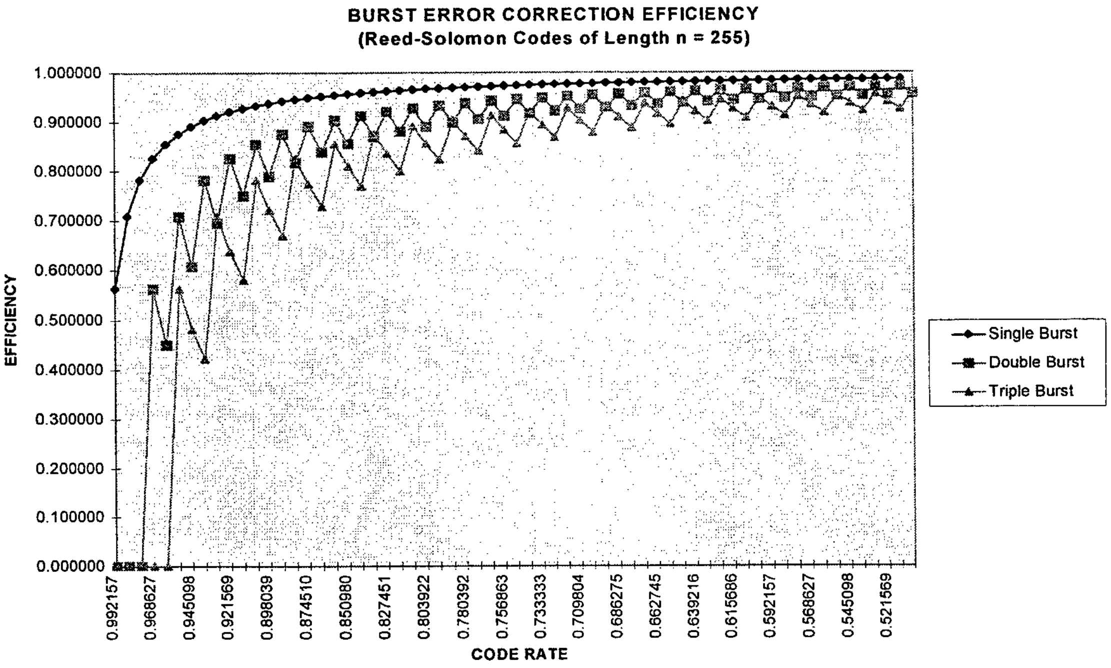


Figure 6.0 Guaranteed Burst Error Correction Efficiency

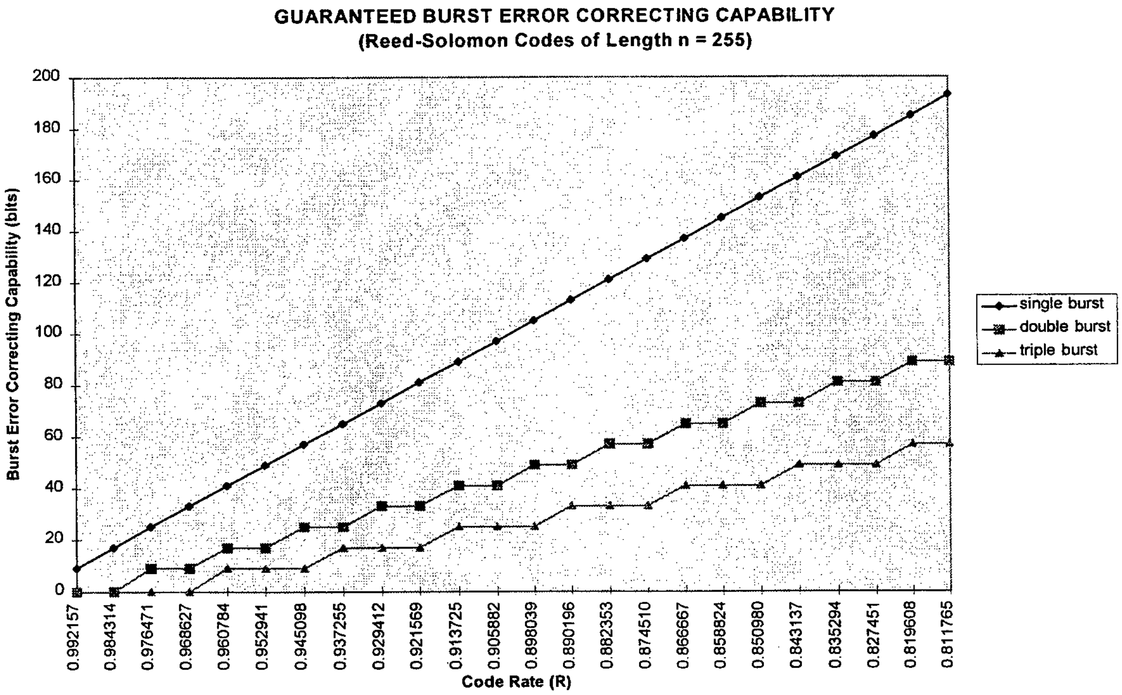


Figure 6.1 Guaranteed Burst Error Correction Capability

7 Burst Error Correction Efficiency

Up to this point the classical definition of efficiency for a linear block code has been expanded to include non-binary RS codes. This research has produced efficiency equations that will be useful in designing RS coding schemes. However, due to the fact that interleaving is commonly used in combating bursty errors, a look at the efficiency of interleaved RS codes is in order. The remainder of this thesis will investigate the efficiency of interleaved RS codes. The question of whether or not the efficiency of a RS coding scheme increases with the use of interleaving will be answered.

7.1 Block Interleaved Guaranteed Burst Error Correction

Recall that block interleaving scrambles codewords to be transmitted over the communications channel thus randomizing the effects of a burst error. Essentially, the burst error is broken into smaller and more manageable errors over multiple codewords. It is the depth (d) of the block interleaver that determines how burst errors will be randomized over the interleaved codewords. Since the depth of the block interleaver is equal to the number of codewords being interleaved, a burst of length b = rdm bits (where r = 2,3,4,... is the number of columns in the interleaver with errors, see Figures 5.2 and 5.3) will produce burst errors of b = rm bits maximum in each de-interleaved codeword. Therefore, codewords with the ability to correct burst errors of length b = rm bits and an interleaver of depth d is all that is required to correct burst of length b = rdm bits.

Reed-Solomon codes that can correct t = r random symbols errors can be used to correct burst errors up to b = rm bits. However, there are instances when burst errors do not occur exactly over t = r symbols and the random error-correcting capability t is exceeded (t < r). The equations for guaranteed burst error correction were given in Section 4 and are restated here. Guaranteed single burst error correction is $b_1 = m(t - 1) + 1$ bits, guaranteed double burst error correction is $b_2 = m(\lfloor t/2 \rfloor - 1) + 1$ bits and guaranteed triple burst error correction is $b_3 = m(\lfloor t/3 \rfloor - 1) + 1$ bits. For the general case of s burst errors the equation is $b_s = m(\lfloor t/s \rfloor - 1) + 1$. These equations again give the burst error correction of a given RS code no matter where the burst error occurs in the codeword.

Combining the operation of the block interleaver with the guaranteed burst error-correcting capability of a given RS code, the following equations are derived.

$$\psi_1 = m(td - 1) + 1$$

$$\psi_2 = m\left[\left\lfloor \frac{t}{2} \right\rfloor d - 1\right] + 1$$

$$\psi_3 = m\left[\left\lfloor \frac{t}{3} \right\rfloor d - 1\right] + 1$$

$$\psi_s = m\left[\left\lfloor \frac{t}{s} \right\rfloor d - 1\right] + 1$$

These equations express the single, double, and triple guaranteed burst error-correcting capability of a block interleaved Reed-Solomon code, where ψ_s is the general form of the equation. These equations simply mean that using a known RS code (n,k,t,m) and a block interleaver of known depth (d), a burst error of length ψ is guaranteed to be corrected no matter where it occurs over the interleaved codewords.

7.2 Block Interleaved Guaranteed Burst Error Correction Efficiency

In Section 6 it was stated that from the Reiger bound and the Gallager bound , the phased burst error correction efficiency of a linear block code is defined as z = 2b/(n-k). If z = 1, then the linear block code is said to be optimal giving the most burst error correction for the amount of parity. This definition of efficiency was for binary linear block codes which was expanded for non-binary RS codes later in Section 6 of this thesis (see γ_1 , γ_2, γ_3 , and γ_s , in Section 6).

Taking the definition of efficiency (z) another step forward, it will now be expanded to give the efficiency of block interleaved RS codes. Since the block interleaver increases the number of parity bits being transmitted to achieve the guaranteed burst error correction given in ψ, the depth (d) is applied to the denominator of z. As before m is applied to the denominator because RS codes are non-binary. Replacing the burst error correction factor b in the numerator of z with ψ, the equations for the block interleaved guaranteed burst error correction efficiency of a RS code have been derived and are listed below.

$$\Psi_1 = \frac{2\left[m(td-1)+1\right]}{m(n-k)d}$$

$$\Psi_2 = \frac{4\left[m\left(\left\lfloor \frac{t}{2} \right\rfloor d-1\right)+1\right]}{m(n-k)d}$$

$$\Psi_3 = \frac{6\left[m\left(\left\lfloor \frac{t}{3} \right\rfloor d-1\right)+1\right]}{m(n-k)d}$$

$$\Psi_s = \frac{2s\left[m\left(\left\lfloor \frac{t}{s} \right\rfloor d-1\right)+1\right]}{m(n-k)d}$$

The importance of the above efficiency equations can be summarized best by examining Figure 7.0 and Table 7.0 below. Figure 7.0 is a plot of the block interleaved guaranteed single burst-error correction efficiency (ψ_1) of length 255 for RS codes of varying interleaving depths. Length 255 is chosen here because of the popular use of the eight bit (m = 8) symbols. Also, only the high code rate (R > 75%) RS codes are shown to reduce the congestion of the graph.

It is immediately obvious from Figure 7.0 that an increase in interleaver depth produces an increase in efficiency of any given RS code. Even more importantly, examining the data from Table 7.0, it is seen that a lower code rate (R) non-interleaved code can be replaced (based on efficiency) with a higher code rate (R) interleaved RS code. This means that a simpler RS code can be interleaved and produce the same burst error correction and burst error correction efficiency as that of a more powerful non-interleaved RS code. By using a simpler RS code there is a reduction in hardware cost and the number of computations needed for encoding and decoding is reduced. As a reminder, interleaving will produce delays in communication and/or data recording systems. Therefore, replacing a more complex RS code with a simpler interleaved code can only be done if the system can tolerate the interleaving delays introduced.

Table 7.0 Block Interleaved Guaranteed Single Burst Error Correction Efficiency

		no interlv		d=2		d=3		d=4		d=5		d=10	
(n-k)	**t**	**b_1**	**eff_1**	**b_1**	**eff_1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**
(255,253)	1	1	0.125	9	0.5625	17	0.708333	25	0.78125	33	0.825	73	0.9125
(255,251)	2	9	0.5625	25	0.78125	41	0.854167	57	0.890625	73	0.9125	153	0.95625
(255,249)	3	17	0.708333	41	0.854167	65	0.902778	89	0.927083	113	0.941667	233	0.970833
(255,247)	4	25	0.78125	57	0.890625	89	0.927083	121	0.945313	153	0.95625	313	0.978125
(255,245)	5	33	0.825	73	0.9125	113	0.941667	153	0.95625	193	0.965	393	0.9825
(255,243)	6	41	0.854167	89	0.927083	137	0.951389	185	0.963542	233	0.970833	473	0.985417
(255,241)	7	49	0.875	105	0.9375	161	0.958333	217	0.96875	273	0.975	553	0.9875
(255,239)	8	57	0.890625	121	0.945313	185	0.963542	249	0.972656	313	0.978125	633	0.989063
(255,237)	9	65	0.902778	137	0.951389	209	0.967593	281	0.975694	353	0.980556	713	0.990278
(255,235)	10	73	0.9125	153	0.95625	233	0.970833	313	0.978125	393	0.9825	793	0.99125
(255,233)	11	81	0.920455	169	0.960227	257	0.973485	345	0.980114	433	0.984091	873	0.992045
(255,231)	12	89	0.927083	185	0.963542	281	0.975694	377	0.981771	473	0.985417	953	0.992708
(255,229)	13	97	0.932692	201	0.966346	305	0.977564	409	0.983173	513	0.986538	1033	0.993269
(255,227)	14	105	0.9375	217	0.96875	329	0.979167	441	0.984375	553	0.9875	1113	0.99375
(255,225)	15	113	0.941667	233	0.970833	353	0.980556	473	0.985417	593	0.988333	1193	0.994167
(255,223)	16	121	0.945313	249	0.972656	377	0.981771	505	0.986328	633	0.989063	1273	0.994531
(255,221)	17	129	0.948529	265	0.974265	401	0.982843	537	0.987132	673	0.989706	1353	0.994853
(255,219)	18	137	0.951389	281	0.975694	425	0.983796	569	0.987847	713	0.990278	1433	0.995139
(255,217)	19	145	0.953947	297	0.976974	449	0.984649	601	0.988487	753	0.990789	1513	0.995395
(255,215)	20	153	0.95625	313	0.978125	473	0.985417	633	0.989063	793	0.99125	1593	0.995625
(255,213)	21	161	0.958333	329	0.979167	497	0.986111	665	0.989583	833	0.991667	1673	0.995833
(255,211)	22	169	0.960227	345	0.980114	521	0.986742	697	0.990057	873	0.992045	1753	0.996023
(255,209)	23	177	0.961957	361	0.980978	545	0.987319	729	0.990489	913	0.992391	1833	0.996196
(255,207)	24	185	0.963542	377	0.981771	569	0.987847	761	0.990885	953	0.992708	1913	0.996354
(255,205)	25	193	0.965	393	0.9825	593	0.988333	793	0.99125	993	0.993	1993	0.9965
(255,203)	26	201	0.966346	409	0.983173	617	0.988782	825	0.991587	1033	0.993269	2073	0.996635
(255,201)	27	209	0.967593	425	0.983796	641	0.989198	857	0.991898	1073	0.993519	2153	0.996759
(255,199)	28	217	0.96875	441	0.984375	665	0.989583	889	0.992188	1113	0.99375	2233	0.996875

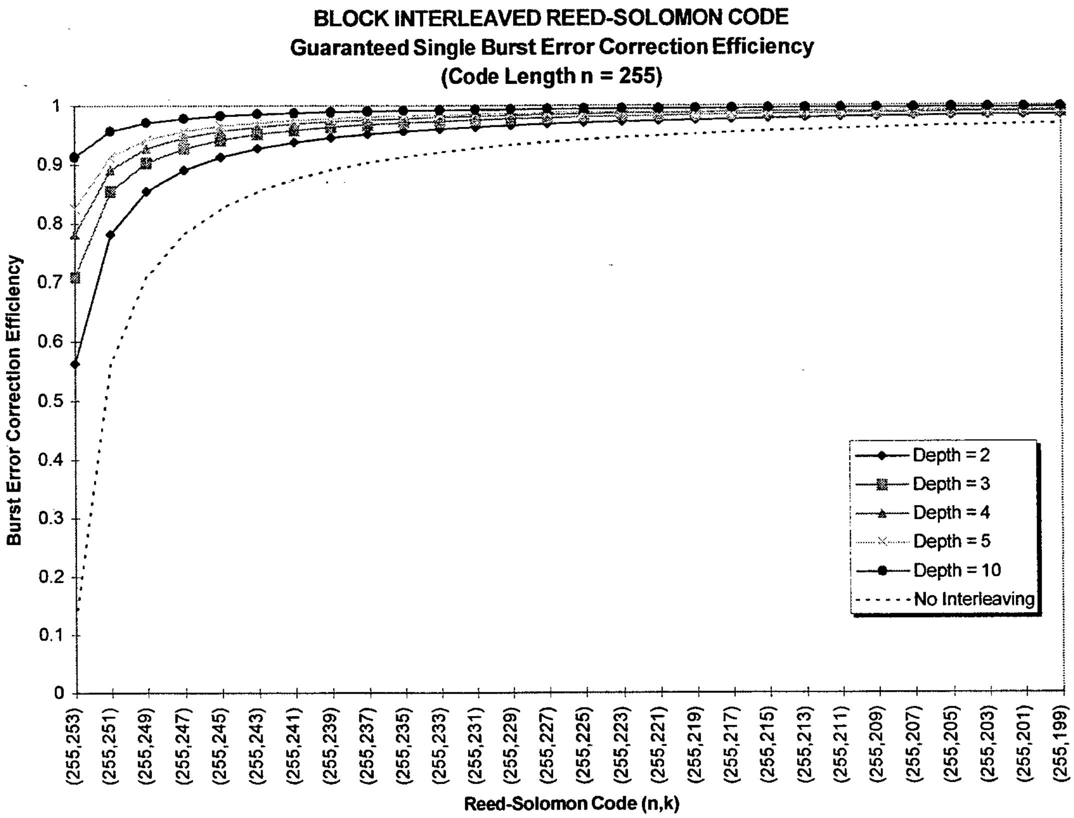


Figure 7.0 Block Interleaved Guaranteed Single Burst Error Correction Efficiency

7.3 Convolutional Interleaved Guaranteed Burst Error Correction

Convolutional interleaving is very much like block interleaving in that the idea is to scramble codewords and make them appear random to the communications channel. Convolutional interleaving was shown in Section 5 to be more efficient than block interleaving, in that it requires less storage. For the convolutional interleaver it is the depth (d) and the number of storage cells (ℓ) on each level that determine how burst errors will be randomized. From Figure 5.4 it can be seen that each symbol of a burst error will be separated by d ℓ symbols after convolutional interleaving.

If $X = (x_0, x_1, x_2, x_3, x_4, \ldots)$ is a codeword applied to the convolutional interleaver of Figure 5.4, then the data stream to be transmitted over the communications channel would be that of Figure 7.1. From Figure 7.1 it is clear that a single burst of length

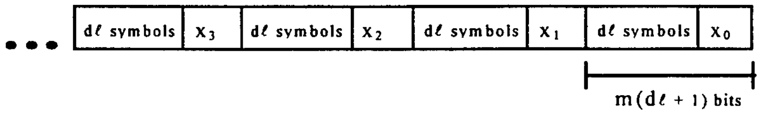


Figure 7.1 Output of a Convolutional Interleaver

$b = m(d\ell + 1)$ bits will produce a single random error in each RS codeword after deinterleaving. Combining this fact about convolutional interleaving with the concept of guaranteed burst error correction (see guaranteed burst error correction capability in Section 6) the following equations are derived. These equations express the single, double, and triple guaranteed burst error correction capability of a convolutional interleaved RS code no matter where the burst error occurs, where ϕ_s is the general form of the equation for s burst errors.

$$\phi_1 = m\left[t(d\ell + 1) - 1\right] + 1$$

$$\phi_2 = m\left[\left\lfloor \frac{t}{2} \right\rfloor (d\ell + 1) - 1\right] + 1$$

$$\phi_3 = m\left[\left\lfloor \frac{t}{3} \right\rfloor (d\ell + 1) - 1\right] + 1$$

$$\phi_s = m\left[\left\lfloor \frac{t}{s} \right\rfloor (d\ell + 1) - 1\right] + 1$$

7.4 Convolutional Interleaved Guaranteed Burst Error Correction Efficiency

As stated in Section 6 the phased burst error correction efficiency is classically defined as $z = 2b/(n-k)$. Throughout this thesis the classical definition of efficiency has been modified for various applications of RS codes. In Section 6 the efficiency (z) was expanded for the case of guaranteed burst error correction for RS codes (γ_1, γ_2, γ_3, γ_s,). In this section the definition of efficiency (z) has been expanded to include the case of block interleaved guaranteed burst error correction efficiency of RS codes. Similarly, in this section the definition of efficiency (z) is now expanded to include the case of convolutional interleaved guaranteed burst error correction efficiency. The equations provided below are derived by substituting ϕ_1, ϕ_2, ϕ_3, and ϕ_s for b in the equation $z = 2b/(n-k)$, m is applied to the denominator because RS codes are non-binary. The term $d\ell + 1$ is applied to the denominator because there must be $d\ell + 1$ codewords processed before achieving the guaranteed burst error correction of ϕ_1, ϕ_2, ϕ_3, and ϕ_s. These equations represent the guaranteed burst error correction efficiency achievable using convolutional interleaving.

$$\Phi_1 = \frac{2m\left[t(d\ell + 1) - 1\right] + 1}{m(n-k)(d\ell + 1)}$$

$$\Phi_2 = \frac{4m\left[\left\lfloor \frac{t}{2} \right\rfloor (d\ell + 1) - 1\right] + 1}{m(n-k)(d\ell + 1)}$$

$$\Phi_3 = \frac{6m\left[\left\lfloor \frac{t}{3} \right\rfloor (d\ell + 1) - 1\right] + 1}{m(n-k)(d\ell + 1)}$$

$$\Phi_s = \frac{2sm\left[\left\lfloor \frac{t}{s} \right\rfloor (d\ell + 1) - 1\right] + 1}{m(n-k)(d\ell + 1)}$$

As can be observed in Figures 7.1 through 7.4 and Tables 7.1 through 7.4 the efficiency of any given RS code increases with an increase in convolutional interleaving depth. Also, it is clear that as the additional delay per each set of registers (ℓ) is increased the single burst error-correcting efficiency is increased. Most importantly, notice that as the amount of convolutional interleaving (delay) is increased (here as ℓ increases in figures 7.1 to 7.4) the efficiency is very high for the simplest RS codes. For example the (255,253) RS code with convolutional interleaving of d = 2 and $\ell = 4$ (figure 7.4) has a guaranteed single burst error correction efficiency of more than ninety percent and can correct a single burst of sixty-five bits with a code rate (R) greater than ninety-nine percent. Very impressive!

Table 7.1 Convolutional Interleaved Guaranteed Single Burst Error Correction Eff. (ℓ=1)

l =1		no interlv		d=2		d=3		d=4		d=5		d=10	
(n-k)	**t**	**b_1**	**eff$_1$**	**b_1**	**eff$_1$**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**
(255,253)	1	1	0.125	17	0.708333	25	0.78125	33	0.825	41	0.854167	81	0.920455
(255,251)	2	9	0.5625	41	0.854167	57	0.890625	73	0.9125	89	0.927083	169	0.960227
(255,249)	3	17	0.708333	65	0.902778	89	0.927083	113	0.941667	137	0.951389	257	0.973485
(255,247)	4	25	0.78125	89	0.927083	121	0.945313	153	0.95625	185	0.963542	345	0.980114
(255,245)	5	33	0.825	113	0.941667	153	0.95625	193	0.965	233	0.970833	433	0.984091
(255,243)	6	41	0.854167	137	0.951389	185	0.963542	233	0.970833	281	0.975694	521	0.986742
(255,241)	7	49	0.875	161	0.958333	217	0.96875	273	0.975	329	0.979167	609	0.988636
(255,239)	8	57	0.890625	185	0.963542	249	0.972656	313	0.978125	377	0.981771	697	0.990057
(255,237)	9	65	0.902778	209	0.967593	281	0.975694	353	0.980556	425	0.983796	785	0.991162
(255,235)	10	73	0.9125	233	0.970833	313	0.978125	393	0.9825	473	0.985417	873	0.992045
(255,233)	11	81	0.920455	257	0.973485	345	0.980114	433	0.984091	521	0.986742	961	0.992769
(255,231)	12	89	0.927083	281	0.975694	377	0.981771	473	0.985417	569	0.987847	1049	0.993371
(255,229)	13	97	0.932692	305	0.977564	409	0.983173	513	0.986538	617	0.988782	1137	0.993881
(255,227)	14	105	0.9375	329	0.979167	441	0.984375	553	0.9875	665	0.989583	1225	0.994318
(255,225)	15	113	0.941667	353	0.980556	473	0.985417	593	0.988333	713	0.990278	1313	0.994697
(255,223)	16	121	0.945313	377	0.981771	505	0.986328	633	0.989063	761	0.990885	1401	0.995028
(255,221)	17	129	0.948529	401	0.982843	537	0.987132	673	0.989706	809	0.991422	1489	0.995321
(255,219)	18	137	0.951389	425	0.983796	569	0.987847	713	0.990278	857	0.991898	1577	0.995581
(255,217)	19	145	0.953947	449	0.984649	601	0.988487	753	0.990789	905	0.992325	1665	0.995813
(255,215)	20	153	0.95625	473	0.985417	633	0.989063	793	0.99125	953	0.992708	1753	0.996023
(255,213)	21	161	0.958333	497	0.986111	665	0.989583	833	0.991667	1001	0.993056	1841	0.996212
(255,211)	22	169	0.960227	521	0.986742	697	0.990057	873	0.992045	1049	0.993371	1929	0.996384
(255,209)	23	177	0.961957	545	0.987319	729	0.990489	913	0.992391	1097	0.993659	2017	0.996542
(255,207)	24	185	0.963542	569	0.987847	761	0.990885	953	0.992708	1145	0.993924	2105	0.996686
(255,205)	25	193	0.965	593	0.988333	793	0.99125	993	0.993	1193	0.994167	2193	0.996818
(255,203)	26	201	0.966346	617	0.988782	825	0.991587	1033	0.993269	1241	0.994391	2281	0.996941
(255,201)	27	209	0.967593	641	0.989198	857	0.991898	1073	0.993519	1289	0.994599	2369	0.997054
(255,199)	28	217	0.96875	665	0.989583	889	0.992188	1113	0.99375	1337	0.994792	2457	0.997159

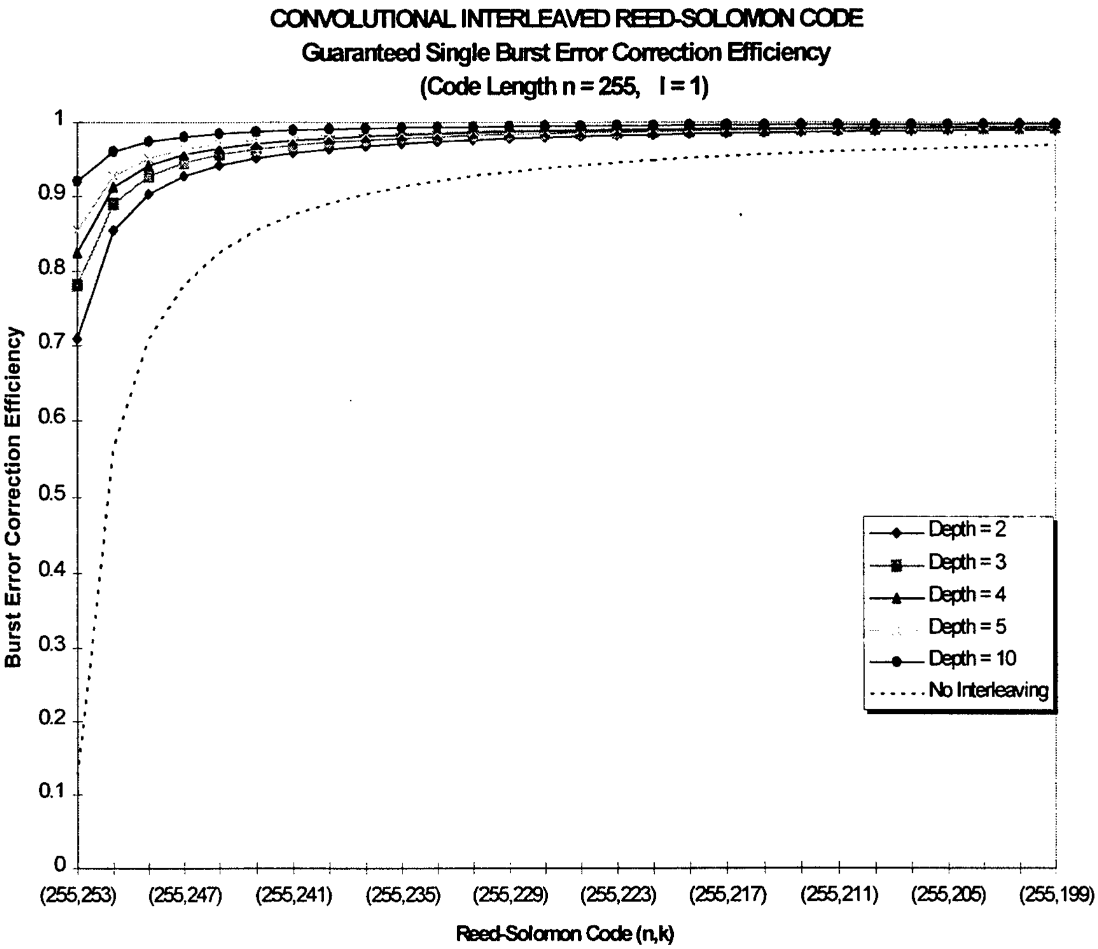


Figure 7.1 Convolutional Interleaved Guaranteed Single Burst Error Correction Efficiency (ℓ=1)

Table 7.2 Convolutional Interleaved Guaranteed Single Burst Error Correction Eff. (λ=2)

I=2		no interlv		d=2		d=3		d=4		d=5		d=10	
(n-k)	**t**	**b_1**	**eff$_1$**	**b_1**	**eff$_1$**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**
(255,253)	1	1	0.125	33	0.825	49	0.875	64.88657	0.901202	81	0.920455	161	0.958333
(255,251)	2	9	0.5625	73	0.9125	105	0.9375	136.7736	0.949817	169	0.960227	329	0.979167
(255,249)	3	17	0.708333	113	0.941667	161	0.958333	208.6612	0.966024	257	0.973485	497	0.986111
(255,247)	4	25	0.78125	153	0.95625	217	0.96875	280.5492	0.974129	345	0.980114	665	0.989583
(255,245)	5	33	0.825	193	0.965	273	0.975	352.4376	0.978993	433	0.984091	833	0.991667
(255,243)	6	41	0.854167	233	0.970833	329	0.979167	424.3264	0.982237	521	0.986742	1001	0.993056
(255,241)	7	49	0.875	273	0.975	385	0.982143	496.2157	0.984555	609	0.988636	1169	0.994048
(255,239)	8	57	0.890625	313	0.978125	441	0.984375	568.1053	0.986294	697	0.990057	1337	0.994792
(255,237)	9	65	0.902778	353	0.980556	497	0.986111	639.9952	0.987647	785	0.991162	1505	0.99537
(255,235)	10	73	0.9125	393	0.9825	553	0.9875	711.8855	0.98873	873	0.992045	1673	0.995833
(255,233)	11	81	0.920455	433	0.984091	609	0.988636	783.7761	0.989616	961	0.992769	1841	0.996212
(255,231)	12	89	0.927083	473	0.985417	665	0.989583	855.667	0.990355	1049	0.993371	2009	0.996528
(255,229)	13	97	0.932692	513	0.986538	721	0.990385	927.5582	0.990981	1137	0.993881	2177	0.996795
(255,227)	14	105	0.9375	553	0.9875	777	0.991071	999.4497	0.991518	1225	0.994318	2345	0.997024
(255,225)	15	113	0.941667	593	0.988333	833	0.991667	1071.341	0.991983	1313	0.994697	2513	0.997222
(255,223)	16	121	0.945313	633	0.989063	889	0.992188	1143.234	0.99239	1401	0.995028	2681	0.997396
(255,221)	17	129	0.948529	673	0.989706	945	0.992647	1215.126	0.99275	1489	0.995321	2849	0.997549
(255,219)	18	137	0.951389	713	0.990278	1001	0.993056	1272.808	0.982105	1577	0.995581	3017	0.997685
(255,217)	19	145	0.953947	753	0.990789	1057	0.993421	1343.911	0.982391	1665	0.995813	3185	0.997807
(255,215)	20	153	0.95625	793	0.99125	1113	0.99375	1415.015	0.982649	1753	0.996023	3353	0.997917
(255,213)	21	161	0.958333	833	0.991667	1169	0.994048	1486.118	0.982883	1841	0.996212	3521	0.998016
(255,211)	22	169	0.960227	873	0.992045	1225	0.994318	1557.222	0.983095	1929	0.996384	3689	0.998106
(255,209)	23	177	0.961957	913	0.992391	1281	0.994565	1628.326	0.983289	2017	0.996542	3857	0.998188
(255,207)	24	185	0.963542	953	0.992708	1337	0.994792	1699.43	0.983467	2105	0.996686	4025	0.998264
(255,205)	25	193	0.965	993	0.993	1393	0.995	1770.535	0.98363	2193	0.996818	4193	0.998333
(255,203)	26	201	0.966346	1033	0.993269	1449	0.995192	1841.639	0.983782	2281	0.996941	4361	0.998397
(255,201)	27	209	0.967593	1073	0.993519	1505	0.99537	1912.744	0.983922	2369	0.997054	4529	0.998457
(255,199)	28	217	0.96875	1113	0.99375	1561	0.995536	1983.849	0.984052	2457	0.997159	4697	0.998512

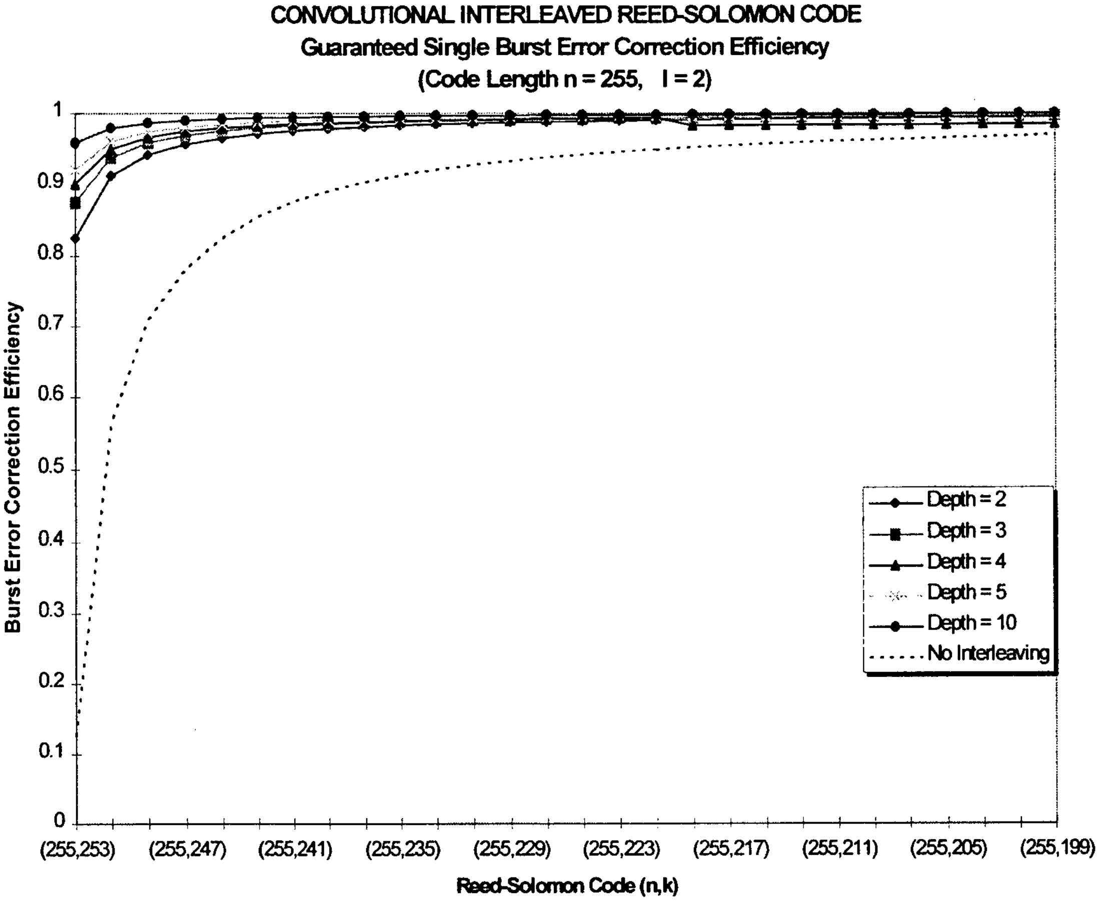


Figure 7.2 Convolutional Interleaved Guaranteed Single Burst Error Correction Efficiency (ℓ=2)

Table 7.3 Convolutional Interleaved Guaranteed Single Burst Error Correction Eff. (ℓ=3)

l =3		no interlv		d=2		d=3		d=4		d=5		d=10	
(n-k)	**t**	**b_1**	**eff_1**	**b_1**	**eff_1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**
(255,253)	1	1	0.125	49	0.875	73	0.9125	96.94267	0.932141	121	0.945313	241	0.971774
(255,251)	2	9	0.5625	105	0.9375	153	0.95625	200.8857	0.965797	249	0.972656	489	0.985887
(255,249)	3	17	0.708333	161	0.958333	233	0.970833	304.829	0.977016	377	0.981771	737	0.990591
(255,247)	4	25	0.78125	217	0.96875	313	0.978125	408.7727	0.982627	505	0.986328	985	0.992944
(255,245)	5	33	0.825	273	0.975	393	0.9825	512.7166	0.985994	633	0.989063	1233	0.994355
(255,243)	6	41	0.854167	329	0.979167	473	0.985417	616.6609	0.988239	761	0.990885	1481	0.995296
(255,241)	7	49	0.875	385	0.982143	553	0.9875	720.6054	0.989843	889	0.992188	1729	0.995968
(255,239)	8	57	0.890625	441	0.984375	633	0.989063	824.5501	0.991046	1017	0.993164	1977	0.996472
(255,237)	9	65	0.902778	497	0.986111	713	0.990278	928.4951	0.991982	1145	0.993924	2225	0.996864
(255,235)	10	73	0.9125	553	0.9875	793	0.99125	1032.44	0.992731	1273	0.994531	2473	0.997177
(255,233)	11	81	0.920455	609	0.988636	873	0.992045	1136.386	0.993344	1401	0.995028	2721	0.997434
(255,231)	12	89	0.927083	665	0.989583	953	0.992708	1240.331	0.993855	1529	0.995443	2969	0.997648
(255,229)	13	97	0.932692	721	0.990385	1033	0.993269	1344.277	0.994288	1657	0.995793	3217	0.997829
(255,227)	14	105	0.9375	777	0.991071	1113	0.99375	1448.223	0.994659	1785	0.996094	3465	0.997984
(255,225)	15	113	0.941667	833	0.991667	1193	0.994167	1552.17	0.994981	1913	0.996354	3713	0.998118
(255,223)	16	121	0.945313	889	0.992188	1273	0.994531	1656.116	0.995262	2041	0.996582	3961	0.998236
(255,221)	17	129	0.948529	945	0.992647	1353	0.994853	1760.063	0.995511	2169	0.996783	4209	0.99834
(255,219)	18	137	0.951389	1001	0.993056	1433	0.995139	1853.79	0.990273	2297	0.996962	4457	0.998432
(255,217)	19	145	0.953947	1057	0.993421	1513	0.995395	1957.169	0.99047	2425	0.997122	4705	0.998514
(255,215)	20	153	0.95625	1113	0.99375	1593	0.995625	2060.549	0.990648	2553	0.997266	4953	0.998589
(255,213)	21	161	0.958333	1169	0.994048	1673	0.995833	2163.928	0.99081	2681	0.997396	5201	0.998656
(255,211)	22	169	0.960227	1225	0.994318	1753	0.996023	2267.308	0.990956	2809	0.997514	5449	0.998717
(255,209)	23	177	0.961957	1281	0.994565	1833	0.996196	2370.687	0.99109	2937	0.997622	5697	0.998773
(255,207)	24	185	0.963542	1337	0.994792	1913	0.996354	2474.067	0.991213	3065	0.997721	5945	0.998824
(255,205)	25	193	0.965	1393	0.995	1993	0.9965	2577.447	0.991326	3193	0.997813	6193	0.998871
(255,203)	26	201	0.966346	1449	0.995192	2073	0.996635	2680.827	0.99143	3321	0.997897	6441	0.998914
(255,201)	27	209	0.967593	1505	0.99537	2153	0.996759	2784.207	0.991527	3449	0.997975	6689	0.998955
(255,199)	28	217	0.96875	1561	0.995536	2233	0.996875	2887.588	0.991617	3577	0.998047	6937	0.998992

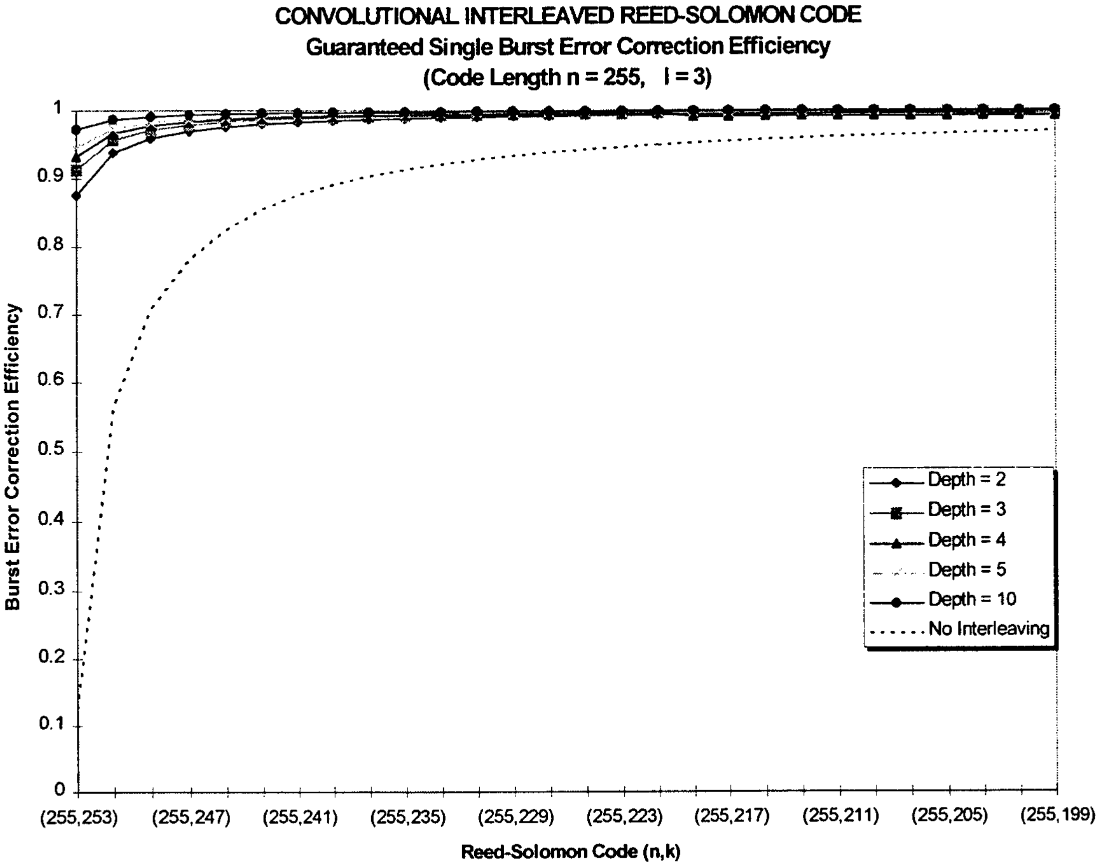


Figure 7.3 Convolutional Interleaved Guaranteed Single Burst Error Correction Efficiency (ℓ=3)

Table 7.4 Convolutional Interleaved Guaranteed Single Burst Error Correction Eff. (ℓ=4)

l =4		no interlv		d=2		d=3		d=4		d=5		d=10	
(n-k)	**t**	**b_1**	**eff_1**	**b_1**	**eff_1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**	**b1**	**eff1**
(255,253)	1	1	0.125	65	0.902778	97	0.932692	128.9647	0.94827	161	0.958333	321	0.978659
(255,251)	2	9	0.5625	137	0.951389	201	0.966346	264.9298	0.974006	329	0.979167	649	0.989329
(255,249)	3	17	0.708333	209	0.967593	305	0.977564	400.895	0.982586	497	0.986111	977	0.992886
(255,247)	4	25	0.78125	281	0.975694	409	0.983173	536.8605	0.986876	665	0.989583	1305	0.994665
(255,245)	5	33	0.825	353	0.980556	513	0.986538	672.8262	0.98945	833	0.991667	1633	0.995732
(255,243)	6	41	0.854167	425	0.983796	617	0.988782	808.7922	0.991167	1001	0.993056	1961	0.996443
(255,241)	7	49	0.875	497	0.986111	721	0.990385	944.7583	0.992393	1169	0.994048	2289	0.996951
(255,239)	8	57	0.890625	569	0.987847	825	0.991587	1080.725	0.993313	1337	0.994792	2617	0.997332
(255,237)	9	65	0.902778	641	0.989198	929	0.992521	1216.691	0.994029	1505	0.99537	2945	0.997629
(255,235)	10	73	0.9125	713	0.990278	1033	0.993269	1352.658	0.994601	1673	0.995833	3273	0.997866
(255,233)	11	81	0.920455	785	0.991162	1137	0.993881	1488.625	0.99507	1841	0.996212	3601	0.99806
(255,231)	12	89	0.927083	857	0.991898	1241	0.994391	1624.592	0.995461	2009	0.996528	3929	0.998222
(255,229)	13	97	0.932692	929	0.992521	1345	0.994822	1760.559	0.995791	2177	0.996795	4257	0.998358
(255,227)	14	105	0.9375	1001	0.993056	1449	0.995192	1896.526	0.996075	2345	0.997024	4585	0.998476
(255,225)	15	113	0.941667	1073	0.993519	1553	0.995513	2032.494	0.99632	2513	0.997222	4913	0.998577
(255,223)	16	121	0.945313	1145	0.993924	1657	0.995793	2168.461	0.996535	2681	0.997396	5241	0.998666
(255,221)	17	129	0.948529	1217	0.994281	1761	0.996041	2304.429	0.996725	2849	0.997549	5569	0.998745
(255,219)	18	137	0.951389	1289	0.994599	1865	0.996261	2432.428	0.993639	3017	0.997685	5897	0.998814
(255,217)	19	145	0.953947	1361	0.994883	1969	0.996457	2567.953	0.99379	3185	0.997807	6225	0.998877
(255,215)	20	153	0.95625	1433	0.995139	2073	0.996635	2703.478	0.993926	3353	0.997917	6553	0.998933
(255,213)	21	161	0.958333	1505	0.99537	2177	0.996795	2839.004	0.994049	3521	0.998016	6881	0.998984
(255,211)	22	169	0.960227	1577	0.995581	2281	0.996941	2974.529	0.994161	3689	0.998106	7209	0.99903
(255,209)	23	177	0.961957	1649	0.995773	2385	0.997074	3110.055	0.994263	3857	0.998188	7537	0.999072
(255,207)	24	185	0.963542	1721	0.995949	2489	0.997196	3245.581	0.994357	4025	0.998264	7865	0.999111
(255,205)	25	193	0.965	1793	0.996111	2593	0.997308	3381.107	0.994443	4193	0.998333	8193	0.999146
(255,203)	26	201	0.966346	1865	0.996261	2697	0.997411	3516.633	0.994523	4361	0.998397	8521	0.999179
(255,201)	27	209	0.967593	1937	0.996399	2801	0.997507	3652.159	0.994597	4529	0.998457	8849	0.99921
(255,199)	28	217	0.96875	2009	0.996528	2905	0.997596	3787.685	0.994665	4697	0.998512	9177	0.999238

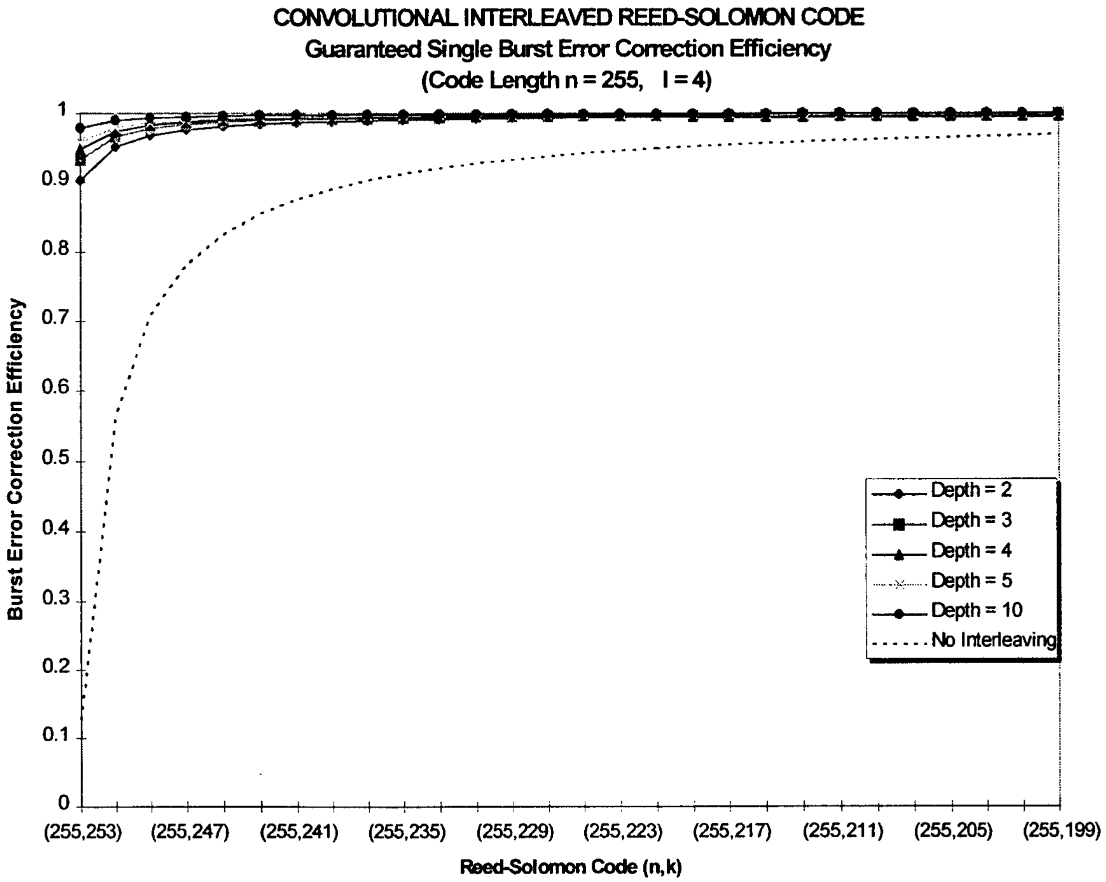


Figure 7.4 Convolutional Interleaved Guaranteed Single Burst Error Correction Efficiency (ℓ=4)

8 Conclusion

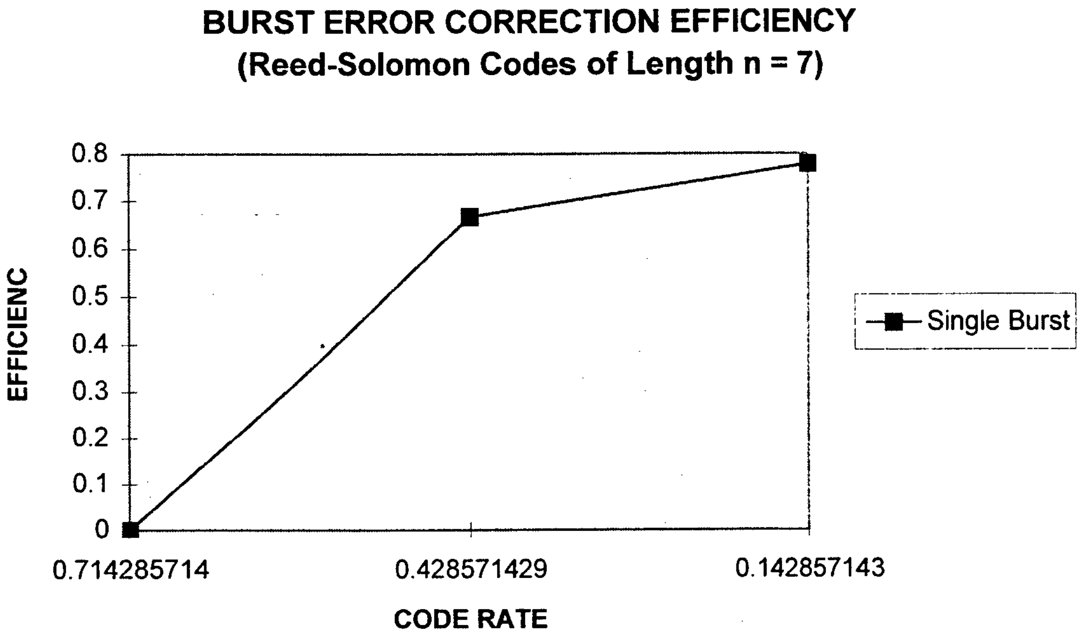
Burst errors are commonplace in today's digital communication and recording systems. Reed-Solomon codes are known to be the best at combating burst errors because they are non-binary Maximum Distance Separable (MDS) codes. It was the intent of this research effort to explore the burst error-correcting capabilities and efficiencies of Reed-Solomon codes.

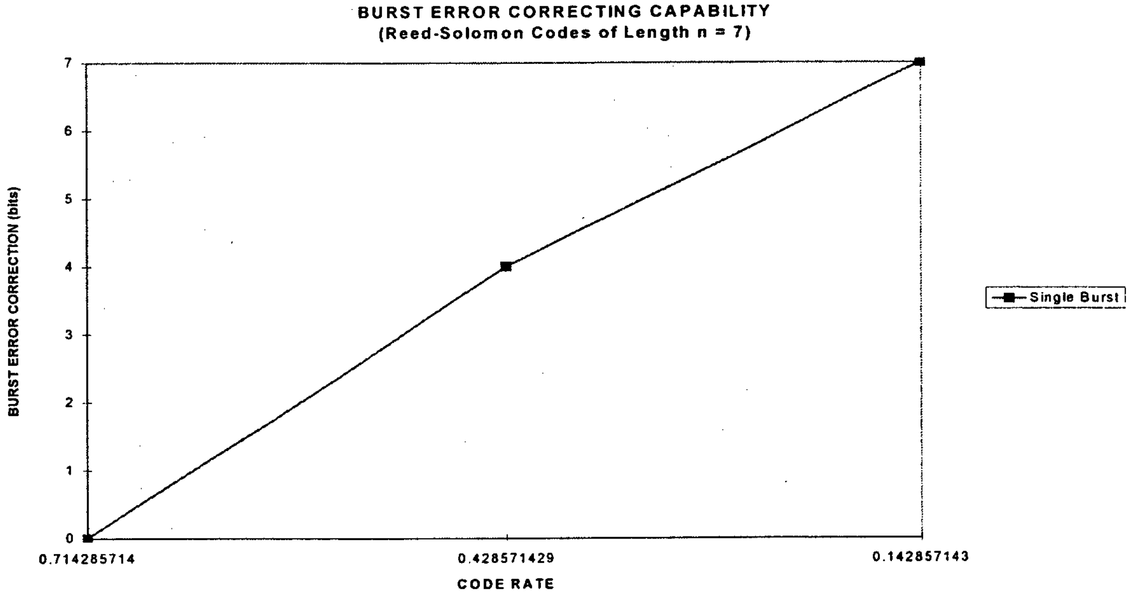
This thesis discussed the phased burst error-correcting capabilities of RS codes. The concept of guaranteed burst error correction was then examined. The idea of guaranteed burst error correction was then expanded to include the popular block and convolutional interleaving techniques illustrating the increase in guaranteed burst error correction achievable using interleaving. The classical definition of error correction efficiency was then expanded to include the guaranteed burst error correction of RS codes and interleaved RS codes.

It was shown that RS codes are very efficient at correcting burst errors. In fact many RS codes approach optimal efficiencies while correcting considerably large burst errors. All of the equations, tables, and graphs generated in this thesis can be used by code designers as additional criteria in choosing the most efficient burst error-correcting system for their needs.

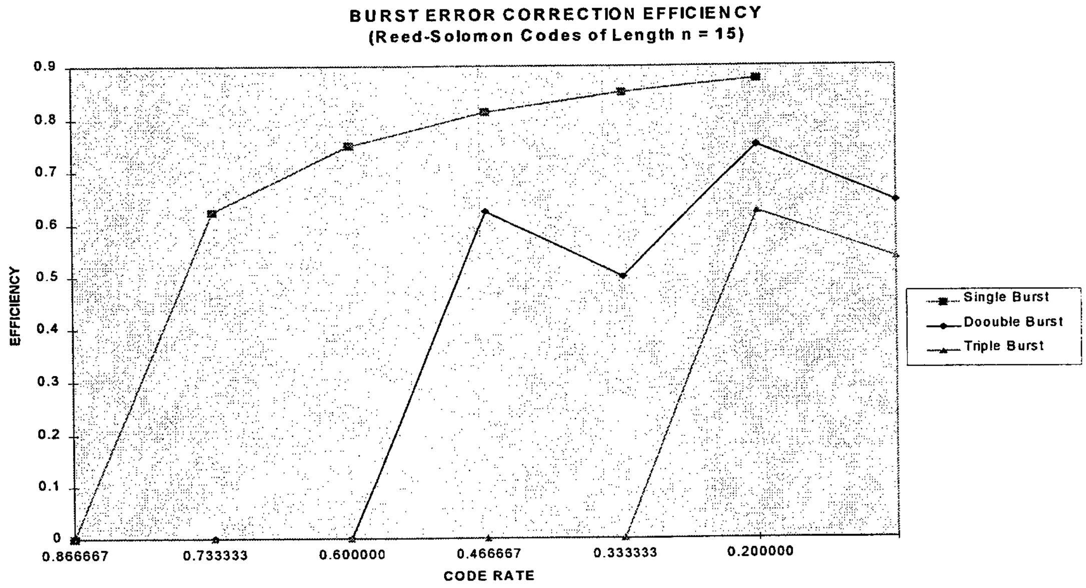
9 Appendix A

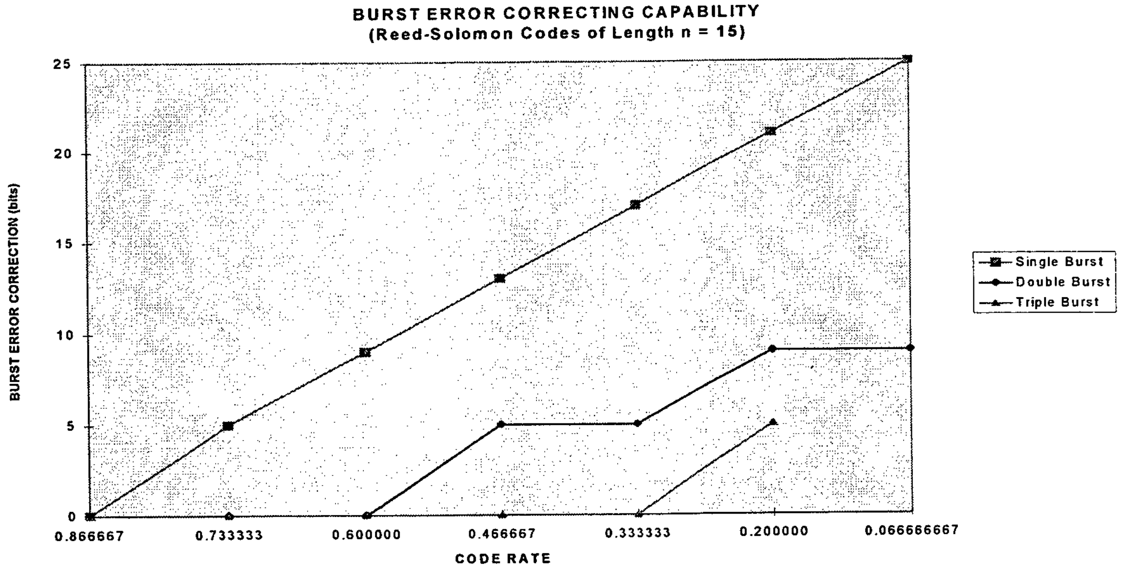
(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(7,5)	1	0.714286	-	-	-	-	-	-
(7,3)	2	0.428571	4	0.666667	-	-	-	-
(7,1)	3	0.142857	7	0.777778	-	-	-	-





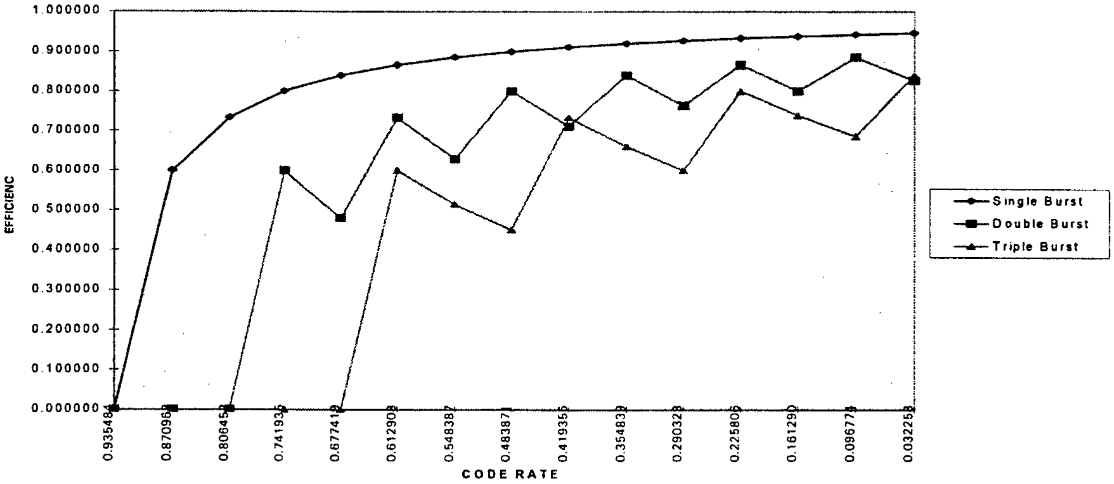
(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(15,13)	1	0.866667	-	-	-	-	-	-
(15,11)	2	0.733333	5	0.625000	-	-	-	-
(15,9)	3	0.600000	9	0.750000	-	-	-	-
(15,7)	4	0.466667	13	0.812500	5	0.625000	-	-
(15,5)	5	0.333333	17	0.850000	5	0.500000	-	-
(15,3)	6	0.200000	21	0.875000	9	0.750000	5	0.625000
(15,1)	7	0.066667	25	0.892857	9	0.642857	5	0.535714



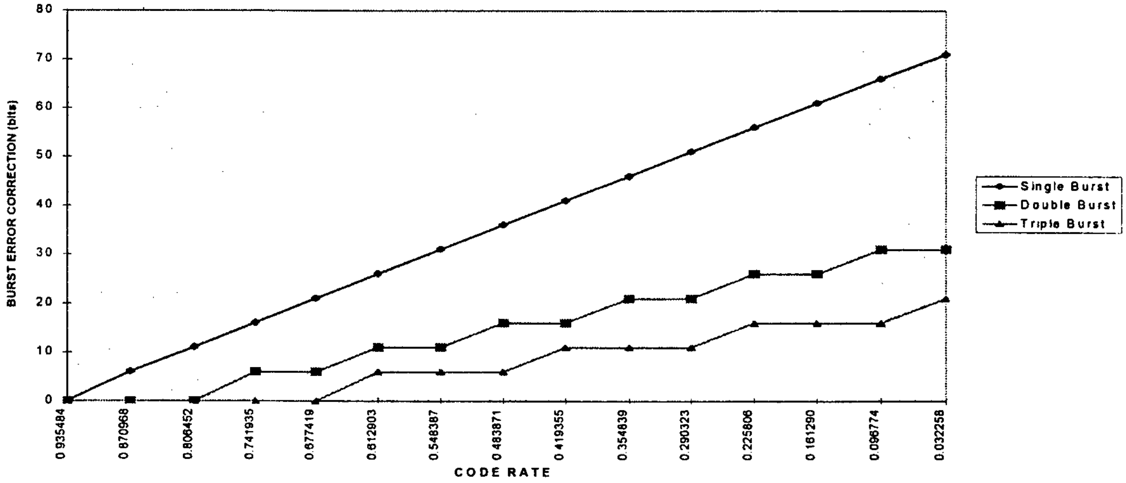


(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(31,29)	1	0.935484	-	-	-	-	-	-
(31,27)	2	0.870968	6	0.600000	-	-	-	-
(31,25)	3	0.806452	11	0.733333	-	-	-	-
(31,23)	4	0.741935	16	0.800000	6	0.600000	-	-
(31,21)	5	0.677419	21	0.840000	6	0.480000	-	-
(31,19)	6	0.612903	26	0.866667	11	0.733333	6	0.600000
(31,17)	7	0.548387	31	0.885714	11	0.628571	6	0.514286
(31,15)	8	0.483871	36	0.900000	16	0.800000	6	0.450000
(31,13)	9	0.419355	41	0.911111	16	0.711111	11	0.733333
(31,11)	10	0.354839	46	0.920000	21	0.840000	11	0.660000
(31,9)	11	0.290323	51	0.927273	21	0.763636	11	0.600000
(31,7)	12	0.225806	56	0.933333	26	0.866667	16	0.800000
(31,5)	13	0.161290	61	0.938462	26	0.800000	16	0.738462
(31,3)	14	0.096774	66	0.942857	31	0.885714	16	0.685714
(31,1)	15	0.032258	71	0.946667	31	0.826667	21	0.840000

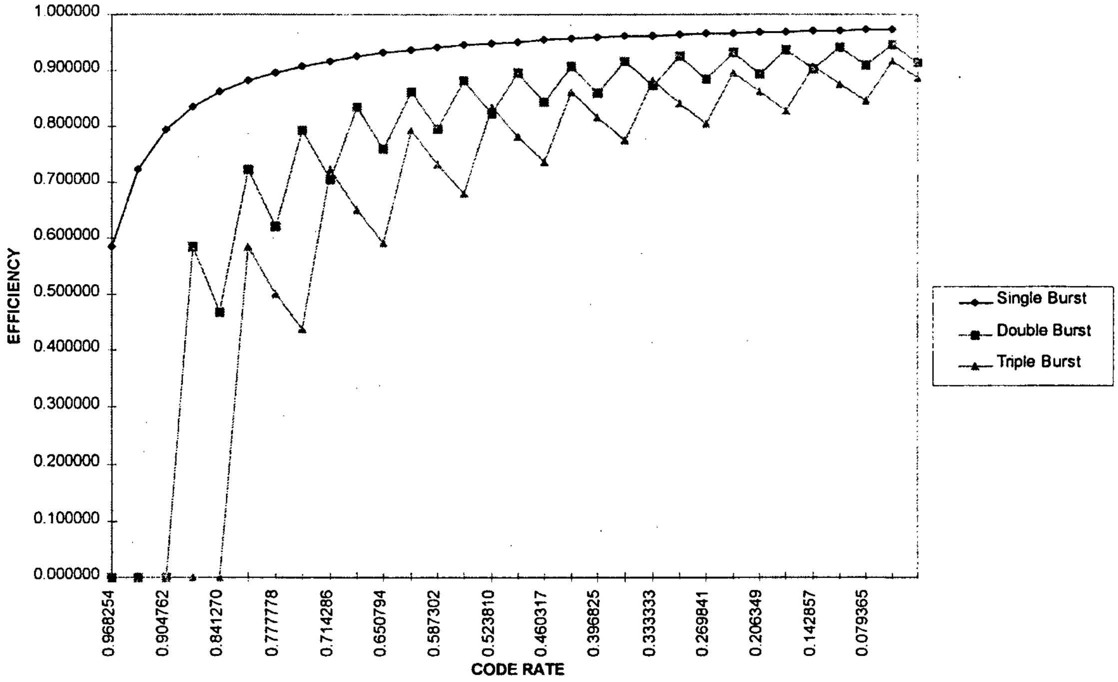
BURST ERROR CORRECTION EFFICIENCY
(Reed-Solomon Codes of Length n = 31)



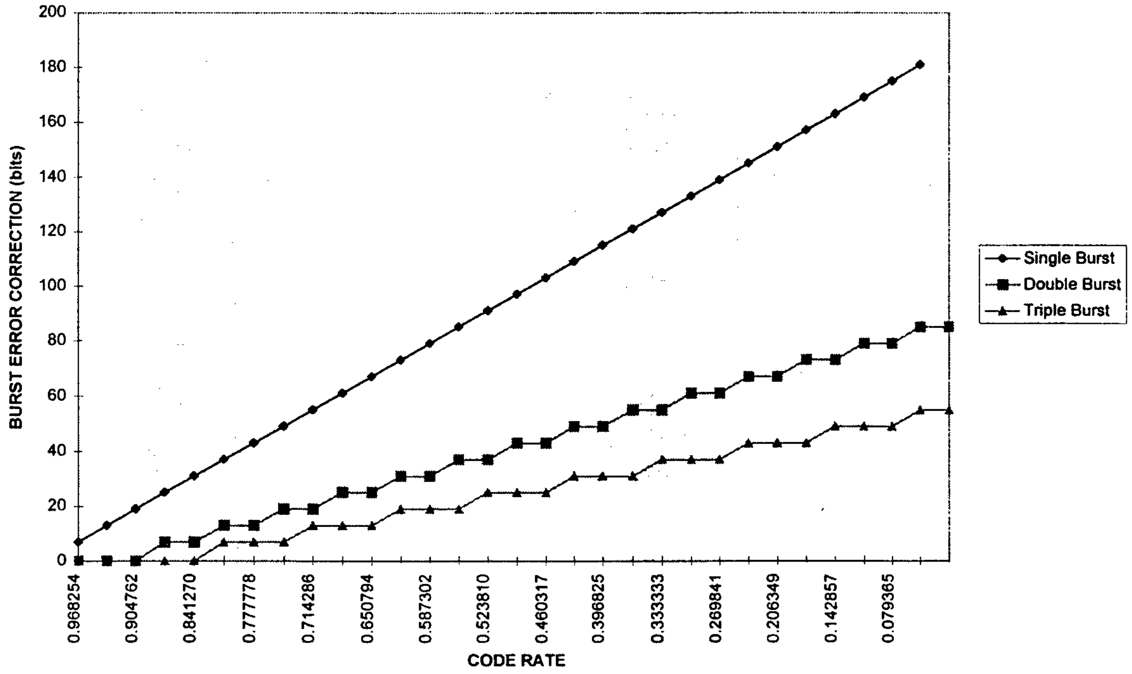
BURST ERROR CORRECTING CAPABILITY
(Reed-Solomon Codes of Length n = 31)



(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(63,61)	1	0.968254	-	-	-	-	-	-
(63,59)	2	0.936508	7	0.583333	-	-	-	-
(63,57)	3	0.904762	13	0.722222	-	-	-	-
(63,55)	4	0.873016	19	0.791667	7	0.583333	-	-
(63,53)	5	0.841270	25	0.833333	7	0.466667	-	-
(63,51)	6	0.809524	31	0.861111	13	0.722222	7	0.583333
(63,49)	7	0.777778	37	0.880952	13	0.619048	7	0.500000
(63,47)	8	0.746032	43	0.895833	19	0.791667	7	0.437500
(63,45)	9	0.714286	49	0.907407	19	0.703704	13	0.722222
(63,43)	10	0.682540	55	0.916667	25	0.833333	13	0.650000
(63,41)	11	0.650794	61	0.924242	25	0.757576	13	0.590909
(63,39)	12	0.619048	67	0.930556	31	0.861111	19	0.791667
(63,37)	13	0.587302	73	0.935897	31	0.794872	19	0.730769
(63,35)	14	0.555556	79	0.940476	37	0.880952	19	0.678571
(63,33)	15	0.523810	85	0.944444	37	0.822222	25	0.833333
(63,31)	16	0.492063	91	0.947917	43	0.895833	25	0.781250
(63,29)	17	0.460317	97	0.950980	43	0.843137	25	0.735294
(63,27)	18	0.428571	103	0.953704	49	0.907407	31	0.861111
(63,25)	19	0.396825	109	0.956140	49	0.859649	31	0.815789
(63,23)	20	0.365079	115	0.958333	55	0.916667	31	0.775000
(63,21)	21	0.333333	121	0.960317	55	0.873016	37	0.880952
(63,19)	22	0.301587	127	0.962121	61	0.924242	37	0.840909
(63,17)	23	0.269841	133	0.963768	61	0.884058	37	0.804348
(63,15)	24	0.238095	139	0.965278	67	0.930556	43	0.895833
(63,13)	25	0.206349	145	0.966667	67	0.893333	43	0.860000
(63,11)	26	0.174603	151	0.967949	73	0.935897	43	0.826923
(63,9)	27	0.142857	157	0.969136	73	0.901235	49	0.907407
(63,7)	28	0.111111	163	0.970238	79	0.940476	49	0.875000
(63,5)	29	0.079365	169	0.971264	79	0.908046	49	0.844828
(63,3)	30	0.047619	175	0.972222	85	0.944444	55	0.916667
(63,1)	31	0.015873	181	0.973118	85	0.913978	55	0.887097

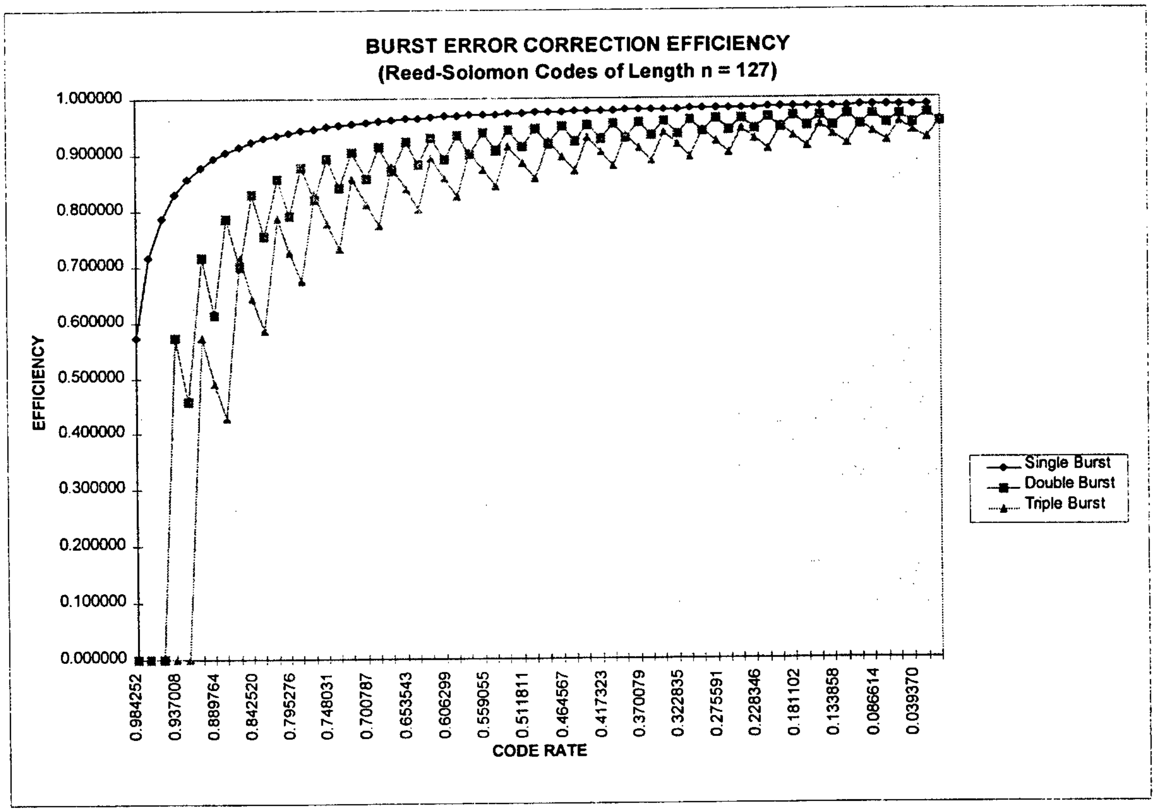


BURST ERROR CORRECTION EFFICIENCY
(Reed-Solomon Codes of Length n = 63)

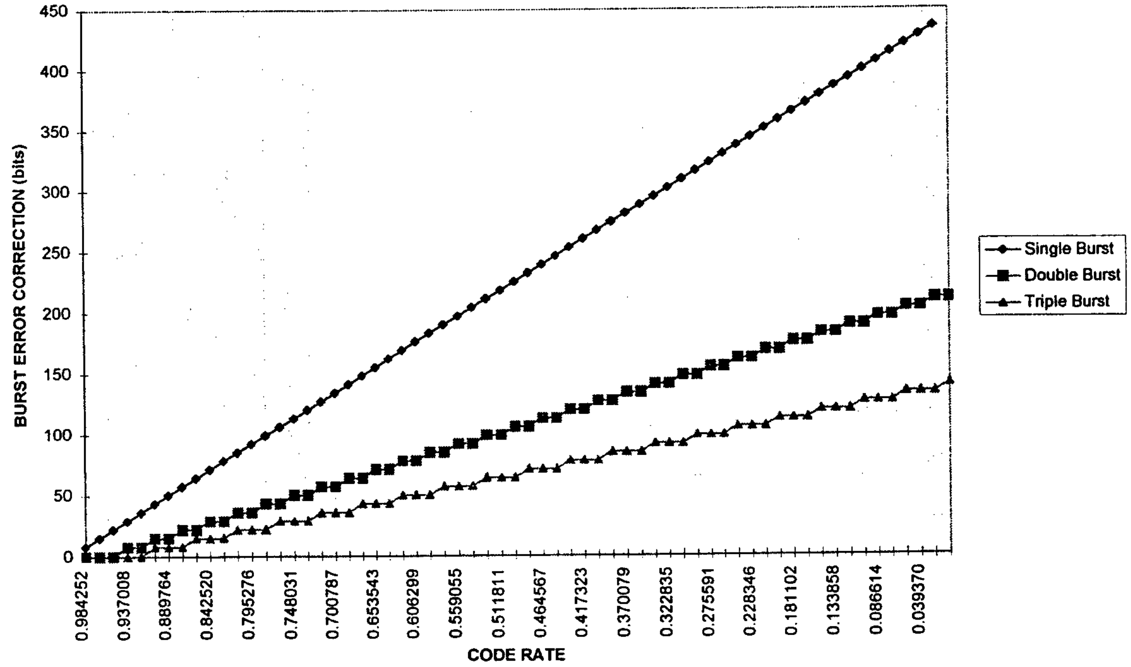


BURST ERROR CORRECTING CAPABILITY
(Reed-Solomon Codes of Length n = 63)

(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(127,125)	1	0.984252	-	-	-	-	-	-
(127,123)	2	0.968504	8	0.571429	-	-	-	-
(127,121)	3	0.952756	15	0.714286	-	-	-	-
(127,119)	4	0.937008	22	0.785714	8	0.571429	-	-
(127,117)	5	0.921260	29	0.828571	8	0.457143	-	-
(127,115)	6	0.905512	36	0.857143	15	0.714286	8	0.571429
(127,113)	7	0.889764	43	0.877551	15	0.612245	8	0.489796
(127,111)	8	0.874016	50	0.892857	22	0.785714	8	0.428571
(127,109)	9	0.858268	57	0.904762	22	0.698413	15	0.714286
(127,107)	10	0.842520	64	0.914286	29	0.828571	15	0.642857
(127,105)	11	0.826772	71	0.922078	29	0.753247	15	0.584416
(127,103)	12	0.811024	78	0.928571	36	0.857143	22	0.785714
(127,101)	13	0.795276	85	0.934066	36	0.791209	22	0.725275
(127,99)	14	0.779528	92	0.938776	43	0.877551	22	0.673469
(127,97)	15	0.763780	99	0.942857	43	0.819048	29	0.828571
(127,95)	16	0.748031	106	0.946429	50	0.892857	29	0.776786
(127,93)	17	0.732283	113	0.949580	50	0.840336	29	0.731092
(127,91)	18	0.716535	120	0.952381	57	0.904762	36	0.857143
(127,89)	19	0.700787	127	0.954887	57	0.857143	36	0.812030
(127,87)	20	0.685039	134	0.957143	64	0.914286	36	0.771429
(127,85)	21	0.669291	141	0.959184	64	0.870748	43	0.877551
(127,83)	22	0.653543	148	0.961039	71	0.922078	43	0.837662
(127,81)	23	0.637795	155	0.962733	71	0.881988	43	0.801242
(127,79)	24	0.622047	162	0.964286	78	0.928571	50	0.892857
(127,77)	25	0.606299	169	0.965714	78	0.891429	50	0.857143
(127,75)	26	0.590551	176	0.967033	85	0.934066	50	0.824176
(127,73)	27	0.574803	183	0.968254	85	0.899471	57	0.904762
(127,71)	28	0.559055	190	0.969388	92	0.938776	57	0.872449
(127,69)	29	0.543307	197	0.970443	92	0.906404	57	0.842365
(127,67)	30	0.527559	204	0.971429	99	0.942857	64	0.914286
(127,65)	31	0.511811	211	0.972350	99	0.912442	64	0.884793
(127,63)	32	0.496063	218	0.973214	106	0.946429	64	0.857143
(127,61)	33	0.480315	225	0.974026	106	0.917749	71	0.922078
(127,59)	34	0.464567	232	0.974790	113	0.949580	71	0.894958
(127,57)	35	0.448819	239	0.975510	113	0.922449	71	0.869388
(127,55)	36	0.433071	246	0.976190	120	0.952381	78	0.928571
(127,53)	37	0.417323	253	0.976834	120	0.926641	78	0.903475
(127,51)	38	0.401575	260	0.977444	127	0.954887	78	0.879699
(127,49)	39	0.385827	267	0.978022	127	0.930403	85	0.934066
(127,47)	40	0.370079	274	0.978571	134	0.957143	85	0.910714
(127,45)	41	0.354331	281	0.979094	134	0.933798	85	0.888502
(127,43)	42	0.338583	288	0.979592	141	0.959184	92	0.938776
(127,41)	43	0.322835	295	0.980066	141	0.936877	92	0.916944
(127,39)	44	0.307087	302	0.980519	148	0.961039	92	0.896104
(127,37)	45	0.291339	309	0.980952	148	0.939683	99	0.942857
(127,35)	46	0.275591	316	0.981366	155	0.962733	99	0.922360
(127,33)	47	0.259843	323	0.981763	155	0.942249	99	0.902736
(127,31)	48	0.244094	330	0.982143	162	0.964286	106	0.946429
(127,29)	49	0.228346	337	0.982507	162	0.944606	106	0.927114
(127,27)	50	0.212598	344	0.982857	169	0.965714	106	0.908571
(127,25)	51	0.196850	351	0.983193	169	0.946779	113	0.949580
(127,23)	52	0.181102	358	0.983516	176	0.967033	113	0.931319
(127,21)	53	0.165354	365	0.983827	176	0.948787	113	0.913747
(127,19)	54	0.149606	372	0.984127	183	0.968254	120	0.952381
(127,17)	55	0.133858	379	0.984416	183	0.950649	120	0.935065
(127,15)	56	0.118110	386	0.984694	190	0.969388	120	0.918367
(127,13)	57	0.102362	393	0.984962	190	0.952381	127	0.954887
(127,11)	58	0.086614	400	0.985222	197	0.970443	127	0.938424
(127,9)	59	0.070866	407	0.985472	197	0.953995	127	0.922518
(127,7)	60	0.055118	414	0.985714	204	0.971429	134	0.957143
(127,5)	61	0.039370	421	0.985948	204	0.955504	134	0.941452
(127,3)	62	0.023622	428	0.986175	211	0.972350	134	0.926267
(127,1)	63	0.007874	435	0.986395	211	0.956916	141	0.959184

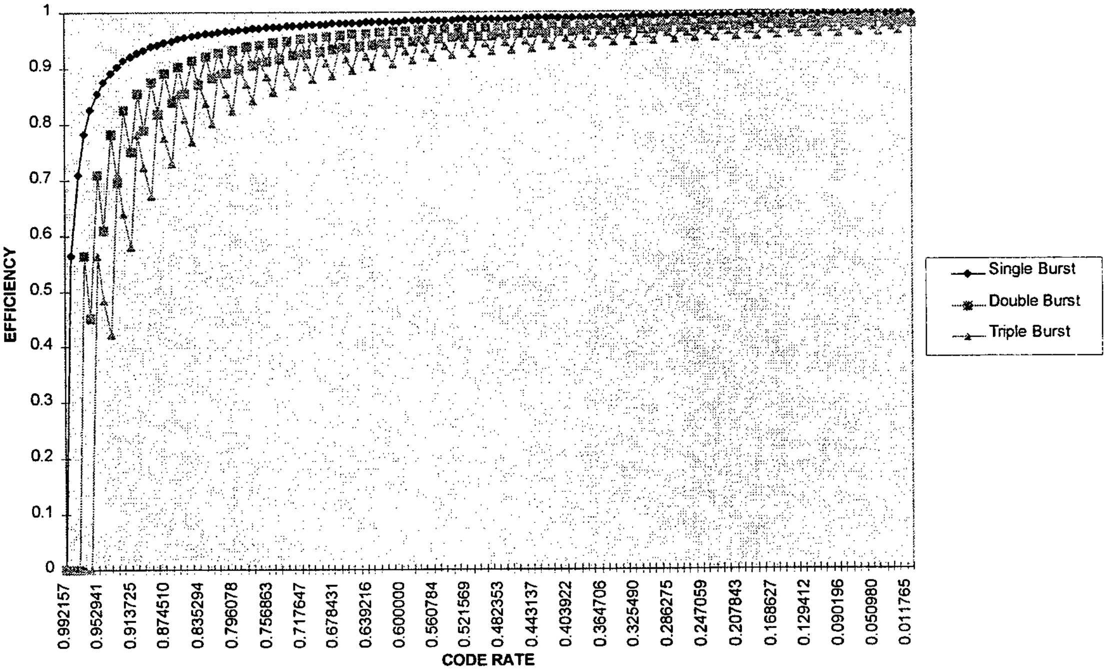

BURST ERROR CORRECTION EFFICIENCY
(Reed-Solomon Codes of Length n = 127)

BURST ERROR CORRECTING CAPABILITY
(Reed-Solomon Codes of Length n = 127)

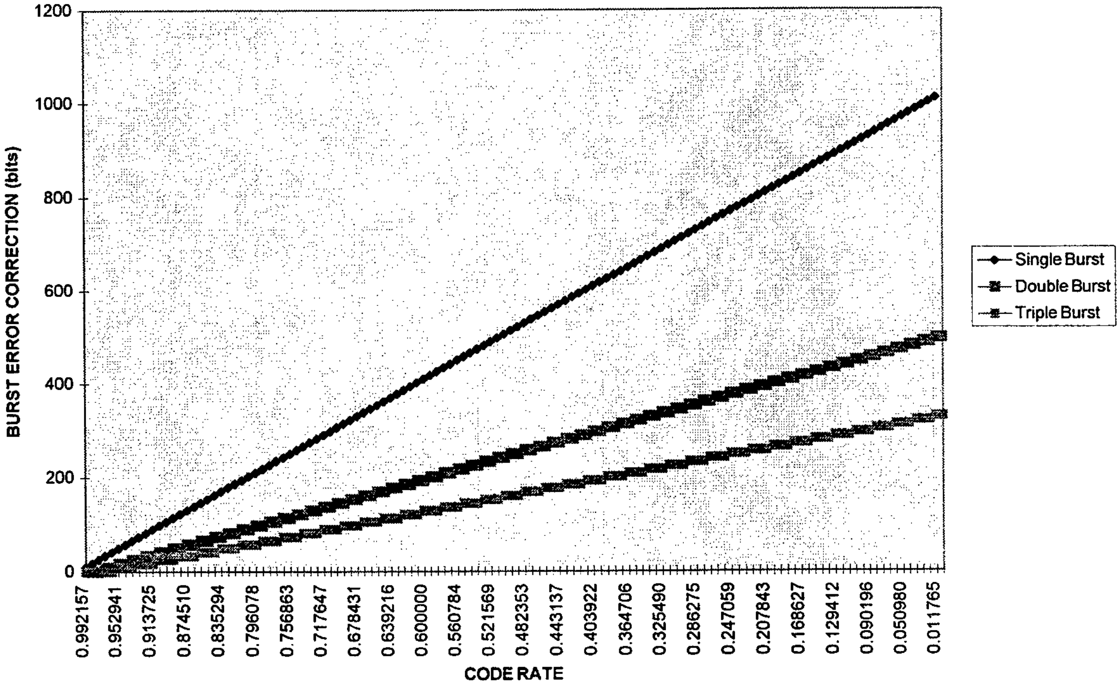


(n,k)	t	R_c	b_1	γ_1	b_2	γ_2	b_3	γ_3
(255,253)	1	0.992157	-	-	-	-	-	-
(255,251)	2	0.984314	9	0.562500	-	-	-	-
(255,249)	3	0.976471	17	0.708333	-	-	-	-
(255,247)	4	0.968627	25	0.781250	9	0.562500	-	-
(255,245)	5	0.960784	33	0.825000	9	0.450000	-	-
(255,243)	6	0.952941	41	0.854167	17	0.708333	9	0.562500
(255,241)	7	0.945098	49	0.875000	17	0.607143	9	0.482143
(255,239)	8	0.937255	57	0.890625	25	0.781250	9	0.421875
(255,237)	9	0.929412	65	0.902778	25	0.694444	17	0.708333
(255,235)	10	0.921569	73	0.912500	33	0.825000	17	0.637500
(255,233)	11	0.913725	81	0.920455	33	0.750000	17	0.579545
(255,231)	12	0.905882	89	0.927083	41	0.854167	25	0.781250
(255,229)	13	0.898039	97	0.932692	41	0.788462	25	0.721154
(255,227)	14	0.890196	105	0.937500	49	0.875000	25	0.669643
(255,225)	15	0.882353	113	0.941667	49	0.816667	33	0.825000
(255,223)	16	0.874510	121	0.945313	57	0.890625	33	0.773438
(255,221)	17	0.866667	129	0.948529	57	0.838235	33	0.727941
(255,219)	18	0.858824	137	0.951389	65	0.902778	41	0.854167
(255,217)	19	0.850980	145	0.953947	65	0.855263	41	0.809211
(255,215)	20	0.843137	153	0.956250	73	0.912500	41	0.768750
(255,213)	21	0.835294	161	0.958333	73	0.869048	49	0.875000
(255,211)	22	0.827451	169	0.960227	81	0.920455	49	0.835227
(255,209)	23	0.819608	177	0.961957	81	0.880435	49	0.798913
(255,207)	24	0.811765	185	0.963542	89	0.927083	57	0.890625
(255,205)	25	0.803922	193	0.965000	89	0.890000	57	0.855000
(255,203)	26	0.796078	201	0.966346	97	0.932692	57	0.822115
(255,201)	27	0.788235	209	0.967593	97	0.898148	65	0.902778
(255,199)	28	0.780392	217	0.968750	105	0.937500	65	0.870536
(255,197)	29	0.772549	225	0.969828	105	0.905172	65	0.840517
(255,195)	30	0.764706	233	0.970833	113	0.941667	73	0.912500
(255,193)	31	0.756863	241	0.971774	113	0.911290	73	0.883065
(255,191)	32	0.749020	249	0.972656	121	0.945313	73	0.855469
(255,189)	33	0.741176	257	0.973485	121	0.916667	81	0.920455
(255,187)	34	0.733333	265	0.974265	129	0.948529	81	0.893382
(255,185)	35	0.725490	273	0.975000	129	0.921429	81	0.867857
(255,183)	36	0.717647	281	0.975694	137	0.951389	89	0.927083
(255,181)	37	0.709804	289	0.976351	137	0.925676	89	0.902027
(255,179)	38	0.701961	297	0.976974	145	0.953947	89	0.878289
(255,177)	39	0.694118	305	0.977564	145	0.929487	97	0.932692
(255,175)	40	0.686275	313	0.978125	153	0.956250	97	0.909375
(255,173)	41	0.678431	321	0.978659	153	0.932927	97	0.887195
(255,171)	42	0.670588	329	0.979167	161	0.958333	105	0.937500
(255,169)	43	0.662745	337	0.979651	161	0.936047	105	0.915698
(255,167)	44	0.654902	345	0.980114	169	0.960227	105	0.894886
(255,165)	45	0.647059	353	0.980556	169	0.938889	113	0.941667
(255,163)	46	0.639216	361	0.980978	177	0.961957	113	0.921196
(255,161)	47	0.631373	369	0.981383	177	0.941489	113	0.901596
(255,159)	48	0.623529	377	0.981771	185	0.963542	121	0.945313
(255,157)	49	0.615686	385	0.982143	185	0.943878	121	0.926020
(255,155)	50	0.607843	393	0.982500	193	0.965000	121	0.907500
(255,153)	51	0.600000	401	0.982843	193	0.946078	129	0.948529
(255,151)	52	0.592157	409	0.983173	201	0.966346	129	0.930288
(255,149)	53	0.584314	417	0.983491	201	0.948113	129	0.912736
(255,147)	54	0.576471	425	0.983796	209	0.967593	137	0.951389
(255,145)	55	0.568627	433	0.984091	209	0.950000	137	0.934091
(255,143)	56	0.560784	441	0.984375	217	0.968750	137	0.917411
(255,141)	57	0.552941	449	0.984649	217	0.951754	145	0.953947
(255,139)	58	0.545098	457	0.984914	225	0.969828	145	0.937500
(255,137)	59	0.537255	465	0.985169	225	0.953390	145	0.921610
(255,135)	60	0.529412	473	0.985417	233	0.970833	153	0.956250
(255,133)	61	0.521569	481	0.985656	233	0.954918	153	0.940574
(255,131)	62	0.513725	489	0.985887	241	0.971774	153	0.925403
(255,129)	63	0.505882	497	0.986111	241	0.956349	161	0.958333
(255,127)	64	0.498039	505	0.986328	249	0.972656	161	0.943359
(255,125)	65	0.490196	513	0.986538	249	0.957692	161	0.928846
(255,123)	66	0.482353	521	0.986742	257	0.973485	169	0.960227
(255,121)	67	0.474510	529	0.986940	257	0.958955	169	0.945896

(255,117)	69	0.458824	545	0.987319	265	0.960145	177	0.961957
(255,115)	70	0.450980	553	0.987500	273	0.975000	177	0.948214
(255,113)	71	0.443137	561	0.987676	273	0.961268	177	0.934859
(255,111)	72	0.435294	569	0.987847	281	0.975694	185	0.963542
(255,109)	73	0.427451	577	0.988014	281	0.962329	185	0.950342
(255,107)	74	0.419608	585	0.988176	289	0.976351	185	0.937500
(255,105)	75	0.411765	593	0.988333	289	0.963333	193	0.965000
(255,103)	76	0.403922	601	0.988487	297	0.976974	193	0.952303
(255,101)	77	0.396078	609	0.988636	297	0.964286	193	0.939935
(255,99)	78	0.388235	617	0.988782	305	0.977564	201	0.966346
(255,97)	79	0.380392	625	0.988924	305	0.965190	201	0.954114
(255,95)	80	0.372549	633	0.989063	313	0.978125	201	0.942188
(255,93)	81	0.364706	641	0.989198	313	0.966049	209	0.967593
(255,91)	82	0.356863	649	0.989329	321	0.978659	209	0.955793
(255,89)	83	0.349020	657	0.989458	321	0.966867	209	0.944277
(255,87)	84	0.341176	665	0.989583	329	0.979167	217	0.968750
(255,85)	85	0.333333	673	0.989706	329	0.967647	217	0.957353
(255,83)	86	0.325490	681	0.989826	337	0.979651	217	0.946221
(255,81)	87	0.317647	689	0.989943	337	0.968391	225	0.969828
(255,79)	88	0.309804	697	0.990057	345	0.980114	225	0.958807
(255,77)	89	0.301961	705	0.990169	345	0.969101	225	0.948034
(255,75)	90	0.294118	713	0.990278	353	0.980556	233	0.970833
(255,73)	91	0.286275	721	0.990385	353	0.969780	233	0.960165
(255,71)	92	0.278431	729	0.990489	361	0.980978	233	0.949728
(255,69)	93	0.270588	737	0.990591	361	0.970430	241	0.971774
(255,67)	94	0.262745	745	0.990691	369	0.981383	241	0.961436
(255,65)	95	0.254902	753	0.990789	369	0.971053	241	0.951316
(255,63)	96	0.247059	761	0.990885	377	0.981771	249	0.972656
(255,61)	97	0.239216	769	0.990979	377	0.971649	249	0.962629
(255,59)	98	0.231373	777	0.991071	385	0.982143	249	0.952806
(255,57)	99	0.223529	785	0.991162	385	0.972222	257	0.973485
(255,55)	100	0.215686	793	0.991250	393	0.982500	257	0.963750
(255,53)	101	0.207843	801	0.991337	393	0.972772	257	0.954208
(255,51)	102	0.200000	809	0.991422	401	0.982843	265	0.974265
(255,49)	103	0.192157	817	0.991505	401	0.973301	265	0.964806
(255,47)	104	0.184314	825	0.991587	409	0.983173	265	0.955529
(255,45)	105	0.176471	833	0.991667	409	0.973810	273	0.975000
(255,43)	106	0.168627	841	0.991745	417	0.983491	273	0.965802
(255,41)	107	0.160784	849	0.991822	417	0.974299	273	0.956776
(255,39)	108	0.152941	857	0.991898	425	0.983796	281	0.975694
(255,37)	109	0.145098	865	0.991972	425	0.974771	281	0.966743
(255,35)	110	0.137255	873	0.992045	433	0.984091	281	0.957955
(255,33)	111	0.129412	881	0.992117	433	0.975225	289	0.976351
(255,31)	112	0.121569	889	0.992188	441	0.984375	289	0.967634
(255,29)	113	0.113725	897	0.992257	441	0.975664	289	0.959071
(255,27)	114	0.105882	905	0.992325	449	0.984649	297	0.976974
(255,25)	115	0.098039	913	0.992391	449	0.976087	297	0.968478
(255,23)	116	0.090196	921	0.992457	457	0.984914	297	0.960129
(255,21)	117	0.082353	929	0.992521	457	0.976496	305	0.977564
(255,19)	118	0.074510	937	0.992585	465	0.985169	305	0.969280
(255,17)	119	0.066667	945	0.992647	465	0.976891	305	0.961134
(255,15)	120	0.058824	953	0.992708	473	0.985417	313	0.978125
(255,13)	121	0.050980	961	0.992769	473	0.977273	313	0.970041
(255,11)	122	0.043137	969	0.992828	481	0.985656	313	0.962090
(255,9)	123	0.035294	977	0.992886	481	0.977642	321	0.978659
(255,7)	124	0.027451	985	0.992944	489	0.985887	321	0.970766
(255,5)	125	0.019608	993	0.993000	489	0.978000	321	0.963000
(255,3)	126	0.011765	1001	0.993056	497	0.986111	329	0.979167
(255,1)	127	0.003922	1009	0.993110	497	0.978346	329	0.971457



BURST ERROR CORRECTION EFFICIENCY
(Reed-Solomon Codes of Length n = 255)



BURST ERROR CORRECTING CAPABILITY
(Reed-Solomon Codes of Length n = 255)

10 References

[Bae] Luc Baert, Luc Theunissen, Guido Vergult, Digital Audio and Compact Disc Technology, Great Britain, Richard Clay Ltd, 1988.

[Ben] G. Benelli, C. Bianciardi, V. Cappellini, “Redundancy Bounds for Multiple-Burst Correcting-Codes,” Alta Frequenza, Volume XLVII, Number 9, pp 679-683, 1978.

[Ber1] E. R. Berlekamp, Algebriac Coding Theory, New York: McGraw-Hill, 1968.

[Ber2] E. R. Berlekamp, “Nonbinary BCH Decoding,” paper presented at the 1967 International Symposium on Information theory, San Remo, Italy.

[Ber3] E. R. Berlekamp, “A Hypersystolic Reed-Solomon Decoder,” in Reed-Solomon Codes and Their Applications, Wicker and Bhargava (ed.), New York:IEEE Press, 1994.

[Cap] V. Cappellini, Data compression and Error Control Techniques with Applications, Academic Press, 1985.

[Car] M.G. Carasso, J. B. H. Peek, and J. P. Sinjou, “The Compact Disc Digital Audio System,” Philips Technical Review, Volume 40, Number 6, pp. 151-156, 1982.

[Cla] G. C. Clark, Jr. and J. B. Cain, Error-Correction Coding for Digital Communications, New York: Plenum Press, 1981.

[Chi] R. T. Chien, “Cyclic Decoding Procedure for the Bose-Chaudhuri-Hocquenghem Codes,” IEEE transactions on Information Theory, Volume IT-10, pp. 357-363, October 1964.

[For1] G. D. Forney, Jr., “Burst Correcting Codes for the Classic Bursty Channel,” IEEE Transactions on Communication, Volume COM-19, pp. 772-781, October 1971.

[For2] G. D. Forney, Jr., “On Decoding BCH Codes,” IEEE Transactions on Information Theory, Volume IT-11, pp. 549-557, October 1965.

[Gal1] R. G. Gallager, Information Theory and Reliable Communication, New York: John Wiley & Sons, 1968

[Gal2] R. G. Gallager, “Binary Codes for Burst Error Correction,” IEEE Transactions on Information Theory, Volume IT-12, 1966.

[Gor] D. Gorenstein and N. Zierler, "A Class of Error-Correcting Codes in pm Symbols," Journal of the Society of Industrial and Applied Mathematics, Volume9, pp. 207-214, June 1961.

[Has] M. A. Hasan, V. K. Bhargava and T. Le-Ngoc, "Algorithms and Architectures for the Design of a VLSI Reed-Solomon Codec", in Reed-Solomon Codes and Their Applications, Wicker and Bhargava (ed.), New York:IEEE Press, 1994.

[Her] D. Herskovitz, "A Sampling of Recording Devices", Electronic Defense, Volume 19, No. 2, pp. 44-49, February 1996.

[Imm1] K. A. S. Immink, Coding Techniques for Digital Recorders, Englewood Cliffs, NJ.: Prentice-Hall International, 1991.

[Imm2] K. A. S. Immink, "Reed-Solomon Codes and the Compact Disc", in Reed-Solomon Codes and Their Applications, Wicker and Bhargava (ed.), New York: IEEE Press, 1994.

[Kam1] B. Kamali, "Some Applications of Wireless Communications in Intelligent Vehicle Highway Systems ", Proceedings of IEEE Vehicular Technology Conference, VTC- 96, Atlanta GA, pp. 1796 - 1800, May 1996.

[Kam2] B. Kamali, Lecture notes on Reed-Solomon codes, Mercer University, Macon GA, 1995

[Kam3] B. Kamali, and R. L. Brinkley, "Some New Outlooks on Burst Error-correcting Capabilities of Reed-Solomon Codes and Their Application in HDTV", draft.

[Kam4] B. Kamali, and R. L. Brinkley, "Reed-Solomon Coding and GMSK Modulation for Digital Cellular Packet Data Systems", Proceedings of IEEE Pacific Rim Conference (PACRIM97), Victoria, B.C., Canada, August 1997

[Lin] S. Lin and D.J. Costello Jr., Error Control Coding: Fundamentals and Applications, Englewood Cliffs, NJ: Prentice Hall, 1983.

[Mas] J. L. Massey, "Shift Register Synthesis and BCH Decoding," IEEE Transactions on Information Theory, Volume IT-15, Number 1, pp. 122 - 127, January 1969.

[Mee] C. D. Mee, E. D. Daniel, Magnetic Recording, Volume II: Computer Data Storage, New York: McGraw-Hill, 1988.

[McE1] R. J. McEliece, Finite Fields for Computer Scientists and Engineers, Boston: Kluwer Academic Publishers, 1987.

[McE2] R. J. McEliece and L. Swanson, "Reed-Solomon Codes and the Exploration of the Solar System", in Reed-Solomon Codes and Their Applications, Wicker and Bhargava (ed.), New York: IEEE Press, 1994.

[Nas] NASA Technical Support Package, "Tutorial on Reed-Solomon Error Correcting Coding", NASA Tech Briefs, MSC-21834.

[Pet1] W. W. Peterson and E. J. Weldon Jr., Error-correcting Codes, 2nd ed., Cambridge, MA: MIT Press, 1972.

[Pet2] W. W. Peterson, "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes," IRE Transactions on Information Theory, Volume IT-6, pp 459-470, September 1960.

[Pur] M. B. Pursley and W. E. Stark, "Antijam Capability of Frequency-Hop Spread-Spectrum with Reed-Solomon Coding," proceedings of MILCOM 1983, Volume 1, pp. 7-10, October 1983.

[Ram] J. L. Ramsey, "Realization of Optimum Interleavers," IEEE transactions on Information Theory, Volume IT-15, pp. 338-345, May 1970.

[Ree] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," SIAM J. Appl. Math., Volume 8, pp. 300-304, June 1960.

[Sim1] M. K. Simon, S. M. Hinedi, and W.C. Lindsey, Digital Communication Techniques Signal Design and Detection, Englewood cliffs, NJ: Prentice-Hall, 1995.

[Sim2] M. K. Simon, J. K. Omura, R. A. Scholtz, B. K. Levitt, Spread Spectrum Communications, Rockville, MD: Computer Science Press, Inc., 1985

[Sin] R. C. Singleton, "Maximum Distance q-nary Codes," IEEE Transactions on Information Theory, Volume IT-10, pp. 116-118, April 1964.

[Skl] B. Sklar, Digital Communication, Fundamentals and Applications, Englewood Cliffs, NJ: Prentice Hall, 1988.

[Ste] R. Steele, Mobile Radio Communication, New York: IEEE Press, 1992

[Swe] P. Sweeney, Error Control Coding An Introduction, Englewood Cliffs, NJ.:Prentice-Hall International, 1991.

[Tor] D. J. Torrieri, Principles of Military Communication Systems, Dedham, MA,:Artech House Inc., 1981.

[Vau] R. G. Vaughan, “Signals in Mobile Communications: A Review”, IEEE Transactions on Vehicular Technology, Volume VT-35, No. 4, pp. 133-145, November 1986.

[Vri] L. B. Vries and K. Odaka, “CIRC - The Error-Correcting Code for the Compact Disc Digital Audio System,” Collected Papers AES Premiere Conference, Rye, New York, pp. 178-188, New York: Audio Engineering Society, 1982.

[Wic1] S. B. Wicker, Error Control Systems for Digital Communication and Storage, Englewood Cliffs, NJ: Prentice Hall, 1995.

[Wic2] S. B. Wicker and V. K. Bhargava, “An Introduction to Reed-Solomon Codes”, in Reed-Solomon Codes and Their Applications, Wicker and Bhargava (ed.), New York: IEEE Press, 1994.

[Wic3] S. B. Wicker, “Reed-Solomon Error Control Coding for Rayleigh Fading Channels with Feedback”, IEEE Transactions on Vehicular Technology, Volume 41, No. 2, pp. 124-133, May 1992.

[Yue] On-Ching Yue, “Design Trade Offs in Cellular/PCS Systems,” IEEE Communications Magazine, Volume 34, No. 34, pp 146-152, Sept. 1996.

[Zie] R. E. Ziemer and R. L. Peterson, Digital Communications and Spread Spectrum Systems, New York, NY,:MacMillan, 1985.

6. Conclusions And Plans For Future Activity

Program participants have remarked frequently on the success of the RAPCEval contract and on the quality of the results that have been produced to date. The cooperation among various engineering communities and their fruitful interaction has been a very satisfying byproduct of the entire project. Students, university faculty, government and private sector engineers have united in the common attack on a number of priority Air Force electronic warfare concerns. The student master's degree research has focused on topics that have immediate impact on the transitioning of new and improved software and hardware technologies into fielded systems.

Recently, several students have had the privilege of presenting their work at national and international conferences, where comments have been quite favorable. Ron Brinkley presented a paper entitled "Reed-Soloman Coding and GMSK Modulation for Cellular Digital Packet Systems" at an IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PACRIM' 97). Henderson Benjamin gave a talk entitled "Selection of the Most 'Efficient' Reed-Solomon Code for a Specifie Application Using Neural Networks" at the International Conference on Communications (ICC '99), in Vancouver, British Columbia. These accomplishments are appreciated all the more when it has been noted that it is very unusual for a master's level graduate student to present a paper at national and international conferences.

Twelve master's degrees have been awarded at the time of this report. The title of each report is listed in the table together with availability information for each document.

Author	*Report Title*	*Classification*	*Availability*
Mark Astin	"Application of Parallel Computing Techniques to the RAD Algorithm"	Classified	available from MERC or AFRL/SNRP: #AFRL-SN-WP-TR-1998-1088
Henderson Benjamin	"Neural Network System the Selects Reed-Solomon Codes for a Specific Application"	Unclassified	available from MERC or AFRL/SNRP: #AFRL-SN-WP-TR-1998-1056, Section 5.2.9
Ron Brinkley	"Burst Error Correction with Reed-Solomon Codes"	Unclassified	Submitted to AFRL/SNRP for publication – anticipate availability 12/99
Mark Campbell	"Auto-Regressive Spectral Analysis - EW Applications"	Unclassified	available from MERC or AFRL/SNRP: #WL-TR-94-1057, Apx. E

Randy Ford	"Comparison of Differential Evolution to the Simplex Method in Optimization During Passive Emitter Location"	Unclassified	Available from MERC In process of submission for publication to AFRL/SNRP
Claus Franzkowiak	"Four-Pulse Primary RAD Filter Development"	Classified	available from MERC or AFRL/SNRP: #AFRL-SN-WP-TR-1998-1087
Neal Garner	"Error Correction and Prediction for Improved Communication of Time and Time Measurements"	Unclassified	available from MERC or AFRL/SNRP: #WL-TR-96-1161, Apx. D
Joseph Kelley	"A Parameter Determination Alternative for RAD Analysis"	Classified	Available from AFRL/SNRP #WL-TR-95-1005
Joseph Kelley	"MultiGroup Simultaneous RAD Parameter Selection"	Classified	Available from AFRL/SNRP #WL-TR-97-1094
Max Roesel	"Agile RF/PRI Radar Analysis via RAD"	Classified	Available from AFRL/SNRP #WL-TR-95-1020
Dave Schuler	"Comparison of Algorithms for Geolocation of Radar Signals"	Unclassified	Available from MERC- requires need-to-know
Kirk Wright	"Object Oriented Modeling of the AN/ALQ-172"	Classified	available from MERC or AFRL/SNRP: #AFRL-SN-WP-TR-1998-1086

At the present time, several students are planning to start new research. Reports from various colleagues in the military and industry indicate that additional students will be interested in participating in projects of the kind sponsored by the RAPCEval program.