AN INVESTIGATION OF ERROR CORRECTING TECHNIQUES FOR OMV DATA

NASA Grant: NAG8-104

Final Report

September 15, 1992

Submitted to:

NASA/Marshall Space Flight Center
Mr. Dave Harris
Mr. Reggie Inman
Mrs. Lee Ann Thomas
EB-33
MSFC, Alabama 35812

Submitted by:

Mississippi State University
Department of Electrical and Computer Engineering
Frank Ingels, Principal Investigator
John Fryer, Assistant Investigator
P. O. Drawer EE
Mississippi State, MS 39762

AN INVESTIGATION OF ERROR CORRECTING TECHNIQUES FOR OMV DATA

NASA Grant: NAG8-104

Final Report

September 15, 1992

Submitted to:

NASA/Marshall Space Flight Center
Mr. Dave Harris
Mr. Reggie Inman
Mrs. Lee Ann Thomas
EB-33
MSFC, Alabama 35812

Submitted by:

Mississippi State University
Department of Electrical and Computer Engineering
Frank Ingels, Principal Investigator
John Fryer, Assistant Investigator
P. O. Drawer EE
Mississippi State, MS 39762

REPORTS CONTENTS

SECTION	TITLE	PAGE
1.0 INTRODUCTION	.....	3
ATTACHMENT #1	Considerations of Testing the OMV System ..... With Class; Monthly Report February 12, 1989	5
ATTACHMENT #2	OMV Class Test Results (First Go Around), ... Monthly Report, May 12, 1989	32
ATTACHMENT #3	Equivalent System Gain Available From R–S ..... Encoding Versus a Desire to Lower the Power Amplifier From 25 Watts to 20 Watts for OMV, Memo, 27 August 1989	58
ATTACHMENT #4	Command Word Acceptance/Rejection Rates ..... for OMV, Memo Included in 12 November 1989 Monthly Report	68
ATTACHMENT #5	A Memo Concerning Energy–to–Noise Ratio for ..... the Viterbi–BSC Channel and the Impact of Manchester Coding Loss, Memo. 22 January 1990	89
ATTACHMENT #6	Probability of False Polynomial Division ..... Synchronization Using Shortened Cyclic Codes, Included in 16 May 1991 Monthly Report	104
ATTACHMENT #7	An Investigation of Error Correcting Techniques . for OMV and AXAF, Quarterly Report, 16 November 1991	Separate Bound Cover
ATTACHMENT #8	Investigation of WISP Near Fields	Separate Bound Cover

1.0 INTRODUCTION

The grant, NAG8-104, was instituted May 12, 1988, with a first-year funding of $35,995. A second funding for $46,145 was awarded on May 12, 1989, and a final funding for $50,350 was awarded on June 7, 1990. Originally proposed as a three-year grant to study error correcting techniques for OMV, the investigation took several interesting turns due to redirection of the OMV project and then cancellation of the OMV project. After the OMV project was cancelled, the investigation turned toward investigative support of the WISP project. Several no-cost extensions were requested throughout the life of the grant, and the final expiration of the grant is September 1992, some four and a quarter years after the initiation of the grant.

Mississippi State has felt that the no-cost extension allowed a better use of resources and resulted in more effort and hopefully more results for the dollar expenditure for NASA/MSFC. The average cost per year of this grant has been $29,442, as opposed to the originally proposed average cost per year of $44,163, yet the work has been steady and at times it was felt to be fruitful.

Whereas the original investigations revolved around studies and simulations of the error protection for the OMV image compression data link, other interesting and related issues were addressed as well, including an analysis of OMV command word acceptance/rejection rates, the OMV transponder lock spin/dock problem, some CCSDS issues relating to OMV and AXAF, testing of AHA/NASA Reed-Solomon ECC chips and Investigation of the Electromagnetic Field Structure in the Space Shuttle Cargo Bay due to the WISP antenna.

Several reports have been delivered during the performance of this grant. In particular, the following reports concerned specific topics and/or present the culmination of work on a topic:

1. Considerations of testing the OMV system with Class, Monthly Report, February 12, 1989, Attachment 1.
2. OMV Class Test Results (first go around), Monthly Report, May 12, 1989, Attachment 2.
3. Equivalent System Gain Available From R-S Encoding Versus a Desire to Lower the Power Amplifier from 25 Watts to 20 Watts for OMV, Memo to L. A. Thomas, 27 August 1989, Attachment 3.
4. Command Word Acceptance/Rejection Rates for OMV, Memo to L. A. Thomas, 30 September 1989, and again on 12 November 1989, Attachment 4.
5. A memo concerning Energy-to-Noise Ratio for the Viterbi-BSC Channel and the Impact of Manchester Coding Loss, Memo to L. A. Thomas, 22 January 1990, Attachment 5.
6. Probability of False Polynomial Division Synchronization Using Shortened Cyclic Codes, Anna Lynn Schauer and Frank M. Ingels, Report to L. A. Thomas, April 1991, (an in-depth analysis extending the work of report two listed above), Attachment 6.

7. An Investigation of Error Correcting Techniques for OMV and AXAF (Testing of AHA/ NASA Reed-Solomon ECC Chips), Final Report, John N. Fryer, Ken Lawrence and Frank Ingels, September 28, 1991 (included in November 16, 1991 Quarterly Report). Separate cover.

8. A Determination of the Near Field Strengths of the WISP Antenna in the Space Shuttle Orbiter Bay, September, 1992.

While a diverse topic range was covered during the performance of the Grant, the work was interesting, informative, and it was hoped that it was beneficial to NASA/MSFC.

ATTACHMENT 1

Considerations of Testing the OMV System With Class, Monthly Report, February 12, 1989

AN INVESTIGATION OF ERROR CORRECTING TECHNIQUES FOR OMV DIGITIZED DATA

NASA GRANT: NAG8-104
QUARTERLY REPORT
February 12, 1989

SUBMITTED TO:

Mrs. Lee Ann Thomas
Mr. Glenn Parker
EB33
NASA/MSFC
MSFC, AL 35812
205+544-3662

SUBMITTED BY:

Frank Ingels, Ph.D.
Principal Investigator
Mississippi State University
Mississippi State, MS 39762
601+325-3912

1.0 Work Summary (November 1988 - February 1989)

During the last of October and early November, a three-day meeting of the OMV Video Compression data link design, development and CLASS testing principals was held at NASA/MSFC.

The major topic was the actual hardware concept/design versus software simulation for CLASS testing of the VCU, R-S encoder, helical interleaver, TDRS link (including the convolutional encoder-interleaver and the Viterbi decoder interleaver), R-S decoder and deinterleaver sync and the VRU including subframe replacement.

Fairchild, LinCom, CYCLOTOMICS, TRW, NASA/GSFC and NASA/MSFC were represented at this meeting. Preparation for the meeting included compiling a question list, informing the participants of the need to prepare a briefing and laying out a schedule including time for in-depth discussion of details raised during the briefing. Mrs. Lee Ann Thomas took the lead in the meeting preparation and conduct.

The results were satisfactory in many ways. Details of tape conversions were solved, details in technical operations were uncovered, discussed and agreed upon by all parties. A tentative schedule of procedures to CLASS test was agreed upon, as were the CLASS test requirements and desired data outputs.

In the appendix to this report is a typed copy of the notes taken during this meeting. These notes are not edited but are preserved as taken to serve as a memory aid in recalling the meeting particulars.

Pertinent results of this meeting are noted below:

1. Hardware is not complete, thus modeling will be as faithful as possible but not 100% accurate. Variable rate buffer finished in strategy is not yet fixed in final design. R-S synchronizer strategy is fixed but final choice of counter presets and thresholds are not fixed.

   Until hardware finalized and tested against CLASS with RFI and random errors, the system is not frozen. Hence, software details must be flexible to allow variations.

2. Probability of false synch lock in R-S synch strategy is very low.

   P(of .8 bits of data looking like R-S synch word) = 2^{-8} = 3.9×10^{-3}.

   P(of X consecutive sets of 8 bits of data looking like synch) ~ $(3.9 \times 10)^{X}$.

   Thus, 16 successive sets of 8 bits of data to pass synch thereby making a count of 16 in the threshold counters thereby creating a false synch lock. The probability of this happening is

   $$P(\text{False Synch}) = (3.9 \times 10^{-3}) = 2.94 \times 10^{-39}.$$

   For a Viterbi output random error rate of 10^{-5}, the probability of counting up to 16 from a starting count of 0 is

   $$P(\text{Sync Lock}) \sim (1 - 10^{-5})^{8^{16}} = .99872.$$

   For a threshold count of 10 rather than 16, the corresponding probability of false lock and of successively acquiring lock is respectively

   $$P(\text{false lock}) = (3.9 \times 10^{-3})^{10} = 8.14 \times 10^{-25}$$

   and

$$P(\text{sync lock}) = (1 - 10^{-5})^{8} = .9992.$$

Thus, a tradeoff of the false lock probability versus successful synch lock for a threshold setting is illustrated. The threshold setting is not yet determined.

3. A standardization of word sending structure (least significant bit first, most significant bit last) was agreed upon. A copy of this structure is appended to this report.

4. LinCom and Fairchild apparently worked out most difficulties in transferring code. Not all problems were solved, however. It would be best for Fairchild's computer software programmer to go to LinCom/NASA/GSFC to oversee changes necessary.

5. Use of the RS undecodeable flag is not presently incorporated in the VRU.

6. A set of test vectors is being constructed for use by Fairchild and LinCom. These test vectors will be used to validate the R-S coder/decoder software.

7. Output of CLASS test tape format agreed upon.

8. Root polynomial and root used to generate R-S field is to be (has been) provided to LinCom and NASA/MSFC.

9. The functional software flow and functional software construction in modular form provided by LinCom is illustrated in Figure A.1 in the Appendix.

10. A list of questions developed for the meeting is attached to this memo.

Since the meeting at MSFC during 30 October 1988 to 2 November 1988, a series of telecons have been held so as to monitor the progress to test data. These telecons have extended through February 1989. A meeting was held in December at NASA MSFC between Lee Ann Thomas, Glenn Parker and Frank Ingels.

Several items were uncovered during the software conversion by LinCom. The non-use of the R-S undecodable flag has been bandied about and to date is not incorporated into the VRU. The channel ID which determines the alternate camera frames was not incorporated into the VRU software. This is being fixed so that two different scenes may be simulated and camera de-multiplexing tested during CLASS testing.

A software anomoly in the 4 and 10 bit line subframe replacement was uncovered and is being fixed. It is normal to experience a small boundary disturbance between subframes due to change from 2 dimensional encoding to 1 dimensional encoding.

A new bit rate controller design is to be incorporated into the software. This design takes double the memory of the original design.

Full configuration 1 testing is absolutely recommended. It is necessary to see and study the results of configuration 1 testing before relinquishing this demand. Consistent test results for sync loss and recovery time are the most important guidelines for success of configuration 1 testing. Testing is now scheduled to start in early March, 1989.

QUESTIONS PREPARED BY MS. LEE ANN THOMAS FOR NOV 88 MEETING

1. How is the sync circuit of the R/S modeled? LinCom stated that Cyclotomics has not yet decided on the counter for this circuit; has LinCom allowed for this by using variables as counters?

2. What type of RF channel testing is planned for the CLASS test?

3. What type of video display and error statistics are being used by LinCom for presenting reconstructed video?

4. Does LinCom now have all the data, algorithms and schematics necessary for development of the R/S sync/encoder/decoder/helical interleaver/deinterleaver/bit sync/Viterbi/video subframe sync?

5. Does LinCom feel that there is a problem with the 'windowing' scheme necessary for achieving the 24-bit sync word for the video sub-frames? Has this been incorporated into the model?

6. What is necessary for interfacing various hardware/software for the test? Who coordinates this effort? What is the status concerning interfaces of the Fairchild code, the LinCom code and the CLASS code?

7. How is the flag indicating uncorrectable errors handled by the VRU? Has this been coordinated between Fairchild and LinCom?

8. How is the sub-frame replacement handled?

9. Exactly what functions does the McIntosh perform? What algorithms/answers are used? Are error characteristics of the burst evaluated at the input of the Viterbi, the output of the Viterbi, and the output of the R/S decoder.

10. What steps occur between the VRU software and the actual display of the video?

11. Define 'real-time' and discuss how this applies to the OMV CLASS test.

12. What are the factors involved in the selection of techniques (why was a burst error generator designed to use for testing instead of using the RF link?

13. What is the status of the tape transfer from VAX/VMS to HP/UNIX?

14. LinCom stated that they did not have the correct R/S sync word from Cyclotomics. Verify that they have obtained the correct word.

15. How is the video sub-frame sync modeled?

16. LinCom stated that interfaces must be developed between the bit sync, Viterbi decoder, and the R/S sync. Please define this problem and describe algorithms necessary to obtain these interfaces.

17. Define all computers/interfaces necessary to achieve the CLASS test.

18. Does the video software model the bit filler/elastic buffer/change of rate?

19. Where will the hooks for the CLASS test be located for the VCU, VRU, R/S encoder/decoder, CLASS system?

20. Do we have the ability to take a burst hit on every sync (bit sync, Viterbi, R/S decoder, and video sub-frame sync) and go back and re-acquire sync in all cases?

21. CLASS Model

    - What is the degree of fidelity between the software model and the actual hardware implementation (this includes all hardware modeled end-to-end for the CLASS)?
    - How are the four levels of sync modeled?
    - What is meant by transient environment?
    - How is the receiver modeled?
    - What happens when there is receiver saturation?
    - How is the Costas Loop simulated?
    - How is the bit-by-bit decoding after bit sync accomplished?
    - Does the Viterbi use hard or soft decision?
    - How is the convolutional coding accomplished?
    - How is the periodic interleaver/deinterleaver modeled?

22. LinCom listed four or five possible reasons for losing video sub-frame sync. Will the software model allow the user to identify which of these possibilities actually caused the loss of sync?

23. Has LinCom contacted Cyclotomics to verify that the use of a different primitive root in the R/S model will have no affect on the CLASS test?

24. What is Stanford's involvement in the CLASS test?

25. What type of computers will be used for the CLASS test? Since LinCom (Stanford) is converting the Fairchild VAX/VMX/Fortran 77 to the HP9000/UNIX, will this code undergo another software conversion?

26. How will integration/verification of all these computational modules be accomplished? Who will write/approve this?

27. Where is the data from the CLASS being analyzed? Name the specific points where data will be available for evaluation of system performance.

28. Can Goddard develop test cases to validate their model?

29. What hardware verifications have been performed to validate the CLASS?

30. Will CLASS tell if the error correction (Viterbi and R/S) scheme is (optimum? non-optimum?) in performance under normal and adverse (RFI and dropouts) conditions?

31. Does CLASS have to be re-initialized with another sub-routine if a data dropout occurs?

32. Will CLASS model the WSGT receiver under adverse conditions?

    - How long of a dropout can be tolerated without unlocking the receiver?
    - How long does it take to relock the signal?
    - The NASCOM to JSC hop contains Doppler effects from the satellites - buffering on the ground takes out these effects; is this modeled for CLASS?
    - How is the clock handled for the bit sync, Viterbi, R/S? The Viterbi and R/S are at two different locations - how is this clock modeled for CLASS? Is this similar to the actual technique implemented?

33. What hardware/software configuration is necessary to demonstrate the Fairchild tape (end-to-end)? Does CLASS have a hardware configuration to support this?

34. Description of 'hooks' in Fairchild tape.

35. Does Cyclotomics have a software model for the R/S encoder/decoder?

36. Does LinCom have all information and complete code from Fairchild?

37. What is the status of concatenated channel error pattern generator algorithm?

38. Why was LinCom planning on using a <u>statistical</u> sub-frame corruption generator instead of RF link?

39. What is the status of STI?

40. What is the MacII being used for?

41. What is the status of Run Control, R/S subsystem, PCI, dePCI, Receiver, Viterbi sync, sub-frame sync?

42. Verification by Fairchild to LinCom the following:

    1) RAM size
    2) disk size necessary

UNEDITED FIGURES AND

31 OCTOBER 1988 NOTES TAKEN DURING OMV NASA/MSFC

CLASS TEST MEETING

Steve Jones:

Launch scheduled for late 93-94, trying to go for Late Qtr 93. CDR slipped from 1990 to 1991. Steve will push for hardware test following CLASS. Our objective here to be ready for CLASS/HARDWARE test. (Phone Mess. 544-3626)

Lee Ann Thomas:

5 Objectives:

1. Evaluate present software approach used for CLASS testing.
2. Determine if present CLASS system can be modified to be HI FI model of OMV video downlink.
3. Detailed review of software/hardware tests on algorithms.
4. Verify all hardware/software.
5. Discuss additional hardware/software test plans.

Bob Godfrey:

Discussion of CLASS Communications Link Analysis and Simulation System.

Question in CLASS models: to what detail is modeling? Bit-by-bit or statistical estimate.

Function - combines several components together (speedy).

Simulator - bit-by-bit full Hardware Emulation (validation).

R/S can be done either way. Originally planned to be functional.

1st CLASS test was functional.

If 200-300 bits in error, then there will be no cycle slip.

1500-1600 average bits in error for cycle slip assuming sufficient SNR to resync.

TDRS requires PCI at these data rates.

Takes longer to recover sync if low SNR.

Almost never in standard Gaussian environment with TDRS.

Purpose of PN cover sequence is to prevent Viterbi decoder to false lock-up on wrong bit pairs.

Bob Godfrey has more faith in functional test with TDRS portion. Because not all hardware is finished in design! Good point. Also each specific box is different. Which do you simulate? Functional covers global results.

Full simulator approach is being planned after last two telecons. Functional test also available. RFI is statistical emulation and non-stationary.

Robert Godfrey recommends functional use simulation to validate functional.

Some pieces (RFI) are only statistical models. If any part is statistical, then functional is best. Functional is (analytic/hybrid with hardware) average over some set of bits. Simulation is watching a bit all the way through.

RFI updated in 3-month intervals if necessary.

Is channel environment (XMIT and RCVR) modeled OMV output to WS input? Transients modeled.

2 MAJOR WAYS TO USE CLASS:

Static: (Put user in orbit and set up environment and go). Select any set of parameters you want (once in 100 years).

Dynamic: In normal mode environment is anticipated designed for typical flight condition. Could put in once in 100 year occurrence using override.

We can worst case environment until we kill ourselves. Antenna switching is modeled.

CLASS models forward link.

Validation tests on down link were within 1/2 db of actual.

Validation tests on forward link were within .01 db of actual.

31 OCTOBER 1988

GAR LEWIS in place of JERRY O'CONNER

Cyclotomics test vectors being made up.

FW gives to NASA/GSFC and then FW wants test vectors back.

CAN WE HAVE TEST VECTORS APPROPRIATE TO EACH:	FW	Ask
CODE EMULATIONS??	LinCom	Cyclotomics

The Cyclotomics sync is clever!! Combined with undecodable error flag!

Preload R/S sync counters with unknown number?

What is R/S sync codeword (8 bits)?

Can R/S be set to sync up with less than 16 R/S sync patterns?

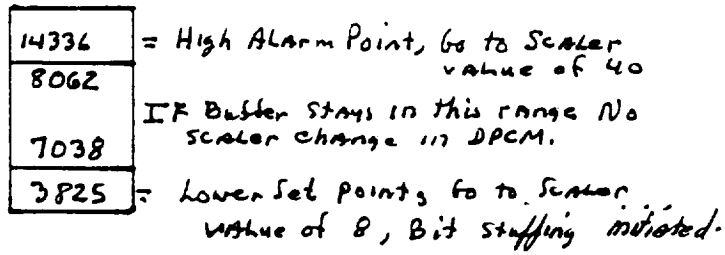
In Cyclotomics algorithm 'clear sync counters . . .' means?

CURRENT FW ENCODER INTERLEAVES DATA AS WELL AS PARITY.

31 OCTOBER 1988

JERRY O'CONNER

BIT RATE CONTROLLER



FW has yet to find a picture which overflows the buffer!

If the buffer overflows, you will recover, but old data used for recover period.

Roger says some function calls are missing in their code.

But FW says they don't use image processing in the code that went to LinCom.

How to find

Glenn said 'What do we need at MSFC to run FW tapes with software/hardware test here at MSFC?'

FW said MSFC needs a buffer to store 150 frames, and VAX will do it.

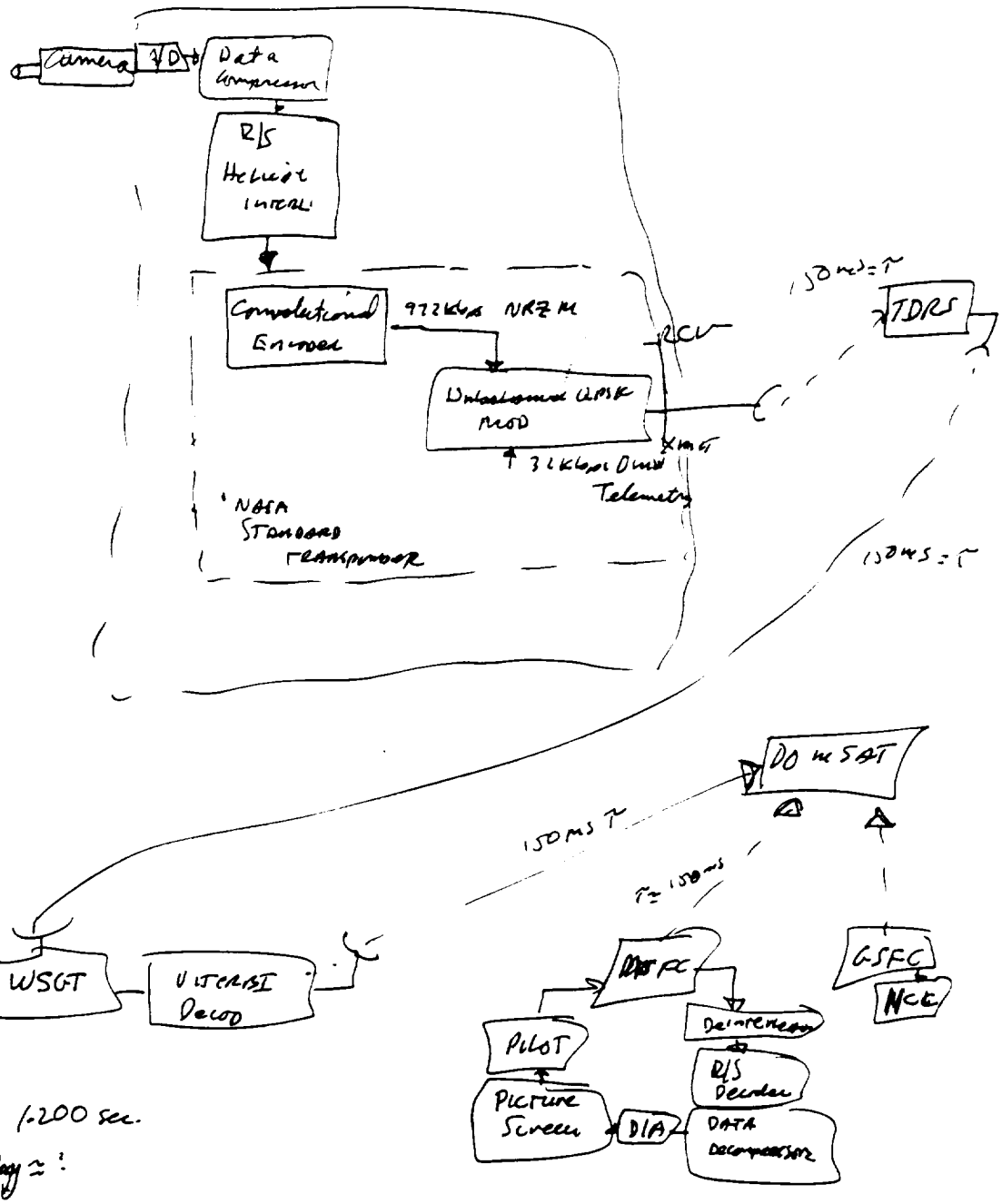
RAM requirements are 64 K Bytes x 16 = 1024 M Bytes.

Side discussion 30 minutes Roger and Jerry code conversion!!

5

31 OCT 88

Roger Huant



δT = 1.200 sec.

Pilot delay ≃ ?

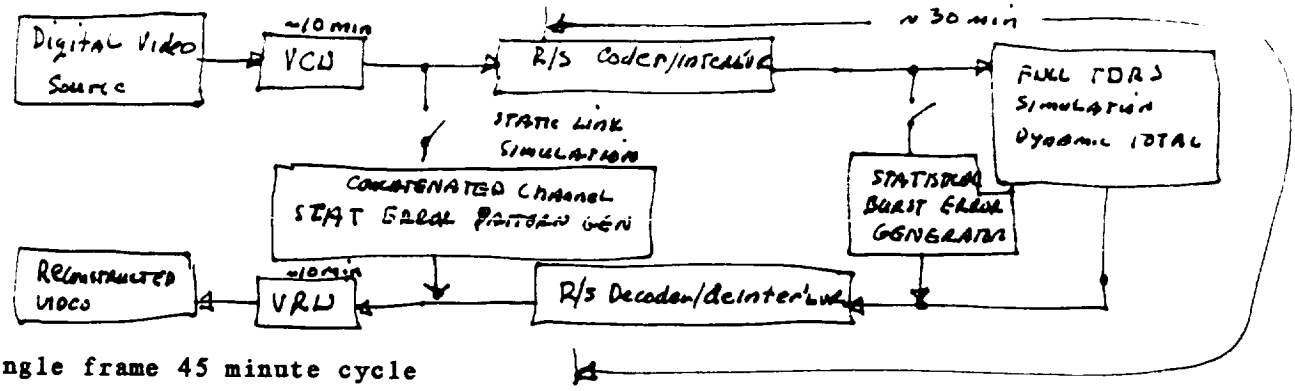
OMV Video/Telemetry Channel.

31 OCTOBER 1988

To run full simulation for 150 frames, computing time would be 2 weeks for 1 mth. This would be 56 different test configurations x 1 mth.

Statistical burst generator can shorten this time <u>if</u> you are synced up!!

Could add concatenated channel statistical error pattern Gen. and by pass all of system.



Single frame 45 minute cycle

<u>Error Performance</u>

Total Pixel Errors vs EIRP Ratio

Pixel SNR vs EIRP Ratio

Error Propagation Statistics vs EIRP

Subjective

THESE ARE IMPORTANT:

Sync transient curves: How long were you out of sync, etc.?

What caused the errors?

1 NOVEMBER 1988

Bob Law - Cyclotomics and Tze-Hwa Liu

GCC Hardware:

1. R/S decoder (No. PLL) upgraded (to space level?)
2. Bit error monitor included.
3. Super channel monitor.
4. Update/report bit errors and undecodables.
5. BCD displayed.
6. OP code defined by customer.

COMPATABILITY TO OMV OF MODIFIED 888C UNIT AT MSFC

Difference of OMV and 888C at MSFC

(Functionally: same) (Same sync, same helical interleaver, same decoding algorithm)

1. Interface: bursty data vs continuous data
2. Clock: No. PLL
3. Interleaving depth fixed to 8
4. Same helical interleaving
5. Undecodable flag: indicate following block is wrong
6. Data rate: 1.544 Mbps vs 972 (OMV) Kbps, master/slave driven by user
7. Byte: separate byte for OMV (built-in test equipment)

LINCOM TREATS THE R/S SYSTEM AS FIVE UNITS

1. R/S Encoder
2. Helical Interleaver
3. R/S Sync Circuit
4. Helical Deinterleaver
5. R/S Decoder

Primitive or root poly.

Which roots used to generate field?

Two polynomials: Primitive Poly.

Parity Generator Poly.

Test vectors must be different for different roots.

8

IN0088

Dennis LAI

25 (254 + 1 Sync, 238) $\lambda = 8$ Parity Checks = 16

2040 bits = 1 frame

2032 bits = RS WORD +8 bits Sync.

1. Gen. Poly $G(X) = \prod_{J=R_S}^{R_S+15} (X+\alpha^i) = \sum_{K=0}^{16} G_K X^K$

α = Prim. element in $GF_1(2^8)$

2. Encoding: obtain $M(X) = \sum_{K=0}^{237} M_K X^K$ obtain $Z(X) = X^{16} M(X) \bmod (G(X)) + \sum_{K=0}^{15} R_K X^K$

$C(X) = X[X^{16} M(X) + R(X)] + Sync(X)$

3. $R(X) = X[X^{16} M(X) + R(X)] + E(X) + Sync(X) = C'(X) + Sync(X)$

Decod evaluate $C'(X)$ at the root of the Gen-Poly

$S_j = C'(\alpha^j)$ $S = S_0 + S_1 X + S_2 X^2 + \cdots$

$b_0 \cdots b_7 \; b_8 \cdots b_{15} \quad \cdots\cdots \quad b_{1896} \cdots b_{1903} \; b_{1904} \cdots b$

M_{237} M_{236} M_0 M_{237}

How a Symbol is Defined on Incoming Bit Stream.

⊗ [LIN COM uses MASSEY - BERLEKAMP ALGORITHM + [First Element of Finite Field.]
CYCLOTOMIC " Berlekamp Algorithm]

→ Find Polynomials
Find roots of Error Locator Poly. If # of distinct roots of degree of Error Polynomial = Undecodable.

1 NOVEMBER 1988

QUESTIONS:

1. LSB or MSB 1st
2. Which symbol comes first?
3. Massey-Berlekamp versus Berlekamp. Differences?
4. Test vector impact!
5. Where in sync block is FW ID?
6. What is FW ID channel 1?
7. What is FW ID channel 2?

TEST VECTOR SIGNATURE ANALYSIS

Generate a periodic 2040x8 test pattern

Unknown phases

THE LEGAL ASPECTS OF TEST VECTORS

I recommended two sets: to go from FW to LinCom

A. 1 with answers

B. 1 with no answers

This agreed to by Gar Lewis and Bob Law.

By 18 Nov 88 these vectors will be sent to LinCom by FW.

Steve entered the discussion.

1 NOV 88

Dennis asks one more question regarding test vector.

AGREED FORMAT FOR TEST VECTORS

(1,7) is:

MSB ⟶ LSB

0 0 0 1 0 1 1 1 Transmission is LSB first, MSB last.

$[1_7\ 1_6\ 1_5\ 1_4\ 1_3\ 1_2\ 1_1\ 1_0]$

1st Bit Received

$1_0\ 1_1\ 1_2\ 1_3\ 1_4\ 1_5\ 1_6\ 1_7$ — 1st Byte

$1_0\ 1_1\ 1_2\ 1_3\ 1_4\ 1_5\ 1_6\ 1_7$ — 2nd Byte

R/S FORMAT:

Sync $M_{237}\ M_{236}\ \cdots\ M_0\ C_{15}\ C_{14}\ \cdots\ C_0$ Sync $M_{237}\ \cdots$

$M_{237}\ M_{236}\ \cdots\ M_0\ C_{15}\ C_{14}\ \cdots\ C_0$ — R/S Word I

$M_{237}\ \cdots$ — R/S Word I + 1

First Received M_{237}, Second Received M_{236}, etc.

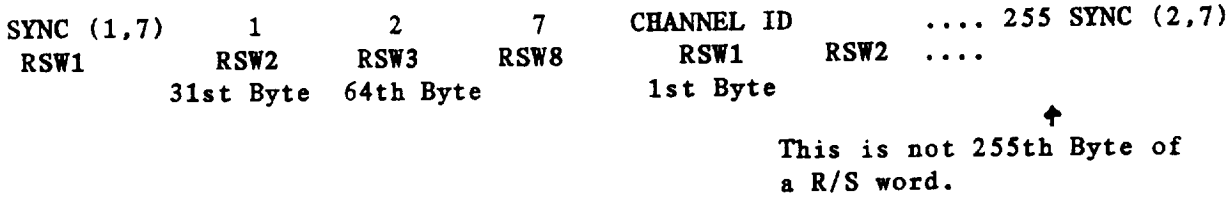
Sync is 0 0 0 1 0 1 1 1 ≡ 1,7 as above.

OUTPUT FORMAT SAME AS ABOVE.

1	UNDECODABLE PATTERN	1ST BYTE
0		2ND BYTE
.		.
.		.
.		.
0		255TH BYTE SYNC (1,7)
.		.
.		.
.		.
0		2040TH BYTE SYNC (1,7)

1 NOV 88

SYNC BLOCK



SUB-FRAME SYNC

1. Hardware won't look for sub-frame sync until opening window near bottom of sub-frame. This prevents compressed video from looking like sub-frame. This windowing doesn't occur until sync is acquired.
2. If you don't get garbage look with window (if not sync'd Huffman decoding fails).
3. If you don't get sync - open window to find sync. Then go back to window.
4. Where to reinsert undecodable flag in VRU?
5. Test points in VRU and VCU.
6. Process 150 frames of data. Results?

   Can we count pixel errors? Rather than just sub-frame replacement? As it stands now, 1 pixel error can cause sub-frame replacement. However, after one pixel in error, the VRU stops! All pixels after are bad.
7. Bit sync Viterbi Sync RS Sync Sub-frame sync

   Lose Δ value, lose pixel, lose Huffman table

   Lose channel ID (Rare! - it has multiple cks and flywheels if you miss one.)

Channel 1 1 0 1 1 1 0 0 0 ID 8-bit Irig Unique Word

Channel 2 0 1 0 0 0 1 1 1 ID 8-bit Irig Unique Word

LSB?

8. Can we determine which problem caused loss of picture, i.e., which sync?

9. Can we determine re-sync acq time for each? For inter-reaction of sync's? Interdependence, etc.

10. Simulation is done in NRZ-M.

2 NOV 88

What equipment is being tested, where, what?

VCU/VRU Compatibility test

Then tied together as unit including camera. No end-to-end test.

Suggest end-to-end test using Goddard Test Van Fairchild. Goddard has req for end-to-end using only compatibility test and test van.

Compatibility test (is pre-flight readiness) is not performance test but only run with usually low power testing to see data format and all connectors, etc. are compatible with TDRS link. Channel environment and flight dynamics are not simulated in NASA/GSFC compatibility test van. Assures signals will pass through TDRS.
User does not drive these tests. They are wet and dried tests. It is certification process required before actual TDRS usage.

To schedule van. 1 year lead time. 1 week typical max usage at time. Could use your own transponder and dish and schedule TDRS time. Ironically, it is much easier to schedule TDRS time than to schedule the compatibility test van.

TRW will define the requirements for compatibility and other tests while in the van.

Glenn: Is anybody here running the system level BER tests?, i.e., duplicating CLASS with hardware to check hardware.

2 NOV 88

TEST PLAN

Boris

Objectives:

Model OMV video interleaver to CLASS perform end-to-end

SIM TEST

1. Sync versus E_b/No. Viterbi and PCI
2. RS frame sync
3. Helical deinterleaving and RS coding
4. Reconstruct sub-frames
5. Verify video picture quality

E_b = Channel Bit or Channel Symbol

∴ ~ .5 μsec wide

INDWT

DATA TO RUN TEST ON DIGITIZED RS-170 VIDEO DATA TEST PATTERN(S)

FOR CALIBRATION - illustrate freq response LinCom/GSFC

(30 seconds = 150 frames) (10 seconds = 50 frames)

10 seconds of static space craft scene FWSI

10 seconds of static rotating space craft scene FWSI (This is 6 tapes 150 frame tape FW)

N seconds of TBD scene from NASA/MSC (No docking tape - this could be docking)

Comment 200 frames required to get good, stable statistics

.'. Run 40 seconds - recommended

Tape Record

VRU output R/S Enod/nter Output

Input to R/S

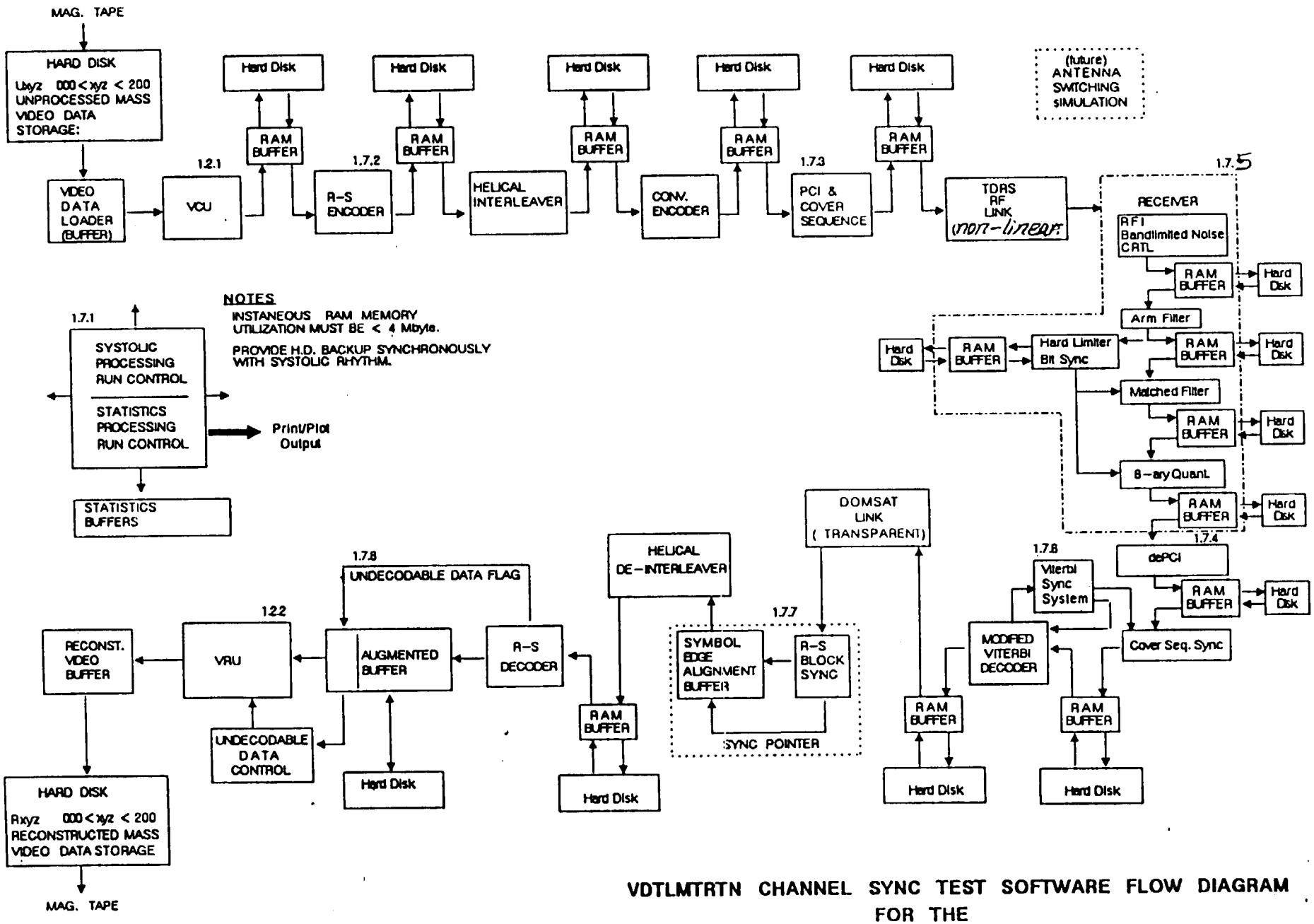
Records

Time for R/S sync (Lee Ann will give me copy)

(Phase II testing = dynamics test -- not sure we'll run!)

Additional time to modify VRU to give sub-frame replacement statistics and accept R/S undecodable flag.

TL
31 OCT 88



VDTLMTRTN CHANNEL SYNC TEST SOFTWARE FLOW DIAGRAM FOR THE CLASS OMV DATACOMP TEST

ATTACHMENT 2

OMV Class Test Results (First Go Around), Monthly Report, May 12, 1989

AN INVESTIGATION OF ERROR CORRECTING TECHNIQUES FOR OMV DIGITIZED DATA

NASA GRANT: NAG8-104
QUARTERLY REPORT
MAY 12, 1989

SUBMITTED TO:

Mrs. Lee Ann Thomas
EB33
NASA/MSFC
MSFC, AL 35812
205+544-3662

SUBMITTED BY:

Frank Ingels, Ph.D.
Principal Investigator
Mississippi State University
Mississippi State, MS 39762
601+325-3912

WORK SUMMARY - FEBRUARY 12, 1989 - MAY 12, 1989

During this period, several meetings at NASA/GSFC were attended. These meetings concerned CLASS simulation test runs for OMV. In Addendum 1 is a report that was submitted concerning the 26-30 March 1989 meeting at NASA/GSFC, and in Addendum 2 is a report that was submitted concerning the 17 April 1989 meeting.

Other factors have arisen since the 17 April 1989 meeting. These factors concern the User Signal Constraint budget as delineated by TRW memo of 11 April 1989 from C. Y. Yoon to B. Dobrotin and the Modified OMV Return Link Calculation as performed by Ted Kaplan, 17 May 1989 (telephone 301+464-8900).

On the first item, User Signal Constraint budget, I have requested the TRW IOC on "The Effect of Gain/Phase Imbalance on the Performance of the DBS, Part II," by C. Yoon, February 11, 1983. The data of Figure 1, Degradation Due to Modulator Imbalance in the 11 April 1989 memo from C. Y. Yoon is, I hope, verified in the above requested IOC report.

A question as to the Gain Slope of .1 db/MHz or .2 db/MHz has been discussed in a telecon held 16 May 1989. C. Y. Yoon has stated that the test data meets the 0.1 db/MHz figure although the 0.2 db/MHz figure has been used in the past. THIS ISSUE IS NOT OFFICIALLY RESOLVED AS YET.

The second item, the Modified OMV Return Link Calculation, it is noticed that despite the discussion in the 16 May 1989 telecon, there is no antenna calibration factor of 0.5 db taken off the antenna gain as recommended by Mr. Lee Malone. (In a telephone contact with Mr. Ted Kaplan on 19 May 1989, Mr. Kaplan stated he did not include that since he wasn't sure what number to use or even if that factor had been agreed upon.)

The Q channel power is listed as 14 dbw, but note that in the Return Link Calculation item 6, User Data/Total Power Ratio, subtracts 1.0 db for power sharing. This is appropriate since it effectively reduces the 25 watts (14 dbw) in Q channel to 20 watts (13 dbw) in the Q channel.

Finally, note that the Effective User Margin, item 14, states that there is 3.7 db. However, we must keep in mind that the NASA/MSFC specification requires 3.0 db Effective User Margin. Hence, TRW has .7 db extra margin above the NASA/MSFC requirement. If we deduct Mr. Malone's 0.5 db, then TRW has a .2 db extra.

ADDENDUM 1

TRIP REPORT - CLASS TEST SCHOOL 26-30 MARCH 1989

At the 26-30 March 1989 CLASS Test School, the time to recover from synchronization loss was discussed. Since it appears that simulating a bit synchronizer bit slip during an otherwise usable channel does not appear feasible in the immediate future, this investigator has estimated some synchronization recovery times due to various parts of the system.

Viterbi Decoder Recovery Time From Bit Synchronizer Slip

Worst Case Estimate

Conceivably it might be necessary to search all 30 PN cover sequence states and to clear the interleaver contents before each state search. The interleaver buffer contains 3,600 Viterbi input symbols* (equivalent to 1,800 bits) of interleaved information. Furthermore, approximately 600 input symbols are searched to determine if the error metrics are indicating correct PN sequence lock. Thus, it could take, for a search of all 30 PN sequence states,

$$30\ (3600 + 600) = 126{,}000 \text{ symbols}$$

to recover from a symbol sync loss. Each symbol is equivalent to approximately 0.514 usec (Mode-A), hence a total worst case recovery time is estimated to be 65 ms. (At the slower data rate of 486 Kbps (Mode-C), the recovery time is approximately 130 ms.)

Average Case Estimate

On the average, a PN cover sequence search could encompass only 2 states. For this case, the number of bits required would be

$$2\ (3600 + 600) = 8{,}400 \text{ symbols}$$

to recover, and an average recovery time is estimated to be 4.32 ms for the two camera (Mode-A) data rate of 972 Kbps and 8.64 ms for the one camera (Mode-C) data rate of 486 Kbps.

* A symbol is the output of the rate 1/2 convolutional encoder. The symbol rate is 2 times the data rate.

Reed/Solomon (R/S) Deinterleaver and Decoder Recovery Time From Bit Synchronizer Slip

There are 2,040 bits per R/S word and a depth 8 interleaving, thus 16,320 bits (at 2.06 usec per bit) per R/S interleaved block.

Assuming the need to first clear the interleaver and then to fill at least 15 additional code words to resynch the R/S code word to provide and channel ID, the estimated resynchronization time of the R/S deinterleaver/decoder is

16,320 bits at 2.06 usec per bit = 33.62 ms Clear Interleaver
30,600 bits at 2.06 usec per bit = 63.00 ms Resynch R/S Counters
2,040 bits at 2.06 usec per bit = 4.20 ms To provide Channel ID.

Thus, we have:

Maximum Total R/S Recovery Time = 100.82 ms (Mode-C)
Maximum Total R/S Recovery Time = 50.41 ms (Mode-A).

VRU Subframe Replacement Sync Loss Recovery Time

20 Line Subframes

Approximately 15,300 bits fill 20 lines and if a scene is just missed, hence requiring two subframes of missed data, then the worst case recovery time would be

30,600 bits at 2.06 usec per bit = 62.9 ms (Mode-C),

and the best case recovery time would be

15,300 bits at 2.06 usec per bit = 31.5 ms (Mode-C).

A scenario of recovery times is now estimated as:

MODE-C (486 Kbps data rate, 972 Kbps RF Symbol Rate)

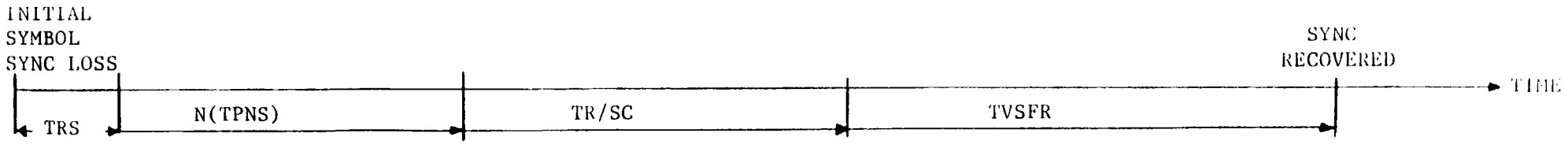
SCENARIO	ESTIMATED BIT SYNC LOSS RECOVERY TIME
A. Worst Case Viterbi plus worst case subframe replacement sync.	130 ms + 100.8 ms + 62.9 ms = 293.72 ms
B. Average Viterbi (2 State Search) plus worst subframe replacement sync.	8.64 ms + 100.82 ms + 62.9 ms = 172.36 ms
C. Average Viterbi (2 State Search) plus best subframe replacement sync.	8.64 ms + 100.82 ms + 31.5 ms = 140.96 ms.

Mode-A recovery times are approximately one-half that of the Mode-C recovery times. The above recovery times do not include the 400 to 700 symbols it takes the Viterbi to detect synchronization loss after an actual symbol synchronization loss occurs. This would add a maximum of 0.36 ms for Mode-A and 0.72 ms for Mode-C operation, not a significant factor in either case. Figure 1 illustrates synchronization recovery procedures.

A constant source of confusion has been the EIRP and EIRP Margin terminology. In an effort to straighten this out, the following exposition is offered.

Figure 2 illustrates a simplified view of the OMV to TDRS to White Sands facility and the three important system points concerning EIRP, EIRP Margin, SNR and expected Viterbi decoded and deinterleaved bit error rate (BER). As noted on the figure, an OMV EIRP adjusted for 46,000 Km free space propagation loss only results in 7.12 db EIRP Margin at TDRS for the 25-watt OMV transmitter.

However, it must be remembered that any waveform distortions (user constraint) polarization losses, plume loss, RFI loss effects and dynamic motion losses are not included as yet. If, at White Sands, a .3 db polarization loss is assumed, and 2.0 db losses assumed for the combination



Time for Viterbi to recognize sync loss after actual sync loss.

Time to clear PCI interleaver (3600 symbols) and to search 600 symbols to decide sync lock or not. This takes place for each PN sequence state searched. Time for 1 PN sequence state search is TPNS, for N PN sequence state search N(TPNS).

Time to clear R/S Deinterleaver (16,320 data bits) and to fill 15 additional code words (30,600 data bits) to resync the R/S code words and to provide channel ID.

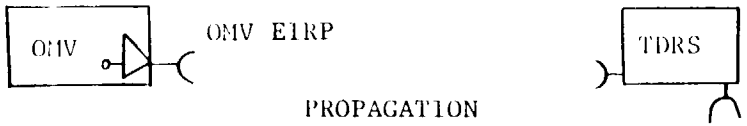
Time to fill 20 lines if a scene is just missed (15,300 data bits), hence requiring two subframes of missed data.

RESYNC TIME = TRS + N(TPNS) + TR/SC + TVSFR

There are 2 symbols emitted from the rate ½ convolutional coder for each input data bit.

FIGURE 1 - SYNC RECOVERY TIME

(-168.02 dbw required to achieve 10^{-5} BER at Viterbi output at White Sands)



PROPAGATION
LOSS
-192.9 db at 46,000 Km Range

OMV EIRP = XMIT power (dbw) + Gimbal loss + RF combiner loss + cabling loss + VSWR losses + antenna gain
= XMIT power (dbw) - 1.6 - 1.8 - .9 - .3 + 23.6
= XMIT power (dbw) + 19 db

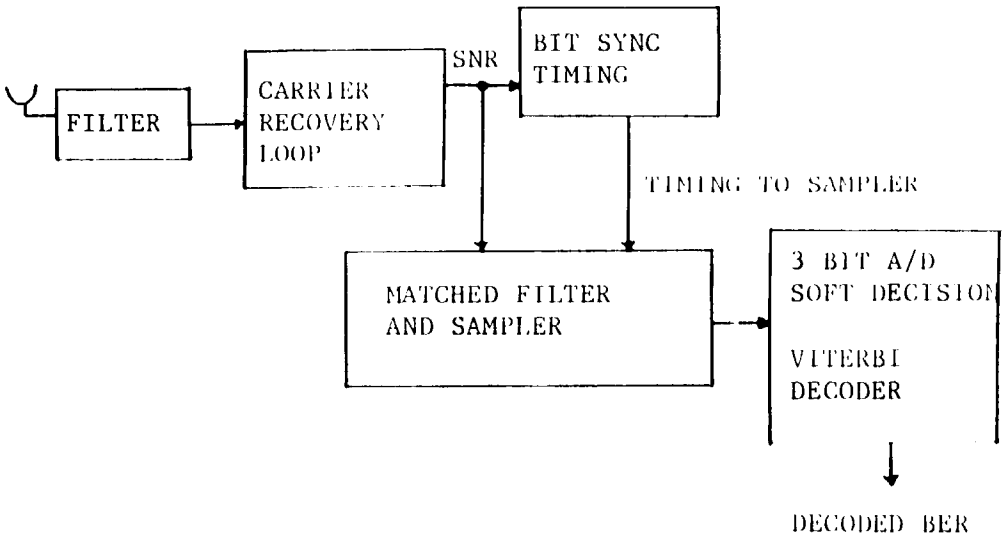
For 25 watt transmitter (20 watts Q channel XMIT power)
OMV EIRP = 32 dbw

EIRP Required at TDRS to achieve Viterbi output BER of 10^{-5} is -168.02 dbw

EIRP from OMV -192.9 db propagation loss = -160.9 dbw for 25 watt transmitter

EIRP Margin at TDRS for 25 watt OMV transmitter is -160.9 dbw + 168.02 dbw = 7.12 db

The above does not include degradation due to RFI, user constraint losses, polarization losses, plume loss, dynamic motion losses.



An SNR of 2.2 db yields a Viterbi output error rate of 10^{-5}. Thus, 0 EIRP Margin at TDRS is roughly equivalent to 2.5 db SNR at Viterbi input.

FIGURE 2 - SIMPLIFIED OMV-TDRS-W.S. LINK

of user constraint losses, plume loss and dynamic motion losses, then the resulting EIRP Margin at TDRS is reduced to 4.82 db for the 25-watt OMV transmitter!

For no RFI, this would be a healthy margin, but if RFI is assumed to create an apparent 2.7 db loss on the average, then the EIRP Margin at TDRS is reduced to 2.12 db, just above that required for a 10^{-5} BER out of the Viterbi decoder. What does this mean to the OMV video system? Figure 3 illustrates a video frame that might have been corrupted by a subframe replacement (4 line subframe) caused by undecodable R/S words. The R/S word errors resulted in a simulation of a channel which had assumed an EIRP Margin at TDRS of 0 db. The channel simulation included RFI effects which actually created an apparent 2.7 db loss (high RFI used) for an effective EIRP Margin (RFI adjusted) of -2.7 db. From this, we might conclude that the 25-watt OMV transmitter situation will produce a very healthy video, better than that of Figure 3.

Once again, we must exercise caution! Subject to waveform distortion measurements of the OMV 25-watt transmitter high power amplifier, we might have as little as 0 db loss or as much as 2.5 db loss due to waveform distortion (user constraint) losses alone! Thus, the above assumed 2.0 db assumed for the combination of user constraint, plume loss and dynamic motion losses might prove to be underestimated.

To ascertain the possible video degradation effects, a series of simulations are planned using CLASS. Phase 1.0 will include only a "static test", i.e., (see page 28 of "Class Testing of the OMV Video Telemetry Channel," by R. Avant, 2nd Draft, 27 January 1989) no dynamic motion, no signal frequency or power level or data rate variations, full synchronization lock conditions. RFI effects will be considered.

The OMV EIRP levels and RFI situation planned is listed on page 52 of "CLASS Testing of the OMV Video Telemetry Channel" mentioned above. These values are listed in Table 1, along with EIRP Margin at TDRS and the Effective Margin at TDRS adjusted for 0.3 db polarization loss and 2.0 db total combined other losses (not including RFI).

From Table 1 we observe that Test 1 is planned with a -1.0 db RFI adjusted level. This will not be as much channel degradation as the channel for the situation depicted in Figure 3.

TABLE 1

PHASE 1 PLANNED TEST PARAMETERS

(Excerpts from "CLASS Testing of the OMV Video Telemetry Channel," by R. Avant, 2nd Draft, 27 Jan 1989, p. 52)

. RFI Level 1 (TDRS-E SSA Worst Case), VCU Mode-A, Data Rate 972 Kbps, 46,000 Km Range

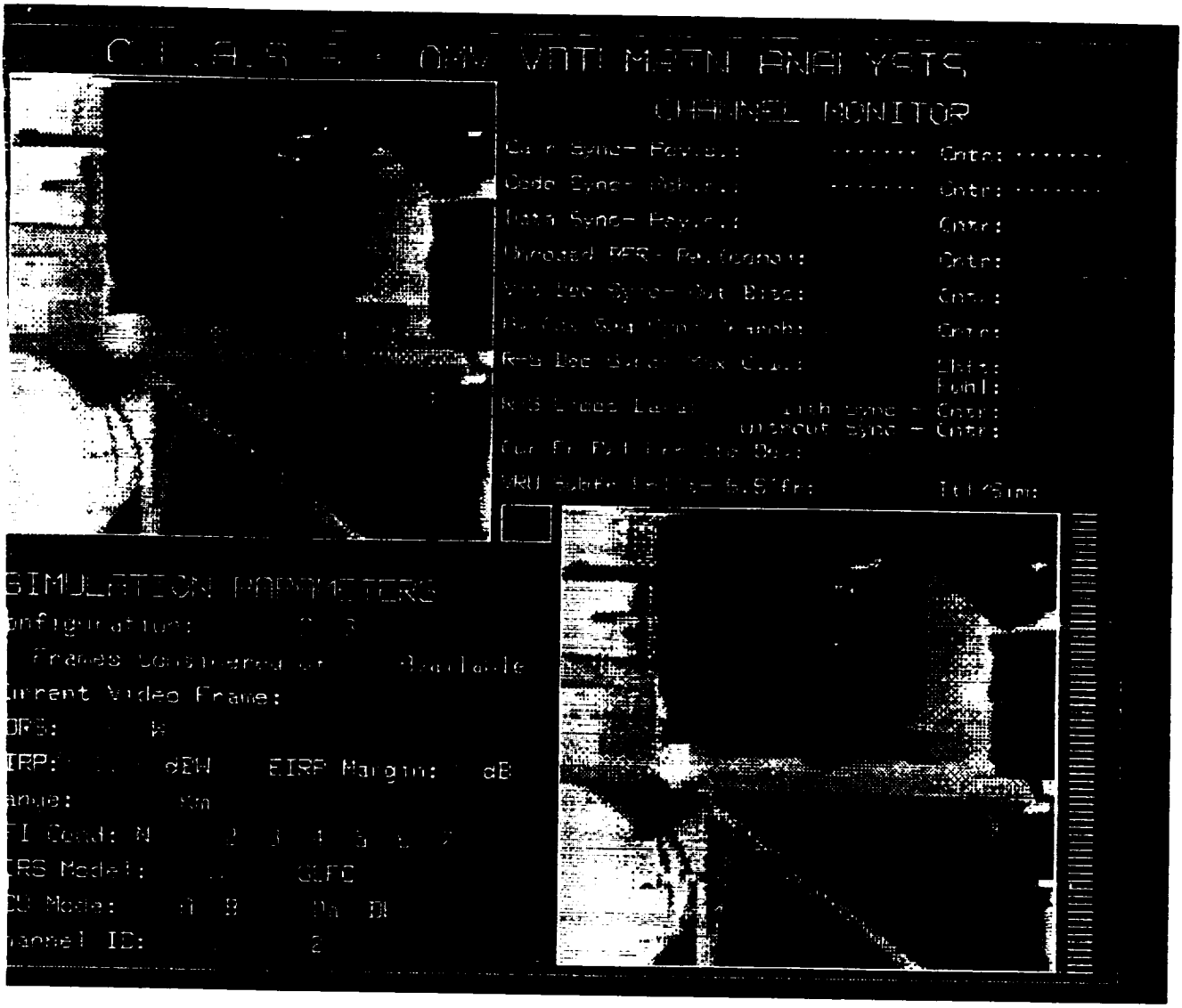
Test No.	OMV Transmit Power	Q Channel Power	OMV EIRP*	EIRP Margin (TDRS)	Effective EIRP Margin (TDRS)**	RFI Adjusted EIRP Margin***
1	15 W	12 W	28.9 dbw	4.0 db	1.7 db	-1.0 db
2	20 W	16 W	31.1 dbw	6.2 db	3.9 db	1.2 db
3	25 W	20 W	32.1 dbw	7.2 db	4.9 db	2.2 db
Recommended test at these levels:				1.0 db	-1.3 db	-4.0 db
				0.0 db	-2.3 db	-5.0 db
				-4.9 db	-6.0 db****	-8.7 db

* OMV EIRP does not include: user constraint losses, dynamic motion losses, polarization losses, plume losses or RFI effects.

** Effective EIRP Margin includes: .3 db polarization losses and 2.0 db other combined losses.

*** RFI adjusted EIRP Margin is effective EIRP Margin plus an additional 2.7 db of losses due to RFI effects.

**** Effective EIRP Margin includes: .3 db polarization losses and .3 antenna pointing loss and .5 user constraint losses.



ORIGINAL PAGE
COLOR PHOTOGRAPH

FIGURE 3 - A SIMULATION RESULT FOR HIGH LEVEL RFI

CONSIDERING THE LACK OF DYNAMIC MOTION LOSSES AND OTHER FACTORS NOT INCLUDED IN PHASE 1.0, IT IS RECOMMENDED THAT A TEST WITH HIGH RFI (LEVEL 1) WITH -3.0 EIRP MARGIN (TDRS), HENCE AN EFFECTIVE EIRP MARGIN (TDRS) OF -5.3 dB (ADJUSTED FOR POLARIZATION AND USER LOSSES OF .3 dB AND 2.0 dB, RESPECTIVELY) AND AN RFI ADJUSTED EIRP MARGIN OF -8.0 dB BE CONDUCTED. THIS WOULD BE 3.0 dB WORSE THAN THE SIMULATION WHICH PRODUCED FIGURE 3.

SOFTWARE ANOMALIES UNCOVERED (OCT/NOV 1988 - APRIL 1989)

At an October/November 1988 meeting in Huntsville, Alabama, a consortium of OMV personnel gathered to review the upcoming (Dec 88) OMV CLASS test. Since that meeting, many anomalies of the software/CLASS test development effort have been uncovered. The actual OMV CLASS test is being (will be) initiated March 26-31, 1989. A partial list of problem anomalies discovered by NASA/MSFC, NASA/GSFC, Fairchild and TRW is given below. These problems would not have been discovered until well into the hardware development had the OMV CLASS test not been insisted upon and insisted upon to the fidelity required by NASA/MSFC personnel. At the close of the October/November 1988 meeting, it was felt that the OMV CLASS test could be held December 1989 and that the software was complete by Fairchild and all software conversion could be expected to proceed (finally) by a two to three week date. Since that date, a series of telecons were held and an in-person meeting conducted in December, 1988, at NASA/MSFC with Dr. Frank Ingels, Mrs. Lee Ann Thomas and Mr. Glenn Parker.

Telecons and/or telephone discussions were held on the following dates:

T1. 4 January 1989
T2. 5 January 1989
T3. 12 January 1989
T4. 19 January 1989
T5. 1 February 1989
T6. 14 February 1989
T7. 1 March 1989
T8. 7 March 1989
T9. 14 March 1989
T10. 20 March 1989

The following items were brought to light in pursuit of a high fidelity OMV CLASS test:

1. Lack of correct interleaving/deinterleaving when using two different video scenes. Different scene usage insisted on by NASA/MSFC/MSU.
2. Use of and thus test of 4 and 10 line subframe replacement. This feature was insisted on by NASA/MSFC/MSU. This feature did not function properly when tried the first time! It was inadvertently commented out.
3. Use of R/S decoder erroneous decoding (invalid data) flag was not incorporated in Fairchild software! This feature insisted on by NASA/MSFC/MSU. Decided it would be incorporated in hardware in May or July, 1990.
4. It was discovered that Huffman tables in the software were not correct! Evidently cross-loaded in software!
5. New buffer/step size control algorithm was uncovered! OMV CLASS test will not utilize this feature but will utilize the old feature.
6. When the ten line subframe replacement feature was successfully integrated, it was found the four line subframe replacement feature was not acting properly.
7. Use of two different data rates (48 Kbps and 972 Kbps) were not integrated at first. This was accomplished later.
8. VRU apparently did not contain channel ID check. The assumption was that every other code word is in every other camera! This created a problem of sync lost! This was fixed.
9. Only after use of very different scenes did. When VCU.1 and VCU.2 produced different bit counts per frame, it was discovered that the position of both ends of residual bits were not being properly identified and tagged! This was fixed.
10. Bit rate controller was not being kept operational for all two channel modes of operation! This was fixed.
11. SSA end-to-end calibration BER performance was reviewed by R. Godfrey in a memo to NASA/MSFC/MSU. This gave us more confidence in the CLASS calibration than I felt before.

12. An unknown VCU/VRU software problem (as of 27 March 1989).
13. Throughout the postponement of OMV CLASS test from December, 1988, to mid-January, 1989 to mid-February, 1989 to mid-March, 1989, the NASA/MSFC/MSU contingent successfully withstood all requests to go to configurations 2 and 3 testing after very few (if any) configuration 1 tests. We keep in mind the following:
    1. Only in configuration 1 can we judge resynchronization time after bit slip.
    2. Only configuration 1 uses the RF link equipment and the Viterbi decoder and the PCI decoder.
14. It was also discovered CLASS Viterbi software was not originally written in sufficient detail to allow synchronization and resynchronization studies on a bit-by-bit basis.
15. It was discovered that in a hurry to perform a 10 frame test run for CLASS school demonstration the CLASS software concerning the R/S decoder was modified but not revalidated using the test vectors. This illustrates the need for careful validation of each software setup!

ADDENDUM 2

Report: Meeting at GSFC, 17 April 1989

A visit was made to GSFC on Monday, 17 April 1989, to inspect an initial few configuration 1 data runs for CLASS simulation testing of the OMV system. Mr. R. Godfrey kicked off the meeting, answered some questions I had prestored and let Mr. Ted Kapplan (301+464-8900) and Mr. Dave Wampler (301+286-6767) guide me through the maze of detailed statistics for a configuration 1 data run. The statistics that will be available for a configuration 1 run are tabulated in TABLE 1. Figure 1 depicts the system points at which data will be taken in a configuration 1 run. Figure 2 is a condensed summary of the error statistics for the 50-frame runs with 0 db and -2 db EIRP margin at TDRS (relative to no RFI) respectively.

A set of 10-frame runs were conducted to determine the approximate EIRP margin (TDRS) at which the video reconstruction would deteriorate to an unusable picture. The runs were made for 0 db, -1 db and -2 db. No detailed statistical results were kept. These short runs were made to determine at what EIRP margin the 50-frame runs should be made. A comparison of the error rate out of the Viterbi decoder for the 10- and 50-frame runs is of interest because it can indicate whether 50-frames will be sufficient to prove a statistical sample. These results are indicated in TABLE 3.

The results of TABLE 3 indicate that the error statistics after the Viterbi decoder are roughly the same for the 10-frame and 50-frame runs. However, there are no available statistics on the random/burst mixture, so we cannot make any judgement on 50-frame statistics for random/burst

mixture. A-200 frame run is planned to compare with the 50-frame runs. Perhaps the 10-frame runs could be repeated and the detailed statistics of TABLES 1 and 2 could be recorded and compared to the 50-frame runs to allow judgement of the sufficiency of the statistical sample size of 50 frames.

Also from TABLE 3, it is apparent that the video link is tracked between -1 db and -2 db EIRP margin (TDRS). It is unfortunate that a 50-frame run was not made at a EIRP margin (TDRS) that provided some frames with replacements and some frames with no replacements, probably around -1.5 db EIRP margin (TDRS). I recommend this be done.

TABLE 2 illustrates the system performance for a good working video link (0 db) and a completely trashed video link (-2 db). In fact, at -2 db EIRP margin (TDRS), there were no video frames reconstructed. That this should be the case is logical. The following discussion will explain why we expect the results in TABLE 2 for both the 0 db case and the -2 db case.

First, one must realize that deinterleavers of any type will dispense a long burst of errors into smaller bursts distributed more or less on a periodic basis throughout the deinterleaved symbol stream. Also, a uniformly random distribution of error events into a deinterleaver produces a uniformly random distribution of errors in the output symbol stream and a uniformly random distribution of short bursts input will create an approximate uniform random distribution of small bursts in the output symbol stream. With this in mind, let's look at the data in TABLE 2.

We see that the statistics into the DPCI are a random mixture of random and burst error events. The mean length, $\bar{L}$, of the bursts for both 0 db and -2 db runs are fairly short, of the order of 11 to 18 symbols long. Even with two standard deviations, the burst error lengths are only of the order

of 26 and 44 symbols. After the DPCI, we see the error events are still a rather random mixture of random and burst error events with bursts of mean lengths 12 and 18 symbols and two standard deviation lengths of 30 and 48 symbols, respectively.

The Viterbi decoder with a free distance of 10 across the constraint length of about 64 symbols can correct most error bursts of 4 to 5 symbols in a span of about 64 symbols. The output of the DPCI for the 0 db runs produces a burst event on the average of 1.6 burst events every 100 symbols. These bursts have an average length of 12 symbols and an average of 3 errors per burst. Thus, the Viterbi should be able to correct most of these error events; and it does a good job, reducing the cumulative error rate from 6.478×10^{-2} to 3.335×10^{-4} (a factor of 200 reduction).

However, at -2 db EIRP margin (TDRS), the burst error events out of the DPCI occur at a rate of 2.2 per 100 symbols with an average length of 18 symbols and an average of 4 to 5 errors per burst. Now we see the Viterbi should start having trouble decoding the average burst, especially with an occasional random error included in the 64 symbol constraint length, and it does have trouble reducing the cumulative error rate from 11×10^{-2} to only 2.4×10^{-2} (a factor of 5 reduction).

The error statistics out of the Viterbi indicate a fairly uniform mixture of random and burst error events and after the R/S deinterleaver we expect a similar error makeup (there are no long strings of bursts to deinterleave!).

As a result, the probability of symbol errors in a 2040 bit R/S codeword is conservatively approximated by assuming all error events are simply

random events. Thus, a cumulative error event rate of (6.74×10^{-5} plus 7.48×10^{-5}) 14.225×10^{-5} (0 db margin case) will produce an average of .29 errors per 2040 bits, hence the R/S decoder should correct these, and it does! (No subframe replacements occurred and no R/S codewords failed to decode after initial synchronization was achieved.)

However, for the -2 db margin runs, the Viterbi output error event rate (1.879×10^{-3} plus 3.789×10^{-3}) of 5.668×10^{-3} produces an average of 11.56 error events per 2040 bits or 11.56 errored R/S symbols. Now we expect the R/S decoder to fail to decode the errors, and it does! (All video frames failed to reconstruct.)

The above is a heuristic discussion to provide insight into how the system should work. The system test results for this STATIC case agree with the "gut" feeling. One must remember that the DYNAMIC case will produce non-linear situations and will be the REAL test!

It must be remembered that 0 db EIRP margin at TDRS (relative to no RFI) corresponds to an EFFECTIVE_OMV_EIRP of 24.88 dbw after all losses are accounted for in the OMV budget analysis. We and TRW must be absolutely sure to be fair and objective in accounting for all possible losses.

An interesting observation from the TABLE 2 test results is that a 0 EIRP margin (TDRS) with no RFI should produce 10^{-5} error rate after the Viterbi decoder. In the 10- and 50-frame test, HRFI runs, the Viterbi decoder error rate at 0 EIRP margin (TDRS) was approximately 3×10^{-4}. Thus, the RFI raises the average error rate by a factor of approximately thirty (30).

An estimate of synchronization time was made in my report of 5 April 1989. I estimated for the average Viterbi search and best subframe replacement synchronization case that 140 ms would be the acquisition time. The test runs for 50 frames took 30 R/S codewords to acquire synchronization. This is 68,040 bits. At a bit rate of 486,000 bits per second, 140 ms is 68,040 bits! What a lucky coincidence. In fact, though the first two video frames (one of each channel) are lost due to the need to have a correct frame header for each frame and the fact that channel frames are interleaved.

Another interesting observation is that after initial synchronization, neither the bit sync, Viterbi sync, nor R/S sync were dropped in the -2 db runs. Although "hits" were occurring in the R/S sync words, there was never a danger of losing R/S sync; the video was lost because there were simply too many errors to correct!

TABLE 1

I. Bitsync.dat
   1) mean & standard deviation of symbol interval as a fraction of the symbol time
   2) clock jitter spectrum

II. Quansync.dat & PCI.dat
   1) slip rate = # of slips/# of sym
   2) random error rate
   3) # of bursts
   4) burst characterization
      i) mean & standard deviation of burst duration
      ii) histogram of burst duration
      iii) # of errors/burst (mean & standard deviation)
      iv) spacing between errors in a burst (mean & standard dev.)
   5) 8-ary transition probabilities (Helps to check ops of VA decoder, also to see the type of channel noise)

III. Vsync.dat & CCPCI.dat (ALL OF THESE STATISTICS ARE AFTER DPCI BUT BEFORE VITERBI DECODER)
   1) random error rate (should go up relative to burst errors)
   2) # of bursts (should go down after de-interleaving)
   3) burst characterization - same as II-4

Note: Any two erroneous symbols which have <12 error-free symbols between them are considered to be in the same burst.

IV. Vbit.dat & Helicc.dat (AFTER VA DECODER)
   1) # of bits it takes for initial acquisition
   2) sync statistics table

Out-of-Sync Counter	# of bits it takes to detect sync loss input to DPCI to output of VIT decoder	# of bits it takes to re-acquire sync	# of DPCI commutator shifts
1			
2			
.			
.			
.			

Note: If the commutator shifts a multiple of 30, declare the out-of-sync a false alarm! Have to search for this visually.

3) total number of bursts
   6 data bits equal burst window (after decoding)
4) histogram of number of bursts versus burst length
5) histogram of mean of erroneous bits within a burst
6) histogram of standard deviation of erroneous bits within a burst
7) bit error rate for "in sync" bits static statistic

V. Rssync.dat (Reed/Solomon Coder) (See table in CLASS book at end on R/S sync positions.)

1) Table - Sync Position

Codeword Cntr	Sync Position	Cntr starting at sync pos.	Adjacent Values -- -	Sync Cntr Value	Adjacent Values + ++
counts each code-word	stores the cur-rent sync	counts the no. of codewords while at a sync position		max value 15	

2) Table - Freewheeling

Freewheeling Count	Location (Codeword) of Freewheeling Event	# of Codewords	Lowest Freewheeling Value

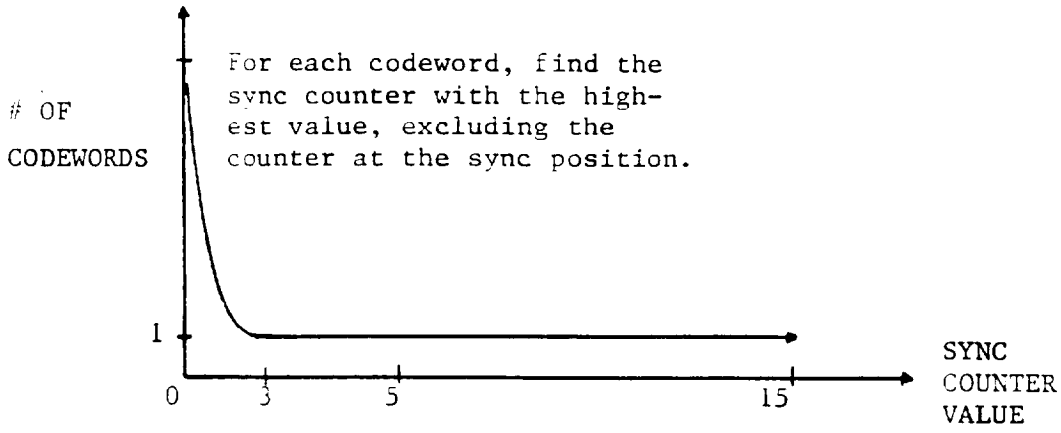
3) Table - Out-of-Sync

Out-of-Sync Count	Out-of-Sync Location	# of Codewords

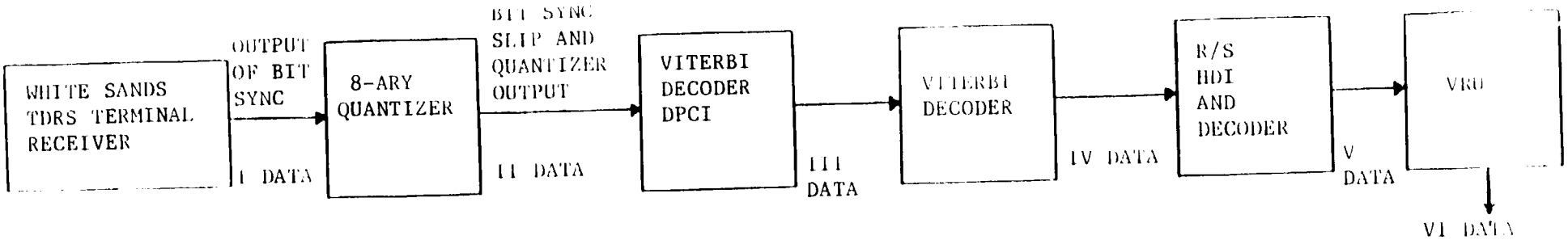
4) # of decodable codewords during in-sync
5) # of decodable codewords during out-of-sync
6) # of undecodable codewords during in-sync
7) # of undecodable codewords during out
8) # of miscorrections
9) miscorrection table

Transmitted Codeword in Hex	Codeword in Hex Before the R/S Decoder	Miscorrected Codeword in Hex	# of Bit Errors in the Mis-correction

10) Histogram of max sync cntr's for each codeword



VI. RSVRV.dat
1) total channel bit error rate



CONCISE SUMMARY

DATA POINT	
I DATA	BIT SYNC OUTPUT CLOCK JITTER SPECTRUM AND SYMBOL (BIT SYNC CLOCK OUTPUT) TIME INTERVAL VARIATION STATISTICS
II DATA	BIT SYNC BIT SLIP STATISTICS, 8-ARY TRANSITION PROBABILITIES, RANDOM AND BURST ERROR STATISTICS PRIOR TO DPCI
III DATA	BURST AND RANDOM ERROR STATISTICS AFTER DPCI, BUT BEFORE VITERBI DECODING
IV DATA	VITERBI DECODED OUTPUT BURST ERROR STATISTICS AND SYNC STATISTICS. NUMBER OF BITS IT TAKES FOR VITERBI/DPCI SYSTEM TO RECOGNIZE SYNC LOSS AND NUMBER OF BITS IT TAKES TO REACQUIRE SYNC
V DATA	R/S DECODER SYNC STATISTICS AND UNDECODABLE, DECODABLE AND MISCORRECTED R/S WORD STATISTICS
VI DATA	TOTAL CHANNEL BIT ERROR STATISTICS

FIGURE 1 - CONCEPTUAL OMV CLASS SIMULATION DATA RECORDING POINTS, CONFIGURATION 1 RUNS

STATISTICS FOR TWO 50 FRAME RUNS 17 APRIL 1989

STATISTIC	BEFORE DPCI (II DATA)		AFTER DPCI (III DATA)		VITERBI DECODER OUTPUT	
EIRP MARGIN	0 db	-2 db	0 db	-2 db	0 db	-2 db
1. Bit Slip Rate	-0-	-0-	-0-	-0-	-0-	-0-
2. Number of Random Errors	112,390	86,646	135,688	76,320		
3. Random Error Rate	1.229×10^{-2}	9.447×10^{-3}	1.484×10^{-2}	8.35×10^{-3}	6.74×10^{-5}	1.879×10^{-3}
4. Number of Burst Errors	126,137	192,000	147,850	202,471		
5. Burst Error Rate	1.379×10^{-2}	2.1×10^{-2}	1.617×10^{-2}	2.21×10^{-2}	7.48×10^{-5}	3.789×10^{-3}
6. Total Number of Errors	592,311	1,010,088	592,104	1,009,729		
7. Cumulative Error Rate	6.478×10^{-2}	.1105	6.478×10^{-2}	.1105	3.335×10^{-4}	2.461×10^{-2}
8. Average Error Burst Length, $\bar{L}$	10.694	16.66	12.109	18.514	5.657	10.28
9. Std. Dev. of $\bar{L}$	8.0824	14.16	9.206	15.983	3.837	9.64
10. Average Errors Per Burst, $\bar{E}/\bar{B}$	3.8047	4.809	3.087	4.61	3.555	5.998
11. Std. Dev. of $\bar{E}/\bar{B}$	2.173	3.342	1.502	3.07	1.938	5.366
12. Average Error Spacing Per Burst, $\bar{E}\bar{S}/\bar{B}$	2.921	3.39	4.316	3.856		
13. Std. Dev. of $\bar{E}\bar{S}/\bar{B}$	3.051	3.06	3.116	3.034		
14. Viterbi In-Sync Error Rate					3.335×10^{-4}	2.46×10^{-2}
15. Number of Bits for Viterbi to Reacquire Sync					No Sync Drop	
16. Number of Video Frames With Subframe Replacements					None	No Video Frames Reconstructed

TABLE 3

COMPARISON OF 10 FRAME AND 50 FRAME RUNS
(CONFIGURATION 1)

No. of Frames/Run	EIRP*: 0 db	-1 db	-2 db
Subframe Replacements:			
10 Frame Run	0	0	ALL
50 Frame Run	0	-	ALL
Bit Error Rate After Viterbi Decoder:			
10 Frame Run	3.09×10^{-4}	2.43×10^{-3}	2.33×10^{-2}
50 Frame Run	3.34×10^{-4}		2.46×10^{-2}
Bit Error Rate After Bit Sync and Matched Filter But Before DPCI:			
50 Frame Run	6.478×10^{-2}		11.04×10^{-2}

25 frames per camera channel, 50 frames total, statistics based on 50 frames. 20 lines per subframe, 12 subframes per frame. approximately 20 R/S codewords per frame.

* EIRP margin at TDRS for no RFI. (The high RFI channel model was used for the runs.)

ATTACHMENT 3

Equivalent System Gain Available From R-S Encoding Versus a Desire to Lower the Power Amplifier From 25 Watts to 20 Watts for OMV, Memo, 27 August 1989

MISSISSIPPI STATE UNIVERSITY
COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING
DRAWER EE
MISSISSIPPI STATE, MISSISSIPPI 39762
PHONE (601) 325-3912

27 August 1989

To: Ms. Lee Ann Thomas

Subject: Memo From Boris Dobrotin, 20 July 1989, Concerning Equivalent System Gain Available From R-S Encoding AND B. Dobrotin's Desire to Lower the Power Amplifier From 25 Watts to 20 Watts

The memo references two reports by Joe Oldenwalder concerning Error Control Coding. These references are the references 1 and 2 at the end of this report. Mr. Dobrotin's memo contains a curve drawn from reference 1 (Figure 5-11, page 124 of reference 1) which I have included herein as Figure 1. Figure 2 of this report depicts the unaltered curves of Figure 1 drawn from my own copy of reference 1. Figure 3 of this report depicts the unaltered curves of Figure 7.4, page 198 of reference 1 for concatenated code block error probability performance for K = 7, rate = 1/2 convolutional inner codes and various R/S, 8 bit/symbol, codes for the outer code. Figure 4 of this report depicts the bit error probability performance for K = 7, rate = 1/2 convolutional code inner codes and various R/S, 8 bit/symbol, codes for the outer code.

Inspection of the curves of Figure 1 and Figure 3 of this report and noting the circled points of Figure 3, we see Mr. Dobrotin's added curve for the R/S concatenated 8 bits per symbol, 8 symbol correcting outer code has been transposed correctly with 2.8 dB E_b/N_o required to achieve Bit Error Probability of 10^{-5} for the concatenated system and approximately 4.4 dB E_b/N_o required for the Viterbi soft decision Rate 1/2 K = 7 coding along. Thus the conclusion by Mr. Dobrotin that the concatenation of the

R/S code on the convolutional code does result in a Bit Error Probability of about 10^{-5} for an apparent reduction in E_b/N_o of 4.4 dB - 2.8 dB = 1.6 dB.

(One might be interested in knowing that the same curve as Figure 3 of this report (Figure 7.3, page 197 of Reference 1) appears in many literature sources and, in particular, occurs in Figure 17.3, page 536 of Reference 3).

So, Mr. Dobrotin's conclusion is verified **if** all synchronization has been achieved. **A VERY BIG IF!** The carrier recovery loop and the bit synchronizer and the Viterbi decoder work with raw E_b/N_o levels. By lowering the Power Amplifier from 25 watts to 20 watts (a .96910013 or approximately a 1.0 dB loss), approximately 1 dB loss in raw E_b/N_o is incurred at the carrier recovery loop and bit synchronizer input and the error rate of the BPSK demodulated data to the Viterbi decoder will now correspond to a 1 dB lower E_b/N_o BPSK signal. The bit error rate into the Viterbi is about 5.10^{-2} (based on 5 dB E_b/N_o input approximately), and a 1 dB reduction yields a bit error rate of about 8.10^{-1} for the BPSK signal, NOT INCLUDING POSSIBLE SYNC LOSS AT CARRIER RECOVERY LOOP OR AT BIT SYNC OR VITERBI NODE SYNC LOSS.

Table 5.3, page 135, of Reference 1 depicts Average Error Burst Length in Bits and Average Number of Errors per Burst for the K = 7 Rate 1/2 system with soft decision. Inspecting the Table (Table 1 of this report) we see a reduction of 1 dB in E_b/N_o from 4 dB to 3 dB would cause a burst length increase of only 1.4 bits on the average from 6.2 bits to 7.6 bits. However, a decrease of 1 dB from 3.0 dB to 2.0 dB in E_b/N_o would cause an average burst length increase of 3.3 bits from 7.6 to 10.9 bits.

Now suppose we have a 4.4 dB E_b/N_o BPSK input signal and a 10^{-5} bit error rate out of the Viterbi decoder. With a 1.5 dB apparent degradation due to RFI, we would have about 2.9 dB BPSK input to Viterbi and an average burst length of about 8 bits out of Viterbi. **HOWEVER, with Mr. Dobrotin's suggestion we would have a (no RFI) 3.4 dB E_b/N_o BPSK input signal to the Viterbi with a (no RFI) 2.10^{-4} bit error rate output from**

Viterbi and approximately a (no RFI) 7 bit average burst length out of Viterbi. Again, with a 1.5 dB apparent degradation due to RFI, we would have about 1.9 dB BPSK input to Viterbi and an average burst length of 11 bits out of Viterbi and an average bit error rate of 10^{-2}. At this point, we could expect the system to fall apart due to Viterbi Node Sync Loss or Carrier recovery Loop Sync Loss or Bit Sync Loss!! Most likely the Viterbi Node Sync would slip. Testimony to this is a paper by Liu and Lee in 1984 (reference 4) that investigated concatenation of R/S and Convolutional codes for space imagery. They point out that the Viterbi decoder exhibits frequent Node Pair resynchronization (about 100 bits duration, hence a burst of about 100 bits will be emitted during Node Pair resynchronization) if the Viterbi input E_b/N_o ratio is about 2.5 dB or less. Thus reducing the Power Amplifier by 1.0 dB as noted above could cause "mucho" problems during RFI.

I CANNOT AGREE TO REDUCING POWER AMPLIFIER OUTPUT FROM 25 WATTS TO 20 WATTS.

FURTHERMORE, AS I HAVE STRESSED IN THE IMMEDIATE PAST, THE CURRENT CLASS TESTS DO NOT INVOLVE DYNAMICS, ARE ONLY SIMULATIONS NOT EXPERIMENTAL TEST RESULTS OF HARDWARE AND SHOULD NOT BE TREATED AS GOD-GIVEN!

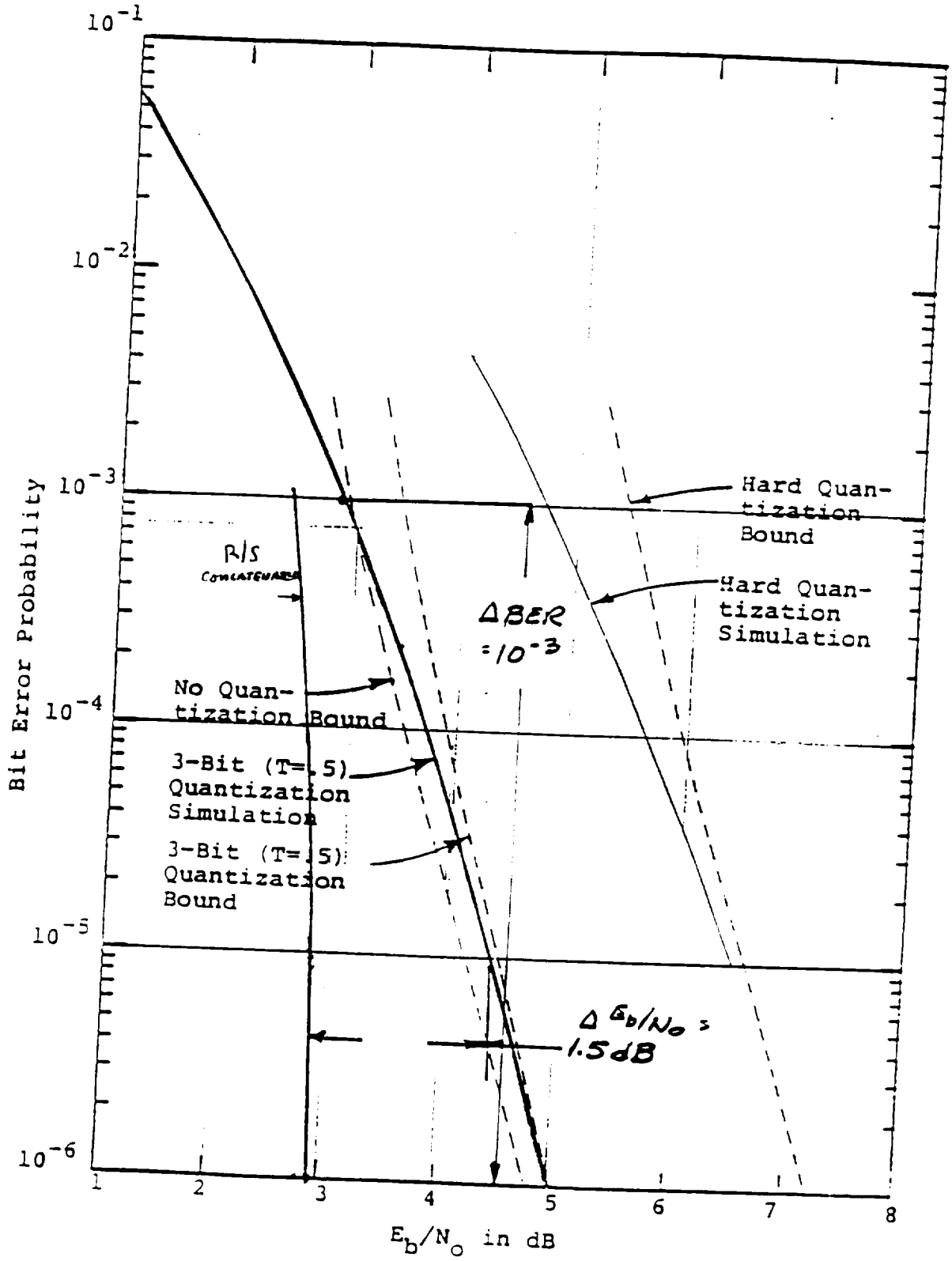


Figure 5.11 Bit error probability versus E_b/N_o performance of a K=7, R= 1/2 convolutional coding system with BPSK modulation and an AWGN channel.

CURVE FROM B. Dobrushin's Memo.

FIGURE 1

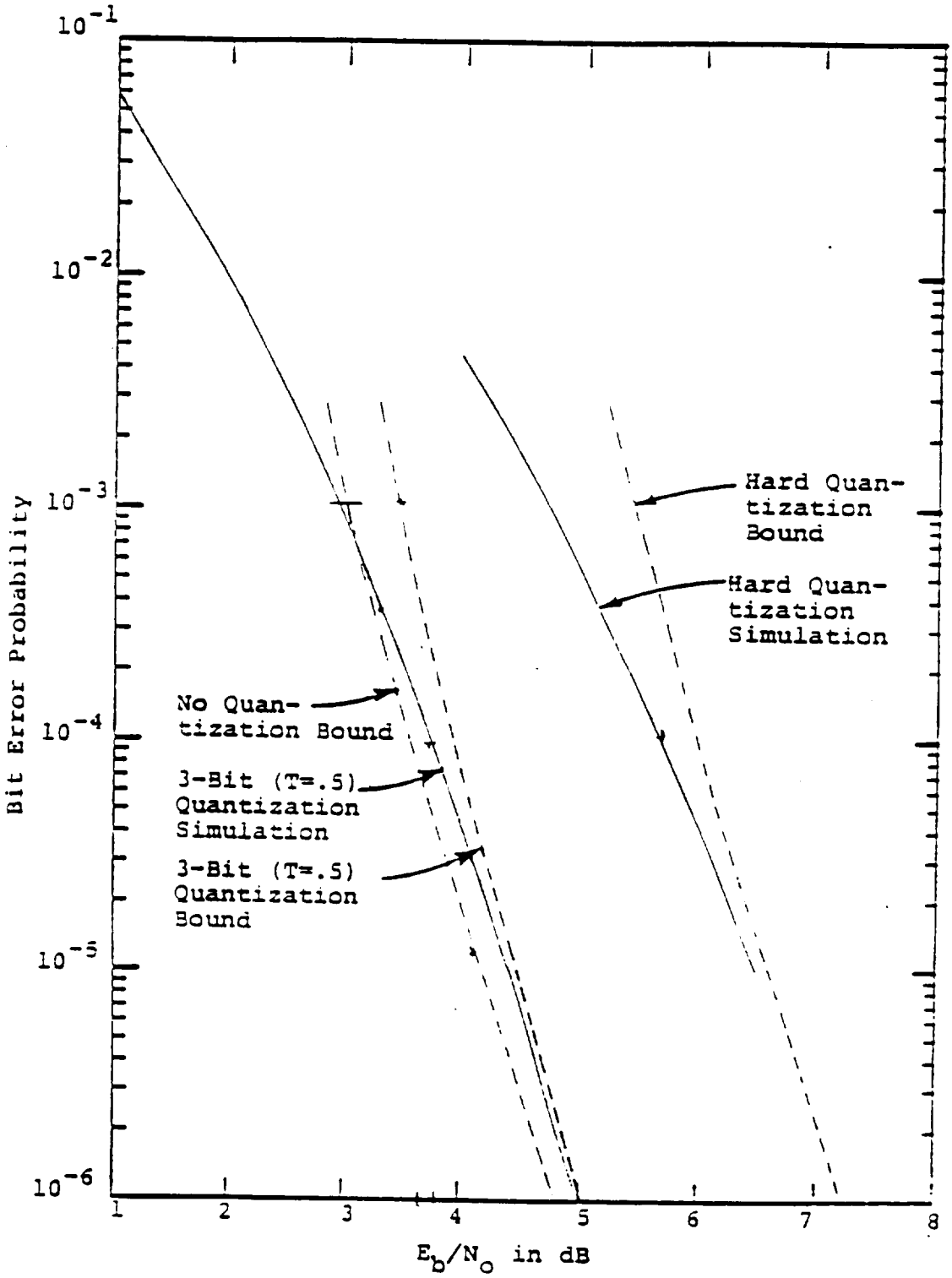


Figure 5.11 Bit error probability versus E_b/N_o performance of a K=7, R= 1/2 convolutional coding system with BPSK modulation and an AWGN channel.

ORIGINAL CURVES UNALTERED

FIGURE 2

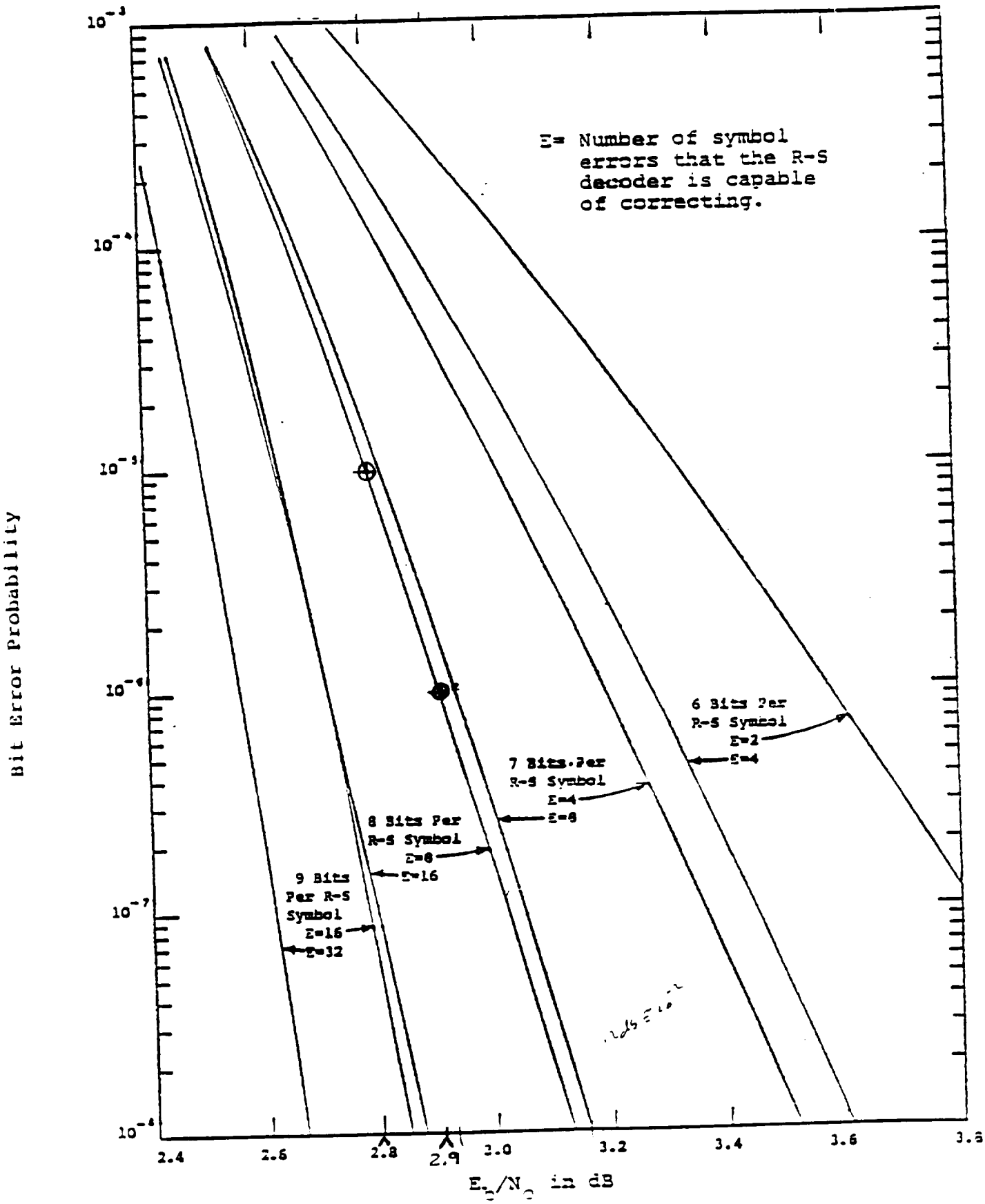


Figure 7.3 Summary of concatenated coding bit error probability performance with a K=7, R=1/2 convolutional inner code and various R-S outer codes.

FIGURE 3

E_b/N_o	Average Error Burst Length in Bits	Average Number of Errors per Burst
1.0	17.3	12.1
2.0	10.9	5.9
3.0	7.6	4.3
4.0	6.2	3.8

Table 5.3 Error burst statistics for K=7 R= 1/2 system with 3-bit quantization.

From Ref. 1 page 135

TABLE 1

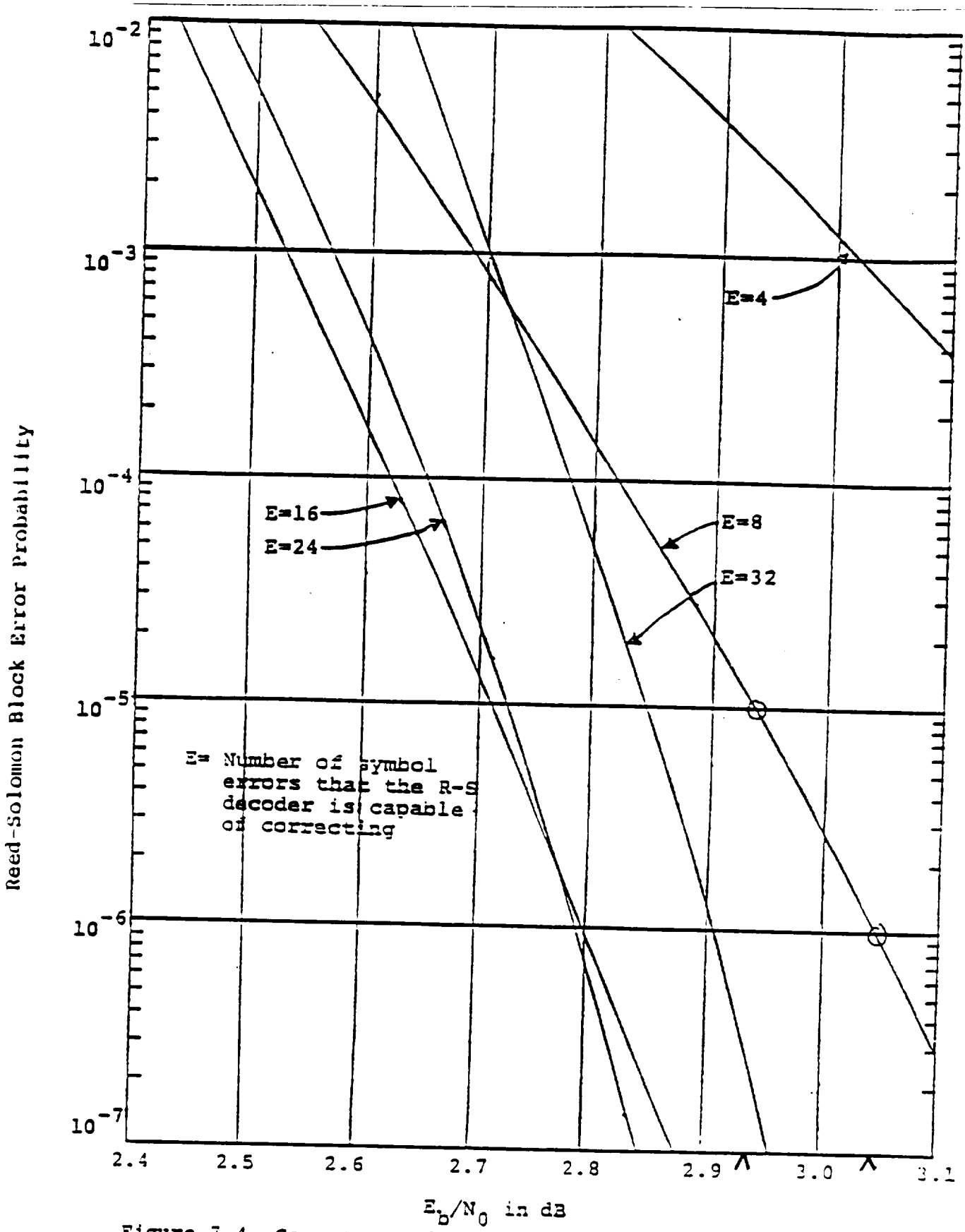


Figure 7.4 Concatenated code block error probability performance with a K=7, R=1/2 convolutional inner code and an 8 bit/symbol R-S outer code

FIGURE 4

REFERENCES

1. Error Control Coding Handbook; Final Report, Oldenwalder J.P., 15 July 1976; Linkabit Corp.; San Diego, California.

2. Hybrid Coding Systems Study, Final Report, Oldenwalder J.P., September 1972; Linkabit Corp., San Diego, California.

3. Error Control Coding: Fundamentals and Applications, Shu Lin and Daniel Costello, Prentice Hall Series in Computer Applications in Electrical Engineering, 1983, ISBN-0-13-283796-X.

4. Recent Results on the Use of Concatenated Reed-Solomon/Viterbi Channel Coding and Data Compression For Space Communication," Liu and Lee, IEEE-COM-32, No. 5, May 1984, page 518.

ATTACHMENT 4

Command Word Acceptance/Rejection Rates for OMV, Memo Included in 12 November 1989 Monthly Report

AN INVESTIGATION OF ERROR CORRECTING TECHNIQUES FOR OMV DIGITIZED DATA

NASA GRANT: NAG8-104
QUARTERLY REPORT
NOVEMBER 12, 1989

SUBMITTED TO:

Mrs. Lee Ann Thomas
EB33
NASA/MSFC
MSFC, AL 35812
205+544-3662

SUBMITTED BY:

Frank Ingels, Ph.D.
Principal Investigator
Mississippi State University
Mississippi State, MS 39762
601+325-3912

WORK STATEMENT

During the quarter from August 12, 1989, through November 12, 1989, a series of activities concerning OMV Video CLASS simulation tests were conducted. A brief discussion of three of these activities is summarized in this report. More detailed memos, etc., that were transmitted during this period are not included.

The contract expiration date is May 11, 1990.

On 15 August, a summary of comments on the Video End-to-End Test Plan was forwarded to Mrs. H. Thomas. On 27 August, a comment paper on the Equivalent System Gain Available From R-S Encoding and A Desire to Lower the Power Amplifier From 25 Watts to 20 Watts was forwarded to Mrs. Thomas. During September and October, a series of discussions concerning the uplink command word acceptance and rejection probability were conducted. A paper summarizing the results of these discussions was forwarded to Mrs. H. Thomas. These three papers are attached to this report.

16 Aug 89

To: L. A. Thomas

Comments on Video End-To-End Test Plan (Coordination Copy) July 31, 1989. H. Haugen to E. B. Stewart.

1. On front page, it is stated that "it is assumed that the Video Test would be performed at GSFC as a follow-on to the Antenna Switching Test".

   If it does not work out to be this way, what impact does the have on cost, re-scheduling of equipment, etc.? Is there a list of equipment used during Antenna Switching Test that is necessary for the Video End-To-End test that would be dispersed after Antenna Switching Test? What if Antenna Test delayed and Video Test must come first? Impact on cost and scheduling? See item 3 of this memo.

2. Use of breadboard CU is okay **if** all interface commands, docks, timing, etc., are identical to the flight design hardware. Does breadboard CU have NRZ-L to NRZ-M conversion? 972 Kbps data rate?

3. How will RFI be simulated? This is in hardware, correct? How will dynamic antenna switching be simulated or accomplished?

   Note it seems from page 3, 2nd paragraph, that all the Antenna Switching Test Set will be necessary to fulfill the test objectives.

4. Correlation of Link test results will CLASS Simulations is not going to be easy. Making link parameters match any simulation will be interesting.

5. Measure bit error rate before and after Viterbi decoder as well. (See page 3, last paragraph.)

6. Top page 4 is important -- recording of the Input and Output Video. Timing codes to be recorded on video frames so we can correlate one for one input and output video frames with error statistics of that frame? Need timing codes for error statistics that are recorded as well.

7. Figure 1, page 5, does not include RFI insertion in Block 14. Is this a Freudian admission to the possibility we won't simulate injected RFI?

8. On Figure 1, page 5, no data is to be recorded after WSGT but before NASCOM Link (between block 15 and 16). HST End-To-End tests have exhibited quite a few outages (1 NASCOM block per 500,000 blocks which is an error rate less than 10^{-6}) over the NASCOM Link. We must record data at WSGT terminal and keep the data so we can determine if outages or errors occur on NASCOM or TDRS section. See also Block 15, page 8 of Test Plan.

9. Item 7, page 6, implies equipment to be used that is not the same design as the flight design.

10. Proposed RFI emulation as per pages 10, 11 are easier to implement than in front of Viterbi, but injecting RFI in front of Viterbi preferable. I do think this would be valuable, however.

11. Page 19, Pass/Fail Criteria. I would state that in addition to 75% or more of the subframes are being updated per frame that no more than X frames occur between updates of any specific subframe. (X to be specified by MSFC, probably X=2.)

4 Nov 89

To: Lee Ann Thomas

PREFACE

The material received by me by FAX 10/31/89 from Howard Haugen to Lee Ann Thomas (dated October 23, 1989) calculates the acceptance and rejection probabilities as:

2.A	Valid Command Rejection Probability	(4.8×10^{-4}	spec is 10^{-3})	OK
2.b.1	Invalid Command Acceptance Probability Assuming in-synch commands with channel error rate 10^{-5}	(1.9×10^{-15} worst case estimate!	spec is 10^{-9})	OK
2.b.2	Invalid Command Acceptance Probability Assuming out of sync commands with random error rate (.5)	(5.96×10^{-8} with 8 bit pre-sync and 9 fixed bits in 48 bit word.	spec is 10^{-9})	FAILS

A memo from F. Ingels to Lee Ann Thomas of 2 October 89, part 1 a) page 3 agrees with 2.A above.

A memo from F. Ingels to Lee Ann Thomas of 2 October 89, part 1 b) page 4 agrees with 2.b.1 above.

The attached 6 pages calculates 2.b.2 above from a different perspective and reaches the same conclusion as 2.b.2 from Mr. Haugen's memo of 23 Oct 89.

F.I.

OMV COMMAND WORD ACCEPTANCE/REJECTION RATES

Estimation of the acceptance and rejection rates for the OMV command word up link is made for several possible operational situations.

1. **NORMAL UPLINK CONDITIONS:** 48-bit sync assumed as are contiguous 48-bit command words in a digital stream assumed as opposed to random 48-bit patterns.

The probability of errors corrupting a command word and these errors being detected by the triple error detecting, shortened, non-cyclic Hamming Code (48,41) thus resulting in a rejection of the command word is termed the Probability of Rejection of a Command Word, P(RCWD).

The probability of errors corrupting a command word and these errors <u>not</u> being detected by the polycode and thus resulting in false acceptance of a bad command word is termed the Probability of False Acceptance of a Command Word, P(FCWD).

A. P(RCWD) ~ Prob. of 1, 2 or 3 errors (although some larger number of error patterns are detectable, they will not be considered here. Thus, the analysis is conservative towards P(FCWD) plus the probability of a bad spacecraft address (P($\overline{SCA}$) plus the probability of the two fixed bits being in error plus additive combinations.

$$P(1,2,3 \text{ errors}) = \sum_{n=1}^{3} \binom{48}{n} p^{n}(1-p)^{48-n} \quad .$$

The parameter p is the estimated channel error rate.

$$P(1,2,3 \text{ errors}) = p(1-p)^{45}[48(1-p)^{2} + 1128p(1-p) + 17296p^{2}]$$

p	p(1,2,3 errors) ~ P(RCWD)
10^{-5}	$4.79887217 \ .10^{-4} \sim 48 \times 10^{-5}$
10^{-6}	$4.79988720 \ .10^{-4} \sim 4.8 \times 10^{-5}$

Thus for 10^{-5} channel bit error rate, we see a rejection rate for command words is approximately one word rejected for every 2083 command words. The up link rate is approximately 20 words per second, so we should experience a rejection every 104 seconds or so. For a 10^{-6} channel bit error rate, we see about one rejection every 20030 command words or about one every 1040 seconds due to RCWD. As will be shown, this is the dominate error case for rejection of commands.

The probability of a bad spacecraft address is approximately the probability of one or more of the 7 field bits in the 8-bit address having an error.

$$\overline{P(SCA)} \sim 1 - P(0 \text{ errors in 7 bits}) = 1 - (1-p)^{7}$$

p	$\overline{P(SCA)}$
10^{-5}	$6.99979 \times 10^{-5} \sim 7.10^{-5}$
10^{-6}	$6.9999 \times 10^{-6} \sim 7.10^{-6}$

The probability that the two fixed bits are in error is approximately

$$\overline{P(FB)} = 1 - P(0 \text{ errors}) = 1 - (1-p)^2$$

p	$\overline{P(FB)}$
10^{-5}	$\sim 2 \times 10^{-5}$
10^{-6}	$\sim 2 \times 10^{-6}$

We see the P(RCWD) is dominated by the probability of 1, 2 or 3 errors and is approximately 48×10^{-5} for a channel bit error rate of 10^{-5}.

B. The probability of false acceptance of a command word, P(FCWD) is approximately the probability of 4 or more errors in the 48-bit pattern. (This is actually an upper bound.)

$$P(FCWD) \sim \sum_{n=4}^{48} \binom{48}{n} p^n (1-p)^{48-n} .$$

p	P(FCWD)	
10^{-5}	~ 1.95×10^{-15}	(using 1st two terms)
10^{-6}	~ 1.95×10^{-19}	(using 1st two terms)

Thus with a normal operating up link, we anticipate no significant False acceptance of command words.

2. **NORMAL UPLINK EXCEPT IT IS ASSUMED THAT THERE IS NO 48-BIT WORD SYNC.** Furthermore, it is assumed that a sliding window correlator is used and 64 successive attempts are made to find an acceptable command word.

In this situation, we have two valid words (back to back) (possibly with some errors) each partially in the 48-bit correlation window.
The acceptance demands a proper spacecraft address at the proper location (right hand 7 bits) and a division of the 48 bits by the Hamming poly code with remainder of zero.

To illustrate the poly code division and the potential problem, consider the Hamming 7, 4 SEC code. The primitive generator polynomial is $g(x) = 1 + X + X^3$. By multiplying by 1+X, we obtain the 7, 3 SEC-DED minimum distance 4 code polynomial $p(x) = (1+X)\ g(X) = 1 + X^2 + X^3 + X^4$.

There are 8 code words in this code with a minimum of 4 bit changes between code words. They are:

$$v_i(x) = \mathrm{Re}\left[\frac{X^{n-k} m_i(x)}{p(x)}\right] + X^{n-k} M_i(x)$$

$M_i(X)$	$v_i(x)$
0 0 0	0 0 0 0 0 0 0
1 0 0	1 0 1 1 1 0 0
0 1 0	1 1 1 0 0 1 0
0 0 1	0 1 1 1 0 0 1
1 1 0	0 1 0 1 1 1 0
1 0 1	1 1 0 0 1 0 1
0 1 1	1 0 0 1 0 1 1
1 1 1	0 0 1 0 1 1 1

$$x^4 + x^3 + x^2 + 1 \;\overline{)\,x^4}$$

$$\mathrm{Re} = \frac{x^4 + x^3 + x^2 + 1}{x^3 + x^2 + 1}$$

Being a Linear Cyclic code, any two code words added together is also a code word. We see the minimum distance is 4.

To see if any two adjacent (serial adjacent) words shifted could look like a valid code word and hence be successfully divided by the generator polynomial p(x). Consider the following 7 bit sliding window correlator:

$$\underset{\downarrow \leftarrow \text{ window } \rightarrow \downarrow}{1\ 0\ 1\ 1\ 1\ 0\ 0\quad 1\ 1\ 1\ 0\ 0\ 1\ 0}$$

As the above pattern is shifted to right once we have in the window the pattern 0 1 1 1 0 0 1 which is a valid code word! Such agreements are to be expected since the 7, 3, code is cyclic (note cyclic shifts of the

code words in the above table do produce other code words. For example, 1 0 1 1 1 0 0 → 0 1 0 1 1 1 0 → 0 0 1 0 1 1 1 → 1 0 0 1 0 1 1, etc.).

Now, let's shorten the 7,3 code to the 6,2 code which is a SEC DED minimum distance 4 code consisting of those code words from the 7,3 code which ave as the highest message bit a zero.

$M_i(X)$	$V_i(X)$
0 0	0 0 0 0 0 0
1 0	1 0 1 1 1 0
0 1	1 1 1 0 0 1
1 1	0 1 0 1 1 1

This is a linear non-cyclic code. (Note 0 1 0 1 1 1 → 1 0 1 0 1 1, which is not a code word)

Again, consider the sliding correlator.

```
              ↓  window   ↓
  0 1 0 1 1 1   1 0 1 1 1 0
```

Shifting through we see no false words appear. Six shifts produce the following six patterns in the window, none of which are valid patterns:

1 1 0 1 1 1, 1 1 1 0 1 1, 1 1 1 1 0 1, 0 1 1 1 1 0, 1 0 1 1 1 1

However, the following two words yield a false pattern after a shift:

```
                ↓   window    ↓
  1 0 1 1 1 0   1 0 1 1 1 0
```

After one shift, the pattern in the window is 0 1 0 1 1 1 which is a valid code word and would yield a false acceptance! For this 6,2 code, some of the possibilities are:

```
                ↓   window    ↓
  0 1 0 1 1 1   1 0 1 1 1 0     no shifts yield false patterns

  1 1 1 0 0 1 ↓ 0 1 0 1 1 1 ↓   no shifts yield false patterns

  0 0 0 0 0 0 ↓ 1 0 1 1 1 0 ↓   1st shift yields 0 1 0 1 1 1 and false
                                acceptance!  No other shifts yield
                                false acceptance.

  1 0 1 1 1 0 ↓ 1 0 1 1 1 0 ↓   1st shift yields 0 1 0 1 1 1 and false
                                acceptance!  No other shifts yield
                                false acceptance.
```

Altogether there are ten serial combinations, each with 5 shifts for a total of 50 shifts. Out of these 50 shifts, two cause a false acceptance.

Although not many shifts of serial words yield false acceptance, at this point I do not have an analytical answer to the question of how many false acceptances might be possible when any two code words are serially shifted through a sliding correlator window.

A worst case assumption for P(FCWD) under the no sync scenario might be best. Mr. Dave Harris' 2^{32} possibilities out of 2^{48} total vectors is definitely a worst case assumption since any of the possible randomly selected bits patterns are assumed to be a problem. (Remember the two fixed bits and the 7 fixed bits of the spacecraft address are not open for random selection.) The probability of FCWD for this case is 1.528×10^{-5}.

$$P(FCWD) = 1.528 \times 10^{-5}$$

For 64 looks the result should be bounded by

$$P(FCWD) \sim 10^{-3}$$

(Note for the 6,2 code, 2 out of 50 cause a problem. There are $2^2 = 4$ valid possible random patterns and $2^6 = 64$ possible patterns. Thus, we see $2^2/2^6 = .0625$ is a bound over $2/50 = .04$. Thus, the above number is conservative by probably two orders of magnitude for the 48,41 code truncated by 15 bits from the 63,56 code.)

3. ABNORMAL UPLINK, NO 48-BIT SYNC AND RANDOM BITS INPUT TO THE CORRELATOR

UNSYNCHRONIZED CORRELATOR SEARCH FOR 48 BIT WORDS

What is the probability of false acceptance of an error detecting/correcting coded word with random bit probabilities (that is not synchronized to word boundaries and hence the probability each bit equals 1 or 0 is 0.5) and a subset of fixed bits? To answer this question, it suffices to start with a small example. Consider the 7,4 single error correcting code. There are 4 information bits and 3 parity bits.

With 0 fixed bits, there are 2^4=16 possible valid words. There are 2^7=128 total 7 bit words, and hence the probability of a false acceptance of a word is tantamount to a random pattern of 7 bits looking like one of the 16 valid patterns out of 128 possible patterns.

$$P(FWA) = \frac{2^4}{2^7} = \frac{16}{128} = \frac{1}{2^3} = .125 \quad .$$

This presumes the receiver/decoder checks the complete 7 bit pattern in toto. Another possible receiver approach is to re-encode the 4 bit information pattern and then compare the resulting 3 parity bits against those received. The probability of a false acceptance of a word in this case amounts to the probability of the 3 parity bits resulting from re-encoding equalling the specific 3 parity bit pattern received. Thus,

$$P(FWA) = \frac{1}{2^3} = .125 \quad .$$

We see the receivers perform the same.

Now consider fixing one of the bits of the word, as might be the case for creating a synchronization ID code within a word which contains an error checking polynomial parity check set. After fixing one bit, inspection of the 7,4 binary code illustrated in Table 1.0 will reveal that there are 8 rather than 16 possible code words. (Note fixing one bit is restrained to the information bits since the parity check bits cannot be constrained and still preserve the error correcting/detecting capability of the code.)

One should also note that the parity check codes are repeated in the full set of 16 words, but that for each bit position of the information bits if that bit is a 1, there are 8 unique parity checks and if that bit is 0, the same 8 unique parity checks occur again. Thus, fixing 1 bit leaves 8 possible words each with a unique parity check set.

Finally, we may observe that the probability of a random pattern of bits being accepted as a valid word now amounts to one of 8 possible patterns out of the 128 possible patterns. Conversely, one might consider the probability of the fixed bit being emulated at random times the probability of the 8 remaining patterns occurring out of $2^6=64$ possible patterns (the fixed bit being removed from the 7 bit word). The result is the same:

$$P(FWA) = \frac{1}{2^1} \cdot \frac{2^{k-1}}{2^{N-1}} = \frac{2^{k-1}}{2^N} = \frac{2^3}{2^7} = .0625 \quad .$$

TABLE 1.0
7,4 BINARY SINGLE ERROR CORRECTING CODE

K = 4 Information Bits
N = 7 Total Bits Per Word
N - K = 3 Parity Check Bits

K	N-K
0 0 0 0	0 0 0
1 0 0 0	1 0 1
1 1 0 0	0 1 0
1 1 1 0	1 0 0
1 1 1 1	1 1 1
0 0 0 1	0 1 1
0 0 1 1	1 0 1
0 1 1 1	0 1 0
1 0 1 1	0 0 0
1 1 0 1	0 0 1
1 0 0 1	1 1 0
0 1 1 0	0 0 1
0 1 0 0	1 1 1
0 0 1 0	1 1 0
0 1 0 1	1 0 0
1 0 1 0	0 1 1

Similarly, for X fixed bits, we have

$$P(FWA) = \frac{1}{2^X} \frac{2^{K-X}}{2^{N-X}} = \frac{2^{K-X}}{2^N} \quad ; \qquad X \leq K \ .$$

Thus, if we fix all K = 4 information bits, there is one acceptable valid pattern out of the 2^N possible patterns and

$$P(FWA) = \frac{1}{2^N} \qquad X = K \ .$$

What of the receiver that calculates parity checks using the received bits of information and then comparing them to the received parity bits when we fix X information bits? The answer lies in the realization that only one of the possible 2^X bit patterns combined with only one of the possible 2^{K-X} bit patterns will produce the required parity checks. From Table 1.0 consider the patterns with the first two bits fixed at 11. There are four

1 1 0 0	0 1 0
1 1 1 0	1 0 0
1 1 1 1	1 1 1
1 1 0 1	0 0 1

We see that each of the four have different parity checks. The probability a random set of bits will pass both the fixed bit pattern of 11 and simultaneously pass the exact three parity bits that are received

(which, being derived from random bits, is an equal probability of being one of the 2^3=8 possible patterns) is equal to

$$P(FWA) = \frac{1}{2^2} \times \frac{1}{2^3} = \frac{1}{2^5} = \frac{2^{K-X}}{2^N} ,$$

the same as for the receiver that divides all 7 bits by the generator polynomial g(x) and expects all zeros as the answers.

Summarizing

Thus, the probability of false word acceptance for either receiver type for X fixed bits out of K information bits in an N bit word, with no word boundary synchronization (thus, each bit is equally likely to be a 1 or 0) is

$$P(FWA) = \frac{2^{K-X}}{2^N} \qquad X \leq K .$$

For a (48,41) Hamming code with only 8 fixed bits of synchronization ID code within the (48,41) code word, the probability of false acceptance of a word when operating with a sliding correlator and no synchronization and no code word boundaries known is (on a single look in the correlator)

$$P(FWA)_{\substack{\text{Single look}\\ \text{in correlator}\\ \text{(only 8 total fixed bits!)}}} = \frac{2^{41-8}}{2^{49}} \sim 1.526\times10^{-5}$$

For 64 looks with the correlator (each look is independent), the cumulative probability of false command word acceptance would be

$$P(FWA)_{\substack{\text{64 looks}\\ \text{in correlator}}} = 64\, P(FWA)_{\substack{\text{single look}\\ \text{in correlator}}} \sim 9.76\times10^{-4} \sim 10^{-3} .$$

If a γ bit preface code (such as a 7 bit Barker) is placed in front of the 48 bit word, then the probability of false acceptance is the combination of false acceptance of the 48 bit word <u>and</u> of false acceptance of the γ bit preface code.

$$P(FWA)_{\substack{\text{combination}\\ \text{code}}} = 2^{-\gamma}\, P(FWA)_{\substack{\text{64 looks in}\\ \text{correlator}}} \sim 2^{-8} \times 9.76\times10^{-4} = 3.8125\times10^{-6}$$

For 8 bit preface & 64 looks & only 8 total fixed bits.

IF EACH COMMAND WORD IS PREFACED BY AN 8 BIT PRE-SYNC SEQUENCE AND EACH 48 BIT COMMAND WORD HAS INTERNAL TO ITSELF 7 FIXED ADDRESS BITS AND 2 CORRECT FIXED BITS FOR A TOTAL OF 9 FIXED BITS IN THE 41 BIT FIELD, THEN THE PROBABILITY OF FALSE COMMAND ACCEPTANCE IS (SINGLE LOOK BASIS)

$$P(FWA)_{\substack{\text{combination code,}\\ \text{single look}}} = 2^{-8} \times \frac{2^{41-9}}{2^{48}} = 5.96\times10^{-8} .$$

THIS IS SHORT OF THE SPECIFICATION OF 10^{-9}.

CONCLUSIONS

Obviously, the P(FCWD) is unacceptably high for situations 2 and 3. Fortunately, there is a possible remedy - one suggested by Mr. Howard Haugen of TRW. The first accepted command word shall not be acted upon, but the succeeding good command words will be acted upon. The probability of two successive FCWDs is approximately $(5.96 \times 10^{-8})^2$ or 3.552×10^{-15}.

One question does stand out. The earlier data handling equipment write-up (from Don Bastion?) mentioned 8 bit spacecraft ID, whereas the recent memo from Howard Haugen mentioned 7 bit spacecraft ID. Which is to be used?

ATTACHMENT 5

A Memo Concerning Energy-to-Noise Ratio for the Viterbi-BSC Channel and the Impact of Manchester Coding Loss, Memo, 22 January 1990

22 Jan 90

To: Mrs. Lee Anne Thomas, EB-33

This started out as a 1-2 hour memo and, believe it or not, I've spent all weekend on this. The first write-ups were 10 pages and seemed so awkward I then searched for a way to present this in a simple manner. Hopefully, this will resolve the questions concerning required energy-to-noise ratios for the Viterbi system.

At the end of the last page is the summary for the Soft Decision Viterbi-BSC channel. In studying this, I made use of Lin and Costello Error Control Coding Text, Chapter 11 and Error Control Coding Handbook by Joseph P. Odenwalder of Linkabit (Final Report).

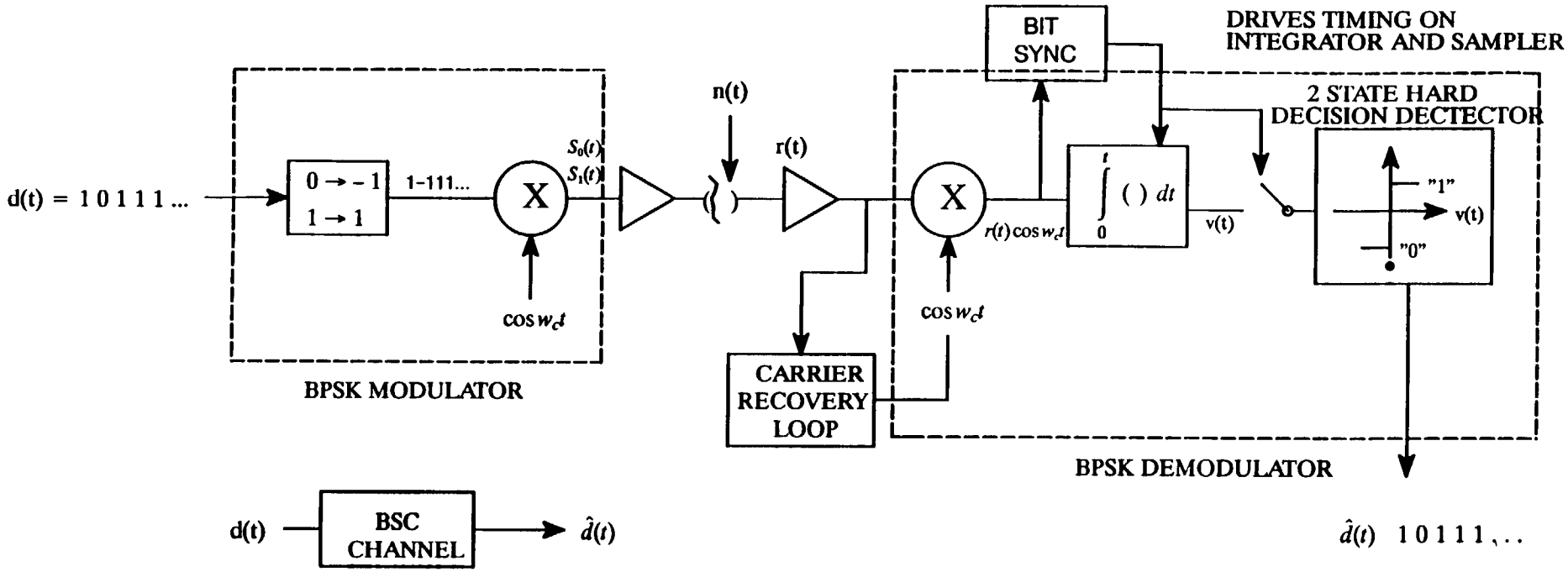
One thing I have realized is that the channel model includes the BPSK Modulator and BPSK Demodulator in what is termed the BSC (Binary Symmetric Channel) Channel model. Thus, we really talk about

- the SNR into the BPSK Demodulator,
- the symbol (or bit) error rate from BPSK demodulator,
- and the bit or data stream error rate out of the Viterbi decoder.

We see from the General Discussion that sending bits twice as fast on a BPSK BSC Channel the error rate goes up even though the average transmitted energy is kept the same. This is due to a shorter integration time in the BPSK demodulator.

Since two Convolutional Coded Symbols result in one data bit, $E_b = 2E_s$.

A. THE NEXT FIVE PAGES DISCUSS THE CODING GAINS AND $\frac{E_s}{N_o}$ and $\frac{E_b}{N_o}$ REQUIREMENTS TO ACHIEVE 10^{-5} AND WHERE THE $\frac{E_s}{N_o}$ and $\frac{E_b}{N_o}$ REQUIREMENTS ARE IN THE SYSTEM.



$S_0(t) = -\cos w_0 t$

$S_1(t) = \cos w_0 t$ $\qquad r(t) = S_i(t) + n(t)$

FIGURE 1 SINUSOIDAL CARRIER SIGNALLING

INPUT TO INTEGRATOR IS

$$S_i(t)\cos w_c t + n(t)\cos w_c t = \pm \cos^2 w_c t + n(t)\cos w_c t$$

OUTPUT OF INTEGRATOR IS

$$\pm \int_o^t \cos^2 w_c t dt + \int_o^t n(t)\cos w_c t dt$$

$$= \pm \frac{t}{2} + N(t) \qquad \text{where N(t) is a gaussian noise function}$$

IF BIT WIDTH IS T SECONDS, t IS CHOSEN TO BE = T..

$$E_0 = E_1 = \int_0^T S_{\frac{0}{1}}{}^2(t)dt = \frac{T}{2} \qquad or \ \frac{A^2T}{2} \ \text{if } A \text{ volts peak value}$$

It is shown in every text on communication theory and in Chapter 7 of Ziemer & Tranter, Principles of Communications that the probability of error of $\hat{d}(t)$ is

$$P(e) = \frac{1}{2} erfc \sqrt{\frac{A^2T}{2N_0}} \qquad \text{and } Z \overset{\Delta}{=} SNR \overset{\Delta}{=} \frac{E_b}{N_0} = \frac{A^2T/2}{N_0}$$

at point of r(t) or input to receiver ant.

Sending data at rate of 1 bit per T seconds yields $10^{-5} = P_1(e)$ for $Z_1 = 9.6dB$. *See Figure 3.1 attached from Odenwalder's Final Report 1976.

Sending data at rate of 2 bits per T seconds yields $10^{-5} = P_2(e)$ for $Z_2 = 9.6$ dB.

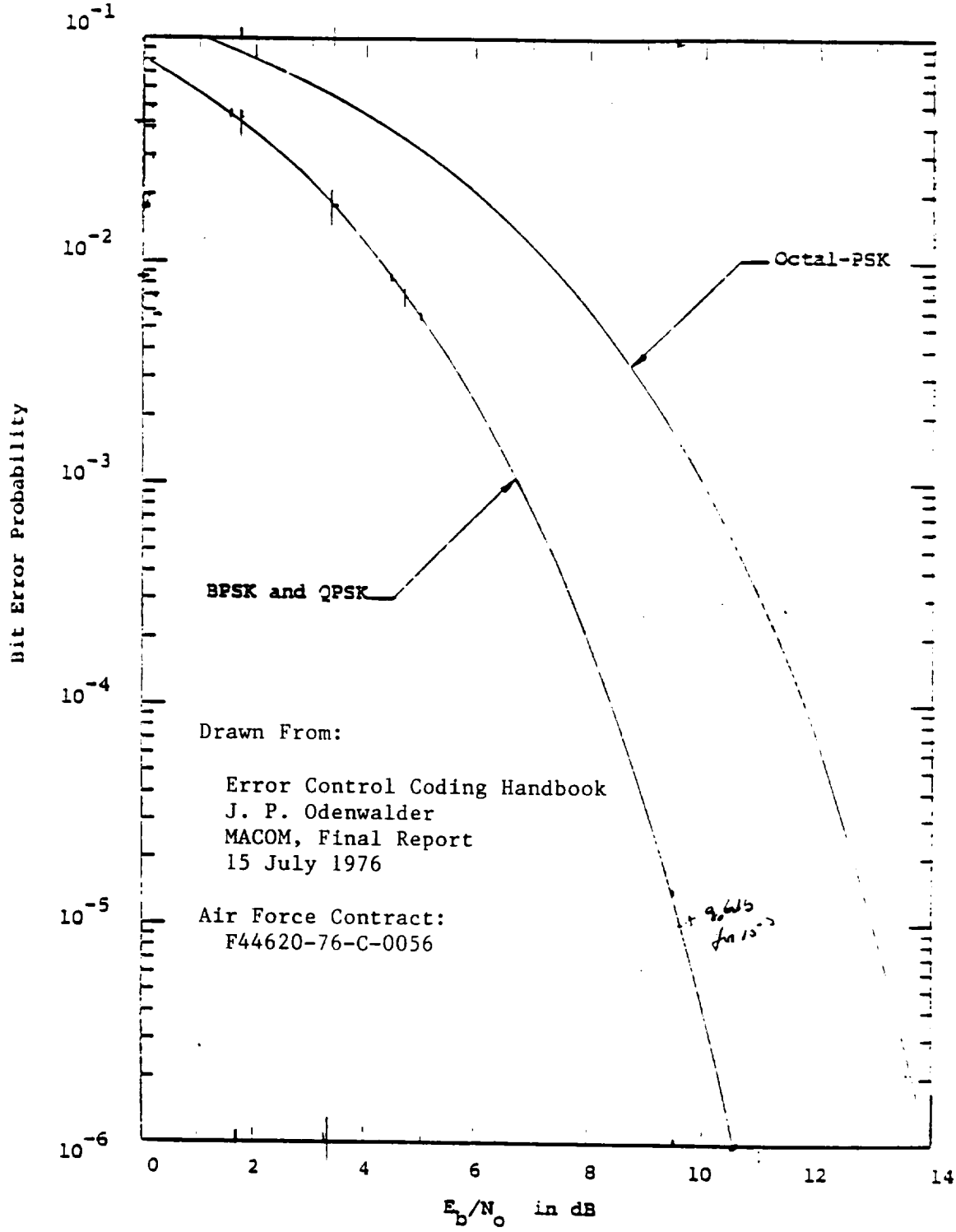


Figure 3.1 Bit error probability versus E_b/N_o performance of coherent BPSK, QPSK, and octal-PSK.

But $\vec{E} = \frac{A^2(T/2)}{2}$ and $E_b = \frac{A^2(T/2)}{2}$

And $Z_2 \triangleq \frac{A^2T}{4N_0} = \frac{1}{2} Z_1 = 6.6dB$ if $Z_1 = 9.6dB$

So, for same pulse amplitude, A, and same noise N_0 as case 1

$$P_2(e) \neq 2 \cdot 10^{-3} \quad Z_2 = 6.6dB$$

Now, let's add convolutional encoding/Viterbi decoding, Figure 2, and see what happens.

Again, sending data, d(t), at rate of 1 bit per T seconds, we ask what is required, Z_c = SNR coded, at input of receiver or at receiver antenna to yield 10^{-5} BER at $\hat{d}(t)$?

FIRST, LET'S USE HARD DECISION VITERBI AND BSC

From Figure 5.15, page 129 of Error Control Coding Handbook (Final Report) by Joseph P. Odenwalder of Linkabit (see attached curve), we see an error rate of 1.8×10^{-2} out of the BSC channel $\underset{BSC}{P(e)}$ will produce an error rate of 10^{-5} for $\hat{d}(t)$ out of the Viterbi decoder.

An error rate out of the BSC of 1.8×10^{-2} will occur for Z_{BSC} = 3.3 dB.

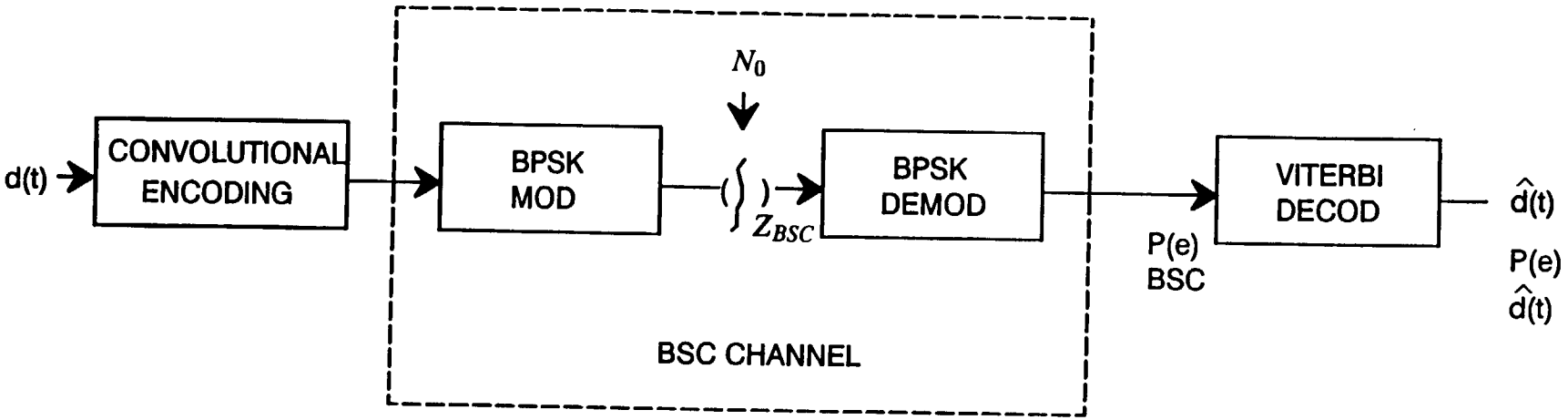
Thus, we see a coding gain of about 6.6 – 3.3 ≈ 3.3 dB relative to the BSC channel with 2 bits per T seconds and hard decision coding.

AND NOTING $E_s \triangleq$ channel symbol energy of T/2 bit time

and $E_b \triangleq$ data bit energy of T bit time.

We see from page before that $E_s = \frac{1}{2}E_b$ due to different bit times if pulse height is maintained at the same value.

C 2


d(t)
CONVOLUTIONAL ENCODING
BPSK MOD
N_0
Z_{BSC}
BPSK DEMOD
BSC CHANNEL
P(e)
BSC
VITERBI DECOD
$\hat{d}(t)$
P(e)
$\hat{d}(t)$

96

FIGURE 2

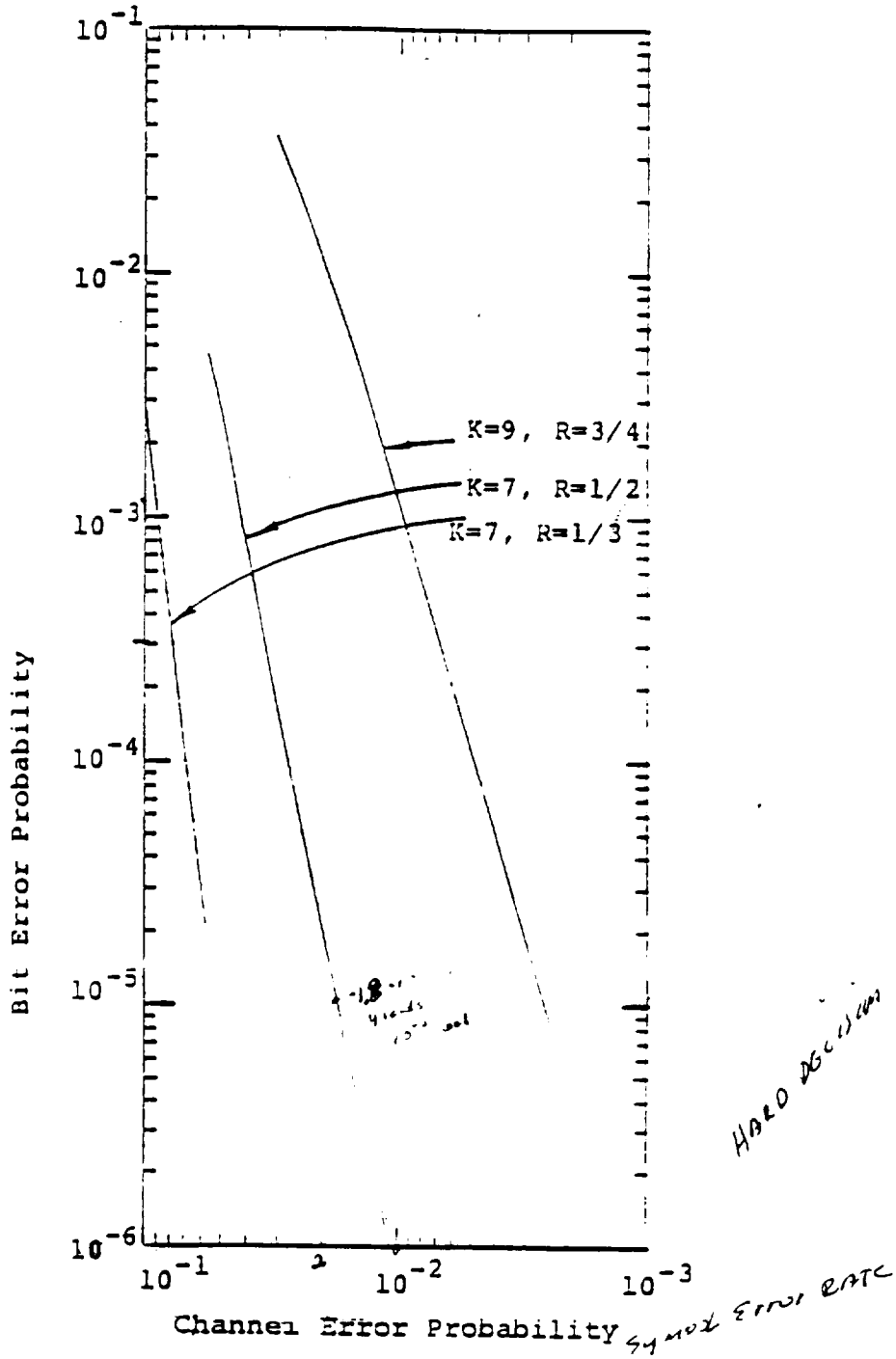


Figure 5.15 Bit error probability versus channel error rate performance of several convolutional coding systems.

awn From:

Error Control Coding Handbook
J. P. Odenwalder
MACOM, Final Report
15 July 1976

r Force Contract:

F44620-76-C-0056

To compare against uncoded BSC with 1 bit per T seconds, we must compare average energy over T second time for both systems.

For uncoded BSC, $\frac{\overline{E}}{N_0} = \frac{A^2T}{2N_0} = \frac{E_b}{N_0} =$ average SNR over T seconds.

For convolutional coded BSC,

$$\overline{E} = \frac{E_s + E_s}{N_0} = \frac{A^2T/4 + A^2T/4}{N_0} = \frac{A^2T}{2N_0} = \frac{E_b}{N_0} = \frac{2E_s}{N_0}$$

So $Z_{BSC} = 3.3dB \Rightarrow Z_{BSC} = 6.6dB$

$\frac{T}{2}$ *bit time* — T bit time

2 T/2 bits

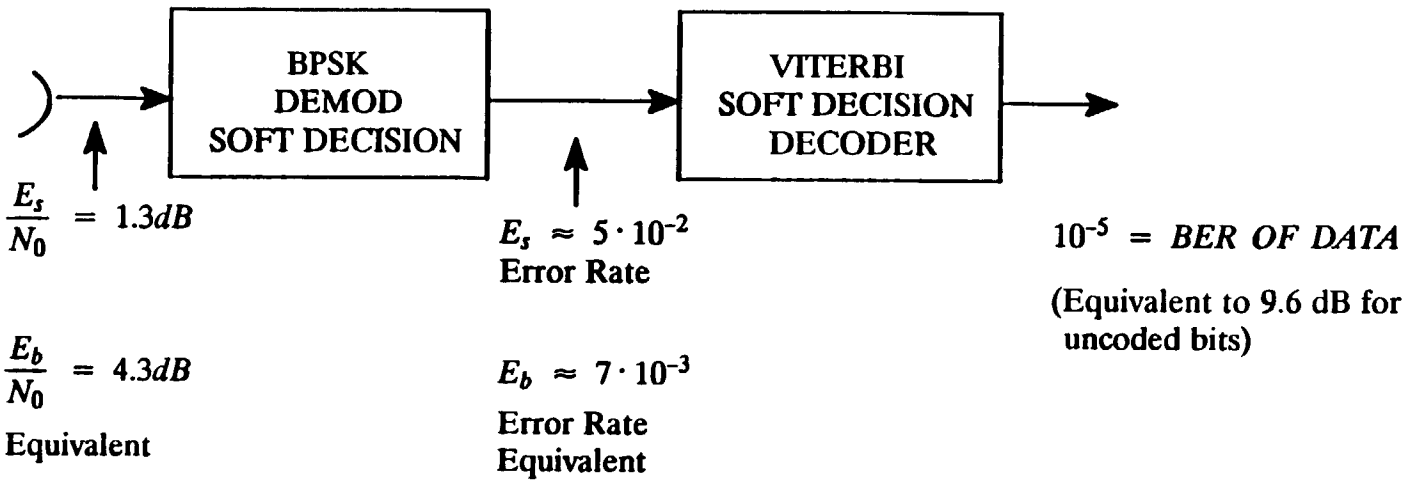
As a result, we see a coding gain of 9.6 – 6.6 ≈ 3 dB over the uncoded BSC channel for the same data rate and same average transmitted energy for hard decision decoding.

SOFT DECISION VITERBI AND BSC

All the above was for a Hard Decision BSC. If we use 3 bit Soft Decision, we will see an additional – 2 dB coding gain relative to uncoded BSC.

Thus, the required $\frac{Z_{BSC}}{\text{T/2 bit time}}$ SNR is ~3.3 dB – 2 dB ≈ 1.3 dB!

which is the required $\frac{E_s}{N_0}$ for soft decision Viterbi decoding. Figure 3 illustrates the results in graphical form.



Summary: Overall Coding Gain Soft Decision BSC and Viterbi = ~ 9.6 – 4.3 ≈ 5.3*dB*

FIGURE 3

B. THE NEXT 3 PAGES ARE A TUTORIAL ON WHAT MANCHESTER CODING FOR RF TRANSMISSION COSTS US IN AN EQUIVALENT dB LOSS OVER USING NRZ BITS FOR RF TRANSMISSION.

IF RF CHANNEL TRANSMISSIONS ARE 6 dB SNR OR MORE, WE LOSE ABOUT .6 dB IF WE USE MANCHESTER CODED NRZ BITS FOR RF TRANSMISSION.

- We see from Figure 4 that Manchester Coding for RF channel costs about .6 dB.

Prob of error of Manchester Pulse (BPSK Modulated) =

$$\frac{1}{2}\ erfc\ \sqrt{\frac{E_{MP}}{N_0}} = \frac{1}{2}\ erfc\ \sqrt{Z_{MP}}$$

Prob of error of NRZ bit
After MP to BRZ decoding

$$= 1 - \left[(Prob\ of\ 1st\ MP\ correct) \cdot (Prob\ of\ 2nd\ MP\ correct)\right]$$

$$= 1 - \left[\left(1 - \frac{1}{2}\ erfc\ \sqrt{Z_{MP}}\right)\left(1 - \frac{1}{2}\ erfc\ \sqrt{Z_{MP}}\right)\right]$$

$$= 1 - \left[1 - 2\,\frac{1}{2}\ erfc\ \sqrt{Z_{MP}} + \left(\frac{1}{2}\ erfc\ \sqrt{Z_{MP}}\right)^2\right]$$

$$P(e)_{NRZ\ BIT} = 1 - 1 + 2\,P(e)_{MP} - P^2(e)_{MP} \approx 2P(e)_{MP}$$

Thus

$$\boxed{P(e)_{NRZ\ BIT} \approx 2P(e)_{MP}}$$

IF NRZ CODED INTO
MANCHESTER PULSES

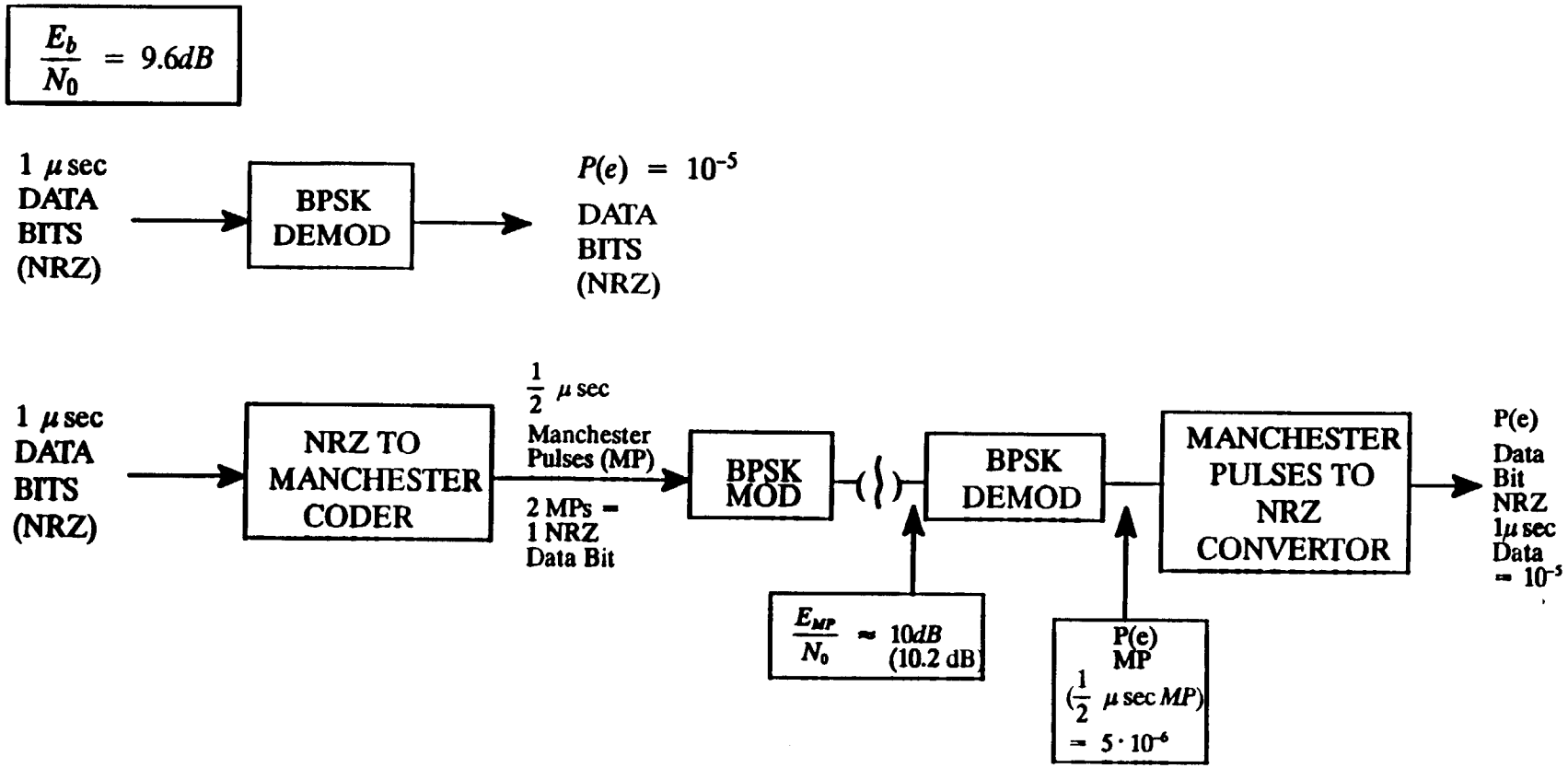


FIGURE 4 EFFECTS OF MANCHESTER CODING

Channel	Modulation / Demodulation	E_b/N_o in dB required for given bit error rate						
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
Additive white Gaussian noise	BPSK and QPSK	-.8	4.3	6.8	8.4	9.6	10.5	11.3
"	Octal-PSK	1.0	7.3	10.0	11.7	13.0	13.9	14.7
"	DBPSK	2.1	5.9	7.9	9.3	10.3	11.2	11.9
"	DQPSK	2.1	6.8	9.2	10.8	12.0	12.9	13.6
"	Noncoherently demodulated binary FSK	5.1	8.9	10.9	12.3	13.4	14.2	14.9
"	Noncoherently demodulated 8-ary MFSK	2.0	5.2	7.0	8.2	9.1	9.9	10.5
Independent Rayleigh fading	Binary FSK, L = 1	9.0	19.9	30.0	40.0	50.0	60.0	70.0
"	Binary FSK, L = 2	7.9	14.8	20.2	25.3	30.4	35.4	40.4
"	Binary FSK, L = 4	8.1	13.0	16.5	19.4	22.1	24.8	27.3
"	Binary FSK, L = 8	8.7	12.8	15.3	17.2	18.9	20.5	22.0
"	Binary FSK, L = 16	9.7	13.2	15.3	16.7	18.0	19.1	20.0
"	Binary FSK, L = 32	10.9	14.1	15.8	17.1	18.1	18.9	19.7

Table 3.1 Summary of uncoded system performances.

Drawn From:
Error Control Coding Handbook
J. P. Odenwalder
MACOM, Final Report
15 July 1976

Air Force Contract: F44620-76-C-0056

ATTACHMENT 6

Probability of False Polynomial Division Synchronization Using Shortened Cyclic Codes, Included in 16 May 1991 Monthly Report

AN INVESTIGATION OF ERROR CORRECTING TECHNIQUES FOR OMV AND AXAF

NASA GRANT: NAG8-104
QUARTERLY REPORT
MAY 16, 1991

SUBMITTED TO:

Mrs. Lee Ann Thomas
EB33
NASA/MSFC
MSFC, AL 35812
205 + 544-3662

SUBMITTED BY:

Frank Ingels, Ph.D.
Principal Investigator
Mississippi State University
Mississippi State, MS 39762
601 + 325-3912

QUARTERLY WORK STATEMENT

During the period February 1991 to May 1991, a report concerning the probability of false synchronization due to shortened cyclic code division was forwarded to Mrs. Thomas. A copy of a condensed version of this report is attached. A telecom was participated in on 16 May 1991.

A meeting with Dr. Gary Maki was attended at NASA/MSFC with Mrs. H. Thomas. This meeting discussed the Reed-Solomon chip set.

A report concerning the results of testing the Reed-Solomon chips is attached. The chips were tested for random and burst error patterns and all decodings were accurate.

The grant has been extended to June 1992, due to a slow down in the AFE project.

PROBABILITY OF FALSE POLYNOMIAL DIVISION SYNCHRONIZATION USING SHORTENED CYCLIC CODES

Anna Lynn Schauer
Mississippi State University*
(Now at Sandia National Laboratories,
Albuquerque, NM)

Frank M. Ingels
Mississippi State University

ABSTRACT

Shortened cyclic codes are not cyclic, but many cyclic shifts of various code words are still part of the shortened code set. This paper addresses the probability of false synchronization obtained through polynomial division of a serial shortened cyclic code stream in a "sliding" window correlator.

Key words: shortened cyclic codes, "sliding" window correlators, serial bit stream synchronization.

INTRODUCTION

Three basic operational modes have been considered for spacecraft uplink command communications. All of these modes have forty–eight bit command words.

The forty–eight bit command words include an eight bit spacecraft address, two fixed bits, thirty–one data bits and a seven bit parity check. Thus, each forty–eight bit command word would be comprised of forty–one bits plus a seven bit parity check. The seven bit parity check is formed by a polynomial division technique commonly referred to as Cyclic Rundancy Check, CRC, using the polynomial $g(x) = x^7 + x^6 + x^2 + x^0$ derived from the CCSDS 201.0–B–1 Standard, paragraph 3.3.1.

This polynomial is a non–primitive generator polynomial for the 63,56 single error correcting (SEC), double error detecting (DED) Hamming code. It in turn is constructed from the generator polynomial for the 63,57 single error correcting code which is $p(x) = x^6 + x + 1$ by multiplication by $1 + x$ to produce the 63,56 cyclic code generator polynomial g(x).

The command word is a shortened version of the 63,56 SEC, DED code. The shortening is accomplished in a virtual sense by assuming the first 15 bits of each word are zeroes. Thus, the forty–eight bit command words are actually a 48,41 SEC, DED code obtained by shortening the 63,56 SEC, DED cyclic code. (If no error correction is attempted, the code, which has minimum distance of four, can be used to detect up to three errors.) Importantly, it must be realized that shortened cyclic codes are not themselves cyclic [1].

The command word acceptance technique would include an exact match for the eight bit spacecraft address code, an exact match of the two fixed bits, a command length of exactly 48 bits and an exact match of the error checking polynomial parity bits as rederived on the spacecraft by division of the first 41 bits by the generator polynomial g(x). It is the probability of false acceptance of the division process that this paper addresses.

Normally the cyclic shift of a code word will result in another valid code word. (A cyclic shift is easily envisioned by placing a code word in a shift register with the output line fed back to the input of the register. A shift of the register, thus, will cycle the output bit of the register to the input of the register. The MSB then becomes the LSB for each cyclic shift thereof.) However, for shortened versions of a cyclic code, this property does not hold for all code words in the shortened code set.

If a synchronization scheme incorporates the concept of division of a serial bit pattern in an X bit window by a generator polynomial and essentially interpreting a remainder of zero as a valid code word, then this amounts to a 'sliding window' correlator, i.e., as the serial bit stream 'slides' by the X bit window, the contents of the window are divided by g(x) on a bit–by–bit basis. False synchronization (false acceptance) occurs when the remainder term from the division is zero and segments of two adjacent words, *x* and *y*, are within the correlator window.

There are two mechanisms by which false synchronization may occur. First, a sync will occur if an apparent *i*th (or (n–i)th) cyclic shift of word *x* (or *y*) occurs, as a serial pattern is shifted through the window. Second, that portion of the bit stream within the correlator window may accidently "coincide" with a different valid code word other than a cyclic shifted version of the word under inspection. The probabilities of each of these occurrences have been derived separately [2]. In order to distinguish the two mechanisms by which false synchronization may occur, the subscripts "CS" (cyclic shift) and "CC" (coincidental correlation) are used. The two mechanisms represent mutually exclusive events, therefore the probability of false sync acquisition is the sum of two individual probabilities,

$$P(FSA) = P(FSA)_{CS} + P(FSA)_{CC}. \tag{1}$$

FALSE SYNCHRONIZATION DUE TO CYCLIC SHIFTS OF *X* AND *Y* (FULL-LENGTH CODES, $l = 0$), $P(FSA)_{CS}$ (l = SHORTENING PARAMETER)

After a valid word is located in the correlator window, a cyclic shift of *x* will occur on the first serial shift of the bit stream through the correlator window if the bit "shifted in" matches the bit "shifted out." Since bit values "1" and "0" are equally likely, the probability that a cyclic shift of *x* results on the first shift or the correlator window is

$$P(\text{False sync on shift no. 1})_{CS} = 0.5.$$

False synchronization due to a cyclic shift of *x* occurs on the second shift if each of the two bits "shifted in" match each of the two bits "shifted out." Again, since code word bit values of "1" and "0" are equally likely, the probability that a cyclic shift of *x* occurs on the second shift of the correlator window is

$$P(\text{False sync on shift no. 2})_{CS} = (1/2)^2 = 0.25.$$

Once the correlator window contains more bits from word *y* than it does from word *x*, false synchronizations due to cyclic shifts are due to cyclic shifts of *y*. Thus, when full length codes are being considered, it is sufficient to calculate the first $\frac{n-1}{2}$ probabilities and use the property of symmetry to determine the remaining values.

In general, the probability of false synchronization due to cyclic shifts of *x* followed by *y* (full-length codes) is expressed as

$$\text{For the first } \frac{n-1}{2} \text{ shifts: } P(\text{False sync on shift } i)_{CS} = (1/2)^i,\ i = 1 \text{ to } \frac{n-1}{2};$$

$$\text{For the last } \frac{n-1}{2} \text{ shifts: } P(\text{False sync on shift } i)_{CS} = (1/2)^{n-i},\ i = \frac{n-1}{2} + 1 \text{ to } n-1. \tag{2}$$

Once the individual shift probabilities $P(\text{False sync on shift i})_{CS}$ have been determined, the average probability of false sync, $P(FSA)_{CS}$, is calculated by

$$P(FSA)_{CS} = \frac{\sum_{i=1}^{n-\ell-1} P(\textit{False sync on shift } i)_{CS}}{n-\ell}, \tag{3}$$

which yields an arithmetic average over *all* shifts.

Using Equation 3 and the symmetry of the probability $P(\text{False sync on shift i})_{CS}$ from Equation 2, the probability of false sync acquisition for full-length codes is [2, pages 18–21]:

$$P(FSA)_{CS} = \frac{2 \times \left[\sum_{i=1}^{\frac{n-1}{2}} (1/2)^i \right]}{n} \quad . \tag{4}$$

FALSE SYNCHRONIZATION DUE TO CYCLIC SHIFTS OF *X* AND *Y*, (CODES SHORTENED BY $l = 1$), $P(FSA)_{CS}$

Once a full-length code is shortened, the cyclic relationship between all code words is lost. (There may still be many cyclic shift code words present but not all cyclic shifts will be present.) Recall that the full-length code word from which the shortened word ($l = 1$) is derived has a zero in the MSB position [1]:

$$x_0\, x_1\, \ldots\, x_{n-3}\, x_{n-2}\, [0] \,.$$

The bracketed zero, [0], indicates the bit that is removed in order to shorten the code.

The full-length code words from which shortened words are derived will be used in the following discussion to demonstrate the conditions under which a cyclic shift of a valid (shortened) code word yields another valid code word.

Consider a full-length code word of the form $x_0\, x_1\, \ldots\, x_{n-3}\, x_{n-2}\, 0$. The first cyclic shift of this word has the form $0\, x_0\, x_1\, \ldots\, x_{n-3}\, x_{n-2}$. The original word, $x_0\, x_1\, \ldots\, x_{n-3}\, x_{n-2}\, 0$, may be shortened to form a word in a shortened code set. The new word of the full-length code formed by the first cyclic shift, $0\, x_0\, x_1 \ldots x_{n-3}\, x_{n-2}$, may also be shortened if the last bit, x_{n-2}, is a zero.

On the next cyclic shift, the word $x_{n-2}\, 0\, x_0\, x_1 \ldots x_{n-3}$ is formed, and may be shortened if the last bit, x_{n-3}, is a zero. At each shift of a valid full-length code word, a valid code word in shortened set, $l = 1$, can be created if the last bit of the full-length word is a zero. Because the form of the original full-length code word is specified with a zero in the last bit position, cyclic shifts of the word must track the location of this zero as well. Thus, two bit positions may be monitored at each cyclic shift:

1) the last bit, and
2) the location of the zero (or zeroes, for $l > 1$).

As a bit stream comprised of shortened code words passes through a correlator window, successive "cyclic shifts" of a word x are defined as follows:

Original word in the correlator window:	$x_0\, x_1\, x_2 \ldots x_{n-3}\, x_{n-2}$
First shift:	$b_m\, x_0\, x_1 \ldots x_{n-4}\, x_{n-3}$
Second shift:	$b_n\, b_m\, x_0 \ldots x_{n-5}\, x_{n-4}$

where b_n and b_m represent bits from the next word in the bit stream. At each shift i, a cyclic shift is said to occur and synchronization is flagged when

Rule 1) the $n-l$ bits in the correlator window form the first $n-l$ bits of the ith shift of the full-length word from which the original shortened code word is derived, and

Rule 2) the last l bits in the correlator window on shift $i-l$ are zeroes.

These two stipulations relate to the two bit positions which may be monitored in cyclic shifts of full-length code words of the form $x_0\, x_1\, x_2 \ldots x_{n-l}\, 0_{n-l} \ldots 0_{n-1}$.

With these guidelines in place, it is now possible to determine P(False sync on shift no. 1) when $l = 1$. On the first shift, the bit pattern in the correlator window is $b_m\, x_0\, x_1 \ldots x_{n-4}\, x_{n-3}$. Synchronization will

occur if the bit $\mathbf{b_m}$ is a zero (since the first cyclic shift of the full-length word from which the original shortened code word is derived is $\mathbf{0\ x_0\ x_1 \ldots x_{n-3}\ x_{n-2}}$) *and* bit $\mathbf{x_{n-2}}$ is a zero (since a full-length word must have a zero in the last location in order to be a valid shortened word when $l = 1$). Stated differently, synchronization will occur on the first shift if a zero is shifted in and a zero is shifted out. Thus, the corresponding probability of this event is

$$\begin{aligned} P(\text{False sync on shift no. 1})_{CS} &= P(\text{Zero is shifted out}) \bullet P(\text{Zero is shifted in}) \\ &= (1/2)\ (1/2) \\ &= 0.25. \end{aligned}$$

Note that the events "zero is shifted out" and "zero is shifted in" are independent.

On the second shift through the correlator window, there are four combinations of "1"s and "0"s of bits to be shifted in: **00, 01, 10,** and **11**. By the definition of a cyclic shift, the bits which are "shifted in" are also "shifted out." However, since the word on which cyclic shifts are being performed is a full-length word of the form $\mathbf{x_0\ x_1\ x_2 \ldots x_{n-3}\ x_{n-2}\ [0]}$, the bits which are shifted out are the last two bits, $\mathbf{x_{n-2}}$ and **0**. Of these combinations, **01** and **11** cannot yield synchronization due to cyclic shift since a "1" is in the **[0]** location. This preceding discussion provides an intuitive interpretation for Rule 1 (above).

In order to evaluate the remaining combinations **00** and **10**, Rule 2 must be considered. This is best illustrated by example. If **00** is shifted in on shift $i = 2$, the bit pattern in the window on each of the first two shifts is

$\mathbf{x_0\ x_1\ x_2 \ldots x_{n-4}\ x_{n-3}\ x_{n-2}}$	*Original (shortened) word in correlator window*
$\mathbf{b_m = 0\ x_0\ x_1\ x_2 \ldots x_{n-4}\ x_{n-3}}$	*First shift*
$\mathbf{b_n = 0\ b_m = 0\ x_0\ x_1\ x_2 \ldots x_{n-4}}$	*Second shift.*

Recall that the full-length word from which this shortened word is derived has the form $\mathbf{x_0\ x_1\ x_2 \ldots x_{n-4}\ x_{n-3}\ x_{n-2}\ x_{n-1}}$. If $\mathbf{b_n = 0\ b_m = 0}$ is cyclic shifted in, then $\mathbf{x_{n-2} = 0\ x_{n-1} = 0}$ must be shifted out. Therefore, for false synchronization to occur, the following three events must occur. First, $\mathbf{b_m}$ must be a zero (in accordance with Rule 1). Second, since $\mathbf{b_n = 0}$, $\mathbf{x_{n-2}}$ must also be a zero. Third, if a valid shortened word is to occur on shift $i = 2$, then the bit shifted out on $i = 1$ must be a zero (in accordance with Rule 2). Thus, $\mathbf{x_{n-3}}$ must be a zero.

Now consider the combination **10**. First, by Rule 1, if $\mathbf{b_n = 1\ b_m = 0}$ is cyclic shifted in, then bit $\mathbf{x_{n-2}}$ must be a one. Second, since $\mathbf{b_n = 1}$, $\mathbf{x_{n-2}}$ must also be a one. Third, by Rule 2, $\mathbf{x_{n-3}}$ must be a zero.

The probability of false sync on the second shift is calculated as follows. First, only those combinations of bits shifted in which do not violate Rule 1 must be considered. For each combination that can yield false sync acquisition, the following independent probabilities must be considered:

1) The probability of shifting in bit pattern $\mathbf{b_n b_m}$ must be evaluated.
2) Since cyclic shifts are being considered, the bit $\mathbf{b_m}$ corresponds to the 0 that is removed when the full-length code is shortened, and bit $\mathbf{b_m}$ corresponds to the bit $\mathbf{x_{n-2}}$ which is shifted from the correlator window on shift $i = 1$. By definition of cyclic shifts, $\mathbf{x_{n-2} = b_n}$. The probability of this event must be considered.
3) Even though bit pattern $\mathbf{b_n b_m}$ may be shifted in on the first two shifts, and bit $\mathbf{x_{n-2} = b_n}$ may be shifted out on shift $i = 1$, there is still only a 50/50 chance that bit $\mathbf{x_{n-3}}$ is a 0, in accordance with Rule 2.

Thus, the probability of false sync acquisition on shift $i = 2$ becomes

$$P(\text{False sync on shift no. 2})_{CS} = P(\mathbf{00}\text{ shifted in}) \bullet P(\mathbf{x_{n-2} = 0}) \bullet P(\mathbf{x_{n-3} = 0})$$

$$+ \quad P(\mathbf{10}\text{ shifted in}) \bullet P(x_{n-2}=1) \bullet P(x_{n-3}=0)$$

$$= \quad (1/2)(1/2) \bullet (1/2) \bullet (1/2) + (1/2)(1/2) \bullet (1/2) \bullet (1/2)$$

$$= \quad 0.125.$$

Probability derivation for the third shift proceeds in a similar manner. There are 2^3 possible bit combinations shifted in and out. Of these, the combinations **001, 011, 101,** and **111** cannot result in false synchronization due to cyclic shifts of the original word x since a "1" is found in the deleted zero location of the full-length word from which the shortened word is derived. Each of the remaining combinations **000, 010, 100,** and **110** must be evaluated, and the same three independent probabilities must be considered on shift $i=3$ as were considered on shift $i=2$.

The probability of false sync on the third shift is

P(False sync on shift no. 2)$_{CS}$

$$= \quad P(\mathbf{000\text{ shifted in}}) \bullet P(x_{n-2}=0\ x_{n-3}=0) \bullet P(x_{n-4}=0)$$

$$+ \quad P(\mathbf{010\text{ shifted in}}) \bullet P(x_{n-2}=0\ x_{n-3}=1) \bullet P(x_{n-4}=0)$$

$$+ \quad P(\mathbf{100\text{ shifted in}}) \bullet P(x_{n-2}=1\ x_{n-3}=0) \bullet P(x_{n-4}=0)$$

$$+ \quad P(\mathbf{110\text{ shifted in}}) \bullet P(x_{n-2}=1\ x_{n-3}=1) \bullet P(x_{n-4}=0)$$

$$= \quad 4\ [(1/2)\ (1/2)\ (1/2) \bullet (1/2)\ (1/2) \bullet (1/2)]$$

$$= \quad 0.0625.$$

If the bits within the correlator window are comprised evenly of portions from x and y, cyclic shifts of either word are equally possible. This special case, where cyclic shifts of *both* x and y may occur, is referred to as the *center probability*, CP. The probabilities P(False sync on shift i)$_{CS}$ are symmetric about the center probability.

In general, the probabilities of false sync due to cyclic shifts of code words shortened by $l = 1$ are as follows [2, pages 21–26 contain the full derivation]:

For the first int $\left(\frac{n-2}{2}\right)$ *shifts*:

$$P(\text{False sync on shift } i)_{CS} = 2^{i-1}\,(0.5)^{\,i + (i-1) + 1}$$

$$= 2^{-(i+1)},\ i = l \text{ to int } \left(\frac{n-2}{2}\right);$$

For shift number int $\left(\frac{n-2}{2}\right) + 1$ (*Center Probability*):

$$P(\text{False sync on shift } i)_{CS} = 2^{i} \bullet (0.5)^{\,i + (i-1) + 1}$$

$$= 2^{-i},\ i = \text{int } \left(\frac{n-2}{2}\right) + 1;$$

For the last int $\left(\frac{n-2}{2}\right)$ *shifts*:

$$\text{P(False sync on shift } i)_{CS} = 2^{n-i-2}\,(0.5)^{\,n-i-1+(n-i-2)+1}$$

$$= 2^{-n+i},\ i = \text{int}\left(\frac{n-2}{2}\right) + 2 \text{ to } n-2.^{*} \tag{5}$$

From Equation 5 (above) and Equation 3,

the probability of false sync acquisition may be written as

$$P(FSA)_{CS} = \frac{2\left[\sum_{i=1}^{\sigma} 2^{i-1}\,(0.25)^{i}\right] + 2^{\sigma+1}\,(0.25)^{\sigma+1}}{n-1} = \frac{2\left[\sum_{i=1}^{\sigma} 2^{-(i+1)}\right] + 2^{-i}}{n-1}\ , \tag{6}$$

for the $l = 1$ shortened cyclic code using the symmetry property where σ = integer value of (n–2)/2.

FALSE SYNCHRONIZATION DUE TO CYCLIC SHIFTS OF *X* AND *Y*, (GENERAL CASE: CODES SHORTENED BY *l*), $(P(FSA)_{CS}$

Following the preceding discussion for codes shortened by $l = 1$, a general probability expression for codes shortened by l has been derived for each shift of the correlator window:

For the first l shifts:

$$\text{P(False sync on shift } i)_{CS} = (0.25)^{i},\ i = 1 \text{ to } l;$$

For the next $\text{int}\left(\frac{n-l-1}{2}\right) - l$ *shifts*:

$$\text{P(False sync on shift } i)_{CS} = 2^{-(i+l)},\ i = l+1 \text{ to } \text{int}\left(\frac{n-l-1}{2}\right);$$

IF $\left[\frac{n-l-1}{2} - \text{int}\left(\frac{n-l-1}{2}\right)\right] = 0.5$, **THEN** (*Center Probability*):

$$\text{P(False sync on shift } i)_{CS} = 2^{-(i+l-1)},\ i = \text{int}\left(\frac{n-l-1}{2}\right) + 1\ . \tag{7}$$

The property of symmetry is used to derive the remaining values for shift numbers greater than $[(n-l-1)/2] + 1$.

As in the previous case for $l = 1$, Equation 3 can be used to determine the average value for $P(FSA)_{CS}$. A computer algorithm was written for ease in performing the calculations for $P(FSA)_{CS}$.

FALSE SYNC ACQUISITION DUE TO COINCIDENTAL CORRELATION, $P(FSA)_{CC}$

Coincidental correlation occurs when a valid code word is formed from segments of two adjacent words on shift *i*, but this windowed code word is not the i^{th} cyclic shift of word *x*, or the $(n-l-i)^{th}$ shift of word *y*. In order to derive an expression for $P(FSA)_{CC}$, a shift-by-shift analysis of word *x* as it passes through the correlator window was performed.

*The operator *int* extracts the integer portion of the argument, i.e., *int*(4.7) = 4.

Analogous to cyclic shifts, the average coincidental correlation false synchronization, $P(FSA)_{CC}$, is:

$$P(FSA)_{CC} = \frac{\sum_{i=1}^{n-l-1} P(\textit{False sync on shift } i)_{CC}}{n - l} \quad . \tag{8}$$

Performing a shift-by-shift analysis of a non-shortened word x as it passes through the correlator window reveals that individual shift probabilities are calculated from

$P(\text{False sync on shift } i)_{CC} =$
$P(\text{Valid code word}) \bullet P(\text{At least } d_{min} \text{ differences or "errors" occur on shift } i).$ (9)

The probability term P(Valid code word) was derived for full-length and shortened codes. The probability term P(At least d_{min} "errors" occur on shift i) derivation is somewhat involved and refers to the possibility that a new different valid code word is accidently formed after i shifts. A conservative approximation for P(At least d_{min} "errors" occur on shift i) is presented for shortened codes, and an exact expression is given for full-length codes. The results of both terms are applied to the final expressions for P(False sync on shift $i)_{CC}$, and a computer algorithm for computing upper and lower bounds for $P(FSA)_{CC}$ was written.

Because the derivation of $P(FSA)_{CC}$ is somewhat involved, an outline of the procedure is given in Figure 1.

Derivation of P(FSA), A Bottom-up Approach

Step 1: Derivation of P(Valid code word)

Step 2: Derivation of P(At least d_{min} "errors" occur on shift i)

a) Compute $N_e(i; d_{min}, n, k)$

b) Compute probabilities associated with each bit pattern containing d_{min} or more "errors"

i) *For* $l = 0$, an exact expression for P(At least d_{min} errors occur on shift i) may be derived

ii) *For* $l > 0$, upper and lower bounds for P(At least d_{min} errors occur on shift i) may be derived

Step 3: Derivation of $P(\text{False sync on shift i})_{CC}$

Step 4: Computation of $P(FSA)_{CC}$

Figure 1. Outline of $P(FSA)_{CC}$ Derivation

On the first shift of a full-length code word x through the correlator window, the bit pattern in the window will appear as a cyclic shift of x, or a shifted version of x in "error" by one bit in the first location*:

(First shift)

$x_{n-1}\ x_0\ x_1\ x_2 \cdots x_{n-3}\ x_{n-2}$	$\epsilon\ x_0\ x_1\ x_2 \cdots x_{n-3}\ x_{n-2}$ ·
(No "errors")	("Error" in first bit location)

If a cyclic shift has not occurred, then assuming for the moment that the minimum distance of the code is $d_{min} = 3$, division of the word $\epsilon\ x_0\ x_1\ x_2 \cdots x_{n-3}\ x_{n-2}$ by the generator polynomial g(X) will result in a non-zero remainder, and false synchronization due to coincidental correlation cannot occur.

On the second shift of the bit stream through the correlator window, the bits within the window will appear to be one of the following four cases:

(Second shift)

$x_{n-2}\ x_{n-1}\ x_0\ x_1\ x_2 \cdots x_{n-3}$	$\epsilon\ x_{n-1}\ x_0\ x_1\ x_2 \cdots x_{n-3}$
(No "errors")	("Error" in first bit location)
$x_{n-2}\ \epsilon\ x_0\ x_1\ x_2 \cdots x_{n-3}$	$\epsilon\ \epsilon\ x_0\ x_1\ x_2 \cdots x_{n-3}$ ·
("Error" in second bit location)	("Errors" in first bit locations)

The first of the four cases represents the probability of false synchronization due to a cyclic shift of the word x, equal to 0.25. This coincides exactly with the probability of false sync on shift $i = 2$ found in the previous section. However, since $d_{min} = 3$, there are not enough accumulated "errors" (possible differences) to cause false synchronization due to coincidental correlation.

It is on the third shift of the bit stream through the correlator window that at least one case accumulates at least d_{min} "errors:"

(Third shift)

$x_{n-3}\ x_{n-2}\ x_{n-1}\ x_0\ x_1\ x_2 \cdots x_{n-4}$	$x_{n-3}\ \epsilon\ x_{n-1}\ x_0\ x_1\ x_2 \cdots x_{n-4}$
$x_{n-3}\ x_{n-2}\ \epsilon\ x_0\ x_1\ x_2 \cdots x_{n-4}$	$x_{n-3}\ \epsilon\ \epsilon\ x_0\ x_1\ x_2 \cdots x_{n-4}$
$\epsilon\ x_{n-2}\ x_{n-1}\ x_0\ x_1\ x_2 \cdots x_{n-4}$	$\epsilon\ \epsilon\ x_{n-1}\ x_0\ x_1\ x_2 \cdots x_{n-4}$
$\epsilon\ x_{n-2}\ \epsilon\ x_0\ x_1\ x_2 \cdots x_{n-4}$	$\epsilon\ \epsilon\ \epsilon\ x_0\ x_1\ x_2 \cdots x_{n-4}$·
("Errors" in 1st and 3rd bit locations)	("Errors" in first three bit locations)

Again note that the first of the eight cases listed above will generate false synchronization due to a cyclic shift of the word x, corresponding to the 0.125 P(False sync on shift no. 1) found from Equation 2 in the previous section. The last case, where the word in the correlator window appears to be the third cyclic shift of word x corrupted by "errors" in the first bit locations, may result in false synchronization. The probability that at least d_{min} "errors" occur on shift i must be "weighted" by the probability that a random set of $n-l$ bits forms a valid code word, where $l = 0$ for full-length codes. For full-length codes, all patterns have equal probability, whereas for shortened codes the pattern probabilities differ.

DERIVING P(VALID CODE WORD)

In any full-length (n, k) code, the total number of possible combinations of n bits is 2^n, and the number of valid code words in the set is equal to 2^k. Therefore, the probability that any random set of n bits is a valid code word is

* The word "error" as used here is unrelated to the channel bit error rate.

$$P(Valid\ code\ word) = \frac{2^k}{2^n} = 2^{(k-n)} = \frac{1}{2^{(n-k)}} \quad . \tag{10a}$$

In a shortened $(n-l, k-l)$ code, the same equation is derived:

$$P(Valid\ code\ word) = \frac{2^{k-l}}{2^{n-l}} = 2^{(k-n)} = \frac{1}{2^{(n-k)}} \quad . \tag{10b}$$

This expression may be substituted directly into Equation 9.

DERIVING P(AT LEAST d_{min} "ERRORS" OCCUR ON SHIFT i)

At each of the first $int\left(\frac{n-l-1}{2}\right)$ shifts of word x through the correlator window, the pattern of bits in the correlator window at shift i will appear to be one of the following:

- the i^{th} cyclic shift of word x, or
- the i^{th} cyclic shift of word x corrupted by "errors" in one or more of the first i bit positions.

When at least d_{min} or more "errors" have accumulated in the i^{th} cyclic shift of word x, it is possible for the bit pattern to match a valid code word in the code set. For all shifts i such that $d_{min} \le i \le n - d_{min} - l$, there will be at least one possible pattern containing at least d_{min} "errors." Associated with each pattern containing at least d_{min} "errors" is a probability of occurrence; by summing the probabilities for each possible pattern containing at least d_{min} "errors" at each shift i, a value for P(At least d_{min} "errors" occur on shift i) can be derived. Thus, the probability P(At least d_{min} "errors" occur on shift i) is obtained in a two-step process:

1) The number, N_ϵ (i; d_{min}, n, k), of possible patterns with at least d_{min} "errors" must be computed for each shift i; and
2) For each shift i, the probabilities associated with all possible bit patterns with at least d_{min} "errors" must be summed together to obtain P(At least d_{min} "errors" occur on shift i).

We note that as i increases, virtually all the patterns have d_{min} or more "errors."

In general, the number of possible bit patterns with at least d_{min} "errors" in shift number i can be expressed as follows:

No. of ways in which d_{min} *or more "errors" can occur on shift* i $= N_\epsilon$ (i; d_{min}, n, k)

$= 0, \quad i < d_{min}$

$$= \sum_{m=d_{min}}^{i} \binom{i}{m}, \qquad d_{min} \le i \le \text{int}\left(\frac{n-1-l}{2}\right)$$

IF $\left(\frac{n-1-l}{2}\right) - \text{int}\left(\frac{n-1-l}{2}\right) = 0.5$, **THEN**

$$= \sum_{m=d_{min}}^{i} \binom{i}{m} , \quad i = \text{int}\left(\frac{n-1-l}{2}\right) + 1$$

$$= \sum_{m=i}^{n-d_{min}-l} \binom{n-m}{d_{min}} , \quad n-l-\text{int}\left(\frac{n-1-l}{2}\right) \leq i \leq n-d_{min}-l$$

$$= 0 , \quad i > n-d_{min}-l . \tag{11}$$

Thus, the upper bound for the probability at least d_{min} "errors" occur on shift i is 1.0. A lower bound for shortened codes has been derived by assuming that each of N_ϵ (i; d_{min}, n, k) patterns have a minimum probability of occurrence. Thus, the general lower bound expression is (l = shortening parameter):

$$\begin{aligned} P(\text{At least } d_{min} \text{ "errors" occur on shift } i)_{LB} &= N_\epsilon (i; d_{min}, n, k) (0.25)^l && i = 1 \text{ to } l \\ &= N_\epsilon (i; d_{min}, n, k) (.25)^l (0.5)^{i-l} && i = l+1 \text{ to CP} \end{aligned} \tag{12}$$

Again, symmetry may be used to determine the values for the remaining shifts.

For full-length codes, it is possible to calculate exactly the probability of at least d_{min} "errors" occur on shift i. These probabilities for full-length codes are:

$$\begin{aligned} P(\text{At least } d_{min} \text{ "errors" occur on shift } i) &= N_\epsilon (i; d_{min}, n, k) (0.5)^{-i} && i = 1 \text{ to } \frac{n-1}{2} \\ &= N_\epsilon (i; d_{min}, n, k) (0.5)^{n-i} && i = \frac{n}{2} \text{ to } n \end{aligned} \tag{13}$$

COMPUTING P(FALSE SYNC ON SHIFT i)$_{CC}$

Expressions for P(Valid code word) and P(At least d_{min} errors occur on shift i) have been derived in the previous section. In summary, the probability of coincidental correlation on shift i is computed as the product of two probabilities, P(False Acceptance) and P(At least d_{min} errors occur on shift i), and is equal to (l = shortening parameter):

P(False sync on shift i)$_{CC(l=0)}$

$$= P(\text{At least } d_{min} \text{ "errors" occur on shift } i) \bullet P(\text{False Acceptance})$$

$$= N_\epsilon(i; d_{min}, n, k) \bullet \frac{1}{2^i} \bullet \frac{1}{2^{(n-k)}}$$

$$= N_\epsilon(i; d_{min}, n, k) \bullet \frac{1}{2^{(n-k+i)}} ; \tag{14a}$$

For $l > 0$:

P(False sync on shift i)$_{CC,UB}$

$$= P(\text{At least } d_{min} \text{ "errors" occur on shift } i) \bullet P(\text{False Acceptance})$$

$$= 1.0 \bullet \frac{1}{2^{n-k}}$$

P(False sync on shift i)$_{CC,LB}$

$$= \text{P(False Acceptance)} \bullet \text{P(At least } d_{min} \text{ "errors" occur on shift } i)$$

$$= \left(\frac{1}{2^{n-k}}\right) \bullet (0.25)^l \bullet N_\epsilon(i; d_{\min}, n, k), \quad i \le l$$

$$= \left(\frac{1}{2^{n-k}}\right) \bullet \left(\frac{1}{2^i - l}\right) \bullet (0.25)^l \bullet N_\epsilon(i; d_{\min}, n, k), \quad i > l. \tag{14b}$$

and P(FSA)$_{CC}$ is the average value of P(False sync on shift i)$_{CC}$:

$$P(FSA)_{CC} = \frac{\sum_{i=1}^{n-l-1} P(\textit{False sync on shift } i)_{CC}}{n - l}. \tag{15}$$

A computer algorithm was developed for computing exact values for this expression when $l = 0$, and upper and lower bounds when $l > 0$. Comparisons of exact values and the upper and lower bound approximations are given in Table 1 for two codes. The (14,10) code has been shortened from the (15,11) code, and the (30,25) code has been shortened from the (31,26) code. A simulated serial bit stream was used to calculate actual probabilities of false synchronization on shift i for various codes. Assuming all valid code words are equally possible, shortened code words selected at random were simulated. The results in Table 2 were compiled using twenty sets of 1000 serially transmitted code words and averaging the results.

CONCLUSIONS

For full-length codes, where the probability of false sync acquisition may be calculated exactly, the theoretical and simulated results are virtually identical. The predicted value tends to be conservative, and this result may be predicted by examining the expression for P(FSA)$_{CC}$. The probability of P(FSA)$_{CC}$ is developed using the probability of false sync acquisition on any shift i. The P(False sync on shift i) is found to be the product of two probabilities, P(Valid code word) and P(At least d_{min} "errors" occur on shift i). The probability P(Valid code word) is an approximation to the more conservative (smaller) probability, P(Valid code word given the position of the accumulated "errors"). For example, no code word from the (7,4) code differs from any other word in the code set in three (and only three) consecutive locations. Thus, the third shift through the correlator window of a code word x from the (7,4) code set will not yield a coincidental match. Therefore, false synchronizations from this code set are a function of cyclic shifts only. The predicted value for P(FSA), however, will account for some coincidental correlation. The percent error in each is small: less than 4% of the predicted value in each of the three cases (1.17% for the (15,11) code; 1.35% for the (31,26) code; and 3.5% for the (63,57) code).

For shortened codes, where the actual probability of false sync acquisition lies between theoretical upper and lower bounds, the results again compare well; the simulated results lie between the theoretical bound for all but the (12,8) and (58,52) codes. For these codes, the error can be explained by examining the shift-by-shift probabilities of false sync acquisition, where the bulk of the error can be traced to the second and second-from-last shift probabilities. Whereas the predicted probability of false sync acquisition on these shifts is approximately 0.0625, the simulated value is approximately twice that, or 0.125. Simulations of (13,9) and (59,53) codes (where the code has been shortened by a lesser amount), and (11,7) and (57,51) codes (where the code has been shortened by a greater amount) yield simulated values which again lie between the predicted bounds. By examining the generator polynomial and the shortened code words, this discrepancy can be explained.

Table 1. Comparison of Exact and Upper and Lower Bound Values for P(False sync on shift i)$_{CC}$

Shift No.	(14,10) Code Exact Value	(14,10) Code Upper Bound	(14,10) Code Lower Bound	(30,25) Code Exact Value	(30,25) Code Upper Bound	(30,25) Code Lower Bound
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0
3	0.011719	0.0625	0.003906	0.005859	0.031250	0.001953
4	0.025391	0.0625	0.009765	0.012695	0.031250	0.004883
5	0.037109	0.0625	0.015625	0.018555	0.031250	0.007813
6	0.045898	0.0625	0.020509	0.022949	0.031250	0.010254
7	0.052002	0.0625	0.024170	0.026001	0.031250	0.012085
8	0.045898	0.0625	0.020509	0.028015	0.031250	0.013367
9	0.037109	0.0625	0.015625	0.029297	0.031250	0.014221
10	0.025391	0.0625	0.009765	0.030090	0.031250	0.014771
11	0.011719	0.0625	0.003906	0.030571	0.031250	0.015114
12	0.0	0.0	0.0	0.030857	0.031250	0.015324
13	0.0	0.0	0.0	0.031025	0.031250	0.015450
14	0.0	0.0	0.0	0.031122	0.031250	0.015524
15	0.0	0.0	0.0	0.031178	0.031250	0.015567
16	0.0	0.0	0.0	0.031122	0.031250	0.015524
17	0.0	0.0	0.0	0.031025	0.031250	0.015450
18	0.0	0.0	0.0	0.030857	0.031250	0.015324
19	0.0	0.0	0.0	0.030571	0.031250	0.015114
20	0.0	0.0	0.0	0.030090	0.031250	0.014771
21	0.0	0.0	0.0	0.029297	0.031250	0.014221
22	0.0	0.0	0.0	0.028015	0.031250	0.013367
23	0.0	0.0	0.0	0.026001	0.031250	0.012085
24	0.0	0.0	0.0	0.022949	0.031250	0.010254
25	0.0	0.0	0.0	0.018555	0.031250	0.007813
26	0.0	0.0	0.0	0.012695	0.031250	0.004883
27	0.0	0.0	0.0	0.005859	0.031250	0.001953
28	0.0	0.0	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0	0.0	0.0

Table 2. Average Probability of False Sync Acquisition Over All Possible Shifts

Full-Length Codes	Simulated Value	Predicted Value
(15,11) Code	0.150233	0.152018
(31,26) Code	0.083545	0.084684
(63,57) Code	0.043075	0.046384

Shortened Codes	Upper Bound	Simulated Value	Lower Bound
(10,6) Code	0.097751	0.091046	0.067039
(12,8) Code	0.093422	0.096597	0.006026
(21,16) Code	0.055536	0.054200	0.031959
(26,21) Code	0.050927	0.048117	0.026302
(28,23) Code	0.050219	0.048978	0.026925
(53,47) Code	0.026729	0.026109	0.012627
(58,52) Code	0.025806	0.029025	0.011918
(60,54) Code	0.025777	0.024082	0.013049
OMV (48,41) Code	0.020562	0.020252	0.013897

Each of these two code sets has a generator polynomial of the form $g(X) = 1 + X + X^{(n-k)}$. When the code sets are shortened by $n-k+1$, the percentage of (shortened) code words which form a shifted version of the generator polynomial is approximately doubled. This may be shown empirically by successively shortening the full-length code sets and monitoring the location of cyclic shifts of the generator polynomial.

It should be noted that these results do not include any bit errors which might be incurred during transmission.

REFERENCES

1. Linn, Shu and Daniel J. Costello, *Error Control Coding Fundamentals and Applications*, Prentiss-Hall, Inc., New Jersey, 1983.
2. Schauer, Anna L., *Analysis of a Proposed Orbital Maneuvering Vehicle (OMV) Synchronization Method*, Master's Thesis, Mississippi State University, May 1991.