# SEMI-ANNUAL STATUS REPORT

## Low-Complexity, High-Performance Bandwidth Efficient Coding and Coded Modulation Techniques for Satellite and Space Communications

NASA Grant Number NAG5-557

Principal Investigator:
Daniel J. Costello, Jr.

November 1993

N94-16507

# Bandwidth Efficient Coding: Theoretical Limits and Real Achievements

Interim Report
Error Control Techniques
for Satellite and Space Communications
NASA Grant NAG5-557
September 1993


Faculty
Dr. Daniel J. Costello, Jr.

Graduate Students
Servanne Couturier
Yannick Levy
Diane G. Mills
Lance C. Perez

Post-Doctoral Associate
Fu-Quan Wang

University of Notre Dame
Department of Electrical Engineering
Notre Dame, IN 46556

# Introduction

- In his seminal 1948 paper "*The Mathematical Theory of Communication*", Claude E. Shannon derived the "channel coding theorem" which gave an explicit upper bound, called the channel capacity, on the rate at which "information" could be transmitted reliably on a given communication channel.

- Shannon's result was an existence theorem and did not give specific codes to achieve the bound. Some skeptics have claimed that the dramatic performance improvements predicted by Shannon are not achievable in practice.

- The advances made in the area of coded modulation in the past decade have made communications engineers optimistic about the possibility of achieving or at least coming close to channel capacity. Here we consider this possibility in the light of current research results.

# Channel Capacity

With respect to coding and coded modulation, the most relevant of Shannon's results is the "noisy channel coding theorem for continuous channels with average power limitations."

This theorem states that for any transmission rate $R$ less than or equal to the channel capacity, $C$, there exists a coding scheme that achieves an arbitrarily small probability of error!

Conversely, if $R$ is greater than $C$, no coding scheme can achieve reliable communication, *regardless of complexity.*

Shannon then shows that the capacity, $C$, of a continuous additive white Gaussian noise (AWGN) channel with bandwidth $B$ and assuming Nyquist signaling is given by

$$C = B \log_2 \left(1 + \frac{E_s}{N_0}\right) \, bits/sec, \tag{1}$$

where $E_s$ is the average signal energy in each signaling interval $T$ and $N_0/2$ is the two sided noise power spectral density.

This bound represents the absolute best performance possible for a communication system on the AWGN channel.

# Restatement of the Capacity Bound

- Shannon's capacity bound can be put in a form more useful for the present discussion by introducing the parameter $\eta$, called spectral efficiency, to represent the average number of information bits transmitted per signaling interval.
- From Shannon's bound, it follows that

$$0 \leq R \leq C \ \ bits/sec,$$

and hence

$$0 \leq \eta \leq C/B \ \ bits/signal.$$

- Substituting the relation

$$E_s/N_0 = \eta E_b/N_0,$$

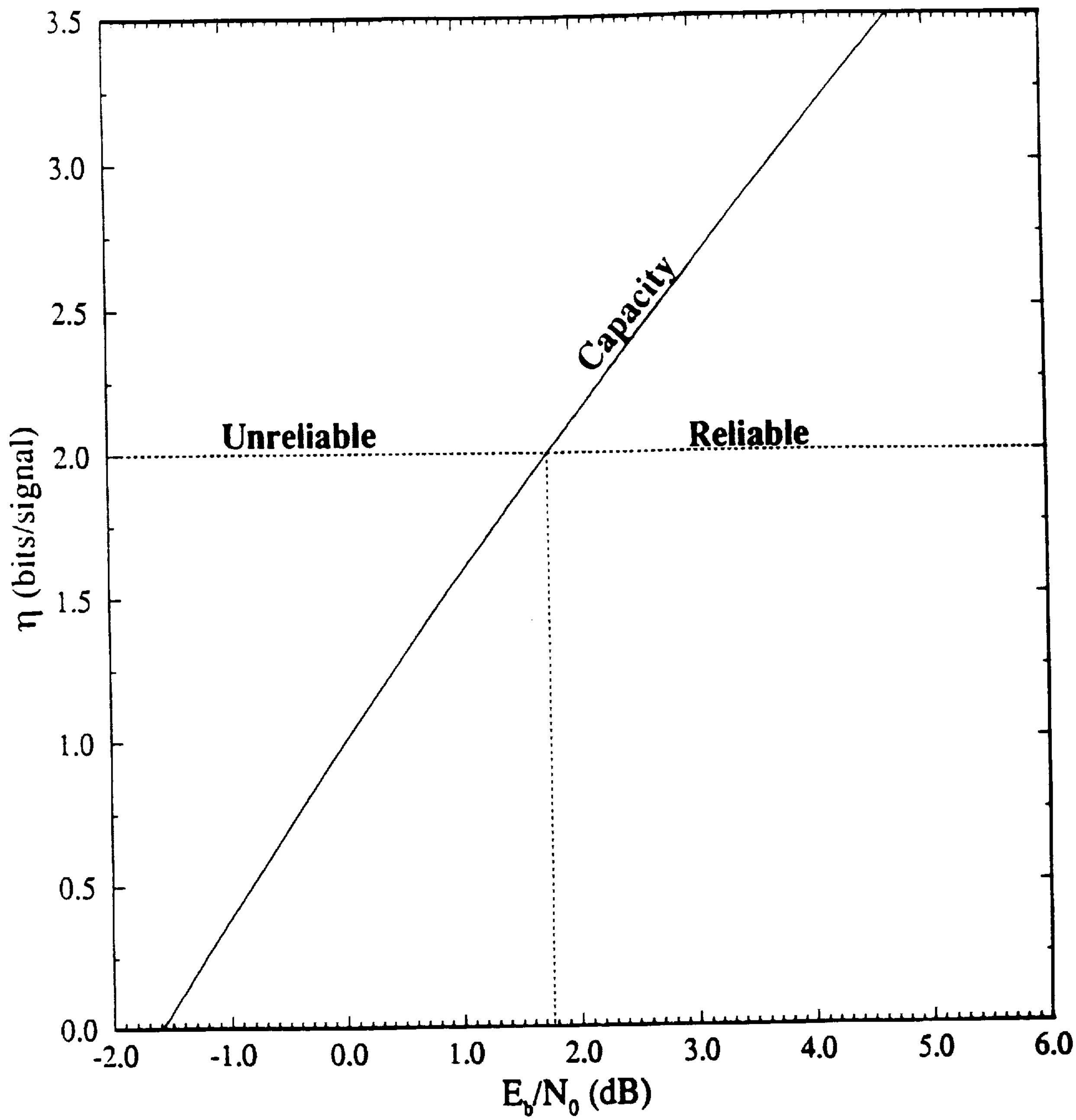
where $E_b$ is the average energy per information bit, into equation (1) and performing some minor manipulations yields

$$E_b/N_0 \geq \frac{2^\eta - 1}{\eta}, \tag{2}$$

which relates the spectral efficiency, $\eta$, to the signal-to-noise ratio (SNR), $E_b/N_0$.

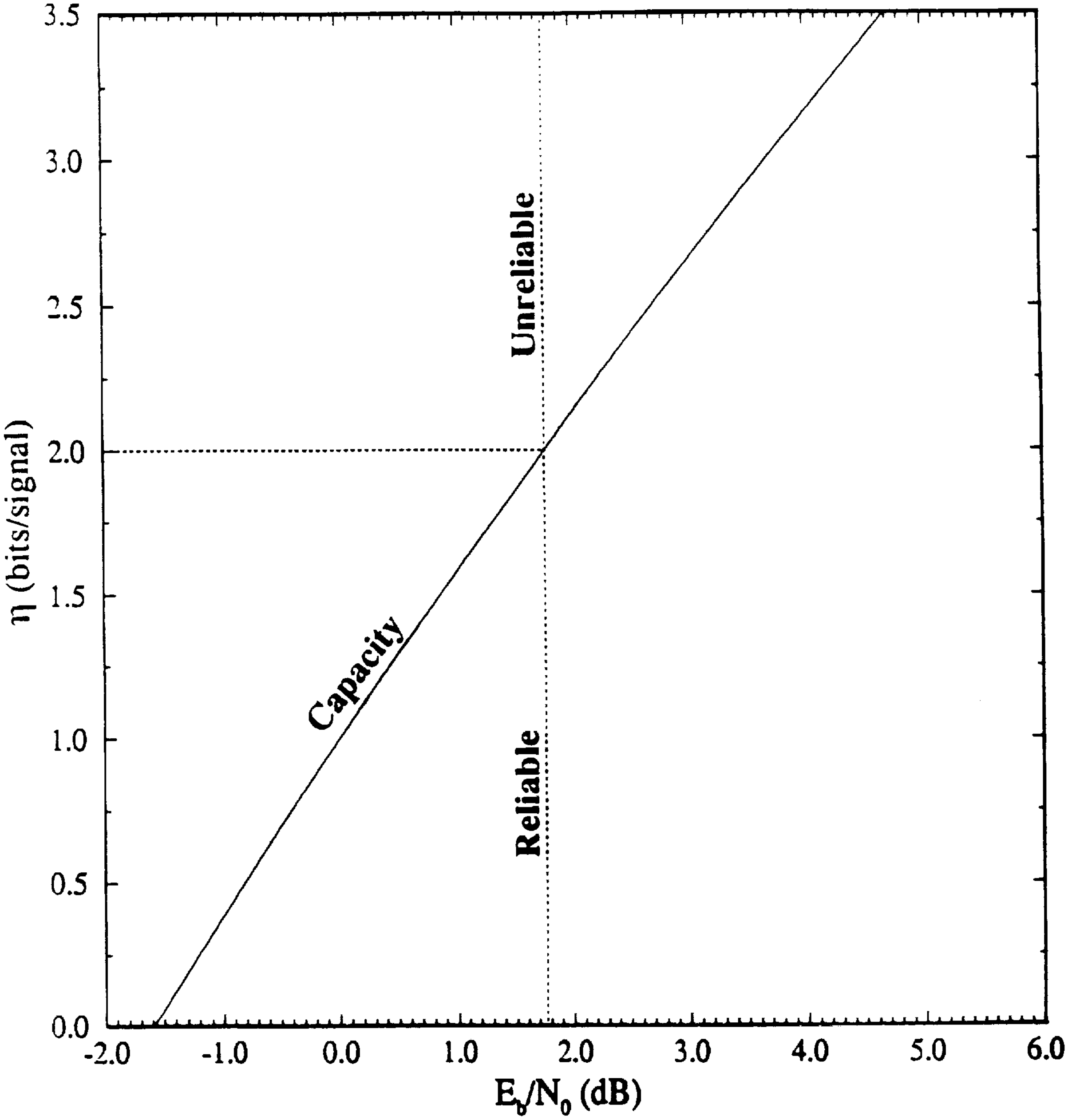
- The bound of equation (2) manifests the fundamental tradeoff between spectral efficiency and SNR. That is, increased spectral efficiency can be reliably achieved only with a corresponding increase in SNR.

# Interpretation of the Capacity Curve



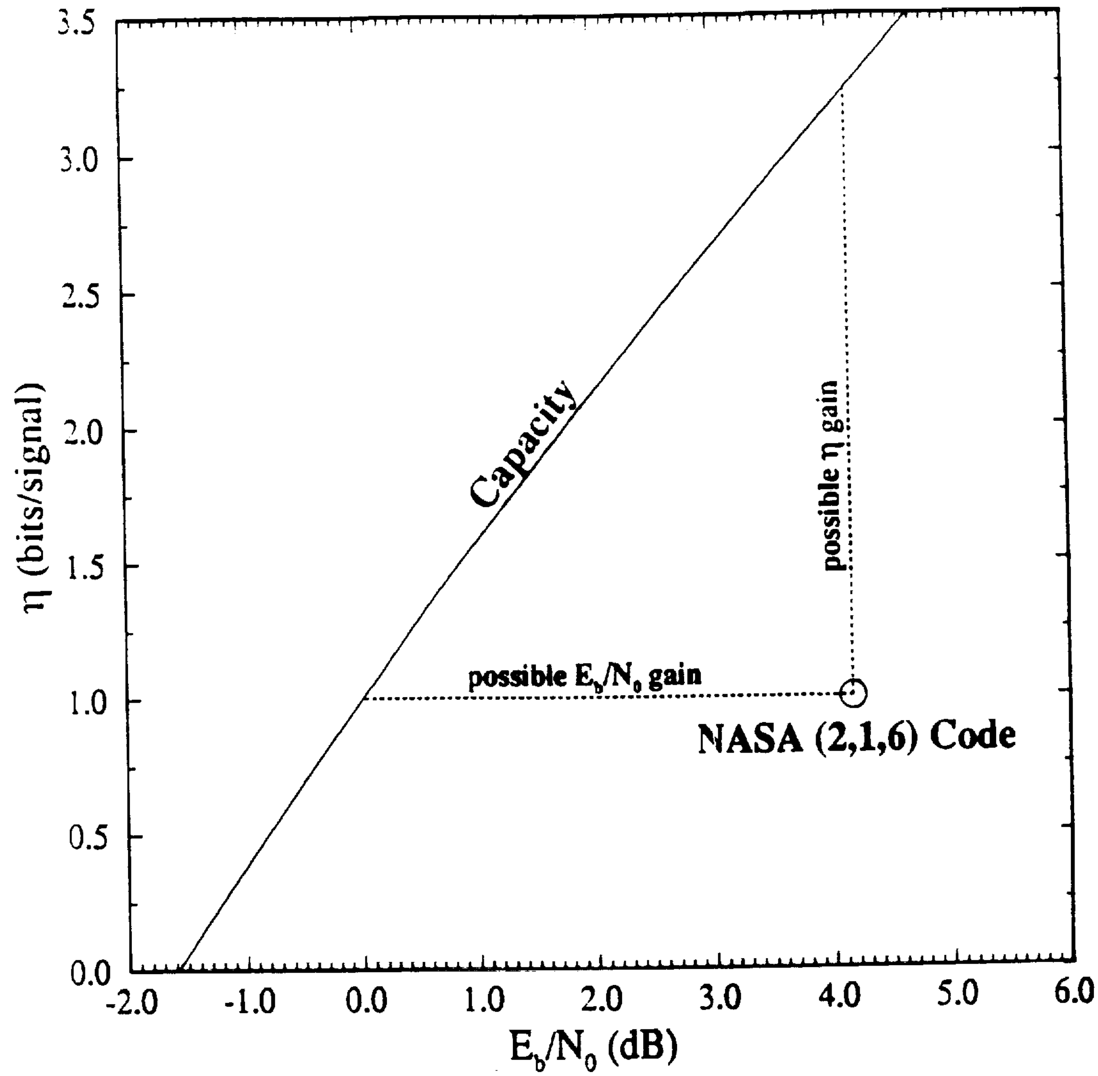

- **Shannon's bound gives the minimum signal-to-noise ratio (SNR) required to achieve a specific bandwidth efficiency with an arbitrarily small probability of error.**
- *Example*: **With $\eta = 2$ information bits per channel signal, there exists a coding scheme that operates reliably with an SNR of 1.76dB.**
- **Conversely, any coding scheme sending $\eta = 2$ information bits per signal with an SNR *less than* 1.76dB will be unreliable, *regardless of complexity*.**

# Interpretation of the Capacity Curve




- **Alternatively, Shannon's bound gives the maximum achievable spectral efficiency for a specific signal-to-noise ratio (SNR).**
- *Example*: **With an SNR of $E_b/N_0 = 1.76$db, there exists a coding scheme cabable of transmitting reliably with a spectral efficiency of $\eta = 2$ bits per signal.**
- **Conversely, any coding scheme operating with an SNR of $E_b/N_0 = 1.76$dB and attempting to transmit** *more than* **$\eta = 2$ bits per signal will be unrealiable** *regardless of complexity.*

# NASA (2,1,6) Convolutional Code




- Historically, most real communications systems operated well below capacity.

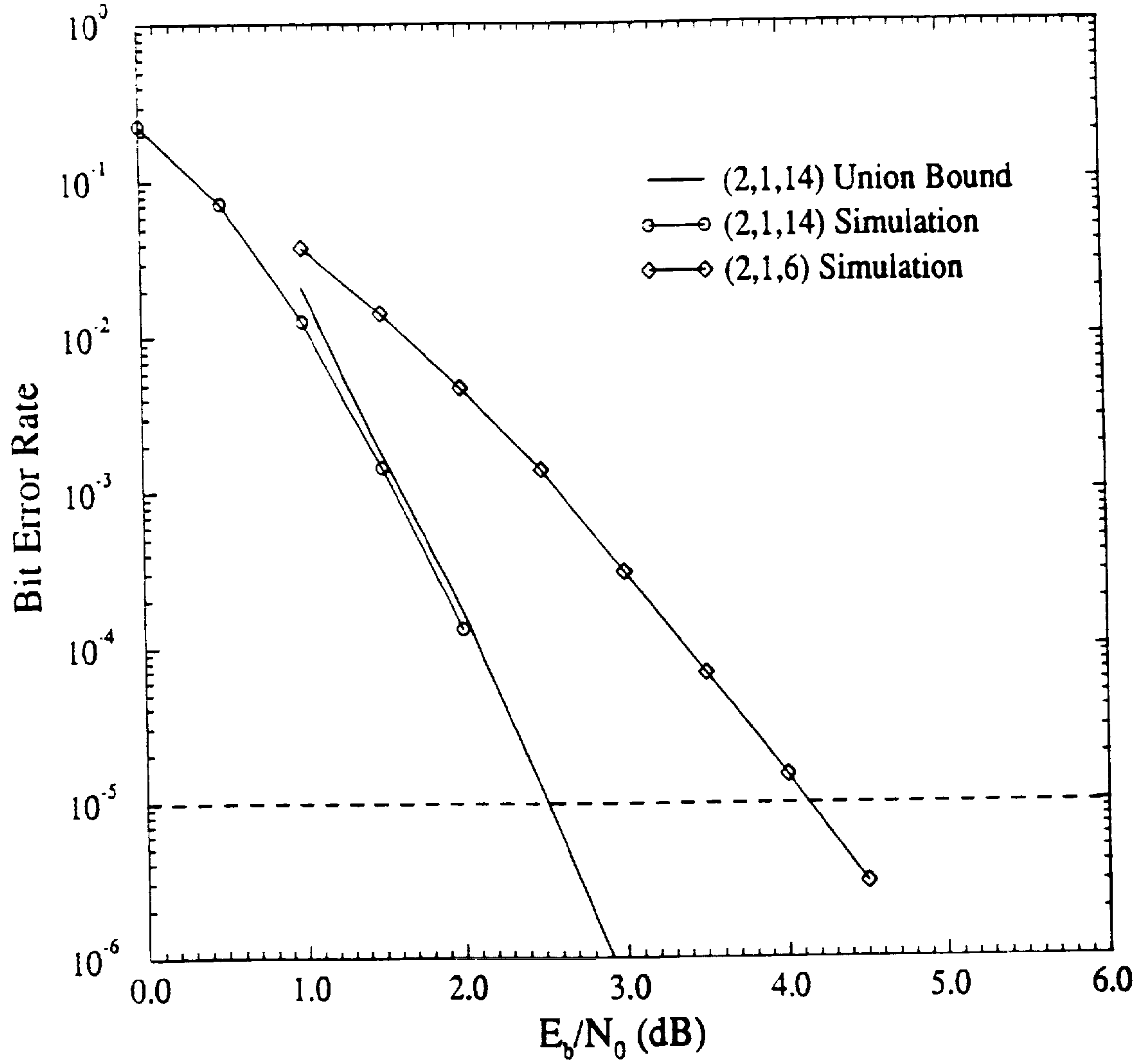
- *Example*: The NASA standard rate (2,1,6) convolutional code with QPSK modulation achieves a spectral efficiency of $\eta = 1$ bit/signal and requires a signal-to-noise ratio (SNR) of $E_b/N_0 =$ 4.15 dB to achieve error free ($10^{-5}$ bit error rate) communication.

- An ideal system operating with the same $E_b/N_0 = 4.15$dB can achieve error free communication with a spectral efficiency as high as $\eta = C = 3.235$ bits per signal.

**OR**

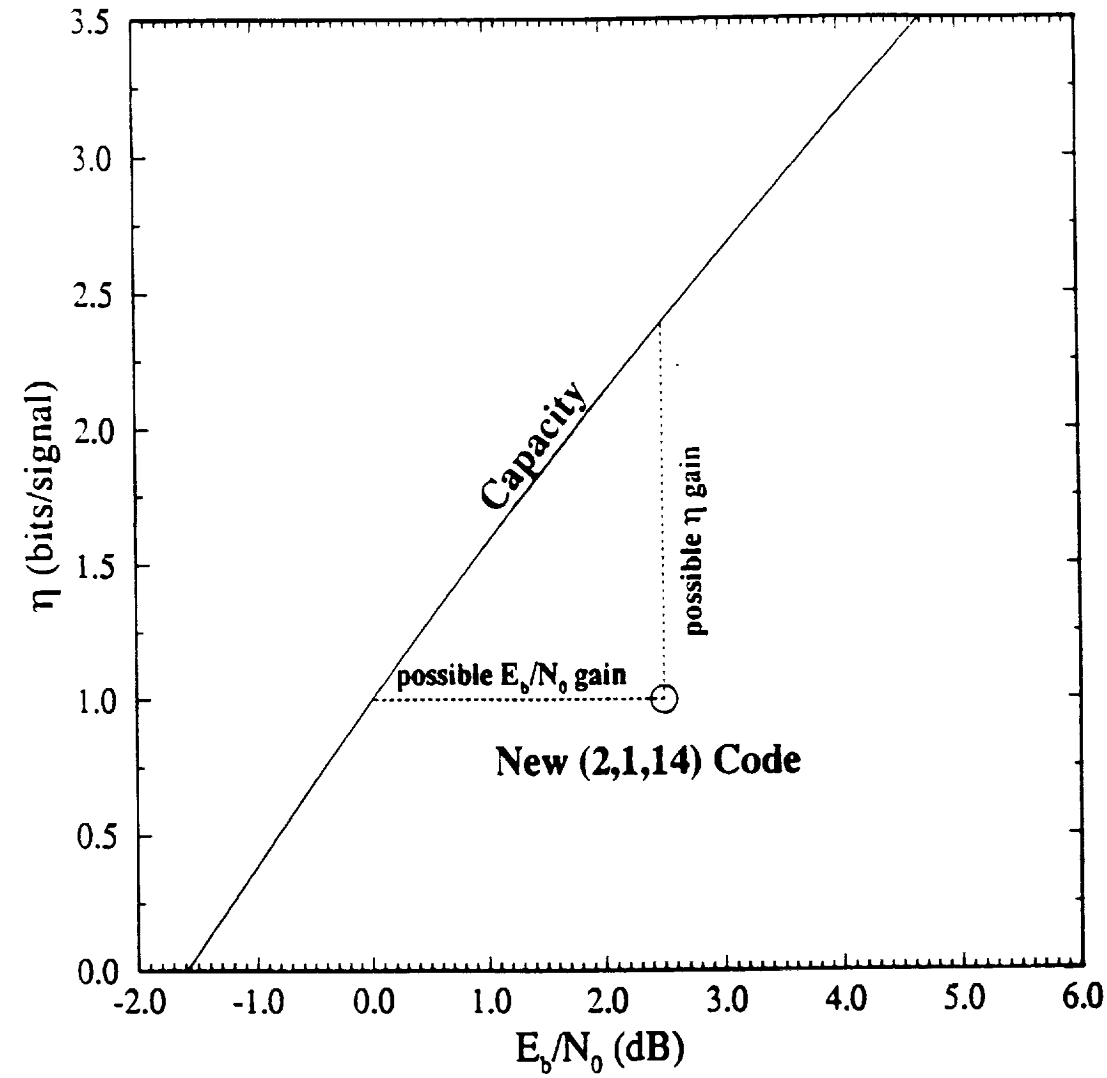
An ideal system operating with the same spectral efficiency of $\eta = 1$ bit per signal would require an SNR of only $E_b/N_0 = 0.0$ dB.

# A New Optimal (2,1,14) Convolutional Code




- A computer search has found the optimum distance spectrum (ODS) (2,1,14) convolutional code.
- This code has optimum minimum free Hamming distance, $d_{free} = 18$, and the smallest number of nearest neighbors, $N_{free} = 26$, of any constraint length 15 code.
- The performance of this code is 1.65 dB better than the (2,1,6) code, but is still 2.5dB away from capacity.

# Optimal (2,1,14) Convolutional Code




- The new optimal (2,1,14) convolutional code requires a signal-to-noise ratio (SNR) of $E_b/N_0 = 2.5$ dB to achieve error free ($10^{-5}$ bit error rate) communication.
- An ideal system operating with the same $E_b/N_0 = 2.5$dB can achieve error free communication with a spectral efficiency as high as $\eta = C = 2.4$ bits per signal.

**OR**

An ideal system operating with the same spectral efficiency of $\eta = 1$ bit per signal would require an SNR of only $E_b/N_0 = 0.0$ dB.

# Practical Bounds

- In real communication systems, there are many practical considerations that take precedence over Shannon's bound in design decisions.
- For example, satellite communication systems that use nonlinear travelling wave tube amplifiers (TWTA's) require constant envelope signaling such as $M$-ary phase shift keying (MPSK).
- Thus, even *if* Shannon's results firmly stated that capacity at a spectral efficiency of $\eta = 3$ bits per signal can be achieved with a (4,3,8) convolutional code using 16 QAM, it would not be feasible to do so on the TWTA satellite link.
- It therefore seems reasonable to ask what is the minimum SNR required to achieve reliable communication, *given* a particular modulation scheme and a spectral efficiency, $\eta$.

# A Signal Specific Bound

- For the discrete input, continuous output, memoryless AWGN channel with $M$-ary one dimensional amplitude modulation (AM) or two dimensional (PSK, QAM) modulation and equiprobable signaling, the capacity bound becomes

$$\eta^* = \log_2(M) - \frac{1}{M}\sum_{i=0}^{M-1} E\left\{\log_2 \sum_{j=0}^{M-1} \exp\left[\frac{|a^i + n - a^j|^2 - |n|^2}{N_0}\right]\right\}, \quad (3)$$

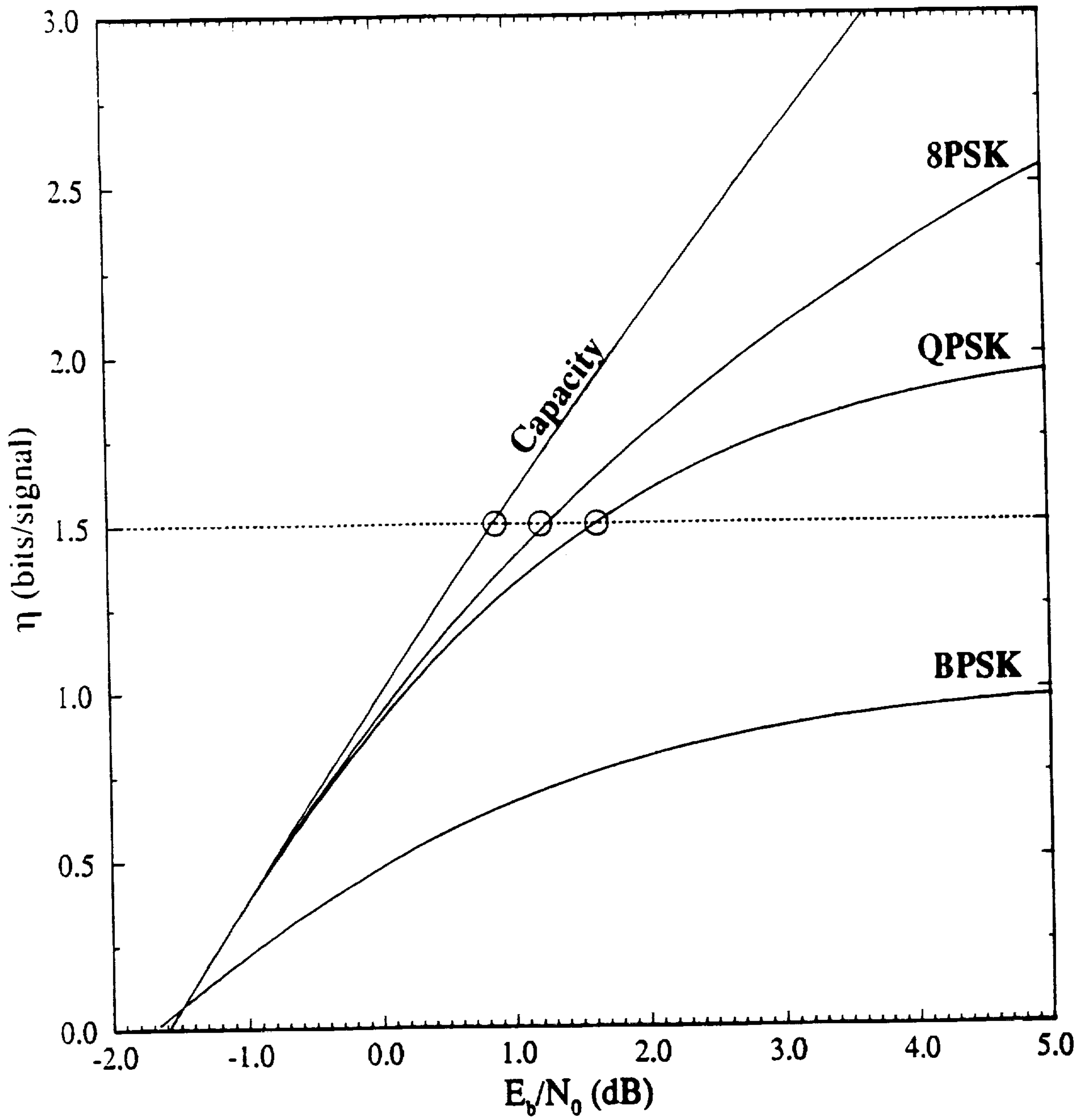
- Here

$$\left\{a^j, j = 0, 1, \ldots M-1\right\} \quad (4)$$

is an $M$-ary modulations set, $a^j$ is a channel signal, $n$ is a Gaussian distributed noise random variable with mean 0 and variance $N_0/2$, and $E$ is the expectation operator.

- For a specified signaling method and spectral efficiency, this bound can be used to compute the minimum SNR required to achieve reliable communication.

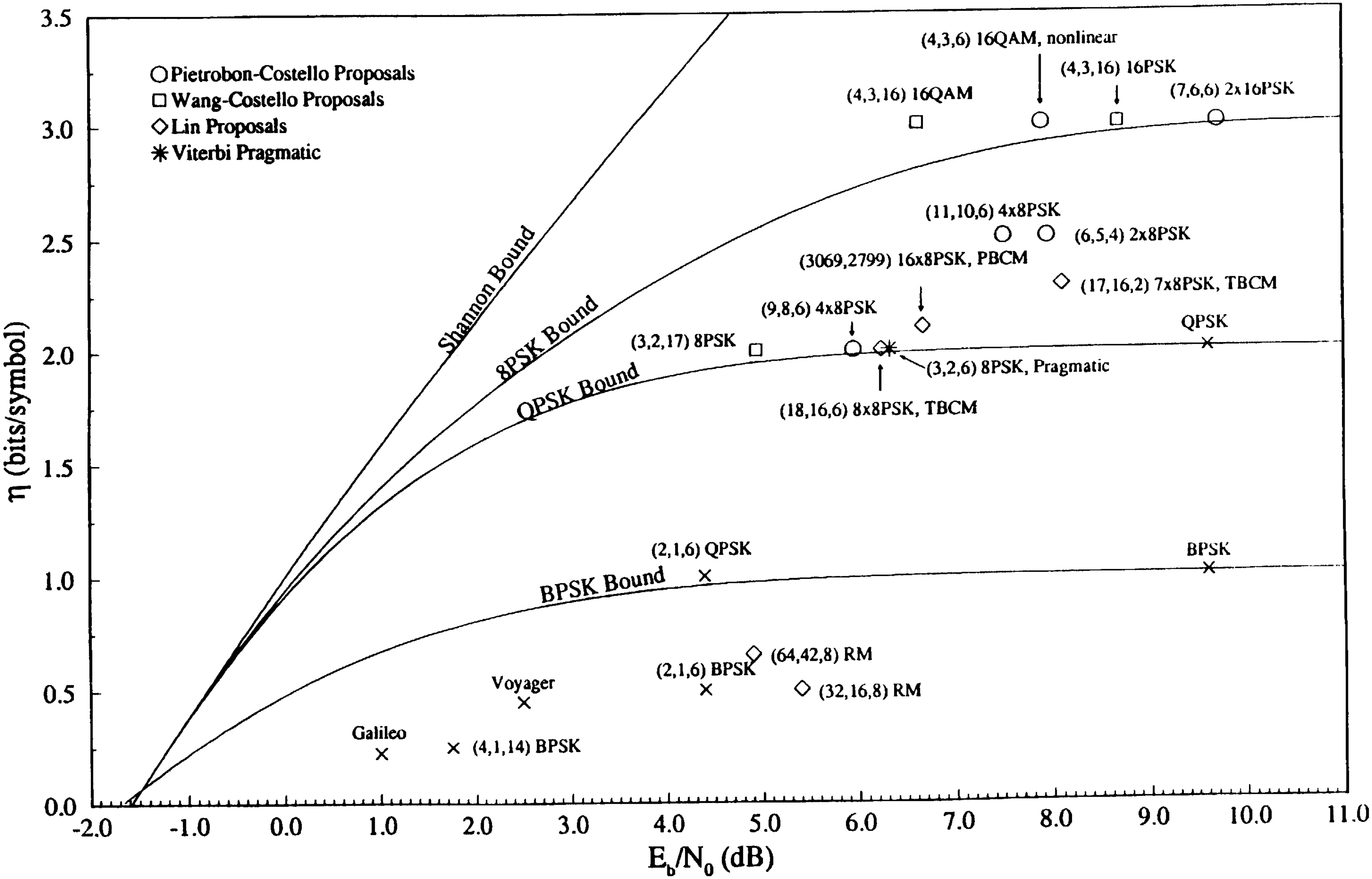
# Interpretation of the Signal Specific Bound




- To send $\eta = 1.5$ information bits per signaling interval, an ideal system *using QPSK modulation* requires a minimum SNR of $E_b/N_0 \approx$ 1.64dB. This is 0.76 dB more than an ideal system without any modulation constraints.
- To send $\eta = 1.5$ information bits per signaling interval, an ideal system *using 8PSK modulation* requires a minimum SNR of $E_b/N_0 \approx$ 1.22dB. This is 0.34 dB more than an ideal system without any modulation constraints.

Plot of Spectral Efficiency, $\eta$, versus $E_b/N_0$ (dB)

Inner Codes at a Bit Error Rate of $10^{-5}$ with Soft Decision Decoding



# Notes About the Graph

All results are with soft decision decoders.

## 1 NASA Codes: denoted by $\times$

**Galileo:** This is the concatenated code used on the Galileo probe.

**(4,1,14) BPSK** The is the (4,1,14) convolutional code developed by JPL.

**Voyager:** This is the concatenated code used on the Voyager probe.

**(2,1,6) BPSK:** This is the NASA standard (2,1,6) convolutional code with BPSK modulation.

**(2,1,6) QPSK:** This is the NASA standard (2,1,6) convolutional code with QPSK modulation.

## 2 Pietrobon-Costello Codes: denoted by $\bigcirc$

**(6,5,4) 2x8PSK:** This is a rate 5/6, 16 state, trellis code using 2x8PSK modulation with Viterbi decoding. It has a spectral efficiency of $\eta = 2.5$ bits/signal and is 45° rotationally invariant. A Viterbi decoder for this code has been built by Steven Pietrobon and tested at New Mexico State.

**(9,8,6) 4x8PSK:** This is a rate 8/9, 64 state, trellis code using 4x8PSK modulation with Viterbi decoding. It has a spectral efficiency of $\eta = 2.0$ bits/signal and is 45° rotationally invariant.

**(11,10,6) 4x8PSK:** This is a rate 10/11, 64 state, trellis code using 4x8PSK modulation with Viterbi decoding. It has a spectral efficiency of $\eta = 2.5$ bits/signal and is 45° rotationally invariant.

**(7,6,6) 2x16PSK:** This is a rate 6/7, 64 state, trellis code using 2x16PSK modulation with Viterbi decoding. It has a spectral efficiency of $\eta = 3.0$ bits/signal and is 45° rotationally invariant.

**(4,3,6) 16QAM, nonlinear:** This is a nonlinear rate 3/4, 16 state, trellis code using 16QAM modulation with Viterbi decoding. It has a spectral efficiency of $\eta = 3.0$ bits/signal and is 90° rotationally invariant.

## 3 Wang-Costello Codes: denoted by $\square$

**(3,2,17)8PSK:** This is a rate 2/3, memory 17, trellis code using 8PSK modulation and sequential decoding with a modified Fano algorithm. It has a spectral efficiency of $\eta = 2.0$ bits/signal and is 180° rotationally invariant.

**(4,3,16) 16PSK:** This a rate 3/4, memory 16, trellis code using 16PSK modulation and sequential decoding with a modified Fano algorithm. It has a spectral efficiency of $\eta = 3.0$ bits/signal and is 180° rotationally invariant.

**(4,3,16) 16QAM:** This is a rate 3/4, memory 16, trellis code using 16QAM modulation and sequential decoding with a modified Fano algorithm. It has a spectral efficiency of $\eta = 3.0$ bits/signal and is 180° rotationally invariant.

## 4 Lin Codes: denoted by $\diamond$

**(32,16,8) RM:** This is a rate 16/32=0.5, 64 state Reed-Muller code using BPSK modulation and Viterbi decoding.

**(64,42,8) RM:** This is a rate 42/64=0.656, 1024 state Reed-Muller code using BPSK modulation and Viterbi decoding.

**(17,16,2) 7x8PSK, TBCM:** This is rate 16/17, 4 state, block coded modulation scheme using 7x8PSK modulation with Viterbi decoding. It has a spectral efficiency of $\eta =$ 2.286 bits/signal and is 180° rotationally invariant.

**(18,16,6) 8x8PSK, TBCM:** This is a 2-level trellis code using 8x8PSK modulation. The first level has 64 states and is decoded with a Viterbi decoder. The second level has 8 states and is deocded with a Viterbi decoder. It has a spectral efficiency of $\eta = 2.0$ bits/symbol and is 180° rotationally invariant.

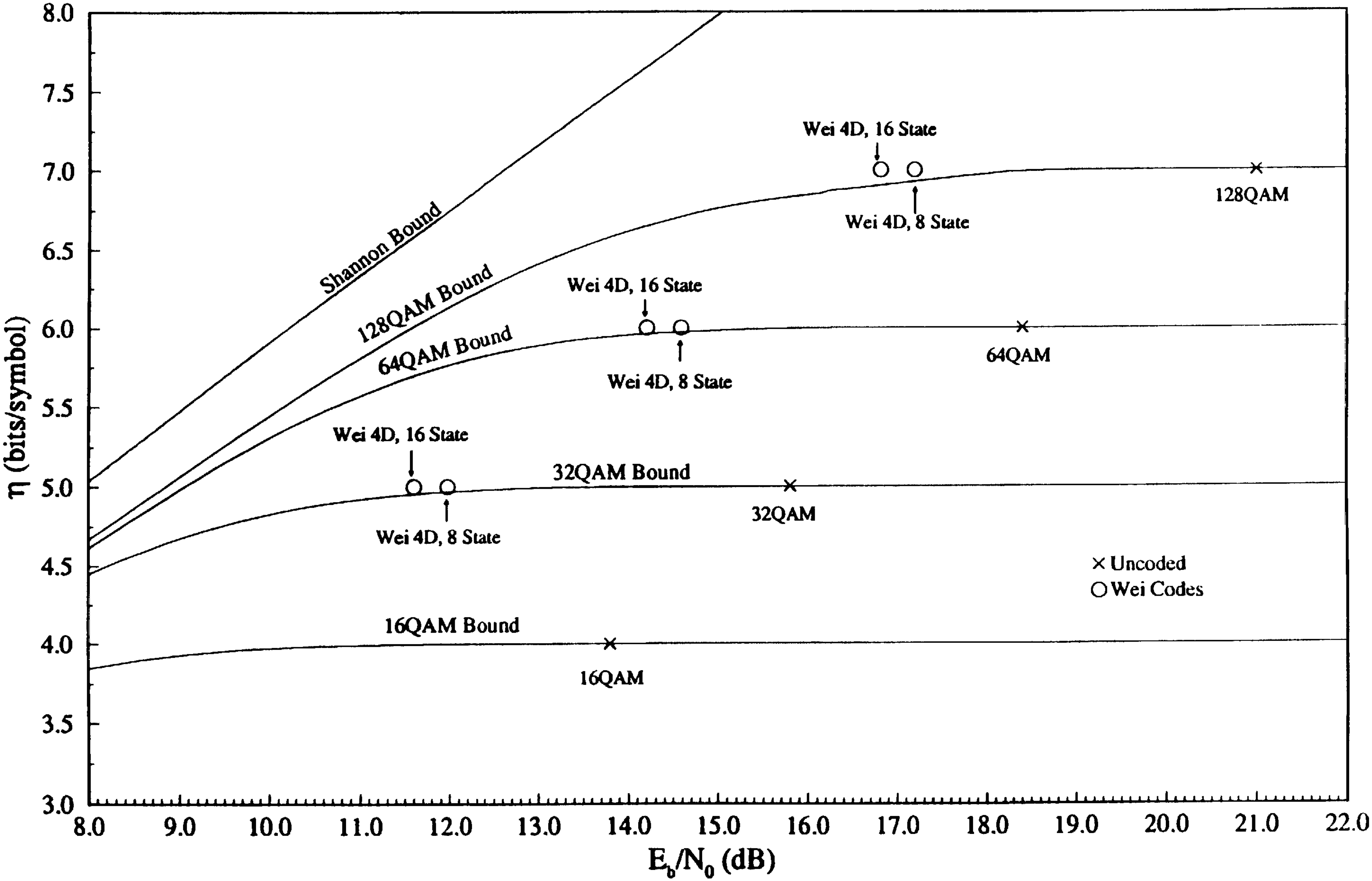
**(3069,2799) 16x8PSK, PBCM:** This is a 3x3 product block coded modulation scheme. The horizontal codes are BCH codes and the vertical code is a 3-level block code. It is decoded using suboptimal multi-stage decoding. It has a spectral efficiency of $\eta = 2.1$ bits/signal and is 45° rotationally invariant.

## 5 Viterbi Pragmatic Code: denoted by $*$

**(3,2,6) 8PSK, Pragmatic:** This is a rate 2/3, 64 state, trellis code using 8PSK modulation and Viterbi decoding. It uses the NASA standard (2,1,6) convolutional code as its basis and is suboptimal. It can be decoded using essentially the same Viterbi decoding chip that is used to decode the NASA standard convolutional code.

Plot of Spectral Efficiency, $\eta$, versus $E_b/N_0$ (dB)

Inner Codes at a Bit Error Rate of $10^{-5}$ with Soft Decision Decoding



# Notes About the Graph

All results are with soft decision decoders.

## 1 Uncoded Systems: denoted by $\times$

**16QAM:** Performance of uncoded 16QAM with a spectral efficiency of $\eta = 4$ bits/symbol from simulation results.

**32QAM:** Performance of uncoded 32QAM with a spectral efficiency of $\eta = 5$ bits/symbol from simulation results.

**64QAM:** Performance of uncoded 64QAM with a spectral efficiency of $\eta = 6$ bits/symbol from simulation results.

**128QAM:** Performance of uncoded 128QAM with a spectral efficiency of $\eta = 7$ bits/symbol from simulation results.

## 2 Wei Codes: denoted by $\bigcirc$

**Wei 4D, 8 state:** This is Wei's 8 state code using a 4-dimensional constellation (of any size). Performance taken from "Coset Codes - Part 1: Introduction and Geometrical Classification," by G. David Forney. The performance was estimated taking into account the minimum squared Euclidean distance and the number of nearest neighbors. Thus, this point shows effective coding gain at a bit-error-rate (BER) of $10^{-5}$.

**Wei 4D, 16 state:** This is Wei's 16 state code using a 4-dimensional constellation (of any size). Performance taken from "Coset Codes - Part 1: Introduction and Geometrical Classification," by G. David Forney. The performance was estimated taking into account the minimum squared Euclidean distance and the number of nearest neighbors. Thus, this point shows effective coding gain at a bit-error-rate (BER) of $10^{-5}$.
This code with a 2x192QAM constellation is being considered by CCITT for the V.FAST ultimate modem standard.

N94-16508

# A Simulation Study of the Performance of the NASA (2,1,6) Convolutional Code on RFI/Burst Channels *

Lance C. Perez
Daniel J. Costello, Jr.

Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

October 14, 1993

*This work was supported by NASA Grants NAG5-557 and NGT-70109.

# 1 Introduction

In an earlier report [1], the LINKABIT Corporation studied the performance of the (2,1,6) convolutional code on the radio frequency interference (RFI)/burst channel using analytical methods. Using an $R_0$ analysis, the report concluded that channel interleaving was essential to achieving reliable performance. In this report, Monte Carlo simulation techniques are used to study the performance of the (2,1,6) convolutional code on the RFI/burst channel in more depth.

The basic system model under consideration is shown in Figure 1. The (2,1,6) convolutional code is the NASA standard code with generators

$$\begin{aligned} g^1 &= 1 + D^2 + D^3 + D^5 + D^6 \\ g^2 &= 1 + D + D^2 + D^3 + D^6 \end{aligned}$$

and $d_{free} = 10$. The channel interleaver is of the convolutional or periodic type first described in [2]. The binary output of the channel interleaver is transmitted across the channel using binary phase shift keying (BPSK) modulation. The transmitted symbols are corrupted by an RFI/burst channel consisting of a combination of additive white Gaussian noise (AWGN) and RFI pulses. At the receiver, a soft-decision Viterbi decoder with no quantization and variable truncation length is used to decode the deinterleaved sequence.

# 2 RFI Channel Models

The RFI/burst channel takes on a variety of forms depending on the characteristics of the RFI pulse and the steps taken to combat it. In this report, the two models described in [1] are used. These models represent the two extremes of the RFI/burst channel.

In the first model, the RFI pulse is assumed to saturate the satellite's transponder to the extent that BPSK symbols at the output of the channel occur with equal probability during the RFI pulse. Thus, the channel output is independent of the channel input during the RFI pulse. This type of RFI can be modeled as a binary symmetric channel (BSC) with crossover probability of 1/2. When an RFI pulse is present, the overall channel is then a cascade of the BSC and the AWGN channel. This channel is called the RFI/burst saturation channel and represents the worst case RFI/burst channel. It is shown in block diagram form in Figure 2. When an RFI pulse is not present, the channel is simply an AWGN channel.

In the second model, it is assumed that RFI pulses can be detected and the satellite saturation then prevented or blanked. In this case, the RFI can be modeled as a binary erasure channel (BEC) with an erasure probability of 1. When an RFI pulse is present, the overall channel is then a cascade of the BEC and the AWGN channel. This channel is called the RFI/burst blank channel and represents the best case of the RFI/burst channel. It is shown in block diagram form in Figure 3. When an RFI pulse is not present, the channel is simply an AWGN channel.

# 3 Simulation Results

For the simulations performed in this study, the channel interleaver and RFI/burst channels were not simulated directly. Instead, empirical data obtained from NASA was used to model the combined convolutional interleaver and channel. This was done in order to address specific questions concerning the performance of the system shown in Figure 1. It is a simple matter to simulate the interleaver and channel in a more direct manner.

The empirical data showed that an RFI pulse with a length of approximately 240 consecutive channel symbols resulted in 1 in 15 symbols out of the convolutional deinterleaver being corrupted by the RFI channel. Similarly, an RFI pulse with a length of approximately 360 consecutive channel symbols resulted in 1 in 10 symbols out of the convolutional deinterleaver being corrupted by the RFI channel. Using these observations, the interleaver, RFI/burst channel, and deinteleaver were combined into a single superchannel consisting of an AWGN channel in cascade with a periodic RFI/burst channel. Thus, to simulate the 240 symbol and 360 symbol RFI pulses the period of the superchannel was set to 15 and 10, respectively. The RFI/burst saturation model and the RFI/burst blank model were both used as the periodic RFI/burst channel.

Figure 4 shows the simulated bit error rate (BER) performance of the (2,1,6) convolutional code on the RFI/burst blank superchannel compared to simulation results of the (2,1,6) code on a pure AWGN channel. Decoder truncation lengths of $\tau = 30$ branches and $\tau = 26$ branches were considered.

The RFI/burst superchannel with a period of 15 symbols resulted in a loss of $\approx 0.5$dB at a BER of $10^{-5}$ compared to the AWGN results. A period of 10 symbols resulted in a loss of $\approx 0.8$dB at a BER of $10^{-5}$. Changing the truncation length had virtually no consequences on the performance of the (2,1,6) code on the RFI/burst blank channel *relative* to the performance on the AWGN channel.

If the (2,1,6) code and the system of Figure 1 are to be used in a concatenated system, the SER performance out of the inner docoder is more significant than the BER. Figure 5 shows the simulated 8-bit symbol error rate (SER) performance of the (2,1,6) code under the same channel conditions that were used in Figure 4. Eight bit symbols were used in order to be compatible with the standard (255,223) Reed-Solomon outer code. The SER performance of the (2,1,6) code degrades in the same manner as the BER.

Figure 6 shows the simulated bit error rate (BER) performance of the (2,1,6) code on the RFI/burst saturation superchannel compared to simulation results of the (2,1,6) code on a pure AWGN channel. As expected, the saturation channel is much more destructive than the blanking channel. With a decoder truncation length of $\tau = 30$, a period of 15 symbols resulted in a loss of $\approx 3.7$dB at a BER of $10^{-5}$. However, with a decoder truncation length of $\tau = 26$, a period of 15 symbols resulted in a loss of $\approx 4.4$dB. The effect of the decoder truncation length was even more significant on the RFI/burst saturation channel when the period was 10 symbols. In this case, there was a loss of $\approx 6.2$dB with $\tau = 30$ and a loss of $\approx 8.2$dB with $\tau = 26$. Thus, reducing the truncation length caused a performance loss of 2.0dB!

In Figure 7 the 8-bit SER performance is shown under the same channel conditions that were used in Figure 6. The SER performance degrades in the same manner as the BER. In particular, the truncation length has a significant effect on performance.

# 4 Conclusions

The simulation results in this report are consistent with the analytical results in [1]. The RFI/burst channel is significantly worse than a pure AWGN channel. It is also clear that the ability to detect and blank RFI pulses greatly enhances performance.

It is unclear at this point why the performance of the (2,1,6) code on the RFI/burst saturation is so sensitive to the decoder truncation length. Simulation results for the AWGN channel, shown in Figure 8, demonstrate that the performance of the (2,1,6) code is fairly robust even with a truncation lenth of $\tau = 24$. When the truncation length is reduced to $\tau = 18$, the performance is reduced considerably, but still does not exhibit the divergent behavior evident on the RFI/burst saturation channel. The cause of this phenomenon is currently being investigated.

# References


[1] J. P. Odenwalder and A. J. Viterbi, "Final Report on RFI/Coding Sensitivity Analysis for Tracking and Data Relay Satellite System (TDRSS)," Technical Report to ORI, Inc., 13 July 1978.

[2] J. L. Ramsey, "Realization of Optimum Interleavers," *IEEE Trans. on Information Theory*, **IT-16**, pp. 338–345, May 1970.

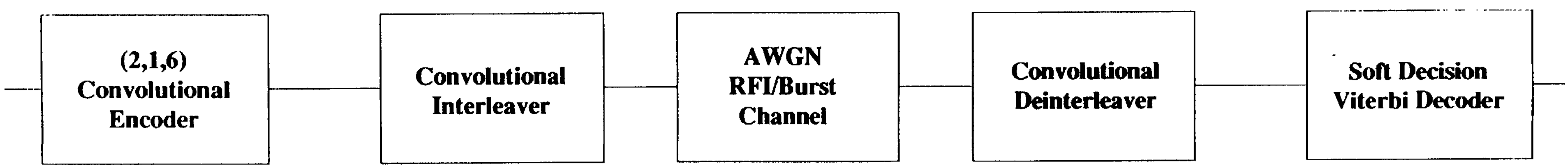



**Figure 1: System Model with the (2,1,6) Convolutional Code**

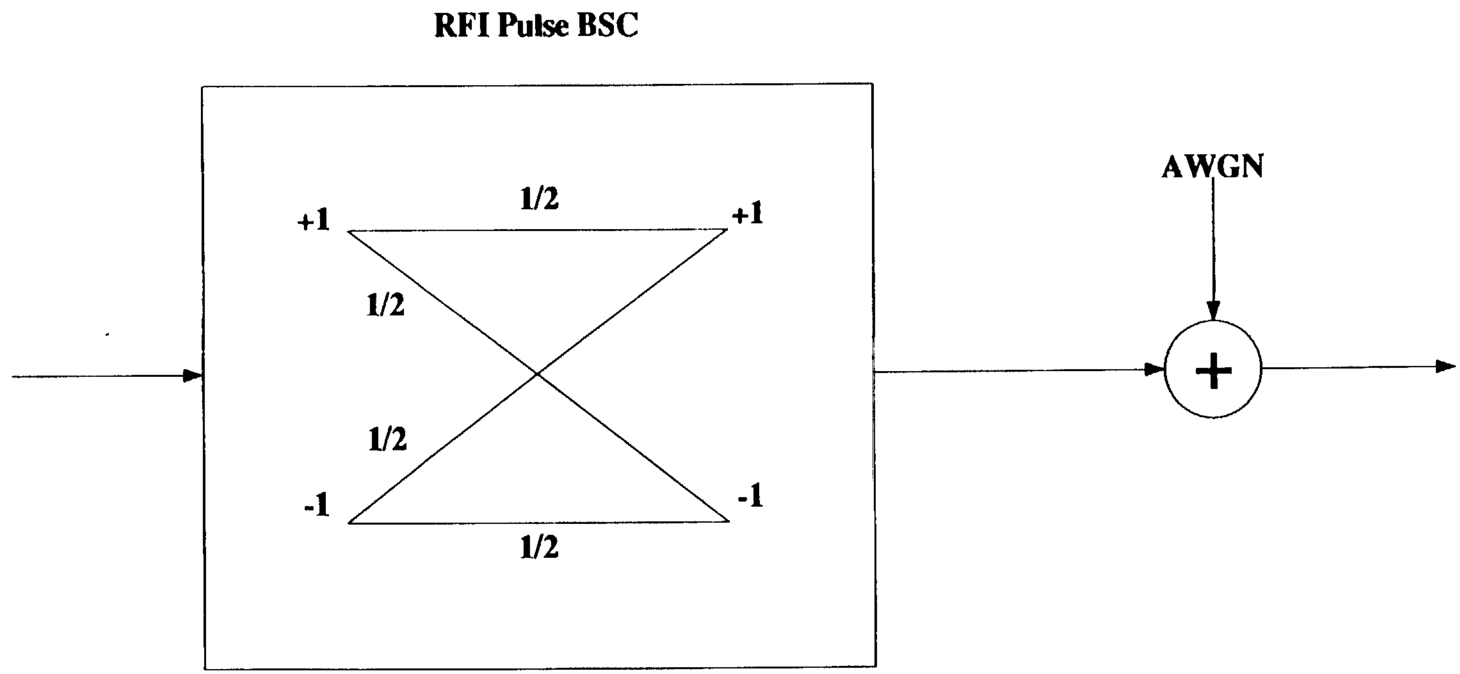



Figure 2: RFI/ Burst Saturation Channel Model

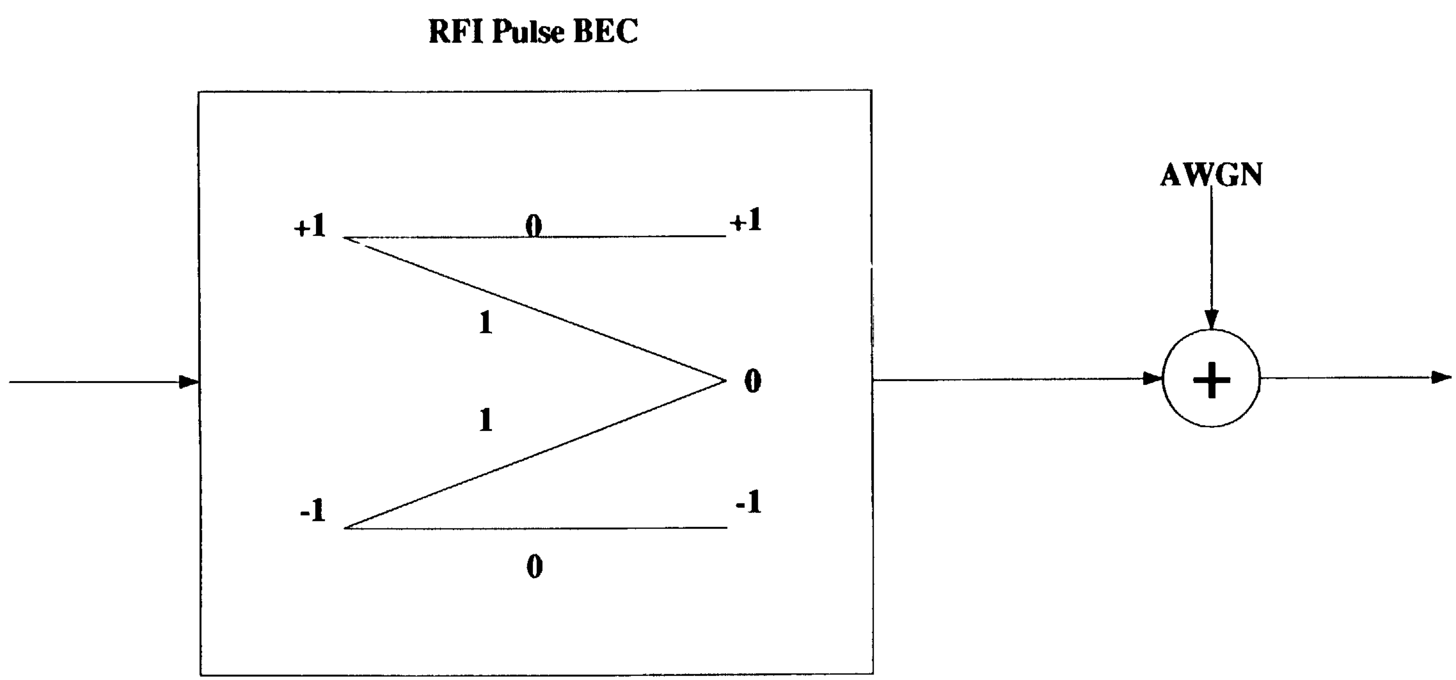



Figure 3: RFI/ Burst Blank Channel Model

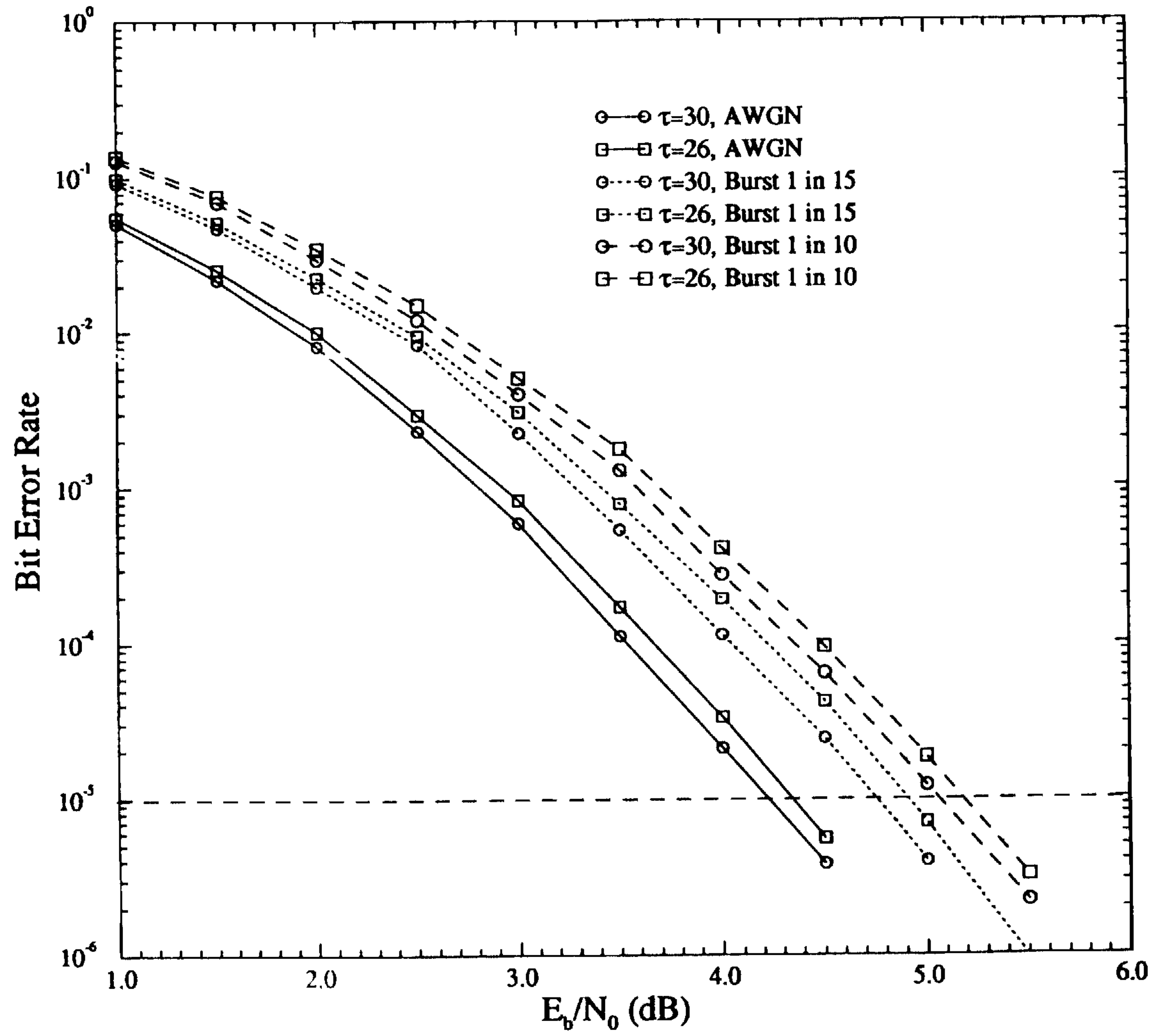



**Figure 4: BER Performance of the (2,1,6) Convolutional Code on the RFI/Burst Blank Channel**

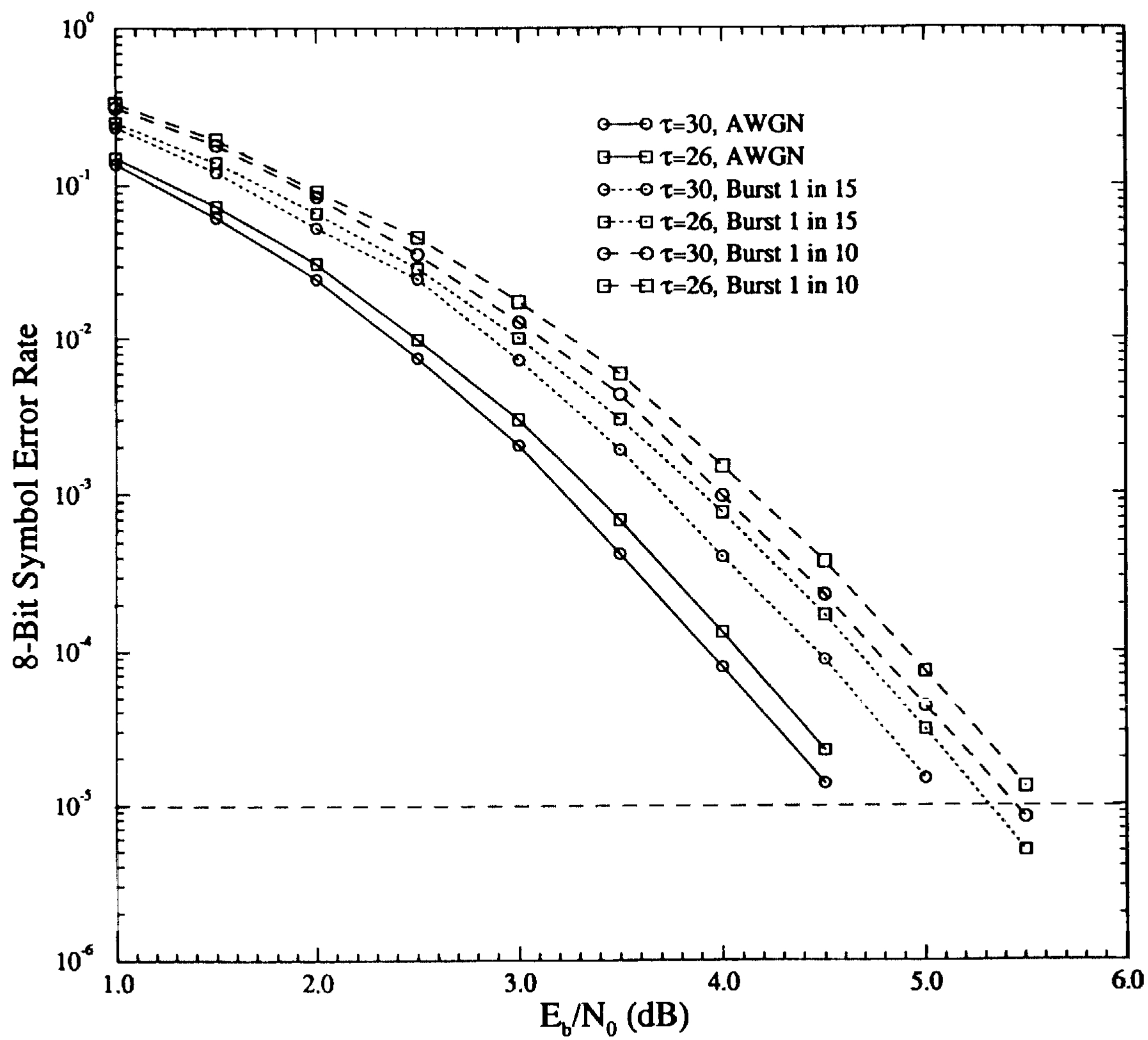



**Figure 5: SER Performance of the (2,1,6) Convolutional Code on the RFI/Burst Blank Channel**

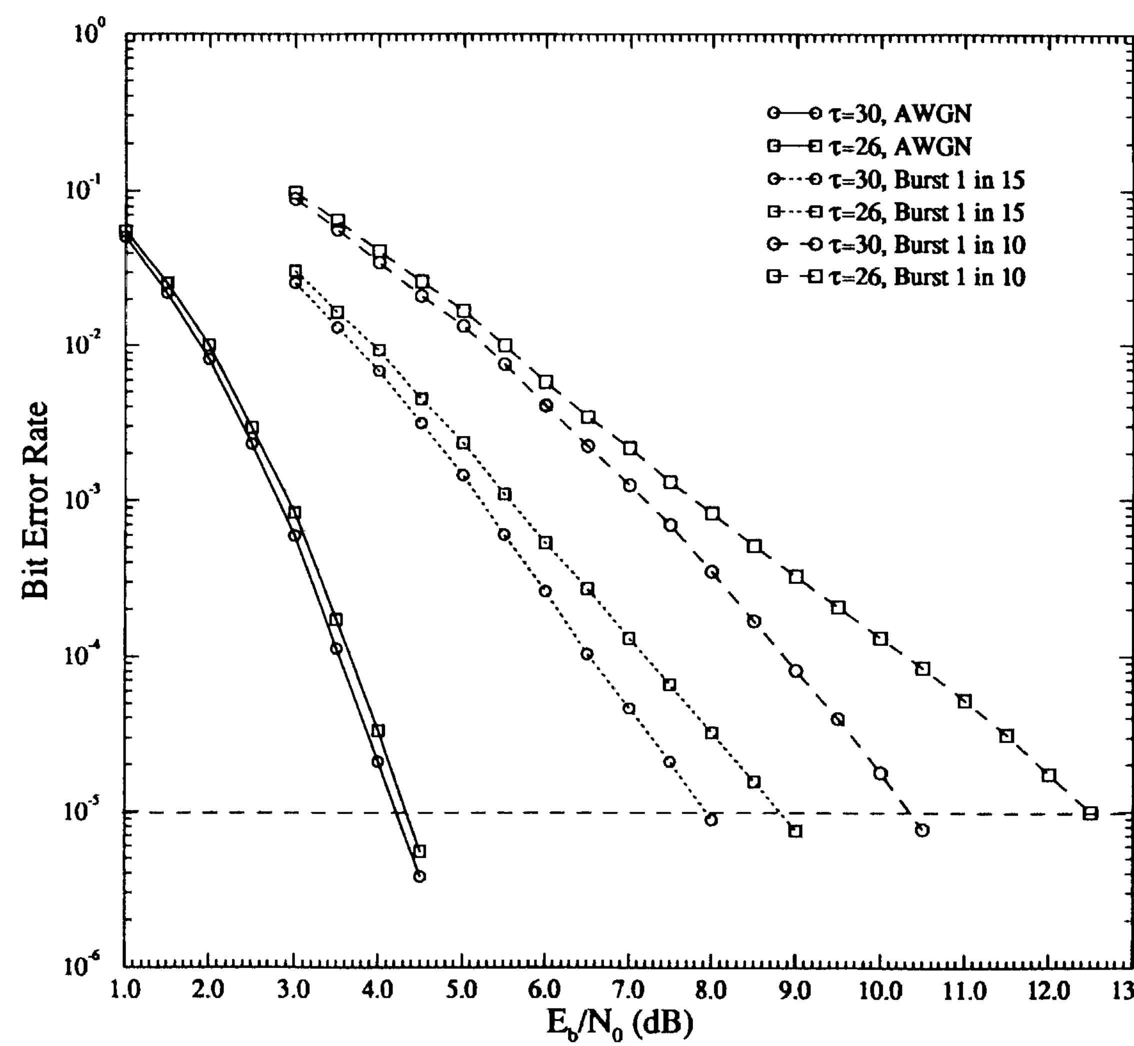



**Figure 6: BER Performance of the (2,1,6) Convolutional Code on the RFI/Burst Saturation Channel**

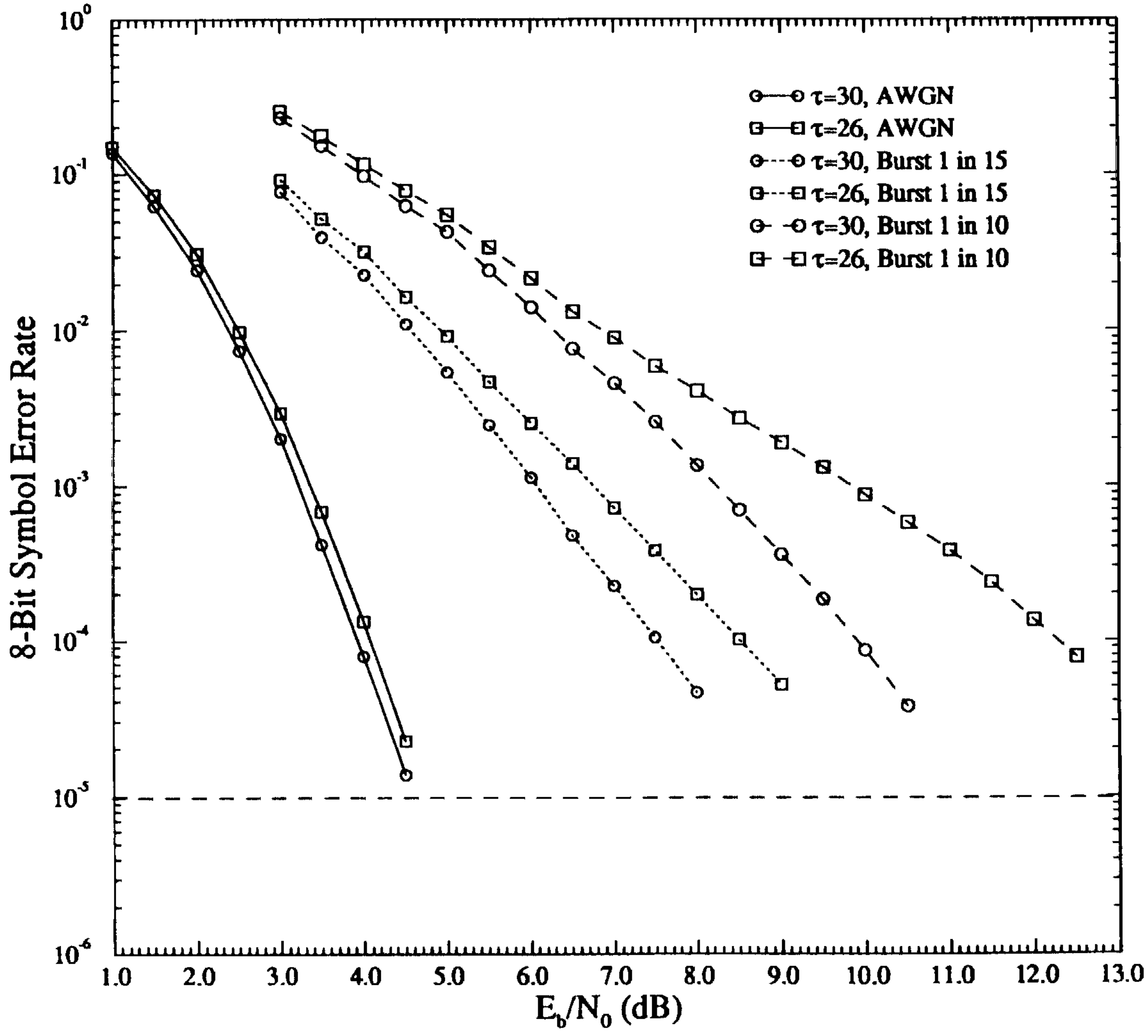



**Figure 7: SER Performance of the (2,1,6) Convolutional Code on the RFI/Burst Saturation Channel**

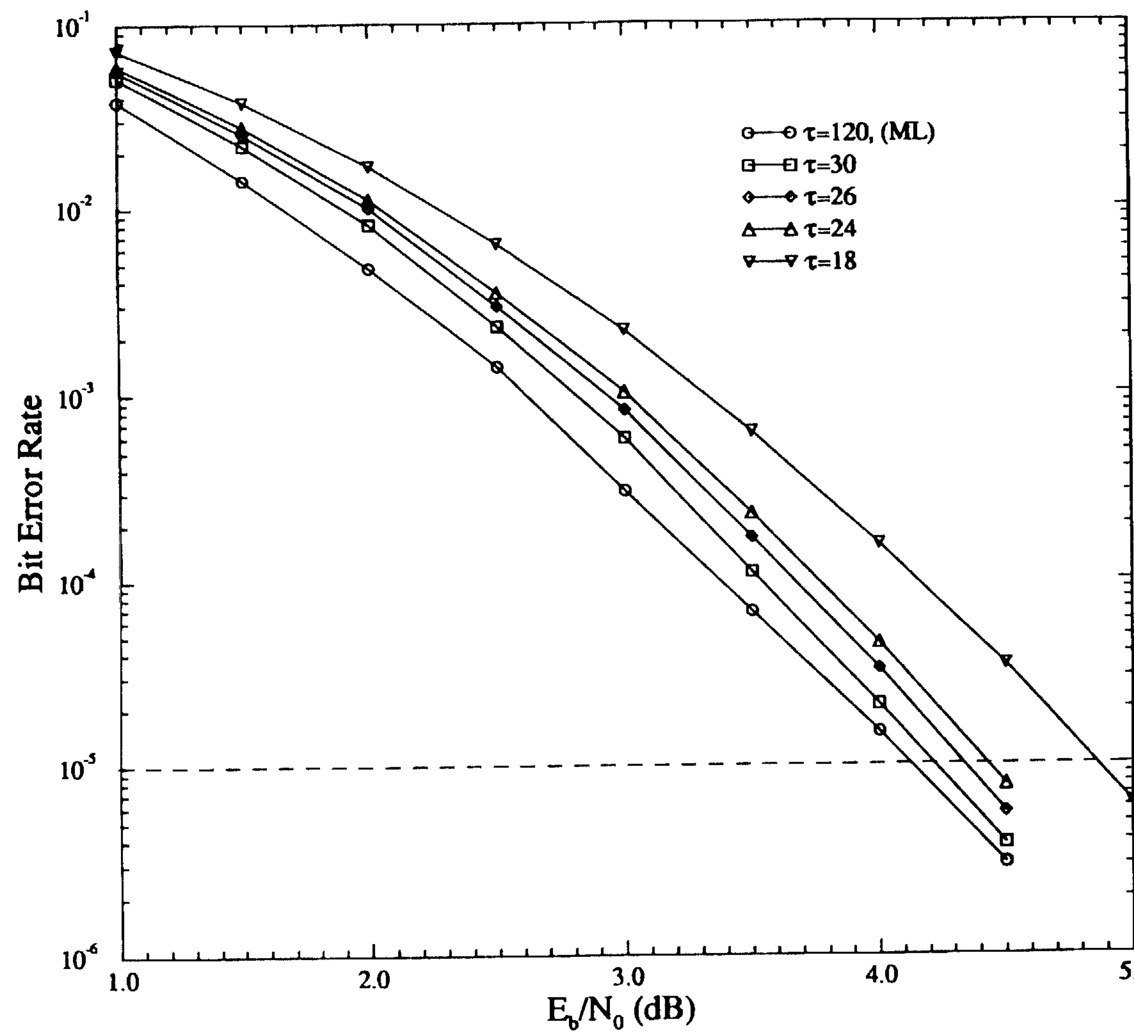



**Figure 8: BER Performance of the (2,1,6) Convolutional Code on the AWGN Channel with Varying Truncation Length**

N94-16509

# Search for Optimal Distance Spectrum Convolutional Codes *

Matthew C. Connor
Lance C. Perez
Daniel J. Costello, Jr.

Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Presented at the 4$^{th}$ Annual Argonne Symposium for Undergraduates in Science, Engineering, and Mathematics

Argonne National Laboratory

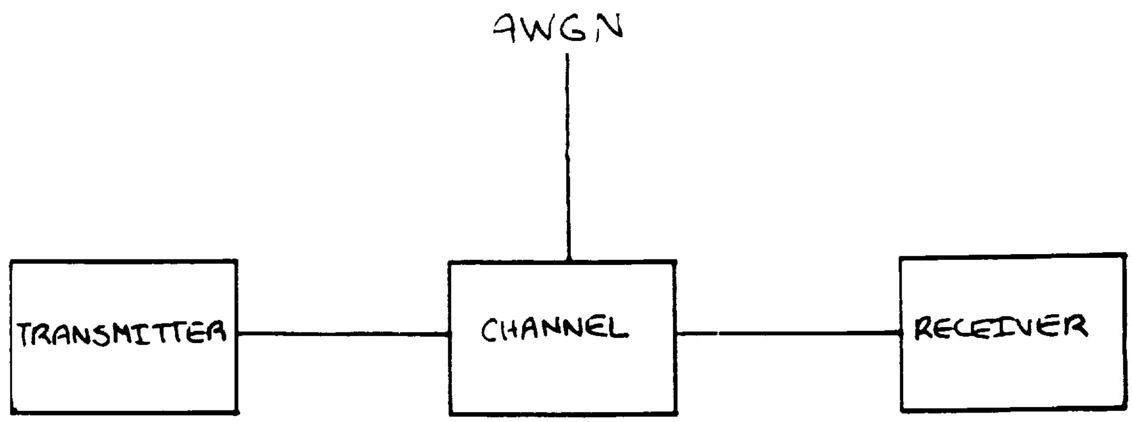
6 November 1993

*This work was supported in part by NASA Grant NAG5-557 and NSF Grant NCR89-03429

# Introduction

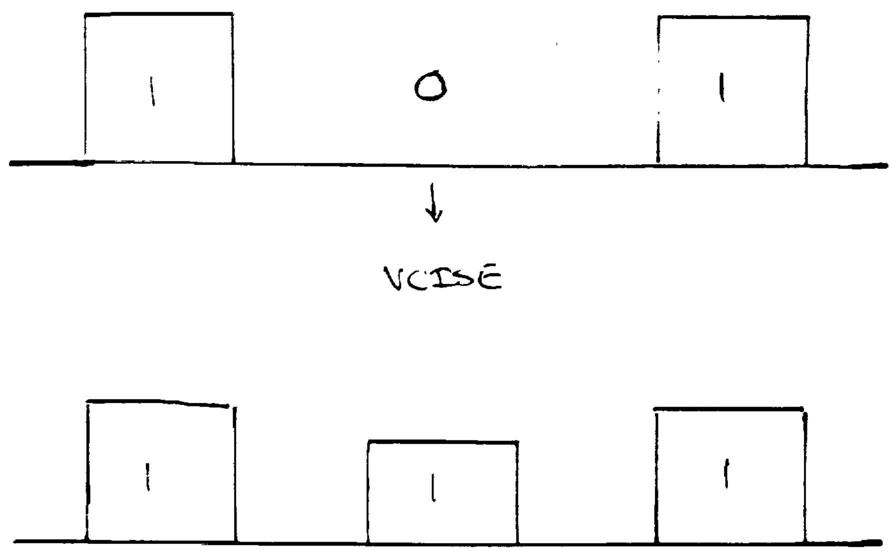
- In order to communicate reliably and to reduce the required transmitter power, NASA uses coded communication systems on most of their deep space satellites and probes (e.g. Pioneer, Voyager, Galileo, and the TDRSS network).
- These communication systems use binary convolutional codes. Better codes make the system more reliable and require less transmitter power.
- However, there are no good construction techniques for convolutional codes. Thus, to find good convolutional codes requires an exhaustive search over the ensemble of all possible codes.
- In this paper, an efficient convolutional code search algorithm was implemented on an IBM RS6000 Model 580. The combination of algorithm efficiency and computational power enabled us to find, for the first time, the optimal rate 1/2, memory 14, convolutional code.

# Digital Transmission Over a Noisy Channel




- When binary digital data is transmitted over a real channel, it is subject to noise (we will assume Additive White Gaussian Noise). The noise can cause errors to occur at the receiver.
- The acceptable bit-error-rate (BER) at the receiver depends on the type of data being transmitted. For example, video signals are more forgiving of errors than computer data.
- One of the goals of forward error correction (FEC) coding, is to allow the receiver to correct errors caused by the channel and thus to increase the reliability of the system and/or reduce the required signal energy.

# Example (Binary Numbers)




- To transmit a 5 in binary, the codeword 101 would be sent. This sequence of transmitted bits is then subject to channel noise.
- If the channel noise is large enough relative to the transmitted signal energy (per bit), the receiver may interpret a transmitted 1 as a 0, or vice-versa.
- For example, an optimum receiver would interpret the received signal shown above as 111 or 8. In this case, the receiver makes one error which in turn causes one codeword, 101, to be converted into another codeword, 111.
- The probability that a 1 is received as a 0 and vice versa is called the channel transition probability, $p$, and is a function of the signal-to-noise ratio (SNR),

$$SNR = \frac{E_S}{N_0}$$

where $E_S$ is the average transmitted signal energy per bit and $N_0$ is the one sided noise spectral density (a measure of the noise power).

# Example (cont.)

Given a channel transition probability of $p$, the probability that 101 is transmitted and 111 is received is given by

$$P_{101,111} = P_1 = (1-p)^2 p$$

This probability can be reduced by increasing the SNR which in turn causes a reduction in $p$.

In this example, one bit error causes one codeword to be converted into another codeword. We say that this code has *minimum free Hamming distance*, of $d_f = 1$.

| | |
|---|---|
| 000 | 100 |
| 001 | 101 |
| 010 | 110 |
| 011 | 111 |

In general, each codeword in this code may be converted into 3 different codewords by a single bit error with probability $P_1$, 3 different codewords by two bit errors with probability

$$P_2 = (1-p)p^2 < P_1$$

and one other codeword with three bit errors with probability

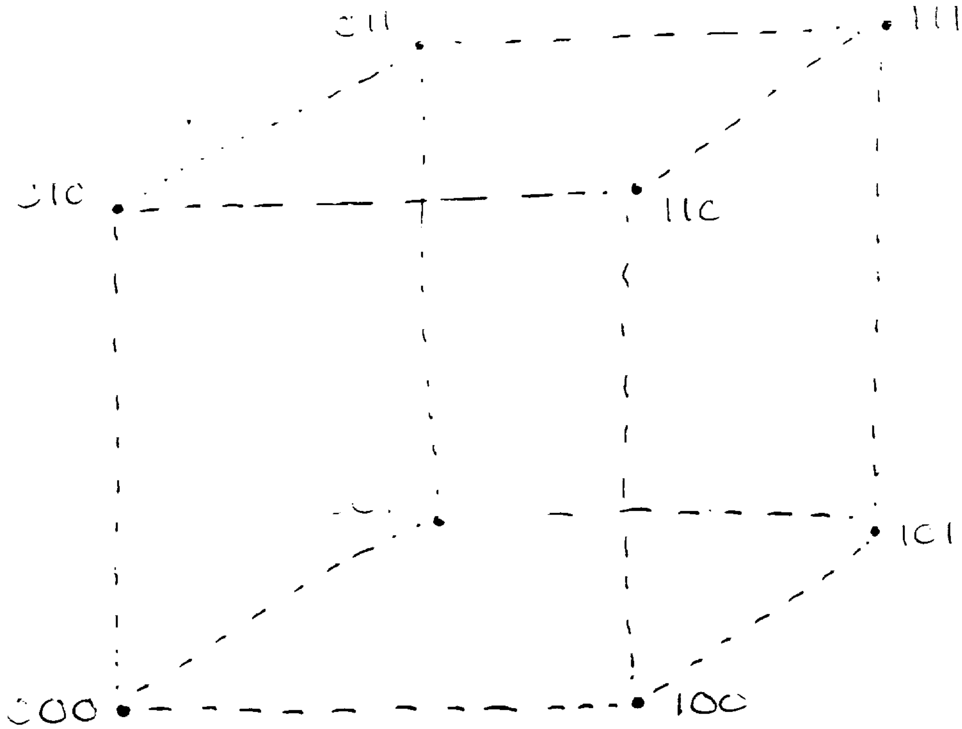
$$P_3 = (1-p)p^2 < P_2$$

The overall *probability of codeword error* is

$$P_C = 3P_1 + 3P_2 + 1P_3$$

which can be reduced by increasing the SNR.

# Geometric Interpretation and Hamming Distance




- Intuitive insight into the error mechanism can be obtained using a geometric perspective.
- From this point of view, each codeword in the previous example is considered a vector in a 3-dimensional vector space. The distance between two vectors is the *Hamming Distance*, $d_H$, which is just the number of positions in which two vectors differ.
- The probability that a codeword is converted into another codeword at a Hamming distance of $d$ is

$$P_d = (1-p)^{3-d}p^d$$

- Notice, that as the Hamming distance between two codewords increases $P_d$ decreases!
- For a fixed SNR and thus a fixed channel transition probability, $p$, the probability of a codeword error can be reduced by increasing the Hamming distance between all pairs of codewords.

# Example: Repetition Coding

- A simple coding technique is known as repetition coding. In this scheme each bit is simply transmitted twice in succession.
- Continuing the previous example, repetition coding leads to the following set of codewords.

| | |
|---|---|
| 000000 | 110000 |
| 000011 | 110011 |
| 001100 | 111100 |
| 001111 | 111111 |

- The mininum free Hamming distance is now $d_f = 2$ and the overall probability of codeword error is

$$P'_C = 3P_2 + 3P_4 + 1P_6 < P_C,$$

because each codeword has 3 codewords at distance 2, 3 codewords at distance 4, and 1 codeword at distance 6.

- The enumeration of the distances between one codeword and all other codewords in the code is called the code *distance spectrum* and is usually depicted in the following way

| $d$ | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| $N_d$ | 3 | 0 | 3 | 0 | 1 |

# Coding Performance Tradeoffs

- The probability of codeword error in digital communications systems on the AWGN channel is determined primarily by three factors:
  1. SNR,
  2. $d_f$, the code's minimum distance, and
  3. the code's distance spectrum.
- Historically, due to the expense of putting large power supplies in space and the relatively large amount of available bandwidth, NASA has chosen to improve system performance by using coding and expanding the required transmission bandwidth.
- With most of its satellites and deep space probes, NASA has chosen to use convolutional codes because of their superior performance characteristics in this application.

# Convolutional Codes

- A binary linear convolutional code with rate $k/n$ is a set of semi-infinite sequences generated by a finite state machine characterized by three parameters:

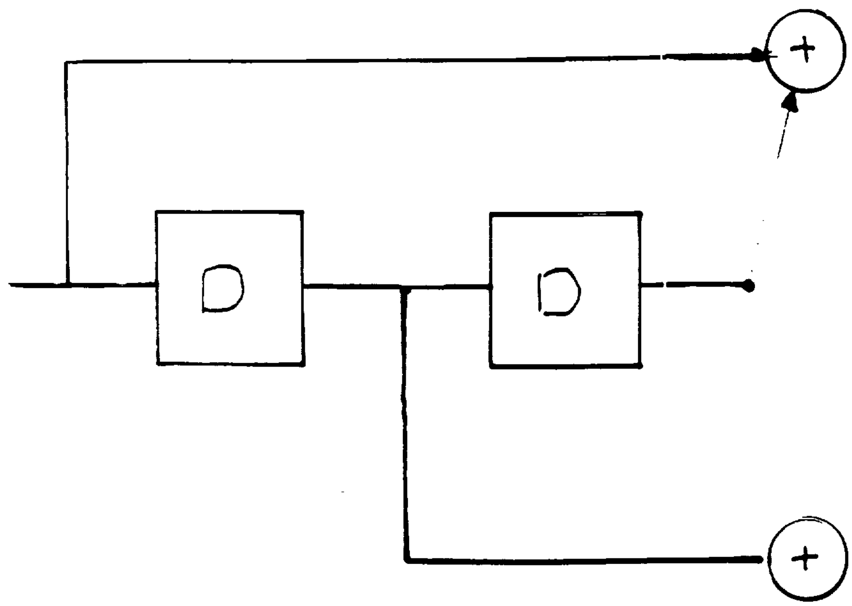
  1. $k$, the number of inputs bits per encoding interval,
  2. $n$, the number of output bits per encoding interval,
  3. $m$, the memory order of the finite state machine.

  The finite state machine has $2^m$ states.

- During each encoding interval, an $(n, k, m)$ convolutional code encodes $k$ information bits into $n$ bits based on the current block of $k$ bits and the past $m$ blocks of $k$ bits.

- The minimum distance between codewords and thus the performance of a convolutional code increases as the rate decreases and the memory increases.

# A (2,1,2) Convolutional Code




- A rate 1/2, convolutional code is specified by a pair of generators denoted by $(g1, g2)$ that describe the connections from the shift register to the output.
- The (2,1,2) code shown above has generators

$$\begin{aligned} g1 &= 101 = 5 \\ g2 &= 010 = 2 \end{aligned}$$

# Optimal Distance Spectrum Codes

- The maximum free distance of a (2,1,14) code is known to be 18. Many good codes have been found that have this free distance. The goal of this research was to find the "best" rate 1/2, memory 14 convolutional code with free distance 18.
- One way to do this is by finding the distance spectrum of every possible code.
- Those codes with fewer paths at a given distance have a lower probability of error, and thus are considered better. If the number of paths are recorded for each code having a minimum free distance of 18, the list could then be sorted and the best code found.
- For example, the maximal free distance (2,1,14) code with generators $(g1, g2)$ =(56721,61713) has 33 paths of weight 18. If another (2,1,14) code with fewer weight 18 paths could be found, this code would be a better code.

# The Problem with Finding Optimal Codes

- Finding the optimal code would be easy if the all of the codes' distance spectra could be evaluated and sorted in a reasonable amount of time. However, there are 1,073,741,824 possible codes of memory length 14.

- Finding a single (2,1,14) code's distance spectrum is a complicated process that takes approximately 30 CPU seconds on the IBM RS6000 Model 580. At this rate, a search of every code would take roughly one millenia, not including the sort routine to find the best code.

- Thus, to make any search feasible, it is necessary to first pare down the number of codes that must be tested by using other techniques for detecting inferior codes.

- In addition, all catastrophic codes must be eliminated before attempting to find their distance spectrum. Catastrophic codes are codes in which a finite weight information sequence can generate an infinite weight codeword.

- This characteristic causes an infinite loop in the distance spectrum algorithm; if not eliminated these codes would make the search impossible.

- Unfortunately, an algorithm to recognize catastrophic codes is very complicated and time consuming because it involves factoring.

# Methods of Reducing the Number of Codes

- The number of codes can be reduced by making certain restrictions regarding the structure of the codes. These restrictions are based on known properties of convolutional codes and do not affect the search results in any way.

- The two primary restrictions used were

  1. both generators must start with a 1, and
  2. one generator must end with a 1.

  These restrictions reduce the number of codes by a factor of 8.

- Second, an upper bound on the free distance can be utilized to eliminate codes that cannot achieve the known maximal free distance. This bound uses the row distance function, which is a decreasing function whose limit is the free distance.

- For most codes, the row distance function converges quickly and is a very effective way of reducing the number of codes.

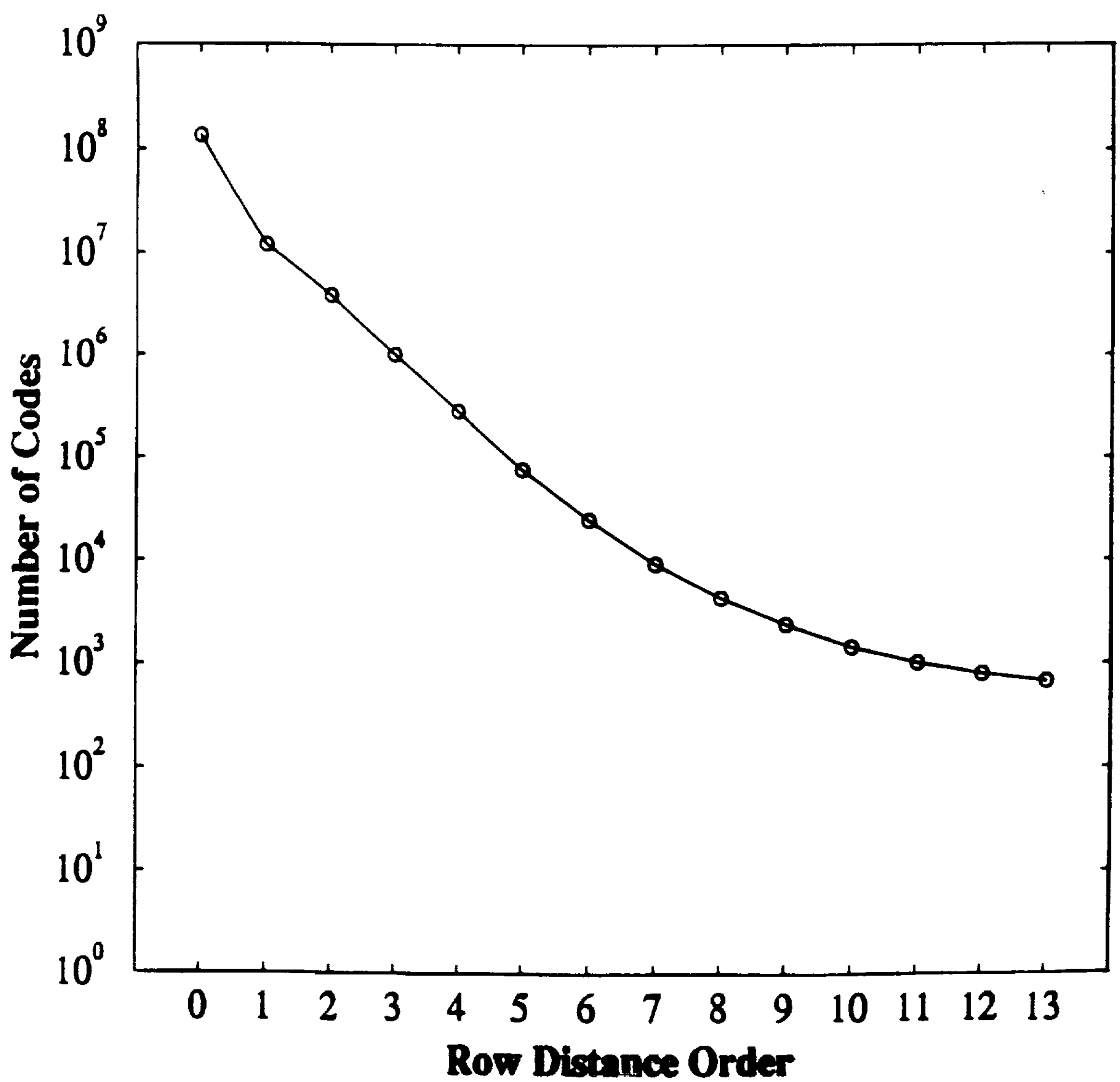
- Third, codes whose generators are mirror images of each other can be eliminated, because they generate identical sets of codewords and thus identical distance spectrums.

# Effectiveness of Schemes to Eliminate Codes

Initially, there are 1,073,741,824 possible codes and the search would have taken 1021 years.

After placing the two restrictions on the code generators, the number is reduced to 134,217,728. This search would have required 127 years.

The row distance evaluations, which require significant computational time, reduce the number of codes to a few hundred.



# The Optimal Distance Search : FAST

With a reduced list of generators, evaluating the distance spectrum becomes feasible. This was done by implementing a version of the FAST algorithm (A Fast Algorithm for Searching a Tree) published by Cedervall and Johannesson.

Given a set of generators, the FAST algorithm builds and searches the code tree to determine the weight of all relevant code sequences. Using column distance function bounds to limit and speed the search, it ultimately returns the number of paths for the ten lowest weights.

Efficient programming and compiler optimization resulted in a CPU time of 30 seconds for the distance spectrum evaluation of one (2,1,14) code.

After using FAST to evaluate the candidate codes, the distance spectrum results must be sorted.

# Search Results and Conclusions

- The (2,1,14) code with generators $(g1, g2)$ = (63057,44735) was found to be the optimal distance spectrum code.
- This code has only 26 weight 18 paths, as opposed to the previously known best (2,1,14) code which has 33 weight 18 paths. Thus, the new code is optimum for high signal-to-noise ratios.
- The new code is being simulated using computer models Notre Dame and a real decoder at the Jet Propulsion Laboratory in Pasadena, California, to determine if it is the best code for moderate SNR's.
- The techniques used in this code search are being refined and extended to find more complex codes for future NASA applications.

### Optimal Rate 1/2, $K = 15$ ($m = 14$), Convolutional Codes

The rate 1/2, $K = 15$ ($m$=14) convolutional code found by Cedervall and Johanneson [1] is the optimum distance spectrum (ODS) code. The generators for this code are

$$\begin{aligned} g^{(1)} &= 63057 = 1 + D + D^4 + D^5 + D^9 + D^{11} + D^{12} + D^{13} + D^{14} \\ g^{(2)} &= 44735 = 1 + D^3 + D^6 + D^7 + D^8 + D^{10} + D^{11} + D^{12} + D^{14} \end{aligned}$$

and its distance spectrum is

| $d$ | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|---|---|---|---|---|---|---|---|---|---|---|
| $N_d$ | 26 | 0 | 165 | 0 | 845 | 0 | 4844 | 0 | 28513 | 0 |

The generators for the code in Lin and Costello [2] are

$$\begin{aligned} g^{(1)} &= 56721 = 1 + D^2 + D^3 + D^4 + D^6 + D^7 + D^8 + D^{10} + D^{14} \\ g^{(2)} &= 61713 = 1 + D + D^5 + D^6 + D^7 + D^8 + D^{11} + D^{13} + D^{14} \end{aligned}$$

and its distance spectrum is

| $d$ | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|---|---|---|---|---|---|---|---|---|---|---|
| $N_d$ | 33 | 0 | 136 | 0 | 835 | 0 | 4787 | 0 | 27941 | 0 |

Both of these codes are invariant to $180^0$ rotations of the QPSK signal set.

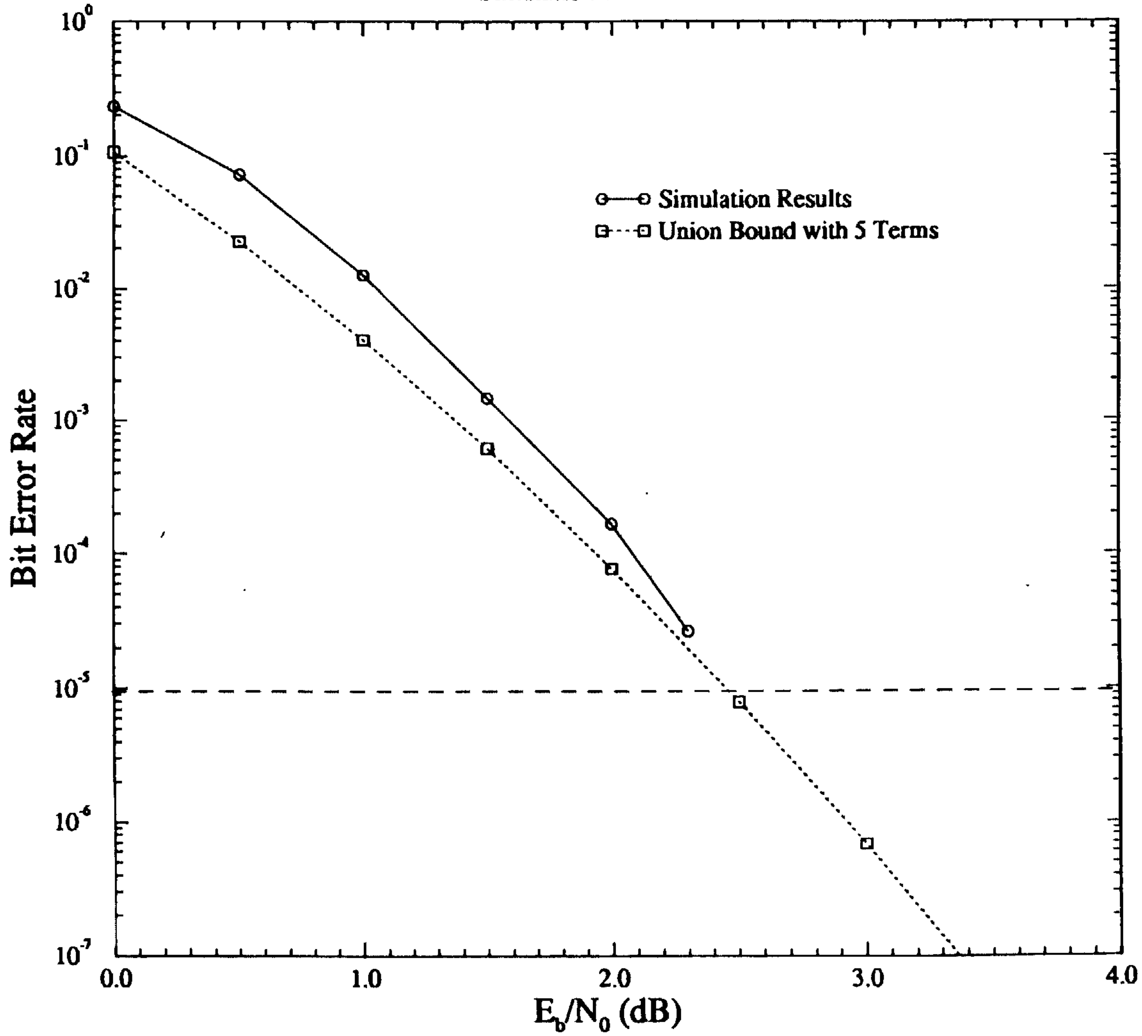
# References

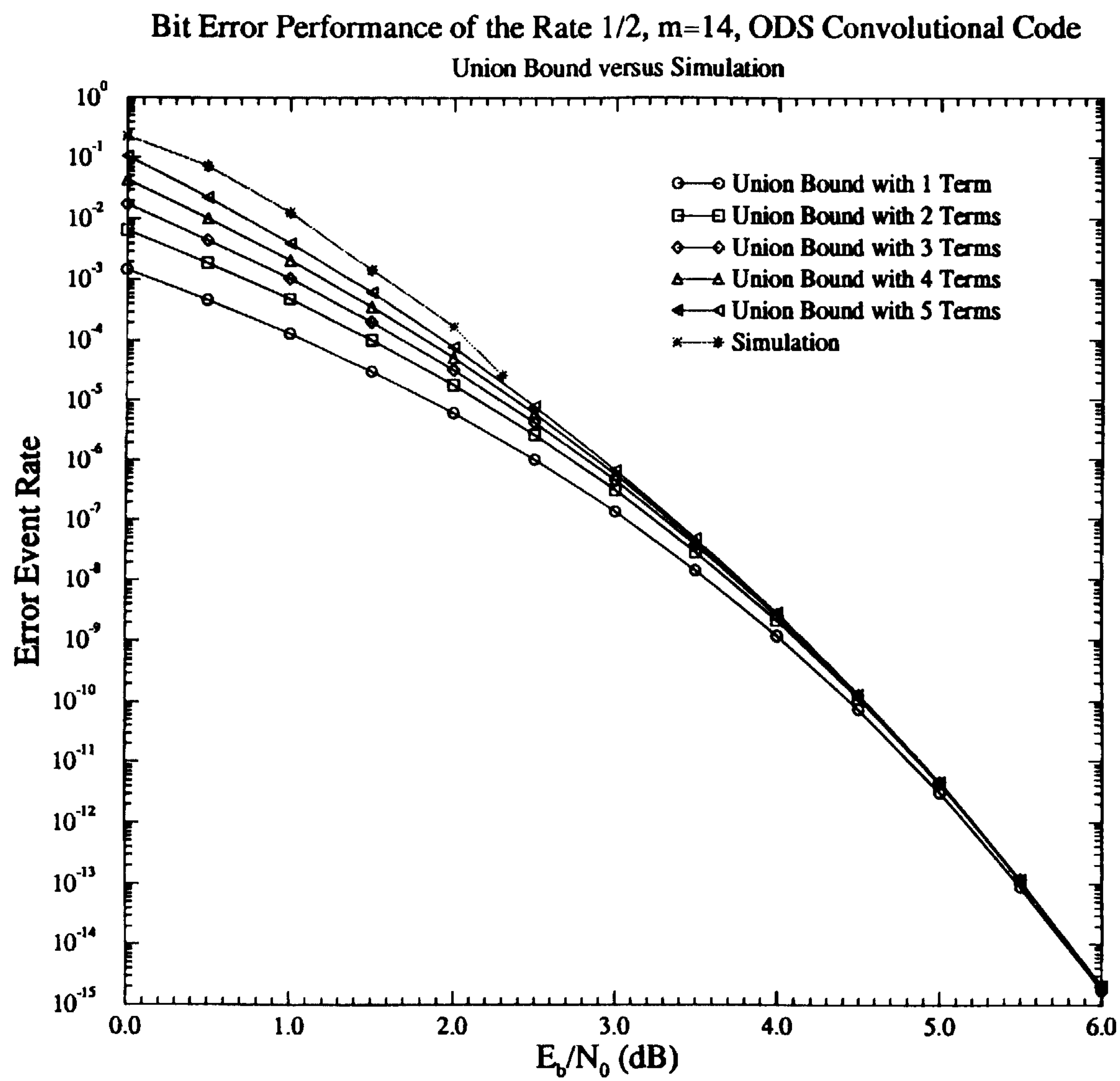

[1] M. Cedervall and R. Johanneson, "A Fast Algorithm for Computing Distance Spectrum of Convolutional Codes," *IEEE Trans. Inform. Theory*, **IT-35**, pp. 1146-1159, November 1989.

[2] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, New Jersey, 1983.

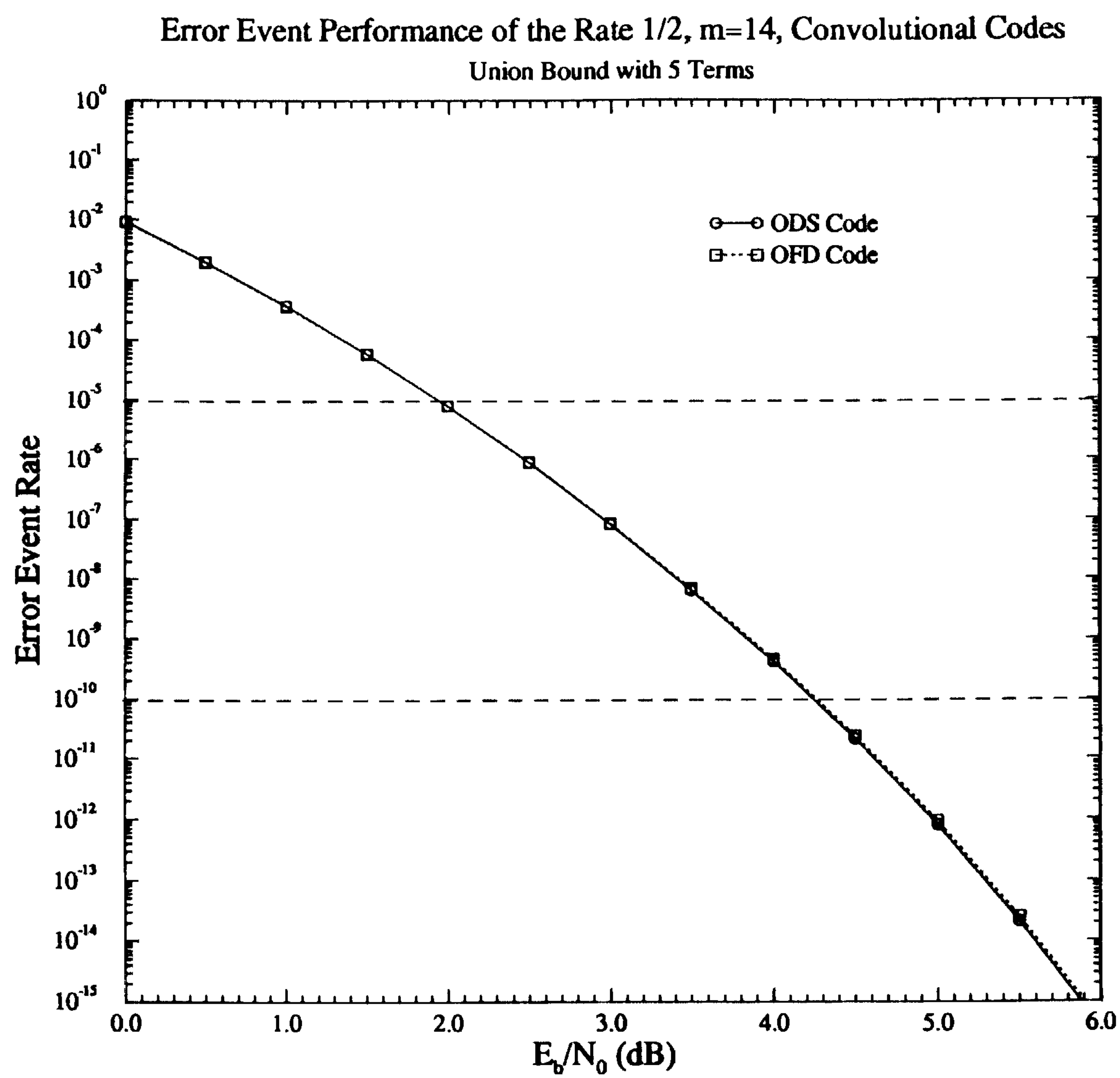
Bit Error Rate Performance of the ODS Rate 1/2, m=14, Convolutional Code

Simulation versus Bound





Bit Error Performance of the Rate 1/2, m=14, ODS Convolutional Code
Union Bound versus Simulation
Union Bound with 1 Term
Union Bound with 2 Terms
Union Bound with 3 Terms
Union Bound with 4 Terms
Union Bound with 5 Terms
Simulation
Error Event Rate
$10^{0}$
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
$10^{-6}$
$10^{-7}$
$10^{-8}$
$10^{-9}$
$10^{-10}$
$10^{-11}$
$10^{-12}$
$10^{-13}$
$10^{-14}$
$10^{-15}$
0.0
1.0
2.0
3.0
4.0
5.0
6.0
$E_b/N_0$ (dB)



Error Event Performance of the Rate 1/2, m=14, Convolutional Codes
Union Bound with 5 Terms
ODS Code
OFD Code
Error Event Rate
$10^{0}$
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
$10^{-6}$
$10^{-7}$
$10^{-8}$
$10^{-9}$
$10^{-10}$
$10^{-11}$
$10^{-12}$
$10^{-13}$
$10^{-14}$
$10^{-15}$
0.0
1.0
2.0
3.0
4.0
5.0
6.0
$E_b/N_0$ (dB)