COMBINED SOURCE–CHANNEL CODING USING MULTILEVEL/PHASE MODULATION

By ANDREW TODD WEITZNER

A thesis submitted to the

Graduate School – New Brunswick

Rutgers, The State University of New Jersey

in partial fullfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical Engineering

Written under the direction of

Professor D. G. Daut

and approved by

_____

_____

_____

New Brunswick, New Jersey

October, 1988

ABSTRACT OF THE THESIS

Combined Source–Channel Coding Using Multilevel/Phase Modulation

by Andrew T. Weitzner

Thesis Director: Professor David G. Daut

A combined source-channel coding and modulation approach is implemented for the encoding, transmission and remote reconstruction of a first-order one-dimensional Gauss-Markov source described by the correlation coefficient ρ. The source encoder employs differential pulse code modulation (DPCM). This is an efficient encoding scheme in the absence of channel errors. However, DPCM degrades rapidly in the presence of channel errors. Trellis-Coded Modulation is used to provide error protection without incurring a bandwidth expansion. This is accomplished by using m-bit DPCM with a rate $m/(m+1)$ trellis code and a 2^{m+1}-ary modulation technique such as multiple phase-shift keying (MPSK) or quadrature amplitude modulation (QAM). The trellis codes used are those developed by Ungerboeck. Performance of the overall system is evaluated on a mean square reconstructed error (MSRE) versus channel signal-to-noise ratio (SNR_{in}) criterion for an AWGN Channel. The system performance is increased by 3 to 7 dB as compared to an uncoded system and results are within 1.0 to 2.3 dB of the R_0 bound. The technique demonstrates an effective bandwidth

efficient means for transmitting source encoded data represented by more than one bit per source symbol which has not been previously demonstrated in the technical literature.

Acknowledgement

I would like to give my sincere thanks to Dr. David G. Daut for the guidance and motivation he has provided during this research effort and also during my graduate and undergraduate studies. I am indebted to Dr. Daut for my interest in communication systems. I am grateful to the members of the thesis committee, Dr. P. Sannuti and Dr. V. Lawrence for their consideration and helpful suggestions during the preparation of this thesis.

I am deeply indebted to Mr. Elias Eter for his contributions to this study. Our enlightning technical discussions and the continuous support he provided during this effort were well appreciated. I would also like to thank Mr. Mark Russo for his encouragement and extensive help in the preparation of this manuscript.

Sincere gratitude is extended to my family and friends for the moral support they provided which made this effort an enjoyable one.

List of Tables

1 Introduction

The use of combined source coding and channel coding techniques provide effective methods for maintaing the integrity of source encoded data when it is tranmitted over a noisy channel. Popular data compression techniques such as DPCM and the block Cosine transform require error protection in noisy channel environments. With the advent of Trellis-Coded Modulation (TCM) [14], system performance may be improved without the bandwidth expansion penalty that was once associated with the use of coding. It is therefore the goal of this thesis to study the proper matching of source coding, channel coding and modulation techniques to maximize system performance.

A combined source coding, channel coding and modulation approach is described for the efficient encoding, transmission and reconstruction of a highly-correlated sampled data source. The approach follows the combined source-channel coding work done by Modestino and Daut [1,2], which used Binary Phase Shift Keying (BPSK) modulation exclusively. At that time, a desire for good high-rate convolutional codes was expressed. Codes of rate $m/(m+1)$ are now available in the form of trellis codes. Trellis codes have also made use of multi-level and multiphase modulation schemes practical. This leads to the combined source coding, channel coding and modulation approach described here.

One may question whether it is necessary to "match" separately optimized source encoders channel encoders and modulators. Although a principle result

of Shannon's information theory is that the source and channel coding functions are fundamentally separable, it is shown clearly in [3] that this assumption is justifiable only in the limit of arbitrarily complex encoders and decoders. In practical situations there are limitations on complexity and a combined source coder, channel coder and modulator may be simpler to design and implement [4,6].

The method used here is that of attempting to acheive performance levels attaining rate-distortion bounds. A useful and accurate rate-distortion bound must first be developed. This requires accurate models to represent the information source and the channel charcteristics. Careful selection of a performance evaluation criterion is also needed. A strategy can then be devised to meet those bounds for various system parameters. In this thesis it is argued that a strategy in which all system components are designed in tandem, to a single system performance criterion, will provide superior performance over a system which combines separately optimized system components. Therefore the proper selection of source encoder, channel encoder and modulation parameters must include a knowledge of available (or necessary) channel characteristics and transmitter power in order to achieve the best possible system performance. The important distinction to be made is that the source encoder designer should understand the channel and specify how the data is to be transmitted.

In addition to the work of [1] and [2] the combined source-channel concept recent advances have been cited in the literature. Two studies have focused on the

use of trellis source codes [4,6] while another has studied quantizer optimization [5]. The source trellis codes in [4] and [6] are effective techniques for approaching the 1-bit per symbol rate-distortion bound in noisy channel environments. While in [1] and [2] greater than 1-bit per symbol was achieved at the expense of a bandwidth expansion This study is unique in that it the modulation technique is considered as a design variable in the overall system configuration. An emphasis in this study is to efficiently achieve performance above the 1-bit bound when sufficient signal power is available without introducing a bandwidth expansion. The approach is twofold; first, the system performance is bounded based on the source model, the available channel bandwidth and channel input signal-to-noise ratio. Second, a suitable source encoding method is selected and then channel coding and modulation techniques are chosen which optimize the transmission of the specific source data.

The system under consideration is shown in Figure 1. The approach here is to take a one-dimensional (1-D) Gauss-Markov source with correlation coefficient ρ and encode it using an m-bit differential pulse code modulation (DPCM) technique. One source symbol will then be transmitted for each channel symbol.[1] This will be accomplished by a system using a rate $r = m/(m+1)$ trellis code in combination with a 2^{m+1}-ary modulation technique such as PSK or QAM. Comparison will be made to a reference system using uncoded PSK with 2^m signal

[1] Although other combinations are possible, this demonstrates a source encoder to channel encoder relationship of interest.

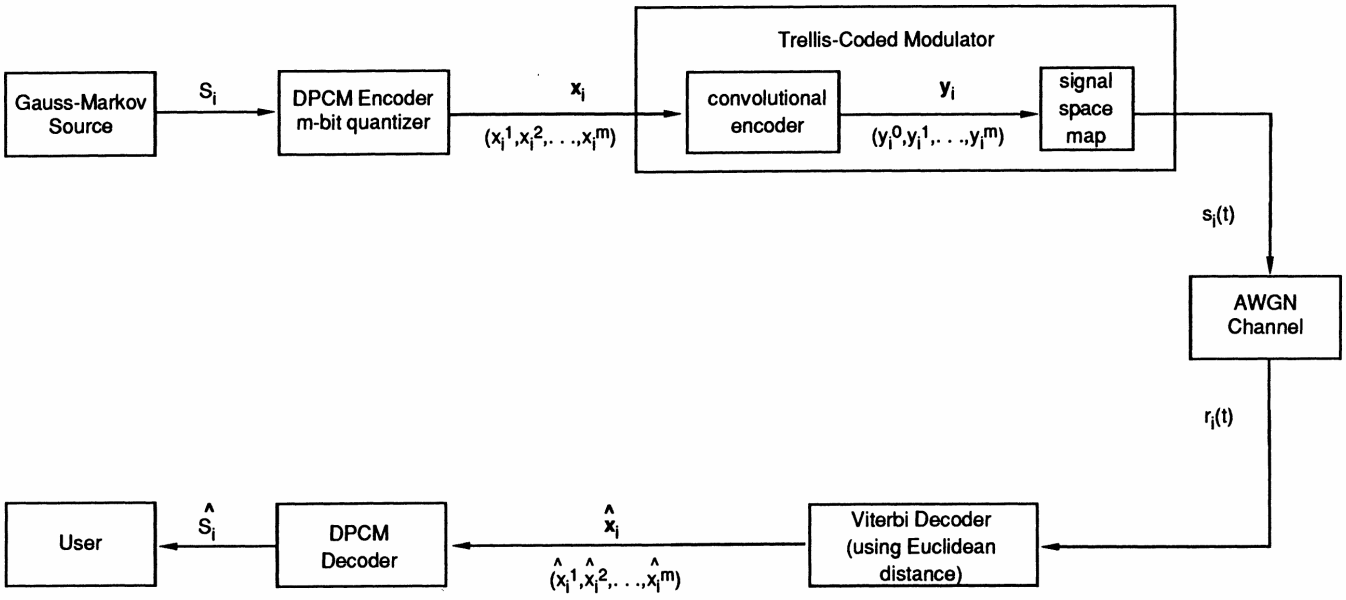

Figure 1: Overall system block diagram

research are presented. In addition, Appendix A details the implementation and justification for the design of the communication system software analysis package. Appendix B contains the derivation of the R_0 results and Appendix C documents the software used in the study.

where N is the number of times that the channel is used to transmit a codeword and R is the information rate in bits per second. $E(\tilde{R})$ and $E(R)$ are the respective upper and lower bounds on the channel "reliability function." The coefficient A is a function weakly dependent on N. For any given modulation scheme and channel, knowledge of these parameters allows one to make $\Pr(\varepsilon)$ as small as desired by choosing N sufficiently large.

For the discrete memoryless channel a typical form of the reliability function is shown in Figure 2. In particular, we see that $E(R)$ and $E(\tilde{R})$ coincide for $R_c \leq R \leq C$, where C is the channel capacity. Since the reliability function is convex cap with slope $= -1$ at $R = R_c$, we notice that

$$E(R) \geq R_0 - R \tag{2}$$

Hence R_0 is the first-order lower bound on $E(R)$ and

$$Pr(\varepsilon) \leq 2^{-NE(R)} \leq 2^{-N[R_0 - R]} \tag{3}$$

While this bound is weaker than Equation 1 for some values of R, it is exponentially optimum for rate $R = R_c$. The importance is that a relatively accurate bound exists which is described by a single parameter, namely R_0.

While the previous results represent the performance of a system using block codes, similar results have been made for convolutional codes. Viterbi [22] has shown that the R_0 parameter also correctly describes the performance of convolutional codes for certain rates as

$$\Pr(\varepsilon) \leq c2^{-\nu(R_0/R)}, \text{ for } R < R_0 \tag{4}$$

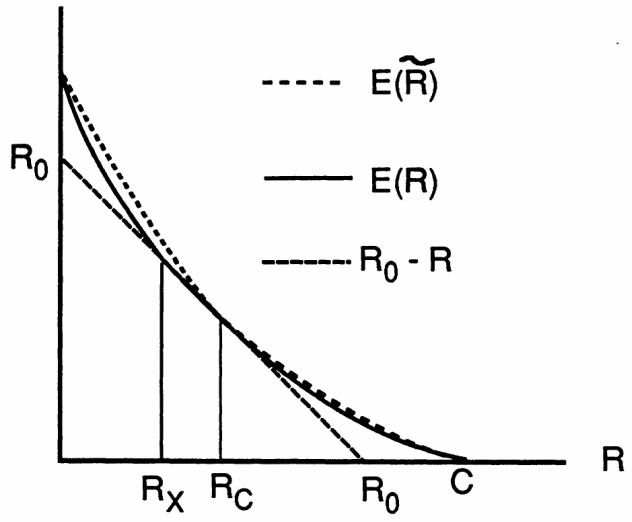

Figure 2: The Reliability function

where ν is the constraint length and is roughly equivalent to the block length N. The parameter c is a function weakly dependent on R. An important aspect of convolutional codes is that their performance is generally superior to block codes having similar complexity. Also, they may be simpler to encode and the decoding algorithms available are more amenable to soft-decision methods.

Massey argues that it is more constructive to design a communications system to an R_0 criterion rather than a raw probability of error, $\Pr(\varepsilon)$, and capacity, C. Whereas, capacity defines the limit of achievable rates of reliable communications over a channel, there is no way to show how to achieve those rates with finite system complexity.

On the other hand, a system designed to a error probability criterion is sen-

sible only if coding is not used. The problem here, Massey contends, is that the communications engineer may design the system to an error probability criterion and then attempt to use coding strictly as an error-correction procedure. This separation of coding and modulation may lead to inefficient designs which require excessive redundancy.

The inefficiency is due to differences between the Hamming distance used to design convolutional codes and the Euclidean space in which modulation schemes exist. Although separate coding and modulation have been successful for systems using binary phase-shift keying (BPSK) and quadrature phase-shift keying (QPSK or 4-PSK), these schemes transmit 1 bit per orthogonal dimension for which increasing Hamming distance can be mapped into increasing Euclidean distance. When the same techniques are applied to general M-level modulation systems, we are lead to believe that so-called "high-level" modulation schemes which use greater than 1 bit per dimension are inefficient, requiring very high signal-to-noise ratios and extensive coding to produce satisfactory results. This is because a relationship between Hamming distance and a modulation schemes Euclidean distance may not exist.

In contrast, for modulation over the AWGN channel, R_0 is determined directly from the Euclidean distance relationship between signal points in the modulation scheme. R_0 correctly describes the performance of high-level modulation schemes when coding is used. It is then of interest to derive and analyze the R_0 function for specific modulation schemes. From these results a theoretical reference is

available which can be compared to the system simulation results of Section 6. The trellis codes used in the simulation are discussed in Section 3 and are designed to maximize Euclidean distance so that high-level modulation improvements are possible. The improvements are a direct result of exploitation of the properties of the R_0 function.

Before determining the specific R_0 results it is meaningful to emphasis some of the advantages of using the R_0 function in communication system design. One of the most attractive properties is its simplicity. In Appendix B it is shown that R_0 can be arrived at by a Union bound approximation of the forward proof of the coding theorem. For an AWGN channel with equally likely symbols this leads to a result of the form

$$R_0 = -\log_2 \frac{1}{M^2} \sum_{l=1}^{M} \sum_{h=1}^{M} e^{-d_{lh}^2/4N_0} \tag{5}$$

Where the only unknown is d_{lh}^2, the square Euclidean distance between signal points in a modulation scheme. This result implies that it is simply to determine R_0 for a given signal constellation. Therefore, we can use R_0 to determine the bound on the coded performance of various modulation schemes and make relative comparisons between them, in terms of required coding complexity. In turn R_0 can be used to design (seek) modulation schemes which are optimal when using coding. This result may very well differ from constellations which are designed without coding in mind.

The R_0 function can also be used to determine the effect of receiver quantiza-

tion. The result is represented as R_0' and is calculated by modeling the receiver front end as an additional discrete channel cascaded onto the AWGN channel. The quantizer can be modeled as a continuous signal input to discrete n-level output channel with transition probabilities assigned due to the quantization levels. In the case of $n = 2$, the hard-decision versus soft-decision tradeoff can be assessed. This results in a 2 dB loss between the unquantized and hard-decision systems. Also, when soft-decision decoding is used the concept that coding is an error correction procedure is lost because the demodulator does not make hard errors which require correction by a subsequent decoder.

Another important concept involving R_0 was developed by Savage [30] and recently amended by Arikan [31]. It is shown that R_0 is an upper bound to R_{comp}, the computational cutoff rate for sequential decoders. This is the rate at which the memory requirements for sequential decoders becomes unbounded. This alone has been cited as a strong argument for R_0 as the practical bound on communication system performance [12].

Finally, we see that R_0 can be viewed as a realistic bound on achievable signalling rates. Whereas capacity may only be guaranteed with infinite length codewords, R_0 promises specific probability of error bounds for a given codelength used with a modulation scheme of interest.

channel. This allows us to interpret the abstract quantity of dimensions in terms of the physical characteristic of bandwidth. For MPSK and QAM the specific bandwidth is due to the shape of the low-pass signal used to represent the data and modulate the carrier. Using a rectangular pulse for the symbol duration T seconds, the bandwidth is approximately $1/T$ seconds^{-1}.

The bandwidth efficiency of a digital modulation technique is the ratio R/W, where R is the information rate in bits per second and W is the required bandwidth in hertz. Bandwidth efficiency can be measured in units bits/sec/Hz. MPSK and QAM both have efficiency approximated by $\log_2 M$ bits/sec/Hz.

The R_0 development is based on a system in which we have a source that produces one of L messages and a channel which has an alphabet of size M. We would like to transmit a message by using the channel N times. The modulator transmits a character in its alphabet by sending a waveform $s_i(t)$ onto the channel, where $i = 1, \ldots, M-1$ to represent M unique waveforms. By using the channel N times, a vector representing message l is formed $\mathbf{s_l} = (s_{l1}, s_{l2}, ..., s_{lN})$.

Gallager [9] has shown for such a system that the R_0 function is given by

$$R_0 = \max_{q_i} -\log \sum_j \left[\sum_i q_i \sqrt{\Pr_{ij}} \right]^2 \tag{6}$$

for which q_i is the probability of transmitting letter i and $\Pr_{ij}$ is the channel transition probability representing the *a prior* probability of the receiver deciding in favor of letter j given that letter i was transmitted.

For channel transition probabilities determined by the additive white Gaus-

sian noise (AWGN) channel with zero-mean and variance $N_0/2$ it has been shown in [11] that R_0 becomes

$$R_0 = \max_{q_i} -\log \sum_i \sum_j e^{-d_{ij}^2/4N_0} q_i q_j \tag{7}$$

The expression d_{ij}^2 is equivalent to $||s_i - s_j||^2$, the square Euclidean distance between signal points s_i and s_j in a given constellation. When all symbols are equally likely in an M-point constellation the result simplifies to

$$R_0 = -\log_2 \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} e^{-d_{ij}^2/4N_0} \tag{8}$$

From the above equation, it is seen that R_0 is easy to determine for a modulation technique from the coordinates of the signal points in the modulation scheme. The R_0 function can be used to determine the relative performance of various modulation techniques. In developing and rating modulation schemes it would also be of useful to appreciate the bounds on the R_0 function itself.

Shannon had shown that for the AWGN channel, there exists an R_0 upper bound for any modulator-demodulator pair given by

$$\begin{aligned} R_0^* &= \frac{\log_2 e}{2} \left[1 + E_N/N_0 - \sqrt{1 + (E_N/N_0)^2}\right] \\ &\quad + \frac{1}{2} \log_2 \left[\frac{1}{2}(1 + \sqrt{1 + (E_N/N_0)^2})\right] \end{aligned} \tag{9}$$

constrained only in energy

$$\{|\mathbf{s}_i|^2\} \leq N E_N;\ i = 0, 1, \cdots, M-1. \tag{10}$$

E_N represents the energy per dimension. Equation 10 implies that the signal constellation energy must average $\sqrt{E_N}$ per dimension. For 2-dimensional constellations the average energy per symbol, $E_s = 2E_N$, should lie within a circular boundary $\mathbf{E}\{|s_i|^2\} \leq 2E_N$ per symbol.

The average energy constraint is a consequence of the bounding technique used. In practical situation, some have argued that a peak power constraint is more realistic [12,28]. In this case, the 2-dimensional signal sets abide to a power limitation defined by

$$|s_i| \leq \sqrt{2E_N} \text{ per symbol}^2 \tag{11}$$

A peak constraint is typical of many physical channels as such travelling wave tube amplifiers (TWTA) used in satellite communications.

We have seen that reliable communication is possible for rates less than R_0^*. While this may be a practical bound, it is not the ultimate bound and is a consequence of a union bound technique. The capacity, C, represents the upper bound on communication. The equation

$$C = \frac{1}{2}\log\left(1 + \frac{2E_N}{N_0}\right) \tag{12}$$

is 2 to 3 dB superior to R_0^* but may require infinite coding complexity to achieve.

R_0 is now used to determine the relative performance of the modulation

[2] The definition for the peak energy value is correct for the 2-dimensional bound, $2R_0^*$. This makes the MPSK average and peak constraint signal sets identical

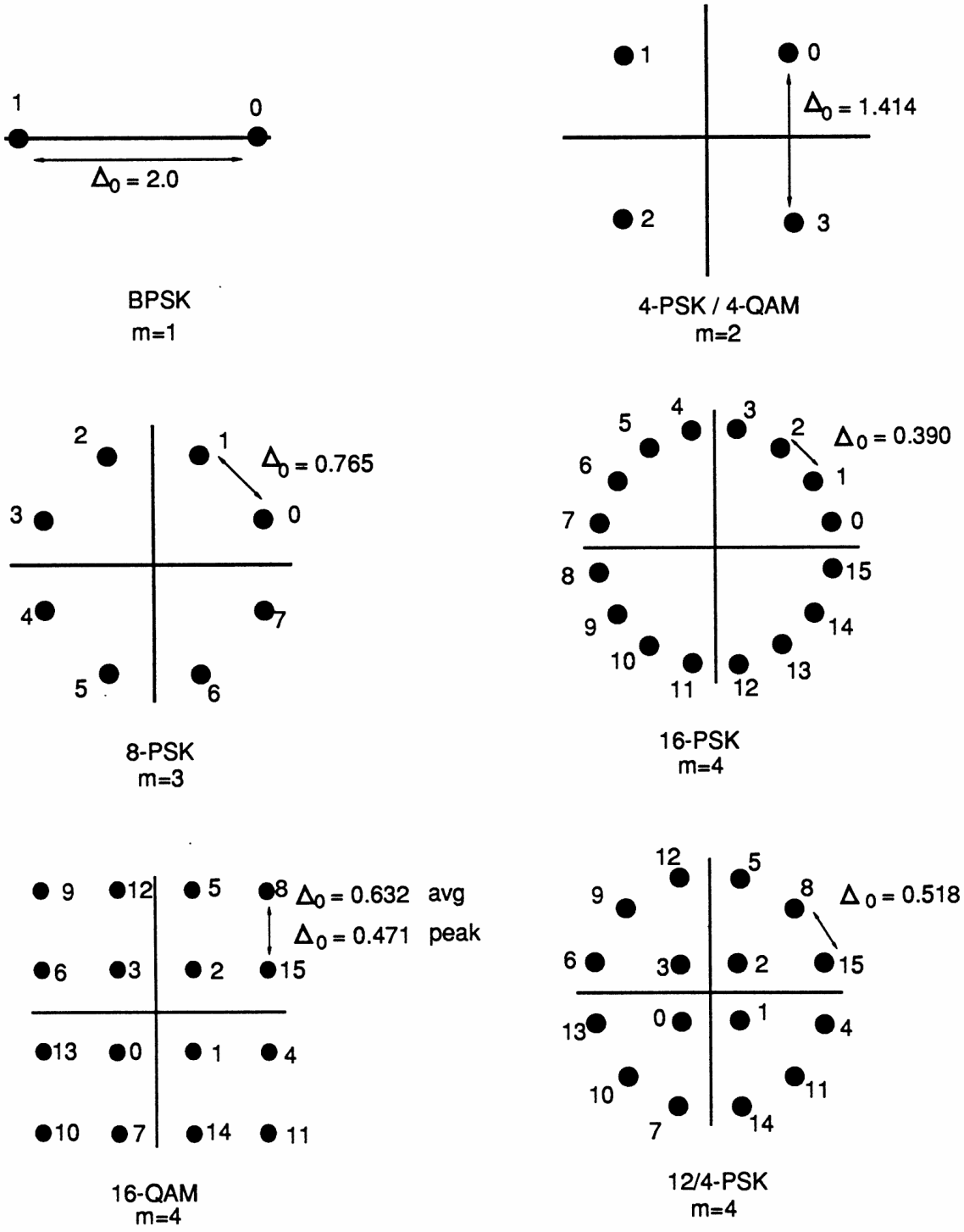

Figure 3: Channel signalling sets

schemes used in the simulation. The constellations are shown in Figure 3. Based on this signal space representation the R_0 function been evaluated by the computer program in Appendix C. The algorithm which were used in the software are based on the expressions derived in Appendix B for M-ary PSK and M-ary QAM.

For M-ary PSK the following expression was derived:

$$R_0 = -\log_2 \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \exp\left\{ \frac{N_s E_N}{N_0} \left[\sin \frac{\pi}{M}(i-j) \right]^2 \right\} \text{ bits/symbol} \qquad (13)$$

where N_s is the number of dimensions per symbol and is 1 for $M = 2$ and 2 for all integer values of M greater than 2.

For M-ary QAM a closed form expression can also be used:

$$R_0 = -\log_2 \frac{1}{M^2} \sum_{i_1=1}^{\sqrt{M}} \sum_{j_1=1}^{\sqrt{M}} \sum_{i_2=1}^{\sqrt{M}} \sum_{j_2=1}^{\sqrt{M}} \exp\left\{ 2a^2 \frac{E_N}{N_0} \left[(i_1 - j_1)^2 + (i_2 - j_2)^2 \right] \right\} \qquad (14)$$

here $a = 1/\sqrt{10}$ for the average power constraint and $a = 1/\sqrt{18}$ for the peak power constraint in figure 3.

In addition, the R_0 function was determined for a QAM-PSK hybrid constellation called 12/4-PSK. This constellation is shown in Figure 3 and will be discussed further. Due to its unusual construction a closed form expression for R_0 was not feasible and calculation is best performed using Equation (7).

The R_0 curves are shown in Figures 4,5 and 6. Each graph displays a family of curves for a modulation type. The R_0 functions for M-ary PSK are in Figure 4. Curves are plotted for BPSK, 4-PSK, 8-PSK and 16-PSK. The curve

represents a practical bound on the amount of information which can be which can transmitted by the modulation technique at a given channel signal-to-noise ratio with arbitrarily small probability of error. Using Equations (3) or (4), as long as $R \leq R_0$ we are guaranteed that a code exists for which the probability of error can be made small by choosing N sufficiently large. Notice that as $R_0 - R$ approaches zero, N must approach ∞ to maintain a small value of $\Pr(\varepsilon)$.

The following can be observed in Figure 4: Each curve has a threshold at which the amount of information saturates. For an M-ary constellation this occurs at $\log_2 M$ bits/symbol. Above SNR_i of 20 dB channel errors are rare and the information rate is limited by the constellation size. Notice that the family of PSK curves overlap at low SNR_i until the saturation level of each constellation is reached. This implies that increased quantization by the transmitter does not necessarily degrade information transfer.

For BPSK saturation at 1 bit/symbol occurs near 6 dB. Notice that 4-PSK saturates at 2 bits/symbol for the same channel SNR. This is due to the increase from 1 to 2 dimensions and therefore a doubling in channel capacity between BPSK and 4-PSK. In practical systems that extra dimension may not be available since it is common to transmit two independent data sources on the inphase and quadrature components of a QPSK system. In this study the focus is on two available dimensions.

As channel input signal-to-noise ratio increases, so does the channel capacity. For example, when SNR_i surpasses 12 dB the use of 4-PSK is wasteful of channel

capacity since 3 bits/symbol can be supported by the use of 8-PSK. Similarly, 16-PSK becomes efficient at $SNR_i > 18$ dB. The ultimate performance is specified by capacity, C, and the more practical bound R_0^* which is identical to the R_0 functions for small SNR_i. As shown in Figures 4 through 6 the proper bounds for two-dimensional modulation are twice the values of Equations 9 and 12 when the results are presented in bits per symbol.

The R_0 curves for QAM are shown in Figure 5. The QAM curves are more efficient than PSK for M greater than 2 bits/symbol as compared with R_0^*. Specifically at $M = 16$, QAM achieves saturation at 4 dB less than PSK. Notice in the QAM functions the R_0 values for family of curves are not aligned at low SNR_i as does PSK. This can be remedied by optimizing the probabilities of each channel letter as shown in [28]. Although in practice, the loss is small enough that the extra complexity is considered not to be necessary.

A direct comparison of PSK and QAM signal constellations is shown in Figure 6. For M=4 the sets are identical. At M=16 the average power constrained QAM is superior to 16-PSK as was stated. Also included in Figure 6 is the performance of a QAM system with the peak-power constraint applied. The peak power constrained 16-QAM signals must lie on or within the circle which the 16-PSK signals form. This causes the peak version of 16-QAM to be 2.5 dB inferior to the average constraint. While the peak power constrained 16-QAM is only slightly superiority to 16-PSK, the PSK modulation is preferred in this circum-

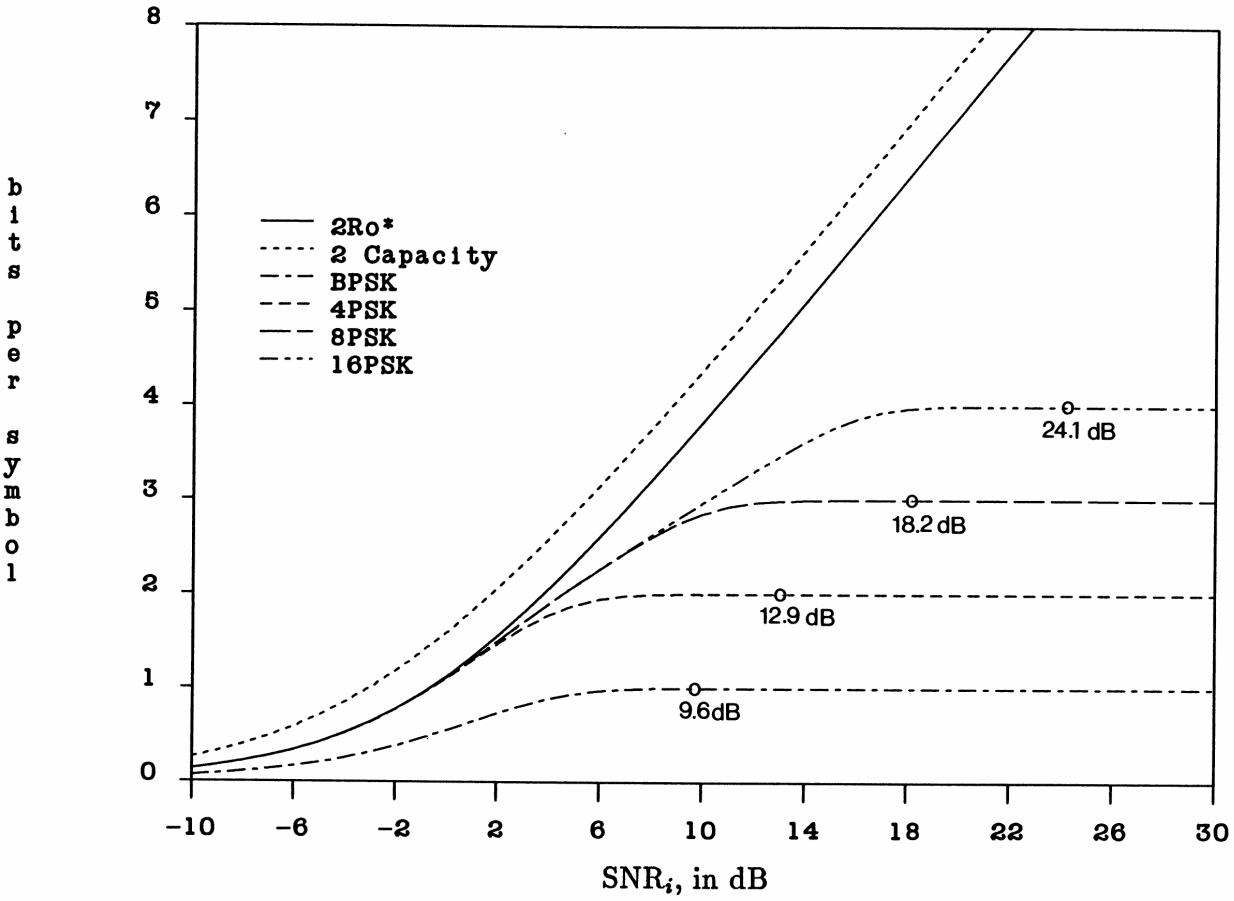

Figure 4: R_0 for PSK modulation

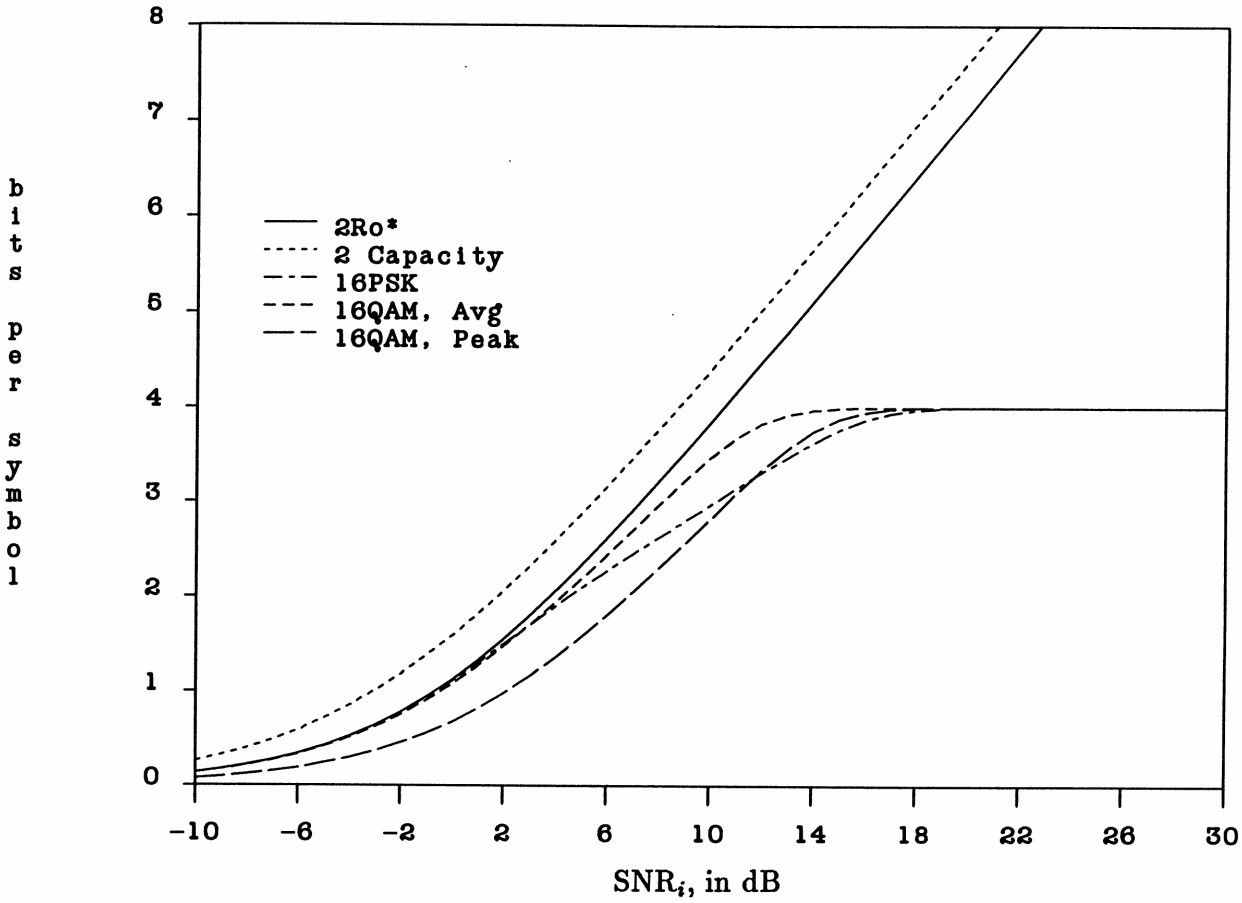

Figure 5: R_0 for QAM modulation

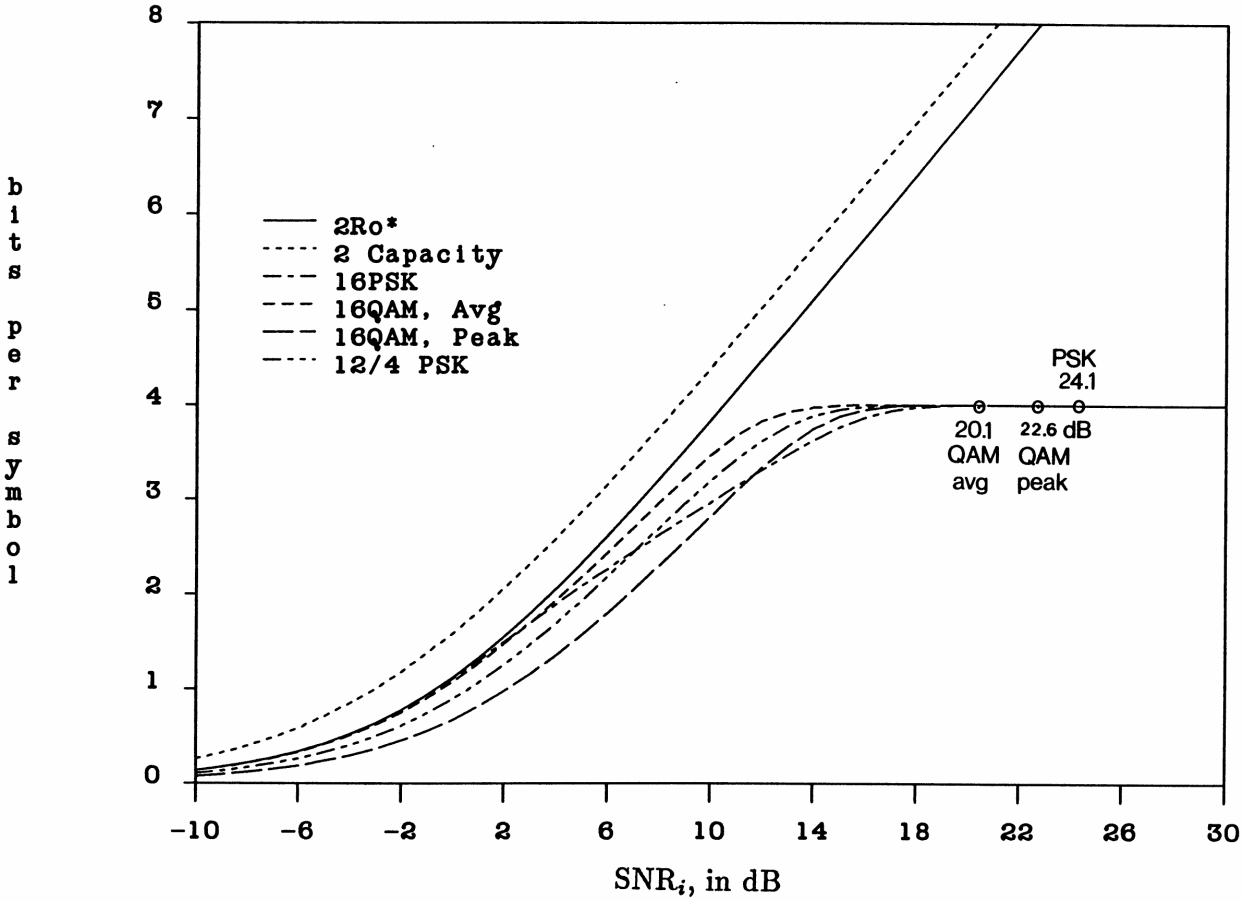

Figure 6: R_0 for PSK versus QAM modulation

3 Trellis Coded Modulation

We now discuss the coding which will be used to achieve the bounds of the previous section. Trellis Coded Modulation (TCM) is a bandwidth efficient modulation technique. It does so by regarding the channel coder and the modulator as single entity. The technique adds redundancy into the signal via coding and also expands the channel signalling set to "make room" for coding without a bandwidth expansion of the resultant signal.

Trellis codes may thought of as a generalized form of convolutional codes. Convolutional codes are designed to maximize Hamming distance and generally they can only perform efficiently for modulation techniques in which maximization of the Hamming distance corresponds to maximizing Euclidean distance. This is typical of the orthogonal and bi-orthogonal signal sets which operate at bandwidth efficiencies of $R/D \leq 1$ bit/dimension. The most common examples are BPSK and 4-PSK. For bandwidth efficient modulation (i.e., bandwidth efficiency of greater than 1 bit/dimension) a relationship does not exist between increasing Hamming distance corresponding to increasing Euclidean distance.

Alternatively, trellis codes are constructed by maximizing the Euclidean distance between messages as they appear on the channel for a given signal constellation. This is the sense in which TCM regards the coder and the modulator as an entity. Ungerboeck developed codes which maximize Euclidean distance by a method called "mapping by set partitioning". The technique partitions the

signal set successively into subsets with increasing minimum distance between nearest neighbors in the new subset of the signal constellation.

The existence of trellis codes can be seen from the R_0 function for multi-level modulation. Trellis codes take advantage of the larger value of R_0 that occurs when the number of letters in a signal constellation are expanded. By keeping the information rate constant, the $R_0 - R$ term becomes larger in Equation (3) and allows the same performance with a reduced codelength. Examining Figure 4 we see that if it is desired to transmit 2 bits/symbol at $\Pr(\varepsilon) = 10^{-5}$, this can be done by using 4PSK at 12.9 dB. Alternately R_0^* predicts that 2 bits/symbol can be achieved at 3.8 dB and R_0 for 8-PSK predicts 2 bits/symbol at 4.8 dB. Therefore by using a rate 2/3 code it should be possible to transmit 2 bits/symbol by using 8-PSK with a savings of 8 dB. For this case, $R_0 - R$ is small and Equation (3) would imply the need for a very large code length. However at $\mathrm{SNR}_i = 7$ dB, $R_0 - R$ becomes 0.5 and Equation (3) guarantees that performance can be described as

$$\Pr(\varepsilon) \leq 2^{N/2} \tag{15}$$

Therefore a code must exist such that $\Pr(\varepsilon) \leq 10^{-5}$ for $N \approx 33$. In practice codes with good performance of much smaller length exist. The conservative prediction of R_0 is due to method of bounding used. This is twofold: 1) R_0 is not an ultimate upper bound on coding performance; 2) R_0 is determined by averaging the set of all possible codes. This includes mostly poor codes which

stem from the elegant random coding argument used in the forward proof of the coding theorem. The removal of some of these codes may lead to a tighter bound but the simplistic R_0 bound would be lost. Hence, the R_0 function should be used to determine where good short constraint length codes may exist when $R \leq R_0$.

The trellis codes developed by Ungerboeck have the same form as the previous example. To transmit m bits of information per symbol a rate $m/(m+1)$ code is used. The $m+1$ encoded bits are transmitted using a channel signalling set from the previous section with an alphabet of size 2^{m+1}. Performance gain is measured against a reference system of m bits per symbol using an uncoded modulation with 2^m letters in the channel signalling alphabet.

The trellis codes are constructed by the mapping by set partioning technique aimed at maximize free Euclidean distance (ED). The mapping takes a channel signalling set and successively partions it into subset with increasing minimum distances $\Delta_0 < \Delta_1 < \Delta_2 \cdots$ between the signals of the subsets. The technique is illustrated in Figure 7 for 8-PSK. A result of the mapping is a numbering scheme assigned to each signal point. The numbering has been included for each modulation scheme in Figure 3. Notice that for MPSK the partion results in the natural binary mapping of the channel signals. [3]

The following rules are used by Ungerboeck [14] in developing the 8-PSK

[3] Although other mappings are possible by permuting subsets.

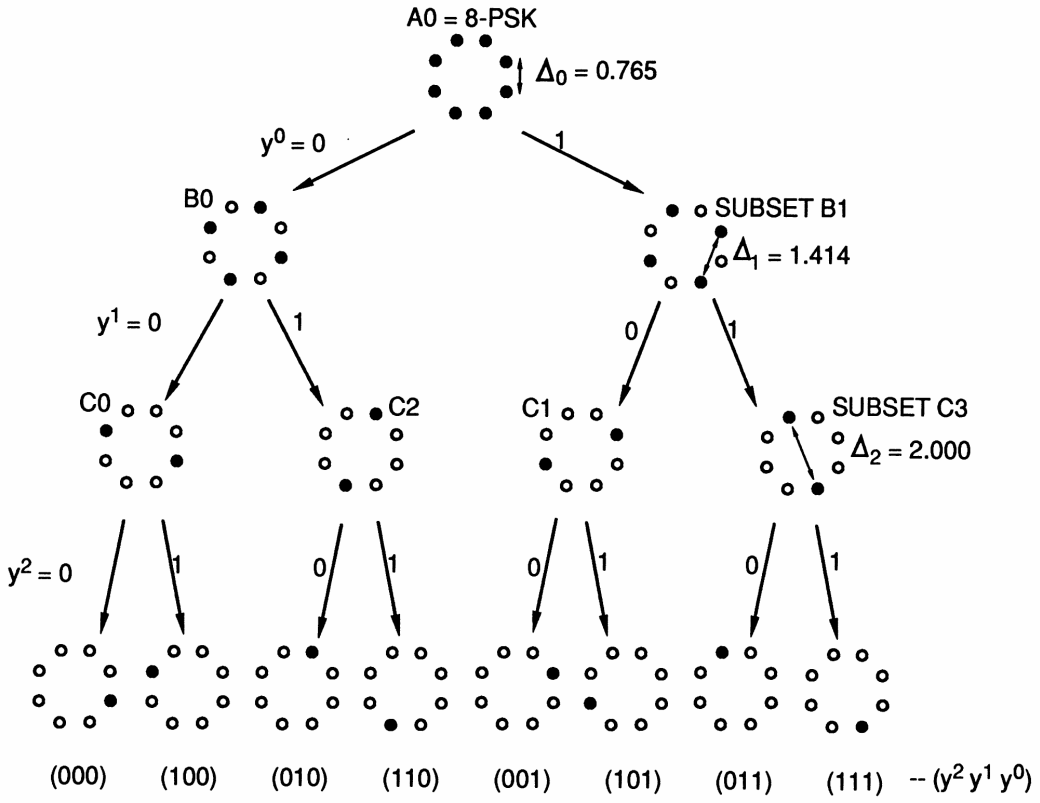

Figure 7: Subset partioning for 8-PSK

trellis codes:

1) each of 8-PSK signal should occur regularly and with equal probability.

2) transitions starting from the same state come from either B0 or B1.

3) transitions joining in the same state receive signals either from subset B0 or B1.

4) parallel transitions receive signals either from subset C0 or C1 or C2 or C3.

The technique is extended in a similar fashion for 16-QAM. The resulting signal map is in Figure 8. The transmission rate in this case is 3 bits/symbol using at rate 3/4 code. This will be compared to uncoded 8-PSK as a reference.

The encoders are of two forms as illustrated in Figure 9. In [38] it is shown that every convolution encoder has a equivalent systematic encoder which may contain feedback. Furthermore, the systematic encoder can be realized with the same number of memory elements as the minimal feedback-free encoder. The feedforward encoder can be realized directly from the parity generator polynomial, $G(D)$ and the dual systematic feedback code is taken from the parity check polynomial $H(D)$, where D is the delay operator similar to z^{-1} in z-transforms. The two functions are related by

$$G(D)H(D) = \mathbf{0} \tag{16}$$

Algebraic details can be found in [38]. It is worth mentioning when $G(D)$ represents a non-systematic feed-forward code it may be very difficult to find

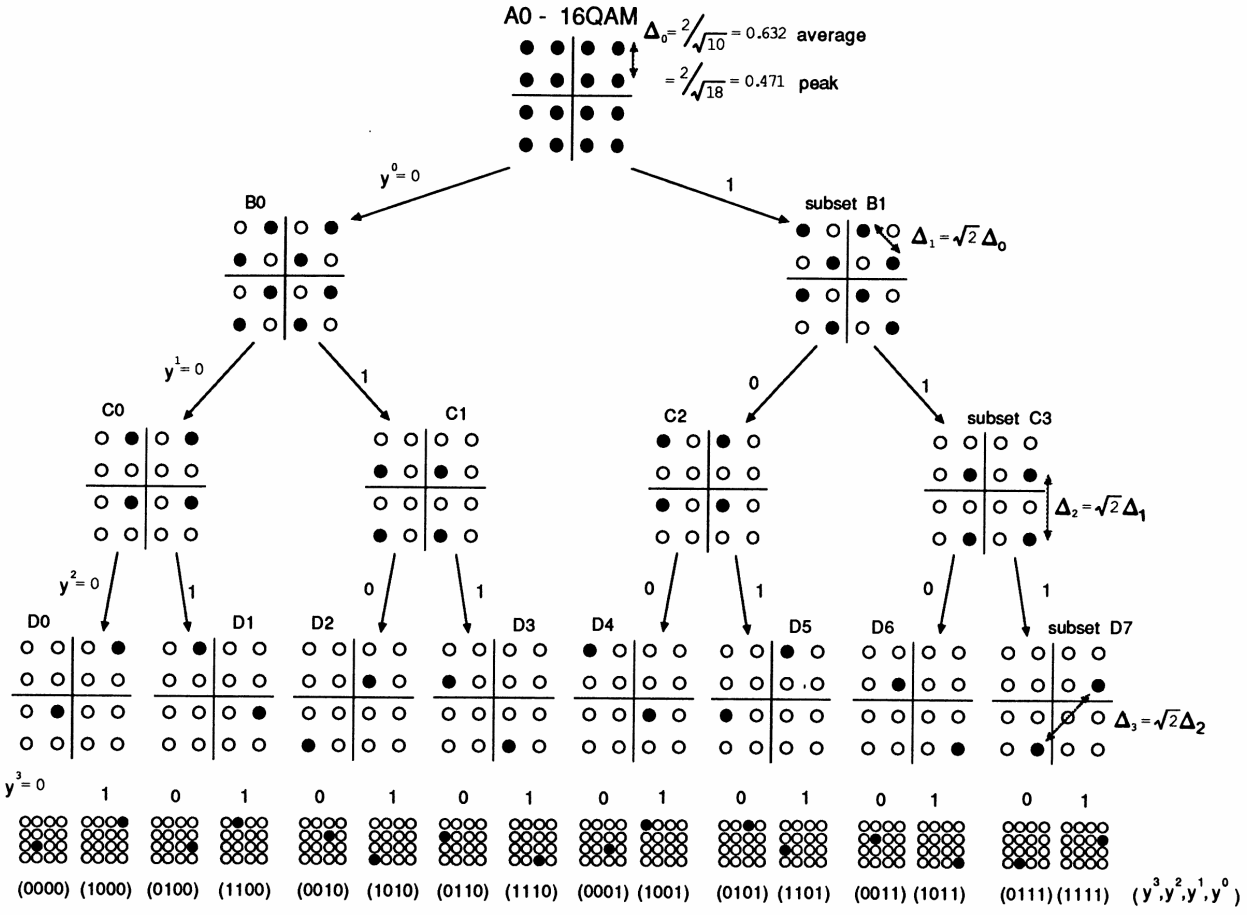

Figure 8: Subset partioning for 16-QAM

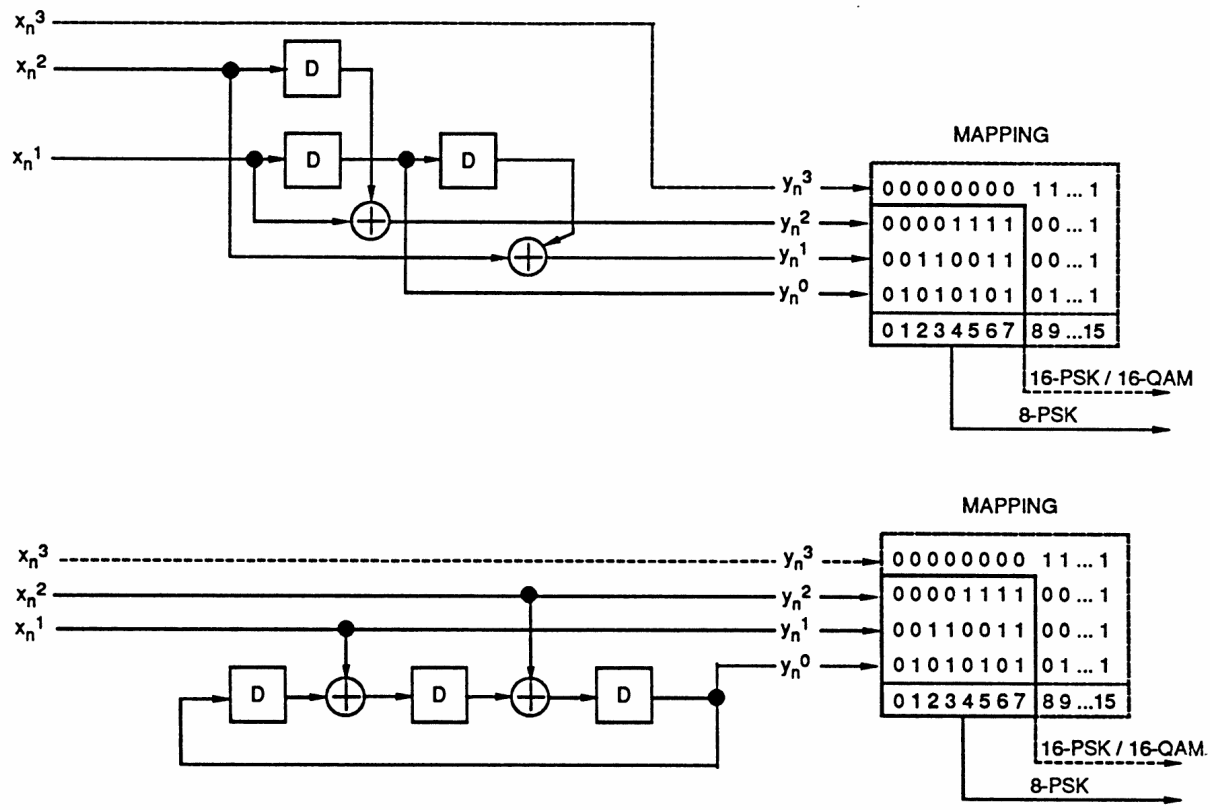

Figure 9: Convolutional encoders a) feedforward version b) systematic feedback version

$G(D)$ from $H(D)$ and vice versa.

While the two forms provide equal performance, it is advantageous to search for good systematic codes because they cannot have catastrophic error propagation. This greatly reduces the search space and the associated computation time. In addition the systematic code may be preferred since the data is available at the receiver when the channel is clean. This could be taken advantage of during periods when the channel is noise-free to free-up computational power.

The codes used in this study are in Tables 1 through 3. Codes for 4-PSK

Constraint Length ν	CCV octal		Asympt coding gain (dB) $G_{4PSK/BPSK}$
2	5	7	3.98
4	23	35	6.02
6	133	171	7.00

Table 1: Rate 1/2 feedforward codes

are standard convolutional codes of rate 1/2 from Odenwalder [15]. The implementation of these codes is optimal since Hamming distance (HD) equals Euclidean distance for gray mapped 4-PSK. These codes will be implemented in non-systematic feed-forward form. The rate 2/3 and 3/4 codes are from Ungerboeck [24]. These codes are for use with 8-PSK, 16-PSK and 16-QAM. The codes will be implemented in the systematic feedback form as given in [24]. Also, this study has avoided the use of rate 2/3 codes for 8-AMPM from [14]. Although 8-AMPM performs similar to QAM in the uncoded case and has an R_0 advantage over 8-PSK by 1.3 dB, the codes found for 8-PSK out-perform coded 8-AMPM for constraint length less than $\nu = 9$. Subsequently, 8-AMPM codes were dropped from Ungerboeck's follow-up paper [24].

Tables 1 through 3 also include the asymptotic coding gain provided by each code. This is determined by the increase in free Euclidean distance over the uncoded case. The free Euclidean distance is the distance of the minimum path in the code trellis which diverges and remerges with the all zero path. The free

distance of the uncoded case is Δ_0 for the reference modulation technique. The gain is calculated by example for 2 bits/symbol according to

$$20\log(d_{\text{free}}(\text{8PSK})/\Delta_0(\text{4PSK})) \tag{17}$$

For the 4-PSK codes the free distance is calculated by noticing that

$$d_{\text{free,ED}} = \sqrt{2d_{\text{free,HD}}} \tag{18}$$

The event error probability can be determined with knowledge of d_{free} and $N(d_{\text{free}})$, the average multiplicity of error events with distance d_{free}. Using maximum likelihood decoding, the error-event probability will asymptotically approach the lower bound [16]

$$\Pr(e) \gtrsim N(d_{\text{free}})\text{erfc}\frac{1}{2}(d_{\text{free}}/2\sqrt{2}\sigma) \tag{19}$$

More accurate descriptions of trellis-coded modulation performance can be found in [39]. The preceding equation was also used in Figures 4 and 5 for determination of the SNR_i which yields $\Pr(\varepsilon) = 10^{-5}$ for uncoded modulation. In this case the multiplicity of errors is the number of nearest neighbors. The calculations are documented in Appendix C.

4 Source Encoding and Performance Evaluation

In the previous sections the means for data transmission have been developed. The discussion will now focus on the data to be transmitted. The presentation has been made in this order because the emphasis in this study is to consider source coding for bandwidth efficiency as opposed to a pure data reduction approach. The data source under consideration is the one-dimensional first-order Gauss-Markov process which is one of the basic models for a correlated data source. It is described according to

$$S_i = \rho S_{i-k} + W_i; \quad \text{for } i \geq 0 \tag{20}$$

where ρ is the autoregression or correlation coefficient and must be in the range $0 \leq |\rho| \leq 1$ and W_i is a sequence of independent and identically distributed (i.i.d.) random variables with common variance σ_w^2. The sequence $\{S_i\}$ then has autocorrelation given by $R_{SS}(k) = \sigma_w^2 \rho^k/(1-\rho^2)$ and the variance is therefore $\sigma_S^2 = \sigma_w^2/(1-\rho^2)$.

The DPCM system block diagram is shown in Figure 10. The system uses linear prediction to determine future values of S_i as

$$\hat{S}_i^+ = \sum_{k=0}^{K} \alpha_k Y_{i-k} \tag{21}$$

The prediction error is minimized when the prediction filter is equivalent to the system model [18], therefore we should have $\alpha_0 = \rho$, $\alpha_k = 0$ for all $k \neq 0$. Although in noisy systems it may be optimal to choose other values of ρ [19]. In

this study it is sufficient to maintain $\alpha_0 = \rho$.

The DPCM encoder operates by making a prediction, $\hat{S}_i^+$, of the current value of S_i using past values of S_i and produces the difference between the predicted and actual values in the sequence E_i. The difference sequence is quantized and transmitted over the channel for remote reconstruction at the receiver. The sequence E_i has a much lower variance than does S_i and it is possible to represent it with less bits than S_i would require to maintain similar reproduction accuracy. In doing so, the data rate is reduced but the sensitivity to channel errors may be increased.

The following is an overview of the DPCM system . The quantity

$$Y_i = S_i + Q_i \tag{22}$$

represents the local estimate of S_i which is applied as the input to the predictor. In the absence of channel errors this is identical $\hat{S}_i^+$, the reproduced estimate of S_i transmitted to the destination. The error incurred in this case are due only to the instantaneous quantization error Q_i. For our specific example with $\alpha = \rho$, in the absence of quantization errors the quantity

$$\hat{S}_i^+ = \rho Y_i \tag{23}$$

represents the causal least mean-square (LMS) predicted estimate of S_i. Notice that the quantization error will generally cause $\hat{S}_i^+$ to differ from S_i. To optimize the prediction made at the decoder , the predictor in the encoder does not make

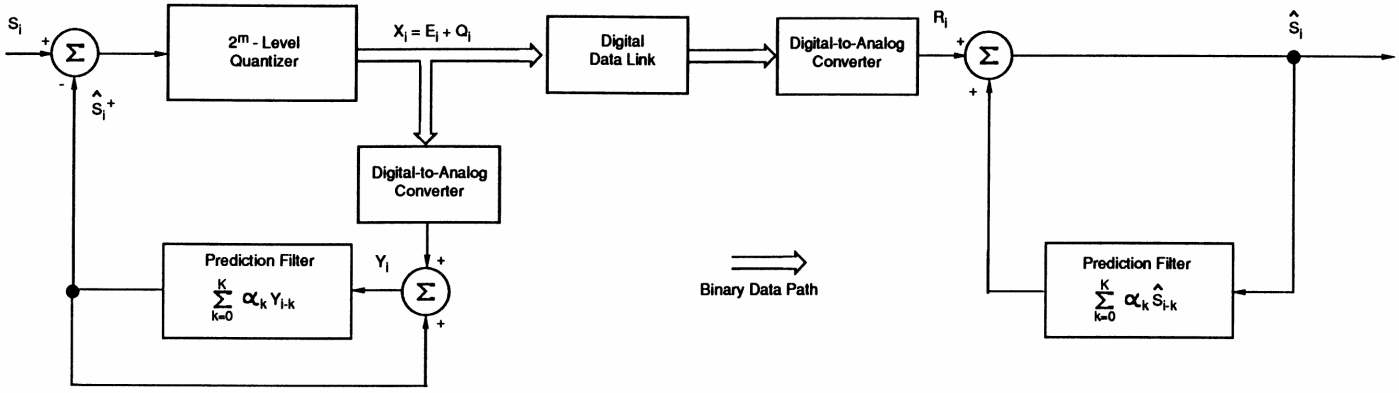

Figure 10: DPCM system block diagram

Number of Output Levels Q	Normalized Step Size δ	Mean Square Quantizer noise
2	1.596	0.3634
4	0.9957	0.1188
8	0.5860	0.03744

Table 4: Optimum uniform gaussian quantizer

use S_i. Instead, it makes use of Y_i which is identical to information available at the decoder when no channel errors occur.

The transmitted sequence $\{X_i\}$ is represented by an m-bit uniform quantizer. Since the signal is gaussian it can be represented by one of $Q = 2^m$ level from a Max quantizer [25] as

$$X_l = [l - (Q-1)/2]\delta\sigma_e; \quad l = 0, 1, \ldots, Q-1 \tag{24}$$

where σ_e is the standard deviation of the error sequence $\{E_i\}$ and δ is the optimal step sizes determined by Max in relation to normalized σ_e. For a given number of quantizer levels the proper values of δ are shown in Table 4. The Gauss-Markov source, when used with the matched predictor in the DPCM scheme yields $\sigma_e = \sigma_w$. Note that the phrase "m-bit DPCM" will used in this document to refer to a DPCM encoder containing an m-bit quantizer.

The quantizer in the DPCM system outputs symbols with a natural binary mapping. For more than 2 levels, the quantizer outputs bits which are not of

equal weight. An error to the most-significant bit (MSB) will cause more distortion than a least significant bit (LSB) error. The natural mapping is such that an MSB makes a greater contribution to the variance of the DPCM encoder output than does an LSB. In developing a transmission system it may be beneficial to minimize MSB errors during transmission at the expense of LSB errors.

4.1 System performance measurement

The validity of a source encoding method must verified by a measure of quality. In voice and image systems the criterion may be a subjective one and difficult describe mathematically. In this study the system performance is based on a channel input signal-to-noise ratio (SNR_i) versus output signal-to-noise ratio (SNR_o). The results are quite similar in form to the R_0 curves. The SNR_o is calculated on a mean square reconstruction error (MSRE) basis. The mean-square error is the most common performance measurement in statistical systems and has been used extensively to analyze DPCM and Gauss-Markov systems [19][27]. It is also attractive from a mathematical standpoint because it easy implement and analyze.

The output signal-to-noise ratio (SNR_o) in decibels is defined as:

$$\text{SNR}_o = 10\log(\sigma_s^2/e_T^2) \tag{25}$$

where σ_s^2 is the variance of original signal produced by the source sequence $\{S_i\}$,

and

$$e_T^2 = E\{[S_i - \hat{S}_i]^2\} \tag{26}$$

is the mean-square reconstruction error between the original source data and the data reproduced at the user.

The MSRE can be expressed in terms of three components [19]

$$e_T^2 = \epsilon_q + \epsilon_m + \epsilon_c \tag{27}$$

here ϵ_q represents the quantization error. Similarly, ϵ_c represents the error due to channel noise. The term ϵ_m is a mutual error term which has been found to be much less significant in DPCM systems with respect to the other terms [19]. In a more general sense to all source encoding techniques, ϵ_q reflects the information efficiency of the source coding technique and ϵ_c reflects the sensitivity of the source coding technique to channel errors.

The input signal-to-noise ratio (SNR_i) is based on the signal energy per channel use in the presence of AWGN noise with double-sided spectral density $N_0/2$. The comparison between the different modulation techniques is accomplished using energy per symbol as opposed to energy per bit. For 2-D channel symbols the input channel signal-to-noise ratio in decibels is defined according to

$$\begin{aligned} \mathrm{SNR}_i &= 10\log\left[\frac{2E_{\mathrm{N}}}{N_0/2}\right] \\ &= 10\log\left[2E_s/N_0\right] \end{aligned} \tag{28}$$

Figure 11 illustrates the system performance using uncoded modulation for the first-order Gauss-Markov source with autoregression coefficient $\rho = 0.95$.

The gaussian quantizer was used for 1,2 and 3 bits per DPCM symbol. The data was transmitted as one source symbol per channel symbol by using BPSK for 1 bit, 4-PSK for 2 bits and 8-PSK for 3 bits.

Included in Figure 11 is the rate-distortion bound and R_0^* bounds on SNR_o for operation over the AWGN channel. As demonstrated in [1] the rate-distortion bound for the Gauss-Markov source with a mean-square fidelity criterion is equated to the channel capacity of Equation (12) and is used to solve for SNR_o in terms of SNR_i. The same technique is used to obtain the equivalent R_0^* bound in terms of maximum SNR_o. The development of the 1-D Gauss-Markov with a mean-square error criterion is from Berger [27]. The rate-distortion bound can be expressed parametrically as

$$D = \left(\frac{1}{2\pi}\right)^2 \int_{-\pi}^{\pi} \min\{\theta, \mathcal{S}(\lambda)\} d\lambda \tag{29}$$

and

$$R(D) = \left(\frac{1}{2\pi}\right)^2 \int_{-\pi}^{\pi} \max\left\{0, \frac{1}{2}\log_2 \frac{\mathcal{S}(\lambda)}{\theta}\right\} d\lambda \tag{30}$$

where D is the fidelity measure and $R(D)$ is the information rate in bits/sample at which D may be achieved. The term $\mathcal{S}(\lambda)$ is the discrete power spectral density of the Gauss-Markov process

$$\mathcal{S}(\lambda) = \frac{(1-\rho^2)\sigma_s^2}{1 - 2\rho\cos\lambda + \rho^2} \tag{31}$$

The rate-distortion bound and R_0^* bound have been solved for by using the

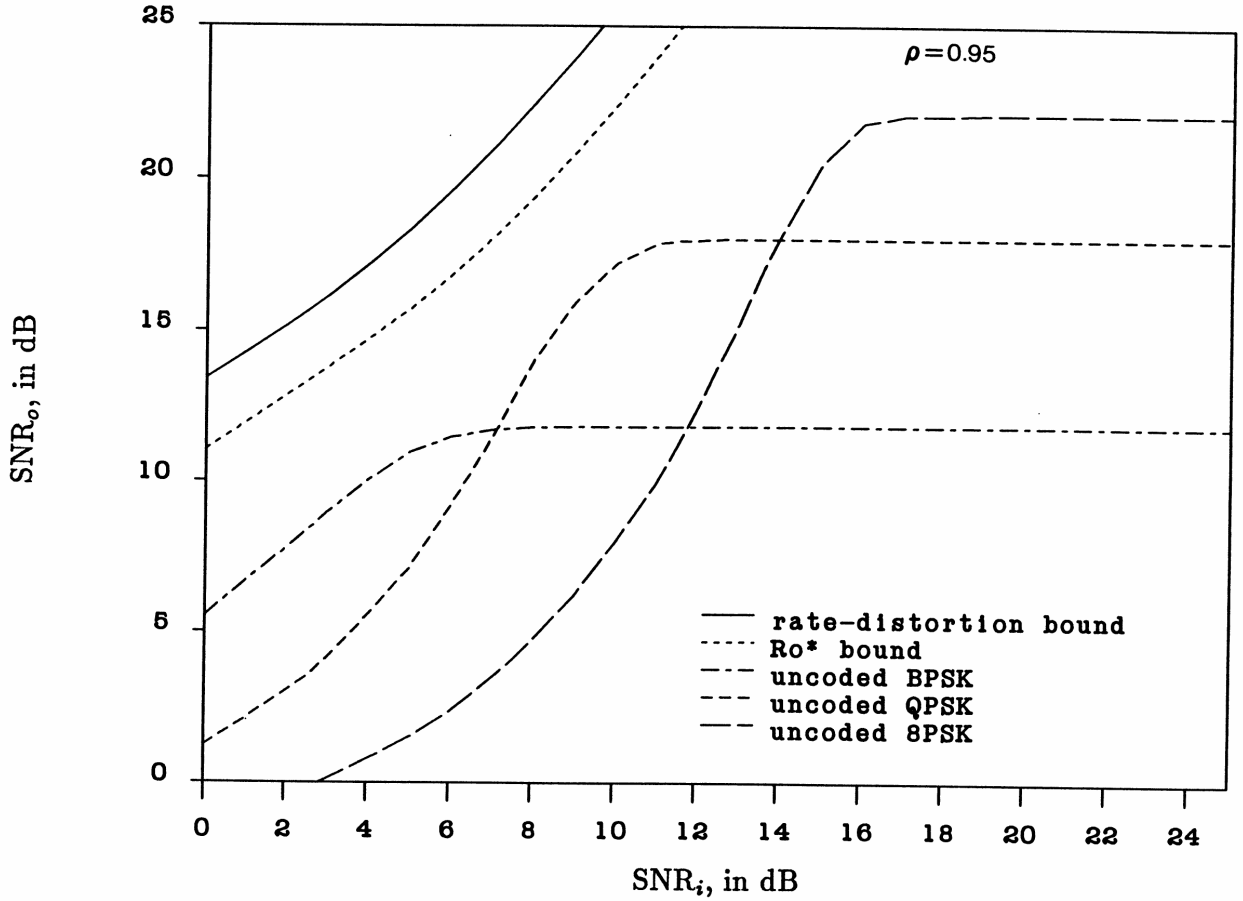

Figure 11: Uncoded system performance

MathCAD© [4]software package and are documented in Appendix C. By using Equation (12) the expression

$$R(\sigma_s^2/\mathrm{SNR}_o) = \frac{\mathsf{N}}{2}\log_2\left[1 + \frac{\mathrm{SNR}_i}{\mathsf{N}}\right] \tag{32}$$

allows a value for SNR_i to determine $R(D)$ which is then used to "look-up" the corresponding D on the rate-distortion curve. The output signal-to-noise ratio is defined similar to Equation (27) as $\mathrm{SNR}_o = \sigma_s^2/D$. The R_0^* bound can also be solved for in a similar manner.

The uncoded system performance curves of Figure 11 resemble the R_0 curves. For each quantization level, the characteristic linear and saturation regions exist. In the linear region the noise has the greatest effect on system performance, here performance is proportional to SNR_i. In the saturation region relatively few errors are made and the performance is due solely to quantization error. At low SNR_i the channel can only support 1 bit per channel use and 1-bit DPCM with BPSK performs best, providing SNR_o =11.8 dB at saturation. As SNR_i is increased above 12 dB the channel can now support 2 bits per use and SNR_o =18 dB is available with the 2-bit DPCM over 4-PSK combination. Similarly, above SNR_i =18 dB the system can achieve SNR_o =22.1 dB per symbol with 3-bit DPCM using 8-PSK. The performance at this level is due only to quantization noise and the 3-bit system is 10.3 dB and 4 dB above the 1 and 2 bit systems, respectively and clearly it would be wasteful of channel capacity to use them for

[4]MathCAD is copyright protected by MathSoft, Inc.

high SNR_i.

For low channel signal-to-noise ratio the situation reverses. At $SNR_i = 9$ dB for example, the 1-bit system provides 11.8 dB. Use of the 2 bit system can provide 16 dB but the performance is in the linear region and highly sensitive to channel noise fluctuation, while the 3-bit system would yield less than 6.5 dB of reconstructed signal-to-noise ratio at the receiver. This differs from the R_0 results which show that an increase in the alphabet size of an optimized modulation scheme does not decrease the information throughput. The performance loss should be attributed to the sensitivity to channel noise of the source encoding technique. Performance of the system can be improved by either seeking source encoding methods with less noise sensitivity or by decreasing the effect of channel noise. In Section 6 the later will be addressed and shown to have significant improvement by using trellis coding.

4.1.1 Source coding performance bound

In the previous section the rate-distortion bounds gave limits on the system performance based on the channel quality and the specific source model. The bound was independent of the source encoding technique. In this section a simple approximation to the performance of a source encoder for the Gauss-Markov source is given.

The upper bound for the output signal-to-noise ratio of Gauss-Markov source represented by m bits per symbol is found simply by finding the fidelity D which

Bits per symbol	Rate-Distortion bound (dB)	Quantization Noise bound (dB)	Simulation Result (dB)
1	16.1	14.5	11.8
2	22.2	19.4	18.0
3	28.2	24.4	22.1

Table 5: Source coding performance bounds, $\rho = 0.95$

corresponds to rate $R(D) = m$ bits from the rate-distortion function. Since the function is described by parametric equations of integrals, an accurate approximation was more easily determined by using a cubic spline method to interpolate specific values. The results are shown in Table 5. The values exceed the simulation results by 4.2 to 6.1 dB.

The difference is attributed to the efficiency of DPCM as a source encoding method. Its performance is limited by the quantization noise in the DPCM encoder. In [19] it is shown that quantization error term in Equation (27) is equivalent to the quantizer variance, i.e.

$$\epsilon_q = \sigma_q^2 \tag{33}$$

Using Equations (25) and (27) the value of SNR_o due to quantization noise for the noiseless channel is

$$\mathrm{SNR}_o = \sigma_s^2/\sigma_q^2 \tag{34}$$

For the uniform Max quantizer the values of σ_q^2 for 2,4 and 8 quantizer levels

are given in [25]. The corresponding SNR_o values are in Table 5 and represent a performance bound on DPCM using Max quantizing for the various quantization levels. The results are within 2.7 dB of the simulated values and accounts for most of the difference from the ideal source encoder. The remaining difference may be due to the actual probability density function of the DPCM error sequence which is not truly gaussian since it is effected by the the propagation of quantization errors.

While the DPCM system is limited to the quantization noise bound, it is possible to achieve greater results using other source encoding methods. An adaptive DPCM system makes it possible to reduce the quantization noise. In [6] a comparison of trellis source coding methods are shown which perform above the Max quantizer bound but below the 1-bit rate-distortion bound. It is important to realize that in this study an alternative is offered to devising complex source encoders which approach the 1-bit rate-distortion bound. That alternative is to increase channel signal-to-noise ratio and use 2-bit DPCM and operate above the 1-bit bound.

5 Combined Source Coding, Channel Coding and Modulation

5.1 Fundamental concepts

A combined source coding, channel coding and modulation approach represents a logical extension to the combined source-channel coding of [1] and the combined channel coding and modulation of [14]. This research has focussed on analyzing such a combined system approach. The evaluation of a sample system is performed in an empirical study to promote interest for more advanced analytic work. A discussion is presented on the motivation and goals of the combined system and its differences from previous approaches.

Communication systems may be implemented from designs in which the source coder, channel coder and modulator are considered separate and independent subsystems. The resultant system may fall considerably short of what is achievable. In this study it is argued that a combined source coding, channel coding and modulation approach can optimize overall system performance. Massey had made a similar argument for the design of joint channel coding and modulation schemes. This position lead to the discovery of Trellis Coded Modulation.

One of the most important aspects of the combined system is that the design goal is to minimize distortion. This may differ slightly from the combined channel

coding and modulation system where the goal is to minimize the probability of a bit error. While both systems are designed to minimize errors, the distinction lies in how the two approaches minimize the effects of errors when they occur. One of the most tangible methods for minimizing the effect of errors is the use of Gray coding in an uncoded system. In Gray coding all nearest neighbors in a signal space have a hamming distance of 1 bit. This reduces the effect of typical channel noise since it will usually cause only one bit to be in error. In general, the approach should be to make the distortion proportional to the noise.

In many systems the source produces data with a natural mapping. This is true of most analog source encoders. Consider an example of a simple PCM system for the encoding, transmission and remote reconstructing a black and white image over an AWGN channel. The system assigns one octal digit per pixel with octal 0 representing white and octal 7 as black. Between 0 and 7 lies graduated shades of grey. A distortion measure is assigned as the absolute value between the transmitted and received signal values. So if a dark pixel represented by 6 is transmitted and channel noise causes a 4 to selected at the receiver, then a distortion of 2 is assigned. During the transmission of this data, we would like to minimize the average distortion. Using 8-PAM modulation and no coding, the natural mapping of Figure 12 is the best since it causes the distortion to be proportional to the noise for all symbols. The same situation using Gray coding is no longer optimum. For example if a 2 is transmitted, noise causing the selection of the right nearest-neighbor causes a distortion of 4 units.

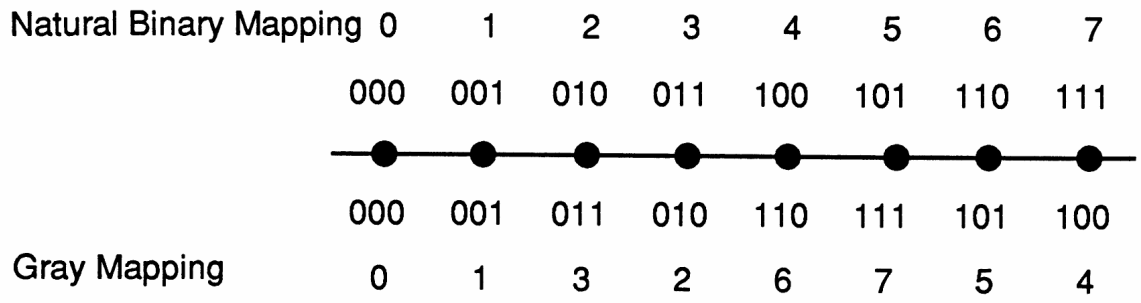

Figure 12: Channel mapping for example

For other modulation schemes the choice is not as obvious. When using 8-PSK, natural and Gray coding are almost equivalent because the natural map is penalized for having 0 and 7 as adjacent signals. The problem becomes more complex when coding is introduced. In a coded system, the concept of nearest neighbor may not exist as the erroneous output of the decoder may depend on both channel noise and the data sequence from the source.

5.2 System objectives

The objectives of a combined source coding, channel coding and modulation system can now be stated and discussed. The following may be simplistic, but it is used to develops some important points.

1) The goal of a combined source coding, channel coding and modulation system is to minimize the distortion of the reconstructed signal at the receiver.

2) The primary goal of channel coding in the system is to maximize free euclidean for the given modulation technique. The coding should also have a secondary goal to maintain "on average" the mapping relationship of the source encoder letters.

Item 1 implies that the components of the system must be optimized together. It also implies that the primary criterion used to evaluate the system should be an end-to-end performance measure which may be similar to source encoding evaluation methods, i.e., a fidelity criterion. In this approach individual components can be evaluated and compared by their performance within the system. For example, two source coding methods can be compared for their sensitivity to channel noise. While one method may be more efficient at achieving the rate-distortion bounds in a noiseless system, another method which is less sensitive to channel noise may provide better overall system performance in certain situations. Similarly, channel coding and modulation techniques can be compared on their ability to improve overall system performance.

Item 2 implies that channel codes can be sought from existing techniques but codes which are similar may provide different performance in conjunction with a source coding method. This is attributed to the code's ability to minimizes distortion of the source letters when channel errors are made. It may also be possible to find coding and modulation schemes whose performance is brought to light only within the context of the combined system. As discussed in section 4, such a scheme may increase the reliability of MSB transmission at the expense of lower order bits in a natural binary mapped symbol.

In [1] a technique was used to minimization the effect of channel error on symbols produced by a DPCM source. The system assumed BSPK modulation exclusively and therefore coding had to be used judiciously since it increased the

efficient transmission.

3) The use of DPCM as an effective method of source encoding. While only one method of encoding is analyzed, the use of DPCM allows for a comprehensive study of all system components parameters because of the ability to increase quantization and thereby increase system performance.

4) Rate-distortion tools to determine the performance limitations for the transmission of a Gauss-Markov source.

From these tools a system approach is taken. The rate-distortion bounds on capacity and R_0^* for the Gauss-Markov process in Figure 11 tell us what is possible for one channel use. The R_0 curves for specific modulation techniques tell us how to achieve near these bounds in practice. In [1] we found out that the system worked best when all bits were coded. That work concluded that good high rate codes were needed. By using high rate codes, such as $r = m/(m+1)$, all bits could be coded with a minimum in bandwidth expansion. The trellis codes presented provide the high rate codes needed and do so with no bandwidth expansion. In this study this is the approach taken, although other options may be desirable to trade bandwidth for a decrease in power requirements. If we consider the objective of this study to maximize fidelity to the user, then the purpose is to determine what is the best way to represent and transmit data from a given source in a fixed bandwidth for a given signal-to-noise ratio. An alternate viewpoint of the same situation is to determine what source quantization, bandwidth and signal-to-noise ratio are needed to achieve a desired fidelity at the receiver.

We can now review the basic system of Figure 1. The system takes a DPCM symbol quantized to m bits and transmits it over the channel with a rate $m/(m+1)$ trellis code using either PSK or QAM modulation containing 2^{m+1} signal points in its constellation. At the receiver, decoding is performed by a Viterbi decoder based on Euclidean distance. The Viterbi decoder provides maximum likelihood (ML) decoding of convolutional codes and is feasible for code constraint lengths less than ten. System performance is compared against a reference system of m-bit DPCM transmitted over uncoded 2^m-ary PSK. The choice of m depends on the available channel signal-to-noise ratio and the fidelity required. Increasing m can increase fidelity but only if sufficiently large signal-to-noise ratio is available. In this study values of $m = 1, 2$ and 3 are demonstrated. The values chosen allow for an informative and complete set of tradeoffs to be performed. The value $m = 1$ demonstrates a system with the least complexity and the least required channel power. While values above $m = 3$ may have practical use but repeat similar tradeoffs that add little insight to the study.

There are a number of tradeoffs to be investigated. The first issue is the effectiveness of the coded system. If the codes are effective then we should expect to approach the asymptotic coding gains predicted by in Tables 1 through 3. A true coding gain should replicate the uncoded system performance only offset at a lower SNR_i, as opposed to only improving fidelity (SNR_o) in the linear performance region. Included in this issue is the need to verify if the complexity due to increased constraint length is justifiable by an appropriate increased in

system performance.

The next tradeoff is the use of modulation technique. The tradeoff is most interesting at 3 bits/symbol, where a choice of uncoded 8-PSK, coded 16-PSK, coded 16-QAM and coded 12/4-PSK are available. Below 3 bits/symbol the options between coded systems provide little performance gain as compared to the simplicity of coded PSK modulation. Above 3 bits/symbol the PSK signals become excessively inefficient as compared to QAM; while at 3 bits/symbol a valid tradeoff exists.

A closely related topic is the peak-power versus average-power constraint imposed by the possible channel characteristics as discussed in Section 2. The PSK signals are not effected by this situation. The issue is the relative performance of the 16-QAM signal with the power constraint applied and the use of 16-QAM modified into 12/4-PSK to counter the situation. While R_0 has given insight into how the modulation schemes will perform, the actual performance may differ because of effects which only become obvious within the context of the combined system.

The last issue to be examined is that of the DPCM encoder output to trellis coder mapping. This will determine whether the trellis codes have a preferential mapping which best maintains the natural binary mapping of the source symbols, thereby reducing distortion. It may be difficult to explain such a phenomenon in analytics terms but the results will yield insight into the problem. If such a result exists then each trellis code should have an appropriate mapping associated with

it when transmitting weighted binary data.

Specific to this study, the use of Ungerboeck's trellis codes represent one solution for how the DPCM symbols should be coded and mapped onto the channel. Robust performance from the trellis code is expected since all bits from the source encoder are channel coded. The mapping by set partitioning specifies the coder to signal space relationship. This leaves how the DPCM symbols are applied to the trellis encoder. The Ungerboeck codes used all have a uniform error probability (UEP). The UEP property implies that each coded bit has an equal probability of error independent of the mapping. In the next section we shall see that a mapping which follows the set partioning produces the best results. Since the codes are systematic the MSB specifies the subset with the greatest intra-signal distance and the LSB specifies the subset with the smallest intra-signal distance (the subset at distance Δ_0 is always specified by the "parity bit", y_0, output by the convolutional encoder embedded in the trellis coder).

6 Simulation results

The use of a system simulation allows for a more complete understanding of the interaction between components which make-up the total system. The result of a simulation is of course only as good as the model on which it is based. The details of the implementation are in Appendix A. For clarity all system parameters assumptions are restated here:

1) The Gauss-Markov source uses autoregression coefficient $\rho = 0.95$ and its random input sequence $\{W_i\}$ is defined by $\sigma_W^2 = 1$ and $\mu_W = 0$.

2) The DPCM encoder and decoder use first-order predictors with $\alpha = 0.95$

3) The channel has infinite bandwidth and is an AWGN channel with a power spectrum of $S(\omega) = N_O/2$

4) The Viterbi decoder uses soft-decisions maintained at 100 levels of quantization. This leads to performance which is indistinguishable from infinite quantization. A decoding depth of 100 bits is used.

5) The simulation results are based on 100,000 symbols iterations per point. Points were simulated at the following values of SNR_i: $E_s/N_0 = 0, 1, 2, \ldots, 25$ dB and $E_s/N_0 = 2.5, 6.5, 11.5, 12.5$ and 13.5 dB.

The simulation results are in Figures 13 through 20. The coefficient $\rho = 0.95$ implies a highly correlated source which is typical of voice and image signals. The simulation of 100,000 source symbols was chosen per the $10/\Pr(e)$ rule discussed in Appendix A so as to guarantee accurate results for $\Pr(e) > 10^{-4}$.

As demonstrated in Appendix A, this choice leads to a 99% confidence that the results are within 6% of the actual probability. Probabilities of error smaller than 10^{-4} are visually indistinguishable when using the mean square error criterion. Also, the size chosen would be a practical example for a frame of image data.

6.1 Effect of coding and constraint length

The improvement due to coding with various code constraint length is demonstrated in Figures 13 through 16. Simulations for codes of constraint length $\nu = 2, 4$ and 6 were performed. The decoding complexity grows exponentially with constraint length and $\nu = 6$ is a good limitation on practical complexity. In the following section ***coding gain*** will be used to refer to the difference between the value of SNR_i that can be used by two different techniques to achieve identical fidelity (SNR_o) by using a different combination of either source resolution, code constraint or modulation type. While ***performance gain*** will refer to the increased fidelity, measured in SNR_o, that can be achieved by different systems at the same SNR_i.

The following characteristics can be observed: the coded systems offer improvements which approach the 3 to 7 dB asymptotic coding gains predicted in Section 2. In each case the saturation level of SNR_0 is reached at a much lower value of SNR_i. The saturation effect is quite similar to the R_0 function. Notice that as the constraint length is increased the saturation effect becomes much

sharper. Also, most of the coding gain is achieved by the short $\nu = 2$ codes. The larger constraint lengths add an additional 1 to 2 dB of coding gain.

The 1-bit DPCM system with rate 1/2 and $\nu = 2$ code for 4-PSK has a 3.5 dB coding gain over the uncoded system with BPSK. Alternatively, the coded system may be viewed as having a 1.8 dB performance gain over the coded system at $\text{SNR}_i = 4$ dB. The $\nu = 4$ code adds another 0.5 dB of coding gain and the $\nu = 6$ is approximately 1.1 dB superior to the $\nu = 2$ code. At this point performance is less than 2 dB from the R_0^* bound and it is expected that any other method used can add only marginal improvement. Part of the improvement of the coded 4-PSK is due to the extra dimension which BSPK cannot access as was discussed in the Section 2.

These results are similar to [1] where 1-bit DPCM was coded onto two uses of BPSK with the use of the same rate 1/2 codes. Although that work was for two-dimensional image information, a relative comparison in terms of coding gain and proximity to R_0^* is justifiable. In this case the two approaches are basically the same since the uncoded data is transmitted via BSPK and the coded scheme uses two channel uses. In this study the modulation is 4-PSK formed by two orthogonal BPSK signals.

In Figure 14 the 8-PSK trellis coded system with 2-bit DPCM offers superior performance over the 1-bit systems when $\text{SNR}_i > 8$ dB is available. By going to 2-bit DPCM the system performance can be increased to $\text{SNR}_o \approx 18$ dB.

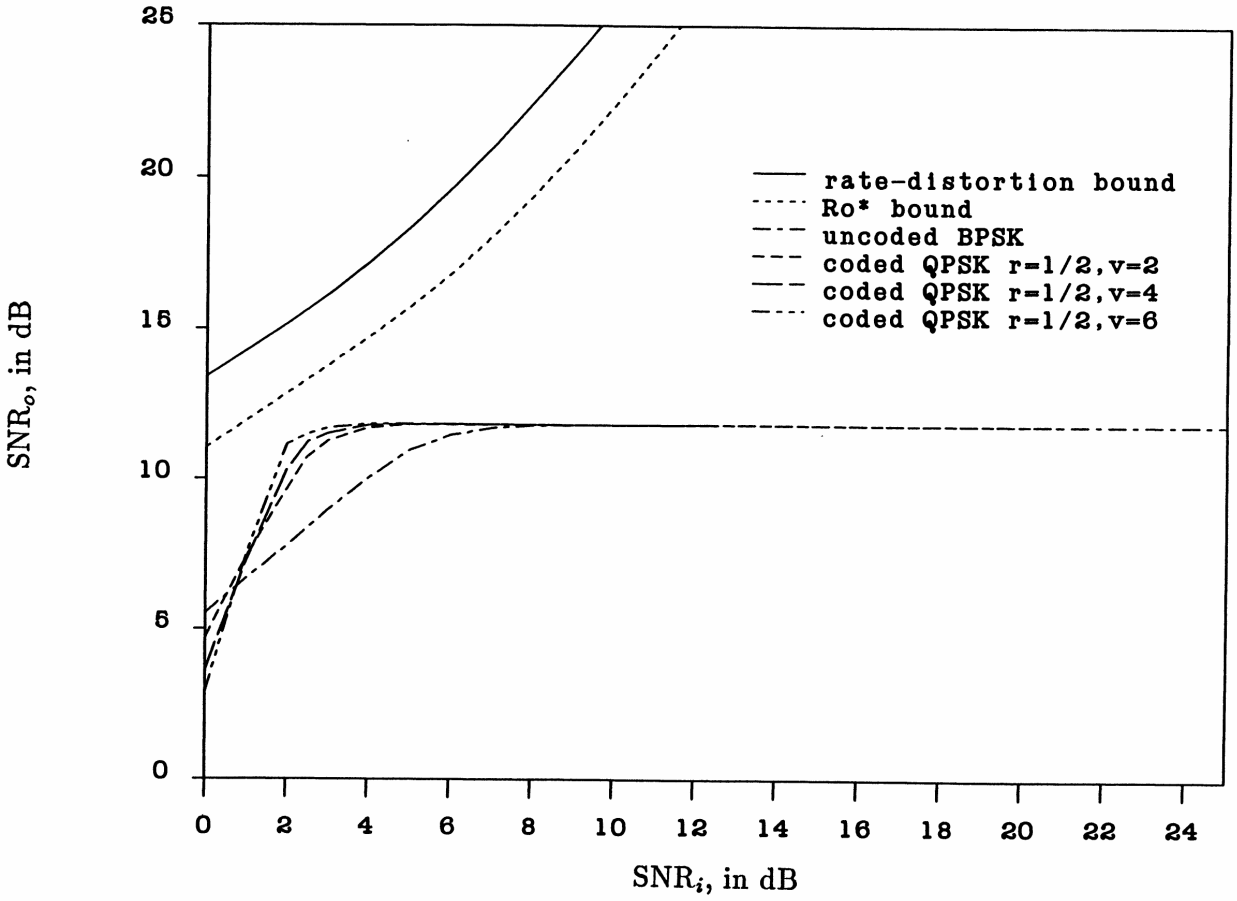

Figure 13: System performance:1-bit DPCM and 4-PSK

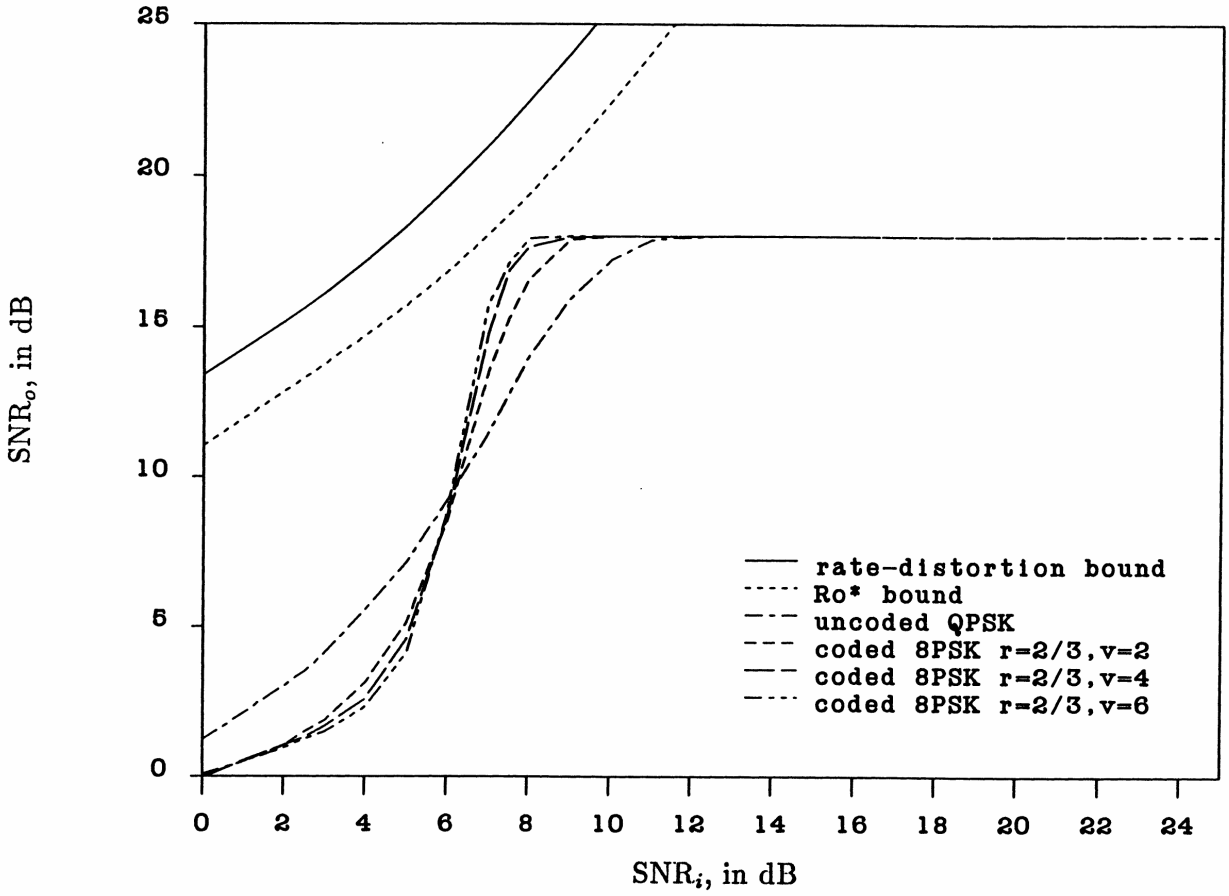

Figure 14: System performance: 2-bit DPCM and 8-PSK

The coded 8-PSK has a 2 dB coding gain over the uncoded 4-PSK system at $SNR_i = 9$ dB for $\nu = 2$. It is reasonable to expect that for larger SNR_i this gain will approach the asymptotic coding gain of 3 dB which is predicted in Table 2 An additional coding gain of almost 4 dB occurs with constraint length 6 and this is within 1 dB of the R_0^* bound. For all of the results in Figure 14 the DPCM symbols were mapped to the encoder as MSB= x_2 and LSB= x_1. The alteration of the mapping will be discussed later. In comparison to [1] a true equivalent can not exist. This is because there is no added dimension or bandwidth associated with the uncoded or coded systems. Instead a relative comparison to a system with a bandwidth expansion is made.

In [1] a 2-bit DPCM symbol is transmitted via an uncoded system using 2 BPSK channel uses and the performance saturates at $SNR_i = 16$ dB or 8 dB from $2 \cdot R_0^*$. By using a rate $r = 1/2$ and $\nu = 6$ coding on each bit and then expanding to 4 BPSK channel uses, the same performance was achieved at $SNR_i = 11$ dB or 4 dB from $4 \cdot R_0^*$. This is a 5 dB coding gain or a 50% gain relative to R_0^*. In contrast, the system in Figure 14 saturates at only 12 dB for the uncoded system and 8 dB for the coded system with $\nu = 6$. While the coding gain is only 4 dB, the improvement relative to R_0^* is 80% with no bandwidth expansion.

The results for 3-bit DPCM transmitted using 16-PSK trellis coded modulation are shown in Figure 15. The use of 8 level quantization brings performance up to $SNR_o \approx 22$ dB when $SNR_i > 13.5$ dB is available. As anticipated by R_0, 16-PSK becomes inefficient as compared to R_0^*. The $\nu = 2$ code has an 3 dB

gain over the uncoded system which is in agreement with the asymptotic coding gain. Another 1 dB can be achieved with the $\nu = 6$ code. This is about 3.5 dB from the R_0^* bound.

In comparing again to [1] the 3-bit system here offers performance gains which were not possible in the other approach. A later study of the block cosine transform [2] introduced higher rate codes but no specific results for rate 3/4 codes using two channel symbols are available. A result in [1] was obtained for 3-bit DPCM transmitted over 4 BPSK channel uses with a 1/2 rate code on the MSB. Fidelity improvements were found only in the linear region and not in the saturation region.

6.2 QAM versus PSK

Some of the inefficiency of 16-PSK relative to R_0^* can be made up by using 16-QAM in Figure 16 which makes more efficient use of the signal space. The use of 16-QAM TCM is generally 1.5 dB more efficient than 16-PSK TCM for codes of equal constraint length when an average power constraint is applied. The constraint length $\nu = 6$ 16-QAM code is within 2.2 dB of the R_0^* bound. Notice that at low SNR_i the QAM performance falls off more sharply than does PSK, this may have to do with the number of channel letters which are accessible for a given intensity of noise. In PSK the number grows roughly proportional whereas in QAM the number grows more rapidly.

Again as in the uncoded case it is important to realize that using increased quantization levels improves system performance only when there exist sufficient transmitter power. For example, at $SNR_i = 5$ dB R_0^* implies that 15.6 dB is obtainable. For codes with $\nu = 2$, the 1-bit system with 4-PSK TCM achieves $SNR_o = 11.8$ dB while 2-bit DPCM with 8-PSK produces only 5 dB performance at the receiver. The performance is poorer for 3-bit DPCM, which yields only 3 dB with 16-PSK and SNR_o less than zero with 16-QAM.

6.2.1 Average versus peak power constraint

As discussed previously, a peak power constraint is sometimes considered a more realistic model of a practical channel. For the peak constraint the channel signals should lie on or within a circle representing the peak power. This means that the PSK signal set remains the same but the QAM intra-set distance must be reduced from $2/\sqrt{10}$ to $2/\sqrt{18}$ which results in a 2.5 dB loss from the average QAM signal set. The relationship is shown in Figure 17. For the peak constraint, QAM looses its superiority to PSK. This result has been verified in studies of non-linear channels [13], which is similar to the peak constraint because the transmitter power is "backed off" and limited into a linear region of the channel.

For the peak power constraint channel some of the losses incurred by 16-QAM may be recovered by using the 12/4-PSK constellation of Section 2. One of the attractive properties of 12/4-PSK is that intra-signal distance are only increased over peak constrained 16-QAM so that codes developed for 16-QAM

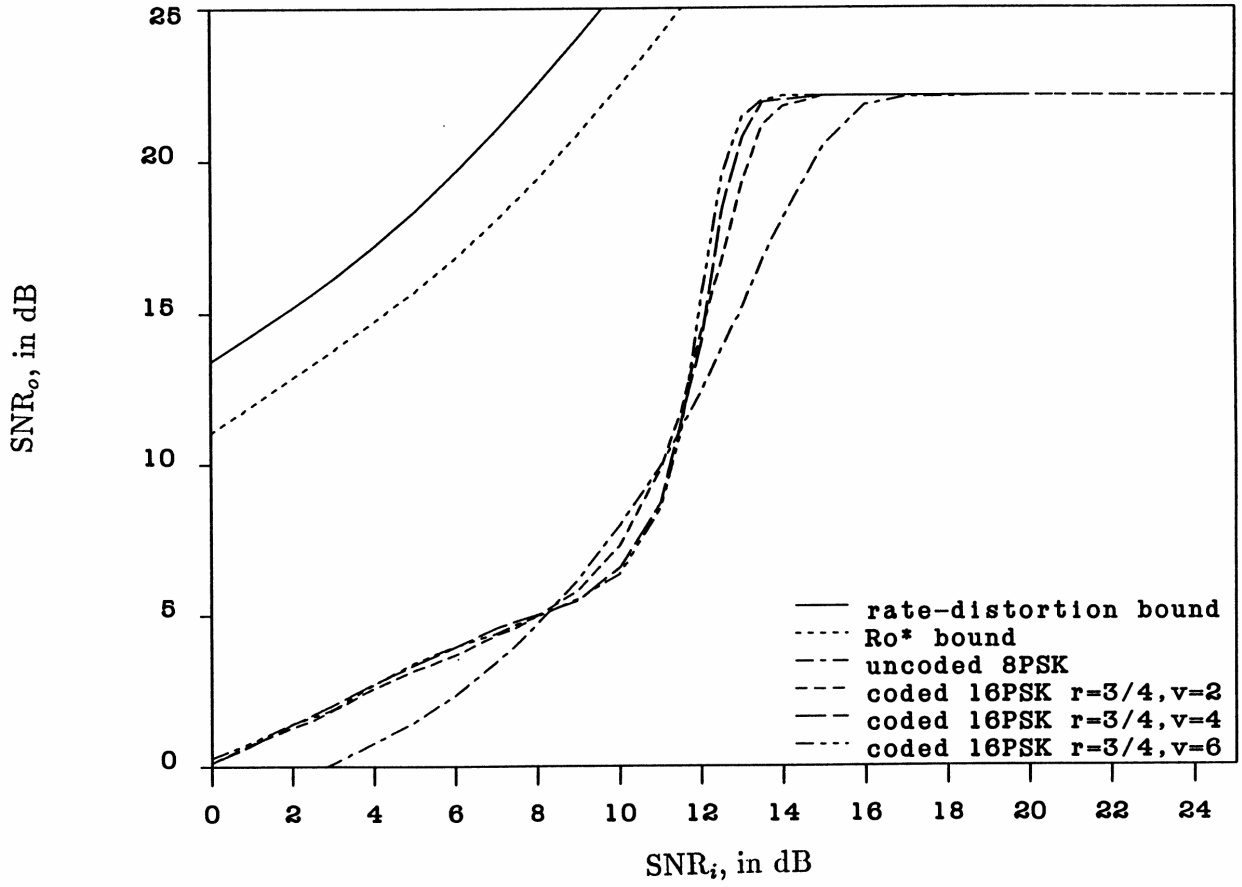

Figure 15: System performance: 3-bit DPCM and 16-PSK

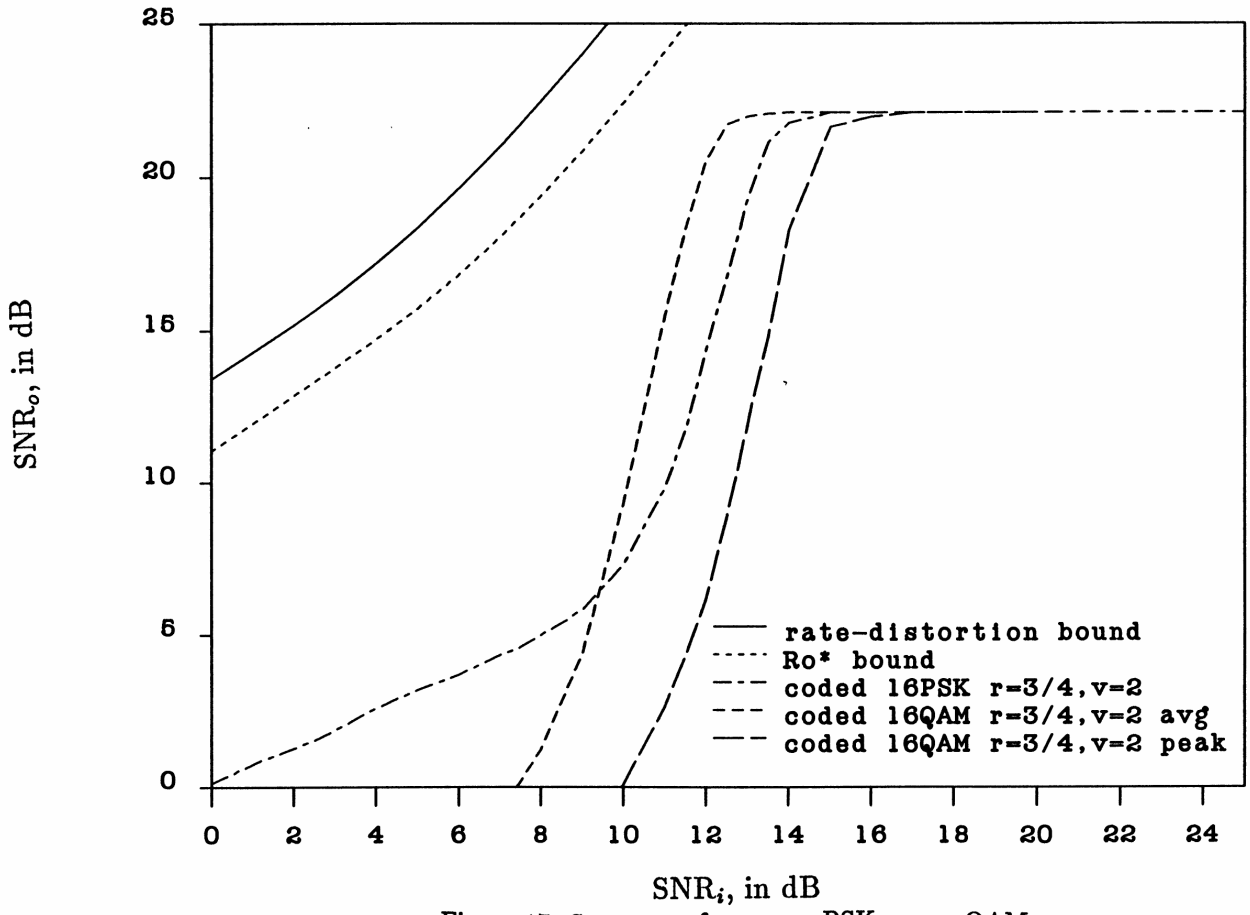

Figure 17: System performance: PSK versus QAM

can be used without modification. The result in Figure 18 is that 12/4-PSK is slightly superior to 16-PSK, about (0.5 dB). This can be compared to the 1 dB difference in R_0 between 16-PSK and 12/4-PSK in Figure 6 at $\mathrm{SNR}_i = 13$ dB.

6.2.2 Effect of bit mapping

In Figures 19 and 20 the effect of the bit mapping between the DPCM quantizer and the trellis coder is investigated. In both cases the greatest effect is at low SNR_i values before the threshold effect occurs. In 2-bit DPCM with 8-PSK TCM the effect diminishes at the threshold, which should always be operated above. For the 3-bit case with 16-PSK TCM the effect is more pronounced. An extra 1 dB of SNR_i is required to achieve the saturation level at $\mathrm{SNR}_o = 22$ dB.

In all cases the performance seems to correspond to the subset distance which the bits specify in the channel signal partion. It appears that the MSB from the quantizer should be mapped to the highest order bit into the encoder, that is x_m for the rate $m/(m+1)$ trellis codes. Likewise the LSB should correspond to x_1 in all cases. Since the codes are systematic, the encoder inputs $x_m, \ldots, x_1$ correspond directly to the embedded convolutional coder output $y_m, \ldots, y_1$ which specifies the signal map.

A similar phenomenon has also been reported by Omura [42]. For a rate 2/3 code it was determined that bits entering the top of the encoder had a smaller probability of error than bits entering the bottom of an encoder.

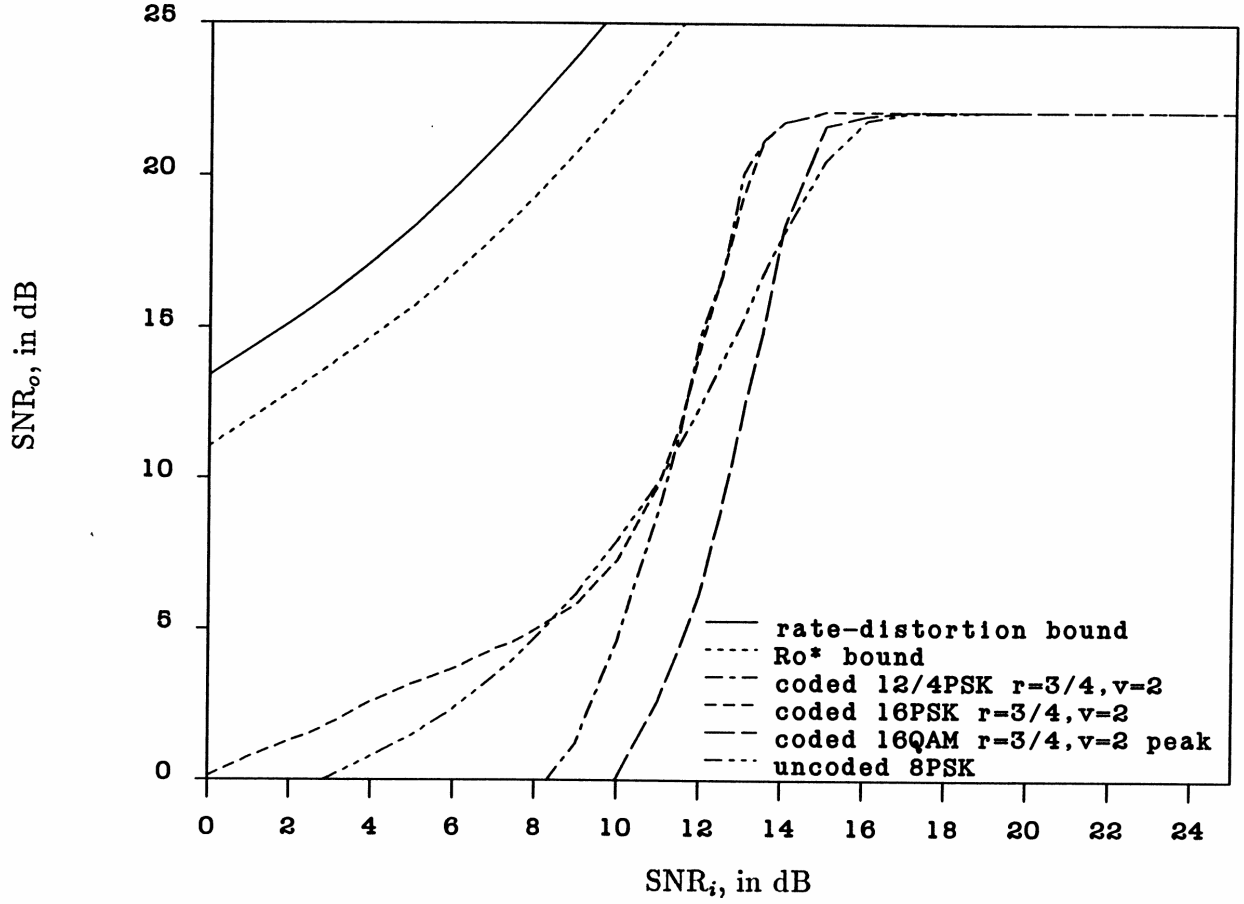

Figure 18: System performance: 3-bit DPCM and 12/4-PSK

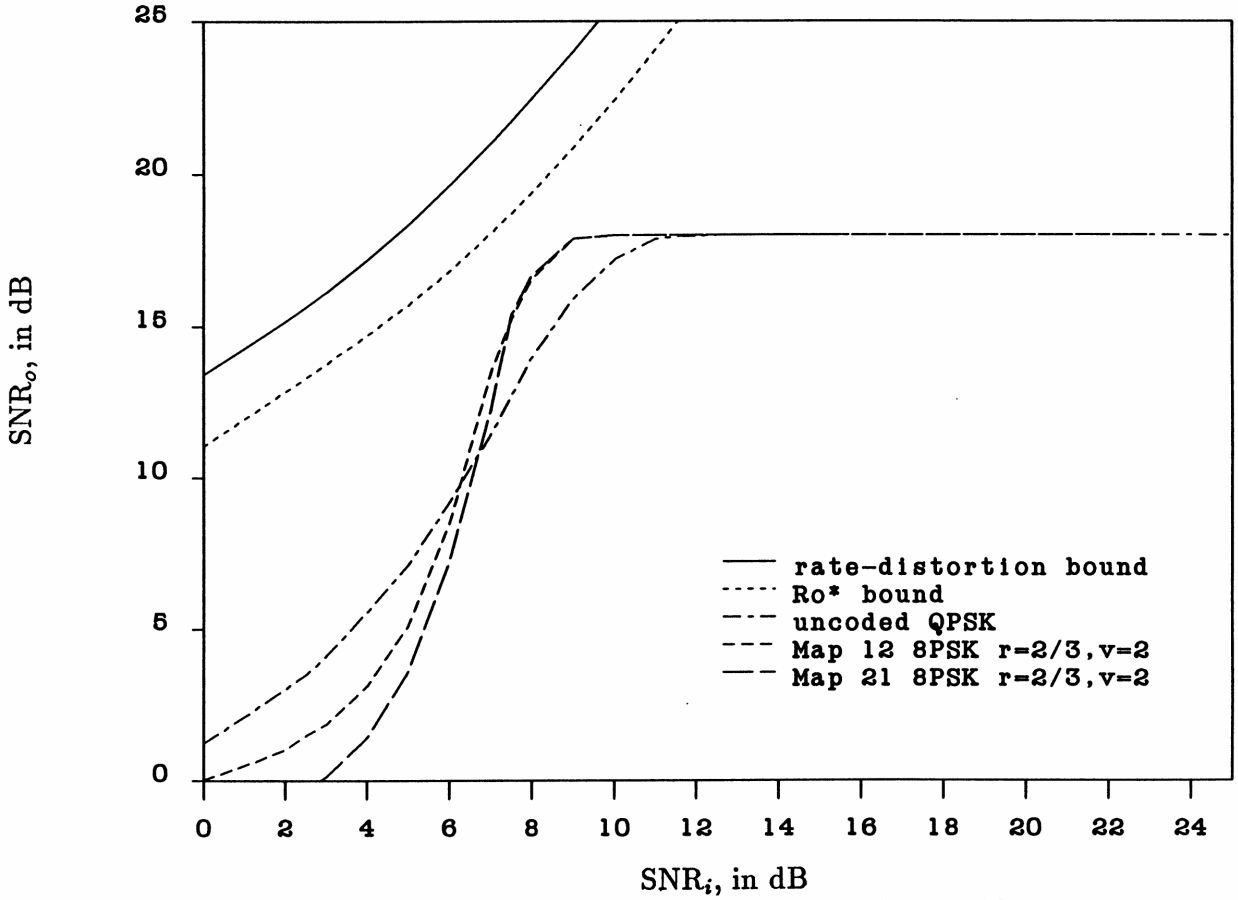

Figure 19: Effect of bit mapping, 2-bit DPCM

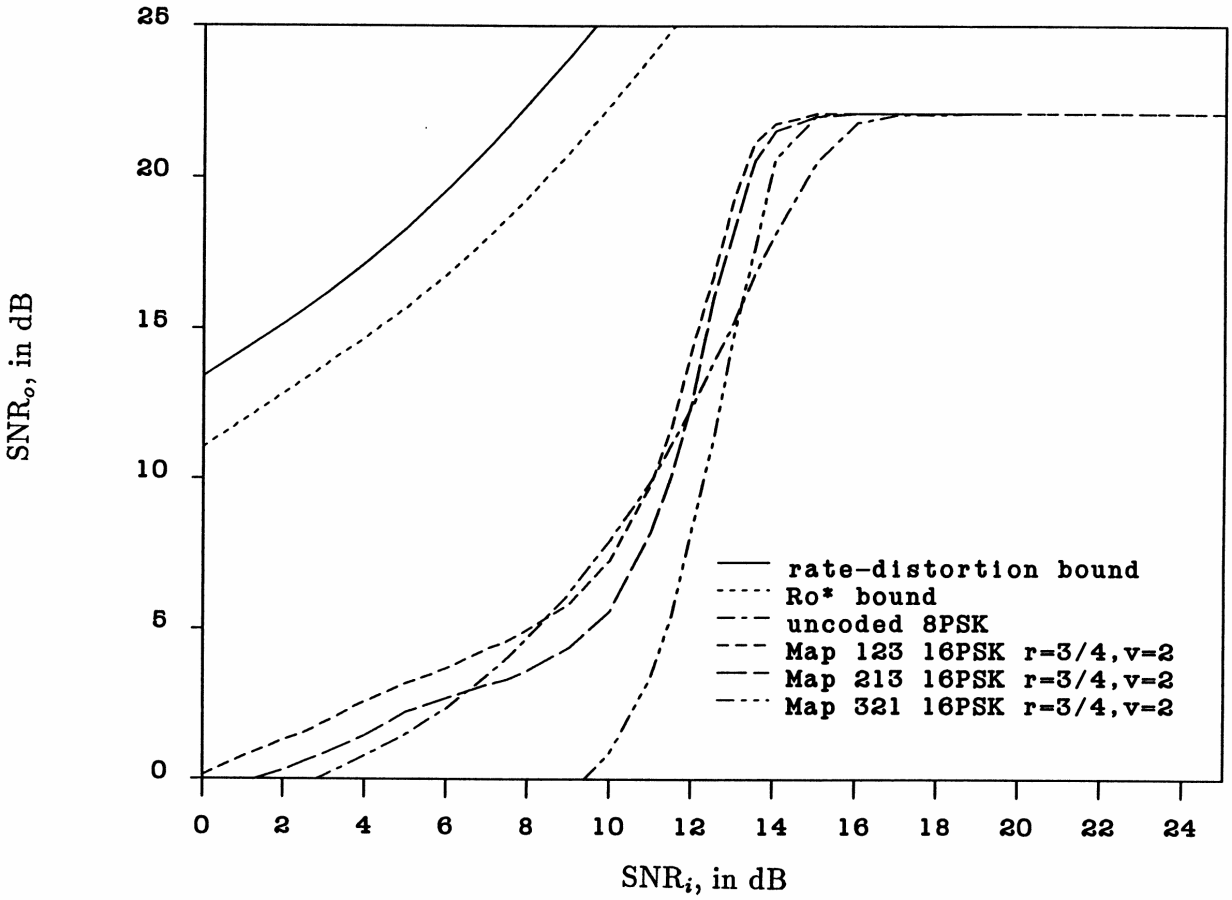

Figure 20: Effect of bit mapping, 3-bit DPCM

6.2.3 Comparison to R_0

The simulation section is concluded by some remarks on the comparison of the simulation results to the R_0 results. It should first be noted that direct comparison to the individual R_0 function plots is difficult because the trellis coded systems use modulation techniques with 2^{m+1} points while the uncoded systems use modulation with 2^m points. Nevertheless, certain observations can be made with the use of either Equation (3) or (4). Good performance is expected when $R_0 - R$ is sufficiently greater than 0 or when R_0/R is sufficiently greater than 1. Comparison of the R_0 curves and the simulation results seem to yield a general trend that the knee in the simulation results curves occur at the same value of SNR_i as does when $R_0 - R \approx 0.6$ for $\nu = 6$ codes and $R_0 - R \approx 0.9$ for the $\nu = 2$ trellis codes. For example, in Figure 14 the result for 2-bit DPCM using 8-PSK with $\nu = 6$ has its saturation knee at about 8 dB. The corresponding R_0 curve in Figure 4 shows that 8-PSK has an R_0 value of 2.6 bits per symbol at $\text{SNR}_i = 8$ dB. Since the information rate is 2 bits per symbol, $R_0 - R$ is 0.6. This supports the claim made that R_0 can be used to find where good short constraint length codes exist.

As shown in Section 2, one of the best uses of R_0 is to compare the relative performance of various modulation technique. The relative performance predicted by R_0 for 16-PSK, 16-QAM and 12/4-PSK was verified in the simulation results. The results were accurate for both the average power constraint and

peak power constraint channels.

Finally, R_0^* represents a practical goal for system performance. For constraint length $\nu = 6$ codes, performance was 1.0 to 3.5 dB from R_0^*. Absolute comparison to R_0^* is possible because it is independent of the the modulation technique used.

7 Summary and Conclusions

This thesis has investigated a combined source coding, channel coding and modulation approach to communication system design. The intention is to show that a system which is optimized as a whole is more robust than one designed by "pasting together" separately optimized pieces of the system. In this study the technique was applied to a specific example. Namely, the transmission of a first-order gauss-markov source over a AWGN channel using DPCM source encoding and two-dimensional modulation. The process and concepts were presented in general a manner so that the analysis could be applied to many systems.

The method used can be outlined as follows:

1) For a given source model and selected performance criterion, the bounds on system performance can be determined based on knowledge of the channel characteristics.

2) The channel symbol rate is set equal to source symbol rate. Thereby creating a fixed bandwidth system.

3) A tradeoff is done to determine the best grouping of source encoding, channel encoding and modulation technique for a given level of channel input signal-to-noise ratio.

In analyzing system performance, the source encoder, channel coder and modulator were studied within the context of the total system and improvements were possible by making tradeoffs between the three sections. The quantization of the

source encoder was modified between 2, 4 and 8 levels. It was determined that increased quantization can improve system performance by also increasing the channel signalling set but only when adequate channel input signal-to-noise ratio is available. At low SNR_i, increased quantization actually deteriorates performance.

The use of trellis coding greatly increased system performance without bandwidth expansion. The choice of code rate is determined by the source quantization. It was found that most of the improvement comes with short constraint codes while additional improvement is possible with longer constraint lengths. Also investigated was the source coder to channel coder bit map. The simulation indicated that a proper mapping exists which corresponds to the set partitioning method.

In the area of modulation, the R_0 function was used to show that for constellation of size 8 or less, the set of M-ary PSK schemes are the simplest and provide near optimal performance as compared to R_0^*. At 16 points the PSK method become inefficient and 16-QAM provides superior performance for the average power constraint channel. These results were verified by the simulation for the trellis coded versions of PSK and QAM. For the peak power constraint channel QAM losses its superiority but some of this may be regained by a modified version of 16-QAM called 12/4-PSK.

The use of R_0 led to insightful information as to the performance capabilities of each modulation technique. By using longer constraint length codes the R_0^*

bound was approached within 2.0 dB for 1-bit using 4-PSK, 1.0 dB with 2-bits using 8-PSK while the 16-QAM with 3-bit DPCM combination came within 2.2 dB of R_0^*. This supports the argument that R_0 is a good bound on achievable performance but it was found to be overly conservative on estimating coding complexity.

It is important to realize that the technique can be generalized to study other systems. The source encoder was investigated for quantization levels of DPCM but this could be expanded to include comparison of different source encoder techniques. The rate-distortion bound shows that performance of DPCM is sub-optimal.

The selection of the method of system performance measurement should be done carefully. In this study, a mean square error criterion was used because it is mathematically attractive and it provides a good measure of average performance. An average criterion is appropriate for systems with random noise fluctuations such as gaussian noise. The averaging technique has a threshold effect which makes noise variations below a certain level indistinguishable. For example 10^{-4} and 10^{-10} $\Pr(\varepsilon)$ might provide about the same mean square error and this may be a useful statement for a practical system which is not sensitive to infrequent errors. Alternatively, a systems may be sensitive to long but infrequent burst errors which can be obscured by averaging technique. Furthermore, in the real world system performance measures may be highly subjective and difficult to define by analytically means. This does not imply that

simple analytic methods are not useful, rather it seems intuitively correct that subjective performance should be monotonically increasing with decreasing mean square error, even if the relationship is non-linear or has insensitive plateau-like regions.

7.1 Future research

This study can be expanded in a number of areas. These can be placed into two basic categories. The first category involves studying more complex aspects of each subsystem. This can be used to extend the knowledge on optimizing the implementation of practical systems. The second category is in the development of analytical work to better describe and bound the performance of the combine source-channel coding and modulation system.

The first category would include performance studies of more complex models of realistics source models and source encoders for image and voice systems. More realistic channels models should be implemented to truly appreciate the robustness of various source encoding methods. In the coding and modulation the symbol rate and channel rate where set equal. This forced system performance to only three unique levels. By using recent results in Multiple Trellis Coded Modulation [40] more finely quantized system performance is possible by using codes with non-integer throughput.

The second category raises the need for a more complete theory. One area

which needs to be understood more completely is the source encoder to channel encoder mapping relationship. Related to this is the possibility of finding codes which are designed to minimize distortion for specific data types. In doing so it would be worthwhile to gain a more thorough understanding of Viterbi decoder error statistics as these determine the distortion.

Finally, there is a need to develop systems which represent a truly combined source coding, channel coding and modulation technique. In this thesis, an example which may be considered as "matched" source coding, channel coding and modulation was investigated. A promising area would be the development of source trellis codes for multilevel/ multiphase modulation.

A Appendix A: Simulation Approach and Design

As part of this study a software simulation package was developed for the analysis of the communications system under consideration. In the this section the methodology of simulation will be discussed. Also the scope of effort will be documented.

The simulation package objective is to implement the overall system of Figure 1. The system includes the Gauss-Markov source, the DPCM encoder and decoder, trellis coder, AWGN channel, the Viterbi decoder and finally the system performance statistics. The system is implemented in a custom software package included in Appendix C.

A.1 Simulation Methodology

In approaching the the simulation, two methods can be identified in ways to determine the effects of interference in noisy communication systems [20]. One method is to simulate the deterministic noise-free signal paths and apply a statistical understanding to the noisy-paths and calculate the effect. The second approach is to use a Monte-Carlo techniques where all random processes are simulated and the results are determined by running large numbers of trials to get statistical averages on the frequency of occurrence of error events.

Each method has advantages and disadvantages [20]. The first approach allows for a saving in computation but requires a complete knowledge of the statistical properties of the random processes involved. In most systems , especially ones with non-linearities, the statistical properties may not be possible to determine. Furthermore, accuracy of the results are highly dependent on the completeness of the knowledge of the random properties. The second approach is a brute force method which can be computationally exhaustive. Its major advantage is that it does not require the intimate details of the statistical properties. The major disadvantage is that events with low frequency of occurrence take considerable computational time to get results with high confidence factors.

The approach taken here is similar to that taken by [20]. This involves a hybrid approach in which the all signal paths are simulated in a deterministic manner except the channel noise and the random component in the source model. A Monte-Carlo approach is then used to subject the Viterbi decoder to the the variations in channel noise as well as variation in the source sequence pattern.

This approach is most appropriate due to the nonlinearity of the Viterbi decoder. The performance of the Viterbi decoder is not only affected by the channel noise but includes the sequence of data transmitted over the channel. All the remaining components in the system are deterministic linear systems who's operation is either well understood or defined.

A.1.1 Monte Carlo method

The Monte Carlo method is based on simulating the random processes in a system with the help of random number generators. By using number generators which accurately simulate the random process, useful statistical averages can be developed by performing a sufficient number of trials.

In discussing the Monte Carlo method it should first be mentioned that the simulation was implemented in two phases. The first phase involved the development of the trellis coder and the Viterbi coder. The system was verified by implementing known codes from Ungerboeck. In this phase performance was measured on a probability of error basis. To verify small probabilities of error (i.e. $P(\varepsilon) = 10^6$) extensive use of monte carlo simulation is necessary. The second phase was the complete simulation of the system in Figure 1.

In running large simulations it is important to estimate the number of trials which will be required for accurate results. The Monte-Carlo method is based on the *Law of Large Numbers* in probability theory [32]. Consider an experiment of repeated Bernoulli trials where the probability of success is p and the probability of failure is $1-p$. If the experiment is repeated n times with $\mathbf{S}_n$ successes then it is expected that $\mathbf{S}_n/n$ approaches p for n large. This is our intuitive understanding of probability. The law of large numbers states this mathematically as:

$$\Pr\left\{\left|\frac{\mathbf{S}_n}{n} - p\right| < \epsilon\right\} \rightarrow 1 \tag{35}$$

here ϵ is the sampling error. For finite n the right side of Equation (35) is less

the one. The actual value is referred to as the "confidence factor", .i.e., the probability that the experimental results are a good approximation (within ϵ) to the actual probability. The confidence factor can be well approximated by a form of the *DeMoivre-Laplace limit theorem* [32].

$$\Pr\left\{\left|\frac{\mathbf{S}_n - np}{\sqrt{np(1-p)}}\right| < \beta\right\} \rightarrow 2Q(\beta) - 1 \tag{36}$$

where $Q(\beta)$ is the normal distribution. For a given sampling error and confidence factor ,γ, a sufficient number of trials can be determined by noticing that $p(p-1) \leq 1/4$ and solving for n as

$$n \geq \frac{1}{4\beta} Q^{-1}\left(\frac{\gamma+1}{2}\right) \tag{37}$$

here $Q^{-1}(\cdot)$ represents the inverse Gaussian error-probability function. This leads to the rule of thumb that $10/p$ trials are sufficient [33], as it implies a 99% confidence that the sample probability is within 6% of the actual value. For example, to prove that a certain coded system produces a probability of bit error $\Pr(\varepsilon) = 10^{-4}$ the simulation should use a sequence of a least 100,000 bits.

A.1.2 Random number generation

The random number generator which is used is very important in the outcome of the simulation results. The ideal uniform random number generator should have a sample autocorrelation function which approximates the impulse function $R_{nn}(\tau) = (N_0/2)\delta(\tau)$. A class of important generator are the linear congruential

generators (LCG). The generator used here is the Wichmann-Hill generator [34] which uses 3 LCG's. This is an exceptionally effective algorithm shown to have excellent statistical properties [35]. It is generated by the following iterative set of equations

$$\begin{aligned} I_n &= (171 \cdot I_{n-1}) \bmod 30269 \\ J_n &= (172 \cdot J_{n-1}) \bmod 30307 \\ K_n &= (170 \cdot K_{n-1}) \bmod 30323 \\ u_n &= \left[\frac{I_n}{30269} + \frac{J_n}{30307} + \frac{K_n}{30323}\right] \bmod 1.0 \end{aligned} \tag{38}$$

The generator produces the uniform random number u_n which is uniform over the interval $[0,1)$ and has mean $\mu_u = 1/2$ and variance $\sigma_u = 1/12$. Gaussian random numbers are generated by use of the *central limit theorem* which is generalization of Equation (36). It states that the sum of i.i.d random variables approaches a gaussian distribution. A gaussian number,ψ, with $\mu_\psi = 0$ and $\sigma_\psi^2 = 1$ can be generated from the uniform numbers of Equation (38) as

$$\psi = \left[\sum_{a=1}^{12} u_{n+a}\right] - 6 \tag{39}$$

The generator has been shown to have a period of $7.0 \cdot 10^{12}$ in [35]. This period is very large in respect to the simulation which calls on the generator at most 36 times per bit and runs through between 10^5 and 10^7 bits per simulation.

A.2 System Components

The Gauss-Markov source, DPCM encoder and decoder are implemented in a straightforward manner as would be a digital filter per Figure 10. The input data sequence $\{W_i\}$ is developed by the gaussian random number generator with mean $\mu_w = 0$ and variance $\sigma_w^2 = 1$ for simplicity. The DPCM encoder differs from ordinary digital filters in that it contains an embedded quantizer tp produce the channel data in a digital format. The encoder therefore contains simulated analog-to-digital and digital-to-analog converters. Similarly, the DPCM decoder must first use an analog-to-digital converter before operating on the data.

The most significant section of the simulation package includes the trellis coder, channel and Viterbi decoder. The AWGN channel is implemented in a complex baseband representation. The baseband model is appropriate because any channel effects caused during modulation, transmission and demodulation can be included in the model. This includes transmission and receiver filters, intersymbol interference, carrier phase tracking problems, non-linear channel distortion and other effects. Although these effects are not implemented in this study for the sake of studying more basic concepts. Since the channel noise is gaussian, the in-phase and quadrature noise are independent and their effects may be implemented separately in simulation.

The trellis coder combines the channel coder and modulator. This is accomplished by a convolutional encoder which is cascaded to a signal space map. The

signal map determines the specific location of the channel letter in the signal space based on the convolutional encoder output and the modulation technique.

A.2.1 Viterbi decoder

The Viterbi decoder is a maximum likelihood decoding algorithm for convolutional codes. A detailed explanation is found in [22]. Briefly, the decoders function is to search through the familiar trellis diagram associated with the noiseless convolutional encoder and find the path that differs least from the received sequence. That difference is called a metric and is measured in square Euclidean distance.

The Viterbi decoder is the most complex part of the system. Its design is based on the implementation recommended by Heller and Jacobs [21]. The major difference is that the Viterbi decoder implemented for this study is based on Euclidean distance rather than Hamming distance. The decoder is a pipeline structure composed of the following functional units; the branch metric calculator, the add-compare-select (ACS) section, the path memory and the data output section.

The branch metric calculator acts as a bank of M "matched filters" at the receiver front end. A branch metric is calculated based on a the likelihood function, $\Pr(y|x)$ the probability that y was received given that x was transmitted. For the system under consideration with equally-likely symbols this is calculated

as:

$$m_l = b_{max} - |y - x_l|^2 \tag{40}$$

where b_{max} is the maximum branch metric value and $|y - x_l|^2$ is the squared Euclidean distance between the received signal and the l_{th} point in the signal constellation, x_l. This results in the branch metric, m_l, which is inversely proportional to the distance between the received signal y and x_l.

The received signal y is quantized before the branch metric calculation. For two quantizer levels the system is called a hard-decision decoder and the branch metric becomes the "optimal receiver". By using infinite quantizer level, an addition 2 to 3 dB in performance is made available. A system with more than 2 levels is called a soft decision decoder. For 8 or more levels the loss becomes negligible. The simulations performed in this study will use 100 level of quantization throughout.

The ACS section is responsible for updating the new state metric associated with each of the 2^{ν} states in the trellis. For each state the ACS determines which of the 2^b paths merging into the new state has the largest metric. The largest value is added to the branch metric and this becomes the new state metric for the current state.

The path memory maintains the history of decisions made to get to each state. In [21] it was determined that the path history should be at least 4 constraint lengths deep. This modification implies that the decoder is no longer optimal,

but the impact is minimal. In this simulation, a path memory of length 100 is used throughout.

Finally, the decoder is responsible for outputting data. The data is available after the 100 bit path memory delay. The data can then be reconstructed by the DPCM decoder and compared to the transmitted data for performance measurement. The performance measure is that of Equations (25) and (26). Specifically, for a simulation of length L symbols

$$\mathrm{SNR}_o = 20 \log \left[\frac{\sigma_W^2 L}{(1-\rho^2) \sum_{i=1}^{L} [S_i - \hat{S}_i]^2} \right] \tag{41}$$

with the values $L = 100000$, $\sigma_W^2 = 1$ and $\rho = 0.95$ chosen for this study.

B Appendix B: R_0 Bound

B.1 Derivation of R_0 results

The following section develops the R_0 results of chapter 2. First the R_0 function is derived as a result of the forward proof of the coding theory as was shown by Gallager [10]. The R_0 function is then determined for the AWGN channel per Wozencraft and Jacobs [11]. From this result the specific R_0 functions for M-ary PSK and QAM are presented.

The proof of the coding theory uses the random coding technique of Shannon. Instead of proving probability of error for specific codes, the random coding procedure bounds the average probability of error for all possible codewords which can be formed. The derivation is based on the assumption that the channel will be used N times to transmit a message m. The channel has an input alphabet of size M. The average probability of error can be written as

$$\bar{P}_{e,m} = \sum_{\mathbf{x}_m} \sum_{\mathbf{y}} q_N(\mathbf{x}_m) P_N(\mathbf{y}|\mathbf{x}_m) \Pr[\text{error}|m, \mathbf{x}_m, \mathbf{y}] \tag{42}$$

for message m, the probability of transmitting m and receiving it in error for all possible channel input sequences, $\mathbf{x}_m$, transitioning into channel output sequences $\mathbf{y}$. This is calculated for all possible codewords that can represent m, of which there are M^N combinations. The term $P_N(\mathbf{y}|\mathbf{x}_m)$ is the transition probability of sequence $\mathbf{x}_m$ being transformed into received sequence $\mathbf{y}$ due to the statistical noise properties of the channel. The codeword sequences are assigned

the arbitrary probability $q_N(\mathbf{x}_m)$ of choosing sequence $\mathbf{x}_m$ to represent m. While $\Pr[\text{error}|m, \mathbf{x}_m, \mathbf{y}]$ represents the probability of an error for the combination of m, the chosen sequence of $\mathbf{x}_m$ and the possible received sequence $\mathbf{y}$.

Aside: The union bound

For a given set of events A_m

$$P\left(\bigcup_m A_m\right) \leq \left[\sum_{m=1}^{M} P(A_m)\right]^{\rho}; \quad 0 < \rho \leq 1 \tag{43}$$

this is true since the events may "overlap". The use of ρ generalizes the result and is restricted to a range of interest.

Using the maximum likelihood decoding rule, the receiver should choose in favor of message m' if

$$\Pr(\mathbf{y}|\mathbf{x}_{m'}) = \max_j \Pr(\mathbf{y}|\mathbf{x}_{m_j}). \tag{44}$$

An error event occurs if $m' \neq m$. Therefore for the given values of $m, \mathbf{x}_m$ & $\mathbf{y}$ we call $A_{m'}$ the error event that $\mathbf{x}_{m'}$ is selected such that $P_N(\mathbf{y}|\mathbf{x}_{m'}) \geq P_N(\mathbf{y}|\mathbf{x}_m)$. Then

$$\Pr[\text{error}|m, \mathbf{x}_m, \mathbf{y}] \leq P\left(\bigcup_{m' \neq m} A_{m'}\right) \tag{45}$$

$$\leq \left\{\sum_{m' \neq m} P(A_{m'})\right\}^{\rho} \tag{46}$$

where $A_{m'}$ is the probability of all events which satisfy Equation (44). This can be stated simply as

$$Pr(A_{m'}) = \sum_{\mathbf{x}_{m'}: P_N(\mathbf{y}|\mathbf{x}_{m'}) \geq P_N(\mathbf{y}|\mathbf{x}_m)} q_N(\mathbf{x}_{m'}) \tag{47}$$

The last equation can provide some very useful information by bounding the result with more tangible information. For the set of error events due to maximum likelihood decoding it must be true that for $s > 0$

$$\left[\frac{P_N(\mathbf{y}|\mathbf{x}_{m'})}{P_N(\mathbf{y}|\mathbf{x}_m)}\right]^s \geq 1 \tag{48}$$

Then an upper bound can be written as

$$Pr(A_{m'}) \leq \sum_{\mathbf{x}_{m'}} q_N(\mathbf{x}_{m'}) \left[\frac{P_N(\mathbf{y}|\mathbf{x}_{m'})}{P_N(\mathbf{y}|\mathbf{x}_m)}\right]^s \tag{49}$$

Substitution into equation 46 and noting that there are $M-1$ choices of $m' \neq m$ yields

$$\Pr[\text{error}|m, \mathbf{x}_m, \mathbf{y}] \leq \left\{(M-1)\sum_{\mathbf{x}_{m'}} q_N(\mathbf{x}_{m'}) \left[\frac{P_N(\mathbf{y}|\mathbf{x}_{m'})}{P_N(\mathbf{y}|\mathbf{x}_m)}\right]^s\right\}^\rho ; \quad \begin{matrix} s > 0 \\ 0 < \rho \leq 1 \end{matrix} \tag{50}$$

Substitution into (42) and replacement of the dummy variable $\mathbf{x}_{m'}$ by $\mathbf{x}$ results in

$$\bar{P}_{e,m} \leq (M-1)^\rho \sum_{\mathbf{x}_m} \sum_{\mathbf{y}} \left\{q_N(\mathbf{x}_m) P_N(\mathbf{y}|\mathbf{x}_m) \left(\sum_{\mathbf{x}_m} q_N(\mathbf{x}) \left[\frac{P_N(\mathbf{y}|\mathbf{x})}{P_N(\mathbf{y}|\mathbf{x}_m)}\right]^s\right)^\rho\right\} \tag{51}$$

this can be rewritten

$$\bar{P}_{e,m} \leq (M-1)^\rho \sum_{\mathbf{y}} \left[\sum_{\mathbf{x}_m} q_N(\mathbf{x}_m) P_N(\mathbf{y}|\mathbf{x}_m)^{1-sp}\right] \left[\sum_{\mathbf{x}} q_N(\mathbf{x}) P_N(\mathbf{y}|\mathbf{x}_m)^{sp}\right] \tag{52}$$

Replacing the dummy variable $\mathbf{x}_m$ by $\mathbf{x}$ and setting $s = 1/(1+\rho)$ leads to the bound of interest

$$\bar{P}_{e,m} \leq (M-1)^\rho \sum \mathbf{y} \left[\sum_{\mathbf{x}} q_N(\mathbf{x}_m) P_N(\mathbf{y}|\mathbf{x}_m)^{1+p}\right]^{1+\rho} ; \quad 0 < \rho \leq 1 \tag{53}$$

In this study we are interested in a discrete memoryless channel (DMC) with AWGN noise. For a DMC each channel symbol is independent so that

$$P_N(\mathbf{y}|\mathbf{x}) = \prod_{n=1}^{N} P(y_n|x_n) \tag{54}$$

$$q_N(\mathbf{x}) = \prod_{n=1}^{N} q(x_n) \tag{55}$$

The DMC bound is then

$$\bar{P}_{e,m} \leq (M-1)^{\rho} \sum_{\mathbf{y}} \left[\sum_{\mathbf{x}} \prod_{n=1}^{N} q(x_n) \prod_{n=1}^{N} P(y_n|x_n)^{1+p} \right]^{1+\rho} ; \quad 0 < \rho \leq 1 \tag{56}$$

The product term can then be represented

$$\bar{P}_{e,m} \leq (M-1)^{\rho} \prod_{n=1}^{N} \sum_{y} \left[\sum_{x} q(x_n) P(y_n|x_n)^{1+p} \right]^{1+\rho} \tag{57}$$

For a channel with K-input letters and J-output letters this can be written

$$\bar{P}_{e,m} \leq (M-1)^{\rho} \left\{ \sum_{j=0}^{J-1} \left[\sum_{k=0}^{K-1} q(k) P(j|k)^{1/1+\rho} \right]^{1+\rho} \right\}^{N} \tag{58}$$

An exponential form can be realized by defining rate $R = \log_2 M/N$ and $M = 2^{RN}$. For N selected as $M - 1 < 2^{RN} \leq M$ the bound becomes

$$\bar{P}_{e,m} \leq 2^{\{-N[E_0(\rho,\mathbf{q})-\rho R]\}} \tag{59}$$

where

$$E_0(\rho, \mathbf{q}) = -\log_2 \sum_{j=0}^{J-1} \left[\sum_{k=0}^{K-1} q(k) P(j|k)^{1/1+\rho} \right]^{1+\rho} \tag{60}$$

The average probability over all messages is then bounded as

$$\bar{P}_e = \sum_{m=1}^{M} \Pr(m) \bar{P}_{e,m} \leq 2^{\{-N[E_0(\rho,\mathbf{q})-\rho R]\}} \tag{61}$$

Since ρ and $\mathbf{q}$ are arbitrary, the tightest bound occurs when

$$\bar{P}_e \leq 2^{-N \max_{\rho,\mathbf{q}}[E_0(\rho,\mathbf{q}) - \rho R]} \tag{62}$$

The R_0 bound is then obtained by setting $\rho = 1$

$$\begin{aligned} R_0 &= \max_{\mathbf{q}} E_0(1,\mathbf{q}) \\ &= \max_{\mathbf{q}} -\log_2 \sum_{j=0}^{J-1} \left[\sum_{k=0}^{K-1} q(k) \sqrt{P(j|k)} \right]^2 \end{aligned} \tag{63}$$

For the AWGN channel the transition probability is approximated as [11]

$$P(j|k) = Q\left(\frac{d_{jk}}{\sqrt{2N_0}}\right) \tag{64}$$

where $Q(\cdot)$ is the Gaussian error function and d_{jk} is the euclidean normal distance between channel letters i and j defined as $d_{jk} = ||(s_j - s_k)||$. Using this information the R_0 bound becomes

$$R_0 = \max_{\mathbf{q}} -\log_2 \sum_{j=0}^{J-1} \left[\sum_{k=0}^{K-1} q(k) \sqrt{Q\left(\frac{||s_j - s_k||}{\sqrt{2N_0}}\right)} \right]^2 \tag{65}$$

The Error function can be over-bound as $Q(\alpha) \leq e^{-\alpha^2/2}$. At this point it is assumed that the channel has M input and output letters. The squared term in (65) is expanded as the product of a summation in k and a summation in the dummy variable i. The exponential terms are grouped and the j terms drop-out. Then replacement of dummy variable k by j results in

$$R_0 = \max_{\mathbf{q}} -\log_2 \sum_{i=1}^{M} \sum_{j=1}^{M} \left[q(i)q(j) \exp\left(\frac{||s_i - s_j||^2}{4N_0}\right) \right] \tag{66}$$

Finally for equally likely symbol this reduces to

$$R_0 = -\log_2 \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left[\exp\left(\frac{||s_i - s_j||}{4N_0} \right) \right] \tag{67}$$

To determine R_0 for a modulation technique in the presence of AWGN it is necessary only to have knowledge of the coordinates of the letters in the signal space map and use this to determine the euclidean distance between letters.

For programming purposes it is useful to obtain a closed-form expression. The R_0 function for M-PSK is arrived at by realizing an expression for d_{ij}. The MPSK signal uses 1 dimension per symbol for $M = 2$ and 2 dimensions for $M \geq 3$. In general the N_s-dimension symbol must lie on a hypersphere of radius $\sqrt{N_s E_N}$. From the signal space geometry it is clear that

$$d_{ij} = ||s_i - s_j|| \tag{68}$$

$$= 2\sqrt{N_s E_N} \sin \frac{\pi}{M}(i-j); \text{ for } \begin{cases} i = 1, 2, \ldots, M \\ i = 1, 2, \ldots, M \end{cases} \tag{69}$$

Therefore R_0 for MPSK is

$$R_0 = -\log_2 \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \exp\left\{ \frac{N_s E_N}{N_0} \left[\sin \frac{\pi}{M}(i-j) \right]^2 \right\} \text{ bits/symbol} \tag{70}$$

The closed form R_0 expression relies finding an expression which translate the number used to identify two points into the distance between them, therefore the relationship may result in an algorithm with little mathematical insight. For QAM a closed form expression is determined which in initial form is useful, while the final form is mostly for software implementation.

The expression is arrived at by noticing that the QAM signals form a square lattice of M points equally spaced by $2a\sqrt{2E_N}$. The value $\sqrt{2E_N}$ represents the energy limit for the 2 dimensional symbol and a equal $\sqrt{1/10}$ for an average power constraint while $a = \sqrt{1/18}$ for the peak power constraint. Using a two component vector to enumerate the signal points as

$$\mathbf{i} = (i_1, i_2) \text{ for } \begin{cases} i_1 = 1, 2, \ldots, \sqrt{M} \\ i_2 = 1, 2, \ldots, \sqrt{M} \end{cases} \tag{71}$$

The R_0 function can be determined with the $\mathbf{j}$ representing the channel output coordinates as follows

$$R_0 = -\log_2 \frac{1}{M^2} \sum_{\mathbf{i}} \sum_{\mathbf{j}} \exp\left\{ \frac{[2aN_s E_N ||\mathbf{i} - \mathbf{j}||]^2}{4N_0} \right\} \tag{72}$$

$$= -\log_2 \frac{1}{M^2} \sum_{\mathbf{i}} \sum_{\mathbf{j}} \exp\left\{ a^2 N_s \frac{E_N}{N_0} ||\mathbf{i} - \mathbf{j}||^2 \right\} \tag{73}$$

$$= -\log_2 \frac{1}{M^2} \sum_{i_1=1}^{\sqrt{M}} \sum_{j_1=1}^{\sqrt{M}} \sum_{i_2=1}^{\sqrt{M}} \sum_{j_2=1}^{\sqrt{M}} \exp\left\{ 2a^2 \frac{E_N}{N_0} \left[(i_1 - j_1)^2 + (i_2 - j_2)^2 \right] \right\} \tag{74}$$

1 Appendix C: Software

1.1 R_0 and capacity determination

```
C  THIS PROGRAM GENERATES THE R0 FUNCTION FOR

C  PAM, PSK, QAM AND 12/4-PSK MODULATION IN AWGN

      PARAMETER IDIM=61

      COMMON PI, A, ASQ, ART,RSIN,RCOS

      DIMENSION R0(IDIM),D(64,64)

      DIMENSION RCOS(16),RSIN(16)

      INTEGER DB,L,H

      REAL DIST,N

      CHARACTER*4 NAME

      PI = 4.0 * ATAN(1.0)

      NAME = 'DUMY'

      WRITE(1,*) 'ENTER 0 FOR PAM, 1 FOR PSK, 2 FOR QAM, 3 for 12/4PSK'

      READ(1,*) IOPTION

      WRITE(1,*) 'ENTER CONSTELLATION SIZE,AMPLITUDE,OUTPUT FILE NAME'

      READ(1,*) A,N

      READ(1,*) NAME

      ASQ = A ** 2.0

      ART = SQRT(A)

C##############################################

C    PRE-CALCULATE SIGNAL DISTANCES        #
```

```
C##############################################
      DUMMY=1.0
      CALL DATA(DUMMY)
      DO 50 L=1,A,1
      DO 50 H=1,A,1
50     D(H,L)= DIST(IOPTION,L,H)
C##############################################
C   CALCULATE R0 FOR SIG-TO-NOISE RATIOS   #
C##############################################
    DO 10  DB = -10,50,1
      ETON= 10.0 ** (FLOAT(DB)/10.0)
      I= DB+11
      VALUE = 0
        DO 20  L=1,A,1
        DO 20  H=1,A,1
20       VALUE = VALUE + EXP( D(H,L)* ETON*N)
10     R0(I) = (-1)*LOG( VALUE/ASQ)/LOG(2.0)
    OPEN (UNIT=10,FILE= NAME )
    WRITE (10,*) R0
    CLOSE (UNIT=10)
    END
C##############################################
C   END OF MAIN                #
```

```
C##############################################

      FUNCTION DIST(IOPTION,L,H)

C##############################################

C     SUBROUTINE TO CALC SQUARE DIST        #

C##############################################

      COMMON PI, A, ASQ, ART,RSIN,RCOS

      DIMENSION RSIN(16),RCOS(16)

      COMPLEX CL, CH

      INTEGER L,H

      IF (IOPTION .EQ. 0 ) THEN

           DIST = (-1) * ( (L-H)/(A-1))**2

      ELSEIF (IOPTION .EQ. 1) THEN

           DIST = (-1) * ( SIN(PI*(L-H)/A) )**2

      ELSEIF (IOPTION .EQ. 2) THEN

           CL = CMPLX( ANINT(L/ART+.49), AMOD( L-1.0,ART)+1)

           CH = CMPLX( ANINT(H/ART+.49), AMOD( H-1.0,ART)+1)

           DIST = (-1)* ( CABS(CL-CH)/(ART-1))**2

      ELSE

       DIST =-1.0*((RSIN(L)-RSIN(H))**2+(RCOS(L)-RCOS(H))**2)

      ENDIF

      RETURN

      END

C ######################################
```

```
C THIS SUBROUTINE DEFINES THE POSITION #
C OF THE 12/4-PSK SIGNAL POINTS      #
C ######################################
      SUBROUTINE DATA(DUMMY)
      COMMON PI, A, ASQ, ART,RSIN,RCOS
      DIMENSION RSIN(16),RCOS(16)
        ALPHA= SIN(PI/12.0)
        BETA= COS(PI/12.0)
        B=SQRT(1.0/18.0)
         RCOS(16)=-B
         RSIN(16)=-B
         RCOS(1)=B
         RSIN(1)=-B
         RCOS(2)=B
         RSIN(2)=B
         RCOS(3)=-B
         RSIN(3)=B
         RCOS(4)=BETA
         RSIN(4)=-ALPHA
         RCOS(5)=ALPHA
         RSIN(5)=BETA
         RCOS(6)=-BETA
         RSIN(6)=ALPHA
```

```
      RCOS(7)=-ALPHA
      RSIN(7)=-BETA
      RCOS(8)=3*B
      RSIN(8)=3*B
      RCOS(9)=-3*B
      RSIN(9)=3*B
      RCOS(10)=-3*B
      RSIN(10)=-3*B
      RCOS(11)=3*B
      RSIN(11)=-3*B
      RCOS(12)=-ALPHA
      RSIN(12)=BETA
      RCOS(13)=-BETA
      RSIN(13)=-ALPHA
      RCOS(14)=ALPHA
      RSIN(14)=-BETA
      RCOS(15)=BETA
      RSIN(15)=ALPHA
      RETURN
      END

C  THIS PROGRAM GENERATES THE CAPACITY AND R0* FUNCTIONS
      DIMENSION R0STAR(61), C(61)
```

```
      INTEGER DB
      DO 10  DB = -10,50,1
           ETON= 10.0 ** (FLOAT(DB)/10.0)
           I= DB+11
           VALUE = LOG( EXP(1.0))/LOG(2.0)/2.0
           VALUE = VALUE*(1+ETON- SQRT(1.0+ETON**2))
           R0STAR(I) = VALUE+ LOG(.5*(1+ SQRT(1+ETON**2)))/ LOG(2.)/2.
10         C(I) = .5 * LOG( 1 + 2*ETON)/LOG(2.0)
      OPEN (UNIT=10,FILE='STAR')
      OPEN (UNIT=11,FILE='CAP')
      WRITE (10,*) R0STAR
      WRITE (11,*) C
      CLOSE (UNIT=10)
      CLOSE (UNIT=11)
      END
```

1.2 Rate-distortion bound calculation

This section contains the formatted input to use the engineering spreadsheet program MathCAD. The first MathCAD file calculates the SNR_i which causes uncoded modulation to yeild $Pr(\varepsilon) \geq 10^{-5}$. The second file is used to calculate the rate-distortion bound of a Gauss-Markov.

THIS MATHCAD FILE CALCULATES AN APPROXIMATION FOR

FOR FINDING THE SNR REQUIRED TO YEILD

P(ϵ)=10^-5 FOR VARIOUS MODULATION TECHNIQUES

$$TOL := 10^{-10}$$

$$l := 1 \,..\, 8$$

$$Q(\beta) := \begin{bmatrix} 1 \\ - \\ 2 \end{bmatrix} \cdot \left[1 - \operatorname{erf}\left[\frac{\beta}{\sqrt{2}} \right] \right] \qquad \delta := 1$$

$$ans_l := root\left[Nfree_l \cdot Q(\delta) - 10^{-5}, \delta \right]$$

$$S_l := 10 \cdot \log\left[\left[\frac{ans_l}{dfree_l} \cdot 2 \right]^2 \cdot \frac{1}{2} \right]$$

Nfree :=

l
1
2
2.25
2
3
2
3
2

dfree :=

l
2
$\sqrt{2}$
$\sqrt{0.8}$
$2 \cdot \sin\left[\frac{\pi}{8} \right]$
$\sqrt{0.4}$
$2 \cdot \sin\left[\frac{\pi}{12} \right]$
$\sqrt{\frac{2}{9}}$
$2 \cdot \sin\left[\frac{\pi}{16} \right]$

SNR for Pr(e)=10^5 in dB

S_l	
9.58816909	BPSK
12.90324282	4PSK
16.93245539	8AMPM
18.23614965	8PSK
20.0621662	16QAM
21.63301825	12/4PSK
22.61489125	16QAMpeak
24.08822835	16PSK

THIS MATHCAD FILE CALCULATES THE RATE DISTORTION BOUND FOR A FIRST ORDER GAUSS-MARKOV PROCESS DESCRIBED BY CORRELATION COEFFICIENT ρ AND DISTORTION MEASURED AS THE MEAN-SQUARED ERROR

A RELATIONSHIP BETWEE INPUT SNR VS. OUTPUT SNR IS DETERMINED IN UNITS OF DECIBEL

The following section assigns values and does some pre-calulation to minimize computation time. The value ρ is the autoregressive coefficient of the Gauss-Markov source. The value of x were choosen by trial and error to obtain sufficient range and detail of the R(d) and D curves. Note that in MathCAD ':=' means assign while '=' displays the current value of a variable

Assignments:

$\rho := 0.95$

$\theta := 0 \ldots 11$

$x_\theta :=$

2
1.5
1.2
1
.5
.25
.1
.05
.025
.01
.005
.001

Pre-calculations:

$log2 := \log(2)$

$\rho 1 := 1 - \rho^2$

$\rho 2 := 1 + \rho^2$

$\sigma := \sqrt{\frac{1}{\rho 1}}$

$\rho 1\sigma := \rho 1 \cdot \sigma^2$

$c := \frac{1}{2 \cdot log2}$

CALCULATION OF RATE-DISTORTION FUNCTION

MathCAD calculations such as integrals and roots of functions are calculated with increasing resolution until a set tolerance is met. The tolerance choosen below was determined to provide sufficient resolution while keeping calculation time reasonable.

$$TOL := 0.0001$$

The following parameteric equations define the rate-distorion function for a Gauss-Markov Source. The function d[θ] represents the distorion measure while r[θ] determines the minimum rate at which d[θ] is possible.

$$s(\lambda) := \frac{\rho 1 \sigma}{\rho 2 - 2 \cdot \rho \cdot \cos(\lambda)} \qquad g(\lambda) := c \cdot \log\left[\frac{s(\lambda)}{x_\theta}\right]$$

$$d_\theta := \left[\frac{1}{2 \cdot \pi}\right] \cdot \int_{-\pi}^{\pi} \text{if}\left[s(\lambda) < x_\theta, s(\lambda), x_\theta\right] d\lambda$$

$$r_\theta := \frac{1}{2 \cdot \pi} \cdot \int_{-\pi}^{\pi} \text{if}(g(\lambda) > 0, g(\lambda), 0)\ d\lambda$$

$$WRITE(r) := r_\theta \qquad WRITE(d) := d_\theta$$

Input SNR versus outpur SNR Calculation

The $d[\theta]$ is related to SNRout per equation

While SNRin is calculated in terms of $r[\theta]$

$$SNRout_{\theta} := 10 \cdot \log\left[\frac{\sigma^2}{d_{\theta}}\right] \qquad Ns := 2$$

Ns is dimensions per symbol

This equation is a form of the capacity expression for the AWGN channel

$$SNRin_{\theta} := 10 \cdot \log\left[Ns \cdot \left[2^{r_{\theta} \cdot \frac{2}{Ns}} - 1\right]\right]$$

The Ro* equivalent is best calculated by use of a root determination function

SNR := 0 initial guess

N := 2 dimensions per symbol

Nl := N·log(e)

Ro* function

$$R01(SNR) := \frac{Nl}{2 \cdot log2} \cdot \left[1 + SNR - \sqrt{1 + SNR^2}\right]$$

$$R0star(SNR) := R01(SNR) + \frac{N}{2 \cdot log2} \cdot \log\left[\frac{1}{2} \cdot \left[1 + \sqrt{1 + SNR^2}\right]\right]$$

$$val_{\theta} := root\left[R0star(SNR) - r_{\theta}, SNR\right]$$

$$SNRin2_{\theta} := 10 \cdot \log\left[val_{\theta} \cdot 4\right] \qquad WRITE(capacity) := SNRin_{\theta}$$

In this section a cubic spline method is used to determine SNRin values at integer values from the results for graphical purposes.

$i := 0 \ldots 20$

$S1 := cspline(SNRin, SNRout)$

$S2 := cspline(SNRin2, SNRout)$

$RATE_i := interp[S1, SNRin, SNRout, snr_i]$

$ROstar_i := interp[S2, SNRin2, SNRout, snr_i]$

write interpolated values

to disk

$WRITE(CAPsnr) := RATE_i$

$WRITE(ROsnr) := ROstar_i$

$snr_i :=$

i	
0	
1	
2	
2.5	
3	
4	
5	
6	
7	
7.5	
8	
9	
10	
11	
11.5	
12	
12.5	
13	
13.5	
14	
15	

CALCULATED THE SNR UPPER-BOUND FOR 1,2 & 3 BITS USING

THE GIVEN ρ

$\rho = 0.95$ $\quad k := 1..5$

$s3 := cspline(r, SNRout)$

$bits_k :=$

k
1
1.904
2
2.761
3

$SNRoutMAX_k := interp[s3, r, SNRout, bits_k]$

$bits_k$	$SNRoutMAX_k$	
1	16.131	dB, 1 bit optimal & uniform quant.
1.904	21.573	dB, 2 bit uniform quantizer
2	22.151	dB, 2 bit optimal
2.761	26.733	dB, 3 bit uniform quantizer
3	28.172	dB, 3 bit optimal

1.3 System simulation

This section contains the system simulation program for the rate 3/4 trellis codes. The rate 1/2 and 2/3 programs are of a highly similar nature, but are kept as separate programs to the decrease computation time. The structure of the Viterbi decoder is based on [44]. Sample input and output files follow the program.

```
C    FB3/4.F **** FEEDBACK ENCODER VERSION ***

C    COMMUNICATION SYSTEM SIMULATION PROGRAM

C  SOURCE:         GAUSS-MARKOV, ONE-DIMENSION

C  SOURCE ENCODER:  DPCM 3-BITS, MATCHED TO SOURCE MODEL

C  CHANNEL CODER:   RATE 3/4 TRELLIS CODES (CCV EXTERNAL)

C  MODULATION:      16-PSK, 16-QAM, 12/4PSK OR GRAY 8-PSK (UNCODED)

C  CHANNEL:         AWGN USING LCG WITH PERIOD 7E12

C  CHANNEL DECODER: VITERBI ALGORITHM BASED ON EUCLIDEAN DISTANCE

C  SOURCE DECODER:  APPROPRIATE DPCM EXPANDER


C  PROGRAM INPUTS: SOURCE CORRELATION COEFFICIENT; CODER CCV, MEMORY

C    LENGTH & QUANTIZATION LEVEL; LCG SEEDS

C  PROGRAM OUTPUTS: VITERBI DECODER STATISTICS, SOURCE SEQUENCE AND

C    RECONSTRUCTED OUTPUT DATA SEQUENCE.

     PARAMETER(M=2**12)

     INTEGER LINKLIST(0:M-1,8), IDIST(0:M-1),IOUT(4),STATEBUFFERTP(15)

     INTEGER METRIC(0:M-1), IPATH(0:M-1,150),TERM(0:15)

     DIMENSION IDNEW(0:M-1),RCOS(0:15),RSIN(0:15)

     DIMENSION PATHMEM(150),IPATHMEM(3,150),INPUT(0:M-1,3)

     REAL X1,X2
```

```
      INTEGER H(0:3,12),V,GRAY(0:15)
      COMMON STATEBUFFER(15),KBITS,DELTA(5),P
C     GRAY MAP (FOR 8-PSK) AND MAX QUANTIZER STEPS
      DATA GRAY/0,1,3,2,6,7,5,4,12,13,15,14,10,11,9,8/
      DELTA(1)=1.596
      DELTA(2)=0.996
      DELTA(3)=0.586
      DELTA(4)=0.335
      DELTA(5)=.1881
      INITS=0
      INITD=0
      INITE=0
      INITC=0
      OPEN(UNIT=10,FILE='F34STATS')
      OPEN(UNIT=11,FILE='[WEITZNER.HOME.SIM]F34SET')
      WRITE(10,*)'ENTER 1 TO SUPRESS DATA I/O'
      READ(11,*)NOIO
      IF (NOIO .NE. 1)THEN
        OPEN(UNIT=12,FILE='SOURCE.DAT')
        OPEN(UNIT=14,FILE='SINK.DAT')
      ENDIF
      ISYNC=26
      POWER=1.
```

```
WRITE(10,*)'INPUT V, THE CONTRAINT LENGTH'

READ(11,*)V

KBITS=V+3

WRITE(10,*)'ENTER 0 FOR 16-PSK, 1 FOR 16-QAM'

 READ(11,*)MAP

WRITE(10,*)'ENTER THE SCALING FACTOR FOR THE QUANTIZER'

READ(11,*) FACTOR

WRITE(10,*)'ENTER THE LENGTH OF THE SURVIVOR TO BE PATHMEMD'

READ(11,*)  IMEM

WRITE(10,*)'ENTER THE SEEDS FOR THE RANDOM NUMBER GENERATOR'

READ(11,*)  ISEED1,ISEED2,ISEED3

WRITE(10,*)'ENTER BITS PER ROUND ?'

READ(11,*) NTOTAL

WRITE(10,*)'ENTER REPORT FREQUENCY IN BITS'

READ(11,*) IFREQ

WRITE(10,*)'ENTER DPCM TO CHANNEL BIT MAP'

READ(11,*) MAP1,MAP2,MAP3

WRITE(10,*)'ENTER BITS PER SOURCE SYMBOL'

READ(11,*) NUMBIT

WRITE(10,*)'ENTER SOURCE CORRELATION COEFF'

READ(11,*) P

WRITE(10,*)'ENTER 0 FOR ES, 1 FOR EB'

READ(11,*) ITEST
```

```
      PI=ACOS(-1.0)
     IF (ITEST .EQ. 0) THEN
       KGAIN=1
     ELSE
       KGAIN=NUMBIT
     ENDIF
      ISHFT=2**(3-NUMBIT)
IFREQ=(IFREQ/3)*3
C    DPCM TO TRELLIS ENCODER MAPPING
      MAP1=2**(MAP1-1)
      MAP2=2**(MAP2-1)
      MAP3=2**(MAP3-1)
C    H(I,J) IS THE PARITY-CHECK MATRIX COEFFICIENTS WHICH DETERMINE
C    THE CONNECTIONS IN THE FEEDBACK ENCODER. H0 IS THE COEFF'S
C    FROM THE H0(D) POLYNOMIAL - ENTER LOWSET ORDER FIRST
C    RCOS AND RSIN ARE EVALUATED AND PATHMEMD SO THAT THEIR VALUES MAY
C    BE RETREIVED LATER WITHOUT COMPUTATION WHEN NEEDED
C     NOTE: TO SIMULATE UNCODED 8-PSK, H3 IS SET TO 0 AND
C     H0, H1 AND H2 ARE SET APPROPRIATELY
     WRITE(10,*)'INPUT THE PARITY CHECK COEFFS FOR H0'
     READ(11,*)  (H(0,I),I=0,V)
     WRITE(10,*)'INPUT THE PARITY CHECK COEFFS FOR H1'
     READ(11,*)  (H(1,I),I=0,V)
```

```
      WRITE(10,*)'INPUT THE PARITY CHECK COEFFS FOR H2'
      READ(11,*)  (H(2,I),I=0,V)
      WRITE(10,*)'INPUT THE PARITY CHECK COEFFS FOR H3'
      READ(11,*)  (H(3,I),I=0,V)
C     SECTION TO DEFINE SIGNAL SPACE MAP; MAP=0 FOR 16-PSK WITH
C     NATURAL MAP, -1 FOR 8-PSK WITH GRAY MAP, 1 FOR 16-QAM
C     WITH AVERAGE POWER CONSTRAINT, 2 FOR 16-QAM WITH PEAK POWER
C     CONSTRAINT, AND 3 FOR 12/4-PSK WITH PEAK CONSTRAINT
      IF (MAP .EQ. 0) THEN
       DO 15 I11=0,15
          RCOS(I11)=COS((I11)*PI/8.+PI/16.)
          RSIN(I11)=SIN((I11)*PI/8.+PI/16.)
 15     CONTINUE
      ELSEIF (MAP .EQ. -1) THEN
DO 16 I11=0,15
   RCOS(GRAY(I11))=COS((I11)*PI/8.+PI/16.)
   RSIN(GRAY(I11))=SIN((I11)*PI/8.+PI/16.)
16     CONTINUE
      ELSEIF (MAP .LT. 3) THEN
  A=SQRT(1.0/10.0)
IF (MAP .EQ. 2) A=SQRT(1.0/18.0)
          RCOS(0)=-A
          RSIN(0)=-A
```

```
RCOS(1)=A

RSIN(1)=-A

RCOS(2)=A

RSIN(2)=A

RCOS(3)=-A

RSIN(3)=A

RCOS(4)=3*A

RSIN(4)=-A

RCOS(5)=A

RSIN(5)=3*A

RCOS(6)=-3*A

RSIN(6)=A

RCOS(7)=-A

RSIN(7)=-3*A

RCOS(8)=3*A

RSIN(8)=3*A

RCOS(9)=-3*A

RSIN(9)=3*A

RCOS(10)=-3*A

RSIN(10)=-3*A

RCOS(11)=3*A

RSIN(11)=-3*A

RCOS(12)=-A
```

```
      RSIN(12)=3*A
      RCOS(13)=-3*A
      RSIN(13)=-A
      RCOS(14)=A
      RSIN(14)=-3*A
      RCOS(15)=3*A
      RSIN(15)=A
    ELSE
     ALPHA= SIN(PI/12.0)
BETA= COS(PI/12.0)
      A=SQRT(1.0/18.0)
      RCOS(0)=-A
      RSIN(0)=-A
      RCOS(1)=A
      RSIN(1)=-A
      RCOS(2)=A
      RSIN(2)=A
      RCOS(3)=-A
      RSIN(3)=A
      RCOS(4)=BETA
      RSIN(4)=-ALPHA
      RCOS(5)=ALPHA
      RSIN(5)=BETA
```

```
      RCOS(6)=-BETA
      RSIN(6)=ALPHA
      RCOS(7)=-ALPHA
      RSIN(7)=-BETA
      RCOS(8)=3*A
      RSIN(8)=3*A
      RCOS(9)=-3*A
      RSIN(9)=3*A
      RCOS(10)=-3*A
      RSIN(10)=-3*A
      RCOS(11)=3*A
      RSIN(11)=-3*A
      RCOS(12)=-ALPHA
      RSIN(12)=BETA
      RCOS(13)=-BETA
      RSIN(13)=-ALPHA
      RCOS(14)=ALPHA
      RSIN(14)=-BETA
      RCOS(15)=BETA
      RSIN(15)=ALPHA
   ENDIF
C    TABLES FOR DECODING ARE GOING TO BE SET UP. EACH STATE IS
C    REPRESENTED BY A NUMBER CURSTATE. TOTSTATE IS THE TOTAL NUMBER OF STATES ,
```

```
C    THIS IS AN EXTENDED STATE SO THAT KNOWLEDGE OF THE STATE DIRECTLY
C    YEILDS THE ENCODER OUTPUT..I.E. STATEBUFFER(V+2) AND STATEBUFFER(V+1) CORRESPOND
C    TO THE SYSTEMATIC INPUT BITS
     TOTSTATE=2**(V+3)
     DO 1 CURSTATE=0,TOTSTATE-1
C
     CALL DECTOBIN(CURSTATE)
    METRIC(CURSTATE)=STATEBUFFER(V+3)*8+STATEBUFFER(V+2)*4+STATEBUFFER(V+1)*2+STATEBUFFER(1)
C
C    THE INPUT TO THE ENCODER CORRESPONDING TO EACH STATE IS
C    COMPUTED. THEN THE PREVIOUS STATE OF THE ENCODER WHICH IS LINKED TO
C    THE PRESENT STATE BY THE BRANCH(I3) IS FOUND
33   INPUT(CURSTATE,1)=STATEBUFFER(V+3)
     INPUT(CURSTATE,2)=STATEBUFFER(V+2)
     INPUT(CURSTATE,3)=STATEBUFFER(V+1)
     DO 708 I=1,V+3
708     STATEBUFFERTP(I)=0
     DO 4 IA=0,1
     DO 4 IB=0,1
     DO 4 IC=0,1
I3=IA*4+IB*2+IC+1
      DO 705 K=1,V-1
      ISUMM=STATEBUFFER(K)+H(1,K)*IC+H(2,K)*IB
    &       +H(3,K)*IC + H(0,K)*STATEBUFFER(V)
```

```
 705   STATEBUFFERTP(K+1)=MOD(ISUMM,2)
       STATEBUFFERTP(1)=STATEBUFFER(V)
       IVAL=0
       DO 706 K=1,V
 706     IVAL=IVAL+STATEBUFFERTP(K)*2**(K-1)
       LINKLIST(CURSTATE,I3)=IVAL+IC*2**V+IB*2**(V+1)
      &              +IA*2**(V+2)
 4   CONTINUE
 1   CONTINUE
 71  WRITE(6,*)'INPUT THE ENERGY PER BIT TO NOISE RATIO IN DB'
     READ(5,*) DBNO
IF(DBNO .GT. 210) GO TO 2020
     WRITE(10,*)'EB/NO=',DBNO
     RTEMP=SQRT((0.125/KGAIN)*10**(-DBNO/10.))
     RNOISE=RTEMP
C    STATE METRIC(IDIST), THE PAST NOISE SAMPLES (PNSE) THE CONTENT
C    OF THE REGISTERS OF THE ENCODER(STATEBUFFER), THE PRESENT AND PAST
C    CHANNEL SYMBOL OUTPUT (HEX) ARE INITIALIZED TO BE ZERO
     DO 79 I=0,TOTSTATE-1
 79  IDIST(I)=0.
     IE=0
     INBITS=0
     DO 21 I=1,V+3
```

```
      STATEBUFFER(I)=0
21    CONTINUE
      HEX1=0
      HEX2=0
      INDEX=1
C     THE FOLLOWING REPRESNTS AN ENCODER . THE PROGRAM READS AN OCTAL
C     CHARACTER FROM THE STDIN (IE UNIT 5) ,CONVERTS IT TO BINARY, AND
C     AND INPUTS THE THREE BITS INTO THE ENCODER. THE PREVIOUS
C     CHANNEL SYMBOLS (HEX1 AND HEX2) ARE ADVANCED AND THE ENCODER
C     PUTS OUT A NEW HEX1.
C     THE VALUES MAP1 TO MAP3 ARE USED TO EXTRACT INDIVIDUAL BITS
C     FROM KDATA. MAPX SHOULD EQUAL 1,2 OR 4 TO GET AT THE LSB,
C     MIDDLE BIT AND MSB,REPECTIVELY. NATURAL MAPPING IS MAP[1,2,3]
C     EQUAL [1,2,4]
18    CONTINUE
      DO 707 K=V-1,1,-1
      ISUMM=STATEBUFFER(K+1)+H(1,K)*STATEBUFFER(V+1)+H(2,K)*STATEBUFFER(V+2)
     &      +H(3,K)*STATEBUFFER(V+3) + H(0,K)*STATEBUFFER(1)
707   STATEBUFFERTP(K)=MOD(ISUMM,2)
      STATEBUFFER(V)=STATEBUFFER(1)
      DO 709 K=V-1,1,-1
709     STATEBUFFER(K)=STATEBUFFERTP(K)
C     ************ GET THE PCM CHARACTER ****
```

```
      CALL SOURCE(SOUT,INITS)
      IF (NOIO .NE. 1)WRITE(12,*)SOUT
      CALL DPCM(SOUT,KDATA,INITD,NUMBIT)
C     ******* SHIFT KDATA LEFT **************
C     ******* NUMBIT=1 : 2 BITS **************
C     ******* NUMBIT=2 : 1 BIT **************
      KDATA=KDATA*ISHFT
      STATEBUFFER(V+1)=MOD(KDATA/MAP1,2)
      STATEBUFFER(V+2)=MOD(KDATA/MAP2,2)
      STATEBUFFER(V+3)=MOD(KDATA/MAP3,2)
      HEX3=HEX2
      HEX2=HEX1
      HEX1=STATEBUFFER(V+3)*8+STATEBUFFER(V+2)*4+STATEBUFFER(V+1)*2+STATEBUFFER(1)
C     ICR AND ICI FORM THE REAL AND IMAGINARY PART OF THE SUFFICIENT
C     STATISTICS. TERM CORRESPONDS TO THE FIRST TERM OF THE
C     EXPRESSION FOR THE METRIC
53    X1= GRAN(0.0,POWER,ISEED1,ISEED2,ISEED3)
      X2= GRAN(0.0,POWER,ISEED1,ISEED2,ISEED3)
      ICREAL=ROUND((RCOS(HEX2)/2+RNOISE*X1)*FACTOR**2)
      ICIMAG=-ROUND((RSIN(HEX2)/2+RNOISE*X2)*FACTOR**2)
      DO 7 I5=0,15
C     IF (MAP .EQ. 0) THEN
C     TERM(I5)=ROUND((RCOS(I5)*ICREAL-RSIN(I5)*ICIMAG)*2)
```

```
C     ELSE
      TERM(I5)= FACTOR**2/2-ROUND((RCOS(I5)/2*FACTOR**2-ICREAL)**2+
     &  (RSIN(I5)/2*FACTOR**2+ICIMAG)**2)
C     ENDIF
7     CONTINUE
C     THE FOLLOWING SIMULATE A DECODER WHICH INPUTS THE SUFFICIENNT
C     STATISTICS ICREAL AND ICIMAGE (OR ANY INPHASE AND QUADRQTURE SAMPLED
C     VOLTAGES OF THE DEMODULATOR) AND TRELLIS SEARCH FOR THE MAXIMUM
C     LIKLIHOOD SEQUENCE. METMAX IS THE LARGEST METRIC FO RTHE
C     STATES AT A STAGE OF DECODING
      METMAX=-10000000
C     FOR EACH STATE, THERE ARE FOUR BRANCHES (I6) MERGING INTO IT.
C     IDMERGE IS THE METRIC OF THE SURVIVOR, WHICH LAST BRANCH IS IBRCH.
C     THE SURVIVOR IS PATHMEMD IN THE TABLE IPATH. THE STATE METRIC IS
C     THEN UPDATED.
      DO 8 CURSTATE=0,TOTSTATE-1
      IDMRGE=IDIST(LINKLIST(CURSTATE,1))
      IBRCH=1
      DO 9 I6=2,8
      ITEMP=IDIST(LINKLIST(CURSTATE,I6))
      IF (IDMRGE .GE. ITEMP) GO TO 9
      IBRCH=I6
      IDMRGE=ITEMP
```

```
9    CONTINUE
     IDNEW(CURSTATE)=IDMRGE+TERM(METRIC(CURSTATE))
     IPATH(CURSTATE,INDEX)=LINKLIST(CURSTATE,IBRCH)
C    THE STATE WITH THE LARGEST METRIC (LARGMET) IS FOUND AND PATHMEMD
     IF (IDNEW(CURSTATE) .LE. METMAX) GO TO 8
     METMAX=IDNEW(CURSTATE)
     LARGMET=CURSTATE
8    CONTINUE
C    THE SURVIVOR WITH THE LARGEST METRIC IS TRACED BACK A NUMBER OF
C    STATES TO FIND THE DECODED INFORMATION SEQUENCE. IPOINT SERVES
C    AS A POINTER TRACING FROM ONE STATE TO ANOTHER. THE LOCATION
C    OF STORAGE FOR LARGMET AT THE PRESENT DECODING STATE IS POINTED TO
C    BY THE POINTER CALLED INDEX, WHICH IS INCREMENTED BY MODULO
C    ARITHMETICS. THE INPUT TO THE ENCODER IS PATHMEMD BY A CIRCULAR
C    STRUCTURE CALLED IPATHMEM, SO THAT IT MAY BE RETRIEVED LATER FOR
C    COMPARISON WITH THE DECODED SEQUENCE.,'IS1'
     IPOINT=LARGMET
     ITR1=IPATHMEM(1,INDEX)
     ITR2=IPATHMEM(2,INDEX)
     ITR3=IPATHMEM(3,INDEX)
     TR=PATHMEM(INDEX)
     IPATHMEM(1,INDEX)=IDLAY1
     IPATHMEM(2,INDEX)=IDLAY2
```

```
      IPATHMEM(3,INDEX)=IDLAY3
      PATHMEM(INDEX)=DELAY
      IDELAY1=STATEBUFFER(V+3)
      IDELAY2=STATEBUFFER(V+2)
      IDELAY3=STATEBUFFER(V+1)
      DELAY=SOUT
      DO 10 I7=1,INDEX
      IPOINT=IPATH(IPOINT,INDEX+1-I7)
10    CONTINUE
      IF (INDEX .EQ. IMEM) GO TO 17
      ITIMES=IMEM-INDEX
      DO 11 I8=1,ITIMES
      IPOINT=IPATH(IPOINT,IMEM-I8+1)
11    CONTINUE
17    IF ((ITR1 .NE. INPUT(IPOINT,1)).AND.(INBITS .GT. IMEM*3))
     & IE=IE+1
      IF ((ITR2 .NE. INPUT(IPOINT,2)).AND.(INBITS .GT. IMEM*3))
     & IE=IE+1
      IF ((ITR3 .NE. INPUT(IPOINT,3)).AND.(INBITS .GT. IMEM*3))
     & IE=IE+1
      IF (INBITS .GT. IMEM*3) THEN
C     ******* FORM THE DECODER OUTPUT BITS ****
C     ******* INTO AN OCTAL CHARACTER.    ****
```

```
      KOUT=INPUT(IPOINT,1)*MAP3+INPUT(IPOINT,2)*MAP2+
     &    INPUT(IPOINT,3)*MAP1
C     ****** REMOVE RIGHT PADDED ZEROS ********
        KOUT=KOUT/ISHFT
        CALL EXPAND(KOUT,EXPDAT,INITE,NUMBIT)
        IF (NOIO .NE. 1)WRITE(14,*)EXPDAT
        CALL SNR(TR,EXPDAT,INITC,SNROUT)
      ENDIF
C     THE DISTANCE TABLE IS UPDATED
      DO 209 CURSTATE=0,TOTSTATE-1
      IDIST(CURSTATE)=IDNEW(CURSTATE)
209   CONTINUE
C     THE INDEX AND THE COUNT FOR THE NUMBER OF DECODED BITS ARE
C     INCREMENTED. IE IS THE NUMBER OF BIT ERRORS MADE. THE BIT ERROR
C     PROBABILITY IS COMPUTED FOR EVERY 10000 BITS, THE BER WOULD BE PRINTED
C     UNTIL THE DECODER HAS DECODED THE REQUIRED NUMBER OF BITS (NTOTAL).
      INDEX=INDEX+1
      IF (INDEX .GT. IMEM) INDEX=1
      INBITS=INBITS+3
      IF (MOD(INBITS,IFREQ) .NE. 0) GO TO 75
      BER=IE*1./INBITS
      WRITE(10,*) INBITS,'BITS DECODED,ERROR=',IE,' BER=',BER
      WRITE(6,*)'SNR OUT:',SNROUT
```

```
180   FORMAT(1X,I8,' BITS ARE DECODED,ERROR=',I5,' BER=',F11.10)
75    IF (INBITS .LT. NTOTAL) GO TO 18
      WRITE(10,270)
270   FORMAT('     X---X---X---X---X---X')
      IF (1 .EQ. 1) GO TO 71
      CLOSE(10)
      CLOSE(12)
      CLOSE(14)
      CLOSE(11)
2020  STOP
      END
C     THE SUBROUTINE DECTOBIN CONVERTS A DECIMAL NUMBER INTO A BINARY
C     NUMBER OF LENGTH KBITS: MSB IS STATEBUFFER(1), LSB IS STATEBUFFER(KBITS)
      SUBROUTINE DECTOBIN(IDEC)
      COMMON STATEBUFFER(15),KBITS,DELTA(5),P
      IQUOT=IDEC
      DO 16 I=1,KBITS
      STATEBUFFER(KBITS+1-I)=IQUOT/2**(KBITS-I)
      IQUOT=IQUOT-STATEBUFFER(KBITS+1-I)*2**(KBITS-I)
16    CONTINUE
      RETURN
      END
C     ROUND PERFORMS A ROUNDING FUNCTION.
```

```
C     RETURNS NEAREST WHOLE NUMBER
      FUNCTION ROUND(RE)
      IF (RE .GE. 0.) ROUND=RE+.5
      IF (RE .LT. 0.) ROUND=RE-.5
      RETURN
      END
      SUBROUTINE NOISE(X1,X2,POWER,ISEED)
C       GENERATES GAUSSIAN NOISE FROM TWO RANDOM NUMBERS WHERE X1 OR X2 IS
C     THE AMOUNT OF NOISE TO BE ADDED TO THE SIGNAL BASED ON THE
C     NUMBER POWER.
C     POWER —¡ NOISE VARIANCE.
C     X1 AND X2 ARE INDEPEDENT GAUSSIAN DISTRIBUTED RANDOM
C     VARIABLES.
C     DECLARE VARIABLES.
C     ISEED MUST BE KEPT AS A GLOBAL VARIABLE (I.E. KEEP
C     ITS CHANGES AT DIFFERENT CALLS OF NOISE)
      REAL X1,X2,POWER,U1,U2,PI,T1
      INTEGER ISEED
      PI=2*ACOS(-1.0)
      POW=POWER
      U1=RAN4(ISEED1,ISEED2,ISEED3)
      U2=RAN4(ISEED1,ISEED2,ISEED3)
      T1=(-2.0*ALOG(U1))**0.5
```

```
      X1=POW*T1*COS(2.0*PI*U2)
      X2=POW*T1*SIN(2.0*PI*U2)
      RETURN
      END
C     UNIFORM RANDOM NUMBER GENERATOR SUBROUTINE
C     OVER INTERVAL (0,1)
      FUNCTION RAN4(ISEED1,ISEED2,ISEED3)
      INTEGER A1,A2,A3,C1,C2,C3
      PARAMETER (A1=171,C1=0,M1=30269)
      PARAMETER (A2=172,C2=0,M2=30307)
      PARAMETER (A3=170,C3=0,M3=30323)
        ISEED1=MOD(A1*ISEED1+C1,M1)
        ISEED2=MOD(A2*ISEED2+C2,M2)
        ISEED3=MOD(A3*ISEED3+C3,M3)
        R1=FLOAT(ISEED1)/FLOAT(M1)
        R2=FLOAT(ISEED2)/FLOAT(M2)
        R3=FLOAT(ISEED3)/FLOAT(M3)
        RAN4= AMOD(R1+R2+R3,1.0)
      RETURN
      END
C  SUBROUTINE GRAN.F77                *
C  ;;; GENERATES A GAUSSIAN RANDOM NUMBER *
      FUNCTION GRAN(MEAN,SIGMA,ISEED1,ISEED2,ISEED3)
```

```
      REAL MEAN
          U=0
          DO 1 I=1,12
1           U = U +  RAN4(ISEED1,ISEED2,ISEED3)
          GRAN = SIGMA * (U-6) + MEAN
      RETURN
      END
      SUBROUTINE SOURCE(OUT,INIT)
      COMMON STATEBUFFER(15),KBITS,DELTA(5),P
      SAVE ISEED1,ISEED2,ISEED3
      IF (INIT.EQ.0)THEN
        ISEED1=12
        ISEED2=234
        ISEED3=3456
        OUT=0.0
        INIT=1
      ENDIF
        W=GRAN(0.0,1.0,ISEED1,ISEED2,ISEED3)
        OUT = P * OUT + W
      RETURN
      END
      SUBROUTINE DPCM(INDAT,OUTSIG,ISTAT,NUMBIT)
      REAL INDAT
```

```
      SAVE SHAT
      IF (ISTAT .EQ. 0) THEN
        SHAT=0.0
        ISTAT=1
      ENDIF
        ERROR=INDAT-SHAT
        CALL QUANT(ERROR,OUTSIG,NUMBIT,EST)
        CALL PRED(EST,SHAT)
C       WRITE(6,*)ERROR,EST,OUTSIG
      RETURN
      END
      SUBROUTINE EXPAND(DATIN,SHAT,ISTAT,NUMBIT)
      COMMON STATEBUFFER(15),KBITS,DELTA(5),P
      INTEGER DATIN
      IF (ISTAT .EQ. 0) THEN
        SHAT=0.0
        ISTAT=1
      ENDIF
        Q=2**NUMBIT
        OUT2=DELTA(NUMBIT)*(DATIN-(Q-1)/2.0)
        SHAT=SHAT+OUT2
        CALL PRED(0.0,SHAT)
      RETURN
```

```
      END
      SUBROUTINE PRED(X1,X2)
      COMMON STATEBUFFER(15),KBITS,DELTA(5),P
        X2=P* (X1+X2)
      RETURN
      END
      SUBROUTINE QUANT(VALIN,OUT1,NUMBIT,OUT2)
      COMMON STATEBUFFER(15),KBITS,DELTA(5),P
      INTEGER OUT1
        OUT1 =0
        TESTVL=VALIN+DELTA(NUMBIT)*2**(NUMBIT-1)
        DO 1 TESTBIT=NUMBIT,1,-1
C       IOUT(NUMBIT-TESTBIT+1)=0
          SARVAL=DELTA(NUMBIT)*2**(TESTBIT-1)
          IF (TESTVL.GE.SARVAL)THEN
C           IOUT(NUMBIT-TESTBIT+1)=1
            OUT1=OUT1+2**(TESTBIT-1)
            TESTVL=TESTVL-SARVAL
          ENDIF
1         CONTINUE
        Q=2**NUMBIT
        OUT2=DELTA(NUMBIT)*(OUT1-(Q-1)/2.0)
        RETURN
```

```
      END
     SUBROUTINE SNR(SIG1,SIG2,INIT,SNROUT)
     COMMON STATEBUFFER(15),KBITS,DELTA(5),P
     SAVE ERR, NTOTAL
C    PROGRAM CALCULATES THE SNR DIFFERENCE BETWEEN TWO
C    REAL VALUED SIGNAL FILES AS:
C    SNR=10 LOG( 1/E-[S1-S2]**2~*(1/(1-P**2)) )
     IF (INIT .EQ. 0) THEN
       NTOTAL=0
       ERR=0.0
       INIT=1
     ENDIF
     GAIN=1.0/(1-P**2)
     NTOTAL=NTOTAL+1
1      ERR=(SIG1-SIG2)**2+ERR
     ET2=ERR/NTOTAL
     SNROUT=10.0*LOG10( (1.0/ET2)*GAIN)
     RETURN
     END
```

The follow is the input file v34set.dat

```
1 SURPRESS I/O
0 0 2  MEMORIES PER TAP BIN
1      PSK(0),QAM(1),GREYPSK(-1),QAMPK(2),12/4PSK(3)
```

```
10     SCALING FACTOR

100    SURVIVOR LENGTH

12

234

3456   ISEED1,ISEED2,ISEED3

300300 RUN LENGTH

300300 REPORT FREQUENCY

1 2 3  DPCM BIT ORDER INTO CODER

3      DPCM BITS PER SYMBOL

.95    CORRELATION COEFFICIENT

0      USE ENERGY PER SYMBOL

1 0 0 0 0 H0 TAPS

0 1 0 0 0 H1 TAPS

0 0 1 0 1 H2 TAPS

0 0 0 1 0 H3 TAPS

100    SNR

800    excape snr (exceeds 210)
```

The following is the sample output of the simulation program

```
ENTER 1 TO SUPRESS DATA I/O

INPUT V, THE CONTRAINT LENGTH

INPUT 0 FOR 16-PSK, 1 FOR 16-QAM

INPUT THE SCALING FACTOR FOR THE QUANTIZER

INPUT THE LENGTH OF THE SURVIVOR TO BE STORED

INPUT THE SEEDS FOR THE RANDOM NUMBER GENERATOR
```

References

[1] J.W. Modestino and D.G. Daut, "Combined Source-Channel Coding of Images," *IEEE Trans. on Commun.*, vol COM-27, pg 1644-1659, November , 1979.

[2] J.W. Modestino, D.G. Daut and A.L. Vickers, "Combined Source-Channel Coding of Images Using the Block Cosine Transform," *IEEE Trans. on Commun.*, vol COM-29, pg 1261-1274, September , 1981.

[3] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding.* New York, McGraw-Hill, 1979.

[4] E. Ayanoğlu and R.M. Gray, " The Design of Joint Waveform Coders," *IEEE Trans. Information Theory*, IT-33, pg 855-865, November 1987.

[5] N. Farvardin and V. Vaishampayan, " Optimal Quantizer Design for Noisy Channels: An Approach to Combined Source- Channel Coding," *IEEE Trans. Information Theory*, IT-33, pg 827-838, November 1987.

[6] G. Freeman *et al*, " Trellis Source Codes Designed by Conjugate Optimization," *IEEE Trans. Commun.*, vol COM-36, pg 1-12, January 1988.

[7] J.L. Massey, "Coding and Modulation in Digital Communications," *Proc. 1974 Int. Zurich Seminar on Digital Commun.*, Zurich, Switzerland, pp E2(1)-(4), March 1974.

[8] J.M. Wozencraft and R.S. Kennedy, "Modulation and Demodulation for Probabilistic Coding," *IEEE Trans. Inform. Theory*, vol. IT-12, no. 3, July 1966.

[9] R.G. Gallager, " A Simple Derivative of the Coding Theorem and Some Applications," *IEEE Trans. Information Theory*, IT-11, 3-18, January 1965.

[10] R.G. Gallager, *Information Theory and Reliable Communication.* New York, Wiley, 1968.

[11] J.M. Wozencraft and I.M. Jacobs, *Principles of Communication Engineering.* New York, Wiley, 1965.

[12] A. Saleh and J. Salz, "On the computational cutoff rate, R_0, for the peak-power-limited gaussian channel," *IEEE Trans. Commun. Theory*, vol. COM-35, no. 1 , Jan 1987.

[13] E. Biglieri, "High-Level Modulation and Coding for Nonlinear Satellite Channels," *IEEE Trans. Commun. Theory*, vol. COM-32, no. 5, Jan 1984.

[14] G. Ungerboeck,"Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp 55-67, Jan 1982.

[15] J.P. Odenwalder, "Optimal decoding of convolutionl codes", Ph.D. dissertation, Dep Syst. Sci., Eng. Appl. Sci. Univ. California, Los Angeles, 1970.

[16] G.D. Forney,"The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp 268-287, March 1973.

[17] J.C. Proakis, *Digital Communications*. New York, McGraw-Hill, 1983.

[18] S.J. Orfanidis, *Optimum Signal Processing*. New York, Macmillan, 1985.

[19] K. Chang and R.W. Donaldson, "Analysis, Optimization, and Sensitivity Study of Differential PCM Systems Operating on Noisy Communication Channels," *IEEE Trans. Commun. Theory*, vol COM-20, no. 3, June 1972.

[20] D.P. Taylor and H.C. Chan, "A simulation of two bandwidth-efficient modulation techniques," *IEEE Trans. Commun. Theory*, vol COM-29, Jan 1981.

[21] J.A. Heller and I.M. Jacobs, "Viterbi decoding for satellite and space communication," *IEEE Trans. Commun. Theory*, vol. COM-19, Oct 1971.

[22] A.J. Viterbi, "Convolutional Codes and Their Performance in Communcation Systems," *IEEE Trans. on Commun.*, vol COM-20, No. 5, pg 751-772, October, 1971.

[23] G. Ungerboeck,"Trellis-coded modulation with redundant signal sets-Part I:Introduction," *IEEE Communications Magazine*, vol 25, no. 2, February 1987.

[24] G. Ungerboeck,"Trellis-coded modulation with redundant signal sets-Part II:State of the Art *IEEE Communications Magazine*, vol 25, no. 2, February 1987.

[25] J. Max,"Quantizing for Minimum Distortion," *IRE Trans. Infor. Theory*, vol. IT-6, pp.7-12, March 1960.

[26] S. Wilson *et al*,"Rate 3/4 Convolutional Coding of 16-PSK: Code Design and Performance Study," *IEEE Trans. on Commun.*, vol COM-32, no. 12 , December 1984.

[27] T. Berger, *Rate-Distortion Theory: A Mathematical Basis for Data Compression*, Prentice Hall, Englewood Cliffs,N.J.,1971.

[28] G. Einarsson," Signal Design for Amplitude-Limited Gaussian Channel by Error Bound Optimization," *IEEE Trans. on Commun.*, vol COM-27, no.1, January 1979.

[29] M.L. Honig, "On Constructing Embedded Multilevel Trellis Codes," *IEEE Trans. on Commun.*, vol COM-36, No. 2, February 1988.

[30] J.E. Savage, "Sequentail decoding - The computational problem," *Bell Syst. Tech. J.*,vol. 45, pp. 149-175, January 1966.

[31] E. Arikan,"An Upper Bound on the Cutoff Rate of Sequential Decoding," *IEEE Trans. Inform. Theory*, vol IT-34, no. 1, January 1988.

[32] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. I, 3^{rd} Edition, John Wiley, 1968.

[33] M.C. Jeruchim,"Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," *IEEE Journal on Selected Area in Commun.*, vol. SAC-2, no. 1, pp. 153–170, January 1984.

[34] B.A. Wichmann and I.D. Hill, "An Efficient and Portable Pseudo-random Number Generator," *Applied Statistics*, vol 31, no. 2, pp. 188–190, 1982.

[35] R.F.W. Coates *et al* "Monte Carlo Simulation and Random Number Generation," *IEEE Journal on Selected Area in Commun.*, vol. SAC-6, no. 1, pp. 58–66, January 1988.

[36] A.R. Calderbank and N.J.A. Sloane,"New Trellis Codes Based on Lattices and Cosets," *IEEE Trans. Inform. Theory*, vol IT-33, no. 2, March 1987.

[37] A.R. Calderbank and J. Mazo,"A New Description of Trellis Codes," *IEEE Trans. Inform. Theory*, vol IT-30, no. 6, March 1984.

[38] G.D. Forney, "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Inform. Theory*, vol IT-16, no. 6, November 1970.

[39] E. Zehavi and J. Wolf, "On the Performance Evaluation of Trellis Codes," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 2, March 1987.

[40] D. Divsalar and M.K. Simon, " Multiple Trellis Coded Modulation," *IEEE Trans. on Commun.*, vol. 36, no. 4, April 1988.

[41] G.D. Forney *et al*,"Efficient Modulation for Band-limited Channels," *IEEE Journal on Selected Area in Commun.*, vol. SAC-2, no. 5, September 1984.

[42] J.K. Omura, "Performance bound for Viterbi algorithm," *Conf. Rec., Int.Conf.Commun.*, Denver, CO, June 1981, pp. 2.2.1-2.2.5.

[43] H.Taub and D.L. Schilling, *Principles of Communication Systems*, 2^{nd} Edition. New York, McGraw-Hill,1986, pp. 594-603.

[44] J. Hui,"Joint Coding and Modulation Designs for the Bandlimited Satellite Channel," Masters Thesis, Massachuset Institute of Technology, 1981.