SIMULATION OF BINARY CONTINUOUS PHASE MODULATION COMBINED WITH (1,2)-CONVOLUTIONAL ENCODER

Frank Carden
William Osborne
Karim Alhussiny

Center for Space Telemetering and Telecommunications Systems, New Mexico State University, Las Cruces New Mexico 88003

Abstract

In this paper the performance of a (1,2)-convolutional encoder combined with continuous phase modulation is simulated. A binary sequence is used inconjunction with the above encoder and a modulation index of (h=1/4). A full response 1REC frequency shaping function is used to maintain phase continuity. A binary (uncoded)CPM with the above modulation index is also simulated. The performance in terms of the probability of bit error event is plotted against signal to noise ratio for both coded and the uncoded CPM schemes. The asymptotic performance of both schemes is plotted along with MSK for comparison purposes. The simulation algorithm used in this paper utilizes the Block Oriented System Simulator known as BOSS. The major components of this simulation are the encoder and the decoder. The encoder consists of binary random data generator and a (1,2)-convolutional encoder combined with a channel vector encoder and a random white gaussian noise generator. The decoder consists of the following modules: sequential vector bank, inner product unit, metric calculator unit, multi stage trellis, symbol decoder and error counter module.

Introduction

In recent years many papers have been published in the area of Continuous Phase Modulation (CPM) [1],[2],[3],[4]. This class of signals is characterized by its constant envelope and continuous phase. CPM provides power-bandwidth tradeoff in terms of coding gain. Figure 1 depicts a general block diagram of the combined coding with continuous phase modulation. In this paper we present a systematic approach in our simulation of the above class of modulation schemes.

The end-to-end System behavior is as follows: It is shown in figure 3 that a binary stream of data is encoded by the (1,2)-convolutional encoder. The output of this encoder is applied to the continuous phase modulator to produce the noise free channel symbols. These symbols are transmitted over a noisy channel characterized by AWGN. After demodulation and orthogonalization using the Gramm-Schmidt procedure, each of these demodulated symbols are represented by a four dimensional vector[5]. The received symbols are now contaminated with this noise; specifically each component of the four dimensional vector is corrupted with white Gaussian noise. Each component of the noise vector is represented by an independent Gaussian random variable. The received vector is projected on eight vectors per stage of the trellis. A total of 16 vectors are stored in the sequential vector bank module. Every time a symbol is received, namely the four dimensional vector, a projection is performed using the inner product unit. This projection is now stored as a metric in the CPM metric calculator unit. Only four vectors out of the 8 original vectors are stored[5]. The cumulative metric is stored in the running metric and symbol decider module and the surviving symbol, namely(+/-1) is stored in the CPM trellis module. This process is done for a total of 30 symbols before a decision is made [1]. Owing to the inherent correlative memory among the adjacent symbols. this delay is necessary for the optimum detection of this class of signals as it is required by the MLSE[4]. On the other hand only ten symbols were observed before making a decision on the received symbol when we treated the uncoded case (figure 2).

Finally the decoded symbol is compared with the original random data symbol using the symbol error counter module. This module registers an error every time there is a disagreement between the input symbols. The total number of errors made in a single simulation for a given noise variance represents a point in Figure 5.

Simulation Approach And Module Description

Figure 4 shows a block oriented diagram used in the simulation of both the coded and the uncoded cases. The simulation starts with the generation of the received 4-dimensional vector. That is, the encoder module consists of a sequential finite state machine that generates a vector corresponding of a symbol at the receiver. This module contains 16 vectors stored in the memory of this module. These vectors are called upon request by the finite state machine. For a given state of the encoder the next state is decided on the basis of a symbol one or zero from the random data generator. This causes interdependence among the adjacent successive symbols.

In addition, continuous phase modulation exhibits this intrinsic correlative property among the neighboring symbols, and hence justifies the need for a Vitrebi algorithm and the utilization of the so called MLSE[l] for optimum detection.

The vector that has just been produced by the previous module is now , added to a four dimensional random Gaussian noise. Each component of this vector is independent of the other. This noise vector is characterized by its mean and variance. The variance represents the power associated with the noise. This noise variance is exported to the highest level of the simulation as a parameter that can be varied to accommodate different signal to noise levels. The addition of the noise simulates the stationary channel over which the symbols are communicated.

In the foregoing development, we have implied that these noisy received vectors are the output of the synchronous receiver, and now for optimum detection, a maximum likelihood sequence estimator is the next processor in this simulation.

At this point, we now introduce the noisy received vector to the first stage of the decoder. The first stage of the decoder is represented by the trellis module. This module contains four states per stage of the trellis[6]. At any given time we only have four stages[4]. We depart a particular state to another depending upon whether a symbol one or zero is received. The decoder depth is 30 and 10 symbols interval for the coded and the uncoded simulations respectively[6]. The trellis module is simulated by delay elements to simulate the memory property of the trellis.

Every time a vector, namely a symbol, is received, a projection is performed on each of the vectors of the trellis. The projection process is performed in the inner product unit. This module contains 8 submodules that performs the dot product operation. The output of this module is a metric that is used along with the previous cumulative metric in deciding which of the two contenders into a given node or state, survives.

Eight vectors are projected on the received vector. These 8 vectors are stored in the sequential vector bank module along with the remaining 8 vectors that correspond to the next stage of the trellis. These 8 vectors are introduced every other symbol interval for projection process. The projection of each vector is stored along with the symbol that has survived for that specific stage[5]. The surviving metric is stored in the running metric module as shown in Figure 4. The symbol is stored within the delay elements of the trellis awaiting the final decision after observing some integer number of intervals decided by the decoder depth. In our treatment of the coded case the decision depth is taken to be 30 symbols interval. This distance is taken such that all merged or unmerged paths overcome the minimum euclidean distance of that decoder[6].

The running metric module is made out of two submodules, one controller submodule sequentially enables the other submodule that in structure resembles the two different stages of the trellis except it tracks the metric rather than symbols.

Up to this point we have stored the cumulative metrics in the running metric module and the associated symbols in the the delay elements of the trellis, this is done for a total of 30 symbols interval for the coded case (Figure 3)and a total of 10(Figure 2) for the uncoded case, before we decide which symbol was transmitted. Having observed the corresponding decision depth, a decision now is made on the symbol that was transmitted 30 symbols ago. This final decision is made using a submodule in the running metric and symbol decider unit as shown in Figure 4. The cumulative metric, over the last 30 (coded) or 10 (uncoded) intervals, selects among four final contenders. This selection is predicated upon the path with the largest cumulative metric.

To evaluate the performance of this signaling scheme a comparator is designed to compare the original (noise free) transmitted symbol with the decoded symbol (Figure 4). This is done using the symbol error counter that starts comparing after 30 symbols interval, this is done to provide ample time for the decoder to overcome the transient response (start up time) and achieve a steady state performance.

Finally for each run a point is plotted for the given noise power(variance) and the symbol energy against the total number of errors registered by the symbol error counter. The results of all simulations are shown in Figure 5. Results show that as the signal to noise level increases for either the coded or the uncoded scheme, the simulated results tends to approach the asymptotic bound developed by Aullin and Sundberg[l]. A gain of 4.3 db is achieved when the binary data is encoded with above (1,2)-encoder.

Conclusion

Performance of continuous phase modulation combined with (1,2)-encoder is simulated and plotted against signal to noise ratio. The uncoded case is also plotted on the same graph. Simulations were done using the BOSS simulator which is a fortran oriented code. The simulation was casted in two parts, the encoder and the decoder, all possible transitions were represented by 16 vectors. The vectors are stored in a bank and presented every symbol interval. A symbol is stored in the trellis and its corresponding metric is stored in the running metric module. A decision is made after observing symbols over the entire decision depth. An error event is counted using the error counter module. The performance is obtained by plotting the probability of bit error against symbol energy to noise ratio. The coding gain is observed over the uncoded case along with the asymptotic peformance.

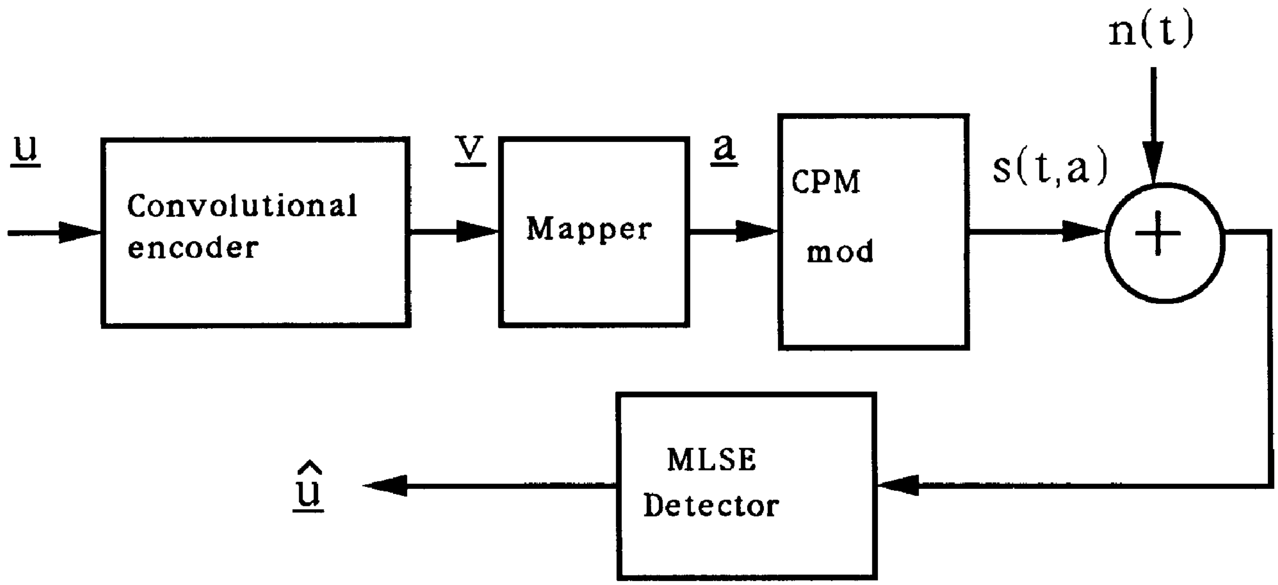

Figure 1. The modulation scheme,channel,and receiver.

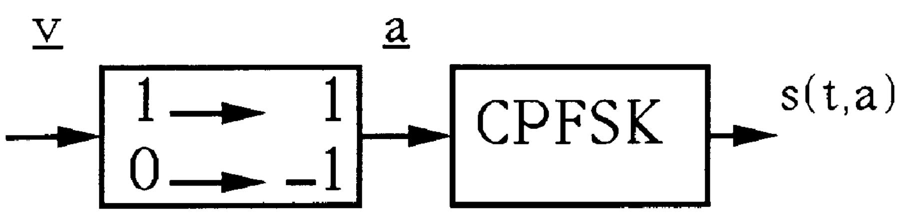

Figure 2. Binary CPFSK(uncoded).

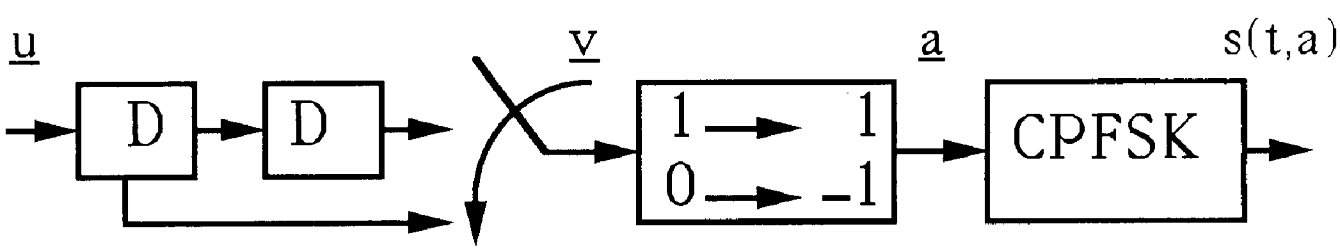

Figure 3. Binary CPFSK with (1,2)-encoder.

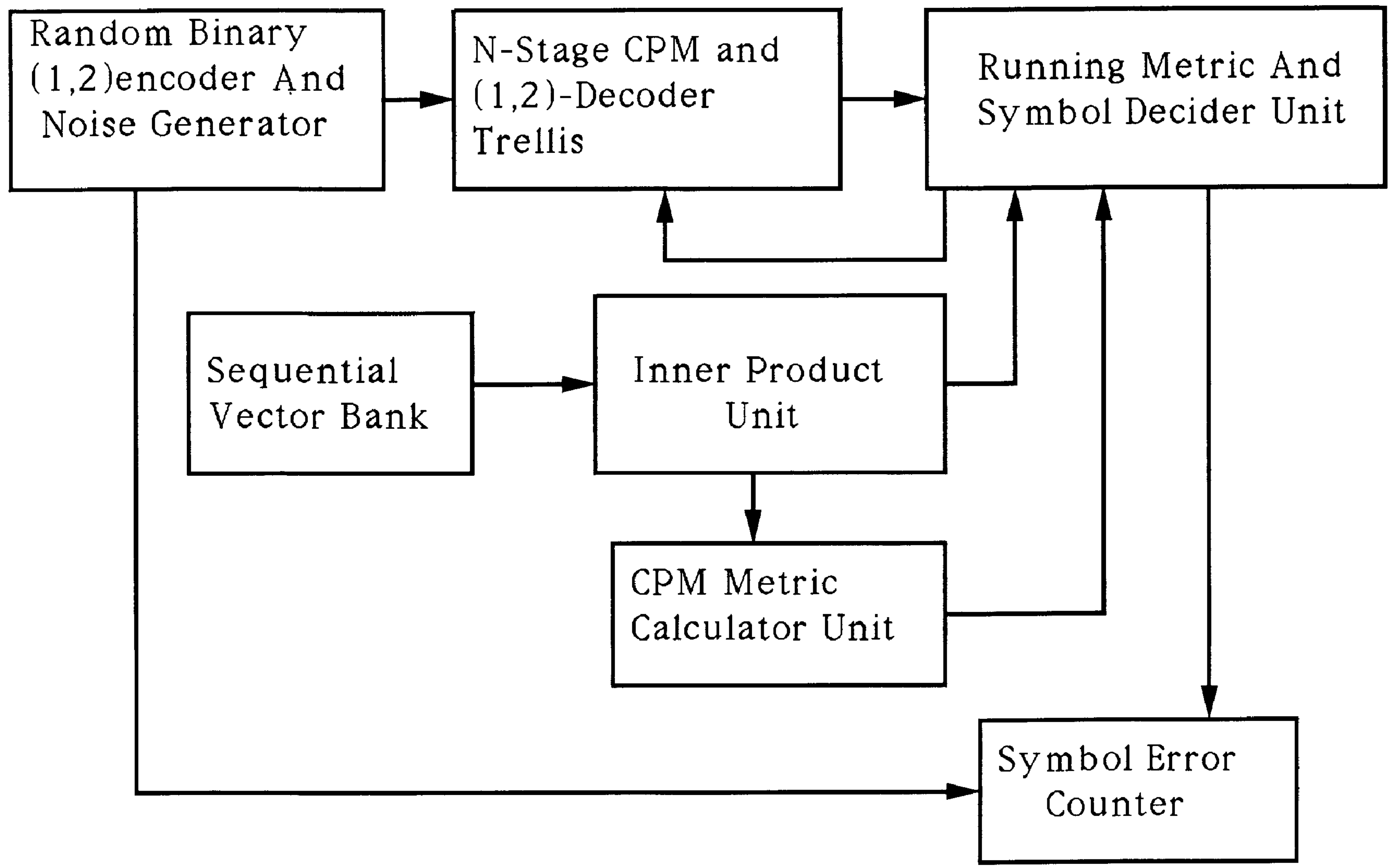

Figure 4. Encoder and decoder block diagram.

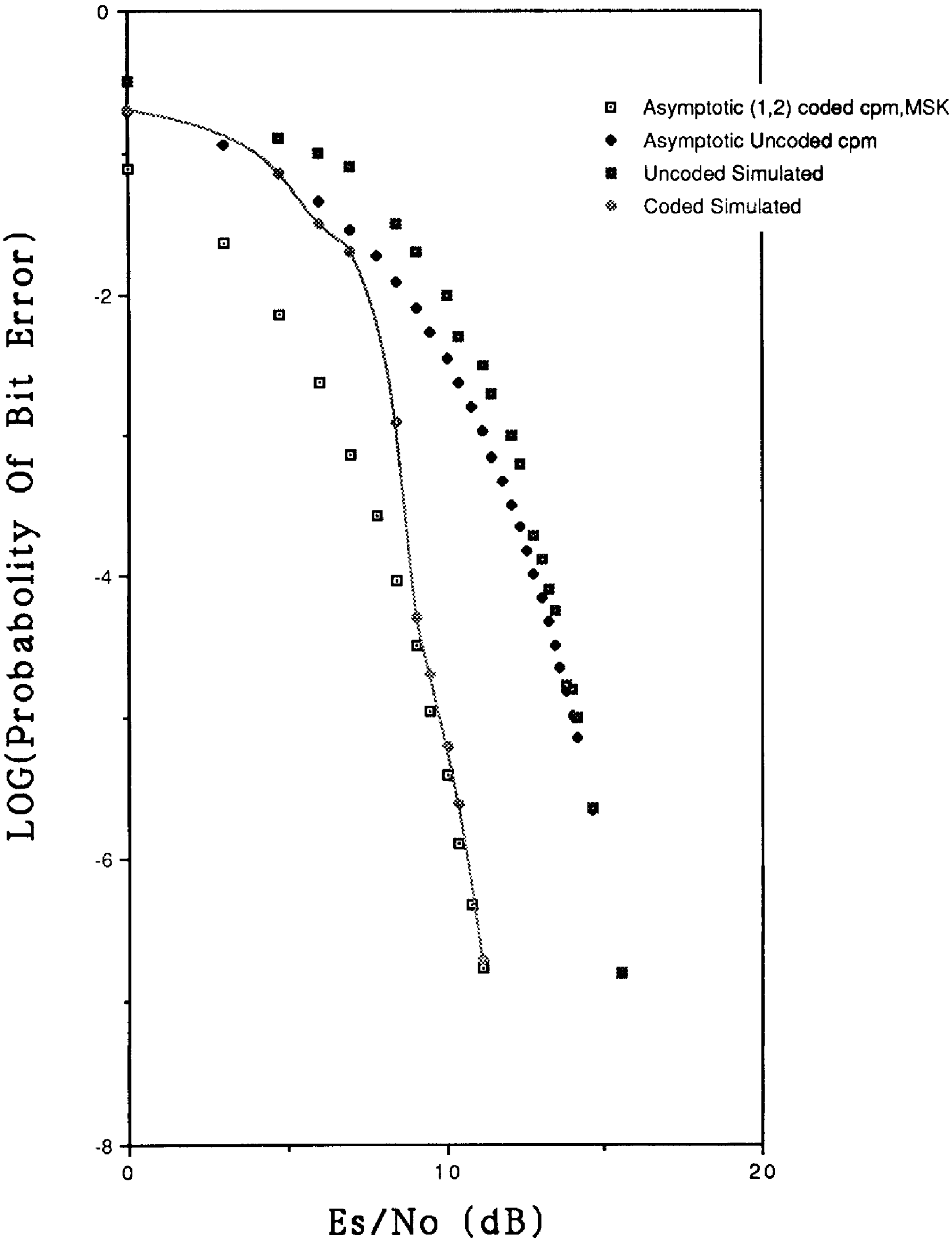

FIGURE 5. SIMULATED RESULTS

References

[1] Frank Carden , W.P. Osborne, Karim Alhussiny, unpublished notes, May 1992.

[2] T. Aullin and C. E. Sundberg, "Continuous Phase Modulation, Part 1: Full Response Signaling," IEEE Commun., Vol. COM-29 Trans., pp. 196-209, March.

[3] T. Aullin N. Redbeck, and C. E. Sundberg, "Continuous Phase Modulation, Part 11: Partial Response Signaling," IEEE Commun., Vol. COM-29 Trans. Commun., Vol. COM-29, pp. 210-225 March 1981.

[4] Frank Carden and Michael Ross, "A Spectrally Efficient Communication System Utilizing a Quantized Euclidean Decoder," Proceedings of the the International Telemetering Conference, Vol. XXV, pp. 575-582, October 1989.

[5] A. premji and D.P. Taylor, "Receiver structures, for multi-h signaling formats," IEEE Trans. Comm., vol COM -35, pp.439-451, Apr.1987.

[6] T. Aullin and C. E. Sundberg, "On the minimum Euclidean distance for a class of signal space codes, IEEE Trans. Inform. Theory, vol.IT-28, pp. 43 - 55, Jan. 1982.

[7] Rodger E. Ziemer and Roger L. Peterson, Digital Communications and Spread Spectrum Systems. 1985.

[8] J. M. Wozencraft and I. M. Jacobs, Principals of Communication Engineering New York: Wiely, 1965.

[9] S. Benedetto, E. Biglieri V. Castell Ani. Digital Transmission Theory. 1987 Prentice-Hall, Inc.

[10] J. Anderson, T. Aulin C-E. Sundberg. Digital Phase Modulation.