View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by NASA Technical Reports Server

NASA-CR-191903

IN-17-CR

141143

p. 26

SEMI-ANNUAL STATUS REPORT

Telemetering and Telecommunications Research

Period Covered January 1991 - July 1991

NASA Grant No. NAG 5-1491

Principal Investigator: Dr. William P. Osborne

New Mexico State University
Electrical and Computer Engineering
Box 30001 - Dept. 3-0
Las Cruces, New Mexico 88003

(NASA-CR-191903) TELEMETERING AND TELECOMMUNICATIONS RESEARCH Semiannual Status Report, Jan. - Jul. 1991 (New Mexico State Univ.) 26 p

N93-18777

Unclas

G3/17 0141143

Introduction

The research center activities during the reporting period have been focused in three areas.

These are:

(1) Developing the necessary equipment and test procedures to support the testing of 8PSK-TCM through TDRSS from the WSGT.

(2) Extending the theorical decoder work to higher speeds with a design goal of 600MBPS at 2 bits/Hz.

(3) Completing the initial phase of the CPFSK Multi-H research and determining what subsets (if any) of these coding schemes are useful in the TDRSS environment.

Results

During the reporting period, January 1992 to July 1992, significant progress was made in the three areas discussed in the introduction. This report will summarize the WSGT test and the high speed decoder work while focusing upon the Multi-H work.

The equipment for the WSGT TCM testing has been completed and is functioning in the lab at NMSU. Measured results to date indicate that the uncoded system with the modified HRD and NMSU symbol sync operates at 1 to 1.5 dB from theory when processing encoded 8PSK. The NMSU pragmatic decoder when combined with these units produces approximately 2.9 dB of coding gain at 10^{-5} BER. The testing of this equipment will move to the WSGT in late July. The test results and the theorical analysis of this work will be the focus of the next 6 month report.

Th high speed decoder effort has continued during the reporting period. Mike Ross's Ph.D. work has focused on this effort. Mr. Ross will receive his Ph.D. in August and the detailed design of a high speed 16 state TCM codec will be complete at that time. The intent of the program is to investigate how fast this decoder design can operate if implemented in various VLSI technologies. This work will take place in the fall 1992. The design which has been developed by Mr. Ross is a pipelined technique with a unique approach to the metric calculations that should result in much faster operation than is possible using metric looking tables. The details of this design and its projected performance will be included in the January report.

Our study of CPFSK with Multi-H coding has reached a critical stage. The work to date and the detailed results are contained in the next section. The principal conclusions reached in this activity are:

(1) No scheme using Multi-H alone investigated by us or found in the literature produces power/bandwidth trades that are as good as TCM with filtered 8PSK.

(2) When Multi-H is combined with convolutional coding, one can obtain better coding gain than with Multi-H alone but still no better power/bandwidth performance than TCM and these gains are available only with complex receivers.

(3) The only advantage we can find for the CPFSK schemes over filtered MPSK with TCM is that they are constant envelope. However, constant envelope is of no benefit in a multiple access channel and of questionable benefit in a single access channel since driving the TWT to saturation in this situation is generally acceptable.

(4) Based upon these results the center's research program will focus on concluding the existing CPFSK studies and closing out this task.

AN OVERVIEW OF

MULTI-H RESEARCH AT NMSU

MULTI-H PHASE MODULATION

Multi-h Background

Multi-h phase modulation is a technique for transmitting information through the phase component of a carrier wave in such a way that the phase changes continuously over the interval to prevent jumps in the phase component.
The general form for a multi-h continuous phase modulation signal is

$$s(t) = \sqrt{\frac{2E_s}{T_s}}\cos[2\pi f_c t + \Phi(t,\alpha) + \phi_o]$$

where

E$_s$ is the energy of the signal
T$_s$ is the time length of the signaling interval
f$_c$ is the carrier frequency
ϕ_o is the arbitrary initial phase of the carrier
$\Phi(t,\alpha)$ is the information carrying phase function given by

$$\Phi(t,\alpha) = 2\pi\int_{-\infty}^{t}\sum_{i=-\infty}^{\infty} h_i\alpha_i g(t-iT_s)dt$$

where

α_i is the data value
h$_i$ is the phase modulation index
g(t) is the frequency pulse-shape function which determines how the phase changes over time.

Many pulse shaping functions are available. Some of the more common one are: LREC (rectangular pulse shape function with pulse length L), and LRC (raised cosine with pulse length L) [17]. A commonly-used and easy-to-implement pulse shape function is 1REC. Using the 1REC pulse shaping function and using the property that the function is periodic (as is the cosine function), then the signal can be written as

$$Y_i = \sqrt{\frac{2E_s}{T_s}}\cos\{2\pi[f_c t + \frac{\alpha_i h_i}{2}(\frac{t}{T_s} - (i-1))] + \Phi_i\}$$

$$(i-1)T_s \le t \le iT_s$$

$$\Phi_i = \pi\sum_{j=-\infty}^{i-1}\alpha_j h_j$$

The quantity Φ_i is interpreted as the excess phase due to all previous information digits.

To receive the signal, one forms all ideal received signals, S_{ij} which are written as

$$S_{ij} = \sqrt{\frac{2E_s}{T_s}}\cos[2\pi t(f_c + \frac{\alpha_j h_i}{2T_s}) + \Phi_{ij}]$$

where Φ_{ij} is the excess phase due to previous information digits over all possible received signals, j and α_j is all possible received data.

Without noise, the received signal should be chosen to be the S_{ij} such that

$$Y_i - S_{ij} = 0$$

where Y_i is the actual received signal. With noise, the S_{ij} should be chosen to minimize this difference.

To aid in this decision, the innerproduct (correlation) between Y_i and S_{ij} is formed, that is

$$\beta_{ij} = (Y_i, S_{ij}) = \int_{(i-1)T_s}^{iT_s} Y_i(t)S_{ij}(t)dt$$

The signal S_{ij} that gives the largest metric (best correlation), β_{ij}, is then chosen as the received signal.

Using the form for the ideal signal S_{ij} as

$$S_{ij} = \sqrt{\frac{2E_s}{T_s}}\{\cos[2\pi t(f_c+\frac{\alpha_j h_i}{2T_s})]\cos[\Phi_{ij}] - \sin[2\pi t(f_c+\frac{\alpha_j h_i}{2T_s})]\sin[\Phi_{ij}]\}$$

a basis expansion for data output, α_j , can be formed using the basic functions that make up S_{ij}:

where j = 1, 2, . . . , M, where M is the number of different output data. For binary output $\alpha_j = +1, -1$ and the basis functions are:

An orthonormal basis is then formed to expand the received signal into four coefficients.

$$\sqrt{\frac{2E_s}{T_s}}\cos[2\pi t(f_c + \alpha_j \frac{h_i}{2T_s})]$$

$$\sqrt{\frac{2E_s}{T_s}}\sin[2\pi t(f_c + \alpha_j \frac{h_i}{2T_s})]$$

$$\sqrt{\frac{2E_s}{T_s}}\cos[2\pi t(f_c + \frac{h_i}{2T_s})]$$

$$\sqrt{\frac{2E_s}{T_s}}\sin[2\pi t(f_c + \frac{h_i}{2T_s})]$$

$$\sqrt{\frac{2E_s}{T_s}}\cos[2\pi t(f_c - \frac{h_i}{2T_s})]$$

$$\sqrt{\frac{2E_s}{T_s}}\sin[2\pi t(f_c - \frac{h_i}{2T_s})]$$

These coefficients are given by:

for $\alpha_j = +1$,

$a_{1i} = \cos(\Phi_i)$
$a_{2i} = -\sin(\Phi_i)$
$a_{3i} = 0.0$
$a_{4i} = 0.0$

or by:

for $\alpha_j = -1$,

$a_{1i} = C_{1i} \cos(\Phi_i) + C_{2i} \sin(\Phi_i)$
$a_{2i} = C_{2i} \cos(\Phi_i) - C_{1i} \sin(\Phi_i)$
$a_{3i} = D_i \cos(\Phi_i)$
$a_{4i} = -D_i \sin(\Phi_i)$

where

Φ_i = the phase state during time t,
$iT_s \leq t \leq (i+1)\, T_s$

and,

$$C_{1i} = \frac{\sin(2\pi h_i)}{2\pi h_i}$$

$$C_{2i} = \frac{[\ 1 - \cos(2\pi h_i)]}{2\pi h_i}$$

$$D = \sqrt{1 - C_{2i}^2 - C_{1i}^2}$$

These coefficients are then used to calculate all possible metrics for each possible phase state and each possible input. The largest metric for each phase state, the data input, and the trellis path are then stored. As more data comes in, a cumulative metric is kept along with the appropriate trellis path and data bits. After a specified time interval, a decision is made as to the appropriate path based on the cumulative metric and data is output from the appropriate trellis path. The value chosen in the first part of our multi-h work was taken as 5 times the constraint length of the code. The constraint length is the number of time intervals it takes for the trellis to have its first merge. That is, starting at one particular phase state, two different paths can be taken to arrive at the same point. The longer the constraint length, the less the chance of error and the better the code. Through recent simulations, it has been found that if the trellis length (or memory length) is taken too long, then errors can be introduced due to longer merge paths which may then give the same result as the correct path[1]. See also the simulation results in Table 7. Hence, there is an optimal length for the trellis that will need to be found for each multi-h scheme used. But complexity of the trellis, and complexity of calculations are trade-offs. To offset the complexity of the calculations, all possible four-tuple vectors consisting of the coefficients can be calculated ahead of time. This vector can be given by:

$$A_i = (a_{1i}, a_{2i}, a_{3i}, a_{4i})$$

This A vector over the ith time interval, is in a four dimensional space. For multi-h values of h = (1/2, 3/4), eight four-dimensional vectors are stored at each stage of the trellis. Each vector represents the possible transition on a symbol for a given phase state. The trellis for the h = (1/2, 3/4) decoder consists of 4 different stages. The trellis would then repeat these four stages until the desired trellis length is reached. Using the 4 stage trellis, a total of 32 vectors are needed and stored (or the 8 needed vectors can be calculated each time) to represent all possible transitions and stages. As noise is added to the system, the received vector is projected onto all vectors stored in the appropriate stage and a metric is calculated and stored to be used for the maximum likelihood sequence estimator (viterbi). The above simulation procedure utilizes the geometric representation of the multi-h wave form rather than the time dependent representation. In so doing, the simulation time is reduced and the memory requirement is minimized.

In general [20], for M-ary data, K values of h can be used. The values of h_i are selected so that

$$h_i = L_i/q$$

where L_i is the sequence $i = 1, 2, \ldots, K$ and q are integers such that $q \geq M^K$ and the weighted sums of the h_i must not be an integer value. That is,

$$a_1h_1 + a_2h_2 + \cdots + a_Kh_K \neq \text{integer}$$

for $a_1, a_2, \cdots, a_K \in \{0, 1, \ldots, M\text{-}1\}$.

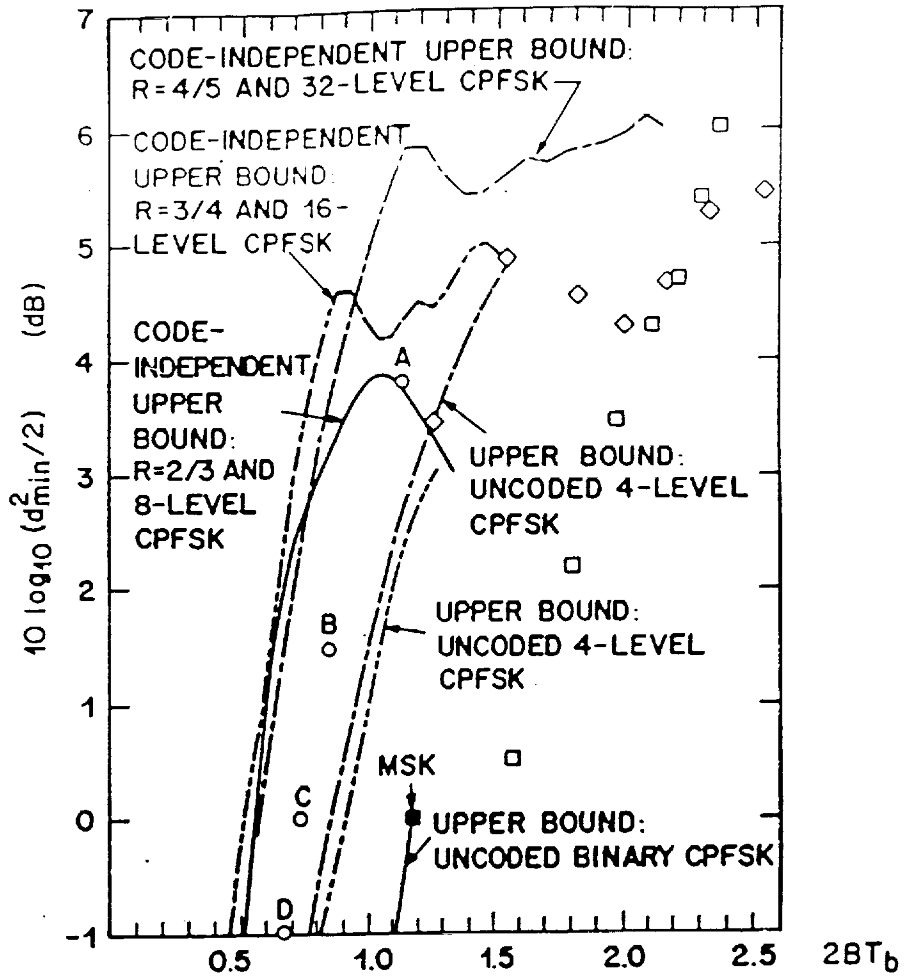
The number of phase states per time period, T_s, is equal to q. The distance between the phase states on the trellis is $2\pi/q$.

The purpose behind Multi-h is to increase the constraint length of the code, thereby decreasing the probability of error, while keeping the bandwidth small, or reducing bandwidth by choosing appropriate values of h. Since Multi-h is a type of CPM (continuous phase modulation), then it has constant envelope as well as improved detection efficiency due to intrinsic memory [14]. Many schemes of multi-h modulation have been investigated [1-20]. A summary of much of the current work can be found in [15]. High coding gain codes consist of 3 or 4 h's with q values of 13 to 16 (giving rise to large trellises due to the many possible phase states). The highest achieved coding gain is 3.66 dB [15], with 99.0% bandwidth of 0.85. Bandwidth efficient codes consist of 2 to 4 h values with q values 8 to 16. The coding gains here are reduced to 2.19 dB with 99.0% bandwidth of 0.59 [15]. The summary tables and figures are reproduced in Figure 1, along with a comparison of some of the best known multi-h CPFSK schemes [17]. The schemes for the multi-h are given below:

Table 1. Best known multi-h CPFSK schemes [17]

Scheme	h_1	h_2	d^2_{min}	99% bandwidth
A	0.4	0.46	4.8	1.13
B	0.25	0.29	2.8	0.84
C	0.25	0.23	2.0	0.73
D	4/16	3/16	1.6	0.67

Fonseka and Davis [8], have done multi-h work using 2 values of h where the h's are repeated in a particular pattern like: (h_1 , h_1 , h_2) or (h_1 , h_1 , h_1 , h_2). Short rate 1/2 convolutional coding was also added to this patterned multi-h to obtain higher coding gain. The highest normalized squared minimum distance, d^2_{min} for $\upsilon = 1$ was 4.495 with h's of (0.2, 0.2,0.3) and for $\upsilon = 2$, $d^2_{min} = 5.151$ for h's of (0.25,0.25, 0.25, 4/9). (υ = k-1) . Only one simulation was run to find optimum memory length (trellis length) and to check performance with d^2_{min}. The simulation approached the d^2_{min} bound as E_b/N_o is increased [8]. Based on the summary of results in [15], the increased complexity of the multi-h schemes is not worth the small improvement in power and/or bandwidth. The Power Bandwidth tradeoff for m-ary multi-h linear phase schemes is taken from [10] and re-produced here in Figure 2.



Multi-*h* code	*K*	Coding gain [dB]	Bandwidth			Decision depth
			95.0%	99.0%	99.5%	
2/16 10/16 11/16 8/16	4	3.66	0.53	0.85	1.00	26-38
0/16 11/16 12/16	3	3.36	0.53	0.89	1.03	26-27
3/13 9/13 10/13	3	3.19	0.53	0.90	1.03	17-20
1/16 12/16 10/16 8/16	4	3.16	0.53	0.85	1.00	11-27
12/16 11/16 10/16 8/16	4	3.16	0.53	0.85	1.00	11-34
9/16 10/16 11/16	3	3.13	0.51	0.81	0.98	24-26
8/14 9/14 10/14	3	3.13	0.53	0.84	0.99	18-20
12/16 9/16 10/16 8/16	4	3.03	0.51	0.81	0.98	28
9/14 10/14 11/14	3	2.96	0.54	0.91	1.04	17-26
12/16 10/16 9/16 8/16	4	2.91	0.51	0.81	0.98	29-30

A

Multi-*h* code	*K*	Coding gain [dB]	Bandwidth			Decision depth
			95.0%	99.0%	99.5%	
7/16 8/16	2	2.19	0.45	0.59	0.68	21
5/16 7/16 8/16	3	2.10	0.43	0.59	0.66	20
6/14 7/14	2	2.07	0.44	0.59	0.68	19
5/14 6/14 7/14	3	1.96	0.43	0.58	0.65	16-18
4/10 5/10	2	1.72	0.44	0.59	0.66	11
5/15 7/15	2	1.67	0.43	0.58	0.65	20
6/16 4/16 7/16 8/16	4	1.65	0.40	0.56	0.64	7-20
5/15 6/15 7/15	3	1.53	0.40	0.56	0.64	20
4/12 5/12 6/12	3	1.47	0.41	0.58	0.65	13
3/8 4/8	2	1.45	0.43	0.59	0.66	8

B

Multi-*h* code	*K*	Coding gain [dB]	Bandwidth			Decisi dept
			95.0%	99.0%	99.5%	
3/15 4/15 5/15	3	-2.40	0.29	0.46	0.54	11
3/14 4/14	2	-2.67	0.26	0.46	0.54	8
3/16 4/16 5/16	3	-2.92	0.26	0.46	0.53	11
3/16 5/16	2	-3.03	0.28	0.46	0.55	11
3/13 4/13	2	-2.08	0.29	0.48	0.56	8
3/15 5/15	2	-2.50	0.29	0.48	0.56	11
4/16 5/16	2	-1.57	0.30	0.49	0.56	12
2/11 3/11 4/11	3	-2.48	0.30	0.49	0.56	9-10
2/9 3/9	2	-2.54	0.30	0.49	0.58	6
4/15 5/15	2	-1.08	0.31	0.50	0.58	12

C

Figure 1 Summary of Multi-h results [15,17]

Table A: Multi-h phase codes with high coding gain

Table B: Multi-h phase codes with high coding gain and bandwidth efficiency

Table C: Bandwidth efficient multi-h codes

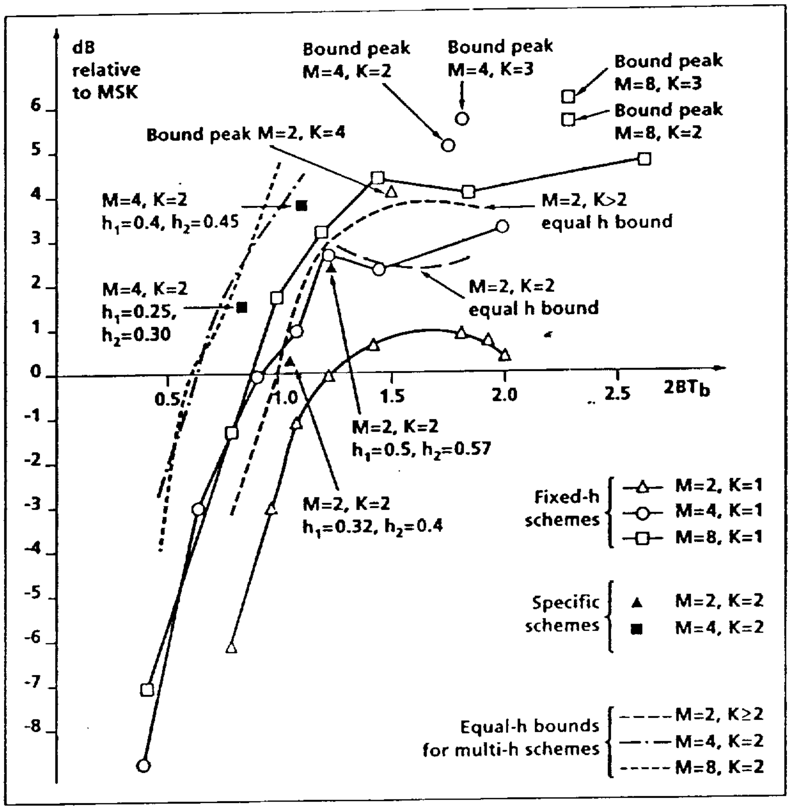


Figure 2 Power Bandwidth tradeoff for M-ary multi-h, linear phase, 1REC schemes [15]

Much work has also been done using convolution and continuous phase modulation. Here many single values of h have been used to find the optimal values of h, the optimal mapping schemes and the optimal codes for different rate codes, for various values of υ [2,5,7,10,11,12,14,16,17,18]. In a summarizing paper, Anderson and Sundberg [3], show that gains can be made in the constant envelope continuous phase modulation by using particular h values, coding schemes and raised cosine phase functions. The results are reproduced in Figure 3. To achieve gains, one needs to increase the complexity or have a tradeoff between the power and the bandwidth. Table 2 [17], shows several different coding schemes.

Table 2. Power, Bandwidth and Complexity for several Coding Schemes[18]

system	h	υ	E_b/N_o for $P_e=10^{-6}$	bandwidth	complexity
R=4/5 32 CPFSK	2/15	4	5.0 dB	1.25	240
R=3/4 16 CPFSK	2/15	4	6.0 dB	0.95	240
R=2/3 8 PSK		4	7.0 dB	0.65	16
R=2/3 8 CPFSK	1/10	4	9.5 dB	0.65	320
4-PSK			10.5 dB	0.65	1
MSK			10.5 dB	1.18	1

It can be seen that as the power is decreased, the bandwidth is increased. Still, the gains over QPSK (4 PSK) may not be worth the effort since the gains are minimal. Gains of up to 6.5 dB (with h=1.38) can be achieved using the raised cosine phase, but the bandwidth is more than doubled [6]. Further development of non-linear phase continuous phase modulation (CPM) by Fonseka and Mao [9] has found a non-linear coded CPM scheme with 4.5 dB improvement over MSK (minimum shift keying).

It is now clear that multi-h alone is not going to give the gains we had hoped for. Convolutional coding searches over various h's has also been completely examined for values of $\upsilon \leq 4$. Higher rate codes and higher υ values show promise for added gains, but also added complexity [17]. A summary of results for various rates, and υ are reproduced in Figure 4 [2,3]. Multi-h combined with convolutional coding may also give some gains over QPSK [17].

ORIGINAL PAGE IS
OF POOR QUALITY

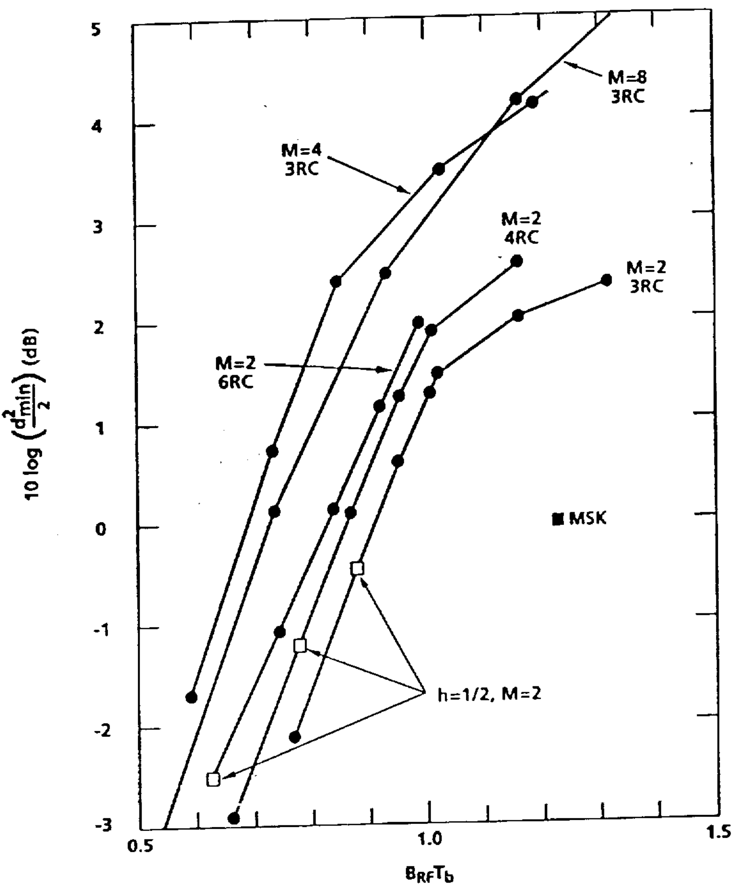
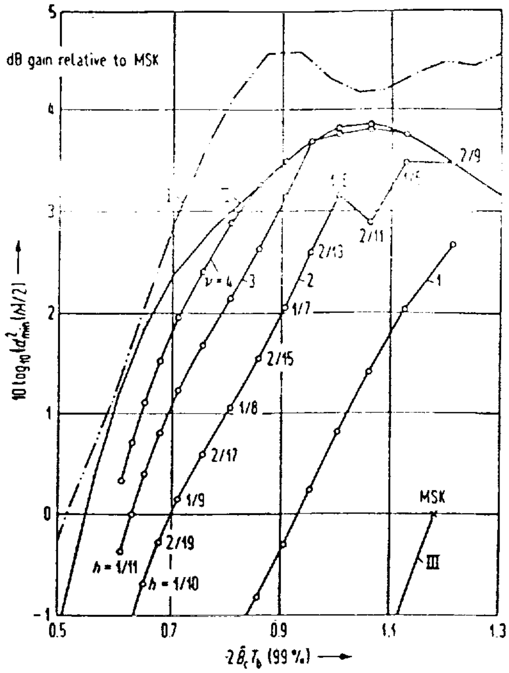


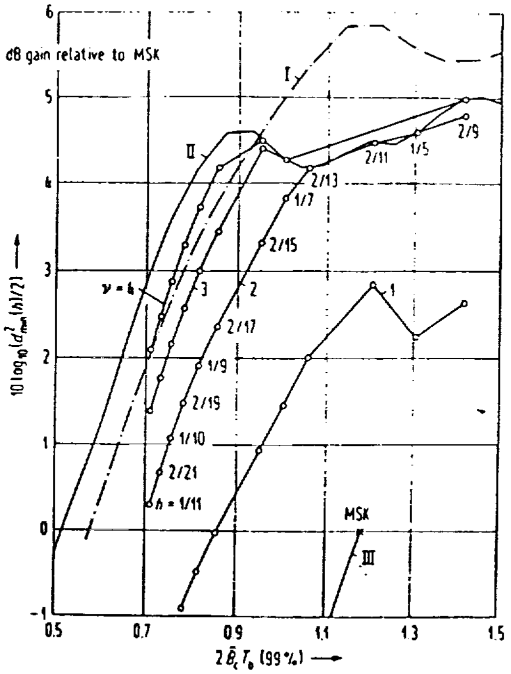
Figure 3 Minimum distance of RC trellis codes with 2,4, and 8 levels plotted against 99% bandwidth. Modulation index increases along each trajectory. Distance scale is relative to MSK [3]

Figure 11.28. Power-bandwidth trade-off for some of the coded eight-level CPFSK schemes considered in this chapter. ν = 1, 2, 3, and 4. The 99% power in band definition of bandwidth is used. I is the upper bound for coded 16-level CPFSK schemes, II is the upper bound for coded 8-level CPFSK, and III is binary CPFSK.

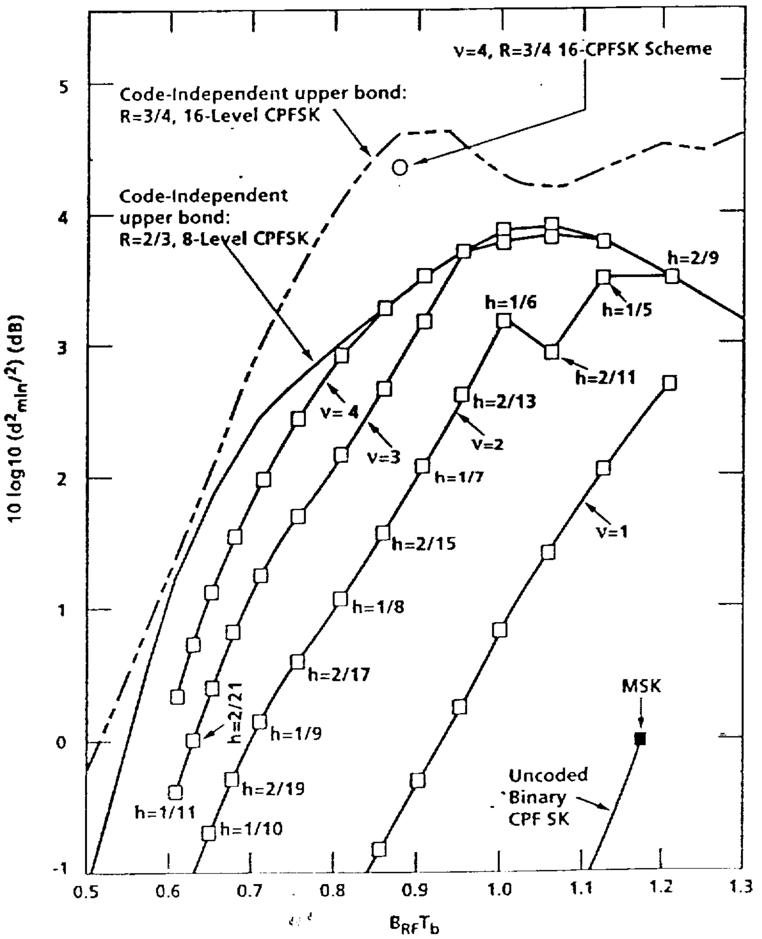


(a)

Figure 11.29. Power-bandwidth trade-off for some of the coded 16-level CPFSK schemes considered in this chapter. ν = 1, 2, 3, and 4. The 99% power in band definition of bandwidth is used. I is the upper bound for the coded 32-level CPFSK, II is the upper bound for coded 16-level CPFSK, and III is binary CPFSK.



(b)



(c)

Figure 4. Power bandwidth tradeoffs for some of the coded (a) 8-level (b) 16-level CPFSK schemes [2], and (c) for some convolutional/-CPFSK concatenated codes [3]

ORIGINAL PAGE IS
OF POOR QUALITY

PROGRESS TO DATE:

A multi-h generator has been modeled and tested using the simulation laboratory facilities at New Mexico State University. Also using the simulation laboratory, a correlation receiver has been modeled and tested. The multi-h metric and trellis have been modeled and tested using two h values of 1/2, and 3/4. The results have been consistent with the literature. No advantage was seen in continuing this particular form of multi-h, so work was directed toward combining convolutional encoding and multi-h. It is hoped that by expanding the trellis to allow for the different convolutional states, that an increase in the minimum distance will lead to greater coding gains. Work on a combined convolutional multi-h encoder/decoder has been completed. The simulated encoder/decoder can handle rate 1/g codes for values of g ranging from 1 to 3, and various h values, allowing up to four different values of h to be used. The decoder generates the decoding trellis associated with each of the different h's and g values, allowing calculation of the minimum distance for each of the different set-ups.

The performance of the multi-h decoder has been simulated and the probability of symbol error vs signal to noise has been obtained. A combination of convolutional encoding and continuous phase has been simulated in an attempt to increase the minimum distance between paths to reduce the probability of error. A rate 1/2 convolutional encoder has been simulated along with continuous phase modulation using a constant h of 1/4. This simulation has shown that the performance of the CPM scheme with mod index of 1/4 achieves that of MSK (uncoded, h=1/2) when the above convolutional encoder is incorporated. The set up of the decoder is similar to that of the multi-h with minor changes to accommodate the convolutional encoder. The basic communication block diagram for convolutional multi-h is given in figure 5.

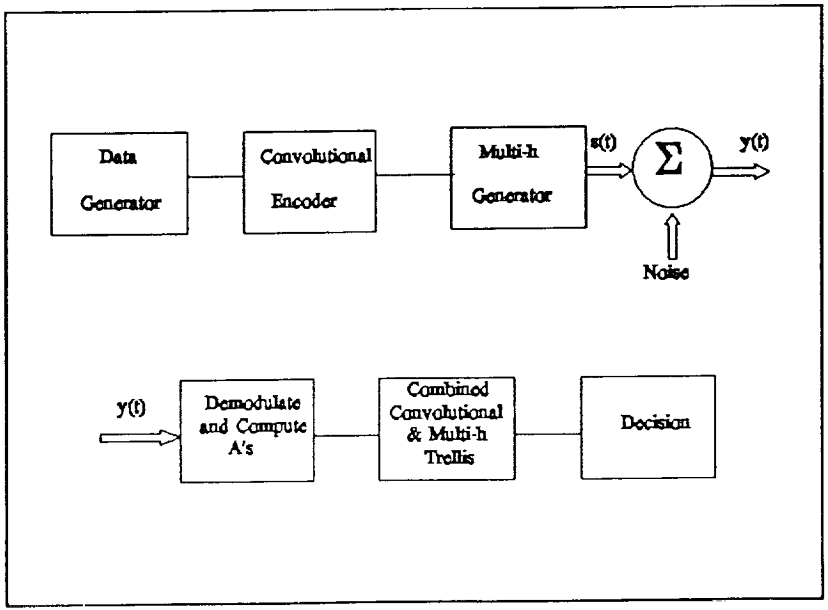


Figure 5 - Convolutional Multi-h Communications System

The transmitted signal, S(t) is given by:

$$S(t) = \sqrt{\frac{2E_s}{T_s}} \cos[2\pi f_c t + \pi \alpha_i h_i (\frac{t}{T_s} - (i-1)) + \Phi_i]$$

$$(i-1)T_s \le t \le iT_s$$

where α_i is the convolutionally encoded data, h_i is the multi-h parameter (the modulation index), f_c is the carrier frequency in Hertz, T_s is the length of time of the signalling interval (the duration of the signal), i is the current time interval beginning with i=0 at starting time t=0, and ϕ_i is the excess phase defined previously. The rectangular pulse shaping function [t - (i-1)] / T over the time interval (i-1) T_s $\le t \le$ i T_s is used to give a smooth transition between the different phase states.

The α_i are obtained from a k state convolutional encoder where k is number of memory cells including the current input. Figure 6 is an example of a convolutional encoder with k=3. The current input is stored in location i. The connections of the g functions with the memory sends the data in that file (0 or 1) into the binary adder. The actual connections determine what type of convolutional code is used. Each g function will be a zero or a one. When combined as g_1 g_2 ... g_n and converted from binary to a data code, this will give the convolutionally encoded data, α_i . A common code is given in Table 3. In Figure 6, the output g_1 takes the input in location i and i-2 and adds them modulo 2. The output g_2 adds the data in memory locations i and i-1 in binary. The transmitted signal is then corrupted by Additive White Gaussian Noise. The received signal is passed through a system of matched filters as seen in Figure 7 to produce expansion coefficients. There will be n = 2*2^ (ng) matched filters and coefficients for each data interval, where ng is the number of outputs from the convolutional encoder (the number of g functions).

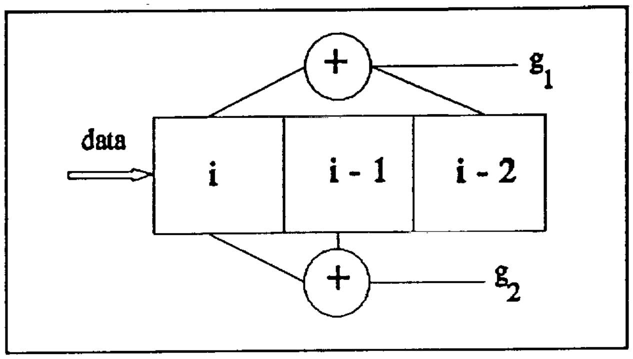


Figure 6 - Rate-1/2 Convolutional Encoder

Table 3. Output Code

g_1	g_2	α_i
0	0	-3
0	1	-1
1	0	1
1	1	3

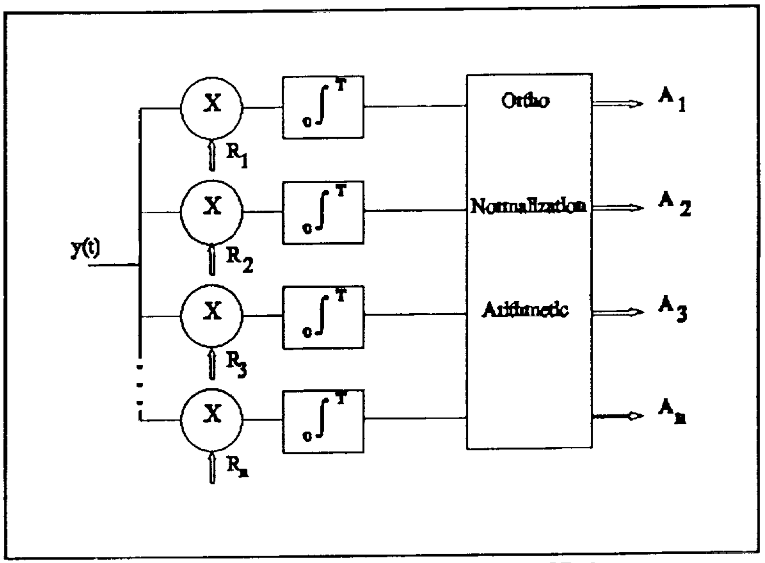


Figure 7 - Generation of Expansion Coefficients

A typical set of expansion functions is given as follows:

$$R_{j,i} = \sqrt{\frac{2}{T_s}} \cos(2\pi f_c t + \pi \alpha_i h_i)$$

$$R_{j+1,i} = \sqrt{\frac{2}{T_s}} \sin(2\pi f_c t + \pi \alpha_i h_i)$$

where j = 1,3, . . ., $2*2^{ng}$ - 1, and α = ±1, ±3, . . . , etc to cover all possible outputs from the convolutional encoder.

A trellis configuration can then be used to determine possible phases and convolutional states to find the correlations between the received signal and all possible signals. An example trellis for two modulation indices, h=(½, ¾) is given in Figure 8.

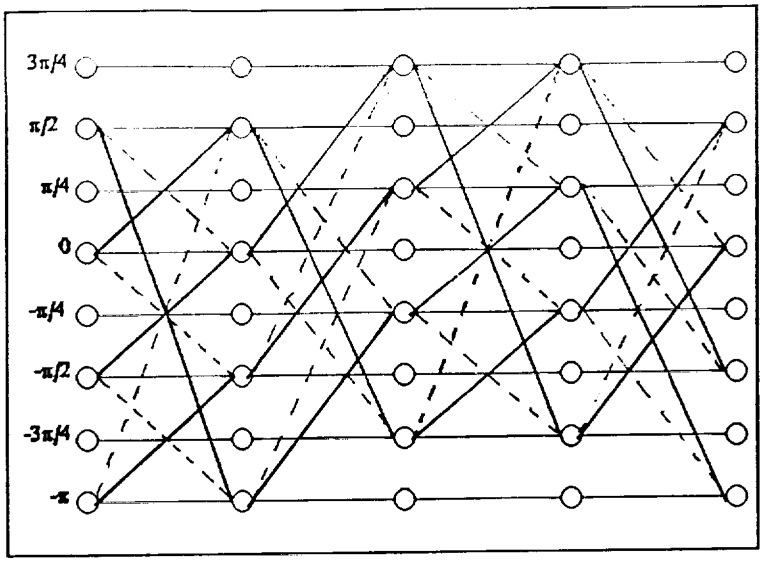


Figure 8 - Multi-h Trellis
h= (1/2, 3/4), +1 is the solid line, -1 is the dotted line

When convolution is added to the trellis, then each phase is split into convolutional states as well. For the encoder from Figure 8, and the example code given in Table 3, the input/output table for all possible inputs is given in Table 4. Since the ith digit is the current data point, Table 4 can be rewritten to give a next state form, given in Table 5.

Table 4. Encoder Output

i	i-1	i-2	g_1	g_2	α_i
0	0	0	0	0	-3
0	0	1	1	0	1
0	1	0	0	1	-1
0	1	1	1	1	3
1	0	0	1	1	3
1	0	1	0	1	-1
1	1	0	1	0	1
1	1	1	0	0	-3

Table 5. Next State Table

state number	current state		next state given input = 0			next state given input = 1		
	i-1	i-2	i-1	i-2	state #	i-1	i-2	state #
1	0	0	0	0	1	1	0	3
2	0	1	0	0	1	1	0	3
3	1	0	0	1	2	1	1	4
4	1	1	0	1	2	1	1	4

Using this next state table, a convolutional trellis can be formed as in Figure 9. The convolutional trellis and the multi-h trellis can now be combined into a convolutional multi-h trellis, where each phase has a sub-trellis consisting of the convolutional trellis. An example trellis can be seen in Figure 10.

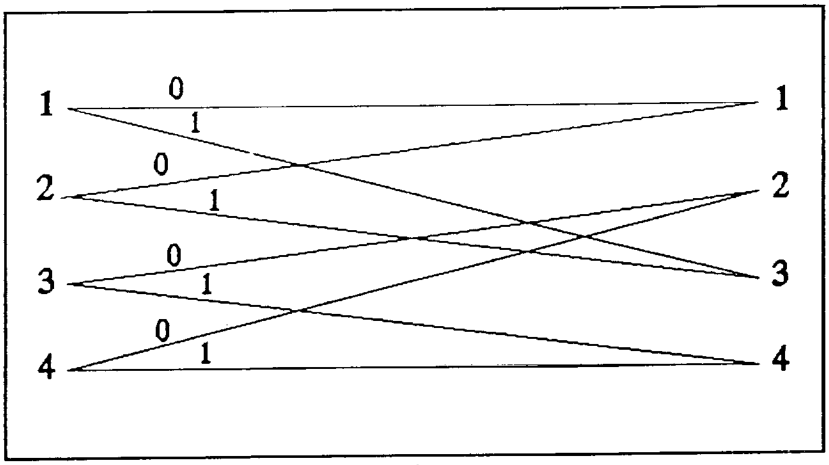


Figure 9 - Convolutional Trellis

Phase Value	Convolution State	Trellis			
φ_1	1	○	○	○	○
	2	○	○	○	○
	3	○	○	○	○
	·				
	·				
	nconv	○	○	○	○
φ_2	1	○	○	○	○
	2	○	○	○	○
	3	○	○	○	○
	·				
	·				
	nconv	○	○	○	○
	·	·	·	·	·
	·	·	·	·	·
	·	·	·	·	·
φ_{np}	1	○	○	○	○
	2	○	○	○	○
	3	○	○	○	○
	·				
	·				
	nconv	○	○	○	○

Figure 10 Combined Convolution and Multi-h Trellis

A metric (correlation between the received signal and all signals which could possibly be received during the time interval) is calculated for each possible transition. The metric used is given by:

$$\beta_{ij} = \int_{(i-1)T_s}^{iT_s} Y(t)\ S_{ij}(t)\ dt$$

where i is the ith data point and j is over all possible received inputs, all possible phase states and all possible convolutional states for the ith data point. The $S_{ij}(t)$ are all possible received signals. At each merge of the trellis, the path that gives the largest accumulated correlation (metric), is selected as the path to continue on through the trellis. The associated data point is also stored and carried along through the trellis. When the end of the trellis is reached, the path with the largest correlation is selected as the correct one and the output is selected as being the binary digit at that end of the path.

Simulation Program:

A BASIC program was written to accept several user parameters. The needed parameters are: desired length of the trellis, the number of h's (modulation indices) to use, the number of outputs from the convolutional encoder, the individual values of the modulation index, the individual values of the encoder outputs (input in decimal), the number of data points to be simulated, the common denominator of the modulation indices, and the desired signal to noise ratio in dB to be simulated. The program then produces the convolutional mapping, and the output mapping for the run. Two trellis' are used to keep track of the data and allow for shifting of data when paths are selected. The metrics are then calculated for each data point and the trellis followed to make a final decoding decision. Four different test runs have been made. The four run parameters are summarized in Table 6.

Table 6. Simulation Runs

Name	#h's	h	#g's	g	k
MSK	1	1/2	1	1.0	1
multi-h	2	1/2, 3/4	1	1.0	1
convolution	1	1/4	2	5.0,2.0	3
convolutional multi-h	3	0.2,0.2,0.3	2	1.0,2.0	2

The program is designed to be flexible to test different parameters like trellis length, signal to noise ratio, as well as the various different types of codes. The program will handle up to 4 values of h, 3 values of g functions, and up to a k value of 7. Trellis lengths of up to 100 are acceptable, and can increase with runs on the Fortran version on the Vax Work-station.

Simulation Results

The parameters in Table 6 were run varying trellis length, number of data points, and signal to noise ratio. Figure 11 is a plot of the results of several simulations holding the trellis length for each type of run a constant for that run, and varying the signal to noise ratio and number of data points. The theoretical curve for MSK is included for comparison. Table 7 shows the numerical results of each of the simulations. These results appear to be consistent and within error bounds set in other works, [8,15].

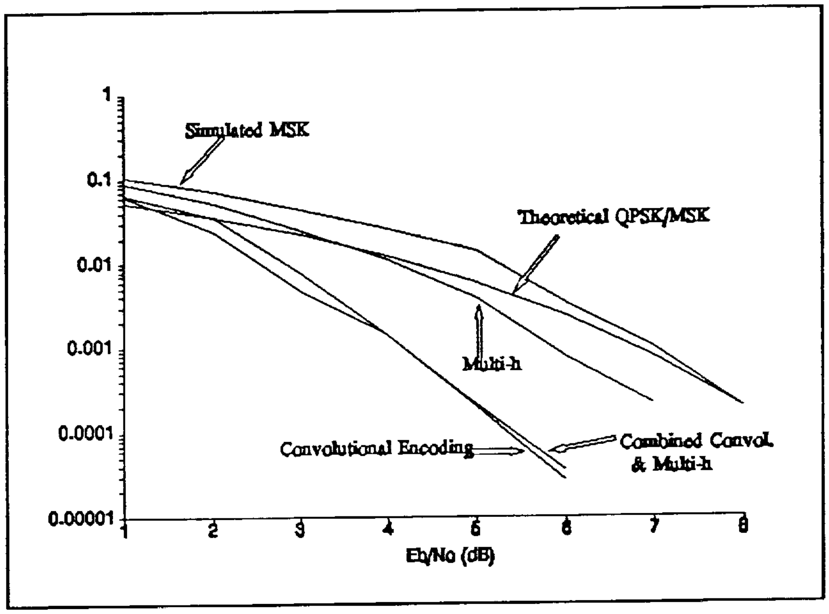


Figure 11 - BER Test Results

Table 7. Simulation Results

Simulation	E_b/N_o	Prob. of error	number of data points	trellis depth
Combined Convolution and Multi-h	1.0	0.0804	5000	20
	1.0	0.0784	5000	30
	1.0	0.086	5000	40
	1.0	0.0649	10000	20
	2.0	0.0358	10000	20
	3.0	0.0079	10000	20
	4.0	0.0014	10000	20
	5.0	0.000215	200000	20
	6.0	0.000036	450000	20
MSK	1.0	0.1074	10000	6
	1.0	0.1048	10000	10
	1.0	0.1062	10000	20
	2.0	0.0718	10000	6
	3.0	0.0438	10000	6
	4.0	0.0258	10000	6
	5.0	0.0142	10000	6
	6.0	0.0034	10000	6
	7.0	0.001	20000	6
	8.0	0.00019	200000	6
Multi-h	1.0	0.08774	100000	15
	2.0	0.05206	100000	15
	3.0	0.02523	100000	15
	4.0	0.01109	100000	15
	5.0	0.00393	100000	15
	6.0	0.00079	100000	15
	7.0	0.000215	200000	15

Table 7. Simulation Results

Simulation	E_b/N_o	Prob. of error	number of data points	trellis depth
Convolution	1.0	0.0626	10000	30
	2.0	0.0245	10000	30
	3.0	0.0048	10000	30
	4.0	0.00144	100000	30
	5.0	0.0002	100000	30
	6.0	0.000027	900000	30

Conclusions:

A computer program has been developed to perform simulations of convolutionally encoded multi-h signals which is used to generate symbol error rates for various codes and will aid in determining which codes will be applicable for communication systems. A new type of trellis has been developed that combines the convolutional states into the multi-h phase trellis as a combined decoding algorithm. Other research [8], has shown that convolutional coding applied to the multi-h modulation has a better power bandwidth tradeoff than uncoded multi-h. Most research has been focused on finding the minimum distance for various codes and modulation indices [see 15], with few simulations. This program allows the flexibility needed to run many simulations. This program will allow vigorous modeling of various communications links using rational modulation indices (up to 4 indices), with various convolutional codes (rate 1/2 to rate 1/3; with $k \leq 7$) to obtain useful combinations that will reduce bandwidth.

Increasing the rate of the code and/or increasing the υ value gives better gains [2,3] in power and bandwidth. The lower the h value used, the smaller the bandwidth.

Planned work:

More codes will be simulated using this program in an effort to find good codes that approach the theoretical lower error bounds. In particular, work will concentrate on larger k values than has been done in the literature. Codes using $\upsilon \leq 4$ ($k \leq 5$) have been used. The case of $k=7$ will be investigated. Rate 1/3 codes have been shown to give no worthwhile gains [2], therefore only rate 1/2 codes will be investigated. Further work could include higher rate codes if the current work indicates promise of good gains.

References:

1. Anderson, J. B., Simulated Error Performance of Multi-h Phase codes, IEEE Transactions on Information Theory, Vol IT-27, #3, May 1981, pages 357-362

2. Anderson, J. B., Aulin, T., Sundberg, C-E, Digital Phase Modulation, Chapter 11, Appendix D, Plenum Press, NY, 1986

3. Anderson, J. B., Sundberg, C-E, Advances in Constant Envelope Coded Modulation, IEEE Communications Magazine, Dec. 1991, pages 36-45

4. Anderson, J. B., Sundberg, C-E, Aulin, T., Rydbeck, N., Power-Bandwidth Performance of Smoothed Phase Modulation Codes, IEEE Transactions on Communications, Vol Com-29, #3, March 1981, pages 187-195

5. Aulin, T., Lindell, G., Sundberg, C-E, Minimum Euclidean Distance for Short Convolutional Codes, GLOBECOM'82, Conference Record, pages 1101-1105

6. Aulin, T., Rydbeck, N., Sundberg, C-E, Continuous Phase Modulation - Part II: Partial Response Signalling, IEEE Transactions on Communications, Vol Com-29, #3, March 1981, pages 210-225

7. Aulin, T., Sundberg, C-E, Continuous Phase Modulation - Part I: Full Response Signaling, IEEE Transactions on Communications, Vol Com-29, #3, March 1981, pages 196-209

8. Fonseka, John P., and Davis, George R., Combined Coded/ Multi-h CPFSK Signaling, IEEE Transactions on Communications, Vol 38, #10, October 1990, pages 1708-1715

9. Fonseka, J.P., Mao, T.R., Coded Nonlinear Continuous Phase Modulation, IPCCC, 1992, pages 218-223

10. Lindell, G., Sundberg C-E, Multilevel Continuous Phase Modulation with High Rate Convolutional Codes, GLOBECOM'83, Conference Record, San Diego, pages 1021-1026.

11. Lindell, G., Sundberg, C-E, Power and Bandwidth Efficiency of Coded CPM-A Comparison to Coded QAM, Proceedings of the 1987 International Conference on Communications Tech., Nov 1987, Nanjing China, pages 229-231.

12. Lindell, G., Sundberg, C-E., Aulin, T., Minimum Euclidean Distance for combinations of short Rate 1/2 Convolutional Codes and CPFSK Modulation, IEEE Transactions on Information Theory, vol IT-30, #3, May 1984, pages 509-519.

13. Mazur, B. A., Taylor, D. P., Demodulation and Carrier Synchronization of Multi-h Phase Codes, IEEE Transactions on Communications, Vol Com-29, #3, March 1981, pages 257-266.

14. Pizzi, Steven, and Wilson, Stephen, Convolutional Coding Combined with Continuous Phase Modulation, IEEE Transactions on Communications, Vol. 33, #1, January 1985, pages 20-29

15. Sasase, Iwao, and Mori, Shinsaku, Multi-h Phase-Coded Modulation, IEEE Communications Magazine, Vol 29, #12, December 1991, pages 46-56

16. Sundberg, C-E, Combined Convolutional Channel Coding and Constant Amplitude Continuous Phase Modulation, Stochastische Modelle und Methoden in der Informationstechnik, Vol 12, bis 14, April 1989, Nuremburg Germany.

17. Sundberg, C-E, Combined Channel Coding and Constant Amplitude Continuous Phase Modulation, Lecture Notes in Control and Information Sciences, Vol 128, Springer-Verlag, 1989, pages 137-176.

18. Wilson, S.G.,Bandwidth-Efficient Modulation and Coding: A Survey of Recent Results, IEEE International Conference on Communication, Vol 2, June 1986, Toronto Canada, pages 965-969.

19. Wilson, S.G., Gaus, R. C., Power Spectra of Multi-h Phase Codes, IEEE Transactions on Communications, vol Com-29, #3, March 1981, pages 250-256.

20. R. E. Ziemer and R. L. Peterson, Digital Communications and Spread Spectrum Systems, Macmillan, 1985, pg. 228-250.