Communications Research Centre

A SIMULATION OF FREQUENCY DEHOPPED BINARY AND 4-ARY NCFSK SIGNALS

by

R.J. Keightley

This work was sponsored by the Department of National Defence, Research and Development Branch, under Project No. 32A56. The author is with NDHQ/DCEM and performed the work while seconded to the MSAT Technical Project Office.

CRC TECHNICAL NOTE NO. 722
OTTAWA, APRIL 1984

Government of Canada
Department of Communications

Gouvernement du Canada
Ministère des Communications

Canada

COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS
CANADA

A SIMULATION OF FREQUENCY DEHOPPED BINARY AND 4-ARY NCFSK SIGNALS

by
R.J. Keightley

CRC TECHNICAL NOTE NO. 722

April 1984
OTTAWA

This work was sponsored by the Department of National Defence,
Research and Development Branch, under Project No. 32A56.
The author is with NDHQ/DCEM and performed the work
while seconded to the MSAT Technical Project Office.

TABLE OF CONTENTS

A SIMULATION OF FREQUENCY DEHOPPED BINARY AND 4-ARY NCFSK SIGNALS

by

R.J. Keightley

ABSTRACT

This report describes the simulation of frequency dehopped binary and 4-ary non-coherent frequency shift keyed signalling (NCFSK) over a Gaussian channel. Orthogonal FSK signals and a hopping rate of one hop per FSK symbol were simulated. Simulations were performed with and without forward error correction. Rate 1/3, constraint length 8 and rate 1/2, constraint length 7 convolutional codes with hard decision Viterbi decoding were used. A family of curves showing the probability of bit error versus E_b/N_o was produced. Coding gains of about 2 dB at an error rate of 1 in 10^5 were observed for binary NCFSK and a coding gain of less than 1.0 dB was observed at the same error rate for 4-ary NCFSK. Some extensions of the work are suggested for future study.

1. INTRODUCTION

In recent years, there has been a trend towards an increase in electronic counter-counter-measures (ECCM) in military satellite communications (MILSATCOM). These measures often include the use of spread-spectrum modulation to help achieve a high measure of anti-jam (AJ) protection. The reader is referred to several introductory sources [1-6] if unfamiliar with spread-spectrum fundamentals. Frequency hopping is the preferred spread-spectrum technique when the information signal is to be spread over the wide bandwidths available at the extremely high frequency (EHF) band.

Satellite communication systems have traditionally been power limited rather than bandwidth limited and this situation is exacerbated in the EHF band because of the present state-of-the-art of device technology. In order to conserve power, it is highly desirable to minimize the energy-per-bit-to-noise-power density (E_b/N_o) required to support communication at a specified error rate. Unfortunately, most frequency-hopping systems are constrained to the use of non-coherent detection because the phase continuity of a signal is not preserved from one hop to the next. As a result, one cannot use the more efficient coherent modulation techniques, such as minimum shift keying (MSK) or quadrature phase shift keying (QPSK).

For this reason, a non-coherent technique, such as non-coherent frequency shift keying (NCFSK), is often used and was chosen as the baseline modulation of the MSAT EHF military payload [7].

The use of forward-error-correction (FEC) coding is one method of reducing the required E_b/N_0. However, non-coherent demodulation losses in the receiver restrict the achievable coding gain to more modest levels than would be the case in a similar system using coherent demodulation.

In the classical case of communication in the presence of additive white Gaussian noise (AWGN), the performance of a coded system can be maximized by the use of soft decision decoding. However, if the channel introduces non-stationary or non-white Gaussian noise or non-Gaussian noise, a different receiver is required. It has been shown in [8] that in the presence of such interference one should not use soft decisions unless one has some information about the interference. The reader is refered to [9], pp 371 - 380, for treatment of the non-stationary Gaussian noise case. In order to establish a performance baseline, the performance of systems using only hard decision decoding was investigated here.

A great deal of work, including [8-16], has appeared recently on the performance of coded and uncoded M-ary NCFSK in various interference environments. As part of the Department of National Defence (DND) recoverable program at Communications Research Centre in MILSATCOM, studies of modulation and coding schemes for frequency-hopped signals were undertaken. These studies included the error performance of dehopped M-ary NCFSK with imperfections in the dehopping process such as frequency offset and phase jitter [17], as well as the study of coding techniques for the interference channel [8].

It was considered desirable to create a capability for the simulation of modulation and coding techniques applicable to EHF MILSATCOM. This report presents the results of the first phase of the study, namely the simulation of frequency-dehopped binary and 4-ary NCFSK signals in the presence of Gaussian noise. A hopping rate of one hop per FSK symbol was simulated. Binary NCFSK was simulated, without coding as well as with rate 1/2, constraint length 7, and rate 1/3, constraint length 8, convolutional coding and Viterbi decoding. Four-ary NCFSK was simulated both without coding and with the rate 1/2 code. Binary phase shift keying (BPSK) was simulated for purposes of comparison with FSK. Standard Monte Carlo techniques were used in the simulation. The method is generally the same as that used by Heller and Jacobs [18], who studied BPSK, and by Pawula and Golden [10], who studied continuous-phase NCFSK. Although hard decisions are used in this work, conversion to soft decision decoding, in the conventional Gaussian noise case, would be straightforward. Simulation of the system in the presence of different types of jamming is the subject of on-going study.

In Section 2 of this report, the computer model and algorithms used in the simulation are described. The results are presented and discussed in Section 3. Recommendations for further work are included in Section 4, including proposed improvements to the existing simulation software to decrease the simulation time.

2. COMPUTER MODEL AND SIMULATION ALGORITHMS

2.1 NCFSK COMMUNICATION SYSTEM

A typical frequency-hopped NCFSK communication system may be viewed as shown in Fig. 2.1. Information bits are generated by some source, FEC encoded, FSK modulated and frequency hopped before transmission over some physical channel. The received signal is dehopped, demodulated, decoded and sent to some destination or data sink. Noise, modelled as Gaussian in this simulation, is added to the transmitted signal in the channel. The noise is the source of errors in the receiver, which strives to estimate the original information sequence from observation of the received signal.

The modulator, channel and demodulator are mathematically modelled as a digital channel, the input of which is a FEC coded bit stream and the output of which is corrupted by errors. Comparison of the output of the data source with the input to the data sink yields an error count from which the probability of error of the system may be calculated.

2.2 OVERVIEW OF THE SIMULATION

The aim of the simulation was to assess the coded performance of binary and 4-ary NCFSK modulation. The communication system was simulated on a DEC LSI 11/23 microprocessor. A family of curves showing the probability of bit error, P_e, versus E_b/N_0, the ratio of the energy-per-information-bit to the single-sided noise spectral density, was produced. The simulation flowchart is shown in Fig. 2.2. The simulation programs are organized into a main program, which handles the simulation input and output, and a family of subroutines that simulate the encoder, channel, modem and decoder. BPSK was similarly simulated.

The programs were first written in FORTRAN but in order to expedite the execution of the simulation, the convolutional encoder and Viterbi decoder subroutines were re-written in MACRO. This modification resulted in a decrease of about an order of magnitude in the overall simulation time.

2.3 PROGRAM INITIALIZATION

The main program is interactive and asks the user for certain initializing parameters including:

1. the range of E_b/N_0 over which the simulation is to be run;

2. the increment in E_b/N_0 between the points to be calculated;

3. specification of the modulation to be BPSK, binary NCFSK or 4-ary NCFSK;

4. whether or not FEC coding is to be used; and

5. choice of rate 1/2 or rate 1/3 coding with BPSK or binary NCFSK. If coding is selected with 4-ary NCFSK, a rate 1/2 code is used.

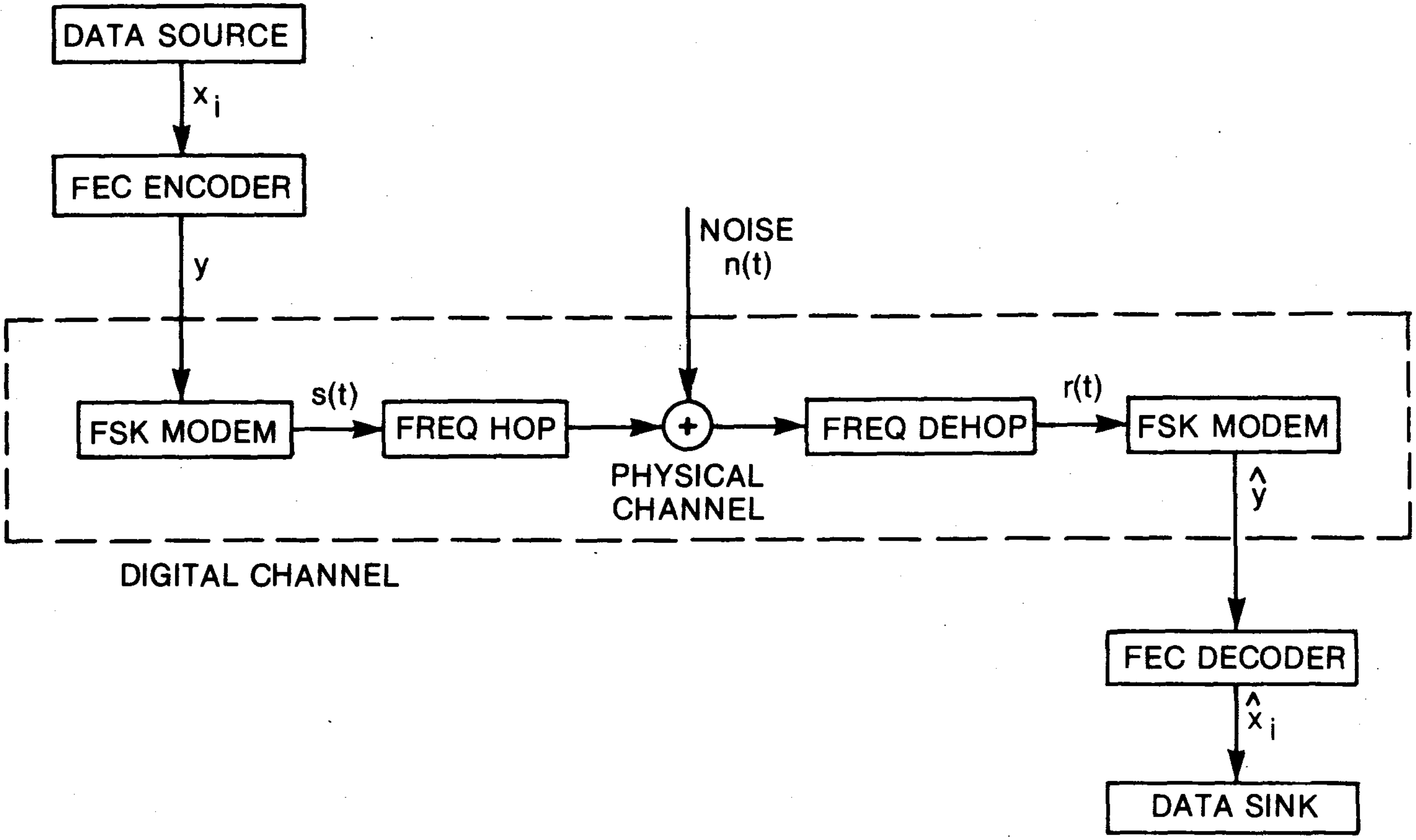


Fig. 2.1 A typical frequency-hopping FSK communications system.

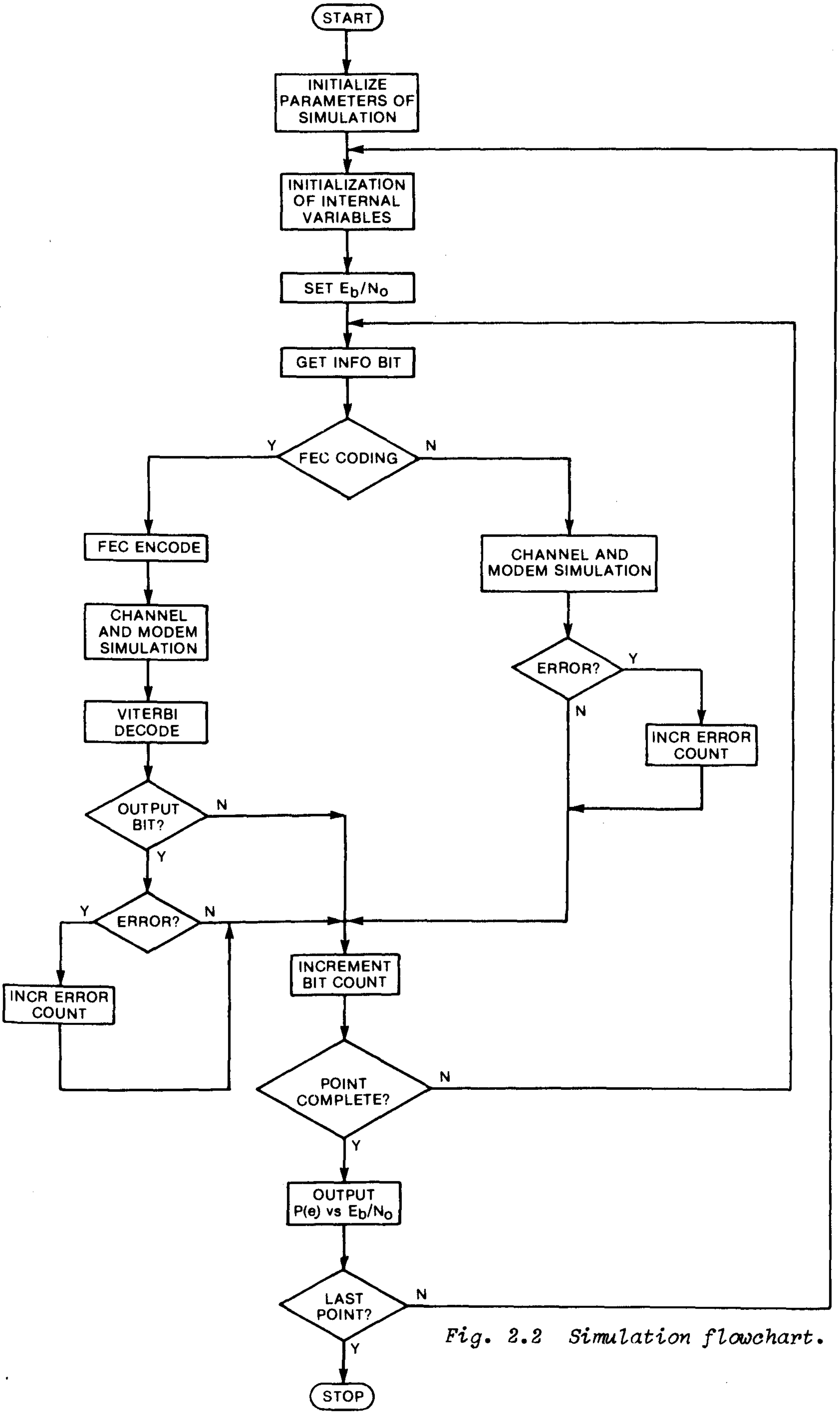


Fig. 2.2 Simulation flowchart.

The program then performs the initialization of internal variables as required. The information sequence, which was pre-computed, is then read from a disk file and stored in memory. A linear, maximal length, pseudo-random sequence (m-sequence) of length 1023 is used as the data sequence. It is generated from the irreducible, primitive polynomial [19]

$$G(x)= x^{10} + x^3 + 1. \tag{2.1}$$

A sequence length of 1023 was chosen because it is important to have the data appear random and to drive the encoder shift register through all possible states while conserving memory requirements. The sequence is repeated as necessary to supply as long an information sequence as required for the simulation.

2.4 FEC ENCODING

The simulation is capable of using either a rate 1/2, constraint length 7 or a rate 1/3, constraint length 8 convolutional code. The description of a convolutional encoder is illustrated with an example from [18].

Consider the rate 1/2, constraint length 3 encoder shown in Fig. 2.3. The contents of two highest order stages, that is the two oldest information bits, specify the state of the encoder; thus there are four possible states. The output of the encoder is completely specified by the state of the encoder and the most recent input bit. The code words, or sequences of code symbols, generated by the encoder for various input information bit sequences are shown in the code trellis of Fig. 2.4. The code trellis is just a state diagram for the encoder of Fig. 2.3. The four states are represented by points labelled on the left with the contents of the first two stages of the encoder. The lines or "branches" joining two stages indicate state transitions due to the input of single information bits. Dashed and solid lines correspond to "1" and "0" input information bits respectively. The encoder is in state 00 at time 0. If the first information bit were a "1", the output code symbols would be 11 and the encoder would go to state 01. The code symbols are shown adjacent to the trellis branches. As an example, the input data sequence 101... generates the code symbol sequence 111000....

The encoder shift register taps are often represented in a polynomial form. For the codes used in this simulation, the generating polynomials are 117.166 from [10] and 367.331.225 from [9] pp 402, where the polynomials are in octal notation. For example, 117 is expanded as 1001111 and is meant to represent the polynomial

$$P(x) = x^6 + x^3 + x^2 + x + 1.$$

Similarly 367 is expanded as 11110111 and represents the polynomial

$$Q(x) = x^7 + x^6 + x^5 + x^4 + x^2 + x + 1.$$

The encoder shift register feedback taps are shown in Figs. 2.5 and 2.6 respectively.

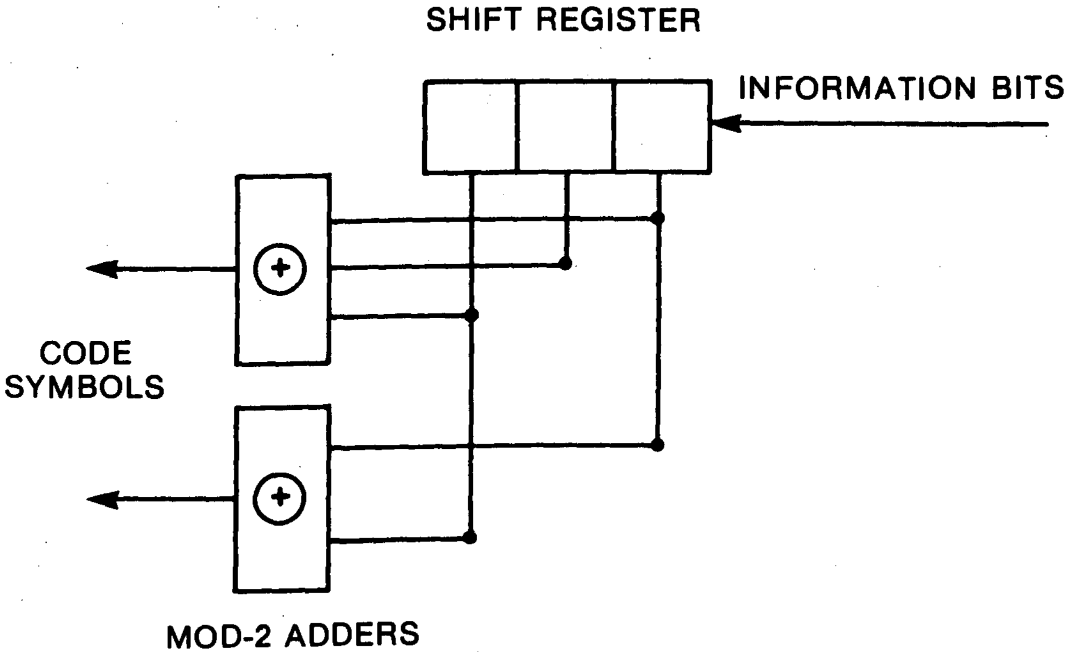


Fig. 2.3 Rate 1/2, constraint length 3 convolutional encoder.

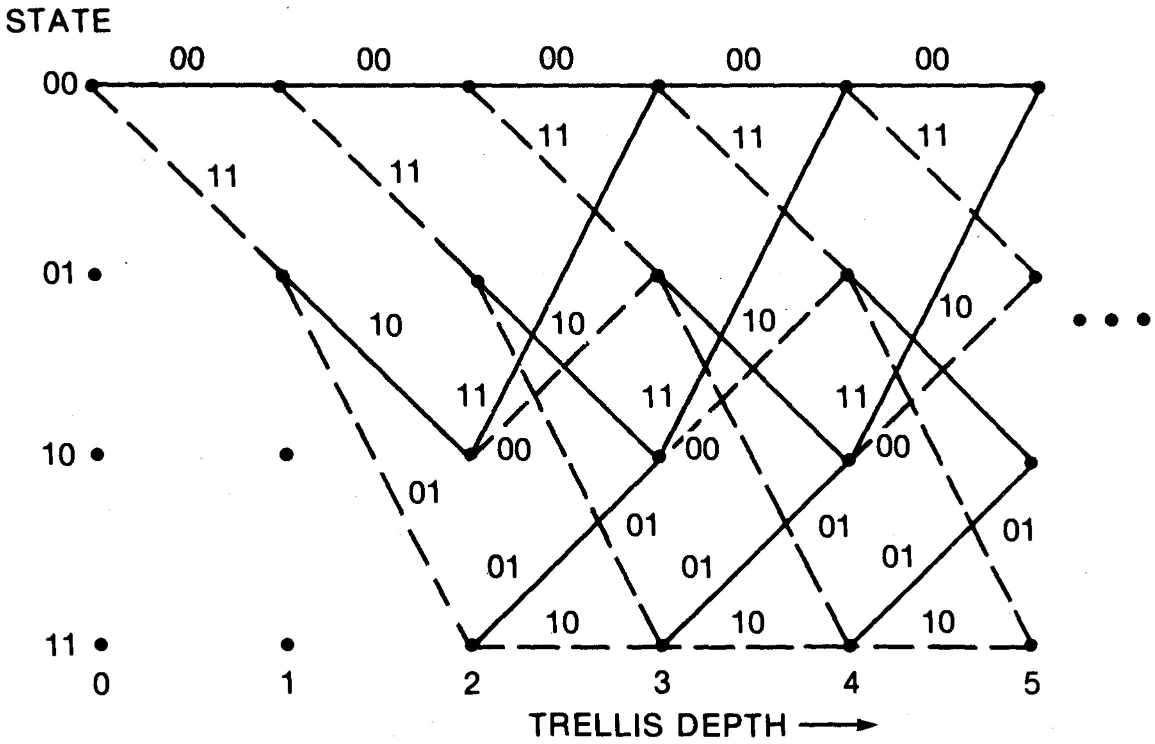


Fig. 2.4 Trellis diagram of the rate 1/2, constraint length 3 convolutional encoder.

The encoder is simulated by a MACRO subroutine in which the encoder shift register taps are represented by a word containing the generating polynomial. The code symbols are calculated by performing a logical AND operation between the encoder shift register and the appropriate code generator. The resultant word is used as an index into a look-up table which contains the modulo-2 summation of the bits in the word. The encoding procedure is shown in Fig. 2.7 for one of the code symbols of the rate 1/2 code.

In practice, the DEC LSI 11/23 does not have an AND instruction. However, the BIC A,B instruction performs the operation {(NOT A) AND B}. Therefore, the complement of the code generator is used with the BIC instruction, as shown in Fig. 2.8, to yield the equivalent result.

2.5 MODULATION

In conventional applications, continuous-phase NCFSK modulation is usually preferable to non-continuous-phase NCFSK since continuous phase NCFSK does not produce the phase discontinuities in the transmitted signal that tend to generate spectral impurities. Another advantage of continuous phase NCFSK is that multi-symbol demodulation, which exploits the phase continuity of the signal from symbol to symbol, can improve the receiver performance [10,20]. However, the frequency-hopping synthesizer at the transmitter of a frequency-hopping system may not be able to generate continuous-phase FSK at the required hopping rate. In any case, multi-symbol demodulation at the receiver may not be possible in a frequency hopping system since phase continuity from hop to hop is not generally preserved. For these reasons, non-continuous-phase NCFSK modulation is simulated.

The following assumptions are made:

1. The hopping rate is one hop per FSK symbol;
2. The hop timing is perfectly synchronized to the symbol timing; and
3. Orthogonal signalling is used, implying that

$$h = 2\, f_d\, T \log_2 M = .5,\ 1.0,\ 1.5,\ \ldots \qquad (2.2)$$

where h is the modulation index or deviation ratio, f_d is the frequency deviation, T is the information bit period and M indicates the number of FSK tones used. The FSK symbol period is $T \log_2 M$.

For binary NCFSK, the possible signals are

$$s_0(t) = \cos\{2\pi(f_c - f_d)t + \phi\} \qquad (2.3)$$

$$s_1(t) = \cos\{2\pi(f_c + f_d)t + \phi\} \qquad (2.4)$$

where ϕ is a random variable uniformly distributed on the interval $(0,2\pi)$.

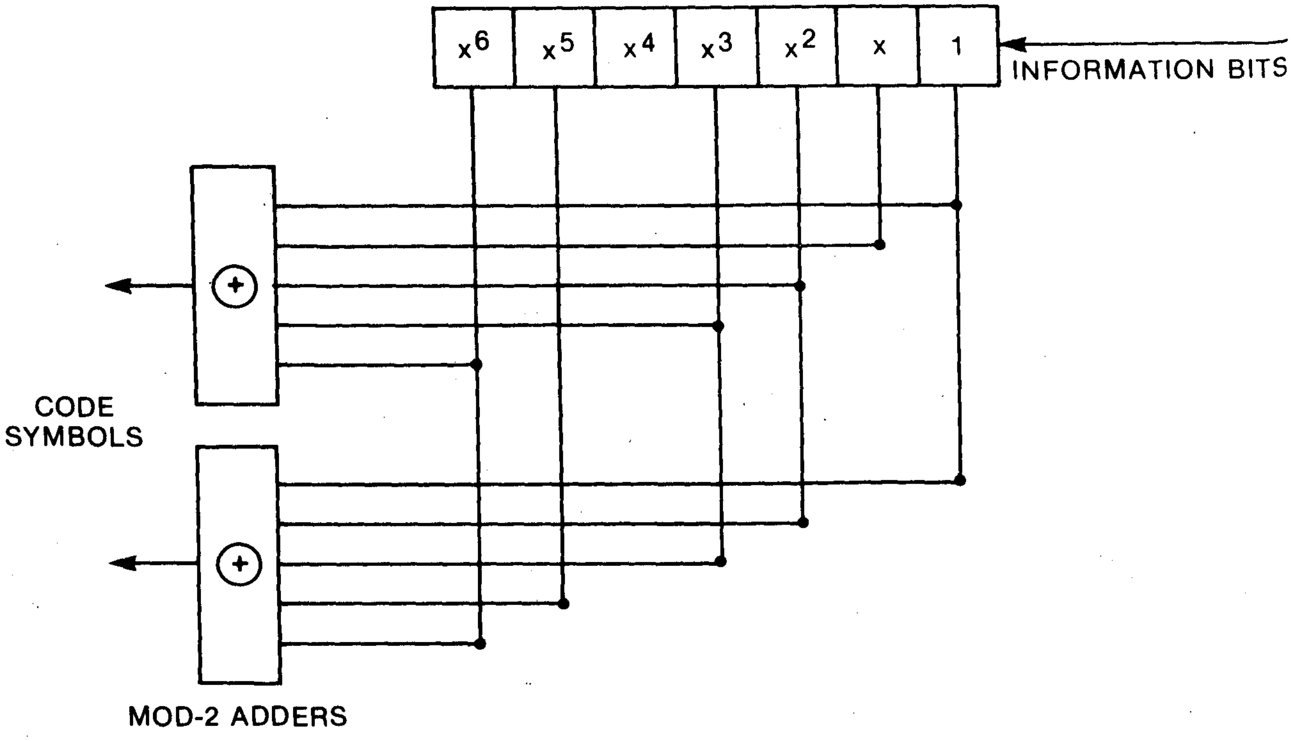


Fig. 2.5 Rate 1/2, constraint length 7 convolutional encoder.

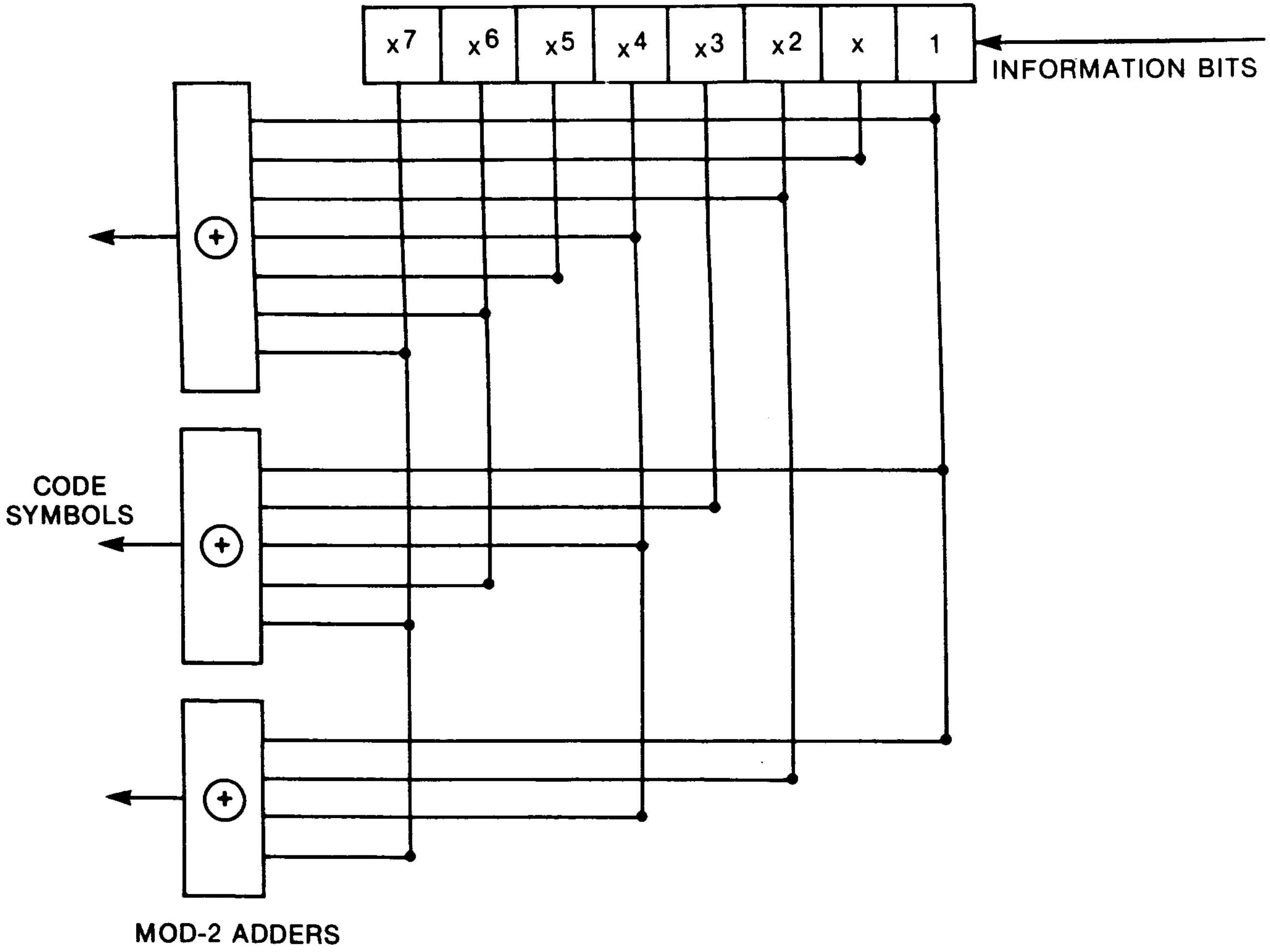


Fig. 2.6 Rate 1/3, constraint length 8 convolutional encoder.

A =	1	0	0	1	1	1	1	GENERATING POLYNOMIAL
B =	1	0	1	1	0	0	0	ENCODER SHIFT REGISTER
AUB =	1	0	0	1	0	0	0	OFFSET INTO LOOK-UP TABLE

CODE SYMBOL = CONTENTS OF ADDRESS $\left\{\text{TABLE} + 1001000\right\}$

Fig. 2.7 Code symbol conceptual calculation.

$\bar{A}$ =	0	1	1	0	0	0	0	COMPLEMENT OF GENERATING POLYNOMIAL
B =	1	0	1	1	0	0	0	ENCODER SHIFT REGISTER
BIC ($\bar{A}$,B)=($\overline{\bar{A}}$) UB =	1	0	0	1	0	0	0	OFFSET INTO LOOK-UP TABLE

Fig. 2.8 Code symbol implemented calculation.

For 4-ary NCFSK, the possible signals are

$$s_{00}(t) = \cos\{2\pi(f_c-3f_d)t + \phi\} \tag{2.5}$$

$$s_{01}(t) = \cos\{2\pi(f_c-f_d)t + \phi\} \tag{2.6}$$

$$s_{11}(t) = \cos\{2\pi(f_c+f_d)t + \phi\} \tag{2.7}$$

$$s_{10}(t) = \cos\{2\pi(f_c+3f_d)t + \phi\} \tag{2.8}$$

It is shown in Appendix A that the binary NCFSK signals (2.3) and (2.4) are indeed orthogonal for $h = 0.5, 1.0, 1.5$ ..... The extension of the proof to 4-ary NCFSK is straightforward.

2.6 CHANNEL MODEL

In this work, the channel is modelled as introducing Gaussian noise. For binary NCFSK, the input to the demodulator is

$$r(t) = A\cos\{2\pi(f_c-f_d)t + \Theta\} + n(t) \tag{2.9}$$

if a "0" were sent and

$$r(t) = A\cos\{2\pi(f_c+f_d)t + \Theta\} + n(t) \tag{2.10}$$

if a "1" were sent. Here

1. A is the amplitude of the received signal;

2. Θ is the phase angle of the received signal and is assumed to be uniformly distributed on the interval $(0,2\pi)$; and

3. $n(t)$ is zero mean, white Gaussian noise with double sided spectral density of $N_0/2$ watts/Hz.

2.7 DEMODULATION

The optimum receiver for M-ary non-coherent detection is well known [21] and is presented in Fig. 2.9 for the binary case. It is shown in Appendix B that the normalized output of the demodulator may be expressed as

$$L_n = i\,[(\sqrt{2E_s/N_0}\cos\theta + v_1)^2 + (\sqrt{2E_s/N_0}\sin\theta + v_2)^2 - (v_3{}^2 + v_4{}^2)] \tag{2.11}$$

where:

1. the actual output L is normalized via;

$$L_n = L/\sqrt{N_0E_s/2}$$

2. i is +1 if the transmitted symbol were a "0", or is -1 if the transmitted symbol were a "1";

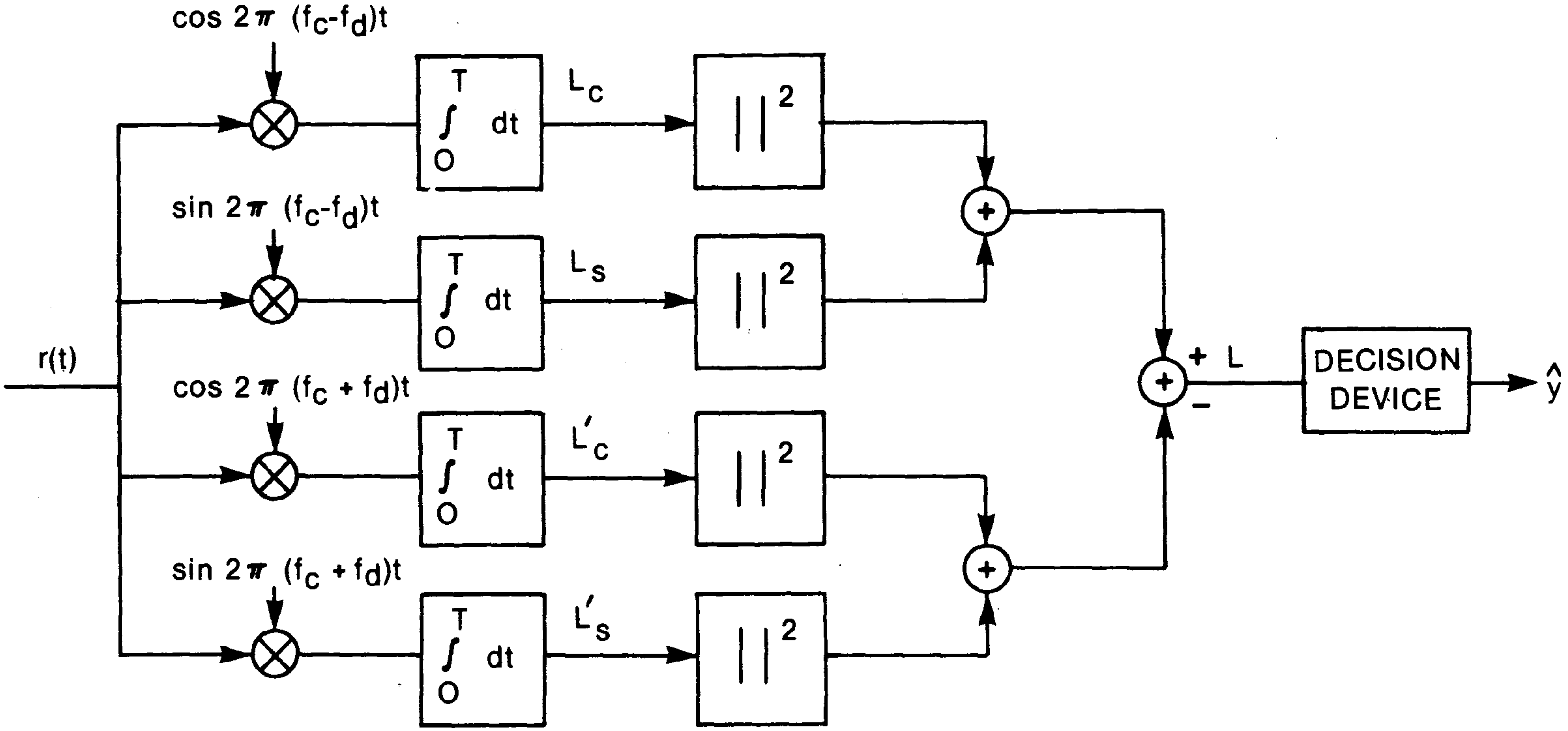


Fig. 2.9 Optimum receiver for binary NCFSK.

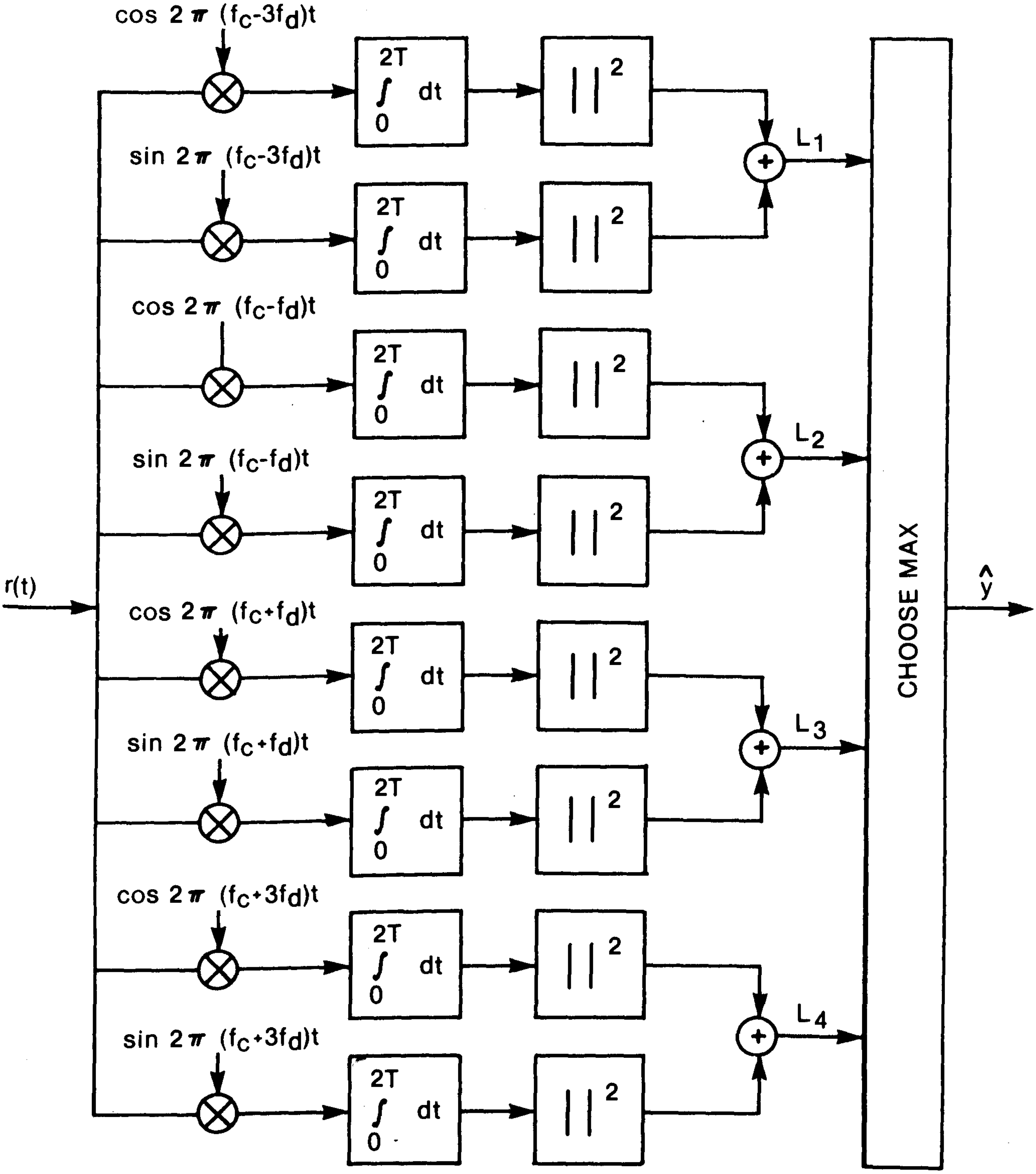


Fig. 2.10 Optimum receiver for 4-ary NCFSK.

3. v_1, v_2, v_3, and v_4 are all zero mean, unit variance, Gaussian random variables; and

4. $E_s = kE_b$, where $k = \log_2 M$.

The phase angle, Θ, is computed for each received bit by standard random number generation techniques, which will be described in more detail later in the report. Four independent, zero mean, unit variance Gaussian random numbers are also computed. L_n is then calculated and a hard decision is made with a "0" being declared received if L_n is greater than zero and a "1" being declared if L_n is less than or equal to zero. The demodulated bit is passed to the Viterbi decoder for further processing.

The 4-ary NCFSK receiver is shown in Fig. 2.10. It should be noted that the integration is now performed over kT = 2T seconds, that is, each FSK symbol now transmits two information bits. The fact that the decision is now based on a higher symbol energy than in the binary case results in a significant improvement in error performance. With a similar analysis to the binary case, it can be shown that the normalized output of the i th summer, is

$$\begin{aligned} L_i &= (\sqrt{2E_s/N_0} \cos \Theta + w_1)^2 \\ &\quad + (\sqrt{2E_s/N_0} \sin \Theta + w_2)^2, \qquad \text{symbol present in channel } i \\ &= w_3{}^2 + w_4{}^2, \text{ symbol not present in channel } i \end{aligned} \tag{2.12}$$

where w_1, w_2, w_3 and w_4 are all zero mean, unit variance, Gaussian random variables.

The theoretical bit error performance of the binary NCFSK receiver can be shown to be [21]

$$P_e = 1/2 \exp \left(-E_s/2N_0\right) \tag{2.13}$$

The theoretical symbol error performance of the 4-ary NCFSK receiver can be shown to be [21]

$$P_e = \sum_{k=1}^{3} \frac{(-1)^{k+1}}{k+1} \genfrac{}{}{0pt}{}{3}{k} \exp \frac{-kE_s}{(k+1)N_0} \tag{2.14}$$

where

$$\binom{3}{k} = \frac{3!}{k!\,(3-k)!} \,.$$

2.8 DECODING

A thorough discussion of the Viterbi decoding algorithm can be found in many references including [9]. This section of the report will briefly describe the basic Viterbi algorithm and then will discuss the implementation of the algorithm used in the simulation. The description of the basic Viterbi algorithm is from [18].

For the code trellis diagram of Fig. 2.4, a brute force maximum likehood decoder would calculate the likehood of the received data for code symbol sequences on all paths through the trellis. The path with the largest likehood would then be selected and the information bits corresponding to that path would form the decoder output. Unfortunately, the number of paths for an N bit information sequence is 2^N. Therefore, this brute force decoding quickly becomes impractical as N increases.

With Viterbi decoding, it is possible to reduce greatly the effort required for maximum likehood decoding. It can be seen from Fig. 2.4 that after depth 3 (in general after a depth equal to the constraint length) each of the four states can be entered from either of two preceeding states. A Viterbi decoder calculates the likehood of each of the two paths entering a given state and eliminates from further consideration all but the most likely path that leads to that state. This is done for each of the $2^{(k-1)}$ states at a given trellis depth, where k is the encoder constraint length. After each decoding operation, only one path remains to each state. The decoder then proceeds one level deeper into the trellis and repeats the process. It is important to note that in eliminating the less likely paths entering each state, the Viterbi decoder will not reject any path that would have been selected by the brute force maximum likehood decoder.

The decoder must, at some point in time, output the decoded information sequence. As described thus far, however, it only maintains a set of $2^{(k-1)}$ paths after each decoding step. Each retained path is the most likely path to have entered a given encoder state. One way of selecting a single most likely path is to periodically force the encoder into a known state by inputting a pre-arranged k-1 bit information sequence. The decoder can then select the most likely path leading to the known state as its output.

Note that the Viterbi algorithm is a maximum likelihood sequence estimator (MLSE) and that the assignment of the metric, or measure of the probability of a given sequence, is based on knowledge of the probability density function of the noise. In practice, a non-optimum linear metric assignment is usually used that results in a negligible degradation from optimum performance [9]. When the noise is Gaussian, there is some advantage in using soft decision decoding. However, it has been shown that in the case of jamming, soft decision decoding should not be used without some knowledge of the jammer state, or statistics of the interfering signal [8]. For reasons previously given, this simulation uses hard decision decoding, even though the performance is studied over the Gaussian channel.

A trellis diagram, similar to Fig. 2.4, is a convenient means of showing the decoding algorithm. There are $2^{(k-1)}$ states to the trellis, where k is the number of stages in the encoder shift register and is called the constraint length. Fig. 2.11 shows one step in the trellis diagram for the rate 1/2, constraint length 7 code. There are 64 input states and 64 output states. Each state has a metric and path history associated with it. The metric is related to the probability of the path history being the transmitted data sequence. In this simulation, the path history of each state contained 32 bits (2 words of computer memory). The solid lines denote the hypothesis of a "0" being the next information bit while the dashed lines denote a "1".

The decoding algorithm is illustrated with the following example of the rate 1/2, constraint length 7 code. The input state metric, $MINP_{i,n}$, is defined to be the metric of the path history for input state i at the nth step in the trellis. The output state metric, $MOUT_{j,n}$, is similarly defined. $MINP_{i,n}$ is calculated using

$$\begin{aligned} MINP_{i,n} &= MOUT_{i,n-1} \quad ; \quad n > 1 \\ &= 0 \quad ; \quad n = 1 \end{aligned}$$

In hard decision decoding, the transition metric, $M_{i,j}$, is the Hamming distance between the demodulated code symbols and the symbols corresponding to the transition from input state i to output state j. Let the symbols corresponding to the transition from input state 0 to output state 0 be represented by S_1 and S_2 respectively. If the demodulated code symbols are C_1 and C_2 then

$$M_{0\,0} = D_1 + D_2$$

where

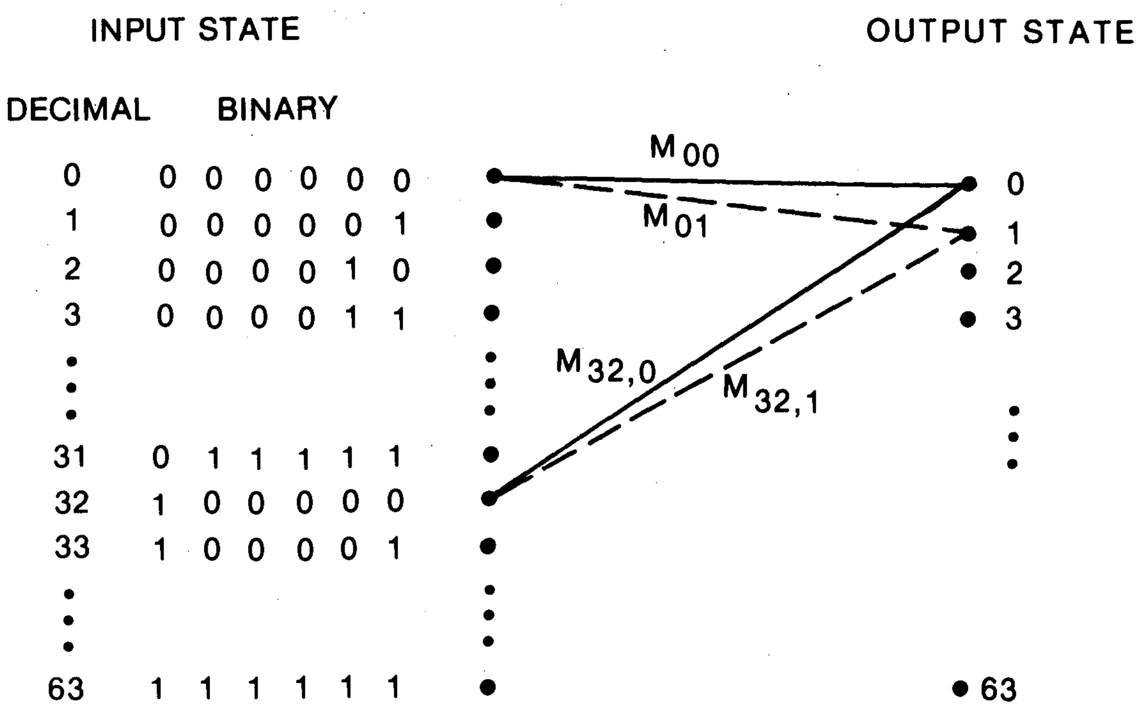
$$\begin{aligned} D_1 &= 0 \text{ ;for } S_1 = C_1 \\ &= 1 \text{ ;otherwise} \\ D_2 &= 0 \text{ ;for } S_2 = C_2 \\ &= 1 \text{ ,otherwise} \end{aligned}$$

The resultant metric, M_r, corresponding to this transition is the sum of the input state metric and the transition metric, $M_{0\,0}$.

$$M_r = MINP_{0,0} + M_{0,0}.$$

The procedure is repeated for the transition from input state 32 to output state 0. The path corresponding to the lower resultant metric is retained and is called the survivor. The other path is discarded. At this point, the surviving path memory is transfered to output state 0 and the surviving resultant metric becomes the new output state metric, $MOUT_{0,n}$. The metric and path memory for output state 1 is then similarly calculated using input states 0 and 1 and transitions corresponding to a transmitted "1".

This process is repeated for input states (1,33) and output states (2,3) and is continued for all the remaining states. When complete, all 64 output states will have updated metrics and path memories associated with them. The metrics are scaled to avoid any possible word overflow problems. The minimum metric is set to zero and the other metrics are adjusted accordingly. A maximum value of 100 is assigned to a metric if its value would otherwise exceed that amount. The limiting of the metric can be shown to have no effect on the performance of the algorithm [9]. The decoder now proceeds one step deeper into the trellis and repeats the process.



INPUT STATE

DECIMAL	BINARY
0	0 0 0 0 0 0
1	0 0 0 0 0 1
2	0 0 0 0 1 0
3	0 0 0 0 1 1
⋮	
31	0 1 1 1 1 1
32	1 0 0 0 0 0
33	1 0 0 0 0 1
⋮	
63	1 1 1 1 1 1

Fig. 2.11 One step of the decoding trellis for the constraint length 7 code.

Once 32 bits information bits have been decoded and stored in memory, the oldest bit in the path history of the output state having the lowest metric is output at each subsequent step through the trellis. Hence, 32 bits of the decoded sequence is stored in memory at any one time. This procedure avoids the undesireable requirement of periodically forcing the decoder into a known state and does not significantly affect the decoder performance [18]. When the simulation is finished, the final 32 bits are taken to be the 32 bit path history corresponding to the state with the minimum metric.

A similar decoding procedure is used with the rate 1/3 code, except that there are three code symbols for every information bit and the decoder has 128 states because of the increased code constraint length.

The decoding algorithm does not make use of the starting state of the encoder because it is desirable to use a receiver that can start decoding at any point in the transmission without any prior knowledge of the transmission.

2.9 RANDOM NUMBER GENERATION

Uniformly distributed random numbers X_1, $X_2,\ldots,X_n$, $X_{n+1},\ldots$ are generated on the interval (0,1) using the linear congruential method [22] with the recursion formula

$$Y_{n+1} = [(aY_n + 1) \text{ modulo } m] \qquad (2.16)$$

$$X_{n+1} = \frac{Y_{n+1}}{(m-1)} \qquad (2.17)$$

where

1. $m = 2^{31}$,
2. $a = 31{,}415{,}925$,
3. Y_n is the n^{th} element of the sequence Y_1, $Y_2 \ldots Y_n$, $Y_{n+1}, \ldots$
4. Y_1 is the seed and is input at run time.

Gaussian random numbers are generated using the polar method described by Knuth [22]. The algorithm calculates two independent, normally distributed random variables X_1 and X_2 as follows:

1. Two independent random variables, U_1, U_2, uniformly distributed between zero and one are generated.

2. Calculate

$$V_1 = 2U_1 - 1$$
$$V_2 = 2U_2 - 1.$$

3. Calculate

$$S = V_1{}^2 + V_2{}^2.$$

4. These steps are repeated until S is less than 1.

5. The desired random numbers are calculated using:

$$X_1 = V_1 \sqrt{(-2 \ln S)/S} \tag{2.18}$$

$$X_2 = V_2 \sqrt{(-2 \ln S)/S} \tag{2.19}$$

3.0 SIMULATION RESULTS

3.1 GENERAL

The aim of the simulation was to assess the performance of coded and uncoded binary and 4-ary NCFSK modulation over a Gaussian channel. The results are presented in the form of curves of the probability of error versus E_b/N_0. The data points are based on a minimum of 4096 simulated information bits. If at least 100 errors were not detected, then sufficient additional information bits were simulated to produce at least 100 errors for all but the last points on the curves. These last points were based on fewer errors because the length of time required to give 100 errors would have been excessive. In these cases, the probability of error was based on the number of errors observed in one million information bits. The curves were determined by performing a least-squares fit of a third order polynomial [23] to the points calculated by simulation.

3.2 SIMULATION RESULTS FOR BPSK

The performance curves for the BPSK simulation are shown in Fig. 3.1. The uncoded results agree well with theory. The coded performance corresponds closely to other simulation results [10,18]. The curve for the rate 1/2 encoded case almost exactly reproduces the corresponding curve of Fig. 7 of [18]. The BPSK results verify the correct operation of the code simulation.

The rate 1/3, constraint length 8 code gives a coding gain in excess of 3.5 dB at an error rate of 1 in 10^5. The cross-over point, the point at which there is no coding gain, occurs at an error probability of about .02. The rate 1/2, constraint length 7 code does not perform quite as well but still gives a significant improvement in system error performance. Note that with both codes there is always a coding gain over the range of all practical error rates.

3.3 SIMULATION RESULTS FOR BINARY NCFSK

The performance curves for binary NCFSK are presented in Fig. 3.2. Once again, the uncoded simulations agree well with theory and the coded results correspond to what one would expect from earlier work [10]. It is immediately apparent that there is significantly less coding gain with binary NCFSK than with BPSK. With binary NCFSK there is a coding gain of about 2 dB at an error rate of 1 in 10^5. As well, there is negligible difference in the performance of the rate 1/3, constraint length 8 code and

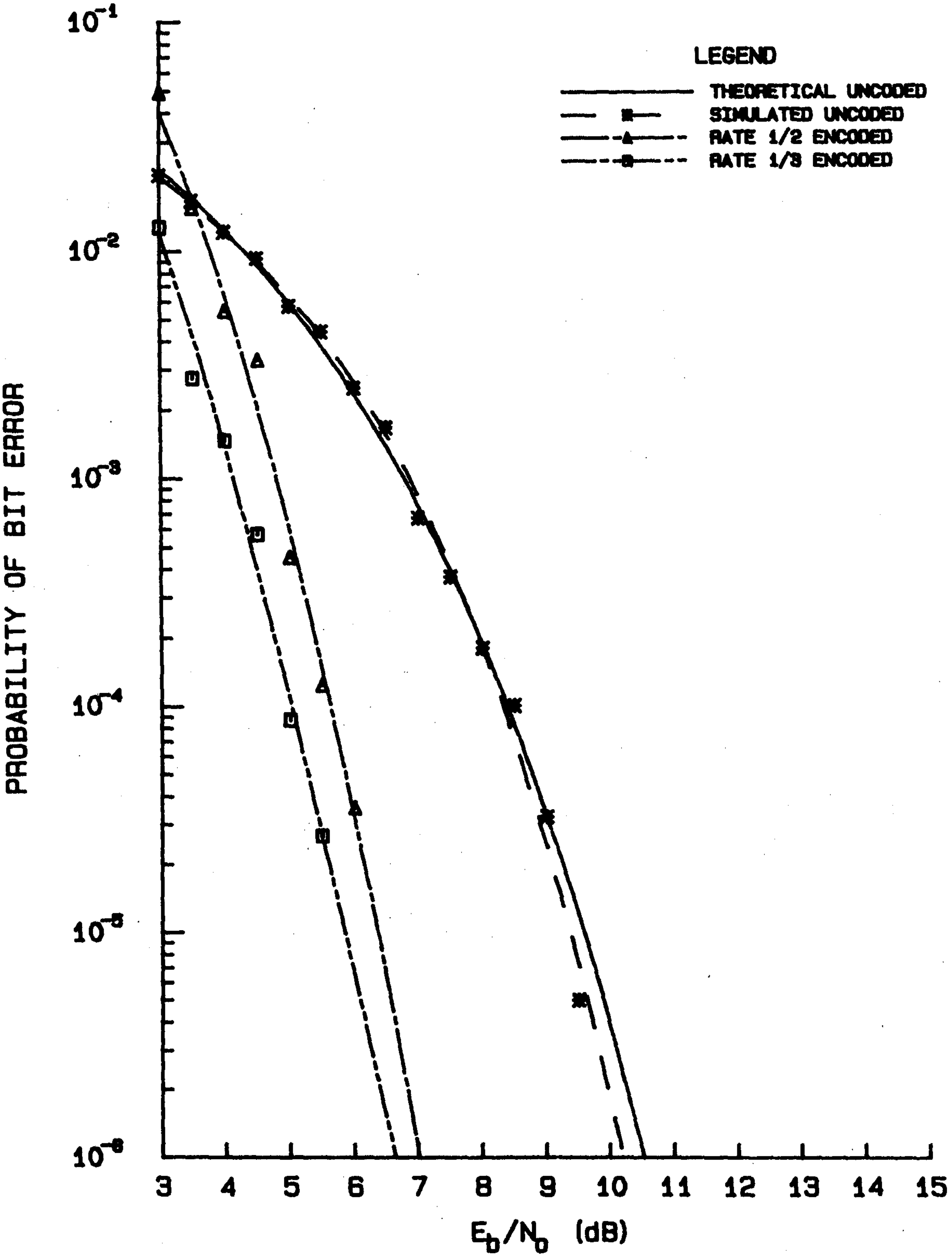


Fig. 3.1 Error performance curves for BPSK.

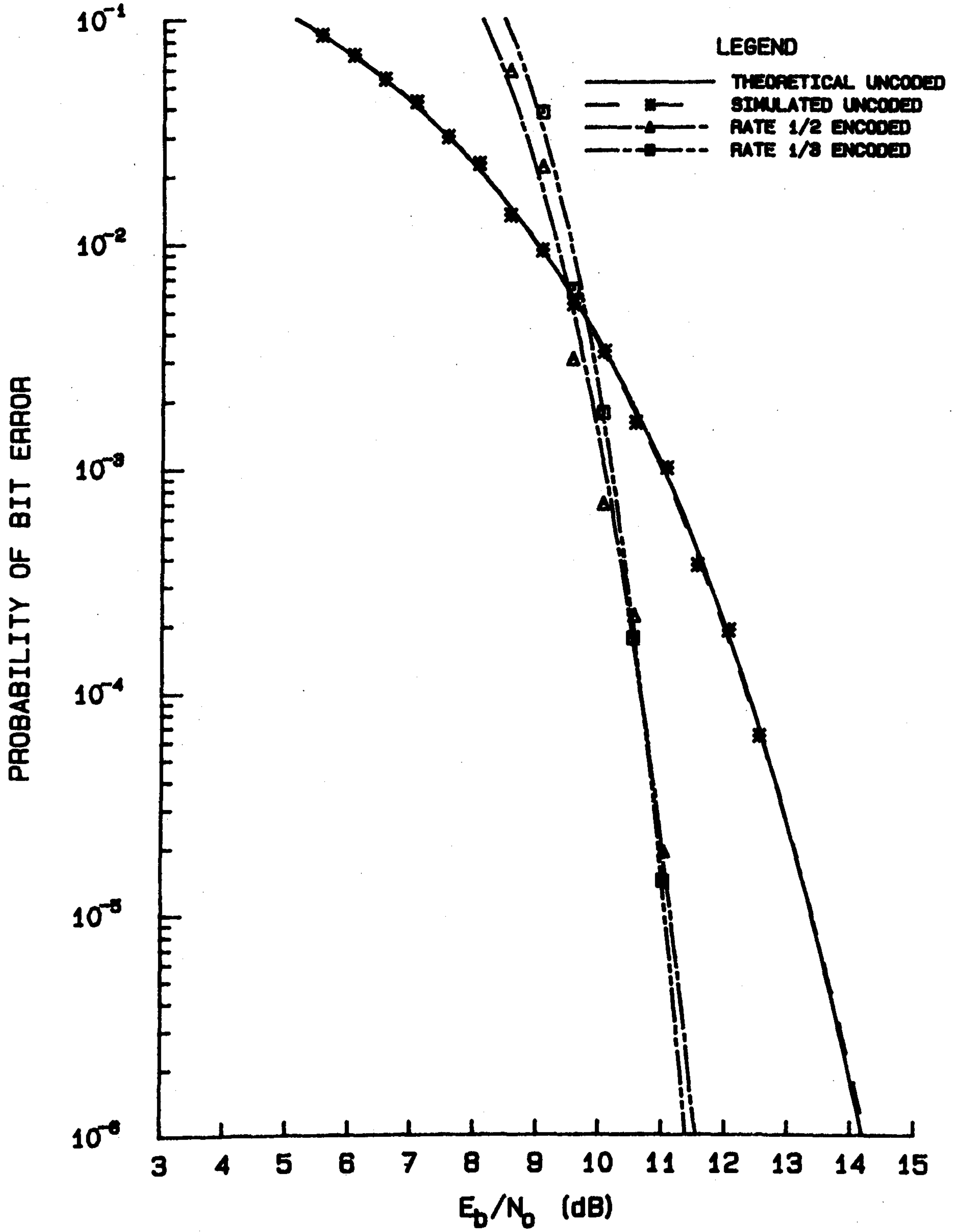


Fig. 3.2 Error performance curves for binary NCFSK.

the rate 1/2, constraint length 7 code. The cross-over point is at an error rate of about .006, a significant degradation from coded BPSK.

3.4 SIMULATION RESULTS FOR 4-ARY NCFSK

The performance curves for 4-ary NCFSK are given in Fig. 3.3. As before, there is close agreement with theory for the uncoded case and with what one would expect from previous results [10] for the coded case. The coding gain at an error rate of 1 in 10^5 is just under 1.0 dB and the cross-over point is at a bit error rate of about 6 in 10^4. Note that two of the curves shown in Figure 3.3 show symbol error rate while the other two curves show bit error rate.

From Fig. 3.3 it can be seen that the bit error performance is always superior to the symbol error performance. This fact is observed since there are always twice as many bits as symbols transmitted but a single symbol error does not always cause two bit errors. The simulation counted both symbol and bit errors and it was found that in all cases for 4-ary NCFSK,

$$P_{eb} \approx .66 \, P_{es}$$

where P_{eb} is the probability of bit error and P_{es} is the probability of symbol error. This result is in close agreement wih theory [24] that states for M-ary transmission

$$P_{eb} = \frac{2^{k-1}}{2^k-1} \, P_{es} \qquad (3.1)$$

where $k = \log_2 M$. For 4-ary transmission, $k = 2$ and

$$P_{eb} = \frac{2}{3} \, P_{es}. \qquad (3.2)$$

It is instructive to compare the binary with the 4-ary NCFSK results. It is observed that the performance of uncoded 4-ary NCFSK is everywhere about 2.5 dB superior to uncoded binary NCFSK. The performance gain is due to the increase in the integration time from one to two information bit periods and is at the expense of increased signal bandwidth and an increased receiver complexity. In a power limited system operating in the EHF band, there may be sufficient bandwidth available to make use of this gain.

3.5 DISCUSSION OF THE RESULTS

The results show that use of convolutional encoding and Viterbi decoding would be of some benefit in a binary or 4-ary NCFSK communication system employing orthogonal signalling and hard decision demodulation operating over a Gaussian channel.

Pawula and Golden [10] investigated the use of three-symbol demodulation with continuous phase NCFSK and Viterbi decoding. Their

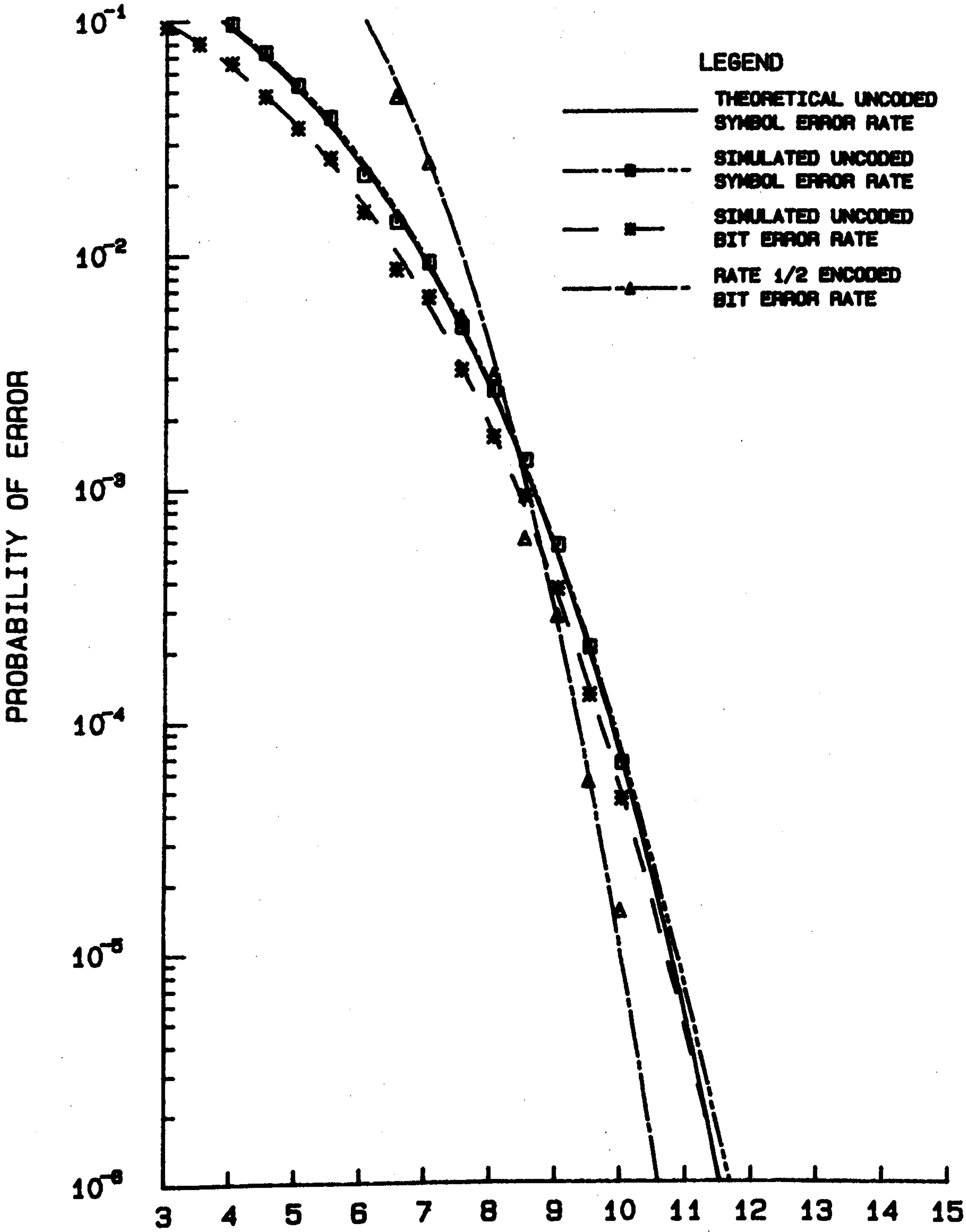


Fig. 3.3 Error performance curves for 4-ary NCFSK.

simulation results showed less than 1 dB of coding gain in the binary case. For the 4-ary case with rate 1/2 coding and modulation index of .8, a coding gain in excess of 2 dB was achieved at a bit error rate of 1 in 10^5. However, the coding gain dropped to approximately 1 dB when a modulation index of 1.0 (orthogonal signalling) was simulated. The reader should exercise caution in applying these results to a frequency-hopping system since it is not clear to what extent multi-symbol demodulation could be exploited in view of the phase discontinuity of the received signal from hop to hop.

Clark and Cain [9], pp 344, claim that Viterbi decoding with 8-ary NCFSK will give a coding gain of between 5.5 and 6.5 dB at an error rate of one in 10^5 relative to uncoded binary NCFSK on a Gaussian channel. However, it is also shown that 4.4 dB of that gain is due to the 8-ary NCFSK modulation and that the coding itself only gives between 1 and 2 dB of additional gain.

With the relatively modest coding gains discussed so far one might ask why one should bother with FEC coding on a frequency hopped system that is designed for many tens of dB of processing gain. On the Gaussian channel, it is true that coding does not significantly improve the performance of a single bit demodulation NCFSK system. However, Clark and Cain [9], pp 375 - 380, claim that a soft decision Viterbi decoding scheme, using knowledge of the variance of the interference can provide 35-40 dB coding gain at a bit error of one in 10^5 against a worst-case partial-band jammer. It is expected that even a much simpler coding scheme, such as simple time diversity will also yield impressive gains against such a jammer [16]. Gains in the order of 30 - 40 dB are far from trivial and one is motivated to implement such a system. Care must be taken in the system design, however, that one does not increase resistance to one type of jamming at the cost of increased vulnerability to other types. One must be fully aware of what one is trading for the increased jam resistance.

It follows from the law of large numbers that as the number of bits simulated increases to infinity, the estimate of the probability of error, P_e, will converge to its true value. Therefore, a measure of the accuracy of the simulation results may be obtained from a study of the variation of P_e as a function of the number of bits simulated. Table 3.1 shows this variation in P_e for uncoded BPSK with an E_b/N_0 of 5.0 dB. It can be seen from Table 3.1 that P_e rapidly converges to within 10% of the theoretical value. From Fig. 3.1, it can be seen that a ±10% variation in P_e about the simulation point results in less than a 0.25 dB variation in E_b/N_0. Since the simulation results are based on a minimum of 100 errors, the variation in P_e should be well within 10% of the true value. Therefore, the simulation results should be accurate to within .25 dB.

4.0 CONCLUSION

4.1 SUGGESTIONS FOR IMPROVEMENT OF PRESENT PROGRAMS

The simulation programs were initially written in FORTRAN but the convolutional encoder and Viterbi decoder subroutines were re-written in MACRO to decrease the simulation time. Since the decoding time increases

Number of simulated information bits	10	20	50	100	200	500	1000	2000	5000	10000	20000	50000	100000
Number of declared errors	1	1	1	1	2	3	6	9	28	54	104	290	603
Estimated probability of error	.1	.05	.02	.01	.01	.006	.006	.0045	.0056	.0054	.0052	.0058	.00603
Percentage variation from the theoretical value of $P_e \approx .00561$	1682.5	791.3	256.5	78.3	78.3	7.0	7.0	-19.8	-0.2	-3.7	-7.3	3.4	7.5

Table 3.1 Variation in the estimate of the probability of error as a function of the number of simulated information bits and the number of declared errors for uncoded BPSK with E_b/N_o of 5.0 dB.

exponentially as the time required to resolve one cell in the decoding trellis, it is important that the subroutine which does the cell calculations be as efficient as possible. The MACRO version of the decoder decreased the execution time by about a order of magnitude. However, there are other modifications which could be implemented to further reduce this time, including:

1. the pre-computation of the branch metrics for all decoder states and their storage in an appropriate table. This change would reduce the decoding to a look-up table procedure and would significantly decrease decoding time.

2. a reduction in the number of index-mode instructions by their substitution, where possible, with register-mode instructions.

3. the elimination of memory arrays for state metrics and path histories for both input and output states. This change would be facilitated by the look-up table decoding procedure described above.

4. a reduction in the length of path memory from 32 bits to 16. Heller and Jacobs [16] showed that only a modest penalty is paid for the reduction in path memory to 16 bits. The savings in memory by implementing the last two changes would at least partially offset the increase in memory by the change to table look-up procedures.

If the above modifications were implemented, the execution time would be governed by the simulation of the modem and channel. These subroutines would not easily be programmed in MACRO. Some enhancement may be possible in the selection and implementation of the random number generation algorithms but further improvements in speed could probably best be achieved by use of a faster computer, such as the VAX, or by a multi-processor approach to the simulation.

4.2 SOME RECOMMENDED FUTURE WORK

A major limitation to the present study is that the simulations were based on orthogonal signalling over a Gaussian channel. Without these assumptions, the simulation of the modulator, channel and demodulator would require a significant increase in processing time. It would be highly desirable to develop a separate simulator for these components which would be able to simulate a wider range of conditions. This effort would include the development of modem, codec and channel simulators that could be programmed for a range of modulation and coding techniques and for several interference channels.

It is recommended that the simulation be extended to include 8-ary and higher order NCFSK modulation with both rate 1/2 and rate 1/3 FEC coding. To attain better coding gain, some soft decision decoding strategy should be included to enable the receiver to distinguish between "good" and "bad" data. It is expected that even a sub-optimum soft decision algorithm could give significant gains in the presence of jamming.

Other recommended work includes:

1. an assessment of the applicability of multi-symbol demodulation of continuous phase NCFSK to frequency-hopped systems;
2. an extension of the simulation to include hopping rates faster than one hop per FSK symbol; and
3. an investigation of other coding techniques such as block coding.

4.3 SUMMARY

Frequency dehopped binary and 4-ary NCFSK, signals were simulated without FEC coding, using orthogonal signalling over a Gaussian channel. Orthogonal FSK signals were used. Simulations were performed with forward error correction (FEC) and without FEC. A hopping rate of one hop per FSK symbol was simulated. A rate 1/3, constraint length 8 and a rate 1/2, constraint length 7 convolutional code with hard decision Viterbi decoding were used. A family of curves showing the probability of bit error versus E_b/N_0 was produced. Coding gains of about 2 dB at an error rate of 1 in 10^5 were observed for binary NCFSK. A coding gain of less than 1.0 dB was observed at the same error rate for 4-ary NCFSK.

Further work is required to extend the results to include more complex modulation and coding techniques and to interference models other than the Gaussian model.

5.0 ACKNOWLEDGEMENTS

The author wishes to express his gratitude to Dr. J.L. Pearce and Dr. E.B. Felstead for their guidance and support throughout the project.

This work was performed for and under the sponsorship of DND, Chief of Research and Development, Task 32A56.

6.0 REFERENCES

[1] W.F. Utlaut, "*Spread spectrum principles and possible application to spectrum utilization*", IEEE Communications Magazine, Vol. 16, No. 5, pp 21-31, September 1978.

[2] C.E. Cook and H.S. Marsh, "*An introduction to spread spectrum*", IEEE Communications Magazine, Vol. 21, No. 2, pp 8-16, March 1983.

[3] M. Spellman, "*A comparison between frequency hopping and direct spread PN as anti-jam techniques*", IEEE Communications Magazine, Vol. 21, No. 2, pp 37-51, March 1983.

[4] R.L. Pickholtz, D.L. Shilling and L.B. Milstein, "*Theory of spread spectrum communications, a tutorial*", IEEE Trans. on Commun., Vol. COM-30, pp 855-884, May 1982.

[5] R.C. Dixon, *Spread Spectrum Systems*, New York, NY: John Wiley and Sons, 1976.

[6] D.J. Torrieri, *Principles of Military Communications Systems*, Dedham, MA: Artech House, 1981.

[7] J.L. Pearce and E.B. Felstead, "*MSAT DND experimental EHF communications system concept* ", CRC Technical Note 716 (SECRET), 1983.

[8] I.F. Blake and J.W. Mark, "*Study for coding techniques for interference channels*", University of Waterloo Research Institute, 1983, DSS Contract No. OSU82-00056.

[9] G.C. Clark and J.B. Cain, *Error Correcting Coding for Digital Communications*, New York, NY: Plenum Press, 1981.

√[10] R.F. Pawula and R. Golden, "*Simulations of convolutional coding/Viterbi decoding with noncoherent CPFSK*", IEEE Trans. Commun., Vol. COM-29, pp 1522-1526, October, 1981.

[11] J.M. Geist, "*Viterbi decoder performance in gaussian noise and periodic erasure bursts*", IEEE Trans. on Commun., Vol. COM-28, pp 1417-1422, August 1980.

[12] I.R.A. Richer, "*A simple interleaver for use with viterbi decoding*", IEEE Trans. on Commun., Vol. COM-26, pp 406-408, March 1978.

[13] T.C. Huang and L. Yen, "*Error probability of a noncoherent MFSK/FH receiver in the presence of interference and gaussian noise*", IEEE MILCOM '82, Paper 14.3.

[14] R.S. Orr, "*Performance of MFSK in simultaneous partial-band and repeat-back jamming*", IEEE MILCOM '82, Paper 28.5.

[15] A.J. Viterbi, "*A robust ratio-threshold technique to mitigate tone and partial band jamming in coded MFSK systems*", IEEE MILCOM '82, Paper 22.4.

[16] B.K. Levitt, "*Use of diversty to improve FH/MFSK performance in worst case partial band noise and multitone jamming*", IEEE MILCOM '82, Paper 28.2.

[17] P.H. Wittke, P.J. McLane, and P. Ma, "*Study of the reception of frequency dehopped M-ary FSK*", Queen's University Department of Electrical Engineering Research Report 83-1, March 1983, DSS Contract No. OSU82-00167.

[18] J.A. Heller and I.M. Jacobs, "*Viterbi decoding for satellite and space communication*", IEEE Trans. on Commun. Technology, Vol. COM-19, pp 835-848, October 1971.

[19] W.W Peterson and E.J. Weldon Jr., *Error Correcting Codes* Appendix C, pp 476, Cambridge, MA: MIT Press, 1972.

[20] T.A. Schonhoff, "*Symbol error probabilities for M-ary CPFSK: coherent and noncoherent detection*", IEEE Trans. Commun., Vol. COM-24, pp 644-652, June 1976.

[21] J.M. Wozencraft and I.M. Jacobs, *Principles of Communication Engineering*, pp 511-523, 577, New York, NY: John Wiley and Sons, 1967.

[22] D.E. Knuth, *The Art of Computer Programming*, pp 9-25, 170-173, 114-121, New York, NY: John Wiley and Sons, 1981.

[23] R.G. Stanton, *Numerical Methods for Science and Engineering*, pp 52 - 60, Englewood Cliffs, N.J.: Prentice Hall, 1961.

[24] K.S. Shanmugam, *Digital and Analog Communications Systems*, pp 227, New York, NY: John Wiley and Sons, 1979.

APPENDIX A

CONDITIONS FOR ORTHOGONALITY OF BINARY NCFSK SIGNALS

Consider the binary NCFSK signals of (2.3) and (2.4). Let

$$\rho = \int_0^T s_1(t)s_0(t)\, dt \qquad (A.1)$$

$$= \int_0^T A^2 \cos(a+b) \cos(a-b)\, dt$$

where $a = 2\pi f_c t+\phi$,

and $b = 2\pi f_d t$.

If the trigonometric identities

$$\cos(a+b) = (\cos a \cos b)-(\sin a \sin b)$$

$$\cos(a-b) = (\cos a \cos b)+(\sin a \sin b)$$

are used in (A.1)

$$\rho = \int_0^T A^2[(\cos^2 a \cos^2 b)-(\sin^2 a \sin^2 b)]\, dt. \qquad (A.2)$$

Now, use the following trigonometric identities in (A.2)

$$\cos^2 a = 1/2\ (1+\cos 2a)$$

$$\sin^2 a = 1/2\ (1-\cos 2a)$$

$$\sin^2 b = 1 - \cos^2 b$$

and after the simplification we get

$$\rho = (A^2)/2 \int_0^T (\cos 2b + \cos 2a)\, dt$$

$$= (A^2)/2\ [\int_0^T \cos\{4\pi f_c t+\phi\}\, dt$$

$$+ \int_0^T \cos\{4\pi\ f_d\ t\}\, dt\ ]. \qquad (A.3)$$

For large f_c

$$\rho = (A^2)T/2\ (\sin\{4\pi\ f_d\ T\})/(4\pi\ fd\ T)$$

$$= (E_s\ \sin\{2\pi\ h\})\ /\ (2\pi\ h) \qquad (A.4)$$

$$= 0 \qquad ;\ \text{for } h = n/2,\ \text{where } n = 1,2,...$$

Therefore, the binary NCFSK signals $s_0(t)$ and $s_1(t)$ are orthogonal for h = .5, 1.0, 1.5, ...

APPENDIX B

ANALYSIS OF BINARY NCFSK RECEIVER

The expression for the input to the binary NCFSK demodulator of Fig. 2.9, is given by (2.9) or (2.10) depending upon whether a "0" or "1" was sent. Assume here that a "0" was sent. Rewrite (2.9) as

$$r(t)= A[\ \cos\{2\pi(f_c-f_d)t\}\cos\Theta\}$$

$$-\sin\{2\pi(f_c-f_d)t\}\sin\Theta] + n(t).$$

Consider L_c, the output of the first integrator:

$$L_c = \int_0^T r(t)\cos\{2\pi(f_c-f_d)t\}$$

$$=A^2[\ \int_0^T \cos^2\ \{2\pi(f_c-f_d)t\}\ \cos\Theta\ dt$$

$$-\int_0^T \sin\{2\pi(f_c-f_d)t\}\ \cos\{2\pi(f_c-f_d)t\}\ \sin\Theta$$

$$+\int_0^T n(t)\cos\{2\pi(f_c-f_d)t\ dt] \qquad (B.1)$$

The first term in (B.1) is

$$A^2 \int_0^T \cos^2\ \{2\pi(f_c-f_d)t\}\ \cos\Theta\ dt$$

$$= \cos\Theta \int_0^T A^2\ \cos^2\ \{2\pi(f_c-f_d)t\}\ dt$$

$$= E_s\ \cos\Theta$$

where E_s is the energy in the received signal during one FSK symbol period.

The second term in (B.1) is

$$\int_0^T \sin\{2\pi(f_c-f_d)t\}\ \cos\{2\pi(f_c-f_d)t\}\ \sin\Theta$$

$$= \sin \Theta/2 \int_0^T \sin\{4\pi(f_c-f_d)t\}$$

$$= \sin \Theta/2 \; [(\cos\{4\pi(f_c-f_d)t)\}/(4\pi(f_c-f_d)t)\Big|_0^T]$$

$$= 0$$

Since n(t) is zero-mean Gaussian noise with variance No/2, the last term in (B.1) will be Gaussian, with zero mean and variance σ^2, where

$$\sigma^2 = E\{(\int_0^T n(t)A \cos\{\pi(f_c-f_d)t\} \; dt)^2\}$$

$$= E\{\int_0^T n(t)A \cos\{2\pi(f_c-f_d)t\} \; dt \int_0^T n(p)A \cos\{2\pi(f_c-f_d)p\} \; dp\}$$

$$= E\{\int_0^T\int_0^T n(t)n(p) \; A \cos\{2\pi(f_c-f_d)t\} \; A \cos\{2\pi(f_c-f_d)p\} \; dt \; dp\}$$

$$= \int_0^T\int_0^T E\{n(t)n(p)\}A \cos\{2\pi(f_c-f_d)t\} \; A \cos\{2\pi(f_c-f_d)p\} \; dt \; dp$$

$$= A^2 \; No/2 \int_0^T \cos^2 \; \{2\pi(f_c-f_d)t\} \; dt$$

$$= N_0 \; E_s/2 \tag{B.2}$$

Therefore

$$L_c = E_s \cos \Theta + u_1$$

where u_1 is a zero-mean Gaussian random variable of variance $N_0E_s/2$.

It can be similarly shown that L_s, L'_c and L'_s (the output of the second, third and fourth integrators respectively) may be expressed as:

$$L_s = E_s \sin \theta + u_2 \tag{B.3}$$

$$L'_c = u_3 \tag{B.4}$$

$$L'_s = u_4 \tag{B.5}$$

where u_2, u_3 and u_4 are all zero mean Guassian random variables of variance $N_0E_s/2$.

Therefore, for a zero transmitted

$$L = L^2{}_c + L^2{}_s - (L'{}_c{}^2 + L'{}_s{}^2)$$

$$= [(E_s \cos\theta + u_1)^2 + (E_s \sin\theta + u_2)^2] - (u_3{}^2 + u_4{}^2) \qquad (B.6)$$

Similarly, it can be shown that for a one transmitted

$$L = -[(E_s \cos\theta + u_1)^2 + (E_s \sin\theta + u_2)^2] + u_3{}^2 + u_4{}^2. \qquad (B.7)$$

Therefore, in general

$$L = i\,[(E_s \cos\theta + u_1)^2 + (E_s \sin\theta + u_2)^2 - (u_3{}^2 + u_4{}^2)] \quad (B.8)$$

where i is +1 if the transmitted symbol were a zero or is -1 if the transmitted symbol were a one.

For ease of computation in the simulation, L was normalized by dividing by $\sqrt{N_0\, E_s/2}$ to give

$$L_n = i\,[(\sqrt{2E_s/N_0}\cos\theta + v_1)^2 + (\sqrt{2\,E_s/N_0}\sin\theta + v_2)^2 - (v_3{}^2 + v_4{}^2)] \qquad (B.9)$$

where v_1, v_2, v_3 and v_4 are all zero mean, unit variance Gaussian random variables.

UNCLASSIFIED

Security Classification

DOCUMENT CONTRDL DATA – R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATING ACTIVITY

COMMUNICATIONS RESEARCH CENTRE
Department of Communications,
Ottawa, Ontario, K2H 8S2

2a. DOCUMENT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. DOCUMENT TITLE

A SIMULATION OF FREQUENCY-DEHOPPED BINARY AND 4-ARY NC-FSK SIGNALS.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

CRC Technical Note 722

5. AUTHOR(S) (Last name, first name, middle initial)

KEIGHTLEY, R.J.

6. OOCUMENT OATE

April 1984

7a. TOTAL NO. OF PAGES

34

7b. NO. OF REFS

23

8a. PROJECT OR GRANT NO.

65610

9a. ORIGINATOR'S DOCUMENT NUMBER(S)

TN 722

8b. CONTRACT NO.

9b. OTHER DOCUMENT NO.(S) (Any other numbers that may be assigned this document)

10. OISTRIBUTION STATEMENT

UNLIMITED

11. SUPPLEMENTARY NOTES

12. SPONSORING ACTIVITY

DEFENCE RESEARCH ESTABLISHMENT OTTAWA

13. ABSTRACT

This report describes the simulation of frequency dehopped binary and 4-ary non-coherent frequency shift keyed signalling (NCFSK) over a Gaussian channel. Orthogonal FSK signals and a hopping rate of one hop per FSK symbol were simulated. Simulations were performed with and without forward error correction. Rate 1/3, constraint length 8 and rate 1/2, constraint length 7 convolutional codes with hard decision Viterbi decoding were used. A family of curves showing the probability of bit error versus E_b/N_o was produced. Coding gains of about 2 dB at an error rate of 1 in 10^5 were observed for binary NCFSK and a coding gain of less than 1.0 dB was observed at the same error rate for 4-ary NCFSK. Some extensions of the work are suggested for future study.

UNCLASSIFIED

Security Classification

KEY WORDS

Simulation
Frequency Hopping
FEC Coding

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the organization issuing the document.

2a. DOCUMENT SECURITY CLASSIFICATION: Enter the overall security classification of the document including special warning terms whenever applicable.

2b. GROUP: Enter security reclassification group number. The three groups are defined in Appendix 'M' of the DRB Security Regulations.

3. DOCUMENT TITLE: Enter the complete document title in all capital letters. Titles in all cases should be unclassified. If a sufficiently descriptive title cannot be selected without classification, show title classification with the usual one-capital-letter abbreviation in parentheses immediately following the title.

4. DESCRIPTIVE NOTES: Enter the category of document, e.g. technical report, technical note or technical letter. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the document. Enter last name, first name, middle initial. If military, show rank. The name of the principal author is an absolute minimum requirement.

6. DOCUMENT DATE: Enter the date (month, year) of Establishment approval for publication of the document.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the document.

8a. PROJECT OR GRANT NUMBER: If appropriate, enter the applicable research and development project or grant number under which the document was written.

8b. CONTRACT NUMBER: If appropriate, enter the applicable number under which the document was written.

9a. ORIGINATOR'S DOCUMENT NUMBER(S): Enter the official document number by which the document will be identified and controlled by the originating activity. This number must be unique to this document.

9b. OTHER DOCUMENT NUMBER(S): If the document has been assigned any other document numbers (either by the originator or by the sponsor), also enter this number(s).

10. DISTRIBUTION STATEMENT: Enter any limitations on further dissemination of the document, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this document from their defence documentation center."

(2) "Announcement and dissemination of this document is not authorized without prior approval from originating activity."

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document, even though it may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall end with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (TS), (S), (C), (R), or (U).

The length of the abstract should be limited to 20 single-spaced standard typewritten lines; 7½ inches long.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a document and could be helpful in cataloging the document. Key words should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context.