AD-A241 041

(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
S ELECTE D
OCT 0 3 1991
D

THESIS

APPLICATION OF MULTI-FREQUENCY
MODULATION (MFM)
FACSIMILE MACHINES

by

James T. Nickerson

September 1990

Thesis Advisor: Paul H. Moose

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority	3 Distribution/Availability of Report Approved for public release; distribution is unlimited.
2b Declassification Downgrading Schedule	
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School — 6b Office Symbol *(if applicable)* 32	7a Name of Monitoring Organization Naval Postgraduate School
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization — 8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number
8c Address *(city, state, and ZIP code)*	10 Source of Funding Numbers Program Element No | Project No | Task No | Work Unit Accession No

11 Title *(include security classification)* APPLICATION OF MULTI-FREQUENCY MODULATION (MFM) TO FACSIMILE MACHINES

12 Personal Author(s) James T. Nickerson

13a Type of Report	13b Time Covered	14 Date of Report *(year, month, day)*	15 Page Count
Master's Thesis	From To	September 1990	54

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	facsimile, MFM, communications

19 Abstract *(continue on reverse if necessary and identify by block number)*

Multi-Frequency Modulation (MFM) has been developed at NPS using both differential quadrature-amplitude-modulation (DQAM) and differential quadrature-phase-shift-keying (DQPSK) encoding formats. This report dicusses the use of each of these formats in transmitting a facsimile encoded message over a voice frequency channel. The satisfactory transmission and receipt of facsimile messages was achieved using both DQPSK and D16-QAM encoding formats. Research and testing for this report included the use of variable facsimile transmission rates in an attempt to optimize MFM operating parameters. Experimental results revealed a higher error rate when decoding messages containing similar contiguous characters.

20 Distribution Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual P.H. Moose	22b Telephone *(include Area code)* (408) 646-2838	22c Office Symbol 62Me

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

APPLICATION OF MULTI-FREQUENCY MODULATION (MFM) TO FACSIMILE MACHINES

by

James T. Nickerson
Lieutenant Commander, United States Navy
B.S.E.E., U.S.N.A., 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1990

Author: James T. Nickerson

James T. Nickerson

Approved by: P.H. Moose

P.H. Moose, Thesis Advisor

Roberto Cristi

R. Christi, Second Reader

Michael A. Morgan

Michael A. Morgan, Chairman,
Department of Electrical and Computer Engineering

ABSTRACT

Multi-Frequency Modulation (MFM) has been developed at NPS using both differential quadrature-amplitude-modulation (DQAM) and differential quadrature-phase-shift-keying (DQPSK) encoding formats. This report discusses the use of each of these formats in transmitting a facsimile encoded message over a voice frequency channel. The satisfactory transmission and receipt of facsimile messages was achieved using both DQPSK and D16-QAM encoding formats. Research and testing for this report included the use of variable facsimile transmission rates in an attempt to optimize MFM operating parameters. Experimental results revealed a higher error rate when decoding messages contained similar contiguous characters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. BACKGROUND

The information explosion of the last two decades has been fueled in large part by the introduction of computers on a personal level. One of the aspects of this opportunity yet to be fully realized is the ability to route the information available to those who require it in an efficient and accurate fashion. It is toward this end that the digitization of the communications industry has taken place. This process has made communications more flexible, more reliable and less expensive. By combining the power of the personal computer (PC) with the advantages of digital communications techniques a large step in meeting the needs of an information hungry environment can be made. The utilization of Multi-Frequency Modulation (MFM) can help allow that step to be taken. MFM is a signal modulation technique that adapts easily to a variety of communication mediums. It can emulate most existing modulation formats and can generate new formats if required. In addition, by modulating and multiplexing, demodulating and demultiplexing through the use of the hardware and software of a host PC, costly analog equipment can be eliminated. With the addition of a few expansion boards to existing industry standard PC hardware the flexibility and diversity of MFM can be fully realized.

The focus of this thesis is the use of MFM over a voice frequency channel to modulate coded information produced by a standard Group III facsimile. Facsimiles typically rely on built-in modems to transmit their information. Since, at present, there are no standard high-speed (or low-speed) modems that use MFM as a modulation technique there are similarly no facsimiles that rely on MFM for transmission of data.

The layout of this thesis is as follows. Chapter II contains a functional description of Group III digital facsimile machines. In Chapter III the implementation of the voice band MFM system and its interface with a Canon FAX-222 is discussed. Chapter IV lists and analyzes the results of a performance evaluation of the system by comparing two different encoding formats: Differential Quadrature-Phase-Shift-Keying (DQPSK) and Differential 16-Quadrature-Amplitude-Modulation (D16-QAM). The conclusions and recommendations encompass Chapter V.

B. THEORY OF MULTI-FREQUENCY MODULATION

MFM is discussed in detail in Refs. 1 and 2. The following summary is included to provide continuity for the convenience of the reader.

1. MFM Signal Packet

In the description of MFM the following definitions are germane:

T: Packet length in seconds

ΔT: Baud length in seconds

L: Number of bauds per packet

$\Delta f = 1/\Delta T$: Frequency spacing between MFM tones

k_x: Baud length in number of samples

Δt: Time between samples in seconds

$f_x = 1/\Delta t$: Sampling frequency in Hz for D/A and A/D conversion

K: Number of MFM tones

ϕ_{lk}: Phase of the k^{th} tone in the l^{th} baud

A_{lk}: Amplitude of the k^{th} tone in the l^{th} baud

The basic unit of MFM is referred to as a "packet". Each packet is arbitrarily located in time and in the specific frequency spectrum in use. Contained in each packet are multiple slots dimensioned in time and frequency. Figure 1 gives an illustration of this concept.

Each packet contains, in the time domain, subintervals referred to as "bauds". A packet is constructed of L bauds each comprised of K tones. These tones occur simultaneously in each baud of the packet. The mathematical representation of the analog signal packet during the l^{th} baud is

$$x_l(t) = \sum_{k=1}^{k_x/2-1} A_{lk} \cos(2\pi k \Delta f t + \phi_{lk}), \qquad (l-1)\Delta T \le t \le l\Delta T, \qquad (1)$$

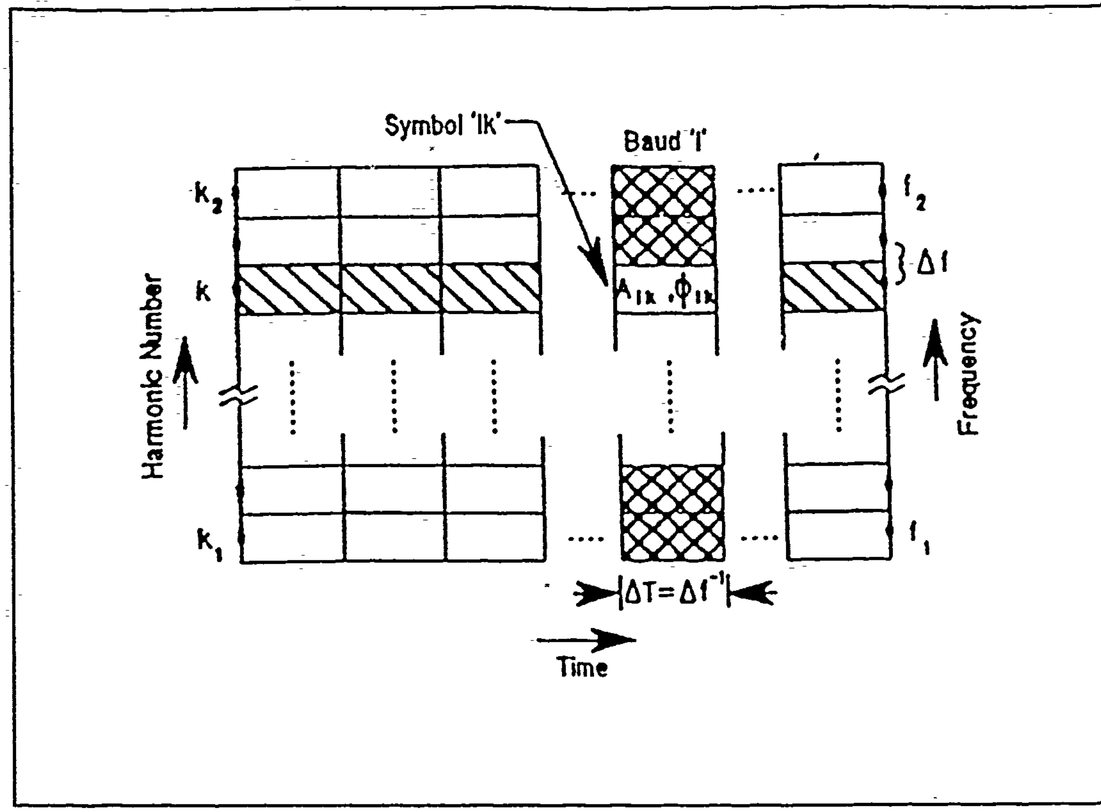

Figure 1. MFM signal packet: (after Ref. 1: p. 3).

and the corresponding discrete time signal of length k_x samples is [Ref. 1: pp. 2-3]

$$x_l(n) = \sum_{k=1}^{k_x/2-1} A_{lk} \cos\left(\frac{2\pi kn}{k_x} + \phi_{lk} \right), \qquad 0 \leq n \leq k_x - 1. \tag{2}$$

In order to send data, the information to be transmitted is independently amplitude and,or phase modulated onto the K tones. Since adjacent tones have a frequency separation of $1/\Delta T$ they form an orthogonal set. Additionally, signals in different bauds are orthogonal since they do not overlap in time. Accordingly a packet is an orthogonal set of $L\ K$ subsignals or symbols. [Ref. 2]

Concurrence with the Nyquist sampling theory demands that the highest frequency in the signal spectrum be less than half the sampling frequency f_x. By combining $\Delta t = \Delta T / k_x$, with the above definitions it can be shown that the sampling frequency must be $f_x = k_x \Delta f$. In order to be strictly within the Nyquist requirement an MFM baud is limited to a maximum of $k_x/2 - 2$ harmonic tones. These are spaced at intervals of Δf Hz from Δf Hz to $f_x/2 - \Delta f$ Hz. For the purposes of this thesis this interval will encompass the voice frequency band between 200 Hz and 3400 Hz.

Table 1. DESIGN PARAMETERS FOR A 1/2.5 SECOND MFM SIGNAL PACKET IN A 200-3400 HZ PASSBAND: (Ref. 12: p. 5)

Baud length(sec)	ΔT	1/40	1/20	1/10	1/5	1/2.5
No. of bauds	L	16	8	4	2	1
Tone spacing	Δf	40	20	10	5	2.5
Lowest harmonic	k_1	5	10	20	40	80
Lowest tone freq.	f_1	200	200	200	200	200
Highest harmonic	k_2	85	170	340	680	1360
Highest tone freq	f_2	3400	3400	3400	3400	3400
Samples per baud	k_x	256	512	1024	2048	4096
Sampling freq	f_x	10240	10240	10240	10240	10240
No. of tones	$k_2 - k_1$	80	160	320	640	1280

The values of $k_1 = f_1/\Delta f$ and $k_2 = f_2/\Delta f$ define the limits of the harmonic tones that may be assigned to the bauds in the packet. Those harmonics not located between k_1 and k_2 will be assigned an amplitude of zero. This implies that the maximum number of tones per baud will be $K = k_2 - k_1 + 1$. It is worth noting that since the bandwidth is described by $W = K \times \Delta f$ and time (the length of a packet is seconds) may be represented by $T = L\Delta T$, the time bandwidth product for the entire signal packet is $TW = L\Delta T \times K\Delta f$, or simply LK. This result equates to the total number of symbols that can be contained in a packet. Table 1 reflects the MFM parameters used in this thesis.

2. MFM Generation and Demodulation

The generation of MFM is performed a baud at a time using the properties of the Fast Fourier Transform (FFT) algorithm. The amplitude and phase of each of the tones to be included in the MFM signal are loaded into the front half of a complex valued array. The length of this array is equal to the number of samples per baud, k_x. In completing the array by loading the later part with the complex conjugates of the values already entered a symmetry about the midpoint is created. This insures that the Inverse FFT (IFFT) will yield a real valued sequence also containing k_x values. These samples are stored as binary numbers in the transmit PC. Clocking them through a digital-to-analog (D/A) converter at f_x samples per second is the final step in generating MFM for a particular baud. A complete signal packet is processed by the repetition of this sequence for each baud in the packet.

MFM is demodulated by simply reversing the process used to create it. The analog signal entering the receiving PC is converted back into digital signal format using an analog-to-digital (A/D) converter which samples the signal at a rate of f_x times per second. The resulting k_x complex values are loaded into the real parts of a complex valued array while the imaginary parts are set to zero. By applying the FFT algorithm to the array a new complex valued array is produced which contains, in its first half, the magnitude A_{lk} and phase ϕ_{lk} values of the tones of the MFM signal. Since the upper half of the array contains redundant values it is disregarded. Those harmonics present in the lower half that were not used can also be discarded since, in the absence of noise they are valued at zero and with noise present their exclusion is equivalent to filtering.

C. VOICE FREQUENCY MFM ENCODING SCHEMES

This thesis employs two different encoding formats in testing the facsimile applications of MFM. Each utilizes Gray-encoding techniques and can be software implemented on standard PCs. The following paragraphs give a brief description of each.

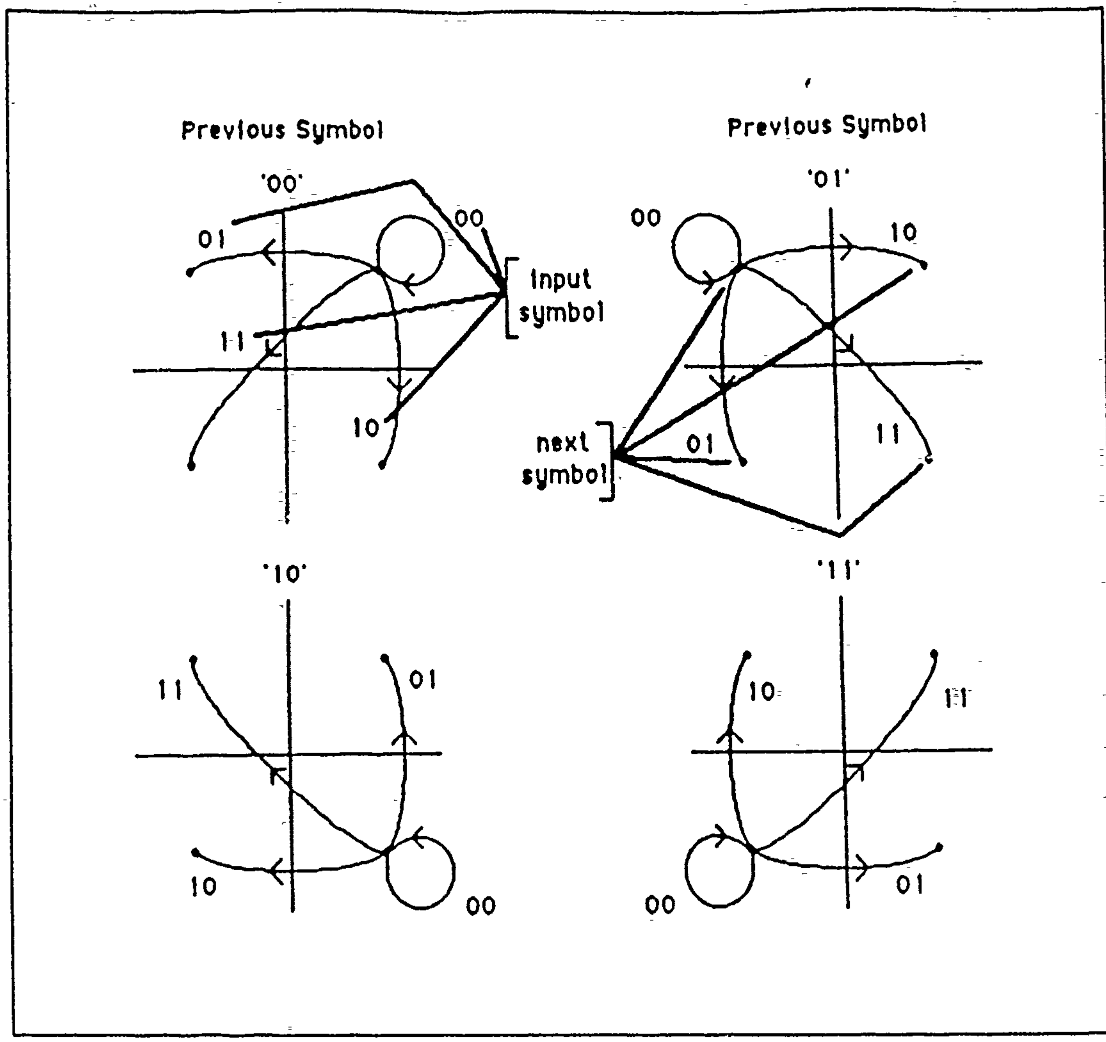

Figure 2. DQPSK encoding scheme: (from Ref. 2: p. 10).

1. DPQSK

In QPSK binary source digits are collected two at a time. These sequential digits instruct the system modulator which of four waveforms to produce. A single bit of information is carried by both the in-phase and quadrature components of each tone. DQPSK encoding is similar to QPSK encoding in that they both use the same four Gray-encoded two-bit symbols in the signal constellation. The difference between DQPSK and QPSK is in the elimination of the phase ambiguity. QPSK must rely on coherent regeneration using complex synchronization techniques while DQPSK overcomes the problem by generating a new differential two-bit symbol based on the similarity or difference of the present symbol with the preceding one. This process is shown in Figure 2. As illustrated in the figure, the following rules govern the generation of new symbols.

- An input of '00' produces a new symbol in the same quadrant as the previous symbol.
- An input of '01' rotates the new symbol $+\pi/2$ radians from the previous symbol.
- An input of '10' rotates the new symbol $-\pi/2$ radians from the previous symbol.
- An input of '11' rotates the new symbol π radians from the previous symbol.

DQPSK decoding of the MFM signal in the receiving PC is accomplished by determining the phase difference between successive symbols. [Ref. 1: p. 9-11]

2. D16-QAM

D16-QAM consists of two independently amplitude-modulated carriers in quadrature. It differs from QPSK in that two bits of information are carried by both the in-phase and quadrature components of each tone. Figure 3 depicts the gray-encoded D16-QAM signal constellation used for this thesis. Although the data rate for D16-QAM is doubled that of DQPSK it also has a higher bit error rate. This is a result of the greater possibility of magnitude and phase errors due to the increase in the number of possible magnitude states and the smaller sector size. A detailed description of D16-QAM encoding/decoding is contained in [Ref. 3: pp. 10-13].

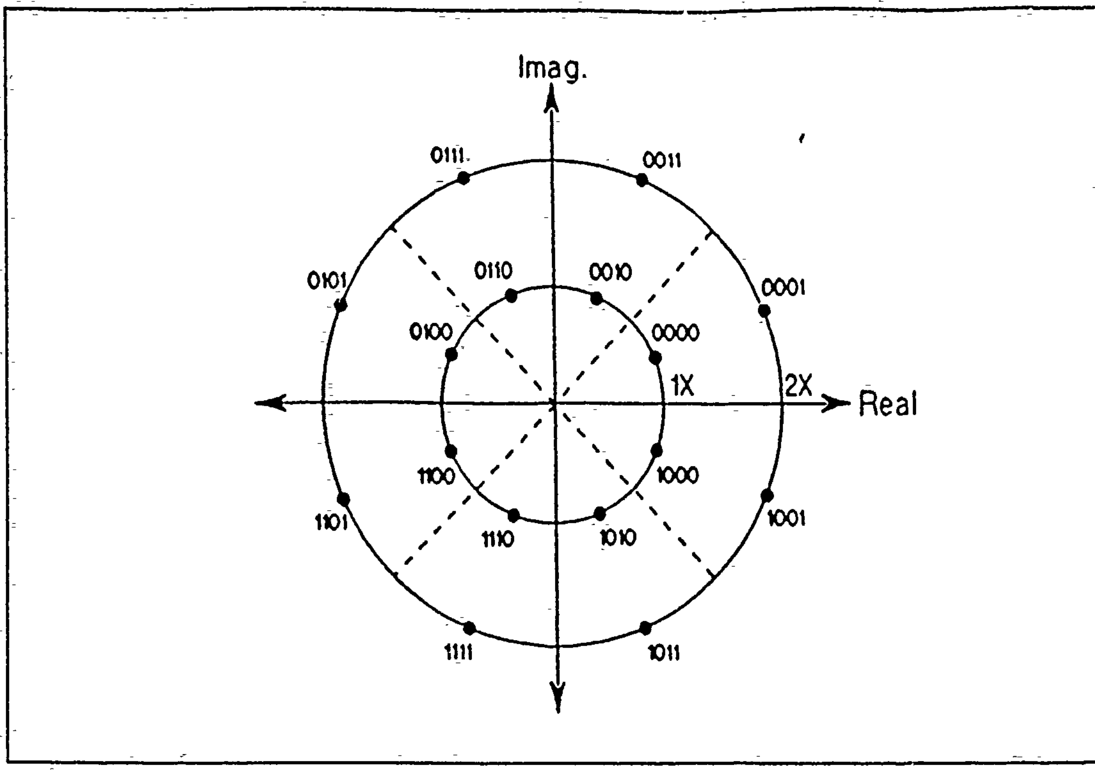

Figure 3. D16-QAM signal constellation.

II. FUNCTIONAL DESCRIPTION OF FACSIMILES

The original facsimile technique, identified as Group I and Group II machines, utilized analog methods and required more than three minutes to transmit a standard size document. This was due to the restricted bandwidth of voiceband telephone circuits. Most documents likely to be sent via facsimile transmission, can be effectively reproduced when quantized and transmitted in a digital format, using two tones. This prompted the research into and eventual adoption of digital transmission techniques for future designs. In 1977, Study Group XIV of The International Telegraph and Telephone Consultative Committee (CCITT) set the following standard for this new class of facsimiles denoted as Group III: [Ref. 4]

- standard - able to accept size A4 documents (8 ½ " x 11")
  option - able to accept size A3 documents (11" x 17") with no loss in resolution
- standard - 1728 samples/scan line on an A4 document
  option - up to 2600 samples/scan line on an A3 document with no loss in resolution
- standard - 3.85 scan lines/mm
  option - 7.7 scan lines/mm giving higher resolution
- standard - 4800 bits/sec transmission rate
  option - 9600 bits/sec transmission rate
- A4 document received in a minute or less over general telephone lines

Recently a standard for Group IV facsimiles has also been established. This class of machines is capable of faster transmission times with fewer errors. They are primary designed to take advantage of the Intergrated Services Digital Network (ISDN) if and when it is widely available.

Facsimile transmission is the process of transmitting a two-dimensional image as a sequence of successive line scans. The position of the scan lines on the page and the position along a scan line create a spatial location that define a grid of picture elements called *pixels*. The facsimile scans each line and performs a two level quantization that identifies each pixel as black (B) or white (W). Almost without exception, this is accomplished by an adaptive-threshold or floating-threshold device that filters out small intensity variations. This enhances the probability of long, unbroken runs of black or white pixels. [Ref. 5: p. 661]

Run length coding exploits such a pattern. In this coding a two-valued input process is transformed (non-linearly) into a many run-length output process. Figure 4 illustrates this sequence.

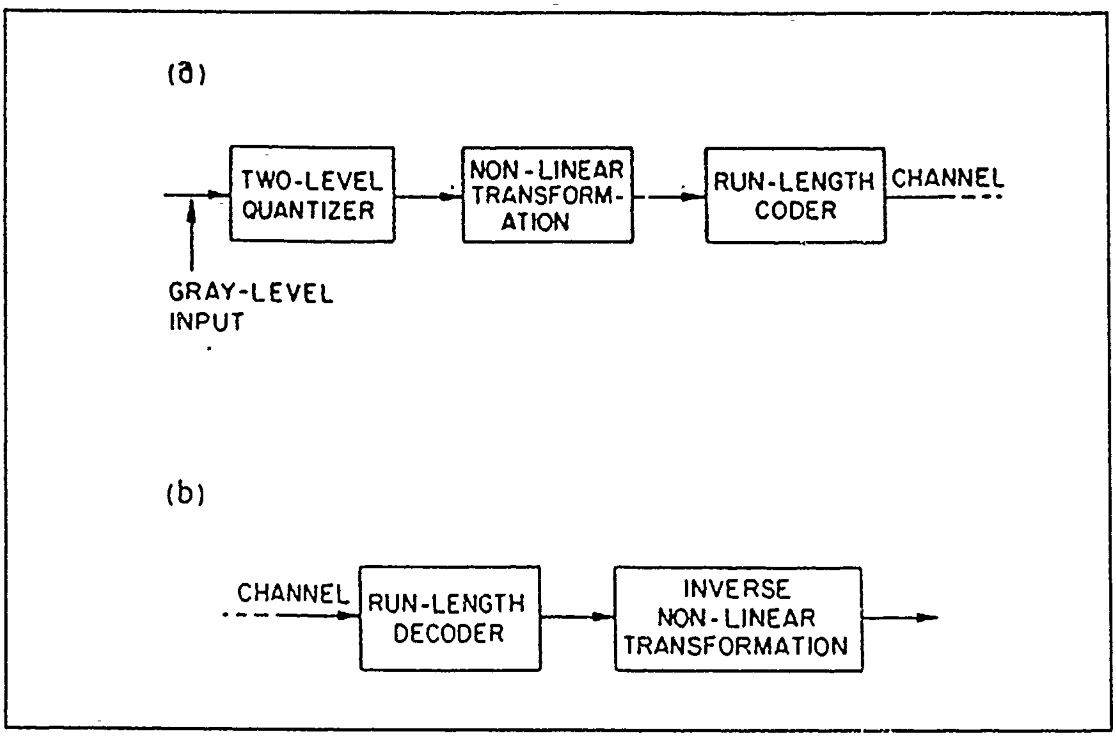

Figure 4. Block diagram of run-length coding; (a) coder, (b) decoder: (Ref. 7: p. 468).

The following advantages stem from the use of a two-level input: [Ref. 6: p. 466]

1. A W-run must be followed by a B-run eliminating the need to transmit separate amplitude and timing information.
2. For correctly reproduced run lengths the quantization or reconstruction noise is zero allowing for exact replication of inputs.
3. Since facsimile communication is not real-time there is no constraint on the time taken for encoding a line other than the average time of transmission for a document class over a given digital channel.
4. Two-level facsimile are robust enough so that noise induced transitions to a neighboring gray level do not affect the run length properties.

A. ONE-DIMENSIONAL CODE; MODIFIED HUFFMAN

1. Attributes

Run-length codes have been widely employed in data reduction systems. However it was thought that they did not recover quickly enough from transmission errors for facsimile use. Instead comma and block codes were proposed. It was subsequently found that although these codes were easier to implement they were more sensitive to changes in document type and usually produced significantly lower compression factors. As a result different run-length codes were tested. In searching for a standard that would be compatible with a wide variety of machine designs and capabilities the CCITT chose a modified Huffman code. Some of the reasons for this choice are listed below.

- High compression efficiency
- One dimensional; it exploits only the relationship between adjacent pixels on the same scan line eliminating the possibility of propagating errors to subsequent lines.
- Compact; it is the optimum method for constructing the smallest average codeword length for a given source alphabet.

- Instantaneous; since it is a prefix code no codeword can appear as the beginning of any other codeword making it possible to decode each word as it arrives.
- Exhaustive; any sequence of binary digits devoid of errors can be decoded correctly.
- Free from patent restrictions

Initially Huffman codes were overlooked for facsimile use because they required large amounts of storage, were slow in operation and were thought to be highly susceptible to errors. In the late [illegible] faster methods for decoding Huffman codes were devised that required only a modest amount of storage [Ref. 7]. In addition, studies at that time indicated that its recovery properties were satisfactory enough to preclude the requirement for automatic repeat request (ARQ) or forward-acting error correcting (FEC) methods. This was important since these techniques were considered complex, added extra cost to the equipment, and increased transmission times.

2. Coding Scheme

As stated before, each standard scan line is composed of 1728 pixels. Each of these pixels are shaded black or white. In order to maintain color synchronization the first pixel is always assumed to be white. If, in fact, the first pixel is black a white run of zero pixels is inserted at the beginning of the line.

The number of consecutive black or white pixels (referred to as a run) determines the actual code to be transmitted. Tables 2 and 3 outline the codes used for each color at various run lengths, the most often occurring lengths having the shortest codes. Studies have indicated that separate black and white tables increase the compression factor by 25 percent. [Ref. 4: p. 857]

Table 2. MODIFIED HUFFMAN CODE -- TERMINATING CODEWORDS

Run length	White	Black	Run length	White	Black
0	00110101	0000110111	32	00011011	000001101010
1	000111	010	33	00010010	000001101011
2	0111	11	34	00010011	000011010010
3	1000	10	35	00010100	000011010011
4	1011	011	36	00010101	000011010100
5	1100	0011	37	00010110	000011010101
6	1110	0010	38	00010111	000011010110
7	1111	00011	39	00101000	000011010111
8	10011	000101	40	00101001	000001101100
9	10100	000100	41	00101010	000001101101
10	00111	0000100	42	00101011	000011011010
11	01000	0000101	43	00101100	000011011011
12	001000	0000111	44	00101101	000001010100
13	000011	00000100	45	00000100	000001010101
14	110100	00000111	46	00000101	000001010110
15	110101	000011000	47	00001010	000001010111
16	101010	0000010111	48	00001011	000001100100
17	101011	0000011000	49	01010010	000001100101
18	0100111	0000001000	50	01010011	000001010010
19	0001100	00001100111	51	01010100	000001010011
20	0001000	00001101000	52	01010101	000000100100
21	0010111	00001101100	53	00100100	000000110111
22	0000011	00000110111	54	00100101	000000111000
23	0000100	00000101000	55	01011000	000000100111
24	0101000	00000010111	56	01011001	000000101000
25	0101011	00000011000	57	01011010	000001011000
26	0010011	000011001010	58	01011011	000001011001
27	0100100	000011001011	59	01001010	000000101011
28	0011000	000011001100	60	01001011	000000101100
29	00000010	000011001101	61	00110010	000001011010
30	00000011	000001101000	62	00110011	000001100110
31	00011010	000001101001	63	00110100	000001100111

This advantage clearly outweighs any inconvenience due to duplication.

Table 3. MODIFIED HUFFMAN CODE -- MAKE-UP CODEWORDS

Run length	White	Black	Run length	White	Black
64	11011	0000001111	960	011010100	0000001110011
128	10010	000011001000	1024	011010101	0000001110100
192	010111	000011001001	1088	011010110	0000001110101
256	0110111	000001011011	1152	011010111	0000001110110
320	00110110	000000110011	1216	011011000	0000001110111
384	00110111	000000110100	1280	011011001	0000001010010
448	01100100	000000110101	1344	011011010	0000001010011
512	01100101	0000001101100	1408	011011011	0000001010100
576	01101000	0000001101101	1472	010011000	0000001010101
640	01100111	0000001001010	1536	010011001	0000001011010
704	011001100	0000001001011	1600	010011010	0000001011011
768	011001101	0000001001100	1664	011000	0000001100100
832	011010010	0000001001101	1728	010011011	0000001100101
896	011010011	0000001110010	EOL	000000000001	000000000001

Use of straight Huffman code tables for each of the black and white runs would require a huge number of codewords (1728 in each). The "modified" version was designed to reduce that number and simplify implementation. Each run length on a scan line is translated into one or a combination of two codeword types: Terminating codewords (TC; Table 2) and Make-up codewords (MUC; Table 3). Runs between 0 and 63 pixels in length are transmitted using a single TC. Run lengths from 64 to 1728 pixels are sent using a MUC followed by a TC. A combination of pixels which equals $64 \times N$ (where N is an integer between 1 and 27) that is the same as or shorter than the run to be transmitted is represented by the MUC. The difference between that and the actual value of the run is contained in the appended TC. The combination of the MUC and TC is what is transmitted.

As noted earlier, the Group III standard contains an optional extension that allows A3 size documents to be transmitted. This requires an increase in pixels/scan line to almost 2600. In order to provide for this contingency an extended code table is necessary. Such a table is listed below. It should be noted that use of the Extended Modified Huffman Code requires that EOL codeword by modified to 11 X "0" + "1" otherwise it is possible through certain code combinations to create a false EOL.

Table 4. EXTENDED MODIFIED HUFFMAN CODE

Run Length (White or Black)	Make-up Codewords
1792	00000001000
1856	00000001100
1920	00000001101
1984	000000010010
2048	000000010011
2112	000000010100
2176	000000010101
2240	000000010110
2304	000000010111
2368	000000011100
2432	000000011101
2496	000000011110
2560	000000011111

The variety in the number of possible run lengths per scan line is large. Similarly the total number of coded bits per line can fluctuate significantly. In order to rectify this problem and enable transmitters and receivers to remain in sync on a line-by-line basis minimum and maximum scan line times (MSLT) were established. The maximum is set at five seconds. If this is exceeded the receiver proceeds to disconnect the line. The standard minimum is 20 ms (this equates to 96 coded bits at the standard transmission rate of 4800 bits/sec). There are options for 40 ms (192 bits), 10 ms (48 bits), 5 ms (24 bits) and 0 ms (ie, no restriction). Should a scan line contain fewer than the requisite number of bits for the MSLT in use "fill" bits (strings of zeros) are added to make up the difference. These bits may only be added between data and the End of Line (EOL), illustrated in Figure 5, and are easily recognized at the receiver and discarded.

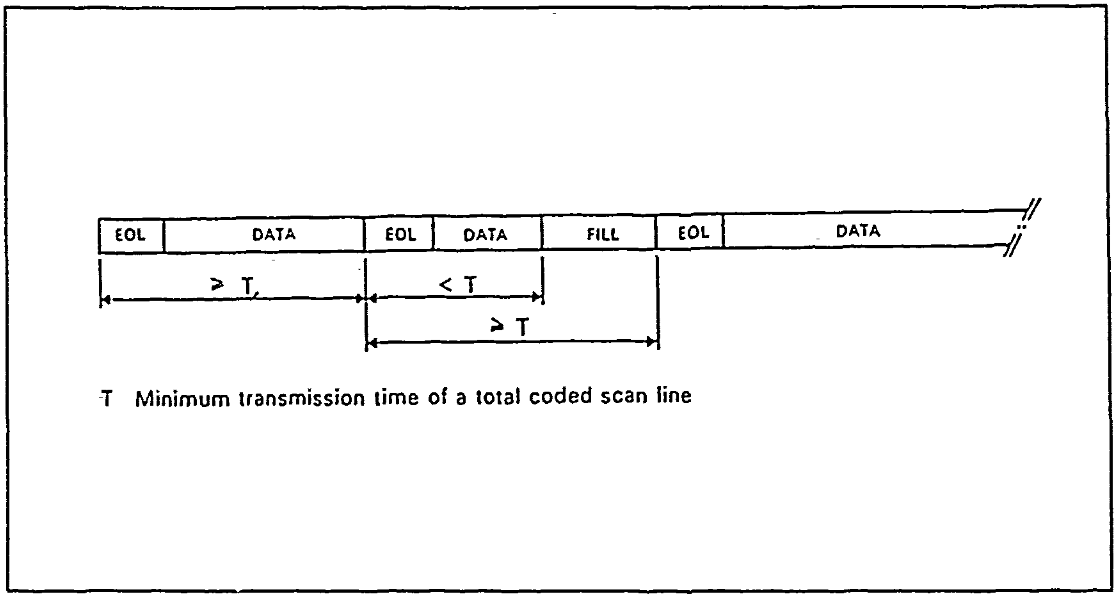

Figure 5. MSLT and "fill" bit locations on a scan line.

Each run on the scan line is coded in turn until all have been transmitted. A unique EOL codeword is transmitted at the end of each line. The EOL codeword is written 10 X "0"+"1". Since fill bits may increase the number of zeros at the end of a line to more than ten, the "1" at the end serves to indicate the start of the next scan line. A perusal of Tables 2 and 3 will reveal that no codeword ends in a sequence of more than three "0"s or begins with a sequence of "0"s larger than six. This ensures that in a non-error environment the EOL forms a sequence which cannot be produced by any combination of codewords. Should an error to the EOL codeword occur it will take one of two forms.

1. *Lost EOL.* This is the result of an error that so corrupts the EOL codeword that it can no longer be recognized. An error concealment technique is required to remedy this situation. The most popular of these techniques are: 1) to replace the damaged line by an all white line, or 2) to repeat the previous line. Alternative methods are outlined in detail in [Refs. 8 and 9].
2. *False EOL.* An error occurs in either the coded scan line or fill bits that creates a spurious or false EOL. This results in an extra line being added to the document. These errors are easy to locate because the lines associated with them contain fewer bits than specified by the MSLT.

The quality of phone lines, in the US at least, are such that the probability of EOL errors occurring in sufficient quantities to pose a significant problem is extremely limited.

The end of document transmission is indicated by 6 consecutive EOL codewords which form the return to control (RTC) signal. Figure 6 shows the last coded scan line on a page.

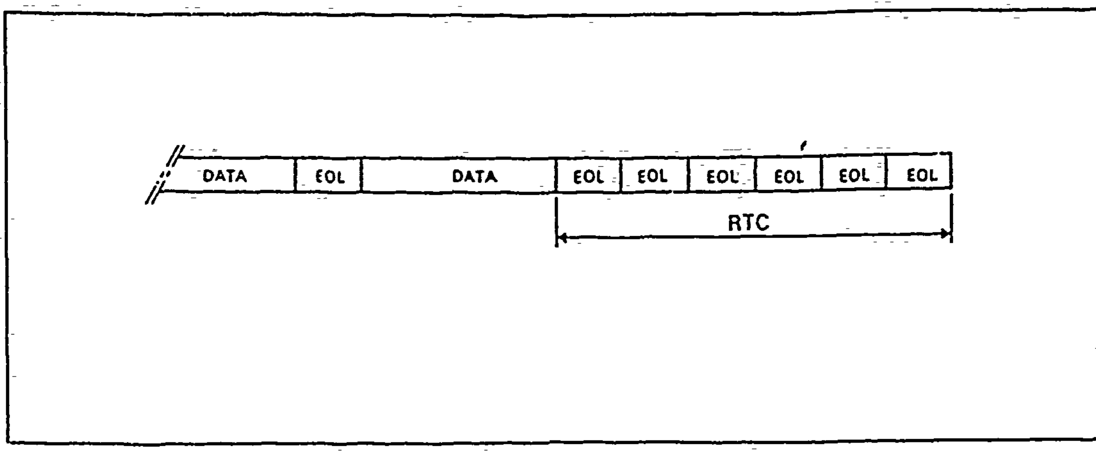

Figure 6. Diagram of RTC configuration.

B. TWO-DIMENSIONAL; MODIFIED READ

Two-dimensional facsimile coding techniques were originally proposed by Japan as a way of speeding up the transmission of documents scanned at high resolution without significantly increasing system costs. CCITT study Group XIV received proposals from IBM Europe, 3M Company, AT & T, British Post Office, Federal Republic of Germany, Japan and Xerox corporation for inclusion with the standard as an option for Group III facsimiles. Each method was extensively tested and their performances compared. The results showed that there was little difference in terms of compression efficiency and error susceptibility among all of the entries. The relative element address designate (READ) [Ref. 10] code submitted by Japan was strongly supported because it had already been successfully implemented in a number of commercial machines. Eventually a modified version of READ, containing several alterations aimed at simplifying implementation, was adopted.

1. Attributes

Although the READ code is faster than the one-dimensinal method it is also significantly more complex. Similar to the Huffman the modified READ is a line-by-line coding scheme. It utilizes the relative position of changing pixels on both a reference line and a coding line to code a run of like colored pixels. A changing pixel is merely a pixel whose color (black or white) is different from that of the previous pixel along that scan line.

There are five different changing pixels used in application of the coding algorithm. They are situated at various locations on the reference and coding lines depending on pixel color patterns. The different types are defined below with illustrations given in Figure 7.

a_0 - The reference or starting changing pixel on the coding line. At the start of the coding line, a_0 is set on an imaginary white changing pixel located immediately before the first actual element on the coding line. Subsequently its position is dictated by the coding mode of the previously coded pixel run.

a_1 - The next changing pixel to the right of a_0 on the coding line. It must be opposite in color to a_0 and is the next changing pixel to be coded.

a_2 - The next changing pixel to the right of a_1.

b_1 - The next changing pixel on the reference line to the right of a_0 with the same color as a_1.

b_2 - The next changing pixel on the reference line to the right of b_1.

Following the coding of a scan line, that coded line becomes the reference line for the the next line to be coded. One of the disadvantages of two-dimensional codes is that errors can be propagated on to successive lines. In order to minimize this problem a one- dimensionally coded line is inserted between every $K - 1$ two- dimensionally coded lines. K is usually set equal to 2 for normal resolution and 4 for high resolution. Increasing the number of one-dimensionally

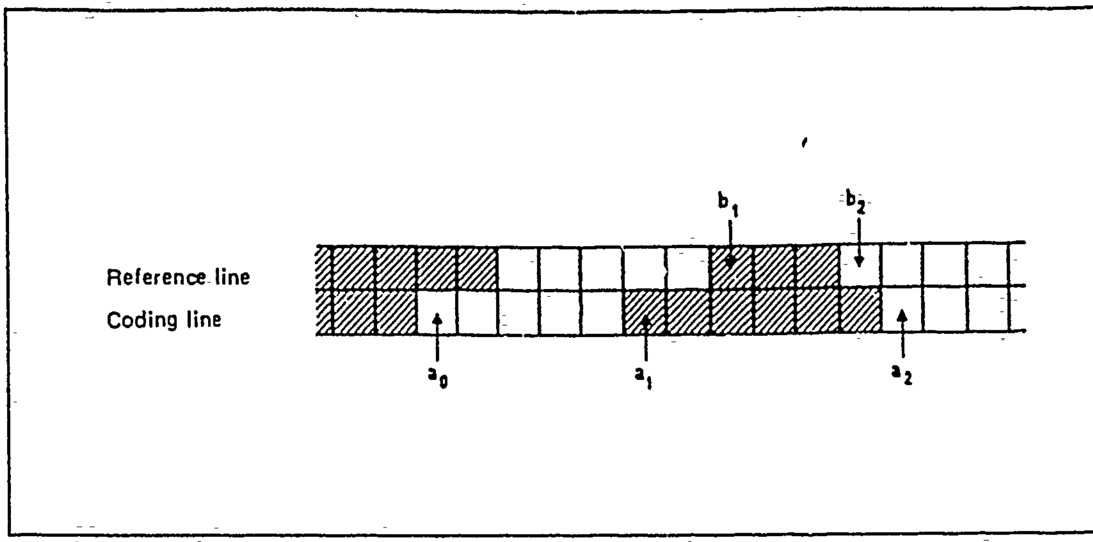

Figure 7. Changing pixels used in 2-D coding: (Ref. 11: p. 21).

coded scan lines interspersed among the two- dimensionally coded ones when such a procedure would improve compression rates or error susceptibility can be accomplished without affecting compatibility.

2. Coding Procedure

The coding process uses three separate coding modes. Each is listed below and pictured in its respective figure. [Ref. 11: p. 21]

Pass Mode Coding. As seen in Figure 8 this mode is identified when the position of b_2 lies to the left of a_1 This mode locates white or black runs on the reference line that do not correspond in color and length to a run on the coding line. A single codeword represents the passmode as indicated in Table 4. When this mode has been coded, a_0 is set on the coding line pixel below b_2 in preparation for the next coding run (ie. a_0').

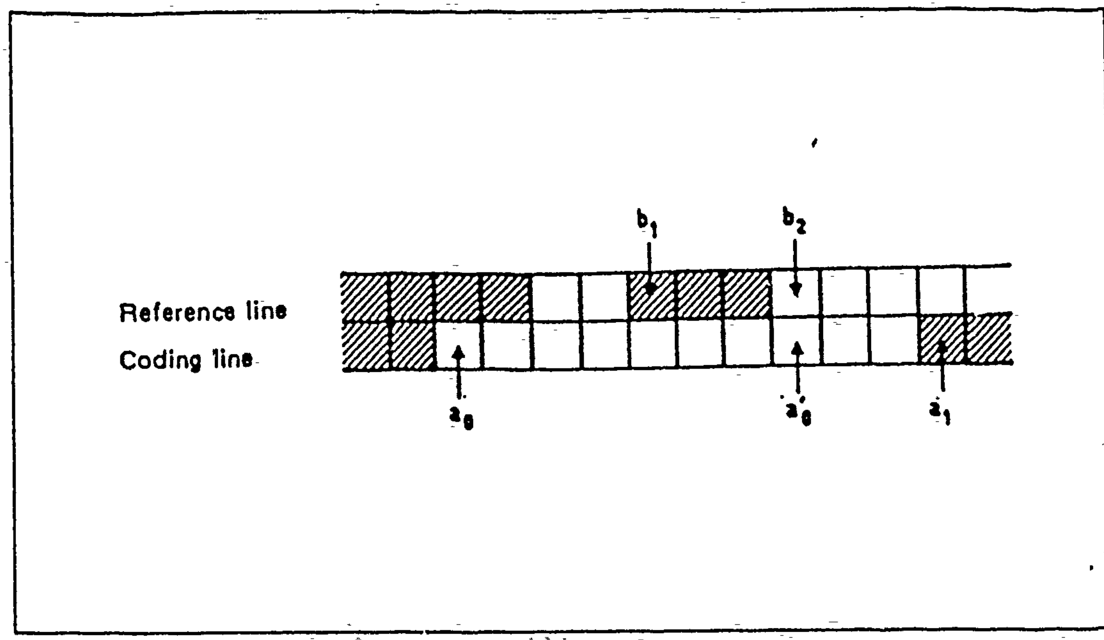

Figure 8. Pass Mode Coding: (Ref. 11: p. 22).

Vertical Mode Coding. When this mode is identified, the code is produced by the relative position of a_1 to b_1. This relative distance, a_1b_1, can take on one of seven values; $V(0)$, $V_L(1)$, $V_L(2)$, $V_L(3)$, $V_R(1)$, $V_R(2)$, $V_R(3)$. As shown in Table 4, each of these values is represented by a separate codeword. The subscripts L and R indicate whether a_1 is to the left or right, respectively, of b_1 . The number in brackets represents the distance or number of pixels between a_1 and b_1. When this mode has been coded a_0 is set on the pixel previously referred to as a_1. See Figure 9.

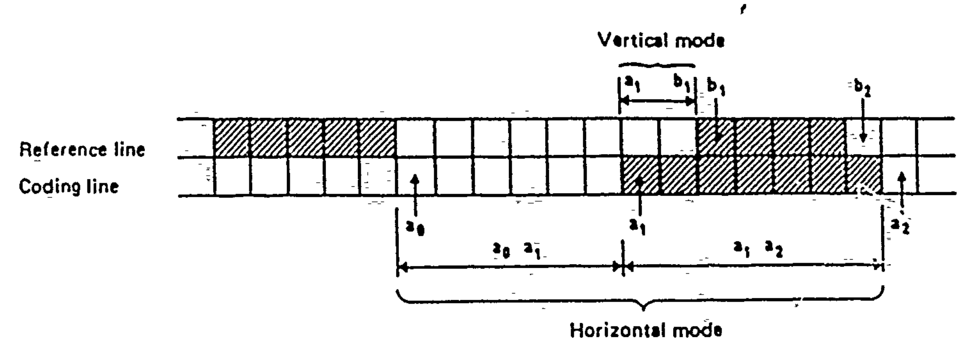

Figure 9. Vertical & Horizontal Mode Coding: (Ref. 11: p. 23).

Horizontal Mode Coding. This mode is used when the position of a_1 cannot be coded using vertical mode coding. When this mode is identified the combined codeword $H + M(a_0a_1) + M(a_1a_2)$. H is the flag codeword "001" as indicated in Table 4. $M(a_0a_1)$ and $M(a_1a_2)$ are codewords which represent the color and length of runs a_0a_1 and a_1a_2 respectively. They are assigned from appropriate white or black one-dimensional codetable (Table 2, 3, or 4). When this mode has been coded a_0 is set on the pixel previously referred to as a_2. See Figure 9.

Examples of the application of each of these modes are contained in the Appendix.

Table 5. TWO-DIMENSIONAL CODE

MODE	ELEMENTS TO BE CODED		NOTATION	CODEWORD
PASS	b_1, b_2		P	0001
HORIZONTAL	a_0a_1, a_1a_2		H	$001 + M(a_0a_1) + M(a_1a_2)$
VERTICAL	a_1 JUST UNDER b_1	$a_1b_1 = 0$	$V(0)$	1
	a_1 TO THE RIGHT OF b_1	$a_1b_1 = 1$	$V_R(1)$	011
		$a_1b_1 = 2$	$V_R(2)$	000011
		$a_1b_1 = 3$	$V_R(3)$	0000011
	a_1 TO THE LEFT OF b_1	$a_1b_1 = 1$	$V_L(1)$	010
		$a_1b_1 = 2$	$V_L(2)$	000010
		$a_1b_1 = 3$	$V_L(3)$	0000010
END OF LINE (EOL) CODEWORD				000000000001
1-D CODING OF NEXT LINE				EOL + '1'
2-D CODING OF NEXT LINE				EOL + '0'

At the commencement of each coding session the position of each of the changing pixels a_1, a_2, b_1 and b_2 is determined. Position a_0 was fixed at the end of the preceding session. The coding sequence identifies the next coding mode by the relative positions of the changing pixels and selects the appropriate codeword from Table 4. This procedure is illustrated by the flow diagram shown in Figure 10. Basically it consists of two steps. [Ref. 4: p. 864]

STEP 1

i) If a pass mode is recognized the code word "0001" is issued. The pixel under b_2 is regarded as the new starting pixel a_0 in preparation for the next coding sequence.

ii) If a pass mode is not detected Step 2 is followed.

STEP 2

i) Determine the number of pixels that separate a_1 and b_1 .

ii) If $|a_1b_1| \leq 3$, then the vertical mode coding is used and the appropriate codeword from Table 4 is issued.

iii) If $|a_1b_1| > 3$, then the horizontal mode coding is used. A codeword composed of a combination of the flag "001", and the code from the one-dimensional table for the runs between a_0 and a_1, and between a_1 and a_2 is issued.

Should the first pixel on the coding line require horizontal mode coding the value a_0a_1 is replaced by $a_0a_1 - 1$ to ensure the correct run-length value is transmitted. This allows that if the first pixel is actually black then the codeword $M(a_0a_1)$ will represent a white run of zero length. The coding of an imaginary changing pixel adjacent to the last actual pixel signals the end of coding for that line.

Fill bits, when required, are implemented in the same manner as those used in the one-dimensional coding technique. In the two-dimensionally coded procedure a single tag bit is added to the EOL codeword used for one-dimensional coding to indicate whether the next line is one or two-dimensionally coded. As depicted at the bottom of Table 4, EOL + "1" portends a subsequent one-dimensionally coded line while EOL + "0" identifies an upcoming two-dimensionally coded line. The return to control (RTC) is also slightly different

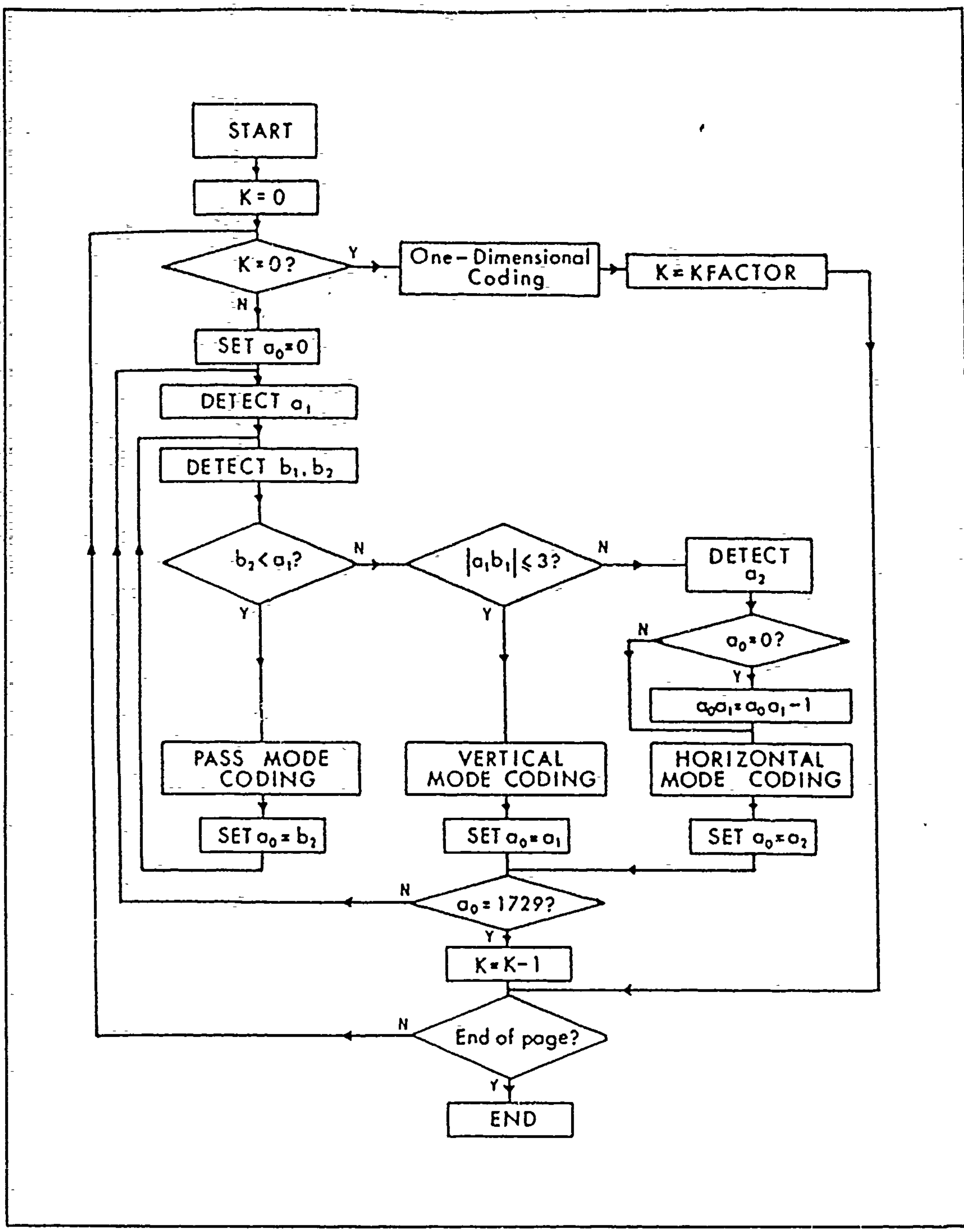

Figure 10. Flow diagram for 2-D coding.

from the one-dimensional scheme. In this case a "1" tag bit is appended to each of the six consecutive EOL codewords that form the RTC. See Figure 11.

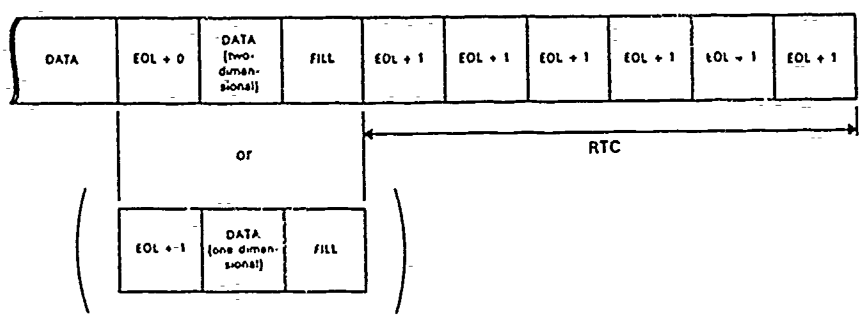

Figure 11. Diagram of RTC configuration for 2-D coding.

C. SUMMARY

The combination of the one-dimensional modified Huffman code with the two-dimensional modified READ code provides the communication industry with a standard that represents an excellent blend of compression efficiency, error susceptibility and implementation complexity. The development of a digital facsimile standard is indicative of the direction of this industry toward exploiting the advantages of digital processing techniques.

III. SYSTEM IMPLEMENTATION

The MFM system developed by LT Terry K. Gantenbein, USN, in Ref. 2 and modified by LT Peter G. Basil, USCG, in Ref. 3 and LCDR Charles P. Salsman, USN, in Ref. 12 provide the backdrop for the research of this thesis. Coded information generated by facsimile machines replaced the simple ASCII files used by Salsman for system testing. The results of tests using the facsimile produced messages are given in Chapter IV. A simple diagram of the system as configured for this thesis is pictured below.

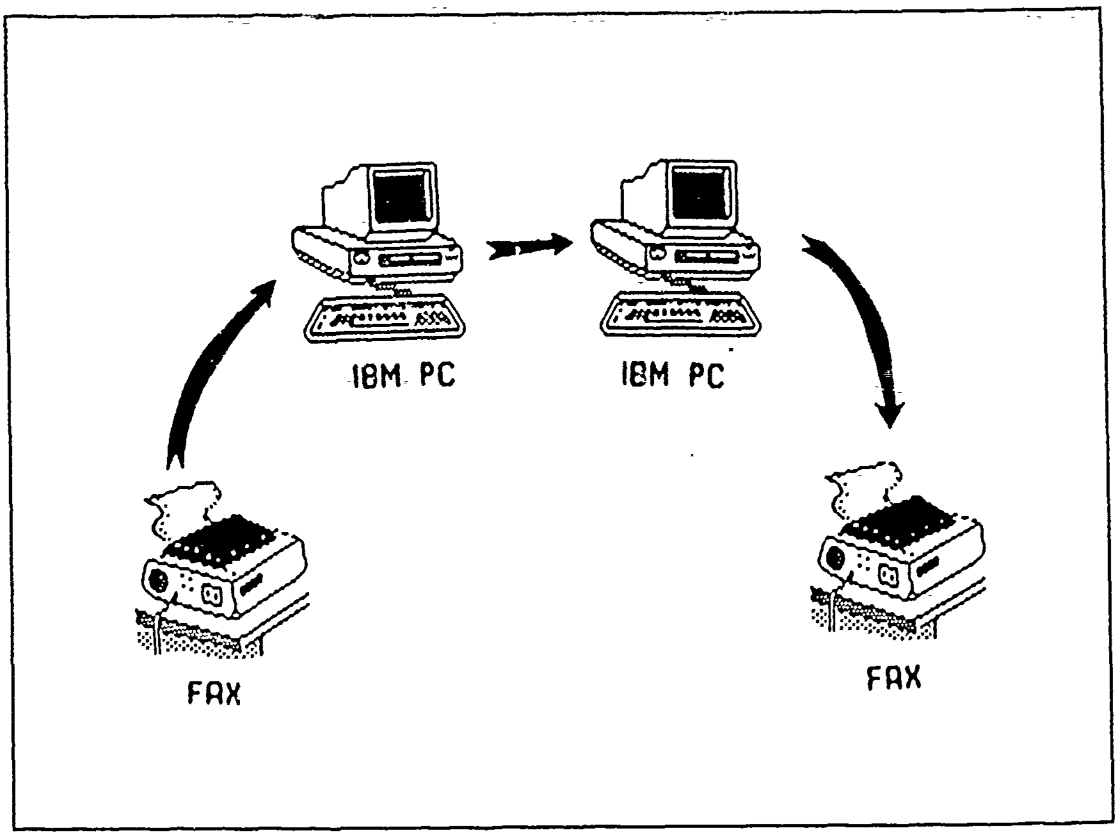

Figure 12. Facsimile/MFM Diagram.

A. FACSIMILE AND INTERFACE

Commercially produced facsimile machines compatible with CCITT standards were used in this research; specifically the Canon FAX-222. There were several reasons this particular make and model was chosen. One was its ability to transmit and receive data via an RS-232C serial interface port. This capability was utilized in order to provide an interface between the facsimiles and the PC's involved with transmitting and receiving the MFM signal. The data was encoded in the facsimile as described in Chapter II and then transferred through the RS-232C port to the transmit PC. The other requirement this fax fulfilled was the ability to transmit data at multiple rates through the RS-232C port. The need for this additional ability is discussed in the next chapter.

In order to precipitate data transfer and file management some software for the PC's was needed. Due to the proprietary design of the Canon RS-232C interface it was necessary to purchase the software produced for Canon by L. A. Business Systems to fulfill this requirement. It is called CAN-FAX and facilitates numerous facsimile/PC interactions. Descriptions of the options specifically used is this thesis follow:

1. **Scanning.** A document is scanned by the facsimile, coded via the methods described in Chapter II and transmitted via the RS-232C port to a designated file in the PC.
2. **Printing.** The reverse of scanning. A file is transferred from the PC via the RS-232C port to the facsimile where it is decoded and printed out.

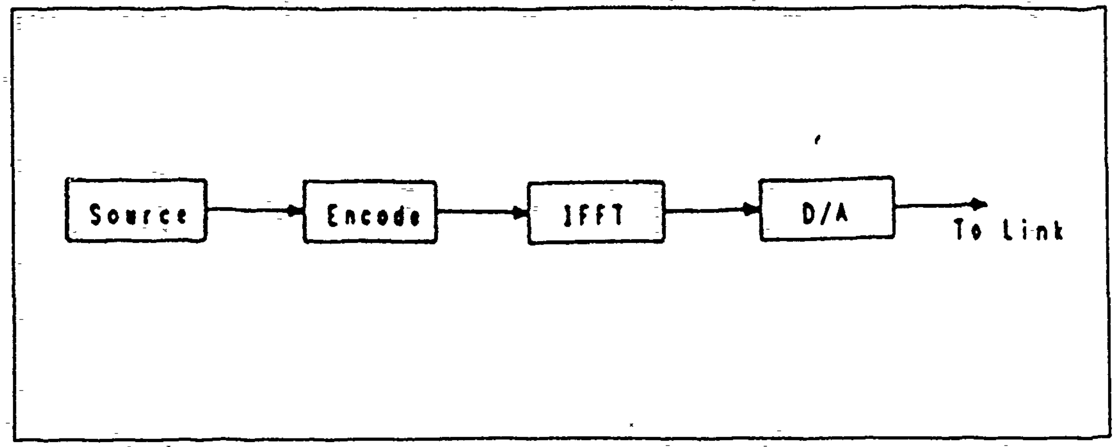

Figure 13. Block diagram of the MFM transmitter.

B. MFM TRANSMITTER

Transmitting MFM requires a specific interaction of hardware and software. Figure 14 is a functional block diagram of an MFM transmitter. The source in the figure refers to the facsimile. Its output is encoded and transferred as described above, appearing as a serial binary signal at the input to the "encode" block. This data is encoded and an IFFT is performed by a collection of Turbo Pascal procedures contained in the transmit PC. A further description of the programs involved along with a listing of the code is contained in [Ref. 12: pp. 46-61].

The D/A conversion is performed by a circuit built on an IBM PC/XT interface breadboard placed in an expansion slot of the transmit PC. Further discussion on the evolution of the converter is given in [Ref. 12: p. 24].

C. MFM RECEIVER

As with the transmitter a combination of hardware and software are utilized in the receiver. The block diagram below depicts the flow of data.

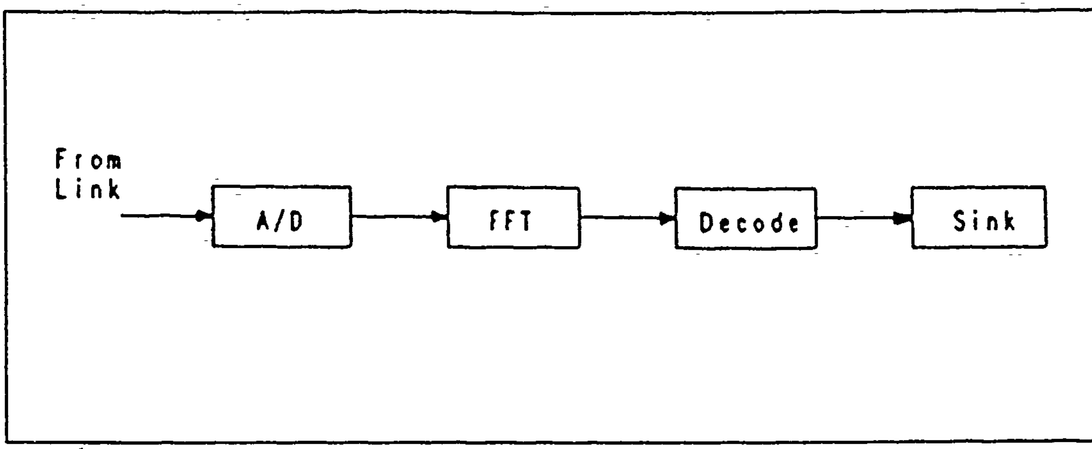

Figure 14. Block diagram of the MFM receiver.

A Metrabyte, Inc., DASH16F data acquisition board used in conjunction with various software routines accomplish 12 bit A/D conversion. The FFT and decoding process is performed by PL1250 floating point processor (FPP) board from Eighteen-Eight Laboratories supported by several Turbo Pascal subroutines. Several of these routines required minor modifications to allow the MFM output files to be properly interfaced with the facsimile. A more complete description of the correlation of these different participants along with a listing of applicable code is contained in [Ref. 12: pp. 62-76].

D. MFM SYNCHRONIZER AND FILTER.

MFM communications requires that each transmitted packet be synchronized at the receiver before it is acquired and demodulated. As such a synchronization baud of predetermined length and sample value polarities is appended to the beginning of each packet. The process of synchronizing is accomplished by a TRW programmable correlator chip. The details of implementation are discussed by Basil [Ref. 3].

Since facsimile transmissions typically travel via telephone lines a 200-3400Hz channel characteristic of the public switched telephone network was simulated. The implementation of a filter to render this simulation was carried out by Salsman and is documented in [Ref. 12: pp. 26-27].

IV. SYSTEM TESTING AND RESULTS

This chapter contains a performance evaluation of the combined facsimile and MFM system. A short section discussing the significance of phase response in optimizing system performance is followed by a section discussing the effects of using various facsimile transmission rates. The final section shows the results of system testing in terms of output signal-to-noise ratio (SNR_{out}), bit errors, and successful facsimile transmissions for both DQPSK and D16-QAM.

A. PHASE RESPONSE

The phase response for a system is essentially the difference between the phase of the transmitted signal and that of the received signal. It is important because it helps determine the proper delay for the syncbaud in order to produce the most accurate synchronization. Salsman, in Ref. 12, shows that a delay of one provides both DQPSK and D16-QAM encoding formats with minimum possibility of errors in decoding phase for a voice channel. A delay of one will be used for the research in this thesis.

B. FACSIMILE TRANSMISSION RATE

One of the objectives of this thesis is to discover how the transmission rate of the facsimile impacts the MFM system performance. As mentioned earlier the combination of MSLT and transmission rate determines the number of coded bits per scan line. Changing this number by altering either the transmission rate or the MSLT will raise or lower the number of scan lines required to fill a baud.

For the research associated with this thesis the MSLT will remain unchanged. The changes in performance associated with alterations in the facsimile transmission rate are documented in the next section.

C. BIT ERRORS AND SNR

The essence of performance for any digital communications system is a measure of the number of bits in error at the receiver. This bit error rate is directly related to both the input signal-to-noise ratio (SNR_{in}) and the size of the message (or baud in the case of MFM). An (SNR_{in}) of 22dB, typical for most American phone networks, and baud lengths of 256, 512, 1024, and 2048 along with with facsimile transmission rates of 4800 bits/sec and 9600 bits/sec were used to quantify system performance. For each combination the number of bit errors is obtained and the SNR_{out}, defined as the ratio of the mean of each of the complex multiplied adjacent tones to their respective variances, is estimated. In addition a record of the number of facsimiles successfully transmitted and received is listed.

1. Performance using DQPSK

Approximately 25,000 bits of data were DQPSK encoded for each baud length. A program developed by Basil in [Ref. 3] was used to count the number of bit errors and estimate the SNR_{out} for each combination. An effective representation of DQPSK performance is shown in Figure 15 which depicts SNR_{out} versus frequency spacing, Δf. The graph, as expected, shows improved performance when the tones are spaced closer together. This is because the nearer the tones are to each other the smaller the phase difference between them.

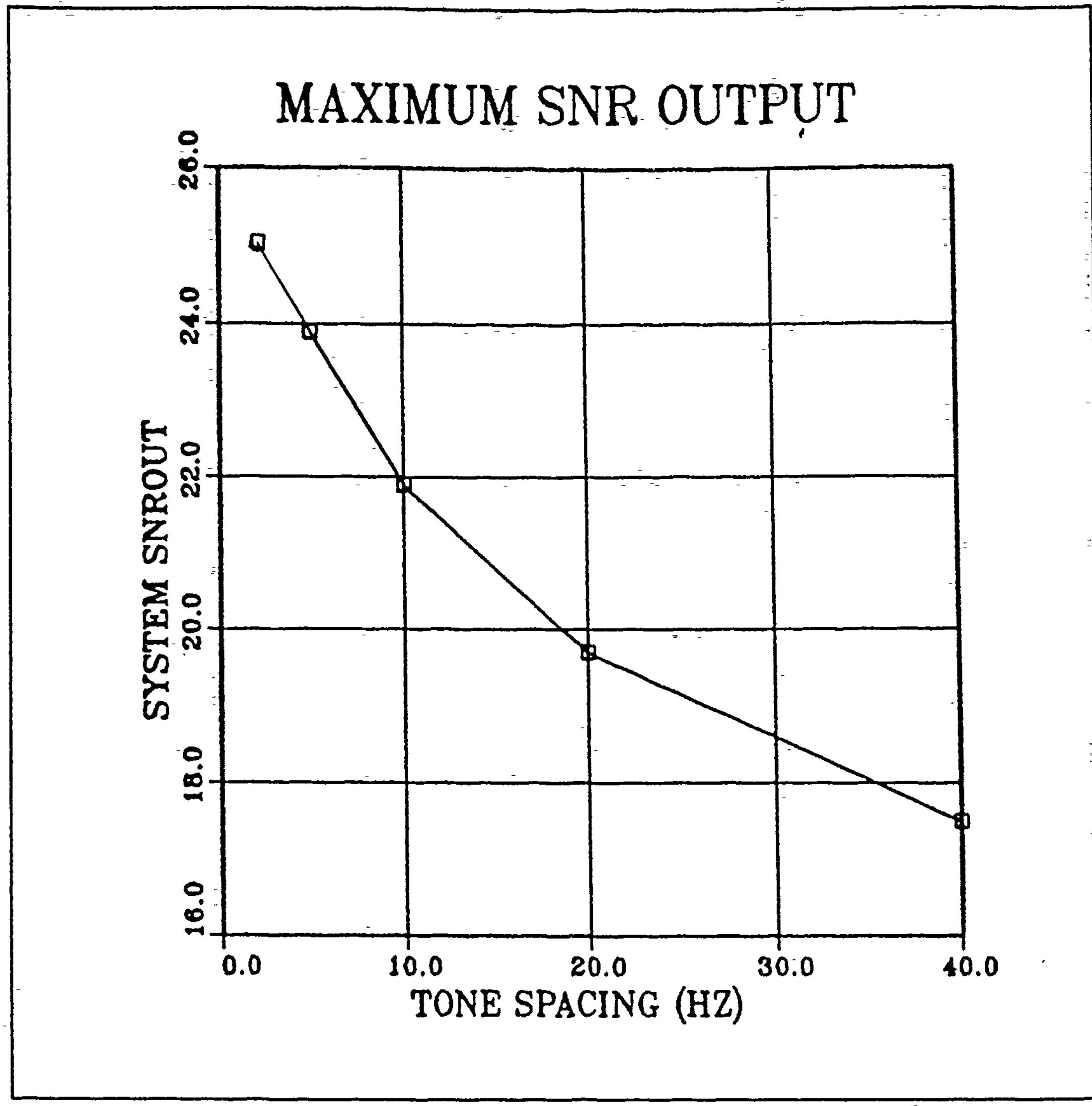

Figure 15. DQPSK system SNR output.

The bit errors registered for each baud length (frequency spacing) at various facsimile transmission rates are listed in Table 6. Ten separate facsimile encoded messages were sent at each baud length. Included in the table is the number of times, out of ten, that a legible message was printed.

Table 6. BIT ERRORS AND FAX SUCCESS IN 25,000 BITS TRANSMITTED

Δf	k_x	4800 bits/sec bit errors	4800 bits/sec successful fax	9600 bits/sec bit errors	9600 bits/sec successful fax
40	256	18	10	26	10
20	512	33	10	38	9
10	1024	60	6	67	6
5	2048	70	7	83	6
2.5	4096	92	5	110	3

In theory larger baud sizes will perform better because the tones are spaced closer together. At first glance these results do not appear to be in concert with that idea. In obtaining these results a message with the information located in the middle of the page was scanned by the facsimile machine. When processed each scan line at the beginning and end, where no printing existed, produced code that was similar. Hence the initial and final parts of the files sent via MFM were composed of repetitive symbols. The vast majority of the bit errors for the larger baud lengths occurred during the decoding of these long unbroken strings of similar characters. The smaller bauds' errors came, for the most part, from the "text" of the message. Thus as the bauds got larger the number of bit errors increased overall even though the "text" portion reflected fewer errors. It is clear that bauds larger than 512 are presently susecptible to increased errors when processing strings of like symbols. The performance of these larger bauds with consistently varying characters appears to agree with theory and mirror previous studies.

2. Performance using D16-QAM

Data was generated and analyzed for D16-QAM encoding by transmitting approximately 50,000 bits for each baud length and transmission rate combination. Another program developed by Basil in [Ref. 3] was used to count the number of bit errors.

D16-QAM has a decoding procedure similar to DQPSK in that phase differential is obtained through the complex multiplication of adjacent tones. However, in the D16-QAM this results in three possible magnitude levels vice the one obtained in DQPSK. The three possible magnitude levels are created when one of the following possible combinations of symbols from the inner (small) and/or outer (large) rings of the D16-QAM constellation occur.

- Small times small ⇒ Lowest magnitude
- Small times large ⇒ Middle magnitude
- Large times large ⇒ Highest magnitude

The lowest magnitude level is most affected by noise and interference and therefore reflects the worst performance. A higher magnitude level will produce higher performance.

The number of magnitude and phase bit errors encountered for each combination of baud length and transmission rate is shown in Table 8. Even to a larger degree than was seen with DQPSK, the performance of D16-QAM suffers from the inability to correctly decode like sequential symbols. Also similar to DQPSK, the largest share of errors in the larger bauds were confined to the "non-text" portions of the message. Noteworthy is the fact that most of the phase bits errors are one bit errors. Since the data is Gray-encoded this indicates that the received phase is only one 45 ° sector away from the transmit phase. This tables also lists the number of successful facsimile transmissions, out of ten possible, for each baud length/transmission rate combination.

Table 7. BIT ERRORS AND FAX SUCCESS IN 50,000 BITS TRANSMITTED

Baud Size	256	512	1024	2048	4096
Fax rate	**4800 bits/sec**				
Mag bit errors	190	175	162	147	182
1 bit phase errors	214	262	339	392	420
2 bit phase errors	0	0	12	0	7
3 bit phase errors	0	0	2	0	0
successful fax	10	9	5	4	3
Fax rate	**9600 bits/sec**				
Mag bit errors	212	207	185	174	196
1 bit phase errors	237	251	372	397	442
2 bit phase errors	0	0	10	0	9
3 bit phase errors	0	0	0	0	4
successful fax	10	8	4	4	2

V. CONCLUSIONS AND RECOMMENDATIONS

The MFM system, using either DQPSK or D16-QAM, performed adequately when using facsimile encoded data as a tranmission message. Overall these messages were more susceptible to errors than those composed of only ASCII characters used in earlier research. In addition, as noted previously, the problem encountered by bauds larger than 512 in differentiating and correctly decoding strings of similar characters increased bit errors significantly.

The number of successful facsimile transmissions mirrored to some degree the bit error rate. It is felt that the interface program, CAN-FAX, contributed to some of the failures as its file transfer characteristics were not originally designed for this type of exchange. Specifically, a twelve digit number as the first entry in a file identified that file to facsimile as one that could be read and printed. Any errors in that number caused that entire transmission to fail.

The use of different transmission rates was somewhat significant. The messages sent from the facsimile to the transmit PC at 4800 bits/sec had consistently fewer errors upon receipt at the receive PC. Further study using variable MSLTs would provide more insight into the most efficient combination for use with MFM.

As noted in previous studies, the BER for DQPSK is lower than that of D16-QAM for every case of baud-length tested. This advantage is particularly important when a switched telephone netwc k that generally has higher noise levels, such as overseas lines is used. The D16-QAM format, because it carries

4 bits per tone per baud (two more than DQPSK), has a higher throughput capability which is advantageous when using telephone networks with guaranteed maximum noise levels, such as the private systems.

Areas of further study should include an effort to modify the synchronization board to allow multiple packets to be transmitted. The present research system requires manual intervention to reset the system sync, limiting each transmission to one packet. Use of a facsimile card in a PC expansion slot, vice a separate facsimile machine, should be considered for testing purposes. Additionally, alterations to existing Pascal software to include error control coding to improve BERs is recommended. The groundwork for such an effort has already been laid by LT Robert W. Ives [Ref. 13]. Finally, a facsimile incorporating a modem with an MFM encoding format should be designed and built to provide a complete testing opportunity.

APPENDIX

The following figures give examples of how facsimile two-dimensional coding works for each of the different modes at various locations on the scan line.

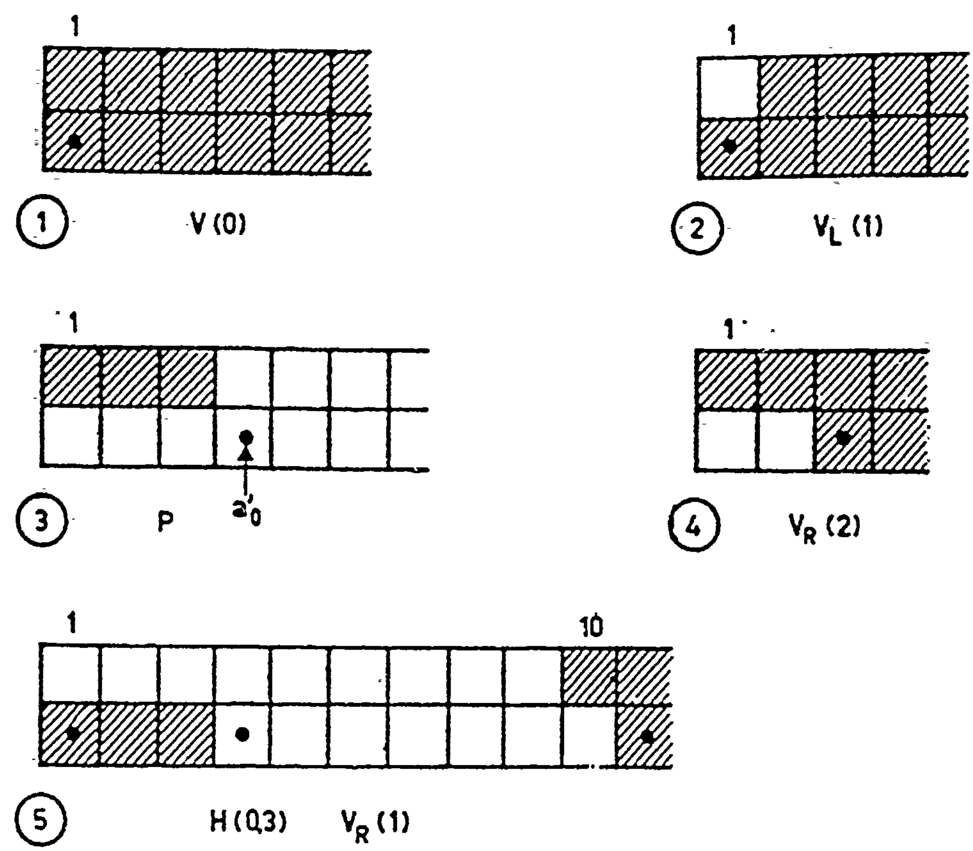

Figure 16. Coding at the beginning of the scan line: (from Ref. 11: p. 27).

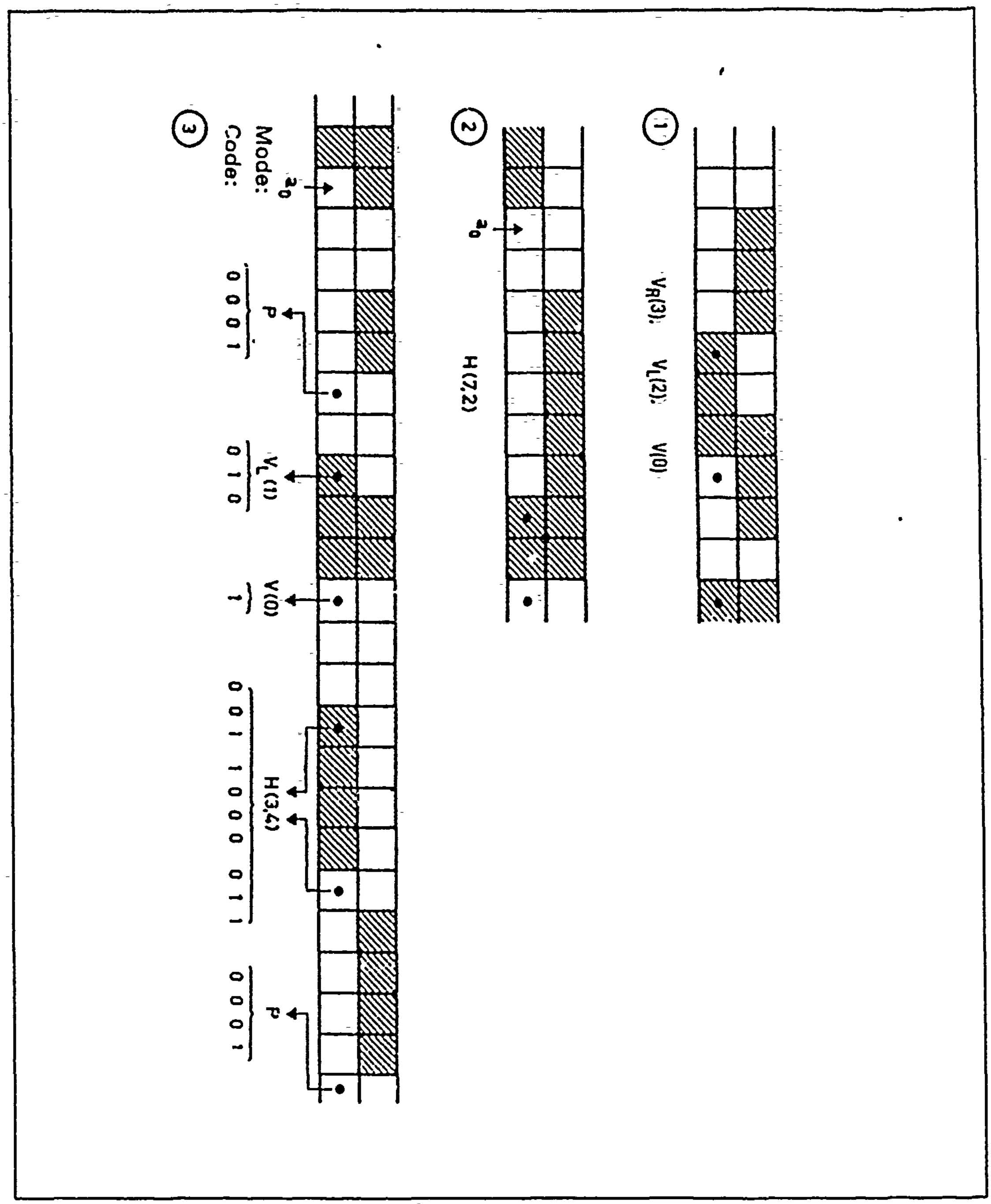

Figure 17. Coding in the middle of the scan line: (from Ref. 11: p. 28).

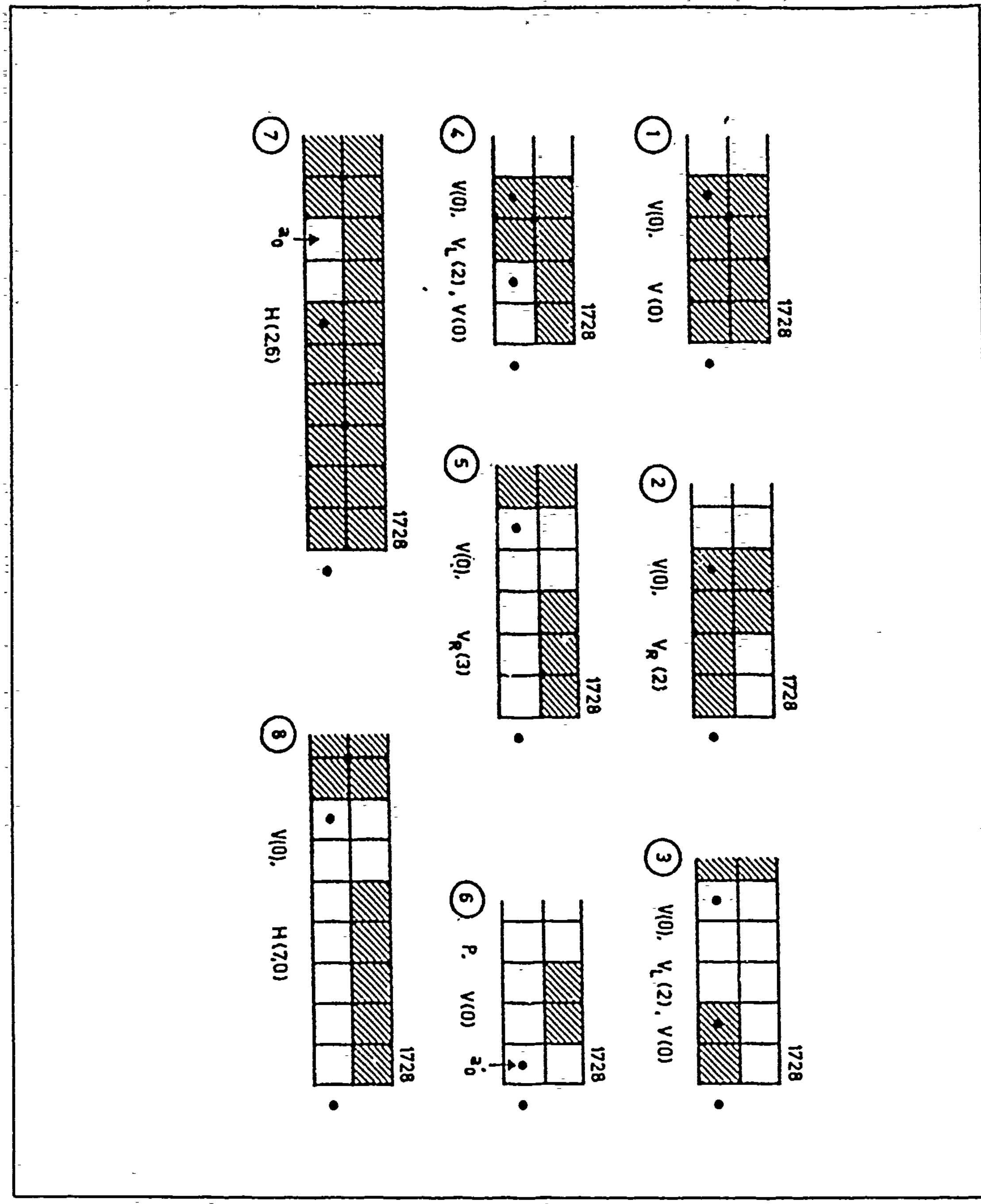

Figure 18. Coding at the end of the scan line: (from Ref. 11: p. 27).

LIST OF REFERENCES

1. Terry K. Gantenbein, "Implementation of multi-frequency modulation on an industry standard computer," Master's Thesis, Naval Postgraduate School, Monterey, California, September 1989

2. Paul H. Moose, "Theory of multi-frequency modulation (MFM) digital communications," Technical Report No. NPS-62-89-019, Naval Postgraduate School, Monterey, California, May 1989

3. Peter G. Basil, "Real time multi-frequency modulation using differentially encoded signal constellations," Master's Thesis, Naval Postgraduate School, Monterey, California, June 1990

4. Roy Hunter & A. Harry Robinson,"International digital facsimile coding standards," *Proceedings of the IEEE*, Vol. 68, No. 7, pp. 854- 867, 1980

5. Bernard Sklar, *Digital Communications; Fundamentals and Applications*, Prentice Hall, New Jersey, 1988

6. N. S. Jayant & Peter Noll, *Digital Coding of Waveforms; Principles and Applications to Speech and Video*, Prentice-Hall Inc., New Jersey, 1984

7. A. J. Frank, "Uniform decoding of minimum-redundancy codes," U.S. Patent 3 883 847, May 13, 1975

8. U. Rothgordt, "Localization of disturbances in digitally transmitted facsimile pictures," presented at the 1979 Picture Coding Symp., Ipswich, England, July 9-11

9. Y. Eto, Y. Hirano, & T. Fukinuki, "Facsimile signal decoding which localizes the effect of run-length code errors," *IEEE Trans. Commun.*, Vol. COM-25, pp. 1170-1173, October 1977

10. Japan, "Proposal for draft recommendation of two-dimensional coding scheme ," CCITT SGXIV Doc. No. 42, July 1978

11. The International Telegraph and Telephone Consultative Committee (CCITT) Red Book Vol. X, 8th Plenary Assembly, Malaga-Torremolinos, 1984

12. Charles P. Salsman, "Application of multi-frequency modulation (MFM) for high-speed data communications to a voice frequency channel," Master's Thesis, Naval Postgraduate School, Monterey California, June 1990

13. Robert W. Ives, "Error control coding for multi-frequency modulation," Master's Thesis, Naval Postgraduate School, Monterey, California, June 1990