1-1-1975

Joint detection and decoding of convolutional codes for channels with intersymbol interference.

Munaf A. Sattar

JOINT DETECTION AND DECODING OF CONVOLUTIONAL CODES FOR CHANNELS WITH INTERSYMBOL INTERFERENCE

by

Munaf A. Sattar

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

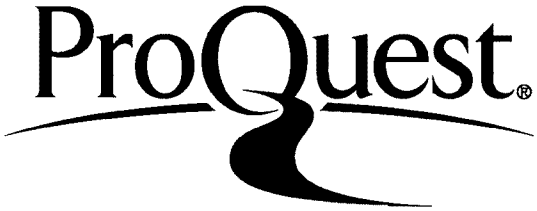
Master of Science

in

Electrical Engineering

Lehigh University

1974



This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

December 1, 1975
(date)

Professor in Charge

Chairman of Department

ACKNOWLEDGEMENTS

I am grateful to my advisor, Prof. Bruce D. Fritchman, for inspiring me to undertake this project and for his encouragement. I wish also to express sincere thanks to Prof. Joseph C. Mixsell, who spent many hours guiding me throughout the course of this work.

TABLE OF CONTENTS

	Page
List of Figures	v
Abstract of the Thesis	1
CHAPTER 1 - INTRODUCTION	2
CHAPTER 2 - COMMUNICATION SYSTEM MODELS	4
2.1 Sequential Detector	4
2.2 Sub-optimum System	7
2.3 Optimum System	11
CHAPTER 3 - SIMULATION RESULTS	16
CHAPTER 4 - SUMMARY AND CONCLUSIONS	20
References	21
Appendix A - Sequential Detector Simulation	22
Appendix B - Sub-optimum System Simulation	32
Appendix C - Optimum System Simulation	45
Vita	52

LIST OF FIGURES

Figure 1.1 - General Block Diagram of a Digital Communication System

Figure 2.1 - Communication System without Encoder, Decoder

Figure 2.2 - Communication System with Sub-optimum Receiver

Figure 2.3 - Encoder for a Rate 1/n Convolutional Code of Memory Order ν

Figure 2.4 - Binary Symmetric Channel

Figure 2.5 - Communication System with Optimum Receiver

Figure 3.1 - Sequential Detector Performance

Figure 3.2 - Performance of Sub-optimum and Optimum Receivers

Figure A.1 - Definitions and Initial Conditions

Figure A.2 - Start-up Procedure and Calculation of Discrete Channel Outputs

Figure A.3 - Computation of Received Signal for Generated Symbol

Figure A.4 - Calculation and Normalization of Joint Probabilities

Figure A.5 - Start-up for Recursive Rule

Figure A.6 - Calculation of 'Old Probabilities' and Shifting of Symbols

Figure A.7 - Decision Segment

Figure A.8 - Formation of Two-Dimensional Binary Array

Figure B.1 - Definitions

Figure B.2 - Main Program

Figure B.3 - Calculation of Discrete Encoded Symbols

Figure B.4 - Calculation of Discrete Channel Outputs

Figure B.5 - Generation and Encoding of Source Symbols

Figure B.6 - Shift-Registers and Transmitter

Figure B.7 - Calculation of 'New' and 'Old' Probabilities for Channel Symbols

Figure B.8 - Decision Segment for Detector

Figure B.9 - Completion of Detector Subroutine

Figure B.10- Calculation of 'New' and 'Old' Probabilities for Source Symbols

Figure B.11- Decision Segment for Decoder

Figure C.1 - Definitions

Figure C.2 - Start-up Procedure and Symbol Generation

Figure C.3 - Encoder and Transmitter

Figure C.4 - Calculation of 'New' and 'Old' Probabilities

Figure C.5 - Decision Segment

Figure C.6 - Formation of Binary Array and Calculation of Discrete Channel Outputs

ABSTRACT

A new algorithm, based on the sequential compound detector, is derived in this thesis for the joint detection and decoding of information transmitted at high speed through intersymbol interference channels using convolutional encoding. Performance results, obtained using computer simulation, show the 'joint' algorithm to considerably outperform the separate detection and decoding procedure, without a significant increase in computational complexity. In addition, the hardware requirements for the joint detector-decoder are substantially less than those for the separate detector-decoder for short constraint lengths.

CHAPTER 1

INTRODUCTION

This thesis examines the design of receivers for communication systems with convolutional encoders and intersymbol interference channels. Much research has been done in recent years for handling intersymbol interference with various types of codes and receiver structures, but most of it has been directed at the development of sub-optimum schemes. This work introduces a new scheme which is optimum, in the sense that the receiver minimizes the average probability of error for the joint detection-decoding problem. The new scheme is a direct extention of the sequential detector algorithm originally proposed by Abend and Fritchman [1].

Consider the typical "sub-optimum" receiver shown in Fig. 1.1, composed of a separate detector and a separate decoder. Most researchers up to now have aimed at improving the performance of these individual components. The use of two components for decision-making purposes, however, is clearly redundant and inefficient, as potentially useful information is lost each time an intermediate decision is made. We, therefore, propose a receiver which makes joint detection and decoding possible, and as such has to make only one hard decision for each source symbol. We call this the "optimum receiver".

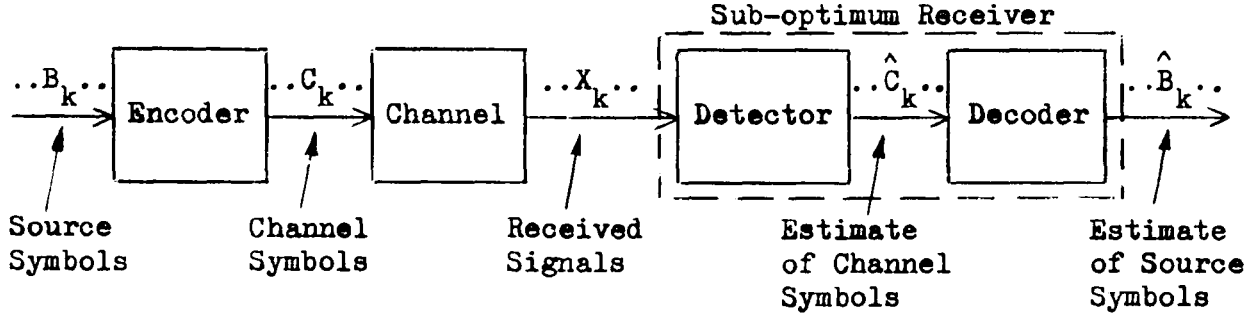


Fig. 1.1 General Block Diagram of a Digital Communication System

Chapter 2 begins with a review of the sequential detector algorithm. This is followed by a discussion of the 'sub-optimum system', i.e. the entire communication system composed of the encoder, the channel, and sub-optimum receiver. The chapter concludes with a discussion of the optimum receiver.

In Chapter 3, computer simulation results are summarized, and a comparison of the performance of the sub-optimum and optimum receivers is made for various delay constraints. Finally, Chapter 4 presents some conclusions pertaining to the future course of communication systems design.

Flowcharts of computer programs used for simulating the sequential detector, the sub-optimum system, and the optimum system are included in the appendices.

CHAPTER 2

COMMUNICATION SYSTEM MODELS

We begin this chapter with a review of the sequential detector algorithm, first described by Abend and Fritchman, which forms the basis for this work. This is followed by a description of the sub-optimum and optimum system models.

2.1 Sequential Detector

The reduced communication system of interest is shown in Fig. 2.1. In this figure, the encoder and decoder of Fig. 1.1 have been omitted so as to focus attention on the detector. Further, the channel has been modeled to consist of a known finite memory part followed by a gaussian noise source. Many practical channels exhibit such finite memory characteristics, e.g. multipath in radio channels, dispersion in scatter channels, and amplitude and delay distortion in telephone channels. The assumption of additive gaussian noise is also satisfied for most such channels.

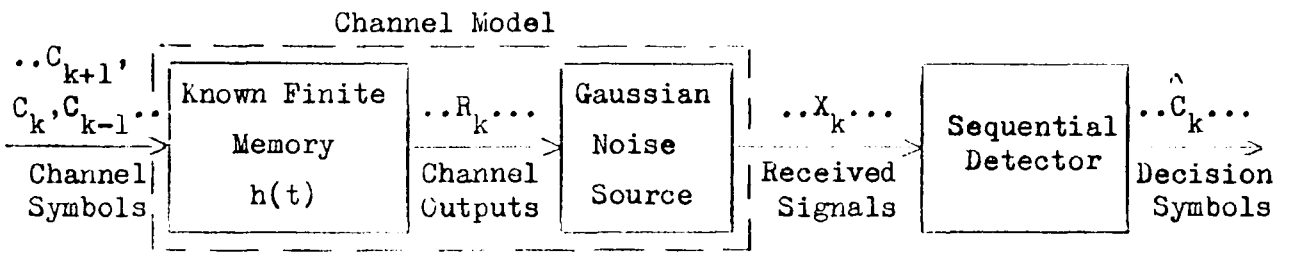


Fig. 2.1 Communication System without Encoder-Decoder

In Fig. 2.1, let C_k denote the $k^{\underline{th}}$ symbol transmitted through the channel at time $t=k\tau$, where τ is the pulse period and k is a non-negative integer, $k=1,2,\ldots$. Assume these input channel symbols are independent, equilikely, and drawn from the set $\{1,-1\}$.

The output of the known finite memory part of the channel, of constraint length L, then depends deterministically on L of the most current channel symbols, C_k, C_{k-1} ,·····, C_{k-L+1} . Inter-symbol interference is present for $L>1$. If the impulse response of the channel is represented by the terms h_0, h_1,·····, h_{L-1}, then the channel output is given by:

$$R_k = C_k h_0 + C_{k-1} h_1 + \cdots\cdots + C_{k-L+1} h_{L-1} \qquad (2.1.1)$$

The real output next passes through the noisy memoryless part of the channel. In most communication channels, noise is independent of the channel inputs, and can be modeled to be independent at each time. Thus, assuming random noise components N_k to be additive white gaussian noise, the received signal during the $k^{\underline{th}}$ interval becomes:

$$X_k = R_k + N_k \qquad (2.1.2)$$

The communication problem, stated simply, is to detect the transmitted symbol C_k from the received information X_1, X_2,···, X_{k+D}, where D is a finite delay, and X_{k+D} is the signal received during the $k+D^{\underline{th}}$ time interval. The reason for making the decision after a lag of D time intervals is to minimize the degradation in performance due to possibly premature decisions. We shall now restate the equations derived in [1] for the case $D \leq L-1$, though the results are easily extended to the case $D > L-1$.

The sequential detector, at each time interval, computes the two probabilities:

$$P(C_k = c \,|\, X_1, \cdots, X_{k+D}) \quad , \qquad c \in \{-1, 1\} \qquad (2.1.3)$$

and selects that value of $\hat{C}_k = c$ which maximizes (2.1.3). Because $p(X_1, \cdots, X_{k+D})$ is a proportionality factor, it suffices to compute the probabilities $p(C_k, X_1, \cdots, X_{k+D})$. i.e.

$$P(C_k \,|\, X_1, \cdots, X_{k+D}) = p(C_k, X_1, \cdots, X_{k+D}) / p(X_1, \cdots, X_{k+D}) \qquad (2.1.4)$$

Therefore the decision rule is equivalent to choosing the value of C_k as the transmitted value that maximizes $p(C_k, X_1, \cdots, X_{k+D})$. These two probabilities can be computed if the 2^L joint probabilities of the form $p(C_{k+D}, \cdots, C_k, \cdots, C_{k+D-L+1}, X_1, \cdots, X_{k+D})$ are known by summing over the appropriate information symbols, i.e.

$$p(C_k, X_1, \ldots, X_{k+D}) = \sum_{C_{k+D}} \cdots \sum_{C_{k+1}} \sum_{C_{k-1}} \cdots \sum_{C_{k+D-L+1}} \cdot p(C_{k+D}, \ldots, C_k, \ldots, C_{k+D-L+1}, X_1, \ldots, X_{k+D}) \tag{2.1.5}$$

It is shown in [1] that the 2^L joint probabilities can be computed recursively using the following rule:

$$p(C_{k+D}, \ldots, C_k, \ldots, C_{k+D-L+1}, X_1, \ldots, X_{k+D}) = P(C_{k+D}) \cdot p(X_{k+D} | C_{k+D}, \ldots, C_{k+D-L+1}) \cdot \sum_{C_{k+D-L}} \cdot p(C_{k+D-1}, \ldots, C_{k+D-L}, X_1, \ldots, X_{k+D-1}) \tag{2.1.6}$$

Note that this expression is the product of three factors. The first factor is the apriori probability of transmitting symbol C_{k+D}, and is assumed to be equal to $\frac{1}{2}$. The second factor is one of 2^L probability densities which can be computed from a knowledge of X_{k+D} and the 2^L mappings of the channel outputs, R_i, corresponding to all possible L-length symbol sequences. If $f(\cdot)$ is the probability density of the noise component N_k, then according to eqn. (2.1.2), we get:

$$p(X_{k+D} | C_{k+D}, \ldots, C_{k+D-L+1}) = p(X_{k+D} | R_i) = f(X_{k+D} - R_i), \quad \text{where } i = 1, 2, \ldots, 2^L \tag{2.1.7}$$

The form of binary sequences to be used in calculating R_i, in accordance with eqn. (2.1.1), is outlined at the end of Appendix A.

The third factor in eqn. (2.1.6) is one of 2^{L-1} probabilities determined by a summation of joint probabilities during the preceding time interval. It summarizes all the required information about the received signals preceding X_{k+D}.

2.2 Sub-Optimum System [1]

The sub-optimum system shown earlier in Fig. 1.1 is redrawn below with some minor modifications. Since the channel and the sequential detector have already been described, the part of Fig. 2.2 enclosed by dashed lines will be treated as a binary channel, and emphasis will be placed on the operation of the encoder and the decoder.

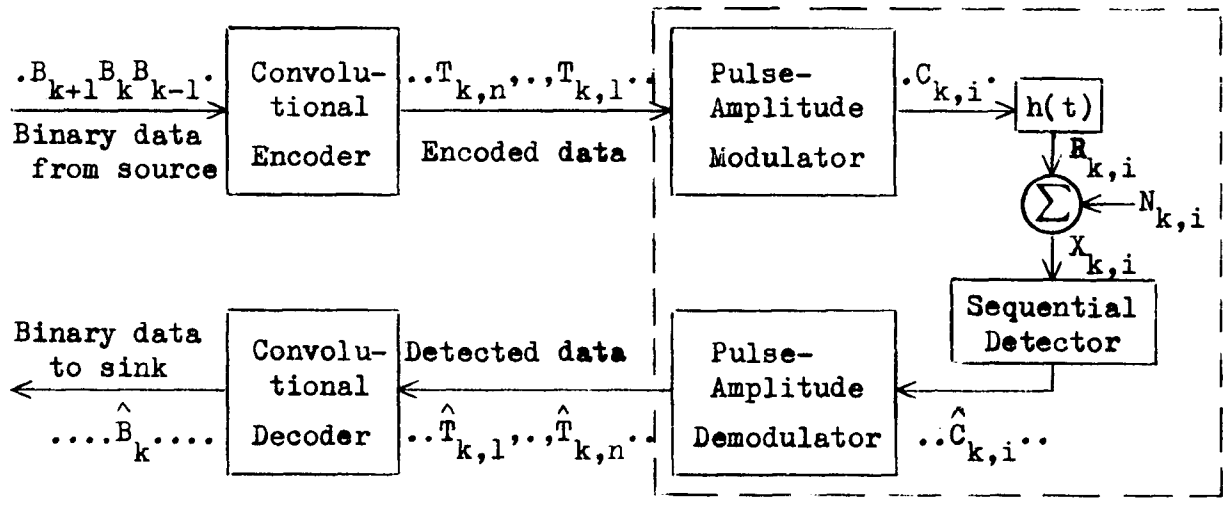


Fig. 2.2 Communication System with Sub-optimum Receiver

The reason for using an encoder and a decoder in the communication system is to further improve the reliability of information transmission. Since Shannon [2] first demonstrated that this is

---

[1] In this section and the next, the word symbol will be used to refer to a binary vector (or n-tuple), and the word digit to refer to a binary scalar. In the rest of the thesis, however, 'symbol' is used interchangeably for both scalars and vectors, its meaning being clear from context.

possible, many different types of coding techniques have been investigated. In this work, we restrict ourselves to the use of convolutional codes, whose decoding has been shown to be equivalent to the detection of digital sequences in the presence of intersymbol interference [3]. Recently, Mixsell [4] has investigated the decoding of convolutional codes using the sequential detector algorithm, and has developed some of the properties of the 'sequential decoder'. Equations describing this decoder are reviewed toward the end of this section.

Referring to Fig. 2.2, let us assume that the binary digits or 'bits' B_k generated by the information source are drawn from the set $\{0,1\}$, are equally likely, and are statistically independent of each other. The information bits are first fed through a convolutional encoder, whose output is then passed through a modulator to obtain channel symbols for transmission over the channel.

The convolutional encoder is a coding device with memory ν, which transmits the n-tuple $\underline{T}_k = (T_{k,1}, \ldots, T_{k,n})$ corresponding to each source bit B_k that is shifted into the encoder register at time $k\tau$, where $1/n$ is the rate of the convolutional code. (n is assumed to be an integer.) Clearly, the n-bit encoder output at time $k\tau$ depends on B_k as well as the previous $\nu-1$ source bits. The mapping of the source sequence $\{B_k\}$ onto the encoder output sequence $\{T_k\}$ is specified by the choice of a convolutional code generator matrix, G:

$$G = \begin{bmatrix} g_{1,1}, & g_{1,2}, & \cdots, & g_{1,n} \\ g_{2,1}, & g_{2,2}, & \cdots, & g_{2,n} \\ \vdots & \vdots & & \vdots \\ g_{\nu,1}, & g_{\nu,2}, & \cdots, & g_{\nu,n} \end{bmatrix} \qquad (2.2.1)$$

The elements g_{ij} in this matrix are either "1" or "0". In implementation, these elements actually represent connections, or lack of connections, between the ν shift register stages and the n modulo-two adders of the encoder, as shown in Fig. 2.3.

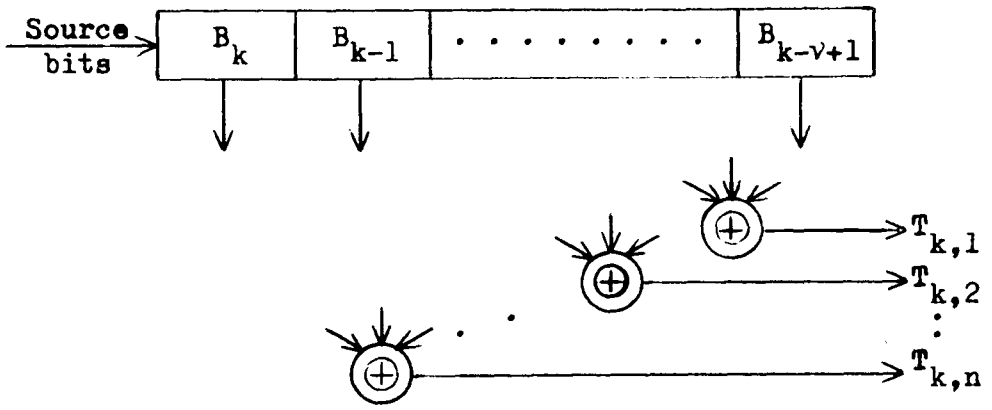


Fig. 2.3 Encoder for a rate 1/n convolutional code of memory order ν

If $B_k,\ldots,B_{k-\nu+1}$ represents the "state" of the encoder at time $k\tau$, the encoder output sequence of binary digits contained in the symbol $\underline{T}_k$ is given by:

$$\begin{aligned} T_{k,1} &= g_{1,1}\, B_{k-\nu+1} \oplus \cdots \oplus g_{\nu,1}\, B_k \\ T_{k,2} &= g_{1,2}\, B_{k-\nu+1} \oplus \cdots \oplus g_{\nu,2}\, B_k \\ &\vdots \\ T_{k,n} &= g_{1,n}\, B_{k-\nu+1} \oplus \cdots \oplus g_{\nu,n}\, B_k \end{aligned} \qquad (2.2.2)$$

It should be observed that the encoder output bit $T_{k,i}$ depends on the encoder 'state' and on the elements of the $i^{\underline{th}}$ column in G.

To obtain the channel symbol $\underline{C}_k = (C_{k,1}, C_{k,2},\ldots, C_{k,n})$ corresponding to $\underline{T}_k$, the encoder output bits are passed through a pulse-amplitude modulator, which sets $C_{k,i} = -1$ if $T_{k,i}=0$, and $C_{k,i} = 1$ if $T_{k,i} = 1$. These channel digits are then serially transmitted through the channel.

The transmission of channel digits over the channel and their estimation at the detector were described in the preceding section. We now treat the channel as a binary symmetric channel, as shown,

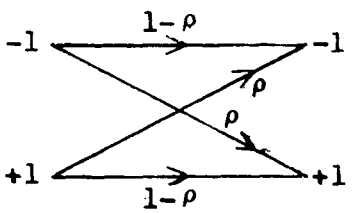


Fig. 2.4 Binary Symmetric Channel

and assume that the channel digits are detected with a bit-error probability of ρ.

After the channel digits are estimated by the sequential detector, they are fed into a pulse-amplitude demodulator, which converts them back to 0's and 1's. Thus, at time $k\tau$, the input at the decoder is the binary n-tuple $\hat{\underline{T}}_k = (\hat{T}_{k,1},\ldots,\hat{T}_{k,n})$. From the sequence of these symbols, we must obtain an estimate of the source bits using the sequential decoder.

The equations which describe the sequential decoder are very similar to those that describe the sequential detector, and as such are reviewed here very briefly. We let δ designate the decoder delay, and consider the case $\delta \leq \nu-1$. As shown in [4], the 2^ν joint probabilities that need to be calculated at the decoder are given by the recursive rule:

$$p(B_{k+\delta},\ldots,B_k,\ldots,B_{k+\delta-\nu+1},\hat{\underline{T}}_1,\hat{\underline{T}}_2,\ldots,\hat{\underline{T}}_{k+\delta}) = P(B_{k+\delta})\cdot p(\hat{\underline{T}}_{k+\delta}|B_{k+\delta},\ldots,B_{k+\delta-\nu+1}) \sum_{B_{k+\delta-\nu}} \cdot p(B_{k+\delta-1},\ldots,B_{k+\delta-},\hat{\underline{T}}_1,\ldots,\hat{\underline{T}}_{k+\delta}) \tag{2.2.3}$$

Upon a comparison of (2.2.3) with (2.1.6), the similarity between the two expressions is readily apparent. The first and third factors in these expressions have almost identical interpretations. The second factor in eqn. (2.2.3), however, is one of 2^ν incremental probabilities associated with the binary n-tuple $\hat{\underline{T}}_{k+\delta}$ and the 2^ν distinct encoder 'states'--to each of which corresponds a discrete encoded binary n-tuple $\underline{T}_i$. These incremental probabilities are calculated as follows:

$$p(\hat{\underline{T}}_{k+\delta}\,|\,B_{k+\delta},\ldots,B_{k+\delta-\nu+1}) = P(\hat{\underline{T}}_{k+\delta}\,|\,\underline{T}_i) = \prod_{j=1}^{n} p(\hat{T}_{k+\delta,j}|T_{i,j}) \qquad \text{where } i=1,2,\ldots,2^\nu \tag{2.2.4}$$

The above equation factors as it does because the noise sequence is assumed to act independently on the succeeding digits of the

transmitted symbols, and hence the digits themselves may be assumed to be independent of each other. The incremental probabilities are thus defined directly in terms of the bit-error probability ρ.

The decision at the decoder is made in the same way as it was at the detector, i.e. by summing the joint probabilities calculated in eqn. (2.2.3) for the two different values of B_k, and comparing the two sums.

In conclusion, we point out a fact that is useful in comparing the performance of the sub-optimum and optimum receivers. The lag d_{eff}, or effective delay, between the time a bit is transmitted by the source and time a decision is made on it by the decoder is related to the detector and decoder delays as follows:

$$d_{eff} = \delta + \left\lceil \frac{D}{n} \right\rceil \qquad (2.2.5)$$

where $\left\lceil \frac{D}{n} \right\rceil$ represents the smallest integer $\geq \left(\frac{D}{n}\right)$. The logic behind eqn. (2.2.5) becomes clear upon a bit-by-bit examination of the algorithm. A delay of δ time intervals at the decoder requires δ additional source bits, or equivalently δ additional encoded symbols, to be transmitted and detected. A delay of D digits at the detector, however, requires at most $\lceil D/n \rceil$ additional encoded symbols to be transmitted. Hence, effective delay represents the additional number of source bits that need to be transmitted to account for delays at both the detector and the decoder, before a final decision can be made on a particular source bit.

2.3 Optimum System

In this section, we present the algorithm[1] for the joint detector-decoder, or the optimum receiver. The communication system model of interest is again illustrated in Fig. 2.5.

---

[1] This algorithm was developed jointly by the author and his advisors, Professors B.D.Fritchman and J.C.Mixsell.

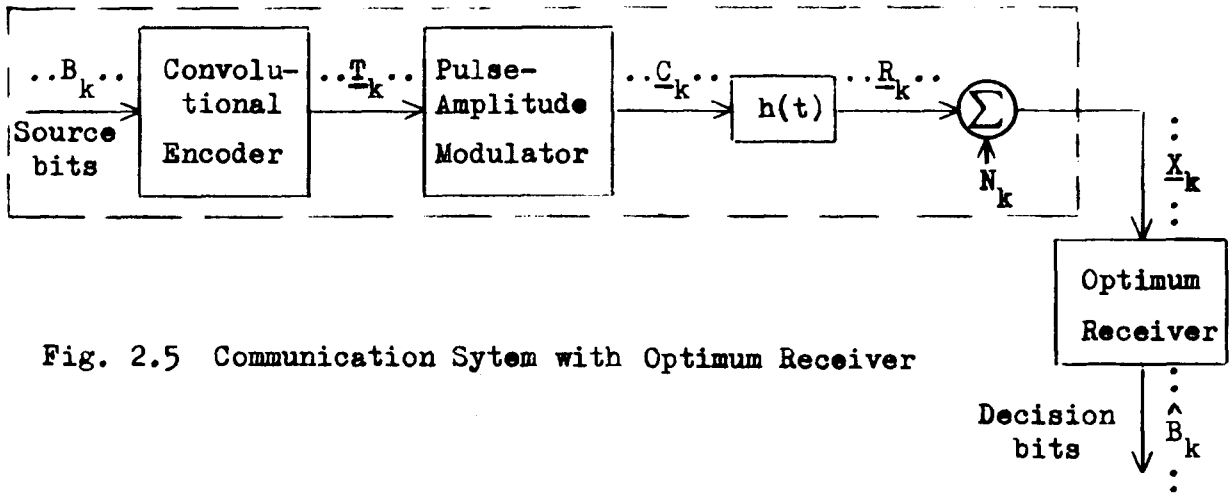


Fig. 2.5 Communication Sytem with Optimum Receiver

The portion of Fig. 2.5 enclosed by dashed lines has been given extensive treatment in the last two sections. We begin here with the assumption that the binary data from the source is transmitted through the encoder, the modulator, and the channel in the same way as it was for the sub-optimum system, and is received at the input to the optimum receiver as a sequence of real valued n-tuples $\{\underline{X}_k\}$. Thus, during the $k^{\underline{th}}$ time interval, the received signal is $\underline{X}_k = (X_{k,1},\ldots,X_{k,n})$. Our object now is to directly estimate the source bits B_k from these received signals.

Before presenting the optimum receiver algorithm, we point out that whereas a decision was made corresponding to each signal $X_{k,i}$ at the detector in the sub-optimum system, the decision procedure at the optimum receiver is not begun until the entire n-tuple $\underline{X}_k$ is received. This inherently removes an information loss associated with intermediate decisions.

The optimum receiver decision process is in a sense similar to the procedure used by the sequential decoder, though we are now working with an n-tuple of real numbers instead of an n-tuple of binary digits. Looked at another way, the optimum receiver algorithm is an extention of the sequential detector algorithm from a scalar- to a vector-field. With these thoughts in mind, and

upon inspection of equations (2.1.6) and (2.2.3) for the detector and the decoder, respectively, we can directly write the recursive rule defining the joint probabilities for the optimum receiver as follows. (The derivation of this equation is analogous to the derivation of eqn. (21a) in [1], and of eqn. (3.09) in [4].)

$$\begin{aligned} p(B_{k+d},\ldots,B_{k+d-\ell+1}, \underline{X}_1,\ldots,\underline{X}_{k+d}) &= p(\underline{X}_{k+d}|B_{k+d},\ldots,B_{k+d-\ell+1}, \underline{X}_1,\ldots,\underline{X}_{k+d-1}) \\ &\cdot p(B_{k+d},\ldots,B_{k+d-\ell+1}, \underline{X}_1,\ldots,\underline{X}_{k+d-1}) \\ &= p(\underline{X}_{k+d}|B_{k+d},\ldots,B_{k+d-\ell+1}) \\ &\cdot P(B_{k+d})\cdot p(B_{k+d-1},\ldots,B_{k+d-\ell+1}, \underline{X}_1,\ldots,\underline{X}_{k+d-1}) \\ &= P(B_{k+d}).p(\underline{X}_{k+d}|B_{k+d},\ldots,B_{k+d-\ell+1}) \sum_{B_{k+d-\ell}} \\ &\cdot p(B_{k+d-1},\ldots,B_{k+d-\ell+1},B_{k+d-\ell}, \underline{X}_1,\ldots,\underline{X}_{k+d-1}) \end{aligned} \tag{2.3.1}$$

The above expression assumes statistical independence of symbols, and also assumes the optimum receiver delay d--between the transmission of, and decision on, a source bit B_k-- to be less than the effective constraint length ℓ of the code and the channel combined, which is given by:

$$\ell = \nu + \left\lceil \frac{L-1}{n} \right\rceil \tag{2.3.2}$$

where $\left\lceil \frac{L-1}{n} \right\rceil$ is the smallest integer $\geq \left(\frac{L-1}{n}\right)$. Eqn. (2.3.2) is based on a logic similar to that used for explaining eqn. (2.2.5), except that we are now dealing with the system's constraint lengths. The integer ℓ represents the maximum number of source bits that influence the received signal at any time, and hence completely incorporates the past history of the encoder and the channel in the received signal.

The form of eqn. (2.3.1) should by now be familiar. The first and third factors have almost the same interpretations as they did in eqn. (2.1.6) for the detector, or eqn. (2.2.3) for the decoder.

The most interesting part of this expression is again the second factor, which represents one of 2^{ℓ} probabilities computed from a knowledge of $\underline{X}_{k+d}$ and the 2^{ℓ} discrete n-tuples $\underline{R}_i=(R_{i,1},\ldots,R_{i,n})$ corresponding to all possible ℓ-length source bit sequences. An explanation of this factor follows a description of the equations which describe $\underline{R}_i$.

To calculate the channel outputs $\underline{R}_i$, each ℓ-length source bit sequence, $B_{\ell},\ldots,B_1$, must first be encoded and converted into a chain of channel digits: $C_{\ell,n},\ldots,C_{\ell,1},C_{\ell-1,n},\ldots,C_{\ell-1,1},\ldots,C_{\nu,n},\ldots,C_{\nu,1}$. Observe that calculation of each $\underline{R}_i$ is based upon a knowledge of the past history of the encoder with memory ν as well as of the channel with memory L. Also note that there are at least L-1+n such digits, i.e.

$$(\ell-\nu+1)\cdot n = \left[\frac{L-1}{n}\right]\cdot n + n \geq L-1+n$$

to be utitlized for calculating the n outputs of the intersymbol interference channel with memory L, corresponding to each source symbol sequence.

From these channel digits, the elements of $\underline{R}_i$ n-tuples are determined as follows:

$$\begin{aligned}
R_{i,n} &= h_0C_{\ell,n} + h_1C_{\ell,n-1} + \cdots \\
&\vdots \\
R_{i,j} &= h_0C_{\ell,j} + \cdots \qquad j=1,2,\ldots,n \qquad (2.3.3) \\
&\vdots \\
R_{i,1} &= h_0C_{\ell,1} + h_1C_{\ell-1,n} + h_2C_{\ell-1,n-1} + \cdots
\end{aligned}$$

where the last term in each equation is a product of h_{L-1} and the (L-1) $\underline{\text{th}}$ channel digit preceding $C_{\ell,j}$ in the chain of channel digits.

The steps outlined above for the calculation of discrete channel outputs $\underline{R}_i$ from distinct source symbol sequences $\{B_i\}$

are easily generalized to the calculation of $\underline{R}_k$ corresponding to a source bit sequence $\{B_k\}$ at time $k\tau$.

Returning to an explanation of eqn. (2.3.1), the probabilities in the second factor are calculated as follows:

$$p(\underline{X}_{k+d}|B_{k+d},\ldots,B_{k+d-\ell+1}) = p(\underline{X}_{k+d}|\underline{R}_i) = \prod_{j=1}^{n} p(X_{k+d,j}|R_{i,j})$$

$$= \prod_{j=1}^{n} f(X_{k+d,j} - R_{i,j}) \qquad \text{where } i=1,2,\ldots,2^{\ell}$$

(2.3.4)

The logic behind the factoring of this expression is the same as that for the factoring of eqn. (2.2.4). Since the noise in the channel is assumed to act independently upon each digit, the succeeding digits may themselves be assumed to be independent of one another.

This concludes the discussion of the various communication system models and their algorithms. Flowcharts of computer programs that were used to simulate the sequential detector, the sub-optimum system, and the optimum system are presented in the appendices. The next chapter contains a summary of the simulation results.

CHAPTER 3

SIMULATION RESULTS

The simulation results obtained using the programs of the three appendices are now presented. Though several different channels and codes were used to test the programs and to compare the performance, only one set of curves is presented here as an illustration.

The intersymbol interference channel that was used is shown in Fig. 3.1. This channel is quite similar to the channel used for illustration purposes in [1]. The transmitted symbols were in all cases binary, and were generated using a uniform-distribution random number generator. The noise samples were obtained from an independent Gaussian-distribution number generator.

Fig. 3.1 shows the detector performance for different values of the delay constraint, or "look-ahead", D. The performance criterion is the bit-error probability, P_e, as a function of the signal-to-noise ratio in dB. The greatest improvement in performance is clearly achieved when D is increased from 0 to L-1, where L=5. This is expected since the symbol that is being estimated at any time does not affect more than L-1 succeeding signals. There is very little improvement for $D > L-1$, and it is not really worth the cost of added computational complexity. These performance results agree reasonably well with those obtained in [1].

The core of the results, as far as this thesis is concerned, is presented in Fig. 3.2, in which the performance of the sub-optimum and optimum receivers is compared. The channel impulse response is the same as before (L=5), and the convolutional code represented by G is a one-third rate code (n = 3), of memory order four (ν=4).

The effective delays for the sub-optimum receiver were computed using equation (2.2.5). Thus, $d_{eff} = 3$ corresponds to δ=2 and D=3, and $d_{eff} = 5$ corresponds to δ=3 and D=4. Note that $d_{eff} = 5$ utilizes a detector delay of L-1, and a decoder delay of ν-1, and hence achieves nearly the best error rates for the given channel and code

constraint lengths. The delay d=5 for the optimum receiver also gives nearly the best error rates for the same reason since, for the specified constraint lengths, $\ell=6$ according to eqn. (2.3.2).

It can be seen from the performance curves of Fig. 3.2 that the optimum receiver allows approximately a 4-dB reduction in required energy-per-bit to noise ratio compared to the sub-optimum receiver at a bit-error rate of 10^{-3}. The improvement is actually larger at lower error rates as is apparent from the slopes of the curves.

Simulations were not performed for error rates below 10^{-4} as they require excessive time. The average computer time, based on compiler seconds, required for the delay=5 simulations was approximately 0.04 sec/source-bit for the suboptimum system, and 0.05 sec/source-bit for the optimum system.

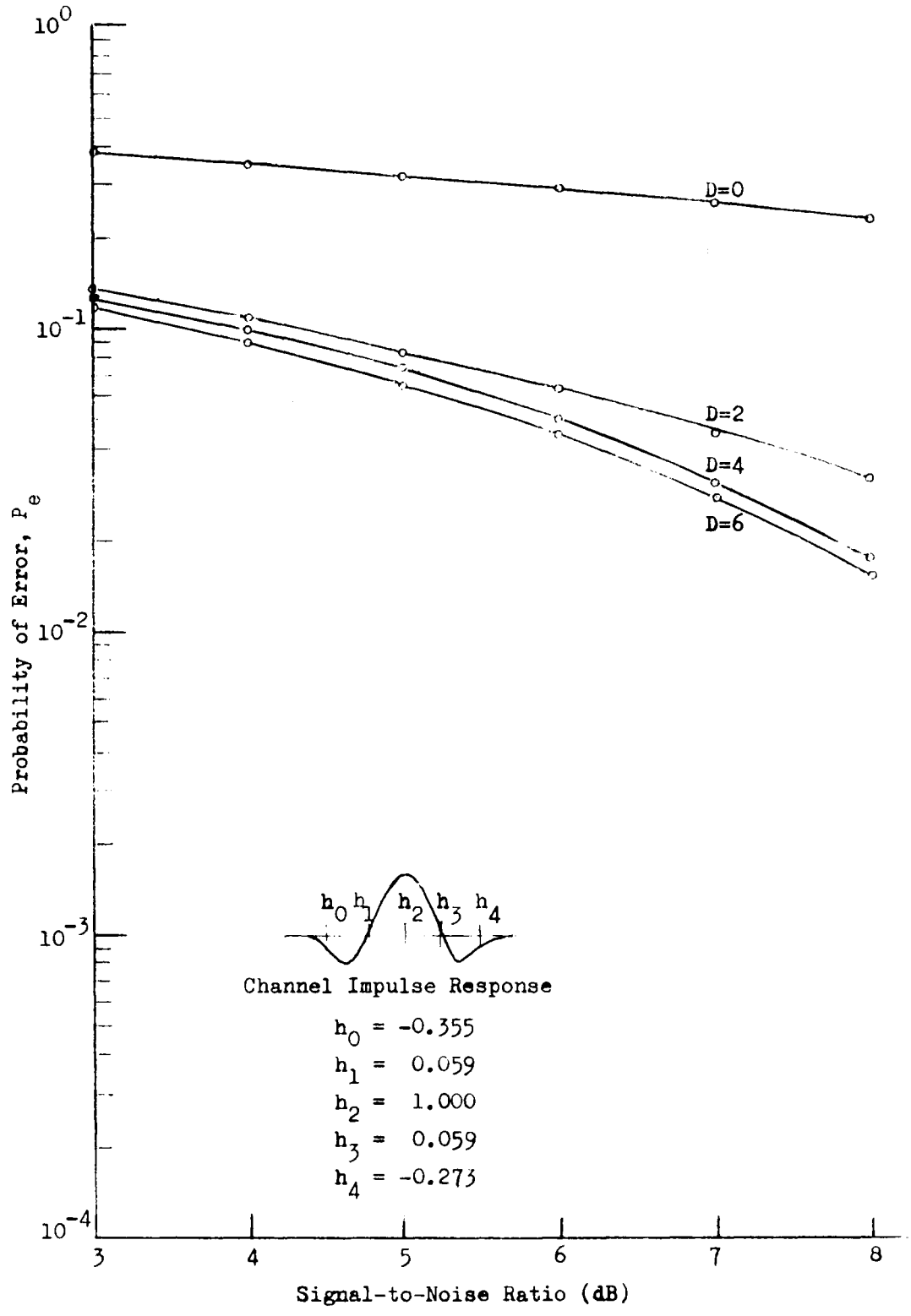


Fig. 3.1 Sequential Detector Performance

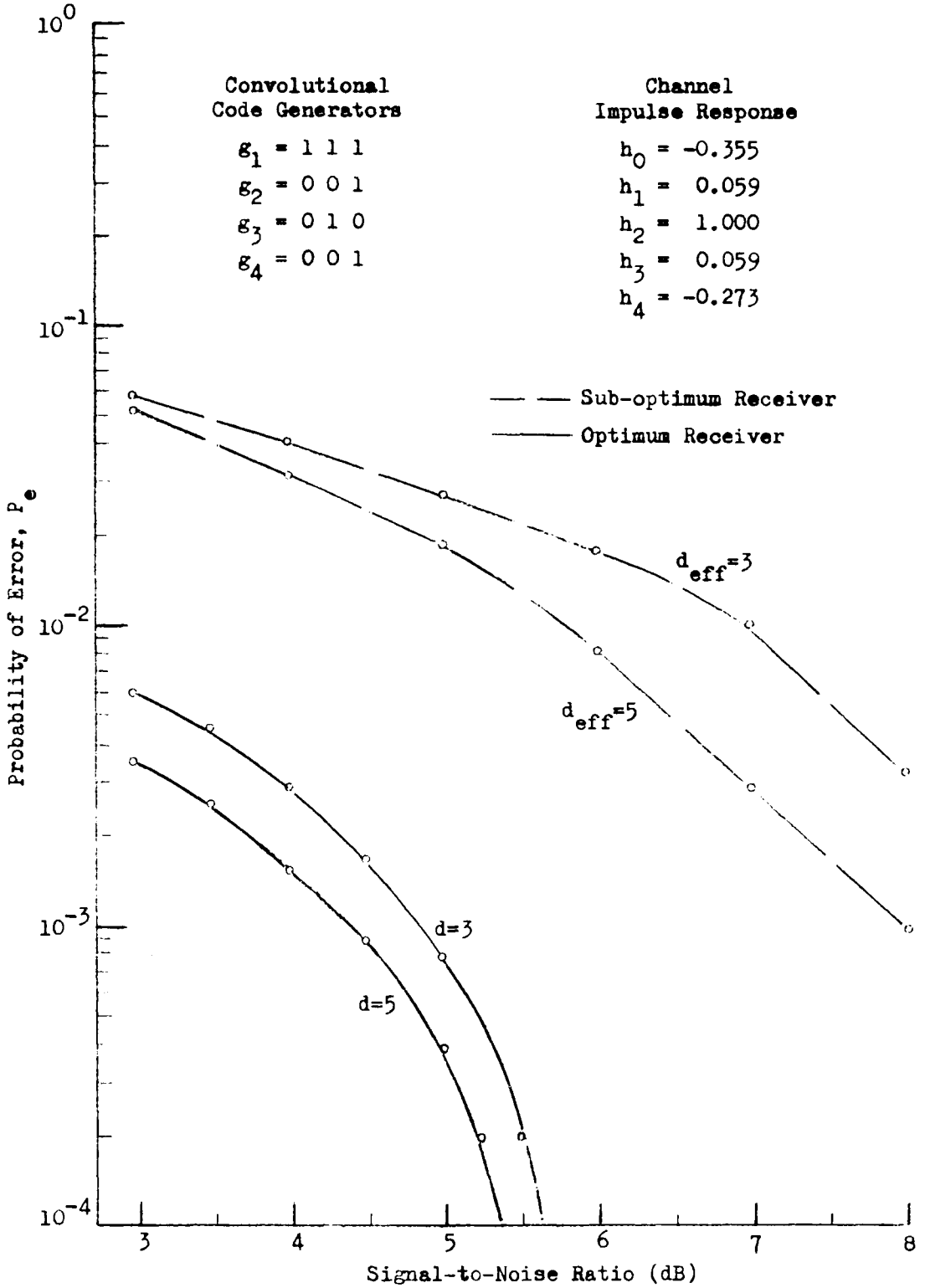


Fig. 3.2 Performance of Sub-optimum and Optimum Receivers

CHAPTER 4

SUMMARY AND CONCLUSIONS

In this work we have investigated an optimum receiver algorithm for digital communication systems with convolutional encoders and intersymbol interference channels. It was shown that this algorithm is a direct extension of the sequential detector [1] and the sequential decoder [4] algorithms.

Computer simulations were performed for the sequential detector, the separate detector-decoder (or sub-optimum receiver) system, and the joint detector-decoder (or optimum receiver) system. Flowcharts for these programs are included in this work. The simulations show that the optimum receiver allows approximately a 4-dB reduction in required energy-per-bit to noise ratio compared to the sub-optimum receiver at a bit-error rate of 10^{-3}, and that the improvement is even better at lower error rates.

The effective constraint length ℓ for the optimum receiver system was found to be $\nu + \left[\frac{L-1}{n}\right]$, which is much smaller than $(\nu+L-1)$ for $n>1$. The value $(\nu+L-1)$ has been assumed as the effective constraint length in an important paper by Omura [5].

A major advantage of the receivers studied in this work is that their structures are fixed regardless of message length, if L and ν are assumed to be unchanging. This is in contrast to the Viterbi decoder [6] whose structure grows linearly with the message length. A disadvantage of the sequential compound receivers, however, is that their computational complexity increases exponentially with the constraint length, whereas the complexity of the receiver in [6] increases only linearly.

Therefore more investigation of sequential compound receivers is needed before they become feasible for most practical applications. It would probably be interesting to study the distance properties of the optimum receiver algorithm using trellis diagrams associated with Viterbi decoding.

REFERENCES

1. Abend. K., and Fritchman, B.D., (1970), "Statistical Detection for Communication Channels with Intersymbol Interference", Proc. of the I.E.E.E., Vol. 58. No. 5, pp. 779-785.

2. Shannon, C.E., (1949), "Communications in the Presence of Noise", Proc. of the I.R.E., Vol. 37, pp. 10-21.

3. Forney, G.D., (1972), "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference", I.E.E.E. Trans. on Information Theory, Vol. IT-18, pp.363-378.

4. Mixsell, J.C., (1973), "Optimum Decoding of Convolutional Codes Using the Sequential Compound Decoder", Ph.D. Dissertation, Lehigh University.

5. Omura, J.K., (1971), "Optimal Receiver Design for Convolutional Codes and Channels with Memory via Control Theoretical Concepts", Information Sciences, Vol. 3, pp. 243-266.

6. Viterbi, A.J., (1967), "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", I.E.E.E. Trans. on Information Theory, Vol. IT-19, No. 2, pp. 260-269.

7. Gear, (1973), Introduction to Computer Science, Science Research Associates.

Appendix A

SEQUENTIAL DETECTOR SIMULATION

This chapter presents a flowchart of the computer program that was used to simulate the sequential detector algorithm, as described completely in [1]. The flowchart notation is generally adopted from [7], though it is in places modified to serve brevity. The computer system that was used to simulate this program, as well as the programs of the next two appendices, is the CDC-6400. The compiler language is FORTRAN IV.

Our interest here is primarily to correlate the symbols and labels used in the flowchart diagrams with those used for explaining the algorithm, and to briefly outline the program set-up.

We begin with an explanation of the labels in Fig. A.1, which defines the various quantities and parameters that are used throughout the program. D and L are the detector delay and the channel constraint length, respectively. (H(I), I=1,L) are the channel impulse response terms, except that the subscripts are shifted by one, i.e. $H(1)=h_0$, $H(2)=h_1$, $\cdots$, $H(L)=h_{L-1}$. SIGPOW is the signal power, calculated with the assumption that channel digits can attain 'levels' of ± 1 volts. SNRDB is the signal-to-noise ratio in dB, and AMEAN, STDEV, and VARNCE are the mean, standard deviation, and variance of the gaussian probability density function, respectively. The expression for variance (σ^2) is determined from the specified SNRDB and the signal power as follows:

$$\text{SNRDB} = 10 \log_{10} \text{SNR} = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{noise}}} = 10 \log_{10} \frac{\sum h^2}{\sigma^2}$$

$$\therefore \sigma^2 = (\sum h^2) \cdot 10^{-(\text{SNRDB}/10.0)}$$

Continuing with an explanation of the labels in Fig. A.1, ISTEP is the iteration step and ISTEPS is the total number of iterations desired. IGAUSS is the initializing integer for the

"GAUSS" built-in subroutine, which supplies a sequence of gaussian noise values according to the specified mean and standard deviation. Similarly, IRANDU is the initializing integer for the "RANDU" built-in subroutine, which generates uniformly distributed random numbers between 0 and 1.

Also in Fig. A.1, NA is an index for the channel symbols that need to be stored. NA is used for start-up purposes and has a maximum value of L for $D \leq L-1$, and a maximum value of D+1 for $D \geq L$. In Fig. A.2, several quantities associated with NA are defined, and the 2^{NA} values of RC, or R_i , associated with the sequences of channel symbols are computed. The sequences of channel symbols are formed in subroutine ARRAY.

In Fig. A.3, the random binary channel symbol, SYMBOL, on which iteration is to be performed is generated using "RANDU", and then transmitted through the channel to obtain X. GNOISE is a random gaussian noise component obtained from "GAUSS".

In Fig. A.4, the 2^{NA} joint probabilities associated with the received signal X are calculated. SYMPRB is the symbol probability, assumed equal to $\frac{1}{2}$, and PDFV represents the value of the gaussian probability density function associated with a discrete noise component Y, i.e.

$$\text{PDFV}\ (Y) = f(Y) = \frac{e^{-Y^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}$$

At the bottom of Fig. A.4, the joint probabilities are normalized by first finding the largest NEWPRB, say MAXPRB, and then dividing each NEWPRB by MAXPRB.

Figs. A.5 & A.6 present the start-up procedure and the general procedure, respectively, for computing the 'old probabilities'--i.e. the third factor of eqn. (2.1.6)--to be used in the next iteration. OLDPRB(1) is initially set equal to 1.0. Fig. A.6 also contains a NA-stage 'shift-register' for the channel symbols.

Fig. A.7 is the decision segment of the program. Most of the labels introduced here are self-explanatory. DECSYM stands for decision symbol, ERRCNT for error count, and ERRPRB for error probability.

Finally, Fig. A.8 presents a method of forming a two-dimensional array consisting of 2^M possible M-length sequences of channel symbols, where M is at least equal to L. (A value of $M > L$ may be used in forming the sequences to allow for programming flexibility.) The form of the binary array is as follows:

Order of Channel Symbols								Row
C_M						C_2	C_1	↓
-1	·	·	·	·	·	-1	-1	1
-1	·	·	·	·	·	-1	+1	2
-1	·	·	·	·	·	+1	-1	3
			·					·
			·					·
			·					·
+1	·	·	·	·	·	+1	+1	2^M

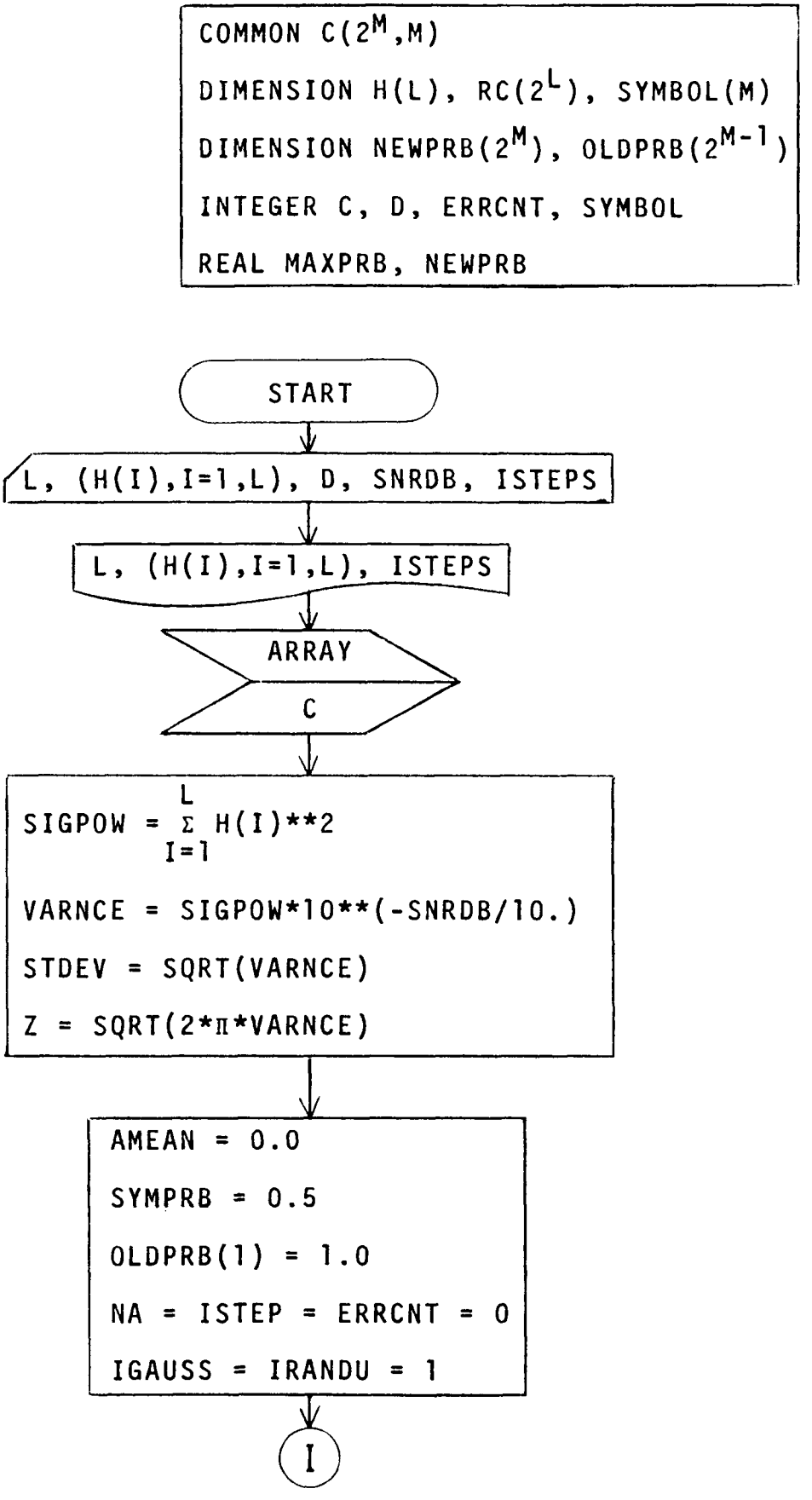


Figure A.1 Definitions And Initial Conditions

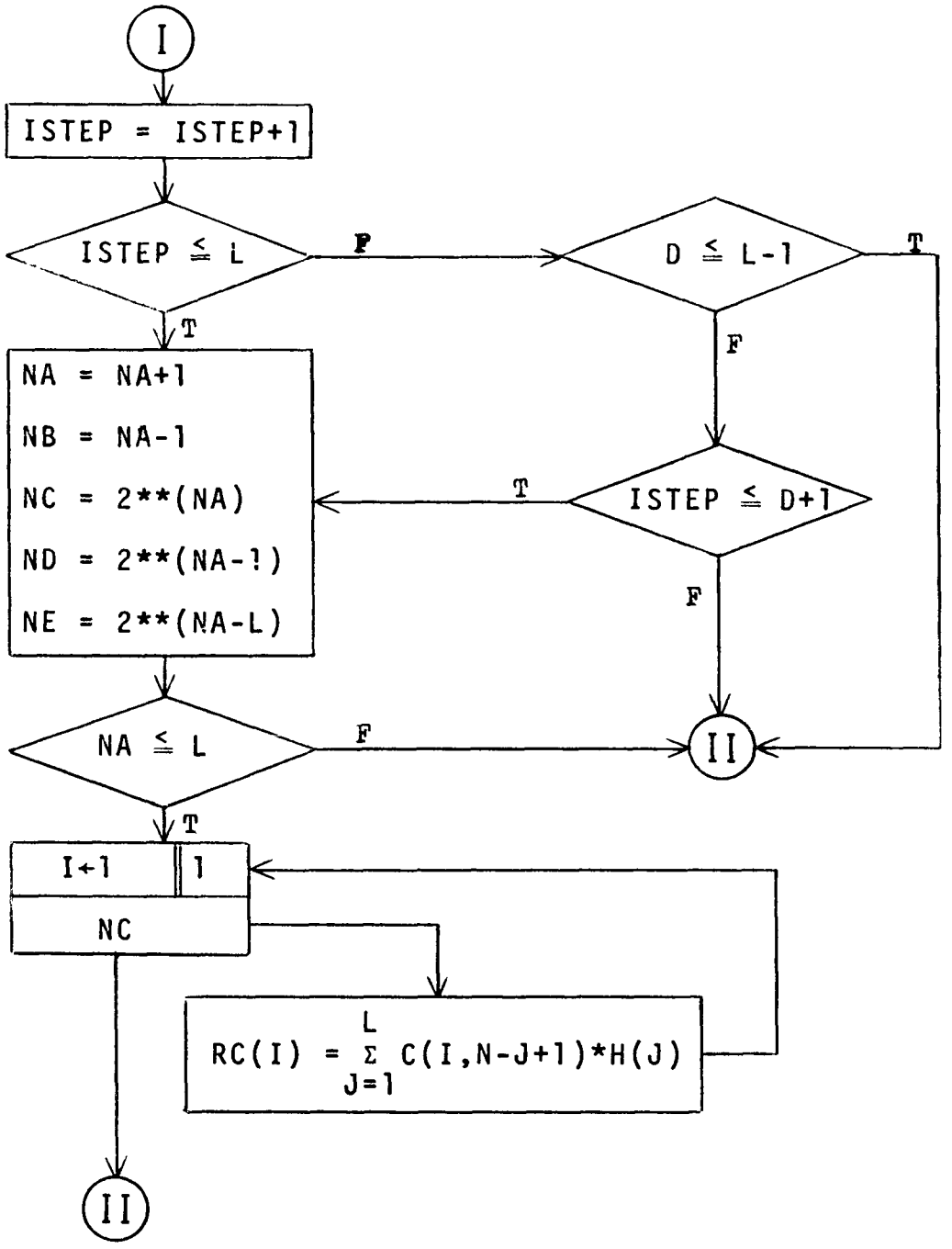


Figure A.2 Start-up Procedure And Calculation of R_i

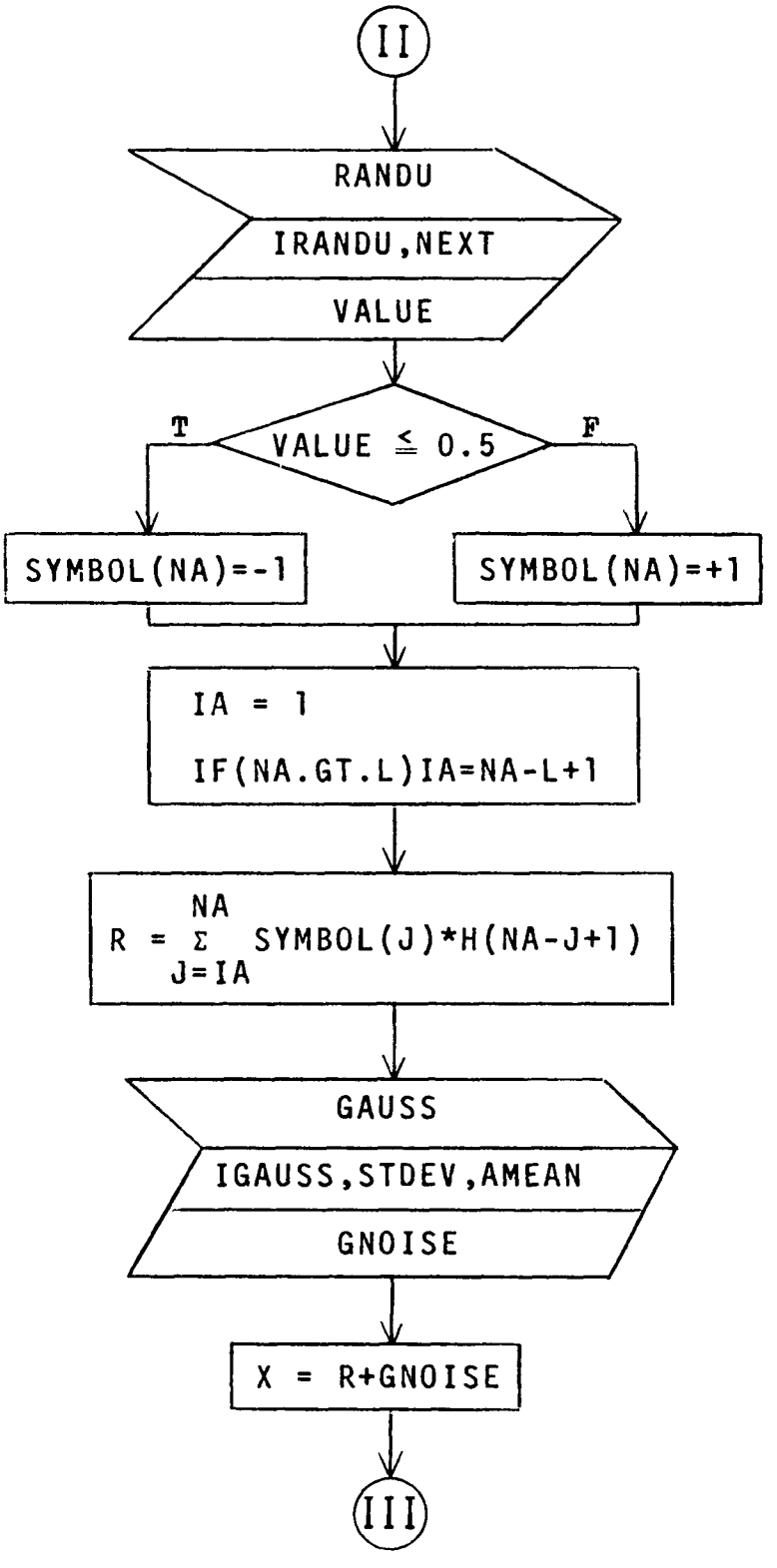


Figure A.3 Calculation of X for Generated Symbol

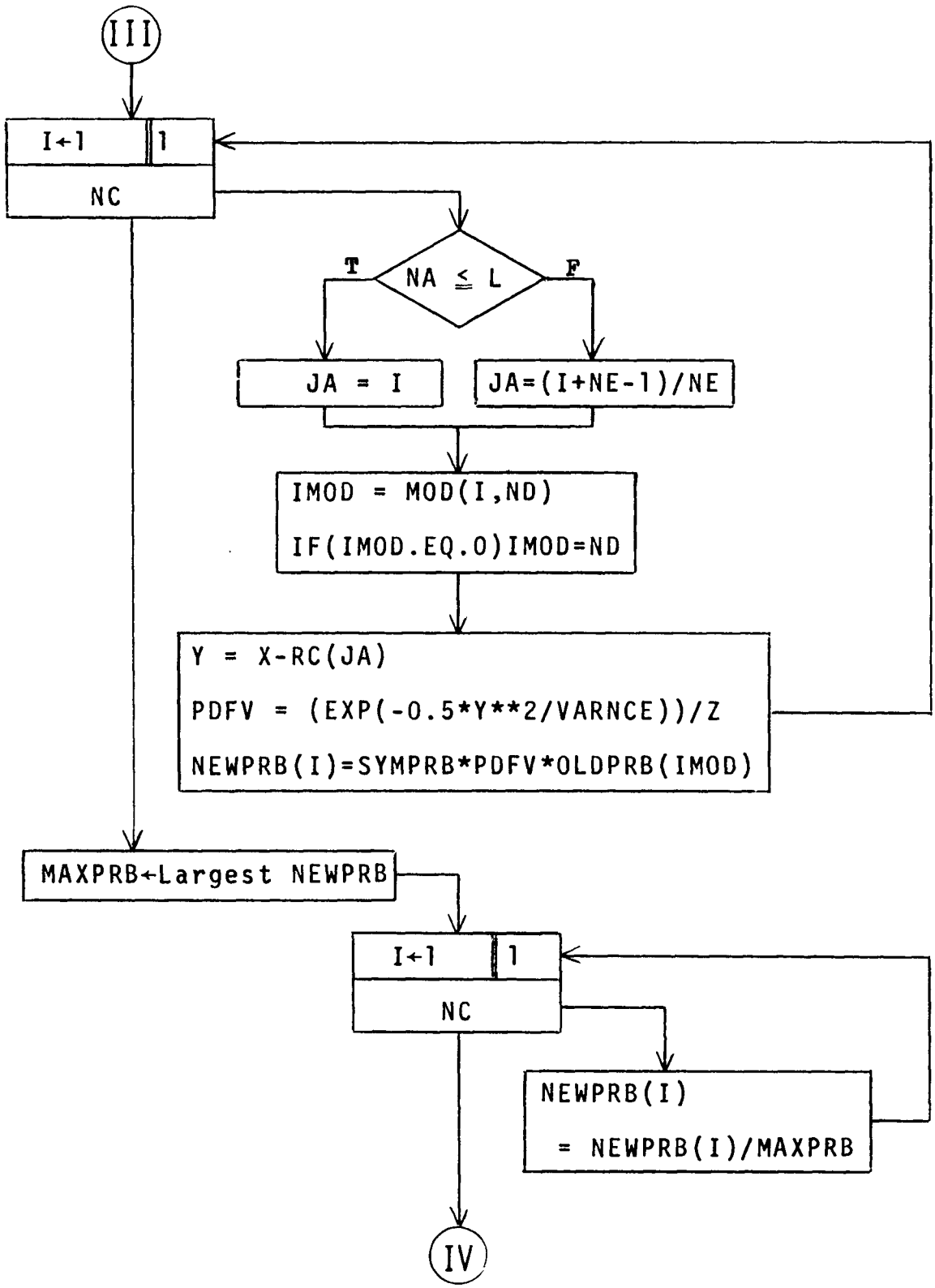


Figure A.4 Computation And Normalization of Joint Probabilities

Figure A.5 → Start-up For Recursive Rule

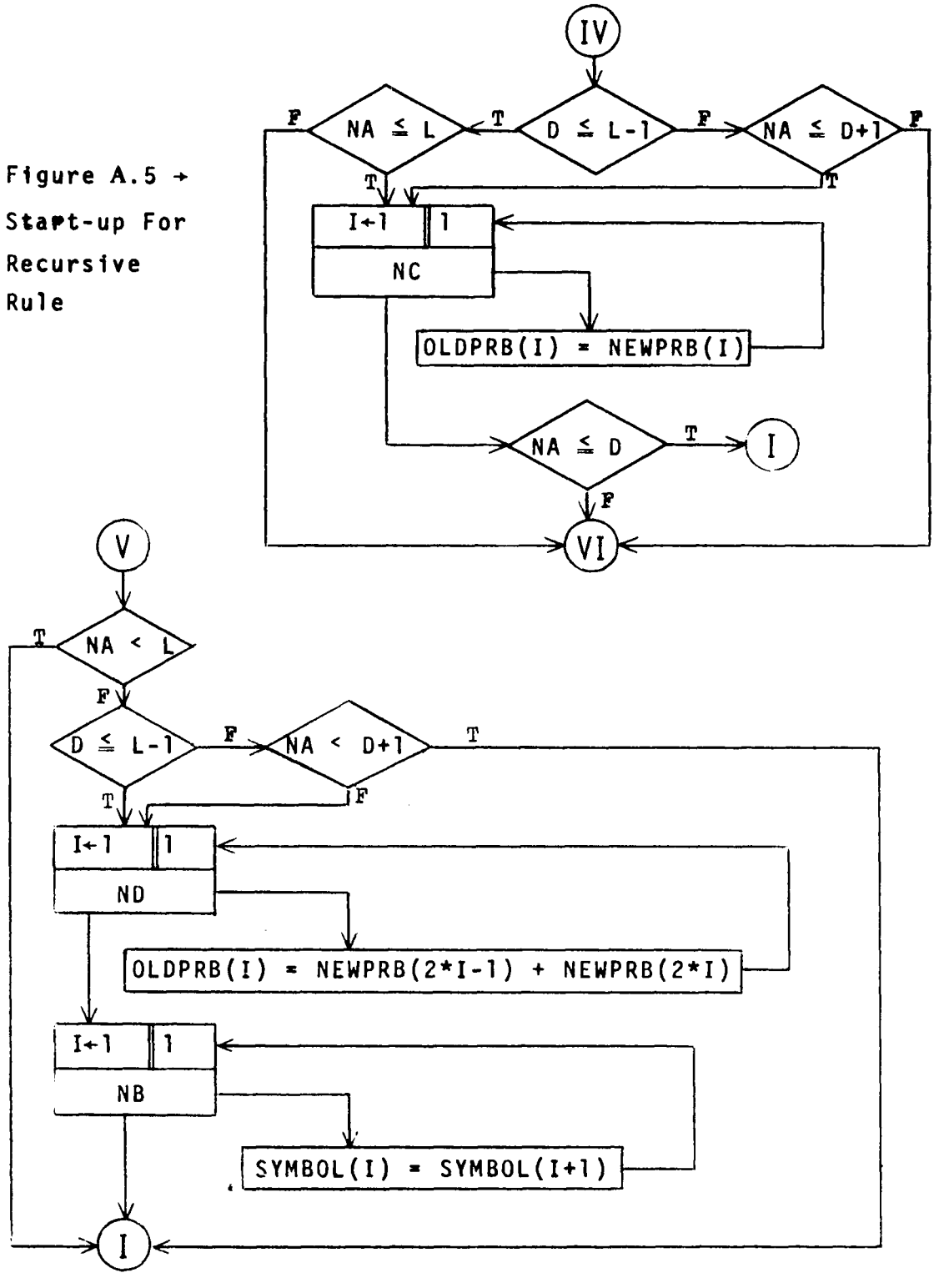


Figure A.6 Calculation of 'Old Probabilities' And Shifting of Symbols

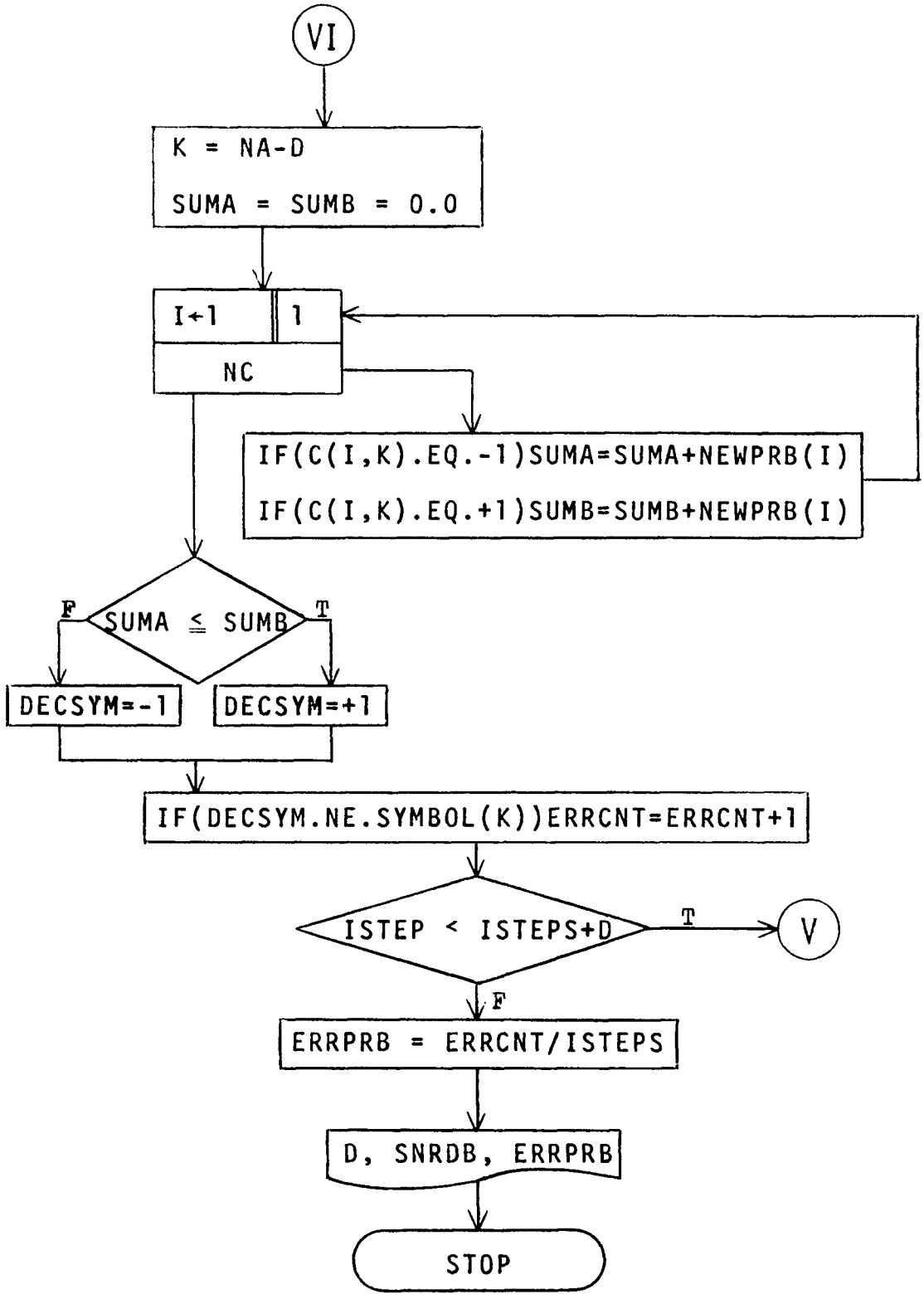


Figure A.7 Decision Segment

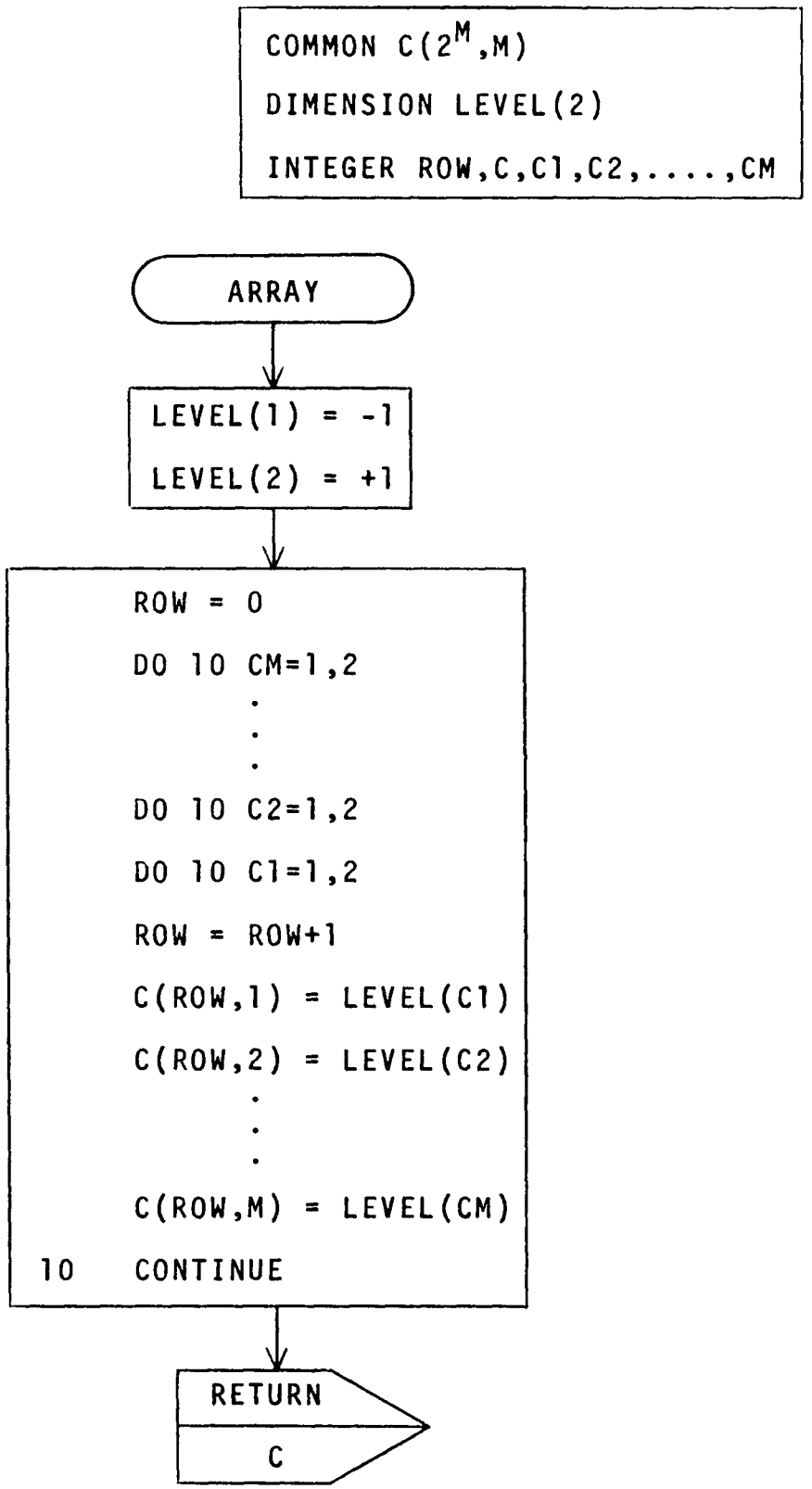


Figure A.8 Formation of Two-Dimensional Binary Array

Appendix B

SUB-OPTIMUM SYSTEM SIMULATION

In this chapter, we present a flowchart of the program that was used to simulate the sub-optimum system. The comments regarding flowcharting and the computer system made in Appendix A are also relevant here. The s-o-s program is essentially composed of four subroutines--i.e. ARRAYS, ENCODR, DETCTR, and DECODR--whose names quite appropriately describe the functions they perform. The DETCTR subroutine is basically the same as the sequential detector program of Appendix A. Therefore, to avoid repetition of explanation, and also to maintain a semblance of continuity, many of the labels used in the last appendix have been retained here.

We start a figure-by-figure analysis of the flowchart. In Fig. B.1, V is the encoder constraint length, ν, and N has the smae meaning as the integer n, i.e. 1/N is the rate of the convolutional code. DELTA and DEFF represent δ and d_{eff}, respectively. ((G(I,J), J=1,N), I=1,V) are the elements of the convolutional code generator matrix, G. CX and CY are the channel symbols at the transmitter and receiver ends, respectively. TX and TY are the encoded symbols corresponding to CX and CY.

Many of the other quantities in Fig. B.1 are similar to those used in Appendix A, and are easily understood from their context. For instance, MA and NA are indexes for the decoder and detector symbols, respectively, and are useful for start-up purposes. MA attains a maximum value of V, and NA a maximum value of L.

Fig. B.2, which contains the primary loop of the program, introduces no new labels of immediate interest. In Fig. B.3, TB, or $\underline{T}_i$, are binary n-tuples associated with all possible ν-length sequences of binary source symbols, $\{B\}$. Similarly, in Fig. B.4, RC, or R_i, are real channel outputs associated with all possible L-length sequences of channel symbols.

ISYMBL in Fig. B.5--which represents the encoder part of the program--is a source symbol, generated using "RANDU"; associated with ISYMBL is the index IV, which attains a maximum value of V. JSYMBL in Fig. B.6 is also a source symbol, but it is kept in a longer 'shift-register' for decision purposes; JK, the index associated with JSYMBL, attains a maximum value of DEFF+1. Fig. B.6 describes shifting operations for both ISYMBL's and JSYMBL's.

After ISYMBL is encoded into a binary n-tuple in Fig. B.5, the encoded outputs are converted to channel digits in Fig. B.6, which are then transmitted serially through the channel one-by-one. Most of the labels used in Figs.B.6 through B.9--which comprise the DETCTR subroutine--are the same as those used in Appendix A. In Fig. B.8, however, P stands for the detector bit-error probability, and IC is an index for the n digits of each symbol. As shown in Fig. B.9, the entire n-tuple must be detected, i.e. IC=N, before a decision can be made at the decoder.

Finally, Figs. B.10 and B.11 make up the DECODR subroutine. In Fig. B.10, the 'new' and 'old' probabilities are calculated. IDIST, in Fig. B.10, is the Hamming distance between the detected TY n-tuple and one of the discrete TB (or $\underline{T}_i$) n-tuples. JDIST is equal to n minus IDIST.

In Fig. B.11, ISTEP is the iteration step, and ISTEPS is the total number of desired iterations. ISKIP is the number of iterations on which decision is skipped at the beginning so as to stabilize the value of the detector error probability P, which is utilized for computing 'incremental' probabilities in Fig. B.10. It is suggested that ISKIP be set approximately equal to $[1000/n]$, as 1000 iterations at the detector yield a fairly stable P for low values of SNRDB.

In conclusion, we point out some of the limitations of the program as described here. They are: $D \leq L-1$, $\delta \leq \nu-1$, and $n \leq \nu$.

```
COMMON B(2^V,V),TB(2^V,N),C(2^L,L),RC(2^L),CX(L)
COMMON G(V,N),H(L),ISYMBL(V),JSYMBL(DEFF+1)
COMMON TX(V,N),TY(V,N),OLDP(2^V-1),OLDPRB(2^L-1)
COMMON D,DELTA,DEFF,L,N,V,IV,JK,MA,NA
COMMON P,SYMPRB,ERRCNT,ERRORS,IGAUSS,IRANDU
COMMON AMEAN,SNRDB,STDEV,VARNCE,Z
COMMON ISTEP,JSTEP,KSTEP,ISKIP,ISTEPS
INTEGER B,C,CX,CY,TB,TX,TY,G,V
INTEGER D,DELTA,DEFF,ERRCNT,ERRORS
```

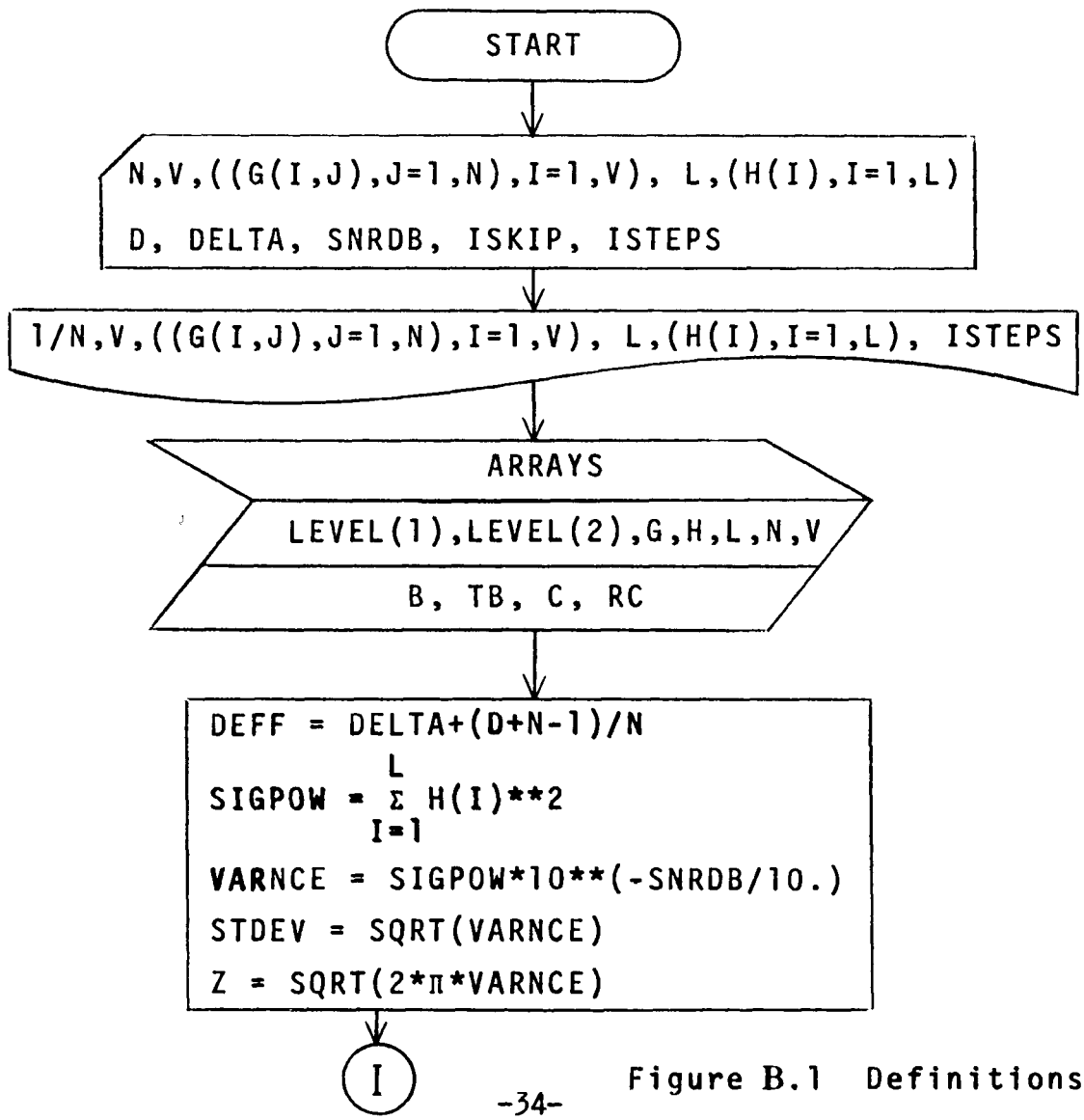


Figure B.1 Definitions

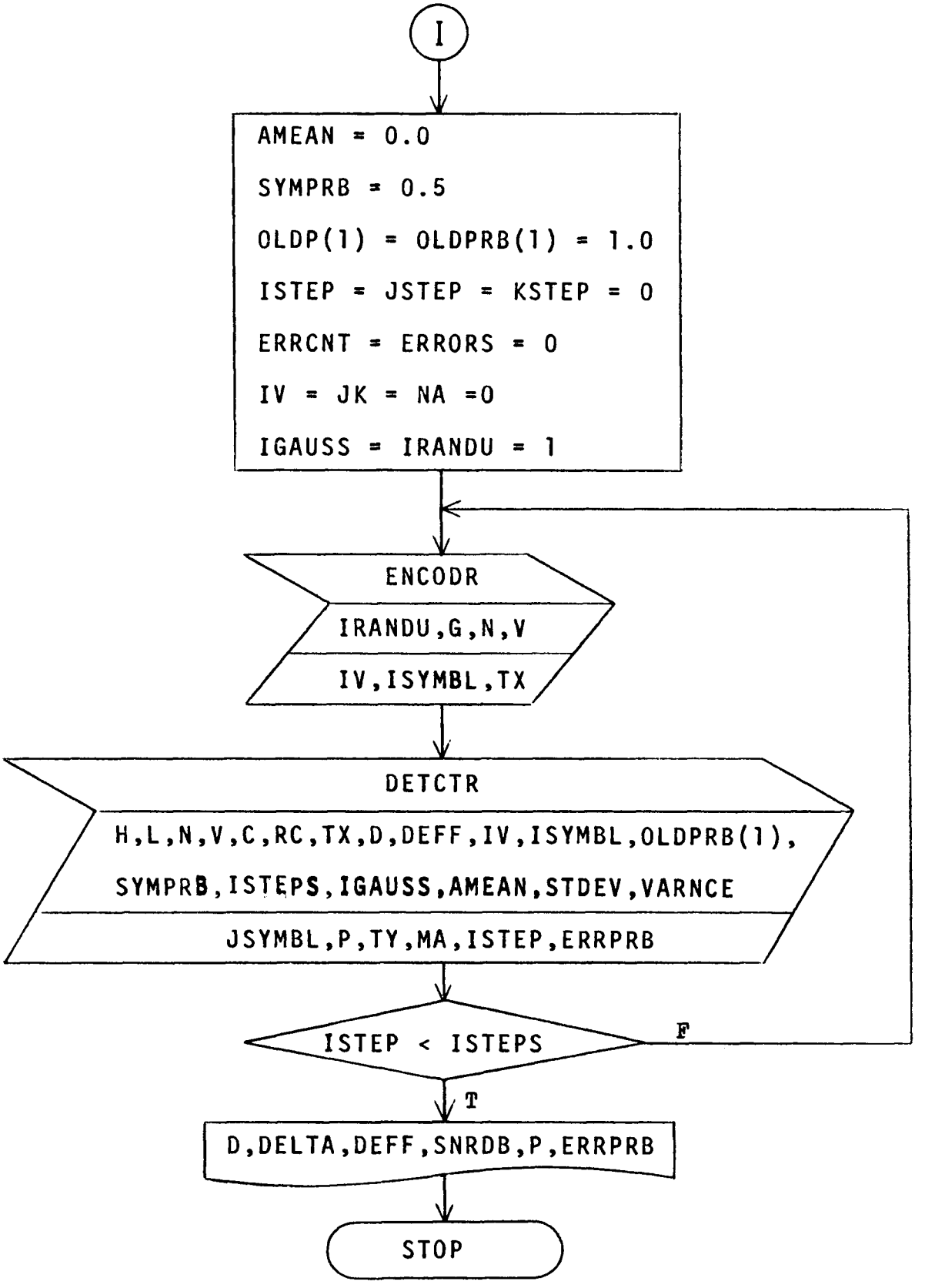
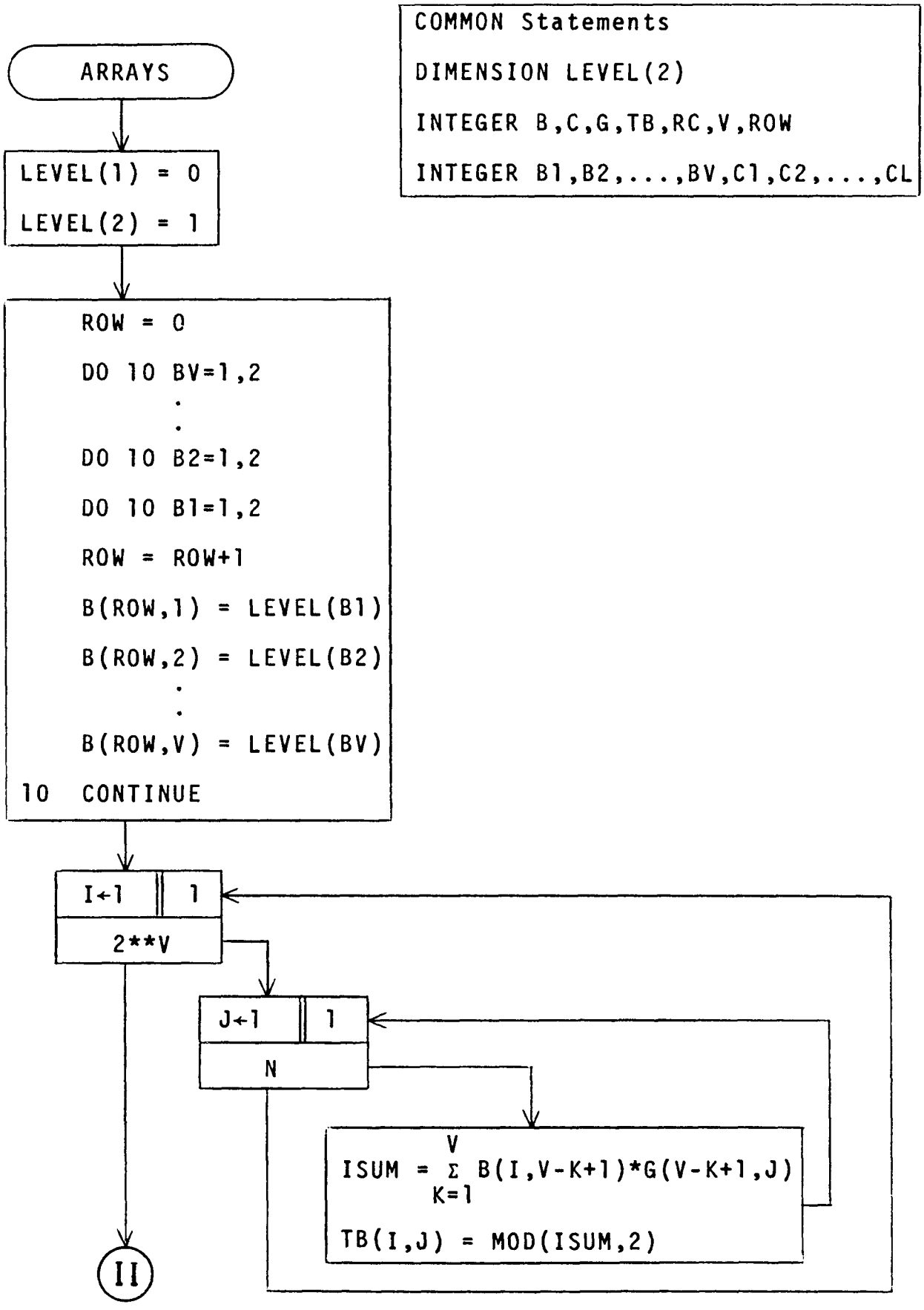


Figure B.2 Main Program



-36- Figure B.3 Calculation of $\underline{T}_i$

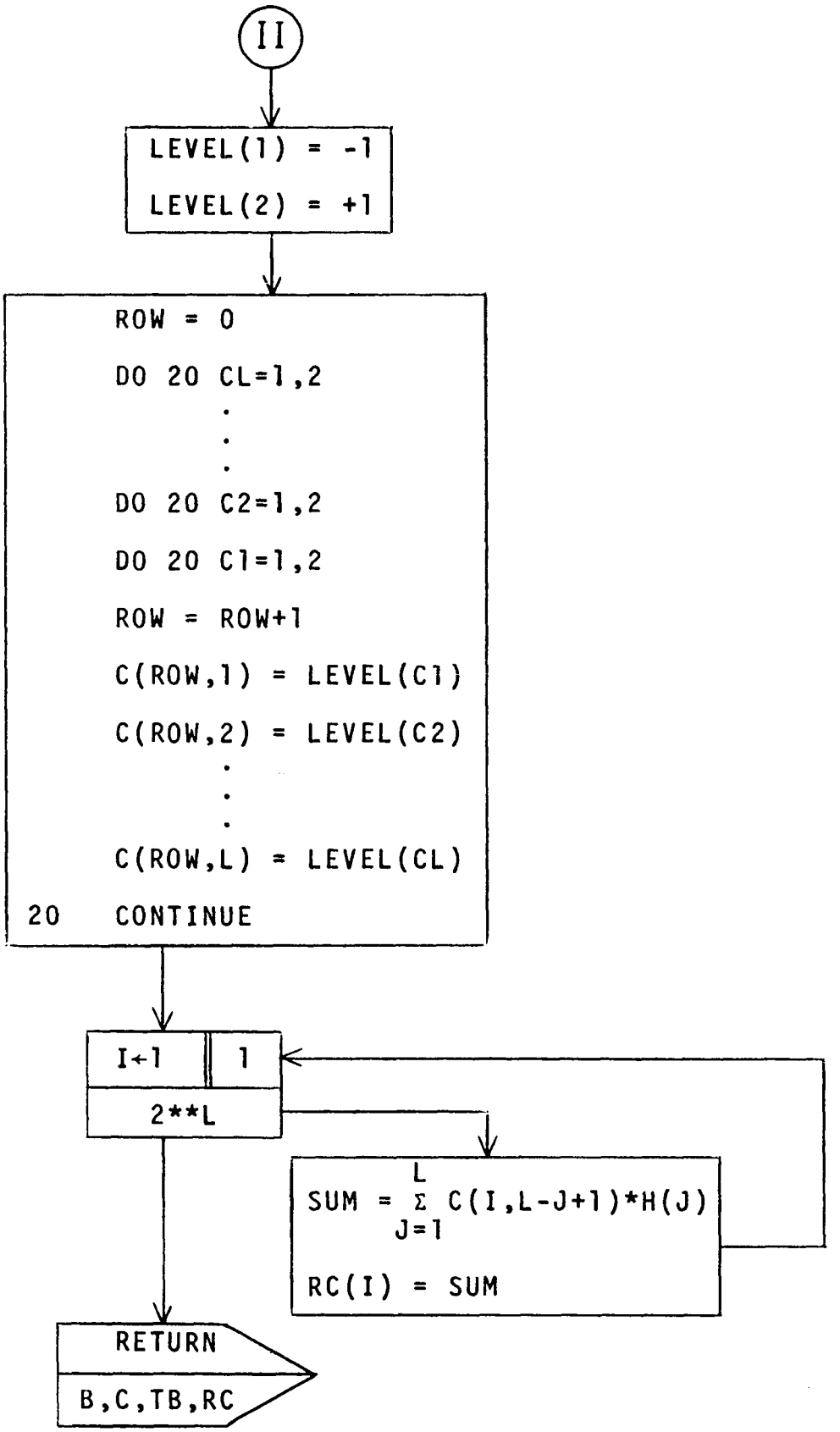


Figure B.4 Calculation of R_i

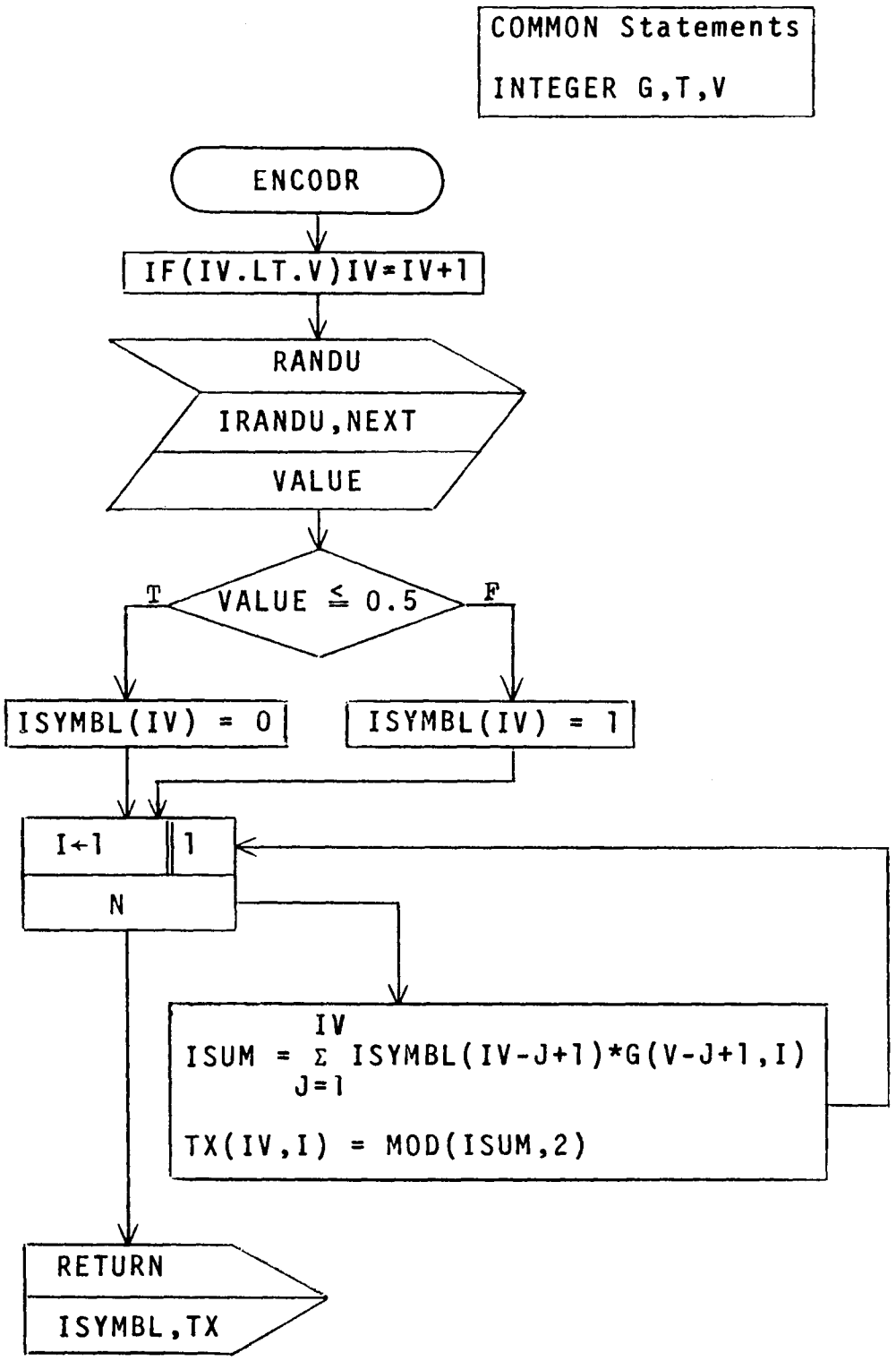


Figure B.5 Generation And Encoding of Source Symbols

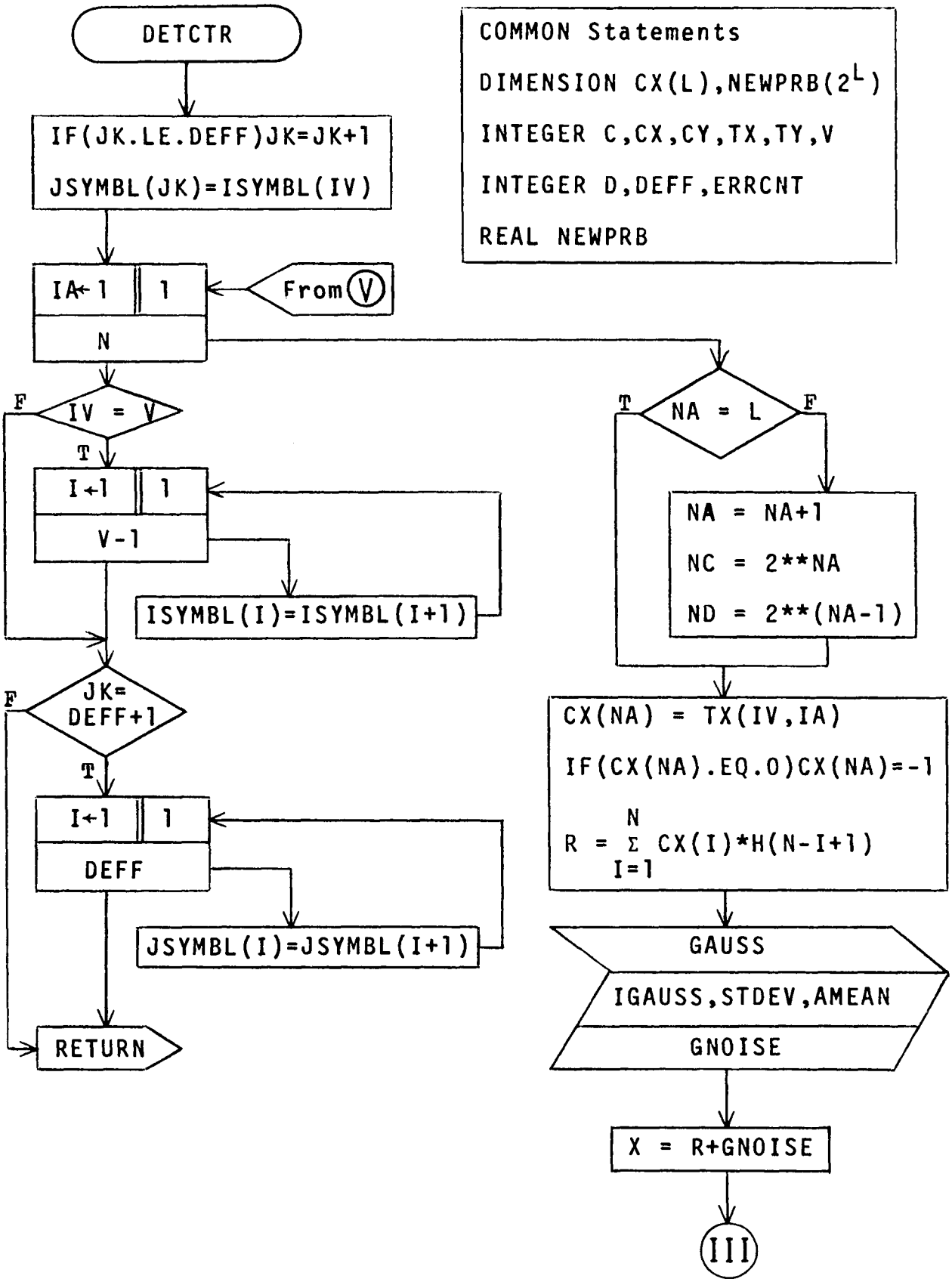


Figure B.6 Shift-Registers & Transmitter

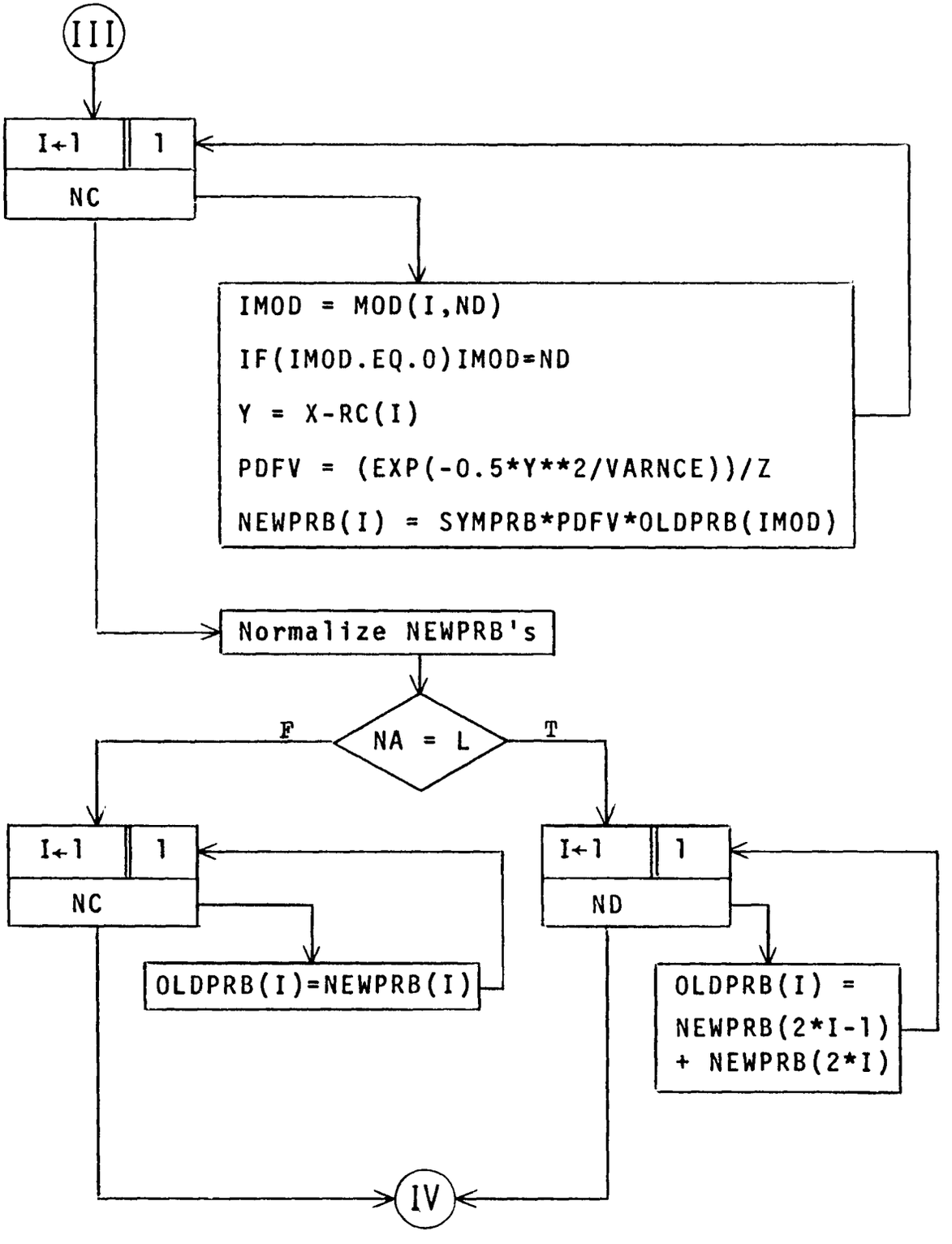


Figure B.7 Calculation of 'New' And 'Old' Probabilities For Channel Symbols

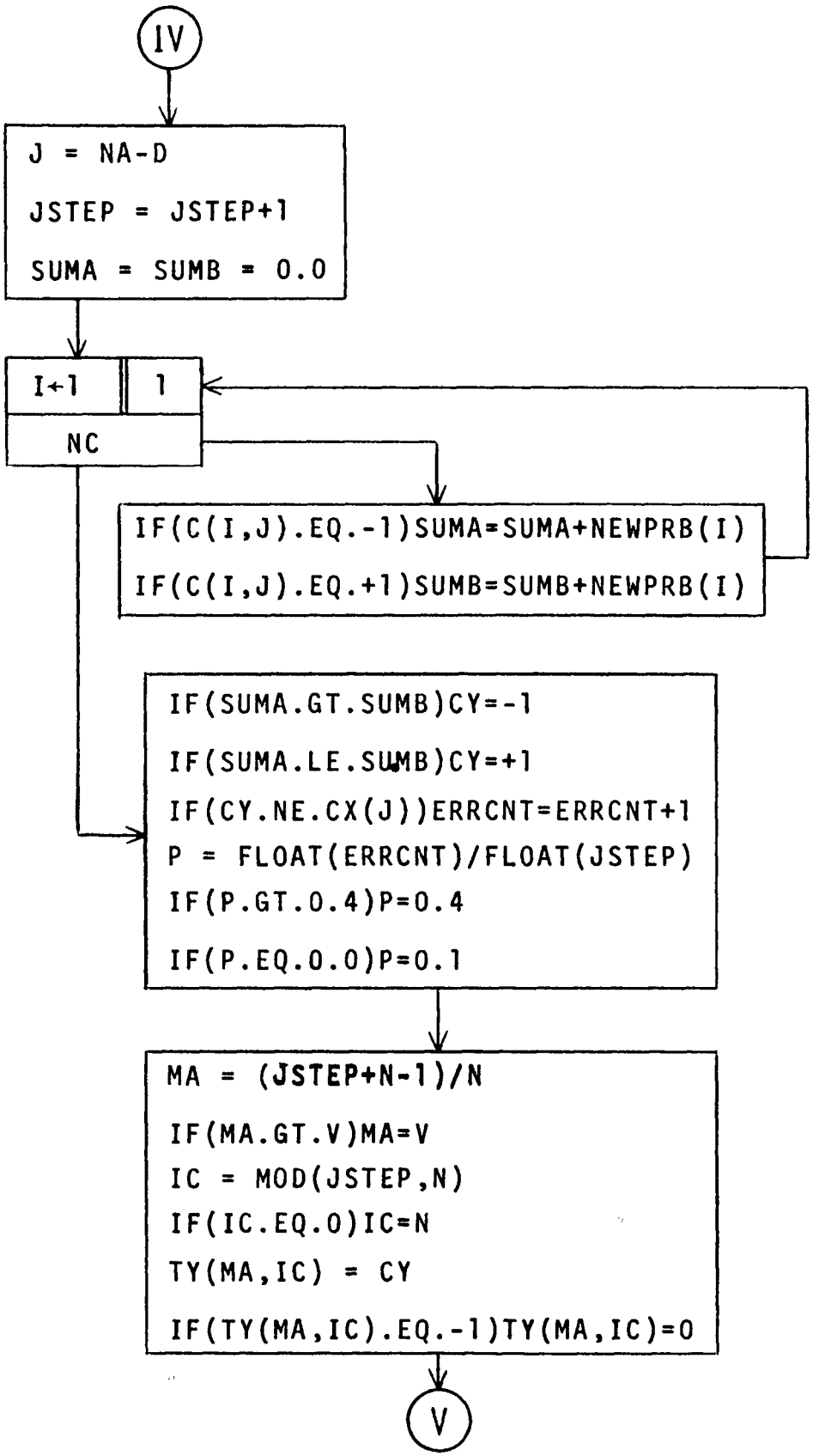


Figure B.8 Decision Segment For Detector

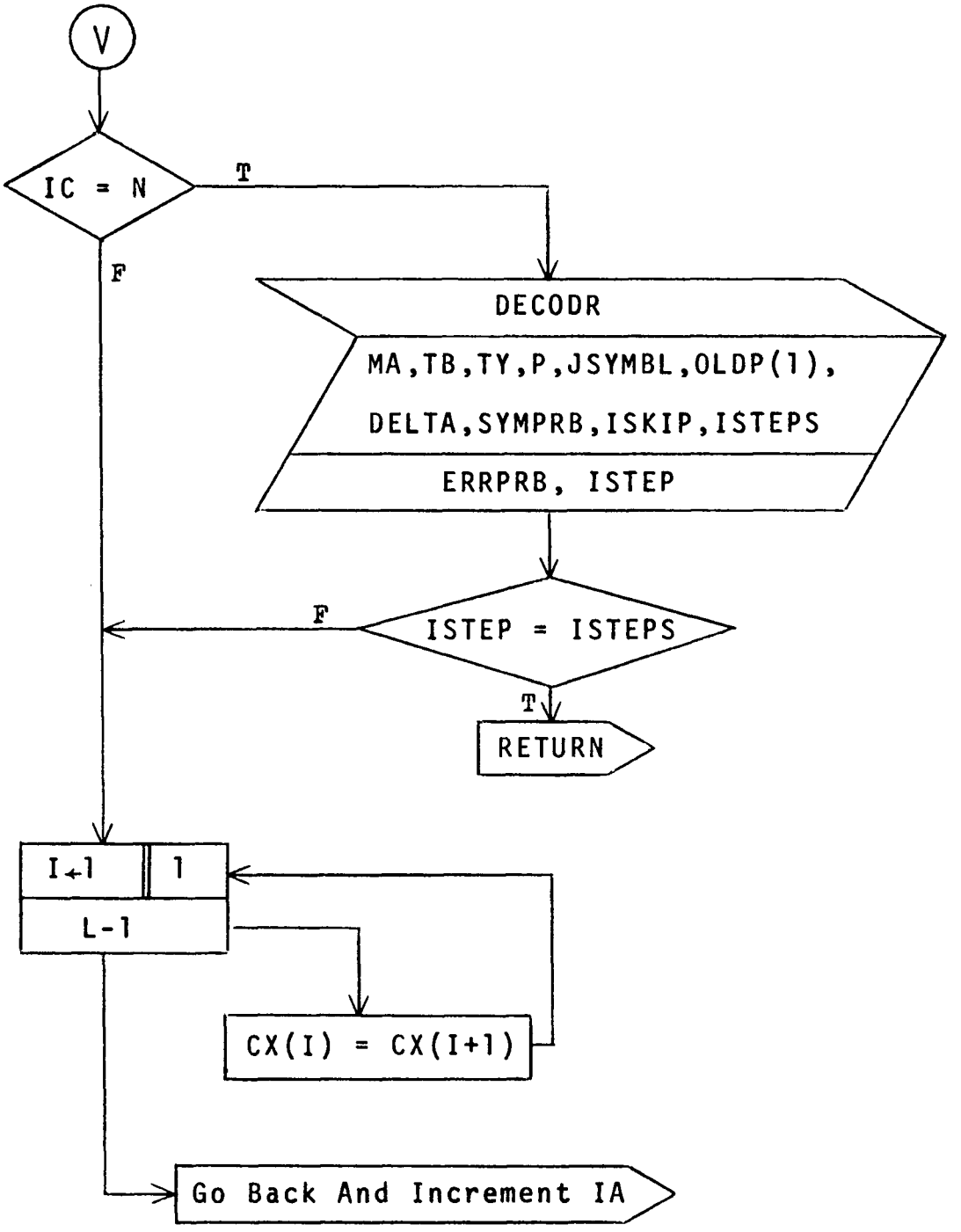


Figure B.9 Completion of Detector Subroutine

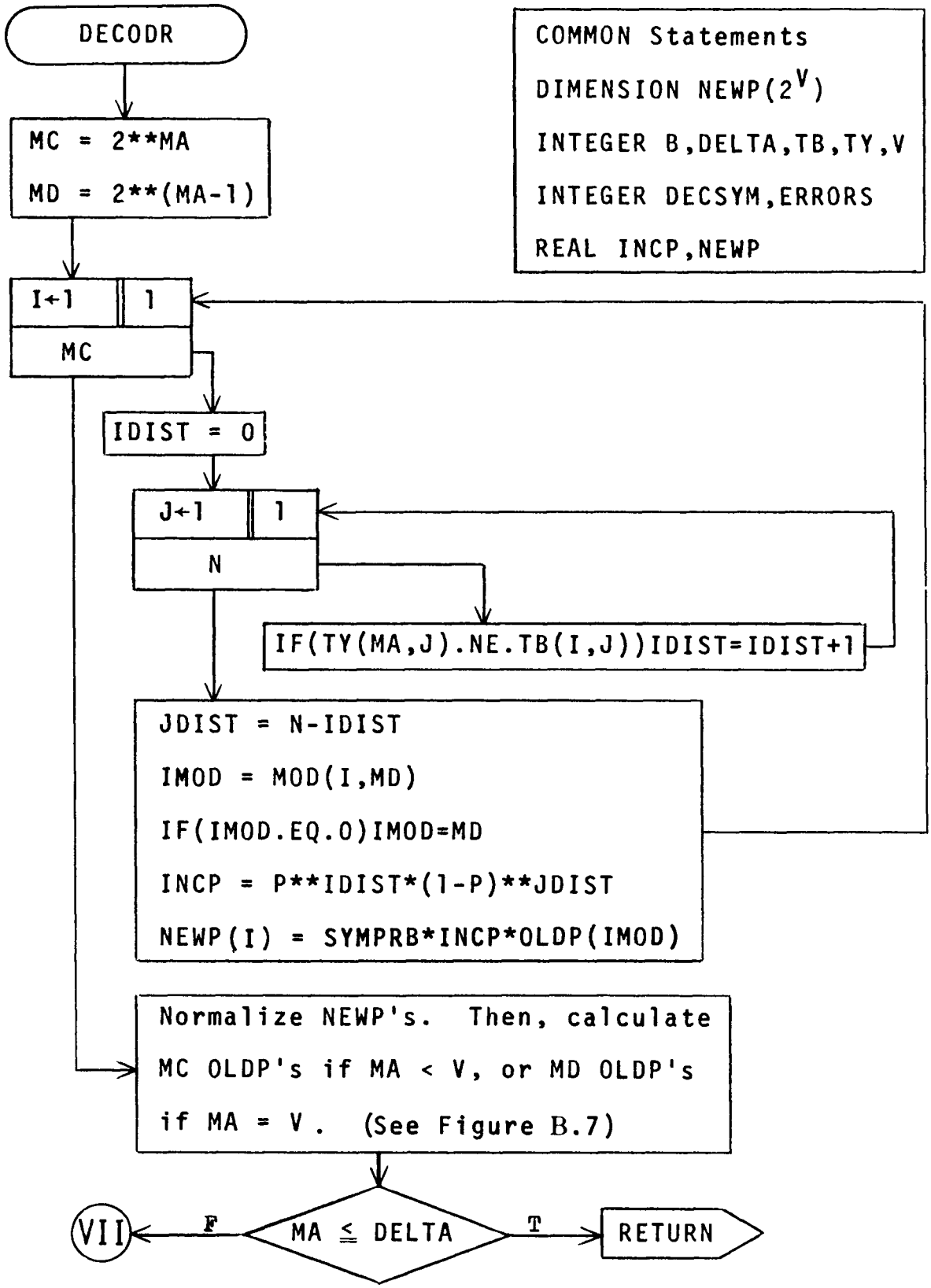


Figure B.10 Calculation of 'New' And 'Old' Probabilities

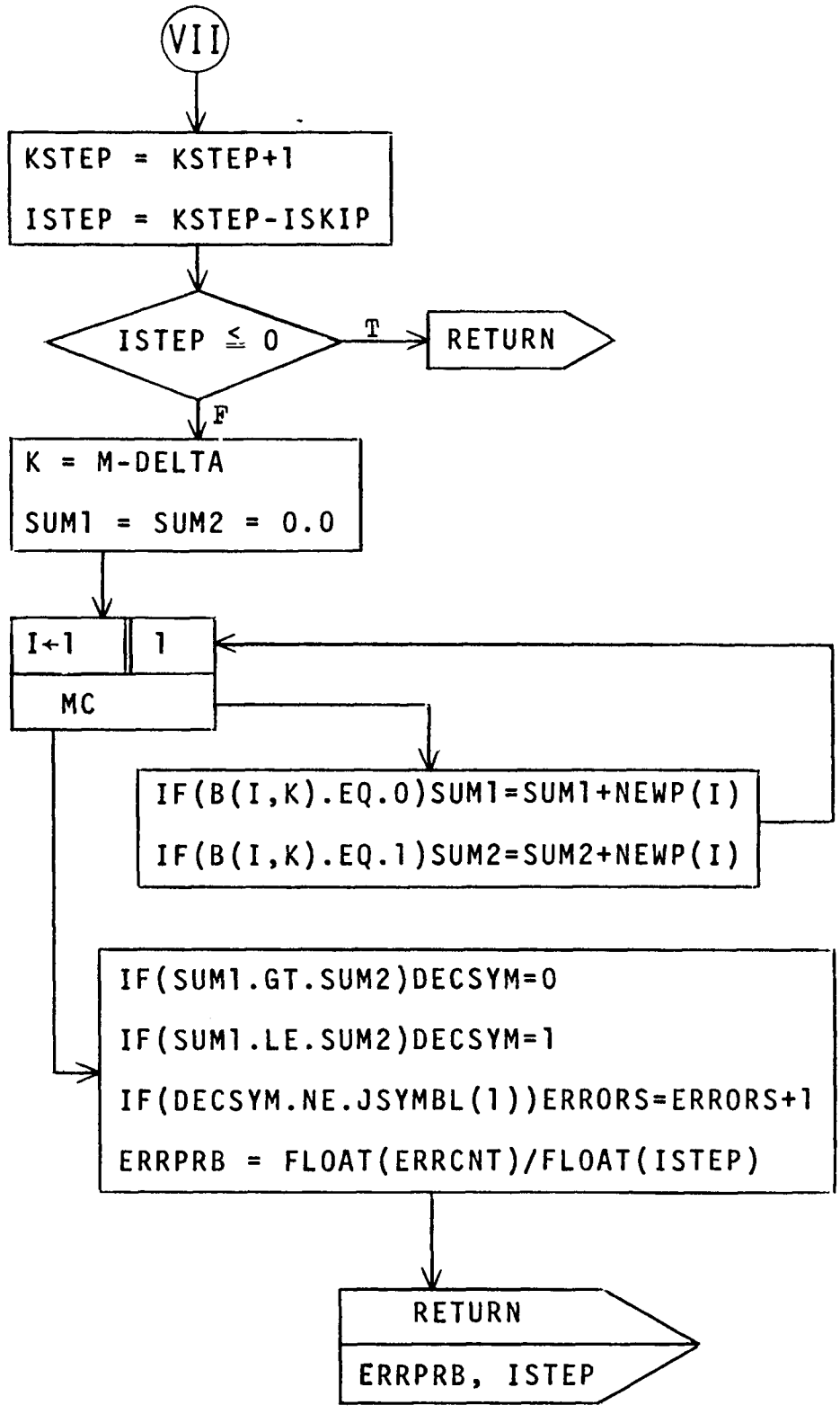


Figure B.11 Decision Segment For Decoder

Appendix C

OPTIMUM SYSTEM SIMULATION

The flowchart presented here should be fairly easy to understand after a study of the last two appendices. Because most of the labels used in this appendix are identical to those used in the previous two appendices, their explanation is not repeated, and only those labels are defined which have not been dealt with previously.

The only unfamiliar labels that appear in Fig. C.1 are LOCH and LEFF. LOCH represents the memory contribution of the channel, i.e. $[(L-1)/n]$ in eqn. (2.3.2), to the effective constraint length, LEFF (or ℓ), for the optimum system. There are practically no other new labels in Figs. C.1 through C.5. The operations described in these figures are also not new.

Perhaps the most distinctive part of the program is the subroutine ARRAYS of Fig. C.6. In this subroutine, initially the 2^{ℓ} possible ℓ-length sequences of binary source symbols are formed. Next the discrete values of channel outputs, RB (or $\underline{R}_i$), corresponding to these sequences are computed. This is accomplished by first generating LOCH*N+N ($\geq$ L-1+n) encoder outputs corresponding to each ℓ-length sequence $\{B\}$ and converting these encoder outputs to channel digits, CB. From these channel digits, the n elements of each RB n-tuple can then be evaluated quite easily using the method of eqn. (2.3.3).

Since the values of RB are calculated for ℓ-length sequences in advance, there will inherently be some errors incorporated in the start-up. Therefore, a provision is made in Fig. C.5 for skipping decisions on the first few iterations through the use of ISKIP, which need be no longer than a few constraint lengths.

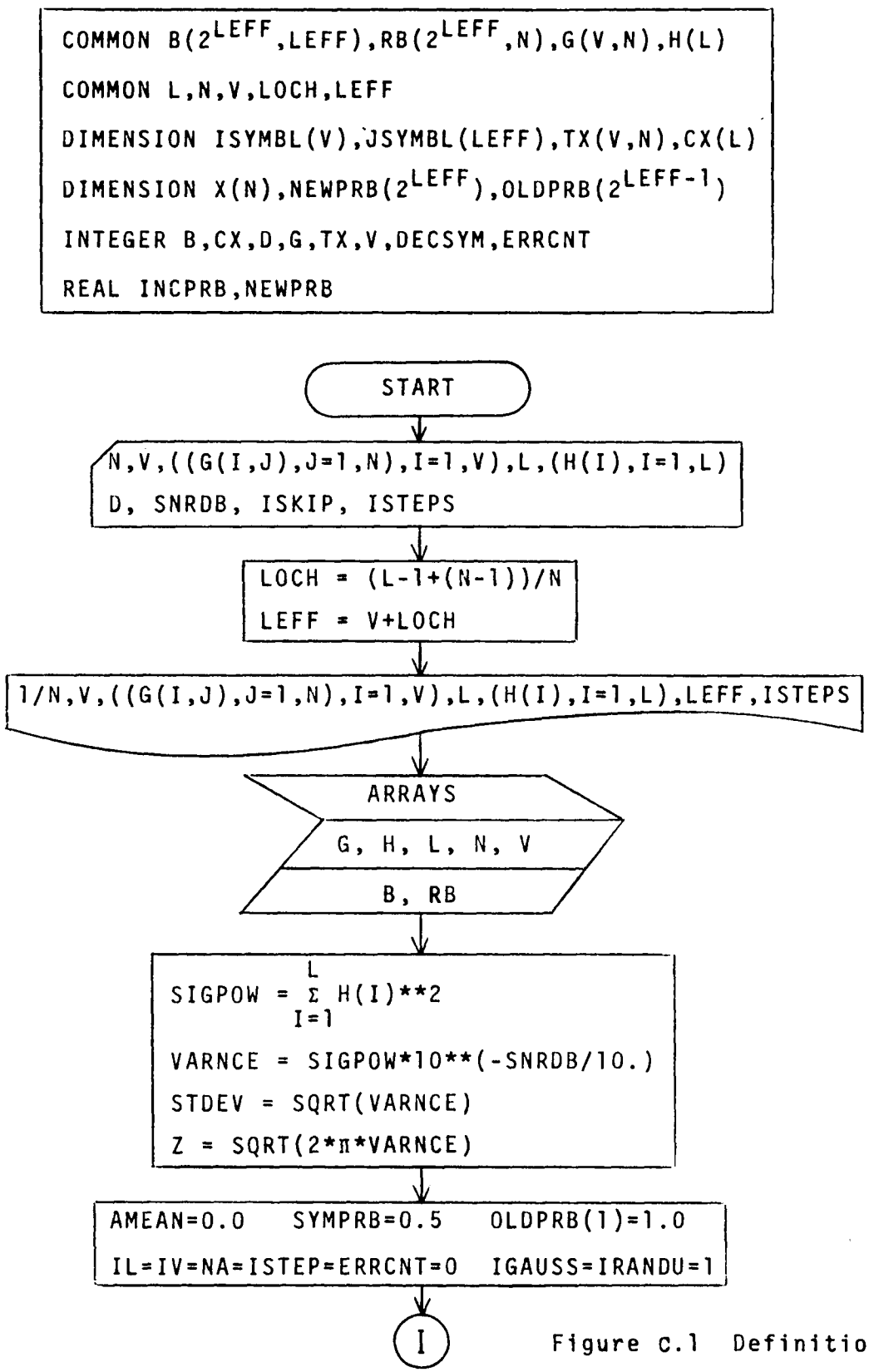


Figure C.1 Definitions

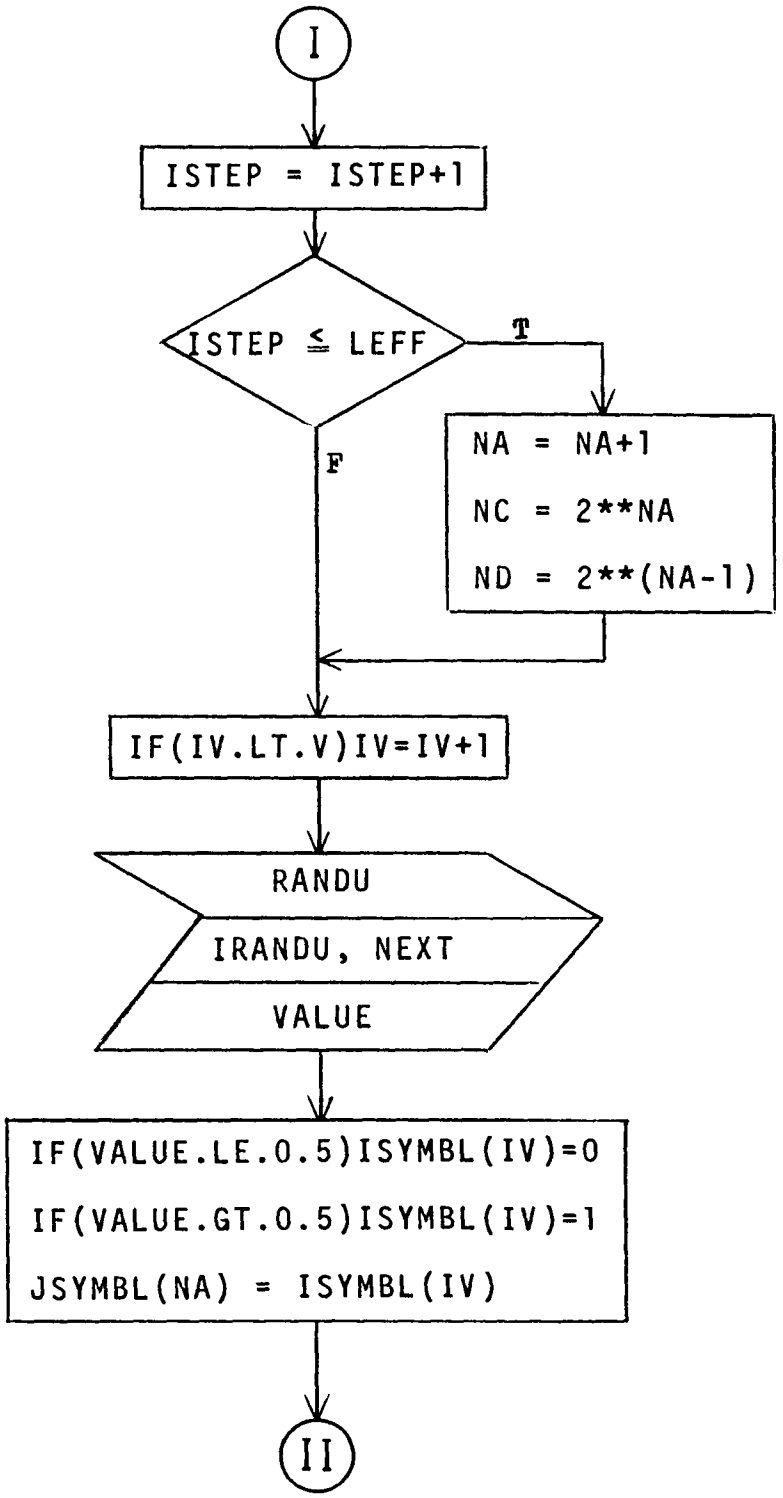


Figure C.2 Start-up Procedure And Symbol Generation

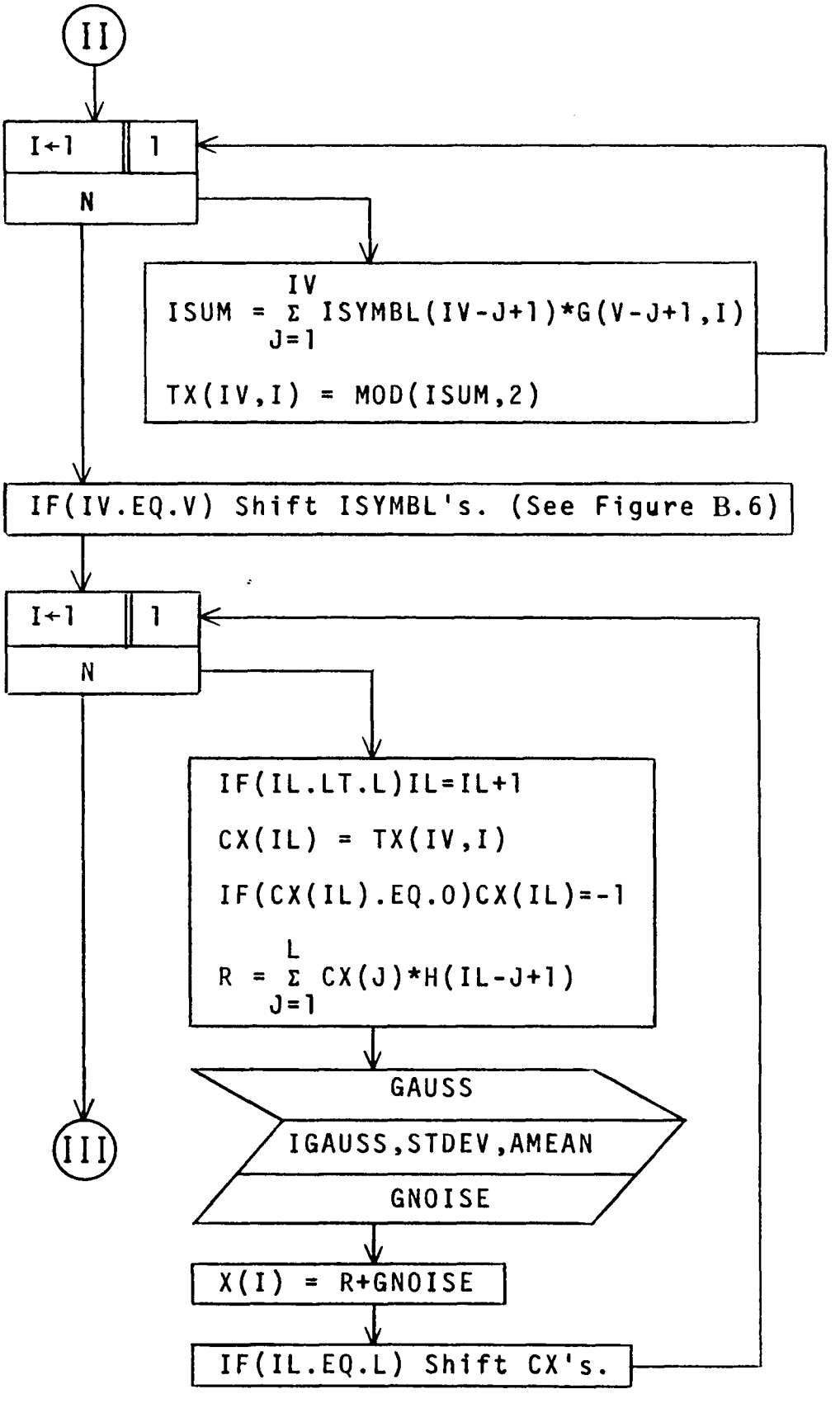


Figure C.3 Encoder And Transmitter

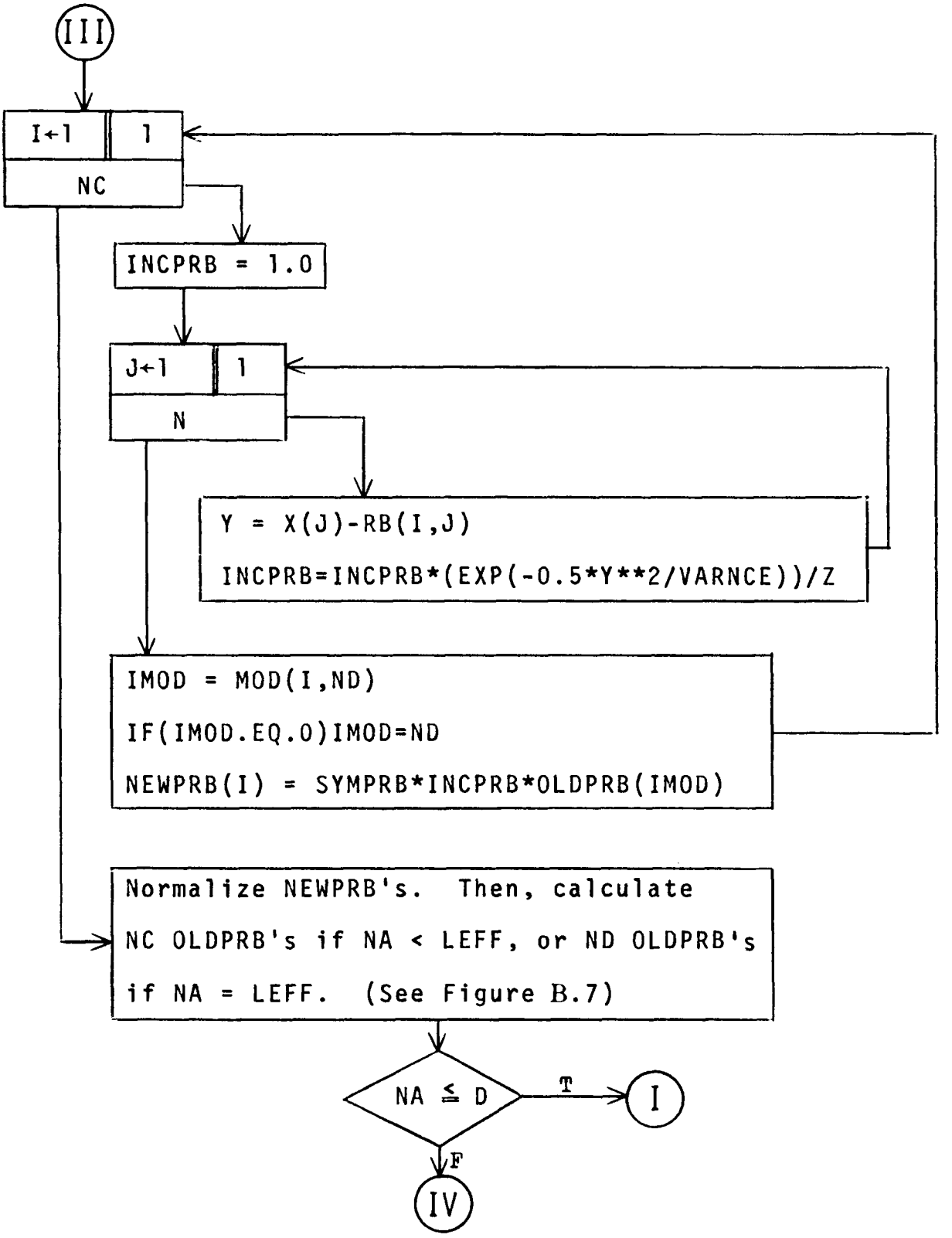


Figure C.4 Calculation of 'New' And 'Old' Probabilities

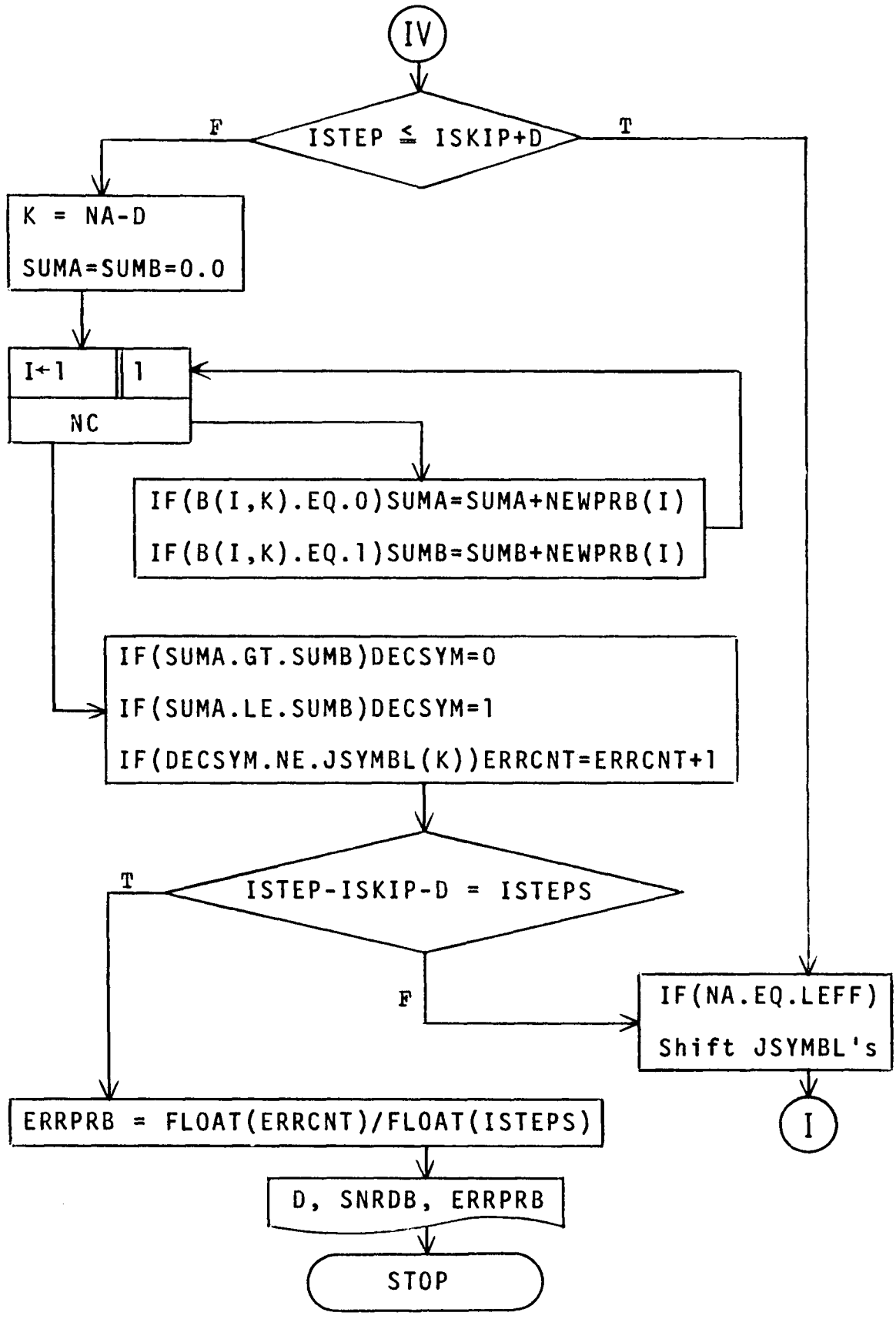


Figure C.5 Decision Segment

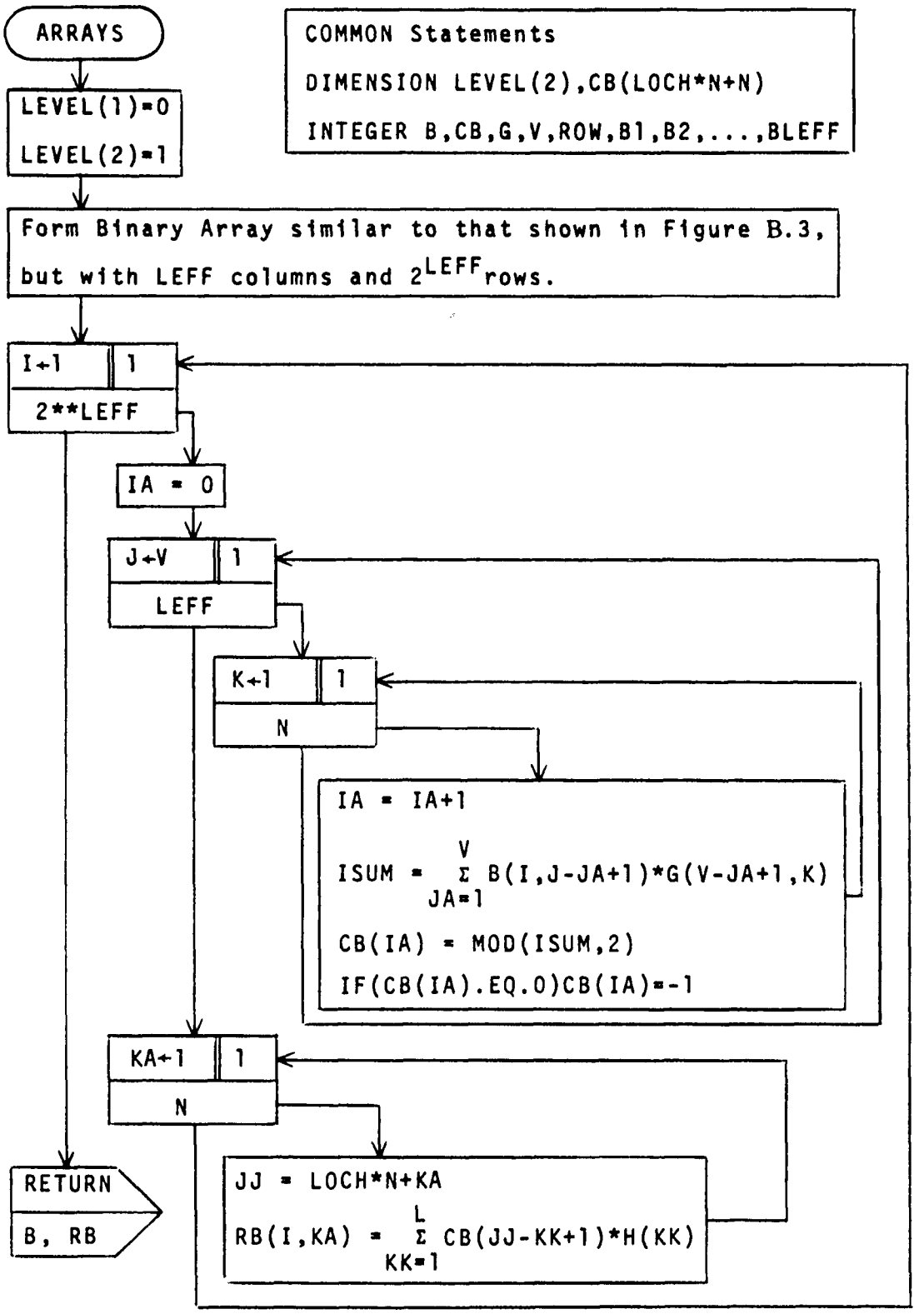


Figure C.6 Formation of Binary Array And Calculation of $\underline{R}_1$

VITA

Munaf A. Sattar, son of Aimna & Abdul, was born in the Punjab Province of Pakistan on May 13, 1951. After obtaining primary and secondary school education in Pakistan, he came to the United States toward the end of 1966. He attended Lafayette College, Easton, Pennsylvania, from 1968-72 and graduated 'cum laude' from the Department of Electrical Engineering. He was a Teaching Assistant at Lehigh University, Bethlehem, Pennsylvania, from 1972-74 where he completed requirements for his M.S.E.E. degree.

Mr. Sattar is a member of Tau Beta Pi and Eta Kappa Nu. His interests include communications, international affairs, and psychology. He is currently engaged in the pursuit of peace, love, and self-actualization.