NAG 2-304

Semi-Annual Report

to

NASA-Ames Research Center
Grant Award No. NAG 2-304

NASA Technical Officer: Karl Anderson

for

A Survey of the State of the Art and Focused Research in Range Systems - Task II

Principal Investigator
Kung Yao

Electrical Engineering Department
University of California
Los Angeles, California 90024

January 1986

(NASA-CR-177034) A SURVEY OF THE STATE OF THE ART AND FOCUSED RESEARCH IN RANGE SYSTEMS, TASK 2 Semiannual Report (California Univ.) 27 p CSCL 09B
N86-29549
G3/61 Unclas 43295

A METHODOLOGY BASED ON REDUCED COMPLEXITY ALGORITHM FOR SYSTEM APPLICATIONS USING MICROPROCESSORS

T. Y. Yan and K. Yao

ABSTRACT

Many communication, control, and information processing subsystems (such as data equalizer, array processor, whitening filter, dynamical system identifier, etc.) are modeled by linear systems incorporating tapped delay lines (TDL). Such optimized subsystems result in full precision multiplications in the TDL. In order to reduce complexity and cost in a microprocessor implementation, these multiplications (consisting of multiple-shift-and-add instructions) can be replaced by single-shift instructions which are equivalent to powers of two multiplications. Since in general the obvious operation of rounding the infinite precision TDL coefficients to the nearest powers of two usually yield quite poor system performance, we consider the optimum powers of two coefficient solution. Detailed explanations on the use of branch-and-bound algorithm for finding the optimum powers of two solutions are given. Specific demonstration of this methodology to the design of a linear data equalizer and its implementation in assembly language on a 8080 microprocessor with a 12 bit A/D converter are reported. This simple microprocessor implementation with optimized TDL coefficients achieves a system performance comparable to the optimum linear equalization with full precision multiplications for an input data rate of 300 baud. The philosophy demonstrated in this implementation is fully applicable to many other microprocessor controlled information processing systems.

T.Y. Yan is with the Communication Research Section of Jet Propulsion Laboratory, Pasadena, California 91109, and K. Yao is with the Electrical Engineering Department, University of California, Los Angeles, California 90024.

1. INTRODUCTION

A large number of multiplications is often encountered in many signal processing situations in modern communication, radar, and information processing systems. The usage of specialized multiplication devices generally increase cost, volume, weight, design time, and possibly decrease reliability. However, designs using general purpose low-cost microprocessors are flexible but yield low throughput rate when much high precision multiplications are required.

Most algorithms implemented on a digital computer are usually contaminated by various quantization effects. There are the usually A/D quantization errors at the input as well as internal arithmetical round-off errors. These errors are quite well understood [1], [2]. On the other hand, the quantization of the multiplication between the data and some basic system parameter intrinsic to the processing algorithm can be controlled to some extent by the system designer. This class of problems generally appeared not to have been studied in detail with respect to signal processing situations with microprocessor implementations.

In Figure 1, we consider a linear tapped delay line (TDL) structure which can be used to model a linear system having a finite implulse response (FIR) [1; p. 18]. This model is conceptually simple since it consists of (2N+1) multipliers, 2N delay units, and 2N summers. If the (2N+1) multiplier coefficients $\{C_n\}$ are fixed then this TDL can model a linear time-invariant system, while if these coefficients are allowed to be time-varying it can model a linear time-variant system. By allowing these coefficients to vary as functions of the changing input under various manners, we can obtain adaptive TDL systems ([3], [4; pp. 15-19]).

The TDL model is basic and is used commonly in the design and analysis of digital data equalization ([5], [6], [7; pp. 147-150]); array processing ([4; p. 400], [8]); digital whitening filtering [7; pp. 272-275]; dynamical system identification and modeling [3; p. 7]; etc. Despite these seemingly different applications, if the analytically tractable minimum mean-square error (MMSE) criterion (which is also justifiable physically from the energy criterion point of view) is used, each of the resulting optimum subsystem uses a set of full precision TDL coefficients $\{C_n\}$ to operate on the input. In practice, for a finite precision implementation using microprocessors, we need to use finite precision and preferrably some "simple" low precision coefficients in the TDL.

In this paper, we shall consider a methodology on the analysis and design of a MMSE criterion linear system incorporating a TDL where all the full precision multiplications in the TDL are constrained to be powers of two. The rationale for considering this class of problems is that without using specialized multiplication hardware devices, the implementation of high order finite precision multiplications by software routines using a microprocessor generally involves numerous multiple-shift-and-add instructions which can be quite time-consuming. However, in using only powers of two multiplications, these operations can be implemented in a microprocessor as single-shift instructions with consequent higher throughput rate. Since the obvious operation of rounding the full precision TDL coefficients to the nearest powers of two usually yields quite large system degradations, we need to find the optimum power of two TDL coefficients with respect to the MMSE criterion. As we shall see in Section 3, this optimization is somewhat involved and needs considerable computational effort. However, now we have the possibility of trading a reduction in real-time on-line computational complexity in the microprocessor implemented system without significant loss of system performance against an increase in off-line computations in the design stage. Equivalently, if we do not want to incur the engineering design cost of off-line optimization, then we can either build a more costly and complicated system using specialized hardware multiplication devices or accept a simpler microprocessor system (performing finite precision multiplications by software) with a lower throughput rate.

In order to demonstrate the philosophy and feasibility of the above discussed methodology, we choose to consider the implementation of the simplest subsystem. Thus, among the various linear systems incorporating TDL devices mentioned above, we consider the well known linear digital data equalizer. Our purpose is not to consider the most sophisticated (and thus complicated) data equalizer nor to use the latest microprocessor hardware. Our basic purpose is to demonstrate in a simple and direct manner the usefulness of the optimization methodology based on powers of two algorithms for system applications using microprocessors. In Section 2, we briefly present a linear equalizer based on the MMSE criterion for the detection and equalization of digital data over a linear dispersive and additive noise channel. In Section 3, some general concepts related to the MMSE criterion derived powers of two solutions formulated as constrained quadratic form minimization is first discussed. Then some details on the use of branch-and-bound algorithm for the solution of this problem is given. In Section 4, hardware block diagram and software flow-chart used in the implementation of this equalizer based on a 8080 microprocessor are summarized. In Section 5, some theoretical and experimental results and conclusions on the reduced multiplication complexity equalizer are given. Specifically, this simple microprocessor implementation with optimized power of two TDL coefficients achieves a system performance comparable to the optimum linear equalization with full precision multiplications for an input data rate of 300 baud. It is interesting to note that if we use regular 8 bit multiplications (in software routines) instead of powers of two left or right shifts, the above equalizer definitely cannot support the 300 baud rate. Of course, a conventional full precision implementation (using specialized multiplication hardware) with comparabale system performance and input data rate would result in a more complicated and costly system.

2. LINEAR EQUALIZER

Consider a linear equalizer for the detection of binary digital data over the linear dispersive and additive noise channel given in Fig. 2. The input digits B_k are assumed to be independent and identically distributed, taking values ± 1 with equal probability, and the data duration is T. The combined transmitter and channel impulse response function is modeled by s(t). The additive noise n(t) is assumed to be a Gaussian zero-mean wide-sense stationary random process of spectral density $S_n(\omega)$. It is well-known that, if the data equalizer is constrained to be linear, the general structure of the equalizer is actually fixed ([5; pp, 94-112], [6]). That is, the equalizer consists of a matched filter, matched to the combined transmitter and channel impulse response function s(t) and the noise process n(t), followed by a sampler with sampling rate R = 1/T, and a tapped delay line (TDL) with basic delay of T seconds between taps with coefficients $\{c_j, j = \pm N, \ldots, \pm 1, 0\}$. Different error criteria, however, affect only the tap coefficients $\{c_j\}$. For this paper we use the mean-square error criterion. Furthermore, in order for the input and the tap coefficients of the TDL to take discrete values, we impose a quantizer Q between the sampler and the TDL in Fig. 2.

The output of the TDL is given by

$$y_k = \sum_{j=-N}^{N} \tilde{r}_{k-j} c_j \tag{1}$$

where $\tilde{r}_j$ is the sampled response of the waveform r(t) after quantization.

We make the usual assumption that the quantization error is uncorrelated with the data B_k and the noise n(t). The mean-square error between B_k and the output of the TDL y_k at k = 0 can be written as

$$\epsilon = E\{(y_0 - B_0)^2\} = 1 + Q(c), \tag{2}$$

where

$$Q(c) = c\Lambda_{cc}c' - 2cu', \tag{3}$$

$$u = (u_N, \ldots, u_0, \ldots, u_{-N}), \tag{4}$$

where u_j is the sampled impulse response of the matched filter. If the data are transmitted and the matched filter is sampled at the Nyquist rate, Λ_{cc} will be a positive-definite matrix. c and ε will have unique optimum solutions in the space of real numbers. The optimum infinite precision real-valued TDL coefficient vector c is given uniquely by

$$\hat{c} = u\Lambda_{cc}^{-1}, \tag{5}$$

The optimum estimate $\hat{B}_k$ is +1 if y_k is positive or -1 if y_k is negative where $\hat{y}_k$ is given by

$$\hat{y}_k = \sum_{j=-N}^{N} \hat{c}_j \tilde{r}_{k-j}. \tag{6}$$

3. BRANCH-AND-BOUND ALGORITHM

In many practical systems, such as the linear equalizer presented in Section 2, the high precision multiplications needed in implementing the TDL equation in (6) may be objectionable. We propose the use of powers of two for each TDL coefficient c_j. A simple rounding of the optimum infinite precision TDL coefficient vector $\hat{c}$ to the nearest powers of two usually yield quite poor system performance (i.e., large M.S. error and large equalizer error probability).

Thus, it is useful to consider the optimal solution of $\tilde{c} = (\tilde{c}_{-N}, \ldots, \tilde{c}_0, \ldots, \tilde{c}_N)$ where each $\tilde{c}_j$ is constrained to be in the space

$$Z = \{z: z = \pm 2^{-t}, t \varepsilon \{0,1,\ldots,b\}\}, \tag{7}$$

where b is a specified integer.

The infinite precision solution of the TDL coefficient vector $\hat{c}$ is given by (5) and its direct implementation in (6) requires (2N + 1) multiplications. However, the presence of the matched filter causes $\{\tilde{r}_k\}$ in Fig. 1 to be symmetric around the zeroth index. This means the TDL coefficients $\{\hat{c}_j\}$ in (5) are symmetric about the zeroth index. Since $\hat{c}_0$ is an arbitrary scaling constant, it can always be set to one. Thus, the solution in (5) has only N degrees of freedom. Now we can constrain $\{\tilde{c}_j\}$ to be symmetric about the zeroth index and thus $\tilde{c}$ has (N+1) degrees of freedom. Unlike the infinite precision case where $\hat{c}_0$ is an arbitrary scaling constant,

$\tilde{c}_0$ is a parameter that needs to be optimized. The optimal solution of $\tilde{c}$ under the power of two constraint becomes

$$\operatorname*{Min}_{c \varepsilon Z^{N+1}} Q\ (c) = Q\ (\tilde{c}) \tag{8}$$

where Z is defined in (7).

A direct brute force search of all finite solutions for $\tilde{c}$ is possible in theory but not practical since the total number of points in Z^{N+1} is $[2(b+1)]^{N+1}$. For example, even for a low order TDL of N=5 and b=8, we have $18^6 = 3.4\text{x}10^7$ number of feasible solutions.

There are various approaches for solving the constrained minimization problem in (8). One practical approach for finding the optimal solution of $\tilde{c}$ in (8) is based on the branch-and-bound algorithm. This algorithm is an efficient tree search procedure for constrained optimization problems in which the constraints need not be convex and some or all of the variables have discrete values. For our constrained minimization problem, the branch-and-bound recursive operation begins by defining an extended solution space with a modified cost function. The solution space is repeatedly divided into smaller and smaller subsets and a bound is computed for the cost of the solutions within each subset. After each subdivision, those subsets with a bound that exceeds the cost of a known feasible solution are exluded from further consideration. This process continues until a feasible solution is found with a cost no greater than the bound for any subset. The precise statement of the branch-and-bound algorithm is quite complicated and lengthy. For details see ([9]-[12]).

In this paper we present the basic operations of the branch-and-bound algorithm by treating a specific two-dimensional example given in detailed graphical form in Figure 3. Consider a generalized quadratic-form Q (c) given by (3), where $c = (c_1, c_2)$. By constraining Q (c) to be some specified real number, the set of c that yields this constant contour is known to be an ellipse in two-dimension. For different constraining values of Q (c), we obtain different sets of concentric ellipses as shown in Figure 3.

The infinite precision solution $\hat{c} = (\hat{x}_1, \hat{x}_2)$ given readily in analytical closed-form by (5), yields the minimum of Q (c) and is in the center of the family of ellipses in Figure 3. This $\hat{c}$ solution is used as the initial solution (i.e., zeroth iteration) of the branch-and-bound algorithm. In Figure 3, we assumed $\varepsilon_0 = Q(c) = 5$. The admissible constrained values of (c_1, c_2) are in the sets spanned by $\{c_{11}, c_{12}, c_{13}, c_{14}\}$ and $\{c_{21}, c_{22}, c_{23}, c_{24}\}$. Since $c_{12} < \hat{x}_1 < c_{13}$ and $c_{22} < \hat{x}_2 < c_{23}$, we can perform the branching operation on either variable. By constraining $x_2 \leq c_{22}$, we find the minimum occurs at (x_{15}, c_{22}) and has a cost of $\varepsilon_1=8$. In Figure 3 and in the tree graph of Figure 4, we label this node ①. Similarly, for $x_2 \geq c_{23}$, we obtain the node ② at (x_{16}, c_{23}) with $\varepsilon_2=6$. Among these two nodes, we branch from the node with the lowest cost ε_2. Since $c_{12} < x_{15} < c_{13}$, by constraining $x_1 \leq c_{12}$ and $x_1 \geq c_{13}$, we obtain nodes ③ and ④ with $\varepsilon_3=20$ and $\varepsilon_4=9$. Among the present active nodes of ①, ③, and ④, the lowest cost is at ε_1. Branching at ① yields nodes ⑤ and ⑥. Now, the active nodes are ③, ④, ⑤, and ⑥. Since ε_4 has the lowest cost, we branch from ④ to obtain ⑦ and ⑧. In general, the algorithm proceeds in this manner until the node with the lowest cost among all the active nodes at the instant is a valid admissible constrained solution. Then the algorithm terminates and that minimum cost admissible solution is the desired solution of (8). In our example on Figure 3, we note nodes ⓪, ①, ②, and ④ are not admissible solutions, while ⑤, ⑥, ⑦ and ⑧ are admissible solutions. In the last set of active nodes {⑤, ⑥, ⑦, ⑧}, we see $\epsilon_8=10$ is lower than $\epsilon_5=12$, $\epsilon_6=25$, and $e_7=14$. Thus, we can terminate the algorithm at node ⑧ with $\tilde{c} = (c_{13},c_{24})$ and a mean-square error cost of 10. It is also interesting to observe that from Figure 3, if we had used simple round of $\hat{c}$ to the nearest admissible solution in the minimum Euclidean norm sense, then $c_r = (c_{13},c_{23})$ is given by node ⑦ and has a cost of 14.

We note that the number of nodes needed to be considered in the branch-and-bound algorithm is highly dependent on the degree of eccentricity of the associated ellipse (or ellipsoid) in the generalized quadratic-form. Indeed, if the generalized quadratic-form is a circle (or sphere), then the rounded solution is the optimum constrained solution. Unfortunately, in most practical problems, when the dimension of the problem becomes large, the associated ellipsoids are almost always highly eccentric manifested in a ratio of largest to smallest eigenvalues that is quite large [13]. In such problems, direct enumeration of all admissible solutions in Z^{N+1} is clearly impossible. Even the use of branch-and-bound algorithm can involve quite large computer storage space for the active nodes during the computation.

4. HARDWARE AND SOFTWARE DESCRIPTIONS

In Figure 5, a block diagram of the hardware used in a 300 bits/second (bps) binary data transmission system is given. The data source is a pseudo-random sequence of TTL level bit stream produced from a Wavetek 132 function generator. The equivalent transmitter and channel filter response is physically modeled by a five tap analog TDL followed by a shaping filter. This subsection is realized by using a SN74164 eight bit shift register with two LM339 quad comparators and two LM308 operational amplifiers. The resistor values in the TDL are adjusted to achieve the desired overall value of $\{\tilde{r}_k\}$ in Figure 2. The noise source is produced from a HP3722A noise generator. The noise is bandlimited white Gaussian with a bandwidth much larger than the data rate. The summer consists of two LM318 operational amplifiers and the receiving filter approximating the theoretical matched filter uses a LM308 operational amplifier as a low-pass filter with an equivalent cut-off frequency of 135 Hz. The synchronization signal is obtained from the sync output of the Wavetek 132 generator. This additional sync signal path does not exist in a real data transmission system. However, for the purpose of verifying the reduced complexity equalization concept, this approach is quite acceptable. The sample and hold subsection uses two LM308 operational amplifiers, a SN74123 monostable multivibrator, and a LM311 comparator. The A/D converter uses a low cost 12 bits AD574JD device and the interface logic and control use two 74LS367 hex tri-state buffer and one each of 7476 JK flip-flop, 7474 D flip flop, 7420 four-input nand-gate, 74LS04 hex inverter, and 7400 quad nand-gate. The data bus is then connected to a 8080 eight-bit microprocessor operating at 750 kHz clock rate.

The equalization TDL is completely implemented in software. It consists of two separate routines: a symbol detection routine and an error counting routine. Each of them is programmed separately using 8080 assembly language in two E&L Microprocessor Training Systems. For real time application, both the detection and the error counting algorithms must finish all computations before a new data symbol arrives. In this experiment, if we count the number of states that each machine lanugage executes, the detection algorithm involves much more computations than the error counting and display algorithm. The maximum allowable data rate for this software detector operating at 750 kHz clock rate is limited to 490 bps.

Software detector flow chart is shown in Figure 6 where we have initialization subroutine and detection subroutine. A hand-shaking control line interfaces the microprocessor and the A/D convertor. When the data available flag is set in the sample and hold subsection, microprocessor will enable the A/D convertor into read mode and will input the sampled 12 bits data in a sequence of 8 bits and then 4 bits. The microprocessor will create a data array from these data and compute the weighted sum according to the TDL coefficients. After finishing these computations, a threshold logic will determine the sign of the weighted average. The detected output is sent to another E&L microprocessor for error counting and display. The entire experiment will be run long enough to generate meaningful statistics.

5. NUMERICAL RESULTS AND CONCLUSIONS

In this section, we consider two explicit examples to illustrate the usefulness of the multiplication-free equalization technique. In both examples, the equalization TDL is restricted to 9 taps, while the A/D converter as well as the processing are limited to 12 bits. In the first example, the sampled channel responses at the input of the TDL are given by (0.1, 0.4, 1, 0.4, 0.1). While this channel responses used in Figure 7 only model a simplistic (i.e., low number of impulse response terms) of a highly distorted linear channel, this type of channel responses are adequate and commonly used ([15; pp. 149-150]) to compare the performances of various forms of equalizers. For this example, four sets of error probabilities as functions of SNR from 5.5 dB to 17.5 dB have been evaluated theoretically and plotted in Figure 7. The solid curve represents the infinite precision TDL performance results. The optimum 12 bits multiplication-free TDL results in the sense of Section 3 are given by the □ points. The performances of the infinite precision TDL with coefficients rounded to the nearest 12 bits multiplication-free values are given by the ○ points. The dashed curve represents the performances of the system with no TDL. For low to medium SNR values, there is slight difference between the rounded multiplication-free solution and the optimum multiplication-free solution. However, at SNR of 17.5 dB, the optimum result is almost 4 times lower in P_e as compared to the rounded result.

In the second example, the sampled channel responses are given by (0.1, 0.3, 1, 0.3, 0.1). These responses represent a fairly distorted channel with moderate intersymbol interference problems. In Figure 8, the experimentally obtained error probabilities, using the procedure discussed in Section 4, for the optimum multiplication-free case as well as for the no TDL case are presented along with the corresponding theoretical results. As can be seen, there are, in general, good agreements among the experimental and theoretical performances. The slight discrepancies at high SNR are due to the mismatch of the implemented low-pass detection filter to the theoretical matched filter. There is only a slight degradation of 0.3 dB between the reduced complexity and the infinite precision performance curves. As expected, there is a significant difference between the reduced-complexity and the no TDL results.

It is interesting to note that if we use full multiplication procedure for the weighting of each data symbol the software will not be able to keep up with the incoming data. (Software multiplication using 8080 assembly language requires at least 666 states for multiplying two unsigned 8 bit data [7]. At 1.33μS clock period, it requires 0.88 ms for a full 8 bit multiplication). This clearly demonstrates the advantage of this shift-only scheme for an efficient and low cost data equalizer based on a microprocessor implementation.

In conclusion, we have presented some analytical and practical results on the implementation of a linear data equalizer. We believe the replacement of high precision multipliers by optimized binary shifts is a useful fast processing technique applicable to various practical signal processing problems. The technique appears to be particularly attractive in conjunction with a low cost microprocessor implementation.

6. ACKNOWLEDGEMENT

This work is supported in part by the Electronic Program of the Office of Naval Research and NASA/Ames research contract NAG-2-304. The authors are appreciative of the experimental works done by Mr. D. O. Anderton.

7. REFERENCES

1. A. V. Oppenheim and R.W. Schafer, Digital Signal Processing, Englewood Cliffs, N.J.: Prentice-Hall, 1975, Chap. 9.

2. L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Englewood Cliffs, N.J.: Prentice-Hall, 1975, Chap. 5.

3. S. Haykin, Introduction to Adaptive Filters, New York: Macmillan, 1984, Chap. 4-5.

4. B. Widrow and S. D. Stearns, Adaptive Signal Processing, Englewood Cliffs, N.J.: Prentice-Hall, 1985.

5. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., Principles of Data Communication, New York: McGraw-Hill, 1968, Chap. 5-6.

6. J. G. Proakis, Digital Communications, New York: McGraw-Hill, 1983, pp.357-381.

7. A. A. Giordano and F. M. Hsu, Least-Square Estimation, New York: J. Wiley, 1985.

8. R. A. Monzingo and T. W. Miller, Introduction to Adaptive Arrays, New York: J. Wiley, 1980, pp.60-64; 433-439.

9. J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling-Salesman Problem," Oper. Res., Vol.11, pp.972-989, 1963.

10. E. Balas, "A Note on the Branch-and-Bound Principle," Oper. Res., Vol.16, pp.442-445, 1968.

11. L. G. Mitten, "Branch-and-Bound Methods: General Formulation and Properties," Oper. Res., Vol.18, pp.24-34, 1970.

12. T. Y. Yan and K. Yao, "A Multiplication-Free Solution for Linear Minimum Meon-Square Estimation and Equalization Using the Branch and Bound Principle," IEEE Trans. Information Theory, Vol.IT-12, pp.26-34.

13. K. Yao and T. Y. Yan, "On the Statistical Optimum Design of a Quantized Coefficient Recursive Digital Filter," Proc. of the European Signal Processing Conf., September 1980, pp.753-757.

14. 8080/8085 Assembly Language Programming Manual, Intel. Corporation, 1978.

15. J. G. Proakis, "Advances in Equalization for Intersymbol Interferences," Advances in Communication Systems-Theory and Applications, Vol. 4, Ed. by A. J. Viterbi, New York: Academic Press, 1975.

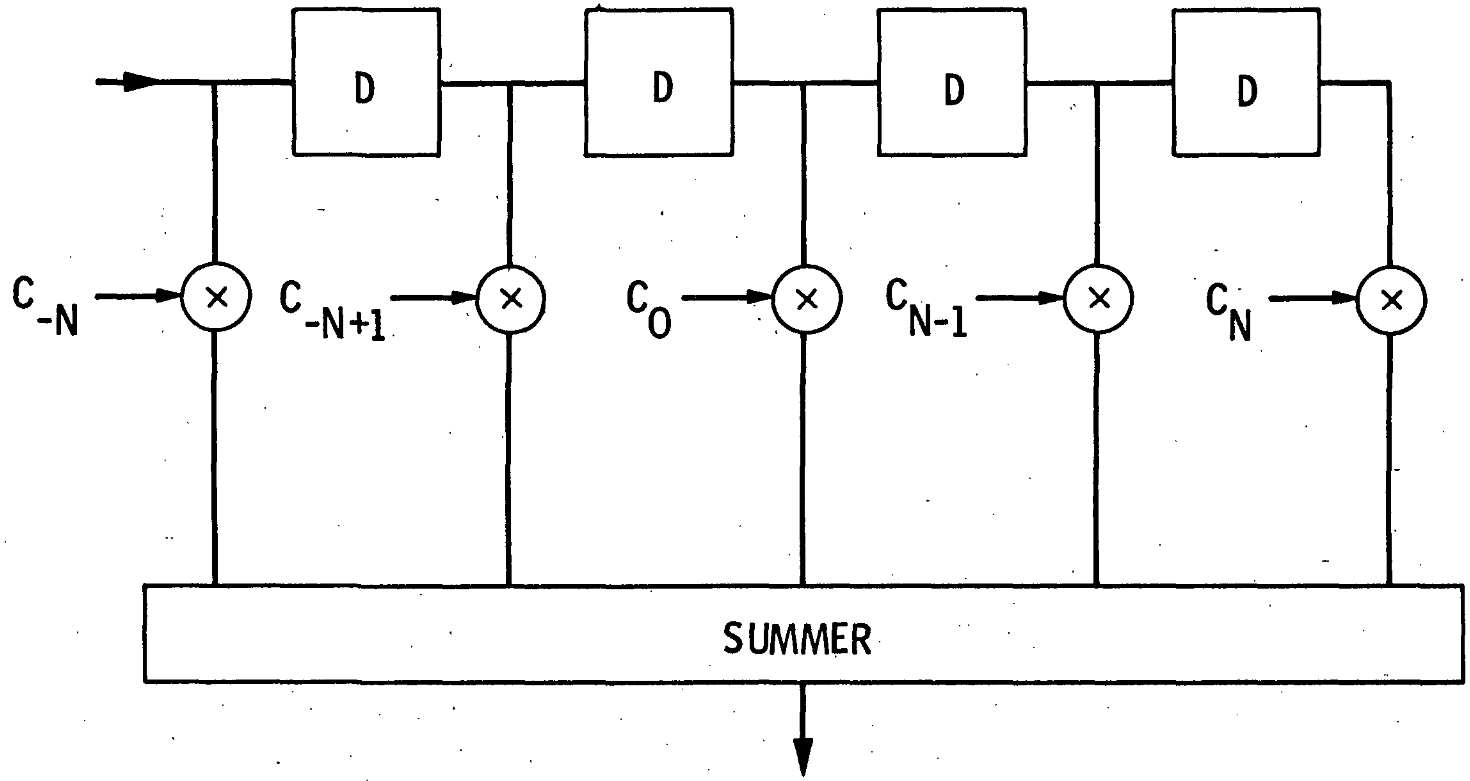

Figure 1. A Linear TDL Model

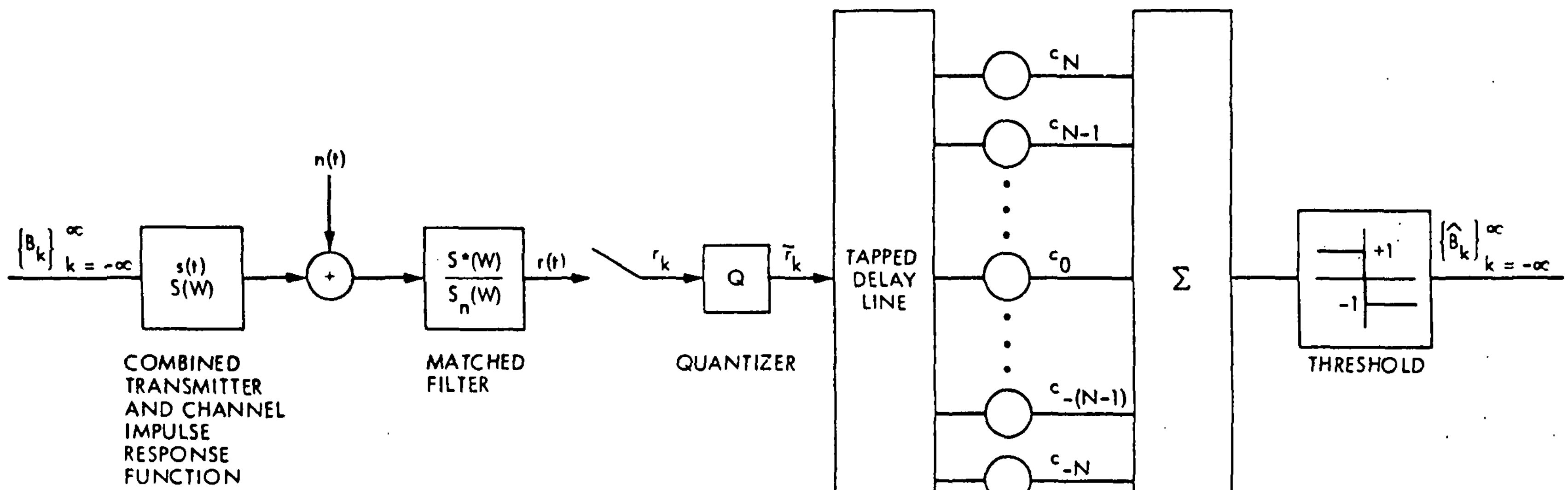

Figure 2. Linear Digital Data Equalizer

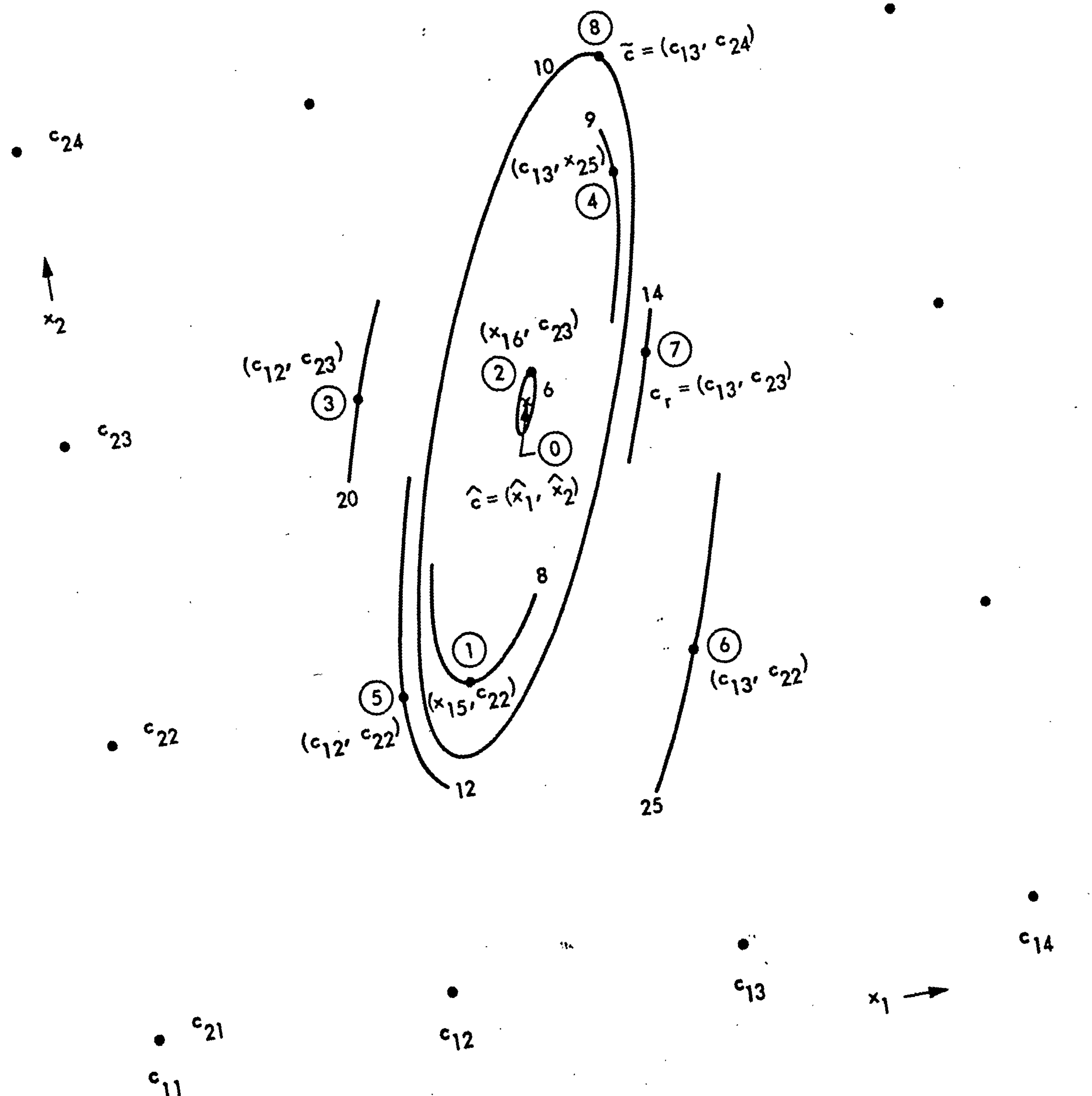

Figure 3. Two-dimensional Constant Contour of Concentric Ellipses

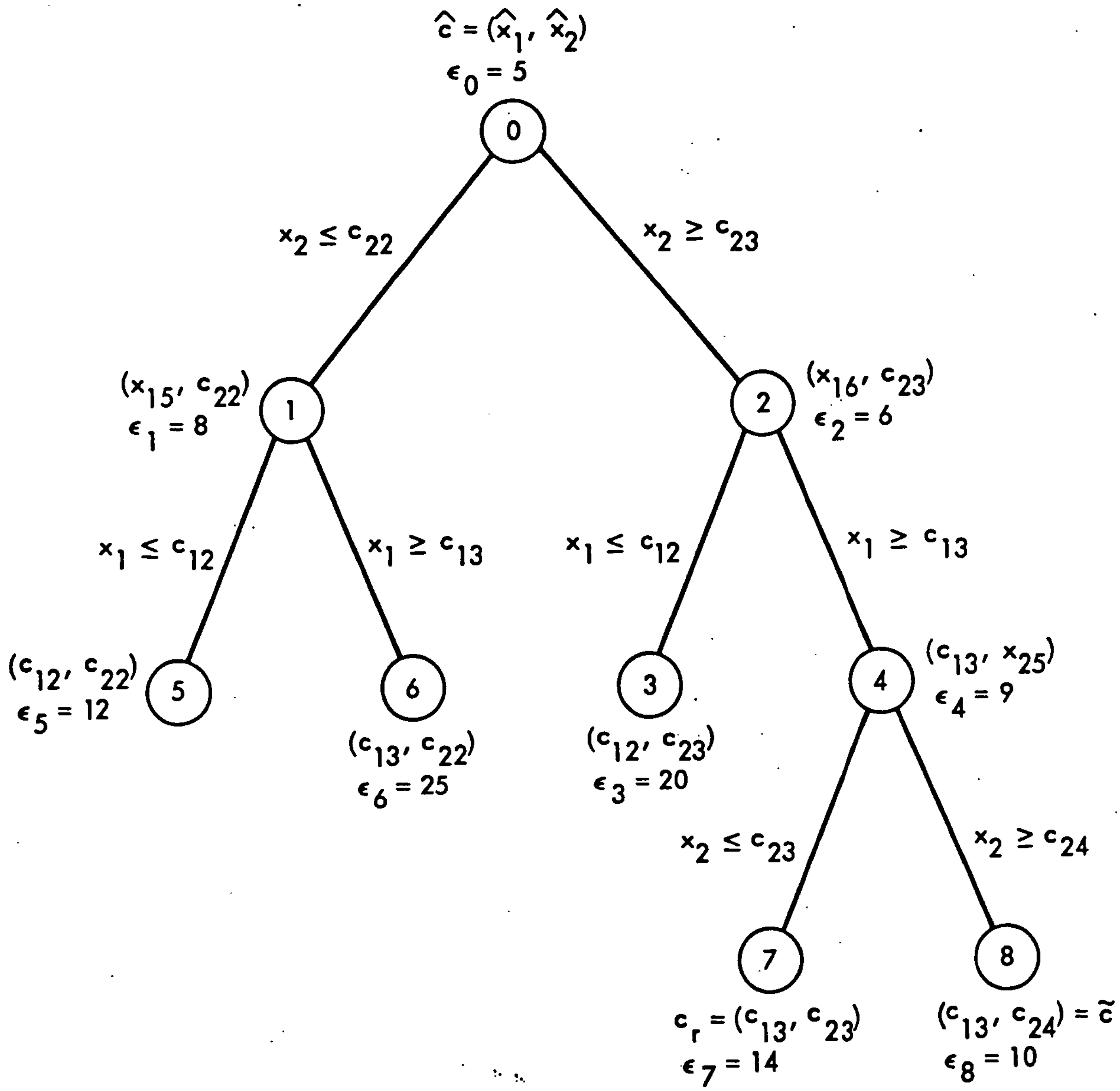

Figure 4. Solution Tree Graph of Fig. 2

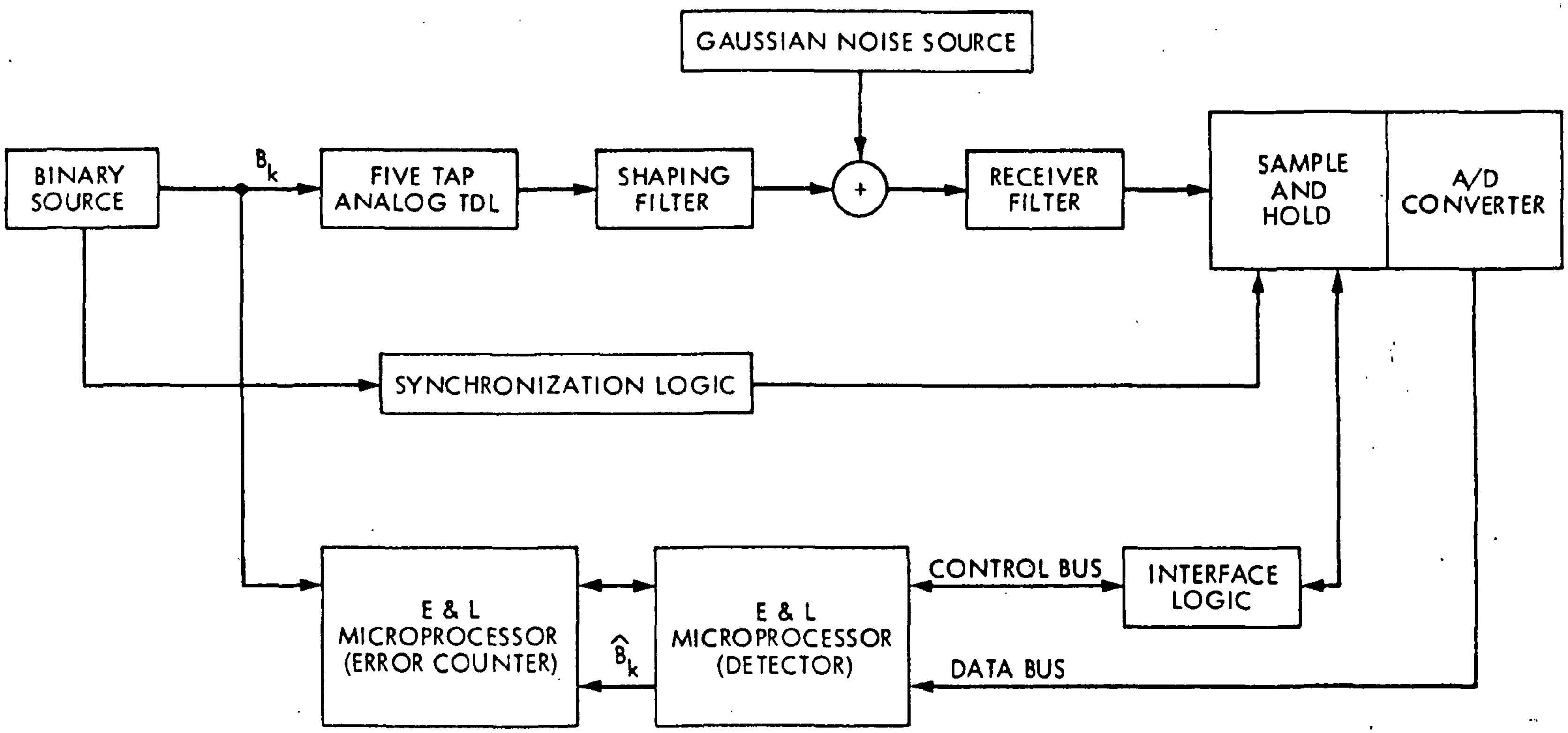

Figure 5. Hardware Subsection Block Diagram

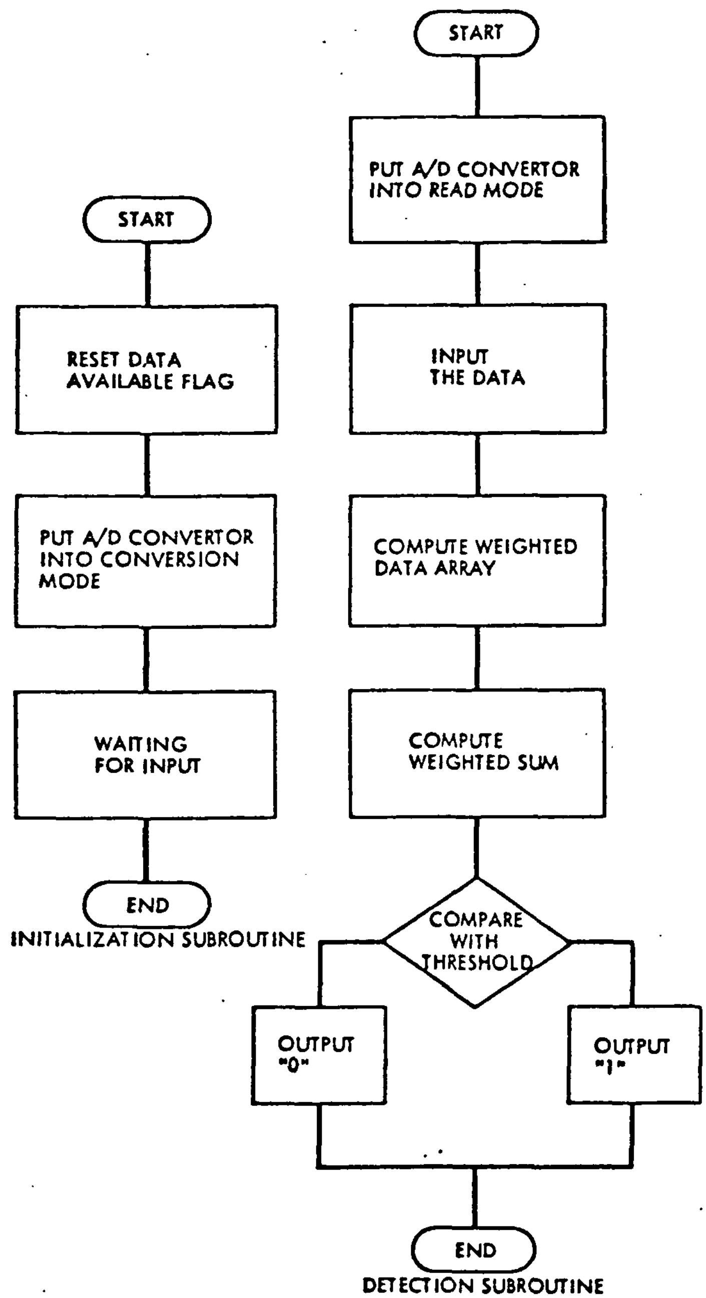

Fig. 6. Flow Chart of Detector Implemented in Software

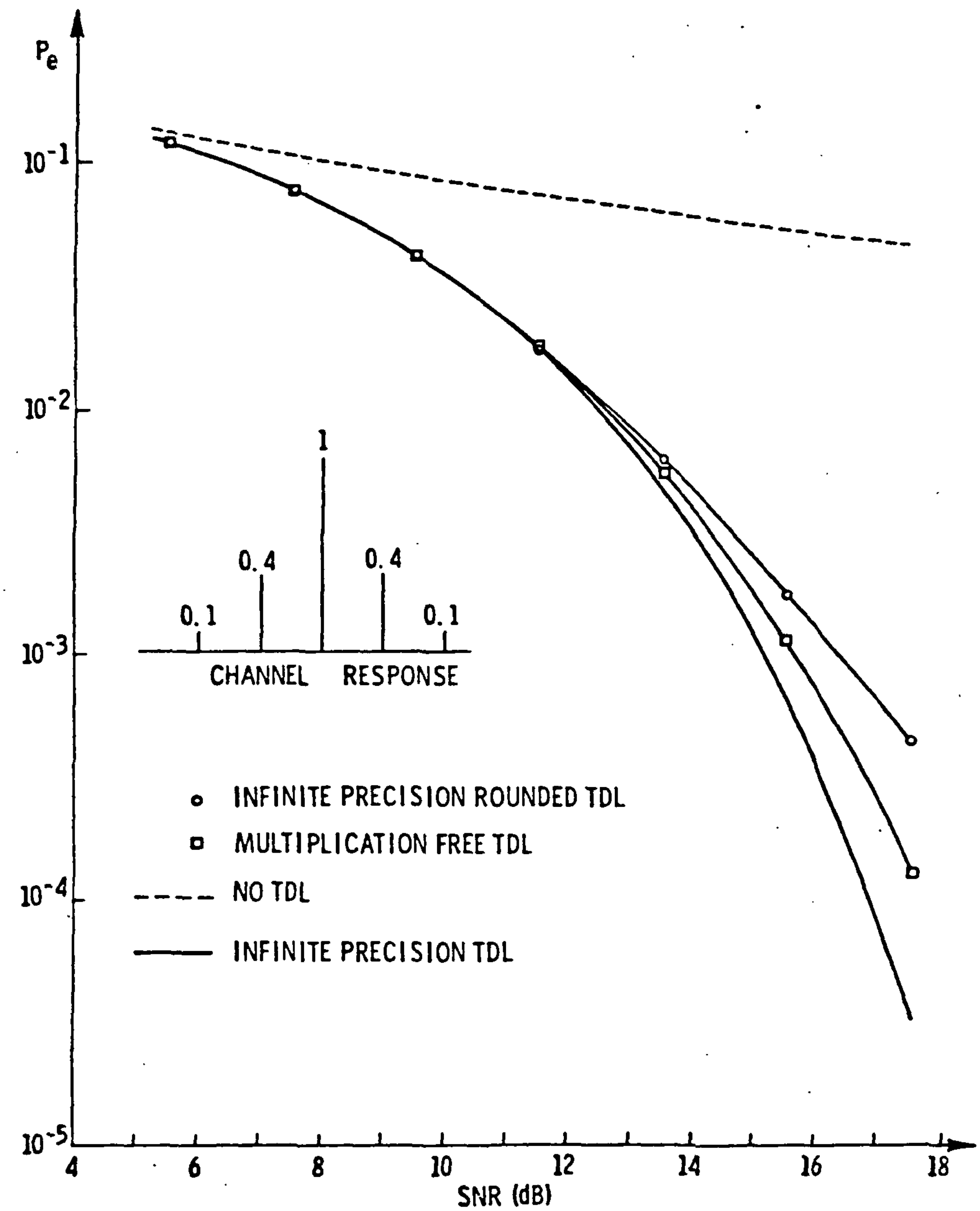

Fig. 7. Theoretical Results of Error Probability Versus SNR at the Input of Equalization TDL

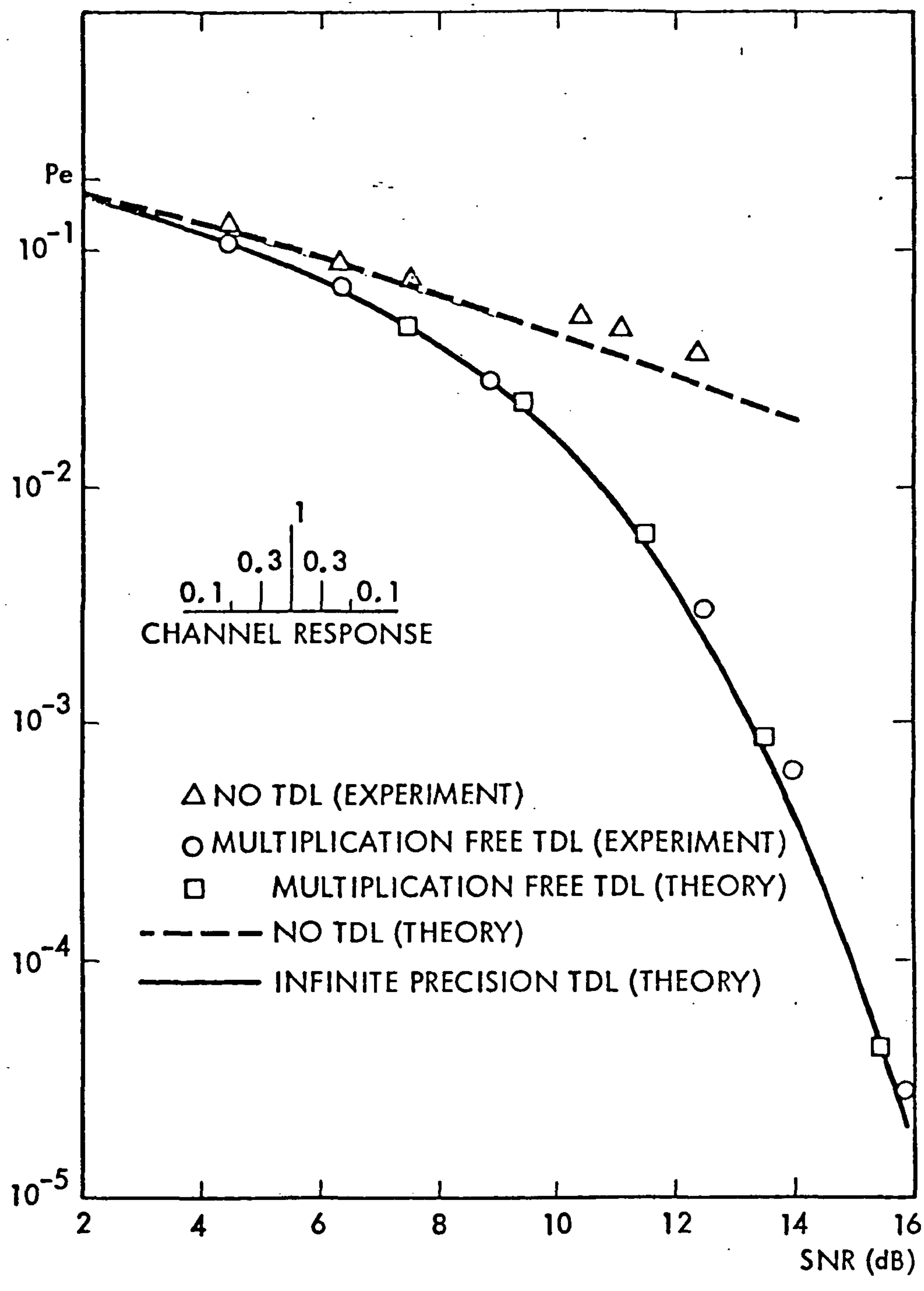

Fig. 8. Theoretical and Experimental Results on Error Probability Versus SNR at the Input of Equalization TDL