TELIDON SYSTEM STUDY

4TH INTERIM REPORT

Contract No: 1ST82-00710
MCS File No: 8342
DSS File No: DSS21ST.36100-2-4380
Date: January 31, 1984

Prepared by: J. H. Lodge, S. T. Craig

Approved by: R. K. Tiedemann

SUBMITTED BY:

MCS MILLER COMMUNICATIONS SYSTEMS LTD.
300 Legget Drive,
Kanata, Ontario,
Canada K2K 1Y5

① Moreland, K.W.

② TELIDON SYSTEM STUDY

4TH INTERIM REPORT

Contract No: 1ST82-00710

MCS File No: 8342

DSS File No: DSS21ST.36100-2-4380

Date: January 31, 1984

Prepared by: J. H. Lodge S. T. Craig
J. H. Lodge, S. T. Craig

Approved by: R K Tiedemann
R. K. Tiedemann

SUBMITTED BY:

MCS MILLER COMMUNICATIONS SYSTEMS LTD.
300 Legget Drive,
Kanata, Ontario,
Canada K2K 1Y5

TABLE OF CONTENTS

1.0 INTRODUCTION

The primary purpose of this report is to lay the groundwork for the performance evaluation of equalization and coding via simulation. In particular this report covers the analysis and algorithm development that are required for the simulation of equalization and various coding strategies. The body of the report is reserved for a concise outline of the proposed simulation approach, while further discussion and technical details can be found in the appendices.

The plan of this report is as follows. Section 2.0 briefly discusses equalization for broadcast teletext. A much more detailed discussion of this topic can be seen in Appendix I. Section 3.0 briefly discusses coding for broadcast teletext, with a more detailed treatment being given in Appendix II. The simulation strategy for evaluating the performance of equalization and coding is outlined in Section 4.0. In Section 5.0 we deviate slightly from the theme of this report and comment on some considerations concerning future measurements for broadcast teletext. The report concludes in Section 6.0.

2.0 EQUALIZATION FOR BROADCAST TELETEXT

The topic of adaptive equalization for broadcast teletext is discussed in detail in Appendix I of this report. Here, we employ some of the conclusions of Appendix I to suggest an equalization strategy to be used by the simulation. Linear transversal equalization is chosen as the most promising approach to the equalization of broadcast teletext.

Nyquist's sampling theorem was used to argue that the equalizer's tap spacing should be no more than 119 ns. This implies that a T/2 equalizer with a tap spacing of 87.3 ns is a reasonable choice. Furthermore, on the basis of the complex impulse responses that were measured by CRC, a T/2 equalizer should have at least 35 taps. However, more taps could be required if the equalizer is to be used in a bigger city or at a higher frequency. Furthermore, some of the more promising hardware approaches to realizing a programmable teletext equalizer are capable of realizing 128-tap equalizers. Therefore, an equalizer with 128 taps at a T/2 spacing, is suggested for the simulation. The equalizer should be placed between the detector and the recovery module in the simulation (see Figure 2.1).

One of the major goals of the present contract is to benchmark the potential performance gains that can be achieved by equalization, coding, and combinations thereof. Convergence and tracking performance of the adaptation algorithm are not topics included here since a study of these is a major job on its own, and these properties are highly dependent upon the algorithm, the hardware and the channels. The least mean square algorithms that are discussed in Appendix I, attempt to minimize the mean square error that is due to both noise and intersymbol interference. As such, the optimal (in the MSE sense)

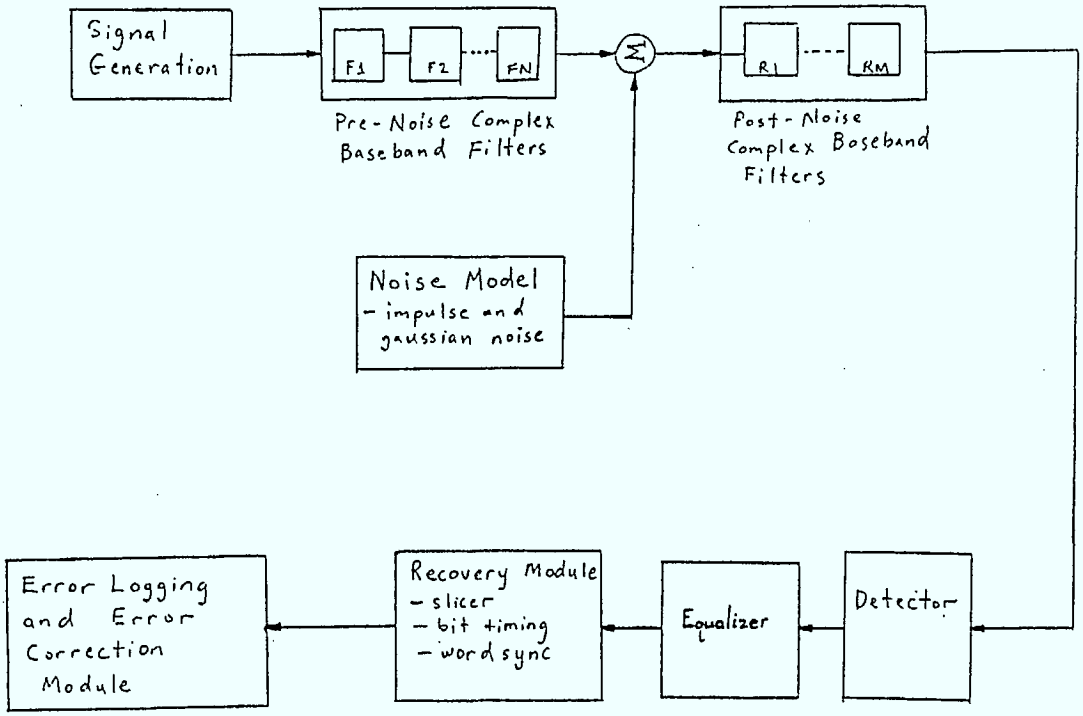


Figure 2.1: A block diagram of the simulation program incorporating the equalizer.

equalizer coefficients are dependent on both the noise and the channel. For the simulation it will be desirable to make several runs, with different types of noise, each at several different signal to noise ratios, for each channel investigated. In order that new equalizer coefficients do not have to be computed for each run with a given channel, it is desirable that an equalization strategy that depends only on the channel be employed to benchmark performance. Certainly, the effects of the noise should not be ignored by the equalization strategy. With these points in mind, we outline an equalization strategy that minimizes the noise power at the data detector, under the constraint that intersymbol interference is completely eliminated. It is assumed that the spectrum of the noise is white. Note, however, that the proposed scheme can easily be extended to account for any known noise spectrum.

The complex baseband model, used here, can be see in Figure 2.2. In this illustration $H_I(f)$ is the transfer function from the input of the baseband teletext transmit filter to the input of the linear equalizer. If

$$H(f) = P(f)T(f)C(f)R(f)L(f), \tag{2.1}$$

then

$$H_I(f) = \frac{1}{2}[H(f) + H^*(-f)]. \tag{2.2}$$

Also,

$$N(f) = R(f)L(f). \tag{2.3}$$

It is assumed that $H_I(f)=0$ for $|f|>f_s$, where f_s is the symbol rate. It can be shown [1] [2] that there is no intersymbol interference if

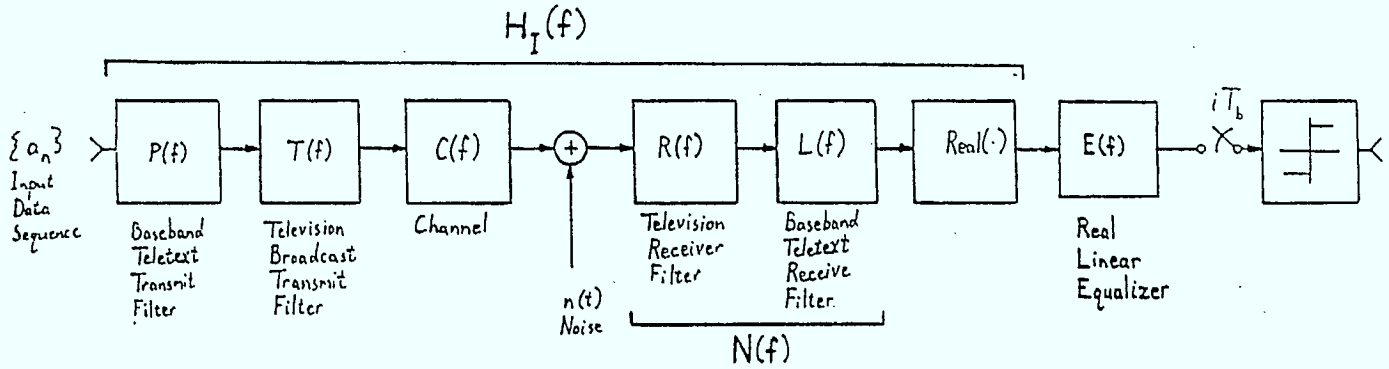


Figure 2.2: The complex baseband model.

$$G(f) = \sum_{k=-\infty}^{\infty} H_I(f-kf_s)E(f-kf_s) = K, \tag{2.4}$$

where K is a real constant and E(f) is the equalizer transfer function. Given the above assumption, the constraint given in equation (2.4) reduces to

$$H_I(f)\ E(f) + H_I(f-f_s)E(f-f_s) = K, \tag{2.5}$$

for $0 < f < f_s$. Thus for no intersymbol interference

$$E(f-f_s) = \frac{K-H_I(f)E(f)}{H_I(f-f_s)}, \tag{2.6}$$

for $0 < f < f_s$. In general, there are an infinite number of choices for the equalizer transfer function, E(f), that satisfies the constraint given in Equation (2.6). One attractive possibility is to choose the equalizer that minimizes the noise bandwidth given by

$$I = \int_{-\infty}^{\infty} W(f)E(f)E^*(f)\ df \tag{2.7}$$

where W(f) is the real non-negative function given by

$$W(f) = N_I(f)\ N_I^*(f) + N_Q(f)N_Q^*(f), \tag{2.8}$$

$$N_I(f) = \frac{1}{2}\left[N(f) + N^*(-f)\right], \tag{2.9}$$

$$N_Q(f) = \frac{1}{2}\left[N(f) - N^*(-f)\right]\ . \tag{2.10}$$

Under the given assumptions, minimizing I is equivalent to minimizing

$$J(f) = W(f)E(f)E^*(f) + W(f-f_s)E(f-f_s)E^*(f-f_s), \quad (2.11)$$

for all f such that $0 \leq f \leq f_s$. Substitution of Equation (2.6) into Equation (2.11) yields

$$J(f) = W(f)\left|E(f)\right|^2 + W(f-f_s)\left|\frac{K-H_I(f)E(f)}{H_I(f-f_s)}\right|^2 . \quad (2.12)$$

In order to determine the value of E(f) that minimizes J(f), we differentiate J(f) with respect to E(f) and set the resulting expression to zero. This yields

$$\frac{dJ(f)}{dE(f)} = 2W(f)E(f) - 2W(f-f_s)\left(\frac{K-H_I(f)E(f)}{H_I(f-f_s)}\right)\left(\frac{H_I^*(f)}{H_I^*(f-f_s)}\right) \quad (2.13)$$

$$= 0.$$

Solving for E(f) yields

$$E(f) = \frac{KW(f-f_s)H_I^*(f)}{W(f)\left|H_I(f-f_s)\right|^2 + W(f-f_s)\left|H_I(f)\right|^2} . \quad (2.14)$$

Equation (2.14) can be used to determine E(f) for $0 \leq f \leq f_s$, while (2.6) can be used to determine E(f) for $0 > f > -f_s$. The impulse response that corresponds to the transfer function E(f) must then be truncated to 128 consecutive samples with a spacing of T/2.

3.0 CODING FOR BROADCAST TELETEXT

Some excellent codes have been developed for the broadcast teletext application, under the direction of CRC [3-8]. The various codes that have been proposed include the product code [8], the one-byte CARLETON code [3], the two-byte C code [4], the bundle code [4], the one-byte SAB code [5], the two-byte SAB code [6], and the three-byte SAB code [6]. Of these codes the one-byte SAB code is of little interest because of its sensitivity to short error bursts [3]. The remaining codes can all be implemented for the simulation within a common framework. The implementation of these codes, for the simulation, is discussed in detail in Appendix II.

It is the product code that is specified as the one-byte code in BS-14 [9] and NABTS [10]. For this reason any code that is not compatible with the product code is unlikely to be incorporated into future broadcast specifications. Fortunately, a couple of very powerful codes (i.e. the C code and the bundle code) have been developed at Carleton University that are fully compatible with the product code. Thus a teletext decoder that can only decode the product code can still be used to receive data that is encoded using the C code or the bundle code. In fact the product code, C code, and bundle code represent an upward compatible family of codes in that the code words of the bundle code are composed of codewords of the C code, and the codewords of the C code are a subset of the codewords of the product code*. Thus, if the teletext data is encoded using the bundle code, it may be received using the bundle code, the C code, the product code, or just byte parity; depending upon which of these options represent the best cost-performance

*The codewords of the C code are also a subset of the codewords of the one-byte CARLETON code.

balance for the individual customer. It is this upward compatibility, along with the excellent performance potential of the C code and the bundle code, that makes this family of codes the recommended target for the simulation effort. It is pointed out in Appendix II that all of these codes can be evaluated simultaneously using data that has been encoded using the bundle code. This is because the computations, that are required for decoding the bundle code, include most of the computations for implementing the byte parity code, the product code, the one-byte Carleton code, and the C code. Thus the upward compatibility results in an advantage for implementing the simulation, in addition to the operational advantage that has already been mentioned.

The effect of severe multipath propagation of the performance of coding strategies should be briefly discussed. Due to intersymbol interference a particular bit sequence will result in a particular sequence of errors with a much higher probability than if the errors were caused by noise alone. This can result in much longer delays than would be predicted using the types of error models suggested in [3] and [11], which do not include be above effect. As an example, consider the case where the eye is closed and the signal to noise ratio is high. In this case there will be certain error sequences that will occur with high probability whenever the corresponding data sequences are sent. If a packet results in such an error sequence, that is detectable but not correctable, the decoder will wait for the packet to be sent again. However, when the packet is sent again the same error pattern will reoccur with a high probability. Thus the decoder can get caught in a state where it is waiting for a very long time. One way to avoid

this problem is to scramble the data. Thus although two bursts may contain the same information, they will consist of different bit sequences. This solution is mentioned for completeness only. It is not recommended because of the resulting increase in complexity of the receive equipment.

In Appendix II and in [11], relatively simple error models are suggested for the generation of error sequences that can be used to compare the various coding strategies. Although the error model approach has some advantages it is not recommended here, in part because of the above deficiency. The recommended approach is outlined in the following section.

4.0 THE SIMULATION STRATEGY FOR EQUALIZATION AND CODING

The simulations involving equalization and coding should help to answer two major questions. How do the performance gains achievable by equalization and coding compare, over "acceptable" multipath channels? How does the performance of the various codes compare? Here, we begin by considering the first question.

It is suggested that the simulation program be run for a couple of different fixed "acceptable" multipath channels, in order to evaluate the relative performance of equalization and coding. Each channel should be tested for both Gaussian noise and a Gaussian impulse noise mix, at several different SNR_{TV}. For each run, the packet rejection rate and the average number of PDI errors per record should be computed and tabulated for the cases when only equalization is used, only coding is used, both equalization and coding are used, and neither equalization nor coding is used. Here, by coding we are refering to the family of codes that are discussed in the previous section. Note that for any given run (i.e. a given channel, noise type, and SNR_{TV}), the number of packets sent is limited to no more than several thousand packets, due to the required computation time. Thus it is unlikely that these runs will be able to accurately quantify the performance of the more powerful coding schemes for high SNR_{TV}. Therefore additional simulation effort is necessary to compare the coding schemes.

One method that has been proposed for comparing the various coding schemes involves the use of relatively simple error models. However, measurements by CRC have resulted in an extensive collection of off-air error sequence records.

These records obviate the need for an error model, and in fact the use of actual error records is clearly preferable to the error model approach since the actual records will incorporate effects that are far beyond the scope of an error model.

From the above, the simulation effort for equalization and coding will consist of two parts. The performance of equalization and coding will be compared using the present teletext simulation program. Then the performance of the high power codes can be evaluated using CRC's error sequence records.

5.0 SOME COMMENTS ABOUT FUTURE MEASUREMENTS FOR BROADCAST TELETEXT

In this section we briefly comment on several points that concern future measurements for broadcast teletext. This discussion is not intended to be comprehensive in any way, but rather its purpose is to highlight several points that should be considered for future decision making. Although this section departs from the overall theme of this report, it has been included at the request of the scientific authority* for this contract. This section begins by discussing the tradeoffs between the measurements of the real baseband signals and the measurement of the complex baseband signals. The the measurement of noise is considered.

The Communications Research Centre is presently considering equipping a van with measurement equipment for the broadcast Telidon application. There has been some controversy over whether this van should be equipped to analyze only the inphase component (i.e. the real baseband signal) of the received signal or both the inphase and quadrature components (i.e. the complex baseband signal) of the received signal. Since the actual baseband signal** is real at both the transmitter and the receivers only inphase signal is required for most purposes. Furthermore, equipping the van to analyze only the inphase component is less expensive than equipping the van to analyze both the inphase and quadrature components. However, there are some penalties associated with having access to only the inphase component. Before the least expensive route is chosen,

*Dr. M. Sablatash of the Communications Research Centre, Shirley's Bay, Ontario, Canada.
**By "actual baseband signal" we are referring to the signal prior to up-conversion at the transmitter or following down-conversion (or detection) at the receivers.

these penalties should be fully appreciated. Thus the following discussion highlights a couple of the advantages of having the capability to analyze both the inphase and the quadrature components of the receive signal.

If ideal synchronous detection (i.e. there is no phase jitter and the mixer behaves like an ideal multiplier) were employed at the receiver, then only the real baseband impulse response of channel would be necessary to compute the noiseless channel output (following detection) given the real baseband input (i.e. the baseband signal at the transmitter). However, ideal synchronous detection is not a good model for the detection processes that are typical of consumer television receivers. Most of the television sets that are currently being manufactured use quasi-synchronous detectors. Due to the wide bandwidths of these detectors, they exhibit relatively large phase jitters. In order to get a feel for what this phase jittering does to the apparent real baseband channel, consider the effect of a fixed phase offset. Let H(f) denote the transfer function of the complex baseband channel. Recall [12] that the transfer function H(f) can be decomposed into two components, a symmetrical component

$$H_I(f) = \frac{1}{2}\left[H(f) + H^*(-f)\right],$$

and an antisymmetrical component

$$H_Q(f) = \frac{1}{2}\left[H(f) - H^*(-f)\right],$$

where H^* denotes the complex conjugate of H. The global symmetry, about f=0, of $H_I(f)$ guarantees that the corresponding impulse response, $h_I(t)$, is purely real.

Therefore, $H_I(f)$ represents the transfer function of the inphase channel. The global antisymmetry, about f=0, of $H_Q(f)$ guarantees that the corresponding impulse response, $h_Q(t)$, is purely imaginary. Therefore, $H_Q(f)$ represents the transfer function of the quadrature channel.

Consider the case where there is some phase offset, θ, due to nonideal carrier recovery. In this case the apparent channel transfer function becomes

$$\hat{H}(f) = cH(f),$$

where $|c|=1$ and $\arg(c) = \theta$. It is straightforward to show that the resulting real impulse response is

$$h_\theta(f) = (\cos\theta)h_I(f) - j(\sin\theta)h_Q(t).$$

Note that this impulse response is a function of both the inphase and the quadrature impulse responses as well as of the phase offset θ. When phase jittering is present, θ is time-varying. For quasi-synchronous carrier recovery the phase jitter is typically dominated by Nyquist slope incidental phase modulation, which is a function of the received signal. The resulting situation can only be accurately modelled and analysed when the complex baseband impulse response is known. Knowledge of only the real (i.e. inphase) baseband impulse response is not enough.

Many of the older television sets use envelope detection. It is well known that envelope detectors introduce envelope distortion. Thus, again both the inphase and the quadrature signals are necessary if accurate modelling and analysis is to be performed. The obvious conclusion here is that both the inphase and quadrature signals are necessary if either of the commonly used detectors are to be modelled and analysed.

Another disadvantage of ignoring quadrature information is that it restricts the utility of the transfer function concept. A very powerful property, that is often used in the analysis, modelling, and measurement of linear systems, is that if two linear systems, with transfer function $H_1(f)$ and $H_2(f)$, are cascaded then the resulting overall transfer function is given by

$$H_3(f) = H_1(f)H_2(f).$$

This property is so basic and used so often that it can easily be taken for granted. Unfortunately, in general

$$H_{I3}(f) \neq H_{I1}(f)\ H_{I2}(f),$$

where $H_{Ik}(f)$ is the conjugate symmetrical component of $H_k(f)$. To illustrate the possible consequences of this observation, consider a simple example. Suppose that we have determined the real baseband impulse response (with the corresponding transfer function $H_I(f)$) of a given teletext channel. Also, suppose that we wish to add an IF or RF filter, with a known transfer function, to the transmission link. We would not have enough information to determine the resulting real impulse response.

In conclusion, although restricting the measurements to the inphase channel is the most economical option, there are a couple of rather severe restrictions with this approach.

A mix of Gaussian and impulse noise seems to be appropriate (from the open literature, e.g. CCIR) for modelling noise in an urban environment for the VHF/UHF bands. However, typical ratios of the powers of these two types of noise are not readily available. Often it is assumed that the man-made noise is purely impulsive and that the Gaussian noise

is due to receiver and galactic noise. Then typical ratios are determined from standard curves. This approach should be either verified or refuted, empirically. Also, typical amplitude probability density functions for the impulses are desirable. Often Rayleigh or log-normal functions are used.

The importance of using a realistic noise model cannot be over emphasized because simulations have shown that the performance is quire sensitive to the assumed noise model. The problem, in practice, is that the noise is variable as a function of location, frequency (i.e. channel), and time. Thus the measurements should include records of the noise taken at various times, locations, and channels. Fitting these records to an appropriate noise model (which may be very difficult) can then be done in software.

6.0 CONCLUSIONS

The primary purpose of this report is to lay the groundwork for the performance evaluations of equalization and coding via simulation. Section 2.0 briefly discusses equalization for broadcast teletext. A much more detailed discussion of this topic can be seen in Appendix I. Section 3.0 briefly discusses coding for broadcast teletext, with a more detailed treatment being given in Appendix II. The simulation strategy for evaluating the performance of equalization and coding is outlined in Section 4.0. In Section 5.0 we deviate slightly from the theme of this report and comment on some considerations concerning future measurements for broadcast teletext.

One of the goals of this study is to compare the performance of equalization and coding. Thus it may be worthwhile to discuss the advantages and disadvantages of these techniques, which may impact on the results.

Linear equalization has the advantage that it can improve the synchronization, when multipath propagation is present, if the equalizer precedes the synchronization circuitry. Real-time equalization has the advantage that the reception of the prefix can be improved (unlike with coding). Also, linear equalization could potentially be employed to improve the video reception in addition to the teletext reception. One drawback of linear equalization is that a linear equalizer can smear a strong impulse, resulting in poor performance in a strong impulse noise environment. Furthermore, the implementation of a suitable equalizer is likely to be relatively expensive.

The primary advantage of coding is that fairly economical implementations have been developed for some very powerful codes [3] [4]. However coding reduces the throughput of data. Also, coding is unlikely to be effective over severe multipath channels (i.e. the eye is closed) because the same error pattern is likely to occur every time a given data sequence is sent.

Of course before the advantages and disadvantages, associated with equalization and coding, can be weighed it is necessary to determine the potential performance gains. It is hoped that the forthcoming simulations will help to identify these potential performance gains.

REFERENCES

[1] R. A. Gibby and J. W. Smith, "Some Extensions of Nyquist's Telegraph Transmission Theory", Bell System Technical Journal, pp. 1487-1510, September 1965.

[2] H. Nyquist, "Certain Topics in Telegraph Transmission Theory", AIEE Trans. 47, pp. 617-644, 1928.

[3] B. Mortimer, "A Study of the Use of Error-Correcting Codes in Broadcast Telidon", Carleton University Report, DSS Contract No. OSV81-00095, February 1982.

[4] B. Mortimer and M. Moore, "A More Powerful Error-Correction Scheme for the Broadcast Telidon System", Carleton University Report, DSS Contract No. OSU82-00164, March 1983.

[5] P. E. Allard et al., "Realization, Economic and Performance Analysis of Error-Correcting Codes and ARQ Systems for Broadcast Telidon and other Videotex Transmission", Concordia University Report, DSS Contract No. OSU80-00133, June 1981.

[6] K. Hari et al., "Further Results on High Rate Codes for Byte-Oriented Information Systems", Concordia University Report, February 1983.

[7] G. E. Seguin et al., "A Class of High Rate Codes for Byte-Oriented Information Systems", IEEE Trans. on Comm., Vol. COM-31, pp. 334-342, March 1983.

[8] M. Sablatash and J. Storey, "Determination of Throughput, Efficiencies and Optimal Block Lengths for an Error-Correcting Scheme for the Canadian Broadcast Telidon System", Can. Elec. Eng. J., vol. 5, no. 4, 1979.

[9] "Broadcast Specifications BS-14", Department of Communications, Government of Canada, Ottawa, Ontario, Issue 1, June 1981.

[10] "Draft North American Broadcast Teletext Specification (NABTS)", EIA/CVCC, September 20, 1983.

[11] K. W. Moreland et al., "Telidon System Study 2nd Interim Report", Miller Communications Systems Ltd. Report, DSS File No. DSS 21ST.36100-2-4380, MCS File No. 8342, June 1983.

[12] K. W. Moreland and J. H. Lodge, "Telidon System Study 3rd Interim Report", Miller Communications Systems Ltd. Report, DSS File No. DSS 21ST.36100-2-4380, MCS File No. 8342, September 1983.

APPENDIX I

ADAPTIVE EQUALIZATION

FOR BORADCAST TELETEXT

APPENDIX I: ADAPTIVE EQUALIZATION FOR BROADCAST TELETEXT

1.0 INTRODUCTION

The purpose of this appendix is to discuss possible approaches to the adaptive equalization of broadcast teletext signals and then to propose a practical adaptive equalizer that can be implemented in hardware and software. In this section, we begin by outlining the complex baseband signal transmission model that will be used here. This is followed by a general discussion about the equalization problem that will be addressed in the remainder of the appendix.

It is assumed here that some form of coherent demodulation is used at the receiver. Therefore the complex baseband model, that is shown in Figure 1, can be applied. Here, P(f) is the teletext transmit filter. Since this filter is implemented at baseband, it will have a purely real impulse response. T(f) is the effective television broadcast transmit filter. It includes the effects of the transmitter's IF filter, zonal filters, harmonic filter, and any fixed (but typically ajustable) equalizer. Since this frequency response must be asymmetrical about the carrier frequency, its baseband representation will have a complex impulse response. C(f) is the frequency response of the channel, including multipath and the transmit and receive antennas.

For the purpose of discussion it will sometimes be assumed that C(f) is of the form

$$C(f) = \hat{C}(f)\left[\sum_{p=1}^{P} c_p(t)e^{-j2\pi f T_p(t)}\right], \qquad (1)$$

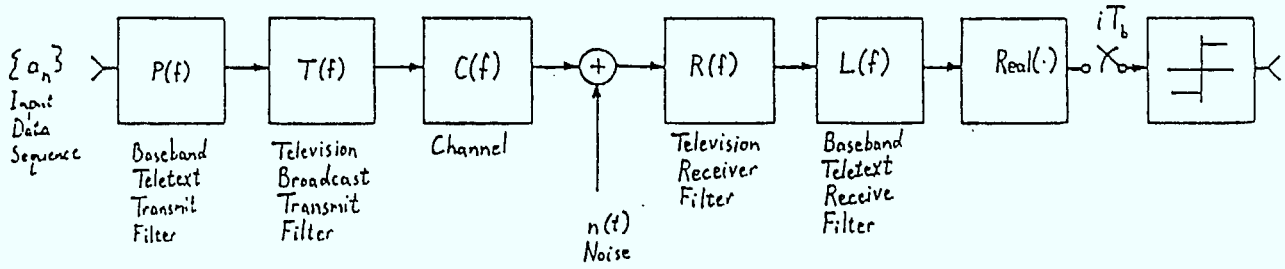


Figure 1 : The complex baseband model for teletext transmission.

where $\hat{C}(f)$ is the channel response in the absence of multipath, P is the number of propagation paths, and $c_p(t)$ and $T_p(t)$ are the complex path gains and the path delays, respectively. Note that the angles of the complex path gains can incorporate any phase offset that is present in the recovered carrier, which in turn is a function of the multipath structure of the channel and the carrier recovery circuitry. The path gains are represented as time-varying functions so that effects such as carrier recovery phase jitter, reflections from moving surfaces (e.g. airplanes), and movement of the transmit antenna due to wind can be accounted for. R(f) is the effective television receiver filter. For vestigal sideband demodulation R(f) must be asymmetrical about the carrier frequency. Therefore its baseband representation will have a complex impulse response. L(f) is the teletext receive filter. This filter will probably be implemented at baseband and therefore will have a real impulse response.

The model illustrated in Figure 1 does not include an adaptive equalizer. The location of the equalizer in this model will be discussed shortly.

In general, adaptive equalization can be used to compensate for a variety of signal transmission impairments, which include:

- Multipath propagation.
- Quadrature distortion.
- Deviations from nominal of the channel and filters (see Figure 1).
- Nonlinear distortion.
- Coloured noise and interference.

In the next few paragraphs each of the above impairments will be briefly discussed.

The first three impairments can all be viewed as linear (but possibly time-varying) distortions in the signal transmission chain. For this reason, these three impairments are grouped together for this discussion. In order to facilitate this discussion we begin by considering transfer functions for the model shown in Figure 1. Here, we are restricting our attention to the case where the complex path gains and the path delays are stationary. Let the transfer function from the input of the baseband teletext transmit filter to the output of the baseband teletext receive filter be given by

$$\hat{H}(f) = H(f)\left[\sum_{p=1}^{P} c_p e^{-j2\pi f T_p}\right]. \qquad (2)$$

Here,

$$H(f) = P(f)T(f)\hat{C}(f)R(f)L(f) \qquad (3)$$

is the transfer function of the channel in the absence of multipath, and the summation of the right-hand side of equation (2) represents the multipath propagation. Also let the argument of c_p be denoted by θ_p. Any deviation from nominal of the channel and filters will manifest itself as a deviation from nominal of H(f). Thus the actual intersymbol interference could be significantly worse than for the nominal H(f).

Here, we begin by considering the case where no multipath is present. In this case $\hat{H}(f) = H(f)$. The transfer function H(f) can be decomposed into two components, a symmetrical component

$$H_I(f) = 1/2[H(f) + H^*(-f)], \tag{4}$$

and an antisymmetrical component

$$H_Q(f) = 1/2[H(f) - H^*(f)], \tag{5}$$

where H^* denotes the complex conjugate of H. The global symmetry, about f=0, of $H_I(f)$ guarantees that the corresponding impulse response, $h_I(t)$, is purely real. Therefore, $H_I(f)$ represents the transfer function of the inphase channel. It is $H_I(f)$ that is typically chosen to approximately meet Nyquist's first criterion.

The global antisymmetry, about f=0, of $H_Q(f)$ guarantees that the corresponding impulse response, $h_Q(t)$, is purely imaginary. Therefore, $H_Q(f)$ represents the transfer function of the quadrature channel. In general $H_Q(f)$ will not satisfy Nyquist's first criterion and indeed for the ideal situation there is no reason why it should. However, consider the case where there is some phase offset θ, perhaps due to nonideal carrier recovery. In this case

$$\hat{H}(f) = c_1 H(f), \tag{6}$$

where $|c_1| = 1$ and $\arg(c_1) = \theta$. It is straightforward to show that the resulting inphase (i.e. that seen by the sampler) impulse response is

$$h_\theta(t) = (\cos\theta)h_I(t) - j(\sin\theta)h_Q(t), \tag{7}$$

where $h_I(t)$ and $h_Q(t)$ are the impulse responses corresponding to $H_I(f)$ and $H_Q(f)$, respectively. Note that $(\cos\theta)h_I(t)$ represents the desired term in equation (7), while $(\sin\theta)h_Q(t)$ represents the undesired term in equation (7). It is this undesired term that results in what is sometimes referred to as quadrature distortion in quasi-synchronous detectors.

We now return to the multipath problem. Equation (2) can be written in the form

$$\hat{H}(f) = \sum_{p=1}^{P} c_p H(f) e^{-j2\pi f T_p}. \tag{8}$$

Recalling that the exponential terms correspond to pure delays in the time domain and applying equations (6) and (7) yields the inphase impulse response given by

$$\begin{aligned} h_{MP}(t) = & \sum_{p=1}^{P} a_p h_I(t-T_p) \\ & -j \sum_{p=1}^{P} b_p h_Q(t-T_p), \end{aligned} \tag{9}$$

where

$$\begin{aligned} a_p &= |c_p| \cos\theta_p \\ b_p &= |c_p| \sin\theta_p. \end{aligned} \tag{10}$$

Note that the direct path may no longer have unity gain and zero phase because multipath propagation will affect both the phase of the carrier and the AGC. When multipath propagation is present with vestigal sideband transmission, the inphase channel does not experience pure multipath but rather a superposition of the multipath channel operating on both the inphase and quadrature channels. Thus the distortion due to multipath propagation can consist of two components, one from the "inphase channel mulitpath" and one from the "quadrature channel multipath".

While on the topic of multipath propagation it is worthwhile discussing the nature of the multipath propagation that is encountered with the braodcast Telidon application. Simulation and analysis that has been performed at MCS [1] [2], and off-air measurements in the Ottawa area performed by CRC [1] provide us with ample information about the type of off-air multipath channels that can be encountered in urban and surburban environments. One important point is that the multipath channel is a fairly strong function of frequency. Simulations indicate that for channel 2, most of the energy is within 2.5 μs [1] but the multipath spread tends to increase as the frequency increases [2]. The off-air measurements were performed using an RF carrier frequency of 201 MHz, which approximately corresponds to channel 11. These measured channels were catagorized as being either acceptable or unacceptable, from a television viewing standpoint, using the approach that is outlined in [1]. A total of 160 impulse responses were measured. Of these, 83 were deemed to represent acceptable channels. Most of the measurements were performed near the downtown core where multipath propagation is known to be a severe problem. However 15 of the measurements were taken well away from the downtown core (i.e. Carlingwood area). Interestingly, 10 of the 15 measurements taken in the Carlingwood area were rejected. It should be noted that the high rejection rate in this largely suburban area is probably partially due to the fact that the investigators were, in effect, looking for multipath channels, thereby biasing the results. For the acceptable channels, most of the echoes have delays less than 3 μs. The longest delay observed, for a significant echo (i.e. DU< 24 dB) was 8 μs.

For certain equalization strategies the number of significant echoes is important. For the 83 acceptable channels that were measured, the mean number of significant

echoes is 3.33 with a standard deviation of 1.97. Therefore, acceptable channels with 4 or 5 significant echoes are relatively common. The maximum number of significant echoes, for an acceptable channel is 10.

A question of interest for equalization is, are the vast majority of multipath channels encountered minimum phase? This question is closely related to the ratio of received energy from the "direct" path to that from the "reflected" paths. In general, if the signal energy received from the direct path is usually significantly greater than the signal energy received from the reflected paths, then the vast majority of the channels will be minimum phase. On the other hand, if the energy received from the reflected paths is usually comparable to or greater than that from the direct path, then it is likely that non-minimum phase channels will be encountered. One situation that will often result in a non-minimum phase channel is when the direct path is either blocked or partially blocked, perhaps by a large building. Of the 83 acceptable channels, only six were found to be non-minimum phase [1].

The multipath structure of a given channel may be time varying. For example, televisions with low gain antennas (e.g. "rabbit ears") can be susceptible to time-varying ghosts due to moving surfaces such as airplanes. Movement of the transmitting antenna in the wind has also been observed to cause time-varying multipath propagation.

Little information is available in the open literature about echoes in CATV, however CRC has provided MCS with a working paper* that contains some useful information. For

*The working paper entitled, "Ghosting on CATV Systems" was supplied to MCS by Dr. M. Sablatash of CRC. The author is not familiar with the original source of this document.

strong echoes, which are caused by effects in the same distribution line, the worst case delay is estimated to be approximately 400 ns. The worst case ghost, due to the cascading of several trunk amplifiers, is estimated to occur at about 6 μs delay with an attenuation of approximately -48 dB. Also, non-minimum phase multipath channels with CATV can occur because the signal propagation is slower in the cable than in free space. Therefore, the receiver can pick up an attenuated version of the signal prior to the reception of the stronger cable signal [3]. In any case, the multipath channels that can be encountered in an off-air urban scenario are likely to be more challenging than those for CATV. Thus it is the former case that is likely to be the ultimate test for adaptive equalization.

Before leaving the discussion of multipath propagation, it should be noted that the channel model, described by equation (1), assumes that the reflections are specular. This will only be a valid assumption when irregularities on the reflecting surface are small compared to the wavelength of the RF carrier*. The wavelength for channel 2 is about 5.56 m, while the wavelength for channel 13 is about 1.43 m. From these observations, the model given by equation (1) is probably valid for the lower VHF band (channels 2 to 6). Also, it may be valid for many multipath situations that can be encountered in the upper VHF band (channels 7 to 13). However, for channels in the UHF band where wavelengths are on the order of 0.5m, many of the reflections are likely to be at least partially diffuse in nature. In this case each reflected path can be represented by its own distinct transfer function.

*Other factors must also be satisfied in order for the above assumption to be valid.

The transmission impairments discussed so far have all been of the linear nature. It has been shown [4] that linear equalization can also be used to improve performance when nonlinear impairments are present, such as the AM to AM and AM to PM conversion that is typical of power amplifiers.

An adaptive equalizer is also capable of improving performance when coloured noise or interference is present. In essence, the adaptive equalizer improves performance by emphasizing the portions of the spectrum where the signal to noise ratio is good and deemphasizing portions of the spectrum where the signal to noise ratio is poor.

While on the topic of noise it is probably worthwhile to briefly discuss the nature of the noise in the VHF and UHF frequency bands. The interested reader is referred to [5] for a more detailed discussion. Figure 2 illustrates typical average power spectral densities for various types of noise as a function of frequency. Note that in urban environments, where multipath propagation is most likely to occur, man-made noise is the dominant type of noise in both the VHF and UHF frequency bands. Also, man-made noise is typically dominated by impulse noise. The fact that the noise may often be dominated by impulse noise has some important implications to equalization, that will be discussed later in this appendix.

There are two different locations in the complex baseband model where linear equalization could be applied. These two possibilities are illustrated in Figure 3. It is well known that in general the optimal linear equalizer is complex (see Figure 3b). In order to perform complex equalization, the equalization must either be performed at IF or quadrature demodulation must be performed and then the equalized signal can be obtained by convolving the complex demodulated signal and the equalizer's complex

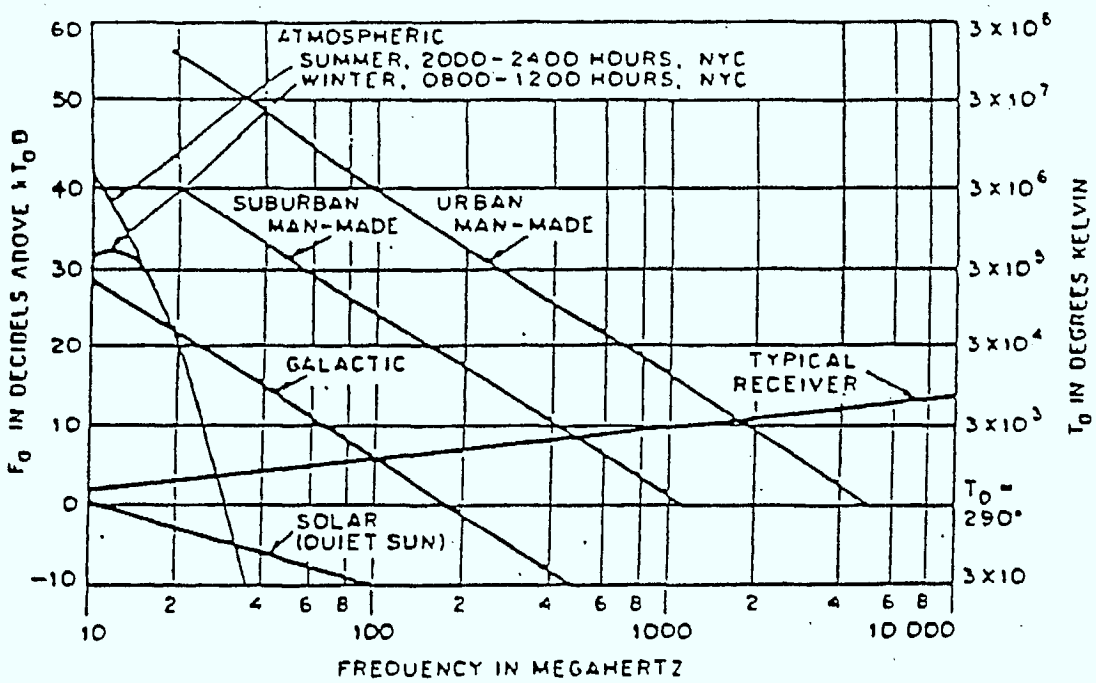


Figure 2 : The median value of average noise power versus frequency for a receiver with an omni-directional antenna located near the earth's surface [6].

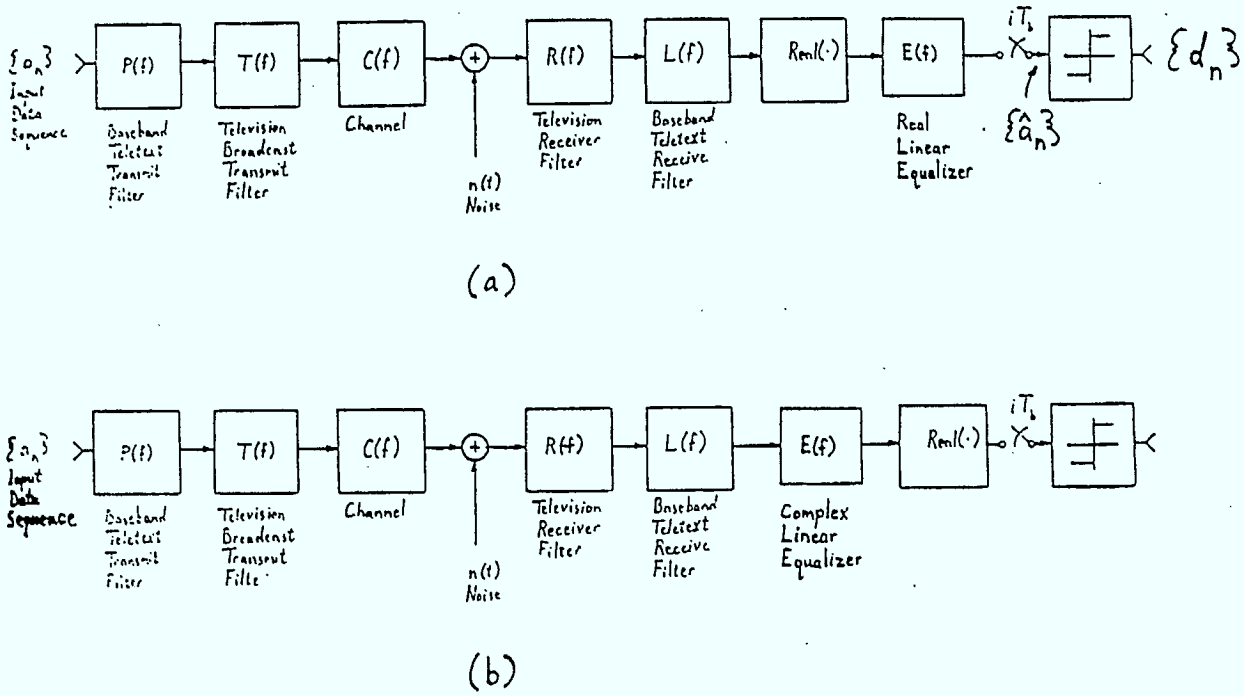


Figure 3: Two possible locations for linear equalization in the complex baseband model.

impulse response. Clearly, complex adaptive equalization represents a degree of complexity that is probably not economically justifiable at this time. A less complex alternative is to perform real linear equalization following video detection (see Figure 3a). This is the approach that has usually been followed in the literature. For typical multipath channels it is expected that the loss incurred, due to ignoring the signal energy in the quadrature channel will be less than a couple of dB. Furthermore, it is expected that much of the future broadcast Telidon receive equipment will only have access to the detected signal (i.e. the real video signal) and will therefore be constrained to use real equalization. Since the goal here is to propose an economically practical adaptive equalizer, we will restrict our attention to real adaptive equalizers. However, it should be remembered that performance closer to optimal could be achieved if complex equalization were used. It should be noted that we are not restricting our attention to linear equalization. Decision feedback taps can be added to both the strategy illustrated in Figure 3a and the one illustrated in Figure 3b. Also, a purely decision feedback equalizer can be considered. The relative merits of the various equalization strategies will be discussed in some of the subsequent sections.

The structure of this appendix is as follows. Section 2.0 outlines some fundamentals of equalization. Linear equalization and decision-feedback equalization are discussed and compared. In Section 3.0 some implementational considerations for teletext equalization are discussed. A brief overview of some hardware building blocks, for constructing an equalizer is given. Also, a structure for implementing the linear equalizer is proposed. This appendix concludes in Section 4.0.

2.0 SOME FUNDAMENTALS OF EQUALIZATION

Over the past decade, the equalization of data communications channels has been a very active area for research and development and hence many excellent papers can be found in the open literature. Chapter 6 of [7] represents a fairly current and comprehensive overview of the theory of equalization, and the interested reader is referred to this reference. A detailed look at the fundamentals of equalization is well beyond the scope of this study. Rather, the purpose of this section is to discuss a few fundamental points that will be useful later in this memo. For the most part, this discussion is at the conceptual level, rather than delving into the details of the mathematics.

Here we begin by looking at the general problem. The input is modelled by a sequence of weighted unit samples, $\{a_n\}$, as is illustrated in Figure 3a. These samples are "smeared" by the transmitter's filters, the channel, and the receiver's filters, which results in undesirable intersymbol interference. In the absence of noise, the goal of the equalizer is to operate on the received signal in such a way that when the output of the equalizer is sampled at the baud rate, the resulting sequence of samples is the same as the input sequence. When noise is present, an optimal equalizer will attempt to trade off the effects of noise and intersymbol interference in an optimal manner. For the purposes of discussion consider the discrete-time model shown in Figure 4. In this diagram the tap spacing of the equalizer is equal to the reciprocal of N times the baud rate, and $k = [m/N]_I$ denotes that $k = m/N$ when m/N is an integer. Note that the equalizer shown in Figure 4 is a linear transversal equalizer. Here, we will begin by considering linear equalization. Other equalization structures are discussed subsequently. Also, techniques

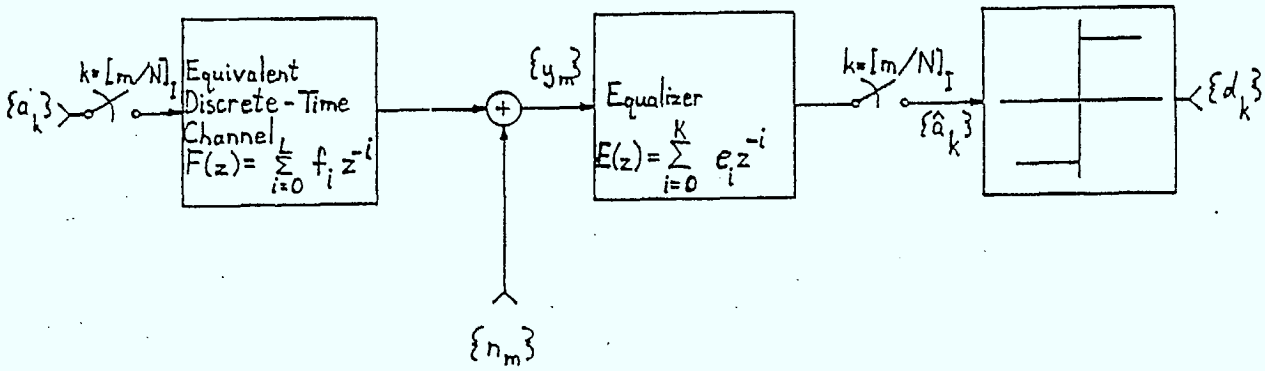


Figure 4: A discrete-time model for linear equalization.

for choosing the equalizer coefficients are discussed later in this section.

2.1 Linear Equalization

For the purposes of discussion, we begin by considering inverse filtering (i.e. zero-forcing equalization). This problem can be divided into two cases of interest. Case 1 is the equalization of minimum phase roots of F(z)*. Case 2 is the equalization of non-minimum phase roots of F(z). The effects of the noise $\{n_M\}$ will be neglected at first and then considered later.

For a known minimum phase channel, the inverse filter can easily be determined. This is best illustrated by a simple example. Consider the minimum phase channel given by

$$F(z) = 1 + 0.1z^{-1} \tag{11}$$

The transfer function of the inverse filter is given by

$$E_1(z) = 1/(1 + 0.1z^{-1}). \tag{12}$$

Note that this transfer function represents a linear recursive filter. If it is desirable to use a linear transversal equalizer then $E_1(z)$ can be closely approximated by

$$E_2(z) = 1 - 0.1z^{-1} + 0.01z^{-2}, \tag{13}$$

*A minimum phase root of F(z) is defined to be a root that lies within the unit circle of the z-plane [8].

which is obtained by truncating the polynomial that results from dividing the denominator of $E_1(z)$ into the numerator. Since $E_2(z)$ is not the true inverse filter for $F(z)$, some residual intersymbol interference will exist. In this example, the transfer function of the equalized channel is

$$F(z)E_2(z) = 1 + 0.001z^{-3}. \tag{14}$$

For the noiseless non-minimum phase channel, it is not possible to simply inverse filter because the inverse filter is unstable. The ideal equalizer can be visualized as a two part process. The first part is to whiten the channel. The second part is to matched filter the whitened channel in order to obtain the equalized channel. This procedure is closely related to inverse filtering because the unit sample response of any "white" channel in cascade with its matched filter always consists of a single weighted unit sample. To do this exactly would require an infinite delay, but for "low Q" roots inverse filtering can be approximately achieved by accepting a finite delay. Again, a simple example is useful for illustrating the concept. Consider the non-minimum phase channel with the transfer function

$$F(z) = 0.01 + z^{-1} \tag{15}$$

The transfer function of the inverse filter is given by

$$E_0(z) = 1/(0.01 + z^{-1}). \tag{16}$$

However, $E_0(z)$ is the transfer function of a filter that is not stable. Therefore, a different approach must be used to equalize the channel. The first step is to whiten the channel. A stable whitening filter can be obtained by "reflecting" the pole of $E_0(z)$ into the unit circle. The transfer function of the resulting whitening filter is

$$E_1(z) = 1/(1 + 0.01z^{-1}). \tag{17}$$

For notational convenience we replace this recursive whitening filter with a transversal filter that approximately whitens the channel. Here, the transfer function of the approximate whitening filter is given by

$$E_2(z) = 1 - 0.01z^{-1}, \tag{18}$$

and the resulting whitened channel is given by

$$F(z)E_2(z) = 0.01 + 0.9999z^{-1} - 0.01z^{-2} \tag{19}$$

The matched filter for the whitened channel is described by the transfer function

$$E_3(z) = -0.01 + 0.9999z^{-1} + 0.01z^{-2}. \tag{20}$$

The resulting equalizer is the cascade of the whitening filter and the matched filter. Thus its transfer function is given by

$$E(z) = E_2(z)E_3(z), \tag{21}$$

and the transfer function of the equalized channel is

$$F(z)E(z) = -0.0001 + z^{-2} + 0.0001z^{-4}. \tag{22}$$

Note that in this example the equalized channel introduces a delay of two sample periods, whereas in the minimum phase case the equalized channel did not introduce any delay. At this point it should be pointed out that a real communications channel is said to be minimum phase if it can be accurately modelled by a pure delay (i.e. the propagation delay) in cascade with a minimum phase linear system (e.g. a filter). The transfer function of this pure

delay is not a strict minimum phase function*. In this case, applying the inverse filter of the minimum phase linear system results in a equalized channel consisting of only the pure propagation delay.

It is important to notice that both minimum phase and non-minimum phase channels (or in fact channel roots) can be effectively equalized using linear equalization. The problem with linear equalization occurs when the channel has "high Q" roots (i.e. the roots are close to the unit circle). These "high Q" roots pose two difficult problems. One problem is that very long equalizers (i.e. many taps) are required to equalize these roots. The other is that the equalization of these roots can result in an enormous amplification of the noise. The problem of noise amplification for inverse filtering is illustrated in Figure 5.

In practice, inverse filtering (or zero-forcing equalization) is often not the best choice because it concentrates entirely on inverting distortions caused by the channel (i.e. intersymbol interference) while it completely ignores the effects of the noise. Ideally one would like to choose the equalizer that balances between the effects of intersymbol interference and noise, in such a way that the bit error rate (BER) is minimized. Unfortunately, adaptive equalization algorithms based on BER tend to be too complex to be practically implemented. A commonly used approach, that yields good but not optimal BER performance and that has been economically implemented for various applications, is to choose the coefficients (e.g. tap weights) of the equalizer such that the mean square error, between the actual equalizer output and the

---

*Some definitions of discrete time minimum phase functions allow for simple poles and zeros on the unit circle.

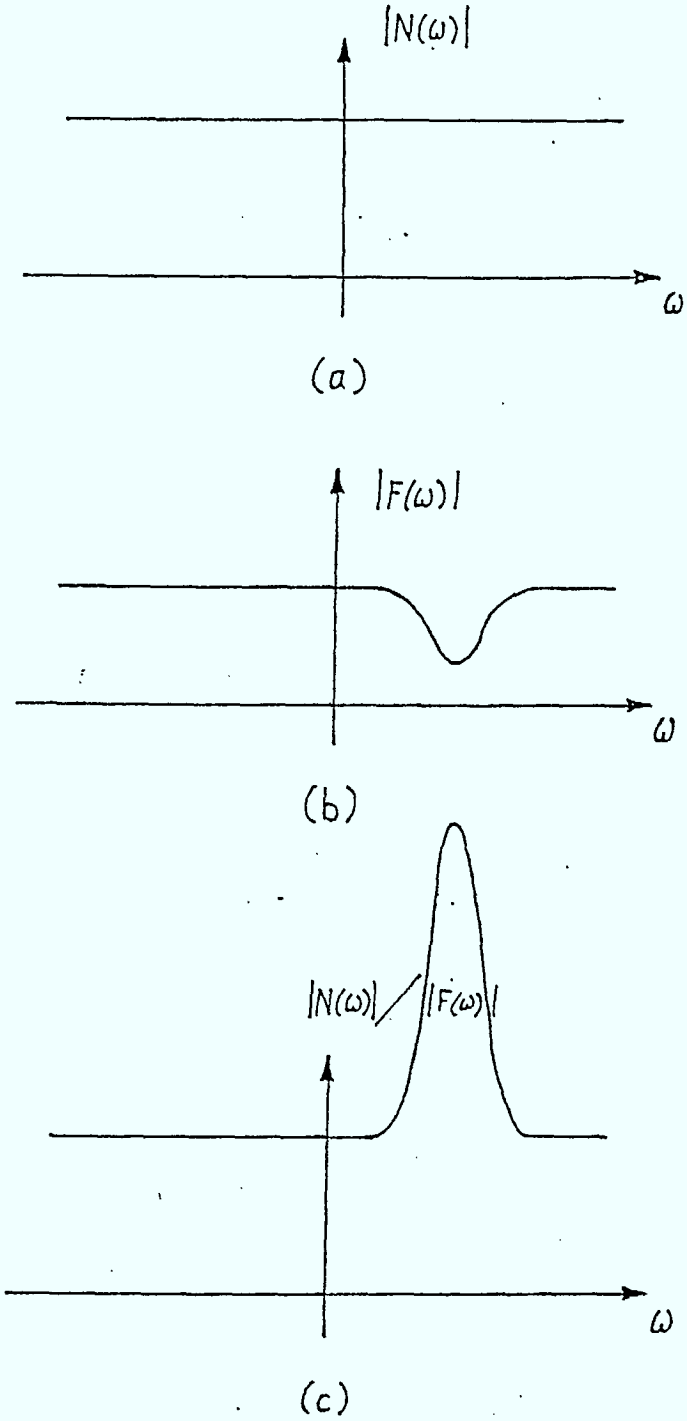


Figure 5: a) The noise spectrum.
b) The channel spectrum.
c) The noise spectrum after inverse filtering.

desired equalizer output, is minimized. Typically the desired equalizer output is either a known training signal or the output of the data detector. It should be emphasized that although this type of equalizer does attempt to achieve an optimum balance (in the mean square sense) between intersymbol interference it still exhibits relatively poor performance for channels with "high Q" roots (i.e. channels with nulls). This is in fact an inherent limitation of any linear equalization scheme. An excellent illustration of this limitation is given by some example channels in [7]. The performance of linear equalization for three example channels is shown in Figure 6*. The amplitude responses for channels 1, 2, and 3 are shown in Figures 7, 8, and 9, respectively. Notice that linear equalization performs quite well for channel 1, which has no in-band nulls. On the other hand, linear equalization performs very poorly on channels 2 and 3, both of which have in-band nulls. In general, if the channel has nulls or "near nulls" then nonlinear techniques (e.g. decision-feedback equalization or maximum likelihood sequence estimation) must be used in order to achieve good BER performance.

A discrete-time linear equalizer can be configured either as a transversal (i.e. FIR) filter, which has only feedforward taps, or as a recursive (i.e. IIR) filter, which has both feedforward and feedback taps. Clearly, the output of a transversal filter is a weighted sum of past (including the present) inputs, while the output of the recursive filter is a weighted sum of past inputs and outputs. In practice, adaptive equalization is almost never done using a linear recursive filter. The primary reason for this is that direct-form recursive filters are very susceptable to limit-cycle behaviour when any

*In [7], the curves for channels 1 and 3 are incorrectly labelled. The correct labelling can be seen in [9].

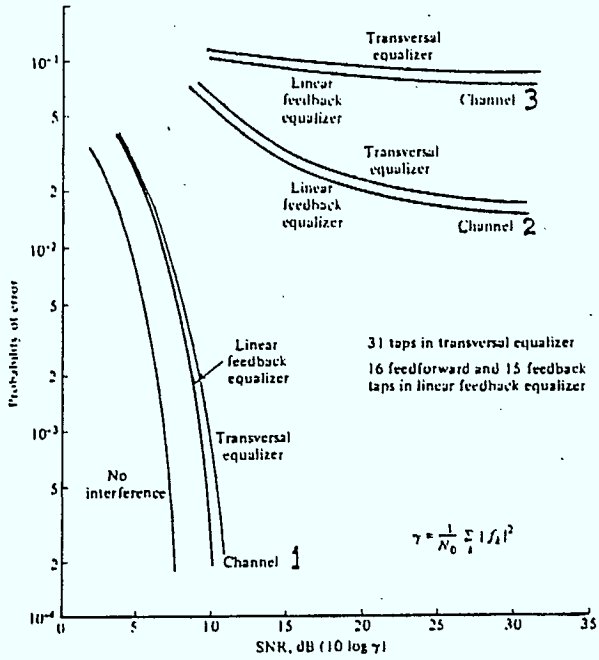


Figure 6: The performance of linear equalization for several example channels[7].

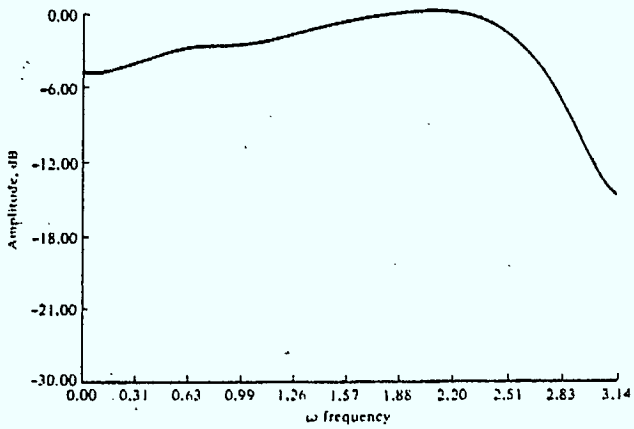


Figure 7: The amplitude response of Channel 1 [7].

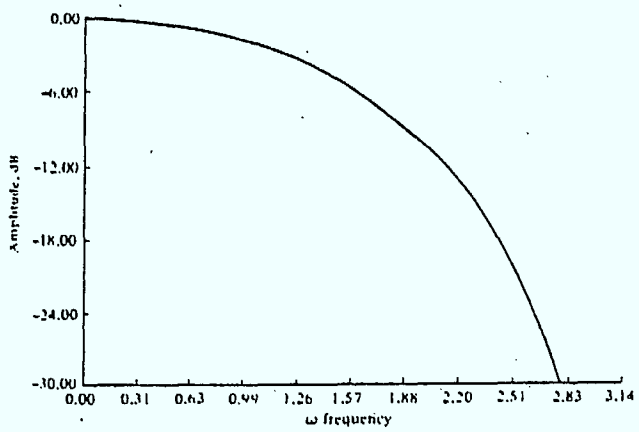


Figure 8: The amplitude response of Channel 2 [7].

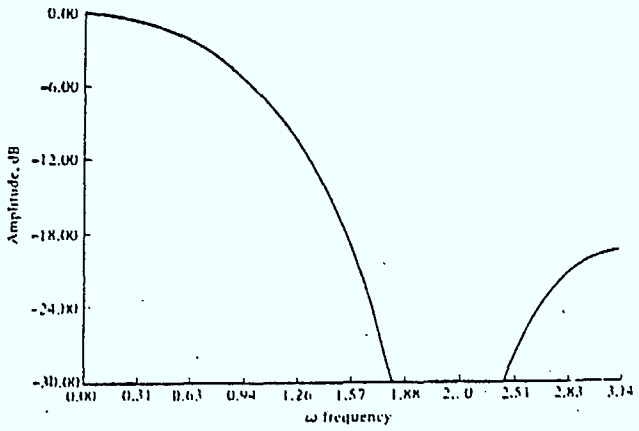


Figure 9: The amplitude response of Channel 3 [7].

nonlinearities (e.g. finite word length effects for a digital implementation) are present in the implementation and when the number of feedback taps is greater than two [10]. Recursive filter structures do exist that are not as susceptable to limit-cycle behaviour, but these structures tend to greatly complicate both the hardware realization and the coefficient adaptation algorithm. Futhermore, there is rarely any motivation to accept this increase complexity because for most channels even ideal recursive filters (i.e. ones that do not exhibit any limit-cycle behaviour) do not yield significantly better performance than a transversal filter (see Figure 6). For these reasons, for the remainder of this study, consideration of linear equalization will be restricted to transversal filters.

If a discrete-time transversal filter is to be used, it is necessary to choose an appropriate tap-spacing (i.e. the duration of the time delay that is applied between consecutive taps). The impulse response of the adaptive transversal equalizer is given by

$$E(z) = \sum_{i=0}^{K} e_i z^{-i} \tag{23}$$

Here, the e_i are the parameters of the equalizer. Note that the sequence $\{e_i\}$ is the unit sample response of the equalizer. Thus $\{e_i\}$ is a discrete-time unit sample response, with a sampling rate that is equal to the reciprocal of the tap spacing. Also, the transfer function of the equalizer can be bandlimited to the NTSC video band, since there is no point in equalizing frequency bands where no signal energy is present. Thus the minimum sampling rate for the equalizer can be determined by applying

Nyquist's sampling theorem. The resulting minimum sampling rate is approximately 8.4 MHz. For implementational reasons, it is often desirable to choose the sampling rate to be a multiple of the baud rate. The lowest multiple of the baud rate that exceeds the minimum sampling rate is 11.454 MHz (i.e. twice the baud rate). This corresponds to a tap spacing of 87.3 ns. Note that, from Nyquist's sampling theorem, absolutely nothing is gained by going to smaller tap spacings (except added complexity). This point is valid regardless of the nature of the multipath channel. The number and the density of the echos is completely irrelevant since the bandlimited channel cannot resolve paths with a differential delay of less than 87.3 ns anyway.

The required number of taps is also a point of interest. Once the sampling rate has been determined, the choice of the number of taps in effect specifies the support* of the unit sample response of the equalizer. Obviously, the support of the equalizer must be at least as large as the spread of the channel if the equalizer is to be effective. Preferrably it should be several times greater. This is clearly illustrated for the special case of echo equalization in [3]. Recall that for measurements taken in the Ottawa area most of the significant echoes have delays less than 3 μs, and longest delay was 8 μs. Thus for multipath equalization near downtown Ottawa an equalizer with a support of at least 3 μs (i.e. at least 35 taps) should be used. Even longer equalizers may be required for larger cities or at higher frequencies (e.g. UHF band).

*By "support" we are referring to the shortest segment of time, beyond which the unit sample response is effectively zero.

It has been brought to the authors attention* that RCA has performed some off-air tests using the Philips teletext equalizer [11]. Apparently, for many of these tests the Philips equalizer actually degraded the teletext performance. Note however that the Philips equalizer was designed to be "capable of reducing short echoes (0.5-1 μs) in Teletext signals" [11]. Thus the support of this equalizer is only on the order of 1 μs. Thus it should not be surprising that it is not capable of equalizing many of the channels in metropolitan New York where many echoes are expected to have delays several times greater than the short echoes for which it was designed. Although the Philips equalizer's performance was dismal for the off-air tests in New York city, it would probably perform quite well for most CATV channels where the dominant echoes are expected to have delays less than 400 ns. It should be emphasized that the reason that the Philips equalizer failed in these off-air tests is that its support is simply not large enough to handle severe multipath channels in an urban environment. It is <u>not</u> due to the density of the paths or the possible non-specular nature of the reflections, as some people have suggested. The author does not know whether the analog techniques used for the Philips equalizer could be used for an equalizer with enough taps (e.g. 50) to perform well in New York.

Another important point about the use of linear equalization in urban environments should be mentioned. The most severe multipath channels are likely to be encountered in urban environments. It is also in urban environments where impulse noise is likely to be the dominant type of noise. Without linear equalization, a strong impulse would typically cause a short burst of

*Personal communication from Dr. M. Sablatash of the Communications Research Centre, Canada.

errors, occurring in the vicinity of the impulse. In fact, if the effects of the receive filter are ignored, the impulse would only result in an error during the bit intervals in which it occurred. However, the impulse will be "smeared" by the receive filters by an amount that is dependent upon the impulse responses of those filters. If linear equalization is employed, where the equalizer has a support of at least 3 μs, then the energy of the impulse would be dispersed by the equalizer over a time duration of at least 3 μs. This dispersion of the impulse's energy may be either beneficial or detrimental depending upon the strength of the impulse. If the impulses energy is relatively weak in comparison to the energy per bit of the data signal, then the dispersion of the impulse's energy is likely to improve the BER. On the other hand if the impulse's energy is orders of magnitude larger than the energy per bit of the data signal, then the dispersion of the impulse's energy might result in significantly more errors than if no equalization were performed! Of course for this to be true, the eye would have to be open when no equalizer is present. However, it is important to note that for certain impulse noise environments, the presence of a linear equalizer can actually degrade the BER performance, even though the intersymbol interference is substantially reduced.

2.1.1 Coefficient Adaptation for Discrete-Time Linear Equalizers

Here, we will briefly discuss some approaches for the adaptation of the coefficients of a discrete-time linear equalizer for teletext reception. It is felt that at this time a fully adaptive real-time adaptive equalizer, with the number of taps required for off-air reception in an urban environment, is not practical. Thus the concept being considered is an equalizer that can slowly track the

channel and that can adapt to a new channel in a reasonably short length of time (i.e. it can adjust quickly enough that the viewer will not be inconvenienced when he changes the channel or turns on his television set). One approach [3] to realizing such an equalizer, is to subdivide the tasks into two areas; (i) filtering, and (ii) coefficient adaptation and control. The idea is to have a programmable filter (e.g. digital or CCD) operating in tandem with a processor (e.g. a microprocessor). Clearly, this type of equalizer would not be capable of tracking time-varying multipath such as that due to the transmit antenna moving in the wind or due to carrier recovery phase jitter, but it should be quite adequate to handle most multipath channels.

Due to its tractability, we will focus our attention on coefficient adaptation based on minimizing the mean squared error criterion. A recommendation of a specific algorithm is beyond the scope of this study since such a recommendation should be based upon careful analysis of computational requirements and numerical stability, particularly taking into account the finite wordlength (and most likely fixed point) nature of the processing. The mathematical development closely follows that found in [9].

Using the notation known in Figure 4, the mean square error criterion is giveng by‡

‡Here, we have carried the notation through in complex form for generality.

$$J(\underline{e}) = E\{|a_{k-P}-\hat{a}_k|^2\} \tag{24}$$

$$= E\{|a_{k-P}-\sum_{j=0}^{K} e_j y_{k-j}|^2\}$$

$$= E\{|a_{k-P}|^2\}-2\text{Real}[\sum_{j=0}^{K} e_j E\{a^*_{k-P} y_{k-j}\}]$$

$$+ \sum_{j=0}^{K}\sum_{i=0}^{K} e_j e_i^* E\{y_{k-j} y^*_{k-i}\},$$

where $\underline{e}$ is the vector of equalizer coefficients, h* denotes the complex conjugate of h, and P is an integer that accounts for a fixed decision delay. Equation (24) is a quadratic functional in the coefficients of the equalizer. The optimum solution for a quadratic functional is well known and is given by

$$\underline{e}_{opt} = A^{-1}\ \underline{b}, \tag{25}$$

where A is a (K+1)x(K+1) Hermitian matrix with ijth element

$$a_{ij} = E\{y^*_{k-i} y_{k-j}\}, \tag{26}$$

and $\underline{b}$ is a (K+1) vector with (i+1)th element

$$b_{i+1} = E\{a_{k-P} y^*_{k-i}\},\ i=0,\ldots,\ K. \tag{27}$$

Here, it is assumed that A is full-ranked and therefore invertible. If the input signal is assumed to be an uncorrelated binary data signal and the noise is assumed to be additive white Gaussian noise then

$$a_{ij} = \begin{cases} R_{i-j} + N_o \, \delta_{ij}, & |i-j| \leq L \\ 0 \quad , & \text{otherwise,} \end{cases} \tag{28}$$

and

$$b_{i+1} = \begin{cases} f^*_{P-i}, & 0 \leq P-i \leq L \\ 0 \quad , & \text{otherwise} \end{cases} \tag{29}$$

Here, N_o is the variance of the noise samples, and $\{R_i\}$ is the sampled autocorrelation autocorrelation function of the channel, which is given by

$$R(z) = F(z)F^*(z^{-1})$$

$$= \sum_{i=-L}^{L} R_i z^{-i} \tag{30}$$

The problem with employing equation (25) directly is that A and $\underline{b}$ must be estimated and then the covariance matrix A must be inverted. In order to avoid these computationally intensive problems, people have turned to steepest descent based methods for the minimization of $J(\underline{e})$.

Perhaps the most commonly used approach is one that was introduced by Widrow and Hoff, which iteratively adjusts the coefficients using a noisy but unbiased estimate of the gradient vector. The structure of this classical type of adaptive equalizer is shown in Figure 10. One modification that can be made to the structure shown in Figure 10, that might improve its performance in an impulse noise environment, is to monitor the size of the error signal ε_k.

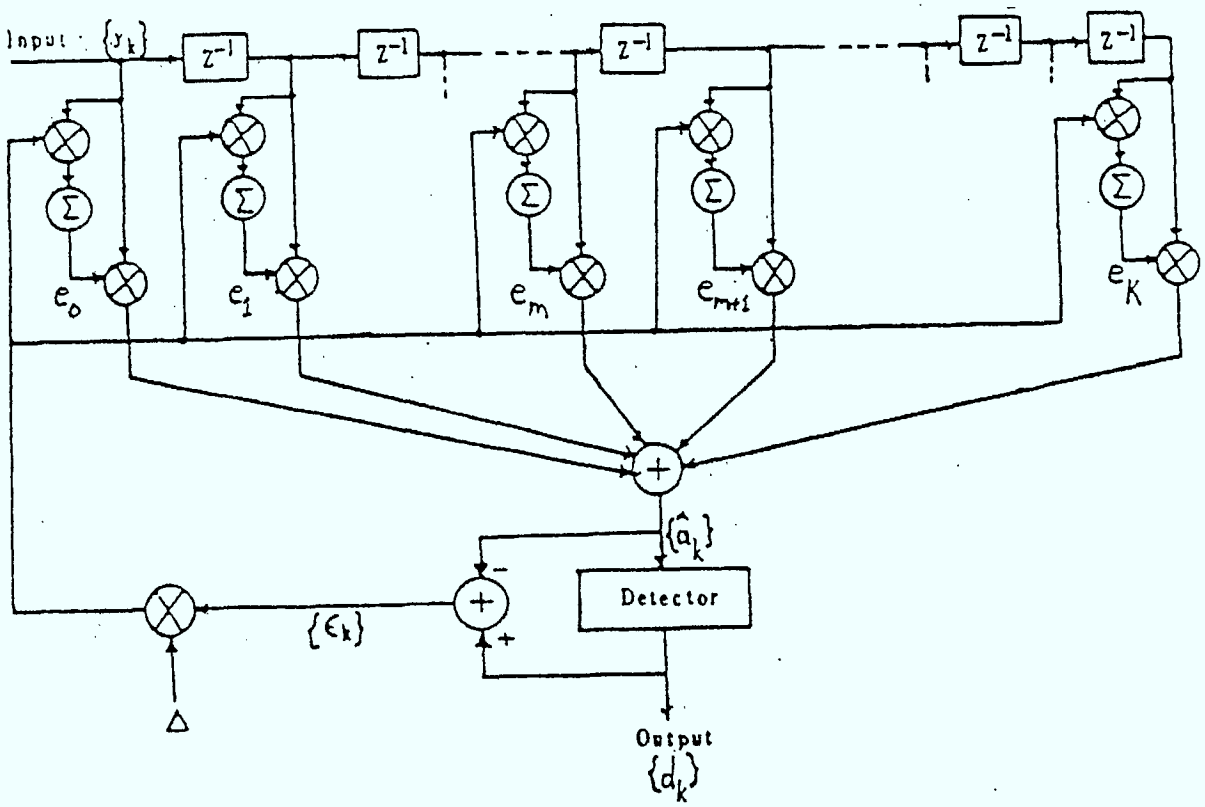


Figure 10: A block diagram illustrating the structure of a typical gradient-based adaptive equalizer [9].

If the magnitude of a given ε_k exceeds some threshold value, then it would be concluded that the signal is being corrupted by a strong impulse and that error signal should not be used for the adjustment of the tap weights. Ideally, this threshold level should be adaptive.

Another modification that might improve the performance in strong impulse noise environment is to precede the equalizer with a limiter that would reduce the magnitude of strong impulses but leave the remainder of the received signal unaltered. Note that this function will probably be at least partially performed by existing hardware (i.e. the demodulator) because of the limited dynamic range of actual devices.

Any steepest descent based algorithm can suffer poor tracking behaviour due to ill-conditioning. To get a feel for the tracking behaviour of this type of equalizer it is instructive to consider the convergence behaviour of the method of steepest descent for the quadratic problem. In the following excerpt from [12], A and α are the maximum and minimum eigenvalues of the covariance matrix (Q), respectively.

"The method of steepest descent converges linearly with a ratio no greater than $[(A-\alpha)/(A+\alpha)]^2$. It also has been shown, by Akaike, that barring certain degenerate starting points, this bound on the rate of convergence is exact. The proof of this fact is fairly complex and we do not give it here; but because of that fact, we say that the convergence ratio of steepest descent is $[(A-\alpha)/(A+\alpha)]^2$.

It should be noted that the convergence rate actually depends only on the ratio $r=A/\alpha$ of the largest to the smallest eigenvalue. Thus the convergence ratio is

$$\left(\frac{A-\alpha}{A+\alpha}\right)^2 = \left(\frac{r-1}{r+1}\right)^2 \tag{31}$$

which clearly shows that convergence is slowed as r increases. The ratio r, which is the single number associated with the matrix Q that characterizes convergence, is often called the condition number of the matrix. Note, however, that even if n-1 of the n eigenvalues are equal and the remaining one is a great distance from these, convergence will be slow, and hence a single abnormal eigenvalue can destroy the effectiveness of steepest descent."

From the above discussion it is apparent that the tracking performance of an adaptive transversal equalizer depends upon the conditioning of the covariance matrix with the elements given in equation (28). The conditioning of this matrix depends upon the correlation of the additive noise, and the response of the channel. Therefore, an ill-conditioned channel (i.e. one with inband nulls or "near nulls") can result in very poor tracking for a gradient-based adaptive transversal equalizer. Note that for these channels not only is the tracking performance poor, but even after the equalizer has converged the BER performance is likely to be highly suboptimal due to the inherent limitation of linear equalization that was discussed earlier in this section.

In order to avoid the above convergence problem, "self-orthogonalizing" adaptive equalization algorithms were developed (e.g. [13]). One type of self-orthogonalizing algorithm that exhibits nice finite word length properties is the lattice algorithms. A gradient-based self-orthogonalizing adaptive lattice equalizer was proposed in [14]. However, better tracking and convergence can be achieved by the exact solution of the least squares problem. This can be accomplished with some increase in

the computational requirements. These exact least squares techniques include fast Kalman equalization [15] and least squares lattice equalization [16]. All of the above techniques are thoroughly discussed in [7].

It is felt that the added complexity (and hence cost) of the self-orthogonalizing techniques is not warranted for the teletext application. Several reasons can be given for this. The self-orthogonalizing techniques exhibit significantly superior tracking performance for ill-conditioned channels. However an ill-conditioned channel would never be used for television reception because an inband null would result in an unacceptable degradation to the video signal. Furthermore, even after convergence, a linear equalizer probably would not improve the BER to an acceptable level for such channels (e.g. see Figure 6). Therefore, an iterative gradient-based adaptive algorithm, similar in nature to the one illustrated in Figure 10, is a reasonable choice. This type of algorithm has been successfully realized for many applications and its behaviour is fairly robust and well understood (e.g. [17]). Some modification of the algorithm may be desirable to take into account the non-real-time nature of the proposed coefficient adaptation approach. However, it should be noted that the iterative closed-loop nature of the adaptive algorithm must be maintained if inherently nonlinear techniques are used to realize the equalizer's programmable filter (e.g. CCD [18] or analog electronics [11]). In these cases an open-loop approach such as the direct solution of equation (25) is unlikely to perform well because the relationship between the tap control signal and the effective tap weight is nonlinear. This is illustrated for a CCD transversal filter [18] in Figure 11. An iterative closed-loop type of adaptation technique can compensate for this kind of monotonic nonlinearity without

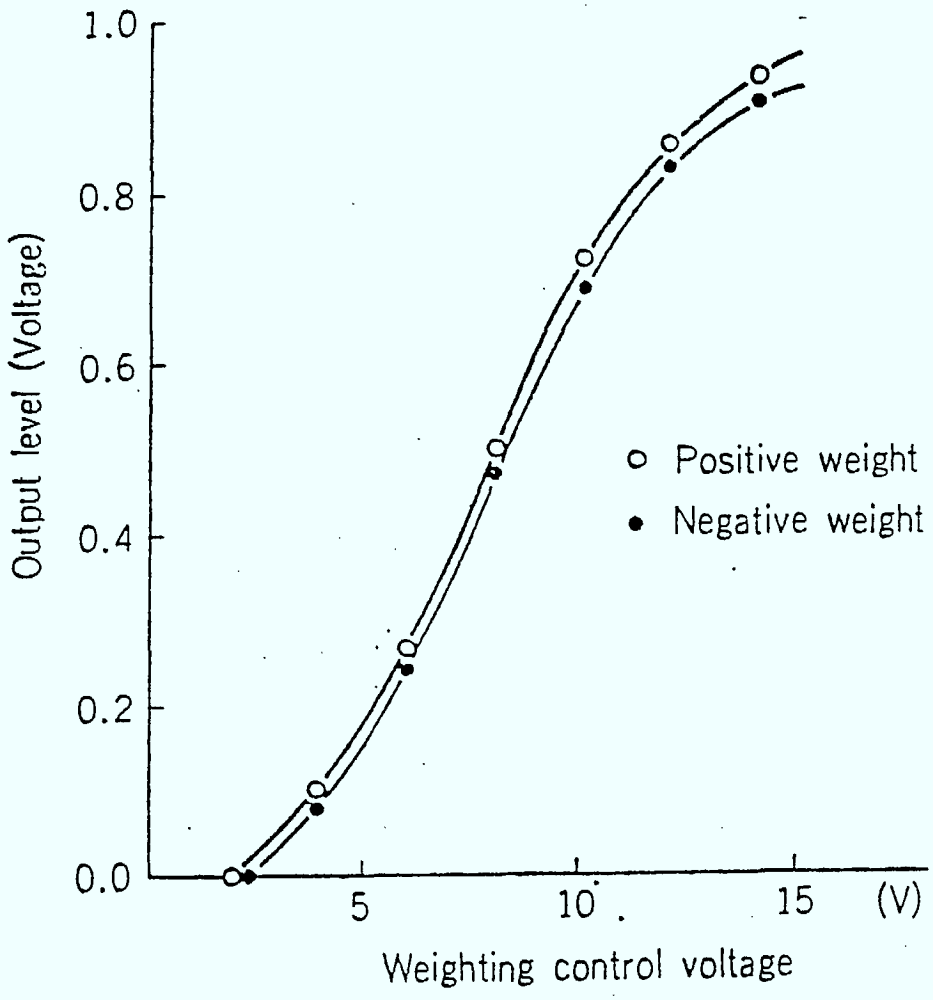


Figure 11: An illustration of the nonlinear control characteristic of a CCD transversal filter tap weight [18].

unduly complicating the algorithm. Typically, the only modification that is required is a modest reduction in the step size (i.e. Δ in Figure 10).

The choice of a possible training signal, for the adaptation of the equalizer, has been discussed in several papers [3] [19] [20]. Recommendation of the training signal is beyond the scope of this study. However, there are a couple of points on this issue that the author would like to mention. Several possible training signals have been suggested in the literature including a truncated (sin x)/x pulse, a 2T raised cosine pulse, and the initial transitions in the vertical serations after the equalizing pulses. Although any one of these possibilities may have some practical advantages, a training signal that looks like an uncorrelated data signal is the best choice from a couple of theoretical points of view. Clearly, the data self itself is such a signal and can be used for the adaptation of the equalizer coefficients if the eye is open. A known training sequence can be transmitted periodically to guarranty proper adaptation even when the eye is closed. Recall that the optimal (in the MSE sense) tap weights are given by equation (25). Note that A and $\underline{b}$ in equation (25) are dependent upon the transmitted signal. Thus the optimal tap weights are also dependent upon the transmitted signal. The clear conclusion is that if the optimal tap weights for the reception of "random" data are desired then the equalizer should be trained using a transmitted signal that closely resembles that for data. The desirability to use a data-like training signal becomes even more evident when one considers the use of the equalizer for compensating for nonlinearities in the channel (e.g. thre transmitter's power amplifier and the receiver's detector). By using a linear equalizer to

compensate for the nonlinear channel, we are implicitly fitting a linear model to the nonlinear channel. It is well known that the linear model, that is the best fit (according to some given criterion) to a nonlinear system, is highly dependent upon the stimulus (i.e. input) that the system is given [21]. Thus the teletext data signals may be the most appropriate channel stimulus, as well as being one of the most convenient.

2.2 Decision - Feedback Equalization

The major equalization alternative to linear equalization is decision-feedback equalization. Typically, a decision-feedback equalizer consists of a linear transversal filter in cascade with a pure decision-feedback section. A block diagram illustrating the structure of a typical gradient-based decision-feedback adaptive equalizer can be seen in Figure 12. In this particular illustration, the tap spacing for the linear transversal filter and for the pure decision-feedback section are equal. For most applications, including teletext, a more desirable arrangement is to have T/2 tap spacing for the transversal filter and T tap spacing for the pure decision-feedback section, where T is the baud period. From Figure 12, it can be seen that the output of decision-feedback equalizer (i.e. just prior to the detector) is a weighted sum of past channel outputs and past detected data symbols. A conceptual explanation of the functioning of a decision-feedback equalizer is as follows. The desired sampled impulse response of the overall channel is a unit sample* (i.e. no intersymbol interference). However when multipath propagation is present, the overall channel is smeared over several samples which results in intersymbol interference.

---

*For synchronous sampling at the bit rate.

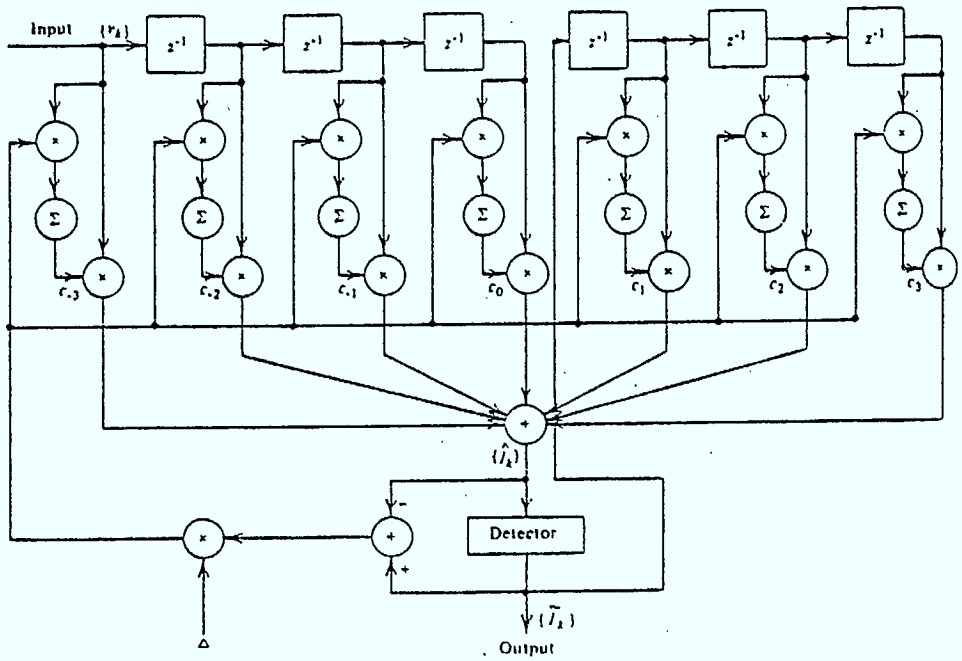


Figure 12: A block diagram that illustrates the structure of a decision-feedback equalizer. The equalizer illustrated here has tap weights that are adapted using an iterative gradient-based approach [7].

A possible "smeared" sampled impulse response can be seen in Figure 13. The purpose of the transversal filter is to equalize the precursor, in such a way that the main sample is the first significant sample. The pure decision-feedback section then cancels the resulting tail. Note that the tail is altered by the transversal filter. Although this functional explanation is over simplified it is of conceptual value. Although the structure of a decision-feedback equalizer is quite similar to that of a recursive linear equalizer, the former does not suffer from the same sensitivity to limit cycle behaviour as the latter. The reason for this is that the presence of the detector in the feedback loop has a stabilizing effect.

One of the advantages of decision-feedback equalization over linear equalization, is the ability of the former to provide reasonably good performance for channels with inband nulls. Recall that for example channel 2 and 3 of Figure 6 linear equalization did not result in very good performance even for relatively high signal to noise ratios. The performance of an adaptive decision-feedback equalizer, for these two example channels, can be seen in Figure 14. The performance is clearly vastly superior to that for linear equalization on these two channels. However, it should be mentioned that too much emphasis should not be placed on these two channels since any channel with inband nulls would certainly be unacceptable for television viewing.

For most of the channel impulse responses that are shown in Volume 2 of [1], there is no precursor. That is that for most of the channels the direct path comes through relatively undistorted, and the tail consists of weaker reflected paths. For these channels the transversal filter

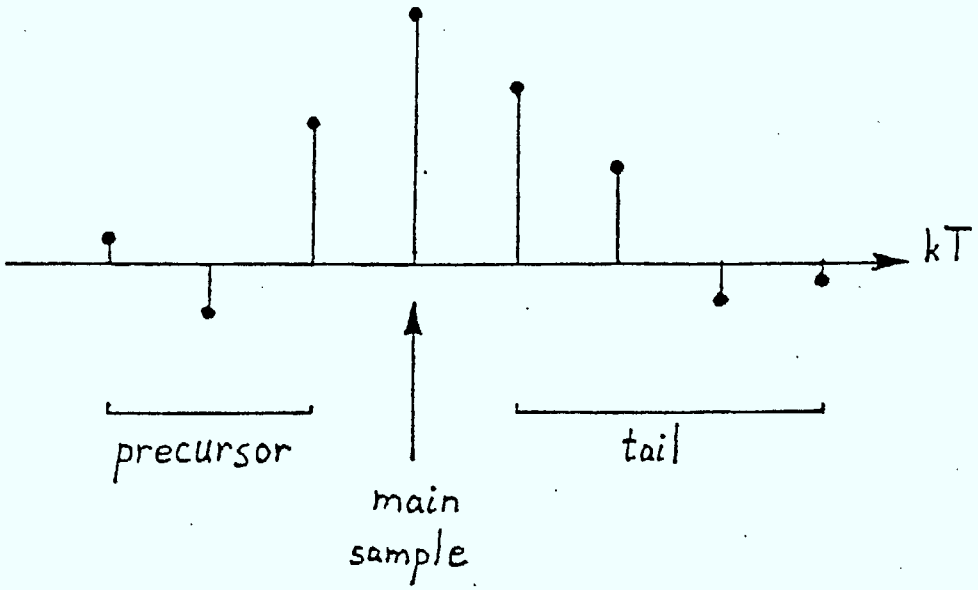


Figure 13: An example of a sampled impulse response that has been smeared by a channel.

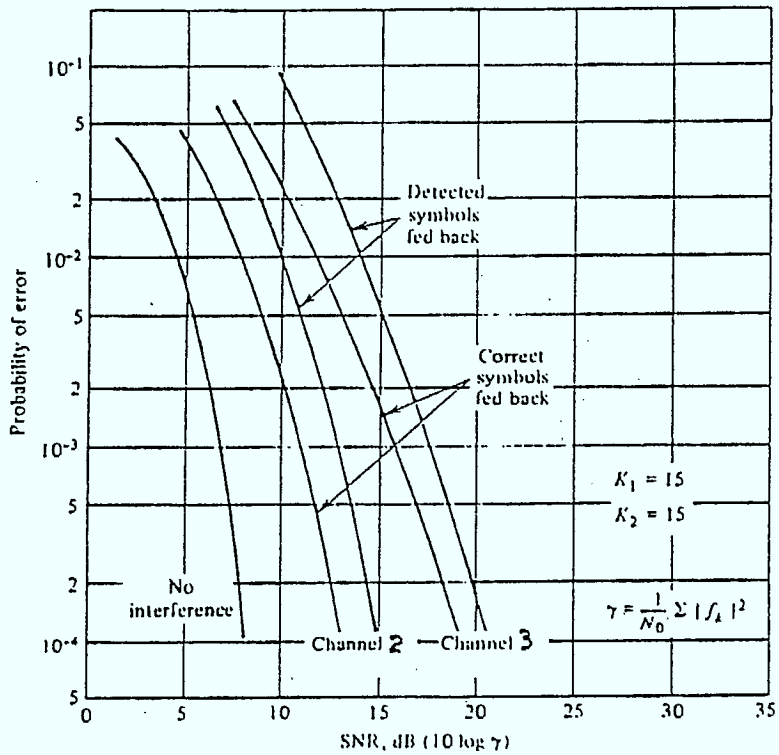


Figure 14: The performance of a decision-feedback equalizer for two example channels [7].

section is unnecessary, and all that is needed is the pure decision-feedback section. An equalizer consisting only of a pure decision-feedback section has several advantages. Since there is no linear transversal section, strong impulse noise will not be "smeared" by the equalizer. Errors that result from a strong impulse will of course effect the equalizer's performance, but no more than detection errors due to any other source.

Also, if the DC offset (i.e. the slicing level) is removed prior to equalization, the detection symbols can be assigned the values of +1 and -1. This has several benefits for digital implementation. The shift registor for the decision-feedback section need only be one bit wide. Secondly, the output of the decision-feedback section consists of a sum of the coefficients multiplied by +1 or -1. Note that a switchable complementer can be used instead of the multipler, that is required for a linear transversal filter. Such a complementer can be implemented as a basic VLSI cell [22], and hence a pure decision-feedback equalizer is well suited for VLSI implementation.

A special case of pure decision-feedback equalization that has some intuitive appeal is a decision-feedback equalizer with a small number of taps with tap delays equal to the estimated path delays. Despite the intuitive appeal, the author feels that this type of equalizer would be a poor choice. There are several reasons for this conclusion. The off-air measurements indicate that a fairly large number of taps would still be necessary (i.e. at least 5). The path delays are very difficult to estimate when there are merging paths. By merging paths we are referring to the case when the differential delay between two paths is too small for the paths to be distinctly resolved given the finite bandwidth of the channel. Furthermore such a

strategy is only valid for handling specular multipath. Diffuse multipath and other channel distortions (e.g. see the discussion in the preceding section) can not be handled by the strategy. For these reasons, this approach will not be addressed further by this study.

The relative merits of linear transversal equalization and pure decision feedback equalization will be compared in a subsequent section.

2.2.1 Coefficient Adaptation for Pure Decision-Feedback Equalizers

Since the pure decision-feedback equalizer uses the first significant sample of the channel's impulse response to form the decision, and then uses the decisions to cancel out the remainder of the channel's impulse response, the coefficients of the pure decision-feedback equalizer are given by the sampled impulse response of the channel. Thus the determination of the coefficients is in fact an impulse response estimation problem. In similar fashion to the previous section, we will begin by reviewing how the optimal (in the least squares sense) impulse response estimate can be computed directly. Then a gradient-based iterative structure is introduced.

Before proceding it is constructive to outline the underlying assumptions of the development. All of these assumptions are at least approximately satisfied for the teletext case. First, it is assumed that the data is sent at a known symbol rate (1/T). Thus, for the complex* baseband modelling considered here, the channel input becomes a train of impulses, with the nth impulse weighted by the nth data symbol a_n. It is further assumed that the

*We are giving the development for the complex case for the sake of completeness.

correct data symbols are known. That is either a known training signal is transmitted or the data decisions from the detector are correct. Next, it is assumed that the channel output is bandlimited, so that the output signal may be completely characterized by samples taken at a sufficiently high rate. Specifically, we initially assume that the channel is synchronously sampled N times per symbol period. Finally, it is assumed that the channel has a finite memory, so that only the M most recent symbols affect the channel output. An illustration of the channel estimation problem can be seen in Figure 15.

Given the discrete-time input and output sequences, the search for the best least squares model can be restricted to those models for which the channel impulse response is represented by MN (=L+1) equally spaced samples $\{f_k\}$, with the samples outside this range taken to be zero. Thus, the discrete-time error function at time nT (neglecting the stationary transmission delay) becomes

$$e_n(\underline{f}) = y_n - \hat{y}_n(\underline{f}), \tag{32}$$

where

$$\hat{y}_n(\underline{f}) = \sum_{k=0}^{MN-1} a_{n-k} f_k = \underline{a}^T \underline{f}, \tag{33}$$

and where superscript T denotes transposition. Note that for the discrete-time model and N samples per symbol period, only one out of every N elements of $\{a_n\}$ is nonzero.

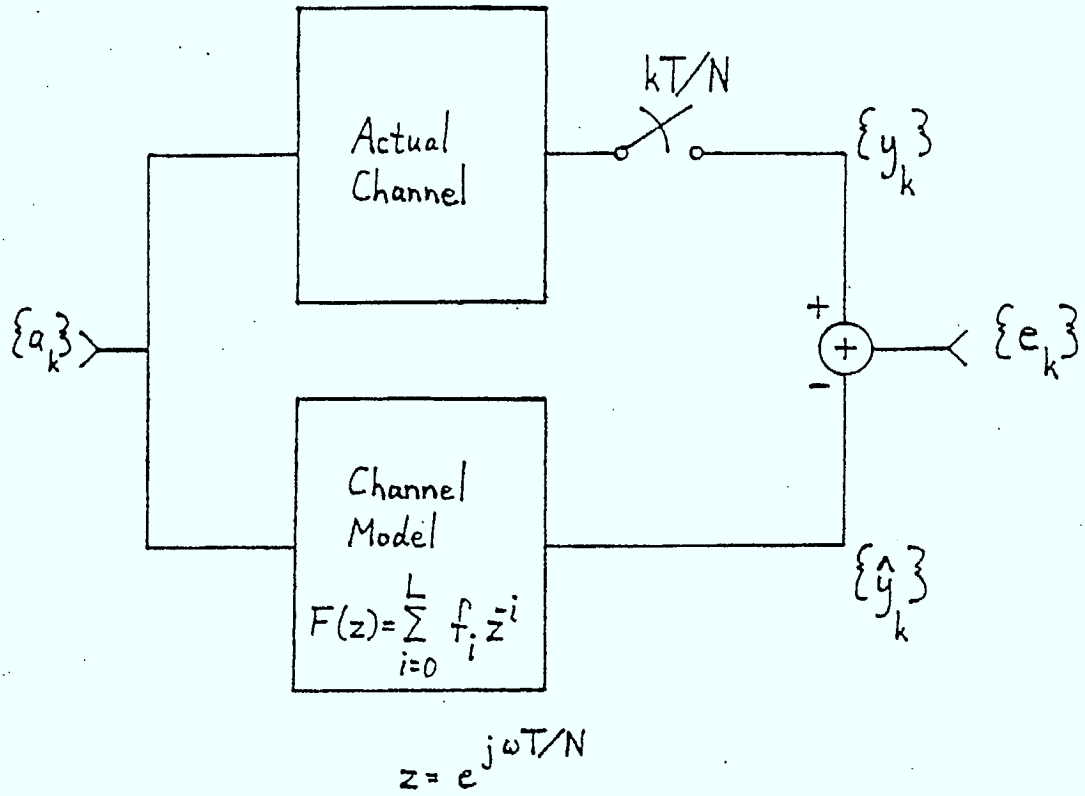


Figure 15: An illustration of the channel impulse response estimation problem.

In general, the least squares norm is given by

$$J(\underline{f}) = \underline{\varepsilon}^H(\underline{f})W\underline{\varepsilon}(\underline{f}), \tag{34}$$

where $\underline{\varepsilon}(\underline{f})$ is a K-vector with elements that are K samples of the discrete-time error function, W is a KxK positive semi-definite Hermitian matrix, and where superscript H denotes complex-conjugate transposition. Note that the matrix W can be used to compensate for the colouring of the input signal. Here, consideration will primarily be restricted to the case where W is a diagonal matrix (e.g. the identity matrix). Note that, in general, the elements of $\underline{\varepsilon}(\underline{f})$ need not be consecutive samples of $\{e_n(\underline{f})\}$.

The error vector $\underline{\varepsilon}(\underline{f})$ may be expressed in the form

$$\underline{\varepsilon}(\underline{f}) = \underline{y} - \hat{\underline{y}}(\underline{f}) = \underline{y} - D\underline{f}, \tag{35}$$

where $\underline{y}$ is a K-vector with elements being samples of the complex baseband channel output, and D is a KxMN matrix with rows being elements of the vector sequence $\{\underline{a}_n^T\}$.

It is well known that the choice of $\underline{f}$ that minimizes the norm defined by equations (34) and (35) is given by [23]

$$\underline{f}_{min} = (D^H WD)^{-1} D^H W\underline{y} \quad . \tag{36}$$

This $\underline{f}_{min}$ is unique only if $D^H WD$ is nonsingular. The probability of this is very high providing the transmitted data is relatively "random", W is positive definite, and $k \gg MN$. All of these conditions are normally satisfied in practice.

Since, for each row of D, only one out of every N elements is nonzero, the minimization problem can be decomposed into N smaller minimization problems by rewriting equation (35) in the form

$$\begin{bmatrix} \underline{\varepsilon}_1(\underline{f}_1) \\ \underline{\varepsilon}_2(\underline{f}_2) \\ \cdot \\ \cdot \\ \cdot \\ \underline{\varepsilon}_N(\underline{f}_N) \end{bmatrix} = \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \cdot \\ \cdot \\ \cdot \\ \underline{y}_N \end{bmatrix} - \begin{bmatrix} D_1 & 0 & & & \\ 0 & D_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & D_N \end{bmatrix} \begin{bmatrix} \underline{f}_1 \\ \underline{f}_2 \\ \cdot \\ \cdot \\ \cdot \\ \underline{f}_N \end{bmatrix}, \qquad (37)$$

where the D_i are $K_i x M$ matrices with elements consisting of data symbols, and $\underline{\varepsilon}_i(\underline{f}_i)$ and $\underline{y}_i$ are K_i-vectors. Here, the order of the elements in the error vector has been rearranged so that $\underline{f}_i$ is an M-vector with elements that represent M equally spaced impulse response samples with a spacing of N sample periods. If W is restricted to the block diagonal form

$$W = \begin{bmatrix} W_1 & & & & & \\ & W_2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & W_N \end{bmatrix}, \qquad (38)$$

where the W_i are $K_i x K_i$ Hermitian matrices, then the least squares norm becomes

$$J(\underline{f}) = \sum_{i=1}^{N} J_i(\underline{f}_i), \tag{39}$$

where

$$J_i(\underline{f}_i) = \underline{\varepsilon}_i^H(\underline{f}_i)\, W_i \underline{\varepsilon}_i(\underline{f}_i) \quad . \tag{40}$$

From equations (39) and (40) it can be seen that the least squares minimization in MN variables can be decomposed into N independent least squares minimization problems, each in M variables. It should be noted that for the determination of the pure decision-feedback coefficients only one of the N minimization problems needs to be solved. The one of interest is the one that yields the subset of impulse response samples, $\underline{f}_i$, that is phase synchronous with the symbol clock, since it is only at the moments when the detector samples the signal that the effects of the tail need to be removed. Aside from the obvious computational advantages, this decomposition is advantageous because each of the smaller minimization problems (including the one of interest to us here) has an effective sampling period equal to the symbol period, instead of 1/N of the symbol period which was initially assumed (and which is required for sampling the channel output in excess of the Nyquist rate). In this case, the D_i matrices are Toeplitz and very efficient techniques exist for solving the least squares estimation problem [24][25]. For practical reasons it may be desirable to sample the channel output using sub-symbol rate sampling. The resulting D_i will no longer be Toeplitz and thus the more computationally intensive standard matrix techniques must be used to solve the minimization problem.

Unfortunately, the direct computation of the equalizer coefficients using the above approach is probably too computationally intensive to be realized on a fixed

wordlength processor (e.g. a microprocessor) in near-real time. Thus, as was the case for transversal linear equalization, a more practical approach may be to employ an iterative procedure such as the one illustrated in Figure 16*. Note that from an implementational point of view the structure of Figure 16 is simpler than the one in Figure 10, because most of the multiplications shown in Figure 16 represent multiplication by +1 or -1 (which is not really multiplication at all). Also, the tap spacing in Figure 16 is equal to the symbol period whereas the tap spacing of a transversal equalizer should preferably be half of the symbol period.

2.3 A Comparison of Linear Transversal Equalization and Pure Decision Feedback Equalization

In this section we have discussed two possibilities for equalization; linear transversal equalization and pure decision-feedback equalization. Here, we will discuss the relative advantages and disadvantages of these two approaches.

The primary advantages of linear transversal equalization stem from the fact that the equalization is performed using a linear filter and therefore, in principle, the equalization is capable of equalizing signals other than just the data signal. Note that this is in contrast to decision-feedback equalizers which are only capable of equalizing the type of data signal for which they were designed. Therefore, with linear equalization, there is the possibility that the same equalizer could be used for both the teletext and the video signals. This would require several conditions to be met. Firstly, the linear filtering portion of the equalizer would have to be capable

*Clearly, for the off-air reception of teletext, many more feedback taps would be required than are shown in Figure 16.

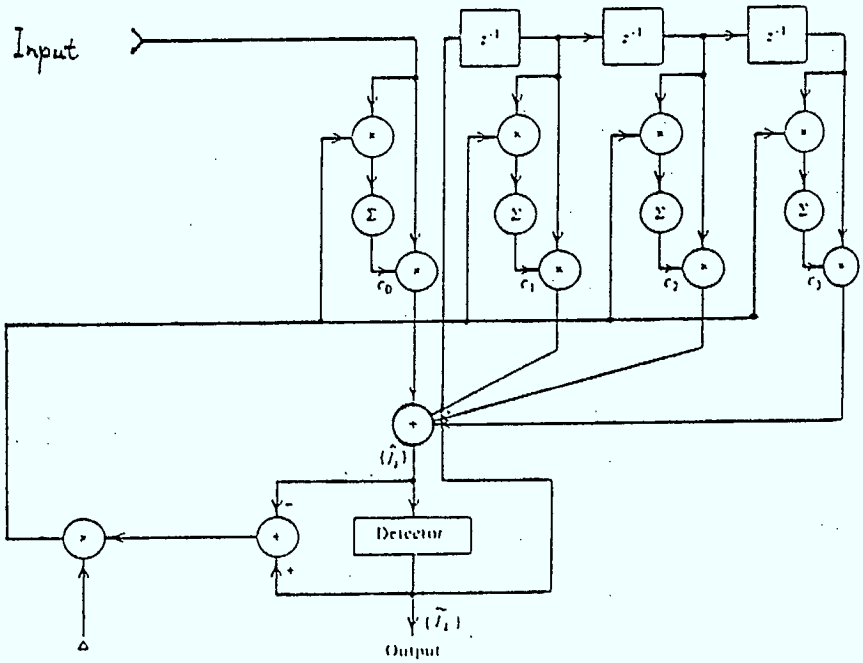


Figure 16: A block diagram illustrating an iterative structure for the adaptation of the coefficients of a pure decision-feedback equalizer.

of operating in real time. The issue or real time or nonreal time equalization will be discussed further in subsequent sections. Secondly, the teletext system would have to be set up so that the same filter transfer function is required for both signals. The possibility of equalizing the video signal as well as the data signal must be viewed as an important advantage, because this could justify the cost and complexity of an adaptive equalizer.

Both simulations and CRC off-air tests indicate that one of the more serious consequences of multipath propagation is synchronization failure to the superposition of the line sync pulse or the colour burst on the two-byte bit synchronization signal [1]. Linear equalization is capable of alleviating this problem if the equalizer is placed in front of the synchronization. On the other hand a decision-feedback equalizer could not cope with this situation because the colour burst and sync pulse are not data-like signals.

Pure decision-feedback has some advantages over linear equalization. These advantages have already been discussed but will be briefly reviewed here for completeness. Perhaps the most important advantage of the pure decision-feedback structure is that it will not "smear" strong impulses the way the a linear equalizer will. The significance of this advantage depends both on the type of noise that must faced and upon the effectiveness of strategies for limiting the effects of strong impulse noise. Unfortunately the author is not aware of any reference that addresses either of these issues in adequate depth for a sound judgement to be formed. Another advantage of the pure decision-feedback approach is that channels with inband nulls can be successfully equalized. As was mentioned previously, this is only an advantage if

one expects to receive teletext over a channel that is not capable of transmitting an acceptable quality video signal.

It was pointed out that a pure decision-feedback equalizer is probably more suited to a VLSI digital implementation than is a transversal linear equalizer. Note however that either type of equalizer can probably be realized by employing CCD techniques.

At this time it is felt that linear equalization represents the most promising approach. However, pure decision-feedback equalization should be kept in mind if strong impulse noise becomes an insurmountable problem for linear equalization. The remainder of this study will focus primarily on transversal linear equalization.

3.0 IMPLEMENTATIONAL CONSIDERATIONS FOR TELETEXT EQUALIZATION

In this section, we discuss some considerations regarding the implementation of an equalizer for the off-air reception of broadcast teletext. In this study we do not intend to go into implementional details. Rather, here we will describe several hardware components that are potentially useful for implementing a teletext equalizer. Then a possible equalizer structure is proposed.

Both advanced coding schemes and equalization are techniques that can be employed to improve the performance over poor channels. Thus, in a sense, equalization can be viewed as an alternative to advanced equalization schemes. Of course, we are not precluding the possibility of using both equalization and an advanced coding scheme simultaneously. Typically, the advanced coding schemes assume that the Hamming-encoded prefix is adequately protected and therefore the advanced coding scheme is the applied only to the data block. This assumption has the practical advantage that the decoder can decode the prefix "on the fly" and then decode the data block only for the lines that are requested by the user (i.e. from a given channel). Since a large number of channels are time-division multiplied, there is a reasonable amount of time for the decoder to decode a line from a given channel before the next line from that channel arrives. A minimum time separation of 4 ms is given in [26]. Thus the decoder has at least 4 ms to decode a line before the next line to be decoded is received. If a truly fair comparison is to be made between coding and equalization, then a mode of equalization where only the data block is equalized should be considered as a possible option. Here, this option will be referred to as "data-only equalization". In this mode, the average data rate that the equalizer must

contend with is no greater than 72 kbits/s. Note that the prefix must be passed through the equalizer's filter, even though only the data blocks will benefit from the equalization. This is necessary because the presence of the prefix will affect the data block due to multipath propagation. Clearly, data-only equalization represents one approach to non-real-time equalization.

The assumption that the Hamming-encoded prefix is adequately protected can not be justified for multipath environments where the eye is closed (i.e. detection errors can occur due to intersymbol interference even when there is no noise present). For example, consider a case where the signal-to-noise ratio is very high but the eye is closed. Suppose that the customer requests a packet address for which the intersymbol interference causes multiple errors. That is that one or more of the prefix bytes is rejected. Assuming that the channel is stationary and that the noise power is significant, lines bearing the desired packet address will continue to be rejected. This is in contrast to the case of noise induced errors, in which case the line would typically be properly received after a small number of retransmissions, for reasonable signal-to-noise ratios. It is for the above reason that equalization (and advanced coding) strategies that can operate on the prefix are desirable.

For vertical blanking interval transmission, up to 12 lines per field can be used for broadcast teletext transmission. Assuming that 288 bits are transmitted on a line, the maximum average bit rate that an equalizer must be capable of handling is 207.36 kbits/s. This observation suggests a second mode of non-real-time equalizer. Clearly this mode is only appropriate for vertical blanking interval transmission, and not for full-field transmission. Here, we will refer to this option as "VBI-only equalization".

Another possibility exists which allows for the non-real-time equalization of the entire line for either vertical blanking interval or full field transmission, will be referred to here as "scheduled equalization". For this scheme, it is assumed that packets on a given channel are always transmitted on a particular line of a particular field. For example, packets on channel 836 are always transmitted on line 16 of field 1. Thus the equalizer (and decoder) need only operate on one line per frame, which results in an average bit rate of 8.64 kbits/s. This is encouraging because this bit rate is in the range of that for high-rate data transmission over a voice grade channel, which is an application where both adaptive equalization and coding have been used entensively. The disadvantage of this approach is that it places a restriction on the format of the transmissions, that is not included in any of the current specifications for broadcast teletext [26][28].

The fourth option considered here is real-time equalization. For this option, a discrete-time transversal filter should be capable of operating at a sampling rate of 11.4545 MHz (i.e. T/2 sampling). Although the equalization of the television video signal will not be discussed in this section, it should be kept in mind that this third option is the only one for which the same equalizer can be used to equalize both the video and the teletext signals.

Before proceding with the description of a possible equalization strategy, we will first perform a brief survey of some useful hardware components. This survey is intended to be representative of the available components rather than be comprehensive in nature.

3.1 Possible Building Blocks for Teletext Equalization

Most of the devices that will be discussed here are devices that are capable of performing convolution (i.e. the programmable linear filtering function) on wideband signals. However, the first device that is discussed is not a convolver. It is a device that can be used for "time expansion" for non-real-time processing.

Not discussed here are the Philips teletext equalizer [11] and SAW wideband convolvers [30][31]. The Philips equalizer is not considered here because it has already been discussed and it was concluded that this device is not appropriate for the equalization of off-air signals in an urban environment. SAW convolvers are not discussed because the author feels that it is unlikely that these devices will provide a cost-effective means of equalizing the teletext signals.

3.1.1 CCD Video Analog Shift Registers

Several manufacturers make video-rate CCD analog shift registers (e.g. the Fairchild CCD321A and the Reticon R5103). Typically these devices are designed to store one full line of the video signal at a sampling rate of four times the colour subcarrier frequency (i.e. 14.318 MHz). The length of these shift registers is therefore usually 910 samples. Clearly one of these shift registers is capable of storing more than a full video line at the 2/T rate of 11.4545 MHz. Once a line has been stored, it can be clocked out of the shift register at a much lower rate for A/D conversion* or some other processing operation.

*Video-rate sample-and-hold and analog-to-digital devices are commercially available, but they are very expensive. Furthermore, high speed digital memory is required to store the outputs of these devices.

3.1.2 CCD Analog-Analog Convolvers

Convolvers can be constructed using CCD registers and analog multipliers. A block diagram illustrating this type of convolver can be seen in Figure 17. Reticon manufactures a 32-tap CCD analog-analog convolver that is capable of operating at sampling rates up to 2 MHz. The length of this device is only marginal for the given application. (Recall that in Section 2.0, 35 taps was suggested as the minimum number of taps for the equalization of off-air signals in downtown Ottawa.) Also, the maximum sampling rate is not high enough for real-time equalization. However, the maximum sampling rate is more than adequate for VBI-only equalization.

A very promising CCD analog-analog correlator has been developed in Japan [18]. This device has 128 taps and was designed to operate at a sampling rate of 10.7 MHz, which is three times the colour subcarrier frequency. Thus it could probably operate at the 2/T rate of 11.4545 Hz. This device has been successfully tested for the equalization of the video signal, but to the best of the author's knowledge it has not been tested for teletext. Thus the author does not know if it possesses sufficient bandwidth and linearity for the teletext application. However, this device does provide some encouragement that a CCD convolver is capable of providing cost-effective real-time equalization of teletext.

3.1.3 CCD Digital-Analog Convolvers

CCD techniques can also be used to realize digital-analog convolvers. A block diagram illustrating the structure of a digital-analog correlator can be seen in Figure 18. Digital-analog convolvers have the advantage over analog-

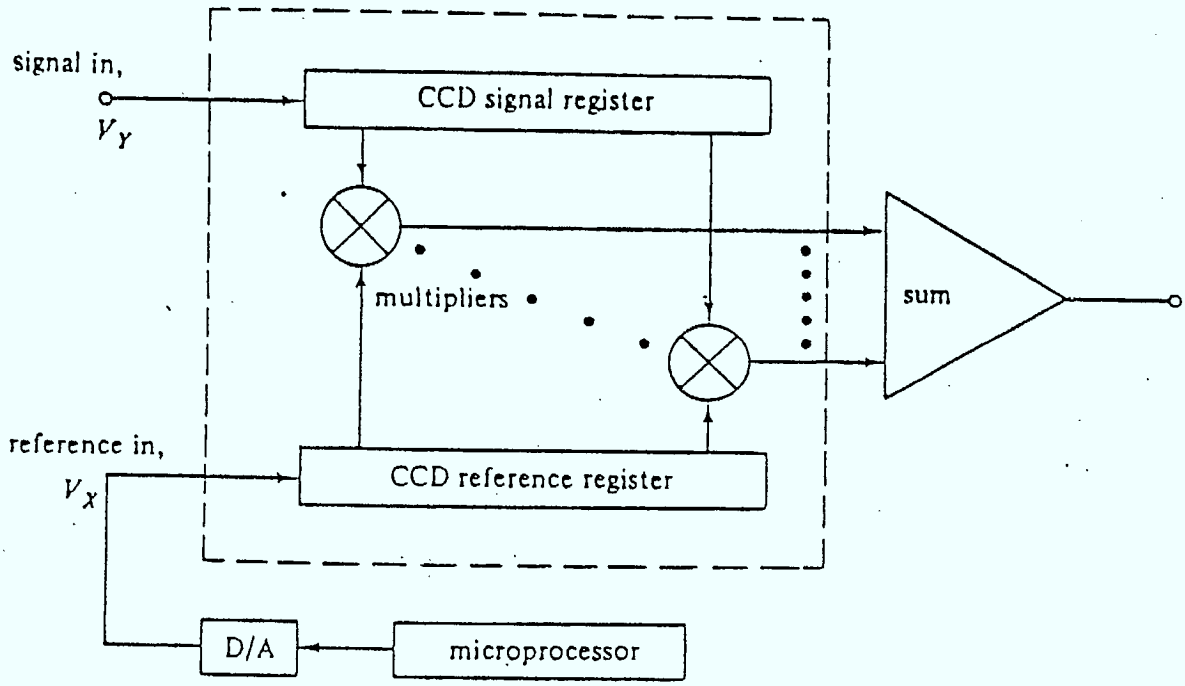


Figure 17: A conceptual block diagram illustrating the structure of a CCD analog-analog convolver [27].

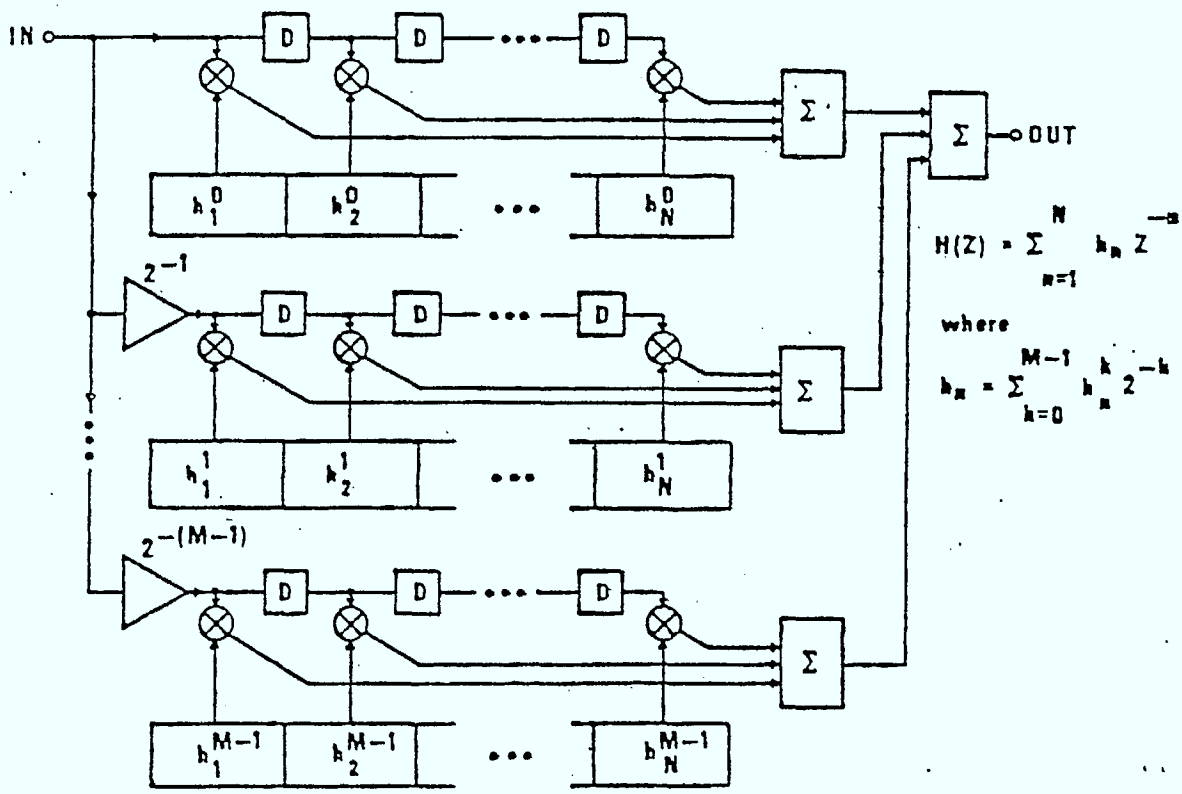


Figure 18: A block diagram of a digital-analog correlator [29].

analog convolvers that the filter coefficients are stored statically on the chip in digital form. A digital-analog convolver has been built at Texas Instruments Inc. [29], that can be configured so that it has 128 taps with 8-bit coefficients. However, the maximum sampling rate for this device is 1 MHz. Thus, it is not fast enough to be used for the real-time equalization of teletext signals. However, it is fast enough for all of the other equalization options.

3.1.4 Components for Digital Filtering

TRW has recently begun to manufacture an FIR digital filter building block that can operate at sampling rates of up to 20 MHz [32] and is therefore capable of providing real-time teletext equalization. This building block is essentially an 8-tap FIR digital filter, with 4-bit coefficients and a 4-bit wide input signal path, which has been configured so that multiple devices can easily be paralleled and cascaded in order to construct a filter with a longer word length and more taps. Although this device has some interesting possibilities, it is felt that a programmable filter with enough taps and a long enough word length for the teletext application, would be too expensive to represent a practical solution at this time.

High-speed digital signal processing components (e.g. hardware multiplier and multiplier-accumulator chips) can be used to construct a high-speed programmable digital filter. An example of this type of filter is the ADF-16 digital FIR filter board that is produced by the Ottawa-based company Interactive Circuits and Systems Limited. This is a 16-bit programmable FIR filter that is capable of operating at 10/N MHz, where N is the number of filter taps. Thus this device could be used for either the data-

only equalization strategy or the scheduled equalization strategy. At present this filter board is sold for approximately one thousand 1983 Canadian dollars. However, it must be considered as a specialty device (i.e. low quantity). If it were manufactured for the consumer market, its cost would probably drop to a couple hundred dollars. Even at this lower cost, it must be considered a relatively high cost solution.

If a strategy such as the scheduled equalization strategy is adopted, then a LSI digital signal processor chip (or chip set) may represent an economical solution to the equalizer realization problem. Examples of this type of device include the Texas Instruments TMS320 [33] and the Nippon Electric NEC μPD7720 [34]. In addition to performing the programmable filtering these devices might also be able to perform the parameter adaptation on a time-shared basis.

3.2 A Possible Structure for the Linear Equalization of Teletext

A conceptual block diagram illustrating a possible structure for the linear equalization of teletext can be seen in Figure 19. This diagram is at the conceptual level in that only the signal paths are shown. The many timing and control lines that would be necessary are not included in this illustration. Several points about this structure should be mentioned. Note that both the input signal to the equalizer and the error signal (i.e. the difference between the signals prior to and following the detector) are stored in the microcomputer's random-access memory. Thus the microcomputer has access to all of the information it needs to perform a closed-loop iterative coefficient adaptation algorithm.

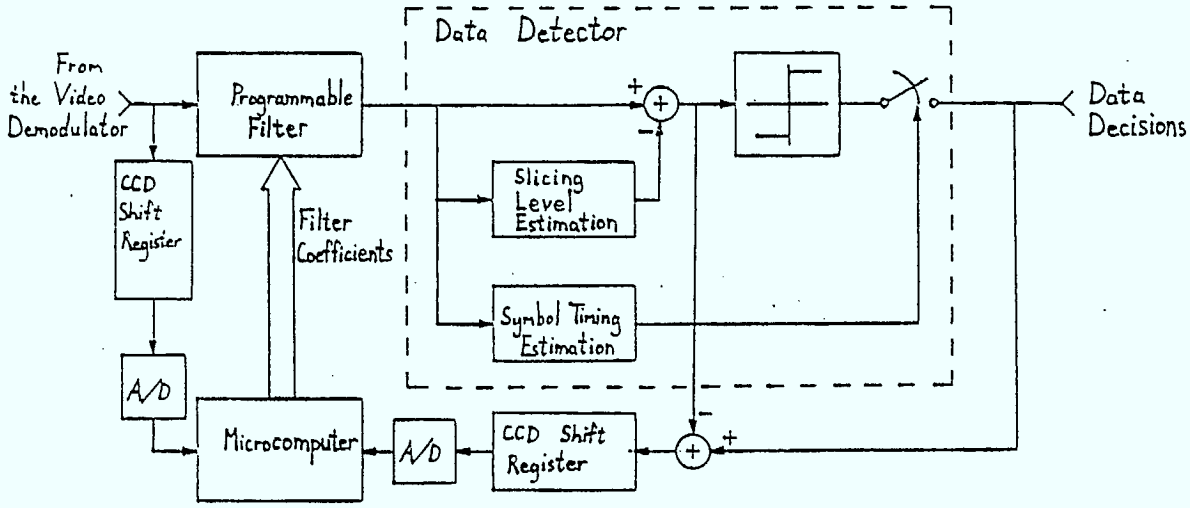


Figure 19: A conceptual block diagram illustrating the structure of the linear equalization concept.

An important point to note is that the slicing level estimation and the symbol timing estimation (i.e. bit clock phase recovery) are implemented after the programmable filter. This is a highly desirable arrangement because multipath propagation can severely degrade the performance of these synchronization circuits.

The microcomputer must aid in the various synchronization and control tasks as well as being responsible for the coefficient adaptation. It may be advantageous to speed up the microcomputer by including a hardware multiplier. One candidate, for the hardware multiplier, is the RCA CDP1855. This CMOS device has a lower cost and a lower power consumption than the faster bipolar multipliers that are manufactured by companies such as TRW and Advanced Micro Devices Inc.

With the coefficient adaptation being implemented on a microcomputer, the adaptation strategy can be made somewhat more intelligent than a direct software implementation of the structure shown in Figure 10. For example a given line might only be used for the coefficient adaptation if no errors are detected by the coding scheme.

The structure illustrated in Figure 19 is for real-time equalization. However, with minor modifications it can be applied to the other equalization options. The required modification, to the equalization structure, for VBI-only equalization, is shown in Figure 20. Note that a bank of multiplexed CCD shift registers are required at the input. A shift register is required for each teletext-bearing line in a field. Thus the number of shift registers is given by the maximum number of lines in a field that can be used for teletext (e.g. 12 in [26]). For the data-only equalization

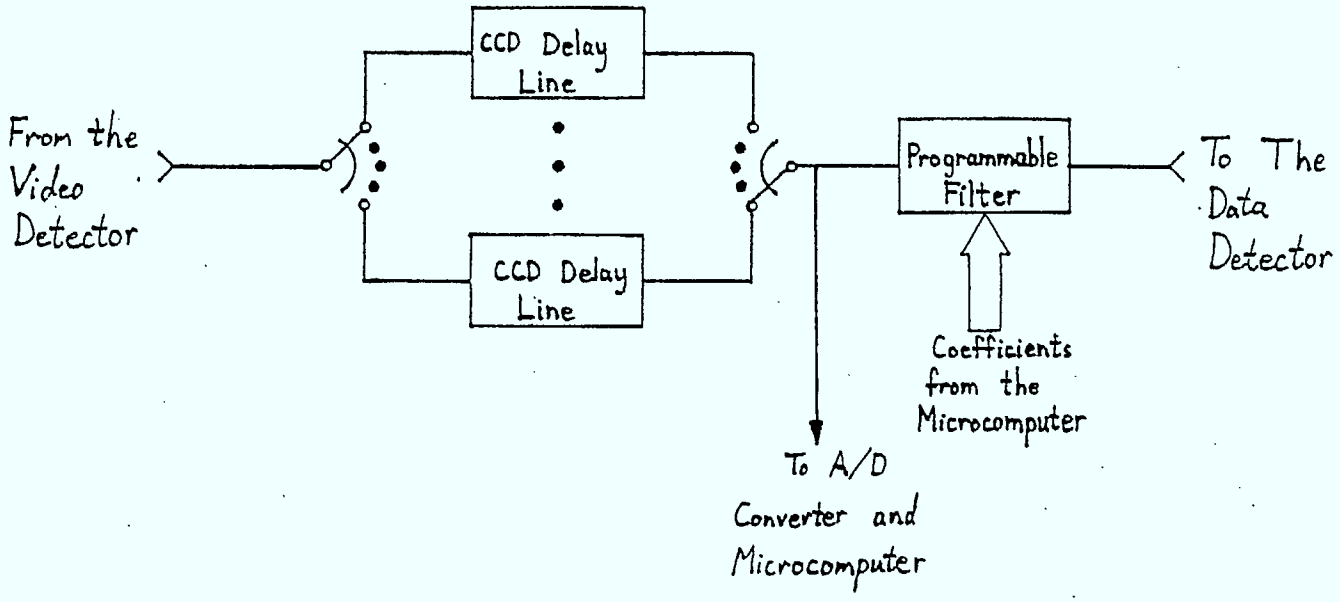


Figure 20: An illustration of the modification, to the equalization structure, for VBI-only equalization

strategy and the scheduled equalization strategy the bank of multiplexed shift registers can be replaced by a single shift register.

In Figure 19, no reference is made as to which portions of the structure are implemented in the digital domain and which portions of the structure are implemented in the analog domain. The reason for this is that the choice of which portions of the structure should be implemented in the given domains is highly technology dependent. Since the candidate technologies are still dynamically evolving a detailed specification of the equalizer is both difficult and undesirable.

4.0 SUMMARY AND CONCLUSIONS

Section 1.0 began with a review of the complex baseband model that is used for the study. It was noted that in some cases the multipath propagation can be time-varying in nature. The signal transmission impairments, for which adaptive equalization can compensate, were said to include multipath propagation, quadrature distortion, deviations from nominal of the channel and filters, nonlinear distortion and coloured noise. These various distortions were then discussed. During the discussion about multipath propagation it was pointed out that, except for possibly the lower VHF channels, the multipath channel will likely often include some non-specular reflected paths. Some of the highlights of the CRC impulse response measurements were reviewed. It was noted that of the acceptable channels that were measured, most but not all of them were minimum phase. The little information that is available about echoes in CATV was reviewed. It was concluded that off-air reception in urban environments, not CATV, is likely to be the ultimate test for adaptive equalization.

The likely noise environment was briefly discussed. The important observation, that the noise is often dominated by impulse noise in the urban environments where multipath propagation is the most severe, was made.

It was pointed out that the equalization can be implemented in the complex baseband (or equivalently at IF) or in the real baseband. Although complex baseband equalization exhibits superior performance, the study focuses on real equalization for practical reasons.

In Section 2.0 some fundamentals of equalization were reviewed. Linear equalization was addressed first. It was shown that linear equalization is capable of reasonably

good performance over both minimum phase and non-minimum phase channels, but that it has difficulty dealing with channels with "high Q" roots. Linear recursive equalization was discarded as a candidate primarily because of its susceptability to limit-cycle behaviour. Thus consideration of linear equalizers was restricted to transversal equalizers. Nyquist's sampling theorem was used to argue that the tap spacing, for such an equalizer should be no more than 119 ns. This implies that a T/2 equalizer with a tap spacing of 87.3 ns is a reasonable choice.

From the fact that most of the significant echoes, in the Ottawa-area measurements, have delays of less than 3 μs, it was concluded that a T/2 equalizer should have at least 35 taps if it is to be used for the off-air reception of teletext in downtown Ottawa. More taps could be required if the equalizer is to be used in a bigger city or at a higher frequency (i.e. UHF).

It was suggested that the reason that the Philips equalizer did not work well for off-air reception in New York city is that its support is not large enough (i.e. it does not have enough taps). Although the Philips equalizer's performance was dismal for the off-air tests in New York city, it would probably perform quite well for most CATV channels.

One possible drawback of linear equalization in strong impulse noise environments was mentioned. If linear equalization is employed, where the equalizer has a support of at least 3 μs, then the energy of a strong impulse would be dispersed by the equalizer over a time duration of at least 3 μs. If the impulse's energy is orders of magnitude larger than the energy per bit of the data signal, then the dispersion of the impulse's energy might result in significantly more errors than if no equalization were

performed. One possibility to improve the performance of linear equalization is to precede the equalizer with a limiter that would reduce the magnitude of strong impulses but leave the remainder of the received signal unaltered.

The computation of the coefficients of a discrete-time transversal equalizer was discussed. Due to its tractability, attention was focused on coefficient adaptation based on the minimizing the mean square error criterion. Both the direct computation of the optimal coefficients and a closed-loop iterative adaptation approach were outlined. It was pointed out that a data-like signal is one of the leading candidates for the equalizer training signal.

Decision-feedback equalization was also discussed. It was pointed out that one of the advantages of decision-feedback equalization over linear equalization is the ability of the former to provide reasonably good performance for channels with inband nulls. However, not too much emphasis should be placed on this advantage since any channel with inband nulls would certainly be unacceptable for television viewing. The special case of decision-feedback equalization, referred to as pure decision-feedback equalization, was discussed. It was noted that a pure decision-feedback equalizer has some features which make it well suited for VLSI digital implementation.

The computation of the coefficients of a pure decision-feedback equalizer was discussed. Both the direct computation of the optimal (in the least squares sense) coefficients and a closed-loop iterative adaptation approach were outlined.

The relative advantages of transversal linear equalization and pure decision-feed equalization were discussed. It was

decided to focus the attention of this study on transversal linear equalization because of the inability of decision-feedback equalization to remove delayed (via reflected paths) versions of the line sync pulse and the colour burst from the teletext prefix.

In Section 3.0 some considerations regarding the implementation of an equalizer for the off-air reception of broadcast teletext, were discussed. Several possible schemes for non-real-time equalization were outlined, in addition to real-time equalization. A brief survey of some useful hardware components was given. The intention of this survey is to be representative rather than comprehensive in nature.

A possible structure for the implementation of a transversal equalizer was proposed. A couple of important features of this structure should be highlighted. Firstly, the equalizer should precede the slicing level estimation and the symbol timing estimation since both of these circuits tend to be sensitive to multipath propagation. Secondly, a closed-loop iterative coefficient adaptation algorithm is suggested due to the robustness of these algorithms in the presence of nonlinearities and other nonideal phenomena.

Since equalization and coding are, in a sense, competitive approaches to improving the performance of broadcast teletext, it is probably appropriate to review the relative strengths of each of these approaches. Coding has the advantage that it is simpler and cheaper to implement, at least at this time. Most of the codes that are under consideration can be implemented using no more than a microcomputer and a look-up table (i.e. read-only memory). Also, coding may be more effective than linear equalization for handling strong impulse noise, although this is yet to

be verified. Note however that coding is not an effective tool when multipath propagation and other sources of distortion close the eye. In this case, intersymbol interference can cause "steady-state" error patterns. If some of these patterns are uncorrectable patterns, then that data can virtually never be correctly received regardless of how many times it is retransmitted.

Linear equalization has the advantage over coding that it can equalize the teletext prefix. This has two important consequences. One is that the reception of the prefix data (e.g. the packet address) can be improved. The other is that synchronization (e.g. slicing level estimation and symbol timing estimation) can be improved. Note that this second point is beyond the scope of coding, even if more powerful coding schemes were applied to the prefix. Another advantage of linear equalization is that it may be possible to employ much of the same hardware (and possibly software) for the equalization of both the teletext signal and the television video signal.

Although it appears that it is possible to construct a linear equalizer for the reception of off-air teletext signals, such an equalizer is likely to be a fairly expensive device. Obviously, the cost versus performance tradeoff is highly dependent upon the current technology. The candidate technologies for teletext equalization are still dynamically evolving. Furthermore, the potential increase in performance that linear equalization offers is yet to be quantified.

REFERENCES

[1] K.W. Moreland and J.H. Lodge, "Telidon System Study 3rd Interim Report", Miller Communications Systems Ltd. Report, DSS File No. DSS 21ST.36100-2-4380, MCS File No. 8342, September 1983.

[2] R.K. Tiedemann, "Simulation of VHF Multipath Propagation Model for Telidon", MCS Memorandum, MCS File No. 8342, April 1983.

[3] W. Ciciora et al., "A Tutorial on Ghost Cancelling in Television Systems", IEEE Trans. on Consumer Electronics, Vol. CE-25, pp. 9-44, February 1979.

[4] B. Mazur et al., "Monitoring and Equalization of In-Service QPSK/TDMA Links by Means of Burst-Mode Channel Impulse Response Measurements", Sixth International Conference on Digital Satellite Communications, pp. I-A-18 - I-A-25, Phoenix, September 1983.

[5] K.W. Moreland et al., "Telidon System Study 2nd Interim Report", Miller Communications Systems Ltd. Report, DSS File No. DSS 21ST.36100-2-4380, MCS File No. 8342, June 1983.

[6] Reference Data for Radio Engineers, ITT, Howard W. Sams and Co., Inc., 1972.

[7] J.G. Proakis, Digital Communications, McGraw-Hill Book Company, New York, 1983.

[8] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Prentice-Hall Inc., Englewood Cliffs NJ, 1975.

[9] J.G. Proakis, "Advances in Equalization for Intersymbol Interference", Advances in Communications Systems, Vol. 4, Academic Press, 1975.

[10] A. Antoniou, Digital Filters: Analysis and Design, McGraw-Hill Book Company, New York, 1979.

[11] J.O. Voorman et at., "One-Chip Automatic Equalizer for Echo Reduction in Teletext", IEEE Trans. on Consumer Electronics, Vol. CE-27, No. 3, August 1981.

[12] D. D. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Co., Reading, MA, 1973.

[13] R. D. Gitlin and F. R. Magee Jr., "Self-Orthogonalizing Adaptive Equalization Algorithms", IEEE Trans. on Comm., Vol. COM-25, pp. 666-672, July 1977.

[14] E. H. Satorius and G. T. Alexander, "Channel Equalization Using Adaptive Lattice Algorithms", IEEE Trans. on Comm., Volume COM-27, pp. 899-905, June 1979.

[15] D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization", IEEE Trans. on Comm., Vol. LCOM-26, pp. 1439-1445, October 1978.

[16] E. H. Satorius and J. D. Pack, "Application of Least Squares Lattice Algorithms to Adaptive Equalization", IEEE Trans. on Comm., Vol. COM-29, pp. 136-142, February 1981.

[17] B. Widrow et al., "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter", Proc. of the IEEE, Vol. 64, pp. 1151-1162, August 1976.

[18] M. Obara et al., "A TV Ghost Canceller Using a CCD Transversal Filter", IEEE Trans. on Broadcasting, vol. BC-29, pp. 60-71, June 1983.

[19] G. K. Goyal and S. C. Armfield, "Reception of Teletext Under Multipath Conditions", IEEE Trans. on Consumer Electronics, Vol. CE-25, pp. 378-392, July 1979.

[20] H. Thedick, "Adaptive Multipath Equalization for TV Broadcasting", IEEE Trans. on Consumer Electronics, pp. 175-181, May 1977.

[21] M. Schetzen, "Nonlinear System Modeling Based on the Wiener Theory", Proc. of the IEEE, Vol. 69, pp. 1557-1573, December 1981.

[22] N. Frenette et al., "A CMOS Implementation of a Binary Correlation for Digital Communications", 1983 Canadian Conference on Very Large Scale Integration, Waterloo, Ontario pp. 108-111, October 1983.

[23] A. S. Householder, The Theory of Matrices in Numerical Analysis, Dover Publications, New York, 1964, pg. 98.

[24] S. L. Marple, Jr., "Efficient Least Squares FIR System Identification", IEEE Trans. Acoust., Speech, and Signal Proc., vol. ASSP-29, pp. 62-73, February 1981.

[25] S. L. Marple, Jr., and L. R. Rabiner, "Performance of a Fast Algorithm for FIR System Identification Using Least Squares Analysis", BSTJ, vol. 62, pp. 717-742, March 1983.

[26] "Broadcast Specification BS-14", Department of Communications, Government of Canada, Ottawa, Ontario, Issue 1, June 1981.

[27] J. D. E. Beynon and D. R. Lamb, Editors, Charge-Coupled Devices and Their Applications, McGraw-Hill Book Company Limited, London, 1980.

[28] "North American Broadcast Teletext Specification", September 20, 1983.

[29] R. A. Haken et al., "A General Purpose 1024-Stage Electonically Programmable Transversal Filter", IEEE Journal of Solid-State Circuits, Vol. SC-15, pp. 984-996, December 1980.

[30] J. D. Maines and E. G. G. Paige, "Surface-Acoustic-Wave Devices for Signal Processing Applications", Proc. of the IEEE, pp. 639-652, May 1976.

[31] L. B. Milstein and P. K. Das, "Surface Acoustic Wave Devices", IEEE Communications Magazine, pp. 25-33, September 1979.

[32] F. Williams, "Video-Speed Filtering Gets Its Own Digital IC", Electronics, pp. 148-151, October 20, 1983.

[33] E. R. Caudel and R. K. Hester, "A Chip Set for Audio Frequency Digital Signal Processing", International Conference on Acoustics, Speech, and Signal Processing, pp. 1065-1068, April 1982.

[34] T. Nishitani et al., "LSI Signal Processor Development for Communications Equipment", International Conference on Acoustics, Speech, and Signal Processing, pp. 386-389, April 1980.

APPENDIX II

DETERMINING THE PERFORMANCE OF ERROR PROTECTIVE CODES FOR TELIDON

1.0 INTRODUCTION

Selection of an appropriate error protection code for Telidon broadcast videotext requires that the performance of candidate codes be observed under realistic error situations. To this end, it is proposed that the codes be applied to error sequences produced by the computer simulation of the channel. This appendix outlines a method of performing the performance measurements for six codes:

(1) Product

(2) C_1 (Carleton 1 byte),

(3) C_2 (Carleton 1 byte),

(4) C_b (Carleton bundle),

(5) SAB_2 (Concordia 2 byte),

(6) SAB_3 (Concordia 3 byte),

The appendix will review and update previous memos on simulation validation, error sequence modelling, code performance measures, and finally a code by code explanation of how each code can be implemented in software.

2.0 SIMULATION VALIDATION

From an error sequence viewpoint, a check on simulation accuracy is to compare the characteristics of the errors measured in the field with those produced by the computer simulation. The characterization of errors is treated in [1]. Of the many statistics that may be collected, four are

chosen because of their applicability in modelling the sequences. The four statistics are P(m,224), G(j), B(j), and a(j), where:

P(m, 224) = prob of m errors in 224 bits,

G(j) = prob. of a gap of j error free bits, (e.g. 11101111 is a gap of 2)

B(j) = prob. of a burst of j errors, (e.g. 00011000 is a burst of 2)

a(j) = error autocorrelation,

$$a(j) \triangleq \frac{\sum_{i=1}^{N-j} z_i \, z_{i+1}}{\sum_{i=1}^{N} z_i}$$

where

$$z_i = \begin{cases} 1 & \text{for an error in bit } i \\ 0 & \text{for no error in bit } i \end{cases}$$

The software for determining these statistics, from records of error positions has been developed, partially by CRC [2] and completed in order to include all four statistics by MCS.

3.0 ERROR SEQUENCE MODELLING

3.1 The Need for a Model

A model must be used to generate error sequences because it is infeasible to generate a sufficient number of packets using the simulation directly. If statistics are to be collected on the performance of codes having a post-decoding BER $< 10^{-6}$, then many million of bits would have to be produced by the simulation. The complexity of the simulation prohibits this. The pre-decoding BER should be much higher (say $\approx 10^{-3}$); hence, it is hoped that reasonable statistics, which characterize the pre-decoder errors, can be obtained from shorter data sequences. These characterizations would then be used to efficiently (computationally) generate error sequences, similar to those produced by the simulation.

Another need for error characterization is for measuring the effects of channel enhancements. By examining the error statistics (or model parameters) changes in error characteristics can be observed and general performance predicted without the use of error protective decoding.

3.2 The Model

The characterization and generation of error sequences is covered in [1]. The model recommended in [1] is a Fitchman model having multiple gap and error states. Unfortunately this model will have to be reduced to one having a single error state in order to facilitate parameterization [3]. The model consists of a measured gap length distribution (G(j)) and geometric burst length distribution (see Figure 1). The geometric distribution would be found by a least squares fit to the measured burst length distribution (B(j)). Error sequences are generated by:

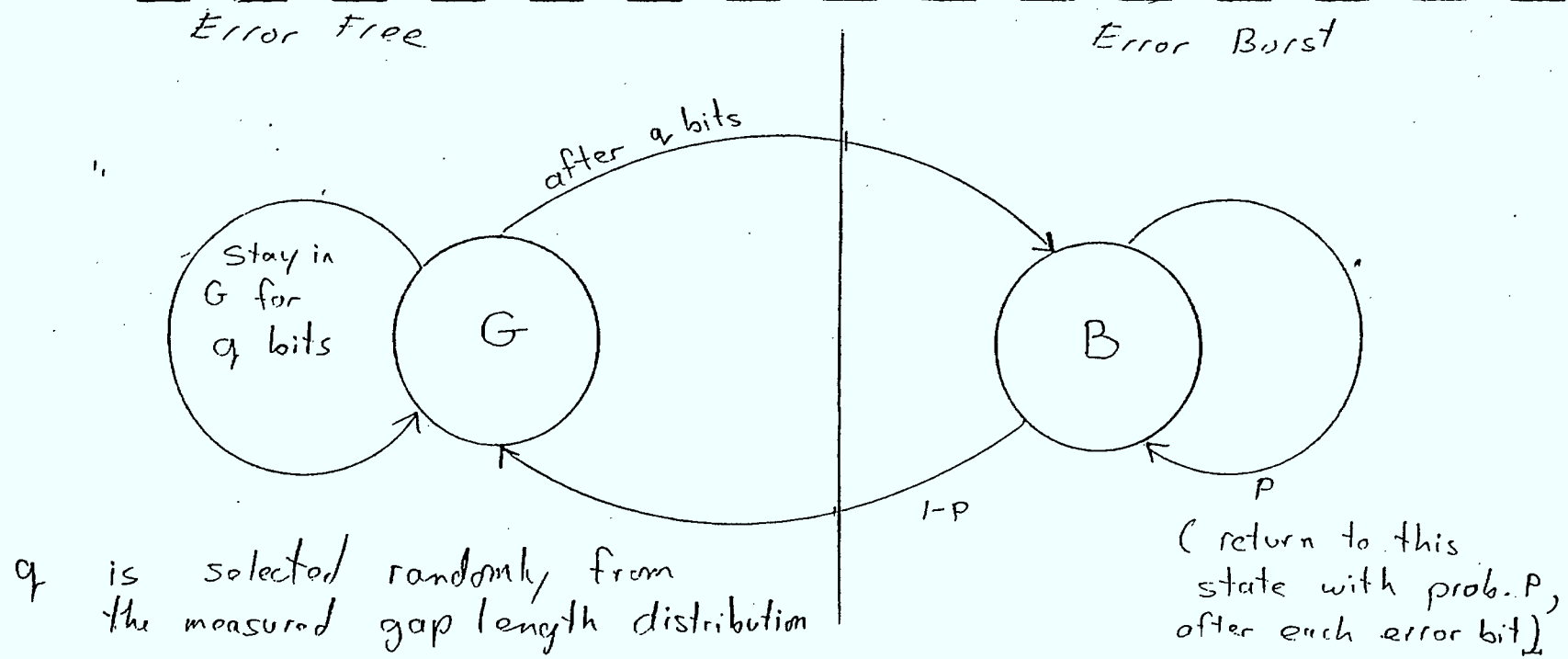


Figure 1: Fritchman's Model

(i) generate a gap of length g by selecting g in accordance with the gap distribution,

(ii) generate a burst of length b by selecting b in accordance with the geometric distribution of bursts,

(iii) return to (i).

Gap length distributions will not be expressed analytically; hence, gaps should be produced by using a measured inverse CDF of the gap lengths and a uniform random number generator. Burst lengths can be generated by:

$$j = \mathrm{INT}\left(\frac{\log(1-U)}{\log(p)}\right) + 1,$$

U is uniform random over [0, 1),

$$B(j) \cong (1-p)p^{j-1},$$

3.2.2 Model Validity

The model assumes that successive gap lengths are uncorrelated (renewal) and that the burst lengths are geometrically distributed. An indication of how true these assumptions are can be obtained by:

(i) calculating the χ^2 statistic of the observed B(j) and the l.s. fit of B(j) to $(1-P)P^{j-1}$,

(ii) using G(j) to calculate a(j) and P(m,224).

The first test simply indicates how valid the geometric assumption is. If the geometric and renewal assumptions are both true, then the predictions, found in the second test of

a(j) and P(m,224) should match those observed directly from the original error sequence. The formula for a(j), P(m,n) are [4]:

$$a(0) = 1, \quad a(1) = G(1),$$

$$a(j) = G(j) + \sum_{s=1}^{j-1} G(s)\, a(j-s), \quad j>1, \tag{1}$$

$$P(m,n) = \sum_{j=1}^{n-m+1} \rho\, Q(j-1)R(m,n-j+1), \quad 1 \leq m \leq n \tag{2}$$

where:

$$R(m,n) = \sum_{j=1}^{n-m+1} G(j)\, R(m-1,\, n-j), \quad 2 \leq m \leq n$$

$$R(1,n) = Q(n-1) = \sum_{j=n}^{\infty} G(j), \text{ and}$$

$$\rho = \frac{1}{\sum_{j=1}^{\infty} jG(j)} = \text{BER}.$$

4.0 PERFORMANCE MEASURES

Referring to [5], performance degradation from a user viewpoint is composed of delays due to detected errors, and image corruption due to undetected errors. If the only coding were the data block coding then the delays and corrupting errors could be determined by counting the failures and errors of the code under consideration. Unfortunately, two codes are used. The packet prefix is only Hamming (8,4) encoded, any header is both Hanning (8,4) then data block (C, SAB, or Product) encoded. The PDIs are encoded by the data block code alone [6]. The problem is simplified by:

(1) assuming all prefix errors and failures result in packet rejection [7],

(2) ignoring the effects of the header because of its infrequent occurance and doubly protective coding.

The task is then simplified to:

(1) Get next packet containing errors,

(2) If there is an uncorrectable prefix error, go to 4,

(3) Decode the data block. If an uncorrectable error is detected then go to 4. If all errors are corrected, go to 1. Otherwise, increment the packet, PDI and bit error counts and return to 1.

(4) Increment the packet rejection count and return to 1.

The use of bundle codes complicates the above procedure because these codes can be used to replace a missing line (rejected packet). For bundle codes the procedure must be modified so that the bundle is assembled and if 2 or more prefix errors occur, a bundle rejection is declared.

For simplicity, a bundle rejection will be treated as h packet rejections, where h is the number of packets in a bundle. The true number may be less, depending on the particular error detection circumstances.

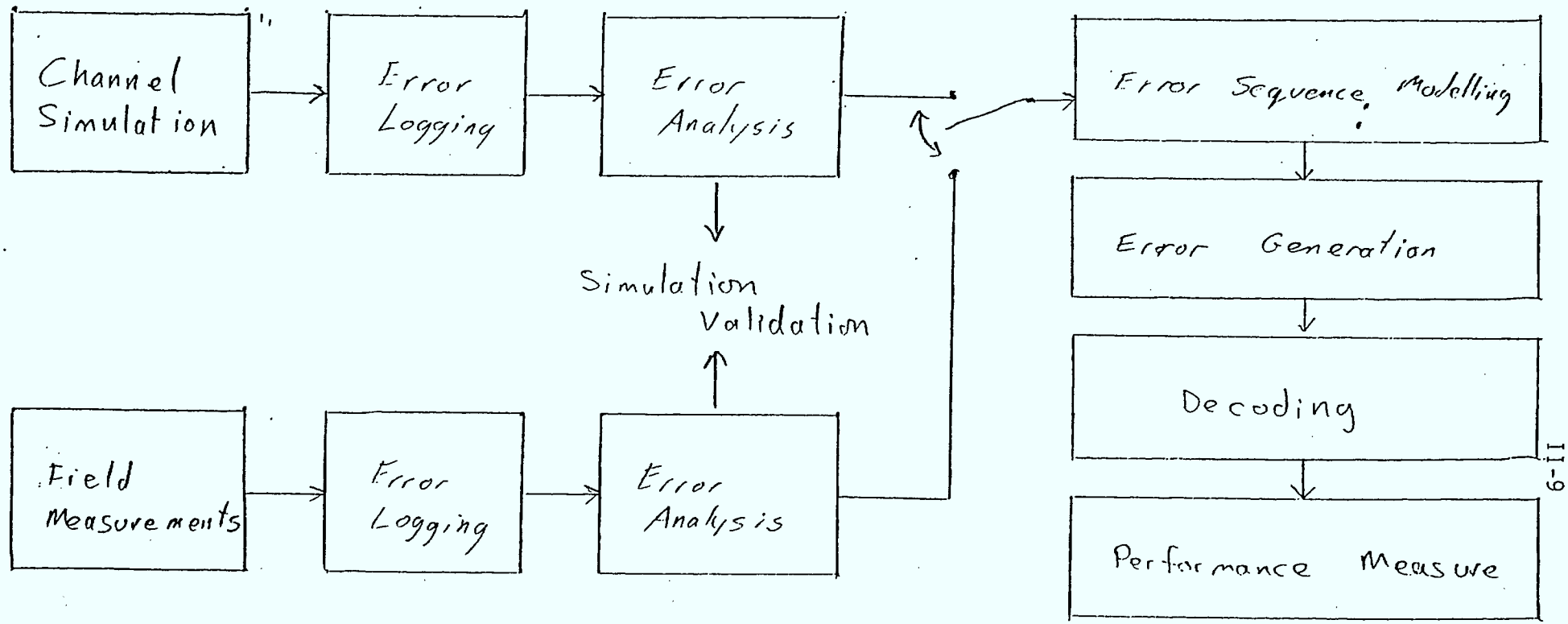


Figure 2: Block Diagram of Performance Measurement Procedures

5.0 DECODING

The steps involved in code performance determination are summarized in Figure 2. The only major portion of Figure 2 not yet covered by this memo is the actual process of decoding. Note that encoding is not required because all the codes are linear; hence, the same "encoded" packets can be used with every error sequence.

Decoding can be greatly simplified by essentially decoding only those bytes in error. This can be done because all of the codes involve modulo 2 8-tuple (byte) sums:

$$S_k = \sum_{i=0}^{27} B_i \gamma_{ki}, \qquad (3)$$

where S_k is the kth "syndrome",

B_i is the ith code byte

γ_{ki} is a weighting, dependent on the code.

Expanding (3):

$$S_k = \sum_{i=0}^{27} (C_i + E_i)\gamma_{ki}$$

$$= \sum_{i=0}^{27} C_i \gamma_{ki} + \sum_{i=0}^{27} E_i \gamma_{ki} \qquad \text{(linear code)}$$

$$S_k = 0 + \sum_{i,\ E_i \neq 0} E_i \gamma_{ki}, \qquad (4)$$

where C_i is the error free ith byte,

E_i is the ith byte error pattern.

The same C_i are used for each packet. Finding S_k can be done by a subroutine, shared by all codes. As input, this routine would be given the i's and corresponding E_i's, and the appropriate weighting vector $\underline{\gamma}_k$.

Decoding can also be simplified by recognizing simple, known correctable error patterns. For instance, some codes can correct any single byte error in the 28 byte data block (packet); therefore, if only one byte contains errors, the simulated decoder would simply remove all these errors.

Before examining each individual code, the arithmetic operations in a Galois Field (GF) are briefly treated.

5.1 Calculations in GF(128)

All the codes require solving algebraic equations in GF(128). The elements of the field are related by what is called a "generator polynomial". For instance, if the elements are denoted α^i, $0 \leq i \leq 127$, and:

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
$\alpha^0 =$	0	0	0	0	0	0	0	1	
$\alpha^1 =$	0	0	0	0	0	0	1	0	
$\alpha^2 =$	0	0	0	0	0	1	0	0	
$\alpha^3 =$	0	0	0	0	1	0	0	0	(5)
$\alpha^4 =$	0	0	0	1	0	0	0	0	
$\alpha^5 =$	0	0	1	0	0	0	0	0	
$\alpha^6 =$	0	1	0	0	0	0	0	0	

For a generator polynomial of $x^7 + x^3 + x^0 = 0$,

$$\alpha^7 = \alpha^3 + \alpha^0$$

$$= 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1.$$

Note that only 7 bits would be used if we were to continue to find all the α^is in this way. The extra bit is used for byte parity. The polynomial can be expanded:

$$(x+1)(x^7 + x^3 + x^0) = 0$$

$$x^8 + x^7 + x^4 + x^3 + x + 1 = 0 \qquad (6)$$

in order to produce an 8 bit, odd parity cyclic code. The corresponding even parity code can be obtained by adding the generator root $(x^7 + x^3 + x^0) \rightarrow 211_8$ to the odd parity byte. Appendix III lists the code bytes for (5) with (6), which are the α^i of the Carleton codes. Appendix IV lists the α^i for the SAB codes. These α^i use the same generator (6), but a different definition of α^i, $0 < j < 6$.

When calculating the S_k, we must find the products $E_i \gamma_{ki}$. This may be rewritten as:

$$E_i \gamma_{ki} = \alpha^{\ell} \alpha^{m}$$

$$= \alpha^{\ell + m}$$

Therefore, multiplication (division) in GF(128) can be accomplished by adding (subtracting) in modulo 127 the "log" of the multiplier and multiplicand (dividend and divisor) and then taking the "antilog" of this sum (difference). The log and antilog operations are best performed by using a look-up table. Note that knowledge of

the parity of E_i is lost when log (E_i) is taken. The special cases of the all zero byte (in even parity, 211_8 in odd) can be termed $\alpha^{-\alpha}$, where $\alpha^{-\alpha}\alpha^i = \alpha^i$, V_i.

5.2 Product Code

The Product code is simply a row-column parity check. It is capable of correcting one error in the 28 byte data block, and detecting any odd number of errors/byte or any error pattern confined to one byte. Odd parity on each byte is used to detect the odd errors in a byte.

If B_{27} is the byte following the prefix and B_0 being the last suffix, then for the product code [7]

$$B_0 \triangleq \sum_{i=1}^{27} B_i,$$

$$S_1 \triangleq \sum_{i=0}^{27} R_i, \quad (\text{i.e. } \gamma_1^i = \alpha^0, V_i) \qquad (7)$$

where R_i is the i-th received byte.

Let $W(p)$ = the number of parity errors detected,

$W(S_1)$ = the weight of S_1, then decoding is as follows:

(1) If $W(p) = 0 \wedge W(S_1) = 0$ accept packet as error free.

(2) If $W(p) = 1 \wedge W(S_1) = 1$ correct the single error by adding (XOR) S_1 to the byte which produced the parity error. Declare the packet "corrected".

(3) If not (1) or (2), declare "failure" (error detected) and reject packet.

5.2.1 High Level Computer Implementation

Product decoding involves finding byte parity errors and S_1. Since in the simulation we know which bytes contain errors, only these bytes need be checked for parity. Although byte parity is a simple function of the number of bits set in a byte, a far simplier approach is to use a 256 element look-up table. A count of parity errors should be maintained as each byte is checked so that as soon as 2 parity errors are found, the packet is rejected.

S_1 can be found as defined by (4) and (7).

5.3 Carleton 1 Byte

Each byte in the C_1 code is encoded to have odd parity. The suffix byte, B_0 is encoded as [8]:

$$B_o \triangleq \sum_{i=1}^{27} B_i \, \alpha^{8i} ,$$

where α^i is given in Appendix III.

Therefore,

$$S_2 = \sum_{i=0}^{27} R_i \, \alpha^{8i} = \alpha^{\ell} \qquad (8)$$

The guaranteed correction/detection capabilities of C_1 are the same as the product code. Decoding is as follows:

(1) If $W(p) = 0 \wedge W(S_2) = 0$ accept packet as error free.

(2) If $W(p) = 1 \wedge W(S_2) \neq 0$, then find $u = \log S_2$. Denote the first bit after the prefix as b_{224} (and the last bit in the suffix as b_1). Now if u or 127+u matches a bit number in the byte having the parity error, then correct that bit.

(3) If not (1) or (2) reject packet.

5.3.1 Implementation

Byte parity checks can be performed in the same way as with the Product code. S_2 can be found by (8) and (4) where the $E_i\alpha^{8i}$ products are found using log, antilog look-up tables. As pointed out in 5.1, parity information is lost in the log operation. Therefore, the parity of E_i must be stored before finding the $E_i\alpha^{8i}$ product, so that this parity ;may be used in the antilog operation.

5.4 Carleton 2 Byte Code

The C_2 code is a combination of the C_1 and product codes. It is capable of correcting any single byte error and any 2 byte erasures (parity errors). The packet is encoded such that B_0 meets the product code requirements (7), and B_1 is choosen such that B_0 also meets the C_1 (8) requirements. This is accomplished as folows:

Let $$P_p = \sum_{i=2}^{27} B_i,$$

$$P_c = \sum_{i=2}^{27} B_i \alpha^{8i},$$

then we wish B_1 B_0 such that:

$$P_c + B_1 \alpha^8 = B_0 \qquad (9)$$

$$P_p + B_1 = B_0 \qquad (10)$$

Solving for B_0 gives:

$$B_0 = P_C \alpha^6 + P_p \alpha^{14}$$

Finding S_1, S_2 and parity checks is done exactly as for the product and C_1 code. The decoder decisions are [7]:

(1) If $W(S_1) = W(S_2) = W(p) = 0$ accept packet as error free.

(2) If $W(S_1) \neq 0$, $W(S_2) \neq 0$ and $W(p) = 0$, then there is a possibility of a single byte error. If the only error is E_j then,

$$S_1 = E_j$$

$$S_2 = E_j \alpha^{8i}$$

$$K = \log(S_2/S_1)$$

$$= 8i \Big|_{\text{mod } 127}$$

If K is a multiple of 8 and $0 \leq k \leq 216$, then correct byte K/8 by XORing it with S_1 and then accept the packet. Otherwise reject the packet. Note that because $K=\text{Log}(S_2/S_1)$ is modulo 127, K and 127+K must checked. For example:

$S_1 = \alpha^{46} = 35_8$, say j= 25, then

$S_2 = \alpha^{46} \alpha^{200}$

$= \alpha^{46} \alpha^{73}$

$= \alpha^{119} = 353_8$

$S_2/S_1 = \alpha^{73}$

$\log(S_2/S_1) = 73 = K$

$K + 127 = 200$

K is not devisible by 8, but K+127 is.

The recovered $j = (K + 127)/8 = 25$ as expected.

(3) If $W(p) = 1$, $W(S_1) \neq 0$ and $W(S_2) \neq 0$, then it is possible that the one byte, B_j, having the parity error is in error by S_1. As before

$S_1 = E_j$

$S_2 = E_j \alpha^{8i}$

except now we suspect we known j; therefore, if

$S_1 \alpha^{8i} = S_2$

correct B_j by XORing it with S_1, otherwise reject the packet.

(4) If $W(p) = 2$, $W(S_1) \neq 0$, and $W(S_2) \neq 0$ then it is possible that only the two bytes with parity errors, B_i, B_j are in error.

It follows that:

$S_1 = E_i + E_j$

$S_2 = E_i \alpha^{8i} + E_j \alpha^{8j}$

Solving for E_j:

$$E_j = \frac{(S_1\alpha^{8i} + S_2)}{\alpha^{8i} + \alpha^{8j}} \qquad (11)$$

In terms of the log, exp (antilog) look-up tables, let

$$\ell = [127-\log(\exp)8i) + \exp(8j)]_{\text{mod } 127}$$

$$K = \log [\exp(\log(S_1) + 8i) + S_2]$$

Then

$E_j = \exp(k + \ell)$ in odd parity representation.

E_i is then found by:

$E_i = S_1 + E_j$.

Bytes B_i, B_j are "corrected" by adding E_i, E_j respectively. The packet is then accepted.

(5) Any other cases result in packet rejection.

5.5 SAB 2 Byte Code

The 2 byte SAB (Seguin, Allard, Bhargava) code has the same guaranteed correction capabilities as the C_2 code. It is defined in even byte parity, with the even α's listed in Appendix IV [9].

Encoding can be done by:

$$\underline{c}^T = \underline{m}^T G,$$

where

$\underline{c}$ is the coded 28 byte block,

$\underline{m}$ is the uncoded (but with parity) 26 byte message,

G is given in Figure 3.

For the received block, $\underline{r}$, $\underline{r} = \underline{c}$ + errors, the decoder syndromes are given as:

$$S_1 = \sum_{i=0}^{27} r_i \alpha^i$$

$$S_2 = \sum_{i=0}^{27} r_i \alpha^{2i} \qquad (12)$$

Interpreting S_1, S_2 for decoding is clearly explained in [9]. The decoding steps given in [9] are copied in Appendix V for reference. Solutions to the resulting algebraic expression can be done in the same way as in (11), using log and exp look-up tables.

5.6 SAB 3 Byte Code

The 3 byte SAB code guarantees correction of any one of the following:

- any single byte error
- any 2 byte error provided at least 1 of the errors is detectable (parity error).
- any 3 byte error, provided all 3 errors are detectable.

Figure 3: The G Matrix for SAR_2 [9]

The generator matrix G in systematic form for this code is as follows.

$$G = \left[\begin{array}{cccccccccc}
\alpha^{3} & \alpha^{32} & 1 & 0 & 0 & 0 & \cdots & & 0 & 0 \\
\alpha^{35} & \alpha^{105} & 0 & 1 & 0 & 0 & \cdots & & 0 & 0 \\
\alpha^{108} & \alpha^{96} & 0 & 0 & 1 & 0 & \cdots & & 0 & 0 \\
\alpha^{99} & \alpha^{55} & & & & & & & & \\
\alpha^{58} & \alpha^{115} & & & & & & & & \\
\alpha^{118} & \alpha^{105} & & & & & & & & \\
\alpha^{108} & \alpha^{97} & & & & & & & & \\
\alpha^{100} & \alpha^{21} & & & & & & & & \\
\alpha^{24} & \alpha^{15} & & & & & & & & \\
\alpha^{18} & \alpha^{13} & & & & & & & & \\
\alpha^{16} & \alpha^{8} & & & & & & & & \\
\alpha^{11} & \alpha^{72} & & & & & & & & \\
\alpha^{75} & \alpha^{115} & & & & & & & & \\
\alpha^{116} & \alpha^{46} & & & & & & & & \\
\alpha^{49} & \alpha^{99} & & & & & & & & \\
\alpha^{102} & \alpha^{54} & & & & & & & & \\
\alpha^{57} & \alpha^{74} & & & & & & & & \\
\alpha^{77} & \alpha^{93} & & & & & & & & \\
\alpha^{96} & \alpha^{62} & & & & & & & & \\
\alpha^{65} & \alpha^{29} & & & & & & & & \\
\alpha^{32} & \alpha^{58} & & & & & & & & \\
\alpha^{61} & \alpha^{107} & & & & & & & & \\
\alpha^{110} & \alpha^{48} & & & & & & & & \\
\alpha^{51} & \alpha^{79} & & & & & & & & \\
\alpha^{82} & \alpha^{49} & 0 & 0 & & & \cdots & & 1 & 0 \\
\alpha^{52} & \alpha^{112} & & & & & & & 0 & 1 \\
 & & & & & & & & 0 & 0
\end{array}\right] \qquad (2.3.2)$$

Encoding can be done as with the SAB_2 code, except m is only 25 bytes, and G is given in Figure 4. The syndromes are then defined as:

$$S_1 = \sum_{i=0}^{27} r_i \alpha^i$$

$$S_2 = \sum_{i=0}^{27} r_i \alpha^{2i} \qquad (13)$$

$$S_3 = \sum_{i=0}^{27} r_i \alpha^{3i}$$

Again, the decoding procedure is taken from [9] and copied in Appendix VI.

5.7 Bundles

A bundle of a bundle code consists of 14 data blocks (packets). Thirteen of these are identical to packets having a 2 byte code. The 14th packet contains 2 byte suffixes for packets formed by interleaving the bytes of the previous 13 data blocks. (see Figure 5). Suffix bytes in the 14th block are calculated on the basis of the 14th block alone.

To encode the 14th block, let

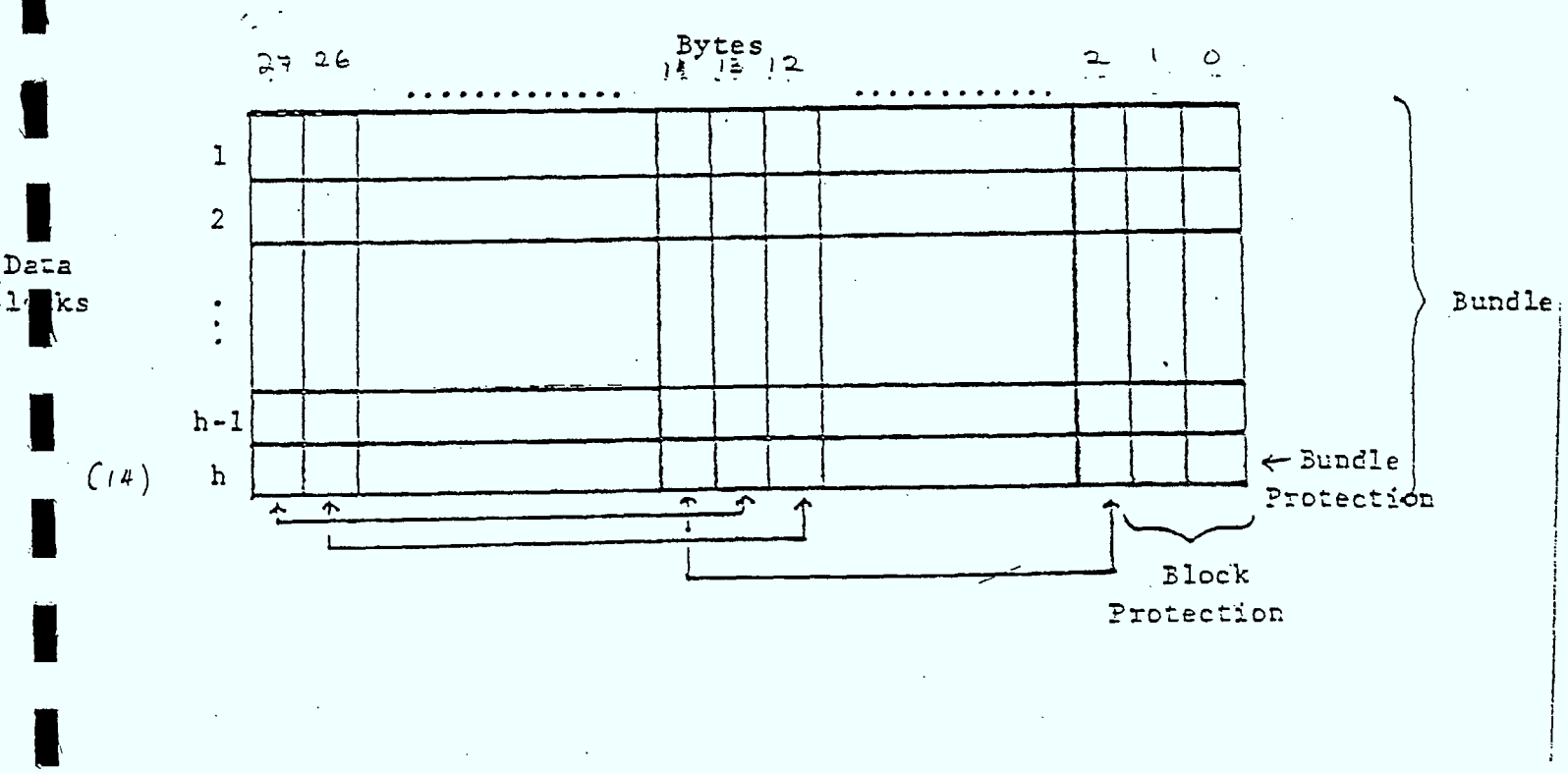
$$C_k = (B_{K,14}, B_{K+13,14}, B_{K,1}, B_{K,2} \ldots B_{K,13}, B_{K+13,1}, B_{K+13,2} \ldots\ldots B_{K+13,13})$$

where $B_{K,i}$ is the kth byte of the ith block.

Figure 4: The G Matrix for SAB_3 [9]

$$G = \begin{bmatrix} \alpha^{6} & \alpha^{106} & \alpha^{104} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \alpha^{110} & \alpha^{1} & \alpha^{40} & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \alpha^{46} & \alpha^{32} & \alpha^{116} & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \alpha^{122} & \alpha^{82} & \alpha^{7} & & & & & & & \\ \alpha^{13} & \alpha^{30} & \alpha^{56} & & & & & & & \\ \alpha^{62} & \alpha^{81} & \alpha^{37} & & & & & & & \\ \alpha^{43} & \alpha^{21} & \alpha^{79} & & & & & & & \\ \alpha^{85} & \alpha^{106} & \alpha^{123} & & & & & & & \\ \alpha^{2} & \alpha^{27} & \alpha^{114} & & & & & & & \\ \alpha^{120} & \alpha^{15} & \alpha^{106} & & & & & & & \\ \alpha^{112} & \alpha^{76} & \alpha^{37} & & & & & & & \\ \alpha^{43} & \alpha^{113} & \alpha^{16} & & & & & & & \\ \alpha^{22} & \alpha^{107} & \alpha^{116} & & & & & & & \\ \alpha^{122} & \alpha^{75} & \alpha^{99} & & & & & & & \\ \alpha^{105} & \alpha^{87} & \alpha^{106} & & & & & & & \\ \alpha^{112} & \alpha^{32} & \alpha^{80} & & & & & & & \\ \alpha^{86} & \alpha^{5} & \alpha^{118} & & & & & & & \\ \alpha^{124} & \alpha^{120} & \alpha^{105} & & & & & & & \\ \alpha^{111} & \alpha^{105} & \alpha^{40} & & & & & & & \\ \alpha^{46} & \alpha^{102} & \alpha^{35} & & & & & & & \\ \alpha^{41} & \alpha^{114} & \alpha^{112} & & & & & & & \\ \alpha^{118} & \alpha^{16} & \alpha^{101} & & & & & & & \\ \alpha^{107} & \alpha^{38} & \alpha^{72} & & & & & & & \\ \alpha^{78} & \alpha^{75} & \alpha^{72} & & & & & \cdots & 1 & 0 \\ \alpha^{78} & \alpha^{41} & \alpha^{104} & 0 & 0 & & & \cdots & 0 & 1 \end{bmatrix}$$

Figure 5: A Bundle



The C_2 (or possibly the SAB_2) encoding strategy is then used on C_k to defind $B_{k,14}$ and $B_{k+13,14}$. That is, for the C_2 code, select $B_{k,14}$, and $B_{k+13,14}$ such that:

$$\sum_{i=0}^{27} C_{ki} = 0, \quad \text{and}$$

$$\sum_{i=0}^{27} C_{ki}\alpha^{8i} = 0$$

The same 2 byte code is then applied to $B_{i,14}$ to find $B_{1,14}$ and $B_{0,14}$.

Decoding procedes as follows:

(1) Perform the 2 byte block decoding on each of the 14 blocks.

(2) If exactly 1 block (packet) is rejected (or missing), replace that block and accept the bundle.

(3) If more than 1 packet is missing, reject the bundle.

(4) If no packets are missing, then perform the "vertical" (interleaved) block decoding for each of the 13 vertical blocks. Note, this is performed on the blocks after "horizontal" correction. If there are any vertical rejections, reject the bundle, otherwise accept it.

Note that a "missing" packet is one with a prefix error, as opposed to one having uncorrectable detectable errors. The decoding steps are identical as those used for the 2 byte code, excepting block replacement.

Replacement can be accomplished by setting all bytes in the missing line to $\alpha^{-\alpha}$, with the parity such that the error is detected (0 for the odd parity codes, 211_8 for the even parity codes). The vertical blocks are then decoded as before.

5.8 Five Codes in One

A bundle encoded with the C_2 code is in fact encoded in:

(1) byte parity,
(2) product code,
(3) C_1 code,
(4) C_2 code, and
(5) C_2 bundle code.

Because of this, all five codes can be used on one encoded bundle, permitting the performance of all these codes to be determined in parallel. Note also that byte parity, S_1, and S_2 must be found for the C_2 bundle anyway, and so determining the performance of the other codes is a small increment to the work load.

5.9 Hamming Decoding

An uncorrectable error in the 5 byte, Hamming (8,4) encoded prefix represents a packet rejection to line codes and a missed packet to a bundle code. Instead of encoding and decoding the first 5 bytes, a single error count/byte can be used to determine the success of the prefix. If there is more than 1 bit/byte in error, in the prefix, then the prefix has failed.

5.10 Overall Implementation Considerations

In summary the decoding operations to be performed can be carried out by:

(1) Use only the error positions, and using a standard (implied) encoded block and bundle.

(2) Syndromes can be calculated in terms of the errors only, as the error free portion contributes 0 to the syndrome.

(3) Hamming (8,4) performance can be judged by counting errors/byte instead of decoding.

(4) The five codes listed in 5.8 can be "decoded" in parallel because each lower order code is a subset of the higher order codes.

(5) Parity checks and "multiplication", "exponentiation", and the "logarithm" operations should be performed by using look-up tables. The look-up tables required are listed in Table 1.

Table 1: Look-up Tables

Function	# of Words	Input	Output
Carleton Log	258	α^i odd parity	i, $0 \leq i \leq 128$, $128 \rightarrow -\infty$
		α^i even parity	$i+129$, $0 \leq i \leq 128$, $128 \rightarrow -\infty$
Carleton exp	129	i, $0 \leq i \leq 128$, $128 \rightarrow -\infty$	α^i in odd parity (add 211_8 for even parity)
SAB Log	258	α^i odd parity	i, $0 \leq i \leq 128$, $128 \rightarrow -\infty$
		α^i even parity	$i + 129$
SAB exp	129	i, $0 \leq i \leq 128$ $128 \rightarrow -\infty$	α^i in odd parity

774 Words total, for both codes.

	α^I ODD PARITY									α^I EVEN PARITY								
I	7	6	5	4	3	2	1	0	OCTAL	7	6	5	4	3	2	1	0	OCTAL
0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	210
1	0	0	0	0	0	0	1	0	2	1	0	0	0	1	0	1	1	213
2	0	0	0	0	0	1	0	0	4	1	0	0	0	1	1	0	1	215
3	0	0	0	0	1	0	0	0	10	1	0	0	0	0	0	0	1	201
4	0	0	0	1	0	0	0	0	20	1	0	0	1	1	0	0	1	231
5	0	0	1	0	0	0	0	0	40	1	0	1	0	1	0	0	1	251
6	0	1	0	0	0	0	0	0	100	1	1	0	0	1	0	0	1	311
7	1	0	0	0	0	0	0	0	200	0	0	0	0	1	0	0	1	11
8	1	0	0	1	1	0	1	1	233	0	0	0	1	0	0	1	0	22
9	1	0	1	0	1	1	0	1	255	0	0	1	0	0	1	0	0	44
10	1	1	0	0	0	0	0	1	301	0	1	0	0	1	0	0	0	110
11	0	0	0	1	1	0	0	1	31	1	0	0	1	0	0	0	0	220
12	0	0	1	1	0	0	1	0	62	1	0	1	1	1	0	1	1	273
13	0	1	1	0	0	1	0	0	144	1	1	1	0	1	1	0	1	355
14	1	1	0	0	1	0	0	0	310	0	1	0	0	0	0	0	1	101
15	0	0	0	0	1	0	1	1	13	1	0	0	0	0	0	1	0	202
16	0	0	0	1	0	1	1	0	26	1	0	0	1	1	1	1	1	237
17	0	0	1	0	1	1	0	0	54	1	0	1	0	0	1	0	1	245
18	0	1	0	1	1	0	0	0	130	1	1	0	1	0	0	0	1	321
19	1	0	1	1	0	0	0	0	260	0	0	1	1	1	0	0	1	71
20	1	1	1	1	1	0	1	1	373	0	1	1	1	0	0	1	0	162
21	0	1	1	0	1	1	0	1	155	1	1	1	0	0	1	0	0	344
22	1	1	0	1	1	0	1	0	332	0	1	0	1	0	0	1	1	123
23	0	0	1	0	1	1	1	1	57	1	0	1	0	0	1	1	0	246
24	0	1	0	1	1	1	1	0	136	1	1	0	1	0	1	1	1	327
25	1	0	1	1	1	1	0	0	274	0	0	1	1	0	1	0	1	65
26	1	1	1	0	0	0	1	1	343	0	1	1	0	1	0	1	0	152
27	0	1	0	1	1	1	0	1	135	1	1	0	1	0	1	0	0	324
28	1	0	1	1	1	0	1	0	272	0	0	1	1	0	0	1	1	63
29	1	1	1	0	1	1	1	1	357	0	1	1	0	0	1	1	0	146
30	0	1	0	0	0	1	0	1	105	1	1	0	0	1	1	0	0	314
31	1	0	0	0	1	0	1	0	212	0	0	0	0	0	0	1	1	3
32	1	0	0	0	1	1	1	1	217	0	0	0	0	0	1	1	0	6
33	1	0	0	0	0	1	0	1	205	0	0	0	0	1	1	0	0	14
34	1	0	0	1	0	0	0	1	221	0	0	0	1	1	0	0	0	30
35	1	0	1	1	1	0	0	1	271	0	0	1	1	0	0	0	0	60
36	1	1	1	0	1	0	0	1	351	0	1	1	0	0	0	0	0	140
37	0	1	0	0	1	0	0	1	111	1	1	0	0	0	0	0	0	300
38	1	0	0	1	0	0	1	0	222	0	0	0	1	1	0	1	1	33
39	1	0	1	1	1	1	1	1	277	0	0	1	1	0	1	1	0	66
40	1	1	1	0	0	1	0	1	345	0	1	1	0	1	1	0	0	154
41	0	1	0	1	0	0	0	1	121	1	1	0	1	1	0	0	0	330
42	1	0	1	0	0	0	1	0	242	0	0	1	0	1	0	1	1	53
43	1	1	0	1	1	1	1	1	337	0	1	0	1	0	1	1	0	126
44	0	0	1	0	0	1	0	1	45	1	0	1	0	1	1	0	0	254
45	0	1	0	0	1	0	1	0	112	1	1	0	0	0	0	1	1	303
46	1	0	0	1	0	1	0	0	224	0	0	0	1	1	1	0	1	35
47	1	0	1	1	0	0	1	1	263	0	0	1	1	1	0	1	0	72
48	1	1	1	1	1	1	0	1	375	0	1	1	1	0	1	0	0	164
49	0	1	1	0	0	0	0	1	141	1	1	1	0	1	0	0	0	350

50	1	1	0	0	0	0	1	0	302	0	1	0	0	1	0	1	1	113
51	0	0	0	1	1	1	1	1	37	1	0	0	1	0	1	1	0	226
52	0	0	1	1	1	1	1	0	76	1	0	1	1	0	1	1	1	267
53	0	1	1	1	1	1	0	0	174	1	1	1	1	0	1	0	1	365
54	1	1	1	1	1	0	0	0	370	0	1	1	1	0	0	0	1	161
55	0	1	1	0	1	0	1	1	153	1	1	1	0	0	0	1	0	342
56	1	1	0	1	0	1	1	0	326	0	1	0	1	1	1	1	1	137
57	0	0	1	1	0	1	1	1	67	1	0	1	1	1	1	1	0	276
58	0	1	1	0	1	1	1	0	156	1	1	1	0	0	1	1	1	347
59	1	1	0	1	1	1	0	0	334	0	1	0	1	0	1	0	1	125
60	0	0	1	0	0	0	1	1	43	1	0	1	0	1	0	1	0	252
61	0	1	0	0	0	1	1	0	106	1	1	0	0	1	1	1	1	317
62	1	0	0	0	1	1	0	0	214	0	0	0	0	0	1	0	1	5
63	1	0	0	0	0	0	1	1	203	0	0	0	0	1	0	1	0	12
64	1	0	0	1	1	1	0	1	235	0	0	0	1	0	1	0	0	24
65	1	0	1	0	0	0	0	1	241	0	0	1	0	1	0	0	0	50
66	1	1	0	1	1	0	0	1	331	0	1	0	1	0	0	0	0	120
67	0	0	1	0	1	0	0	1	51	1	0	1	0	0	0	0	0	240
68	0	1	0	1	0	0	1	0	122	1	1	0	1	1	0	1	1	333
69	1	0	1	0	0	1	0	0	244	0	0	1	0	1	1	0	1	55
70	1	1	0	1	0	0	1	1	323	0	1	0	1	1	0	1	0	132
71	0	0	1	1	1	1	0	1	75	1	0	1	1	0	1	0	0	264
72	0	1	1	1	1	0	1	0	172	1	1	1	1	0	0	1	1	363
73	1	1	1	1	0	1	0	0	364	0	1	1	1	1	1	0	1	175
74	0	1	1	1	0	0	1	1	163	1	1	1	1	1	0	1	0	372
75	1	1	1	0	0	1	1	0	346	0	1	1	0	1	1	1	1	157
76	0	1	0	1	0	1	1	1	127	1	1	0	1	1	1	1	0	336
77	1	0	1	0	1	1	1	0	256	0	0	1	0	0	1	1	1	47
78	1	1	0	0	0	1	1	1	307	0	1	0	0	1	1	1	0	116
79	0	0	0	1	0	1	0	1	25	1	0	0	1	1	1	0	0	234
80	0	0	1	0	1	0	1	0	52	1	0	1	0	0	0	1	1	243
81	0	1	0	1	0	1	0	0	124	1	1	0	1	1	1	0	1	335
82	1	0	1	0	1	0	0	0	250	0	0	1	0	0	0	0	1	41
83	1	1	0	0	1	0	1	1	313	0	1	0	0	0	0	1	0	102
84	0	0	0	0	1	1	0	1	15	1	0	0	0	0	1	0	0	204
85	0	0	0	1	1	0	1	0	32	1	0	0	1	0	0	1	1	223
86	0	0	1	1	0	1	0	0	64	1	0	1	1	1	1	0	1	275
87	0	1	1	0	1	0	0	0	150	1	1	1	0	0	0	0	1	341
88	1	1	0	1	0	0	0	0	320	0	1	0	1	1	0	0	1	131
89	0	0	1	1	1	0	1	1	73	1	0	1	1	0	0	1	0	262
90	0	1	1	1	0	1	1	0	166	1	1	1	1	1	1	1	1	377
91	1	1	1	0	1	1	0	0	354	0	1	1	0	0	1	0	1	145
92	0	1	0	0	0	0	1	1	103	1	1	0	0	1	0	1	0	312
93	1	0	0	0	0	1	1	0	206	0	0	0	0	1	1	1	1	17
94	1	0	0	1	0	1	1	1	227	0	0	0	1	1	1	1	0	36
95	1	0	1	1	0	1	0	1	265	0	0	1	1	1	1	0	0	74
96	1	1	1	1	0	0	0	1	361	0	1	1	1	1	0	0	0	170
97	0	1	1	1	1	0	0	1	171	1	1	1	1	0	0	0	0	360
98	1	1	1	1	0	0	1	0	362	0	1	1	1	1	0	1	1	173
99	0	1	1	1	1	1	1	1	177	1	1	1	1	0	1	1	0	366
100	1	1	1	1	1	1	1	0	376	0	1	1	1	0	1	1	1	167
101	0	1	1	0	0	1	1	1	147	1	1	1	0	1	1	1	0	356
102	1	1	0	0	1	1	1	0	316	0	1	0	0	0	1	1	1	107
103	0	0	0	0	0	1	1	1	7	1	0	0	0	1	1	1	0	216

104	0	0	0	0	1	1	1	0	16	1	0	0	0	0	1	1	1	207
105	0	0	0	1	1	1	0	0	34	1	0	0	1	0	1	0	1	225
106	0	0	1	1	1	0	0	0	70	1	0	1	1	0	0	0	1	261
107	0	1	1	1	0	0	0	0	160	1	1	1	1	1	0	0	1	371
108	1	1	1	0	0	0	0	0	340	0	1	1	0	1	0	0	1	151
109	0	1	0	1	1	0	1	1	133	1	1	0	1	0	0	1	0	322
110	1	0	1	1	0	1	1	0	266	0	0	1	1	1	1	1	1	77
111	1	1	1	1	0	1	1	1	367	0	1	1	1	1	1	1	0	176
112	0	1	1	1	0	1	0	1	165	1	1	1	1	1	1	0	0	374
113	1	1	1	0	1	0	1	0	352	0	1	1	0	0	0	1	1	143
114	0	1	0	0	1	1	1	1	117	1	1	0	0	0	1	1	0	306
115	1	0	0	1	1	1	1	0	236	0	0	0	1	0	1	1	1	27
116	1	0	1	0	0	1	1	1	247	0	0	1	0	1	1	1	0	56
117	1	1	0	1	0	1	0	1	325	0	1	0	1	1	1	0	0	134
118	0	0	1	1	0	0	0	1	61	1	0	1	1	1	0	0	0	270
119	0	1	1	0	0	0	1	0	142	1	1	1	0	1	0	1	1	353
120	1	1	0	0	0	1	0	0	304	0	1	0	0	1	1	0	1	115
121	0	0	0	1	0	0	1	1	23	1	0	0	1	1	0	1	0	232
122	0	0	1	0	0	1	1	0	46	1	0	1	0	1	1	1	1	257
123	0	1	0	0	1	1	0	0	114	1	1	0	0	0	1	0	1	305
124	1	0	0	1	1	0	0	0	230	0	0	0	1	0	0	0	1	21
125	1	0	1	0	1	0	1	1	253	0	0	1	0	0	0	1	0	42
126	1	1	0	0	1	1	0	1	315	0	1	0	0	0	1	0	0	104
127	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	210

APPENDIX IV

CODE WORDS FOR SAB CODES [6]

Table of Elements of $GF(2^7)$

In this appendix, a table of 'i' with the binary 8-tuple representation of α^i for the basis chosen in Equation (2.2.1) is given, where α is a primitive element of $GF(2^7)$. We have used the recursion $\alpha^7 = 1 + \alpha^3$.

0	10000001	40	01101100
1	10000010	41	11010001
2	10000100	42	00101011
3	10001000	43	01010110
4	10010000	44	10100101
5	10100000	45	11001010
6	11000000	46	00011101
7	00001001	47	00111010
8	00010010	48	01110100
9	00100100	49	11100001
10	01001000	50	01001011
11	10011001	51	10011111
12	10110010	52	10111110
13	11100100	53	11111100
14	01000001	54	01110001
15	10001011	55	11101011
16	10010110	56	01011111
17	10101100	57	10110111
18	11011000	58	11101110
19	00111001	59	01010101
20	01110010	60	10100011
21	11101101	61	11000110
22	01010011	62	00000101
23	10101111	63	00001010
24	11011110	64	00010100
25	00110101	65	00101000
26	01101010	66	01010000
27	11011101	67	10101001
28	00110011	68	11010010
29	01100110	69	00101101
30	11000101	70	01011010
31	00000011	71	10111101
32	00000110	72	11111010
33	00001100	73	01111101
34	00011000	74	11110011
35	00110000	75	01101111
36	01100000	76	11010111
37	11001001	77	00100111
38	00011011	78	01001110
39	00110110	79	10010101

80	10101010	104	10001110
81	11010100	105	10011100
82	00100001	106	10111000
83	01000010	107	11110000
84	10001101	108	01101001
85	10011010	109	11011011
86	10110100	110	00111111
87	11101000	111	01111110
88	01011001	112	11110101
89	10111011	113	01100011
90	11110110	114	11001111
91	01100101	115	00010111
92	11000011	116	00101110
93	00001111	117	01011100
94	00011110	118	10110001
95	00111100	119	11100010
96	01111000	120	01001101
97	11111001	121	10010011
98	01111011	122	10100110
99	11111111	123	11001100
100	01110111	124	00010001
101	11100111	125	00100010
102	01000111	126	01000100
103	10000111		

It can be observed that all the binary 8-tuples listed above have even weight and along with the all 0 8-tuple, these 8-tuples form a vector space of dimension 7.

APPENDIX V

DECODING THE SAB_2 CODE [6]

3.1 A Closed Form Decoding Algorithm for d Equal to 3

For the encoding scheme given in section (2.3), the minimum distance of the code is 3 and, therefore, it can correct t errors and s erasures if

$$2t + s < 3 \tag{3.1.1}$$

It is clear from Equation (3.1.1) that the received vector is correctly decoded to the transmitted code word, iff

(a) $s \leq 2$, $t = 0$, i.e. a maximum of two erasures and no error takes place.

(b) $t \leq 1$, $s = 0$, i.e. a maximum of one error and no erasure takes place.

Thus, the following decoding procedure is viable for such an encoding scheme.

(i) Two Detectable Errors in the Received Bytes

Let the error polynomial be

$$e(X) = e_i X^i + e_j X^j \tag{3.1.2}$$

where $e_i, e_j \in GF(2^7)$ and $1 \leq i < j$ and let $r(X)$ be the received polynomial, i.e.

$$r(X) = c(X) + e(X) \tag{3.1.3}$$

where $c(X)$ is the transmitted code vector. Then the two syndromes S_1 and S_2 are given by

$$\begin{aligned} S_1 &= r(\alpha) = e_i\alpha^i + e_j\alpha^j \\ S_2 &= r(\alpha^2) = e_i\alpha^{2i} + e_j\alpha^{2j} \end{aligned} \tag{3.1.4}$$

and the values of i and j are known from the parity bits associated

with the bytes in positions i+1 and j+1. We then have

$$(e_i \quad e_j) \begin{bmatrix} \alpha^i & \alpha^{2i} \\ \alpha^j & \alpha^{2j} \end{bmatrix} = (S_1 \quad S_2) \tag{3.1.5}$$

The latter matrix has an inverse which is

$$\frac{1}{\Delta} \begin{bmatrix} \alpha^{2j} & \alpha^{2i} \\ \alpha^j & \alpha^i \end{bmatrix} \tag{3.1.6}$$

where $\Delta = \alpha^{i+2j} + \alpha^{2i+j}$.

Multiplying both sides of Equation (3.1.5) by (3.1.6), we obtain

$$e_i = \frac{S_1\alpha^{2j} + S_2\alpha^j}{\Delta} \qquad \text{in odd parity} \tag{3.1.7}$$

$$e_j = \frac{S_1\alpha^{2i} + S_2\alpha^i}{\Delta} \qquad \text{in odd parity} \tag{3.1.8}$$

Hence the errors in positions i and j are determined by Equations (3.1.7) and (3.1.8) respectively.

(ii) A Single Detectable Error in the Received Bytes

Suppose there is a detectable error in position i, then the error polynomial is given by

$$e(X) = e_i X^i$$

where $e_i \in GF(2^7)$ and the syndromes S_1 and S_2 are

$$\begin{aligned} S_1 &= r(\alpha) = e_i\alpha^i \\ S_2 &= r(\alpha^2) = e_i\alpha^{2i} \end{aligned} \tag{3.1.9}$$

The value of i is known. Thus,

$$e_i = \alpha^{-i} S_1. \qquad \text{in odd parity} \tag{3.1.10}$$

Also it can be observed that

$$S_2 = \alpha^i S_1. \tag{3.1.11}$$

The Equation (3.1.10) can be used to calculate the value of the error at position i.

(iii) A Single Byte in Error

If there is a single byte in error, then the error polynomial is given by

$$e(X) = e_i X^i$$

where $e_i \in GF(2^7)$ and i is not known.

The two syndromes S_1 and S_2 are calculated as

$$\begin{aligned} S_1 &= r(\alpha) = e_i \alpha^i \\ S_2 &= r(\alpha^2) = e_i \alpha^{2i} \end{aligned} \tag{3.1.12}$$

Solving Equation (3.1.12) for the value of e_i and i, we get

$$\begin{aligned} e_i &= S_1^2 S_2^{-1} \qquad \text{in even parity} \\ \alpha^i &= S_2 S_1^{-1}. \end{aligned} \tag{3.1.13}$$

A look up table of (i, α^i) can be used to determine i.

Hence, the decoding algorithm for the coding scheme presented in section (2.3) can be described as follows

Step I : Check the parity of each of the received bytes.

If they all check, go to II

If exactly one does not check, go to III

If exactly two do not check, go to IV

If more than two do not check, declare a decoding failure.

Proceed to next frame.

Step II : Represent all the bytes as elements of $GF(2^7)$ and compute the syndromes $S_1 = r(\alpha)$ and $S_2 = r(\alpha^2)$.

If both are 0, assume $r(X)$ is error free.

If both are nonzero, assume a single symbol in error and decode it using the procedure (iii) described above.

If exactly one is 0, declare a decoding failure. Proceed to next frame.

Step III: Compute the syndromes. Check for Equation (3.1.11). If it is satisfied, decode it using procedure (ii) given above.

If Equation (3.1.11) is not satisfied, declare a decoding failure. Proceed to next frame.

Step IV : Assume exactly two symbols in error and correct them using procedure (i) described above. Proceed to next frame.

The above decoding algorithm ensures that a decoded code word has bytes that have even parity only. The complete decoding algorithm is given by the flowchart in Figure (3.1).

3.2 A Closed Form Decoding Algorithm for d Equal to 4

The encoding scheme and the generator matrix for d equal to 4 are given in section (3.4). It can correct t errors and s erasures if

$$2t + s < 4 \tag{3.2.1}$$

APPENDIX VI

PROCEDURE TO DECODE SAB_3 [6]

3.2 A Closed Form Decoding Algorithm for d Equal to 4

The encoding scheme and the generator matrix for d equal to 4 are given in section (3.4). It can correct t errors and s erasures if

$$2t + s < 4 \tag{3.2.1}$$

Thus a received vector is correctly decoded to the transmitted code word if one of the following combinations of errors and erasures takes place

(a) No erasure, no error $(2t + s = 0)$

(b) One erasure, no error $(2t + s = 1)$

(c) Two erasures, no error $(2t + s = 2)$

(d) Three erasures, no error $(2t + s = 3)$

(e) One error, no erasure $(2t + s = 2)$

(f) One error, one erasure $(2t + s = 3)$.

A simplified decoding procedure described below is viable for such a coding scheme

(i) A Single Detectable Error in the Received Bytes

Let there be a detectable error in position i, then the error polynomial is

$$e(X) = e_i X^i$$

where $e_i \in GF(2^7)$. The syndromes S_1 and S_2 and S_3 are given by

$$\begin{aligned} S_1 &= r(\alpha) = e_i \alpha^i \\ S_2 &= r(\alpha^2) = e_i \alpha^{2i} \\ S_3 &= r(\alpha^3) = e_i \alpha^{3i} \end{aligned} \tag{3.2.2}$$

Since the value of i is known, solving Equation (3.2.2) for e_i, we get

$$e_i = \alpha^{-i} S_1. \qquad \text{in odd parity} \tag{3.2.3}$$

Also we have to perform a check that there is no error in the received bytes. This can be done by noting that

$$S_2 = e_i\alpha^{2i} = \alpha^{-i} S_1 \alpha^{2i} = S_1\alpha^{i} \tag{3.2.4}$$

and

$$S_3 = e_i\alpha^{3i} = \alpha^{-i} S_1 \alpha^{3i} = S_1\alpha^{2i}. \tag{3.2.5}$$

Combining Equations (3.2.4) and (3.2.5), we get

$$S_3S_1 = S_2^2 \tag{3.2.6}$$

Hence if there is only one detectable error, then

$$e_i = \alpha^{-i}S_1$$

and

$$S_3S_1 = S_2^2.$$

(ii) Two Detectable Errors in the Received Bytes

Let there be detectable errors in positions i and j. The procedure for finding the error values at these positions is given in section (3.1). Additionally, we have to check if there are any other errors in the received bytes. It can be done by observing that

$$S_3 = e_i\alpha^{3i} + e_j\alpha^{3j} \tag{3.2.7}$$

Substituting for e_i and e_j from Equations (3.1.7) and (3.1.8) respectively, we get

$$S_3 = \alpha^{(i+j)}S_1 + (\alpha^{i} + \alpha^{j})S_2$$

Thus if there are only two detectable errors, then

$$e_i = \frac{S_1\alpha^{2j} + S_2\alpha^{j}}{\Delta}$$ in odd parity.

$$A^{-1} = \frac{1}{\Delta}\begin{bmatrix} \alpha^{2j+3k}+\alpha^{3j+2k} & \alpha^{j+3k}+\alpha^{3j+k} & \alpha^{j+2k}+\alpha^{2j+k} \\ \alpha^{2i+3k}+\alpha^{3i+2k} & \alpha^{i+3k}+\alpha^{3i+k} & \alpha^{i+2k}+\alpha^{2i+k} \\ \alpha^{2i+3j}+\alpha^{3i+2j} & \alpha^{i+3j}+\alpha^{3i+j} & \alpha^{i+2j}+\alpha^{2i+j} \end{bmatrix} \tag{3.2.11}$$

where

$$\Delta = \alpha^{i+j+k}[\alpha^{2i}(\alpha^{j}+\alpha^{k}) + \alpha^{2j}(\alpha^{i}+\alpha^{k}) + \alpha^{2k}(\alpha^{i}+\alpha^{j})].$$

Multiply both sides of Equation (3.2.10) by A^{-1}, we get

all in odd parity

$$e_i = \frac{S_1(\alpha^{2j+3k}+\alpha^{3j+2k}) + S_2(\alpha^{j+3k}+\alpha^{3j+k}) + S_3(\alpha^{j+2k}+\alpha^{2j+k})}{\Delta}$$

$$e_j = \frac{S_1(\alpha^{2i+3k}+\alpha^{3i+2k}) + S_2(\alpha^{i+3k}+\alpha^{3i+k}) + S_3(\alpha^{i+2k}+\alpha^{2i+k})}{\Delta}$$

$$e_k = \frac{S_1(\alpha^{2i+3j}+\alpha^{3i+2j}) + S_2(\alpha^{i+3j}+\alpha^{3i+j}) + S_3(\alpha^{i+2j}+\alpha^{2i+j})}{\Delta} \tag{3.2.12}$$

Hence the erasure values are determined from the Equation (3.2.12).

(iv) One Error

The procedure for finding the error magnitude and its position is given in section (3.1). Also, we have to check if there is only one error in the received bytes. This can be done by observing that

$$S_3 = r(\alpha^3) = e_i\alpha^{3i}$$

e_i has even parity

from which it follows that

$$S_1S_3 = S_2^2 \tag{3.2.13}$$

(v) One Detectable and One Undetectable Error in the Received Bytes

Let there be a detectable error in position i and an undetectable error in position j. The error polynomial, therefore, is given by

$$e(X) = e_i X^i + e_j X^j$$

e_i has odd parity
e_j has even parity

where $e_i, e_j \in GF(2^7)$ and i is known.

The syndromes S_1, S_2 and S_3 are then calculated as

$$\begin{aligned} S_1 &= r(\alpha) = e_i\alpha^i + e_j\alpha^j \\ S_2 &= r(\alpha^2) = e_i\alpha^{2i} + e_j\alpha^{2j} \\ S_3 &= r(\alpha^3) = e_i\alpha^{3i} + e_j\alpha^{3j} \end{aligned} \tag{3.2.14}$$

The value of j is calculated from the above equation and is given by

$$\alpha^j = (S_3 + \alpha^i S_2)(S_2 + \alpha^i S_1)^{-1}. \tag{3.2.15}$$

Once the location of error j is determined, the magnitudes e_i and e_j can be calculated as in part (ii).

Hence the decoding algorithm for the coding scheme described in chapter 2 for minimum distance d equal to 4 can be stated as follows.

Step I Check the parity of each of the received bytes.

If all parity bits check, go to II.

If exactly one does not check, go to III.

If exactly two do not check, go to IV.

If exactly three do not check, go to V.

If more than three do not check, declare a decoding failure. Proceed to next frame.

Step II Compute the syndromes

$S_1 = r(\alpha)$, $S_2 = r(\alpha^2)$ and $S_3 = r(\alpha^3)$.

If all three of S_1, S_2 and S_3 are zero, declare $r(X)$ as error free.

If all of S_1, S_2 and S_3 are nonzero, assume a single error and decode it using the procedure (iv) given above. Also check if $S_3S_1 = S_2^2$. If not, declare a decoding failure.

If some of S_1, S_2 and S_3 are zero, declare a decoding failure.

Go to step VI.

Step III Check if $S_3S_1 = S_2^2$. If yes, then go to III(a), else go to III(b).

III(a) Assume a single detectable error and correct it using procedure (i) given above.

Go to step VI.

III(b) Assume that an undetectable and a detectable error have occurred and decode it using procedure (v).

Go to step VI.

Step IV Check if $S_3 = \alpha^{i+j}S_1 + (\alpha^i + \alpha^j)S_2$.

If yes, then assume that exactly two detectable errors have occurred and use procedure (ii) to decode them.

Go to step VI.

If not, declare a decoding failure.

Go to step VI.

Step V Assume exactly three bytes in error and correct them using procedure (iii) given above.

Go to step VI.

Step VI Go to next frame.

Again, this decoding algorithm ensures that a decoded code word has bytes that have even parity only. The complete decoding algorithm is given by the flowchart in Figure (3.2).

MORELAND, K.W.
--Telidon system study

DATE DUE