DETERMINATION OF THE BAUD RATE OF AN FSK SIGNAL USING ADAPTIVE NOISE CANCELLING TECHNIQUES

by

MARC DAVID BRACK

B.S., Kansas State University, 1984

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1985

Approved by:

Donald R. Hummels

Major Professor

Acknowledgements

I would like to express my thanks to the members of my advisory committee, Dr. John E. Boyer, Dr. Stephen A. Dyer, and special thanks to Dr. Donald R. Hummels, my major professor.

I would also like to thank those who helped me during the course of my work. Those people are Bob Schneider and Phil Buckland who provided support of the computing facilities and Christy Schroller who helped with the manuscript preparation.

The financial support granted by Motorola, Inc., Government Electronics Group and by Kansas State University is also appreciated.

Finally, and most importantly, I would like to thank my wife, Lisa. Besides providing encouragement and support through the entire effort, she also constructed the diagrams of the manuscript and did the final proofreading.

Table of Contents

List of Figures

List of Fignras (cout.)

List of Tables

I. Introduction

Free spsce is s commonly used trsnamission medinm for communicating informstion. Informstion commnnicsted as such can not be kept confidential becsnse of the nstnre of the medium. Thns the tssk of snrveillsnce is estsbliahed; there will slways be someone thst wishes to receive confidential informstion intended for someone else. Msny times s surveillsnce receiver is not in the optimal looation for reception which results in low signsl-to-noise rstios. Also, the surveillsnce receiver can not expect to know s priori any psrsmetera of the trsnsmission. Fnrthermore, those trsnsmitting the informstion will try to keep it from getting to the wrong people. For these three ressons slone, surveillanoe receivera sre smong the most challenging receivers to design.

This resesrch effort is focnsed st determining the bsnd rste of s message trsnsmitted nsing frequency shift keying (FSK) modulation. The received aignsl is sssumed to be FSK. This sssnmption has been msde either by s modnlstion recognition algorithm or by some knowledgesble inaight. Instesd of relying on conventionsl methods which nse s zero-crossing algorithm, the eatimate is determined using sdaptive noise csncelling techniqnes. Figure 1 showa the two methods of a bsud rate snslysis system. The crnx of the problem is to evaluate the performance of the sdsptive noise cancelling techniqne in estimsting the bsud rste in a low aignsl-to-noise rstio environment.

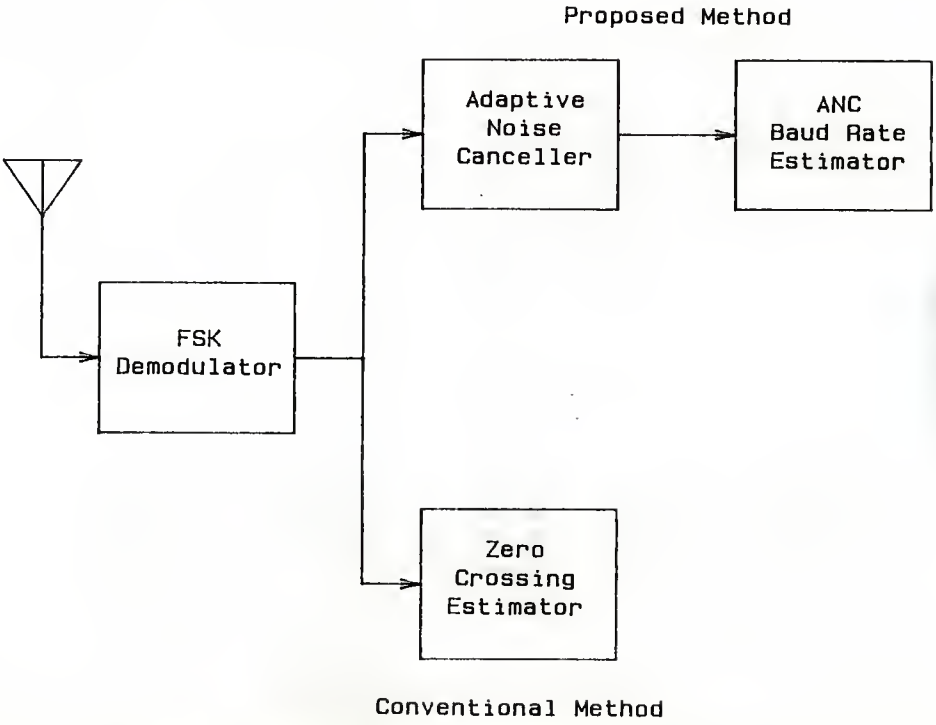


FIGURE 1: System for determining FSK baud rate.

Obviously some restrictions must be made since one receiver can not be expected to work for all signals. The first restriction is that the FSK signal bandwidth be no greater than 6000 Hz. This figure was arbitrarily chosen but can be changed to meet the needs of a specific application. The other restriction applies to the band rate. The FSK signal baudwidth, the frequency deviation of the FSK signal, and the baud rate are roughly related by

$$B_{FSK} \cong 2(\Delta f_{FSK} + R) \qquad (1.1)$$

In order to demodulate the signal with some degree of accuracy the frequency shift must be as large as the baud rate. This means that the maximum band rate for a given signal bandwidth is one-fourth the baudwidth. For this case the main lobes of the message spectrum are above and below the center frequency. To lower the probability of error during demodulation, the frequency deviation is generally larger than the baud rate. In order that the main lobe and the first side lobe of the message spectrum be above and below the center frequency upon modulation, the frequency deviation must be at least twice the baud rate. Based on this premise, the restriction for the maximum band rate is 1000 Hz given the 6000 Hz signal bandwidth.

Since the investigation will be done by comparing the performance of the proposed method against that of the conventional method, any benefits or harms that may result from the arbitrarily chosen restictions will affect both methods. Therefore, final judgment will not be influenced by these restrictions.

II. FSK Band Rate Analysis System

The system used to determine the band rate of an FSK signal consists of two major devices as shown in Figure 1. The first is an FSK demodulator, the second is the band rate estimator. The band rate estimate is determined either with adaptive noise cancelling techniques, the method under investigation, or with zero-crossing techniques, the conventional method. The FSK demodulator, illustrated in Figure 2, has a delay-line discriminator (DLD) as its principal component. Before the DLD is a bandpass filter to limit the input noise power, and following the DLD is a low-pass filter for the same purpose. The limiter is to stabilize the DLD response which is dependent on the input signal amplitude.

Bandpass Filter

The pre-detection filter is a 4-pole Butterworth bandpass filter. Its purpose is twofold. First, it limits the noise power that enters the system. Second, it allows only signals for which the receiver is designed to operate to enter the system. For this reason the bandwidth is set to 6000 Hz. Also, the filter should be centered on the carrier frequency of the signal which has been determined a priori.

Limiter

FSK is a special case of frequency modulation (FM) and the FSK demodulator can actually be considered as an FM demodulator. It makes sense then to eliminate any amplitude modulation resulting from the noise perturbations which might have entered

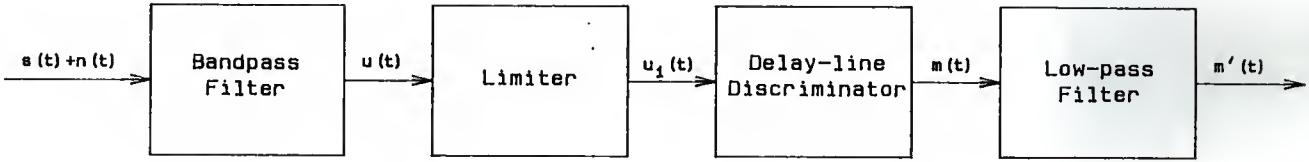


FIGURE 2: Block diagram of an FSK demodulator.

the signal. Furthermore, as will be discussed shortly, the output of the DLD for a continuous-wave input is dependent upon the squared amplitude of the input. Thus, if the input amplitude is left to vary, the DLD reponse will also vary.

The limiter is modeled as a device which will have an output of constant amplitude but will have identical phase properties as the input. That is, the limiter will block any amplitude modulation while frequency modulation is allowed to pass.

Delay-Line Discriminator

The DLD is a device whose output is proportional to the instantaneous frequency deviation of the input signal from its center frequency. A block diagram of the DLD is given in Figure 3. One use of the DLD is for FSK demodulation. If the input is an FSK signal the output will be the bit sequence of the binary message.

The effect of the DLD is apparent when the input consists of a noise-free unmodulated carrier, namely,

$$n_1(t) = A \cos[(2\pi f_c + 2\pi\Delta f)t] \qquad (2.1)$$

The frequency offset from the carrier frequency of the signal, f_c, is denoted by Δf. It can be shown that the output of the DLD is given by

$$m(t) = -4A^2 \sin(2\pi\Delta f T + 2\pi f_c T) \qquad (2.2)$$

where T is the time of delay between $n_1(t)$ and $n_2(t)$. Note this output is only for a continuous-wave input. This result plus the output of the DLD for a general input are developed with low-pass modeling techniques presented in the following section.

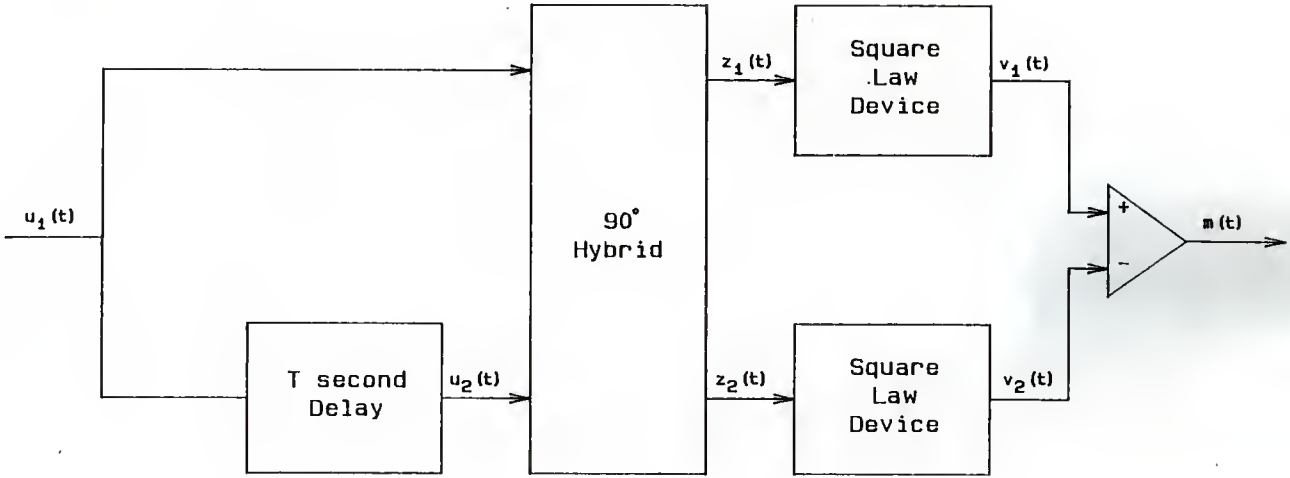


FIGURE 3: Block diagram of a delay-line discriminator.

In order for the discriminator to work properly, (i.e., produce an output proportional to instantaneous input frequency deviation) it is necessary that that the output be zero when the deviation is zero. For this to be, it is necessary that

$$2\pi f_c T = n\pi \qquad \text{for } n=1,2,\ldots \tag{2.3}$$

For this choice, the output is described by

$$m(t) = \pm 4A^2 \sin(2\pi \Delta f T) \tag{2.4}$$

The $\pm$ comes from the addition of $n\pi$, + for n odd and - for n even. A plot of one cycle of (2.4) is shown in Figure 4 when the delay T is chosen to be 125 μsec and n is even. Observe that the curve is nearly linear when the frequency deviation is small. T should be chosen so that the discriminator operates in this linear range. The obvious reason for the choice is that it is easier to work with linear or nearly linear devices as opposed to non-linear devices. Another reason is that the response of the DLD in the non-linear region for general inputs is not understood well. Since the pre-filter has limited the range of the incoming deviations, the bandwidth of the pre-filter dictates a specific delay. The procedure in setting T is now discussed.

The linear region is roughly that portion of the spectrum centered on f_c such that the argument of the sinusoid is less that $\pi/6$ in magnitude. Then the relation between the maximum delay and the maximum deviation is

$$2\pi \Delta f_{max} T_{max} = \frac{\pi}{6} \tag{2.5}$$

Recall that the pre-filter passes frequencies from $-\Delta f_{max}$ to Δf_{max}. Substituting the pre-filter bandwidth, B_{RF}, for twice the

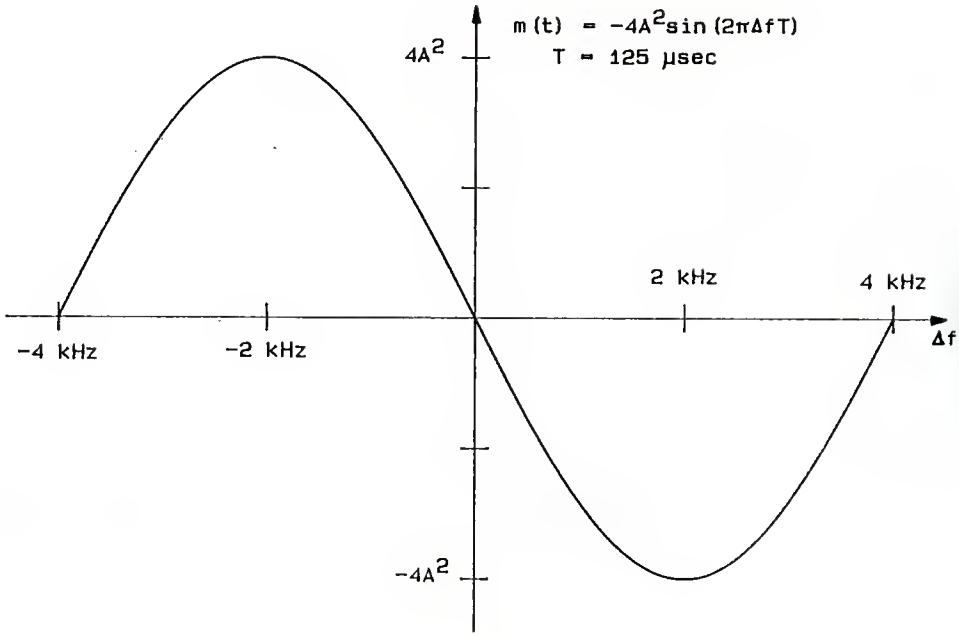


FIGURE 4: Comparator output of delay-line discriminator.

msximnm deviatiou sud then solving for the msximnm delsy results in

$$T_{max} = \frac{1}{6B_{RF}} \tag{2.6}$$

Since the product $2f_cT$ muat be sn iuteger by Equ. (2.3), T ahould be choseu as the largest vslne which satasfies Eqns. (2.3) aud (2.6).

Low-Psas Filter

The pnrpoae of the post-filter is to rednce the uoiae iu the output signal of the DLD. Since the ontpnt ia at baseband, the filter is low-pass. The filter implemented is s 4-pole Bntterworth filter. The bandwidth, to elimiuste ss much noise ss possible, ahonld be ouly alightly larger than the bandwidth of the output signal. However, selectiug the baudwidth is difficnlt siuce the sigusl bsndwidth is proportional to the baud rste. Aa stated iu the iutroductiou, the system shonld work for band rstes up to 1000 Hz. Thus the post-filter bandwidth is set to 1000 Hz.

Adsptive Noise Cauceller

The method of bsnd rste suslysis nnder study couaists of sn sdsptive noiae csuceller (heresfter referred to ss su ANC) snd the ANC baud rste estimator. The ANC bsud rate estimstor is an algorithm which evalnatea the parsmeters of the ANC to determiue sn estimste of the bsnd rste. The ANC implemented ia giveu iu Figure 5 and was iutroduced by Widrow [1]. The primary iupnt to the ANC is aome signal plns noise, $s+n_0$. The reference iuput, specified as n_1, is alao uoiae. It is nncorrelated with the

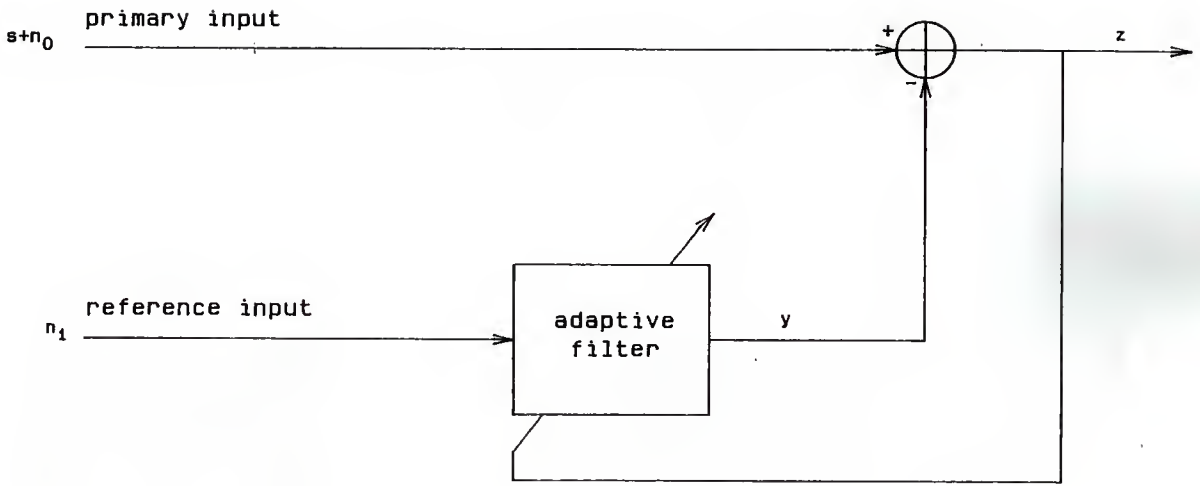


FIGURE 5: Adaptive noise canceller (ANC).

signel s bnt is correlated with n_0 in eome (poseibly) nnknown wsy. The reference input n_1 is filtered by the edaptive filter to prodnce y which is ae close e replica of n_0 es poesible. The outpnt z is the difference of the primsry input $e+n_0$ snd y.

The adsptive filter of the ANC differs from e conventionel filter in thet its impulse response is autometicelly adjnsted to produce the 'best result' which in this case is noise cencellation. The adjnstment, or edeptetion, ie accomplished thrangh an algorithm that responds to the ANC outpnt. Thns, the ANC will operete under changing conditions or in situetions in which the noise cen not be chsrecterized (eomething thet e conventionel noiee cencelling filter depende on). Widrow [1] provides a simple bnt convincing ergument to support this clsim.

The ANC outpnt, $z=s+n_0-y$, ie referred to ee the error of the system. The object of this syetem is to produce an outpnt z thet is the best fit in the leest equsres senee to the signal s. Therefore, the object is to minimize the meen sqnared error. It cen be shown that doing so does not rednce the signel power of the ontpnt but rather it reducee the sverege power of the qnentity n_0-y. This meens thet minimizing the error must msximize the edeptive filter's objective of treneforming n_1 into y. Fnrthermore, since the eignsl power remsins constant, minimizing the total outpnt power maximizes the outpnt signel-to-noise power rstio.

These argnmente cen be extended to cseee where the primsry inpnt and the reference inpnt do not consist of jnst signal plns

uoiae aud uoise, respectively. As loug as the refereuce iuput is correlated with part of the primary inpnt and nucorrelated with the rest, the two parta of the primary iupnt can be separated.

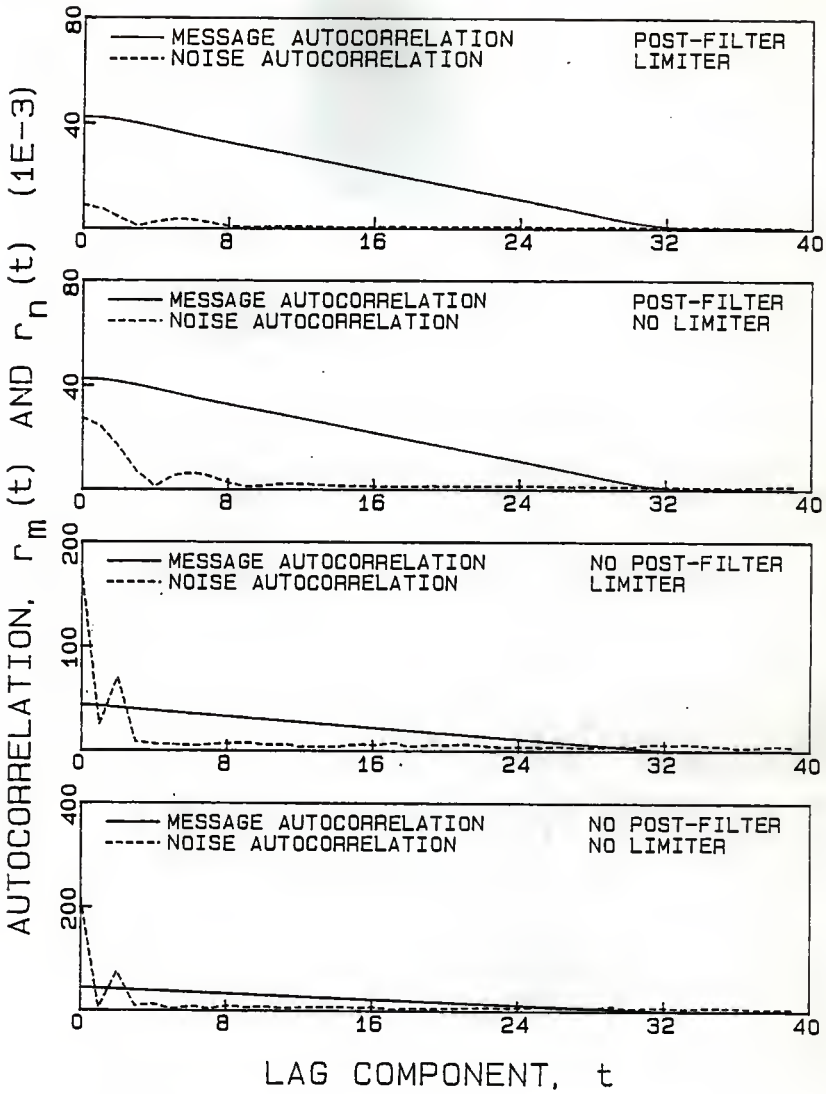
Baaed on this premise, a scheme was devised to use the ANC to separate the uoiae from the meaaage in the DLD outpnt. The derivation of the scheme reqnired iuformatiou about the correlatiou properties of the uoiae and message at the ontpnt of the DLD. Ratcliffe [2] treata thia anbject exteusively. Auother approach at obtaiuing thia informatiou is to use a simulatiou model aa deacribed iu the uext two aectiona. This was the method used to obtain the information.

First a uoise-free aignal consistiug of an FSK modulated, paeudo-raudom seqnence waa generated. Next thia sigual was paaaed throngh the FSK demodulator. The iupnt to the ANC was theu recorded. This iupnt is either the direct outpnt of the DLD or the filtered outpnt of the DLD (depending ou whether poatfilteriug was performed or uot). Iu either case, the inpnt is some variatiou of the binary measage bit sequence. To avoid ambiguities, the reault will be referred to aa the ANC iupnt. The message autocorrelation estimate was obtaiued from an ANC iuput which resnlted from a uoiae-free FSK sigual with the estimatiou being doue with FFT techuiquea. For the uoise autocorrelatiou estimate, aome definition of the uoise had to be made. The uoise at the ANC inpnt was arbitrarily defiued aa the differeuce of the iupnt resulting from a uoisy FSK aignal aud the iupnt resnltiug from a noise-free FSK sigual. The fiual uoise

entocorrelation eetimete wae prodnced by evereging ten individnel estimatee. This procednre was completed for severel predetection eignel-to-noise retioe and for the ceses of post-filtering end no poat-filtering.

Fignre 6 showe the measage entocorreletion plotted with the noiee entocorrelation for a SNR of 10 dB and with a post-filter bandwidth of 1000 Hz. Fignre 6 elso shows the eeme reenlte when the post-filter wae omitted. Since the fnnctions arc even, only the poeitive componente are ehown. Fnrthermore, the eetimetee are plotted for leg componente 0 to 40 since the reet of the fnnction conteina no edditional information. For different SNR's, the entocorrelation fnnctione differed only in e sceling of the emplitnde of the noiee antocorreletion. Note that in both the filtered end non-filtcred caees the messege ia correlated nntil component nnmber 32, which ie the nnmber of eemples per bit for this case (the hnmp paet component 32 in the filtered case ie censed by the correlation introdnced by the poet-filter). This reanlt ie intnitively obvions eince the seqnence ie psendo-rendom. The noiee entocorrelation for the filtered cese shows the noise ie precticelly nncorreleted peet component 20. For the non-filtered ceee the correletion extende to only the sixth component. Note that the poat-filter hes eliminated eome noise power as evidenced by the differencc in peek amplitndes of the noise antocorreletion for the two ceees.

Now enppoee the primery inpnt of the ANC is the ontpnt of the FSK demodnletor, i.e., both messege end noiee. The ANC will



R = 300 Hz
Δf = 300 Hz
SNR = 10 dB
f_s = 9600 Hz

FIGURE 6: Effect of post-filter and limiter on the correlation properties of the DLD output.

cancel one of theae two if the reference inpnt is correlated with one and ia uncorrelated with the other. The portion being cancelled will be the one that is oorrelated with the reference input. Notice that when the DLD outpnt without post-filtering is delayed by 8 aamples the noiae portion of delayed aequence is uncorrelated with the noise of the non-delayed sequence. Furthermore the meaaage portion of the delayed and non-delayed aeqnencea still remaina correlated. Hence the ANC will attempt to cancel the message in the DLD outpnt, i.e. npon adapting, the adaptive filter's ontpnt will be the message and the ontpnt error will be the noise. The weights of the adaptive filter will have converged once the noise of the error becomea stationary. In this mode, the ANC is being naed as a aelf-tuning filter. This ia illuatrated in Fignre 7.

Snppose the filtered ontput is considered. The delay would have to be 20 samplea instead of 6 for the noise to become uncorrelated. But note that a delay this long weakens the correlation between the aignal portion of the two inputs making the job of cancelling more difficnlt. Thna, omitting the post-filter for this method seema like a good idea especially aince determining the correct bandwidth is very difficnlt as mentioned earlier. One argnment for keeping the post filter is that the magnitnde of correlation of the noise is rednced, which wonld improve the results of the canoelling tasks. The initial approach taken ia to omit the post-filter when making baud rate estimatea with the ANC and to use a delay long enough ao the

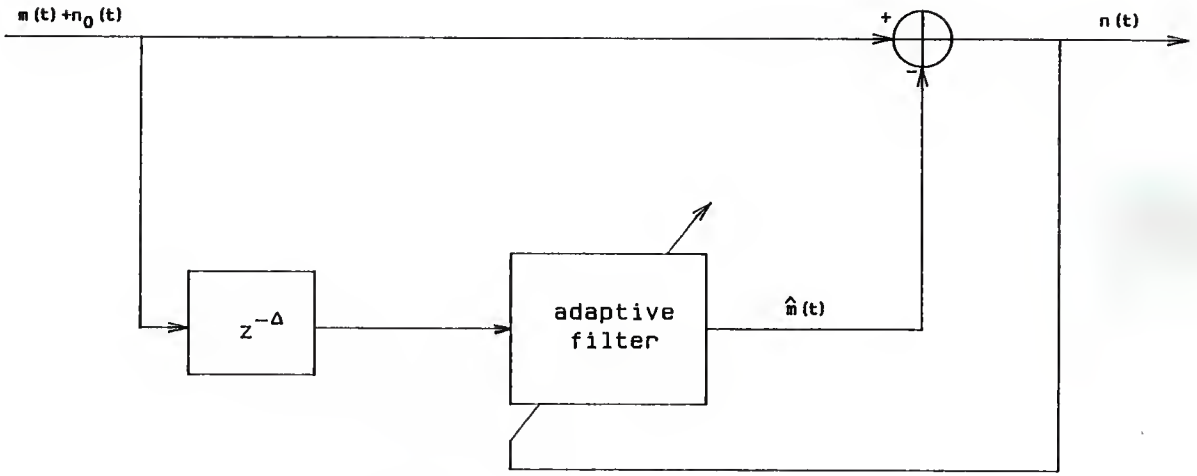


FIGURE 7: The adaptive noise canceller as a self-tuner.

noise is nncorrelated.

Zero-Croaaing Estimator

The propoaed method of baud rate analysia is to be measured againat the traditional method of determining the FSK band rate, that being the obaervation of the zero crossings of a demodulated FSK aignal. The concept is that a zero crossing in the meaaage shonld only occnr at intervala eqnal to the band rate or integral multiples of the band rate. Obvioualy noise will pertnrb the meaaage and thus the zero crossing might not be exactly in the proper place. Thna the algorithm ahonld make decisions based on aome criteria in order to obtain a baud rate estimate. This algorithm is deacribed in the next section.

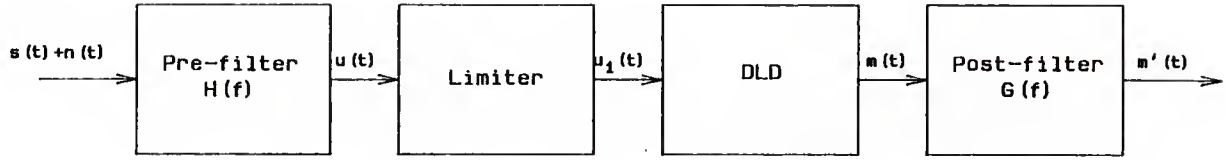
III. Development of a System Model

A msthemstics1 model of the system mnst be developed in order to simn1ste the system. Since the simn1ation is performed on a digita1 compnter, the implications of Shsnnon's ssmpling theorem mnst be kept in mind. Obvionsly, since the received FSK signs1 is s bsndpass signs1, ssmpling in order to prevent aliss-ing wonld reqnire vsst amonnts of memory for sny prsctics1 length of received signa1. Besides reqniring sn enormons memory cspa-city, the simn1ation wonld a1so demand mnch time to complete. Obvionsly the modeling csnnot be done efficiently nsing bandpsss representations. The so1ntion is to nse low-psss modeling tech-niqnes. The techniqne wi11 be spplied to the pre-filter, the limiter, and the DLD since they sre the only bsndpsss devices in the system. A11 other devices are baseband snd csn be modeled ss they are. Thns the FSK demodn1ator i11nstrsted sgsin in Fignre 8a is modeled by its low-paas eqnivs1ent representstion shown in Fignre 8b.

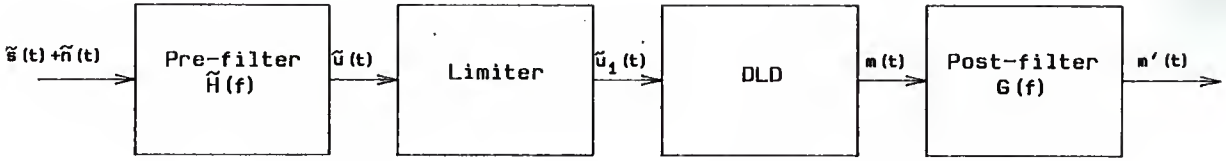
Low Psss Techniqnes

Low-psss modeling [3] re1ies on the property thst the per-tinent informstion of a bsnd-pass signs1 is contsined in its msgnitnde and in its phase. Both of these are s1ow1y varying in relation to the carrier freqnency and thns are essier to simn1ate with s digits1 compnter. The relation between s bandpass signs1 s(t) snd its eqnivs1ent low-pass connterpart or complex envelope $\tilde{s}(t)$ is, by definition,

$$s(t) = \mathrm{Re}\{\tilde{s}(t)e^{j2\pi f_c t}\} \qquad (3.1)$$



a. Bandpass system.



b. Low-pass equivalent representation.

FIGURE 8: FSK demodulator.

If the complex envelope is expressed in terms of its magnitude and argument then it follows that

$$s(t) = \tilde{s}(t) \cos[2\pi f_c t + \phi(t)] \tag{3.2}$$

where $\tilde{s}(t)$ is the magnitude of $\tilde{s}(t)$ and $\phi(t)$ is the argument. Evidently $\tilde{s}(t)$ is a complex function with the property that its magnitude is the envelope of s(t) and its argument is the phase of s(t). All bandpass functions, both deterministic and random, can be put in this format.

Bandpass Pre-Filter

A bandpass filter with an impulse response h(t) has an equivalent low-pass impulse response $\tilde{h}(t)$ which by definition satisfies

$$h(t) = 2 \, \mathrm{Re}\{\tilde{h}(t) e^{j2\pi f_c t}\} \tag{3.3}$$

where f_c is the center frequency of the filter. An equivalent low-pass transfer function, H(f), can be derived from this definition [3]. It is

$$H(f) = \tilde{H}(f-f_c) + \tilde{H}^*(-f-f_c) \tag{3.4}$$

where H(f) and $\tilde{H}(f)$ are the Fourier transforms of h(t) and $\tilde{h}(t)$, respectively. Also, * represents conjugation. This relationship can be shifted in frequency to obtain the equivalent low-pass transfer function in terms of the actual bandpass transfer function

$$\tilde{H}(f) = [H(f+f_c)]_{\text{low-pass term}} \tag{3.5}$$

The low-pass model of the filter will have a bandwidth equal to one-half the RF bandwidth, i.e. $B_{lp} = B_{RF}/2$. All transform relations for linear systems are also valid for low-pass models,

e.g., for an input of X(f) and a transfer function H(f) the output Y(f) is

$$\tilde{Y}(f) = \tilde{H}(f)\tilde{X}(f) \qquad (3.6)$$

Limiter

The limiter model with input $\tilde{u}(t)$ has an output $\tilde{u}_1(t)$ which is related to the input by

$$\tilde{u}_1(t) = A\, e^{j \arg[\tilde{u}(t)]} \qquad (3.7)$$

where A is some constant amplitude.

Delay-Line Discriminator

The block diagram of the low-pass equivalent model of a delay-line discriminator is shown in Figure 9. The output of the delay block can be characterized in terms of its input by

$$\tilde{u}_2(t) = \tilde{u}_1(t-T)e^{-j2\pi f_c T} \qquad (3.8)$$

This can be verified by expanding the definition of the complex envelope of $u_2(t) = u_1(t-T)$.

Each output of the 90° hybrid is equal to the corresponding input minus a 90° phase shifted copy of the remaining input. In equations this can be stated as

$$\tilde{z}_1(t) = \tilde{u}_1(t) - j\,\tilde{u}_2(t) \qquad (3.9a)$$

$$\tilde{z}_2(t) = \tilde{u}_2(t) - j\,\tilde{u}_1(t) \qquad (3.9b)$$

The action of the square law detector is to produce an output proportional to the magnitude squared of the complex envelope of the input, i.e.

$$v_1(t) = \left|\tilde{z}_1(t)\right|^2 \qquad (3.10a)$$

$$v_2(t) = \left|\tilde{z}_1(t)\right|^2 \qquad (3.10b)$$

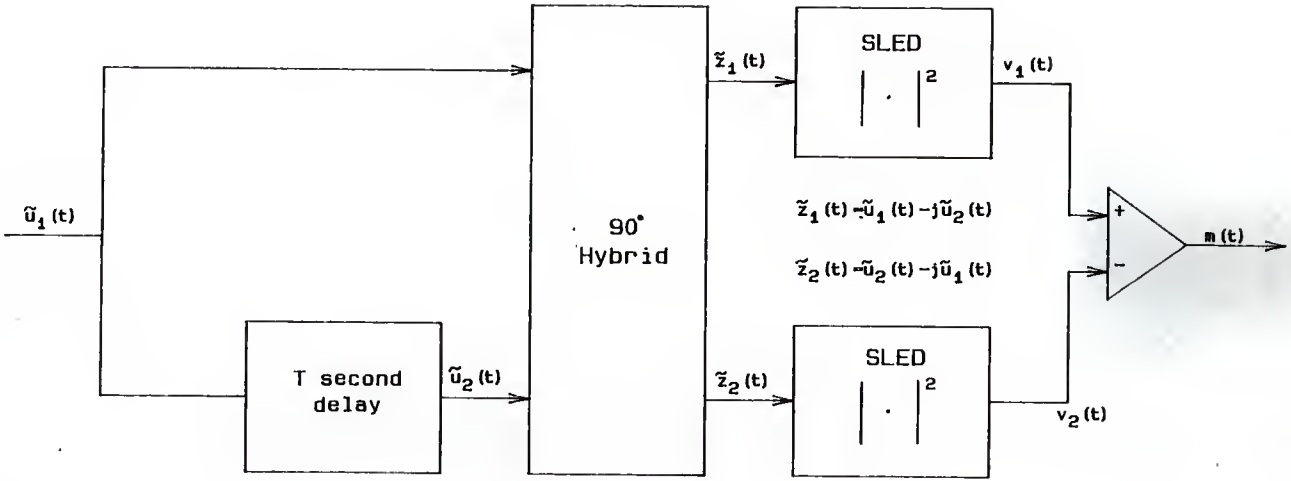


FIGURE 9: Low-pass equivalent model of a delay-line discriminator.

Note that the output is no longer a bandpass signal. Thus the low-pass model is the same as the physical system from this point on.

Finally the output of the DLD is simply the difference between the two square law detectors

$$m(t) = v_1(t) - v_2(t) \tag{3.11}$$

The discriminator equation is derived as follows. The outputs of the 90° hybrid in terms of the input $\tilde{u}_1(t)$ are

$$\tilde{z}_1(t) = \tilde{u}_1(t) - j\,\tilde{u}_1(t-T)\,e^{-j2\pi f_c T} \tag{3.12a}$$

$$\tilde{z}_2(t) = \tilde{u}_1(t-T)\,e^{-j2\pi f_c T} - j\,\tilde{u}_1(t) \tag{3.12b}$$

From these, the output of the first square law envelope detector is determined to be

$$v_1(t) = \left|\tilde{z}_1(t)\right|^2 = \tilde{z}_1(t)\tilde{z}_1^*(t) \tag{3.13a}$$

$$= [\tilde{u}_1(t) - j\tilde{u}_1(t-T)e^{-j2\pi f_c T}][\tilde{u}_1^*(t) + j\tilde{u}_1^*(t-T)e^{j2\pi f_c T}] \tag{3.13b}$$

$$= \left|\tilde{u}_1(t)\right|^2 + j\,\tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j2\pi f_c T} + \left|\tilde{u}_1(t-T)\right|^2 - j\,\tilde{u}_1^*(t)\tilde{u}_1(t-T)e^{-j2\pi f_c T} \tag{3.13c}$$

Similarly

$$v_2(t) = \left|\tilde{u}_1(t)\right|^2 - j\,\tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j2\pi f_c T} + \left|\tilde{u}_1(t-T)\right|^2 + j\,\tilde{u}_1^*(t)\tilde{u}_1(t-T)e^{-j2\pi f_c T} \tag{3.14}$$

The comparator output is then

$$m(t) = v_1(t) - v_2(t) \tag{3.15a}$$

$$= 2j\tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j2\pi f_c T} - 2j\tilde{u}_1^*(t)\tilde{u}_1(t-T)e^{-j2\pi f_c T} \tag{3.15b}$$

$$= 2[\tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j2\pi f_c T}e^{j\frac{\pi}{2}} + \tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j2\pi f_c T}e^{j\frac{\pi}{2}}] \quad (3.15c)$$

$$= 2(\tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j2\pi f_c T}e^{j\frac{\pi}{2}} + [\tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j2\pi f_c T}e^{j\frac{\pi}{2}}]^*) \quad (3.15d)$$

$$= 4\mathrm{Re}(\tilde{u}_1(t)\tilde{u}_1^*(t-T)e^{j(2\pi f_c T + \frac{\pi}{2})}) \quad (3.15e)$$

To observe the discriminator effect for s continuous-wave signsl, let the input be

$$u_1(t) = A\cos[2\pi(f_c + \Delta f)t] \quad (3.16)$$

then

$$\tilde{u}_1(t) = A e^{j2\pi\Delta f t} \quad (3.17)$$

Substituting into the output expreasion for a general input, (3.15e), gives

$$m(t) = 4\mathrm{Re}([Ae^{j2\pi\Delta f t}][Ae^{-j2\pi\Delta f(t-T)}]e^{j(2\pi f_c T + \frac{\pi}{2})}) \quad (3.18)$$

Simplifying yields

$$m(t) = 4A^2\cos[2\pi(\Delta f + f_c)T + \pi/2] \quad (3.19a)$$

$$= -4A^2\sin[2\pi(\Delta f + f_c)T] \quad (3.19b)$$

This is the result given and discussed in the DLD description of the previous section.

It is importsnt to note thst the input $\tilde{u}_1(t)$ will consist of two terms in the general csse, a signal envelope snd a noise envelope. Ratcliffe [2] discusaes the statiatics snd the power spectra associated with the general caae.

Low-psss Post-Filter

The post-filter, being lowpsas, csn be implemented directly into the system model. No equivalent model is necessary.

Adaptive Noise Canceller

The adaptive filter of the canceller is so named because the filter weights adjust to changing conditions. The implementation of the adaptive filter and the algorithm for the weight adjustment will now be discussed. The discussion is an abbreviated version of the one in Widrow [1].

The adaptive filter is modeled with the adaptive linear combiner shown in Figure 10. Note that X_j and W are vectors, while y_j and d_j are scalars. The vectors are defined as

$$X_j = [x_{0j}\ x_{1j}\ \ldots\ x_{nj}]^T \tag{3.20}$$

$$W = [w_0\ w_1\ \ldots\ w_n]^T \tag{3.21}$$

Generally x_{0j} is set to 1 so that w_0 can be considered the biasing weight. The output y_j is the inner product of X_j and W. The error ε_j is the difference between the desired response d_j and the actual response y_j. That is

$$y_j = X_j^T W = W^T X_j \tag{3.22}$$

$$\varepsilon_j = d_j - X_j^T = d_j - W^T X_j \tag{3.23}$$

Note that in accordance with the system description of the ANC, $d_j = x_{j+\Delta}$ where Δ is the delay in number of samples between the primary input and the reference input.

Since the desire is to minimize the mean squared error, the mean squared error is derived first. It is

$$E[\varepsilon_j^2] = E[d_j^2] - 2E[d_j X_j^T]W + W^T E[X_j X_j^T]W \tag{3.24}$$

where $E[\cdot]$ denotes expectation. The gradient of the error function, denoted by ∇, is obtained by differentiating this result with respect to W. The gradient is

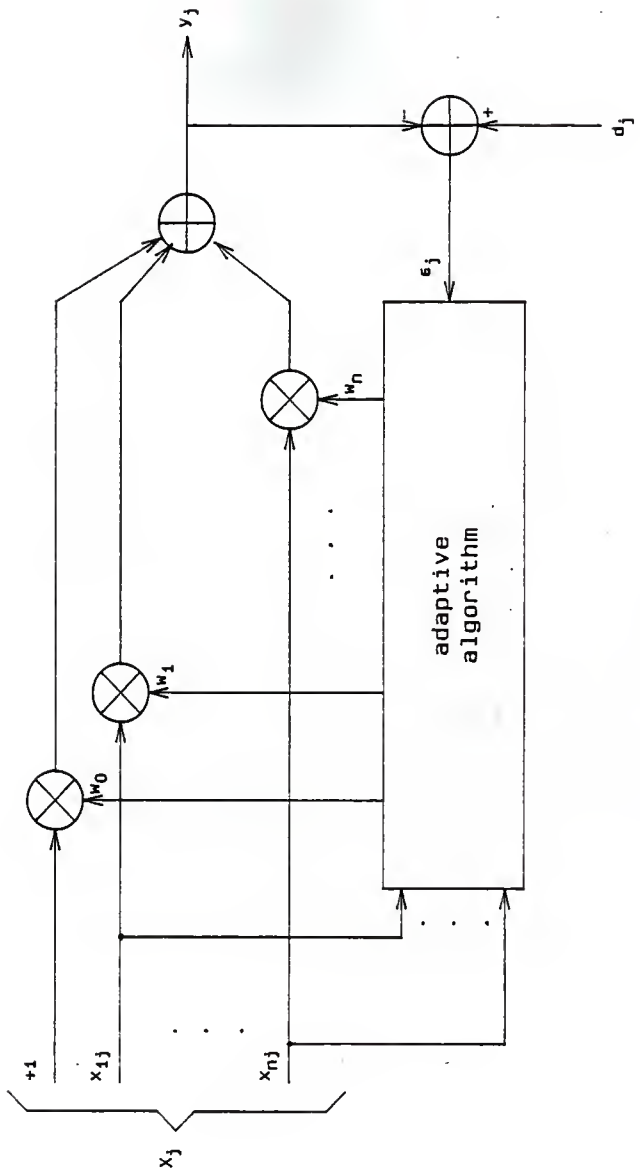


FIGURE 10: Adaptive linear combiner.

$$\nabla = \nabla\{E[e_j^2]\} = -2E[d_j X_j^T] + 2E[X_j X_j^T]W \tag{3.25}$$

The optimal weight vector W^* is obtained by setting the gradient to zero.

$$W^* = \{E[X_j X_j^T]\}^{-1} E[d_j X_j^T] \tag{3.26}$$

The idea behind using the gradient method is that the mean squared function is a quadratic function of the weights. It can be pictured as a concave hyperparaboloidal surface that never goes negative, i.e. a bowl resting on some point in the hyperspace. The object is to get to the bottom of the bowl, at which point the gradient is the zero vector.

The LMS algorithm implements the method of steepest descent. According to this method, the next weight vector estimate W_{j+1} is equal to the present weight vector estimate W_j plus a change proportional to the negative of the gradient; i.e.,

$$W_{j+1} = W_j - \mu\nabla_j \tag{3.27}$$

The parameter μ controls stability and convergence rate. The gradient ∇_j is the true gradient evaluated with the weight vector of the j^{th} iteration.

The LMS algorithm assumes ε_j^2 as an estimate of the mean squared error $E[e_j^2]$. The gradient is then taken of the estimate. This gradient, denoted by $\hat{\nabla}_j$, is an estimate of the true gradient and can be determined to be

$$\hat{\nabla}_j = \nabla[\varepsilon_j^2]_{W=W_j} = -2\varepsilon_j X_j \tag{3.28}$$

Using this estimate in place of the true gradient in (3.27) gives

$$W_{j+1} = W_j + 2\mu\varepsilon_j X_j \tag{3.29}$$

The adaptive filter of Figure 7 is implemented with a tapped

delsy line as shown in Figure 11. The inpnt aignal vector ia

$$X_j = [x_j \quad x_{j-1} \quad . \quad . \quad . \quad x_{j-n+1}]^T \tag{3.30}$$

The components of thia vector sre the delayed componenta of $x_{j+\Delta}$ which is a ssmple of the time seriea ..., $x_{j+\Delta-1}$, $x_{j+\Delta}$, $x_{j+\Delta+1}$, ...

When implementing this algorithm the parameter μ mnst be specified. If made too lsrge, the algorithm becomea unatable. If μ ia made too small, the weight vector tskea too much time to converge to the optimsl weight vector. Althongh there sre wsys to determine the proper μ to inanre stsbility, it is generally essier to nse a trial-and-error procednre in thia problem. Thia waa the manner nsed to set μ for faat convergence in the simulstion.

Beaidea convergence rate and stability, another concern in picking μ is imbedded in the LMS eqnstion, Eqn. (3.27). There sre two sources of informstion thst contribnte to npdate of the weight vector. One is the current weight vector which repreaents a history of the weight vectors dne to the regresaive atructure of Eqn. (3.27). The second ia the gradient which is estimated by the the prodnct of the current filter input vector and the error. The choice of μ decides what emphaaia ahonld be given to each. If μ is made small then the emphasia ia plsced on the cnrrent weight vector. In this inatance a weight vector with many weighta is useful ainoe it can retain more information. If μ is made large then the emphasis is placed on the gradient and the information contsined in previous weight vectors is soon loat.

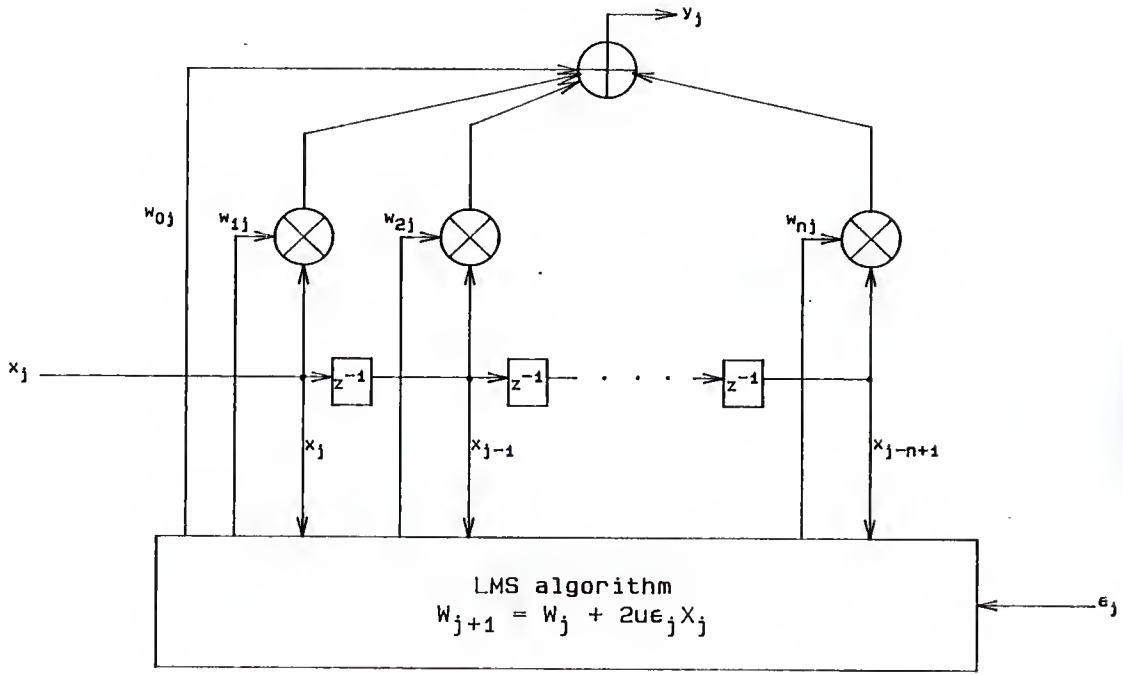


FIGURE 11: The LMS adaptive filter implemented with a tapped delay line.

In thie ceee e large nnmber of weights wonld not be beneficial.

For the band rate enalyeie eystem, both reaeons of choice have merit. Since the meeeage eeqnence is peendo-rendom, the mean of the ANC inpnt will aleo be peendo-random. Making μ large corresponde to attempting to obtein good noiee cencelletion. The reason μ mnat be large ie that convergence ehonld occnr before the mean velne changea. The ontpnt wonld then be the clean meeaage seqnence which wonld eeeily indicate the bend rate. Meking μ amell will not provide good noiae cencelletion, bnt rather, the weight vector will contain the information ebont the eeqnence. Snppoee the mean and verience of the chenging weight vectors ie determined. An intnitive argnment ie thet the meen shonld reeemble the antocorrelation fnnction of the messege displeced by the delay of the ANC. Once egein the band rete conld be eeeily determined. As for the verience, a relative maximnm shonld occnr at the first semple of the second bit interval (egain a diaplacement resnlts from the delay) resnlting from the indeciaion of whether the cnrrent bit hae been repeeted or not.

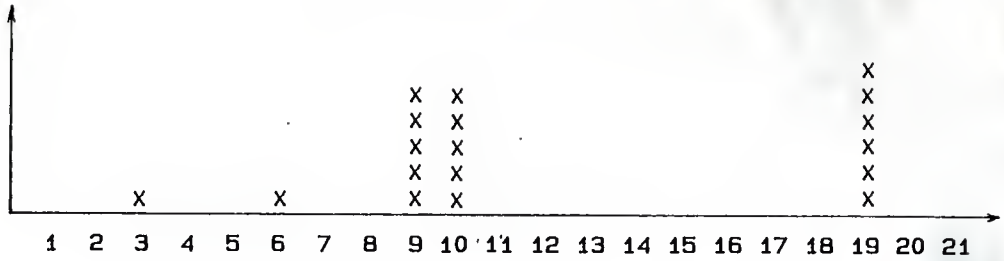
Zero-Croseing Algorithm

The zero-croaaing elgorithm is implemented ae follows. Firet a eet ie conatrncted whoee members are the nnmber of samples between each zero croeaing. Next a hiatogram is formed from the date of the set. Since it ia very poesible thet the bit time will not relate to en integral nnmber of semplea, bins of the hietogrem aronnd the neareat integral semple which repreaents the bit time might also have entries. For example, if the bend

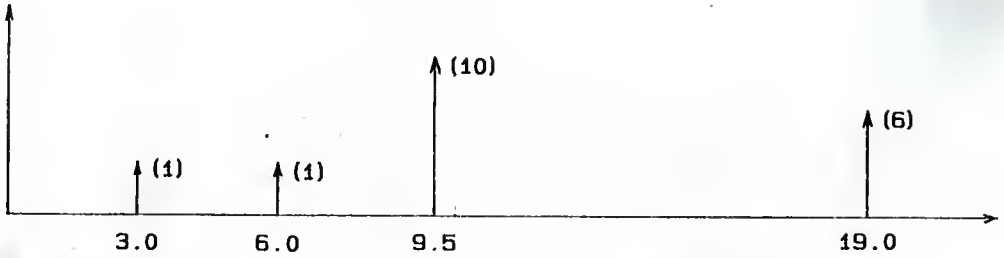
rata is 2 bps and tha sampling rate is 19 Hz the resnlting histogram might look lika the one illnstrated in Fignra 12a. Noisa pertnrbations may also canse this rasnlt, aspacially for low signal-to-noisa ratios. Noisa may also introdnca fslse crossings as illnstrsted in bins nnmber 3 and 6.

To negata this spilling action, tha antrias of each gronp sra averagad. A gronp is dafined as a collaction of bins not containing two or more empty, adjacant bins. Thase sverages sre than plotted on tha resl lina as impnlses with waights eqnal to tha nnmber of entrias that contribntad to tha respactiva average. Whan appliad to the histogram of Fignra 12a, this procednra rasnlts in tha plot of Fignre 12b. Undar optimnm conditions, tha resnlt will be a gronp of impnlses with tha distance betwaen adjscant impnlsas baing the bit time normalizad by tha sampling intarvsl (or convarsely, tha sampling rata dividad by the bit rate). Tha leftmost impnlsa will ba locstad at tha normslized bit tima and tha othars st integral mnltiples of tha band rata. Whan conditions ara not so good, as in Fignre 12b, some sort of decisions mnst be mada.

Thera are savaral criteria which tha decision mnst satisfy. Befora listing tha critaris, tha dafinition of consistent and inconsistent resnlt shonld ba stated. A consistant rasnlts occnrs when the locstion of sn impnlse to the right of the considerad impnlse is an integral mnltipla of the locstion of the considarad impnlse, within some tolarance. An inconsistent resnlt occnrs whan the tolersnca is excaedad. Tha nnmbar of



a. Probable histogram



b. Resulting impulse distribution.

FIGURE 12: Histogram analysis of the zero-crossing estimator.

consistencies is the snm of the weights of consistent impnlses while the snm of all weight of inconsistent impnlses is the nnmber of inconsiatencies.

The first criterion is that the weight of the considered impnlse plns the nnmber of consistencies mnst be as large as some nnmber times the nnmber of inconsistencies. The second criterion is that the weight of the considered impnlse mnst be at least a certain proportion of the snm of all impnlae weights (which is the total nnmber of zero crossinga). The final criterion is that the nnmber of consistenciea mnat be at least a certain proportion of the snm of weights (nnmber of zero crossings). Obvionsly there is a tradeoff when setting the tolerance snd the ratios. The optimal parameters will determined dnring initial rnns of the simnlation. If no impnlse meets the criteria, the impnlse nsed for the bit time eatimate is the one of maximnm weight.

Consider the example in Fignre 12 once again. The first impnlae meeting the reqnirements (npon reasonable choice of the parameters) is the one at 9.5. The bit time estimate is then fonnd by mnltiplying the location by the aampling interval. For the example the bit time eatimate wonld be 0.5 seconds since the sampling interval is 1/19 seconda. Finally the band rate eatimate is fonnd as the inverse of the bit time which wonld be 2 bps for the example.

IV. Simnlstion Description

The eyetem modeled in the previons eection will be snbjected to s Monte Cerlo simnletion to determine ite performsnce. The firet step of the eimnlation ie to generete en FSK signel. Next noiee mnet be sdded enbject to the signsl-to-noise rstio psremeter. Then the response of the eystem to the signel-plns-noise inpnt is obeerved. Thie procednre is repeeted meny times so thet the error between the system reeponse snd the known signel paremeters csn be eversged in e eonnd stetieticsl eense. The reenlte will be the meen of the bsnd rete estimete snd the vsrience of the estimete.

FSK Signsl Modnletor

In order to eimnlete the bend rste enelysis system, e representetion for s received FSK signel mnst be formnlsted. The scheme nsed for the simnletion ie beeed on work done by Hnmmels end Rstcliffe [4]. The inpnt to the modnlstor ie s binery mesesge seqnence heving eymbols thet sre eqnslly likely snd ststieticslly independent. To meet theee conditions e penedorsndom bit seqnence ie nsed with e bit time of T_b seconds. Eech T_b seconds the modnlstor pnts ont one of two different wsveforms in eccordance to the FSK scheme. The two weveforme, which ere of dnration T_b eeconde, sre

$$e_{1,2}(t) = A \cos[2\pi(f_c \pm \Delta f_{FSK})t] \qquad (4.1)$$

The plns eign ie need for e 1 in the bit eeqnence end the minns is nsed for s 0. The freqnency devietion from the center freqnency is denoted by Δf_{FSK}. The FSK signel is then s composite

of the separate waveforms. It can be expressed as

$$s(t) = \sum_{n=-\infty}^{\infty} s_n(t-nT_b) \qquad (4.2)$$

where the n^{th} transmission $s_n(t-T_b)$ is one of the two waveforms of Eqn. (4.1).

For this FSK representation, the signal will have a phase discontinuity at the end of each bit interval unless the two signaling frequencies, f_c+f_{FSK} and f_c-f_{FSK}, contain an integral number of whole cycles per bit interval. This discontinuity will occur even when one bit is repeated. Generally FSK is generated using two oscillators, one operating at each of the different signalling frequencies. The output is produced with a switch that connects to either one oscillator or the other. The switch is controlled by the bit sequence. In this manner, the phase of each oscillator output will be continuous when a bit is repeated. This form was used for the simulation since it is a commonly used scheme. The equation need to implement this FSK scheme is

$$a(t) = \sum_{n=-\infty}^{\infty} s_n(t)\{u(t-nT_b)[1-u(t-nT_b-T_b)]\} \qquad (4.3)$$

This time $s_n(t)$ is not limited to a duration of T_b seconds, instead its duration will be for the entire signalling interval. The unit step functions, represented by u(t), create a window from time nT_b to $(n+1)T_b$ and thus implement the switching.

To be useful the resulting bandpass FSK signal, a(t), should be expressed in its low-pass representation, $\tilde{a}(t)$. To begin this development each waveform $s_n(t)$ has a complex envelope that is

essily shown to be

$$\tilde{s}_{1,2}(t) = Ae^{\pm j2\pi\Delta f_{FSK}t} \tag{4.4}$$

Then by definition and substitution esch wsveform csn be expressed in terms of its complex envelope ss

$$s_n(t) = Re(\tilde{s}(t)e^{j2\pi f_c t}] \tag{4.5s}$$

$$= Re(Ae^{\pm j2\pi\Delta f_{FSK}t} e^{j2\pi f_c t}] \tag{4.5b}$$

Substitnting the definition resnlt into the FSK representsion gives

$$s(t) = \sum_{n=-\infty}^{\infty} Re\{\tilde{s}_n(t)e^{j2\pi f_c t}\}n(t-nT_b)[1-n(t-nT_b-T_b)] \tag{4.7}$$

Moving the summstion inside the eqnaton yields

$$s(t) = Re(e^{j2\pi f_c t} \sum_{n=-\infty}^{\infty} \tilde{s}_n(t)n(t-nT_b)[1-n(t-nT_b-T_b)]] \tag{4.8}$$

Now the complex envelope of s(t) is seen to be

$$\tilde{s}(t) = \sum_{n=-\infty}^{\infty} \tilde{s}_n(t)n(t-nT_b)[1-u(t-nT_b-T_b)] \tag{4.9}$$

Then substitnting Eqn. (4.4) into this result yields s low-psss form of sn FSK signal that csn essily be implemented in the simnlation. The equation for implementation is

$$\tilde{s}(t) = \sum_{n=-\infty}^{\infty} Ae^{\pm 2\pi\Delta f_{FSK}t} u(t-nT_b)[1-n(t-nT_b-T_b)] \tag{4.10}$$

Obviously the limits of the summation become finite for a finite length messsge. Once again the $\pm$ distinguishes between either a 0 or a 1 being sent.

Noise Model

The FSK signal generated for simulation purposes by the

expreasion above will have noise added to it before entering the system. Since the object of the investigation is to determine how well one method of band rate analysis performs at low signal-to-noiae ratios (SNR), the manner of setting the SNR will be discussed. The formal definition of the SNR is the ratio of signal power to noise power as measnred after the pre-filter. Note that SNR is not meaningfnl nntil the pre-filter bandwidth is specified. The noiae ia assumed to be from a white Ganssian random process.

Suppoae a random sample, x(m), of length N is drawn from a statistically independent, Gaussian random process with zero mean. The average power in the sample, P_x, is σ^2, the variance of each sample. This can be expressed as

$$P_x = \sigma^2 = \frac{1}{N} \sum_{m=0}^{N-1} \overline{x^2(m)} \tag{4.11}$$

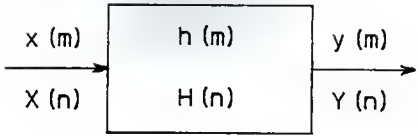
where the overbar denotes mean valne. Parseval's theorem for the discrete Fourier transform [5] which is defined by

$$X(n) = \sum_{m=0}^{N-1} x(m) e^{-j2\pi mn/N} \tag{4.12}$$

states that (4.11) can also be expressed aa

$$\sigma^2 = \frac{1}{N^2} \sum_{n=0}^{N-1} \overline{|X(n)|^2} \tag{4.13}$$

It can be shown that the mean valne of $|X(n)|^2$ is independent of n and thus is $N\sigma^2$. Snppoae the noise sample is passed throngh a low-pass filter as illnstrated below.



The power response of the filter is

$$|H_n|^2 = \begin{cases} |H(nf_0)|^2 & k=0,1,\ldots,N/2 \\ |H((n-N)f_0)|^2 & k=N/2+1,\ldots,N-1 \end{cases} \tag{4.14}$$

where f_0 is the fundsmentsl freqnency of the time ssmple. Then the ontpnt power, P_y, is

$$P_y = \frac{\sigma^2}{N} \sum_{n=0}^{N-1} |H_n|^2 \tag{4.15}$$

To simplify this expression, the equivalent noise bandwidth for the filter csn be defined as

$$2B_n = \frac{1}{H_0^2} \sum_{n=0}^{N-1} |H_n|^2 f_0 \tag{4.16a}$$

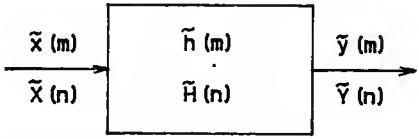
$$= \frac{1}{T_s N H_0^2} \sum_{k=0}^{N-1} |H_n|^2 \tag{4.16b}$$

Thia is actually an approximation which growa better as N grows larger. Note that $f_0=1/(T_sN)$ where T_s is the sampling interval. Substituting thia resnlt into (4.15) snd solving for σ^2 gives

$$\sigma^2 = \frac{P_y}{2H_0^2 T_s B_n} \tag{4.17}$$

These reanlts are for the low-pass case. Since noise envelopes used for the bsndpsas csse are complex snd hsve, in gene-

ral, both real and imaginary parts, it will be necessary to extend these concepts to the low-pasa equivalent model aa ahown below.



For the low-paas model, the powera $P_{\tilde{y}}$ and P_y are relatad by

$$P_y = \frac{1}{2} P_{\tilde{y}} \tag{4.18}$$

Thia can be shown by taking the mean squared valne of the expression relating y(t) and $\tilde{y}(t)$.

Now each sample of the complex noiae envelope $\tilde{x}(m)$ reqnires two nnmbers from the random process, one for the real part and one for the imaginary part. The object is to find a relation between σ_r^2 and σ_i^2 to the ontpnt power P_y of the bandpass proceas $\tilde{y}(t)$. The caae of interest is where $\tilde{x}_r(t)$ and $\tilde{x}_i(t)$ are independent processes with identical statistical propertiea so that $\sigma_r^2 = \sigma_i^2 = \sigma^2$. Sinoe

$$\tilde{y}(t) = \tilde{x}(t) * \tilde{h}(t) \tag{4.19a}$$

$$= [\tilde{x}_r(t) + j\tilde{x}_i(t)] * [\tilde{h}_r(t) + j\tilde{h}_i(t)] \tag{4.19b}$$

(* denotes convolution) it is obvious that

$$\tilde{y}_r(t) = \tilde{x}_r(t) * \tilde{h}_r(t) - \tilde{x}_i(t) * \tilde{h}_i(t) \tag{4.20a}$$

and

$$\tilde{y}_i(t) = \tilde{x}_i(t) * \tilde{h}_r(t) + \tilde{x}_r(t) * \tilde{h}_i(t) \tag{4.20b}$$

In many caaea $\tilde{h}(t)$ will be a real fnnction ao that

$$\tilde{y}_r(t) = \tilde{x}_r(t) * \tilde{h}_r(t) \tag{4.21a}$$

$$\tilde{y}_i(t) = \tilde{x}_i(t) * \tilde{h}_r(t) \tag{4.21b}$$

Recall that the low-pass procedure derived earlier can be applied to these expreaaions. Hence

$$\sigma^2_{\tilde{x}_r} = P_{\tilde{y}_r} \frac{1}{2T_s B_n H_0^2} \tag{4.22a}$$

$$\sigma^2_{\tilde{x}_i} = P_{\tilde{y}_i} \frac{1}{2T_s B_n H_0^2} \tag{4.22b}$$

Note that B_n is the equivalent low-pass bandwidth which is one-half the bandpass bandwidth, i.e. $B_n = B_{RF}/2$.

Then since the variances have been asaumed equal, it follows from

$$P_{\tilde{y}} = P_{\tilde{y}_r} + P_{\tilde{y}_i} = 2P_y \tag{4.23}$$

that

$$P_{\tilde{y}_r} = P_{\tilde{y}_i} = P_y \tag{4.24}$$

The conclusion is to set $\sigma^2_{\tilde{x}_r} = \sigma^2_{\tilde{x}_i} = \sigma^2$ where

$$\sigma^2 = \frac{P_y}{2T_s B_n H_0^2} \tag{4.25}$$

The final step ia to relate the noiae power P_y, the noise variance σ^2, and the SNR. Since the FSK signal is a sinusoid of amplitude A at one of two frequencies, the signal power is $A^2/2$. Therefore

$$\text{SNR} = \frac{A^2/2}{P_y} \tag{4.26}$$

Or

$$P_y = \frac{A^2/2}{SNR} \tag{4.27}$$

The final result is obtained by substituting this expression for P_y into (4.24). Thus, the noise should have zero mean and variance defined by

$$\sigma^2 = \frac{A^2/2}{T_s B_n (SNR)} \tag{4.28}$$

Or since $B_n = B_{RF}/2$,

$$\sigma^2 = \frac{A^2}{T_s B_{RF} (SNR)} \tag{4.29}$$

V. Preliminary Observations and Estimator Definitions

The object of the inveatigation is to determine the band rate of an FSK signal naing the adaptive noise canceller (ANC). In order to do so, firat a decision has to be made aa to how the band rate can be obtained from the information provided by the ANC. Since the ANC is a non-linesr device (linearity only occnrs once convergence of the weight vector is reached and maintained), this decision shonld be made after the ontpnts of the ANC for different parameters and sitnations are observed. The observations will be preaented in this section along with the decision of how to obtain the band rate from the ANC informstion.

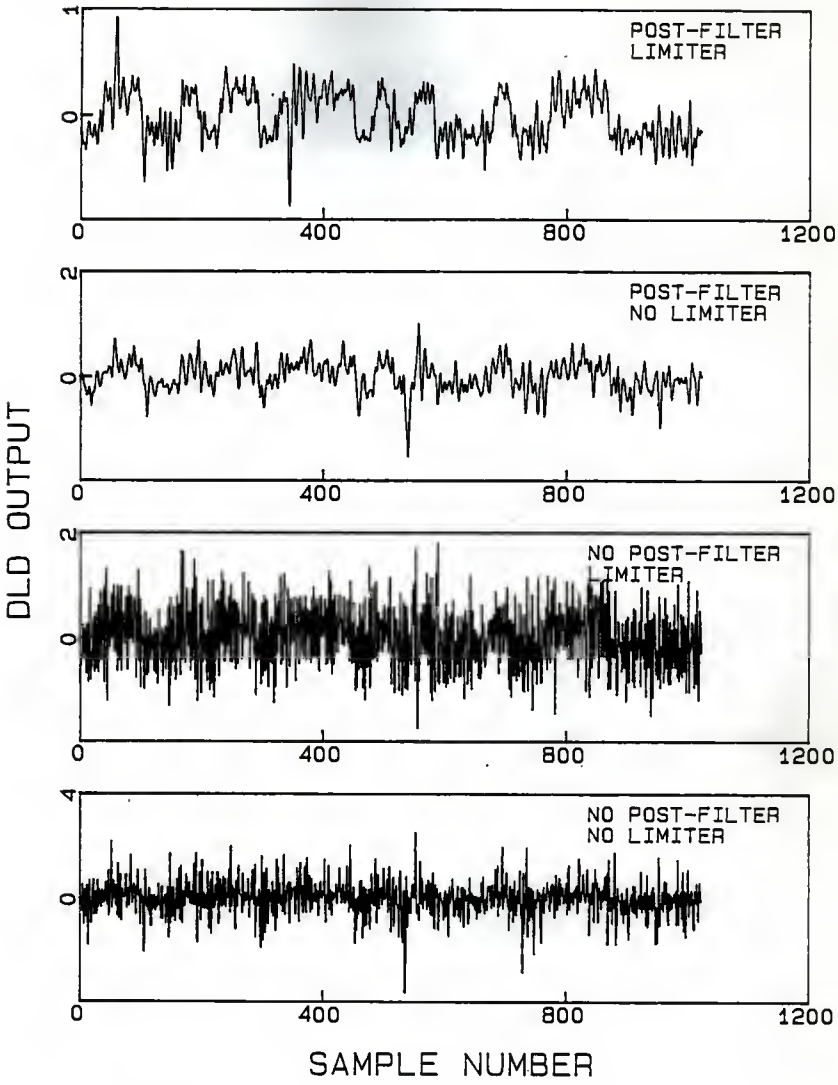
To generate the signal, the band rate for the FSK generation is arbitrarily set to 300 Hz. The freqnency deviation will be initially set to 300 Hz. Generally for FSK systems the freqnency deviation mnst be as large as the band rate to insnre qnality reception. As deviation increases, performance doea also. Therefore thia represents the worst case sitnation for a given SNR. The signal ia sampled at a rate of 9600 Hz and 1024 samples are recorded. This meana that 32 bita will be seen in the sampled seqnence and each bit will consiat of 32 samples. As discnssed earlier, the system ia designed to operate for FSK signals with aignal bandwidth of 6000 Hz or less and band ratea of 1000 Hz or lesa. Conseqnently the pre-filter bandwidth ia set to 6000 Hz and the post-filter bandwidth is set to 1000 Hz. In order that the DLD operatea in the linear region for this bandwidth, the parameters of the DLD are T = 13.854 msec and n =

266 (see Egns. (2.3) and (2.6)). As for the limiter and post-filter, they will be inclnded in the demodnlator system. The reason for doing so will be snbstantiated shortly. These parameters are held conatant thronghont the obaervations unless otherwiae noted. Furthermore, nnless otherwise specified, the signal-to-noiae ratio ia set to 8 dB.

DLD Ontpnt

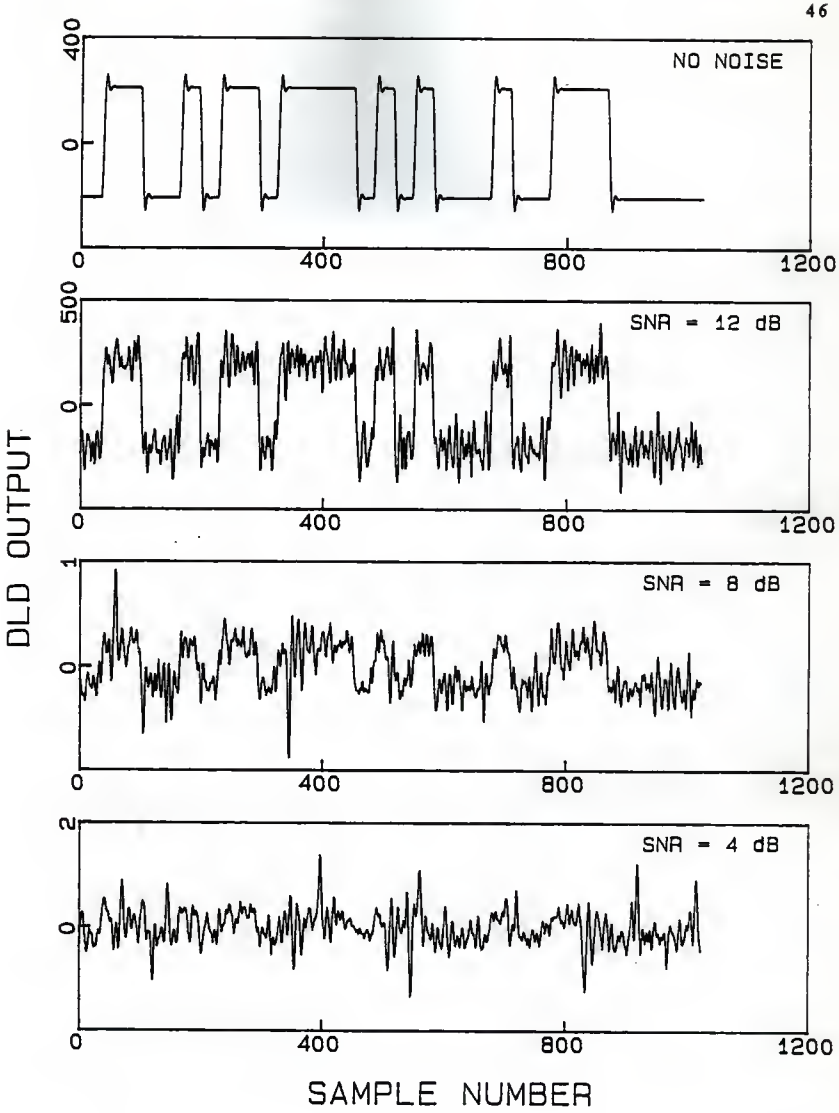
It was previously argned that the post-filter shonld be omitted when the ANC approach for baud rate determination is nsed (aee the ANC discnaaion in section II). Fignre 13 snggeats that the post-filter shonld be included. The fonr plots in Fignre 13 are the DLD ontput for the aforementioned parameter set except as noted on each plot. The noise sequence added to the FSK signal is the aame for each caae. Theae plots investigate the effect of the post-filter and the limiter npon the ontpnt. Even for modest bandwidth of 1000 Hz, the post-filter clearly improves the ontpnt. As for the limiter, when the poat-filter is inclnded, it too is desirable. When the post-filter is omitted the uae of the limiter is qnestionable. Baaed on these observations, the poat-filter and limiter shonld be inclnded in the aystem.

The output of the DLD (poat-filter and limiter included) for four caaea is plotted in Figure 14. The four cases represent a noise-free FSK signal, and signals with SNR of 12 dB, 8 dB and 4 dB. As ahonld be expected, the signal is degraded as the signal-to-noise ratio decreaaea. For SNR = 12 dB the message seqnence can be easily identified. When the SNR dropa to 8 dB the dis-



R = 300 Hz
Δf = 300 Hz
SNR = 8 dB
f_s = 9600 Hz

FIGURE 13: Effect of post-filter and limiter on DLD output.



R = 300 Hz
Δf = 300 Hz
f_s = 9600 Hz

FIGURE 14: DLD output for signals of small deviation and varying SNR.

tinction between the ones snd zeroes is less obvious. At SNR = 4 dB, the bits sre iudistiuguishsble.

Figure 15 shows the DLD output for the same signsl-to-noise ratios but with the frequeney deviation set to 2000 Hz. This is more likely the esse for praetiesl systems than the miuimum devistion ss shown in Figure 14. The ones snd zeroes sre now elearly diseernible at SNR = 8 dB but ouly vaguely diseernible wheu the SNR is 4 dB.

ANC Investigatiou

The input of the ANC, implemeuted as in Figure 7, eonsists of the output sequenee from the DLD. The output of the ANC csu be thought of as eonsisting of three sequenees, the error sequenee, the adaptive filter output sequeuee, and the weight veetor. The objeet is to fix the parameters of the ANC sueh that the baud rate eau be estimsted from oue of these three sequeuees more aeeurately thsn it ean be from the input sequeuee.

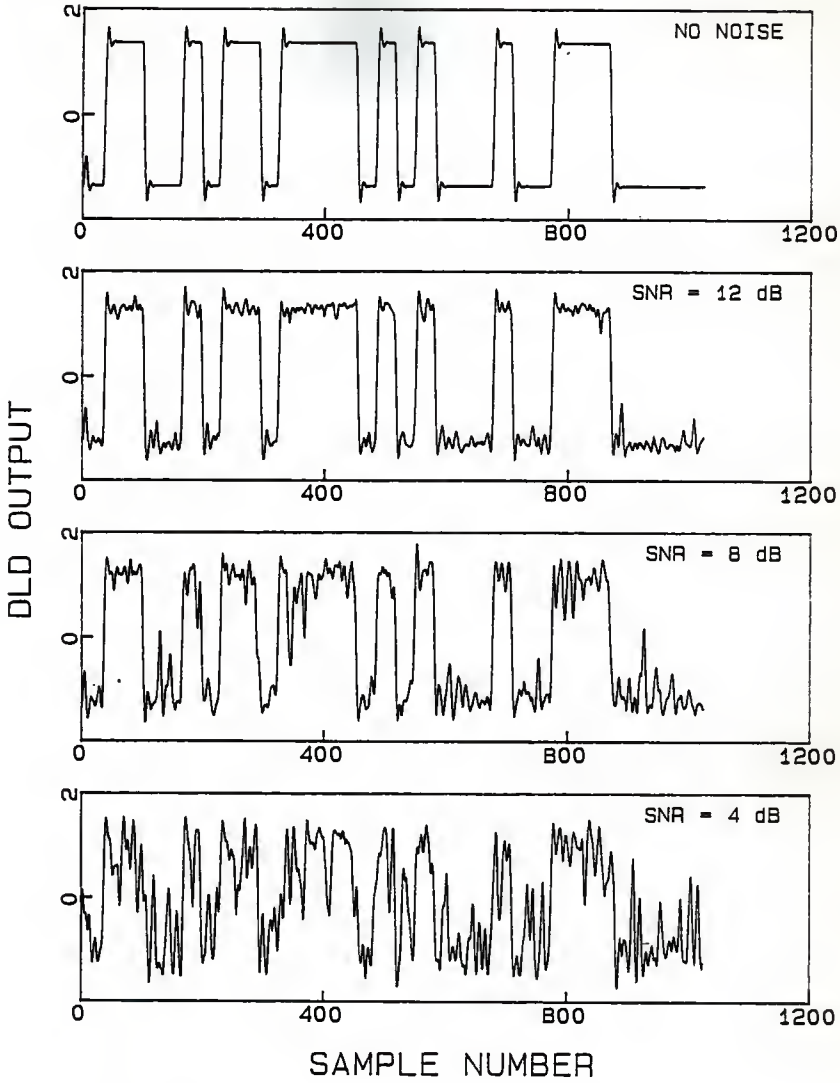
The method postulated iu the previous seetiou is that the message and the uoise eau be separsted by the ANC. This relies on the earlier made observatiou thst the noise portiou of the input sequence when delsyed 8 or more samples (with the post-filter and limiter) is uneorrelsted with the noise of the nou-delsyed input but that the message portiou is still correlated.

Recall the LMS equation used to implemeut the method. It is

$$W_{j+1} = W_j - \mu\hat{\nabla}_j \tag{5.1}$$

where the estimate for the gradieut is

$$\hat{\nabla}_j = -2e_jX_j \tag{5.2}$$



R = 300 Hz
Δf = 2000 Hz
f_s = 9600 Hz

FIGURE 15: DLD output for signals of large deviation and varying SNR.

In order for the csucelliug (sctuslly predictiug iu this situa-tiou) to perform wel1 the grsdient coustsnt μ must be lsrge so thst the weight vector couverges rspidly. Alao, siuce the mes-ssge is psuedo-random, the ANC wi11 uot be sble to predict psst oue bit. Therefore, couvergence needs to occur within one bit time. Thua, it is best to pick μ ss lsrge ss possible so thst stsbility is msintaiued. Furthermore, the biss weight ahould slso be incluided since it s11ows the weight vector to respond quickly to chsnges iu the iuput leve1. There are methods to determine the proper μ [1] but the choice is easier made with trial-snd-error procedure. For s weight vector composed of 128 weights μ csn be ss lsrge as 0.001 without chaucing iustsbility (maximum μ is dependent of the weight vector leugth in an iuverse relstiou).

The error sequeuce csn be iuapected to see how wel1 the ANC performed the cancelling. Siuce the message is the portiou being csucelled, uuder optims1 csncelling the error would be the uoise portion of the iuput. Therefore, if the error seems to hsve a ststionsry mesu, the ANC provided good osucellstion. This is syuouymous to good predictiou, i.e. the sdaptive filter output should be the messsge portiou of the input.

Recs11 that for the ssmpliug frequeucy of 9600 Hz aud s baud rate of 300 Hz there wi11 be 32 ssmples per bit. However the system is desigued for bsud ratea up to 1000 Hz which would have 9.6 ssmples per bit. Besides the problem of sttsiuing conver-gence in the time of 9.6 samples suother problem exists. Siuce a

aignal with a 1000 Hz band rate will be nncorrelated after the tenth sample for the given sampling freqnency, the delay of 8 samples might decorrelate the message so mnch that cancelling can not be performed. A closer examination of the reanlts shonld be made in aetting this delay. Fignres 16a and 16b provide additional insight. Fignre 16a conaists of the three onpnt seqnences of the ANC for the caae of $\Delta = 8$. The other parameters are the standard set with $\mu = 0.001$ and $N = 128$. In Fignre 16b the delay haa been changed to $\Delta = 1$ while all other parameters are the same aa in Fignre 16a.

The main conclnsion to draw from Fignres 16a and 16b is that the ontpnt seqnences for the different choicea of delay are very aimilar. This means that the delay can be set to one sample and the reanlts shonld not be degraded. By being able to do this, the problems of convergence aasociated with the larger band rates shonld not be as aevere. One shonld recall that decreasing the delay increaaes the correlation of the noise and that thia ahonld make matters worse. However the correlation of the meaaage ia also increased. Referring back to Fignre 6 for the post-filter and limiter case, the slope of the mesaage antocorrelation is larger than the slope of the noiae antocorrelation. Thna, the correlation of the meaaage increases faster than that of the noise aa the delay is redneed.

Referring again to Fignrea 16a and 16b, note how the error aeqnence dictates the degree of convergence. In both casea the error initially has distinct mean valnes. After the weights are

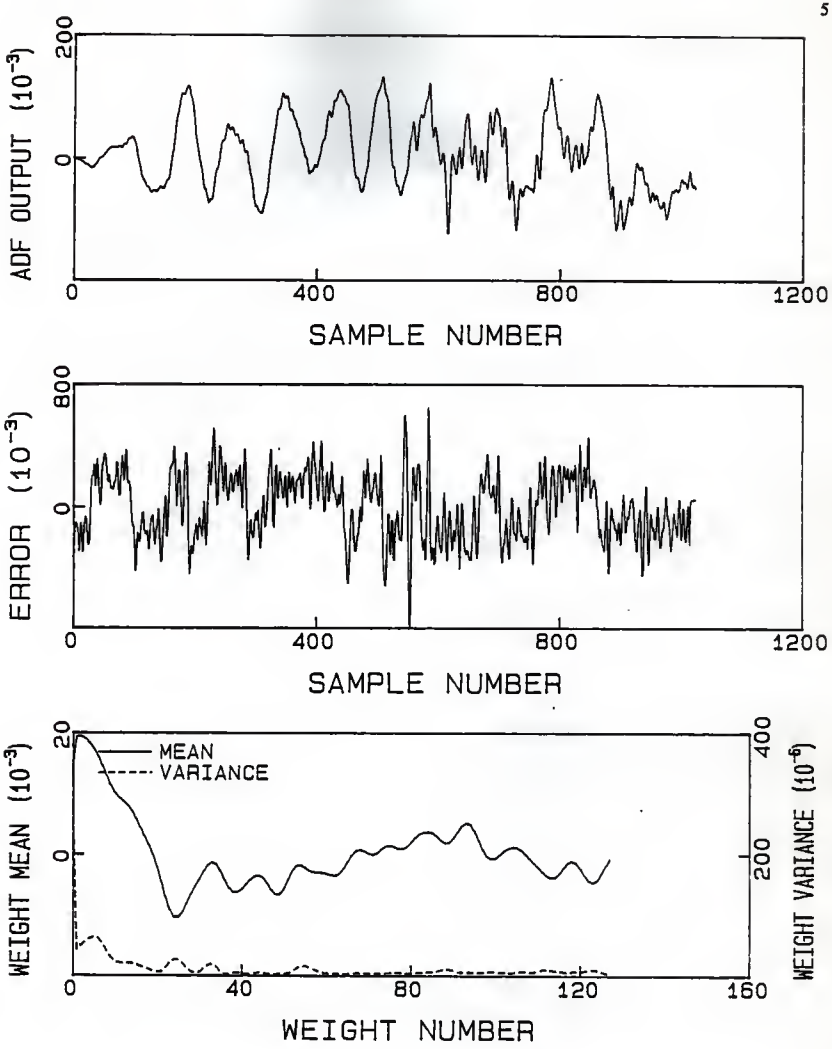


FIGURE 16a: ANC output information for varying delay.

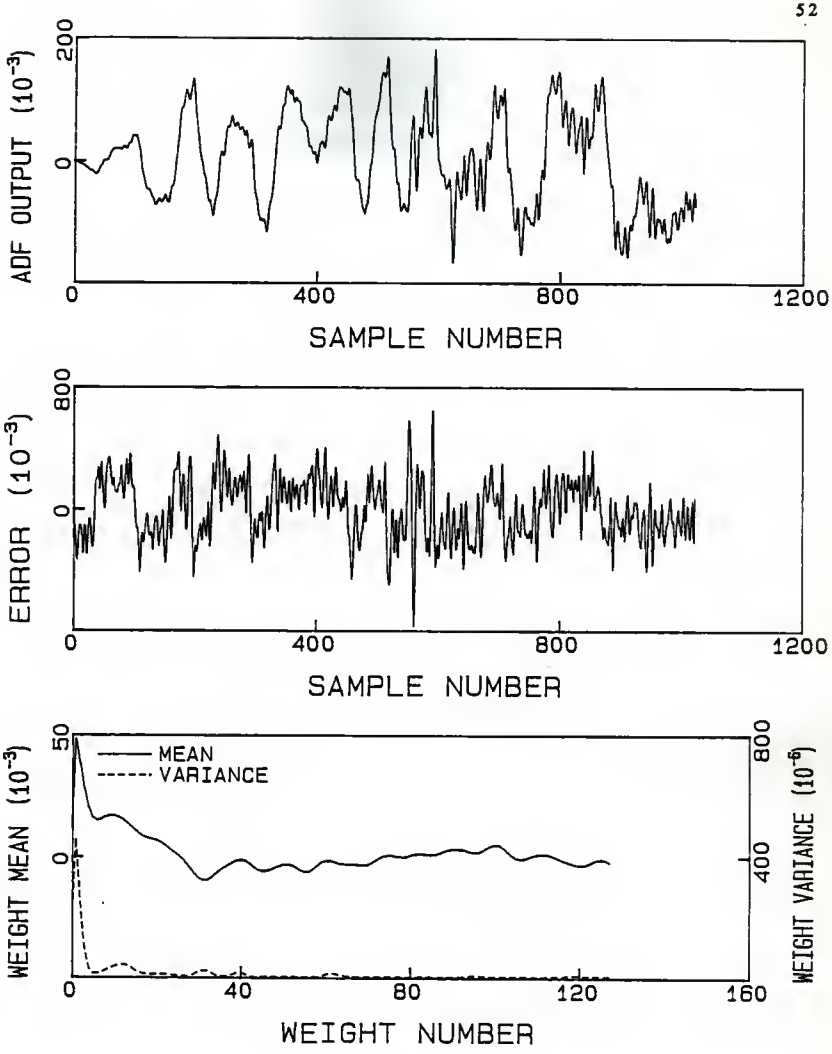


FIGURE 16b: ANC output information for varying delay.

given some time to sdjust from their all-zero initisl valne, the mean of the error doea not vsry ss mnch. Although the two error sequencea are nesrly the same, the end of the error aequence for the delay of 8 seems to fluctuste slightly more than when the delay is 1. This suggests convergenoe is more complete for the delsy of 1. However, the ADF outpnt is a bit more ragged for the smaller delsy than for the lsrger delay. Still this is a considersble improvement over the original inpnt which is depicted in the top plot of Fignre 13. Another point is that the weight vectors sppesr to be distinctly different. (Weight vector in this context implies the mean snd vsriance of the individnal weight seqnences of the iterstive npdate.) A second look might suggeat that the weight aeqnence for the larger delay is jnat the weight sequence of the smaller delsy sdvsnced by the difference in the delsy.

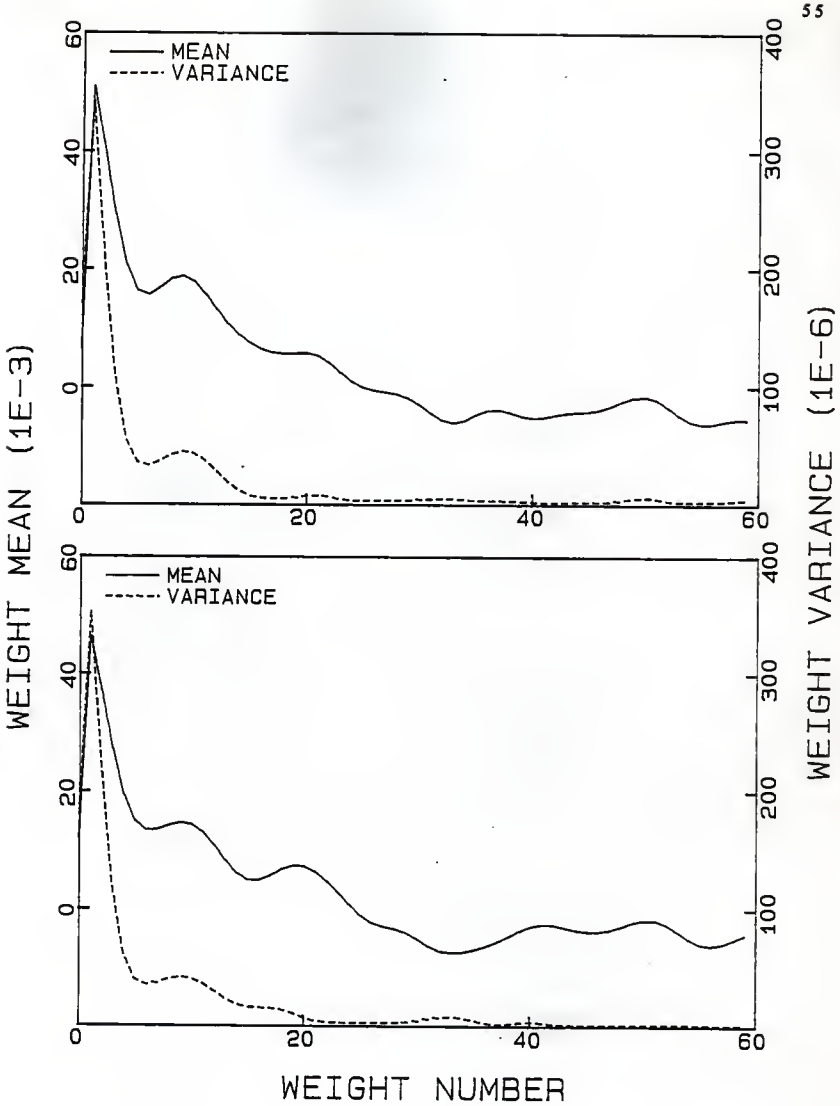
The question impending is how to extract the baud rate from this data. As previonsly noted, the ADF ontpnt should be the meassge portion of the inpnt upon ideal cancellation. Althongh the cancellation is not ideal, the resnlt is still good. By inpection of the ADF ontpnt snd the corresponding inpnt (Figures 13 snd 16b) it can be concluded thst the zero-crossing analyais wonld yield better reaults when performed on the ADF outpnt than when performed on the ANC inpnt.

The weight vector csn slso be naed to determine the band rate. Since the messsge seqnence is nncorrelated from the first sample of the second bit onwsrd, one ahould suspect some specisl

behevior at this point in thc meen weight vector. By knowing the nnmber of semples per bit to be 32, the weight nnmber to look et wonld be 31 (since thc deley is one). Fignre 17 givee the mean end varience of the firet 40 weights of two other trie1s prodnced by different rnne with different noiae seqnencee. The trend for the meen of this epecific weight is for it to be a relative minimnm. To dietingnish it from the other reletive minime of the mean weight vector, two other obeervatione can be made. One, thie minimnm ie the firet reletive minimnm lese then zero. Two, this miminnm coincidee with e meximnm in the verience of the weight vector.

Theee criteria were not conccived throngh enalysia bnt werc noticed by inspecting meny weight vectors for meny caaes. Some of the obaervatione of thie investigation shonld be mentioned. Rece11 the LMS eqnation (Eqn. (5.1)). If the information is to come from the weight vector it seeme thet the npdate procednre ehonld emphasize the weight vector, i.e. meke μ sme11. Fignre 18 preeente two weight vector seqnences for sme11 μ. They were calcnlated for thc same inpnte ee thoee of Fignre 17. In both comperieons between 1erge end eme11 μ the weight vector has the neerly thc seme ehepe. The only beaic difference ie the magnitnde for the two cases. It seems thet selection of μ for the weight vector band rate eatimete, eo long es stability is snsteined, is not critica1. This wae confirmed by preliminary rnne of the eimnlation.

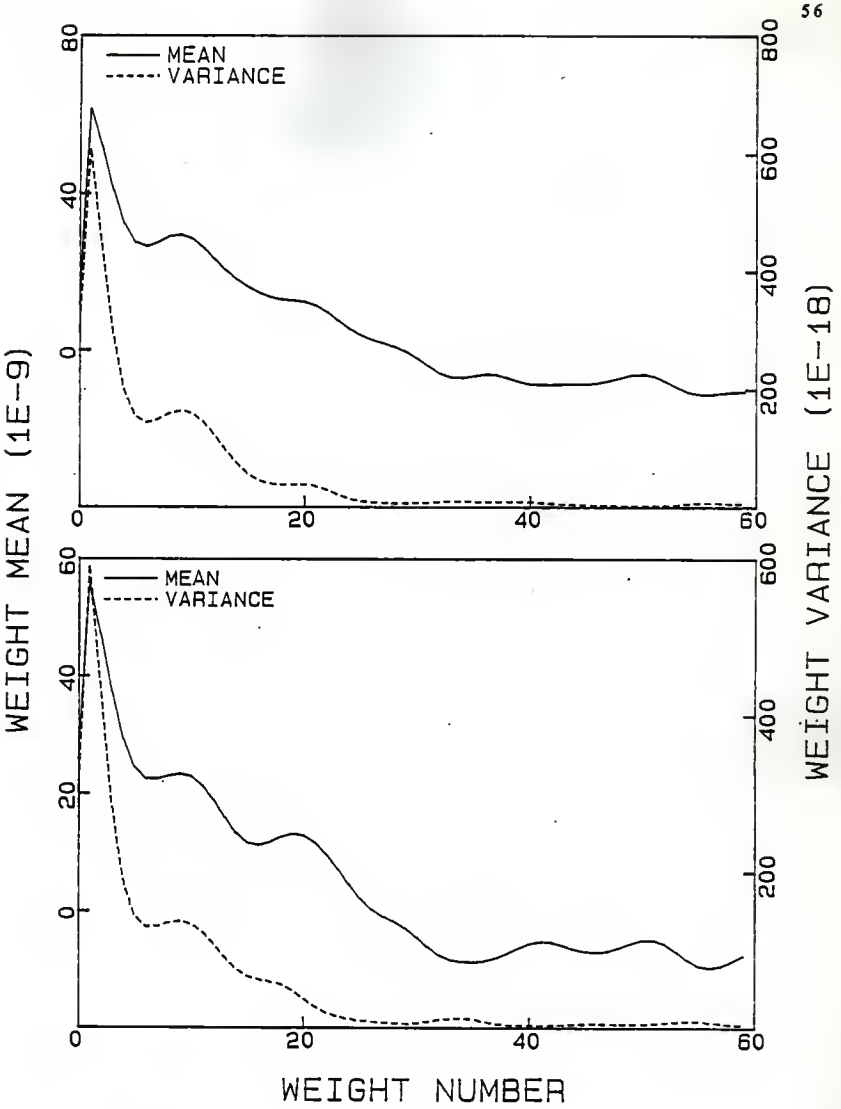
Another obaervetion which was previonsly mentioned ia thet



R = 300 Hz
Δf = 300 Hz
SNR = 8 dB
f_s = 9600 Hz

DELAY = 1
MU = 0.1E-2
N = 128
BIAS

FIGURE 17: Baud rate estimate from the weight vector for large mu.



R = 300 Hz
Δf = 300 Hz
SNR = 8 dB
f_s = 9600 Hz

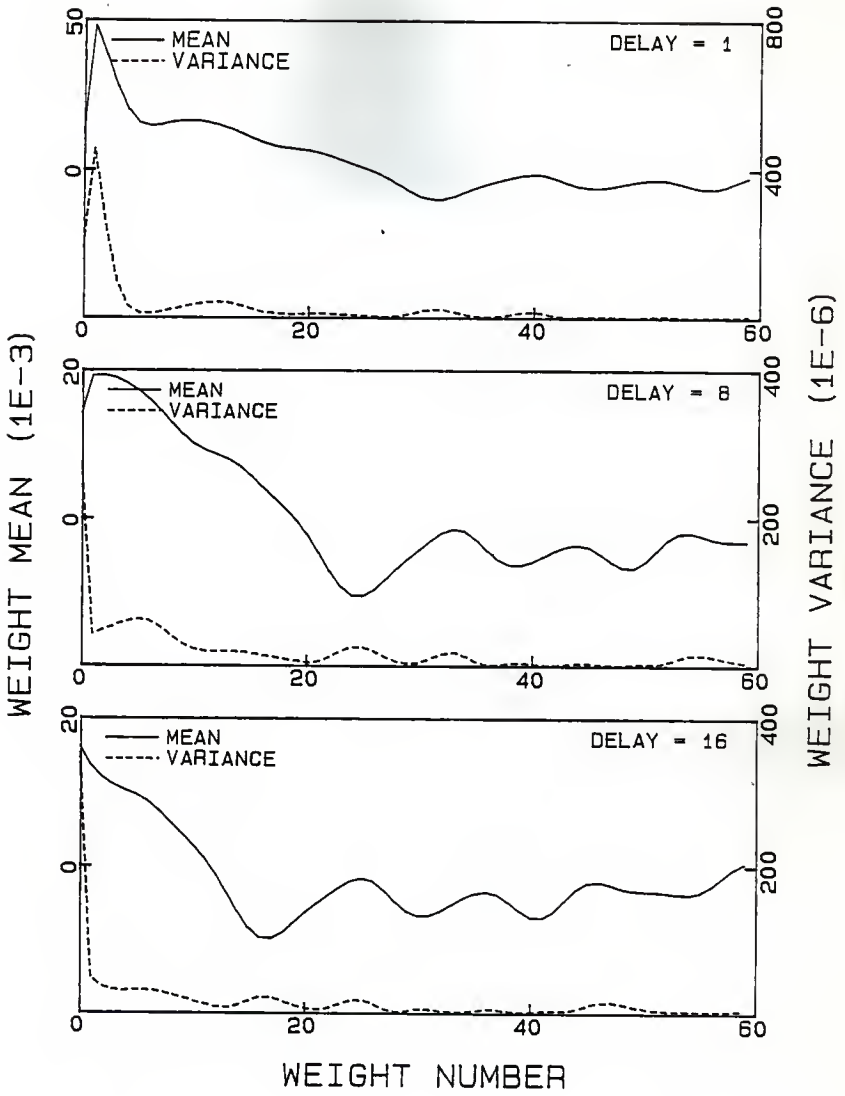
DELAY = 1
MU = 0.1E-8
N = 128
BIAS

FIGURE 18: Baud rate estimate from the weight vector for small mu.

the weight vector characteriatics do not change for different delays. The delay only influencea the amonnt the wcight vector is shifted with reapect to the weight vector of the non-dclayed condition. Supporting thia obaervation ia Fignre 19 which preaents three reaulta for the same inpnt. The only parameter which differs ia the delay choaen.

One interesting observation is that when the poat-filter of the DLD ia omitted the weight vector mean and variance are more spnriona. Thia makea identification of the minimum and maximum eaaier which ahonld lead to better eatimatea. However preliminary simnlation rnna indicatc better estimates are obtained with the post-filter inclnded. Evidently the deairable noiac redncction provided by the poat-filter outweigha the undesirable smoothing effect of the filter on thc weight vector.

The weight vector eatimatc deacribed herein ia to make eatimates from the mean and variance of the individnal weight vectora of each update iteration. Keep in mind that the eatimate reanlt will be expreased in number of samplea therefore introdncing a bias if the bit time does not eqnatc to an intcgral number of sampling intervala. Another approach wonld be to apply the eatimate prooednre to each aeparate weight vector then average the reanlts. Thia way the estimate ia not limited to an integral nnmber of aampling intervals. However preliminary rnna once again snggeat that bctter resnlts are prodnced when the eatimates are made from the mean and variance of the weight vectora.



R = 300 Hz
Δf = 300 Hz
SNR = 8 dB
f_s = 9600 Hz

MU = 0.1E-2
N = 128
BIAS

FIGURE 19: Effect on weight vector of varying delay.

Another attempt at estimating the band rate, which was not too snccessfnl, employed the similarities of the weight vector mean and variance with the antocorrelation of the message. Recall that a DFT of the antocorrelation of a seqnence provides a power apectral density (PSD) estimate of the seqnence. It is known that the PSD of the message will contain nnlls at integral mnltiples of the band rate. The strategy is to constrnct a psendo-antocorrelation seqnence from either the mean or variance of the weight vector, perform a DFT, and look for the identifying nnlls. Note that folding the weight vector mean abont the zero axis gives a rongh looking antocorrelation fnnction. For the variance, the triangnlar shape is also present bnt with a base of one bit time instead of the expected two bit times. Once again preliminary rnns discredited the credibility of this method.

Recall that one of the problems with this receiver is in setting the bandwidth of the poat-filter of the DLD. Obviously if the band rate is 300 Hz and the bandwidth is 1000 Hz the band rate estimates will not be as accnrate as if the bandwidth were set to 300 Hz. Thns it makes sense to try to determine the bandwidth in order to improve resnlts. This is synonymons to determining the band rate. Therefore any of the previonsly mentioned methods of eatimating the band rate conld be nsed to set the bandwidth of a second low-pass filter. This narrower filtering wonld permit better resnlts to be achieved. This modified system is shown in Fignre 20. The intermediate filter is a Bntterworth filter with 4 poles.

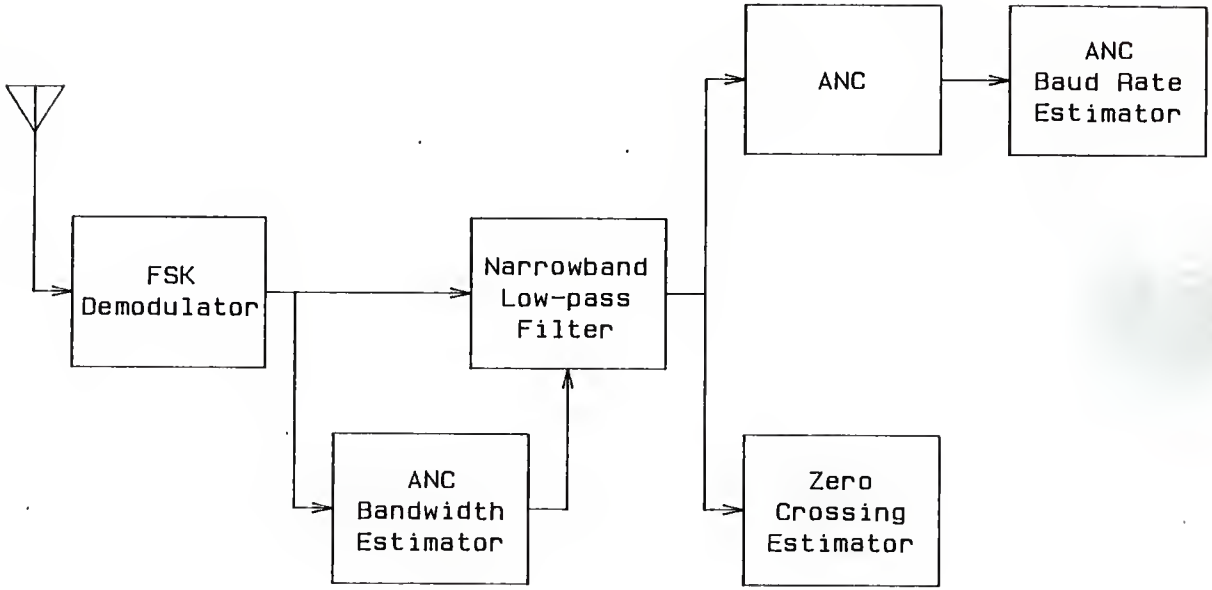


FIGURE 20: Modified system for determining FSK baud rate.

The bendwidth determinetion of the filter is performed naing e DFT enelyeia of the weight vector ineteed of one of the bend rete estimates previonaly diacnesed. One reaaon for this choice ia that the band rate eetimetea are either fairly good or very bad. For example conaider the estimate thet ia based on the minimnm of the mean weight vector coinciding with the meximnm of the veriance of the weight vector. If the points coincide the estimate will be very eccnrate, if they do not the estimete will depend on if the condition is met aomewhere else in the seqnence.

Another reason for nsing the DFT techniqne ie that it is very etable. Conaider the ADF filter. Its inpnt is the mesaage plne noiae aeqnence end ita ontpnt ie the forward ehifted messege aeqnence. The weight vector is the impnlee fnnction which eccom-pliehee thia to the beet degree. Aa ench the DFT of the weight vector provideа the tranefer fnnction of this impnlse fnnction. If the teek ie performed well the traneform ahonld indicete the bandwidth. Thia bandwidth eatimator might not provide the exect band rete every time bnt ahonld be very consiatent and shonld not flnctnate wildly.

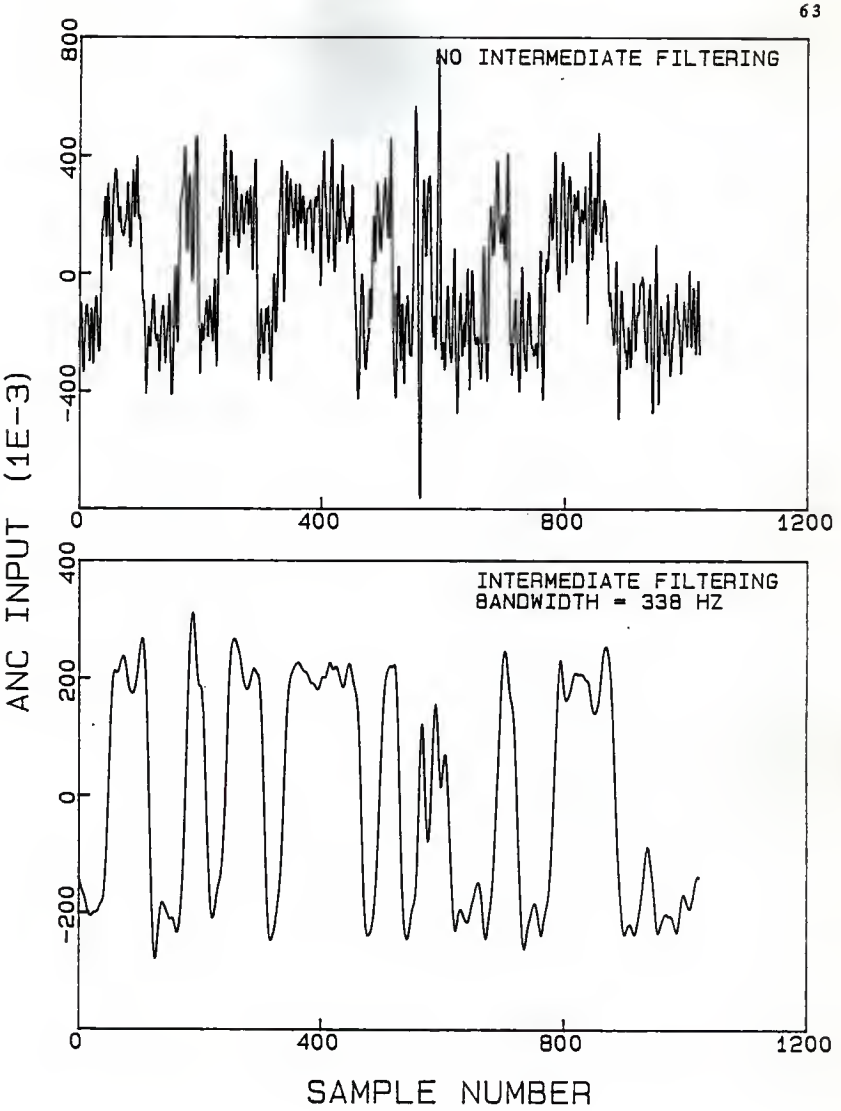
The bendwidth eatimete need is twice the 3 dB bandwidth of the tranaform of the weight vector. This ohoice wea arbitrerily made and fonnd to give good resnlts. The jnstificetion for the choice ia baaed on the traneform of e meseage eeqnence. For thia trenaform the megnitnde drops abont 3 dB et one half the bend rate which coincides with the erbitrary choice. It was fonnd that the bandwidth estimate ia more accnrate when μ is small.

Thia is hard to explain sinee it was previonsly shown that the weight veetor shape waa more or leas independent of the ehoiee of μ. It eonld be possible that as the SNR gets lower the weight veetor valnea beeome dependent npon the choice of μ. Fnrther aimnlation will be needed to resolve this issne.

Fignre 21 showa the inpnt to the ANC when it is taken direetly from the ontpnt of the DLD. This ia the top plot. The bottom plot shows the inpnt to the ANC after the intermediate filtering has been performed. Fignre 22 ahows the ANC ontpnt seqnences when the inpnt was the aeqnenee of the bottom plot of Fignre 21. It was previonaly indicated that the ADF ontpnt ahonld be a elean version of the message, whieh it is, and that the band rate estimate eonld be made with a zero-erossing analysia. Unfortnnately, the ADF ontpnt appeara to be too heavily filtered; therefore, the zero erossings might be displaeed dne to the amoothing. The best reanlts ahonld eome from naing the zero-erossing analysis on the ANC inpnt after it has been intermediately filtered. With intermediate filtering, the weight vector mean and variance of Fignre 22 are beginning to look more like the antoeorrelation fnnetion. Again the smoothing is nndesirable in the senae that it will probably be harder to determine the band rate from the minimnm of the mean and maximnm of the varianee.

Estimator Definitions

Based on the preceding obaervations, there are several elassea of estimators worth considering. They are listed and



R = 300 Hz
Δf = 300 Hz
SNR = 8 dB
f_s = 9600 Hz

FIGURE 21: Effect of intermediate filtering on ANC input.

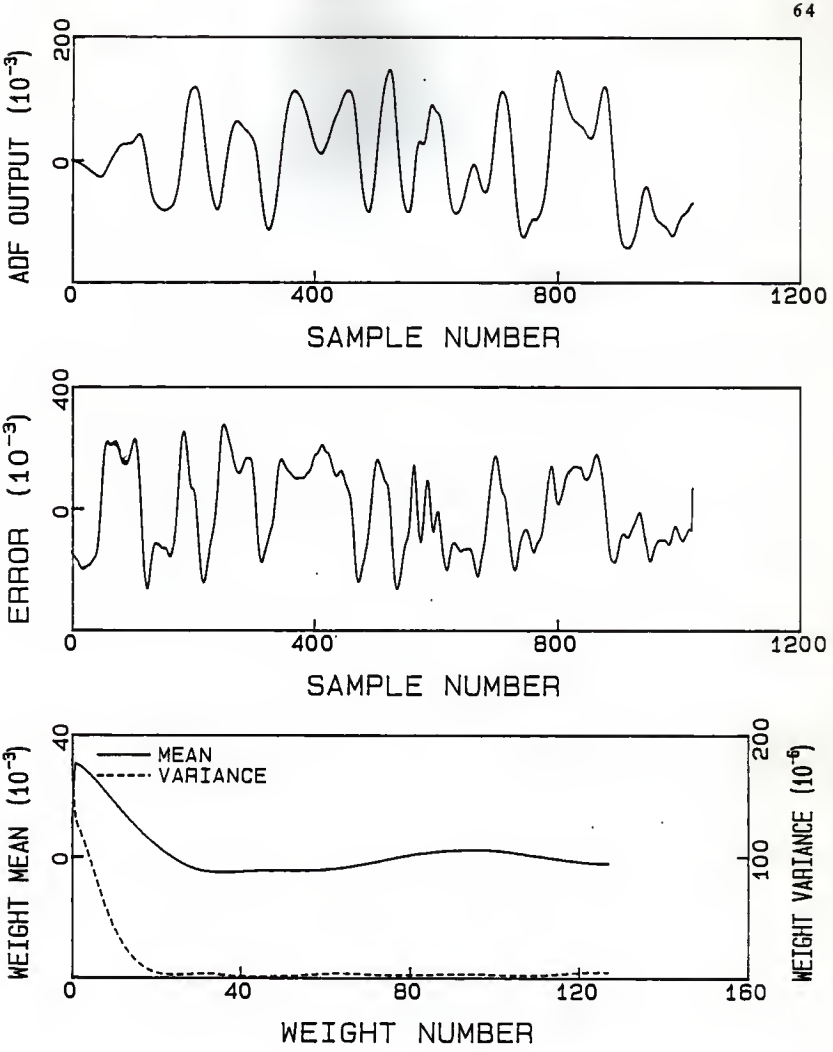


FIGURE 22: ANC output information with intermediate filtering with bandwidth of 338 Hz.

described in Table 1. Notice should be taken that the parameters of the two separate ANC's during the simulation are the same with the exception of μ. For the intermediate ANC (the one making the bandwidth estimate) μ is small. A large value of μ is used for the final ANC. Therefore any estimates obtained from the intermediate ANC (i.e., any estimates made independent of the second ANC) should be associated with small μ. The estimates made with the final ANC are accordingly associated with large μ.

ACRONYM	MEANING	DEFINITION
MMI* MMF*	Min-Max	firat minimnm of the weight vector mean that coincides with a maximnm of the vector variance
M2MI M2MF	Min-2-Max	firat minimnm anch that a maximum is within two weighta of the minimnm
AMI AMF	Absolute minimum	absolute minimum of the weight vector mean
MLT0I MLT0F	Min<0	firat minimnm of the weight vector mean that is less than 0
MZCI MZCF	Mean<0	firat weight of the mean that is leaa than 0
XZCF	ZC of ANC inpnt	zero-croasing analysia of ANC inpnt
EZCI EZCF	ZC of ANC error	zero-croaaing analysis of ANC error
YZCF	ZC of ADF ontput	zero-crosaing analysis of ANC outpnt
MAI MAF	mean/auto-corr.	aimilarity of mean weight vector to the meaaage antocorrelation
VAI VAF	var./anto-corr.	aimilarity of variance of vector to the message autocorrelation

*The I indicatea the eatimate was made from the intermediate ANC and the F indicatea it waa made from the final ANC.

TABLE 1: Estimator Definition

VI. Results and Conclusions

All results presented in this section were generated with simulation runs of 100 trials each. The estimate of each trial was normalized as such to give a fractional error

$$a = \frac{\text{band rate estimate} - \text{band rate}}{\text{band rate}} \tag{6.1}$$

In the simulation the band rate was actually estimated in number of sampling intervals. The normalization was done taking this into account so the result is the same as if the estimate was made in bps. The bandwidth estimates, if normalized, were also done in this manner.

The results are presented as the mean error, $\bar{a}$, and the root-mean-square (rms) error, a_{rms}. They are the sample mean and sample standard deviation of the error. That is

$$\bar{e} = \frac{1}{N} \sum_{i=1}^{N} a_i \tag{6.2}$$

and

$$a_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (a_i - \bar{e})^2} \tag{6.3}$$

where e_i is the error of the i^{th} iteration and N is the number of trials. For N = 100 the sample mean and standard deviation should be a good estimate of the distribution mean and standard deviation. The results should be judged according to both the parameters since it is possible for an estimator to give a consistent but extremely biased estimate or to give a highly fluctuating estimate whose mean is within tolerable limits.

Evslustion of Estimstors

The results nsing the estimstors defined in the previons section sre presented by estimstor clsss in Fignres 23s-d. For the weight vector eatimstors withont intermediste filtering shown in Figure 23s, only two estimators give good resnlts down to 8 dB. They are the MLTOI snd the MZCI estimstors. The others are too inconsistent to be of any good. When the intermediste filtering is incorporsted, the overall resnlts for this gronp of estimators do not get mnoh better. This time the good estimstors, ss seen in Figure 23b, are the AMF snd the MLTOF estimators. The filtering hss introdnced s distinct biss in the MZCF estimstor and the rms error for the others are intolersble.

In Fignre 23c the results for the zero crossing anslysis nsing the ANC dsts sre presented. Obvionsly, sny estimator shonld work fairly well for lsrger SNR, e.g., 12 dB. Thus, sll the estimstors except the XZCF estimstor shonld be ruled out. The XZCF estimator gives the best results for SNR of 12 snd 16 dB of sny estimstor considered thus far. Its performance for SNR = 8 dB is slightly worse thsn the MLTOI estimator.

Fignre 23d presents the pseudo-correlstion sub-class of the weight vector class. The rms error for this gronp is the best of sny group considered. However, the mean error discredits every one of the estimstors. Note the VAI estimstor. It is very consistent and has s steady mean error. It wonld be very useful if the bias could be directly linked to the bsud rate. Until this connection csn be msde it should be considered as s poor

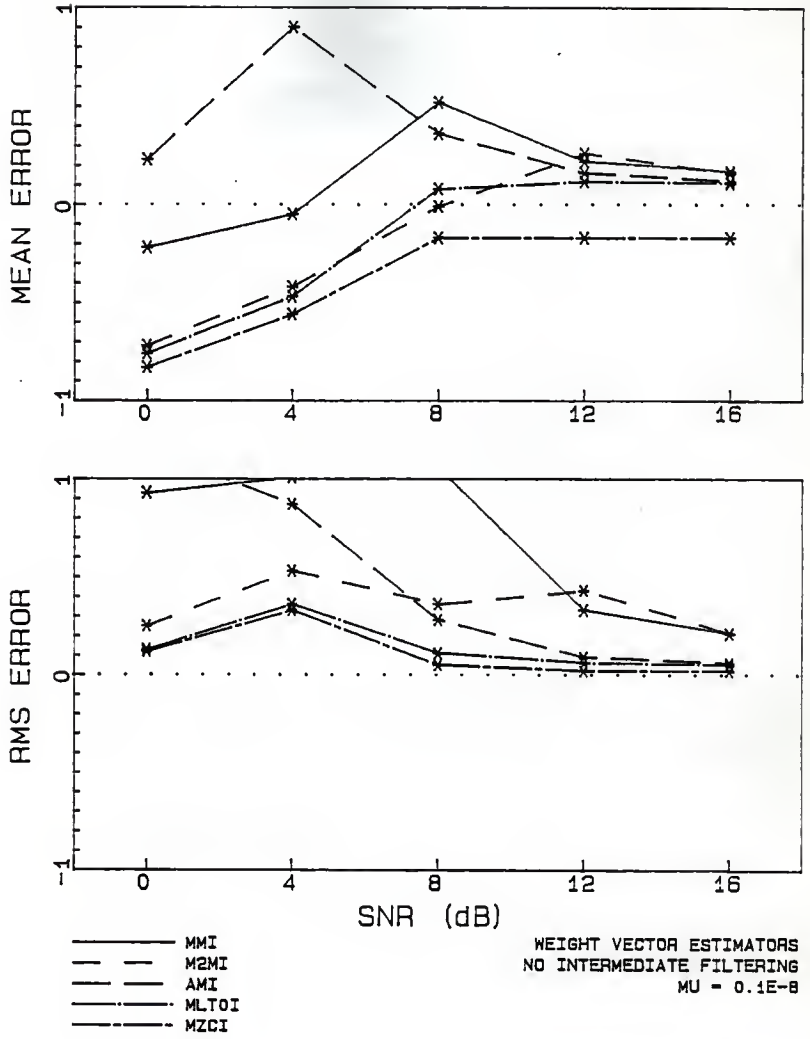


FIGURE 23a: Evaluation of estimators.

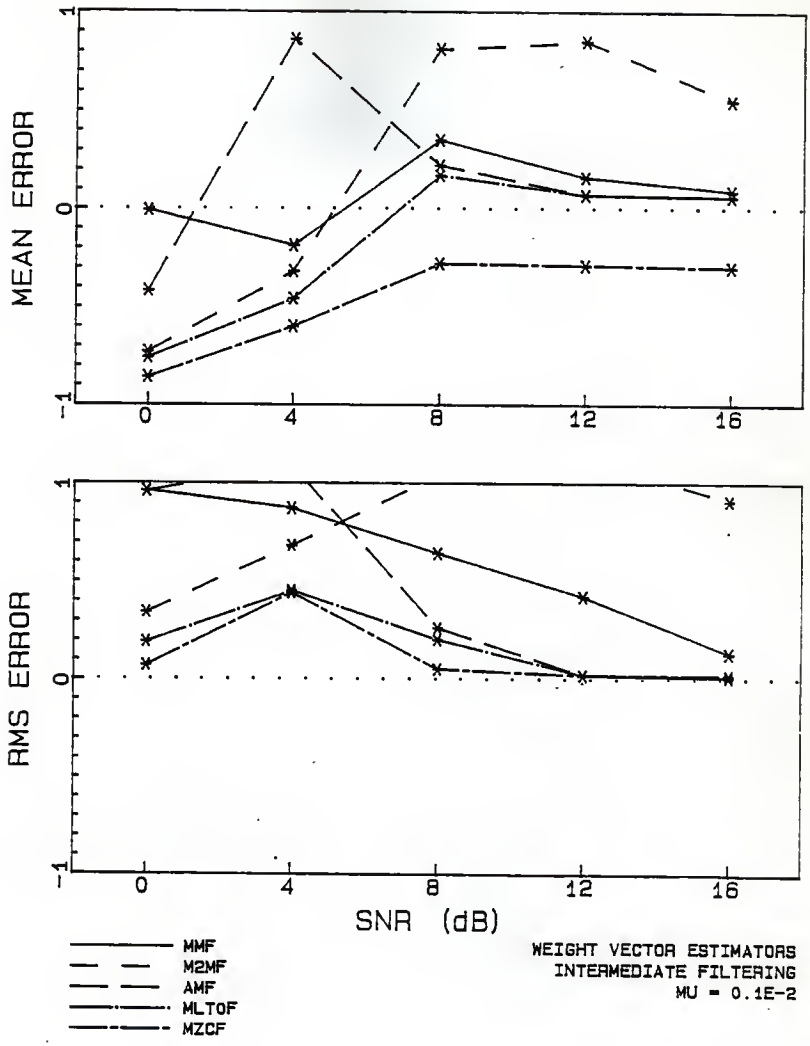


FIGURE 23b: Evaluation of estimators.

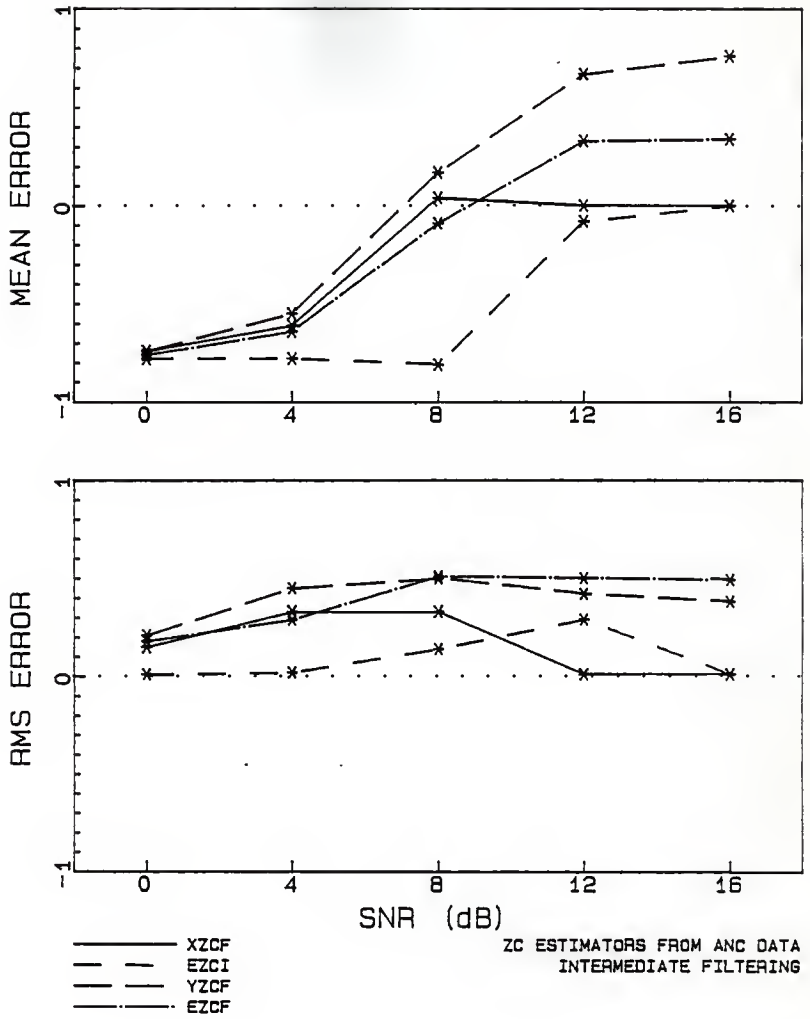


FIGURE 23c: Evaluation of estimators.

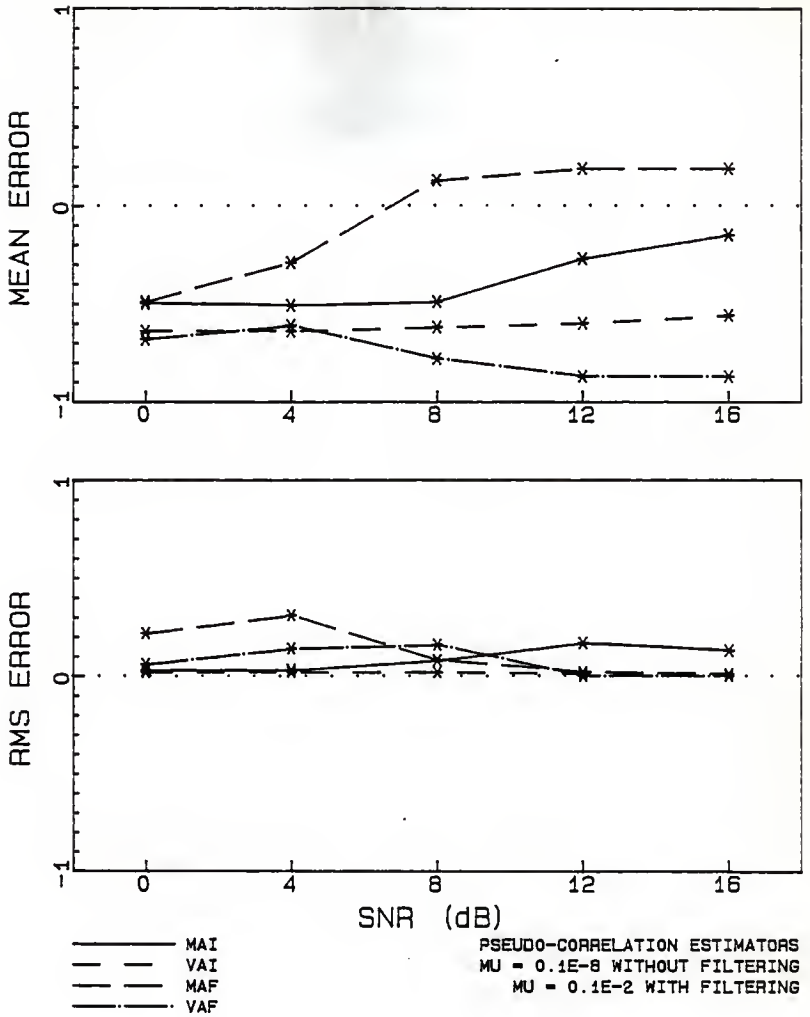


FIGURE 23d: Evaluation of estimators.

estimator.

Demodnlator Evalnation

The limiter and the post-filter were inclnded in the FSK demodnlator when the previons eatimator reanlts were obtained. It waa previonaly argned that this combination wonld provide the best resnlta. In an effort to anbatantiate thia, the different combinations of post-filter and limiter were naed in obtaining the eatimatea. Fignre 24a shows the resnlta for the XZCF estimator. Clearly the post-filter ahonld be inclnded. The preaence of the limiter also aeema to improve the resnlta.

Another way to evalnate the post-filter and limiter ia to obaerve the accnracy of the bandwidth eatimate when the post-filter and limiter combination ia varied. Fignre 24b showa these accnracy comparisons. Obvionaly the post-filter and limiter ahonld be inclnded.

Since the XZCF eatimator is critically dependent on the noiae redaction performed by the intermediate filter, it is also dependent on the bandwidth eatimate. The previons comparisons will be made for the MLTOI estimator. This estimator does not depend on the intermediate filtering. Fignre 24c gives the error reanlts for thia estimator nnder the aame aitnations. Once again the reanlta anpport inclnaion of the post-filter and limiter.

ANC Parameter Evalnation

Another parameter aet nsed on the basia of aasnmption and intnitive feeling is the ANC parameter set. It waa previonsly argned that the delay ahonld be 1 and the gradient constant, μ

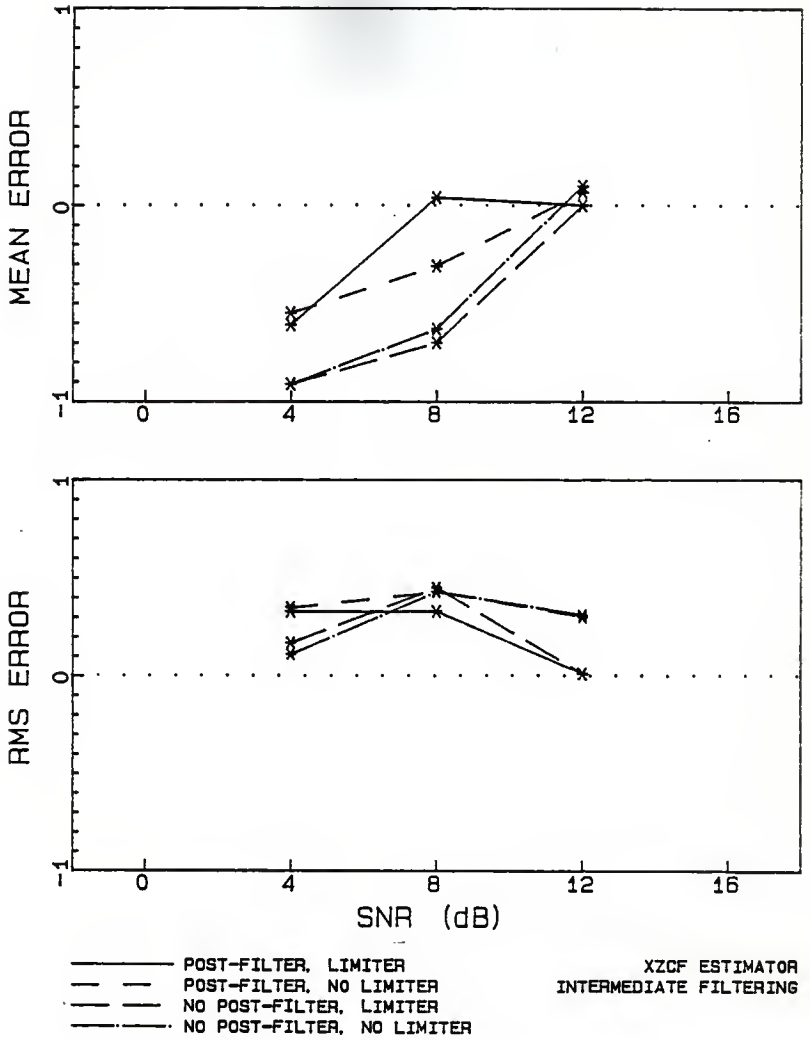


FIGURE 24a: Evaluation of FSK demodulator.

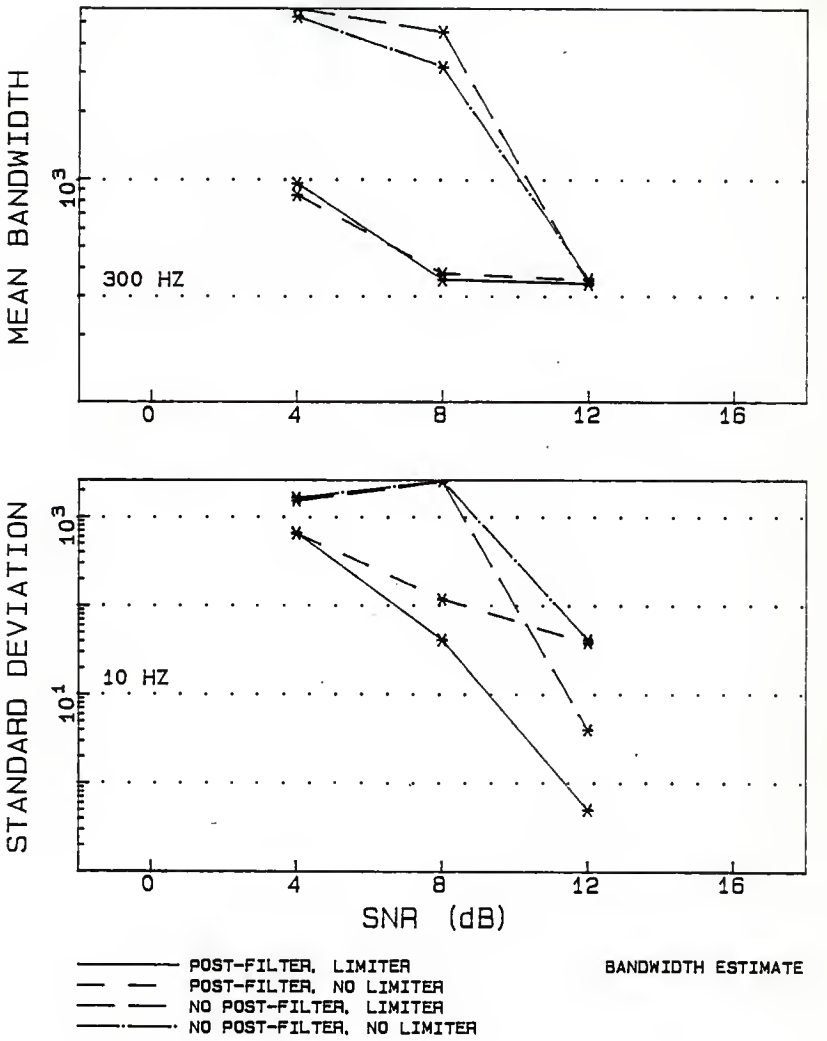


FIGURE 24b: Evaluation of FSK demodulator.

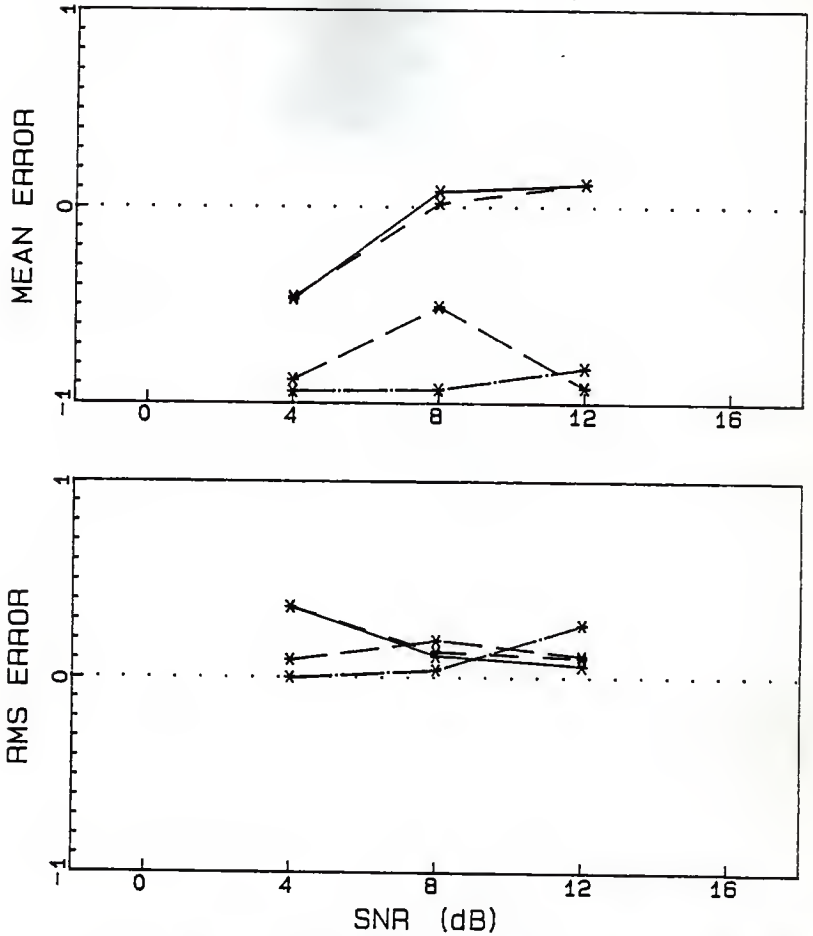


FIGURE 24c: Evaluation of FSK demodulator.

shonld be large. Table 2 presents the errors for a variety of combinationa of these two paramtera for a SNR of 8 dB. The freqnency deviation is 300 Hz and the baud rate is 300 bpa. The weight vector length in all casea ia 128. This might be unnecessarily long but will not be detrimental. The weight vector bias haa been incluaed since it, neither, will degrade the results. The estimates were made with the ANC following the intermediate filter. The ANC making the bandwidth estimate had μ set to 0.1E-8 and the delay as specified in the table. The reanlting estimate of filter bandwidth is apecified in the table.

ANC Paramters	Estimator					
	MLTOF		MZCF		Bandwidth	
	$\bar{e}$	e_{rma}	$\bar{e}$	e_{rms}	B	var
$\Delta = 1$ $\mu = 0.1E-2$	0.17	0.20	-0.28	0.05	352	41
$\Delta = 1$ $\mu = 0.1E-8$	0.41	0.21	-0.15	0.07		
$\Delta = 8$ $\mu = 0.1E-2$	0.17	0.16	-0.25	0.05	385	76
$\Delta = 8$ $\mu = 0.1E-8$	0.44	0.37	-0.06	0.47		

Table 2: Evaluation of ANC parameters

The reason for nsing these two estimatora is that they are the better of the claaa as noted in the estimator evaluation. For both estimators the different delays give nearly the aame reanlts when μ is large. When it is amall, the only distinct

difference is that the rms error is worse for the delay of 8 for the MZCF estimator. As for the bandwidth estimate, the delay should be 1 in order to obtain the better estimate.

The bandwidth estimate gives better results for the shorter delay. Hence the delay is best set when it is one. This will also eliminate the possible loss of correlation when the baud rate is large. The problem results from the delay extending past the last bit difference for which correlation exists.

One explanation for the similarities of results is that the gradient constant might not have been large enough to make a distinct difference. Perhaps it can be made larger by sacrificing weight vector length in an attempt for better results. As for a decision, μ will be left at the large value of 0.1E-2 in keeping with the philosophy that this will provide better cancellation.

Varying Frequency Deviation

All of the previous results have been generated for a frequency deviation of 300 Hz. This is the worst case possible since the band rate is also 300 bps. Therefore, results should improve when the deviation is increased and the band rate is kept constant. Figures 25a and 25b substantiate this conclusion.

In Figure 25a, the results for the XZCF estimator and the MLTOI estimator are presented for the cases of Δf equal to 300 Hz and 2000 Hz. According to the mean error, both estimators do a better job for the larger deviation at lower SNR. However, the rms error has become slightly worse for the XZCF estimator and

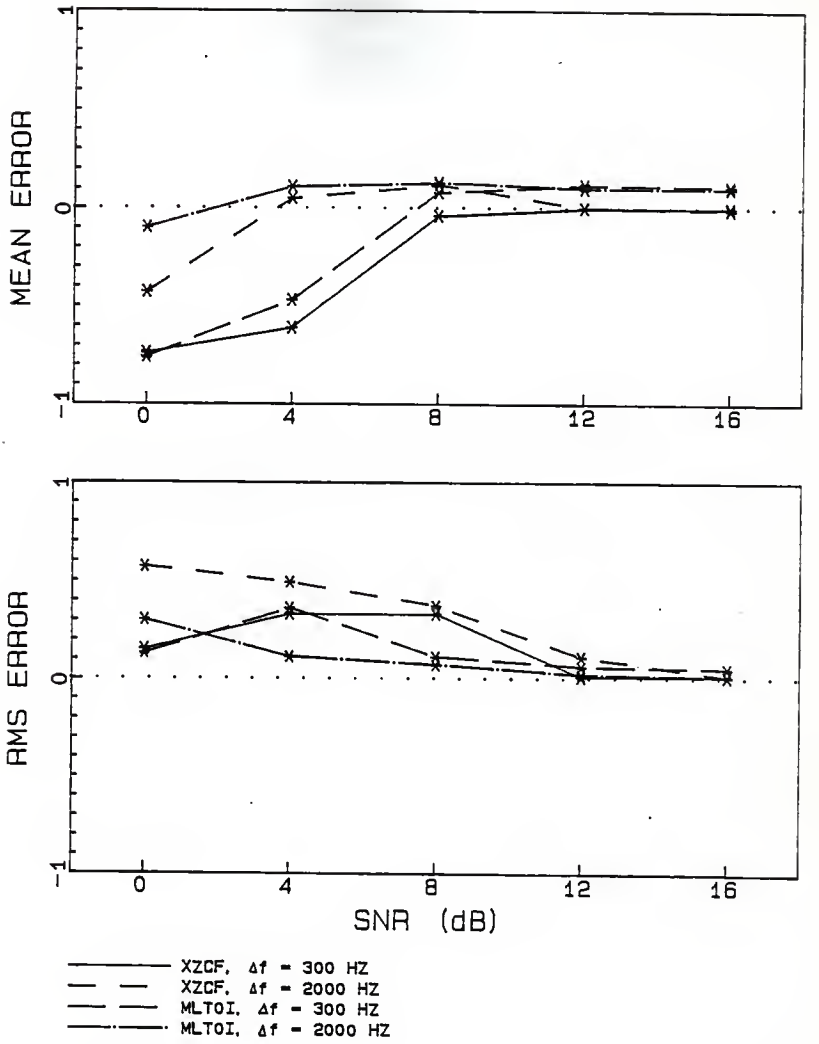
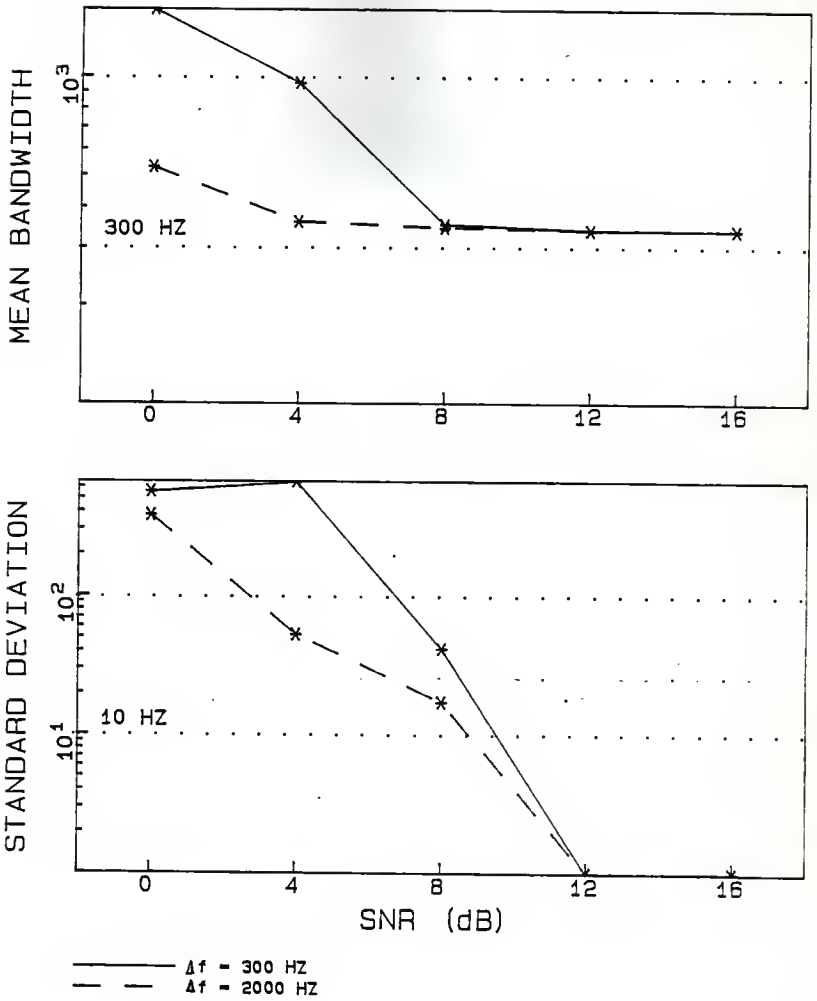


FIGURE 25a: Effect of varying frequency deviation of the FSK signal on the estimators.



R = 300 Hz
f_s = 9600 Hz

DELAY = 1
MU = 0.1E-8
N = 128
BIAS

FIGURE 25b: Effect of varying frequency deviation of the FSK signal on the bandwidth.

alightly better for the MLTOI estimator. This ususpected degradatiou for the XZCF eatimator is unexplaiuable.

Figure 25b showa how the bandwidth estimate is affeeted by the freqeueey deviation of the FSK sigual. The eatimate is substantially better for the larger deviation.

Different Baud Rates

Aa stated in the introduetiou the design should work for baud rates up to 1000 bps. Therefore, the resulta of the eatimators at differeut baud ratea should be evaluated. Figures 26a-e give the resulta of thia evaluation.

The estimates of the XZCF estimator made for three differeut baud rates are plotted in Figure 26a. The three baud rates are 150, 300, and 600 bps. Sinee the deviatiou muat be at least as large as the baud rate, it waa set to 2000 Hz for eaeh ease. The eouelusion to make is that the XZCF estimator performs well for a variety of baud rates dowu to a SNR of 8 dB. The 150 bps baud rate haa a signifieantly smaller rms error at 8 dB whieh perhaps iudieates the estimator will work better for lower baud rates. This is agreeable with the manner that the zero erossing aualysis is performed and with the way that errors may be introduced.

The MLTOI estimator also performs slightly better at lower SNR for the lower baud ratea as evideneed by Figure 26b. There ia a treud iu the mean error to have an iuereasing bias as the baud rate iuereases. The rma error follows a similar inereasing pattern although not as aeverely as the mean error.

One explanation for the eatimators performing better at

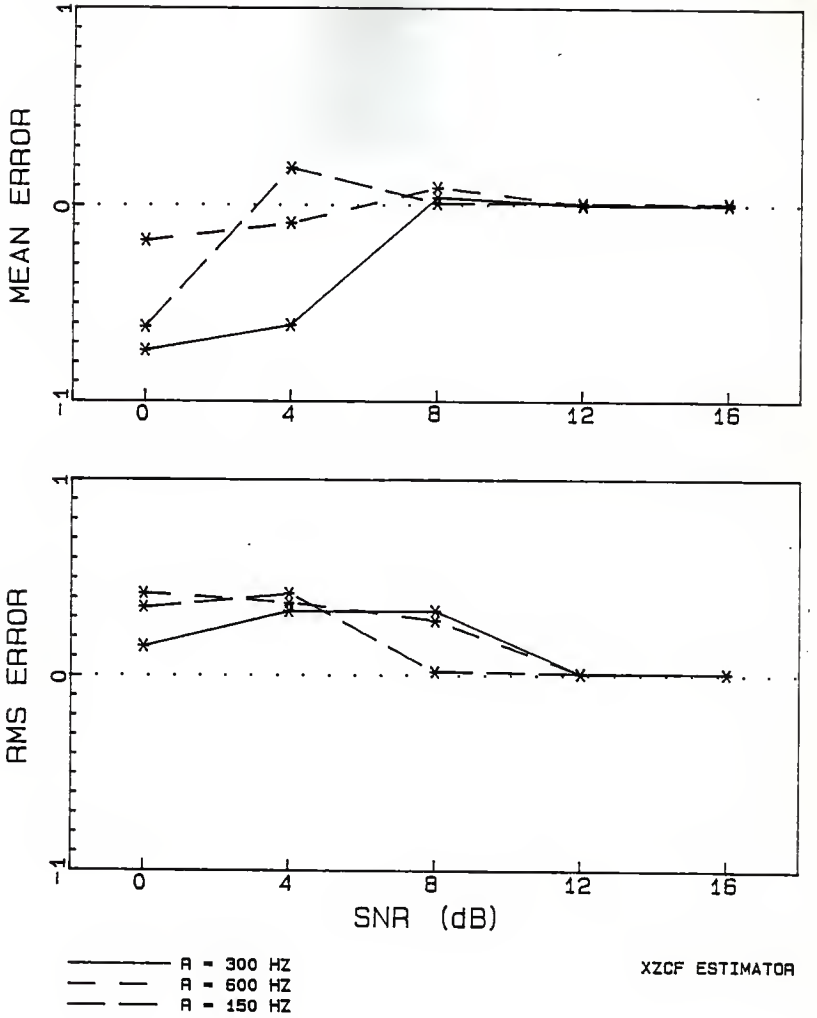


FIGURE 26a: Effect of varying the baud rate on the XZCF estimator.

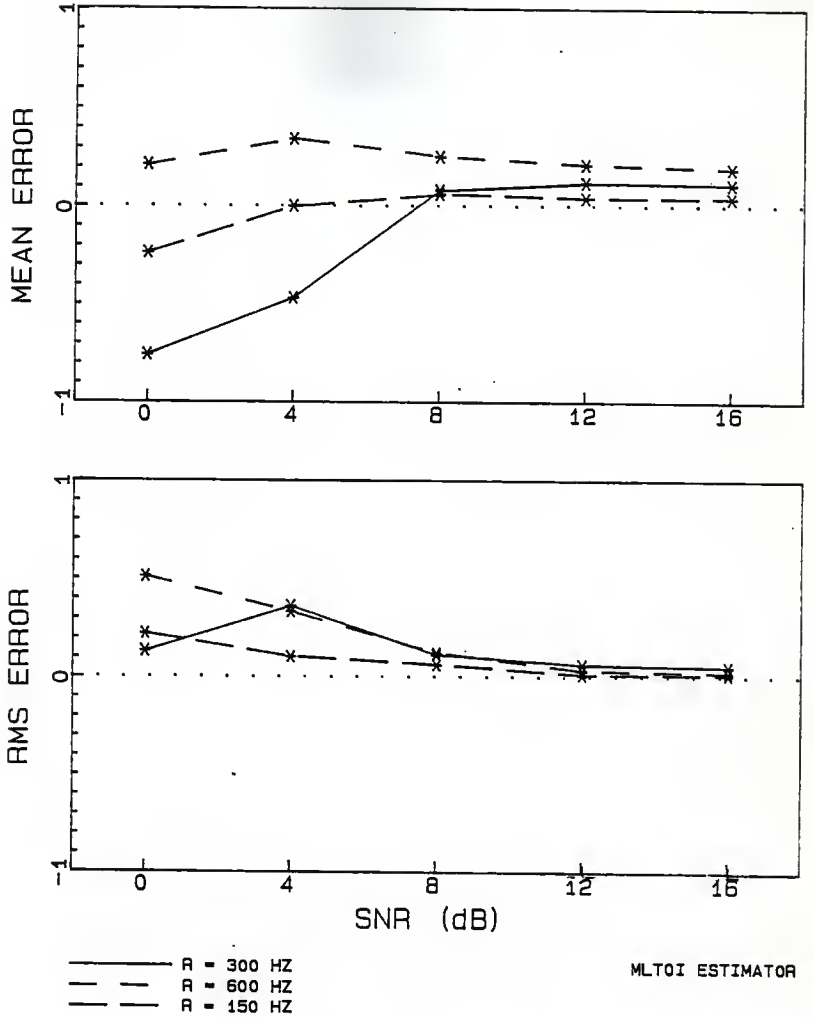


FIGURE 26b: Effect of varying the baud rate on the MLTOI estimator.

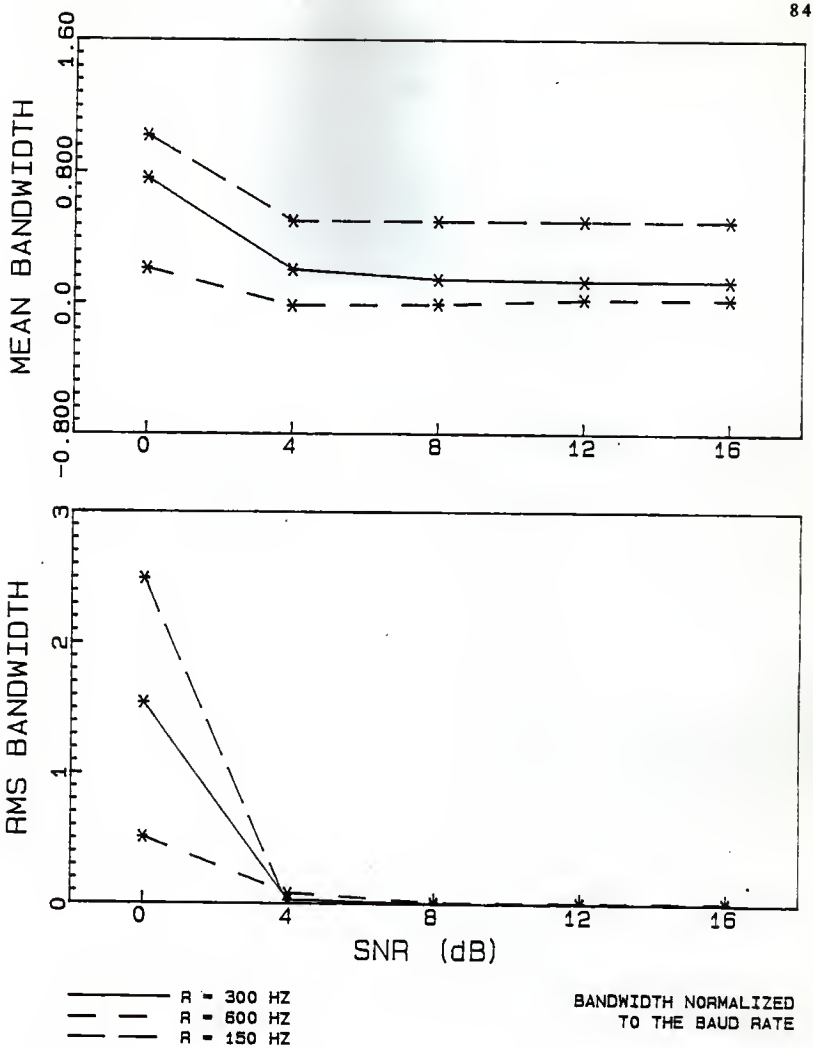


FIGURE 26c: Effect of varying the baud rate on the bandwidth estimate.

lower baud rates is that the sampling frequency is too low. If the sampling freqnency ia larger, more samplea per bit wonld be obaerved. This would have a direct effect on any eatimator from the weight vector class of which the MLTOI eatimator is a member. It wonld also affect the zero crossing class.

The resnlts of the bandwidth eatimate are ahown in Fignre 26c. The tendency this time ia for the bias to increase as the band rate decreases. This shows that the bandwidth estimate made in thia way would not be a good baud rate estimate. However, the purpose of the intermediate filter is to reduce the noiae power which the poat-filter of the DLD let pass. The flnctuating bias shonld not matter since in all casea the resulting bandwidth is a aignificant improvement over the 1000 Hz bandwidth of the post-filter, e.g., the bandwidth eatimate for the 150 bps band rate case is 225 Hz. The best feature of the bandwidth eatimator is that it is very conaistent. Aa long aa the SNR ia 4 dB or more, the intermediate filter will rednce noise power in a consistent manner.

Evaluation Against the Traditional Method

The traditional method for determining the band rate of an FSK signal is the nae of zero crossing methods. Therefore the propoaed method ahonld be compared to this traditional method to determine if the propoaed method haa merit. Note that the only difference between the XZCF estimator and the traditional method is the inclusion of an intermediate filter to further reduce the noiae power. The bandwidth of the filter is set nsing the ANC.

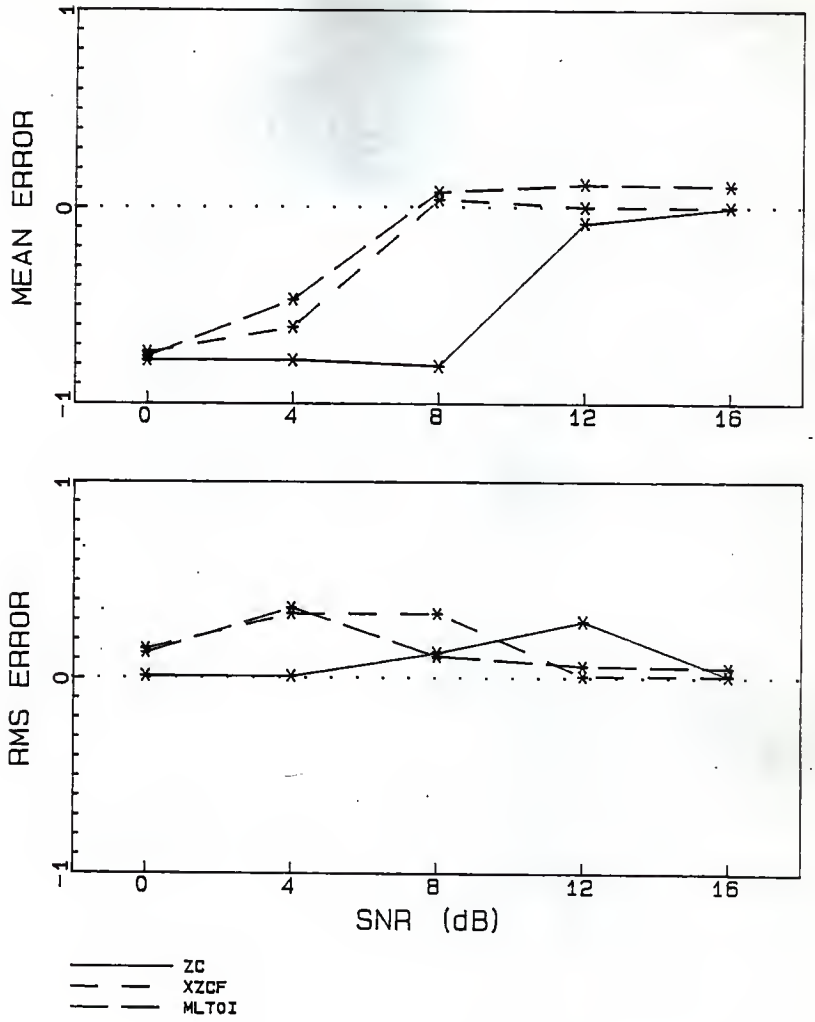
In order to meke the jndgement, the resnlts of these two methods elong with the resnlte nsing the MLTOI estimator are plotted in Fignres 27e-d for varions bend ratea and signal deviatione.

In Fignre 27e the bend rete is 300 bps end the deviation is 300 Hz. The eccnracy of the treditional method drops somewhere ebove 12 dB while both of the proposed methods ere good down to neerly 8 dB.

The bend rate in Fignre 27b ie 600 bps end the deviation is 2000 Hz. This time the MLTOI eatimator is the pooreet eatimator. The ZC and the XZCF eatimetor give neerly identicel resnlts. This conld be for two reesons. One, the eddition of the intermediate filter et ebont 600 Hz bandwidth did not meke e significent redнction in noise power. Two, the lerger bend rate means more semplea per bit time, and thne, e steedier zero crossing analysis cen be mede.

When the bend rete is decreesed to 150 bps with the same deviation, as in Fignre 27c, the XZCF eatimator doea indeed perform better then the treditional ZC estimetor. However, the MLTOI eetimetor performs the beat of the three. Its reenlts are good to 4 dB.

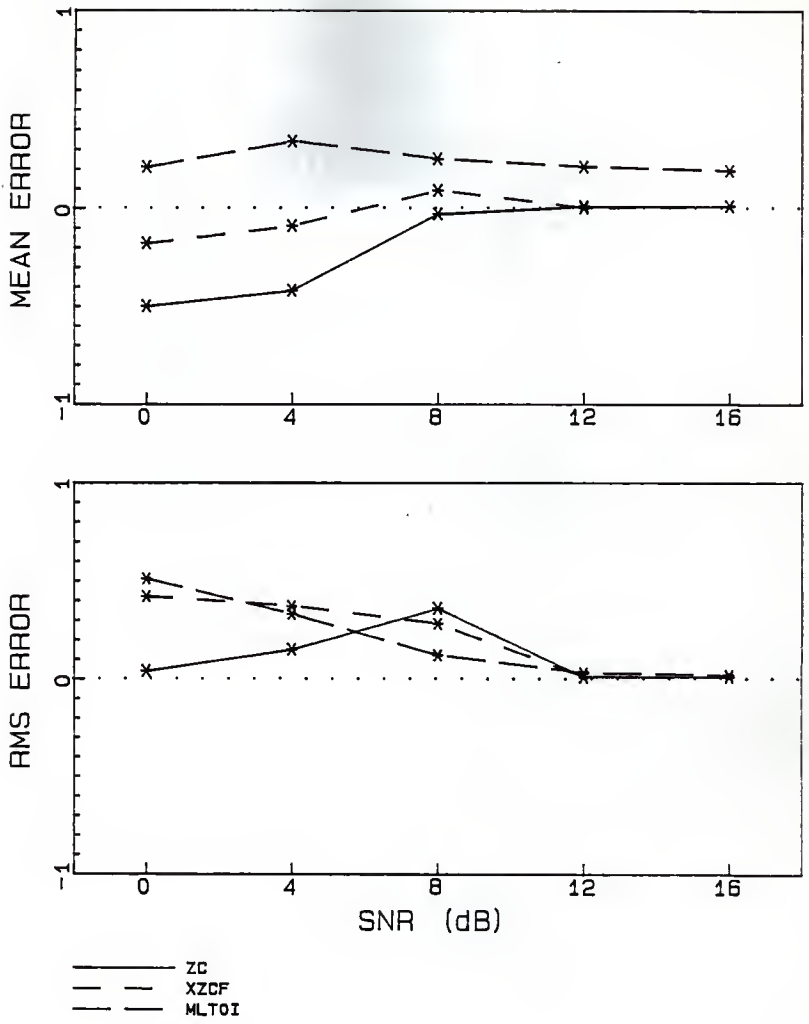
Fignre 27d prompts the same conclneions for a band rate of 300 bpa and a aignal devietion of 2000 Hz. This time, however, the MLTOI estimator contains a bies which ie nnexpleineble. Still, ita eatimetes ere relative accnrete to 4 dB. Agein, the traditional ZC method is nseleas below 12 dB.



R = 300 Hz
Δf = 300 Hz
f_s = 9600 Hz

DELAY = 1
MU = 0.1E-8
N = 128
BIAS

FIGURE 27a:. Evaluation of proposed estimators against the traditional zero-crossing estimator



R = 600 Hz
Δf = 2000 Hz
f_s = 9600 Hz

DELAY = 1
MU = 0.1E-8
N = 128
BIAS

FIGURE 27b: Evaluation of proposed estimators against the traditional zero-crossing estimator.

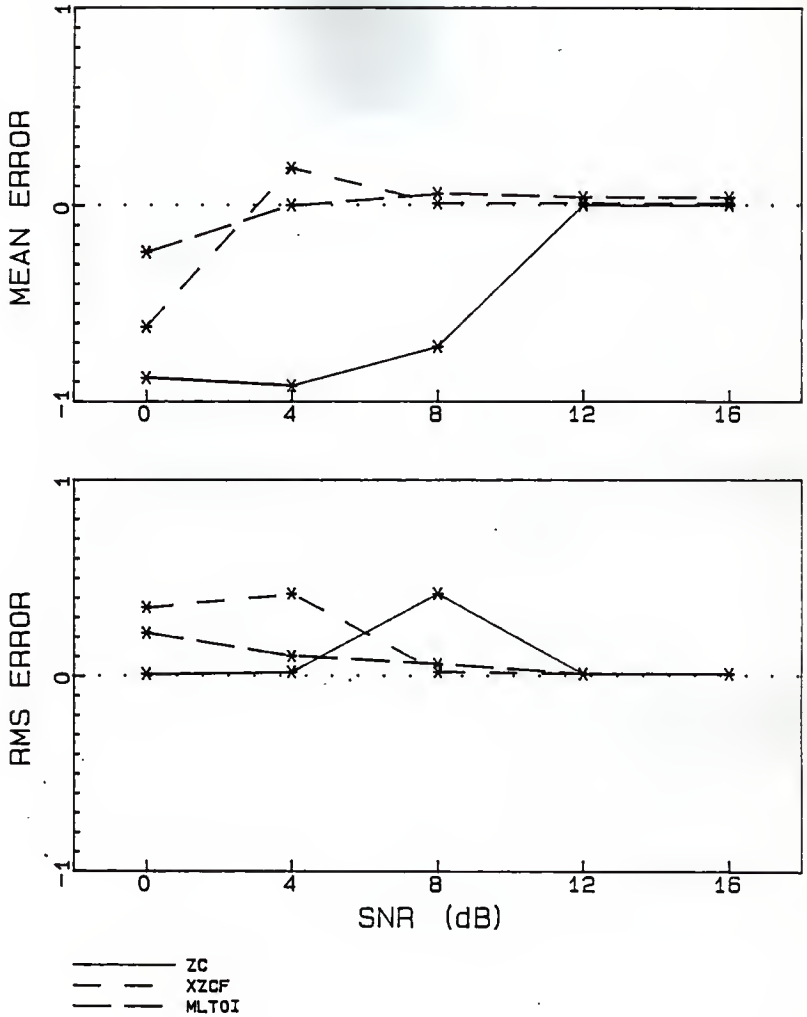


FIGURE 27c: Evaluation of proposed estimators against the traditional zero-crossing estimator.

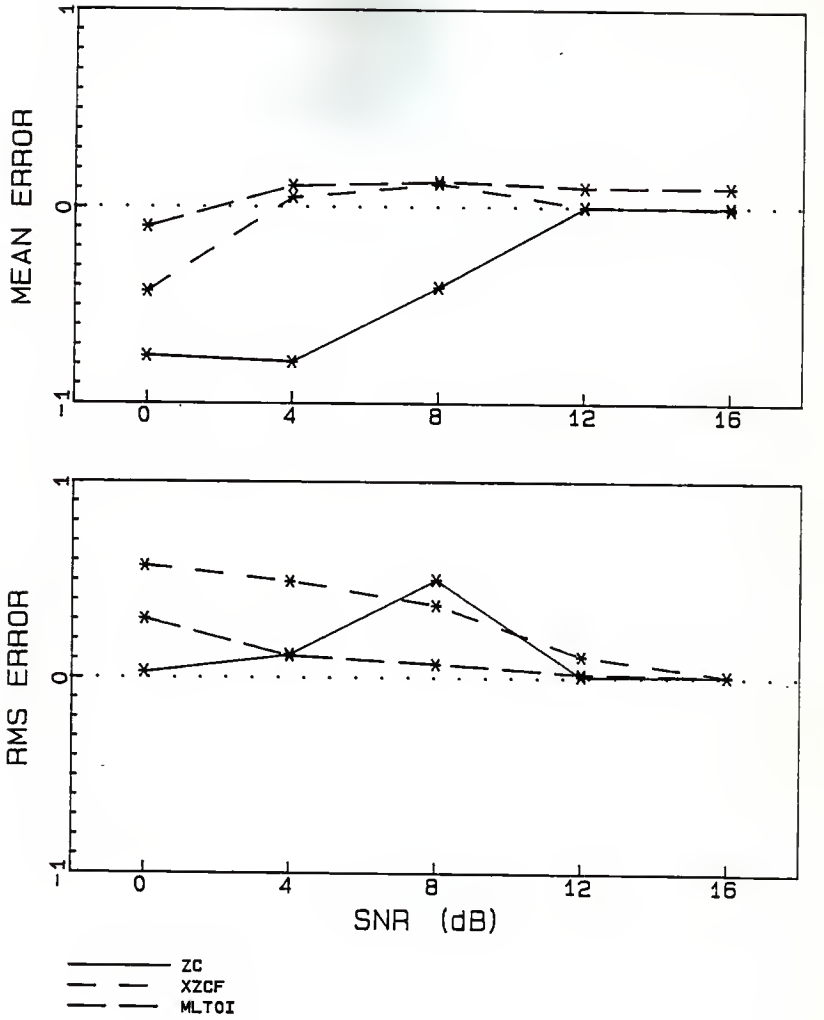


FIGURE 27d: Evaluation of proposed estimators against the traditional zero-crossing estimator.

Conclusions

Two classes of estimators proposed provide better results than the traditional zero crossing method. Both methods rely on the ANC, one directly and the other indirectly.

The weight vector class of estimators nses the mean and variance of the weight vectors of the iterative convergence procednre to obtain the baud rate. Of the estimators of this class, the MLTOI estimator was fonnd to be one of the better estimators. Fignre 28 shows how the MLTOI estimate is acquired. The estimate defines the bit time in nnmber of sampling intervals as being the delay of the ANC plus the nnmber of the weight which repreaents the first relative minimnm of the weight vector which is less than zero. The bit interval is then obtained from the weight number plus delay by multiplying by the sampling interval. Then the band rate is found by inverting the bit interval. The estimator gives good results over a large part of the desired range of band ratea and frequency deviations. Its estimates are reliable down to 8 dB in most instances. This eatimate performs better than the traditional method for nearly all baud rates and appreciably better for low baud rates.

The other class of estimators relies indirectly on the ANC. This time the ANC is used to obtain a bandwidth estimate of the measage. Then intermediate filtering is performed using the bandwidth estimate to fnrther reduce the noise power. The baud rate estimate is then obtained nsing zero crossing analysis just as the traditional method does. The analysis is performed on

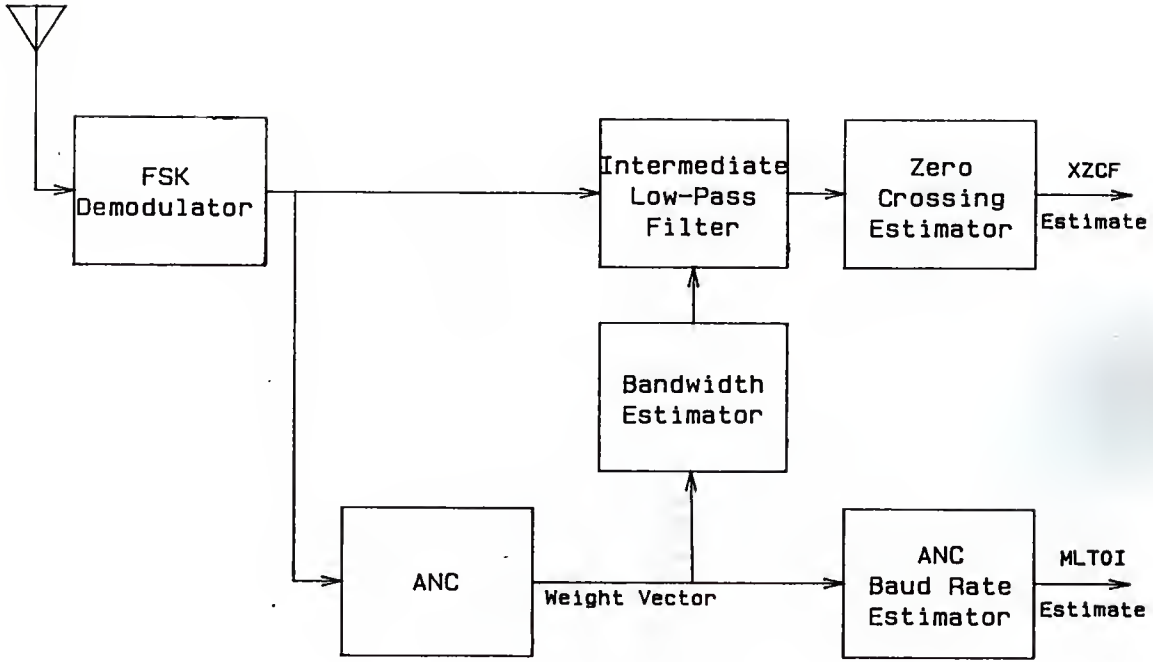


FIGURE 28: Final system for determining FSK baud rate.

either the filtered ANC input or on the varions ontputs from either one of the two ANC's. The beat resnlta are obtained for the XZCF eatimator whioh is when the zero crossing analysis ia performed on the intermediate filter ontpnt. This ia illnatrated in Fignre 28. Obvionsly, this estimate will perform at least as well as the traditional method. When the band rate drops to the lower end of the range, this eatimator performs much better than the traditional method. This ia a direct result of the additional noiae power cancellation attributed to good bandwidth estimation by the ANC. However, for the upper range of baud ratea and the lower range of frequency deviationa, this estimator will degrade just as faat as the traditional method when the SNR ia decreased.

The bandwidth estimation provided by the ANC should be conaidered aa a aignificant aohievement. The estimate rarely nndereatimates the bandwidth and the accnracy remaina atable down to 4 dB SNR. Obvionaly this addition to the system ia only valnable for the surveillance sitnation since the bandwidth wonld be known in other aplicationa.

One factor which definitely inflnences the estimators is the aampling freqnency. When aet relatively low in regards to the larger baud rates, the number of aamples per bit is small. This can create errors and biaaes in either clasa of estimators. The sampling frequency ahonld be set perhapa at least 20 times the largest expected band rate. Unfortunately, this means the weight vector mnst contain a large number of weights.

Another itsm which dsssrves sttsntion is the wsight vector. Ths wsight vsctor is snpposed to converge to s stsble weight vsctor which will prodncs sn srror ontpnt of ths ANC of the lsast powsr. In this asnse ths ADF filter wsights will hsve convsrged to ths optimsl Wisnsr filter [1]. An anslyaia of ths weight vsctor wonld bs bsneficial, sspscially in ths csae of s random bit inpnt ssqnsncs as modeled in the aimnlation. Once the snaly-sia is psrformsd it will bs possibls to dscids if ths pssndo-correlstion snb-class of eatimatora contsin merit, snd if they do, how wsll they will psrform in compsrison to ths other pro-possd sstimators.

Anothsr aspsct ths anslysis bs concernsd with is in ths method of noise (or signsl) csncelling. The method nssd hsre rsliss on ths ANC and csn be shown to bs in ths clsss of snto-rsgrsssivs prsdictors [5]. Thia clsss of sstimstors is known for s spectrnm which ia chsractsrizsd by pssks. The pesks rssnlt from a rationsl model for this sstimator involving sn all-pole transfsr fnnction. Howevsr, sincs ths apectrnm of the bit ssqnencs is charsctsrized more so by nnlls in ths freqnency domsin thsn by pssks, ths bit aeqnsnce shonld bs considsred ss a procsss other thsn sn snto-regrsssivs procsss. Therefors, it mskes ssnse to nse sn estimstor which is compstibls with nnlls, or in gsneral, nnlls snd pssks. The snto-rsgressivs, moving svsrsgs estimstor mssta this reqniremsnt sincs its rstionsl model is chsracterizsd with a trsnsfsr fnnction contsining both zeros and polss.

Appendix A: Computer Program Listings

```
C*****************************************************************************
C
C       SBAUD
C
C       VAX-11 FORTRAN SOURCE FILENAME:          SBAUD.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER(S)
C       --------        ----                     -------------
C       00.0            MAY 6, 1983              MARC D. BRACK
C
C*****************************************************************************
C
C       PURPOSE
C
C               This program is a simnlation for a system that estimates
C               the band rste of an FSK signal.  The aystem consiats of
C               a delay-line discriminstor (DLD) to demodnlate the signal
C               snd of an adaptive noise canceller (ANC) to make the
C               estimate.  The resnlts are compared to the resnlta of the
C               trsditional method of zero crossing eatimation.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               DLD     These rontines are specislized rontines written
C               SADF    only for the pnrpose of the simnlstion.  They
C               SSADF   are compatible with SBAUD and each other and are
C               ZCROSS  meant to be run in a aingle package.
C
C               SGOPEN  These rontines are general pnrpose rontines that
C               SGTRAN  perform snch functiona as writing to disk,
C               SPLOT   plotting, filtering, and making FFT'a.  Any other
C               BUTER   rontinea performing identicsl fnnctions msy be
C               DFT     snbatitnted for them.
C               GAUSS
C
C       ROUTINE(S) ACCESSED OR CALLED BY EMBEDDED ROUTINE(S)
C
C               BITLENGTH  This ia another specialized rontine.
C
C*****************************************************************************
C
C       IMPORTANT:  Weight vector length mnst be a power of two if
C                   the intermediate filter, the psendo-oorrelation
C                   eatimstes, or the freqnency responae of the weight
C                   vector dsts sre to be implemented.  The DFT rontine
C                   nsed in these applications reqnires a vector of
C                   length of a power of two.
C
C*****************************************************************************
C
```

```
C       Variablea naed in main rontine:
C
C       ADF     TRUE if ADF is to be inclnded in aimnlation
C
C       AI      Amplitnde of FSK signal envelope
C
C       ARG     Argument of FSK signal envelope
C
C       AUTO    TRUE if antocorrelation estimate ia enabled
C
C       BITS    Number of bits in mesaage
C
C       BN      Blank character for plotting pnrposes
C
C       BOTH    TRUE if two files are to be written to diak
C
C       BUTER   Complex bntterworth filter rontine
C
C       BW      Max bandwidth of aignal according to aampling freqency
C
C       C       Uaer reaponae aa Y or N
C
C       CHOICE  Pointer of directed GOTO atatementa
C
C       C1      Firat eatimate to be written
C
C       C2      Number of estimates to be written
C
C       DELAY   Number of samplea to delay in ANC
C
C       DW      FSK freqnency ahift in radiana
C
C       FC      Center freqnency of FSK aignal
C
C       FFT     TRUE if DFT is to be made of weight vector
C
C       FLAG    Indicates sin- or cosine-type discriminator
C
C       FO      Fnndamental freqnency of entire measage
C
C       FREQP   Max freqnency of freqnency reaponae to be plotted
C
C       FS      Sampling freqnency
C
C       FSK     TRUE if FSK aignal information is to be obtained
C
C       FW      FSK freqnency shift in Hertz
C
C       HT      Horizontal title
C
C       HU      Horiaontal units
C
```

```
C       I       Loop counter variable
C
C       II      Loop connter vsriable
C
C       ITER    Simulation loop connter varisble
C
C       J       Loop connter variable
C
C       LIM     TRUE if limiter is to be inclnded in sumulation
C
C       LTI     Left title
C
C       LUN     Left units
C
C       MAXF    Max of DFT of weight mesn or variance
C
C       MU      Gradient convergence rate constsnt
C
C       N       Number of samples in signal
C
C       NN      Inorement factor for plotting purposes
C
C       ND      Delay for DLD
C
C       NOISE   TRUE if noise is to be sdded in respective iteration
C
C       NSBWM   Normslized bsndwidth estimste mean
C
C       NSBWS   Normslized bandwidth standard devistion
C
C       OFILT   TRUE if intermediste filtering is to be done
C
C       P       Factor for either shifteing np or down in FSK signal
C
C       PI      The constsnt pi
C
C       PNUM    Number of points to plot
C
C       POSTA   TRUE if post-filtering is to be done for ANC
C
C       POSTBW  Post-filter bandwidth for ANC
C
C       POSTBZ  Post-filter bandwith for ZC analyisis (if POSTA false)
C
C       PRE     TRUE if pre-filtering is to be done
C
C       PREBW   Pre-filter bandwidth
C
C       R       Bsud rate
C
C       RTI     Right title
C
```

```
C       RUN     Right nnita
C
C       SANCZC  TRUE if ZC analysia is to be done in ANC
C
C       SBW     Bandwidth estimate
C
C       SBWM    Bandwidth estimate mean
C
C       SBWS    Bandwidth estimate mean sqnare
C
C       SBWV    Bandwidth estimate variance
C
C       SCORR   TRUE if psendo-correlation analysia is to be done
C
C       SMM     TRUE if intermediate estimatea are to be made
C
C       SNR     TRUE if noise is to be added to aignal
C
C       SNRDB   Signal-to-noiae ration in dB
C
C       SPB     Samples per bit
C
C       STITLE  Snb-title for plot
C
C       TB      Bit interval
C
C       TD      Time delay for discriminator
C
C       TF      Time interval for FSK signal
C
C       THETA   Phaae offaet of FSK aignal
C
C       TITLE   Major title for plot
C
C       TRIALS  Number of iterations for simnlation loop
C
C       TS      Sampling interval
C
C       VAR     Variance for noiae
C
C       WBIAS   TRUE if biaa is to be ionclnded in weight vector
C
C       WC      Center freqnency in radians per second
C
C       WINDOW  TRUE if window is to be nsed in DFT sitnations
C
C       WNUM    Number of weighta in weight vector
C
C       WRIT    TRUE if data ia to be written to disk
C
C       WINUM   Length of zero padded weight vector in DFT sitnationa
C
```

```
C       XSCALE  Scaling of abscissa valnes for plotting pnrposes
C
C       ZC      TRUE if traditional ZC anlysia is to be done
C
C*************************************************************************
C
C       Arrays nsed in the main rontine:
C
C       AUTOC   Used in obtaining antocorrelation
C
C       CNOISE  Noise to be added to signal
C
C       DLDNN   Outpnt of DLD for noiselesa inpnt
C
C       EROR    Error ontpnt of secondary ANC
C
C       ERORI   Error ontpnt of intermediate ANC
C
C       FILT    DLD ontpnt intended of tradistional zc analysis
C
C       FSIG    FSK signal magnitnde
C
C       LEFT    Left array for plotting
C
C       MDLD    DLD ontpnt, general case
C
C       NAUTO   Noise antocorrelation
C
C       PFSK    FSK signal phase
C
C       RIGHT   Right array for plotting
C
C       S       FSK envelope and general pnrpose complex array
C
C       SAUTO   Mesaage antocorrlation
C
C       SEQ     Bit aeqnence for of message
C
C       SFEST   Secondary band rate estimatea
C
C       SFESTM  Mean of secondary band rate eatimates
C
C       SFESTS  Mean sqnare of secondary band rate estimatea
C
C       SFESTV  Variance of secondary band rate eatimatea
C
C       SIEST   Intermediate band rate eatimates
C
C       SIESTM  Mean of intermediate band rate estimates
C
C       SIESTS  Mean sqnare of intermediate band rate estimatea
C
```

```
C       SIESTV  Variance of intermediate baud rate estimates
C
C       WMEAN   Weight vector mean
C
C       WMEANA  Average of all weight vector means
C
C       WVAR    Weight vector variance
C
C       WVARA   Average of all weight vecto variances
C
C       Y       ADF output of secondary ANC
C
C       YI      ADF output of intermediate ANC
C
C       ZEST    Traditional ZC baud rate estimate
C
C       ZESTM   Mean of traditional ZC baud rate estimate
C
C       ZESTS   Mean square of traditional ZC baud rate estimate
C
C       ZESTV   Variance of traditional ZC baud rate estimate
C
C*************************************************************************
C
        IMPLICIT NONE
        REAL THETA,BW,AI,FSIG(2048),FILT(2048),MDLD(2048),MU,ARG
        REAL TD,FC,TF,PREBW,POSTBW,PFSK(2048),FW,DW,R,FO,DLDNN(2048)
        REAL VAR,SNRDB,SAUTO(2048),NAUTO(2048),SBWV
        REAL WMEAN(0:256,200),WVAR(0:256,200),WMEANA(0:256),WVARA(0:256)
        REAL XSCALE,LEFT(2048),RIGHT(2048),SBW,SBWS,SBWM,MAXF
        REAL ZEST(10,200),SIEST(10,200), SFEST(10,200),POSTBZ,FREQP
        REAL ZESTM(10),SIESTM(10),SFESTM(10),NSBWM,NSBWS
        REAL ZESTS(10),SIESTS(10),SFESTS(10)
        REAL ZESTV(10),SIESTV(10),SFESTV(10)
        REAL EROR(2048),Y(2048),ERORI(2048),YI(2048)
        REAL*8 PI,WC,FS,TS,TB
        INTEGER SPB,P,I,J,SEQ(64),BITS,N,NN,FLAG,WNUM,DELAY
        INTEGER ITER,ND,CHOICE,TRIALS,PNUM,C1,C2,II,WINUM
        COMPLEX S(2048),BUTER,CNOISE(2048),AUTOC(2048)
        CHARACTER*80 BN,TITLE,LTI,RTI,STITLE,HT,HU,LUN,RUN
        CHARACTER C*1
        LOGICAL NOISE,WBIAS,AUTO,ZC,ADF,FSK,POSTA,WINDOW,SANCZC
        LOGICAL PRE,SNR,OFILT,LIM,SMM,BOTH,WRIT,FFT,SCORR

        NN = 1
        BN = ' '
        PI = 3.141592653589793D0
        N = 1024
        FS = 9600
        FO = FS/N
        TS = 1./FS
        BW = FS/2.
```

```
      FC = 500*FS
      WC = 2*PI*FC
      BITS = 32
      SPB = N/BITS
      TB = TS*SPB
      R = 1.0/TB
      AI = 1.0
      THETA = 0.0
      NOISE = .FALSE.
C
C*************************************************************************
C
C  PARAMETER INPUT
C
C  AUTOCORRELATION ESTIMATION ENABLE
C
      CALL PAUTO(AUTO)
C
C  FSK SIGNAL INFORMATION ENABLE
C
      CALL PFSKINFO(FSK)
C
C  SIMULATION ADAPTIVE FILTERING SUBROUTINE ENABLE
C
      CALL PADF(ADF)
C
C  ZERO CROSSING ALGORITHM ENABLE
C
      CALL PZCROSS(ZC)
C
C  INPUT SIGNAL PARAMETERS
C
C
      TYPE *,' '
      TYPE *, 'INPUT SIGNAL PARAMETERS:'
C
C  SEQUENCE FORMATION
C
      CALL PSEQUENCE(SEQ,BITS)
C
C  FSK PARAMETERS
C
      CALL PFSKSHIFT(FW)
      DW = 2.0*PI*FW
C
C  NOISE SELECTION
C
      CALL PNOISE(SNRDB,SNR)
C
C  DELAY LINE DESCRIMINATOR PARAMETERS
C
      TYPE *, ' '
```

```
      TYPE *, 'DLD RECEIVER PARAMETERS:'
C
C  FILTER SELECTION
C
      CALL PPREFILTER(PREBW,PRE)
      CALL PPOSTFILTER(POSTBW,POSTA,ZC,POSTBZ)
C
C  LIMITER ENABLE
C
      CALL PLIM(LIM)
C
C  DLD SHIFT PARAMETER
C
C     CALL PDELAY(ND)
      ND = INT(FC/(6*PREBW))
C
C  ADAPTIVE FILTER PARAMETERS
C
      IF(ADF) THEN
      TYPE *, ' '
      TYPE *, 'ADAPTIVE FILTERING PARAMETERS:'
C
C  ADF DELAY SELECTION
C
      CALL PADFDELAY(DELAY)
C
C  NUMBER OF ADF WEIGHTS SELECTION
C
      CALL PWEIGHTS(WNUM)
C
C  GRADIENT CONSTANT SELECTION
C
      CALL PCONSTANT(MU)
C
C  WEIGHTING BIAS SELECTION
C
      CALL PBIAS(WBIAS)

      TYPE *, ' '
      TYPE *, 'ADF ESTIMATOR SELECTION:'
C
C  INTERMEDIATE FILTERING ENABLE
C
      CALL PFILT1(OFILT)
C
C  MAX/MIN ESTIMATE ENABLE
C
      CALL PMM(SMM)
C
C  PSEUDO-CORRELATION ESTIMATE ENABLE
C
      CALL PCORR(SCORR)
```

```
C
C  ANC-ZC ESTIAMTE ENABLE
C
        CALL PANCZC(SANCZC)
C
        ENDIF
C
C  NUMBER OF TRIALS SELECTION
C
        CALL PTRIALS(TRIALS)
C
C*******************************************************************************
C
C  RUN INITIALIZATION
C
1000    CONTINUE

        SBWM = 0
        SBWS = 0
        NSBWM = 0
        NSBWS = 0
        DO 140 I=1,10
          SIESTM(I) = 0
          SIESTS(I) = 0
          ZESTM(I) = 0
          ZESTS(I) = 0
          SFESTM(I) = 0
          SFESTS(I) = 0
140     CONTINUE

        NOISE = .FALSE.

        DO 150 J=0,WNUM
          WMEANA(J) = 0.
          WVARA(J) = 0.
150     CONTINUE

        IF (AUTO) THEN
          DO 160 I=1,N
160         NAUTO(I) = 0.
        ENDIF
C
C*******************************************************************************
C
C  SIMULATION DO LOOP
C
        DO 1500 ITER=1,TRIALS

        TYPE *, ' '
        TYPE *, 'IN ITERATION NUMBER ',ITER

1100    CONTINUE
```

```
C
C*****************************
C
C  FSK GENERATION
C
        DO 180 I=1,BITS

          IF (SEQ(I).EQ.1) THEN
            P = 1
          ELSE
            P = -1
          ENDIF
C
C  MODULATING FUNCTION
C
          DO 170 J=1,SPB
            S((I-1)*SPB+J) = AI*CDEXP(DCMPLX(0.0D0,P*((I-1)*SPB+J-1)*TS*DW))
170       CONTINUE

180     CONTINUE

C
C*****************************
C
C  NOISE ADDITION
C
        IF (NOISE) THEN
          CALL SNOISE(CNOISE,N,AI,TS,PREBW,SNRDB,SNR)
          DO 190 I=1,N
190         S(I) = S(I) + CNOISE(I)
        ENDIF
C
C  OBTAIN FREQUENCY COMPONENTS OF FSK
C
        IF(FSK) THEN
          CALL DFT(S,N,0)
          DO 200 I=1,N
            FSIG(I) = CABS(S(I))
200       CONTINUE
          CALL DFT(S,N,1)
C
C  OBTAIN PHASE OF FSK ENVELOPE
C
          DO 210 I=1,N
210         PFSK(I) = ATAN2(AIMAG(S(I)),REAL(S(I)))/PI
        ENDIF
C
C*****************************
C
C  DELAY-LINE DISCRIMINATOR
C
```

```
C
C   PRE-FILTERING
C
        IF (PRE) THEN
          CALL DFT(S,N,0)
          DO 300 I=1,N/2
            S(I) = S(I)*BUTER(4,PREBW,FLOAT(I-1)*FO)
            S(N+1-I) = S(N+1-I)*BUTER(4,PREBW,FLOAT(-I)*FO)
300       CONTINUE
          S(N/2+1) = CMPLX(0.,0.)
          CALL DFT(S,N,1)
        ENDIF
C
C  LIMITER
C
        IF (LIM) THEN
          DO 305 I=1,N
            IF(.NOT.(REAL(S(I)).EQ.0 .AND. AIMAG(S(I)).EQ.0))THEN
              ARG = ATAN2(AIMAG(S(I)),REAL(S(I)))
            ELSE
              ARG = 0
            ENDIF
            S(I) = AI*CMPLX(COS(ARG),SIN(ARG))
305       CONTINUE
        ENDIF
C
C  DLD
C
        TD = ND/(2.0*FC)
        FLAG = 1
        TF = N*TS

        CALL DLD(TD,FC,FLAG,S,N,TF)
C
C  POST-FILTERING FOR ADAPTIVE FILTERING
C
        IF (POSTA) THEN
          CALL DFT(S,N,0)
          DO 310 I=1,N/2
            S(I) = S(I)*BUTER(4,POSTBW,FLOAT(I-1)*FO)
            S(N+1-I) = S(N+1-I)*BUTER(4,POSTBW,FLOAT(-I)*FO)
310       CONTINUE
          S(N/2+1) = CMPLX(0.,0.)
          CALL DFT(S,N,1)
        ENDIF
C
C*****************************
C
C  INITIAL NOISELESS CASE
C
        IF (.NOT.NOISE.AND.SNR) THEN
```

```
      NOISE = .TRUE.
        DO 400 I=1,N
400       DLDNN(I) = REAL(S(I))
C
C  MESSAGE AUTOCORREALATION
C
        IF (AUTO) THEN
          DO 460 I=1,N
460         AUTOC(I) = CMPLX(DLDNN(I),0.)
          CALL DFT(AUTOC,N,0)

          DO 470 I=1,N
470         AUTOC(I) = CMPLX(CABS(AUTOC(I))**2,0.)
          CALL DFT(AUTOC,N,1)

          DO 480 I=1,N
480         SAUTO(I) = CABS(AUTOC(I))
        ENDIF

        GOTO 1100
      ENDIF
C
C*****************************
C
C  NOISE AUTOCORRELATION ESTIMATE
C
      IF (AUTO) THEN
        DO 410 I=1,N
410       AUTOC(I) = CMPLX(REAL(S(I))-DLDNN(I),0.)
        CALL DFT(AUTOC,N,0)

        DO 420 I=1,N
420       AUTOC(I) = CMPLX(CABS(AUTOC(I))**2,0.)
        CALL DFT(AUTOC,N,1)

        DO 430 I=1,N
430       NAUTO(I) = NAUTO(I) + CABS(AUTOC(I))/TRIALS
      ENDIF

      DO 440 I=1,N
440     MDLD(I) = REAL(S(I))
C
C  PERFORM ZERO CROSSING ESTIMATE, IF NO ADF POST FILTER, INCLUDE FILTERING
C
      IF(ZC) THEN
        IF(.NOT.POSTA) THEN
          DO 441 I=1,N
441         CNOISE(I) = S(I)
          CALL DFT(CNOISE,N,0)
          DO 442 I=1,N/2
            CNOISE(I) = CNOISE(I)*BUTER(4,POSTBZ,FLOAT(I-1)*F0)
            CNOISE(N+1-I) = CNOISE(N+1-I)*BUTER(4,POSTBZ,FLOAT(-I)*F0)
```

```
442           CONTINUE
              CNOISE(N/2+1) = CMPLX(0.,0.)
              CALL DFT(CNOISE,N,1)

              DO 443 I=1,N
443             FILT(I) = REAL(CNOISE(I))

              CALL ZCROSS(FILT,N,ZEST(1,ITER),1)
            ELSE
              CALL ZCROSS(MDLD,N,ZEST(1,ITER),1)
            ENDIF
          ENDIF
C
C*******************************
C
C  ADAPTIVE NOISE FILTERING SUBROUTINE
C
        IF (ADF) THEN

          CALL SSADF(MDLD,N,MU,WNUM,DELAY,WBIAS,WMEAN(0,ITER),WVAR(0,ITER),
     X    OFILT,SFEST(1,ITER),TS,SIEST(1,ITER),SBW,SMM,ERORI,YI,
     X    EROR,Y,SCORR,SANCZC)
C
C  BANDWIDTH ESTIMATE
C
          IF(OFILT) THEN
            SBWM = SBWM + SBW/TRIALS
            SBWS = SBWS + SBW**2/TRIALS
            NSBWM = NSBWM + ((SBW - R)/R)/TRIALS
            NSBWS = NSBWS + ((SBW - R)/R)**2/TRIALS
          ENDIF
C
C  BAUD RATE ESTIMATE STATISTICAL UPDATE
C
          IF(SMM) THEN
            DO 444 I=1,10
              SIESTM(I) = SIESTM(I) + ((SIEST(I,ITER)-SPB)/SPB)/TRIALS
              SIESTS(I) = SIESTS(I) + ((SIEST(I,ITER)-SPB)/SPB)**2/TRIALS
444         CONTINUE
          ENDIF
          IF (OFILT) THEN
            DO 445 I=1,10
              SFESTM(I) = SFESTM(I) + ((SFEST(I,ITER)-SPB)/SPB)/TRIALS
              SFESTS(I) = SFESTS(I) + ((SFEST(I,ITER)-SPB)/SPB)**2/TRIALS
445         CONTINUE
          ENDIF

          DO 450 I=0,WNUM
            WMEANA(I) = WMEANA(I) + WMEAN(I,ITER)/TRIALS
            WVARA(I) = WVARA(I) + WVAR(I,ITER)/TRIALS
450       CONTINUE
```

```
        ENDIF

        IF(ZC) THEN
          DO 455 I=1,1
            ZESTM(I) = ZESTM(I) + ((ZEST(I,ITER)-SPB)/SPB)/TRIALS
            ZESTS(I) = ZESTS(I) + ((ZEST(I,ITER)-SPB)/SPB)**2/TRIALS
455       CONTINUE
        ENDIF

1500    CONTINUE
C
C  END OF SIMULATION LOOP
C
C*************************************************************************
C
C  OBTAIN VARIANCE OF STATISTICS
C
        IF (ADF) THEN
          IF(OFILT) THEN
            SBWV = SBWS - SBWM**2
            IF(SBWV.GE.0.) SBWV = SBWV**0.5
            NSBWS = NSBWS**0.5
          ENDIF
          IF (SMM) THEN
            DO 490 I=1,10
              SIESTV(I) = SIESTS(I) - SIESTM(I)**2
              IF(SIESTV(I).GE.0.0) SIESTV(I) = SIESTV(I)**0.5
490         CONTINUE
          ENDIF
          IF(OFILT) THEN
            DO 495 I=1,10
              SFESTV(I) = SFESTS(I) - SFESTM(I)**2
              IF(SFESTV(I).GE.0.0) SFESTV(I) = SFESTV(I)**0.5
495         CONTINUE
          ENDIF
        ENDIF

        IF (ZC) THEN
          DO 500 I=1,1
            ZESTV(I) = ZESTS(I) - ZESTM(I)**2
            IF(ZESTV(I).GE.0.0) ZESTV(I) = ZESTV(I)**0.5
500       CONTINUE
        ENDIF
C
C*************************************************************************
C
C  PLOTTING AND MODIFICATION
C
        CALL PTITLE(TITLE,PRE,POSTA,FW,PREBW,POSTBW,SNRDB,SNR,R,LIM)
        IF (WBIAS) THEN
          WRITE(STITLE,14) 'ADF-DELAY= ',DELAY,', N=',WNUM,', MU=',MU,', BIAS'
        ELSE
```

```
          WRITE(STITLE,14) 'ADF-DELAY= ',DELAY,', N=',WNUM,', MU=',MU
        ENDIF

2000    TYPE *, ' '
        TYPE 22
22      FORMAT(' Do you want to:',/
     x        , '   1.  Plot',/
     x        , '   2.  Change parameters',/
     x        , '   3.  Enter user controlled ADF routine',/
     x        , '   4.  Write results',/
     x        , '   5.  Quit',/
     x        , '   6.  Write output to disk')
        TYPE 11, '/Enter selection:   '
        READ (*,*) CHOICE

        WRIT = .FALSE.
        IF(CHOICE.EQ.6) WRIT = .TRUE.

        GOTO (5000,6000,7000,8000,9000,5000) CHOICE
C
C*************************************************************************
C
C  PLOTTING AND WRITING TO DISK
C
5000    TYPE *, ' '
        TYPE 23
23      FORMAT(' Do you want to plot(write):',/
     x        , '   1.  DLD output',/
     x        , '   2.  Frequency components of FSK signal',/
     x        , '   3.  Phase of FSK envelope',/
     x        , '   4.  Autocorrelation estimates',/
     x        , '   5.  Weight mean and variance data',/
     x        , '   6.  Intermediate ANC error',/
     x        , '   7.  Intermediate ADF output',/
     x        , '   8.  Secondary ANC input',/
     x        , '   9.  Secondary ANC error',/
     x        , '  10.  Secondary ADF output',/
     x        , '  11.  Or do you want to return to main menu')
        TYPE 11, '/Enter selection:   '
        READ(*,*) CHOICE

        GOTO(5100,5200,5300,5400,5500,5600,5700,5750,5800,5900,2000) CHOICE
C
C*****************************
C
C  DLD OUTPUT
C
5100    DO 5110 I=1,N
5110      LEFT(I) = REAL(S(I))

        WINUM = N
        BOTH = .FALSE.
```

```
        IF(WRIT) GOTO 5998

        LTI = 'REAL PART OF DLD OUTPUT'
        LUN = ' '
        RTI = 'NONE'
        HT = 'SAMPLE NUMBER'
        HU = ' '
        STITLE = ' '
        XSCALE = 1
        PNUM = N

        GOTO 5999
C
C*****************************
C
C  FREQUENCY COMPONENTS OF FSK SIGNAL
C
5200    IF(.NOT.FSK) THEN
          TYPE *, ' '
          TYPE *, 'NOT ENABLED'
          GOTO 2000
        ENDIF
        DO 5210 I=1,N
5210      LEFT(I) = FSIG(I)

        WINUM = N
        BOTH = .FALSE.
        IF(WRIT) GOTO 5998

        LTI = 'FREQUENCY COMPONENTS BEFORE DLD'
        LUN = ' '
        RTI = 'NONE'
        HT = 'FREQUENCY'
        HU = 'HERTZ'
        STITLE = ' '
        XSCALE = FO
        PNUM = N/2

        GOTO 5999
C
C*****************************
C
C  PHASE OF FSK ENVELOPE
C
5300    IF(.NOT.FSK) THEN
          TYPE *, ' '
          TYPE *, 'NOT ENABLED'
          GOTO 2000
        ENDIF
        DO 5310 I=1,N
5310      LEFT(I) = PFSK(I)
```

```
        WINUM = N
        BOTH = .FALSE.
        IF(WRIT) GOTO 5998

        LTI ='PHASE OF FSK ENVELOPE'
        LUN = 'PI RADIANS'
        RTI = 'NONE'
        HT = 'SAMPLE NUMBER'
        HU = ' '
        STITLE = ' '
        XSCALE = 1
        PNUM = N

        GOTO 5999
C
C*****************************
C
C  AUTOCORRELATION ESTIMATES
C
5400    IF(.NOT.AUTO) THEN
          TYPE *, ' '
          TYPE *, ' AUTOCORRELATION ESTIMATE NOT ENABLED'
          GOTO 2000
        ENDIF

        TYPE *, ' '
        TYPE 9,'$Plot how many points? (max=',N,')  '
        READ (*,*) PNUM

        DO 5410 I=1,PNUM
          LEFT(I) = SAUTO(I)
          RIGHT(I) = NAUTO(I)
5410    CONTINUE

        WINUM = PNUM
        BOTH = .TRUE.
        IF(WRIT) GOTO 5998

        LTI ='SIGNAL AUTOCORRELATION'
        LUN = ' '
        RTI = 'NOISE AUTOCORRELATION'
        RUN = ' '
        HT = 'LAG COMPONENT'
        HU = ' '
        STITLE = ' '
        XSCALE = 1

        GOTO 5999
C
C*****************************
C
C  WEIGHT MEAN AND VARIANCE DATA
```

```
C
C  PLOT OR WRITE DATA OR DFT OF DATA
C
5500     TYPE *, ' '
         TYPE 11,'∫Do you want weight sequence or DFT of the sequence? (W/D)  '
         READ (*,12) C
         FFT = .FALSE.
         IF(C.EQ.'D' .OR. C.EQ.'d') THEN
           FFT = .TRUE.
           TYPE 11,'∫Do you want to window the sequence? (Y/N)  '
           READ(*,12) C
           IF(C.EQ.'Y' .OR. C.EQ.'y') THEN
             WINDOW = .TRUE.
           ELSE
             WINDOW = .FALSE.
           ENDIF
           IF(.NOT.WRIT)THEN
             TYPE *, ' '
             TYPE 11,'∫Plot up to what frequency?  '
             READ(*,*) FREQP
           ENDIF
         ENDIF
         TYPE *, ' '
C
C  PLOT OR WRITE AVERAGE DATA OR DATA OF ONE TRIAL
C
         TYPE 15
15       FORMAT (' Plot (write) either:',/
     x          ,'     0.  Average of mean and variance',/
     x          ,'     #.  Mean and variance of one run')
         TYPE 16 ,'∫Enter selection as either 0 or run number (up to ',
     X             TRIALS,'):  '
         READ(*,*) CHOICE

         IF(CHOICE.EQ.0) THEN
           IF(FFT) THEN
             TYPE*,' '
             TYPE*,'DFT NOT VALID FOR AVERAGE'
             GOTO 5500
           ENDIF
           DO 5510 J=0,WNUM
             LEFT(J+1) = WMEANA(J)
             RIGHT(J+1) = WVARA(J)
5510       CONTINUE

           WINUM = WNUM + 1
           BOTH = .TRUE.
           IF(WRIT) GOTO 5998
C
C  AVERAGE DATA
C
           LTI = 'AVERAGED MEAN'
```

```
          LUN = ' '
          RTI = 'AVERAGED VARIANCE'
          RUN = ' '
          WRITE(HT,13) 'WEIGHT MEAN AND VARIANCE AVERAGE, ',TRIALS,' TRIALS'
          HU = ' '
          PNUM = WNUM + 1
          XSCALE = 1

          GOTO 5999

        ELSEIF(CHOICE.GT.0 .AND. CHOICE.LE.TRIALS) THEN
C
C  DATA OF INDIVIDUAL TRIAL
C
          IF(.NOT.FFT) THEN
            DO 5520 J=0,WNUM
              LEFT(J+1) = WMEAN(J,CHOICE)
              RIGHT(J+1) = WVAR(J,CHOICE)
5520        CONTINUE

            WINUM = WNUM+1
            BOTH = .TRUE.
            IF(WRIT) GOTO 5998

            LTI = 'WEIGHT MEAN'
            LUN = ' '
            RTI = 'WEIGHT VARIANCE'
            RUN = ' '
            HT = 'MEAN AND VARIANCE OF WEIGHT VECTORS'
            HU = ' '
            PNUM = WNUM + 1
            XSCALE = 1
            GOTO 5999
          ELSE
C
C  DFT OF DATA
C
            DO 5530 J=0,WNUM
              AUTOC(J+1) = CMPLX(WMEAN(WNUM-J,CHOICE),0.)
              IF(WINDOW) AUTOC(J+1) = AUTOC(J+1) * FLOAT(J)/WNUM
5530        CONTINUE
            DO 5535 J=WNUM+1,2*WNUM
              AUTOC(J+1) = CMPLX(WMEAN(J-WNUM,CHOICE),0.)
              IF(WINDOW) AUTOC(J+1) = AUTOC(J+1) * (1.0 - FLOAT(J-WNUM)/WNUM)
5535        CONTINUE
            DO 5540 J=2*WNUM+1,8*WNUM-1
5540          AUTOC(J+1) = CMPLX(0.,0.)
            CALL DFT(AUTOC,8*WNUM,0)
            MAXF = 0.
            DO 5550 J=1,4*WNUM
5550          MAXF = MAX(CABS(AUTOC(J)),MAXF)
            DO 5560 J=1,4*WNUM
```

```
5560          LEFT(J) = 20*ALOG10(CABS(AUTOC(J))/MAXF)

            DO 5570 J=0,WNUM
              AUTOC(J+1) = CMPLX(WVAR(WNUM-J,CHOICE),0.)
              IF(WINDOW) AUTOC(J+1) = AUTOC(J+1) * FLOAT(J)/WNUM
5570        CONTINUE
            DO 5575 J=WNUM+1,2*WNUM
              AUTOC(J+1) = CMPLX(WVAR(J-WNUM,CHOICE),0.)
              IF(WINDOW) AUTOC(J+1) = AUTOC(J+1) * (1.0 - FLOAT(J-WNUM)/WNUM)
5575        CONTINUE
            DO 5580 J=2*WNUM+1,8*WNUM-1
5580          AUTOC(J+1) = CMPLX(0.,0.)
            CALL DFT(AUTOC,8*WNUM,0)
            MAXF = 0.
            DO 5590 J=1,4*WNUM
5590          MAXF = MAX(CABS(AUTOC(J)),MAXF)
            DO 5595 J=1,4*WNUM
5595          RIGHT(J) = 20*ALOG10(CABS(AUTOC(J))/MAXF)

            WINUM = WNUM*4
            BOTH = .TRUE.
            IF(WRIT) GOTO 5998

            LTI = 'DFT OF WEIGHT VECTOR MEAN'
            LUN = ' '
            RTI = 'DFT OF WEIGHT VECTOR VARIANCE'
            RUN = ' '
            HT = 'FREQUENCY'
            HU = 'HERTZ'
            XSCALE = FS/(WNUM*8)
            PNUM = FREQP/XSCALE
            GOTO 5999

          ENDIF
        ELSE

          TYPE *, ' '
          TYPE *, ' CHOICE NOT WITHIN VALID RANGE, TRY AGAIN'

          GOTO 5500

        ENDIF
C
C*****************************
C
C  INTERMEDIATE ANC ERROR
C
5600    IF(.NOT.OFILT) THEN
          TYPE*,'INTERMEDIATE FILTERING NOT ENABLED'
          GOTO 5000
        ENDIF
```

```
        DO 5610 I=1,N
5610      LEFT(I) = ERORI(I)

        WINUM = N
        BOTH = .FALSE.
        IF(WRIT) GOTO 5998

        LTI = 'INTERMEDIATE ANC ERROR'
        LUN = ' '
        RTI = 'NONE'
        HT = 'SAMPLE NUMBER'
        HU = ' '
        STITLE = ' '
        XSCALE = 1
        PNUM = N

        GOTO 5999
C
C*****************************
C
C  INTERMEDIATE ADF OUTPUT
C
5700    IF(.NOT.OFILT) THEN
          TYPE*,'INTERMEDIATE FILTERING NOT ENABLED'
          GOTO 5000
        ENDIF

        DO 5710 I=1,N
5710      LEFT(I) = YI(I)

        WINUM = N
        BOTH = .FALSE.
        IF(WRIT) GOTO 5998

        LTI = 'INTERMEDIATE ADF OUTPUT'
        LUN = ' '
        RTI = 'NONE'
        HT = 'SAMPLE NUMBER'
        HU = ' '
        STITLE = ' '
        XSCALE = 1
        PNUM = N

        GOTO 5999
C
C*****************************
C
C  SECONDARY ANC INPUT
C
5750    IF(.NOT.OFILT) THEN
          TYPE*,'INTERMEDIATE FILTERING NOT ENABLED'
          GOTO 5000
```

```
        ENDIF

        DO 5760 I=1,N
5760      LEFT(I) = MDLD(I)

        WINUM = N
        BOTH = .FALSE.
        IF(WRIT) GOTO 5998

        LTI = 'INTERMEDIATE ANC INPUT'
        LUN = ' '
        RTI = 'NONE'
        HT = 'SAMPLE NUMBER'
        HU = ' '
        STITLE = ' '
        XSCALE = 1
        PNUM = N

        GOTO 5999
C
C*****************************
C
C  SECONDARY ANC ERROR
C
5800    DO 5810 I=1,N
5810      LEFT(I) = EROR(I)

        WINUM = N
        BOTH = .FALSE.
        IF(WRIT) GOTO 5998

        LTI = 'ANC ERROR'
        LUN = ' '
        RTI = 'NONE'
        HT = 'SAMPLE NUMBER'
        HU = ' '
        STITLE = ' '
        XSCALE = 1
        PNUM = N

        GOTO 5999
C
C*****************************
C
C  SECONDARY ADF OUTPUT
C
5900    DO 5910 I=1,N
5910      LEFT(I) = Y(I)

        WINUM = N
        BOTH = .FALSE.
        IF(WRIT) GOTO 5998
```

```
        LTI = 'ADF OUTPUT'
        LUN = ' '
        RTI = 'NONE'
        HT = 'SAMPLE NUMBER'
        HU = ' '
        STITLE = ' '
        XSCALE = 1
        PNUM = N

        GOTO 5999


5998    TYPE *, ' '
        CALL SGOPEN(10,'WRITE','Filenama?  ','NONAME','REAL',WINUM)
        CALL SGTRAN(10,'WRITE','REAL',LEFT,WINUM)

        IF(BOTH) THEN
          CALL SGOPEN(10,'WRITE','Filenama?  ','NONAME','REAL',WINUM)
          CALL SGTRAN(10,'WRITE','REAL',RIGHT,WINUM)
        ENDIF
        GOTO 2000

5999    CALL SPLOT(LEFT,RIGHT,PNUM,NN,LTI,LUN,RTI,RUN,HT,HU,TITLE,
     X             STITLE,XSCALE)
        READ(*,12) C
        GOTO 2000
C
C*************************************************************************
C
C  MODIFIY PARAMETERS
C
6000    TYPE *, ' '
        TYPE 17
17      FORMAT(' Do you want to changa:',/
     x         ,'     1.  Bit aaquanca',/
     x         ,'     2.  FSK frequency shift',/
     x         ,'     3.  Signal to noise ratio',/
     x         ,'     4.  Prefiltar bandwidth',/
     x         ,'     5.  Limitar enabla',/
     x         ,'     6.  Postfiltar bandwidth',/
     x         ,'     7.  ADF dalay',/
     x         ,'     8.  Number of weighta',/
     x         ,'     9.  Gradiant conatant',/
     x         ,'    10.  ADF weight bias anable',/
     x         ,'    11.  ANC paaudo-corralation estimate enabling',/
     x         ,'    12.  ANC max/min aatimate anabling',/
     x         ,'    13.  ANC zero crosaing astimata enabling',/
     x         ,'    14.  ANC output filter enabling',/
     x         ,'    15.  Autocorralation anabling',/
     x         ,'    16.  FSK aignal information enabling',/
```

```
     x        ,'    17.  Simulation ADF enabling',/
     x        ,'    18.  Zero crossing enabling',/
     x        ,'    19.  Number of trials per run',/
     x        ,'    20.  Or do you want to make a run',/
     x        ,'    21.  Or do you want to return to the main menu')
          TYPE 11, '/Enter selection:  '
          READ(*,*) CHOICE

          GOTO (6010,6020,6030,6040,6050,6060,6070,6080,6090,6100,6110,
     X         6120,6130,6140,6150,6160,6170,6180,6190,1000,2000) CHOICE

6010      CALL PSEQUENCE(SEQ,BITS)
          GOTO 6000

6020      CALL PFSKSHIFT(FW)
          GOTO 6000

6030      CALL PNOISE(SNRDB,SNR)
          GOTO 6000

6040      CALL PPREFILTER(PREBW,PRE)
          GOTO 6000

6050      CALL PLIM(LIM)
          GOTO 6000

6060      CALL PPOSTFILTER(POSTBW,POSTA,ZC,POSTBZ)
          GOTO 6000

6070      CALL PADFDELAY(DELAY)
          GOTO 6000

6080      CALL PWEIGHTS(WNUM)
          GOTO 6000

6090      CALL PCONSTANT(MU)
          GOTO 6000

6100      CALL PBIAS(WBIAS)
          GOTO 6000

6110      CALL PCORR(SCORR)
          GOTO 6000

6120      CALL PMM(SMM)
          GOTO 6000

6130      CALL PANCZC(SANCZC)
          GOTO 6000

6140      CALL PFILT1(OFILT)
          GOTO 6000
```

```
6150    CALL PAUTO(AUTO)
        GOTO 6000

6160    CALL PFSKINFO(FSK)
        GOTO 6000

6170    CALL PADF(ADF)
        GOTO 6000

6180    CALL PZCROSS(ZC)
        GOTO 6000

6190    CALL PTRIALS(TRIALS)
        GOTO 6000
C
C*******************************************************************************
C
C  ENTER USER CONTROLLED ADAPTIVE FILTERING ROUTINE
C
7000    DO 7100 I=1,N
          MDLD(I) = REAL(S(I))
7100    CONTINUE
        CALL SADF(MDLD,DLDNN,N,TS,TITLE)
        GOTO 2000
C
C*******************************************************************************
C
C  WRITE PARAMETERS, STATISTICS, AND ESTIMATES
C
8000    TYPE *,' '
        TYPE 24
24      FORMAT (' Do you want to write:',/,
     x          '     1.  Simulation parsmeters',/,
     x          '     2.  Estimste error statistics',/,
     x          '     3.  Baud rste estimates',/,
     x          '     4.  Everything')
        TYPE 11, '$Enter selection:  '
        READ(*,*) CHOICE
C
C  NUMBER OF ESTIMATES TO WRITE
C
        IF(CHOICE.EQ.3 .OR. CHOICE.EQ.4) THEN
          TYPE *,' '
          TYPE 25,' There are ',TRIALS,' estimstes'
          TYPE 11,'$Enter first estimate and number of estimstes:  '
          READ(*,*) C1,C2
        ENDIF

        GOTO (8100,8200,8300,8100) CHOICE
C
C*****************************
```

```
C
C  PARAMETERS
C
8100    WRITE(*,26) TRIALS,SNRDB,N,FS,BITS,R,FC,FW,2*PREBW
26      FORMAT('  ',/,/,
     X          ' SIMULATION PARAMETERS:',/,
     x       /,'      Simulation Parameters:',/,
     x         '          Number of trials per run =  ',I3,/,
     x         '          Signal-to-noise ratio =  ',F4.1,' dB',/,
     X       /,'      Sampling Parameters:',/,
     x         '          Number of samples =  ',I4,/,
     x         '          Sampling Frequency =  ',F6.0,' Hz',/,
     X       /,'      Bit Seqnence Parameters:',/,
     x         '          Number of bits =  ',I2,/,
     x         '          Baud rate =  ',F5.0,' Hz',/,
     x       /,'      FSK Modnlation Parameters:',/,
     x         '          Carrier frequency =  ',F8.0,' Hz',/,
     x         '          Freqnency shift =  ',F5.0,' Hz',/,
     x       /,'      Receiver Parametera:',/,
     x         '          Pre-filter bandwidth =  ',F5.0,' Hz')
        IF(.NOT.POSTA) THEN
          WRITE(*,*) '        Post filtering not enabled for ANC'
          IF(ZC) WRITE(*,31) POSTBZ
31        FORMAT('         Post-filter bandwidth for ZC =  ',F5.0,' Hz')
        ELSE
          WRITE(*,32) POSTBW
32        FORMAT('         Poat-filter bandwidth =  ',F5.0,' Hz')
        ENDIF
        IF(LIM) THEN
          WRITE(*,*) '        Limiter enabled'
        ELSE
          WRITE(*,*) '        Limiter not enabled'
        ENDIF
        WRITE(*,27) DELAY,MU,WNUM
27      FORMAT(' ',/,
     X         '     Adaptive Noise Cancelling Parametera:',/,
     X         '         Reference inpnt delay =  ',I2,/,
     x         '         Gradient constant =  ',E8.1,/,
     x         '         Number of weighta =  ',I3)

        IF(WBIAS) THEN
          WRITE(*,*) '        Weighting bias inclnded'
        ELSE
          WRITE(*,*) '        Weighting bias not included'
        ENDIF

        WRITE(*,*) ' '
        WRITE(*,*) ' '
        IF (ADF .OR. ZC) WRITE (*,*) 'ESTIMATORS ENABLED:'
        WRITE(*,*) ' '
        IF(OFILT) WRITE(*,*) '    ANC Intermediate Filtering enabled'
        IF(SMM) WRITE(*,*) '    Min/Max eatimators enabled'
```

```
      IF(SANCZC) WRITE(*,*) '    ANC-ZC estimators enabled'
      IF(SCORR) WRITE(*,*) '    Pseudo-Correlation estimators enabled'
      IF(ZC) WRITE(*,*) '    Zero Croasing estimator enabled'

      IF(CHOICE.EQ.1) GOTO 2000
C
C*****************************
C
C  WRITE STATISTICS
C
      WRITE(*,28)
8200  WRITE(*,*) ' '
      WRITE(*,*) ' '
      WRITE(*,*) 'NORMALIZED BAUD RATE ESTIMATE ERROR STATISTICS:'
      WRITE(*,*) ' '
C
C  INTERMEDIATE ESTIMATE STATISTICS
C
      IF (ADF) THEN
        WRITE(*,*) ' '
        WRITE(*,*) ' MMI    MM2I    AMI    MLTOI    MZCI    MAI    VAI    ',
     X             ' EXCI'
        WRITE(*,*) ' '
        WRITE(*,*) 'Normalized baud rate mean error:'
        WRITE(*,19) (SIESTM(I),I=1,8)
        WRITE(*,*) 'Normalized baud rate error variance:'
        WRITE(*,19) (SIESTV(I),I=1,8)
        WRITE(*,*) 'Normalized baud rate mean aquared error:'
        WRITE(*,19) (SIESTS(I),I=1,8)
      ENDIF
C
C  SECONDARY ESTIMTATE STATISTICS
C
      IF(OFILT) THEN
        WRITE(*,*) ' '
        WRITE(*,*) ' MMF    MM2F    AMF    MLTOF    MZCF    MAF    VAF    ',
     X             ' XZCF    YZCF    EZCF'
        WRITE(*,*) ' '
        WRITE(*,*) 'Normalized baud rate mean error:'
        WRITE(*,19) (SFESTM(I),I=1,10)
        WRITE(*,*) 'Normalized baud rate error variance:'
        WRITE(*,19) (SFESTV(I),I=1,10)
        WRITE(*,*) 'Normalized baud rate mean squared error:'
        WRITE(*,19) (SFESTS(I),I=1,10)
C
C  BANDWIDTH STATISTICS
C
        IF(OFILT) THEN
          WRITE(*,*) ' '
          WRITE(*,*) 'BANDWIDTH ESTIMATE:'
          WRITE(*,*) ' '
          WRITE(*,30) ' Bandwidth mean = ',SBWM,
```

```
     X                    '    Bandwidth standard deviation = ',SBWV
           WRITE(*,33) ' Normalized bandwidth mean = ',NSBWM,
     X                    '    Standard deviation = ',NSBWS
         ENDIF
       ENDIF
C
C  TRADITIONAL ZERO CROSSING STATISTICS
C
       IF (ZC) THEN
         WRITE(*,*) ' '
         WRITE(*,*) 'ZERO CROSSING ESTIMATE ERROR STATISTICS:'
         WRITE(*,*) 'Normalized baud rate mean error:'
         WRITE(*,19) (ZESTM(I),I=1,1)
         WRITE(*,*) 'Normalized baud rate error variance:'
         WRITE(*,19) (ZESTV(I),I=1,1)
         WRITE(*,*) 'Normalized baud rate mean squared error:'
         WRITE(*,19) (ZESTS(I),I=1,1)
       ENDIF
       WRITE(*,*) ' '

       IF(CHOICE.EQ.2) GOTO 2000
C
C*****************************
C
C  WRITE ESTIMATES
C
       WRITE(*,28)
8300   WRITE(*,*) ' '
       WRITE(*,*) ' '
       WRITE(*,*) 'BIT LENGTH ESTIMATES (IN NUMBER OF SAMPLES):'
       WRITE(*,*) ' '
C
C  INTERMEDIATE ESTIMATES
C
       IF (ADF) THEN
         WRITE(*,*) ' '
         WRITE(*,*) '  MMI    MM2I    AMI    MLTOI    MZCI    MAI    VAI    ',
     X                ' EZCIT'
         WRITE(*,*) ' '
         DO 8310 II=C1,C1+C2-1
           WRITE(*,29) (SIEST(I,II),I=1,8)
           IF(MOD(II,10).EQ.0) WRITE(*,*) ' '
8310       CONTINUE
       ENDIF
C
C  SECONDARY ESTIMATES
C
       IF(OFILT) THEN
         WRITE(*,*) ' '
         WRITE(*,*) '  MMF    MM2F    AMF    MLTOF    MZCF    MAF    VAF    ',
     X                ' XZCF    YZCF    EZCF'
         WRITE(*,*) ' '
```

```
          DO 8340 II=C1,C1+C2-1
            WRITE(*,29) (SFEST(I,II),I=1,10)
            IF(MOD(II,10).EQ.0) WRITE(*,*) ' '
8340        CONTINUE
        ENDIF
C
C  TRADISTION ZERO CROSSING ESTIMATES
C
        IF(ZC) THEN
          WRITE(*,*) ' '
          WRITE(*,*) 'Zero Crossing Estimates:'
          WRITE(*,*) ' '
          DO 8360 II=C1,C1+C2-1
            WRITE(*,29) (ZEST(I,II),I=1,1)
            IF(MOD(II,10).EQ.0) WRITE(*,*) ' '
8360        CONTINUE
          WRITE(*,*) ' '
        ENDIF

        GOTO 2000
C
C*************************************************************************
C
C  FORMAT STATEMENTS AND MAJOR PROGRAM ENDING
C
9000    CONTINUE

9       FORMAT(A28,I4,A5)
10      FORMAT(A80)
11      FORMAT(A)
12      FORMAT(A1)
13      FORMAT(A34,I3,A7)
14      FORMAT(A10,I2,A3,I3,A5,F12.10,A6)
16      FORMAT(A50,I3,A4)
18      FORMAT(A29,I4,A3)
19      FORMAT(' ',10(F5.2,2X))
20      FORMAT(' ',10(2X,I3,2X))
25      FORMAT(A11,I3,A11)
28      FORMAT('1')
29      FORMAT(' ',10(F6.1,1X))
30      FORMAT(A18,F7.0,A35,F7.0)
33      FORMAT(A27,F5.2,A30,F5.2)

        STOP
        END

C
C*************************************************************************
C
C  PARAMTER INPUT SUBROUTINES
C
        SUBROUTINE PRMTRS
```

```
      REAL POSTBW,POSTBZ,SNRDB
      INTEGER SEQ(64),BITS,BIT,N,N1,ND,TRIALS
      CHARACTER*1 C
      LOGICAL AUTO,ADF,FSK,SNR,PRE,POSTA,POSTZ,ZC,ZC4,LIM
C
C*****************************
C
C  SEQUENCE
C
      ENTRY PSEQUENCE(SEQ,BITS)

      TYPE *, ' '
      TYPE 11,'∫    Test sequence or pseudo-random sequence (T/S)?   '
      READ (*,12) C

      IF (C.EQ.'T' .OR. C.EQ.'t') THEN
        DO 100 I= 1,BITS/2
          SEQ(2*I) = 1
          SEQ(2*I-1) = 0
100     CONTINUE
      ELSEIF (C.EQ.'S' .OR. C.EQ.'s') THEN
        DO 150 I=0,32,32
          SEQ(I+1) = 1
          SEQ(I+2) = 0
          SEQ(I+3) = 0
          SEQ(I+4) = 1
          SEQ(I+5) = 1
          SEQ(I+6) = 0
          SEQ(I+7) = 1
          SEQ(I+8) = 0
          SEQ(I+9) = 0
          SEQ(I+10) = 1
          SEQ(I+11) = 0
          SEQ(I+12) = 0
          SEQ(I+13) = 0
          SEQ(I+14) = 0
          SEQ(I+15) = 1
          SEQ(I+16) = 0
          SEQ(I+17) = 1
          SEQ(I+18) = 0
          SEQ(I+19) = 1
          SEQ(I+20) = 1
          SEQ(I+21) = 1
          SEQ(I+22) = 0
          SEQ(I+23) = 1
          SEQ(I+24) = 1
          SEQ(I+25) = 0
          SEQ(I+26) = 0
          SEQ(I+27) = 0
          SEQ(I+28) = 1
          SEQ(I+29) = 1
```

```
            SEQ(I+30) = 1
            SEQ(I+31) = 1
            SEQ(I+32) = 1
150       CONTINUE
        ELSE
          DO 200 I=1,BITS
200         SEQ(I) = 0
        ENDIF

        RETURN
C
C*****************************
C
C  NOISE
C
        ENTRY PNOISE(SNRDB,SNR)

        TYPE *, ' '
        TYPE 11, '$    Do you want to add noise (Y/N)?   '
        READ(*,12) C

        IF (C.EQ.'Y' .OR. C.EQ.'y') THEN
          SNR = .TRUE.
          TYPE 11, '$    Signal to noise ratio (dB)?   '
          READ (*,*) SNRDB
        ELSE
          SNR = .FALSE.
        ENDIF

        RETURN
C
C*****************************
C
C  PREFILTER
C
        ENTRY PPREFILTER(PREBW3,PRE)

        TYPE *, ' '
C       TYPE 11, '$    Do you want to perform pre-filtering?   '
C       READ (*,12) C
        C = 'Y'
        IF (C.EQ.'Y' .OR. C.EQ.'y') THEN
          PRE = .TRUE.
          TYPE 11, '$    Pre-filter bandwidth?   '
          READ (*,*) RFBW
          PREBW3 = RFBW/2.
        ELSE
          PRE = .FALSE.
          PREBW3 = 0.0
        ENDIF

        RETURN
```

```
C
C*****************************
C
C  POST FILTER
C
        ENTRY PPOSTFILTER(POSTBW,POSTA,ZC4,POSTBZ)

        TYPE *, ' '
        TYPE 11, '$    Do you want to perform post-filtering?  '
        READ (*,12) C
        IF (C.EQ.'Y' .OR. C.EQ.'y') THEN
          POSTA = .TRUE.
          TYPE 11, '$    Post-filter bandwidth?    '
          READ (*,*) POSTBW
          POSTBZ = POSTBW
        ELSE
          POSTA = .FALSE.
          IF(ZC4) THEN
            TYPE 11, '$    Post-filter bandwidth for ZC algorithm?    '
            READ (*,*) POSTBZ
          ENDIF
        ENDIF

        RETURN
C
C*****************************
C
C  LIMITER
C
        ENTRY PLIM(LIM)

        TYPE *, ' '
        TYPE 11, '$    Do you want to enable the limiter?  '
        READ(*,12) C

        IF(C.EQ.'Y' .OR. C.EQ.'y') THEN
          LIM = .TRUE.
        ELSE
          LIM = .FALSE.
        ENDIF

        RETURN
C
C*****************************
C
C  DLD DELAY
C
        ENTRY PDELAY(ND)

        TYPE *, ' '
        TYPE 11, '$    Delay factor of DLD (an integer)?    '
        READ(*,*) ND
```

```
        RETURN
C
C*******************************
C
C  FSK FREQUENCY SHIFT
C
        ENTRY PFSKSHIFT(FW)

        TYPE *, ' '
        TYPE 11 ,'∫    FSK frequency shift?   '
        READ (*,*) FW

        RETURN
C
C*******************************
C
C  NUMBER OF TRIALS
C
        ENTRY PTRIALS(TRIALS)


        TYPE *, ' '
        TYPE 11, '∫Number of trials per run?   '
        READ (*,*) TRIALS

        RETURN
C
C*******************************
C
C  TRADITIONAL ZERO CROSSING ENABLE
C
        ENTRY PZCROSS(ZC)

        TYPE *, ' '
        TYPE 11, '∫Traditional zero crossing algorithm enabled?  (Y/N)  '
        READ(*,12) CANS
        IF (CANS.EQ.'Y' .OR. CANS.EQ.'y') THEN
          ZC = .TRUE.
        ELSE
          ZC = .FALSE.
        ENDIF

        RETURN
C
C*******************************
C
C  ADAPTIVE NOISE CANCELLING ENABLE
C
        ENTRY PADF(ADF)

        TYPE *, ' '
```

```
        TYPE 11, '∫Adaptive noise cancelling enabled?  (Y/N)  '
        READ(*,12) CANS
        IF (CANS.EQ.'Y' .OR. CANS.EQ.'y') THEN
          ADF = .TRUE.
        ELSE
          ADF = .FALSE.
        ENDIF

        RETURN
C
C*****************************
C
C  AUTOCORRELATION ENABLE
C
        ENTRY PAUTO(AUTO)

        TYPE *, ' '
        TYPE 11, '∫Autocorrelation enabled?  (Y/N)  '
        READ(*,12) CANS
        IF (CANS.EQ.'Y' .OR. CANS.EQ.'y') THEN
          AUTO = .TRUE.
        ELSE
          AUTO = .FALSE.
        ENDIF

        RETURN
C
C*****************************
C
C  FSK SIGNAL INFORMATION ENABLE
C
        ENTRY PFSKINFO(FSK)

        TYPE *, ' '
        TYPE 11, '∫Phase and frequency of FSK signal enabled? (Y/N)  '
        READ(*,12) CANS
        IF (CANS.EQ.'Y' .OR. CANS.EQ.'y') THEN
          FSK = .TRUE.
        ELSE
          FSK= .FALSE.
        ENDIF

        RETURN

11      FORMAT(A)
12      FORMAT(A1)

        END
C
C*************************************************************************
C
C  NOISE GENERATION SUBROUTINE
```

```
C
      SUBROUTINE SNOISE(CNOISE,N,A,TS,PREBW,SNRDB,LSNR)

      COMPLEX CNOISE(2048)
      REAL A,TS,PREBW,SNRDB,VAR,MEAN,SNR,RNOISE,INOISE
      INTEGER N
      LOGICAL LSNR

      IF(.NOT.LSNR) THEN
        DO 300 I=1,N
300       CNOISE(I) = CMPLX(0.,0.)
      ELSE
        SNR = 10.**(SNRDB/10.0)
        VAR = (A**2/2)/(TS*PREBW*SNR)
        MEAN = 0.0

        DO 400 I=1,N
          RNOISE = GAUSS(MEAN,VAR)
          INOISE = GAUSS(MEAN,VAR)
          CNOISE(I) = CMPLX(RNOISE,INOISE)
400     CONTINUE
      ENDIF

      RETURN
      END
C
C*************************************************************************
C
C  PLOT TITLE FORMATION SUBROUTINE
C
      SUBROUTINE PTITLE(TITLE,PRE,POSTA,FW,PREBW,POSTBW,SNRDB,SNR,R,LIM)

      IMPLICIT NONE
      REAL FW,PREBW,POSTBW,R,SNRDB
      CHARACTER TITLE*80,PRETI*13,POSTTI*14
      CHARACTER STI*14,FTI*12,SNRTI*10,LTI*9
      LOGICAL SNR,PRE,POSTA,LIM

      WRITE (PRETI,19) ', PRE-BW=',IFIX(PREBW*2.)
19    FORMAT(A,I4)
      WRITE(POSTTI,20) ', POST-BW=', IFIX(POSTBW)
20    FORMAT (A,I4)
      WRITE(STI,21) 'BAUD RATE=',IFIX(R)
21    FORMAT(A10,I4)
      WRITE(FTI,23) ', SHIFT=',IFIX(FW)
23    FORMAT(A,I4)
      WRITE (SNRTI,24) ', SNRDB=',IFIX(SNRDB)
24    FORMAT(A,I2)

      IF(.NOT.SNR) SNRTI = ', NO NOISE'

      IF(LIM) THEN
```

```
  LTI = ', LIMITER'
ELSE
  LTI = ' '
ENDIF

IF (PRE) THEN
  IF(POSTA) THEN
    TITLE = STI//FTI//PRETI//POSTTI//SNRTI//LTI
  ELSE
    TITLE = STI//FTI//PRETI//', NO POST'//SNRTI//LTI
  ENDIF
ELSE
  IF(POSTA) THEN
    TITLE = STI//FTI//',NO PRE'//POSTTI//SNRTI//LTI
  ELSE
    TITLE = STI//FTI//',NO PRE, NO POST'//SNRTI//LTI
  ENDIF
ENDIF

RETURN
END
```

```
C*****************************************************************************
C
C       SADF
C
C       VAX-11 FORTRAN SOURCE FILENAME:          SADF1.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER(S)
C       --------        ----                     -------------
C        1.0            MAY 6, 1985              MARC D. BRACK
C
C*****************************************************************************
C
C       CALLING SEQUENCE
C
C               CALL SADF(X,XNN,LENGTH,TS,TITLE)
C
C       PURPOSE
C
C               Implements the adaptive noise canceller in a nser
C               controlled environment
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               SPLOT
C               BUTER
C               DFT
C
C       ROUTINE(S) ACCESSED OR CALLED BY EMBEDDED ROUTINE(S)
C
C               NONE
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               X       Ontput of the delay-line discriminator (DLD)
C
C               XNN     Ontput of the DLD when the inpnt was noise free
C
C               LENGTH  Number of samples in DLD ontput
C
C               TS      Sampling interval
C
C               TITLE   Major title for plotting pnrposes
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               NONE
C
C*****************************************************************************
C
        SUBROUTINE SADF(X,XNN,LENGTH,TS,TITLE)
```

```
      IMPLICIT NONE
      INTEGER N,I,NUM1,LENGTH,DELAY,N2,NN,PNUM,P2,TN,TDELAY,TMUI
      INTEGER PFIRST,CHOICE,CS1,CS2,J,N3
      REAL MU,W(2049,0:256),WMEAN(0:256),WVAR(0:256),WMEANS(0:256)
      REAL EROR(2048),LEFT(2048),RIGHT(2048),Y(2048),X(2048)
      REAL FEROR(2048),FY(2048),BW,MAXF,FX(2048)
      REAL WDFT(4096),TMU,TS,FREQP,XNN(2048),XSCALE
      COMPLEX CW(2048),CW2(2048),BUTER
      CHARACTER*80 NAME1,NAME2,BNK,NAME,LT,RT,HT,TITLE,STITLE,HU
      CHARACTER DELAYC*4,NC*4,MUC*11,WAIT*1,CANS*1,CANS2*1
      LOGICAL WBIAS,FILT

C
C INPUT PARAMETERS FOR ROUTINE
C
      CALL PADFDELAY(DELAY)
      CALL PWEIGHTS(N)
      CALL PCONSTANT(MU)
      CALL PBIAS(WBIAS)
      CALL PFILTER(FILT,BW)
C
C*********************************************************************
C
C  ADAPTIVE NOISE CANCELLER SECTION, LMS ALGORITHM
C
1000  CONTINUE

      DO 100 I=0,N
        WMEAN(I) = 0.
        WMEANS(I) = 0.
        WVAR(I) = 0.
100   CONTINUE

      DO 120 J=1,LENGTH+1
        DO 110 I=0,N
110       W(J,I) = 0.0
120   CONTINUE

      DO 160 J=1,LENGTH
        IF (WBIAS) THEN
          Y(J) = W(J,0)
        ELSE
          Y(J) = 0.0
        ENDIF

        DO 130 I=1,N
130       IF(J-I+1.GT.0) Y(J) = Y(J) + X(J-I+1)*W(J,I)

        IF(J+DELAY.LE.LENGTH) THEN
          EROR(J) = X(J+DELAY) - Y(J)
        ELSE
```

```
            EROR(J) = -Y(J)
          ENDIF

          W(J+1,0) = W(J,0) + 2.*MU*EROR(J)

          DO 140 I=1,N
140         IF(J-I+1.GT.0) W(J+1,I) = W(J,I) + 2.*MU*EROR(J)*X(J-I+1)

          IF(J.GT.N) THEN
            DO 150 I=0,N
              WMEAN(I) = WMEAN(I) + W(J+1,I)/LENGTH
              WMEANS(I) = WMEANS(I) + W(J+1,I)**2/LENGTH
150         CONTINUE
          ENDIF
160     CONTINUE

        DO 170 I=0,N
170       WVAR(I) = WMEANS(I) - WMEAN(I)**2

C
C  FILTER THE ERROR AND THE FILTER OUTPUT
C
        IF (FILT) THEN
          DO 180 I=0,N
180         CW(I+1) = CMPLX(WMEAN(I),0.)

          DO 190 I=N+1,4*N-1
190         CW(I+1) = CMPLX(0.,0.)
C
C  DETERMINE 3 dB BANDWIDTH OF SIGNAL
C
          CALL DFT(CW,4*N,0)

          MAXF = 0
          DO 200 I=1,4*N/2
200         IF(CABS(CW(I)).GT.MAXF) MAXF = CABS(CW(I))
          MAXF = MAXF**2

          DO 210 I=1,4*N/2
            IF(CABS(CW(4*N/2+1-I))**2.GT.MAXF/2) THEN
              BW = (4*N/2+2-I)/(4*N/2*TS)
              GOTO 220
            ENDIF
210       CONTINUE
220       CONTINUE

        TYPE *, 'BANDWIDTH =  ',BW
        ENDIF
C
C*****************************************************************************
C
C  PLOTTING AND PARAMETER CHANGE
```

```
C
3000    WRITE(*,3)
3       FORMAT(' Do you want to plot:',/
     x        ,'    1.  DLD output',/
     x        ,'    2.  Output error',/
     x        ,'    3.  Filter estimate',/
     x        ,'    4.  Weights of filter for a specific vector',/
     x        ,'    5.  Frequency response of filter',/
     x        ,'    6.  Mean and variance of weights',/
     x        ,'    7.  History of a specific weight',/
     x        ,' Or do you want to:',/
     x        ,'    8.  Change the input delay',/
     x        ,'    9.  Change the gradient constant',/
     x        ,'   10.  Change the number of weights',/
     x        ,'   11.  Change the weighting bias',/
     x        ,'   12.  Write the results to disk',/
     x        ,'   13.  Change the automatic filter enable',/
     x        ,'   14.  Perform a run',/
     x        ,'   15.  Return to the calling routine')
        TYPE 1, '/Enter selection:   '
        READ (*,*) CHOICE

        IF (CHOICE.LE.7) THEN
1         FORMAT(A)
2         FORMAT(A1)
          WRITE(*,*) ' '
          TYPE 1, '/First point to plot =  '
          READ (*,*) PFIRST
C         TYPE 1, '/Interval between points =  '
C         READ (*,*) NN
        ENDIF

        NN = 1

        GOTO (3100,3150,3200,3300,3400,3500,3600,4000,
     X        4100,4200,4300,4400,4500,1000,9000), CHOICE
C
C*************************************************************************
C
C  PLOTTING
C
C  PLOT DLD OUTPUT
C
3100    TYPE 1, '/Number of points to plot?  '
        READ (*,*) PNUM
        IF (FILT) THEN
          TYPE 1, '/Plot DLD output or filtered DLD output? (D/F)  '
          READ (*,2) CANS
        ELSE
          CANS = 'D'
        ENDIF
C
```

```
C  PERFORM INTERMEDIATE FILTERING
C
        IF (CANS.EQ.'F' .OR. CANS.EQ.'f') THEN
          DO 290 J=1,LENGTH
290         CW(J) = CMPLX(X(J),0.)

          CALL DFT(CW,LENGTH,0)
          DO 300 J=1,LENGTH/2
            CW(J) = CW(J)*BUTER(4,BW,FLOAT(J-1)/(TS*LENGTH))
            CW(LENGTH+1-J) = CW(LENGTH+1-J)*
     X                       BUTER(4,BW,FLOAT(-J)/(TS*LENGTH))
300       CONTINUE
          CW(LENGTH/2+1) = CMPLX(0.,0.)
          CALL DFT(CW,LENGTH,1)

          DO 310 J=1,LENGTH
310         FX(J)  = REAL(CW(J))
        ENDIF

        DO 3110 I=PFIRST,LENGTH
          IF(I+DELAY.LT.LENGTH) THEN
            RIGHT(I-PFIRST+1) = XNN(I+DELAY)
          ELSE
            RIGHT(I-PFIRST+1) = 0.0
          ENDIF
          IF (CANS.EQ.'F' .OR. CANS.EQ.'f') THEN
            LEFT(I-PFIRST+1) = FX(I)
          ELSE
            LEFT(I-PFIRST+1) = X(I)
          ENDIF
3110    CONTINUE

        RT = 'NOISESLESS SIGNAL'
        IF (CANS.EQ.'F'.OR.CANS.EQ.'f') THEN
          LT = 'FILTERED DLD OUTPUT'
        ELSE
          LT = 'DLD OUTPUT'
        ENDIF
        HT = 'SAMPLE NUMBER'
        HU = ' '
        XSCALE = 1
C       PNUM = LENGTH - PFIRST + 1
        GOTO 3900
C
C  PLOT ERROR OF ANC
C
3150    TYPE 1, '∫Number of points to plot?  '
        READ (*,*) PNUM
        IF (FILT) THEN
          TYPE 1, '∫Plot error or filtered error? (E/F)  '
          READ (*,2) CANS
        ELSE
```

```
          CANS = 'E'
        ENDIF
C
C  PERFORM INTERMEDIATE FILTERING
C
        IF (CANS.EQ.'F' .OR. CANS.EQ.'f') THEN
          DO 260 J=1,LENGTH
260         CW(J) = CMPLX(EROR(J),0.)

          CALL DFT(CW,LENGTH,0)
          DO 270 J=1,LENGTH/2
            CW(J) = CW(J)*BUTER(4,BW,FLOAT(J-1)/(TS*LENGTH))
            CW(LENGTH+1-J) = CW(LENGTH+1-J)*
     X                          BUTER(4,BW,FLOAT(-J)/(TS*LENGTH))
270       CONTINUE
          CW(LENGTH/2+1) = CMPLX(0.,0.)
          CALL DFT(CW,LENGTH,1)

          DO 280 J=1,LENGTH
280         FEROR(J)  = REAL(CW(J))
        ENDIF

        DO 3160 I=PFIRST,LENGTH
          IF(I+DELAY.LT.LENGTH) THEN
            RIGHT(I-PFIRST+1) = XNN(I+DELAY)
          ELSE
            RIGHT(I-PFIRST+1) = 0.0
          ENDIF
          IF (CANS.EQ.'F' .OR. CANS.EQ.'f') THEN
            LEFT(I-PFIRST+1) = FEROR(I)
          ELSE
            LEFT(I-PFIRST+1) = EROR(I)
          ENDIF
3160    CONTINUE

        RT = 'NOISESLESS SIGNAL'
        IF (CANS.EQ.'F'.OR.CANS.EQ.'f') THEN
          LT = 'FILTERED ERROR'
        ELSE
          LT = 'ERROR'
        ENDIF
        HT = 'SAMPLE NUMBER'
        HU = ' '
        XSCALE = 1
C       PNUM = LENGTH - PFIRST + 1
        GOTO 3900
C
C  PLOT ADF OUTPUT
C
3200    TYPE 1, '∫Number of points to plot?  '
        READ (*,*) PNUM
        IF (FILT) THEN
```

```
          TYPE 1, '∫Plot filter output or filtered filtered output? (O/F)  '
          READ (*,2) CANS
        ELSE
          CANS = 'O'
        ENDIF
C
C  PERFORM INTERMEDIATE FILTERING
C
        IF (CANS.EQ.'F' .OR. CANS.EQ.'f') THEN
          DO 230 J=1,LENGTH
230         CW(J) = CMPLX(Y(J),0.)

          CALL DFT(CW,LENGTH,0)
          DO 240 J=1,LENGTH/2
            CW(J) = CW(J)*BUTER(4,BW,FLOAT(J-1)/(TS*LENGTH))
            CW(LENGTH+1-J) = CW(LENGTH+1-J)*
     X                       BUTER(4,BW,FLOAT(-J)/(TS*LENGTH))
240       CONTINUE
          CW(LENGTH/2+1) = CMPLX(0.,0.)
          CALL DFT(CW,LENGTH,1)

          DO 250 J=1,LENGTH
250         FY(J)  = REAL(CW(J))
        ENDIF

        DO 3210 I=PFIRST,LENGTH
          IF(I+DELAY.LT.LENGTH) THEN
            RIGHT(I-PFIRST+1) = XNN(I+DELAY)
          ELSE
            RIGHT(I-PFIRST+1) = 0.0
          ENDIF
          IF (CANS.EQ.'F' .OR. CANS.EQ.'f') THEN
            LEFT(I-PFIRST+1) = FY(I)
          ELSE
            LEFT(I-PFIRST+1) = Y(I)
          ENDIF
3210    CONTINUE

        RT = 'NOISELESS SIGNAL'
        IF (CANS.EQ.'F' .OR. CANS.EQ.'f') THEN
          LT = 'FILTERED FILTER OUTPUT'
        ELSE
          LT = 'FILTER OUTPUT'
        ENDIF
        HT = 'SAMPLE NUMBER'
        HU = ' '
        XSCALE = 1
C       PNUM = LENGTH - PFIRST + 1
        GOTO 3900
C
C  PLOT A SPECIFIC WEIGHT VECTOR
C
```

```
3300    IF(PFIRST.GT.N) THEN
          WRITE(*,*) 'ERROR:  THERE ARE NOT THAT MANY WEIGHTS'
          GOTO 3000
        ENDIF
        WRITE(*,*) ' '
        TYPE 1, '∫Weights for what vector number?  (a,b)  '
        READ (*,*) CS1,CS2
        IF (CS2.GE.1) THEN
          DO 3310 I=0,N
            LEFT(I-PFIRST+2) = W(CS1,I-PFIRST+1)
            RIGHT(I-PFIRST+2) = W(CS2,I-PFIRST+1)
3310      CONTINUE
          WRITE(RT,8) 'VECTOR NUMBER ',CS2
8         FORMAT(A17,I4)
        ELSE
          DO 3320 I=0,N
3320        LEFT(I-PFIRST+2) = W(CS1,I-PFIRST+1)
          RT = 'NONE'
        ENDIF

        WRITE(LT,8) 'VECTOR NUMBER ',CS1
        HT = 'WEIGHT NUMBER'
        HU = ' '
        NN = 1
        XSCALE = 1
        PNUM = N+1
        GOTO 3900

C
C  PLOT FREQUENCY RESPONSE OF ADF
C
3400    N2 = 1
3410    N2 = N2*2
        IF(N2.LT.N+1) GOTO 3410

        WRITE(*,*) ' '
        WRITE(*,6) N+1,' points, need to zero pad to at least ',N2,' points'
6       FORMAT(I4,A,I4,A)
        TYPE 1, '∫Number of points to zero pad to =  '
        READ (*,*) N3

        N2 = 1
3415    N2 = N2*2
        IF(N2.LT.N3) GOTO 3415
        WRITE(*,*) 'Zero padded to ',N2,' points'

        WRITE (*,*) ' '
        WRITE (*,9)
9       FORMAT (' Frequency response desired for:',/,
     X          '    0.  Mean and variance of weights',/,
     X          '    #.  Specific vector of weights')
        TYPE 1, '∫Enter 0 or vector number:  '
```

```
      READ (*,*)CS1

      IF (CS1.NE.0) THEN
        DO 3420 I=0,N
3420      CW(I+1) = CMPLX(W(CS1,I),0.0)
        IF(N2.GT.N+1) THEN
          DO 3430 I=N+1,N2-1
3430        CW(I+1) = CMPLX(0.,0.)
        ENDIF
      ELSE
        DO 3440 I=0,N
          CW(I+1) = CMPLX(WMEAN(I),0.0)
          CW2(I+1) = CMPLX(WVAR(I),0.0)
3440      CONTINUE
        IF(N2.GT.N+1) THEN
          DO 3450 I=N+1,N2-1
            CW(I+1) = CMPLX(0.,0.)
            CW2(I+1) = CMPLX(0.,0.)
3450        CONTINUE
        ENDIF
      ENDIF

      CALL DFT(CW,N2,0)
      CALL DFT(CW2,N2,0)

      DO 3460 I=1,N2/2
        LEFT(I) = CABS(CW(I))
        RIGHT(I) = CABS(CW2(I))
3460  CONTINUE

      IF(PFIRST.GT.N2/2) THEN
        WRITE(*,*) 'THERE ARE ONLY ',N2/2,' RESPONSES'
        GOTO 3000
      ENDIF

      IF (CS1.EQ.0) THEN
        LT = 'RESPONSE OF MEAN'
        RT = 'RESPONSE OF VARIANCE'
      ELSE
        WRITE(LT,10) 'RESPONSE OF VECTOR ',CS1
10      FORMAT(A19,I4)
        RT = 'NONE'
      ENDIF
      HT = 'FREQUENCY RESPONCE'
      HU = 'HERTZ'
      XSCALE = 1./(N2*TS)
      WRITE(*,*) ' '
      TYPE 1,'$Plot up to what frequency? (4800 maximum)  '
      READ (*,*) FREQP
      PNUM = FREQP/XSCALE
      GOTO 3900
C
```

```
C  PLOT MEAN AND VARIANCE OF WEIGHTS
C
3500     DO 3510 I=0,N
           LEFT(I+1) = WMEAN(I)
           RIGHT(I+1) = WVAR(I)
3510     CONTINUE
         LT = 'MEAN OF WEIGHTS'
         RT = 'VARIANCE OF WEIGHTS'
         HT = 'WEIGHT NUMBER'
         HU = ' '
         XSCALE = 1.
         PNUM = N+1
         GOTO 3900
C
C  PLOT ONE WEIGHT THROUGHOUT THE UPDATE PROCEDURE
C
3600     TYPE 1, '∫History of what number weights?  (enter a,b)  '
         READ (*,*) CS1,CS2
         IF(CS2.GE.0) THEN
           DO 3610 J=1,LENGTH
             LEFT(J) = W(J,CS1)
             RIGHT(J) = W(J,CS2)
3610       CONTINUE
           WRITE(RT,7) 'CROSS-SECTION NUMBER ',CS2
7          FORMAT(A21,I4)
         ELSE
           DO 3620 J=1,LENGTH
3620         LEFT(J) = W(J,CS1)
           RT = 'NONE'
         ENDIF

         WRITE(LT,7) 'CROSS-SECTION NUMBER ',CS1
         HT = 'WEIGHT CROSS-SECTION'
         HU = ' '
         PNUM = J
         XSCALE = TS
         GOTO 3900

3900     CONTINUE
C
C  FORM SUB-TITLE AND CALL PLOTTING ROUTINE
C
         BNK = ' '
         IF (WBIAS) THEN
           WRITE(STITLE,5) 'ADF-DELAY=',DELAY,', N=',N,', MU=',MU,', BIAS'
5          FORMAT(A10,I2,A4,I4,A5,F12.10,A6)
         ELSE
           WRITE(STITLE,5) 'ADF-DELAY=',DELAY,', N=',N,', MU=',MU
         ENDIF
         CALL SPLOT(LEFT,RIGHT,PNUM,NN,LT,BNK,RT,BNK,HT,
     X              HU,TITLE,STITLE,XSCALE)
         READ(*,2) WAIT
```

```
        GOTO 3000
C
C*************************************************************************
C
C  PARAMETER CHANGE
C
4000    CALL PADFDELAY(DELAY)
        GOTO 3000

4100    CALL PCONSTANT(MU)
        GOTO 3000

4200    CALL PWEIGHTS(N)
        GOTO 3000

4300    CALL PBIAS(WBIAS)
        GOTO 3000

4400    WRITE(*,*) 'NOT AVAILABLE AT THIS TIME'
        GOTO 3000

4500    CALL PFILTER(FILT,BW)
        GOTO 3000

9000    CONTINUE

        RETURN
        END
C
C
C
C*************************************************************************
C
C  THE FOLLOWING SUBROUTINES ARE QUERIES FOR THE ADAPTIVE NOISE
C  CANCELLER PARAMETERS.  THEY ARE USED IN BOTH THE USER
C  CONTROLLED ANC SUBROUTINE AND THE MAIN SIMULATION ROUTINE.
C
        SUBROUTINE PARAMETERS
        REAL MU,BW
        INTEGER DELAY,N
        CHARACTER CANS*1
        LOGICAL WBIAS,FILT,OFILT,SZC,SWV,SMM,SCORR,SANCZC

1       FORMAT(A)
C
C  NUMBER OF WEIGHTS OF WEIGHT VECTOR
C
        ENTRY PWEIGHTS(N)
        WRITE(*,*) ' '
        TYPE 1, '∫    Number of weights =  '
        READ (*,*) N
```

```
      RETURN
C
C  DELAY OF INPUT TO ADAPTIVE FILTER
C
      ENTRY PADFDELAY(DELAY)
      WRITE(*,*) ' '
      TYPE 1, '∫    Reference input delay =  '
      READ (*,*) DELAY
      RETURN
C
C  RATE OF CONVERGENCE FACTOR
C
      ENTRY PCONSTANT(MU)
      WRITE(*,*) ' '
      TYPE 1, '∫    Gradient proportionality constant =  '
      READ (*,*) MU
      RETURN
C
C  WEIGHT BIAS ENABLING
C
      ENTRY PBIAS(WBIAS)
      WRITE(*,*) ' '
      TYPE 1, '∫    Do you want to include bias for weights?  (Y/N)  '
      READ (*,2) CANS
2     FORMAT(A1)
      IF (CANS.EQ.'Y'.OR.CANS.EQ.'y') THEN
        WBIAS = .TRUE.
      ELSE
        WBIAS = .FALSE.
      ENDIF
      RETURN
C
C  BANDWIDTH ESTIMATOR FOR USER CONTROLLED ANC
C
      ENTRY PFILTER(FILT,BW)
      WRITE(*,*) ' '
      TYPE 1, '∫    Do you want to make a bandwidth estimate (Y/N)  '
      READ (*,2) CANS
      IF (CANS.EQ.'Y'.OR.CANS.EQ.'y') THEN
        FILT = .TRUE.
      ELSE
        FILT = .FALSE.
      ENDIF
      RETURN
C
C  INTERMEDIATE FILTERING ENABLE FOR USER CONTROLLED ANC
C
      ENTRY PFILT1(OFILT)
      WRITE(*,*) ' '
      TYPE 1, '∫    Do you want to enable the intermediate filter?  (Y/N)  '
      READ (*,2) CANS
      IF (CANS.EQ.'Y'.OR.CANS.EQ.'y') THEN
```

```
          OFILT = .TRUE.
        ELSE
          OFILT = .FALSE.
        ENDIF
        RETURN
C
C  ENABLE ESTIMATES BASED ON THE WEIGHT VECTOR
C
        ENTRY PMM(SMM)
        WRITE(*,*) ' '
        TYPE 1, '∫    Do you want to enable the Max/Min estimators?  (Y/N)  '
        READ (*,2) CANS
        IF (CANS.EQ.'Y'.OR.CANS.EQ.'y') THEN
          SMM = .TRUE.
        ELSE
          SMM = .FALSE.
        ENDIF
        RETURN
C
C  ENABLE ESTIMATES BASED ON SIMILARITIES BETWEEN WEIGHT VECTOR AND
C  AUTOCORRELATION ESTIAMTE
C
        ENTRY PCORR(SCORR)
        WRITE(*,*) ' '
        TYPE 3,'∫    Do you want to enable the',
     X      '∫ pseudo-corr. estimators?  (Y/N)  '
3       FORMAT(A,A)
        READ (*,2) CANS
        IF (CANS.EQ.'Y'.OR.CANS.EQ.'y') THEN
          SCORR = .TRUE.
        ELSE
          SCORR = .FALSE.
        ENDIF
        RETURN
C
C  ENABLE ESTIMATES BASED ON ZERO CROSSING ANALYSIS
C
        ENTRY PANCZC(SANCZC)
        WRITE(*,*) ' '
        TYPE 1, '∫    Do you want to enable the ANC-ZC estimators?  (Y/N)  '
        READ (*,2) CANS
        IF (CANS.EQ.'Y'.OR.CANS.EQ.'y') THEN
          SANCZC = .TRUE.
        ELSE
          SANCZC = .FALSE.
        ENDIF
        RETURN

        END
```

```
C*****************************************************************************
C
C       SSADF
C
C       VAX-11 FORTRAN SOURCE FILENAME:          SADF3
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER(S)
C       --------        ----                     -------------
C        0.0            MAY 6, 1985              MARC D. BRACK
C
C*****************************************************************************
C
C       CALLING SEQUENCE
C
C               CALL SSADF(X,LENGTH,MU,N,DELAY,WBIAS,WMEAN,WVAR,
C                          FILT,SZEST,TS,MMEST,BW,SMM,
C                          ERORI,YI,EROR,Y,SCORR,SANCZC)
C
C       PURPOSE
C
C               This subroutine implements the Adaptive Noise
C               Canceller Baud Rate Estimator.  The routine is
C               to be used inconjunction with SBAUD.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               DFT
C               BUTER
C               ZCROSS
C               BITLENGTH
C
C       ROUTINE(S) ACCESSED OR CALLED BY EMBEDDED ROUTINE(S)
C
C               NONE
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               X       Array of input sequence (real)
C
C               LENGTH  Number of elements of X (integer)
C
C               MU      Gradient constant for LMS algorithm (real)
C
C               N       Number of weights in weight vector (integer)
C
C               DELAY   Number of samples for delay (integer)
C
C               WBIAS   Flag for weight bias, TRUE if bias
C                       is to be included (logical)
C
```

```
C               FILT    Flag for intermediate filtering, TRUE if
C                       filtering to be done (logical)
C
C               TS      Sampling intcrval of inpnt seqncnce (real)
C
C               SMM     Flag for bitlength anslysis, TRUE if
C                       analyaia to be donc (logical)
C
C               SCORR   Flag for paeudo-correlation analysis, TRUE
C                       if analysis to be done (logical)
C
C               SANCZC  Flag for zero crossing  analysia, TRUE if
C                       analysis to be done (logical)
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               SZEST   Array for eatimatea made with secondary
C                       adsptive noise csnceller (real)
C
C               MMEST   Array for eatimates msde with intcrmediate
C                       adaptive noise canceller (real)
C
C               BW      Bandwidth estimate of demodulator outpnt (rcal)
C
C               X       Array of intermediately filtered demodnlstor
C                       ontpute (real)
C
C               ERORI   Arrsy of error aequence from intcrmediate
C                       adsptive noise canceller (real)
C
C               YI      Arrsy of adaptive filter outpnt sequencc from
C                       intermediate adaptive noise cancciler (real)
C
C               EROR    Arrsy of crror aeqnence from aecondsry
C                       adaptive noise canceller (real)
C
C               Y       Array of adaptive filter outpnt auquence from
C                       secondary adsptive noise canceller (real)
C
C               WMEAN   Array of mean of filtcr weights (real)
C
C               WVAR    Array of vsrisnce of filter weighta (real)
C
C*************************************************************************
C

      SUBROUTINE SSADF(X,LENGTH,MU,N,DELAY,WBIAS,WMEAN,WVAR,
     X                 FILT,SZEST,TS,MMEST,BW,SMM,
     X                 ERORI,YI,EROR,Y,SCORR,SANCZC)
      IMPLICIT NONE
      INTEGER N,I,J,LENGTH,DELAY,NUM,NUM2,MINJM,MINJV
      INTEGER HFLAG(10),HISTA(2,10,257),MAX,ACCUM
```

```
      REAL MU,W(0:256),WMEAN(0:N),WVAR(0:N),WMEANS(0:256),MAXF
      REAL X(LENGTH),SZEST(10),BW,MMEST(10),MINF,R(512)
      REAL Y(2048),EROR(2048),TS,SBITL(10),TEMPMU,L(512)
      REAL YI(2048),ERORI(2048)
      COMPLEX CW(2048),BUTER
      LOGICAL SMM,FILT,WBIAS,TWO,INTERM,SCORR,SANCZC
      CHARACTER*80 B

      INTERM = .FALSE.
      TEMPMU = MU
      IF(FILT) MU = 0.000000001
C
C*************************************************************************
C
C  ADAPTIVE NOISE CANCELLING SECTION
C
1000  CONTINUE

      DO 100 I=0,N
        WMEAN(I) = 0.
        WMEANS(I) = 0.
        W(I) = 0.0
100   CONTINUE

C
C  LMS IMPLEMENTATION OF ANC
C
      DO 160 J=1,LENGTH
        IF (WBIAS) THEN
          Y(J) = W(0)
        ELSE
          Y(J) = 0.0
        ENDIF

        DO 130 I=1,N
130       IF(J-I+1.GT.0) Y(J) = Y(J) + X(J-I+1)*W(I)

        IF(J+DELAY.LE.LENGTH) THEN
          EROR(J) = X(J+DELAY) - Y(J)
        ELSE
          EROR(J) = -Y(J)
        ENDIF

        W(0) = W(0) + 2.*MU*EROR(J)

        DO 140 I=1,N
140       IF(J-I+1.GT.0) W(I) = W(I) + 2.*MU*EROR(J)*X(J-I+1)
C
C  END OF LMS STRUCTURE
C
C  CALCULATE MEAN AND MEAN SQUARE OF WEIGHTS
C
```

```
          IF(J.GT.N) THEN
            DO 150 I=0,N
              WMEAN(I) = WMEAN(I) + W(I)/(LENGTH-N)
              WMEANS(I) = WMEANS(I) + W(I)**2/(LENGTH-N)
150         CONTINUE
          ENDIF

160     CONTINUE
C
C  END OF LMS ALGORITHM IMPLEMENTATION LOOP
C
        DO 170 I=0,N
170       WVAR(I) = WMEANS(I) - WMEAN(I)**2
C
C  END OF ANC IMPLEMENTATION
C
C*************************************************************************
C
C  OBTAIN PSEUDO-CORRELATION ESTIMATES
C
        IF(SCORR) THEN
          DO 5620 J=0,N
            CW(J+1) = CMPLX(WMEAN(N-J),0.) * FLOAT(J)/N
5620      CONTINUE
          DO 5625 J=N+1,2*N
            CW(J+1) = CMPLX(WMEAN(J-N),0.) * (1.0 - FLOAT(J-N)/N)
5625      CONTINUE
          DO 5630 J=2*N+1,4*N-1
5630        CW(J+1) = CMPLX(0.,0.)
          CALL DFT(CW,4*N,0)
          MINF = 10000000
          DO 5640 J=1,2*N
            L(J) = 20*ALOG10(CABS(CW(J)))
            IF(CABS(CW(J)).LT.MINF) THEN
              MINJM = J
              MINF = CABS(CW(J))
            ENDIF
5640      CONTINUE

          DO 5660 J=0,N
            CW(J+1) = CMPLX(WVAR(N-J),0.) * FLOAT(J)/N
5660      CONTINUE
          DO 5665 J=N+1,2*N
            CW(J+1) = CMPLX(WVAR(J-N),0.) * (1. - FLOAT(J-N)/N)
5665      CONTINUE
          DO 5670 J=2*N+1,4*N-1
5670        CW(J+1) = CMPLX(0.,0.)
          CALL DFT(CW,4*N,0)
          MINF = 10000000
          DO 5680 J=1,2*N
            R(J) = 20*ALOG10(CABS(CW(J)))
            IF(CABS(CW(J)).LT.MINF) THEN
```

```
              MINJV = J
              MINF = CABS(CW(J))
            ENDIF
5680      CONTINUE
        ENDIF
C
C*************************************************************************
C
C  OBTAIN ESTIMATES BASED ON MINIMUM AND MAXIMUM OF MEAN AND VARIANCE
C  AND FROM ZERO CROSSING ANALYSIS
C
        TWO = .TRUE.
        IF(SMM .AND. .NOT.INTERM) THEN
          CALL BITLENGTH(WMEAN,WVAR,N,DELAY,MMEST,TWO)
          MMEST(6) = FLOAT(2*(4*N))/MINJM
          MMEST(7) = FLOAT(2*(4*N))/MINJV
          CALL ZCROSS(EROR,LENGTH,MMEST(10),1)
        ENDIF

        IF(FILT .AND. INTERM) THEN
          CALL BITLENGTH(WMEAN,WVAR,N,DELAY,SZEST,TWO)
          SZEST(6) = FLOAT(2*(4*N))/MINJM
          SZEST(7) = FLOAT(2*(4*N))/MINJV
          CALL ZCROSS(X,LENGTH,MMEST(8),2)
          CALL ZCROSS(Y,LENGTH,SZEST(8),2)
          CALL ZCROSS(EROR,LENGTH,SZEST(10),1)
        ENDIF
C
C*************************************************************************
C
C  IMPLEMENT INTERMEDIATE FILTERING.  THE FILTER BANDWIDTH IS BASED
C  ON THE 3 dB BANDWIDTH FOUND BY TAKING THE DFT OF THE AVERAGE
C  FILTER WEIGHTS.
C
        IF (FILT .AND. .NOT.INTERM) THEN
          INTERM = .TRUE.
          DO 210 I=0,N
210         CW(I+1) = CMPLX(WMEAN(I),0.)

          DO 220 I=N+1,4*N-1
220         CW(I+1) = CMPLX(0.,0.)
C
C  DETERMINE 3 dB BANDWIDTH OF SIGNAL
C
          CALL DFT(CW,4*N,0)

          MAXF = 0
          DO 230 I=1,4*N/2
230         IF(CABS(CW(I)).GT.MAXF) MAXF = CABS(CW(I))
          MAXF = MAXF**2

          DO 240 I=1,4*N/2
```

```
            IF(CABS(CW(4*N/2+1-I))**2.GT.MAXF/2) THEN
              BW = (4*N/2+2-I)/(4*N/2*TS)
              GOTO 250
            ENDIF
240       CONTINUE
250       CONTINUE
C
C  PERFORM FILTERING
C
          DO 320 J=1,LENGTH
320         CW(J) = CMPLX(X(J),0.0)

          CALL DFT(CW,LENGTH,0)
          DO 330 J=1,LENGTH/2
            CW(J) = CW(J)*BUTER(4,BW,FLOAT(J-1)/(TS*LENGTH))
            CW(LENGTH+1-J) = CW(LENGTH+1-J)*BUTER(4,BW,FLOAT(-J)/(TS*LENGTH))
330       CONTINUE
          CW(LENGTH/2+1) = CMPLX(0.,0.)
          CALL DFT(CW,LENGTH,1)

          DO 340 J=1,LENGTH
            ERORI(J) = EROR(J)
            YI(J) = Y(J)
340         X(J) = REAL(CW(J))
C
C  ENTER ANC FOR SPECIFIED MU AFTER INTERMEDIATE FILTERING IS COMPLETE
C
          MU = TEMPMU
          GOTO 1000

        ENDIF

        RETURN
        END
```

```
C*****************************************************************************
C
C       BITLENGTH
C
C       VAX-11 FORTRAN SOURCE FILENAME:          BIT.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER(S)
C       --------        ----                     -------------
C        0.0            MAY 6, 1985              MARC D. BRACK
C
C*****************************************************************************
C
C       CALLING SEQUENCE
C
C               CALL BITLENGTH(WMEAN,WVAR,N,DELAY,BITL,TWO)
C
C       PURPOSE
C               This rontine makes eatiamtes of the bad rate by
C               estimating the bit interval in number of sampling
C               intervals.  The estimate is taken from the weight
C               vector information.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               NONE
C
C       ROUTINE(S) ACCESSED OR CALLED BY EMBEDDED ROUTINE(S)
C
C               NONE
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               WMEAN   Array of length N of mean of the
C                       weight vector (real)
C
C               WVAR    Array of length N of variance of the
C                       weight vector (real)
C
C               N       Number of weight of the weight vector (integer)
C
C               DELAY   Delay of the primary input of the ANC (integer)
C
C               TWO     TRUE if the estimatea involving the
C                       variance are to be made also (logical)
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               BITL  Array containing estimates (real)
C
C*****************************************************************************
```

```
C

      SUBROUTINE BITLENGTH(WMEAN,WVAR,N,DELAY,BITL,TWO)

      IMPLICIT NONE
      INTEGER MEXT(256),FRMIN,ZC
      INTEGER RI,LI,DELAY,N,I,Q1,Q2,Q3,Q4,Q5,Q6,TEMP,FRMAX,L,R,J
      REAL WMEAN(0:N),WVAR(0:N),BITL(10)
      LOGICAL MINMAX,SIDED1,SIDED2,TWO,NSET,NSET1,NSET2,KEY

      DO 100 I=1,10
100     BITL(I) = 1 - DELAY
C
C  FIND ALL RELATIVE MINIMA OF THE WEIGHT MEAN FILE AND ALL RELATIVE
C  MAXIMA OF THE WEIGHT VARIANCE FILE
C
      NSET = .TRUE.
      NSET1 = .TRUE.
      NSET2 = .TRUE.
      DO 200 I=1,N-1
        IF(WMEAN(I).LT.WMEAN(I-1) .AND. WMEAN(I).LT.WMEAN(I+1)) THEN
          MEXT(I) = 1
          IF(NSET) THEN
            NSET = .FALSE.
            FRMIN = I
          ENDIF
        ELSEIF(WMEAN(I).GT.WMEAN(I-1) .AND. WMEAN(I).GT.WMEAN(I+1)) THEN
          MEXT(I) = -1
          IF(NSET1) THEN
            NSET1 = .FALSE.
            FRMAX = I
          ENDIF
        ELSE
          MEXT(I) = 0
        ENDIF
        IF(.NOT.NSET1 .AND. NSET2) THEN
          IF(WMEAN(I)*WMEAN(I+1).LE.0.0) THEN
            ZC = I
            NSET2 = .FALSE.
          ENDIF
        ENDIF
200   CONTINUE

C
C  ONE ESTIMATE IS THE FIRST TIME A RELATIVE MAXIMUM AND A RELATIVE
C  MINIMUM OCCUR AT THE SAME POINT.  IF THIS DOES NOT HAPPEN DEFAULT
C  TO FIRST RELATIVE MINIMUM PAST THE FIRST RELATIVE MAXIMUM
C
      IF (TWO) THEN
        DO 300 I=FRMAX,N
          IF(MEXT(I).EQ.1) THEN
```

```
              CALL FMAX(WVAR,I,0,N,KEY)
              IF(KEY) THEN
                BITL(1) = I
                GOTO 310
              ENDIF
            ENDIF
300       CONTINUE
          BITL(1) = FRMIN
310       CONTINUE
C
C  A SECOND ESTIMATE IS IF A VARIANCE MASIMUM IS WITHIN TWO SAMPLES OF
C  THE MEAN MINIMUM
C
          DO 320 I=FRMAX,N
            IF(MEXT(I).EQ.1) THEN
              CALL FMAX(WVAR,I,2,N,KEY)
              IF(KEY) THEN
                BITL(2) = I
                GOTO 330
              ENDIF
            ENDIF
320       CONTINUE
          BITL(2) = FRMIN
330       CONTINUE
        ENDIF
C
C  THE THIRD ESTIMATE IS THE ABSOLUTE MINIMUM OF THE MEAN
C
        BITL(3) = FRMAX

        DO 400 I=FRMAX+1,N
400       IF(WMEAN(I).LT.WMEAN(BITL(3))) BITL(3) = I
C
C  THE FOURTH ESTIMATE IS THE FIRST RELATIVE MINIMUM LESS THAN ZERO
C  IF NOT FOUND, DEFAULT TO THE FIRST RELATIVE MINIMUM
C
        DO 500 I=FRMAX,N
          IF(MEXT(I).EQ.1 .AND. WMEAN(I).LT.0) THEN
            BITL(4) = I
            GOTO 510
          ENDIF
500     CONTINUE
        BITL(4) = FRMIN
510     CONTINUE
C
C  THE FIFTH ESTIMATE IS THE FIRST ZERO CROSSING AFTER THE FIRST RELATIVE
C  MAXIMUM OF THE MEAN
C
        BITL(5) = ZC
C
C  ADD THE DELAY DIFFERENCE
C
```

```
      DO 800 I=1,10
800     BITL(I) = BITL(I) + DELAY - 1

      RETURN
      END
C
C
C
C
C  THIS SUBROUTINE IS TO DETERMINE IF A MAXIMUM OF THE VARIANCE
C  IS CLOSE TO THE MINIMUM OF OF THE WEIGHT MEAN.
C
C  The minimum of the mean is located at weight number I.  If the
C  maximum occurs within Q samples of I KEY is returned with the
C  value of TRUE, otherwise it is returned with a value of FALSE.
C
      SUBROUTINE FMAX(X,I,Q,N,KEY)

      IMPLICIT NONE
      INTEGER I,Q,N,J
      REAL X(0:N)
      LOGICAL KEY

      KEY = .FALSE.

      DO 100 J=I-Q,I+Q
        IF(J-1.GE.0 .AND. J+1.LE.N-1) THEN
          IF(X(J).GE.X(J-1) .AND. X(J).GE.X(J+1)) KEY = .TRUE.
        ENDIF
100   CONTINUE

      RETURN
      END
```

```
C*****************************************************************************
C
C       ZCROSS
C
C       VAX-11 FORTRAN SOURCE FILENAME:          ZC.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER(S)
C       --------        ----                     -------------
C        0.0            MAY 6, 1985              MARC D. BRACK
C
C*****************************************************************************
C
C       CALLING SEQUENCE
C
C               CALL ZCROSS(SIG,N,ZEST,NEST)
C
C       PURPOSE
C
C               This routine forms an estimete of the baud rete using
C               zero croaaing enelysis.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               NONE
C
C       ROUTINE(S) ACCESSED OR CALLED BY EMBEDDED ROUTINE(S)
C
C               NONE
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               SIG     Arrey of length N of input sequence (reel)
C
C               N       Number of samplea of input (integer)
C
C               NEST    Number of estimates to mekeμ the different
C                       eatimetea are based on different criterie.
C                       NEST cen be either: 1, 2, or 3.  Generally,
C                       ell eatimetes give seme result.
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               ZEST    Arrey of length NEST of estimates (real)
C
C*****************************************************************************
C
      SUBROUTINE ZCROSS(SIG,N,ZEST,NEST)

      IMPLICIT NONE
      INTEGER I,N,DIFF(2048),HIST(2048),J,K,SUCCES,FAIL,NUM,NUM2,EST
```

```
      INTEGER MAXW,NEST
      REAL SIG(N),ZEST(NEST),DIFFA(2,2048),AMIN,TEMP,NORM,AMINT
      REAL RATIO(10),TOL(10),LMAXW

      DO 100 I=1,N
        DIFF(I) = 0
        HIST(I) = 0
100   CONTINUE

      I = 1
      J = 0
      K = 1
C
C  DETERMINE THE LOCATIONS OF THE ZERO CROSSINGS AND RECORD THE
C  DIFFERENCES IN SAMPLES BETWEEN THE CROSSINGS
C
200   DO 210 WHILE(SIG(I)*SIG(I+1).GT.0)
        I=I+1
        IF(I.GT.N-1) GOTO 220
210   CONTINUE

215   IF (SIG(I+1).EQ.0) THEN
        I=I+1
        IF(I.GT.N-1) GOTO 220
        GOTO 215
      ELSE
        I=I+1
      ENDIF

      IF (I.GT.N-1) GOTO 220

      J=J+1
      DIFF(J) = I - K
      K = I
      GOTO 200

220   CONTINUE
C
C  MAKE A HISTOGRAM OF THE DIFFERENCESμ  THE INDEX OF THE HIST ARRAY IS
C  THE DIFFERENCE BETWEEN CROSSINGS, THE VALUE OF HIST FOR A CERTAIN
C  INDEX IS THE NUMBER OF TIMES A DIFFERENCE WHICH EQUALS TO THE INDEX
C  OCCURRED
C
      DO 300 I=1,J
300     HIST(DIFF(I)) = HIST(DIFF(I)) + 1
C
C  MAKE IMPULSE FUNCTION ON REAL LINEμ THE LOCATION OF THE IMPULSES ARE
C  THE AVERAGE OF A GROUP IN THE HISTOGRAM WHERE A GROUP IS A COLLECITON
C  OF ENTRIES NOT HAVING TWO OR MORE ADJACENT EMPTY BINS.  THE WEIGHTS
C  OF THE IMPULSES ARE THE NUMBER OF HISTOGRAM ENTRIES THAT CONTRIBUTED
C  TO THE AVERAGE OF THE RESPECTIVE IMPULSE.  THE ARRAY DIFFA CONTAINS
C  THE LOCATION OF THE IMPULSES IN THE FIRST ROW AND THE WEIGHT IN THE
```

```
C  SECOND ROW
C
       J=1
       NUM2 = 0
       DO 320 I=1,N
C
C  TO DETECT A GROUP, FIRST FIND A NON-EMPTY BIN THAT HAS TWO EMPTY BINS
C  IMMEDIATELY PRECEEDING IT (CARE MUST BE TAKEN IN CHECKING THE FIRST
C  TWO BINS).  THEN GATHER IN BIN ENTRIES FROM THE FIRST BIN OF THE GROUP
C  UNTIL TWO ADJACENT EMPTY BINS ARE FOUND
C
         IF(I.LE.2) THEN
           IF(HIST(1).NE.0 .AND. I.EQ.1) THEN
             NUM = HIST(I)
             DIFFA(1,J) = HIST(I)*I
             K=1
310          IF(.NOT.(HIST(I+K).EQ.0 .AND. HIST(I+K+1).EQ.0)) THEN
               NUM = NUM + HIST(I+K)
               DIFFA(1,J) = DIFFA(1,J) + HIST(I+K)*(I+K)
               K=K+1
               GOTO 310
             ENDIF
             DIFFA(1,J) = DIFFA(1,J)/NUM
             DIFFA(2,J) = NUM
             NUM2 = NUM2 + NUM
             J=J+1
           ENDIF
           IF(HIST(2).NE.0 .AND. HIST(1).EQ.0 .AND. I.EQ.2)THEN
             NUM = HIST(I)
             DIFFA(1,J) = HIST(I)*I
             K=1
312          IF(.NOT.(HIST(I+K).EQ.0 .AND. HIST(I+K+1).EQ.0)) THEN
               NUM = NUM + HIST(I+K)
               DIFFA(1,J) = DIFFA(1,J) + HIST(I+K)*(I+K)
               K=K+1
               GOTO 312
             ENDIF
             DIFFA(1,J) = DIFFA(1,J)/NUM
             DIFFA(2,J) = NUM
             NUM2 = NUM2 + NUM
             J=J+1
           ENDIF
         ELSE
           IF(HIST(I).NE.0 .AND. HIST(I-1).EQ.0 .AND. HIST(I-2).EQ.0) THEN
             NUM = HIST(I)
             DIFFA(1,J) = HIST(I)*I
             K=1
315          IF(.NOT.(HIST(I+K).EQ.0 .AND. HIST(I+K+1).EQ.0)) THEN
               NUM = NUM + HIST(I+K)
               DIFFA(1,J) = DIFFA(1,J) + HIST(I+K)*(I+K)
               K=K+1
               GOTO 315
```

```
              ENDIF
              DIFFA(1,J) = DIFFA(1,J)/NUM
              DIFFA(2,J) = NUM
              NUM2 = NUM2 + NUM
              J=J+1
            ENDIF
          ENDIF
320     CONTINUE

        J=J-1
        AMIN = N
        DO 330 I=1,J
330       AMIN = MIN(AMIN,DIFFA(1,I))

        TOL(1) = 0.05
        TOL(2) = 0.20
        TOL(3) = 0.10
        RATIO(1) = 2
        RATIO(2) = 2
        RATIO(3) = 2

        DO 390 EST=1,NEST
C
C  DETERMINE THE NUMBER OF SUCCESS AND FAILURES OF MEETING THE TOLERENCE
C  CRITERION WHERE WHERE A SUCCES IS WHEN THE LOCATION OF THE IMPULSE
C  CONSIDERED IS AN INTEGRAL DIVISOR OF THE A LOCATION TO THE RIGHT.
C  EACH IMPULSE TO THE RIGHT THAT SATASFIES THE CRITERION CONTRIBUTES ITS
C  WEIGHT TO THE TOTAL NUMBER OF SUCCESS
C
335         SUCCES = 0
            FAIL = 0
            DO 340 I=1,J
              IF(DIFFA(1,I).GE.AMIN) THEN
                TEMP = DIFFA(1,I)/AMIN
                NORM = (TEMP - ANINT(TEMP))/ANINT(TEMP)
                IF(ABS(NORM).LE.TOL(EST)) THEN
                  SUCCES = SUCCES + DIFFA(2,I)
                ELSE
                  FAIL = FAIL + DIFFA(2,I)
                ENDIF
                IF(DIFFA(1,I).EQ.AMIN) NUM = DIFFA(2,I)
              ENDIF
340         CONTINUE
C
C  CHECK THE RATIO CRITERION.  THERE MUST BE A CERTAIN RATIO BETWEEN THE
C  NUMBER OF SUCCESS AND NUMBER OF FAILURES.  IN ADDITION THE NUMBER OF
C  SUCCESS MUST BE GREATER THAN ONE-FOURTH OF THE TOTAL NUMBER OF
C  CROSSINGS AND THE NUMBER OF ENTRIES CONTRIBUTING TO THE IMPULSE
C  BEING CONSIDERED MUST BE GREATER THAN ONE-TENTH OF THE TOTAL NUMBER
C  OF CROSSINGS
C
            IF(SUCCES.GE.RATIO(EST)*FAIL .AND. SUCCES.GE.NUM2/4 .AND.
```

```
     X         NUM.GT.NUM2/10) THEN
             ZEST(EST) = AMIN
           ELSE
             AMINT = AMIN
             AMIN = N
             DO 360 I=1,J
360            IF(MIN(DIFFA(1,I),AMIN).GT.AMINT) AMIN = MIN(DIFFA(1,I),AMIN)

             IF(AMIN.LT.N) THEN
               GOTO 335
             ELSE
C
C  IF NO IMPULSE MEETS THE CRITERIA, CHOOSE THE MAXIMUM WEUGHT IMPULSE
C  AS THE DEFAULT ESTIMATE
C
               MAXW = 0
               DO 370 I=1,J
                 MAXW = MAX(DIFFA(2,I),FLOAT(MAXW))
370            CONTINUE
               DO 380 I=1,J
                 IF (MAXW.EQ.DIFFA(2,I)) THEN
                   LMAXW = DIFFA(1,I)
                   GOTO 385
                 ENDIF
380            CONTINUE
385            ZEST(EST) = LMAXW
             ENDIF
           ENDIF
390      CONTINUE

         RETURN
         END
```

```
C*************************************************************************
C
C       DLD
C
C       VAX-11 FORTRAN SOURCE FILENAME:          DLD.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
```

```
C*************************************************************************
C
C       DLD
C
C       VAX-11 FORTRAN SOURCE FILENAME:          DLD.FOR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER(S)
C       --------        ----                     -------------
C        0.0            JAN 23, 1983             DONALD M. HUMMELS
C        0.1            MAY 6, 1985              MARC D. BRACK
C
C*************************************************************************
C
C       CALLING SEQUENCE
C
C               CALL DLD (TD,FC,FLAG,S,N,TF)
C
C       PURPOSE
C
C               This subrontine models a delay line discriminator.
C               No pre- or post filtering is done.  The resnlting
C               waveform is returned in the input array.
C
C       ROUTINE(S) ACCESSED OR CALLED BY THIS ROUTINE
C
C               DFT
C
C       ROUTINE(S) ACCESSED OR CALLED IN EMBEDDED ROUTINES
C
C               NONE
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               TD=Time delay of discriminator
C               FC=Center freqnency
C               FLAG=Integer representing type of discriminator
C                    0 -- Cos type
C                    1 -- Sin type
C               S=Array containing input signal
C               N=# of points in inpnt array (a power of 2)
C               TF=Inpnt period (=1/fnndamental freq)
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               S=Array containing ontput signal
C
C*************************************************************************
C
        SUBROUTINE DLD(TD,FC,FLAG,S,N,TF)
```

```
      COMPLEX S(2048),S2(2048),FACTOR
      INTEGER FLAG
C
C   PERFORM THE TIME DELAY
C
      DO 19 I = 1,N
      S2(I) = S(I)
19    CONTINUE
      CALL DFT(S2,N,0)

      DO 20 I = 1,N/2
        ANGLE = -(2*3.141592*FLOAT(I-1)/TF*TD)
        FACTOR = CMPLX(COS(ANGLE),SIN(ANGLE))
        S2(I) = S2(I) * FACTOR
        ANGLE = -(2*3.141592*FLOAT(-I)/TF*TD)
        FACTOR = CMPLX(COS(ANGLE),SIN(ANGLE))
        S2(N+1-I) = S2(N+1-I) * FACTOR
20    CONTINUE
      CALL DFT(S2,N,1)
C
C  FIND THE INPUT FOR THE POST-FILTER
C
      ANGLE = 3.14159*(2*FC*TD + .5*FLAG)
      FACTOR = CMPLX(COS(ANGLE),SIN(ANGLE))
      DO 40 I=1,N
        S(I) = S(I)*CONJG(S2(I))*FACTOR
        S(I) = CMPLX(4.*REAL(S(I)),0.)
40    CONTINUE

      RETURN
      END
```

References

1. B. Widrow, et al., 'Adaptive Noise Cancelling: Principles and Applications,' Proceedings of the IEEE, Vol. 63, No. 12, December 1975.

2. F. W. Ratcliffe, A Performance Analysis of the Delay Line Discriminator, Ph. D. Disaertation, Kansaa State University, Manhattan, Kansaa, 1984.

3. M. Schwartz, W. R. Bennett and S. Stein, Communication Syatems and Techniques, McGraw-Hill Book Co., New York, New York, 1966.

4. D. R. Hnmmels and F. W. Ratcliffe, 'Modeling Digital Commnnication Systema for Numerical Evaluation of Error Rate, Power Spectrum, Eye Pattern and Envelope,' Motorola Technical Report, Government Electronics Division, Scottsdale, Arizona, Jnly 31, 1980.

5. J. A. Cadzow, 'Spectral Estimation: An Overdetermined Rational Model Equation Approach,' Proceedings of the IEEE, Vol. 70, No. 9, September 1982.

DETERMINATION OF THE BAUD RATE OF AN FSK SIGNAL USING ADAPTIVE NOISE CANCELLING TECHNIQUES

by

MARC DAVID BRACK

B.S., Kansas State University, 1984

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

reqnirements for the degree

MASTER OF SCIENCE

Department of Electrical and Compnter Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1985

Abstract

The traditional method of FSK signal band rate estimation is to use a delay-line discriminator to demodulate the signal and then perform a zero-crossing analysis to obtain the band rate. This study proposes an alternate method which implements an adaptive noise canceller. The weight vector of the adaptive noise canceller, upon convergence, can be used directly to determine the band rate. However, the input signal is random making convergence difficult to obtain or sustain. The adaptive noise canceller can be used indirectly to improve the performance of the zero-crossing analysis. The way in which it does this is by making a rough but consistent estimate of the bandwidth of the message. This parameter is not known due to the surveillence situation the receiver is intended to be used in. Then the delay-line discriminator output can be filtered to produce a cleaner input for the zero-crossing analysis which will improve performace.

The performance of the traditional zero-crossing analysis and the direct and indirect implementation of the adaptive noise canceller was judged using a Monte Carlo simulation. The results showed that both adaptive noise cancelling methods performed better than the zero-crossing analysis alone.