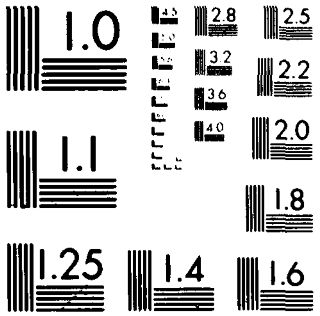
AD-A090 128 RHODE ISLAND UNIV KINGSTON DEPT OF ELECTRICAL ENGIN--ETC F/6 9/2
APPLICATIONS OF EFFICIENT SUBSET SELECTION TO DIGITAL FILTERING--ETC(U)
AUG 80 J KHORAMMI, D TUFTS AFOSR-77-3174

UNCLASSIFIED AFOSR-TR-80-0954 NL

1 OF 2

AD A
090128



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

LEVEL II (2) F

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A090128

(19) REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) AFOSR TR- 80-0954	2. GOVT ACCESSION NO. AD-A090128
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) APPLICATIONS OF EFFICIENT SUBSET SELECTION TO DIGITAL FILTERING AND TO SIGNAL RESOLUTION	5. TYPE OF REPORT & PERIOD COVERED (9) FINAL rept.,
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) JAFIR KHORAMMI ~~AND~~ DONALD TUFTS	8. CONTRACT OR GRANT NUMBER(s) (15) AFOSR-77-3174
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Rhode Island Department of Electrical Engineering Kingston, RI 02881	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F (16) 2304/A3 (17) A3
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, DC 20332	12. REPORT DATE (11) August 1980
	13. NUMBER OF PAGES 99
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (12) 104	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The subset selection algorithm is extended to search for a best subset from a large set of complex-valued basis functions. This algorithm is used to design digital finite-duration impulse response (FIR) filters having fewer coefficients than conventional FIR filters. An optimum conventional FIR filter is derived which has best uniform spacing of the fixed number of samples which are to be used, and examples are presented which show that for the same number of coefficients, the complex-subset-selection filter can give better results than the optimum conventional filter. The complex subset selection method is also →(over→)

UNCLASSIFIED JUB

305570

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) cont

applied to estimation of the frequencies of sinusoids in the presence of noise. A windowing technique is introduced to increase the efficiency and accuracy of the algorithm for frequency estimates. The results are compared with Cramer-Rao bounds.

UNIVERSITY OF RHODE ISLAND

DEPARTMENT OF ELECTRICAL ENGINEERING

FINAL REPORT

ON

GRANT AFOSR-77-3174

APPLICATIONS OF EFFICIENT SUBSET SELECTION TO DIGITAL FILTERING AND TO SIGNAL RESOLUTION

JAFIR KHORAMMI

AND

DONALD TUFTS

AUGUST 20, 1980

ABSTRACT

The subset selection algorithm is extended to search for a best subset from a large set of complex-valued basis functions. This algorithm is used to design digital finite-duration impulse response (FIR) filters having fewer coefficients than conventional FIR filters. An optimum conventional FIR filter is derived which has best uniform spacing of the fixed number of samples which are to be used, and examples are presented which show that, for the same number of coefficients, the complex-subset-selection filter can give better results than the optimum conventional filter.

The complex subset selection method is also applied to estimation of the frequencies of sinusoids in the presence of noise. A windowing technique is introduced to increase the efficiency and accuracy of the algorithm for frequency estimates. The results are compared with Cramer-Rao bounds.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This is a report of work done on digital filter design and sinusoidal-tone estimation by the complex subset selection method. In the complex subset selection method, a set of complex basis functions are chosen and a best subset is selected to represent a desired function. The desired functions investigated are an ideal transfer function of a low-pass digital filter, and the sum of two sinusoidal tones. The summed squared-magnitude error is used as the criterion for subset selection.

One can design a digital lowpass FIR filter by use of the window method (1), frequency sampling (2),and other techniques (3),(4). In all of these methods, the input data elements are uniformly spaced in time with respect to one another. Instead, we search for the best M data elements, which may or may not be uniformly located, to minimize the least squares error in approximating the desired transfer function. We compare the result with a filter having the same number of coefficients, but uniformly spaced locations.

We will also investigate sinusoidal-tone parameter estimation, which is a very important subject in communication systems. Linear prediction (5) and spectral estimation (6) are two well known methods for estimating parameters of tones, such as frequencies. Both of these methods can produce biased estimates of the the signal parameters when more than one signal is present (7). We use complex subset selection to estimate frequencies of tones without such bias. The effect of additive noise is discussed, and the results are compared with Cramer-Rao bounds computed by Rife (8). A windowing technique is applied to increase the accuracy and the efficiency of the algorithm. By "windowing technique", we mean that a small set of basis fuctions is always chosen over potentially productive intervals, which will be searched for a best subset, and other intervals are ignored. That is, potentially productive intervals are first located, and only these intervals are searched for the desired frequencies.

The basis for complex subset selection is reviewed in the following paragraphs:

Regression analysis may be broadly defined as the analysis of relationships among variables. It is one of the most widely used statistical tools because it provides a simple method for establishing a linear functional relationship among variables. The relationship is expressed in the form of an equation connecting the response or dependent variable $\hat{y}$, and one or more independent variables, $x_1, x_2, \ldots\ldots, x_N$. The regression equation takes the form

$$\hat{\tilde{y}} = b_1 x_1 + b_2 x_2 + \ldots\ldots + b_N x_N \quad (1.1)$$

where $b_1, b_2, \ldots\ldots, b_N$ are called the regression coefficients and are determined from the data. However, in some applications, one needs to select M out of the N ($M < N$) (independent variables to best approximate y by

$$\hat{\tilde{y}} = a_1 u_1 + a_2 u_2 + \ldots\ldots + a_M u_M \quad (1.2)$$

where $U = \{u_1, \ldots\ldots, u_M\}$ is a subset selected from $X = \{x_1, \ldots\ldots, x_N\}$. Examining all $\binom{N}{M} = N!/(N-M)!M!$ different subsets to select the best subset involves a lot of computation, especially when N is large.

Hocking and Leslie (H-L) (10) have developed an algorithm for finding the best subset. In their algorithm, they first test the variables which are superior. That is, the ones which, when omitted from the linear regression equation, result in a large residual error. If a best subset is obtained, they stop. Otherwise, the variables with smaller residual error are tested until a best subset is obtained. They have developed a criterion for examining only a fraction of all possible subsets and stopping after the best subset is selected. The squared error criterion was suggested by Mellow (11) to be used in linear regression models.

The H-L algorithm was further investigated by

LeMotte and Hocking (12) for estimation cf the linear regression coefficients to fit a surface of the form in (1.1). They have also introduced some other improvements to reduce the amount of computation. Later, Levasseur and Lewis (13) considered antenna design as a partial basis problem. They, too, applied a generalization of the H-L algorithm. The value of extending the work of Levasseur and Lewis to complex-valued functions was pointed out by D. W. Tufts. He also suggested applying the method to the synthesis of digital filters and to estimation of signal parameters in the presence of noise.

In the following chapter, we restate the methods of H-L and Levasseur and Lewis for complex subset selection to make it useful for the problems considered in this thesis. Digital filter synthesis will be discussed in chapter 3. And chapter 4 is devoted to estimation of the frequencies of sinusoidal signals by the method of complex subset selection.

CHAPTER 2

COMPLEX SUBSET SELECTION

2.1- Introduction

Our objective is to extend an algorithm for partial basis selection to complex-valued basis functions. In this case, we have a large set of exponentials with imaginary arguments as basis functions. the desire is to approximate a given function by choosing a best subset of the basis set with minimum approximation error. The least squares error is used as the criterion for subset selection. The algorithm is designed such that usually only a fraction of all possible subsets is examined, and it comes to a halt as soon as a best subset is selected. Hence, considerable amounts of time and computation can usually be saved by not examining all possible subsets.

This algorithm is suitable for such problems as digital filter design and sinusiodal-tone estimation. This is due to the representation of the transfer function of a nonrecursive digital filter and the waveform of multiple sinusoidal signals by linear combinations of exponential functions with imaginary arguments.

Some work has been done on partial basis selection of real-valued functions by Levasseur and Lewis (13). Their derivation is restated in the next section for complex partial basis selection.

2.2- The Algorithm

It is desired to approximate a given function D(x) by a best subset of M elements chosen from N linearly independent complex basis functions relative to some error noım. The least squares norm is discussed because of its later use, but other norms can be applied, as well.

Let us define a set of N complex basis functions as a vector H, and a subset with M (M < N) elements selected from H as a vector U.

$$H^T = \left[h_1, h_2, \ldots\ldots\ldots, h_N\right] \qquad (2.2.1)$$

$$U^T = \left[u_1, u_2, \ldots\ldots\ldots, u_M\right] \qquad (2.2.2)$$

$$U \subset H$$

$$h_i(t) = EXP(Jw_i t) \qquad i=1,2,\ldots\ldots,N \qquad (2.2.3a)$$

Or

$$h_i(w) = EXP(Jwt_i) \qquad i=1,2,\ldots\ldots,N \qquad (2.2.3b)$$

The subset parameter vector is defined as

$$G = \left[g_1, g_2, \ldots\ldots\ldots, g_M\right] \tag{2.2.4}$$

in which the components of G are distinct and for some value of i

$$g_n = \begin{cases} w_i & \text{for some value of i when } h_i \text{ is defined by (2.2.3a)} \\ t_i & \text{for some value of i when } h_i \text{ is define by (2.2.3b)} \end{cases} \tag{2.2.5}$$

for $n=1,2,\ldots,M$ and $i=1,2,\ldots,N$

"T" means transpose, and $J=\sqrt{-1}$.

The basis functions can have two versions in (2.2.3) so that the approximation can be done either in the time domain or the frequency domain. One can replace time with position, and have basis functions related to location in space. The approximating function A(x) is of the form

$$A(x) = \sum_{k=1}^{M} c_k u_k = C^T U \tag{2.2.6}$$

in which the coefficient vector C is defined by

$$C^T = \left[c_1, c_2, \ldots\ldots\ldots, c_M\right] \tag{2.2.7}$$

The error of approximation is defined to be

$$E(C,G) = \left\| D(x) - A(x) \right\|^2 = \int W(x) \left| D(x) - A(x) \right|^2 dx \tag{2.2.8}$$

and $W(x)$ is a weight function which can be 1 or any suitable function. The error $E(C,G)$ is to be minimized by choice of the subset parameter vector G and the coefficient vector C. Choosing G is equivalent to choosing a subset of M basis functions.

It is impractical to search over all $\binom{N}{M}$ subsets to find the best one, esecially when N is large which is the case in most situations. Hence, the following method is adopted which is based on the Hocking and Leslie (10) algorithm, as extended by Levasseur and Lewis (13) to a partial basis problem.

In the adopted method one searches for the potential basis functions which contribute the most to the error function by their deletion. That is, a set of error functions, E_i's is obtained where E_i corresponds to the error due to a best approximation by using all basis functions excluding the ith one.

$$E_i = \underset{\{c\}}{\text{Min.}} \left\| D(x) - \sum_{\substack{k=1 \\ k \neq i}}^{N} c_k h_k \right\| \tag{2.2.9}$$

Then, the basis functions are re-ordered in a way that their corresponding E_i errors are in ascending order.

$$E_1 \leqslant E_2 \leqslant \ldots\ldots \leqslant E_N \qquad (2.2.10)$$

The basis functions coresponding to the large E_i's are usually the potential basis functions which are good candidates to be kept at the begining of the search. Let r be a set of r= N - M basis functions to be deleted and let ER be the corresponding approximation error. Also, define p to be initially equal to r. Now, the search can be implemented in the following three steps.

STEP-1. Delete the r basis functions corresponding to E_i's with subscripts less than, or equal to r $(i \leqslant r)$, and compute the minimum error ER . If ER $\leqslant E_{p+1}$, go to step-3. If ER $> E_{p+1}$, follow step-2.

STEP-2. Increment p by 1, and compute the error for all $\binom{p}{r}$ combinations of deleting r out of p basis functions where deleted basis functions correspond to E_i's with $i \leqslant p$. Let ER be the minimum error obtained from all possible $\binom{p}{r}$ combinations. If ER $\leqslant E_{p+1}$, go to step-3; otherwise(ER $> E_{p+1}$) re-execute step-2.

STEP-3. If ER $\leqslant E_{p+1}$, a best subset has been achieved, and no further search is necessary. Hence the selected basis functions are those which have been kept for best approximation with minimum error either in step-1 or step-2.

NOTE: Step-1 is to start the search while step-2 is executed over and over until a best subset is obtained.

The search algorithm was outlined in the above three steps, but it was not explained why one stops whenever ER $\leqslant E_{p+1}$. Now, this will be discussed.

Assume p=r + q with $1 \leqslant q < M$ to be a general case. Let $I = \{1, 2, \ldots p\}$ be a set of p indices taken from the set $\{1, 2, \ldots\ldots, N\}$ with largest element p. Let I_q be a set of r indices selected from I which result in minimum error by their deletion, and let $I' \subset \{1, 2, \ldots\ldots, N\}$ be any set of r indices with an element $i' > p + 1$. The intention is to show that if

$$ER = \underset{\{c\}}{\text{Min.}} \left\| D - \sum_{\substack{i=1 \\ i \notin I_q}}^{N} c_i h_i \right\| \leqslant E_{p+1} \qquad (2.2.11)$$

then,

$$ER \leqslant \left\| D - \sum_{\substack{i=1 \\ i \notin I'}}^{N} c_i h_i \right\| = E_{I'} \qquad (2.2.12)$$

for any I'.

The statement (2.2.12) can be proved as follows:

$$ER \leqslant E_{p+1}$$

$$E_{p+1} \leqslant E_{i'} \qquad \text{BECAUSE } i' > p+1$$

$E_{I'}$ is the minimum error obtained by deleting i'th basis function plus some other ones, as well. Hence,

$$E_{i'} < E_{I'}$$

and,

$$ER \leqslant E_{p+1} \leqslant E_{i'} < E_{I'}$$

Thus, at any point, if the statement $ER \leqslant E_{p+1}$ is satisfied, no further search would be necessary. However, if $ER \leqslant E_{p+1}$ is never satisfied, all the possible subsets will be examined to select the best cne.

To show the power of this algorithm, some numerical examples will be presented in the following chapters. Also, a windowing technique will be applied to improve the efficiency of the algorithm.

CHAPTER 3

FILTER SYNTHESIS

3.1- Introduction

A finite-duration impulse response (FIR) filter can be represented with a set of coefficients and delays as depicted in Figure 3.1. The frequency response of such a filter is of the form shown by equation (3.1.1) . The object is to design an FIR filter to have fewer coefficients than equivalent conventional FIR filters, while meeting the same performance objectives.

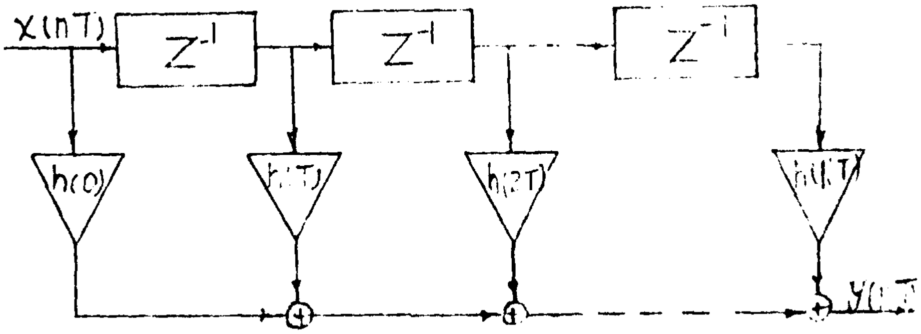


Figure 3.1

$$H(w) = \sum_{n=0}^{N} h(nT).EXP(JwnT) \qquad (3.1.1)$$

Instead of processing all the input data over a given record length or a subset of uniformly spaced data elements, we choose the best M data elements for achieving the desired performance. The fewer the coefficients, the fewer multiplications will be performed, since each coefficient corresponds to a multiplication directly. The complex subset selection method is used to achieve the goal of selecting the best subset of data elements, from a given interval of input data, to be processed by an FIR filter.

3.2- Formulation Of Design

The desired FIR filter is a lowpass filter(LPF) having a passband, a stopband, and a 'don't care' (3) region as depicted in figure 3.2. The desired frequency transfer function, D(w), is real valued and symmetric about frequency zero. Assuming that sampled data is to be processed with a memory no greater than 2qT seconds, or (2q+1) samples, we can approximate D(w) using all the available data by a FIR filter with transfer function A(w) given in formula (3.2.1).

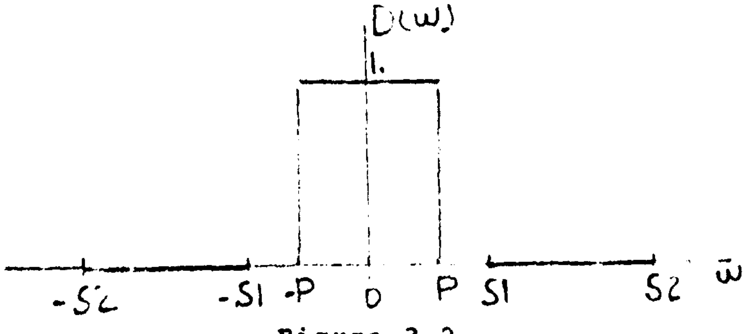


Figure 3.2

$$A(w) = \sum_{n=-q}^{q} a(nT)\ \mathrm{EXP}(JwnT) \qquad (3.2.1)$$

There is no loss of generality by choosing zero phase for D(w), because one can easily synthesize a linear phase filter by shifting the impulse response of this filter properly. Thus, the ideal filter is defined by equation (3.2.2).

$$D(w) = \begin{cases} 1 & , \ |w| \leqslant p \\ 0 & , \ s1 \leqslant |w| \leqslant s2 \\ \text{don't care,} & \text{otherwise} \end{cases} \qquad (3.2.2)$$

Let the observation time be kept fixed. That is, q and T of (3.2.1) satisfy the formula

$$2qT = \text{constant.} \qquad (3.2.3)$$

Using complex subset selection, D(w) can be approximated by a subset having fewer coefficients in the same observation time ($M < 2q+1$) shown by equation (3.2.4),

$$H(w) = \sum_{n=1}^{M} h(n)\ \mathrm{EXP}(JwT_n) \qquad (3.2.4)$$

where the T_n 's may or may not be uniformly spaced. Using the method of complex subset selection, the large set of basis functions of the form FXP(JwnT) are the potential basis functions of (3.2.1) in which

$$\text{nth basis function} = \text{EXP}(JwnT) \qquad (3.2.5)$$

$$n = -q,\ldots,-1,0,1,\ldots,q$$

$$N = 2q + 1 \,. \qquad (3.2.6)$$

A best subset having M coefficients is selected from these N basis functions such that the least squares error between D(w) and H(w) is minimized, i.e.,

$$\underset{\{h(n),T_n\}}{\text{Min. Error}} = \int W(w)\left|D(w) - \sum_{n=1}^{M} h(n)\ \text{EXP}(JwT_n)\right|^2 dw \qquad (3.2.7)$$

where

$$W(w) = \begin{cases} 1 & , \ w| \leqslant p \text{ and } s1 \leqslant w \leqslant s2 \\ 0 & , \ p < w < s1 \end{cases} \qquad (3.2.8)$$

and the T_n 's are selected out of the set (-qT,...,-T,0,T,...,qT).

From the orthogonality principle (14) for optimum choice of the h(n)'s, for given values of the T_n's, it follows that

$$\int W(w)\,(D(w) - \sum_{n=1}^{M} h(n)\ \text{EXP}(JwT_n))\ .\ \text{EXP}(-JwT_k)\ dw = 0.$$

$$\text{for } k = 1,2,\ldots,M \qquad (3.2.9)$$

which can be rewritten as a set of M linear equations of the form

$$AH = B \qquad (3.2.10)$$

with

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots\cdots a_{1M} \\ a_{21} & \cdot\cdot & \\ a_{M1} & \cdots\cdots\cdots\cdots & a_{MM} \end{bmatrix} \qquad (3.2.11)$$

$$H^{T} = \left[h(1), h(2), \ldots\ldots\ldots, h(M) \right] \qquad (3.2.12)$$

$$B^{T} = \left[b_{1}, b_{2}, \ldots\ldots\ldots, b_{M} \right] \qquad (3.2.13)$$

where

$$a_{ij} = \begin{cases} (2/(T_{i}-T_{j}))\,(\,\mathrm{Sin}(s2(T_{i}-T_{j}))-\mathrm{Sin}(s1(T_{i}-T_{j})) \\ \qquad\qquad + \mathrm{Sin}(p(T_{i}-T_{j}))\,) & i\neq j \\ 2(s2-s1+p) & i=j \end{cases} \qquad (3.2.14)$$

$$b_{i} = \begin{cases} (2/(T_{i})).\mathrm{Sin}(pT_{i}) & T_{i}\neq 0 \\ 2p & T_{i}=0 \end{cases} \qquad (3.2.15)$$

$$H = A^{-1} B \qquad (3.2.16)$$

For the detailed derivation of the equations (3.2.9) through (3.2.17), see appendix A.

Since A and B are both real valued matrices, the solution to the equation (3.2.16) results in real coefficients for the filter (i.e. h(n)'s are all real). Solving for H and substituting in the error function of (3.2.7) is equivalent to simplifying (3.2.7) using (3.2.9). The result is

$$\underset{\{h(n)\}}{\text{Min. Error}} = 2p - 2\sum_{n=1}^{M} h(n)\ \text{Sin}(pT_n)/T_n \qquad (3.2.17)$$

in which $\{h(n)\}$ satisfies (3.2.9), or equivalently (3.2.16).

The results obtained here will be compared with a filter having uniformly spaced values of T with frequency response given by (3.3.1), and we claim that the complex subset selection will always result in a better approximation to a desired ideal filter, or at least an approximation as good as the one obtained by the uniformly-spaced method. The uniformly-spaced method is a search for only a constrained set of possible solutions, and many other possibilities are simply ignored. On the other hand, whenever complex subset selecton is used, all possible approximations are considered . If it happens that the best solution is uniformly spaced, it will be included as one of the possible solutions to be examined.

3.3- Filter With Uniformly-spaced Time Samples

The filter obtained by the complex subset selection will be compared with another filter which uses uniformly-spaced time samples. For this filter, the observation time (2qT) is fixed and the number of the coefficients (M) are the same as before . But, the basis function parameters must be spaced uniformly from an initial location 'c' as illustrated in figure 3.3.

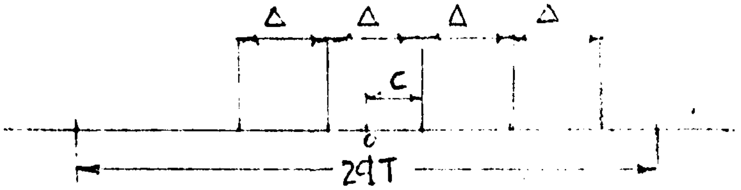


Figure 3.3

The frequency response of this filter is of the form shown by equation (3.3.1), assuming that M is an odd integer of the form M=21+1

$$H_u(w) = \sum_{n=-1}^{1} h_u(n)\ EXP(J(n\Delta+c)w) \qquad (3.3.1)$$

If, for fixed values of Δ and c, we minimize the error given by

$$\text{Error} = \int W(w) \left| D(w) - H_u(w) \right|^2 dw \qquad (3.3.2)$$

by choice of the coefficients h(n) , We obtain, using the orthogonality principle (14), the set of simultaneous linear equations

$$A'H = B' \qquad (3.3.3)$$

in which A' and B' can be evaluated from (3.2.9) by letting $T_n = n\Delta + c$. The coefficients $\{h_u(n)\}$ are also real since A' and B' are real-valued matrices.

In the search for the best c, it was found numerically that the best c is zero for the best Δ for the cases tried (four examples are illustrated in table 3.1 where the C was searched in the interval 0.0 to 3.0 with increment .1, and best Δ was searched in the interval 0. to 2.5 with increment .1). Therefore, we tried to prove this analytically. However, after investigating many different methods, we could only show that C=0 is a good candidate for best C, and could not prove its uniqueness. One method is to differentiate the error function with respect to c, and set it to zero. The error function is of the form

$$E = 2P - 2\sum_{n=-1}^{1} h_u(n) . Sin((n\Delta+c)P)/(n\Delta+c)$$

$$= 2P - B'^{T} A'^{-1} B' \qquad (3.3.4)$$

in which A' and B' are matrices from equation (3.3.3). The B' matrix is dependent on c, whereas the A' matrix is independent of c. Differentiating (3.3.4) with respect to c and setting it to zero, we have

$$dE/dc = - (dB'^{T}/dc) A'^{-1} B' - B'^{T} A'^{-1} (dB'/dc)$$

$$= - 2B'^{T} A'^{-1} (dB'/dc) = 0 \qquad (3.3.5)$$

$$b_n = 2Sin((n\Delta+c)P)/(n\Delta+c)$$

$$db_n/dc = 2(PCos((n\Delta+c)P)/(n\Delta+c)$$

$$- Sin((n\Delta+c)P)/((n\Delta+c)**2)) \qquad (3.3.6)$$

$$dE/dc = - 8\sum_{n=-1}^{1}\sum_{m=-1}^{1} Sin((n\Delta+c)P)/(n\Delta+c)$$

$$\left[PCos((m\Delta+c)P)/(m\Delta+c) - Sin((m\Delta+c)P)/((m\Delta+c)**2)\right]$$

$$.a'(n,m) \qquad (3.3.7)$$

where a'(n,m are elements of A'^{-1}. Since A' is a toeplitz matrix(see equations (3.2.11) and (3.2.14)), it follows that

$$a'(n,-m) = a'(-n,m) \qquad (3.3.8)$$

and

$$dE/dc = \sum_{n=1}^{1} \sum_{m=-1}^{1} \mathrm{Sin}(n\Delta P)/n\Delta .(P\mathrm{Cos}(-m\Delta P)/(-m\Delta)$$

$$-\mathrm{Sin}(-m\Delta P)/((-m\Delta)**2)).a'(n,-m)$$

$$+ \mathrm{Sin}(-n\Delta P)/n\Delta .(P.\mathrm{Cos}(m\Delta P)/m\Delta$$

$$- \mathrm{Sin}(m\Delta P)/((m\Delta)**2)).a'(-n,m) = 0. \qquad (3.3.9)$$

Thus c=0 is a solution of (3.3.5), because all the terms in (3.3.7) cancel one another for c=0. Since we have not been able to show analytically the uniqueness of c, we search for the best c as we do for the best Δ.

In the next section, some examples are presented in order to explain the differences of the two filters.

3.4- Results And Examples

Presented here are the results of the comparison of the two filters discussed in sections 3.2 and 3.3. Many examples have been investigated, and only two will be presented here to show the differences in the two filters. In both examples, 21 basis functions are used with arguments wnT, (n=-q,...,0,...,q , and T=.5), and the subset size is 5 (M=5). The results are tabulated in tables 3.2 and 3.3 for which the following observations are made. Of course, these observations are drawn from the investigation of a large set of examples.

a) The least squares error is smaller for the subset selection filter than the uniformly spaced parameter filter.

b) The magnitude response is flatter in the passband region in the subset selection filter case.

c) The magnitude response rolls off sharply in the transition-band for the subset selection filter.

d) However, the maximum sidelobe level is smaller for the uniform parameter filter.

The above observations can be seen from tables 3.2 and 3.3 and also from magnitude response plots of figure 3.4 and 3.5. The frequency response is symmetric about frequency 1/2T, where T is the sampling period. Since the smallest sampling period for 21 basis functions is T=.5, all frequency responses are plotted over frequencies 0 through 1 hertz (1/2(.5) = 1) .

It was also noticed that the nonzero coefficient locations are moved toward (or away from) the midpoint by increasing (or decreasing) the passband width. That is, the coefficient locations are around the midpoint for a wide passband, and farther away for a narrow passband. The plot in figure 3.6 shows this effect where the abscissa is the second term in equation (3.3.2) ($\sum_{n=-\ell}^{\ell} h_u(n) \cdot \mathrm{Sin}(n\Delta)p/n\Delta$), and the ordinate is Δ . Each curve is for a different passband value while each peak represents the best Δ for that specific passband value.

TABLE 3.1a

M=5
P=.15   S1=.2   S2=.5

C	ERRCR FOR Δ=0.5	ERROR FOR Δ=1.0	ERROR FOR Δ=1.5
0.0	.144201	.092526	.162134
0.1	.147396	.093267	.161562
0.2	.156949	.095505	.159873
0.3	.172754	.099286	.157136
0.4	.194671	.104690	.153471
0.5	.222486	.111818	.149040
0.6	.255916	.120801	.144044
0.7	.294664	.131788	.138728
0.8	.338348	.144936	.133360
0.9	.386559	.160419	.128241
1.0	.438853	.178403	.123694
1.1	.494751	.199062	.120051
1.2	.553746	.222544	.117660
1.3	.615326	.248993	.116866
1.4	.678959	.278519	.118011
1.5	.744113	.311209	.121422
1.6	.810252	.347113	.127416
1.7	.876849	.386238	.136278
1.8	.943399	.428554	.148267
1.9	1.00941	.473985	.163603
2.0	1.07444	.522403	.182472
Best C	0.0	0.0	1.3
ERROR	.144201	.092526	.116866

TABLE 3.1b

N=5
P=.1 S1=.2 S2=.5

C	ERROR FOR Δ=0.5	ERROR FOR Δ=1.0	ERROR FOR Δ=1.5
0.0	.103650	.032061	.036306
0.1	.104927	.032794	.036345
0.2	.108742	.034986	.036466
0.3	.115084	.038536	.036681
0.4	.123900	.043731	.037008
0.5	.135168	.050260	.037473
0.6	.148798	.058206	.038109
0.7	.164734	.067545	.038957
0.8	.182873	.078256	.040061
0.9	.203121	.090311	.041470
1.0	.225369	.103673	.043241
1.1	.249488	.118313	.045435
1.2	.275352	.134188	.048111
1.3	.302826	.151259	.051339
1.4	.331759	.169477	.055175
1.5	.361995	.188798	.059697
1.6	.393391	.209166	.064967
1.7	.425772	.230528	.071049
1.8	.458985	.252826	.078011
1.9	.492862	.276001	.085911
2.0	.527239	.299990	.094807
Best C	0.0	0.0	0.0
Error	.103650	.032061	.036306

TABLE 3.1c

M=5
P=.15 S1=.225 S2=.5

C	ERROR FOR Δ=0.5	ERROR FOR Δ=1.0	ERROR FOR Δ=1.5
0.0	.093154	.068523	.154084
0.1	.096298	.068842	.153555
0.2	.105711	.069827	.152005
0.3	.121291	.071544	.149489
0.4	.146899	.074114	.156113
0.5	.170340	.077695	.142028
0.6	.203352	.082489	.137417
0.7	.241646	.088724	.132498
0.8	.284865	.096650	.127518
0.9	.332622	.106542	.122751
1.0	.384496	.118675	.118493
1.1	.440033	.133334	.115049
1.2	.498750	.150784	.112741
1.3	.560160	.171278	.111889
1.4	.623744	.195045	.112813
1.5	.688995	.222276	.115825
1.6	.755392	.253120	.129281
1.7	.822430	.287677	.129281
1.8	.889607	.326001	.140256
1.9	.956452	.368084	.154369
2.0	1.02250	.413856	.171806
Best C	0.0	0.0	1.3
Error	.093154	.068523	.111889

TABLE 3.1d

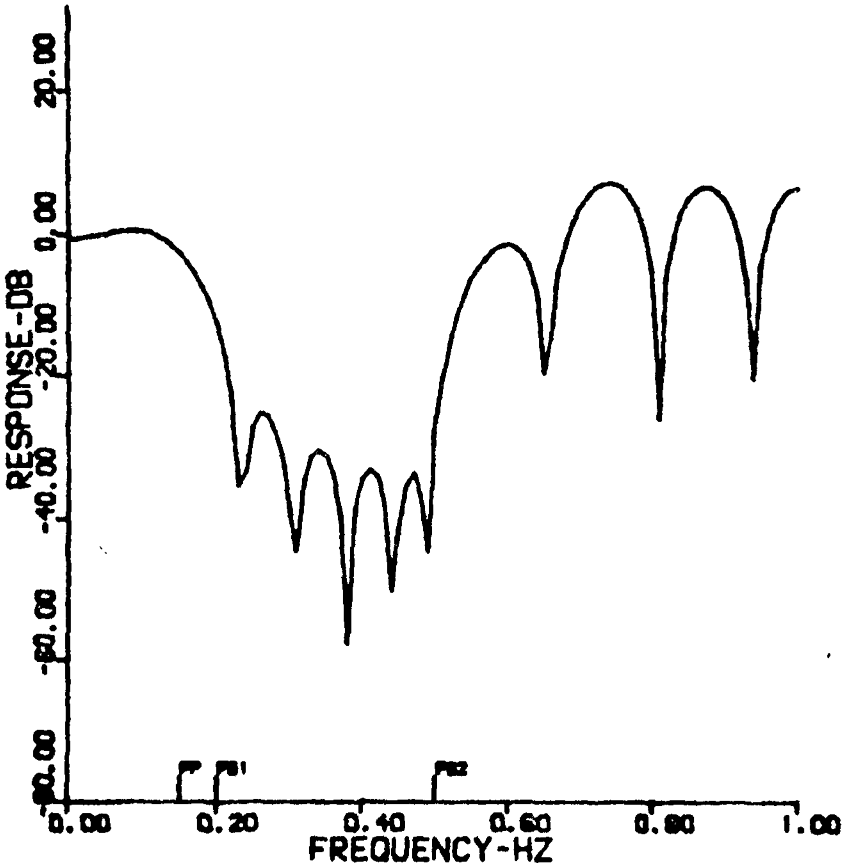
N=5
P=.1 S1=.225 S2=.5

C	ERROR FOR Δ=0.5	ERROR FOR Δ=1.0	ERROR FOR Δ=1.5
0.0	.062489	.018249	.036055
0.1	.063793	.018802	.036085
0.2	.067678	.020460	.036176
0.3	.074139	.023223	.036341
0.4	.083124	.027090	.036599
0.5	.094604	.032058	.036975
0.6	.108495	.038125	.037506
0.7	.124734	.045289	.038225
0.8	.143224	.053546	.039182
0.9	.263868	.062889	.040426
1.0	.186553	.073311	.042016
1.1	.211153	.084809	.044006
1.2	.237535	.097367	.046464
1.3	.265566	.110980	.049453
1.4	.395094	.125629	.053040
1.5	.325968	.141302	.057292
1.6	.358029	.157980	.062277
1.7	.391109	.175643	.068061
1.8	.425056	.194265	.074711
1.9	.459692	.213823	.082287
2.0	.494855	.234284	.090847
Best C	0.0	0.0	0.0
Error	.062489	.018249	.036055

TABLE 3.2

N=21 M=5
f_P=.15 f_{S1} =.2 f_{S2}=.5

	Filter with N coefficients	Uniform parameter filter	Subset selection filter
coefficient locations	nT n=-10,...,10	nΔ 0,±1.0,±2.0	Tn 0,±1.5,±4.5
Max. passband level(db)	.677	1.3	.73
magnitude(db) at f_P	-2.6	-4.5	-3.5
magnitude(db) at f_{S1}	-12.9	-10.1	-17.1
Max. sidelobe level (db)	-25.2	-24.3	-21.0
least squares error	.2098836E-01	.92527E-01	.7460785E-01



Frgure 3.4a- Frequency response of filter having 21 coefficients with $f_p=.15$, $f_{s1}=.2$, and $f_{s2}=.5$.

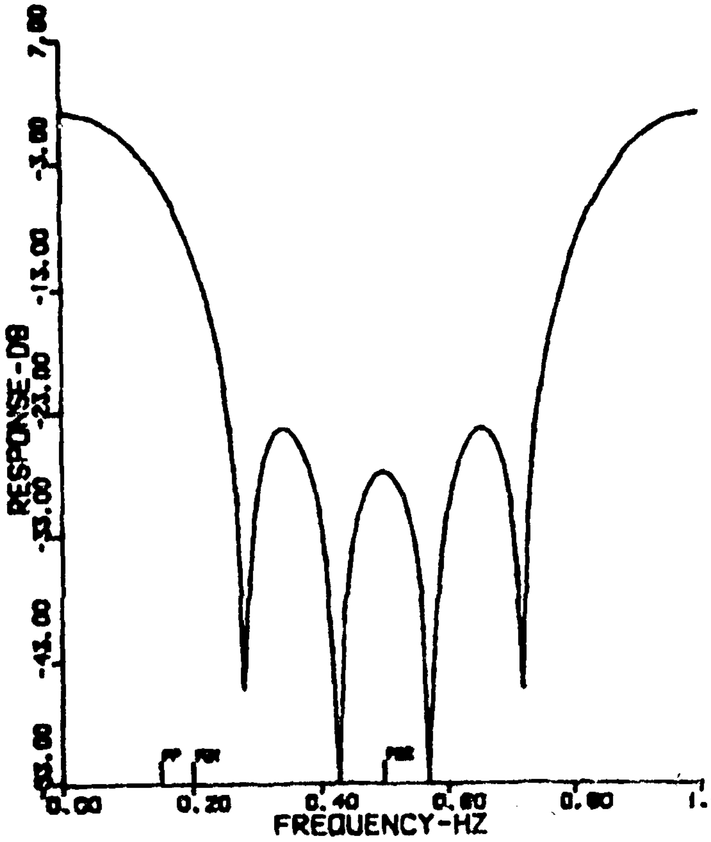


Figure 3.4b- Frequency response of uniform parameter filter having 5 coefficients with f_p=.15, f_{s1}=.2, and f_{s2}=.5 .

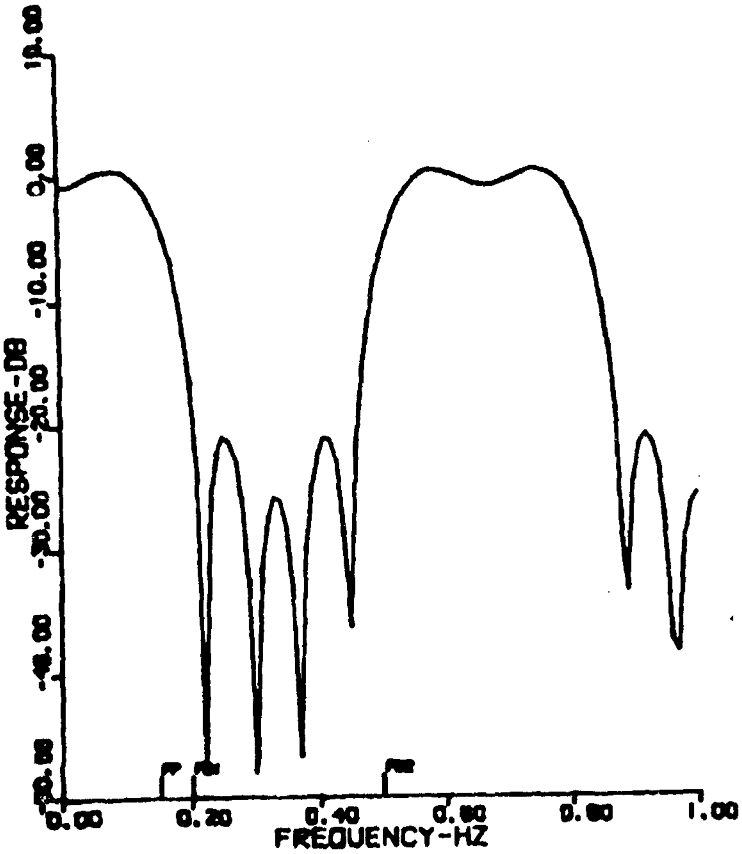


Figure 3.4c- Frequency response of subset selection filter having 5 coefficients with f_p=.15, f_{s1}=.2, and f_{s2}=.5 .

TABLE 3.3

N=21 M=5

f_p=.20 f_{S1} =.3 f_{S2} =.5

	Filter with N coefficients	Uniform parameter filter	Subset selection filter
coefficient location	nT n=-10,...,10	nΔ 0,±0.5,±1.0	T_n 0,±1.0,±3.0
Max. passband level (db)	.30	1.2	.47
magnitude(db) at f_p	-.93	-3.8	-1.7
magnitude(db) at f_{S1}	-21.8	-12.4	-15.1
Max. sidelobe level(db)	-32.7	-27.0	-25.1
least squares error	.2613066E-02	.79270E-01	.1487637E-01

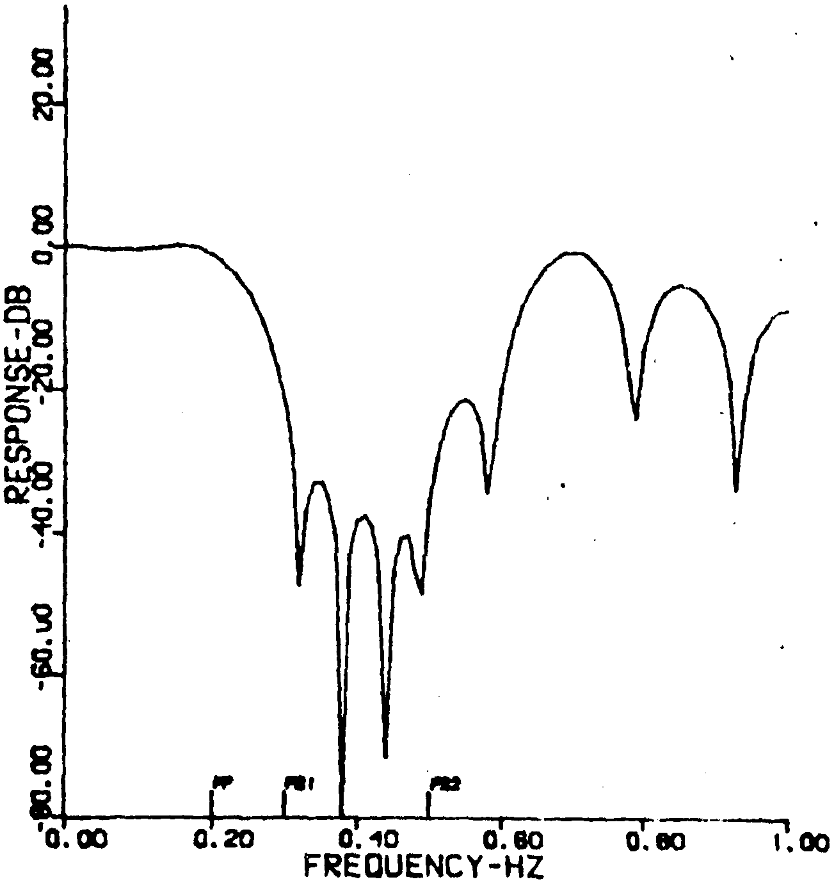


Figure 3.5a- Frequency response of filter having 21 coefficients with f_p=.2, f_{s1}=.3, and f_{s2}=.5 .

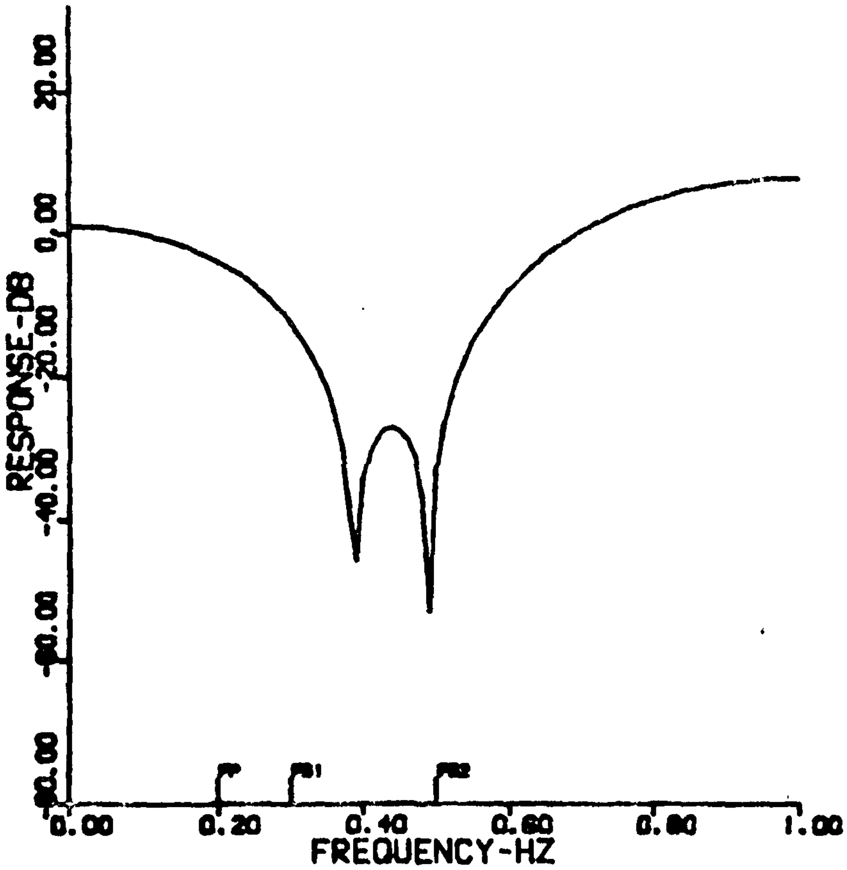


Figure 3.5b- Frequency response of uniform parameter filter having 5 coefficients with f_p .2, f_{s1} .3, and f_{s2} .5 .

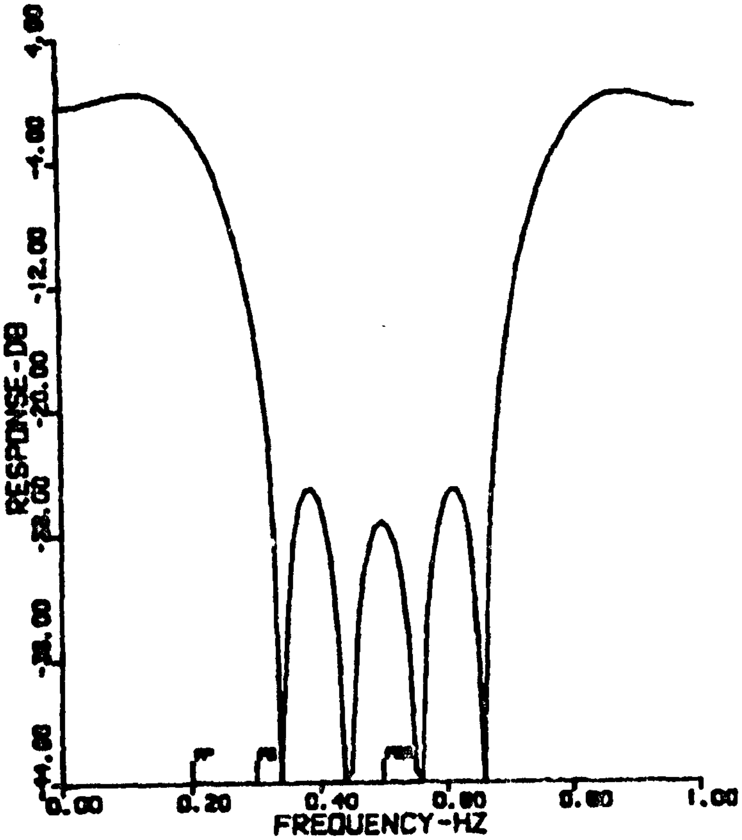


Figure 3.5c- Frequency response of subset selection filter having 5 coefficients with f_p=.2, f_{s1}=.3, and f_{s2}=.5 .

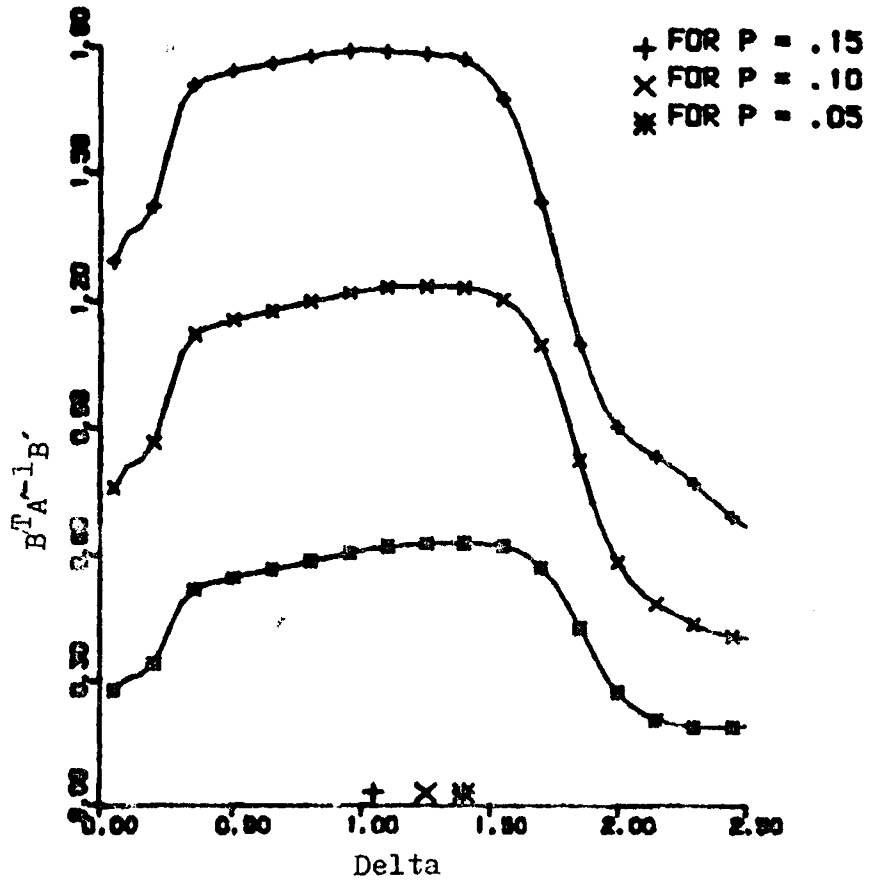


Figure 3.6- $B'^T A'^{-1} B'$ (variable part of the error function) for different values of delta and passband width.

CHAPTER 4
ESTIMATION OF THE FREQUENCIES OF SINUSOIDS

4.1- Introduction

This chapter is devoted to the problem of estimation of the frequencies of sinusoidal tones by the complex subset selection method. In this case, the function to be approximated contains some tones with specific amplitudes and frequencies in the presence or absence of noise. We consider the case of two sinusoidal tones, but the method can be extended to more tones at the cost of more computation.

The framework of this problem will be developed in section 4.2; while the frequency estimation will be presented in section 4.3 . The presence of Gaussian noise in the data will be investigated in section 4.4, and finally section 4.5 will contain some numerical results.

4.2- Two Tone Approximation

The signal tc be approximated contains two sinusiodal tones with each tone having amplitude and frequency (A1,F1) and (A2,F2), respectively. A1 and A2 can be either real cr ccmplex.

$$S(t) = A1\ EXP(J2\pi f_1 t) + A2\ EXP(J2\pi f_2 t) \qquad (4.2.1)$$

A data vector y is formed from N samples of above function (4.2.2).

$$Y^T = \left[y(0), y(1), \ldots\ldots, y(N-1) \right] \qquad (4.2.2)$$

$$y(n) = S(nT)$$

The complex exponential basis functions used here are $h_1(t), h_2(t), \ldots\ldots, h_L(t)$ with arguments of the form $2\pi f_I t$, i.e.,

$$h_I(t) = EXP(J2\pi f_I t) \qquad \text{for } I=1,2,\ldots,L \qquad (4.2.3)$$

A vector is formed for each basis function with N samples taken from each of them .

$$H_I^T = \left[h_I(0), h_I(1), \ldots, h_I(N-1) \right] \qquad \text{for } I=1,2,\ldots,L$$

with (4.2.4)

$$h_I(n) = EXP(J2\pi f_I n)$$

and

$$0 < f_1 < f_2 < \ldots\ldots < f_L < 1$$

Due to the periodicity of exponentials with an imaginary arguments, f_L should be less than one. That is

$$EXP(J2\pi f_L n) = EXP(J2\pi(f_L-1)n)$$

Having H_I's for basis functions, we want to approximate the data vector Y by the complex subset selection method with minimum least squares error. In another words, we want to choose M functions from the L basis functions to best represent the data vector, Y in order to minimize the sum cf the squares of the errors. Assume Y is the data approximation vector of the form

$$Y = \sum_{k=1}^{M} c_k u_k = UC \qquad (4.2.5)$$

where U is a vector formed from M-column vectors U_k's and C is a M-dimensional complex coefficient vector.

$$C^T = \left[c_1, c_2, \ldots\ldots c_M\right] \qquad (4.2.6)$$

$$U = \left[U_1 \vdots U_2 \vdots \ldots\ldots \vdots U_M\right] \qquad (4.2.7)$$

$$U_k^T = \left[u_k(0), u_k(1), \ldots\ldots u_k(N-1)\right] \qquad (4.2.8)$$

$$u_k(n) = EXP(Jw_k n) \qquad (4.2.9)$$

$$w = \left[w_1, w_2, \ldots\ldots\ldots, w_M\right]$$

Note U_K's are chosen from H_I's , and w_K's are selected from $2\pi f_I$'s. The error function is

$$\underset{\{C,w\}}{\text{Min. Error}} = E = \left\|Y - \hat{Y}\right\| \qquad (4.2.10)$$

where $\|\cdot\|$ means the L2 norm of the vector.
To minimize the error one has to find the optimum coefficient vector and frequency vector to be used in specifying $\hat{Y}$.

For the optimum coefficient vector , the ort oqonality principle indicates that Y-Y must be orthogonal to every U_l^* with l=1,.....,M .(* means conjugate transpose).

$$U_l^*(Y - \hat{Y}) = 0 \qquad \text{for } l=1,2,\ldots\ldots,M \qquad (4.2.11)$$

$$U_l^*\hat{Y} = U_l^*Y \qquad \text{for } l=1,2,\ldots\ldots,M \qquad (4.2.12)$$

$$U_l^*UC = U_l^*Y \qquad \text{for } l=1,2,\ldots\ldots,M \qquad (4.2.13)$$

Let $A = U^*U$, and $B = U^*Y$,THEN

$$AC = B \qquad (4.2.14)$$

where

$$A = \begin{bmatrix} U_1^* U_1 & U_1^* U_2 \; \ldots\ldots \; U_1^* U_M \\ \ldots & \\ U_M^* U_1 \; \ldots\ldots\ldots\ldots\ldots & U_M^* U_M \end{bmatrix}$$

with

$$U_1^* U_k = \sum_{n=0}^{N-1} EXP(J(w_k - w_1)n)$$

$$B^T = \left[U_1^* Y, U_2^* Y, \ldots\ldots\ldots, U_M^* Y \right] \qquad (4.2.15)$$

$$b_1 = U_1^* Y = \sum_{n=0}^{N-1} y(n) \; EXP(-Jw_1 n) \qquad (4.2.16)$$

The solution of the linear equation (4.2.14) will give the optimum coefficient vector C. Thus, the error function becomes

$$\begin{aligned} E^2 &= \left\| Y - \hat{Y} \right\|^2 = (Y - \hat{Y})^* (Y - \hat{Y}) \\ &= Y^* (Y - \hat{Y}) - \hat{Y}^* (Y - \hat{Y}) \\ &= Y^* (Y - \hat{Y}) - C^* U^* (Y - \hat{Y}) \qquad (4.2.17) \end{aligned}$$

The right side of the above equation can be simplified by using equation (4.2.11).

$$E^2 = Y^*(Y - \hat{Y}) = |Y|^2 - Y^*\hat{Y}$$

$$= |Y|^2 - Y^*UC = |Y|^2 - B^*C$$

$$= |Y|^2 - B A^{-1} B \qquad (4.2.18)$$

$$E^2 = |Y|^2 - B^*C = \sum_{n=0}^{N-1} \left(|y(n)|^2 - \sum_{k=1}^{M} c_k\, y(n)\, \mathrm{EXP}(-Jw_k n) \right) \qquad (4.2.19)$$

Let us choose the frequencies f_I's of the basis functions to be uniformly spaced. Hence , the functions will be of the form

$$h_I(n) = \mathrm{EXP}(J2\pi fIn) \qquad (4.2.20)$$

such that $\Delta f=1/L$ is the difference between two consecutive grid points. The goodness of the approximation is dependent on L and N, namely, the number of the basis functions and the number of the samples. Since $\Delta f=1/L$, any increase in L will result in having denser grid points which in turn will increase the possibility that f1 and f2 to be on the grid points. f1 and f2 being on the grid points means that two of the basis functions can exactly approximate the data vector. On the other hand , large N means having more data points to process which will also result in better approximation. But L cannot get too large, because the amount of the computation will get impractical for large L. Thus , the windowing technique described below will be applied whenever very dense grid points are needed.

4.3- Frequency Estimation

In some applications one has to estimate the frequencies of the tones as accurately as possible. In those cases, we set the number of the basis functions M in the selected subset to be equal the number of the frequencies contained in the signal. For example, when we are estimating two sinusiodal tones we set M=2. But a very dense set of grid points is needed in order for the estimation to be accurate, and having a very dense grid points is impractical computationally since the amount of the computation will increase significantly as the number of the grid points becomes large. To get around this deficiency, we introduce the windowing technique below.

The windowing technique is applied whenever the grid points (number of basis functions) are large, and the search for a best subset needs too much computer processing. In those cases, a best subset is selected from a less dense grid as a first step. In second step, a denser grid is chosen around the selected subset grid points of step one. As step three, a new best subset is selected from the new grid points of step two. Then, step two and three repeated until no changes are noticed in the selected subset and the error function. One should notice that in the first attempt, potential basis functions are selected where in the second attempt a new set of basis functions are established which are densely populated around those selected potential basis functions, and the best subset is selected. Graphically, the windows in Figure 4.1 indicate where the new densely populated grid points are to be set.

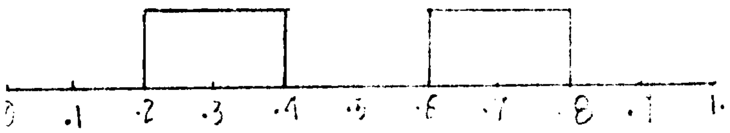


Figure 4.1

To clarify the windowing technique, let us take a look at the two-tone signal, with frequencies f1=.25 and f2=.725. Choose only ten grid points uniformly spaced from zero to one, for the first step. Thus, the basis functions are

$$h_I(n) = EXP(J2\pi fIn) \qquad I=0,1,\ldots.,9$$

with L=10 ,and grid points

$$(0.,0.1,0.2,\ldots\ldots,0.9)$$

of which 2 are selected as potential functions. Assume pth=.3 and qth=.7 are selected. In the next step, we use the following ten new basis functions around the pth, and qth :

$$h_I(n) = EXP(J2\pi(p+\frac{\Delta f}{2}(I-2))n) \qquad I=0,1,2,3,4$$

$$h_I(n) = EXP(J2\pi(q+\frac{\Delta f}{2}(1-7))n) \qquad 1=5,6,7,8,9$$

with L=10, and grid points

(.2,.25,.3,.35,.4,.6,.65,.7,.75,.8).

Note the new grid points are one-half of the previous step. The complex subset selection algorithm is again used to select two potential functions from these new basis functions. In this case, second and nineth (p=.25,q=.75) are selected. Again, choose another ten basis functions with grid points around .25,and .75

$$h_I(n) = EXP(J2\pi(p+\frac{\Delta f}{4}(I-2))n) \qquad I=0,1,..4$$

$$h_I(n) = EXP(J2\pi(q+\frac{\Delta f}{4}(1-7))n) \qquad 1=5,6,..9$$

with L=10, and new grid points

(.2,.225,.25,.275,.3,.675,.7,.725,.75,.775).

Continue as before until the selected frequencies and the error do not change very much for a denser grid points. In this case .25 and .725 are the selected frequencies . The above example is listed in Table 4.4.

If at any time, two windows overlap, that is, two selected frequencies are close to one another, use one wide window instead of two as depicted below.

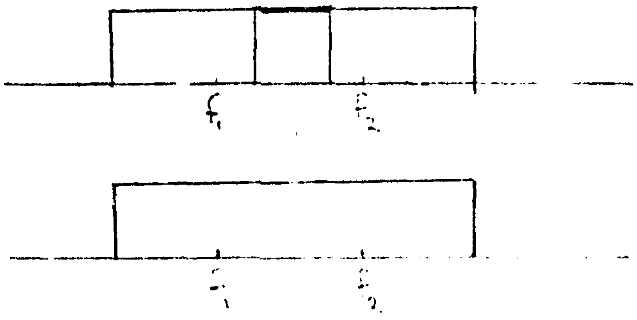


Figure 4.2

The program listings for the two frequency estimation by subset selection and windowing technique are in appendix B.

4.4- The Effect Of Additive Noise

The approximation of a two-tone signal was investigated in secticn 4.2. But, it was for deterministic signals, and in almcst all practical applications the received signal is purturbed by some kind of noise. In here, we investigate the additive Gaussian noise which is the general case for most systems.

Recall the data vector Y (4.2.2) which was obtained from a deterministic signal, only. Now we form a new data vector from signal vector S with noise vector V added to it.

$$Y = S + V \qquad (4.4.1)$$

$$V^T = \left[v(0), v(1), \ldots\ldots\ldots, v(V-1) \right] \qquad (4.4.2)$$

Note that each vector in (4.4.1) is N-dimensional. Assume V consists of complex, independent, zero-mean Gaussian noise components with known variance.

By a close look at linear equation (4.2.1), one will realize that the noise will affect the B vector , only. That is, each row of B is changed by a amount due to the inner product of the noise vector and the corresponding U vector (where subscript corresponds to Ithe row.)

$$B = U^* Y = U^* S + U^* V \qquad (4.4.3)$$

Therefore, the same procedure as before is used to find the coefficient vector , selected basis fuction parameters,and the error function with new data vector.

The effects of Gaussian noise in tone parameter estimation has been well investigated by David C. Rife (8}. He has used a multiparameter generalization of the Cramer-Rao bound (C-R bound) to establish lower bounds for variances. He has also shown that the single-tone bounds are the minimum values attained by two-tone bounds, and they are attained when the two tones have wide frequency separation. Thus, one can use the single-tone bounds (4.4.4) for two-tone bounds whenever two frequencies are far apart.

$$\mathrm{var}(\hat{w}_i) \geqslant (12\sigma^2)/(b_i^2 T^2 N(N^2-1)) \qquad (4.4.4)$$

where

$\hat{w}_i$	frequency estimate in radians
σ^2	noise variance
b_i	tone amplitude
T	time between two samples
N	number of samples

In the next section, this C-R bound is compared with the measured accuracy in frequency estimation which is obtained by applyinq the complex subset selection algorithm to simulated data.

4.5 Results and Examples

Presented here are numerical results for tone estimation by the complex subset selection method. The examples are chosen from three groups to show the power of the complex subset selection method, the windowing technique for frquency estimation, and the effect of the noise.

In the first example, the signal contains two tones with amplitudes and frequencies (A1=1.,A2=1.), (F1=.2,F2=.4), respectively. Twenty basis functions were used where the frequency separation of the basis functions was .05 (f=.05 , grid point separation). Subsets of size 5 (M=5) were searched, and the best one was selected after only examining 136 subsets. This is a small fraction qf all possible subsets (20!/(15!5!) = 15504). However, the two frequencies were exactly on the grid pcints. This example is illustrated in table 4.1 along with the re-ordered E_L errors. For the second example, the amplitudes and frequencies were (A1=1.,A2=1.) and (F1=.2 , F2=.585). Fifteen basis functions with frequency separation .05 were used. In this case, cne cf the frequencies was off the grid points, and subset size 8 were used. Frequency F1 was selected exactly along with frequencies around F2. Note that frequency .6 was selected with a large magnitude which is the closest frequency from the set to F2. The subsets examined were 1716 which are less than 2' per cent of the total possible subsets (15!/(8!7!) = 6435). This example is illustrated in table 4.2 .

The third example was picked to show the improvement obtained by windowing tichnique. Thus, the amplitudes and frequencies were chosen to be the same as example 2. In this case, the selected frequencies were .2 and .58437 after examining only 271 subsets. It was observed that the windowing technique offers better accuracy while the number of subsets examined are kept small. This example is illustrated in table 4.3 along with three more examples in table 4.4 to show the efficiency of this techique. Note that the results are very accurate, even for two tones with small frequency separation.

The next example is an illustration of frequency estimation in the presence of Gaussian noise. Five different signal-to-noise ratios (SNR) were used, and in each case the frequency estimates were averaged over 100 runs to evaluate the mean and the standard deviations. The results were tabulated in table 4.5 along with their coresponding C-R bounds. For high SNR'S, the standard deviations of the estimates were consistant with the C-R bounds, but they differ for low SNR's due to the occurrence of the outliers. Outliers are extreme observations, and they create great difficulty. When we encounter one, our first suspicion is that the observation resulted from a mistake or other extraneous effect, and hence should be discarded. A major reason for discarding it is that under the least squares method, a fit curve is pulled disproportionatly toward an outlying observation because of the squared diviations is minimized. This could cause a misleading fit if indeed the outlier obsevation resulted from a mistake or other extraneous cause. On the other hand, outliers may convey significant information, as when an outlier occurs because of an interaction with another independent variable omitted from the model.

Two frequency estimation has recently been investigated by Kumaresan (15) in a different fashion. In his study, Kumaresan searches over the entire error surface to locate the minimum error, and therefore, the corresponding two frequencies. The same example has been run by his method and the results are listed in table 4.6 for comparison. Both methods give relatively identical results, and consistant with C-R bounds.

A small example with 10 runs is listed in table 4.7 to show how the output frequencies differ from the true values for each run. Two more examples are illustrated in tables 4.8 and 4.9 to show the estimation behavior of the two frequencies when the frequency separation is very small. The results in table 4.8 are consistant with the C-R bounds whereas the results in table 4.9 for stronger tone are lower than the C-R bounds due to the use of information about the tones magnitudes. That is, we have suplied more information than assumed in the C-R bounds for the estimation of the frequencies.

The C-R bounds are for unbiased estimation. We assume that our method, too, gives unbiased estimation since our results are consistant with C-R bounds, and try to show that this assumption is correct by analyzing the results.

We assume, as a working hypothesis, that the frequency error has a normal distribution with zero mean and standard deviation SDT. Then if our frequency error mean falls in the interval -SDT to SDT, we could say that the data is consistant with our zero-mean hypothesis to that accuracy. Since we have 100 independent runs, the probability density function of the frequency error is a chi-square distributicn with 100 degrees of freedom which can be approximated accurately by a normal distibution (16) having the same mean and variance. Let us assume that our experimental standard deviation (SD) is in error by +20% from the true SDT (i.e. SDT = (1±.2)SD). And let us choose the worst case, namely SDT = (1-.2)SD . If the frequency error mean is in the interval -SDT to SDT , we will say that with high probability it is an unbiased estimation. By checking table 4.5, we observe that even for low SNR , the frequency error mean falls in that interval. For instance, when SNR = 5db;

$$F1 - \hat{F}1 = .906E-3$$

$$SD1 = .772E-2$$

$$SDT = (1-.2)SD1 = .6176E-2$$

$$SDT/(F1 - \hat{F}1) = 6.82$$

$$-SDT < F1 - \hat{F}1 < SDT$$

and therefore, our assumption is probably correct.

TABLE 4.1

```
NO. OF SAMMLES =    8
A1=0.1000D 01      A2=0.1000D 01
F1=0.2000D 00      F2=0.4000D 00
N=20
E( 3)=0.1703591D-13   NO.  1 SMALLEST ERROR
E( 4)=0.2899026D-12   NO.  2 SMALLEST ERROR
E( 2)=0.5724064D-12   NO.  3 SMALLEST ERROR
E( 5)=0.8619378D-12   NO.  4 SMALLEST ERROR
E( 6)=0.2004521D-11   NO.  5 SMALLEST ERROR
E( 1)=0.3271099D-11   NO.  6 SMALLEST ERROR
E( 7)=0.8995882D-11   NO.  7 SMALLEST ERROR
E( 8)=0.2952868D-10   NO.  8 SMALLEST ERROR
E( 9)=0.9718023D-10   NO.  9 SMALLEST ERROR
E(20)=0.1602580D-09   NO. 10 SMALLEST ERROR
E(10)=0.1609581D-09   NO. 11 SMALLEST ERROR
E(11)=0.1655543D-09   NO. 12 SMALLEST ERROR
E(12)=0.1662708D-09   NO. 13 SMALLEST ERROR
E(13)=0.1664244D-09   NO. 14 SMALLEST ERROR
E(14)=0.1664624D-09   NO. 15 SMALLEST ERROR
E(15)=0.1664727D-09   NO. 16 SMALLEST ERROR
E(16)=0.1664743D-09   NO. 17 SMALLEST ERROR
E(18)=0.1664748D-09   NO. 18 SMALLEST ERROR
E(19)=0.1664815D-09   NO. 19 SMALLEST ERROR
E(17)=0.1664869D-09   NO. 20 SMALLEST ERROR
N=20  M= 5
NO. OF SUBSETS EXAMINED =   136
D( 1)=0.2000D 00      C( 1)=(0.1000D 01 , 0.7664D-15 )
D( 2)=0.4000D 00      C( 2)=(0.1000D 01 , 0.1628D-14 )
D( 3)=0.8000D 00      C( 3)=(-.7860D-14 , -.7946D-14 )
D( 4)=0.8500D 00      C( 4)=(0.1166D-13 , -.1017D-13 )
D( 5)=0.9000D 00      C( 5)=(0.2896D-14 , 0.1049D-13 )
ERROR=  0.9073D-18
```

TABLE 4.2

```
NO. OF SAMPLES = 10
  A1 =0.20000D 01
  A2 =0.10000D 01
  F1 =0.20000D 00
  F2 =0.58500D 00
  E( 8)=0.1150485D-05  NO.  1 SMALLEST ERROR
  E(10)=0.1152479D-05  NO.  2 SMALLEST ERROR
  E( 6)=0.1154274D-05  NO.  3 SMALLEST ERROR
  E( 7)=0.1170815D-05  NO.  4 SMALLEST ERROR
  E( 9)=0.1179624D-05  NO.  5 SMALLEST ERROR
  E(13)=0.1238354D-05  NO.  6 SMALLEST ERROR
  E( 1)=0.1276429D-05  NO.  7 SMALLEST ERROR
  E(15)=0.1293429D-05  NO.  8 SMALLEST ERROR
  E(14)=0.1298690D-05  NO.  9 SMALLEST ERROR
  E( 2)=0.1327334D-05  NO. 10 SMALLEST ERROR
  E( 5)=0.1353965D-05  NO. 11 SMALLEST ERROR
  E( 3)=0.1435813D-05  NO. 12 SMALLEST ERROR
  E(11)=0.1568995D-05  NO. 13 SMALLEST ERROR
  E(12)=0.1861392D-05  NO. 14 SMALLEST ERROR
  E( 4)=0.1992818D-05  NO. 15 SMALLEST ERROR
 ERROR= 0.18303D-05
  D( 1)=0.2000D 00    C = (0.99997D 00 -.24532D-03)
  D( 2)=0.4000D 00    C = (-.44293D-02 0.77722D-02)
  D( 3)=0.4500D 00    C = (-.36676D-01 -.29364D-01)
  D( 4)=0.5000D 00    C = (0.10543D 00 -.95977D-01)
  D( 5)=0.5500D 00    C = (0.22795D 00 0.34696D 00)
  D( 6)=0.6000D 00    C = (0.69676D 00 -.31457D 00)
  D( 7)=0.6500D 00    C = (0.22999D-01 0.84135D-01)
  D( 8)=0.7000D 00    C = (-.12046D-01 0.13250D-02)
  NO. OF SUBSETS EXAMINED =    1716
```

TABLE 4.3

A1=1.0　　　A2=1.0
F1=.200　　 F2=.585

RUN	$\widehat{F1}$	$\widehat{F2}$	ERROR
1	0.2000	0.60000	.68822E-1
2	0.2000	0.57500	.31433E-1
3	0.2000	0.58750	.19749E-2
4	0.2000	0.58437	.12570E-3
5	0.2000	0.58437	.12570E-3

TOTAL SUBSETS EXAMINED = 271

TABLE 4.4

A1=1.0 A2=1.0
No. of samples = 10

	Case I	Case II	Case III
	F1=.2250	F1=.2500	F1=.1750
	F2=.3125	F2=.7250	F2=.5875
D1	0.2000	0.3000	0.2000
D2	0.3000	0.7000	0.6000
Error	.2194E 0	.6768E 0	.2376E 0
WINDOW 1			
D1	0.2500	0.2500	0.1500
D2	0.3000	0.7500	0.6000
Error	.3485E-1	.1827E 0	.2037E 0
WINDOW 2			
D1	0.2250	0.2500	0.1750
D2	0.3000	0.7250	0.5750
Error	.2367E-1	.2000E-5	.4829E-1
WINDOW 3			
D1	0.2250	0.2500	0.1750
D2	0.3125	0.7250	0.5875
Error	.2000E-5	.2000E-5	.2000E-5
WINDOW 4			
D1	0.2250		0.1750
D2	0.3125	STOP	0.5875
Error	.2000E-5		.2000E-5
	STOP		STOP

TABLE 4.5

A1=1. A2=1.
F1=.25 F2=.375
NRUN=100 No. of samples =12

SNR	$\hat{F}1$	$\|F1 - \hat{F}1\|$	SD1	$\hat{F}2$	$\|F2 - \hat{F}2\|$	SD2
20	.24987	.125E-3	.136E-2	.37497	.314E-4	.148E-2
10	.24945	.547E-3	.424E-2	.37478	.219E-3	.456E-2
5	.24909	.906E-3	.772E-2	.37466	.375E-3	.828E-2
0	.24850	.150E-2	.484E-1	.40651	.315E-1	.128E-0
-5	.32156	.715E-1	.181E-0	.[illegible]6109	.186E-0	.237E-0

SNR signal to noise ratio
SD standard deviation
$\hat{F}i$ mean estimate
$Fi-\hat{F}i$ frequency error

TABLE 4.6

A1= 1. A2=1.
F1=.25 F2=.375
NRUN=100 NO. of samples =12

SNR	$\|F1 - \hat{F}1\|$	SD1	$\|F2 - \hat{F}2\|$	SD2	C-R BOUND
20	.161E-3	.128E-2	.460E-4	.146E-2	.133E-2
10	.250E-3	.573E-2	.230E-3	.550E-2	.420E-2
5	.230E-2	.470E-1	.290E-2	.050E-1	.748E-2
0	.339E-2	.589E-1	.338E-1	.124E 0	.133E-1
-5	.186E-2	.117E 0	.186E 0	.227E 0	.236E-1

TABLE 4.7

```
KN= 12   NRUN= 10
A1=0.1000D 01    A2=0.1000D 01
F1=0.2500D 00    F2=0.4750D 00
SIR = 0.0             SNR =0.1000E 02   NVAR=0.1000E 00
    FREQ1            FREQ-ERROR1          FREQ2           FREQ-ERROR2
0.242187D 00     0.781259D-02    0.478125D 00     0.312483D-02
0.246875D 00     0.312509D-02    0.487500D 00     0.124998D-01
0.259375D 00     0.937491D-02    0.476562D 00     0.156233D-02
0.253125D 00     0.312491D-02    0.478125D 00     0.312483D-02
0.250000D 00     0.894070D-07    0.470312D 00     0.468767D-02
0.248437D 00     0.156259D-02    0.476562D 00     0.156233D-02
0.253125D 00     0.312491D-02    0.478125D 00     0.312483D-02
0.248437D 00     0.156259D-02    0.476562D 00     0.156233D-02
0.251562D 00     0.156241D-02    0.464062D 00     0.109377D-01
0.250000D 00     0.894070D-07    0.492187D 00     0.171873D-01
MEAN1 = 0.250312D 00          MEAN2 = 0.477812D 00
MERROR1=0.312410D-03          MERROR2=0.281233D-02
VARIANCE1=0.184570D-04        VARIANCE2=0.550781D-04
STDIV1  = 0.429616D-02        STDIV2  = 0.742146D-02
```

TABLE 4.8

A1=1.0 A2=1.0
F1=.5 F2=.55
NRUN=100 NO. of samples =12

SNR	$\lvert F1 - \hat{F1} \rvert$	SD1	$\lvert F2 - \hat{F2} \rvert$	SD2	C-R BOUND
20	.187E-3	.263E-2	.171E-3	.262E-2	.272E-2
10	.281E-3	.852E-2	.359E-3	.859E-2	.860E-2
5	.718E-3	.152E-1	.453E-3	.155E-1	.153E-1
0	.193E-1	.857E-1	.594E-1	.118E 0	.272E-1
-5	.420E-1	.167E 0	.321E-1	.179E 0	.484E-1

TABLE 4.9a

A1=1.   A2=10.
F1=.25   F2=.275
NRUN=100   No. of samples =12

SNR	$\widehat{F1}$	$\|F1 - \widehat{F1}\|$	SD1	$\widehat{F2}$	$\|F2 - \widehat{F2}\|$	SD2
20	.25078	.781E-3	.643E-2	.27515	.156E-3	.781E-3
10	.25467	.467E-2	.224E-1	.27402	.984E-3	.264E-2
5	.26035	.103E-1	.428E-1	.27361	.139E-2	.342E-2
0	.27495	.249E-1	.725E-1	.27343	.156E-2	.391E-2
-5	.29096	.409E-1	.887E-1	.27309	.190E-2	.555E-2

TABLE 4.9b

A1=1.   A2=10.
F1=.25   F2=.275
NRUN=100   No. of samples =12

SNR	C-R BD1	C-R BD2
20	.717E-2	.717E-3
10	.226E-1	.226E-2
5	.403E-1	.403E-2
0	.717E-1	.717E-2
-5	.127E+0	.127E-1

CONCIUSION

The complex subset selection algorithm was described in chapter2. This algorithm was used in section 3.2 to design low-pass FIR digital filters with a small number of coefficients using a weighted least squares error criterion. Digital lcw-pass filters with uniformly-spaced time samples were discussed in section 3.3, and it was seen in section 3.4 that, for the same time aperture and number of coefficients, the complex subset selection method results in a) smaller least squares error, b) flatter magnitude response in the passband region, c) sharper roll-off in the transition band, and d) higher peak sidelobe level.

Tone estimation by the complex subset selection method was explained in section 4.2, and was extended to frequency estimation in section 4.3 . The presence of Gaussian noise was discussed in section 4.4 . The numerical results were compared with the C-R bounds in section 4.5, where unbiased estimaticn was assumed, and it was shown that the results were consistant with the C-R bounds. Finally, it should be mentioned that this method can be applied to multi-frequency estimation with a little modification, and with scme more computation.

APPENDIX A

In this appendix, the details of the derivation of digital filter synthesis is presented.

$$D(w) = \begin{cases} 1 & , \quad w < p \\ 0 & , \quad s1 < w < s2 \\ \text{Don't care} & , \quad \text{otherwise} \end{cases} \qquad (A.1)$$

,

$$W(w) = \begin{cases} 1 & , \quad |w| \leqslant p \text{ and } s1 \leqslant |w| \leqslant s2 \\ 0 & , \quad \text{otherwise} \end{cases} \qquad (A.2)$$

$$H(w) = \sum_{n=1}^{M} h(n).EXP(JwT_n) \qquad (A.3)$$

Where T_n's are selected from a set $\{nT\}$ with $n=-q,.....,q$.

$$\underset{\{h(n),T_n\}}{\text{Min. Error}} = \int W(w)\, |D(w) - H(w)|^2\, dw \qquad (A.4)$$

Using orthogonality principle (14), it follows:

$$\int W(w).(D(w) - H(w)).EXP(-JwT_k).dw = 0$$

$$\text{for } k=1,....,M \qquad (A.5)$$

$$\int W(w).H(w).EXP(-JwT_K).dw = \int W(w).D(w).EXP(-JwT_K).dw$$
$$\text{for } k=1,\ldots,M \qquad (A.6)$$

$$A\ H = B \qquad (A.7)$$

with

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & & & \\ a_{M1} & & \cdots & a_{MM} \end{bmatrix} \qquad (A.8)$$

$$H = \Big[h(n), \ldots\ldots\ldots, h(M) \Big] \qquad (A.9)$$

$$B = \Big[b, \ldots\ldots\ldots, b \Big] \qquad (A.10)$$

and

$$\begin{aligned} a_{ij} &= \int W(w).EXP(Jw(T_i - T_j)).dw = \int_{-s2}^{-s1} EXP(Jw(T_i - T_j)).dw \\ &+ \int_{-p}^{p} EXP(Jw(T_i - T_j)).dw + \int_{s1}^{s2} EXP(Jw(T_i - T_j)).dw \\ &= (\ EXP(-Js1(T_i - T_j)) - EXP(-Js2(T_i - T_j))\ )/J(T_i - T_j) \\ &+ (\ EXP(Jp(T_i - T_j)) - EXP(-Jp(T_i - T_j))\ )/J(T_i - T_j) \\ &+ (\ EXP(Js2(T_i - T_j)) - EXP(Js1(T_i - T_j))\ )/J(T_i - T_j) \\ &= 2 \Big[Sin(s2(T_i - T_j)) - Sin(s1(T_i - T_j)) \\ &+ Sin(p(T_i - T_j)) \Big] / (T_i - T_j) \qquad (A.11) \end{aligned}$$

$$b_i = \int W(w).D(w).EXP(-JwT_i).dw = \int_{-p}^{p} EXP(-JwT_i).dw$$

$$= \left[EXP(-JpT_i)) - EXP(JpT_i)) \right] / (-JT_i) = 2Sin(pT_i)/T_i \qquad (A.12)$$

Therefore,

$$H = A\ B \qquad (A.13)$$

$$\underset{h(n),T}{\text{Min. Error}} = \int W(w) \left| D(w) - \sum_{n=1}^{M} h(n).EXP(JwT_n) \right|^{2}.dw$$

$$= \int W(w)\ (D(w) - \sum_{n=1}^{M} h(n).EXP(JwT_n)).D(w).dw$$

$$. \qquad (A.14)$$

$$\underset{\{h(n),T_n\}}{\text{Min. Error}} = \int W(w) \left| D(w) \right|^{2}.dw - \sum_{n=1}^{M} h(n) \int W(w).D\ (w).EXP(JwT_n)).dw$$

$$= \int_{-p}^{p} 1.dw - \sum_{n=1}^{M} h(n) \quad EXP(JwT_n)).dw$$

$$= 2p - \sum_{n=1}^{M} h(n).b_n \qquad (A.15)$$

```
C      APPENDIX B
C
C
C          FOUR DEFFERENT PROGRAMS ARE LISTED BELOW
C      WHICH EACH ONE IS FOR AN ESPECIAL PURPOSE.
C      PROGRAM ONE AND TWO ARE FOR DIGITAL FILTER
C      DESIGN WHILE PROGRAMS 3 AND 4 ARE FOR TONE
C      AND FREQUENCY ESTIMATION.
C
C
C
C      PROGRAM-1
C
C
C      THIS IS THE PROGRAM LISTING FOR DIGITAL FILTER
C      DESIGN BY COMPLEX SUBSET SELECTION.
C
C
C      A(I,J)     A MATRIX
C      COL(I)     B MATRIX
C      SOL(I)     COEFFICIENT MATRIX
C      E(I)       EI ERROR FUNCTIONS
C      AE(I)      RE-ORDER ERRORS
C      TN(I)      BASIS FUNCTION PARAMETERS
C      NQ         Q IN EQUATION (3.2.1)
C      M          NO. OF COEFFICIENTS
C      P          PASSBAND EDGE
C      S1         LEFT STOPBAND EDGE
C      S2         RIGHT STOPBAND EDGE
C      T          GRID POINT SEPARATION
C
C
      DIMENSION Y(101)
      COMPLEX V,ER,H(30),ET,BRSS
      REAL E(30),AE(30),TN(30),CT(10)
      INTEGER IP(30),IND(30)
      INTEGER IP1(30),IP2(30)
      COMPLEX A(30,30),A1(30,30),DCOL(30),COL(30)
      COMPLEX SOL(30),SOL1(30)
      DATA A,COL,A1/900*(0.,0.),30*0.0,900*(0.,0.),/
      DATA IND,DCOL/30*0,30*0.0/
      READ(5,210)NQ,M,NO,P,S1,S2,T
      READ(5,220)MARK,NW,NP
      PI=4.0*ATAN(1.0)
      NQ1=2*NQ+1
      IF(MARK.EQ.1)GO TO 5
      DO 1 1=1,NQ1
      TN(I)=(I-NQ-1)*T
```

```
    1 CONTINUE
      GO TO 8
    5 READ(5,230)(CT(I),I=1,NW)
      TH=T/2
      DO 7 I=1,NP
      DO 7 J=1,NW
      TN(I+(J-1)*NP)=CT(J)+(I-2)*TH
    7 CONTINUE
      NQ1=NP*NW
    8 WRITE(6,150)NQ,M
      WRITE(6,240)P,S1,S2,T
      P=P*2*PI
      S1=S1*2*PI
      S2=S2*2*PI
C
C     EVALUATION OF THE ELEMENTS OF A AND B MATRIX,
C     EQUATIONS (3.2.14) AND (3.2.15)
C
      DO 32 I=1,NQ1
      DO 20 J=I,NQ1
      TIJ=TN(I)-TN(J)
      IF(TIJ.EQ.0.)GO TO 10
      A(I,J)=2*(SIN(TIJ*S2)-SIN(TIJ*S1)+SIN(TIJ*P))/TIJ
      A(J,I)=A(I,J)
      GO TO 20
   10 A(I,J)=2*(S2-S1+P)+.0001
   20 CONTINUE
      IF(TN(I).EQ.0.)GO TO 30
      COL(I)=2*SIN(P*TN(I))/TN(I)
      GO TO 32
   30 COL(I)=2*P
   32 CONTINUE
      CALL AHERIV(A,COL,H,ET1,NQ1)
      ET1=2*P-ET1
      WRITE(6,190)ET1
C
C     EVALUATION AND PLOTTING OF THE REQUENCY RESPONSE
C     FOR THE FILTER USING ALL THE BASIS FUNCTIONS
C
      NO1=NO+1
      L=0
      DO 38 I=1,NO1
      V=CMPLX(0.,0.)
      DO 37 J=1,NQ1
      X=2*PI*L*TN(J)/NO
      V=V+H(J)*CEXP(CMPLX(0.,X))
   37 CONTINUE
      L=L+1
      Y(I)=CABS(V)
      Y(I)=20*ALOG(Y(I)+.0000001)
```

```
  38  CONTINUE
      DX=1./NO
      CALL FPLOTT(Y,NQ1,0.,DX,2.5,-100.0)
C
C     EVALUATION AND RE-ORDERING OF EI ERROR FUNCTIONS,
C     EQUATIONS (2.2.9) AND (2.2.10)
C
      NS=NQ1-1
      DO 40 IE=1,NQ1
      CALL DELT(A,COL,A1,DCOL,NQ1,NS,IE)
      CALL AHERIV(A1,DCOL,SOL,ET1,NS)
      ET1=2*P-ET1
      E(IE)=ET1
  40  CONTINUE
      DO 60 K=1,NQ1
      AE(K)=E(K)
      IP(K)=K
      DO 50 LP=1,NQ1
      IF(AE(K).LE.E(LP))GO TO 50
      AE(K)=E(LP)
      IP(K)=LP
  50  CONTINUE
      E(IP(K))=1.0E 6
      WRITE(6,170)IP(K),AE(K),K
  60  CONTINUE
C
C     BEGINING OF THE SEARCH FOR BEST SUBSET
C
      NR=NQ1-M
      DO 70 I=1,NR
      IND(I)=1
  70  CONTINUE
      DO 80 J=1,M
      IND(J+NR)=0
  80  CONTINUE
      IE=-1
      MP=NR+1
      BR=AE(MP)
      KP=1
      KPT=1
      MI=M+1
      ETAN=1.0E 6
      DO 90 IT=1,MI
      IF(IT.EQ.1)GO TO 82
      KP=(KP*(NR+IT-2))/(IT-1)
      KPT=KPT+KP
  82  DO 86 K=1,KP
      IF(IT.EQ.1)GO TO 83
      CALL COMBO(IND,NQ1,NR,NP)
  83  CALL INTG(A,COL,A1,DCOL,H,IND,IP,IP1,NQ1,TN,NS,IE)
```

```
      CALL AHERIV(A1,DCOL,SOL,ET1,NS)
      ET1=2*P-ET1
      ETA=ET1
      IF(ETA.GT.ETAN)GO TO 86
      ETAN=ETA
      DO 85 IX=1,M
      SOL1(IX)=SOL(IX)
      IP2(IX)=IP1(IX)
   85 CONTINUE
   86 CONTINUE
      IF(ETAN.LE.BR)GO TO 92
      MP=MP+1
      BR=AE(MP)
   90 CONTINUE
   92 DO 93 L=1,M
      WRITE(6,180)L,SOL1(L),TN(IP2(L))
   93 CONTINUE
      WRITE(6,190)ETAN
      WRITE(6,200)KPT
C
C     EVALUATION AND PLOTTING OF THE FREQUENCY RESPONSE
C     FOR A BEST SUBSET SELECTED FILTER
C
      L=0
      DO 94 I=1,NO1
      V=CMPLX(0.,0.)
      DO 95 J=1,M
      X=2*PI*L*TN(IP2(J))/NO
      V=V+SOL1(J)*CEXP(CMPLX(0.,X))
   95 CONTINUE
      L=L+1
      Y(I)=CABS(V)
      Y(I)=20*ALOG(Y(I)+.0000001)
   94 CONTINUE
      CALL FPLOTT(Y,NO1,0.,DX,2.5,-100.0)
  100 FORMAT(2I2,F10.0)
  110 FORMAT(4F10.0)
  120 FORMAT(5X,'D(',I2,')=',E10.2)
  130 FORMAT(/,5X,'C(',I2,')= (',E12.4,'  , ',E12.4,'  )')
  140 FORMAT(//,5X,'RSS FOR FULL BASIS =',E12.4)
  150 FORMAT(//,5X,'N=',I2,'  M=',I2)
  160 FORMAT(/,5X,'E(',I2,')=',E12.4)
  170 FORMAT(/,5X,'E(',I2,')=',E14.7,'  NO. ',I2,
     *' SMALLEST ERROR')
  180 FORMAT(//,'  H(',I2,')=(',E12.4,' ,',E12.4,')',
     *'  T=',E10.4)
  190 FORMAT(//,5X,'ERROR=',E14.7)
  200 FORMAT(//,5X,'NO. OF SUBSETS EXAMINED   ',I6)
  210 FORMAT(3I3,4F5.0)
  220 FORMAT(I1,2I2)
  230 FORMAT(5F5.0)
  240 FORMAT(/,5X,'PBAND =',E10.4,3X,'SBAND1 =',E10.4,3X,
     *'SBANS2 =',E10.4,3X,'T=',E10.4)
      STOP
      END
C
C
C     SUBROUTINE INTG IS FOR SELECTING THE ELEMENTS
C     OF A AND B MATRICES FOR THE INDICATED SUBSET
C
      SUBROUTINE INTG(A,COL,A1,DCOL,H,IND,IP,IP1,NQ1,TN,NS,IE)
      COMPLEX A(30,30),COL(30),A1(30,30),DCOL(30),H(30)
      REAL TN(30)
      INTEGER IND(30),IP(30),IP1(30)
  25  K=0
      K1=1
      DO 26 K2=1,NQ1
      IF(IND(K2).EQ.1)GO TO 26
      IP1(K1)=IP(K2)
      K=K+1
      K1=K1+1
  26  CONTINUE
      NS=K
  30  DO 2 I=1,NS
      DO 2 J=I,NS
      A1(I,J)=A(IP1(I),IP1(J))
      A1(J,I)=A1(I,J)
  2   CONTINUE
      DO 3 K=1,NS
      DCOL(K)=COL(IP1(K))
  3   CONTINUE
      RETURN
      END
C
C
C     SUBROUTINE AHERIV  IS FOR SOLVING THE LINEAR
C     EQUATIONS AH=B, AND EVALUATING THE
C     APPROPRIATE ERROR
C
      SUBROUTINE AHERIV(A,COL,SOL,ET1,K)
      COMPLEX A(30,30),COL(30),RL(30,30),SOL(30),E1,E2,E3,ET
      DO 5 I=1,K
      DO 5 J=1,K
  5   RL(I,J)=CMPLX(0.,0.)
      DO 60 I=1,K
      IP1=I+1
      IM1=I-1
      IF(1.NE.1)GO TO 30
      DO 10 J=1,K
  10  RL(1,J)=A(1,J)
```

```
      DO 20 J=2,K
   20 RL(J,1)=CONJG(RL(1,J))/RL(1,1)
      GO TO 60
   30 DO 40 J=I,K
      RL(I,J)=A(I,J)
      DO 40 M=1,IM1
   40 RL(I,J)=RL(I,J)-RL(I,M)*RL(M,J)
      IF(I.EQ.K)GO TO 60
      DO 45 J=IP1,K
   45 RL(J,I)=CONJG(RL(I,J))/RL(I,I)
   60 CONTINUE
   80 DO 100 I=1,K
      IM1=I-1
      SOL(I)=COL(I)
      IF(I.EQ.1)GO TO 100
      DO 90 L=1,IM1
   90 SOL(I)=SOL(I)-SOL(L)*RL(I,L)
  100 CONTINUE
      DO 140 I=1,K
      J=K-I+1
      IF(J.EQ.K)GO TO 130
      JP1=J+1
      DO 120 L=JP1,K
  120 SOL(J)=SOL(J)-SOL(L)*RL(J,L)
  130 SOL(J)=SOL(J)/RL(J,J)
  140 CONTINUE
C
C     EVALUATION OF ERROR, EQUATION (3.2.17)
C
      ET=0.
      DO 150 I=1,K
      ET=ET+COL(I)*SOL(I)
  150 CONTINUE
      ET1=CABS(ET)
      RETURN
      END
C
C
C     SUBROUTINE COMBO GENERATES STRING OF ONES AND
C     ZEROS FOR DELETION OF BASIS FUNCTIONS
C
      SUBROUTINE COMBO(JN,NQ1,M,NF)
      DIMENSION JN(30)
      ONES=M
      NF=0
    1 NP=1
    2 CONTINUE
      IF(JN(NP).EQ.0)GO TO 10
      IF(NP.EQ.NQ1)GO TO 101
      JN(NP)=0
      NP=NP+1
      ONES=ONES-1
      GO TO 2
   10 JN(NP)=1
      ONES=ONES+1
      IF(ONES.EQ.M)GO TO 100
      GO TO 1
  101 NP=1
  100 RETURN
      END
C
C
C     SUBROUTINE DELT IS FOR DELETING ROW AND
C     COLUMN OF A AND B MATRICES FOR EVALUATION
C     OF EI ERROR FUNCTIONS
C
      SUBROUTINE DELT(A,COL,A1,DCOL,NQ1,NS,IE)
      COMPLEX A(30,30),COL(30),A1(30,30),DCOL(30)
      IEM=IE-1
      IF(IE.EQ.1)GO TO 30
      DO 10 I=1,IEM
      DCOL(I)=COL(I)
      DO 10 J=I,IEM
      A1(I,J)=A(I,J)
      A1(J,I)=A(J,I)
   10 CONTINUE
      DO 20 I=1,IEM
      DO 20 J=IE,NS
      A1(I,J)=A(I,J+1)
      A1(J,I)=A(J+1,I)
   20 CONTINUE
      IF(IE.EQ.NQ1)GO TO 50
   30 DO 40 I=IE,NS
      DCOL(I)=COL(I+1)
      DO 40 J=I,NS
      A1(I,J)=A(I+1,J+1)
      A1(J,I)=A(J+1,I+1)
   40 CONTINUE
   50 RETURN
      END
C
C
C     SUBROUTINE FPLOTT IS FOR PLOTTING THE
C     FREQUENCY RESPONSE OF THE FILTER
C
      SUBROUTINE FPLOTT(Y,NDIM,XI,DX,YMAX,YMIN)
      DIMENSION Y(NDIM)
      DATA IBLANK,IPOINT,IAST,IMINUS/' ','.','*','-'/,IPRINT/6/
      INTEGER PLOT(78)
      IF(YMAX-YMIN)250,5,45
```

```
 5     YMAX=Y(1)
       YMIN=Y(1)
       DO 40 K=1,NDIM
       IF(Y(K)-YMAX)20,20,10
10     YMAX=Y(K)
20     IF(Y(K)-YMIN)30,40,40
30     YMIN=Y(K)
40     CONTINUE
       IF(YMAX-YMIN)250,250,45
45     X=XI-DX
       IF(YMIN)60,60,50
50     IQ=0
       GO TO 70
60     IF(YMAX)62,68,68
62     IQ=76
       GO TO 70
68     IQ=IFIX(76.*(-YMIN/(YMAX-YMIN))+.5)
70     KI=IQ+1
       WRITE(IPRINT,1000)YMIN
1000   FORMAT('  YMIN= ',1PE10.3)
       WRITE(IPRINT,2000)YMAX
2000   FORMAT('  YMAX= ',1PE10.3/)
       DO 200 K=1,NDIM
       YK=Y(K)
       IF(YK-YMAX)81,84,80
80     YK=YMAX
       GO TO 83
81     IF(YK-YMIN)82,84,84
82     YK=YMIN
83     IQUE=IAST
       GO TO 85
84     IQUE=IBLANK
85     CONTINUE
       IP=IFIX(76.*((YK-YMIN)/(YMAX-YMIN))+.5)
       IB=IP+1
       X=X+DX
       IF(YK)90,150,150
90     DO 100 I=1,IP
100    PLOT(I)=IBLANK
       PLOT(IB)=IAST
       J=IB+1
       DO 110 L=J,IQ
110    PLOT(L)=IMINUS
       IF(IP-IQ)130,120,130
120    PLOT(KI)=IAST
       GO TO 140
130    PLOT(KI)=IPOINT
140    WRITE(IPRINT,3000)X,Y(K),IQUE,(PLOT(M),M=1,KI)
       GO TO 200
150    DO 160 I=1,KI
```

```
160    PLOT(I)=IBLANK
       PLOT(KI)=IPOINT
       J=KI+1
       DO 170 I=J,IP
170    PLOT(I)=IMINUS
       PLOT(IB)=IAST
       WRITE(IPRINT,3000)X,Y(K),IQUE,(PLOT(M),M=1,IB)
3000   FORMAT(' X=',1PE10.3,3X,' Y=',1PE10.3,3X,A1,' \',77A1)
200    CONTINUE
       RETURN
250    WRITE(IPRINT,5000)
5000   FORMAT(' ERROR--YMAX=YMIN')
       RETURN
       END
```

```
C     PROGRAM-2
C
C
C
C     THIS IS THE PROGRAM LISTING FOR UNIFORMLY-
C     SPACE PARAMETER FILTER
C
C
C     EVALUATION OF THE ELEMENTS OF A' AND B' MATRICES
C     OF EQUATION (3.3.3)
C
C     A(I,J)     A' MATRIX
C     B(I)       B' MATRIX
C     H(I)       COEFFICIENT MATRIX
C     M          NO. OF COEFFICIENTS
C     P          PASSBAND EDGE
C     S1         LEFT STOPBAND EDGE
C     S2         RIGHT STOPBAND EDGE
C     ND         NO. OF DELTAS FOR SEARCH
C     NC         NO. OF C'S FOR SEARCH
C     DEL        INITIAL DELTA
C     DELD       DELTA INCREMENT
C     CI         INITIAL C
C     DELC       C INCREMENT
C
      DIMENSION DN(10),Y(101),E(100)
      COMPLEX H(10),A(10,10),B(10),HB(10),ET,V
      READ(5,100)NO,M,ND,NC
      READ(5,110)P,S1,S2,DEL,DELD
      READ(5,110)CI,DELC
      PI=4*ATAN(1.)
      D1=DEL
      P=2*PI*P
      S1=2*PI*S1
      S2=2*PI*S2
      SER=1.0E 6
      K1=4
      CO=CI
      DO 50 IK=1,NC
      DO 40 IL=1,ND
      DO 25 I=1,M
      DO 15 J=I,M
      IF(I.EQ.J)GO TO 10
      X=D1*(I-J)
      A(I,J)=2*(SIN(S2*X)-SIN(S1*X)+SIN(P*X))/X
      A(J,I)=A(I,J)
      GO TO 15
   10 A(I,J)=2*(S2-S1+P)+.0001
   15 CONTINUE
      X=(I-3)*D1+CO
      IF(X.EQ.0.)GO TO 20
      B(I)=2*SIN(P*X)/X
      GO TO 25
   20 B(I)=2*P
   25 CONTINUE
      CALL AHERIV(A,B,H,E1,ET,M)
C
C     SELECTION OF THE SMALLEST ERROR AND
C     CORRESPONDING COEFFICIENTS
C
      E(IL)=E1
      ET1=2*P-E1
      WRITE(6,125)CO,D1,ET1
      IF(SER.LT.ET1)GO TO 35
      SER=ET1
      CB=CO
      BD=D1
      DO 30 L=1,M
   30 HB(L)=H(L)
   35 D1=D1+DELD
   40 CONTINUE
      CO=CO+DELC
      D1=DEL
   50 CONTINUE
      WRITE(6,130)CB,BD,SER
      DO 60 I=1,M
   60 WRITE(6,140)HB(I)
      DO 62 I=1,M
   62 DN(I)=(I-3)*BD+CB
      WRITE(6,150)(DN(I),I=1,M)
C
C     EVALUATION AND PLOTTING OF FREQUENCY RESPONSE
C
      NO1=NO+1
      L=0
      DO 70 I=1,NO1
      V=CMPLX(0.,0.)
      DO 65 J=1,M
      X=2*PI*L*DN(J)/NO
      V=V+HB(J)*CEXP(CMPLX(0.,X))
   65 CONTINUE
      L=L+1
      Y(I)=CABS(V)
      Y(I)=20*ALOG(Y(I)+.0000001)
   70 CONTINUE
      DX=1./NO
      CALL FPLOTT(Y,N01,0.,DX,4.,-100.0)
      CALL FPLOTT(E,ND,DEL,DELC,1.8,0.0)
 100  FORMAT(I3,3I2)
 110  FORMAT(5F5.0)
 120  FORMAT(/,5X,'  DELTA=',E10.4,'  ERROR=',3E12.5)
 125  FORMAT(3X,'C=',E9.3,3X,'DELTA=',E8.2,3X,'ERROR=',E12.6)
 130  FORMAT(/,5X,'BEST C=',E10.4,'  BEST DELTA=',E10.4,
     *'  ERROR=',2E12.5)
 140  FORMAT(/,5X,'HB =',2E12.5)
 150  FORMAT(/,5X,5(E10.4,2X))
      STOP
      END
C
C
C     SUBROUTINE AHERIV IS FOR SOLVING THE LINEAR
C     EQUATIONS A'H=B', AND EVALUATING THE
C     CORRESPONDING ERROR FUNCTION
C
      SUBROUTINE AHERIV(A,COL,SOL,E1,ET,K)
      COMPLEX A(10,10),COL(10),RL(10,10),SOL(10),E1,E2,E3,ET
      DO 5 I=1,K
      DO 5 J=1,K
  5   RL(I,J)=CMPLX(0.,0.)
      DO 60 I=1,K
      IP1=I+1
      IM1=I-1
      IF(I.NE.1)GO TO 30
      DO 10 J=1,K
  10  RL(1,J)=A(1,J)
      DO 20 J=2,K
  20  RL(J,1)=CONJG(RL(1,J))/RL(1,1)
      GO TO 60
  30  DO 40 J=I,K
      RL(I,J)=A(I,J)
      DO 40 M=1,IM1
  40  RL(I,J)=RL(I,J)-RL(I,M)*RL(M,J)
      IF(I.EQ.K)GO TO 60
      DO 45 J=IP1,K
  45  RL(J,I)=CONJG(RL(I,J))/RL(I,I)
  60  CONTINUE
  80  DO 100 I=1,K
      IM1=I-1
      SOL(I)=COL(I)
      IF(I.EQ.1)GO TO 100
      DO 90 L=1,IM1
  90  SOL(I)=SOL(I)-SOL(L)*RL(I,L)
  100 CONTINUE
      DO 140 I=1,K
      J=K-I+1
      IF(J.EQ.K)GO TO 130
      IF(J.EQ.K)GO TO 130
      JP1=J+1
      DO 120 L=JP1,K
  120 SOL(J)=SOL(J)-SOL(L)*RL(J,L)
  130 SOL(J)=SOL(J)/RL(J,J)
  140 CONTINUE
C
C     EVALUATION OF ERROR, EQUATION (3.3.4)
C
      ET=0.
      DO 150 I=1,K
      ET=ET+COL(I)*SOL(I)
  150 CONTINUE
      E1=CABS(ET)
      RETURN
      END
```

```
C       PROGRAM-3
C
C
C
C       THIS IS THE PROGRAM LISTING FOR
C       TONE ESTIMATION.
C
C
C       A1,A2        TONES AMPLITUDES
C       B1,B2        TONES FREQUENCIES
C       KN           NO. OF SAMPLES
C       A(I,J)       A MATRIX
C       COL(I)       B MATRIX
C       SOL(I)       COEFFICIENT MATRIX
C       E(I)         EI'S ERROR FUNCTIONS
C       SIR          SIGNAL TO INTERFERENCE RATIO
C       SNR          SIGNAL TO NOISE RATIO
C
        COMMON PI,A1,A2,B1,B2,D1,D2,D3,D4,N1,N2,KN
        REAL*8 D(50),E(50),AE(50),A1,A2,B1,B2,PI,D1,D2
        REAL*8 D3,D4,P1,P2,EC3,P3,BR,ETA,ETAN
        INTEGER IP(50),IND(50),INDR(50),INP(50),JP(50)
        INTEGER JJP(50),LJP(50),R
        COMPLEX*16 A(50,50),COL(50),SOL(50),SOL1(50),ET
        COMPLEX*16 S(50,50),DCOL(50),BRSS,Y(100),VN,E3
        DATA A,COL,IND/2500*(0.0D0,0.0D0),50*(0.D0,0.D0),50*0/
        PI=4.0D0*DATAN(1.0D0)
        READ(5,110)SNR,SIR,B1,B2
        IND1=0
        IND3=0
        A1=1.D0
        A2=A1*(10**(-SIR/20))
        READ(5,105)KN,AK
        IX=3657137
        SD=A1*(10**(-SNR/20))
        VAR=SD**2
        WRITE(6,113)KN
        WRITE(6,115)A1,A2
        WRITE(6,117)B1,B2
   10   READ(5,100)LFI,N,MS,ML,D1,D2
        IF(LFI.EQ.1)GO TO 2
C
C       GENERATION OF BASIS FUNCTION PARAMETERS
C
        D(1)=D1
        DO 1 I=2,N
    1   D(I)=D(I-1)+D2
        GO TO 5
    2   READ(5,220)N1,N2,D3,D4
```

```
      D(1)=D1
      DO 3 I=2,N1
    3 D(I)=D(I-1)+D2
      D(N1+1)=D3
      DO 4 J=2,N2
    4 D(N1+J)=D(N1+J-1)+D4
      N=N1+N2
    5 WRITE(6,210)N
      IF(IND1.EQ.1)GO TO 11
C
C     GENARATION OF GAUSSIAN NOISE IN DATA,
C     EQUATIONS (4.2.2) AND (4.4.1)
C
      DO 7 I=1,KN
      CALL GAUSS(IX,SD,0.,V)
      CALL GAUSS(IX,SD,0.,V1)
      VN=CMPLX(V,V1)
      P1=2.D0*PI*B1
      P2=2.D0*PI*B2
      Y(I)=A1*CDEXP(DCMPLX(0.D0,P1*(I-1)))
      Y(I)=Y(I)+A2*CDEXP(DCMPLX(0.D0,P2*(I-1)))
    7 CONTINUE
   11 IF(IND3.EQ.0)GO TO 8
      P1=2.*PI*D(LJP(1))
      P2=2.*PI*D(LJP(2))
      DO 9 I=1,KN
      Y(I)=Y(I)-SOL1(1)*CDEXP(DCMPLX(0.D0,P1*(I-1)))
      Y(I)=Y(I)-SOL1(2)*CDEXP(DCMPLX(0.D0,P2*(I-1)))
    9 CONTINUE
    8 NS=N
C
C     EVALUATION AND RE-ORDERING OF EI ERROR
C     FUNCTIONS, EQUATIONS (2.2.9) AND (2.2.10)
C
      IE=0
      CALL INTG(A,COL,Y,IND,IP,N,NS,LPI,IE)
      CALL AHERIV(A,COL,SOL,Y,ET,NS,KN)
      ETA=CDABS(ET)
      NS=N-1
      DO 40 IE=1,N
      CALL DELT(A,COL,S,DCOL,N,NS,IE)
      CALL AHERIV(S,DCOL,SOL,Y,ET,NS,KN)
      E(IE)=CDABS(ET)
   40 CONTINUE
      DO 60 K=1,N
      AE(K)=E(K)
      IP(K)=K
      DO 50 LP=1,N
      IF(AE(K).LE.E(LP))GO TO 50
      AE(K)=E(LP)
      IP(K)=LP
   50 CONTINUE
      E(IP(K))=1.0E 6
   60 CONTINUE
      DO 2000 IXX=1,N
      WRITE(6,170)IP(IXX),AE(IXX),IXX
 2000 CONTINUE
C
C     BEGINING OF SEARCH FOR THE BEST SUBSET
C
      DO 99 M=MS,ML
      WRITE(6,150)N,M
      NR=N-M
      DO 70 I=1,NR
      IND(I)=1
      INDB(I)=1
   70 CONTINUE
      DO 80 J=1,M
      IND(J+NR)=0
      INDB(J+NR)=0
   80 CONTINUE
      IE=-1
      MP=NR+1
      BR=AE(MP)
      KP=1
      KPT=1
      MI=M+1
      ETAN=1.0E 6
      DO 90 IT=1,MI
      IF(IT.EQ.1)GO TO 82
      KP=(KP*(NR+IT-2))/(IT-1)
      KPT=KPT+KP
   82 DO 86 K=1,KP
      IF(IT.EQ.1)GO TO 83
      CALL COMBO(IND,N,NR,NF)
   83 CALL INTG(A,COL,Y,IND,IP,N,NS,LFI,IE)
      CALL AHERIV(A,COL,SOL,Y,ET,NS,KN)
      ETA=CDABS(ET)
      IF(ETA.GT.ETAN)GO TO 86
      ETAN=ETA
      ETAN=ETA
      DO 84 I=1,N
      INDB(I)=IND(I)
   84 CONTINUE
      DO 85 J=1,M
      SOL1(J)=SOL(J)
   85 CONTINUE
   86 CONTINUE
   87 IF(ETAN.LE.BR)GO TO 92
   88 MP=MP+1
```

```
      BR=AE(MP)
 90   CONTINUE
 92   J=0
      DO 93 L=1,N
      IF(INDB(L).EQ.1)GO TO 93
      J=J+1
      JP(J)=IP(L)
 93   CONTINUE
      DO 95 LP=1,J
      LJP(LP)=JP(LP)
      JJP(LP)=LP
      DO 94 KP=1,J
      IF(LJP(LP).LE.JP(KP))GO TO 94
      LJP(LP)=JP(KP)
      JJP(LP)=KP
 94   CONTINUE
      JP(JJP(LP))=1E 6
 95   CONTINUE
      WRITE(6,200)KPT
      DO 96 L=1,J
      WRITE(6,180)L,D(LJP(L)),L,SOL1(L)
 96   CONTINUE
      WRITE(6,190)ETAN
 99   CONTINUE
      READ(5,240)IND2,IND1,IND3
      IF(IND3.EQ.1)GO TO 10
      IF(IND2.EQ.1)GO TO 101
      GO TO 10
 101  CONTINUE
 100  FORMAT(I1,3I2,2F10.0)
 105  FORMAT(I3,F10.0)
 110  FORMAT(4F10.0)
 113  FORMAT(/,5X,'NO. OF SAMMLES =',I4)
 115  FORMAT(/,5X,'A1=',E10.4,5X,'A2=',E10.4)
 117  FORMAT(/,5X,'F1=',E10.4,5X,'F2=',E10.4)
 120  FORMAT(5X,'D(',I2,')=',E12.4)
 125  FORMAT(/,5X,'SIR=',E10.4,5X,'SNR=',E10.4,5X,
     *'VAR=',E10.4)
 130  FORMAT(/,5X,'C(',I2,')= (',E12.4.'  , ',E12.4,'   )')
 140  FORMAT(//,5X,'RSS FOR FULL BASIS =',E12.4)
 150  FORMAT(//,5X,'N=',I2,'  M=',I2)
 160  FORMAT(/,5X,'E(',I2,')=',F13.7)
 170  FORMAT(/,5X,'E('.I2,')=',E13.7,'    NO. ',I2,
     *' SMALLEST ERROR')
 180  FORMAT(/,5X,'D(',I2,')=',E10.4,5X,'C('.I2,')=(',
     *E.10.4,' , ',E10.4,' )')
 190  FORMAT(//,5X,'ERROR='.E12.4)
 200  FORMAT(//,5X,'NO. OF SUBSETS EXAMINED = ',I4)
 210  FORMAT(//,5X,'N=',I2)
 220  FORMAT(2I2,2F10.0)
```

```
  230 FORMAT(//,5X,'SUM OF COEFFICIENTS SQUARED =',E12.4)
  240 FORMAT(3I1)
      STOP
      END
C
C
C     SUBROUTINE INTG IS FOR EVALUATION OF ELEMENTS OF
C     A AND B MATRICES FOR EACH SUBSET TO BE EXAMINED,
C     EQUATIONS (4.2.14) THRU (4.2.16)
C
      SUBROUTINE INTG(A,COL,Y,IND,IP,N,NS,LFI,IE)
      COMMON PI,A1,A2,B1,B2,D1,D2,D3,D4,N1,N2,KN
      COMPLEX*16 A(50,50),COL(50),Y(50)
      REAL*8 D(50),AD(50),X,Y1,Y2,Y3,A1,A2,B1,B2,PI
      REAL*8 D1,D2,D3,D4,AK
      INTEGER IND(50),IP(50),JP(50)
      IF(LFI.EQ.1)GO TO 2
      D(1)=D1
      DO 1 L=2,N
  1   D(L)=D(L-1)+D2
      GO TO 5
  2   D(1)=D1
      DO 3 L=2,N1
  3   D(L)=D(L-1)+D2
      D(N1+1)=D3
      DO 4 LL=2,N2
  4   D(N1+LL)=D(N1+LL-1)+D4
  5   IF(IE.EQ.0)GO TO 30
      IF(IE.EQ.-1)GO TO 25
      NS=N-1
      DO 20 L=1,N
      IF(L.NE.IE)GO TO 20
      DO 15 M=L,NS
  15  D(M)=D(M+1)
  20  CONTINUE
      GO TO 30
  25  K=0
      K1=1
      DO 26 K2=1,N
      IF(IND(K2).EQ.1)GO TO 26
      AD(K1)=D(IP(K2))
      K=K+1
      K1=K1+1
  26  CONTINUE
      DO 28 K3=1,K
      D(K3)=AD(K3)
      JP(K3)=K3
      DO 27 K4=1,K
      IF(D(K3).LE.AD(K4))GO TO 27
      D(K3)=AD(K4)
      JP(K3)=K4
   27 CONTINUE
      AD(JP(K3))=1.0E 6
   28 CONTINUE
      NS=K
   30 DO 60 I=1,NS
      DO 60 J=I,NS
      X=(D(J)-D(I))*2.0D0*PI
      IF(I-J)40,50,40
   40 CONTINUE
      A(I,J)=(1.0D0-CDEXP(DCMPLX(0.D0,X*KN)))
      A(I,J)=A(I,J)/(1.D0-CDEXP(DCMPLX(0.D0,X)))
      A(J,I)=DCONJG(A(I,J))
      GO TO 60
   50 A(I,J)=1.0D0*KN
   60 CONTINUE
      DO 80 K=1,NS
      COL(K)=(0.D0,0.D0)
      Y3=-2.D0*PI*D(K)
      DO 80 J=1,KN
      COL(K)=COL(K)+Y(J)*CDEXP(DCMPLX(0.D0,Y3*(J-1)))
   80 CONTINUE
      RETURN
      END
C
C
C     SUBROUTINE AHERIV IS FOR SOLVING THE LINEAR
C     EQUATIONS AC=B AND CORRESPONDING ERROR FUNCTION
C
      SUBROUTINE AHERIV(A,COL,SOL,Y,ET,K,KN)
      COMMON PI,A1,A2,B1,B2,D1,D2,D3,D4,N1,N2
      REAL*8 A1,A2,B1,B2,PI,C,VN(50),C1,C2
      COMPLEX*16 A(50,50),COL(50),RL(50,50),SOL(50)
      COMPLEX*16 XX1,Y(100),E1,E2,E3,ET,XX
      DO 5 I=1,K
      DO 5 J=1.K
    5 RL(I,J)=DCMPLX(0.0D0,0.0D0)
      DO 60 I=1,K
      IP1=I+1
      IM1=I-1
      IF(I.NE.1)GO TO 30
      DO 10 J=1,K
   10 RL(1,J)=A(1,J)
      DO 20 J=2,K
   20 RL(J,1)=DCONJG(RL(1,J))/RL(1,1)
      GO TO 60
   30 DO 40 J=I,K
      RL(I,J)=A(I,J)
      DO 40 M=1,IM1
   40 RL(I,J)=RL(I,J)-RL(I,M)*RL(M,J)
```

```
      IF(I.EQ.K)GO TO 60
      DO 45 J=IP1,K
   45 RL(J,I)=DCONJG(RL(I,J))/RL(I,I)
   60 CONTINUE
   80 DO 100 I=1,K
      IM1=I-1
      SOL(I)=COL(I)
      IF(I.EQ.1)GO TO 100
      DO 90 L=1,IM1
   90 SOL(I)=SOL(I)-SOL(L)*RL(I,L)
  100 CONTINUE
      DO 140 I=1,K
      J=K-I+1
      IF(J.EQ.K)GO TO 130
      JP1=J+1
      DO 120 L=JP1,K
  120 SOL(J)=SOL(J)-SOL(L)*RL(J,L)
  130 SOL(J)=SOL(J)/RL(J,J)
  140 CONTINUE
C
C     EVALUATION OF ERROR, EQUATION (4.2.19)
C
      E2=0.0D0
      E1=0.0D0
      DO 15 I1=1,K
      E2=E2+SOL(I1)*DCONJG(COL(I1))
   15 CONTINUE
      DO 16 I3=1,KN
      E1=E1+CDABS(Y(I3))**2
   16 CONTINUE
      ET=(E1-E2)/KN
      RETURN
      END
C
C
C     SUBROUTINE COMBO IS FOR GENERATING STRING
C     OF ONES AND ZEROS FOR DELETION OF PROPER
C     BASIS FUNCTIONS
C
      SUBROUTINE COMBO(JN,N,M,NF)
      DIMENSION JN(40)
      IONES=M
      NF=0
    1 NF=1
    2 CONTINUE
      IF(JN(NF).EQ.0)GO TO 10
      IF(NF.EQ.N)GO TO 101
      JN(NF)=0
      NF=NF+1
      IONES=IONES-1
      GO TO 2
   10 JN(NP)=1
      IONES=IONES+1
      IF(IONES.EQ.M)GO TO 100
      GO TO 1
  101 NF=1
  100 RETURN
      END
C
C
C     SUBROUTINE GAUSS AND RANDU ARE FOR
C     GENERATING GAUSSIAN NOISE
C
      SUBROUTINE GAUSS(IX,S,AM,V)
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C
C
C     SUBROUTINE DELT IS FOR DELETING PROPER ROW
C     AND COLUMN OF A AND B MATRICES FOR
C     EVALUATION OF EI ERROR FUNCTIONS
C
      SUBROUTINE DELT(A,COL,S,DCOL,N,NS,IE)
      COMPLEX*16 A(50,50),COL(50),S(50,50),DCOL(50)
      IEM=IE-1
      IF(IE.EQ.1)GO TO 30
      DO 10 I=1,IEM
      DCOL(I)=COL(I)
      DO 10 J=I,IEM
      S(I,J)=A(I,J)
      S(J,I)=A(J,I)
   10 CONTINUE
      DO 20 I=1,IEM
      DO 20 J=IE,NS
      S(I,J)=A(I,J+1)
      S(J,I)=A(J+1,I)
```

```
 20   CONTINUE
      IF(IE.EQ.N)GO TO 50
 30   DO 40 I=IE,NS
      DCOL(I)=COL(I+1)
      DO 40 J=I,NS
      S(I,J)=A(I+1,J+1)
      S(J,I)=A(J+1,I+1)
 40   CONTINUE
 50   RETURN
      END
```

```
C     PROGRAM-4
C
C
C
C     THIS IS THE PROGRAM LISTING FOR TWO
C     FREQUENCY ESTIMATION BY COMPLEX SUBSET
C     SELECTION AND WINDOWING TECHNIQUE.
C
C
C     A1,A2       TONES AMPLITUDES
C     B1,B2       TONES FREQUENCIES
C     KN          NO. OF SAMPLES
C     A(I,J)      A MATRIX
C     COL(I)      B MATRIX
C     SOL(I)      COEFFICIENT MATRIX
C     E(I)        EI ERROR FUNCTIONS
C     SIR         SIGNAL TO INTERFERENCE RATIO
C     SNR         SIGNAL TO NOISE RATIO
C     NRUN        NO. OF RUNS
C
      COMMON PI,A1,A2,B1,B2,D1,D2,D3,D4,N1,N2,KN
      REAL*8 D(40),E(40),AE(40),A1,A2,B1,B2,PI
      REAL*8 D1,D2,D3,D4,BR,ETA,ETAN,P1,P2
      REAL*8 ED11(100),ED22(100),DMER1,DMER2,STD1,STD2
      REAL*8 ED1(100),ED2(100),DMEAN1,DMEAN2,VAR1,VAR2
      INTEGER IP(40),IND(40),INDR(40),INP(40),JP(40)
      INTEGER LJP(40),R,JJP(40)
      COMPLEX*16 A(40,40),COL(40),SOL(40),SOL1(40),ET
      COMPLEX*16 VN,S(40,40),DCOL(40),BRSS,Y(100)
      DATA A,COL/1600*(0.0D0,0.0D0),40*(0.D0,0.D0)/
      DATA IND/40*0/
      PI=4.0D0*DATAN(1.0D0)
      READ(5,110)NRUN,SIR,SNR,B1,B2
      READ(5,105)KN,N1,MS,ML
      A1=1.D0
      A2=A1*(10**(-SIR/20))
      WRITE(6,230)KN,NRUN
      WRITE(6,350)A1,A2
      WRITE(6,360)B1,B2
      KPT=0
      IX=3657137
      SD=A1*(10**(-SNR/20))
      VAR=SD**2
      WRITE(6,280)SIR,SNR,VAR
      P1=2.D0*PI*B1
      P2=2.D0*PI*B2
C
C     BEGINING OF EACH RUN
C
      DO 500 IJ=1,NRUN
C
```

```
C     GENERATION OF GAUSSIAN NOISE IN DATA,
C     EQUATIONS (4.2.2) AND (4.4.1)
C
      DO 1 I=1,KN
      CALL GAUSS(IX,SD,0.,V)
      CALL GAUSS(IX,SD,0.,V1)
      VN=CMPLX(V,V1)
      Y(I)=A1*CDEXP(DCMPLX(0.D0,P1*(I-1)))
      Y(I)=Y(I)+A2*CDEXP(DCMPLX(0.D0,P2*(I-1)))+VN
    1 CONTINUE
C
C     GENERATION OF BASIS FUNCTION PARAMETERS
C
      N=10
      NO=0
      LFI=0
      D1=.1
      D2=D1
      D(1)=0.D0
      DO 2 I=2,10
    2 D(I)=D(I-1)+D1
      NS=N
      GO TO 7
    3 LFI=1
      D(1)=D3
      D(6)=D4
      DO 4 J=2,5
      D(J)=D(J-1)+D1
    4 D(J+5)=D(J+4)+D1
      N=10
      NS=N
    7 IE=0
      CALL INTG(A,COL,Y,IND,IP,N,NS,LFI,IE)
      CALL AHERIV(A,COL,SOL,Y,ET,NS,KN)
      ETA=CDABS(ET)
C
C     EVALUATION AND RE-ORDERING OF EI ERROR
C     FUNCTIONS, EQUATIONS (2.2.9) AND (2.2.10)
C
      NS=N-1
      DO 40 IE=1,N
      CALL DELT(A,COL,S,DCOL,N,NS,IE)
      CALL AHERIV(S,DCOL,SOL,Y,ET,NS,KN)
      E(IE)=CDABS(ET)
   40 CONTINUE
      DO 60 K=1,N
      AE(K)=E(K)
      IP(K)=K
      DO 50 LP=1,N
      IF(AE(K).LE.E(LP))GO TO 50
      AE(K)=E(LP)
      IP(K)=LP
  50  CONTINUE
      E(IP(K))=1.0E 6
  60  CONTINUE
C
C     BEGINING OF THE SEARCH FOR BEST SUBSET
C
      DO 99 M=MS,ML
      NR=N-M
      DO 70 I=1,NR
      IND(I)=1
      INDB(I)=1
  70  CONTINUE
      DO 80 J=1,M
      IND(J+NR)=0
      INDB(J+NR)=0
  80  CONTINUE
      IE=-1
      MP=NR+1
      BR=AE(MP)
      KP=1
      KPT=KPT+1
      MI=M+1
      ETAN=1.0E 6
      DO 90 IT=1,MI
      IF(IT.EQ.1)GO TO 82
      KP=(KP*(NR+IT-2))/(IT-1)
      KPT=KPT+KP
  82  DO 86 K=1,KP
      IF(IT.EQ.1)GO TO 83
      CALL COMBO(IND,N,NR,NF)
  83  CALL INTG(A,COL,Y,IND,IP,N,NS,LFI,IE)
      CALL AHERIV(A,COL,SOL,Y,ET,NS,KN)
      ETA=CDABS(ET)
      IF(ETA.GT.ETAN)GO TO 86
      ETAN=ETA
      DO 84 I=1,N
      INDB(I)=IND(I)
  84  CONTINUE
      DO 85 J=1,M
      SOL1(J)=SOL(J)
  85  CONTINUE
  86  CONTINUE
  87  IF(ETAN.LE.BR)GO TO 92
  88  MP=MP+1
      BR=AE(MP)
  90  CONTINUE
  92  J=0
      DO 93 L=1,N
      IF(INDB(L).EQ.1)GO TO 93
      J=J+1
      JP(J)=IP(L)
  93  CONTINUE
      DO 95 LP=1,J
      LJP(LP)=JP(LP)
      JJP(LP)=LP
      DO 94 KP=1,J
      IF(LJP(LP).LE.JP(KP))GO TO 94
      LJP(LP)=JP(KP)
      JJP(LP)=KP
  94  CONTINUE
      JP(JJP(LP))=1E 6
  95  CONTINUE
C
C     APPLYING WINDOWS AROUND SELECTED
C     BASIS FUNCTIONS
C
      NO=NO+1
      D1=D1/2
      D2=D1
      IF(NO.GE.2)GO TO 505
      D3=D(LJP(1))-D1
      D4=D(LJP(2))-D1
      GO TO 506
  505 MS=2
      ML=2
      IF(DABS(D(LJP(1))-D(LJP(2))).LE..002D0)GO TO 550
      D3=D(LJP(1))-2*D1
      D4=D(LJP(2))-2*D1
  506 IF(D3.GE.0.D0)GO TO 520
      D3=0.D0
      IF(D4-D3)525,525,520
  525 D4=D4+2*D1
  520 IF(D4-D3-4*D1)501,502,503
  501 D4=D4+3*D1
      GO TO 503
  502 D4=D4+D1
  503 IF(NO.GE.7)GO TO 98
      IF(ETAN.GT.1.0E-10)GO TO 3
  98  CONTINUE
  99  CONTINUE
  550 ED1(IJ)=D(LJP(1))
      ED2(IJ)=D(LJP(2))
      WRITE(6,300)ED1(IJ),ED2(IJ)
  500 CONTINUE
C
C     EVALUATION OF MEAN AND VARIANCE OF
C     THE ESTIMATES
C
```

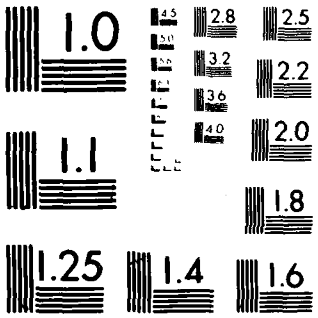
AD-A090 128 RHODE ISLAND UNIV KINGSTON DEPT OF ELECTRICAL ENGIN--ETC F/G 9/2
APPLICATIONS OF EFFICIENT SUBSET SELECTION TO DIGITAL FILTERING--ETC(U)
AUG 80 J KHORAMMI, D TUFTS AFOSR-77-3174

UNCLASSIFIED AFOSR-TR-80-0954 NL

2 OF 2

AD A
090128

END
DATE
FILMED
11-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

```
      CALL VARIAN(ED1,ED2,DMEAN1,DMEAN2,VAR1,VAR2,NRUN)
      WRITE(6,290)
      DO 510 II1=1,NRUN
      ED11(II1)=DABS(ED1(II1)-B1)
      ED22(II1)=DABS(ED2(II1)-B2)
      WRITE(6,300)ED1(II1),ED11(II1),ED2(II1),ED22(II1)
  510 CONTINUE
      DMER1=DABS(B1-DMEAN1)
      DMER2=DABS(B2-DMEAN2)
      STD1=DSQRT(VAR1)
      STD2=DSQRT(VAR2)
      WRITE(6,310)DMEAN1,DMEAN2
      WRITE(6,330)DMER1,DMER2
      WRITE(6,320)VAR1,VAR2
      WRITE(6,340)STD1,STD2
  100 FORMAT(I1,3I2,2F10.0)
  105 FORMAT(2I3,2I2)
  110 FORMAT(I3,4F10.0)
  120 FORMAT(5X,'D(',I2,')=',E14.6)
  130 FORMAT(/,5X,'C(',I2,')= (',E12.4,'  ,  ',
     *E12,4,'  )')
  140 FORMAT(//,5X,'RSS FOR FULL BASIS =',E12.4)
  150 FORMAT(//,5X,'N=',I2,'  M=',I2)
  160 FORMAT(/,5X,'E(',I2,')=',E12.4)
  170 FORMAT(/,5X,'E(',I2,')=',E12.4,'  NO. ',I2
     *,' SMALLEST ERROR')
  180 FORMAT(/,5X,'C(',I2,')=(',E12.4,'  , ',E12.4,' )')
  190 FORMAT(//,5X,'ERROR=',E12.4)
  200 FORMAT(//,5X,'NO. OF SUBSETS EXAMINED = ',I4)
  210 FORMAT(//,5X,'N=',I2)
  220 FORMAT(2I2,2F10.0)
  230 FORMAT(/,5X,'KN=',I3,3X,'NRUN=',I3)
  280 FORMAT(/,5X,'SIR = ',E10.4,3X,'SNR =',E10.4,3X
     *,'NVAR=',E10.4)
  290 FORMAT(//,9X,'FREQ1',7X,'FREQ-ERROR1',7X,'FREQ2'
     *,7X,'FREQ-ERROR2')
  300 FORMAT(/,2X,4(3X,E12.6))
  310 FORMAT(//,5X,'MEAN1 = ',E12.6,7X,'MEAN2 = ',E12.6)
  320 FORMAT(/,5X,'VARIANCE1=',E12.6,5X,'VARIANCE2=',E12.6)
  330 FORMAT(/,5X,'MERROR1=',E12.6,7X,'MERROR2=',E12.6)
  340 FORMAT(/,5X,'STDIV1  = ',E12.6,5X,'STDIV2  = ',E12.6)
  350 FORMAT(/,5X,'A1=',E10.4,3X,'A2=',E10.4)
  360 FORMAT(/,5X,'F1=',E10.4,3X,'F2=',E10.4)
      STOP
      END
C
C
C     SUBROUTINE INTG IS FOR EVALUATION OF ELEMENTS OF
C     A AND B MATRICES FOR ACH SUBSET TO BE EXAMINED,
C     EQUATIONS (4.2.14) THRU (4.2.16)
```

```
C
      SUBROUTINE INTG(A,COL,Y,IND,IP,N,NS,LFI,IE)
      COMMON PI,A1,A2,B1,B2,D1,D2,D3,D4,N1,N2,KN
      COMPLEX*16 A(40,40),COL(40),Y(100)
      REAL*8 D(40),AD(40),X,Y1,Y2,Y3,A1,A2,B1,B2,PI
      REAL*8 D1,D2,D3,D4
      INTEGER IND(40),IP(40),JP(40)
      IF(LFI.EQ.1)GO TO 2
      D(1)=D1
      DO 1 L=2,N
  1   D(L)=D(L-1)+D1
      GO TO 5
  2   D(1)=D3
      D(6)=D4
      DO 3 L=2,5
      D(L)=D(L-1)+D1
  3   D(L+5)=D(L+4)+D1
  5   IF(IE.EQ.0)GO TO 30
      IF(IE.EQ.-1)GO TO 25
      NS=N-1
      DO 20 L=1,N
      IF(L.NE.IE)GO TO 20
      DO 15 M=L,NS
 15   D(M)=D(M+1)
 20   CONTINUE
      GO TO 30
 25   K=0
      K1=1
      DO 26 K2=1,N
      IF(IND(K2).EQ.1)GO TO 26
      AD(K1)=D(IP(K2))
      K=K+1
      K1=K1+1
 26   CONTINUE
      DO 28 K3=1,K
      D(K3)=AD(K3)
      JP(K3)=K3
      DO 27 K4=1,K
      IF(D(K3).LE.AD(K4))GO TO 27
      D(K3)=AD(K4)
      JP(K3)=K4
 27   CONTINUE
      AD(JP(K3))=1.0E 6
 28   CONTINUE
      NS=K
 30   DO 60 I=1,NS
      DO 60 J=I,NS
      X=(D(J)-D(I))*2.0D0*PI
      IF(X)40,50,40
 40   CONTINUE
```

```
      A(I,J)=(1.0D0-CDEXP(DCMPLX(0.D0,X*KN)))
      A(I,J)=A(I,J)/(1.D0-CDEXP(DCMPLX(0.D0,X)))
      A(J,I)=DCONJG(A(I,J))
      GO TO 60
   50 A(I,J)=1.0D0*KN+.00001
   60 CONTINUE
      DO 80 K=1,NS
      COL(K)=(0.D0,0.D0)
      Y3=-2.D0*PI*D(K)
      DO 80 J=1,KN
      COL(K)=COL(K)+Y(J)*CDEXP(DCMPLX(0.D0,Y3*(J-1)))
   80 CONTINUE
      RETURN
      END
C
C
C     SUBROUTINE AHERIV IS FOR SOLVING THE LINEAR
C     EQUATIONS AC=B AND CORRESPONDING ERROR FUNCTION
C
      SUBROUTINE AHERIV(A,COL,SOL,Y,ET,K,KN)
      COMMON PI,A1,A2,B1,B2,D1,D2,D3,D4,N1,N2
      REAL*8 A1,A2,B1,B2,PI,C,VN(50),C1,C2
      COMPLEX*16 A(40,40),COL(40),RL(40,40),SOL(40)
      COMPLEX*16 XX1,Y(100),E1,E2,E3,ET,XX
      DO 5 I=1,K
      DO 5 J=1,K
    5 RL(I,J)=DCMPLX(0.0D0,0.0D0)
      DO 60 I=1,K
      IP1=I+1
      IM1=I-1
      IF(I.NE.1)GO TO 30
      DO 10 J=1,K
   10 RL(1,J)=A(1,J)
      DO 20 J=2,K
   20 RL(J,1)=DCONJG(RL(1,J))/RL(1,1)
      GO TO 60
   30 DO 40 J=I,K
      RL(I,J)=A(I,J)
      DO 40 M=1,IM1
   40 RL(I,J)=RL(I,J)-RL(I,M)*RL(M,J)
      IF(I.EQ.K)GO TO 60
      DO 45 J=IP1,K
   45 RL(J,I)=DCONJG(RL(I,J))/RL(I,I)
   60 CONTINUE
   80 DO 100 I=1,K
      IM1=I-1
      SOL(I)=COL(I)
      IF(I.EQ.1)GO TO 100
      DO 90 L=1,IM1
   90 SOL(I)=SOL(I)-SOL(L)*RL(I,L)
```

```
  100 CONTINUE
      DO 140 I=1,K
      J=K-I+1
      IF(J.EQ.K)GO TO 130
      JP1=J+1
      DO 120 L=JP1,K
  120 SOL(J)=SOL(J)-SOL(L)*RL(J,L)
  130 SOL(J)=SOL(J)/RL(J,J)
  140 CONTINUE
C
C     EVALUATION OF ERROR, EQUATION (4.2.19)
C
      E2=0.0D0
      E1=0.0D0
      DO 15 I1=1,K
      E2=E2+SOL(I1)*DCONJG(COL(I1))
  15  CONTINUE
      DO 16 I3=1,KN
      E1=E1+CDABS(Y(I3))**2
  16  CONTINUE
      ET=E1-E2
      RETURN
      END
C
C
C     SUBROUTINE COMBO IS FOR GENERATING STRING
C     OF ONES AND ZEROS FOR PROPER DELETION OF
C     BASIS FUNCTIONS
C
      SUBROUTINE COMBO(JN,N,M,NF)
      DIMENSION JN(40)
      IONES=M
      NF=0
  1   NP=1
  2   CONTINUE
      IF(JN(NP).EQ.0)GO TO 10
      IF(NP.EQ.N)GO TO 101
      JN(NP)=0
      NP=NP+1
      IONES=IONES-1
      GO TO 2
  10  JN(NP)=1
      IONES=IONES+1
      IF(IONES.EQ.M)GO TO 100
      GO TO 1
  101 NF=1
  100 RETURN
      END
C
C
```

```
C     SUBROUTINE GAUSS AND RANDU ARE FOR
C     GENERATING GAUSSIAN NOISE
C
      SUBROUTINE GAUSS(IX,S,AM,V)
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C
C
C     SUBROUTINE VARIAN IS FOR EVALUATING MEAN
C     AND VARIANCE OF THE FREQUENCY ESTIMATES
C
      SUBROUTINE VARIAN(ED1,ED2,DMEAN1,DMEAN2,VAR1,VAR2,NRUN)
      REAL*8 ED1(100),ED2(100),DMEAN1,DMEAN2,VAR1
      REAL*8 VAR2,X1,X2
      VAR1=0.D0
      VAR2=0.D0
      DMEAN1=0.D0
      DMEAN2=0.D0
      DO 10 I=1,NRUN
      DMEAN1=DMEAN1+ED1(I)
      DMEAN2=DMEAN2+ED2(I)
  10  CONTINUE
      DMEAN1=DMEAN1/NRUN
      DMEAN2=DMEAN2/NRUN
      DO 20 J=1,NRUN
      VAR1=VAR1+(ED1(J)-DMEAN1)**2
      VAR2=VAR2+(ED2(J)-DMEAN2)**2
  20  CONTINUE
      VAR1=VAR1/NRUN
      VAR2=VAR2/NRUN
      RETURN
      END
C
C
C     SUBROUTINE DELT IS FOR DELETING PROPER
C     ROW AND COLUMN OF A AND B MATRICES FOR
```

```
C     EVALUATION OF EI ERROR FUNCTIONS
C
      SUBROUTINE DELT(A,COL,S,DCOL,N,NS,IE)
      COMPLEX*16 A(40,40),COL(40),S(40,40),DCOL(40)
      IEM=IE-1
      IF(IE.EQ.1)GO TO 30
      DO 10 I=1,IEM
      DCOL(I)=COL(I)
      DO 10 J=I,IEM
      S(I,J)=A(I,J)
      S(J,I)=A(J,I)
   10 CONTINUE
      DO 20 I=1,IEM
      DO 20 J=IE,NS
      S(I,J)=A(I,J+1)
      S(J,I)=A(J+1,I)
   20 CONTINUE
      IF(IE.EQ.N)GO TO 50
   30 DO 40 I=IE,NS
      DCOL(I)=COL(I+1)
      DO 40 J=I,NS
      S(I,J)=A(I+1,J+1)
      S(J,I)=A(J+1,I+1)
   40 CONTINUE
   50 RETURN
      END
```

REFERENCES

1. R. W. Hamming, "Digital Filters," Englewood Cliffs, N. J. : Printice-Hall, 1977.

2. Steven A. Tretter, "Introduction to Discrete-time Signal Processing," New York: Wiley, 1976.

3. D. W. Tufts, D. W. Rorabacher, and W. Mosier, "Designing Simple, Effective Digital Filters", IEEE Trasactions on Audio and Electroacoustics, vol. AU-18, pp. 142-158, June 1970.

4. D. W. Tufts, J. T. Francis, "Designing Digital Low-pass Filters - Comparison of Some Methods and Criteria", IEEE Transactions on Audio and Electroacoustics, vol. AU-18, pp. 487-494, Dec. 1970

5. John Makhoul, "Linear Prediction: + Tutorial Review," Proceeding of the IEEE, April 1975.

6. Alan V. Oppenheim & Ronald W. Schafer, "Digital Signal Processing," Prentice-Hall, 1975.

7. M. Kaveh, and G. R. Cooper, "An Empirical Investigation of the Properties of the Autoregressive Spectral Estimator", IEEE Transactions on Information Theory, May 1976.

8. David C. Rife, "Digital Tone Parameter Estimation in the Presence of Gaussian Noise," Department of electrical Engineering, Ph.D. Dissertation, June 1973, Polytechnic

Institute of Brooklyn, Brooklyn, NY.

9. D. C. Rife and G. A. Vincent, "Use of the Discrete Fourier Transform in the Measurement of Frequencies and Levels of Tones," The Bell System Technical Journal, February 1970.

10. R. R. Hocking and R. N Leslie, "Selection of the Best Subset in Regression Analysis," Technometrics, Vol. 9, 1967, pp. 531-540.

11. C. L. Mellows, "Chocsing Variables in a Linear Regression," A Graphical all Presented at the Central Regional Meeting Of the Institude of Mathematical Statistics, 1966, Manhattan Kansas.

12. L. R LeMotte and R. R. Hocking, "Computational Efficiency in the Selection of Regression Variables," Technometrice, Vol. 12, 1970, pp. 83-93.

13. Kenneth M. Levasseur and James T. lewis, "Antenna Design as a Partial Basis Problem," Department of Mathematics, Technical Report No. 66, June 1977, University of Rhode Island, Kingston, RI .

14. A. Papoulis, "Probability, Random Variables, and Stochastic Processes," New York" MaGraw-Hill Book Company, 1965.

15. R. Kumaresan, "Non-linear Frequency Estimatiom", Department of Electrical Engineering, Ms Thesis, August 1979, University Of Rhode Island, Kingston, RI .

16. F. Mosteller, and R. E. K. Rourke, "Sturdy Statistics", Addison-Wesley Publishing Company, 1973.

E

ED

80