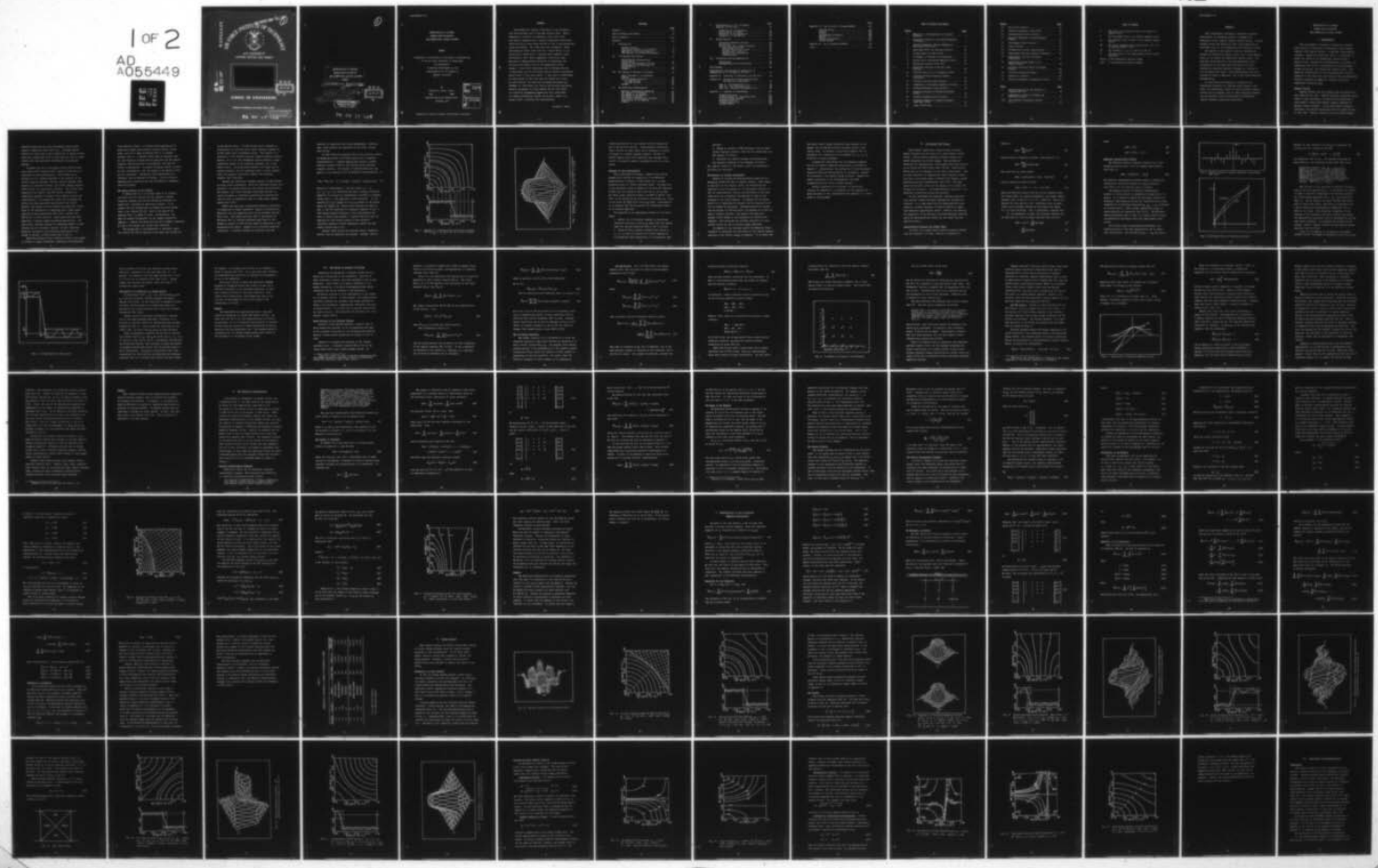
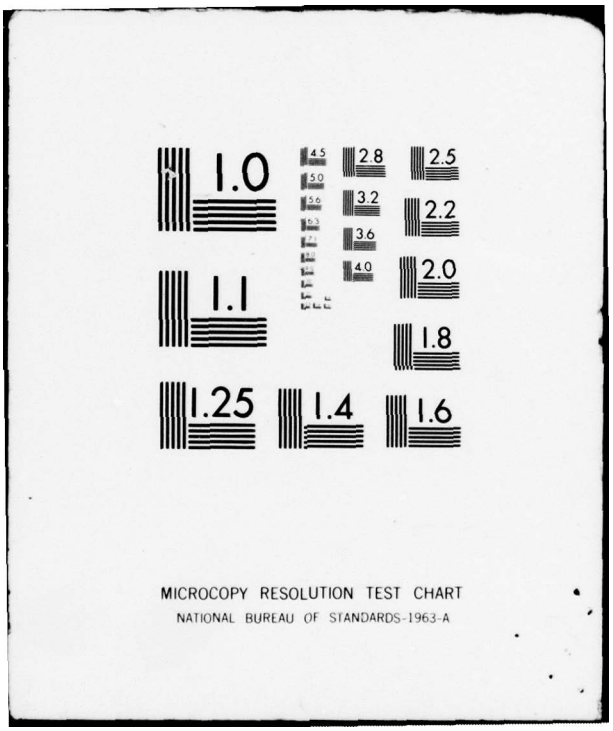
AD-A055 449 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/3
INVESTIGATION OF OPTIMAL LINEAR SHIFT-INVARIANT TWO-DIMENSIONAL--ETC(U)
DEC 77 R B BROWN

UNCLASSIFIED AFIT/GE/EE/77-10 NL

1 OF 2
AD
A055449




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

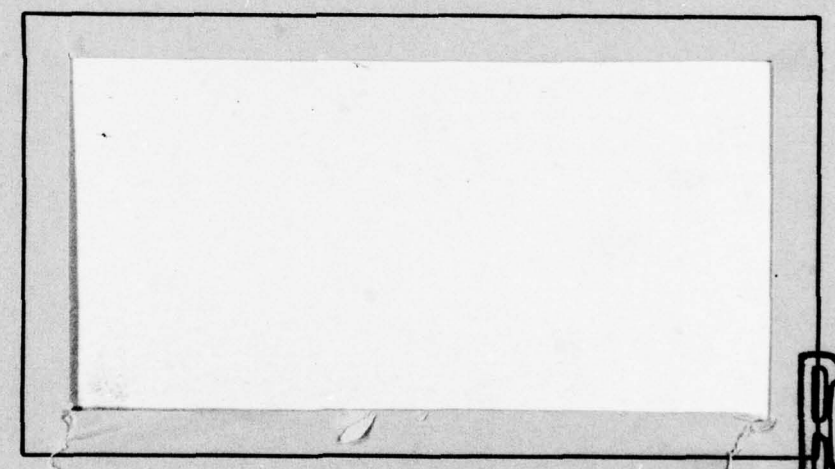
AD A055449

FOR FURTHER TRAN

# AIR FORCE INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY
UNITED STATES AIR FORCE

AD No.
DDC FILE COPY



DDC
RECEIVED
JUN 22 1978
E

## SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

78 06 12 126

(1)

(6) INVESTIGATION OF OPTIMAL
LINEAR SHIFT-INVARIANT
TWO-DIMENSIONAL DIGITAL FILTERS.
THESIS

(14) AFIT/GE/EE/77-10

Richard B. Brown
Capt USAF

(10)

(11) Dec 77 (12) 137 p.

(9) Master's thesis,

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DDC
RECEIVED
JUN 22 1978
E

78 06 13 186

012 225 JOB

AFIT/GE/EE/77-10

# INVESTIGATION OF OPTIMAL LINEAR SHIFT-INVARIANT TWO-DIMENSIONAL DIGITAL FILTERS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Richard B. Brown, B.E.E.
Capt USAF
Graduate Electrical Engineering
December 1977

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

Approved for public release: distribution unlimited.

## Preface

Two-dimensional digital filtering is a new discipline that has developed only in the past several years. When I expressed an interest in studying in this field, Prof. Gary Lamont suggested that I attend a Rennselaer Polytechnic Institute short course being offered in July on two-dimensional signal processing. The course was very worthwhile. While discussing my thesis topic with Prof. Sanjit Mitra, of the University of California, I received some wonderful ideas. Additionally, Prof. Mitra suggested I write Prof. Russ Mercereau of Massachusetts Institute of Technology, who had done some similar work in two-dimensional filter design. I wrote Prof. Mercereau, now of Georgia Institue of Technology, and he very kindly sent me several unpublished papers which I found very useful. I also want to acknowledge the many people in the local area who helped me so much: Dr. Phil Pourier, for help with cataloged procedures and DISSPLA; Dr. John Hines, for time on the Avionics Laboratory graphics equipment; Lt. Stan Larimer and Lt. Steve Enke, for countless programming suggestions; Mrs. Diana Hake, for highly professional typing; and my wife, Randi, for being a model of patience and understanding.

Richard B. Brown

## Contents


| | | Page |
|---|---|---|
| Preface | | ii |
| List of Figures and Tables | | vi |
| List of Symbols | | viii |
| Abstract | | ix |
| I. | Introduction | 1 |
| | Digital Filters | 1 |
| | The Design Problem of 2-D Filters | 3 |
| | Approach of this Investigation | 8 |
| | The Sequence of Problem Development | 9 |
| II. | 1-D Optimal FIR Filters | 11 |
| | Finite-Extent Sequences and Linear Phase | 11 |
| | Chebychev Approximation Problem | 13 |
| | Development of an Efficient 1-D Design Method | 17 |
| | Summary | 18 |
| III. | The Design of Optimal 2-D Filters | 19 |
| | Basic Concepts of 2-D Discrete Signals | 19 |
| | The 2-D Design Objective | 21 |
| | 2-D IIR Filters | 22 |
| | 2-D FIR Filters | 26 |
| | Summary | 31 |
| IV. | The McClellan Transformation | 32 |
| | Spectral Transformation Mappings | 32 |
| | The Change of Variables | 33 |
| | The Range of the Mapping | 37 |
| | The Design Strategy | 38 |
| | The Contour Approximation Problem | 39 |
| | Constraints on the Mapping | 41 |
| | Summary | 49 |

| | | Page |
|---|---|---|
| V. | Determination of the 2-D Impulse Response Coefficients | 51 |
| | Algorithm for the Expansion | 51 |
| | An Analogous 1-D Result | 53 |
| | Extension to Two-Dimensions | 55 |
| | Comparison of Optimality | 58 |
| VI. | Design Results | 62 |
| | Circularly Symmetric Filters | 62 |
| | Fan Filters | 66 |
| | Filters with More Complex Contours | 77 |
| | Oval-Shaped Contours | 77 |
| | Contour Defined by a Cubic | 77 |
| | Non-Monotonic Contours | 80 |
| | Contours not Continuously Differentiable | 80 |
| VII. | Conclusions and Recommendations | 85 |
| | Conclusions | 85 |
| | Recommendations for Future Work | 87 |
| Bibliography | | 88 |
| Supplementary Bibliography of References on Implementation of 2-D Digital Filters | | 91 |
| Appendix A: | Algorithm to Determine $\underline{U}$ for Eq (40) | 92 |
| Appendix B: | The Method of Least Squares Curve Fitting with Constraints | 95 |
| | Case I: No Constraints on $\underline{t}$ | 95 |
| | Case II: One Constraint on $\underline{t}$ | 96 |
| | Case III: Multiple Constraints on $\underline{t}$ | 99 |
| Appendix C: | Comments on Programming | 101 |
| | Program Structure | 101 |
| | Considerations for Interactive Use | 103 |
| | Avoiding SCOPE Commands | 104 |
| | Program Flowchart | 105 |
| | Program Source Code | 105 |
| | Program PROFIL | 105 |

Page

Appendix D: User's Guide to Program MCLTRN . . . . . 108

Identification . . . . . 108
Purpose . . . . . 108
Control . . . . . 108
Programming Information . . . . . 109
Method . . . . . 111
Special Conventions. . . . . 112
Design Example . . . . . 112

Appendix E: Use of Advanced DISSPLA . . . . . 122

Vita . . . . . 124

## List of Figures and Tables


| Figure | | Page |
|---|---|---|
| 1 | Mapping of 1-D Frequencies to Circular Contours . . . . . | 6 |
| 2 | Frequency Response of Circularly Symmetric 2-D Filter. . . . . | 7 |
| 3 | Positive Symmetric Impulse Response of Zero-Phase FIR Filter . . . . . | 14 |
| 4 | Equiripple Error for Function Sin(x/2). . . . | 14 |
| 5 | An Equiripple Low Pass Filter . . . . . | 16 |
| 6 | An Example of Possible 2-D Recursions . . . . | 24 |
| 7 | Slices from a Circularly Symmetric Filter . . | 27 |
| 8 | Elliptical Contours of Eq (67) . . . . . | 45 |
| 9 | Elliptical Contours of Eq (67) after Scaling . . . . . | 48 |
| 10 | Periodic Nature of a 2-D Digital Filter . . . | 63 |
| 11 | Circular Contours Using the Default Constraint . . . . . | 64 |
| 12 | Circular Contours after Scaling . . . . . | 65 |
| 13 | Comparison of Circularly Symmetric Filters. . | 67 |
| 14 | Elliptical-Shaped Filter Contours . . . . . | 68 |
| 15 | Frequency Response of Elliptical-Shaped Filter . . . . . | 69 |
| 16 | Parabolic-Shaped Filter Contours. . . . . | 70 |
| 17 | Frequency Response of Parabolic-Shaped High Pass Filter . . . . . | 71 |
| 18 | Ideal "Fan" Filter . . . . . | 72 |

| Figure | | Page |
|---|---|---|
| 19 | Fan Filter Contours . . . . . | 73 |
| 20 | Frequency Response of Fan Filter . . . . . | 74 |
| 21 | Diamond-Shaped Filter Contours . . . . . | 75 |
| 22 | Frequency Response of Diamond-Shaped Filter . . . . . | 76 |
| 23 | Oval-Shaped Filter Contours . . . . . | 78 |
| 24 | Cubic Contours . . . . . | 79 |
| 25 | Non-Monotonic Contour Specification . . . . . | 81 |
| 26 | Right-Angle Contours; Default Constraint. . . | 82 |
| 27 | Right-Angle Contours: Alternate Constraints . . . . . | 83 |
| 28 | Approximate Design Times of 2-D Design Program . . . . . | 86 |
| 29 | First-Cut Structure Chart for Design Program . . . . . | 102 |
| 30 | Top-Level Program Flowchart . . . . . | 107 |
| 31 | Hyperbolic Contours . . . . . | 120 |
| 32 | Frequency Response of Hyperbolic Filter . . . | 121 |

| Table | | Page |
|---|---|---|
| I. | Storage Required for Two Methods of Expanding Eq (91) . . . . . | 53 |
| II. | Comparison of Error in Optimal and McClellan Filters . . . . . | 61 |
| III. | The Chebychev Polynomials Solved for x . . . . . | 93 |

## List of Symbols

| | |
|---|---|
| $N$ | The order of the digital filter (the number of sample points) |
| $\delta$ | The equiripple error in an optimal digital filter |
| $\delta_i$ | The weighted error in Band i of an optimal digital filter |
| Wt | The error weighting ratio between Band i and j of an optimal digital filter |
| $\underline{t}^T$ | The elements $t_1$, $t_2$, $t_3$, and $t_4$ in Eq (1) |
| $\bar{e}$ | The mean error in the least-squares curve fitting algorithm |
| $\hat{x}(m,n)$ | A two-dimensional discrete signal |
| $x(n)$ | A one-dimensional discrete signal |

## Abstract

This investigation develops an interactive computer aided method for designing lowpass or highpass two-dimensional finite impulse response digital filters. Filters designed using this method will have linear phase and an equiripple error in the transmission and attenuation bands.

The essence of the method is that it transforms an optimal one-dimensional digital filter into a close approximation of an optimal two-dimensional digital filter. The amplitude characteristic of the one-dimensional filter is preserved in the sense that each point of the one-dimensional frequency response is mapped to a contour in the two-dimensional plane. This transformation was first proposed by James H. McClellan, and is now called McClellan Transformation.

By controlling the mapping of a specified one-dimensional frequency to a desired contour shape in the plane, two-dimensional filters of fairly arbitrary specifications can be designed; that is, their frequency response can be determined, and the associated two-dimensional impulse response coefficients calculated.

# INVESTIGATION OF OPTIMAL LINEAR SHIFT-INVARIANT TWO-DIMENSIONAL DIGITAL FILTERS

## I. Introduction

This investigation concerns an interactive computer aided method for designing two-dimensional finite impulse response (FIR) digital filters. A spectral transformation technique, suggested by McClellan (Ref 1), is extended to permit the design of two-dimensional filters within a special class. Through this transformation, an optimal one-dimensional FIR filter is transformed into an equiripple, though sub-optimal, two-dimensional digital filter. The amplitude characteristic of the one-dimensional filter is preserved in the sense that each point of the one-dimensional frequency response is mapped to a contour in the two-dimensional frequency plane.

### Digital Filters

Digital systems deal with signals that are discrete in both amplitude and time. Digital filtering, sometimes called digital signal processing, is involved with operations on such signals, using either general purpose computers or special purpose hardware. In the early years of digital filtering, the required processing could not always be done in real time. However, advances such as the Fast Fourier

Transform algorithm now permit processing times several orders of magnitude faster (Ref 2:3). Although digital filters are often used as mere simulations of analog filters, their most significant value is when they are used to create filters with characteristics unobtainable using analog methods.

To permit the use of well-known linear system theory, only linear shift-invariant (LSI) digital filters will be considered in this investigation. LSI filters can be divided into two classes depending on the duration of the filter's impulse response; infinite impulse response (IIR) filters (or recursive filters), and finite impulse response (FIR) filters (or nonrecursive filters). IIR filters were originally more popular than FIR filters, because the traditional approach to digital filter design involves the transformation of an analog IIR filter into a digital IIR filter with some prescribed specification (Ref 2:197). However, FIR filters have two attractive properties. First, there is the possibility of designing exact linear phase, required in many applications (Ref 7:76). Second, the FIR filter is never unstable. These qualitites are often important in digital signal processing applications. For many years, efficient methods for designing optimal 1-D FIR filters were not available; however, in 1972, Parks and McClellan published a very efficient algorithm for the design of optimal 1-D FIR filters (Ref 3, Ref 4).

With this problem solved, attention turned to the design of filters of higher dimensions, especially two-dimensional

(2-D) digital filters. 2-D filters find applications in areas where signal specification requires spatial coordinates, such as in image processing (Ref 5) or seismic analysis (Ref 6). A digital filter must be designed (that is, the frequency response must be specified and the impulse response coefficients must be determined), and then the filter is implemented. To permit the proper amount of detail in this investigation, only the design of the digital filter will be considered. Recognizing that implementation is certainly just as important as design, a short listing of implementation references has been included as a supplementary bibliography.

The Design Problem of 2-D Filters

A design goal for a 1-D filter might be to minimize the error over the interval $[-\pi,\pi]$ between a desired frequency response $D(\omega)$ and the obtainable frequency response $H(\omega)$, according to some error criteria. It is assumed that the frequencies of interest have been normalized to the interval $[-\pi,\pi]$. This is the case if the sampling time T is equal to unity. Alternatively, one maps the sampled frequencies via the complex exponential mapping $z = \exp(sT)$ and deals with the set of angles, repeating $[-\pi,\pi]$ on the Z-plane unit circle (Ref 2:199-200).

The design goal in two-dimensions is analogous, where the interval $[-\pi,\pi]$ is replaced by the region $[-\pi,\pi] \times [-\pi,\pi]$

in the spatial plane. 2-D FIR filters can be designed by multiplying the infinite-extent ideal frequency response by an NxN sample point rectangular window. This amounts to a truncation of the infinite duration impulse response sequence. However, as in 1-D, the rectangular window produces a large, undesirable ripple in the frequency response, due to the Gibb's effect. By using better windows, the ripple can be reduced somewhat, but the resulting filter is never optimal, since the required convolution "smears" the frequency response (Ref 7:239-250).

Rabiner and others have designed optimal 2-D FIR filters by using linear programming. While the resulting filters were excellent, the design method was very costly and inefficient. The highest order filter designed (9x9 sample points) involved thousands of constraint equations and required more than one hour of computation time on a high speed computer system (Ref 7:471).

Effort has been made to generalize McClellan and Park's 1-D algorithm to higher dimensions, despite an assertion by McClellan that the algorithm could not be extended directly (Ref 1:30). Both Kamp and Thiran (Ref 8) and Hersey and Mercereau (Ref 9) recently published algorithms for the design of optimal 2-D FIR filters based on variations of the Parks-McClellan method. Despite use of efficient numerical techniques, no dramatic decrease in execution time was

achieved in comparison with linear programming. Additionally, these methods are restricted to low order filters (Ref 10:240).

In 1973, McClellan proposed that a more efficient method of designing optimal 2-D filters would be by a "spectral transformation", a complex mapping that carries a stable rational transfer function into another stable rational transfer function. The change of variables McClellan suggested, to be referred to as the McClellan Transformation, is

$$\cos\omega = F(\omega_1,\omega_2) = t_1 + t_2\cos\omega_1 + t_3\cos\omega_2 + t_4\cos\omega_1\cdot\cos\omega_2 \quad (1)$$

Using this transformation, with the values $t_1 = -.5$, $t_2 = t_3 = t_4 = 0.5$, McClellan was able to design circularly symmetric 2-D filters by transforming a suitable 1-D optimal filter (Ref 1). The magnitude of each frequency in the interval $[-\pi,\pi]$ is mapped to an $\omega_1$-$\omega_2$ contour in the square region $[-\pi,\pi]\times[-\pi,\pi]$. Several such contours are shown in Fig. 1, and the resulting 2-D frequency response in Fig. 2. (The design program developed in this investigation was used to obtain these figures.) Because there are many 2-D applications that either require or can tolerate a circularly symmetric filter, McClellan's method has been widely praised (Ref 9:1-2).

Because these filters are obtained using a transform method, they are generally sub-optimal. However, they do

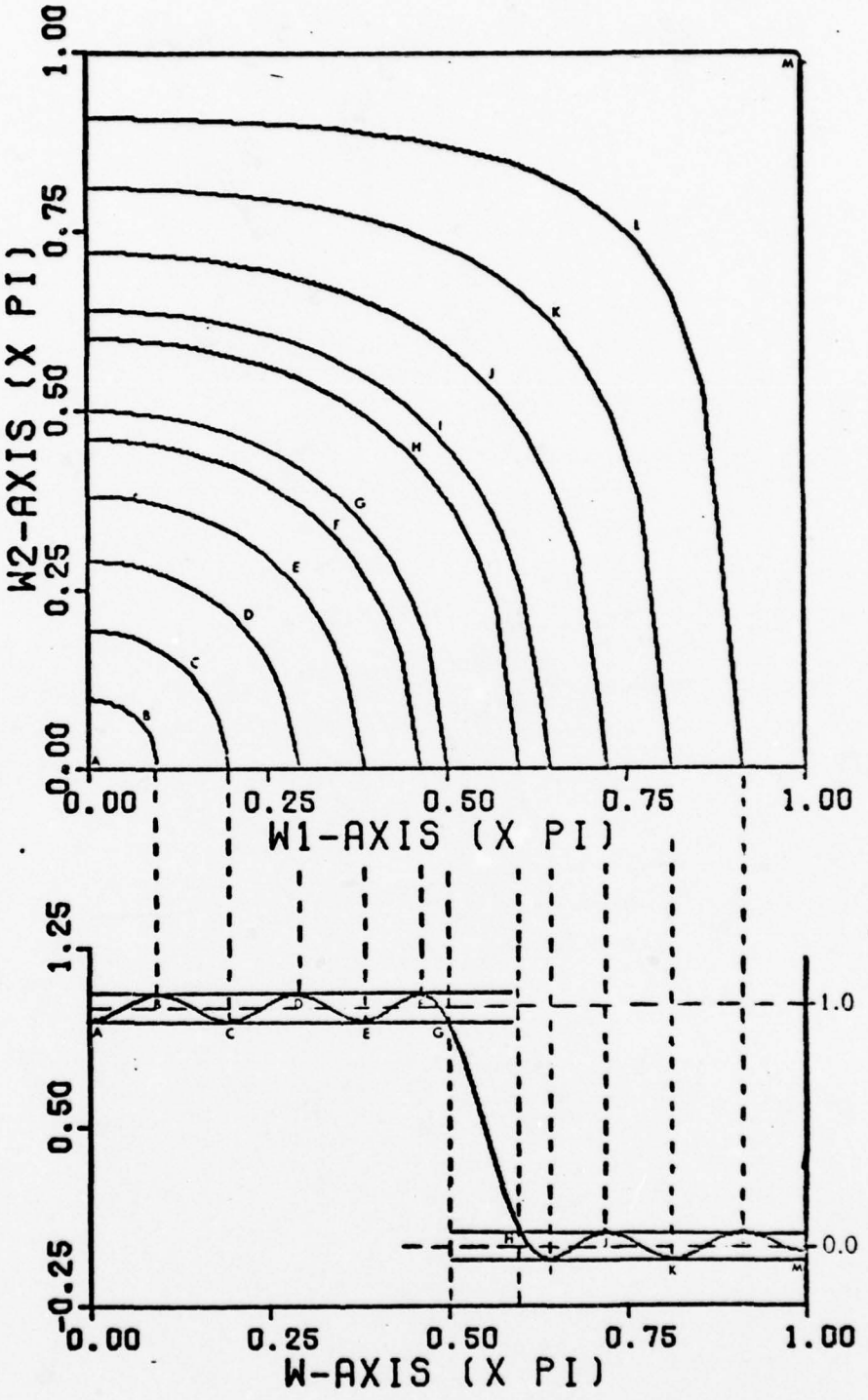



Fig. 1. Mapping of 1-D Frequencies to Circular Contours. N = 21, $\omega_p$ = .5π, $\omega_s$ = .6π, Wt = 1:1, δ = 0.055

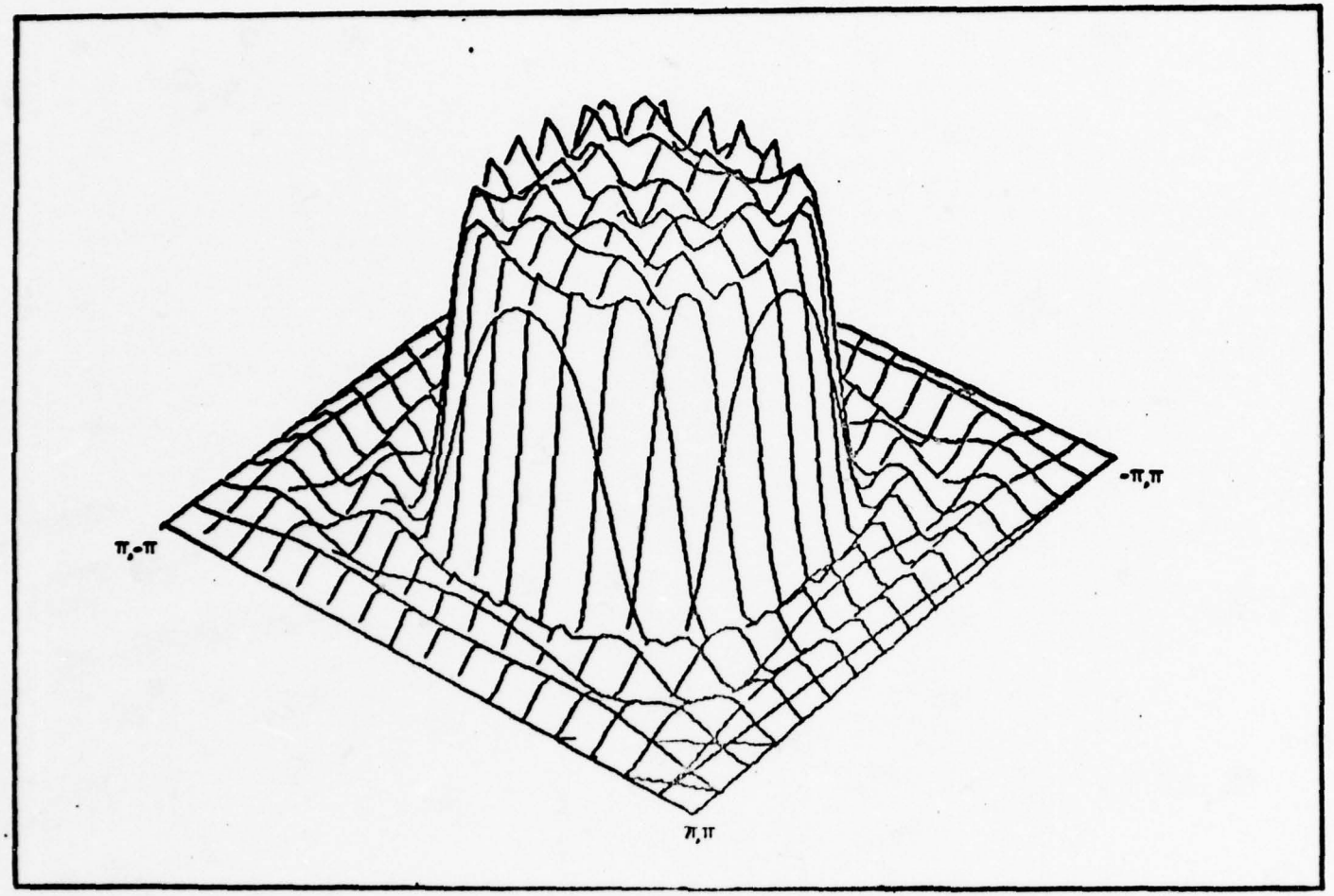



Fig. 2. Frequency Response of Circularly Symmetric 2-D Filter.
N = 21, $\omega_p$ = .50π, $\omega_s$ = .60π, Wt = 1:1, δ = 0.055

closely approximate the true optimal filters designed by any of the previous methods. Using McClellan Transformations, filters of large orders can be designed in a matter of seconds on a general purpose computer. Filters of 127x127 sample points have reportedly been designed (Ref 10:240), although the memory requirements would be considerable.

Approach of this Investigation

This investigation develops a computer-aided design program which extends the capability of the McClellan Transformation to design 2-D FIR filters with a single transition band of a fairly arbitrary shape. Included within the admittable class of shapes are low pass and high pass filters that are symmetric about both the $\omega_1$ and $\omega_2$ axes, and whose single transition band is bounded by smooth, monotonic curves defined by the functional relationship $\omega_2 = F(\omega_1)$ in the region $[0,\pi]x[0,\pi]$ of the $\omega_1$-$\omega_2$ plane. Additionally, it is required that the attenuation and transmission bands be of constant magnitude.

The algorithm to be implemented consists of four major steps:

1. Define the 2-D frequency response by specifying admittable contours in the $\omega_1$-$\omega_2$ plane that will approximate the desired transition band of the 2-D filter.
2. Using an error criteria, determine the values $t_1$, $t_2$, $t_3$, $t_4$ that will produce the closest mapping of 1-D transition band frequencies to 2-D transition band

contours.

3. Design an optimal 1-D FIR prototype, with an appropriate frequency response, that will be transformed into the desired 2-D filter.

4. Calculate the impulse response coefficients and the frequency response of the designed 2-D filter.

The complete details of the method, until now unavailable, are logically developed.

The Sequence of Problem Development

Chapter II reviews the McClellan-Parks algorithm for designing optimal LSI FIR 1-D digital filters. This chapter is included for two reasons: first, the Parks-McClellan algorithm is used to design the 1-D prototypes required for the 2-D transformation; second, the principles in the algorithm serve as an essential introduction to some of the concepts of the later chapters. In Chapter III the mathematics of 2-D sequences is reviewed, and the required tools for 2-D design are developed. Although methods for designing non-optimal 2-D filters exist, the optimal 2-D filter remains a stubborn problem. The reasons that make true optimal filter design in two dimensions so difficult are examined. This motivates the ultimate selection of a suboptimal method for the 2-D filter design algorithm.

In Chapter IV the rationale behind the McClellan Transformation is discussed, and the nature of the complex mapping required in the contour design is examined. It is shown that

the method cannot design transition band contours of all shapes, but can design arbitrary contours within a very useful class. To produce a non-trivial mapping, the requirements for constraints on the elements $t_1$, $t_2$, $t_3$, $t_4$ of Eq (1) is shown necessary.

A method for calculating the 2-D frequency response and the 2-D impulse response coefficients is outlined in Chapter V. Additionally, the optimality of filters designed using the McClellan Transformation is considered. Chapter VI illustrates a few typical design problems. Chapter VII draws conclusions on the investigation, and makes recommendations for future related research.

Several appendices are included. Of particular interest are Appendix C, a discussion of the computer program developed in this investigation and Appendix D, a user's guide to this program.

## II. 1-D Optimal FIR Filters

This chapter begins with a brief review of finite-extent sequences and the nature of a linear phase 1-D FIR filter. Finite duration sequences possess certain very useful properties for filter design. Such problems as stability and implementation, which might affect IIR filters, do not arise with FIR filters (Ref 7:75-76). Additionally, FIR filters can be designed to have exact linear phase. This is often important for applications where frequency dispersion due to non-linear phase is harmful (for example, speech processing and data transmission) (Ref 7:76). For many years no efficient method for the design of 1-D optimal FIR filters was available. The Parks-McClellan algorithm, to be described, effectively solved this problem.

To best understand the Chebychev (optimal) approximation problem, certain necessary mathematical theorems are reviewed. It is then shown that the Parks-McClellan algorithm is a natural extension of earlier efforts to apply this Chebychev approximation theory. However, the resourceful adaptation of the efficient 2nd Remez Exchange algorithm gives the Parks-McClellan method far more power than any previous method.

### Finite-Extent Sequences and Linear Phase

If $\{h(n)\}$ is a causal finite duration sequence defined over the interval $0 \leq n \leq 2N-1$, then the z-transform of

{h(n)} is

$$H(z) = \sum_{n=0}^{2N-1} h(n) z^{-n} \tag{2}$$

and the Fourier transform of {h(n)}, with period $2\pi$, is

$$H(\omega) = \sum_{n=0}^{2N-1} h(n) \exp(-j\omega n) \tag{3}$$

The condition for linear phase,

$$H(\omega) = G(\omega)\exp(j(A + B\omega)), \quad G(\omega) \text{ Real} \tag{4}$$

will be achieved if (Ref 7:77-78)

$$h(n) = h(2N - 1 - n), \quad 0 \leq n \leq 2N-1 \tag{5}$$

Parks and McClellan showed that one of four possible cases for linear phase is a positive-symmetric, odd-length impulse response with, in Eq (4), A = 0, B = -(2N-1)/2. (Ref 1:10). Because the term exp j(A + Bω) contributes only to phase, {h(n)} can be assumed centered at the origin with length N + 1 in each direction. Fig. 3 illustrates a typical case.

If this assumption is made, the frequency response can be rewritten as a sum of cosine functions (Ref 7:81-82):

$$G(\omega) = h(0) + 2 \sum_{n=1}^{N} h(n)\cos\omega n \tag{6}$$

$$= \sum_{n=0}^{N} a(n)\cos\omega n \tag{7}$$

where

$$a(0) = h(0) \tag{8}$$

$$a(n) = 2h(n), \quad n \neq 0 \tag{9}$$

Chebychev Approximation Problem

By defining a desired frequency response $D(\omega)$, and a weighting function $W(\omega)$, then the weighted approximation error $E(\omega)$ is

$$E(\omega) = W(\omega)\left[D(\omega) - G(\omega)\right] \tag{10}$$

The Chebychev approximation problem amounts to finding the coefficients $\{a(n)\}$ that minimize the maximum absolute error $E(\omega)$ over the frequency band of interest.

To properly develop the Chebychev approximation problem, a certain amount of mathematical background is required. It is useful to extend the well-known concept of a polynomial approximating function by considering some linear combinations of prescribed functions $g_1, g_2, \ldots, g_n$ continuous on a fixed space X. Their linear combinations $\Sigma c_i g_i$ are termed generalized polynomials (Ref 11:72). The Existence Theorem guarantees that to each f e X there exists at least one generalized polynomial which best approximates f (Ref 11:20).

For certain types of generalized polynomials, the characterization of the best approximation can be stated very conveniently. The functions $g_1, g_2, \ldots, g_n$ are said to

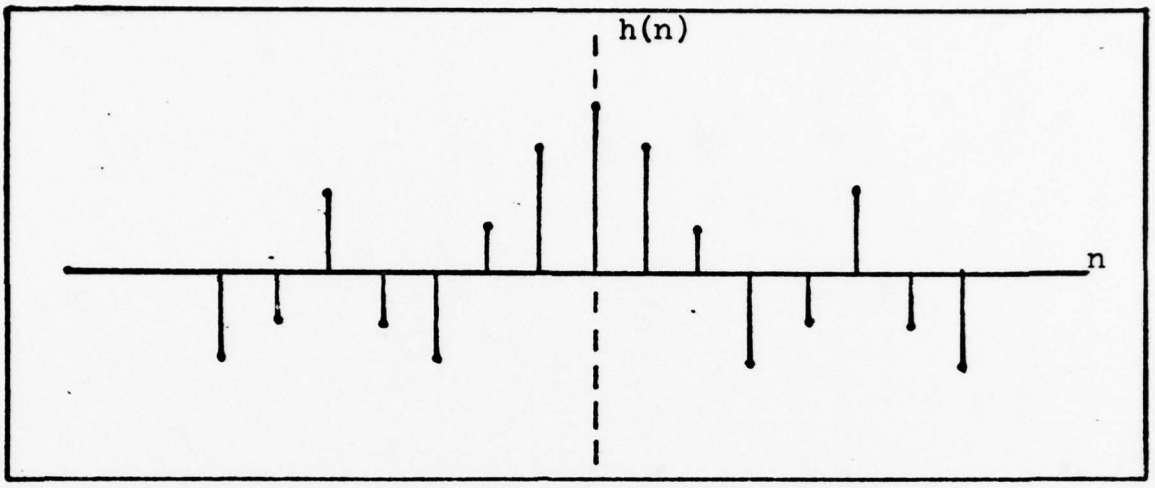



Fig. 3. Positive Symmetric Impulse Response of Zero-Phase FIR filter

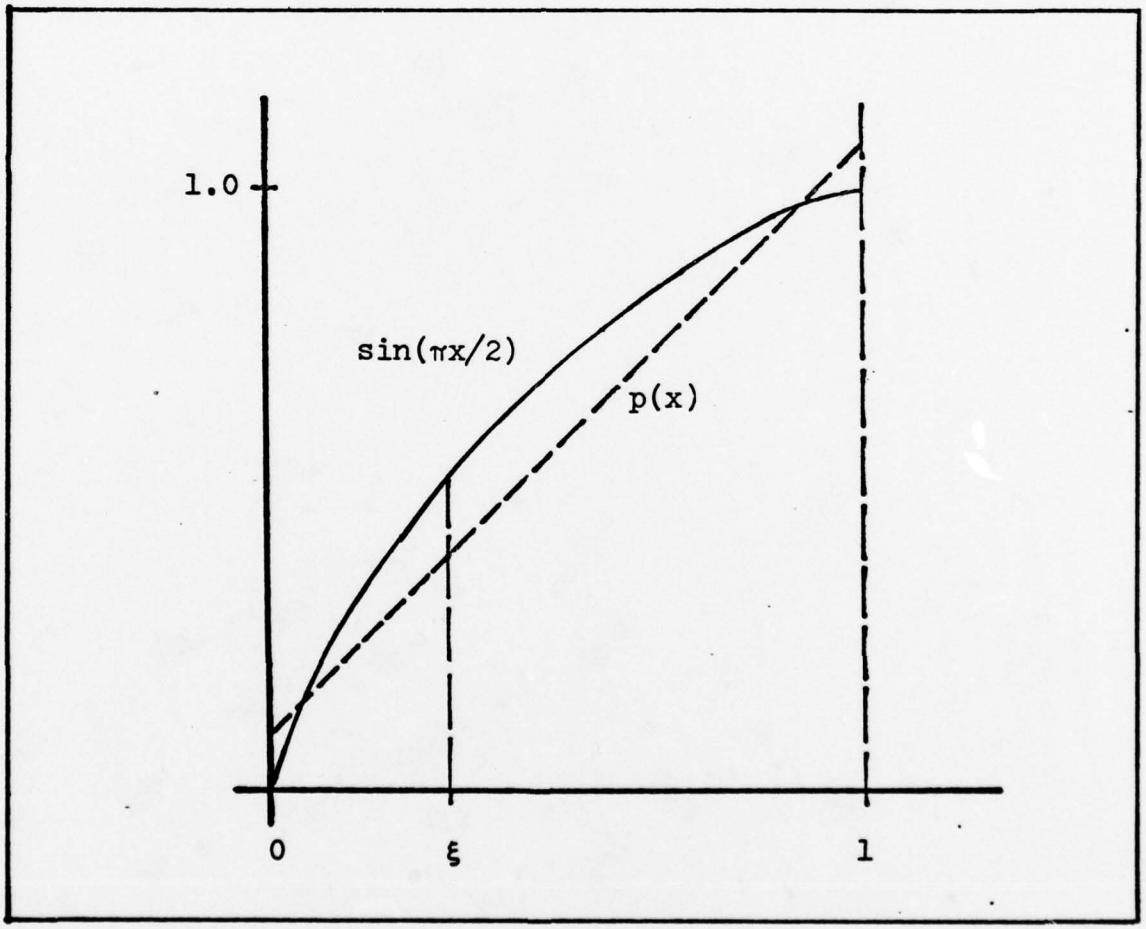



Fig. 4. Equiripple Error for Function sin(πx/2)

satisfy the Haar Condition if each $g_1$ is continuous and if every set of n vectors of the form

$$\underline{x} = [g_1(x), g_2(x), \ldots, g_n(x)] \tag{11}$$

is independent (Ref 11:45). For systems satisfying the Haar Condition, the following theorem holds (Ref 11:75)

> Alternation Theorem. Let $g_1, \ldots, g_n$ be a system of elements of $C[a,b]$ satisfying the Haar condition, and let X be any closed subset of $[a,b]$. In order that a certain generalized polynomial $P = \Sigma c_i g_i$ shall be a best approximation on X to a given $f \in C[X]$ it is necessary and sufficient that the error function $r = f - P$ exhibit on X at least n + 1 alternations. Thus $r(x_i) = -r(x_{i-1})$.

This theorem is very important in numerical determination of the best approximations. To illustrate the theorem, it might be desired to approximate the function $\sin(\pi x/2)$ by a first degree polynomial $P(x) = c_o + c_1 x$ over $[0,1]$ (Fig. 4). By the theorem, the error function must alternate at least three times. The points of alternation are 0 and 1, which can be determined by inspection, and $\xi$, which is not known. This situation is typical: if the location of the extrema of the error function were known, the approximating polynomial could be determined by solving some linear equations. To determine the location of the extrema, however, usually involves solving some non-linear equations (Ref 11:75-76).

If Eq (10) is applied to the design of an optimal bandpass filter, the number of extrema of $E(\omega)$ will be the

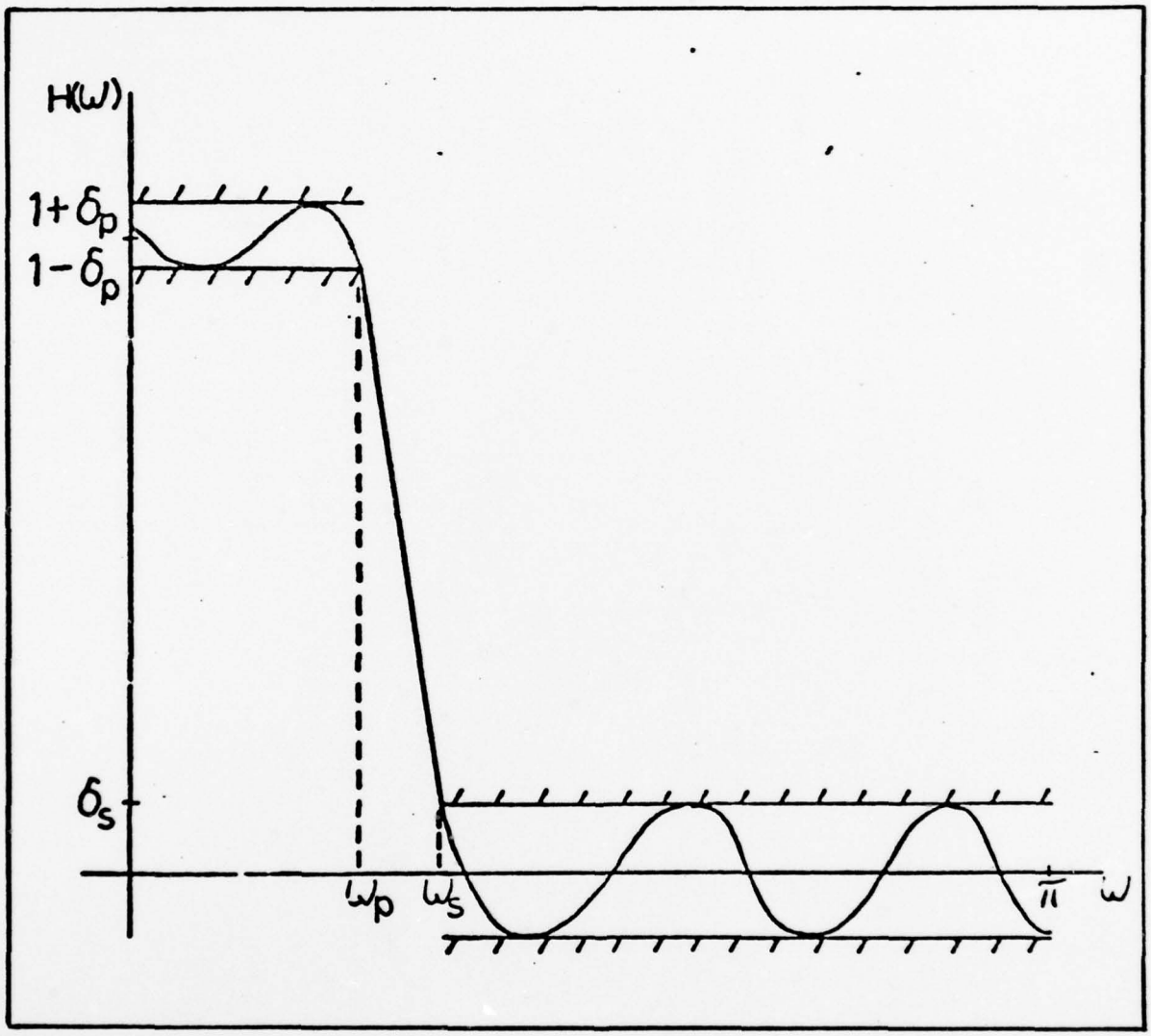



Fig. 5. An Equiripple Low Pass Filter

sum of extrema of $G(\omega)$ plus the mandatory extrema associated with a constraint at each band edge (Fig. 5). In general, the extrema at the band edges (except at $\omega = 0$ and $\omega = \pi$) will not be extrema of $G(\omega)$ (Ref 7:129). "Extra-ripple" may therefore be present, since more than n + 1 alternations might exist.

Development of an Efficient 1-D Design Method

By specifying N and $\delta_1$ ($\delta_2 = \kappa\delta_1$) and allowing $\omega_s$ and $\omega_p$ to be free variables, Hermann designed extraripple optimal FIR filters. He was restricted to designs of fairly low-order filters because of an inefficient method for solving the non-linear equations which locate the extremal frequencies (Ref 2:258).

Using a method of polynomial interpolation "reminiscent of the Remez Exchange algorithm", Hofstetter et al provided an efficient method for determining these extremal frequencies (Ref 12). This avoided the restriction on the filter order, but still left $\omega_s$ and $\omega_p$ as free variables.

Parks and McClellan completed the solution by developing a method where N, $\omega_s$, and $\omega_p$ are fixed, and $\delta_1$ ( $\delta_2 = \kappa\delta_1$) is allowed to vary (Ref 3, Ref 4). By properly interpreting the Alternation Theorem, they developed an algorithm capable of designing optimal FIR filters with the minimum number of error alternations. Following on Hofstetter's work, they incorporated the very efficient 2nd Remez Multiple Exchange algorithm (Ref 13) to provide their method with great speed.

For example, a 100 sample point filter can be designed in about 20 seconds (Ref 4:97). If $\delta_1$ must meet some tolerance, the Parks-McClellan method can be repeated with adjustments in one of the parameters.

The source listing of Parks and McClellan's FORTRAN program for designing optimal FIR filters, as well as an excellent discussion of the 2nd Remez algorithm, is available in Rabiner and Gold's text (Ref 7:187-204). With certain minor modification, this program was used to provide the 1-D prototypes for use in the design of the transformed 2-D filters.

Summary

The Parks-McClellan algorithm provides a very fast method for designing 1-D optimal FIR digital filters. Because this was one of the last remaining major 1-D filtering problems, it is not surprising that research in the past several years has centered on higher-dimensional filtering, especially the two-dimensional case. The insights gained in considering the 1-D optimal design problem prove useful in the discussion of 2-D optimal filter design.

## III. The Design of Optimal 2-D Filters

Techniques for designing 1-D digital filters are now fairly well established in the literature. Very few of these techniques, however, have been extended to two or more dimensions. Since there is no lack of interest in 2-D signal processing, it is worth investigating those design techniques that can be extended to two-dimensions.

Of special interest in this investigation is the design of 2-D optimal filters. In this chapter, the design methods currently available are reviewed, each being evaluated in terms of its optimality, computational efficiency, and ease of understanding. It is shown that no optimal method meets all three criteria. This justifies the selection of a "sub-optimal" design method.

### Basic Concepts of 2-D Discrete Signals

Because 1-D LSI systems represent a special case of multi-dimensional systems, it is not surprising that many of the concepts encountered in 2-D filtering seem familiar. However, certain useful properties are unique to the 1-D case.

Discrete 2-D signals are functions of two integer variables (m,n). A discrete function $\hat{x}(m,n)$ * is not defined unless both m and n assume integer values. In

*The "hat" symbol is used to maintain consistency with the FORTRAN program in Appendix C, where two-dimensional arrays are identified with a "HAT" suffix.

practice, 2-D discrete signals will often be sampled values from a 2-D continuous signal, and appropriate 2-D sampling theorems exist (Ref 14).

As in 1-D, two dimensional LSI systems can be completely specified by their impulse response $\hat{h}(m,n)$. The output $\hat{y}(m,n)$ of a 2-D LSI system is the convolution of the input sequence $\hat{x}(m,n)$ with $\hat{h}(m,n)$:

$$\hat{y}(m,n) = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} \hat{x}(k,\ell)\hat{h}(m-k,\ n-\ell) \tag{12}$$

The complex exponentials $z_1^m$ and $z_2^m$ are the eigenfunctions of LSI systems. Thus,

$$\hat{y}(m,n) = z_1^{\,m}\, z_2^{\,n}\, \hat{H}(z_1,z_2) \tag{13}$$

where $\hat{H}(z_1,z_2)$ is called the system function.

The z-transform of $\hat{x}(m,n)$ is

$$\hat{X}(z_1,z_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \hat{x}(m,n) z_1^{-m}\, z_2^{-n} \tag{14}$$

and the system function can be shown to be the z-transform of the impulse response (Ref 10:231). If the z-transform is evaluated on the surface $z_1 = \exp(j\omega_1)$, $z_2 = \exp(j\omega_2)$, the 2-D Fourier Transform can be evaluated

$$\hat{X}(\omega_1,\omega_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \hat{x}(m,n)\exp(-j(\omega_1 m + \omega_2 n)) \qquad (15)$$

which is periodic over $[-\pi,\pi]$ in both dimensions.

As in 1-D,

$$\hat{Y}(\omega_1,\omega_2) = \hat{X}(\omega_1,\omega_2)\ \hat{H}(\omega_1,\omega_2) \qquad (16)$$

The 2-D discrete Fourier Transform (DFT) is defined to be

$$\hat{X}(k,\ell) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{x}(m,n)\exp(-j(2\pi mk/M + 2\pi n\ell/N)) \qquad (17)$$

As in 1-D, the 2-D DFT has proven to be an extremely useful tool in implementing digital filters, especially when an efficient Fast Fourier Transform (FFT) is used. Although direct convolution can be used to implement filters of low order, it becomes worthwhile to use an FFT for orders of roughly 10x10 sample points or more (Ref 15:405).

The 2-D Design Objective

The normal objective of 2-D filtering is to alter the frequency spectrum of some input function by operating on it with a filter function $\hat{H}(\omega_1,\omega_2)$. The digital filter design problem amounts to the determination of the impulse response coefficients $\hat{h}(m,n)$ which will produce a filter capable of performing the desired operation. Of course, after the filter is designed, it still remains to be implemented.

2-D IIR Filters. For a 2-D IIR filter, the system function will take the form of a ratio of finite-degree polynomials (Ref 10:233)

$$\hat{H}(z_1,z_2) = \frac{\hat{A}(z_1,z_2)}{B(z_1,z_2)} = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \hat{h}(m,n) z_1^{-m} z_2^{-n} \tag{18}$$

where

$$\hat{A}(z_1,z_2) = \sum_{k=0}^{M_1} \sum_{\ell=0}^{N_1} \hat{a}(k,\ell)\, z_1^{-k} z_2^{-\ell} \tag{19}$$

$$\hat{B}(z_1,z_2) = \sum_{k=0}^{M_2} \sum_{\ell=0}^{N_2} \hat{b}(k,\ell)\, z_1^{-k} z_2^{-\ell} \tag{20}$$

then a recursive partial difference equation exists:

$$\hat{y}(m-i,n-j) = \frac{1}{b(i,j)} \sum_{k=0}^{M_1} \sum_{\ell=0}^{N_1} \hat{a}(k,\ell)\hat{x}(m-k,n-\ell)$$

$$- \frac{1}{b(i,j)} \sum_{\substack{k=0 \\ k\neq \ell=0}}^{M_2} \sum_{\ell=0}^{N_2} \hat{b}(k,\ell)\hat{y}(m-k,n-\ell) \tag{21}$$

The order of recursion in Eq (21) is important, for of the many possible schemes of recursing on the variables, only a few will be causal. As a simple illustration, consider the

following partial difference equation:

$$\hat{y}(m,n) = b\hat{y}(m-1,n) + \hat{x}(m,n) \tag{22}$$

where several possible recursions will be considered. In Fig. 6 a few discrete points near the origin are labeled and the boundary condition

$$\hat{y}(m,n) = 0, \quad m \leq 0 \text{ or } n \leq 0 \tag{23}$$

is indicated. Causal relations would be described by any of the following possible recursive paths:

ABC... DEF... GHI...

ADG... BEH... CFI...

A DB GEC ...

However, other paths can be selected that prevent a causal relation:

ABC... ...FED GHI...

AEI... BF... DH...

AB ED GHI FC ...

A 2-D IIR filter can be implemented from a causal partial difference equation, provided the required boundary conditions are known (Ref 16).

While IIR filters offer a much richer set of system functions than do FIR filters, there are disadvantages which make recursive filters unattractive. For any filter,

a prerequisite for stability is that the impulse response be bounded (Ref 16)

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left| \hat{h}(m,n) \right| < \infty \qquad (24)$$

FIR filters are always absolutely summable, due to their finite extent, so they are always stable. IIR filters must be tested for stability.

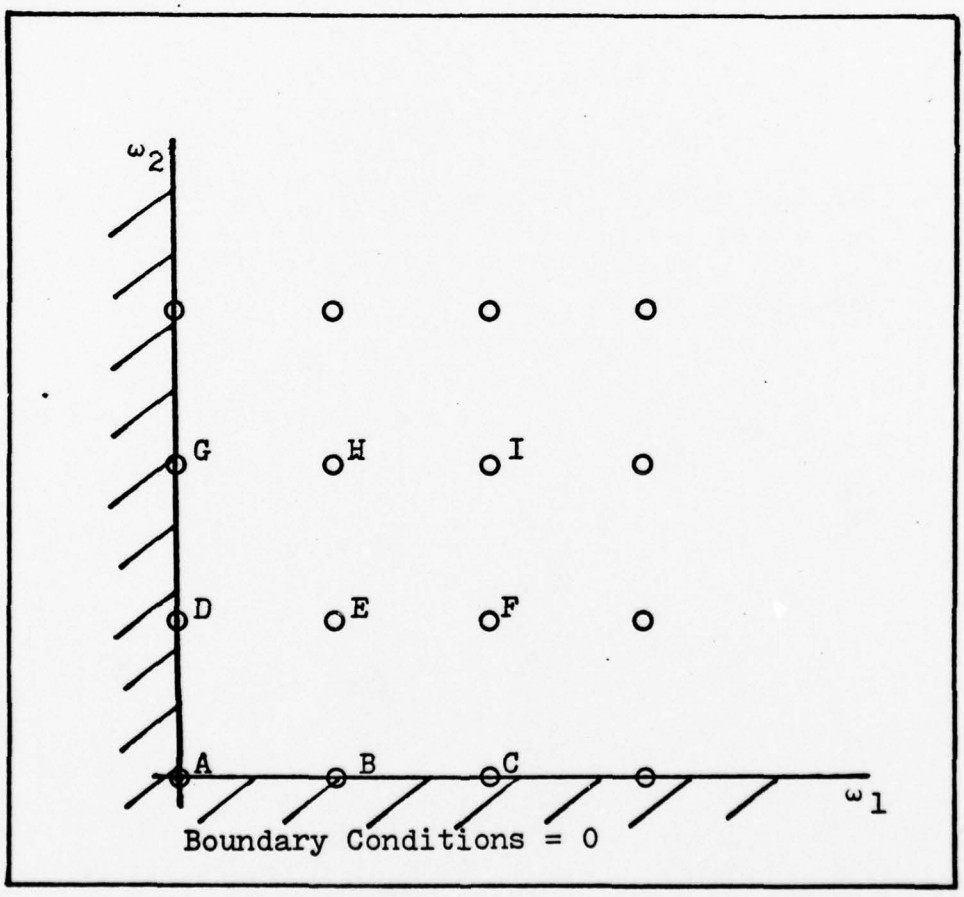



Fig. 6. An Example of Possible 2-D Recursions

In 1-D, an IIR filter of the form

$$\hat{H}(z) = \frac{\hat{A}(z)}{B(z)} \tag{25}$$

can be tested for stability by factoring the functions A(z) and B(z) into cascades of first and second order terms. The Fundamental Theorem of Algebra (Ref 17) guarantees that this can be done. Unfortunately there is no analogous theorem for polynomials of two or more variables. Stability must be tested in a more indirect way.

The most practical IIR stability test is due to Huang (Ref 18). The test can be summarized as follows:

> Huang's Test. A recursive filter $\hat{H}(z_1,z_2)$ is stable if and only if the image of the unit circle $|z_2| = 1$ when mapped into the z plane by $\hat{B}(z_1,z_2) = 0$ does not intersect the region $|z_1| \geq 1$ and, in addition, no point in the region $|z_1| \geq 1$ maps into the point $z_2 = 0$.

Theoretically, this test would require the mapping of the entire $Z_1$-$Z_2$ hyperspace. In practice, only a discrete number of points can be mapped. Improvements on Huang's Test have been published, but the test remains a major computational problem (Ref 10:235).

Even if a stability test is available, more important problems remain: first, how the designer determines the filter coefficients (of a stable filter) to meet a prescribed frequency response specification; second, how an unstable filter can be stabilized without adversely affecting magnitude response.

Complex cepstrum* techniques suffer because they yield infinite degree denominator polynomials which must be approximated by finite degree polynomials, sometimes altering the frequency response significantly (Ref 10:235). A stabilization conjecture due to Shanks (Ref 19), that the double planar least-squares inverse (PLSI) of an unstable filter would yield a stable filter with the desired characteristics, has been used to design many IIR filters. For years, no counter-example appeared. However, very recently, Kamp and Genin provided a counter-example (Ref 20) and then disproved the conjecture (Ref 21).

2-D FIR Filters. Because of these problems with 2-D IIR filters, most 2-D designs have used FIR filters. If the desired 2-D filter either requires or can tolerate circular symmetry, then several design techniques for FIR filters are available. By circular symmetry, it is meant that any slice of the filter that passes thru the origin will be identical (Fig. 7).

The most straight-forward FIR design technique is the direct extension of the windowing method to two-dimensions. Using windows, the infinite-extent 2-D Fourier coefficient $\hat{h}(m,n)$ are multiplied by a finite-extent 2-D window function $\hat{w}(m,n)$ to produce the coefficients

$$\hat{k}(m,n) = \hat{h}(m,n)\cdot\hat{w}(m,n), \quad -M \le m \le M, \quad -N \le n \le N \tag{26}$$

*Defined as the inverse Fourier transform of the complex logarithm of the Fourier transform of a sequence.

The approximated frequency response becomes (Ref 16)

$$\hat{H}(\omega_1,\omega_2) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \hat{k}(m,n)\exp(-j(\omega_1 m + \omega_2 n)) \qquad (27)$$

Huang has shown that "good" 2-D windows can be obtained from "good" 1-D windows by the relation

$$\hat{w}(m,n) = \hat{w}(\sqrt{m^2 + n^2}\,) \qquad (28)$$

where $w(\bullet)$ is a continuous 1-D window (Ref 22). Using windows, circularly symmetric filters with a great variety of shapes can be designed.

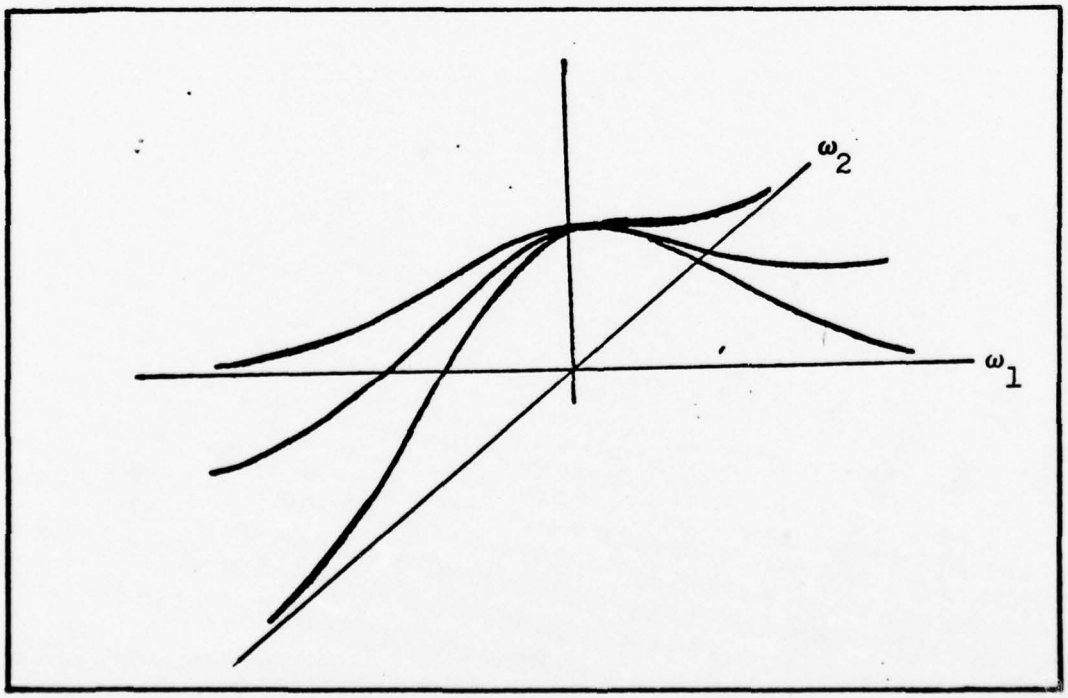



Fig. 7. Slices from a Circularly Symmetric Filter

There are drawbacks to windowing, however. First, in the design of a 1-D prototype window, a closed form expression for the Fourier series coefficients is required:

$$h(n) = \frac{1}{2\pi}\int_0^{2\pi} H(\omega)\exp(j\omega n)d\omega \qquad (29)$$

If this integral proves cumbersome to evaluate, the design process may stop (Ref 7:101). Second, window designed filters cannot meet prescribed frequency cutoff specifications because the convolution process "smears" the frequency response near the ideal response's discontinuities. Although there may exist a "best" window, it would not produce an optimum filter (Ref 7:90).

Rabiner and others have used linear programming to design 2-D FIR filters. This approach can be considered a natural extension of the 1-D frequency sampling technique (Ref 7:105-123), except that all the frequency samples are considered as variables. If $\hat{D}(\omega_1,\omega_2)$ is the desired ideal response, then the constraints

$$\hat{H}(\omega_1,\omega_2) \leq \hat{D}(\omega_1,\omega_2) + \delta \qquad (30)$$

$$\hat{H}(\omega_1,\omega_2) \geq \hat{D}(\omega_1,\omega_2) - \delta \qquad (31)$$

can be applied to a grid of points in the $\omega_1$-$\omega_2$ plane. Linear programming adjusts $\hat{h}(m,n)$ while minimizing $\delta$. Optimal 2-D filters have been designed using linear programming, but the method proved computationally costly.

Rabiner reports that the most ambitious design he attempted (a 9x9 sample point filter with circular symmetry) required more than one hour of execution time on a high speed computer (Ref 7:461-471). Since thousands of constraint equations can be involved in the design of a 2-D optimal filter, linear programming (which is a single-exchange algorithm) would be expected to be an inefficient design method.

Efforts have been made to extend the design method of Parks and McClellan to two-dimensions. McClellan considered the problem, but stopped after realizing that the 2nd Remez algorithm could not be extended to the 2-D case. Recently, in independent efforts, Kamp and Thiran (Ref 8) and Hersey and Mercereau (Ref 9) developed 2-D design techniques similar to the Parks-McClellan approach but based on the 1st Remez Multiple Exchange algorithm (Ref 13). Since a multiple exchange algorithm was used, an impressive speedup in comparison with linear programming was hoped for. However, the speedup was not dramatic, because the bulk of the execution time was spent in the evaluation of the error function, rather than in optimization of parameters (Ref 10:240).

Analysis of these methods reveal that the Chebychev approximation of 2-D FIR filters is fundamentally more difficult and slower than the 1-D case. There are three reasons. First, there is the impossibility that any set of functions defined on a 2-D domain can satisfy the Haar

condition, thus weakening the Alternation Theorem considerably (Ref 9:8). Second, it would not be possible to order the extremal frequencies, as in 1-D, where increasing ordering guarantees that the error sign alternates from point to point. Third, the size of the problem would approach the unmanageable as the number of optimization parameters is increased (Ref 1:30). In the absence of the Haar condition, the 1st Remez algorithm must be used, and convergence may not occur unless suitable perterbations are introduced (Ref 8:262). Lack of a simple alternation in the error complicates the rules for exchanges, at times preventing multiple exchange, the very procedure that makes the Remez algorithms efficient.

As with all the design techniques presented so far, Kamp and Thiran limited their designs to circular symmetry. Hersey and Mercereau improved on this by requiring only octagonal symmetry*. Although they were able to design filters with a more arbitrary frequency response, computational considerations limited their filters to 15x15 sample points or fewer (Ref 9).

Because these methods design truly optimal filters, they have definite value. However, their costly execution time confines their use to relatively small impulse responses. This, coupled with their mathematical sophistication, probably prevents them from being widely accepted.

---

*Symmetry about both axes and the line $\omega_2 = \omega_1$.

## Summary

Until optimal methods become computationally competitive with non-optimal methods, most 2-D filters will probably be designed non-optimally. Fortunately, a new method proposed by McClellan is capable of improving significantly on previous non-optimal methods. It designs filters that are equiripple, although not quite optimal. As such, they have been called "sub-optimal" (Ref 9:1). This new method is discussed in the next chapter.

## IV. The McClellan Transformation

A new method for designing 2-D digital filters, due to McClellan (Ref 1), has been cited as the design method of choice for most applications today (Ref 9:1-2). The essence of the method is that it transforms an optimal 1-D FIR filter into a sub-optimal 2-D FIR filter. These filters are termed "sub-optimal" because they retain the equiripple characteristic of true optimal filters, yet they are inferior to optimal filters when judged very strictly. As mentioned in Chapter I, McClellan has designed circularly symmetric bandpass filters. However, he proposed that his method could be used to design filters with a much less restrictive symmetry (Ref 1:42-43). The important features of the method are that it is not limited to the design of small filters (127x127 point filters have been designed); that it is fast, efficient, and simple to understand; and, surprisingly, in many cases the resulting filters are almost indistinguishable from true optimal filters (Ref 24:93-97). This last point will be considered again in detail in Chapter V.

### Spectral Transformation Mappings

McClellan's method can be considered a spectral transformation mapping. Mitra provides this definition of a spectral transformation (Ref 25:905):

> By a spectral transformation, we mean a complex map that carries a stable rational transfer function into another stable rational transfer function

> exhibiting a different frequency response, at the same time maintaining some desirable characteristics... In general, a spectral transformation must have the following characteristics: 1) it must produce a LSI stable transfer function from an LSI stable transfer function; 2) it must transform a real transfer function into a real transfer function; and 3) it must preserve some basic characteristics of the magnitude response (e.g. the ripple magnitude in the transmission and attenuation domain).

The spectral transformation that McClellan adopted was first stated in Chapter I and is repeated here:

$$\cos\omega = t_1 + t_2\cos\omega_1 + t_3\cos\omega_2 + t_4\cos\omega_1\cdot\cos\omega_2 \qquad (1)$$

where $\omega$, $\omega_1$ and $\omega_2$ are restricted to the interval $[-\pi,\pi]$, as discussed in Chapter II. Eq (1) is referred to as the McClellan Transformation.

The Change of Variables

In Chapter II it was shown that a 1-D linear phase filter of length 2N x 1 has the form

$$H(\omega) = G(\omega)\exp(j(A + B\omega)) \qquad (32)$$

where the term $\exp\ j(A + B\omega)$ contributes only to phase and can be disregarded. (Otherwise it would be carried along unaltered through each transformation to be described.) It followed that

$$G(\omega) = \sum_{n=0}^{N} a(n)\cos\omega n \qquad (33)$$

The change of variables in Eq (1) depends on $G(\omega)$ being expressable by a suitable choice of coefficients $\{b(n)\}$ as the following linear combination of cosine functions:

$$G(\omega) = \sum_{n=0}^{N} a(n)\cos\omega n = \sum_{n=0}^{N} b(n)\cdot(\cos\omega)^n \tag{34}$$

To calculate $\{b(n)\}$, let $x = \cos\omega$. Then

$$T_n(x) = \cos\left[n \cos^{-1}(x)\right] = \cos\omega n \tag{35}$$

where $T_n(x)$ is the nth order Chebychev polynomial of the first kind. Thus,

$$G(\omega) = \sum_{n=0}^{N} a(n)\cos\omega n = \sum_{n=0}^{N} a(n)T_n(x) = \sum_{n=0}^{N} b(n)x^n \tag{36}$$

and by expanding each equation in Eq (36)

$$\begin{aligned} H(\omega) &= a(0)T_0(x) + a(1)T_1(x) + \ldots + a(n)T_n(x) \\ &= b(0)x^0 + b(1)x^1 + \ldots + b(n)x^n \end{aligned} \tag{37}$$

and then using the Chebychev recursion formula

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{38}$$

one can solve for $x^0$, $x^1$, $x^2$,...,$x^n$ (the method to do this is described in Appendix A):

$$\begin{vmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \\ \vdots \\ x^n \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & \cdots \\ .5 & 0 & .5 & 0 & \\ 0 & .75 & 0 & .25 & \\ & \vdots & & & \end{vmatrix} \begin{vmatrix} T_o(x) \\ T_1(x) \\ T_2(x) \\ T_3(x) \\ \vdots \\ T_n(x) \end{vmatrix} \tag{39}$$

or

$$\underline{X} = \underline{U}\ \underline{T} \tag{40}$$

By substituting for $x^0$, $x^1,\ldots,x^n$ and grouping common factors of $T_o(x)$, $T_1(x),\ldots,T_n(x)$ in the right half of Eq (37), one can equate coefficients of $T_o(x)$, $T_1(x),\ldots,T_n(x)$ to yield

$$\begin{vmatrix} a(0) \\ a(1) \\ a(2) \\ a(3) \\ \vdots \\ a(n) \end{vmatrix} = \begin{vmatrix} 1 & 0 & .5 & 0 & \\ 0 & 1 & 0 & .75 & \cdots \\ 0 & 0 & .5 & 0 & \\ 0 & 0 & 0 & .25 & \\ & \vdots & & & \end{vmatrix} \begin{vmatrix} b(0) \\ b(1) \\ b(2) \\ b(3) \\ \vdots \\ b(n) \end{vmatrix} \tag{41}$$

or

$$\underline{A} = \underline{U}^T\ \underline{B} \tag{42}$$

Thus

$$\underline{B} = (\underline{U}^T)^{-1}\ \underline{A} \tag{43}$$

which allows b(0), b(1),..., b(n) to be solved whenever $\underline{U}^T$ is non-singular.

By substituting Eq (1) into the very convenient form of Eq (34),

$$\hat{G}(\omega_1,\omega_2) = \sum_{n=0}^{N} b(n)\left[ t_1 + t_2\cos\omega_1 + t_3\cos\omega_2 + t_4\cos\omega_1\cos\omega_2\right]^n \qquad (44)$$

and completing the expansion, $G(\omega_1,\omega_2)$ can be expressed in this form:

$$\hat{G}(\omega_1,\omega_2) = \sum_{m=0}^{N}\sum_{n=0}^{N} \hat{a}(m,n)\ (\cos\omega_1 m)\ (\cos\omega_2 n) \qquad (45)$$

where the impulse response coefficients are directly related to $\hat{a}(m,n)$ . The methods that take Eq (44) into the form of Eq (45) are involved and will be described in Chapter V.

It is noted that Eq (45) is the desired form for a 2-D FIR zero-phase digital filter (Ref 15:406). This is what makes McClellan's particular spectral transformation so useful. In fact, it is possible to show that there is an entire set of suitable spectral transformations

$$\cos\omega = \sum_{p=0}^{P}\sum_{q=0}^{Q} \hat{t}(p,q)\ (\cos p\omega_1)\ (\cos q\omega_2) \qquad (46)$$

of which Eq (1) is the special case P = 1, Q = 1. Eq (46) has been called the "generalized McClellan Transformation" (Ref 24:20-24). To limit the scope of this investigation, only the case P = 1, Q = 1 has been considered.

The Range of the Mapping

The McClellan Transformation defines a mapping of the interval $[-\pi,\pi]$ of the 1-D frequency axis to the square region $[-\pi,\pi]x[-\pi,\pi]$ of the 2-D frequency plane. The mapping is certainly not one to one, and the range of the mapping may not include the entire square region in $\omega_1$-$\omega_2$. If quadralateral symmetry* is imposed, the problem can be equivalently stated as the mapping $[0,\pi] \rightarrow [0,\pi]\ x[0,\pi]$, since the information in the first quadrant of $\omega_1$-$\omega_2$ is repeated in the other three quadrants.

If $\omega_2$ is a one-valued function of $\omega_1$, then Eq (1) can be solved for $\omega_2$:

$$\omega_2 = \cos^{-1}\left[\frac{\cos\omega - t_1 - t_2\cos\omega_1}{t_3 + t_4\cos\omega_1}\right] \qquad (47)$$

For each fixed value of $\omega_i \in [0,\pi]$ there exists some corresponding contour $C_i$ in the $\omega_1$-$\omega_2$ plane. Along this contour, the magnitude of the 2-D frequency response is identical to the 1-D frequency response at $\omega_i$. By allowing $\omega$ to vary from 0 to $\pi$, a complete family of contours is

*Defined to be symmetry across the $\omega_1$ and $\omega_2$ axes.

generated describing the 2-D frequency response over some portion of the region $[0,\pi]x[0,\pi]$. For example, in the original McClellan transformation, the values $t_1 = -.5$, $t_2 = t_3 = t_4 = 0.5$ produced the circularly symmetric contours shown in Fig. 1. For frequencies below $0.7\pi$, the contours are very close approximations to circles. However, as $\omega \to \pi$, the contours become more squarelike.

This is still a very desirable mapping, because it maintains the desired contour shape over a large area of the 2-D region, and yet there is a correspondence between every point in $[0,\pi] \times [0,\pi]$ and the 1-D interval $[0,\pi]$. Actually, forcing the mapping to have this range will almost always force some distortion in the contour shape. But, if the mapping is not so defined, the resulting frequency response of the 2-D filter will be ill-behaved. This is discussed in detail later in this chapter.

## The Design Strategy

The design strategy and its limitations are now fairly clear. A 1-D filter such as shown in Fig. 5, with transition band frequencies $\omega_p$ and $\omega_s$, is mapped into a 2-D filter filter with transition contours $C_p$ and $C_s$. The magnitude response in the transmission and attenuation domains is carried unaltered from one dimension to two dimensions. If the 1-D prototype filter was an equiripple optimal filter, then the 2-D transformed filter will also be equiripple. However, it will only be optimal along the slice $\omega_2 = 0$.

Everywhere else it will be slightly sub-optimal (Ref 24: 93-97). Because this is usually a small trade-off in comparison with the relative ease and flexibility of design, the McClellan transformation is a very attractive method for designing 2-D FIR filters.

There are limitations on the types of contours that can be mapped using the method. This can be seen by letting $x = \cos \omega$, $u = \cos \omega_1$, and $v = \cos \omega_2$, then Eq (47) becomes

$$v = \frac{x - t_1 - t_2 u}{t_3 + t_4 u} \tag{48}$$

If x is held fixed and a partial differentiation is performed (Ref 1:36-38)

$$\frac{\partial v}{\partial u} = \frac{-t_2 t_3 + t_1 t_4 - t_4 x}{(t_3 + t_4 u)^2} \tag{49}$$

it is seen that v is monotonic, since the sign of the derivative does not change as u varies from -1 to 1. This requires that the contours in $\omega_1$-$\omega_2$ must also be monotonic.

The Contour Approximation Problem

In order to design 2-D filters with arbitrary contour shapes within the admissable class, some method must be developed that maps prescribed 1-D frequencies to a specified 2-D contour. As a first cut at the problem, it is assumed that only one of the 1-D transition band frequencies must be mapped to a particular contour. Hopefully, the contour shapes in the neighborhood of the designated

contour will not be greatly changed. Let the 1-D frequency be $\omega_o$, and the desired contour be $C_o$, where $C_o$ is defined by the single valued relation

$$\omega_2 = F_o(\omega_1) \tag{50}$$

Then for some value of $\underline{t}$

$$\underline{t} = \begin{vmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{vmatrix} \tag{51}$$

$\omega_o$ must be made to map to $C_o$. In general, this is impossible, so a best approximation according to some error criteria is selected. Two of the possible criteria are to minimize the maximum absolute error (minimax criteria) and to minimize the sum of the squared errors (least-squares criteria). Because of the relative ease of implementation, the least-squares criteria was selected. Empirical evidence indicates that the two methods lead to equivalent filters, at least for the case P = 1, Q = 1 in Eq (46) (Ref 2:407-408).

By considering the constant $\cos\omega_o$ to represent a set of identical sample values, the classical least-squares formulation for fitting a set of data points can be used.

Let

$$S(\omega_1) = t_1G_1(\omega_1) + t_2G_2(\omega_1) + t_3G_3(\omega_1) + t_4G_4(\omega_1) \tag{52}$$

where

$$S(\omega_1) = \cos\omega_o = \text{constant} \tag{53}$$

$$G_1(\omega_1) = 1.0 \tag{54}$$

$$G_2(\omega_1) = \cos\omega_1 \tag{55}$$

$$G_3(\omega_1) = \cos( F(\omega_1) ) \tag{56}$$

$$G_4(\omega_1) = \cos(\omega_1) \cos( F_o(\omega_1) ) \tag{57}$$

A requirement that each of the approximating functions (in this case, they are generalized polynomials) be linearly independent is satisfied. The least-squares formulation provides four equations to solve for the four unknowns $\underline{t}$.

Because an explanation of the least-squares formulation is lengthy and might be unnecessary for many readers, the details of this method as applied to the contour approximation problem are outlined in Appendix B. References on the method are included in the bibliography (Ref 27, Ref 28).

Constraints on the Mapping

This much accomplished, the contour approximation problem still remains only half solved. For example, if $\underline{t}$ is free to take on all values, the trivial solution $t_1 = \cos\omega_o$, $t_2 = t_3 = t_4 = 0$ will result in a zero error. This would also result in the frequency $\omega_o$ mapping to the entire 2-D region $[0,\pi]x[0,\pi]$ as well as $C_o$. Evidently, some reasonable constraints must be placed on $\underline{t}$ to reach a useful solution.

A possible set of constraints that seems intuitively satisfactory for the mapping $[0,\pi] \rightarrow [0,\pi] \times [0,\pi]$ might be

$$0 \rightarrow (0,0) \tag{58}$$

$$\pi \rightarrow (\pi,\pi) \tag{59}$$

$$\hat{G}(\omega_1,\omega_2) = \hat{G}(\omega_2,\omega_1) \tag{60}$$

where the second two constraints imply a redundant constraint

$$\pi \rightarrow (\pi, {}_2) \tag{61}$$

Applying the first constraint to the McClellan Transformation yields

$$1 = t_1 + t_2 + t_3 + t_4 \tag{62}$$

while the second constraint yields

$$-1 = t_1 - t_3 + (t_2 - t_4)\cos\omega_1 \tag{63}$$

Because Eq (63) must be true for every $\omega_i \in [0,\pi]$, it is also true that

$$-1 = t_1 - t_3 \tag{64}$$

$$0 = t_2 - t_4 \tag{65}$$

Finally, the constraint of Eq (60) requires that

$$t_2 = t_3 \tag{66}$$

as seen by inspection of the symmetry of Eq (1). Solving Eqs (62)-(66) for $\underline{t}$ yields $t_1 = -.5$, $t_2 = t_3 = t_4 = 0.5$,

which is recognized as the original McClellan transformation for circular symmetry.

A problem with this set of constraints is that it leaves no free variables. To design arbitrary contours, a less restrictive set of constraints is needed. Mitra's paper on spectral transformations provides two theorems that are helpful (Ref 25:910), but a superior technique has been introduced by Mercereau et al (Ref 15:408-409) and is recommended for most contour designs. Their suggestion is to initially perform the contour approximation subject to a single constraint, $0 \rightarrow (0,0)$. This produces a mapping to an intermediate domain $(\Omega_1,\Omega_2)$. A second mapping will produce the final result: a mapping in the $(\omega_1,\omega_2)$ plane that approximates the desired contour $C_o$ and has a range that includes the entire region $[0,\pi]x[0,\pi]$.

As an example, if the elliptical contour

$$\omega_2 = \omega_b \sqrt{1 - (\omega_1/\omega_a)^2} \tag{67}$$

where

$$\omega_b = .50\pi \tag{68}$$

$$\omega_a = .25\pi \tag{69}$$

$$\omega_o = .50\pi \tag{70}$$

is desired, the least-square formulation subject to constraint (explained in Appendix B) yields

$$t_1 = -2.584 \tag{71}$$

$$t_2 = 2.586 \tag{72}$$

$$t_3 = 0.166 \tag{73}$$

$$t_4 = 0.832 \tag{74}$$

with a mean error of 0.0003. However, the range of the contour mapping is inadequate, as shown in Fig. 8. The significance of the cross-hatched area can be understood by considering Eq (1). To stay within the realm of real numbers, the following double inequality must hold:

$$-1.0 \leq \cos\omega \leq 1.0 \tag{75}$$

or equivalently

$$-1.0 \leq \hat{F}(\Omega_1,\Omega_2) \leq 1.0 \tag{76}$$

$$-1.0 \leq t_1 + t_2\cos\Omega_1 + t_3\cos\Omega_2 + t_4\cos\Omega_1\cos\Omega_2 \leq 1.0 \tag{77}$$

The cross-hatched area of Fig. 8 corresponds to values of $(\Omega_1,\Omega_2)$ where Eq (77) does not hold. The magnitude of the frequency response grows rapidly, since it corresponds to complex values of $\omega$ (Ref 15:408).

Two of the basic operations of complex variable mapping theory are translation and expansion (or contraction). These operations do not alter the shape of a given contour,

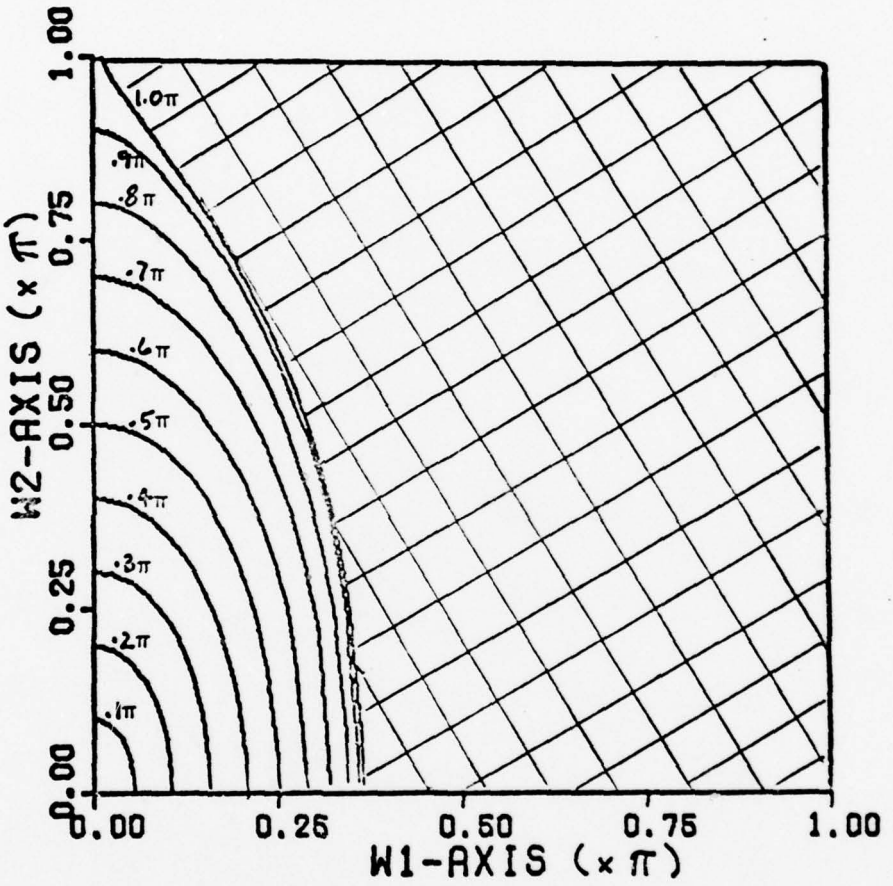



Fig. 8. Elliptical Contours of Eq (67). $\omega_o = 0.5\pi$
$\underline{t}^T = [-2.584, 2.586, 0.166, 0.832]$, $\bar{e} = 0.0003$,
Constraint: $0 \rightarrow (0,0)$

only its orientation and relative size (Ref 17:73). The following mapping should be considered:

$$\cos\omega_o' = \hat{F}'(\omega_1,\omega_2) = k_1\hat{F}(\Omega_1,\Omega_2) - k_2, \quad k_1 > 0 \tag{78}$$

The usefulness of this second mapping stems from consideration of Eq (77) and Fig. 8. Because the filter is either lowpass or highpass, the only contours of importance are $C_o$ and its immediate neighbors, since they control the mapping of the transition band. It is desired that the range of the second mapping include the entire square region, for then Eq (77) would be satisfied. Due to the nature of the second mapping, the contour shapes themselves will be unaffected; however, a new 1-D frequency will be associated with each contour in $\omega_1$-$\omega_2$, the ultimate mapping domain.

To find the necessary values of $k_1$ and $k_2$ in Eq (78), one applies the second mapping to Eq (76) and specifies the constraint of Eq (75):

$$-1.0 \le k_1\hat{F}(\Omega_1,\Omega_2) - k_2 \le 1.0 \tag{79}$$

Solving each inequality separately for its worst case produces two equations in $k_1$ and $k_2$:

$$-1.0 = k_1\hat{F}_{min}(\Omega_1,\Omega_2) - k_2 \tag{80}$$

$$1.0 = k_1\hat{F}_{max}(\Omega_1,\Omega_2) - k_2 \tag{81}$$

where $\hat{F}_{min}(\Omega_1,\Omega_2)$ and $\hat{F}_{max}(\Omega_1,\Omega_2)$ correspond to the least

and greatest magnitudes taken by $F(\Omega_1, \Omega_2)$ over a dense grid of points in $[0,\pi]x[0,\pi]$. By solving Eq (80) and Eq (81) one finds that

$$k_1 = \frac{2}{F_{max}(\Omega_1,\Omega_2) - F_{min}(\Omega_1,\Omega_2)} \tag{82}$$

$$k_2 = k_1 F_{max}(\Omega_1,\Omega_2) - 1 \tag{83}$$

The new 1-D frequency associated with $C_o$ is found by solving Eq (78):

$$\omega_o' = \cos^{-1}(k_1 \cos\omega_o - k_2) \tag{84}$$

Because

$$F'(\omega_1,\omega_2) = t_1' + t_2'\cos\omega_1 + t_3'\cos\omega_2 + t_4'\cos\omega_1\cos\omega_2 \tag{85}$$

a new "scaled" $\underline{t}'$ also results:

$$t_1' = k_1 t_1 - k_2 \tag{86}$$

$$t_2' = k_1 t_2 \tag{87}$$

$$t_3' = k_1 t_3 \tag{88}$$

$$t_4' = k_1 t_4 \tag{89}$$

Using the $\underline{t}'$, a new contour mapping is shown in Fig. 9. It is noted that the shapes of the contours remain unchanged up to the original contour $C_\pi$. In $\omega_1$-$\omega_2$ the contour $C_\pi$ now corresponds to

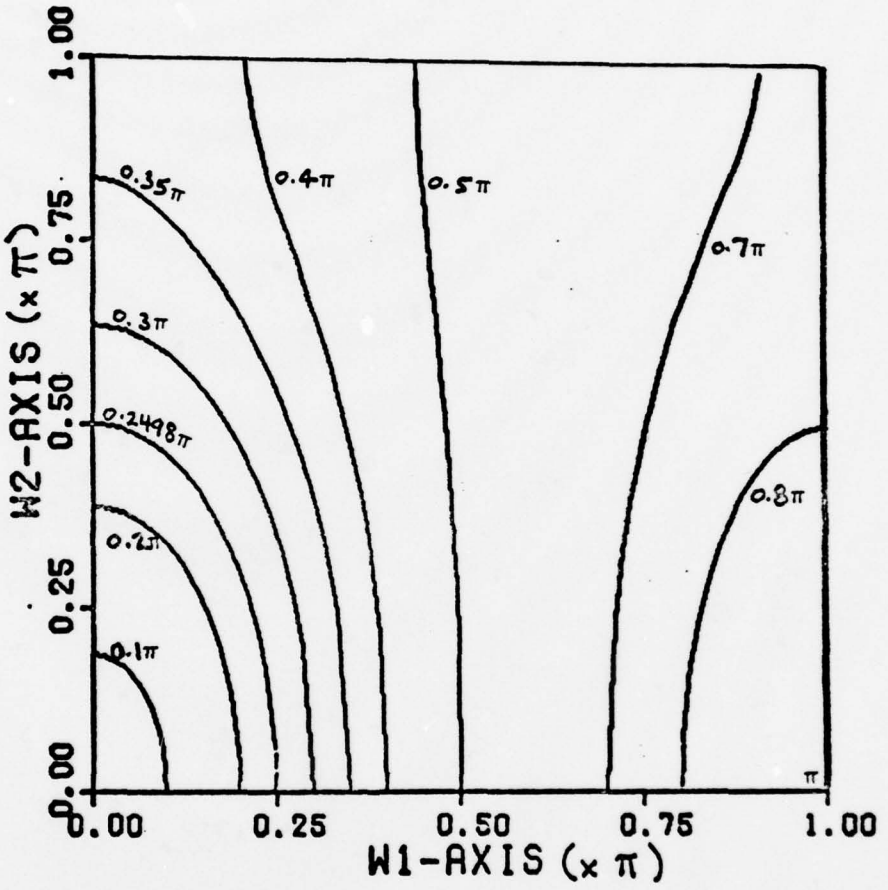



Fig. 9. Elliptical Contours of Eq (67) after scaling.
$\omega_o' = 0.2498\pi$, $\underline{t}^T = [-.0482, .7559, .0482, .2441]$,
$\bar{e} = .0002$, constraint: $0 \rightarrow (0,0)$ with scaling

$$\omega_o' = \cos^{-1}(k_1 \cos\pi - k_2) = \cos^{-1}(-k_1 - k_2) \qquad (90)$$

The remaining contours spread out over $[0,\pi]x[0,\pi]$, giving the final mapping the desired range. Thus, the final frequency response is well-defined.

As mentioned, several theorems attributed to Sanjit Mitra can also be helpful in assuring a well-defined frequency response. Because the development of these theorems is extensive, interested readers are referred to Mitra's paper (Ref 25). In many cases, the resulting constraints will produce in the same final mapping as in the previous section, but this is not always so. To leave the design program as flexible as possible, either method of specifying constraints is permitted. The technique of Mercereau et al is available as a default option; however, an enlightened user may override the default and supply any consistent set of constraints.

Summary

The McClellan Transformation is a spectral transformation that maps 1-D frequencies in the interval $[0,\pi]$ to contours in the $(\omega_1,\omega_2)$-plane over $[0,\pi]x[0,\pi]$. Because the filter is assumed to have quadrilateral symmetry, the area $[0,\pi]x[0,\pi]$ actually defines the total behavior over $[-\pi,\pi]x[-\pi,\pi]$. Because the mapping is generally imperfect, an error criteria is established to determine the best mapping. To insure that the mapping is non-trivial, constraints on $\underline{t}$ are necessary. To insure that the range of

the mapping includes the entire region $[0,\pi]x[0,\pi]$, the technique of Mercereau et al can be used. If the equivalence of Eqs(44) and (45) can be established, the filter design is complete.

## V. Determination of the 2-D Impulse Response Coefficients

As shown in the last chapter, $\underline{t}$ can be found that provides a suitable contour mapping. Thus, the frequency response can be evaluated as a function of $(\omega_1,\omega_2)$

$$\hat{H}(\omega_1,\omega_2) = \sum_{n=0}^{N} b(n)(t_1+t_2\cos\omega_1+t_3\cos\omega_2+t_4\cos\omega_1\cos\omega_2)^n \qquad (44)$$

where $\omega_2 = F(\omega_1)$. This itself is not enough, since it is necessary to convert Eq (44) into a form that relates directly to the impulse response coefficients $\{\hat{h}(m,n)\}$. They will be needed if the z-transform $\hat{H}(z_1,z_2)$ will be used when the filter is implemented.

In this chapter, a method is derived that transforms Eq (44) into the form of a zero-phase 2-D FIR filter. From this form, the impulse coefficients can be determined by inspection. The chapter concludes with some comments on the "optimality" of the McClellan Transformation.

### Algorithm for the Expansion

Let $x = \cos\omega_1$ and $y = \cos\omega_2$, then

$$\hat{H}(x,y) = \sum_{n=0}^{N} b(n)(t_1+t_2x+t_3y+t_4xy)^n = \sum_{n=0}^{N} b(n)[\bullet]^n \qquad (91)$$

The evaluation of Eq (91) can be accomplished by considering the following steps:

$$\hat{H}_o(x,y) = b(0) \tag{92}$$

$$\hat{H}_1(x,y) = \hat{H}_o(x,y) + b(1)[\bullet] \tag{93}$$

$$\hat{H}_2(x,y) = \hat{H}_1(x,y) + b(2)[\bullet][\bullet] \tag{94}$$

$$\hat{H}_3(x,y) = \hat{H}_2(x,y) + b(3)[\bullet][\bullet]^2 \tag{95}$$

$$\vdots$$

$$\hat{H}_k(x,y) = \hat{H}_{k-1}(x,y) + b(k)[\bullet][\bullet]^{k-1} \tag{96}$$

Because the quantity $[t_1 + t_2x + t_3y + t_4xy]^{k-1}$ is already known, the process is recursive. As the order of N gets large, the memory required for the expansion grows very quickly. However, if, after each step of the expansion, all of the coefficients are searched for like terms, the memory requirements are held down considerably. Empirically, it was found that the expression

$$[t_1 + t_2x + t_3y + t_4xy][t_1 + t_2x + t_3y + t_4xy]^{k-1} \tag{97}$$

would require $(k + 1)^2$ words of memory for coefficient storage, provided like terms were combined. If the search for like terms was delayed until the $(N + 1)$th step, the storage required would grow as $4^N$. Table I compares the storage required with the two possible approaches. Obviously, searching for like terms after each step of the expansion is essential except for very low order filter designs. The final result of the expansion is

$$\hat{H}(\omega_1,\omega_2) = \sum_{m=0}^{N}\sum_{n=0}^{N}\hat{b}(m,n)\ (\cos\omega_1)^m\ (\cos\omega_2)^n \qquad (98)$$

which involves every possible combination of $(\cos\omega_1)^m(\cos\omega_2)^n$ up to m = N, n = N.

An Analogous 1-D Result

Eq (98) itself still does not present a direct method of obtaining the impulse response coefficients. Before progressing, it is instructive to consider the analogous 1-D equation

$$\hat{H}(\omega) = \sum_{n=0}^{N} b(n)\ (\cos\omega)^n = \sum_{n=0}^{N} a(n)\cos\omega n \qquad (99)$$

where the coefficients a(n) must be determined. Using the dual of the argument that led to Eq (43) in Chapter IV, let x = cosω then $T_n(x) = \cos\omega n$, and

Table I
STORAGE REQUIRED FOR TWO METHODS OF EXPANDING EQ (91)

| k | $(k+1)^2$ | $4^k$ |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 4 | 4 |
| 2 | 9 | 16 |
| 3 | 16 | 64 |
| . | . | . |
| . | . | . |
| . | . | . |
| 10 | 121 | 1,048,576 |
| . | . | . |
| . | . | . |
| . | . | . |

$$H(\omega) = \sum_{n=0}^{N} b(n)\,(\cos\omega)^n = \sum_{n=0}^{N} b(n)x^n = \sum_{n=0}^{N} a(n)T_n(x) \qquad (100)$$

Using Eq (38), the relationship between $T_o(x)$, $T_1(x)$,..., $T_n(x)$ and $x^0$, $x^1$,...,$x^n$ can be shown to be

$$\begin{vmatrix} T_o(x) \\ T_1(x) \\ T_2(x) \\ T_3(x) \\ \vdots \\ T_n(x) \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & \cdots \\ -1 & 0 & 2 & 0 & \\ 0 & -3 & 0 & 4 & \\ & \vdots & & & \end{vmatrix} \begin{vmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \\ \vdots \\ x^n \end{vmatrix} \qquad (101)$$

or

$$\underline{T} = \underline{V}\,\underline{X} \qquad (102)$$

By substituting for $T_o(x)$, $T_1(x)$,...,$T_n(x)$ and grouping common factors of $x^0$, $x^1$,...,$x^n$ in the right half of Eq (100), one can equate the coefficients of $x^0$, $x^1$,...,$x^n$ to yield

$$\begin{vmatrix} b(0) \\ b(1) \\ b(2) \\ b(3) \\ \vdots \\ b(n) \end{vmatrix} = \begin{vmatrix} 1 & 0 & -1 & 0 & \\ 0 & 1 & 0 & -3 & \cdots \\ 0 & 0 & 2 & 0 & \\ 0 & 0 & 0 & 4 & \\ & \vdots & & & \end{vmatrix} \begin{vmatrix} a(0) \\ a(1) \\ a(2) \\ a(3) \\ \vdots \\ a(n) \end{vmatrix} \qquad (103)$$

or

$$\underline{B} = \underline{V}^T \underline{A} \tag{104}$$

Thus

$$\underline{A} = (\underline{V}^T)^{-1} \underline{B} \tag{105}$$

which allows $\{a(n)\}$ to be solved whenever $\underline{V}^T$ is non-singular.

Extension to Two-Dimensions

This 1-D method for calculating $\{a(n)\}$ can be exploited to calculate $\{\hat{a}(m,n)\}$. Eq (98) is rewritten as

$$\hat{H}(x,y) = \sum_{m=0}^{N} \sum_{n=0}^{N} \hat{b}(m,n) x^m y^n \tag{106}$$

where

$$x = \cos\omega_1 \tag{107}$$

$$y = \cos\omega_2 \tag{108}$$

$$T_m(x) = \cos\omega_1 m \tag{109}$$

$$T_n(y) = \cos\omega_2 n \tag{110}$$

Thus,

$$\hat{H}(x,y) = \sum_{m=0}^{N} x^m \left[ \sum_{n=0}^{N} b(\bar{m},n) y^n \right] \tag{111}$$

where $\bar{m}$ denotes that m is fixed. By expanding Eq (111)

$$\hat{H}(x,y) = x^0 \sum_{n=0}^{N} \hat{b}(\bar{0},n)y^n + x^1 \sum_{n=0}^{N} \hat{b}(\bar{1},n)y^n$$

$$+ \ldots + x^N \sum_{n=0}^{N} \hat{b}(\bar{N},n)y^n \qquad (112)$$

where each individual summation now has the form of Eq (99). One can use the relation defined by Eq (105)*.

$$\hat{H}(x,y) = x^0 \sum_{n=0}^{N} \overset{*}{a}(\bar{0},n)T_n(y) + \ldots + x^N \sum_{n=0}^{N} \overset{*}{a}(\bar{N},n)T_n(y) \qquad (113)$$

$$= \sum_{m=0}^{N} x^m \cdot \sum_{n=0}^{N} \overset{*}{a}(\bar{m},n)T_n(y) \qquad (114)$$

$$= \sum_{m=0}^{N} x^m \cdot \sum_{n=0}^{N} \overset{*}{a}(\bar{m},n)\cos\omega_2 n \qquad (115)$$

$$= \sum_{n=0}^{N} \cos\omega_2 n \left[ \sum_{m=0}^{N} \overset{*}{a}(m,\bar{n})x^m \right] \qquad (116)$$

where the term in brackets in Eq (116) is also of the same form as Eq (99). Repeating the same sequence of steps yields

$$H(x,\omega_2) = \sum_{n=0}^{N} \cos\omega_2 n \cdot \sum_{m=0}^{N} \hat{a}(m,\bar{n})T_m(x) \qquad (117)$$

$$= \sum_{n=0}^{N} \cos\omega_2 n \cdot \sum_{m=0}^{N} \hat{a}(m,\bar{n})\cos\omega_1 m \qquad (118)$$

---

*The notation $\overset{*}{a}(m,n)$ has no mathematical significance. It is used only to emphasize the distinction from $\hat{a}(m,n)$.

$$\hat{H}(\omega_1,\omega_2) = \sum_{m=0}^{N}\sum_{n=0}^{N} \hat{a}(m,n)\cos\omega_1 m \cdot \cos\omega_2 n \qquad (119)$$

which is the desired final form.

As in Chapter II, the assumption is made that the impulse response is centered at the origin. One can now compute the impulse response coefficients $\{\hat{h}(m,n)\}$ as follows:

$$\hat{H}(\omega_1,\omega_2) = \hat{a}(0,0) + \sum_{m=1}^{N} \hat{a}(m,\bar{0})\cos\omega_1 m + \sum_{n=1}^{N} \hat{a}(\bar{0},n)\cos\omega_2 n$$

$$+ \sum_{m=1}^{N}\sum_{n=1}^{N} \hat{a}(m,n)\cos\omega_1 m \cos\omega_2 n \qquad (120)$$

The first term in Eq (120) is the impulse response at (0,0). The second and third terms can be recognized as of the same form as Eq (7) of Chapter II. The fourth term must be further decomposed:

$$\sum_{m=1}^{N}\sum_{n=1}^{N} \hat{a}(m,n)\cos\omega_1 m \cos\omega_2 n = \sum_{m=1}^{N} \cos\omega_1 m \sum_{n=1}^{N} \hat{a}(\bar{m},n)\cos\omega_2 n \qquad (121)$$

$$= \sum_{m=1}^{N} \cos\omega_1 m \sum_{n=1}^{N} 2\overset{*}{h}(\bar{m},n)\cos\omega_2 n \qquad (122)$$

$$= 2\cos\omega_1 \sum_{n=1}^{N} \overset{*}{h}(\bar{1},n)\cos\omega_2 n + \ldots$$

$$+ 2\cos(N\omega_1) \sum_{n=1}^{N} \overset{*}{h}(\bar{N},n)\cos\omega_2 n \qquad (123)$$

$$= 2\cos\omega_1 \sum_{n=1}^{N} 2\hat{h}(I,n)\cos\omega_2 n + \ldots$$

$$+ 2\cos(N\omega_1) \sum_{n=1}^{N} 2\hat{h}(\bar{N},n)\cos\omega_2 n \qquad (124)$$

$$= \sum_{m=1}^{N}\sum_{n=1}^{N} 4\hat{h}(m,n)\cos\omega_1 m \cos\omega_2 n \qquad (125)$$

Thus, knowing $\{\hat{a}(m,n)\}$, one can easily compute $\{\hat{h}(m,n)\}$:

$$\hat{h}(0,0) = \hat{a}(0,0), \quad m=0,\ n=0 \qquad (126)$$

$$\hat{h}(m,0) = 1/2\ \hat{a}(m,0), \quad m\neq 0,\ n=0 \qquad (127)$$

$$\hat{h}(0,n) = 1/2\ \hat{a}(0,n), \quad m=0,\ n\neq 0 \qquad (128)$$

$$\hat{h}(m,n) = 1/4\ \hat{a}(m,n), \quad m\neq 0,\ n\neq 0 \qquad (129)$$

Comparison of Optimality

It is sometimes claimed that the filters designed by the McClellan Transformation are truly optimal. McClellan made this claim for his original circularly symmetric filters and later extended the claim to include "fan" filters (Fig.18). McClellan argued that along any slice of the 2-D filter, the approximation problem degenerates into a 1-D problem. In the case of circular symmetry with $t_1 = .5$, $t_2 = t_3 = t_4 = 0.5$, one considers the slice $\omega_2 = 0$. The 2-D frequency response then becomes a 1-D frequency response in $\omega_1$:

$$\hat{F}(\omega_1,0) = -.5 + .5\cos\omega_1 + .5 + .5\cos\omega_1 \qquad (130)$$

$$F(\omega_1) = \cos\omega_1 \qquad (131)$$

Since this is exactly the same function that was used to generate the optimal 1-D prototype, the 2-D filter is claimed to also be optimal (Ref 1:45-46). Rabiner and Gold have supported this contention (Ref 7:477-478); however, several other filtering authorities continue to refer to the McClellan Transformation as sub-optimal.

Ideally, empirical tests should be conducted to measure McClellan Transformation filters against known optimal filters. However, the design algorithm presented in this investigation would require some modifications. For a fair comparison, both of the 2-D transition band edges should best approximate the desired contour shape (as is done in true optimal methods).

Table II records data obtained on the design of several circularly symmetric optimal lowpass filters designed by Harris and Mercereau (Ref 9:II.2). Each has a passband radius of $0.4\pi$ and a stopband radius of $0.6\pi$. Using the original McClellan transformation, the 0.4 radius corresponded to a 1-D frequency of $0.4000\pi$, while the $0.6\pi$ radius corresponded to a 1-D frequency of $0.5760\pi$ (recall that the contours become non-circular as $\omega \to \pi$). From Table II, one notes that the McClellan error is somewhat larger than the optimal error for each case. If the McClellan Transformation is used with 1-D frequencies of $0.4\pi$ and $0.6\pi$, the error naturally decreases.

Very surprisingly, it actually decreases to less than the optimal error. However, this second case is not a fair comparison: it would be better to design the optimal filters with respect to the contours resulting from this particular McClellan Transformation and then compare the errors. Time has not permitted such an experiment in this investigation.

The data available suggests that the McClellan Transformation is sub-optimal, even for circularly symmetric filters. However, the absolute difference between the optimal error and the McClellan error is very slight. Because of the general design flexibility and significant savings in computation time, the McClellan Transformation compares very favorably with optimal methods for designing 2-D FIR filters.

TABLE II

COMPARISON OF THE ERROR IN OPTIMAL AND MCCLELLAN FILTERS

| PASSBAND | STOPBAND | METHOD | FILTER ORDER | ERROR ($\delta$) | DESIGN TIME (MIN:SEC) |
|---|---|---|---|---|---|
| $.4\pi$ | $.6\pi$ | KAMP-THIRAN | 5x5 | 0.26705 | 1:29[1] |
| $.4\pi$ | $.6\pi$ | HERSEY-MERCEREAU | 5x5 | 0.26705 | 1:19[1] |
| $.4\pi$ | $.576\pi$ | MCCLELLAN | 5x5 | 0.2852 | 0:01[2] |
| $.4\pi$ | $.6\pi$ | MCCLELLAN | 5x5 | 0.2639 | 0:01[2] |
| $.4\pi$ | $.6\pi$ | KAMP-THIRAN | 9x9 | 0.11417 | 19:22[1] |
| $.4\pi$ | $.6\pi$ | HERSEY-MERCEREAU | 9x9 | 0.11417 | 14:39[1] |
| $.4\pi$ | $.576\pi$ | MCCLELLAN | 9x9 | 0.1334 | 0:01[2] |
| $.4\pi$ | $.6\pi$ | MCCLELLAN | 9x9 | 0.1129 | 0:01[2] |

[1] On a PDP 11/50 Computer

[2] On a CDC 6600 Computer

## VI. Design Results

This chapter presents the results from several typical 2-D filter design problems using the computer program developed in this investigation. In the first tests, simple, familiar filters were designed to validate the design program. Gradually, filters with more novel specifications were designed to explore the limits of the method.

### Circularly Symmetric Filters

To test the design program against a known result, circularly symmetric filters were designed. By overriding the default constraint and inputting Eqs (62)-(66), the original McClellan Transformation was duplicated. The associated contour mapping and frequency response for a 21x21 point filter were shown in Figs. 1 and 2, Chapter I. Fig. 10 illustrates the periodic nature of 2-D digital filters.

Circular symmetry was also obtained using the default constraint. Before scaling, the range of the mapping was inadequate (Fig. 11). The scaling algorithm was used to spread out the contours, and the resulting mapping is shown in Fig. 12. Comparing Figs. 1 and 12, it is seen that the contours are essentially the same for values of $\omega$ up to about $0.6\pi$. (Extremely close comparison reveals that the contours

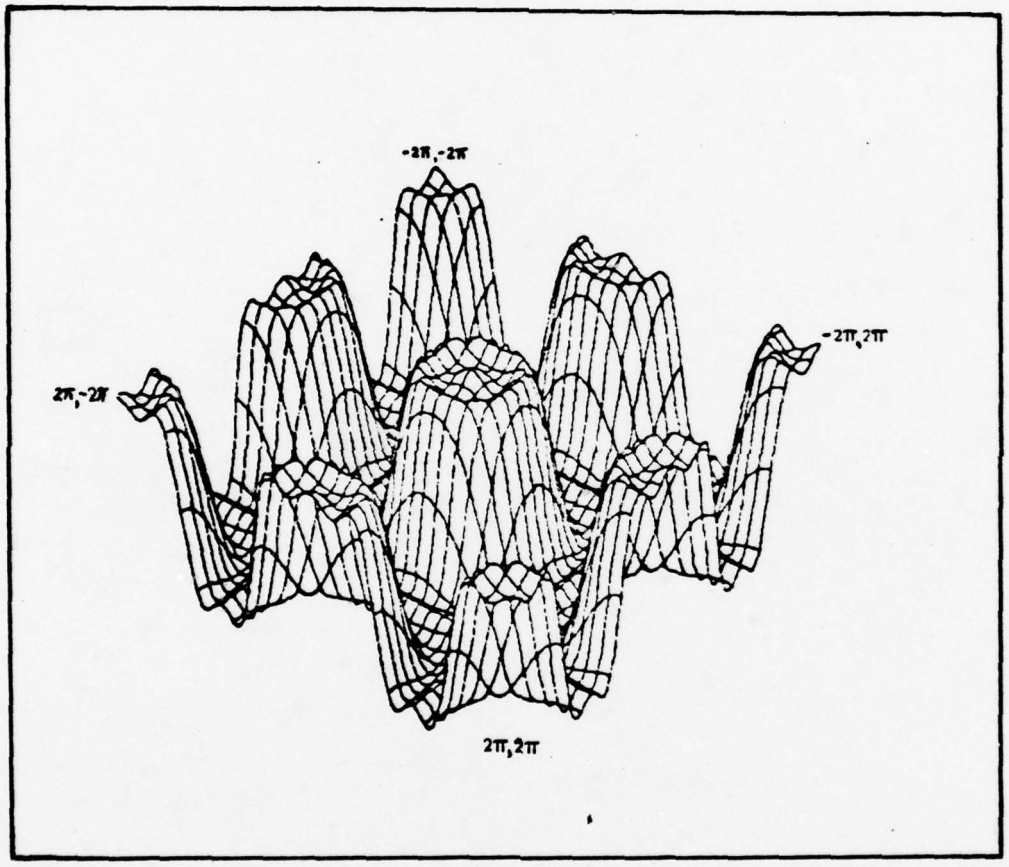



Fig. 10. Periodic Nature of a 2-D Digital Filter

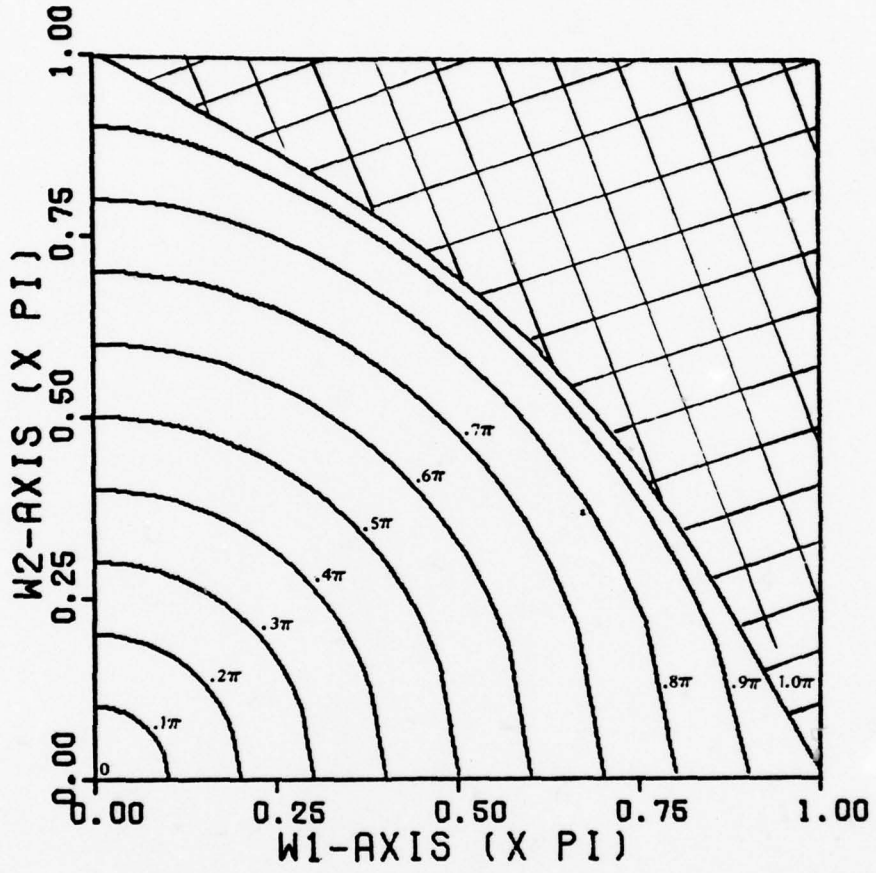



Fig. 11. Circular Contours Using the Default Constraint. $\omega_o = .5\pi$, $\underline{t}^T = [-.6377, -.6377, .6377, .3622]$, $\bar{e} = .00003$

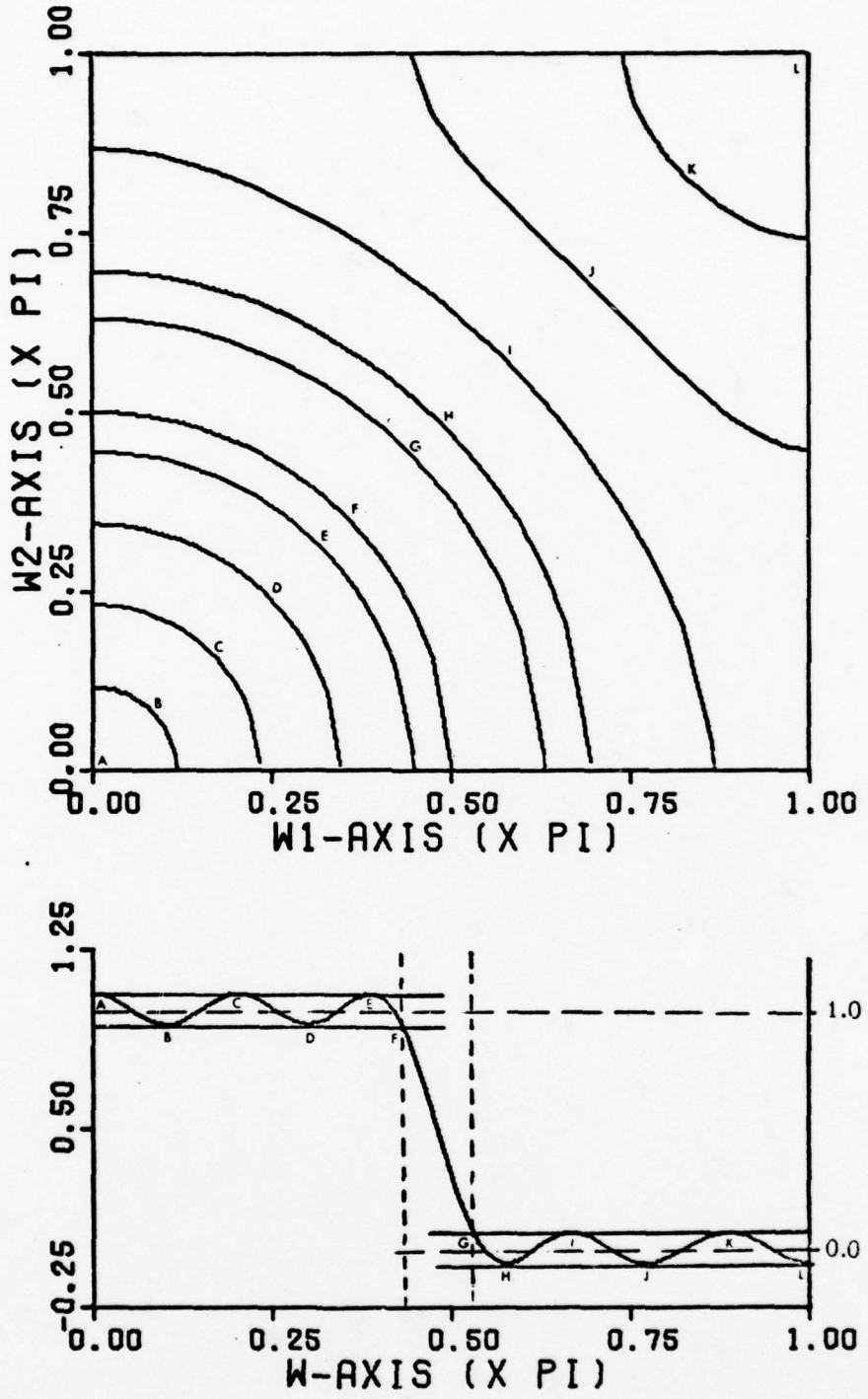



Fig. 12. Circular Contours After Scaling. $\omega_o = .431\pi$
$\underline{t}^T = [-.2840, .5000, .5000, .2840]$, $\bar{e} = .00003$.
Default constraint with scaling. Prototype:
N = 21, $\omega_p = .431\pi$, $\omega_s = .53\pi$, Wt = 1:1, $\delta = .064$

of Fig. 12 are actually more circular.) The contours spread out differently as $\omega \to \pi$. Essentially identical frequency responses can be achieved, as shown in Fig. 13. However, to obtain the same filter specifications, it was necessary to use 1-D prototypes of different order. Because the original filter was designed with a lower order prototype, the original is judged superior.

As an incidental point, note that the information gained from the frequency response drawings is also available from contour mappings of the extremal frequencies of the 1-D prototype. These contours precisely define the ripple in the 2-D filter.

Other smooth shapes successfully designed included elliptical shapes (Figs. 14 and 15), parabolic shapes (Figs. 16 and 17), and hyperbolic shapes (Figs. 31 and 32 of Appendix D).

Fan Filters

Fan filters are used in seismic analysis to filter unwanted velocity components (Ref 26). An ideal fan filter is shown in Fig. 18. McClellan developed four constraint equations and was able to specify $\underline{t}$ as

$$\underline{t}^T = \left[0.0,\ 0.5,\ -0.5,\ 0.0\right] \tag{132}$$

Fan filters were designed using the default constraint, however the resulting value of $\underline{t}$

$$\underline{t}^T = \left[0.0003,\ -0.4992,\ 0.5008,\ -0.0003\right] \tag{133}$$

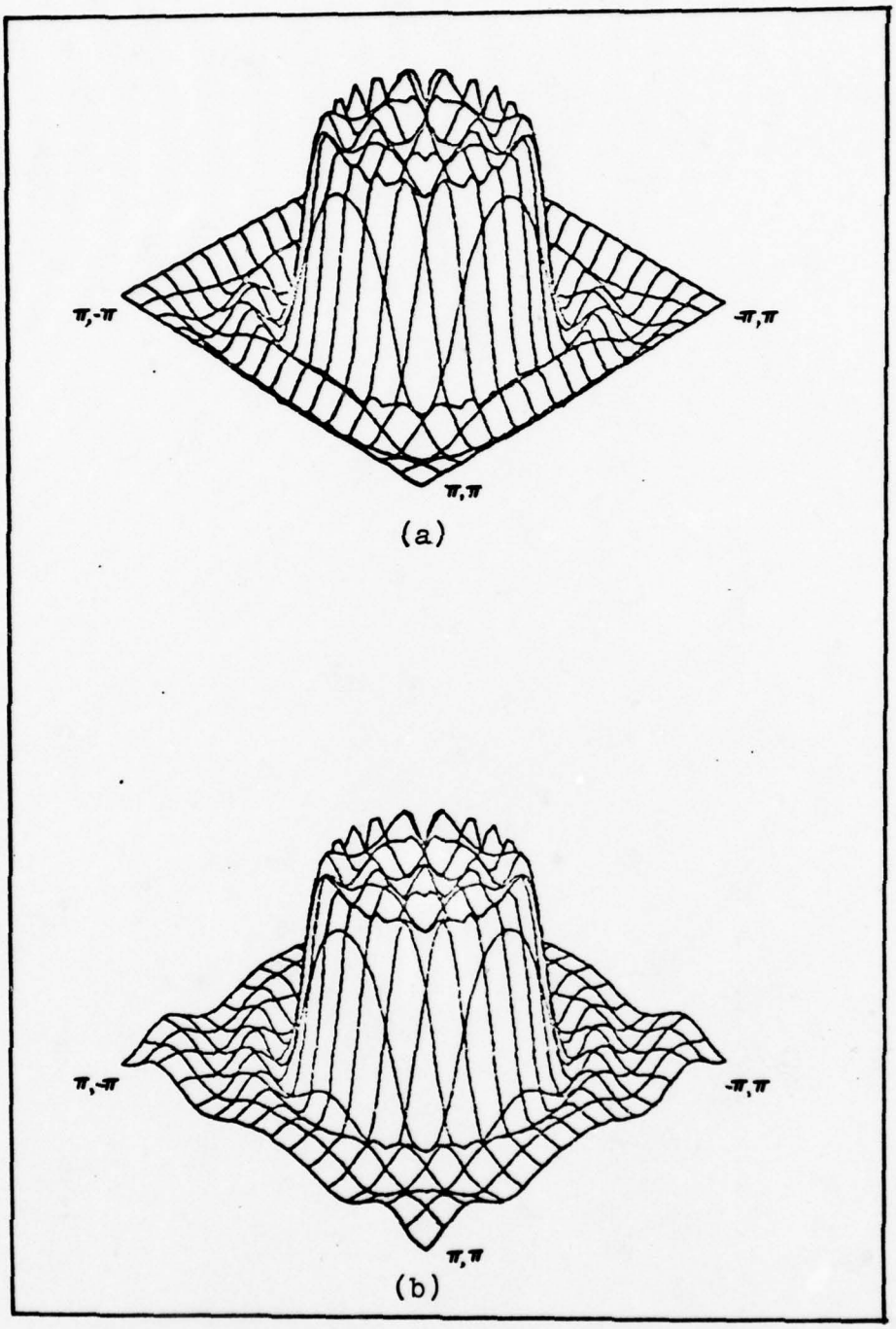



Fig. 13. Comparison of Circularly Symmetric Filters. (a) $N = 19$, $\omega_p = .5\pi$, $\omega_s = .6\pi$, Wt = 1:1, $\delta = .081$, $\underline{t}^T = [-.5, .5, .5, .5]$, $\bar{e} = .00000$. (b) $N = 21$, $\omega_p = .431\pi$, $\omega_s = .53\pi$, Wt = 1:1, $\delta = .065$, $\underline{t}^T = [-.284, .5, .5, .284]$, $\bar{e} = .00003$

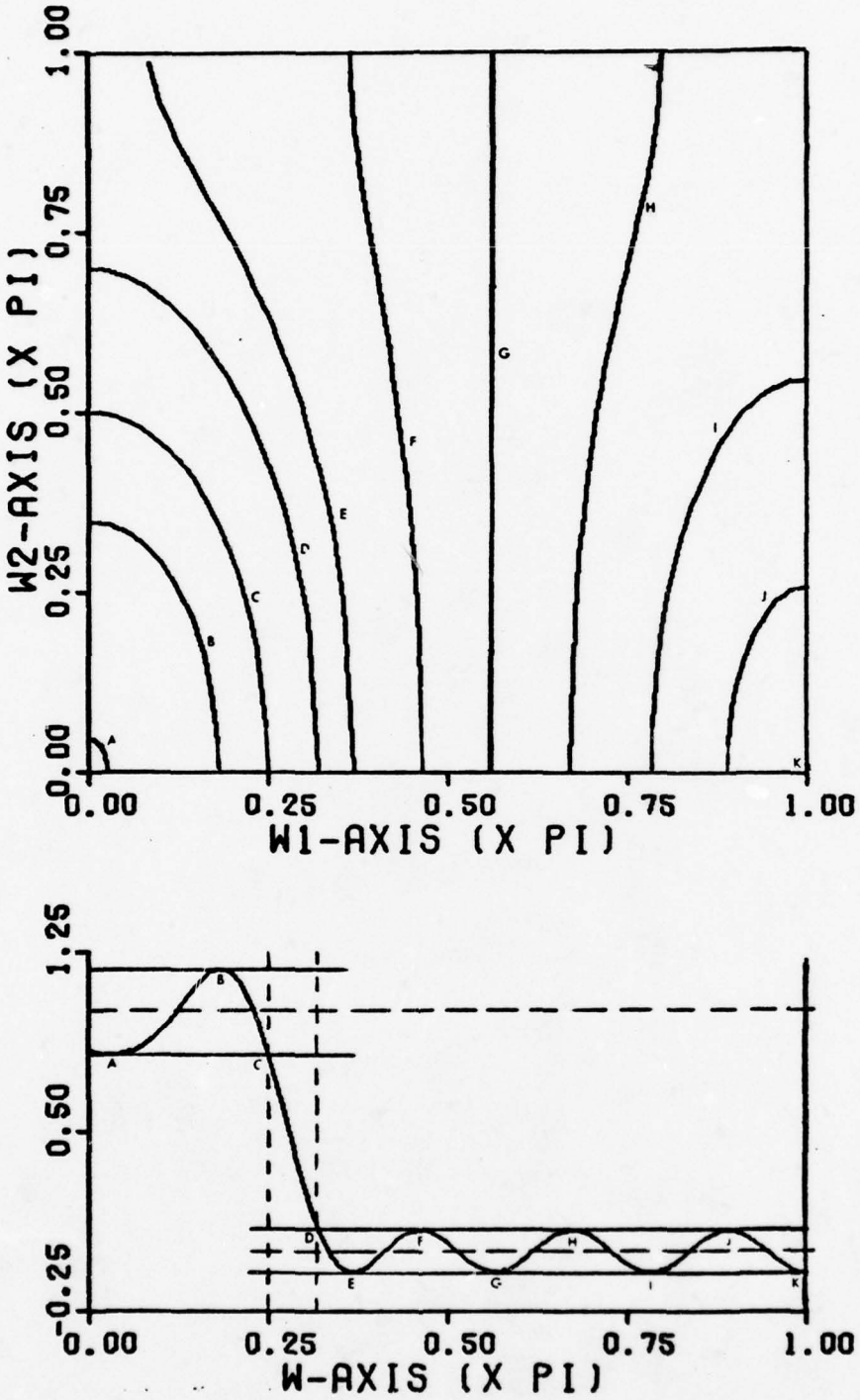



Fig. 14. Elliptical-Shaped Filter Contours. Major axis = .5π, Minor Axis = .25π. N = 19, $\omega_p$ = .2498π, $\omega_s$ = .32π, Wt = 1:2, $\delta_1$ = .177, $\delta_2$ = .087, $\underline{t}^T$ = [-.0482, .7559, .0482, .2440], $\bar{e}$ = .00016.

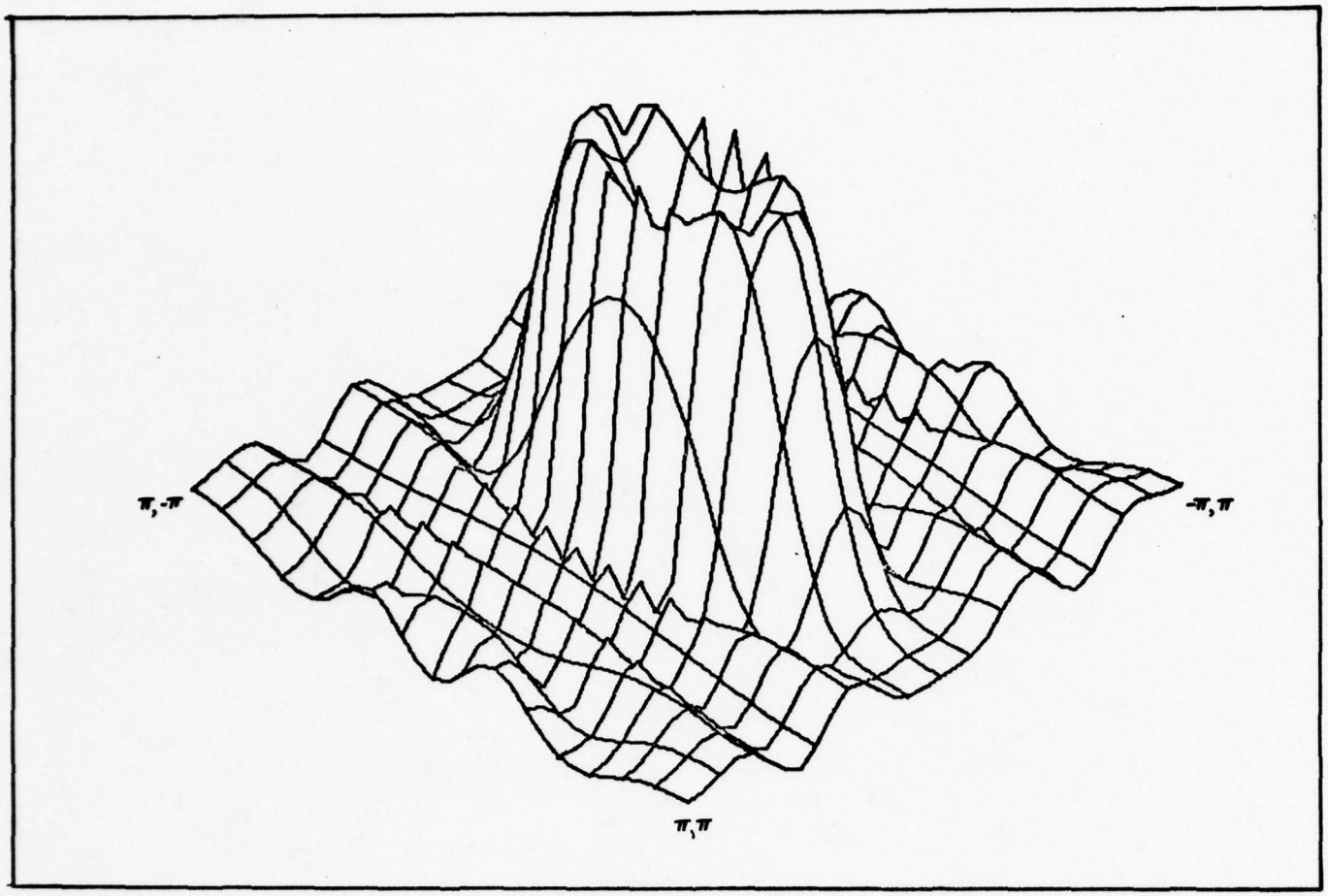



Fig. 15. Frequency Response of Elliptical-Shaped Filter. The specifications are the same as Fig. 14.

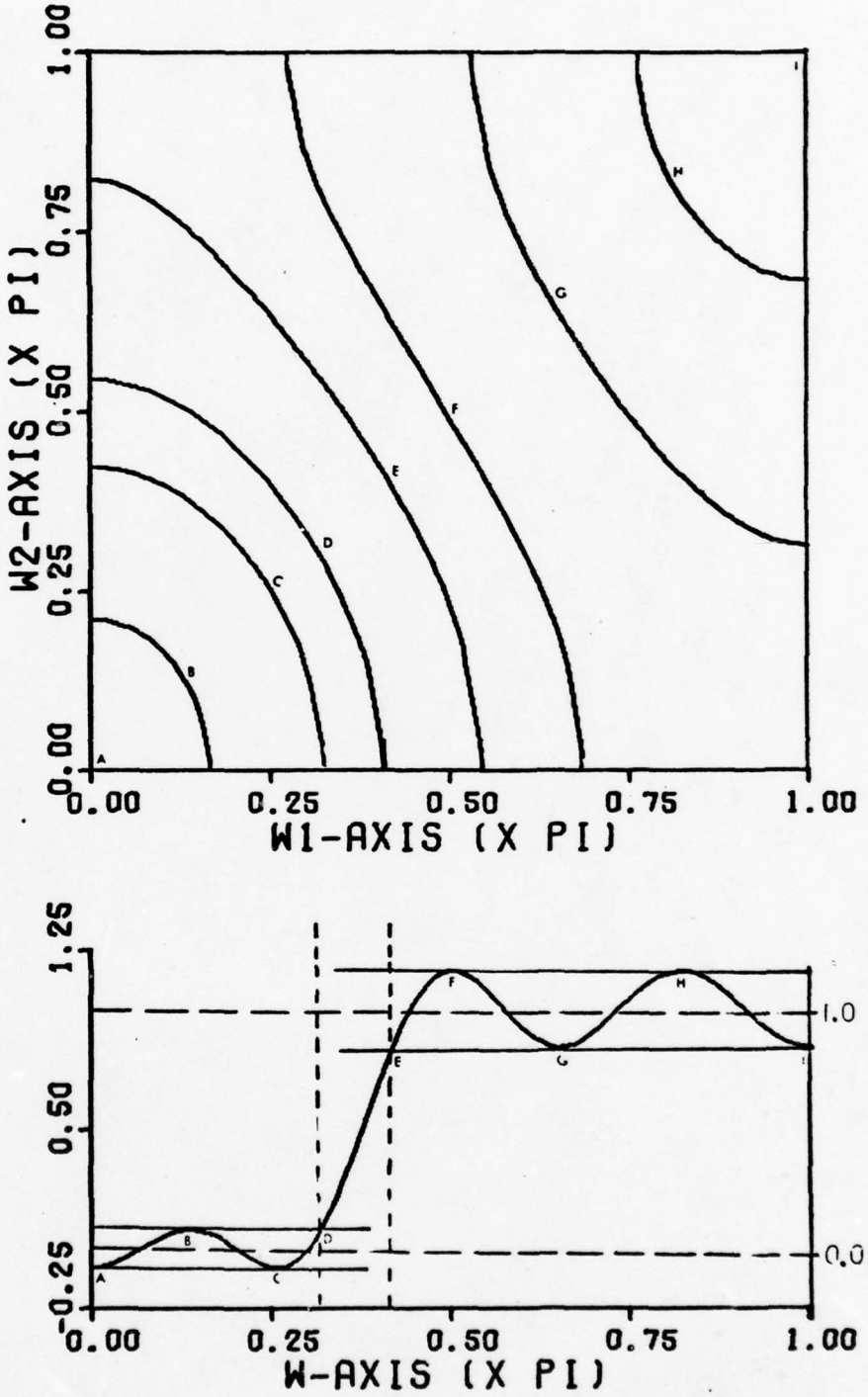



Fig. 16. Parabolic-Shaped Filter Contours. $\omega_2 = .8\pi - 2.4\pi x^2$. N = 15, $\omega_p = .42\pi$, $\omega_s = .32\pi$, Wt = 2:1, $\delta_1 = .079$, $\delta_2 = .160$, $\underline{t}^T = [-.0319, .6249, .3751, .0319]$, $\bar{e} = .019$

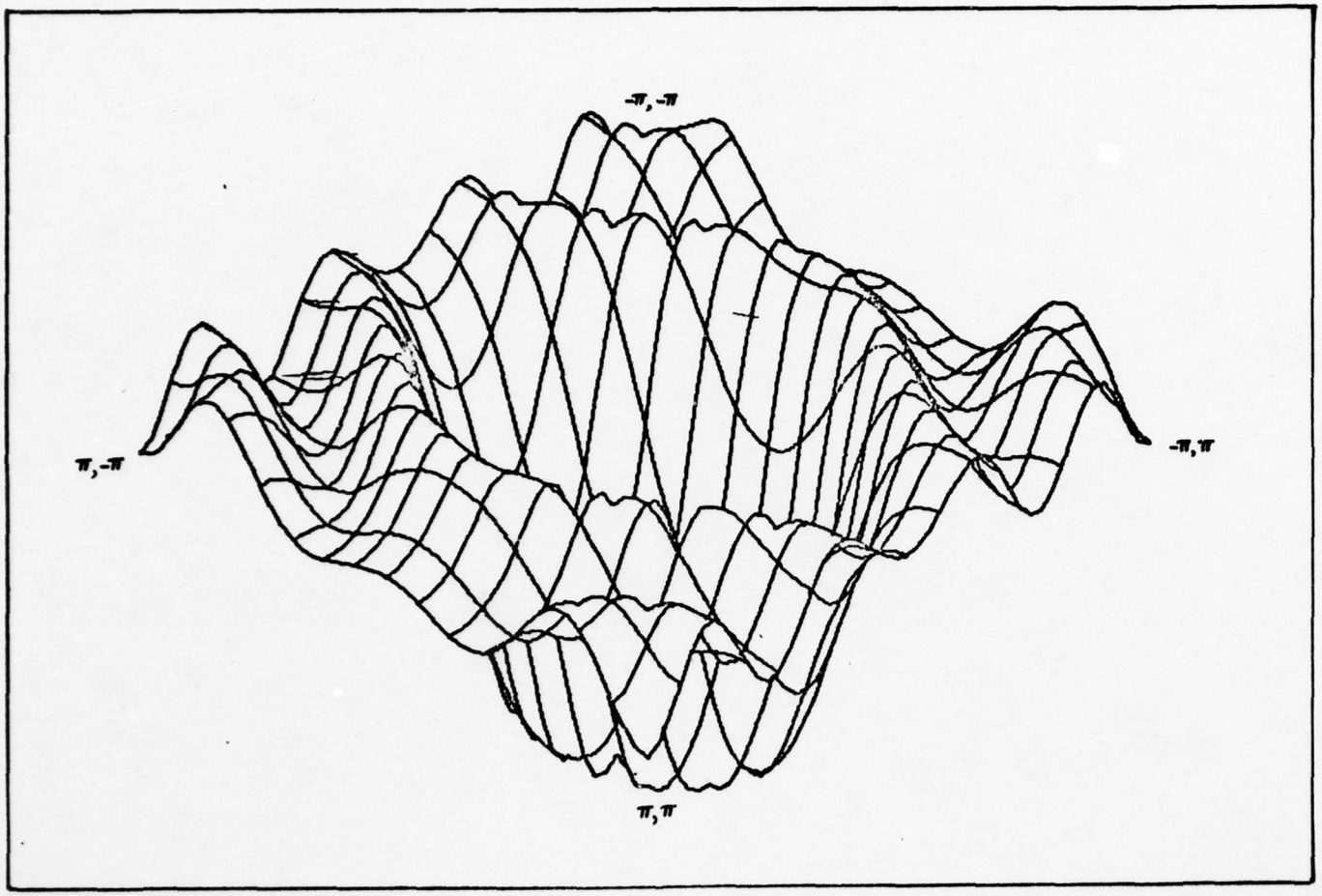



Fig. 17. Frequency Response of Parabolic-Shaped High Pass Filter. The specifications are the same as Fig. 17.

reverses the sense of the regions defined by Fig. 18. One could remedy this by using a high pass 1-D prototype. Another method is to use an alternate constraint, $0 \rightarrow (\pi,\pi)$, and then scale the result. This produces the values of Eq (130). The resulting contour mapping and a frequency response are shown in Figs. 19 and 20.

There are many possible variations of 2-D filters defined by straight lines. A diamond-shaped filter was specified by the passband contour

$$\omega_2 = \pi/2 - \omega_1 \tag{134}$$

The resulting mapping and a frequency response are shown in Figs. 21 and 22.

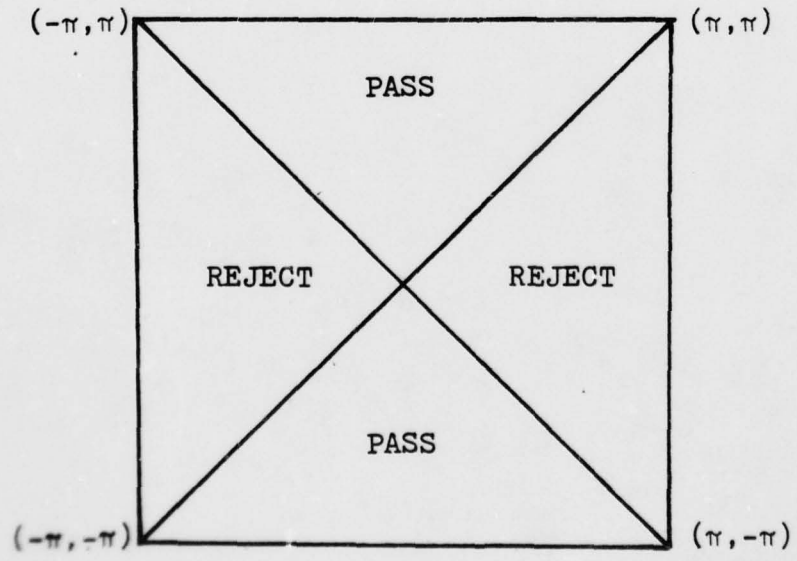



Fig. 18. Ideal "Fan" Filter

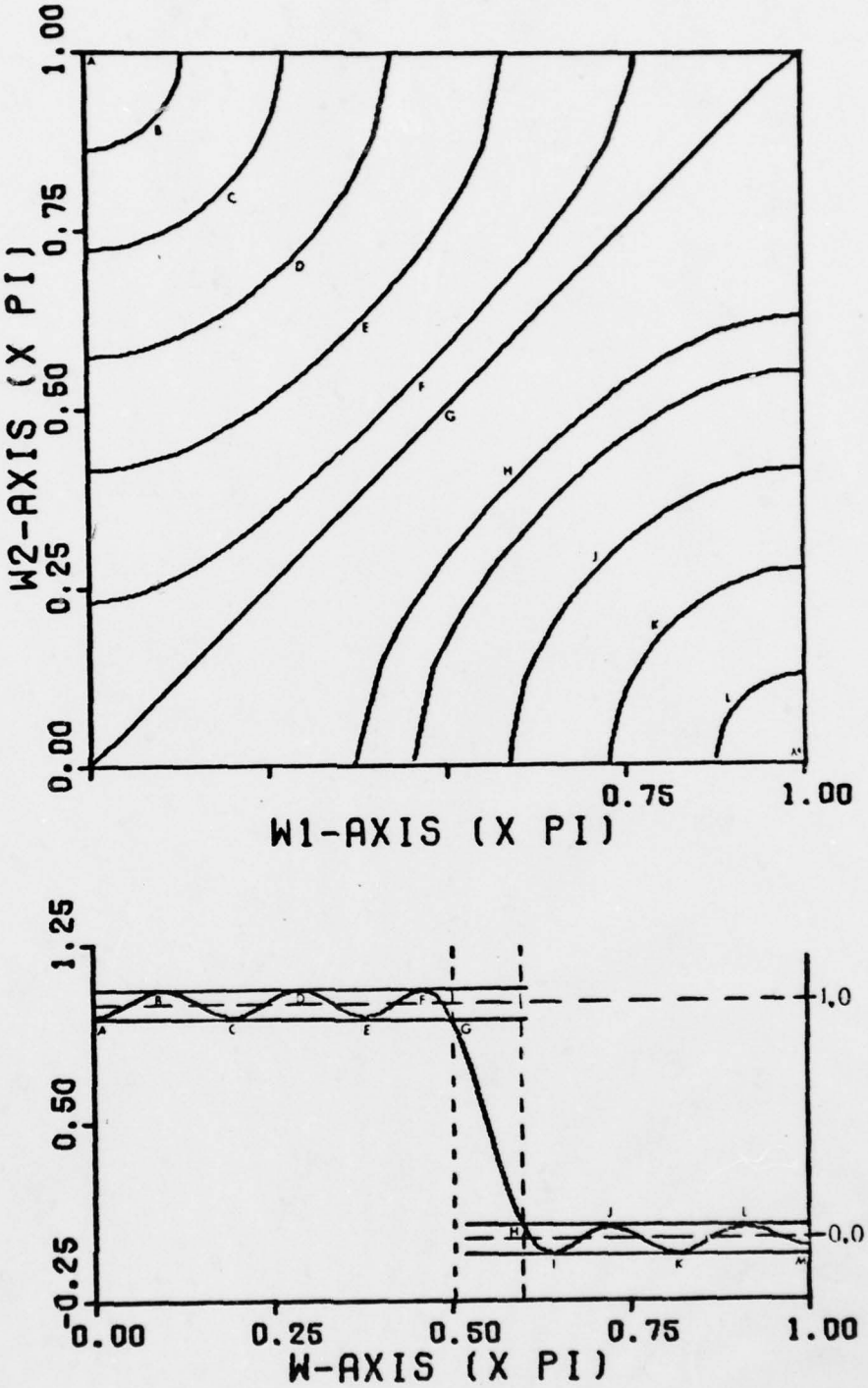



Fig. 19. Fan Filter Contours. $\omega_2 = \omega_1$. N = 21, $\omega_p = .5002\pi$, $\omega_s = .6\pi$, Wt = 1:1, $\delta = .055$, $\underline{t}^T = [-.0003, .4999, -.5008, -.0003]$, $\bar{e} = .00007$. Constraint: $0 \rightarrow (\pi,\pi)$ with scaling.

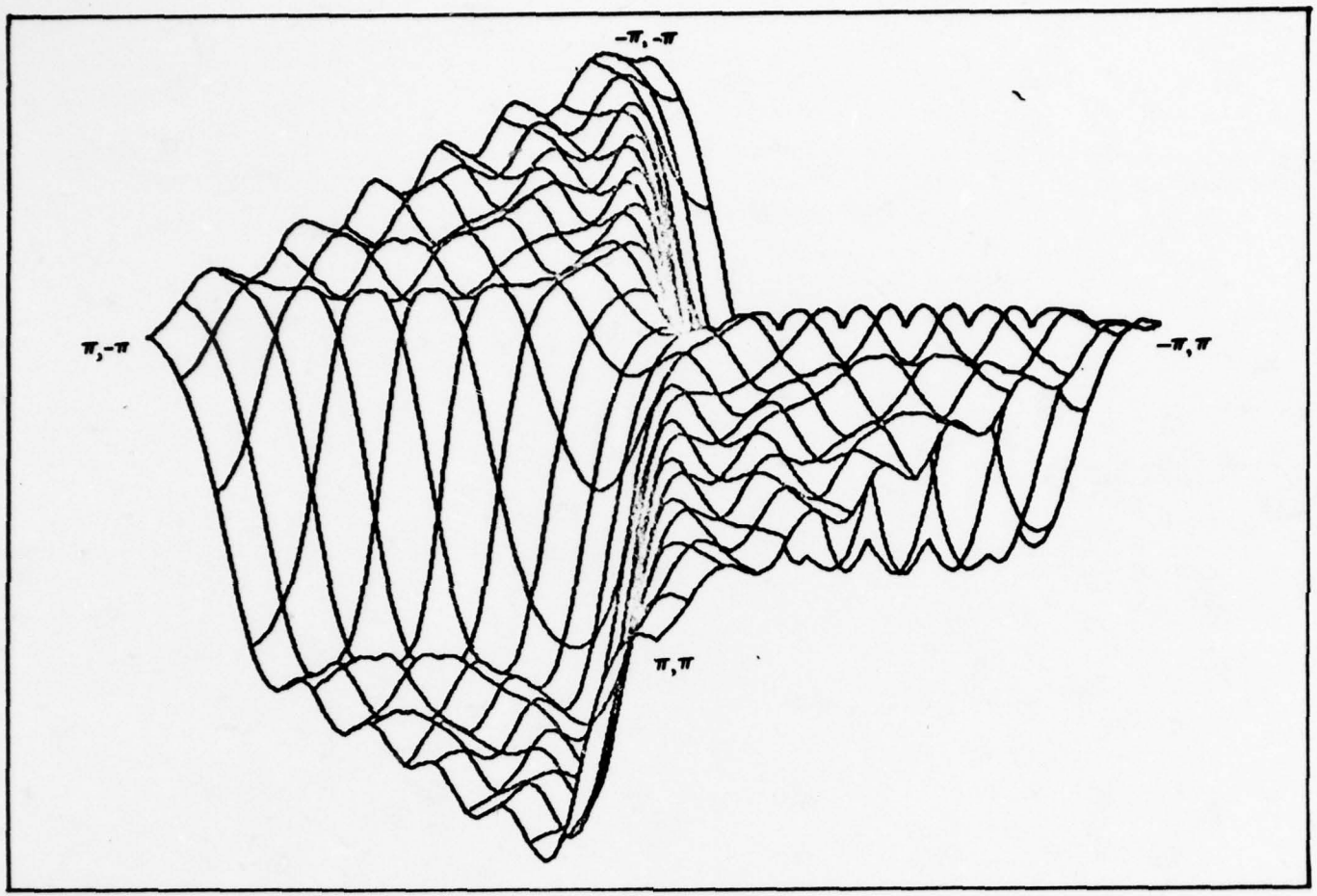



Fig. 20. Frequency Response of Fan Filter. The specifications are the same as in Fig. 19.

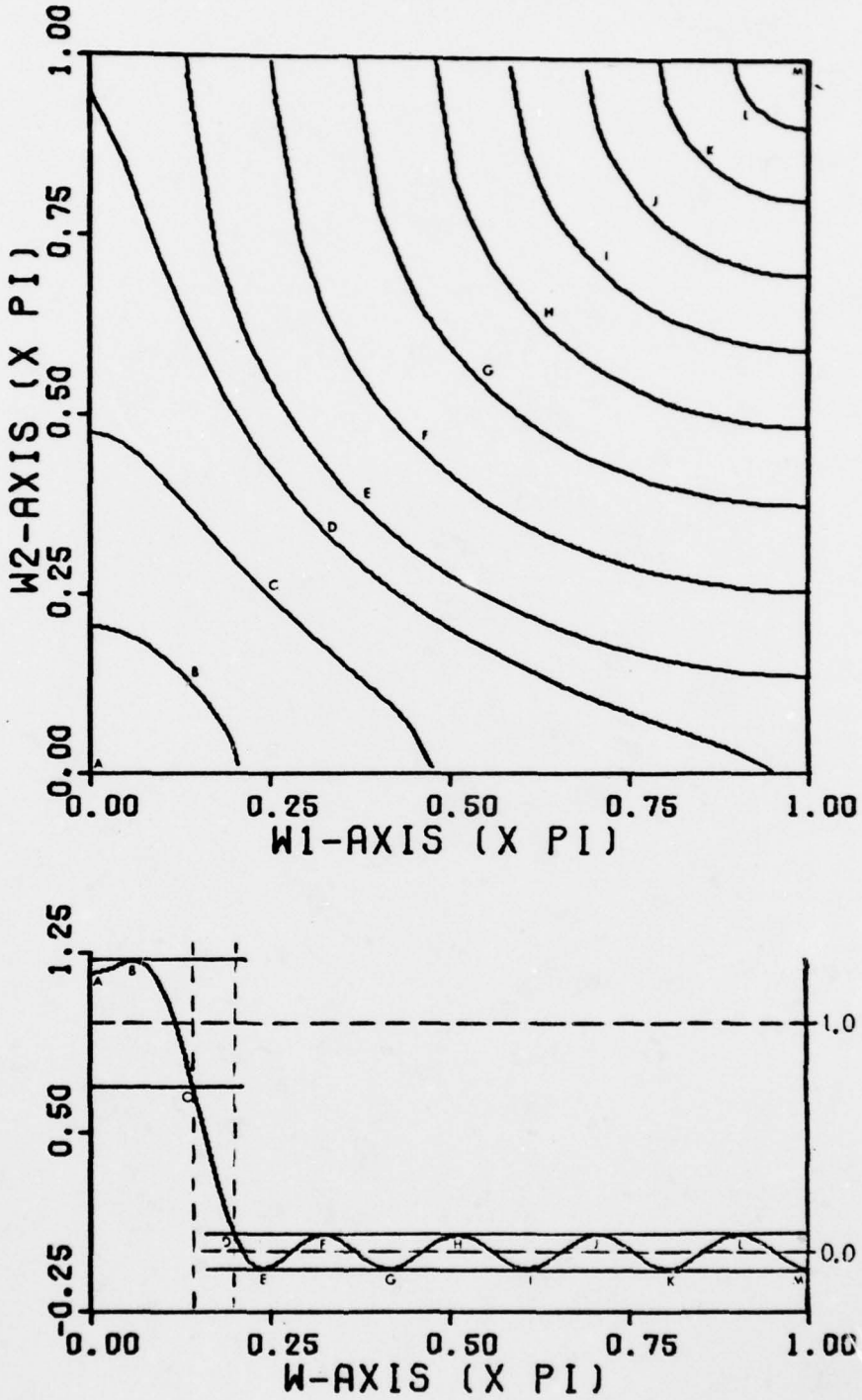



Fig. 21. Diamond-Shaped Filter Contours. $\omega_2 = \pi/2 - \omega_1$. N = 21, $\omega_p = .135\pi$, $\omega_s = .2\pi$, Wt = 1:3, $\delta_1 = .214$, $\delta_2 = .071$, $\underline{t}^T = [.4039, .5, .5, -.4039]$, $\bar{e} = .0030$

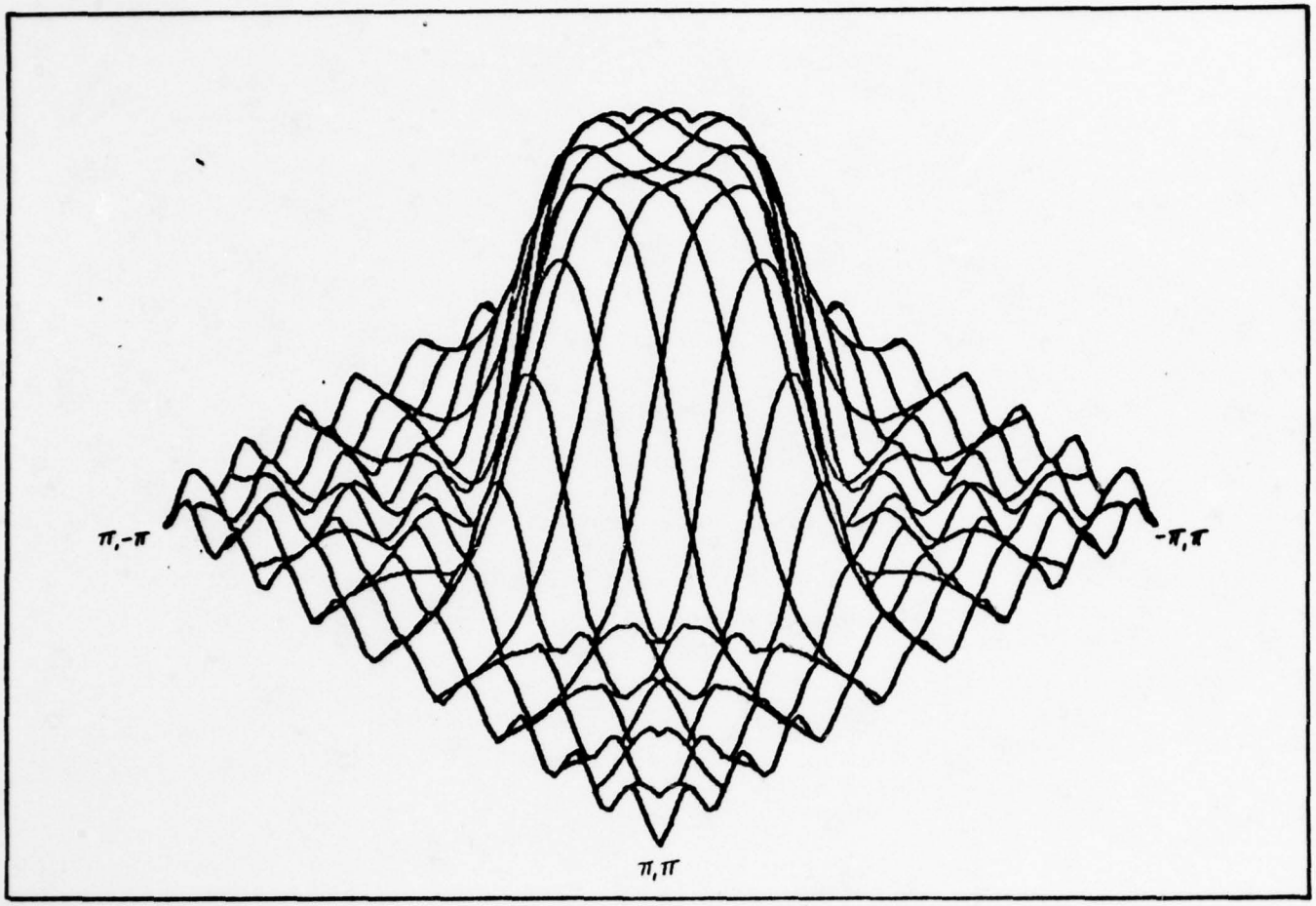



Fig. 22. Frequency Response of Diamond-Shaped Filter.
The specifications are the same as in Fig. 21.

Filters with More Complex Contours

To establish the limits of the design program, several novel filter shapes were designed. The results were generally disappointing, indicating that the method cannot deal with complex contour shapes effectively.

Oval-Shaped Contours. The desired contour for an oval-shaped filter was specified as

$$\omega_2 = \pi/2 \qquad \omega_1 < \pi/2 \tag{135}$$

$$\omega_2 = \sqrt{(\pi/2)^2 - (\omega_1 - \pi/2)^2}, \quad \omega_1 \geq \pi/2 \tag{136}$$

Eq (135) describes a circle of radius $0.5\pi$ centered on the $\omega_1$-axis. The scaled contour mapping is shown in Fig. 23. As $\omega$ varies either side of $\omega_o'$, the contours change dramatically. In this particular case, an assumption made in Chapter IV is clearly wrong: the immediate neighbors of this contour do not maintain the same shape.

Contour Defined by a Cubic. A contour specificed by the cubic

$$\omega_2 = 4/\pi^2 \, (\omega_1 - \pi/2)^3 + \pi/2 \tag{137}$$

produced a mapping with a very large average error. The contour approximation is worst in the vicinity of the origin. By using a properly weighted least-squares criteria, the fit might be improved. However, the primary fault is attributed to the approximating function, Eq (1). The

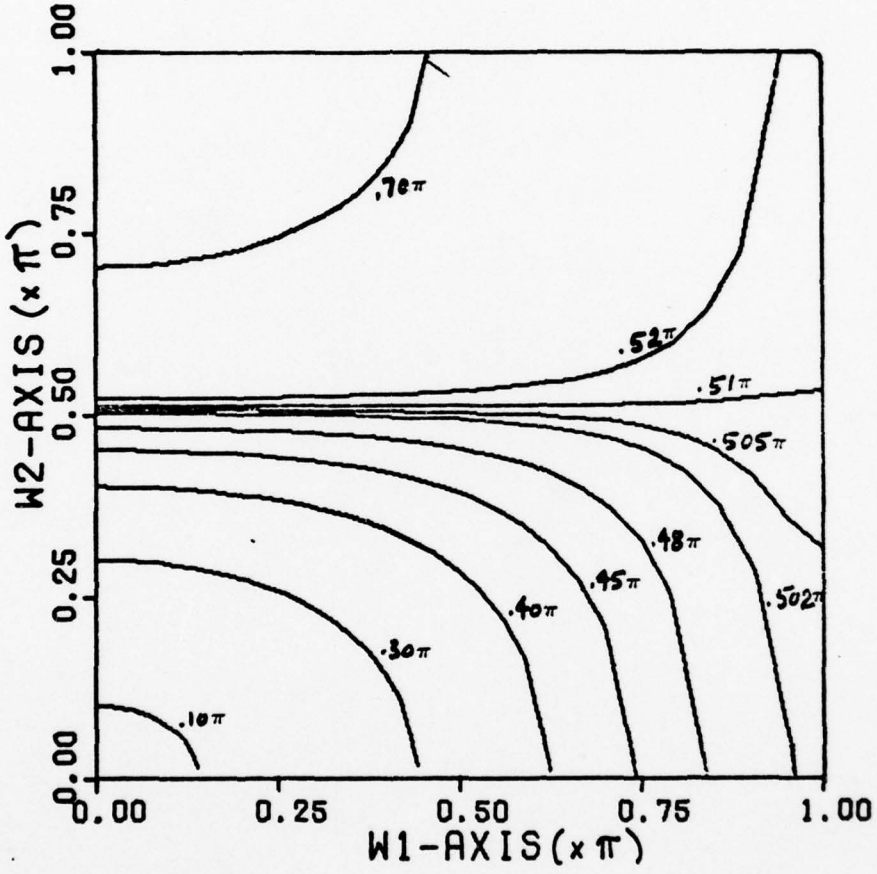



Fig. 23. Oval-Shaped Filter Contours. $\omega_o = .502\pi$, $\underline{t}^T = [-.0174, .0174, .5121, .4879]$, $\bar{e} = .0047$. Default constraint with scaling.

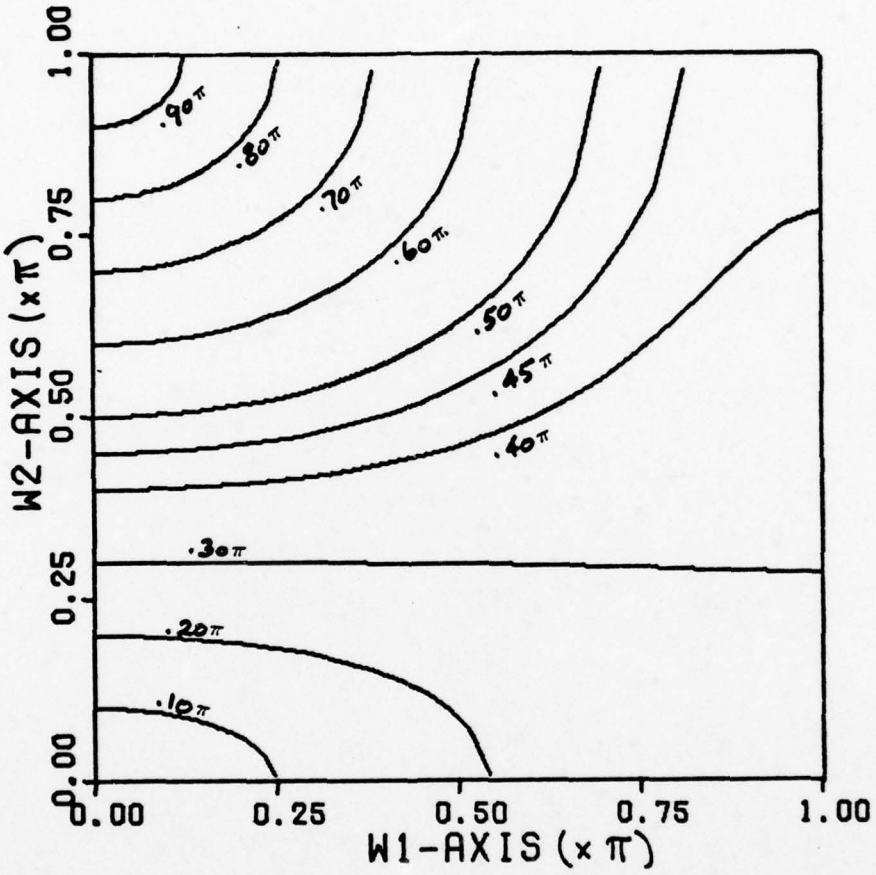



Fig. 24. Cubic Contours. $\omega_o$ = .40774π, $\underline{t}^T$ = [.2314, -.2314, .6000, .4000], $\bar{e}$ = 0.2292. Default constraint with scaling.

function does not have enough terms to fit complicated curves. Adoption of higher order cosine functions in a generalized McClellan Transformation, Eq (46), would solve this problem.

Non-monotonic Contours. In Chapter IV, it was proven that the contour mapping must be monotonic. All empirical evidence verifies this; a monotonic contour has never been observed. This does not imply that all contours for a given transformation will be monotonic in the same direction. However, each individual contour will be monotonic. All attempts to specify a non-monotonic contour led to ill-defined mappings that bore no resemblance to the desired contour. For example, the semi-circle

$$\omega_2 = \sqrt{(\pi/2)^2 - (\omega_1 - \pi/2)^2} \tag{138}$$

resulted in the contour mapping shown in Fig. 25.

Contours not Continuously Differentiable. Several contours that were not continuously differentiable (non-smooth) were used to test the design program. Immediately, problems arose. Fig. 26 shows the contours resulting from an attempt to specify the right-angle contour

$$\omega_2 = \pi/2, \quad \omega_1 < \pi/2 \tag{139}$$

$$\omega_2 = 0, \quad \omega_1 \geq \pi/2 \tag{140}$$

When the default constraint was used, the mapping was not well-defined, even after scaling. By imposing the addi-

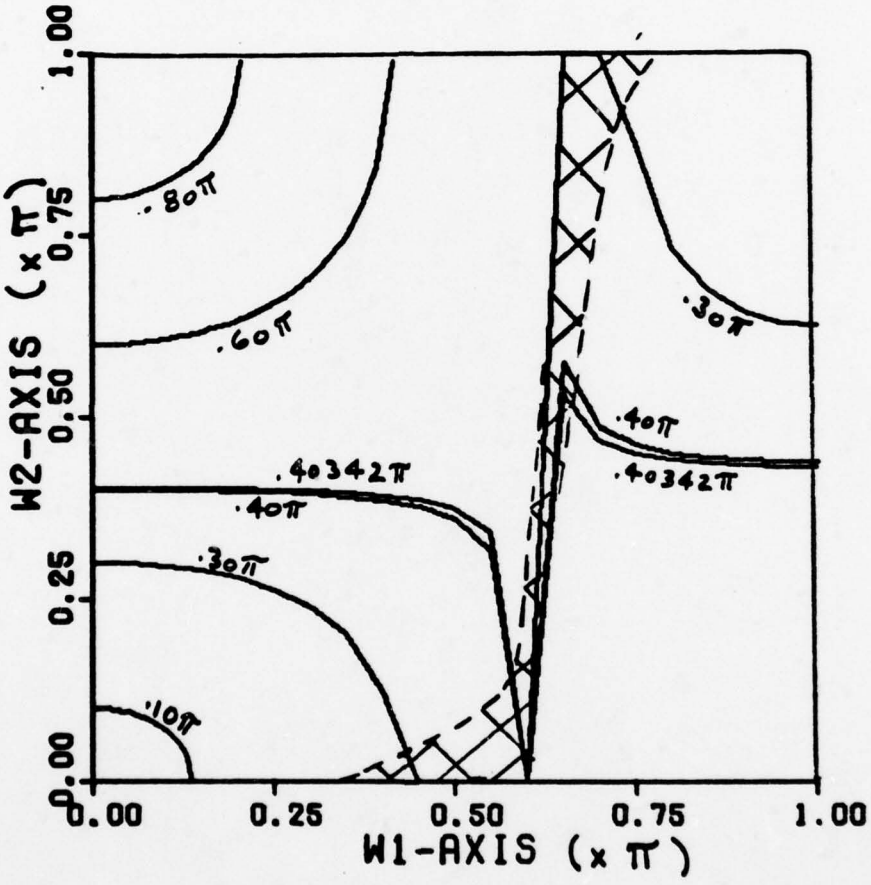



Fig. 25. Non-Monotonic Contour Specification. $\omega_o = .40342\pi$, $\underline{t}^T = [.1998, -.1998, .2577, .7423]$, $\bar{e} = .1286$

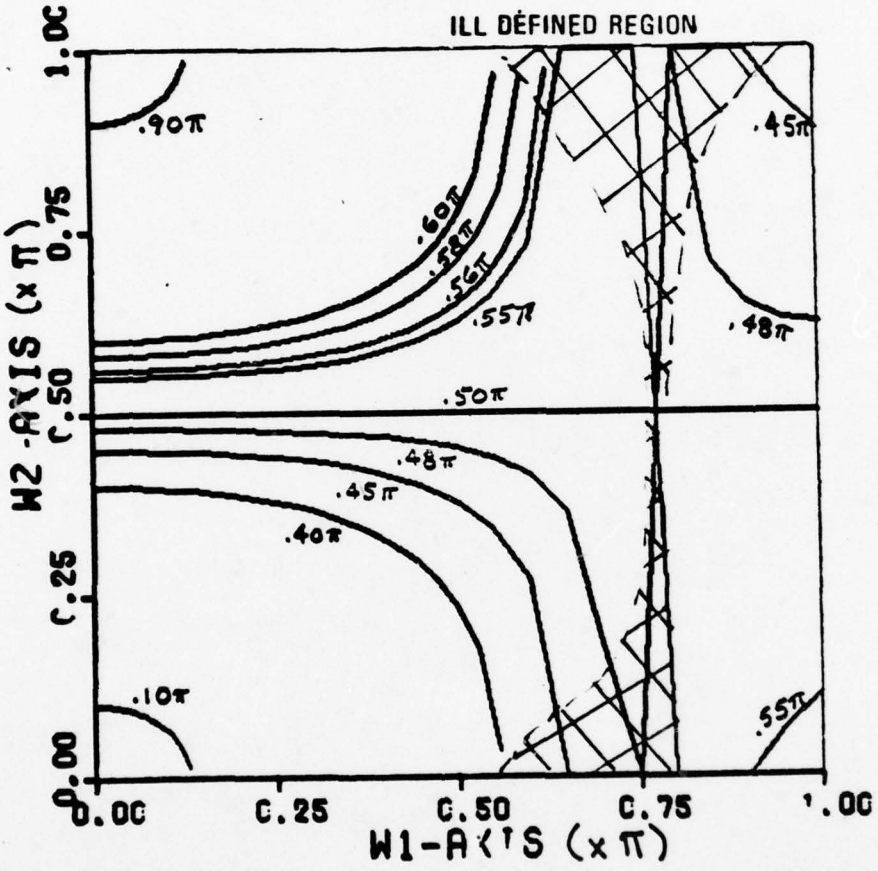



Fig. 26. Right-Angle Contours: Default Constraint. $\omega_o = .50\pi$, $\underline{t}^T = [0.0,\ 0.0,\ 0.4166,\ 0.5834]$, $\bar{e} = .1227$

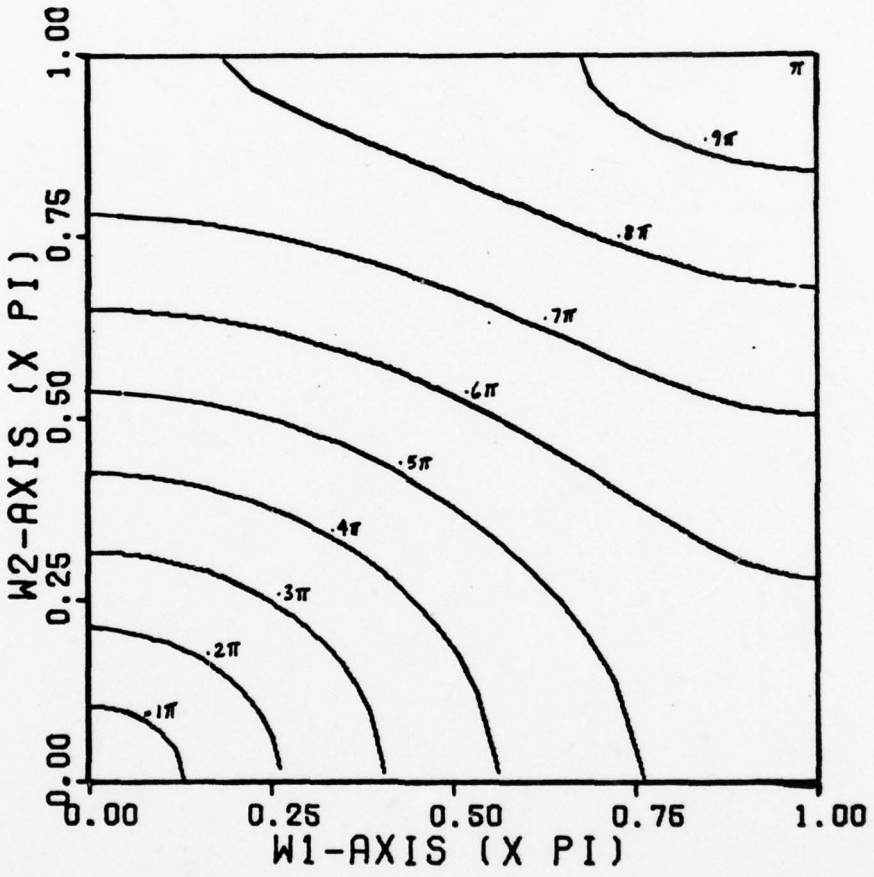



Fig. 27. Right-Angle Contours: Alternate Constraints. $\omega_o = .5\pi$, $\underline{t}^T = [-.2356, .3399, .6601, .2356]$, $\bar{e} = .1465$. Constraint: $0 \rightarrow (0,0)$, $\pi \rightarrow (\pi,\pi)$

tional constraint $\pi \rightarrow (\pi,\pi)$, the mapping became well-defined but the average error was large (Fig. 27). For non-smooth increasing contours, the best constraints to use were $0 \rightarrow (0,\pi)$ and $\pi \rightarrow (\pi,0)$. These constraints were suggested by McClellan (Ref 1:36-38), but they do not always guarantee that the range of the mapping will be adequate. However, the scaling algorithm can still be used to spread out the contours.

# VII. Conclusions and Recommendations

## Conclusions

The primary objective of this investigation was to provide a computer-aided method for designing 2-D digital filters. Available optimal design methods were rejected because of their extreme inefficiency. The McClellan Transformation provided an effective method for designing sub-optimal filters that can meet a wide variety of specifications. Design time for the filter varies in rough proportion with the square of the filter order (Fig. 28). For filters exceeding 21x21 points, the computational time increases rapidly. Even still, the design time compares very favorably with other available methods. The bulk of execution time is spent calculating the impulse response coefficients; if these are not required, the design time is shortened considerably (Fig. 28).

The coded and tested FORTRAN program is available to Air Force Institute of Technology users on the CDC 6600 computer. The structure of the program is outlined in Appendix C, and the use of the program is explained in Appendix D. A complete design example is included.

The limitations of the method must be recognized:

1. The algorithm maps a specific 1-D frequency to an approximation of a desired contour. The "goodness" of the

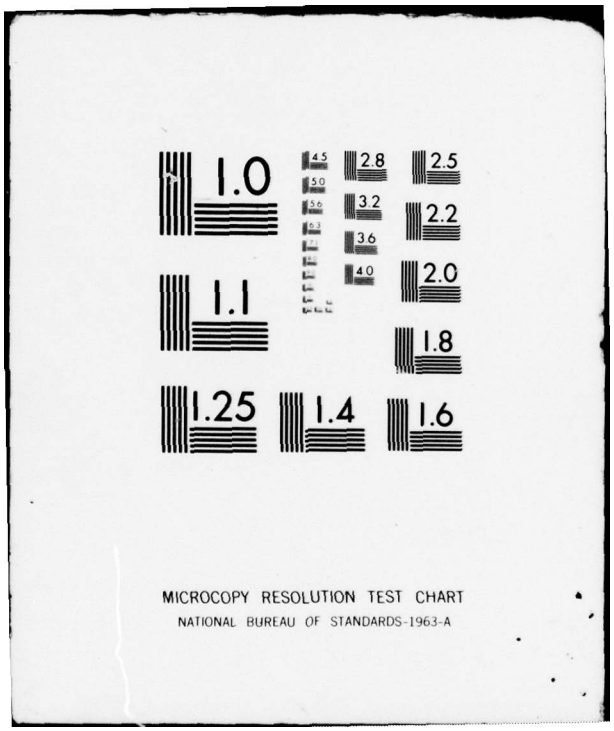
AD-A055 449 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/3
INVESTIGATION OF OPTIMAL LINEAR SHIFT-INVARIANT TWO-DIMENSIONAL--ETC(U)
DEC 77 R B BROWN

UNCLASSIFIED AFIT/GE/EE/77-10 NL

2 OF 2
AD
A055449



END
DATE
FILMED
8 -78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

contour approximation must be evaluated in each design. Contours with high curvature or corners generally lead to unacceptable mappings due to large rms errors.

2. The specified contour must be monotonic in the region $[0,\pi]x[0,\pi]$, limiting the flexibility.

3. For some contours, the default constraints may prove inadequate. The suggestions in Chapter VI should be considered.

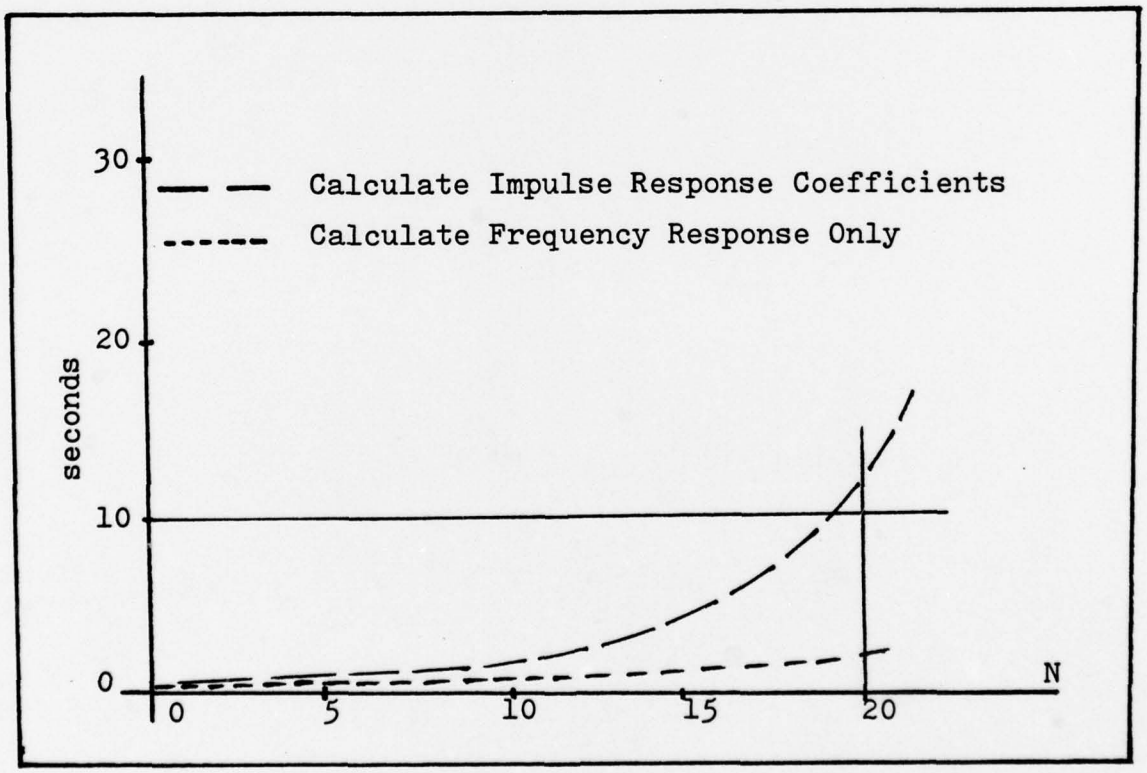



Fig. 28. Approximate Design Times for 2-D Design Program

## Recommendations for Future Work

Two-dimensional digital filtering is an open field for research. This investigation has concentrated on optimal non-recursive filters. Recursive filter design is worth a full investigation. Little has been said regarding practical realization or implementation of 2-D filters. As an aide to research in this important area, a brief supplementary bibliography has been prepared. Also of importance are studies of the practical applications of 2-D filters.

There are several areas of further work directly related to this present investigation. Each of the following suggestions would make the 2-D design program more useful and powerful:

1. It would be advantageous to find $\underline{t}$ such that two or more contours are approximated simultaneously, according to an error criteria.

2. It would be advantageous to adopt the generalized McClellan Transformation, Eq (46), to permit better approximations of complicated contours.

3. It would be advantageous to calculate the impulse response coefficients by a more efficient method. An alternate recursion has been described as being faster than the polynomial expansion used in this investigation (Ref 26).

## Bibliography


1. McClellan, James H. On the Design of One-Dimensional and Two-Dimensional FIR Digital Filters. Rice University, Ph.D., 1973.

2. Openheim, Alan V. and Ronald W. Schafer. Digital Signal Processing. Englewood Cliffs, N. J.: Prentice-Hall, 1975.

3. Parks, Thomas W. and James H. McClellan. "Chebychev Approximation for Non-recursive Digital Filters with Linear Phase." IEEE Transactions on Circuit Theory, CT-19: 189-194 (March 1972).

4. McClellan, James H., Thomas W. Parks, and Lawrence R. Rabiner. "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters." IEEE Transactions on Audio and Electroacoustics, AU-21: 97-117 (December 1973).

5. Hunt, B. R. "Digital Image Processing." Proceedings of the IEEE, 63: 693-708 (April 1975).

6. Wood, Lawrence C. and Sven Treital. "Seismic Signal Processing." Proceedings of the IEEE, 63: 649-661 (April 1975).

7. Rabiner, Lawrence R. and Bernard Gold. Theory and Application of Digital Signal Processing. Englewood Cliffs, N. J.: Prentice-Hall, 1975.

8. Kamp, Y. and J. P. Thiran. "Chebyshev Approximation for Two-Dimensional Non-recursive Digital Filters." IEEE Transactions on Circuits and Systems, CAS-22: 208-217 (March 1975).

9. Mercereau, Russel M. and David B. Harris. "A Comparison of Algorithms for Minimax Design of Two-Dimensional Linear Phase FIR Digital Filters." To be published by IEEE.

10. Mercereau, Russel M. and Dan E. Dudgeon. "Two-Dimensional Digital Filtering." Proceedings of the IEEE, 63: 610-623 (April 1975).

11. Cheney, E. W. Introduction to Approximation Theory. New York: McGraw-Hill, 1966.

12. Hofstetter, E. et al. "A New Technique for the Design of Non-recursive Digital Filters." Proceedings of the 5th Annual Princeton Conference on Information Sciences and Systems: 64-72 (March 1971).

13. Remez, E. Ya. General Computation Methods of Tchebysheff Approximation. Kiev: 1957 (Atomic Energy Translation 4491).

14. Petersen, D. P. and D. Middleton. "Sampling and Reconstruction of Wave Number Limited Functions in N-dimensional Euclidean Space." Information and Control, 5: 279-323 (1962).

15. Mercereau, Russel M. et al. "McClellan Transformations for Two-Dimensional Digital Filtering: I -- Design." IEEE Transactions on Circuits and Systems, CAS-23: 405-414 (July 1976).

16. Woods, John W. "2-D Discrete Linear Systems." Two-Dimensional Digital Signal Processing. Unpublished notes from Rensselaer Polytechnic Institute Office of Continuing Education short course, July 25-28, 1977.

17. Churchill, Ruel V., James W. Brown, and Roger F. Verhey. Complex Variables and Applications. New York: McGraw-Hill, 1974.

18. Huang, Thomas S. "Stability of Two-Dimensional Recursive Filters." IEEE Transactions on Audio and Electroacoustics, AU-20: 158-163 (June 1972).

19. Shanks, J. L. et al. "Stability and Synthesis of Two-Dimensional Recursive Filters." IEEE Transactions on Audio and Electroacoustics, AU-20: 115-128 (June 1972).

20. Kamp, Y. and Y. Genin. "Counter-example in the Least-Squares Inverse Stabilization of 2D Recursive Filters." Electronic Letters, 11: 330-331 (July 1975).

21. -----. "Two-Dimensional Stability and Orthogonal Polynomials on the Hypercircle." Proceedings of the IEEE, 65: 873-881 (June 1977).

22. Huang, Thomas S. "Two-Dimensional Windows." IEEE Transactions on Audio and Electroacoustics, AU-20: 88-90 (March 1972).

23. Openheim, A. V. (editor), Digital Signal Processing Committee. Selected Papers in Digital Signal Processing, II. New York: IEEE Press, 1976.

24. Quatieri, Thomas F. The Design of Two-Dimensional Digital Filters by Generalized McClellan Transformations. Massachusetts Institute of Technology, Department of Electrical Engineering. Masters Thesis, June 1975.

25. Mitra, S. K. and S. Chakrabarti. "Design of Two-Dimensional Filters Via Spectral Transformations." Proceedings of the IEEE, 65: 905-914 (June 1977).

26. McClellan, James H. and Thomas W. Parks. "Equiripple Approximation of Fan Filters." Geophysics, 37:573-583 (August 1972).

27. Stark, Peter A. Introduction to Numerical Methods. London, England: MacMillan, 1970.

28. Nielsen, K. L. Methods of Numerical Analysis. New York: MacMillan, 1964.

29. Constantine, Larry R. and Ed Yourdan. Structured Design. New York: Yourdan, Inc., 1975.

30. Control Data Corporation. FORTRAN Extended Version 4 Reference Manual. Sunnyvale, Cal.: 1976.

31. Larimer, Stanley J. Development of a Complete Computer-Aided Design Package for Digital and Continuous Control System Analysis and Synthesis. Air Force Institute of Technology, Department of Electrical Engineering. Masters Thesis to be submitted.

32. Control Data Corporation. SCOPE 3.4 Reference Manual. Sunnyvale, Cal: 1975.

33. University of Washington Academic Computer Center. Cataloged Procedures and Other Techniques for Manipulating Control Cards. Reprinted by ASD Computer Center, Wright-Patterson Air Force Base, Ohio. October, 1976.

34. Integrated Software Systems Corporation. DISSPLA Advanced Manual, Version I. San Diego, Cal: 1973.

35. McClellan Transformation Two-Dimensional Filter Design Program (MCLTRN). School of Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. December, 1977.

## Supplementary Bibliography

## of

## References on Implementation of 2-D Digital Filters


Ekstrom, Michael P. "Computational Considerations for Filter Realization." Two-Dimensional Digital Signal Processing. Unpublished notes from Rensselaer Polytechnic Institute Continuing Education short course, July 25-28, 1977.

Mitra, Sanjit K. "2-D Digital Filter Realization Methods." Two-Dimensional Digital Signal Processing. Unpublished notes from Rensselaer Polytechnic Institute Continuing Education short course, July 25-28, 1977.

Mitra, Sanjit K. and N. A. Pendergraa. "Realization of Two-Dimensional Recursive Digital Filters." IEEE Transactions on Circuits and Systems, CAS-22:177-184 (March 1975).

Mercereau, Russel M. and W. F. G. Mecklenbrauker. "McClellan Transformations for Two-Dimensional Digital Filtering: II -- Implementation." IEEE Transactions on Circuits and Systems, CAS-23:414-422 (July 1976).

Ming, Duenn Ni, et al. Two-Dimensional Digital Filtering and Its Error Analysis. Report No. AFOSR 72-2371. Air Force Office of Scientific Research/NM. Arlington, VA, 1974 (AD-780653).

## Appendix A

### Algorithm to Determine U for Eq (40)

Given the three Chebychev polynomial identities

$$T_o(x) = 1 \tag{141}$$

$$T_1(x) = x \tag{142}$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{143}$$

one can write an expression for any $T_m(x)$. Each of these equations could, in turn, be used to solve for the powers $x^0$, $x^1$,...,$x^n$. Table III illustrates the first few terms of such a listing.

A recursive algorithm for determining the coefficients of $T_o(x)$, $T_1(x)$,...,$T_n(x)$ in each expression for $x^0$, $x^1$,...$x^n$ has been developed. The algorithm was deduced by writing the equations for $x^i$ in matrix form:

$$\begin{vmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \\ x^4 \\ \vdots \\ x^n \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & \\ 1/2 & 0 & 1/2 & 0 & 0 & \cdots \\ 0 & 3/4 & 0 & 1/4 & 0 & \\ 3/8 & 0 & 4/8 & 0 & 1/8 & \\ & \vdots & & & & \end{vmatrix} \begin{vmatrix} T_o(x) \\ T_1(x) \\ T_2(x) \\ T_3(x) \\ T_4(x) \\ \vdots \\ T_n(x) \end{vmatrix} \tag{144}$$

TABLE III

| The Chebychev Polynomials Solved for x | |
|---|---|
| $T_o(x) = 1 = x^0$ | $x^0 = T_o$ |
| $T_1(x) = x$ | $x^1 = T_1$ |
| $T_2(x) = 2x^2 - 1$ | $x^2 = 1/2\left[T_o + T_2\right]$ |
| $T_3(x) = 4x^3 - 3x$ | $x^3 = 1/4\left[3T_1 + T_3\right]$ |
| $T_4(x) = 8x^4 - 8x^2 + 1$ | $x^4 = 1/8\left[3T_o + 4T_2 + T_4\right]$ |
| ⋮ | ⋮ |

If the rows of Eq (144) are partitioned,

$$x^0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \end{bmatrix} = \underline{U}_o^T \underline{T} \tag{145}$$

$$x^1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots \end{bmatrix} = \underline{U}_1^T \underline{T} \tag{146}$$

$$x^2 = 1/2 \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & \cdots \end{bmatrix} = \underline{U}_2^T \underline{T} \tag{147}$$

$$x^3 = 1/4 \begin{bmatrix} 0 & 3 & 0 & 1 & 0 & \cdots \end{bmatrix} = \underline{U}_3^T \underline{T} \tag{148}$$

$$x^4 = 1/8 \begin{bmatrix} 3 & 0 & 4 & 0 & 1 & \cdots \end{bmatrix} = \underline{U}_4^T \underline{T} \tag{149}$$

⋮

it is noted that for $m > 1$

$$\underline{U}_m^T = \begin{bmatrix} U_{m,0} & U_{m,1} & \cdots & U_{m,n} \end{bmatrix} \tag{150}$$

where

$$U_{m,0} = U_{m-1,1} \tag{151}$$

$$U_{m,1} = 2U_{m-1,0} + U_{m-1,2} \tag{152}$$

$$U_{m,k} = U_{m-1,k-1} + U_{m-1,k+1} \qquad k<m \tag{153}$$

$$U_{m,k} = 1 \qquad k=m \tag{154}$$

$$U_{m,k} = 0 \qquad k>m \tag{155}$$

Thus, a recursive formula for each expression $x^0$, $x^1$,...,$x^n$ is available:

$$x^m = 1/2^{m-1}\ \underline{U}_m^T\ \underline{T} \tag{156}$$

## Appendix B

### The Method of Least-Squares Curve Fitting with Constraints

**Case I: No constraints on $\underline{t}$.**

A known (or measured) function $y = f(x)$ is given. It is desired to approximate y with the function

$$p_m(x) = t_1 g_1(x) + t_2 g_2(x) + \ldots + t_m g_m(x) \tag{157}$$

where each $g_i(x)$ is linearly independent. Defining

$$\delta_k = p_m(x_k) - y_k, \quad k = 1,2,\ldots,N \tag{158}$$

it is desired to minimize

$$\Delta(\underline{t}) = \sum_{k=0}^{N} \delta_k{}^2 = \sum_{k=0}^{N} \left[ p_m(x_k) - y_k \right]^2 \tag{159}$$

where the coefficients $\underline{t}$ are unknown and unconstrained. Let

$$\alpha_{jk} = \alpha_{kj} = \sum_{i=0}^{N} g_j(x_i) g_k(x_i) \tag{160}$$

$$\underline{\alpha} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots \\ \alpha_{21} & \alpha_{22} & \\ \vdots & & \end{bmatrix} \tag{161}$$

$$\beta_j = \sum_{i=0}^{N} y_i g_j(x_i) \tag{162}$$

$$\underline{\beta}^T = \begin{bmatrix} \beta_1 & \beta_2 & \cdots \end{bmatrix} \tag{163}$$

It can be shown that (Ref 27:322-323)

$$\underline{\alpha}\ \underline{t} = \underline{\beta} \tag{164}$$

or

$$\underline{t} = \underline{\alpha}^{-1}\ \underline{\beta} \tag{165}$$

is the best least-squares fit of the approximating function $p_m(x)$ to the data set y, provided that $\underline{\alpha}$ is non-singular.

Case II: One constraint on $\underline{t}$.

The constraint on $\underline{t}$ has the form

$$h(\underline{t}) = c_1 t_1 + \ldots + c_m t_m - c_o = 0 \tag{166}$$

A necessary condition for an extremum is that the total differential* vanish:

$$p_{t_1} dt_1 + \ldots + p_{t_m} dt_m = 0 \tag{167}$$

The total differential of the constraint is

$$h_{t_1} dt_1 + \ldots + h_{t_m} dt_m = 0 \tag{168}$$

If Eq (168) is multiplied by $\lambda$, a Lagrangian multiplier, and added to Eq (167) (Ref 28:169-170)

---

*$p_{t_1}$ is shorthand notation for $\partial p/\partial t_1$.

$$(p_{t_1} + \lambda h_{t_1})dt_1 + \ldots + (p_{t_m} + \lambda h_{t_m})dt_m = 0 \qquad (169)$$

Since Eq (169) is satisfied if

$$p_{t_1} + \lambda h_{t_1} = 0 \qquad (170)$$

$$\vdots$$

$$p_{t_m} + \lambda h_{t_m} = 0 \qquad (171)$$

$$h(\underline{t}) = 0 \qquad (172)$$

this gives (m + 1) equations in the m + 1 unknowns: $\underline{t}$, $\lambda$.

As in Case I, Eq (159) for $\Delta(\underline{t})$ can be written. If one multiplies the term in brackets and performs a partial differentiation with respect to $\underline{t}$,

$$\Delta_{t_1} = \sum_{i=0}^{N} (2t_1 g_1^2 + 2t_2 g_1 g_2 + \ldots - 2g_1 y_i) \qquad (173)$$

$$\vdots$$

$$\Delta t_m = \sum_{i=0}^{N} (2t_1 g_1 g_m + \ldots + 2t_m g_m^2 - 2g_m y_i) \qquad (174)$$

Because Eqs (170) - (172) were determined for any $p_t$, one can substitute $\Delta_t$ for $p_t$. Letting

$$S_{kj} = S_{jk} = 2\sum_{i=0}^{N} g_k(x_i) g_j(x_i) \qquad (175)$$

$$Y_k = 2\sum_{i=0}^{N} g_k(x_i) y_i \qquad (176)$$

$$\underset{\sim}{\overset{*}{S}} = \begin{vmatrix} S_{11} & S_{12} & \cdots & S_{1m} & h_{t_1} \\ \vdots & & & & \vdots \\ S_{m1} & S_{m2} & & S_{mm} & h_{t_m} \\ c_1 & c_2 & \cdots & c_m & 0 \end{vmatrix} \tag{177}$$

$$\underset{\sim}{\overset{*}{Y}}^T = \begin{vmatrix} Y_1 & Y_2 & \cdots & Y_m & c_o \end{vmatrix} \tag{178}$$

$$\underset{\sim}{\overset{*}{t}}^T = \begin{vmatrix} t_1 & t_2 & \cdots & t_m \end{vmatrix} \tag{179}$$

results in

$$\underset{\sim}{\overset{*}{t}} = (\underset{\sim}{\overset{*}{S}})^{-1} \underset{\sim}{\overset{*}{Y}} \tag{180}$$

By noting that

$$\alpha_{kj} = 1/2\ S_{kj} \tag{181}$$

$$\beta_k = 1/2\ Y_k \tag{182}$$

and that

$$2h(\underset{\sim}{t}) = 0 \tag{183}$$

is an equivalent set of constraints, it is easily shown that

$$\begin{vmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \\ c_o \end{vmatrix} = \begin{vmatrix} \alpha_{11} & \cdots & \alpha_{1m} & h_{t_1} \\ \alpha_{21} & & \alpha_{2m} & h_{t_2} \\ \vdots & & \vdots & \vdots \\ \alpha_{m1} & & \alpha_{mm} & h_{t_m} \\ c_1 & \cdots & c_m & 0 \end{vmatrix} \cdot \begin{vmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \\ \lambda \end{vmatrix} \tag{184}$$

or

$$\underset{*}{\underline{t}} = (\underset{*}{\underline{\alpha}})^{-1} \underset{*}{\underline{\beta}} \tag{185}$$

so long as $\underset{*}{\underline{\alpha}}$ is invertable.

Case III: Multiple constraints on $\underline{t}$.

If there are $n \leq m$ constraints on $\underline{t}$, there will be n equations of the form

$$h_i(\underline{t}) = c_{i1}t_1 + \ldots + c_{im}t_m - c_{io} = 0 \tag{186}$$

For each constraint, a Lagrangian multiplier $\lambda_i$ is introduced, resulting in the following set of equations:

$$\Delta_{t_1} + \lambda_1 h_{t_1} + \ldots + \lambda_n h_{t_1} = 0 \tag{187}$$

$$\vdots$$

$$\Delta_{t_m} + \lambda_1 h_{t_m} + \ldots + \lambda_n h_{t_m} = 0 \tag{188}$$

$$h_1(\underline{t}) = 0 \tag{189}$$

$$\vdots$$

$$h_n(\underline{t}) = 0 \tag{190}$$

This gives (m + n) equations in the (m + n) unknowns: $\underline{t}$, $\lambda_1, \lambda_2, \ldots, \lambda_n$.

Extending the argument of Case II, it can be shown that

$$
\left[\begin{array}{c} \\ \underline{\beta} \\ \\ \\ \hline\hline \\ c_{10} \\ c_{20} \\ \vdots \\ c_{n0} \end{array}\right]
=
\left[\begin{array}{cccc|c} & & & & (h_1)_{t_1} \cdots (h_n)_{t_1} \\ & \underline{\alpha} & & & \vdots \quad\quad \vdots \\ & & & & (h_1)_{t_m} \cdots (h_n)_{t_m} \\ \hline\hline c_{11} & c_{12} & \cdots & c_{1m} & \\ & & & & \\ & & & & \underline{0} \\ & & & & \\ c_{n1} & c_{n2} & \cdots & c_{nm} & \end{array}\right]
\left[\begin{array}{c} \\ \underline{t} \\ \\ \\ \hline\hline \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{array}\right]
\tag{191}
$$

## Appendix C

## Comments on Programming

### Program Structure

In structuring this program, the primary goals were that the program would be understandable and modifiable by subsequent users. To permit this, the principles of modern software engineering have been followed as outlined in the text by Yourdan and Constantine (Ref 29).

At the start of this investigation, the program requirements were defined:

1. The program would be interactive. This would provide maximum user flexibility. Additionally, it would allow the program to be semi-educational, potentially useful in a course on digital filtering.

2. The program would design two-dimensional filters with arbitrary transition band contours. (It was later discovered that there is an admittable class of contours.)

3. The minimum output would include the array of 2-D impulse response coefficients and the frequency response of the filter. There would be an optional provision for a computer plot of the frequency response.

4. The program would be simple to use, not requiring a sophisticated knowledge of SCOPE procedures (Ref 32).

A high level structure chart of a first-cut design is shown in Fig. 29 Each of the modules represents an independent process.

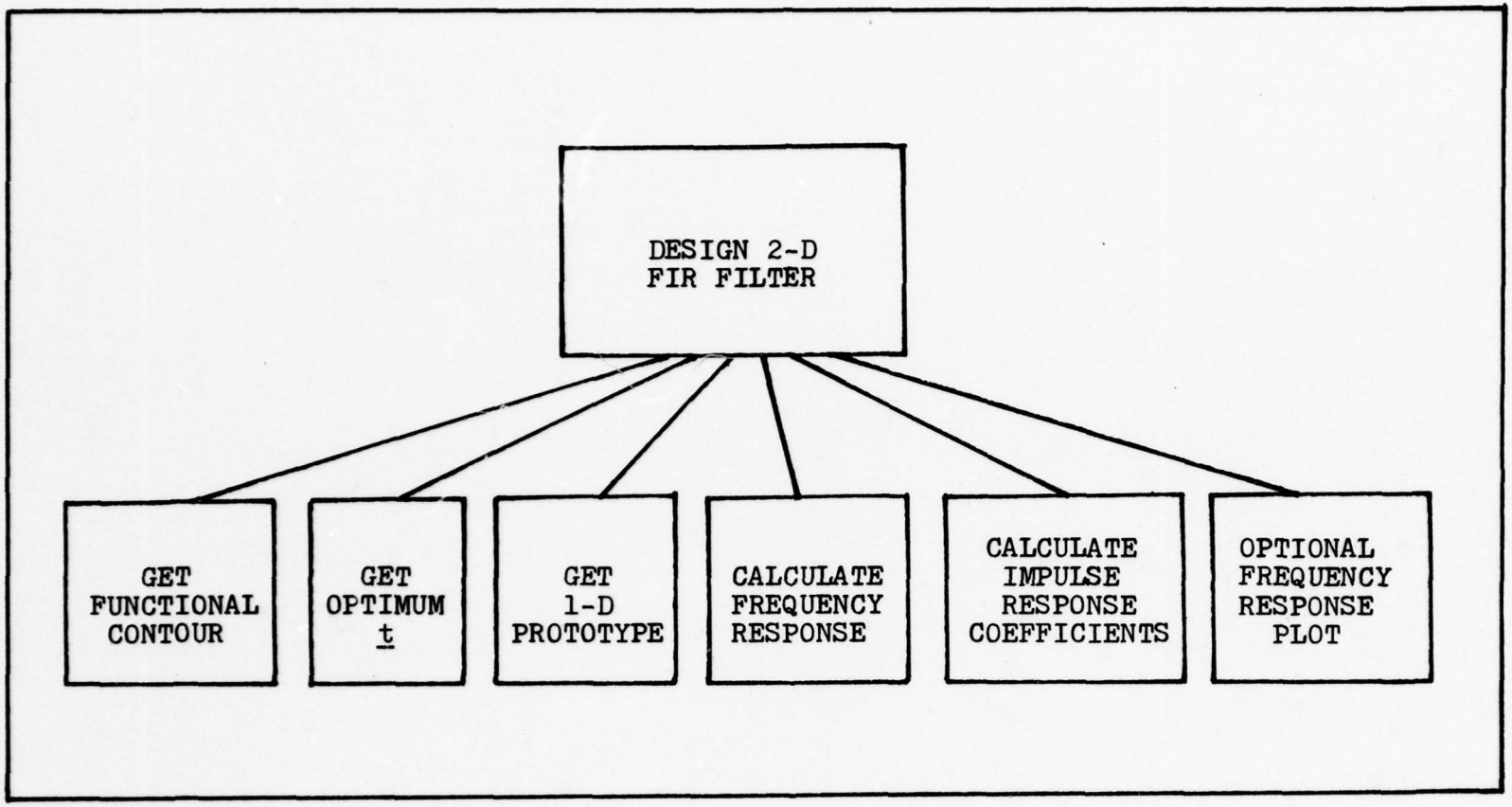



Fig. 29. First-Cut Structure Chart for Design Program

Considerations for Interactive Use

Because the program was required to be interactive, memory management had to be considered. Using Control Data Corporation INTERCOM, local procedures establish a maximum field length of 60K octal memory. This prevents a large program, such as this one, from being loaded. There are two methods available to circumvent this restriction; segmentation and overlays (Ref 30:1-12). Although segmentation is a more elegant method, overlays were selected to permit an eventual interface with a large package of computer-aided design programs concurrently being assembled (Ref 31). Overlays may extend three levels deep. To permit an interface with another program, one overlay level must be left for communication, leaving only two levels to work with.

The main level overlay has been used to serve as a controller for the design program. It corresponds to the top box in the structure chart of Fig. 29. The initial plan was to assign each of the other modules to a separate primary level overlay. Unfortunately, with this program division, the 60K restriction could not be met. Thus, several of the overlays have been artificially divided into two distinct primary overlays, with a great deal of additional control passing required. This compromise hurt the program structure by increasing the dependence of the modules, but it was necessary to permit interactive use.

Avoiding SCOPE Commands

Another troublesome aspect of the program involves the "Get Functional Contour" module of Fig. 29. The program was required to design filters with arbitrary transition band contours (within the admittable class of contours), thus it was impossible to develop a list of available functions for the user. The only realistic approach was to allow the user to create his own subroutine to describe the contour shape. This subroutine is trivial to write; however, integrating such a subroutine with the binary code of the design program is difficult. The subroutine must be compiled and libraried, a tedious task involving a good knowledge of SCOPE commands (Ref 32).

Fortunately, a method was found that relieves the user of any undue burden while still permitting him to supply the required subroutine. A "cataloged procedure", written using University of Washington Control Language (Ref 33), performs all the functions associated with compiling and librarying the subroutine, attaching and executing the design program (MCLTRN), and routing a hardcopy of the output listing and plots. The user creates his subroutine, saves it as a local file, and then issues a set of simple commands. The nature of the subroutine and the required procedure commands are outlined in detail in Appendix D. A listing of the cataloged procedure (PROFIL) is attached to this appendix.

## Program Flowchart

This program was coded with the concepts of structured programming in mind. Unconditional branching has been avoided whenever a reasonable alternative existed. Subroutine calls are made very freely; however, most subroutines are short and functional. As a rule of thumb, modules contain between 10 and 50 lines of code. For these reasons, detailed flowcharts for each module are unnecessary. Fig. 30 shows a top-level flowchart for the design program. The flowchart has been deliberately held to one page, showing only the detail required for a quick overall view of the program.

## Program Source Code

The design program source code, PROGRAM MCLTRN, (Ref 35) involves over 1600 statements, so it has not been included in this report. Copies of the source listing are available from the author or from Prof. Gary Lamont, AFIT/EN.

## Program PROFIL

The cataloged procedure PROFIL controls Program MCLTRN. A listing follows:

Program PROFIL

```
COMPILE(NAME=X)
RETURN,GOGO.
REWIND,NAME.
FTN,I=NAME,L=Ø,B=GOGO.
EDITLIB,I=DA,L=LIST.
RETURN,LIST,DA.
LIBRARY,MYLIB.
/DATA DA
LIBRARY(MYLIB,NEW)
ADD(*,GOGO)
FINISH.
ENDRUN.
RECOMPILE(NAME=X)
RETURN,GOGO.
REWIND,NAME.
FTN,I=NAME,L=Ø,B=GOGO.
EDITLIB,I=DB,L=LIST.
RETURN,LIST,DB.
LIBRARY,MYLIB.
/DATA DB
LIBRARY(MYLIB ,OLD)
DELETE(*)
ADD(*,GOGO)
FINISH.
ENDRUN.
DESIGN
IF(-FILE,AFITSUB)
ATTACH,AFITSUBROUTINES,ID=AFIT.
LIBRARY,AFITSUB,MYLIB.
IF(-FILE,MX)
ATTACH,MX,THESIS,CY=3Ø2,ID=RBB.
REWIND,MX.
MX.
RETURN,MCL.
ROUTE(TERM=BB,USER=KJØ)
REWIND,RESULT.
COPYSBF,RESULT,RRX.
ROUTE,RRX,DC=PR,TID=TERM,FID=USER.
RETURN,RESULT.
IF(FILE,PLOT)
ROUTE,PLOT,DC=PT,TID=TERM,FID=USER.
FILES.
PLFILE
REWIND,TAPE2.
REQUEST,TEMP3D,*PF.
COPY,TAPE2,TEMP3D.
CATALOG,TEMP3D,DATA3DPLOT.
RETURN,TEMP3D.
IF(-FILE,PLOT3D)
ATTACH,PLOT3D,THESIS,CY=4ØØ,ID=RBB.
REWIND,PLOT3D.
BATCH,PLOT3D,INPUT,HERE.
FILES.
3DPLOT
IF(-FILE,PLFILE)
ATTACH,PLFILE,YOURFILE.
IF(-FILE,DISSPLA)
ATTACH,DISSPLA,ID=X654321.
LIBRARY,DISSPLA.
ONLINE.
```

NOTE: Separate procedures with end-of-record.

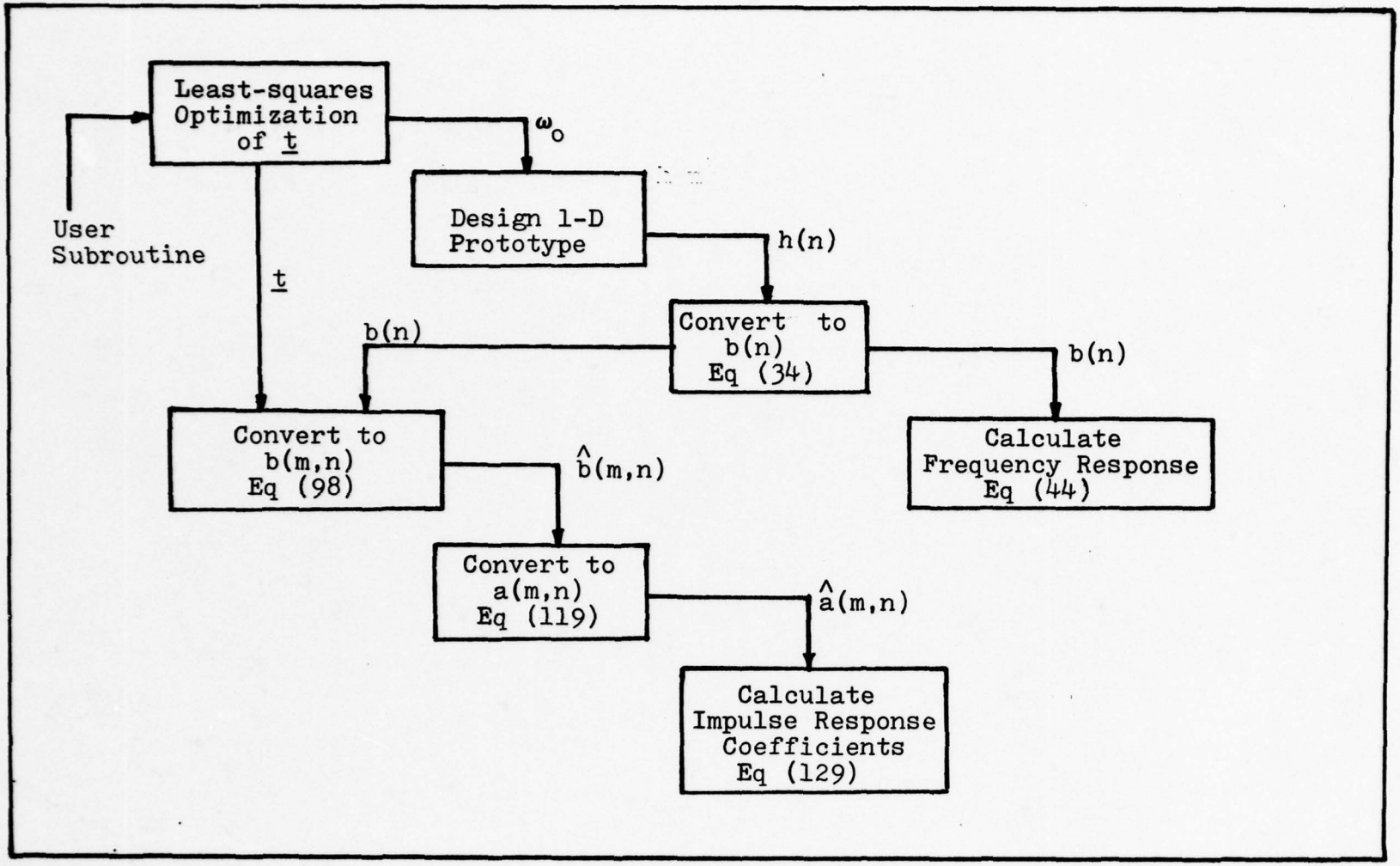



Fig. 30. Top-Level Program Flowchart

## Appendix D

## User's Guide to Program MCLTRN

MCLTRN
15 December 1977

Identification:

MCLTRN - McClellan Transformation Design for Two-Dimensional (2-D) Digital Filters

FORTRAN Extended (FTN) Program

Air Force Institute of Technology

Wright-Patterson Air Force Base, Ohio

Capt. R. B. Brown, GE-77D

Purpose:

MCLTRN will design 2-D finite impulse response (FIR) zero-phase filters of the form

$$G(\omega_1,\omega_2) = \sum_{m=0}^{N} \sum_{n=0}^{N} a(m,n)\cos\omega_1 m \cos\omega_2 n \qquad (45)$$

The filter will be a very close approximation to an optimal filter. Program output includes a 2-D frequency response (optionally plotted), the 2-D impulse response coefficients, and contour maps. MCLTRN is intended for interactive use.

Control:

ATTACH, PROFIL, ID = T77Ø314.

A cataloged procedure (Ref 33) PROFIL is used to control MCLTRN. There are six commands:

BEGIN, COMPILE.
BEGIN, RECOMPILE.
BEGIN, DESIGN.

```
BEGIN, ROUTE.
BEGIN, PLFILE.
BEGIN, 3DPLOT.
```

Programming Information:

1. MCLTRN requires a user supplied FORTRAN SUBROUTINE FUNCT(I, N, X, Y), where X, Y are the spatial axes, N is the number of sample points (supplied by MCLTRN), and I is the current sample index (supplied by MCLTRN). The subroutine defines a smooth monotonic contour in the 2-D region $[0,1] \times [0,1]$. MCLTRN will scale this contour to the region $[0,\pi] \times [0,\pi]$.

e. g. To describe a quarter circle, centered at (0,0) with radius 0.5:

```
      SUBROUTINE FUNCT(I, N, X, Y)
C DEFINE VALID X RANGE ON [0,1] AND INCREMENT X
      XMAX = 0.5
      X = XMAX * (I - 1.0) / N
C CALCULATE CORRESPONDING Y VALUE
      Y = SQRT (XMAX**2 - X**2)
      RETURN
      END
```

2. After creating FUNCT (I, N, X, Y), the edit file must be saved, without sequencing, as local file X:

```
SAVE, X, NOSEQ, OVER.
```

To automatically compile and library the subroutine, command:

```
BEGIN, COMPILE.
```

If there are FORTRAN errors, or if it is necessary to compile another subroutine, use the alternate command:

```
BEGIN, RECOMPILE.
```

(If error messages related to your user library MYLIB appear, check that you saved file X and then attempt to use the recompile command.)

3. To execute MCLTRN, command:

```
BEGIN, DESIGN.
```

4. To obtain a hardcopy of the output listing (RESULT) or contour plots (PLOT), command:

```
BEGIN, ROUTE (, ,XX, YYYYY).
```

where the grouping in parenthesis is optional:

XX = TID code (optional default = BB)

YYYYY = FID code (optional default = USER)

Note: Plots will always have the user identification code as the banner.

5. To create a DISSPLA file for the 3-D drawing of the frequency response, command:

```
BEGIN, PLFILE.
```

Because DISSPLA (Ref 34) requires over 100K memory, the job will be batched to the input queue. After allowing time for execution, check your output file JJJucxx for errors (uc is your user identification code):

```
BATCH, JJJucxx, LOCAL.

EDIT, JJJucxx, SEQ.
```

Assuming there were no FORTRAN errors*, you may create (or preview) and dispose a Calcomp plot file by commanding:

```
BEGIN, 3DPLOT.

DRAW = 1-END$
```

If you are at a graphics terminal, command:

```
TEK4010
TEK4014        as appropriate
etc.
```

To dispose your plot, command:

```
ROUTE, PLOT, TID = XX, FID = YYYYY, DC = PT.
```

*In particular, FORTRAN error 103: No permanent file space for plot file -- job killed.

Method:

MCLTRN maps each frequency of a 1-D bandpass filter to a contour in the spatial plane by the McClellan Transformation:

$$\cos\omega = t_1 + t_2 \cos(x) + t_3 \cos(y) + t_4 \cos(x) \cos(y) \quad (192)$$

In particular, the frequency $\omega_o$ (usually a transition band boundary) is mapped to the contour defined by SUBROUTINE FUNCT (I, N, X, Y). MCLTRN optimizes $\underline{t}$ by the least-squares criteria. For a non-trivial mapping, constraints on $\underline{t}$ are necessary; however, a default constraint that $\emptyset$ maps to $(\emptyset,\emptyset)$ is available. An optional contour scaling algorithm returning a new 1-D frequency $\omega_o$, is recommended to guarantee a well-behaved frequency response. The user may examine the locus of contours generated by $\underline{t}$ as a function of $\omega$, in both tabular or plotted form.

Once $\underline{t}$ is established, a 1-D prototype optimal FIR zero-phase filter is designed. This filter is normally a low-pass or high-pass filter with a stopband or passband frequency of $\omega_o$. At this writing, the filter must be of odd order not exceeding 21. The error magnitude in the 1-D prototype will be passed to the 2-D filter, where each 1-D extremal frequency maps to a 2-D contour. A plot file of this mapping is created automatically by MCLTRN.

Finally, MCLTRN calculates the 2-D frequency response. Optionally, the 2-D impulse response coefficients will be calculated.

Special Conventions:

1. The 2-D filter designed using the McClellan Transformation is symmetric about both the x and y axes. Thus, only a first quadrant design is considered.

2. All frequencies are normalized to the interval $[0,\pi]$ and must be entered as multiples of $\pi$ ($\emptyset.\emptyset$ to $1.\emptyset$).

Design Example:

The following example illustrates the use of PROFIL and MCLTRN. The desired contour is defined by the rectangular hyperbola

$$xy = a^2/2 \tag{193}$$

where $a = 0.30\pi$. The resulting contour mapping is shown in Fig. 31. The frequency response is shown in Fig. 32. Note that user entries are shown in lower-case letters, programming prompts in upper-case letters.

```
ENTER PASSWORD-
ibm
ENTER 3-DIGIT TERMINAL ID-
.
124
11:03:77   LOGGED IN AT  19.41.01.
           WITH USER-ID KJ
           EQUIP PORT 15:003
COMMAND-
editor
..
attach,profil,id=t770314.
 PFN IS
 PROFIL
 PF CYCLE NO. = 001
..
create,sup
ENTER LINES

;subroutine funct(i,n,x,y)
;a=.30
;xmin=.5*a**2
;x=((1.0-xmin)*(i-1.0)/n + xmin)
;y=xmin/x
;return
;end
*
..
l,a
   100=      SUBROUTINE FUNCT(I,N,X,Y)
   110=      A=.30
   120=      XMIN=.5*A**2
   130=      X=((1.0-XMIN)*(I-1.0)/N + XMIN)
   140=      Y=XMIN/X
   150=      RETURN
   160=      END
..

save,x,over,nos
..

begin,compile.
      .056 CP SECONDS COMPILATION TIME
..

begin,design.
```

```
LFN IS
AFITSUB:
PF CYCLE NO. = 022

        ----------------------------------------
        PROGRAM FOR DESIGN OF 2-D FIR FILTERS
        ----------------------------------------


  REMEMBER: ENTER ALL FREQUENCIES AS ANGLES ON UNITS
  CIRCLE, 0.0 TO 1.0 (X PI)


        -----DETERMINE CONTOUR MAPPING VECTOR T-----

 IF YOU HAVE A PREDETERMINED MAPPING VECTOR T, TYPE 1

0


  ENTER ESTIMATED 1-D PROTOPTYPE CUTOFF FREQUENCY,
  NORMALIZED 0.0 TO 1.0 (X PI)

.45

  THE UNCONSTRAINED MCCLELLAN TRANSFORMATION MAY
  RESULT IN A TRIVIAL (AND USELESS) MAPPING.
  THE DEFAULT CONSTRAINT THAT (0) MAPS TO (0,0)
  IS AVAILABLE BY TYPING 1;  IF YOU WANT TO SUPPLY
  SOME OTHER CONSTRAINTS, TYPE 0

1
```



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      INITIAL TRANSFORM VECTOR
 T1 =    -15.07222063      T2 =     15.82563857
 T3 =     15.84240246      T4 =    -15.59582040

WITH THESE T-VALUES, THE MEAN ERROR IS   .027152404

THERE IS NO GUARANTEE THAT YOUR TRANSFORMATION IS
WELL-DEFINED OVER THE REGION (0,PI) X (0,PI); HOWEVER, A
SCALING ALGORITHM WILL USUALLY PRODUCE A WELL-DEFINED
MAPPING. ADDITIONALLY, IT WILL LOCATE A 'BEST' 1-D
PROTOTYPE FREQUENCY FOR THE CONTOUR OF INTEREST.


IF YOU WANT SCALING, TYPE 1;  OTHERWISE TYPE 0

;

      SCALED TRANSFORM VECTOR
 T1 =       .49247822      T2 =       .49973532
 T3 =       .50026468      T4 =      -.49247822

  RECOMMENDED CUTOFF FREQUENCY FOR 1-D PROTOTYPE
    W0' =    .073635   NORMALIZING TO 1.0


WITH THESE T-VALUES, THE MEAN ERROR IS   .000857407

FOR TABULAR DATA ON THE CONTOUR APPROXIMATION

TYPE 1; TO GO ON TYPE 0

1

ENTER A NORMALIZED 1-D VALUE TO BE MAPPED (0+ TO 1.0)
OR ENTER 0 TO GO ON WITH DESIGN

.073636
          U1              U2
        .0480           .9456
        .1432           .3170
        .2384           .1923
        .3336           .1364
        .4288           .1048
        .5240           .0845
        .6192           .0707
        .7144           .0611
        .8096           .0547
        .9048           .0510
       1.0000           .0498

ENTER A NORMALIZED 1-D VALUE TO BE MAPPED (0+ TO 1.0)
OR ENTER 0 TO GO ON WITH DESIGN

.10
          U1              U2
        .0840           .9104
        .1756           .3663
        .2672           .2403
        .3588           .1796
        .4504           .1443
        .5420           .1216
        .6336           .1063
        .7252           .0959
        .8168           .0892
        .9084           .0853
       1.0000           .0841

ENTER A NORMALIZED 1-D VALUE TO BE MAPPED (0+ TO 1.0)
OR ENTER 0 TO GO ON WITH DESIGN

.20
          U1              U2
        .1940           .9173
        .2746           .5150
        .3552           .3890
        .4358           .3188
        .5164           .2743
        .5970           .2442
        .6776           .2234
        .7582           .2091
        .8388           .1998
        .9194           .1944
       1.0000           .1927

ENTER A NORMALIZED 1-D VALUE TO BE MAPPED (0+ TO 1.0)
OR ENTER 0 TO GO ON WITH DESIGN

0
```



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
TO CREATE A PLOT FILE, TYPE 1

  ENTER A VERBAL DESCRIPTION OF THE CONTOURS TO
  BE USED (50 CHARACTERS MAXIMUM)

rectangular hyperbola with a = .30 (x pi)

      SUMMARY FROM THE LEAST SQUARES FIT SUBROUTINE

-----------------------------------------------------------------

            W0 =     .45000
            W0 =     .07363
            T1 =     .49243
            T2 =     .49974
            T3 =     .50026
            T4 =    -.49243

            MEAN ERROR =    .00086

RECTANGULAR HYPERBOLA WITH A = .30 (X PI)

   CONTRAINTS ON T-VECTOR:
    1.00(T1) + 1.00(T2) + 1.00(T3) + 1.00(T4) = 1.00
   WITH ROUTINE-SUPPLIED SCALING

-----------------------------------------------------------------


DECIDE ON ONE OF THE FOLLOWING OPTIONS BY NUMBER:
   1) REPEAT CONTOUR APPROXIMATION FOR NEW T VECTOR
   2) CONTINUE THE PROBLEM BY DESIGNING 1-D PROTOTYPE
   3) STOP THE DESIGN TO EXAMINE OUTPUT OR PLOTS

2

-----DETERMINE BEST 1-D PROTOTYPE FILTER-----

   ENTER AN ODD FILTER ORDER 21 OR LESS
   (EVEN ORDERS LACK SYMMETRY REQUIRED FOR TRANSFORM)

15

   HOW MANY TRANSITION BANDS THIS FILTER?

1

   ENTER THE BAND EDGES FOR EACH TRANSITION BAND.
   RECALL THAT THE BEST CUTOFF FREQUENCY WAS  .073635

.073635,.20

   ENTER AN IDEAL ABSOLUTE MAGNITUDE FOR EACH BAND
   OF THIS PROTOTYPE FILTER (USUALLY 1 OR 0)

1, 0

   ENTER AN ERROR WEIGHTING (RELATIVE TO 1.0)
   FOR THE RIPPLE IN EACH BAND OF THE PROTOTYPE

1,1
```



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
ONE-DIMENSIONAL FIR PROTOTYPE  (ZERO PHASE)

          FILTER LENGTH =  15

                        BAND  1        BAND  2        BAND
LOWER BAND EDGE      0.000000000     .200000000
UPPER BAND EDGE       .073635000    1.000000000
DESIRED VALUE        1.000000000    0.000000000
WEIGHTING            1.000000000    1.000000000
DEVIATION             .097813975     .097813975

DEVIATION IN DB  -20.191981862  -20.191981862

EXTREMAL FREQUENCIES
   0.0000000     .0736350     .2000000     .2781250     .4109375
    .5593750     .7078125     .3562500    1.0000000

IF YOU ARE NOT SATISFIED WITH THE PROTOTYPE, TYPE 1
IF THE PROTOTYPE IS ACCEPTABLE, TYPE 0

1

  ENTER ONE OF THE FOLLOWING OPTION NUMBERS
    1) CHANGE EVERYTHING
    2) CHANGE ONLY THE FILTER ORDER
    3) CHANGE ONLY THE TRANSITION BAND EDGES
    4) CHANGE ONLY THE ERROR WEIGHTING

2

  ENTER AN ODD FILTER ORDER 21 OR LESS
  (EVEN ORDERS LACK SYMMETRY REQUIRED FOR TRANSFORM)

19
```

```
ONE-DIMENSIONAL FIR PROTOTYPE  (ZERO PHASE)

          FILTER LENGTH =  19

                        BAND  1        BAND  2        BAND
LOWER BAND EDGE      0.000000000     .200000000
UPPER BAND EDGE       .073635000    1.000000000
DESIRED VALUE        1.000000000    0.000000000
WEIGHTING            1.000000000    1.000000000
DEVIATION             .063757791     .063757791

DEVIATION IN DB  -23.909334755  -23.909334755

EXTREMAL FREQUENCIES
   0.0000000     .0736350     .2000000     .2500000     .3437500
    .4500000     .5625000     .6687500     .7812500     .8875000
   1.0000000

IF YOU ARE NOT SATISFIED WITH THE PROTOTYPE, TYPE 1
IF THE PROTOTYPE IS ACCEPTABLE, TYPE 0

1

1

  ENTER ONE OF THE FOLLOWING OPTION NUMBERS
    1) CHANGE EVERYTHING
    2) CHANGE ONLY THE FILTER ORDER
    3) CHANGE ONLY THE TRANSITION BAND EDGES
    4) CHANGE ONLY THE ERROR WEIGHTING

4

  ENTER AN ERROR WEIGHTING (RELATIVE TO 1.0)
  FOR THE RIPPLE IN EACH BAND OF THE PROTOTYPE

1,2
```



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
**********************************************************************
   ONE-DIMENSIONAL FIR PROTOTYPE  (ZERO PHASE)

            FILTER LENGTH =  19

                        BAND  1          BAND  2          BAND
 LOWER BAND EDGE     0.000000000       .200000000
 UPPER BAND EDGE      .073635000      1.000000000
 DESIRED VALUE       1.000000000      0.000000000
 WEIGHTING           1.000000000      2.000000000
 DEVIATION            .097073283       .048539142

 DEVIATION IN DB  -20.257558258  -26.278158171

 EXTREMAL FREQUENCIES
    0.0000000     .0736350     .2000000     .2500000     .3437500
     .4500000     .5562500     .6687500     .7812500     .8875000
    1.0000000

**********************************************************************

 IF YOU ARE NOT SATISFIED WITH THE PROTOTYPE, TYPE 1
 IF THE PROTOTYPE IS ACCEPTABLE, TYPE 0

 0
```



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

19 X 19 POINT FILTER

MAGNITUDES OF 2-D FREQUENCY RESPONSE
FREQUENCIES NORMALIZED 0.0 TO 1.0 (X PI)
ORIGIN U2-AXIS PI

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ! | 1.10 | 1.09 | 1.09 | 1.07 | 1.06 | 1.04 | 1.02 | 1.01 | 1.00 | .99 | .98 |
| ! | 1.09 | 1.08 | 1.05 | 1.00 | .93 | .86 | .79 | .73 | .68 | .65 | .63 |
| ! | 1.09 | 1.05 | .95 | .78 | .58 | .39 | .24 | .13 | .06 | .03 | .02 |
| ! | 1.08 | 1.00 | .78 | .47 | .18 | .00 | -.05 | -.03 | -.01 | .01 | .02 |
| ! | 1.06 | .94 | .59 | .18 | -.03 | -.03 | .04 | .05 | .02 | -.00 | -.01 |
| ! | 1.04 | .87 | .39 | .00 | -.03 | .05 | .01 | -.05 | -.04 | -.01 | -.00 |
| ! | 1.03 | .79 | .24 | -.05 | .04 | .01 | -.05 | .02 | .05 | .03 | .02 |
| ! | 1.01 | .73 | .13 | -.03 | .05 | -.05 | .02 | .03 | -.04 | -.04 | -.03 |
| ! | 1.00 | .66 | .06 | -.01 | .02 | -.04 | .05 | -.04 | -.00 | .05 | .04 |
| ! | .99 | .65 | .03 | .01 | -.00 | -.01 | .03 | -.04 | .05 | -.03 | -.05 |
| ! | .99 | .64 | .02 | .02 | -.01 | -.00 | .02 | -.03 | .04 | -.05 | .05 |

U1-AXIS

DO YOU WANT TO CREATE A DISSPLA 3-D PLOT OF FREQ RESP?
IF YES-- TYPE 1

1

(THE COEFFICIENTS FOR DISSPLA ARE NOW ON TAPE2)

YOU MAY NEXT CALCULATE THE IMPULSE RESPONSE COEFFICIENTS.
HOWEVER, THIS IS TIME CONSUMING AND NOT NECESSARY
FOR JUST A FREQUENCY RESPONSE
IF YOU WANT TO CALCULATE THE IMPULSE COEFFICIENTS, TYPE 1

1



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

(after allowing time for execution)

```
NOTE:  THE IMPULSE RESPONSE COEFFCIENTS ARE
BEING WRITTEN TO YOUR HARDCOPY OUTPUT FILE (RESULT)
REMEMBER TO USE BEGIN,ROUTE(, ,XX,YYYYY) COMMAND.
    (SEE USERS GUIDE FOR DETAILS.)


AT THIS POINT DO ONE OF THE FOLLOWING:
CREATE NEW SUBROUTINE FUNCT---BEGIN,RECOMPILE.
   (AFTER CREATING AND SAVING)

ROUTE OUTPUT AND PLOTS--------BEGIN,ROUTE, ,(TID),(FID)
REPEAT THE DESIGN-------------BEGIN,DESIGN.
CREATE A DISSPLA PLOT FILE FOR 3-D FREQ RESPONSE PLOT
                  -------------BEGIN,PLFILE.
   (SEE USERS GUIDE FOR COMPLETE DETAILS)


    STOP    NCPMAL END
    11.269 CP SECONDS EXECUTION TIME
..

begin,route, ,bb,brown.
--LOCAL FILES--
 *AFITSUB    MYLIB     $INPUT     $OUTPUT    *PROFIL
 *MX         TAPE2
--REMOTE OUTPUT FILES--
  PLFIL7Y    JJJKJ7Y
..

begin,plfile.
--LOCAL FILES--
 *PLOT3D    *AFITSUB    MYLIB     $INPUT     $OUTPUT
 *PROFIL    *MX         TAPE2
--REMOTE INPUT FILES--
  JJJKJ8R
--REMOTE OUTPUT FILES--
  PLFIL7Y    JJJKJ7Y
 NEWCYCLE CATALOG
 RP = 002 DAYS
 CT ID=  T770314 PFN=DATA3DPLOT
 CT CY= 004  00000192 WORDS.!
..
```

```
FILES
--LOCAL FILES--
  PLFILE    $INPUT     $OUTPUT    *PROFIL
--REMOTE OUTPUT FILES--
  JJJKJPD

BATCH,JJJKJPD,LOCAL.
..
EDIT,JJJKJPD,SEQ
LINES > CH - YES=TRUNCATE LINES, NO=ABORT EDIT

Y
..
L
  1640= 18.15.19.EJ  END OF JOB, KJ  T770314.

..
L,1500,L
  1500= 18.15.18.ROUTE,PLFILE,TID=KJ,FID=PLFIL,DC=PR.
  1510= 18.15.18.OF  00001088 WORDS - FILE PLFILE , DC 40
  1520= 18.15.19.PURGE,TAPE2.
  1530= 18.15.19.PR ID=  T770314 PFN=DATA3DPLOT
  1540= 18.15.19.PR CY= 004  00000192 WORDS.
  1550= 18.15.19.OP  00000576 WORDS - FILE OUTPUT , DC 40
  1560= 18.15.19.MS   17920  WORDS  (   17920 MAX USED)
  1570= 18.15.19.SCM 110000  WORDS MAXIMUM
  1580= 18.15.19.CPU        4.450 SEC.          1.931 ADJ.
  1590= 18.15.19.IO         3.298 SEC.          1.650 ADJ.
  1600= 18.15.19.CM       176.193 KWS.          1.409 ADJ.
  1610= 18.15.19.CRUS                           4.991

  1620= 18.15.19.COST                  $          .29
  1630= 18.15.19.PP         10.341 SEC.      DATE 11/07/77
  1640= 18.15.19.EJ  END OF JOB, KJ  T770314.

..
BEGIN,3DPLOT.
 PFN IS
 DISSPLA:
 PF CYCLE NO. = 003
DISSPLA POSTPROCESSOR FOR ONLINE CALCOMP PLOTTER.
ENTER DIRECTIVES.

DRAW=1-END$

ROUTE,PLOT,TID=BB,FID=BROWN,DC=PT
```



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

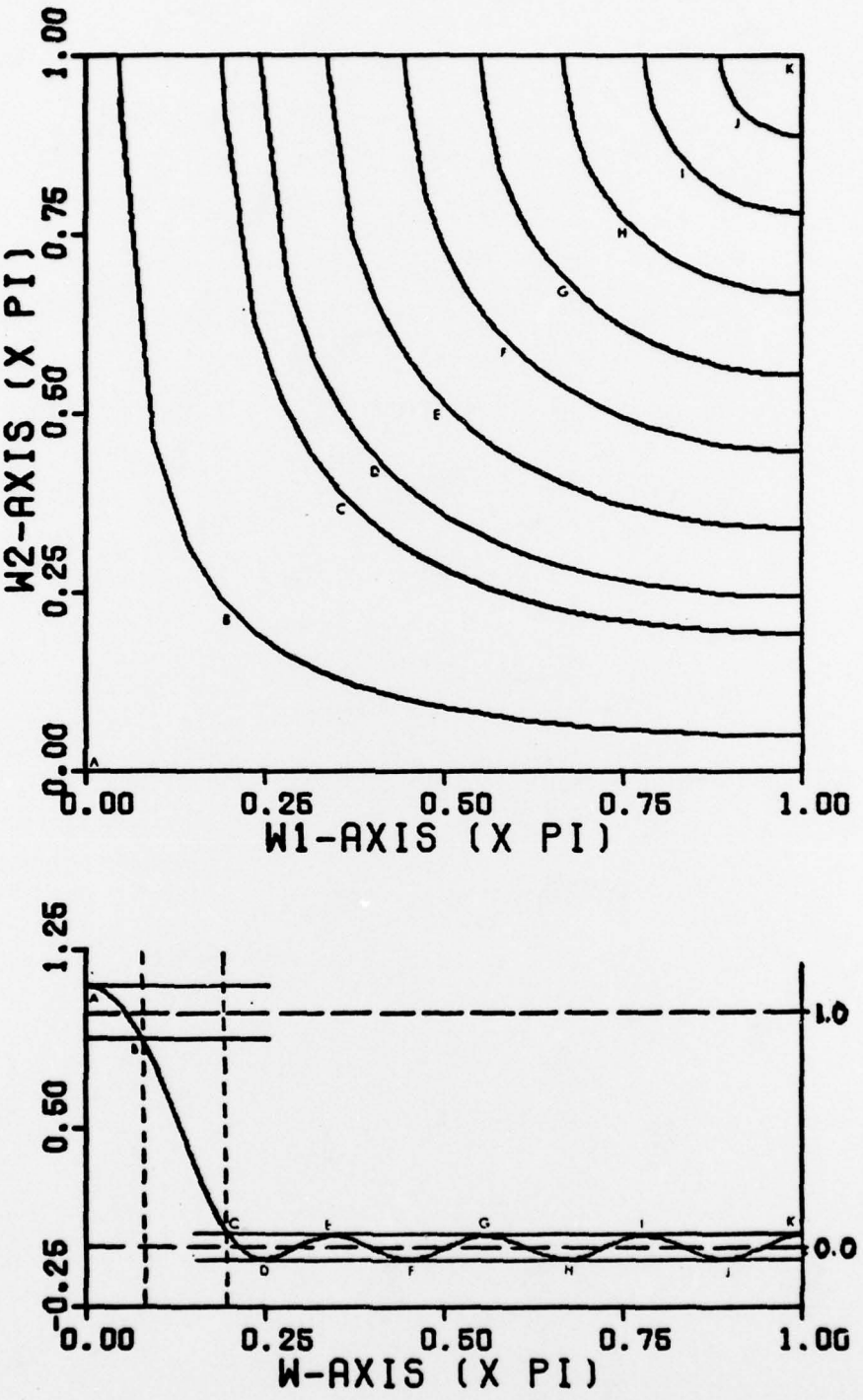



Fig. 31. Hyperbolic Contours. $\omega_o = .0736\pi$, $\underline{t}^T = [.4925, .4997, .5003, -.4925]$, $\bar{e} = .0009$, $\omega_p = .0736\pi$, $\omega_s = .20\pi$, $\delta_1 = .0971$, $\delta_2 = .0485$

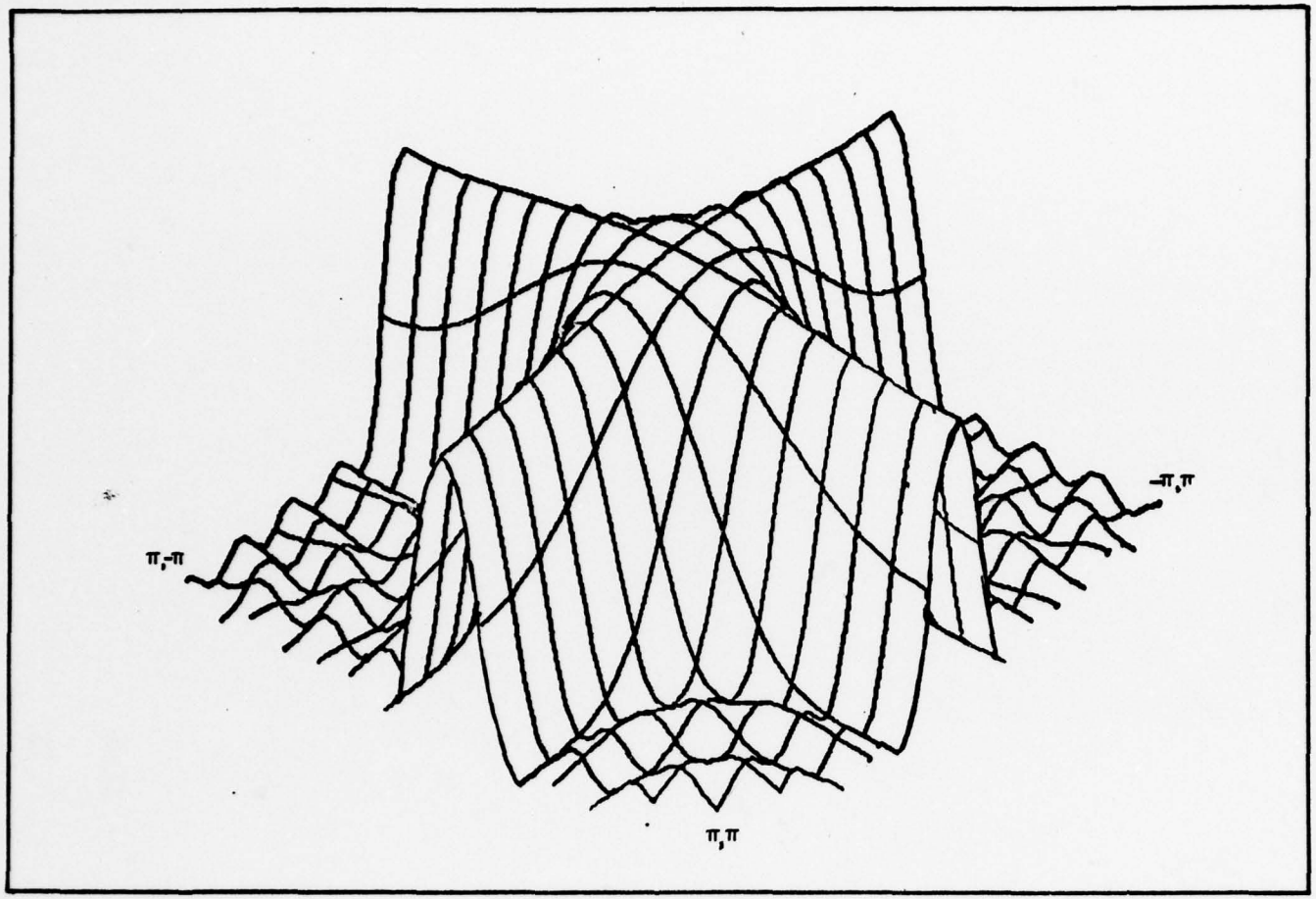



Fig. 32. Frequency Response of Hyperbolic-Shaped Filter. The specifications are the same as Fig. 31.

## Appendix E

### Use of Advanced DISSPLA

DISSPLA is a software graphics package developed by Integrated Software Systems Corporation. The DISSPLA Reference Manual (Ref 34) is augmented locally by preliminary and post-processing instructions available from the ASD Computer Center.

One of the features of Advanced DISSPLA is a capability for projecting three-dimensional (3-D) surfaces and lines (Ref 34:Adv-C). 3-D surfaces are plotted with "hidden lines" automatically removed. Any viewpoint not actually within the surface being drawn is allowed.

The basic plotting subroutine for 3-D surfaces is CALL SURFUN(ZFUN, IXPTS, XDELTA, IYPTS, YDELTA, WORK), where

ZFUN is the function name for a two-variable function to be evaluated, producing a magnitude z.

IXPTS is the number of sample points to be evaluated in each XDELTA interval.

IYPTS is analogous to IXPTS, but over the YDELTA interval.

XDELTA is the interval between lines parallel to the x-axis. The smaller this interval, the greater the detail.

YDELTA is the interval between lines parallel to the y-axis.

WORK is an internal array, dimensioned at least equal to the number of perimeter sample points plus 4.

The limits on x, y, and z are specified by calls to GRAF3D and AXES3D. For information on these and other calls, the DISSPLA Advanced Reference Manual, Part C, should be reviewed.

A source listing of the FORTRAN program used to generate the 3-D frequency response drawings of this investigation has been included as an illustration of 3-D DISSPLA. Function G evaluates Eq (44) over a grid of points in x-y. Tape 2 contains t, and the coefficients {b(n)}.

```
JJJ,T20,CM120000.T770314,MCLTRN USER
FTN(R=0)
ATTACH,TAPE2,DATA3DPLOT.
PERIND,TAPE2.
ATTACH,DISSPLA,ID=X654321.
LIBRARY,DISSPLA.
REQUEST,PLFILE,*PF.
LGO.
CATALOG,PLFILE,YOURFILE.
PURGE,TAPE2.
*EOR
      PROGRAM PLT3D(INPUT,OUTPUT,TAPE2,PLFILE=0)

      DIMENSION WORK(650),B(11),T(4)
      COMMON /ONE/ B,NP,T
      EXTERNAL G
      READ(2,11) NP,T
11    FORMAT(I3,4E18.9)
      READ(2,12)(B(M),M=1,NP)
12    FORMAT(5E18.9/5E12.9/E18.9)
      PRINT 12,T
      PRINT 12,(B(M),M=1,NP)
      CALL COMPRS
      CALL BGNPL(1)
      CALL TITL3D(1H ,-1,7.0,6.0)

      CALL UUMBS(12.,10.,12.)
      CALL AXES3D(0,0,0,0,0,0,2.0,2.0,1.5)
      CALL GRAF3D(-1.0,0.1,1.0,-1.0,0.1,1.0,-.25,0.1,1.25)
      CALL SURFUN(G, 3,.125, 3,.125, WORK)
      CALL ENDPL(1)
      CALL DONEPL
      STOP
      END

      FUNCTION G(U1,U2)
      COMMON /ONE/ B,NP,T
      DIMENSION B(11),T(4)

      PI=3.1415926536
      UU1=U1*PI
      UU2=U2*PI
      X=COS(UU1)
      Y=COS(UU2)
      Z=T(4)*X*Y
      X=T(2)*X
      Y=T(3)*Y
      G=0.00
      DO 30 M=1,NP
30    G=G + B(M)*((T(1)+X+Y+Z)**(M-1))
      RETURN
```

## Vita

Captain Richard Blackstock Brown was born in Minneapolis, Minnesota in 1948. He attended Lawrence University and the University of Minnesota, graduating in 1971 with a Bachelor's degree in Electrical Engineering. He was commissioned through Air Force ROTC. After completing flight training, he served four years as a T-37 instructor pilot, and then came directly to the Air Force Institute of Technology. While at AFIT, he received a regular commission. Upon graduation, he will be assigned to the Avionics Laboratory, Wright-Patterson Air Force Base, Ohio.

Permanent address: c/o Charles Chandler
4421 Barbados
Wichita Falls, Texas 76308

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>AFIT/GE/EE/77-10 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)*<br>INVESTIGATION OF OPTIMAL LINEAR SHIFT-INVARIANT TWO-DIMENSIONAL DIGITAL FILTERS. | | 5. TYPE OF REPORT & PERIOD COVERED<br>MS Thesis<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Richard B. Brown<br>Capt. USAF | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Air Force Institute of Technology (AFIT/EN)<br>Wright-Patterson AFB, Ohio 45433 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Air Force Institute of Technology (AFIT/EN)<br>Wright-Patterson AFB, Ohio 45433 | | 12. REPORT DATE<br>December 1977<br>13. NUMBER OF PAGES<br>134 |
| 14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)* | | 15. SECURITY CLASS. *(of this report)*<br>UNCLASSIFIED<br>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

Approved for public release; IAW AFR 190-17

JERRAL F. GUESS, Captain, USAF
Director of Information

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Digital Filters
Two-Dimensional
Optimal
Computer Program
McClellan Transformation

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

This investigation develops an interactive computer-aided method for designing lowpass or highpass two-dimensional finite impulse response digital filters. Filters designed using this method will have linear phase and an equiripple error in the transmission and attenuation bands. The essence of the method is that it transforms an optimal one-dimensional digital filter into a close approximation of an optimal two-dimensional digital filter. The amplitude characteristic of the one-

→ next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. Abstract (Continued).
dimensional filter is preserved in the sense that each point of the one-dimensional frequency response is mapped to a contour in the two-dimensional plane. This transformation was first proposed by James H. McClellan, and is now called McClellan Transformation. By controlling the mapping of a specified one-dimensional frequency to a desired contour shape in the plane, two-dimensional filters of fairly arbitrary specifications can be designed; that is, their frequency can be determined, and the associated two-dimensional impulse response coefficients calculated.