A DESIGN TECHNIQUE FOR APPROXIMATELY CIRCULARLY SYMMETRIC LOW-PASS 2-D RECURSIVE FILTER

CHIU-LUN YU

A MAJOR TECHNICAL REPORT

in

The Faculty

of

Engineering

Presented in Partial Fulfillment of the requirements
for the Master of Engineering Degree at
Concordia University
Montreal, Canada
June, 1980

ABSTRACT

A technique is presented for designing stable two-dimensional (2-D) recursive filters whose magnitude response is approximately circularly symmetric. This is achieved by cascading two individual elementary one-dimensional (1-D) filters and they are designed from an approximating function which possess circularly symmetry properties. The function is obtained by Taylor Series expansion about the origin and using the Laplace transformation to obtain the corresponding analog filter, and then using the bilinear z-transformation to obtain the corresponding digital filter. Stability of these filters is considered in detail and the results for each second-order section obtained and stated in the form of three constraints. Finally, the method has been developed into a computer-aided design program.

ACKNOWLEDGEMENTS

The author expresses his deep sense of gratitude to Professor V. Ramachandran for suggesting the problem and for his guidance throughout the course of this investigation and for his advice during the preparation of the manuscript. He expresses his thanks to Dr. A. Antoniou for valuable discussions and suggestions.

To my parents Yu, King-Tong and Mei-Ching

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

2-D	Two-Dimensional
1-D	One-Dimensional
1-S	One-sided
2-S	Two-sided
1S1D	One-Sided One-Dimensional
2S1D	Two-Sided One-Dimensional
2S2D	Two-Sided Two-Dimensional
$H_N(s)$	Nth order approximation function One-Sided One-Dimensional
$H_{2N}(s)$	Nth order approximation function Two-Sided One-Dimensional
$H_{2DN}(s_1,s_2)$	Nth order approximation function Two-Sided Two-Dimensional
BIBO	Bounded-Input Bounded-output

CHAPTER I

INTRODUCTION

1.1 GENERAL

Two-dimensional (2-D) digital filter offers many advantages in digital signal processing. Applications exist in the processing of images and geophysical data. One of the ways is to process the image signal with filters whose response is approximately circularly symmetric.

The design techniques for recursive filters in one-dimensional (1-D) case can often be extended to the two-dimensional case in order to obtain the circularly symmetric magnitude response [1] . On the other hand, the design of recursive filters in 2-D becomes difficult due to the fact that a polynomial in two variables $B(z_1, z_2)$ cannot be factored (generally) into first order polynomials. It is also difficult to test the stability of 2-D recursive filters except of some simple cases.

This report is restricted to the study of approximately circularly symmetric systems, which allows us to use the technique of Taylor Series Expansion analysis. This approximation technique has proved its value in numerous problems of practical interest in both 1-D and 2-D system [1] .

In this resport, we will discuss how to obtain a 1-D recursive filter based on the technique of Taylor Series Expansion of function $e^{x_c(t_1^2+t_2^2)}$. Also, we will show how to cascade a number of such filters in order to obtain a stable circularly symmetric 2-D digital filters.

In the course of this discussion, it is essential to study some of the important simularities and differences between 1-D and 2-D digital systems.

1.2 DEFINITIONS

Some accepted definitions of 1-D and 2-D digital system functions are given below [8].

Definition 1

A infinite response linear shift invariant 1-D systems can be described by finite difference equations of the form

$$y_n = \sum_{K=0}^{N-1} a_k\, x(n-k) - \sum_{K=0}^{M-1} b_k\, y(n-k) \qquad (1.1)$$

and they have rational z transforms

$$H(z) = \frac{\sum_{K=0}^{N-1} a_k z^{-k}}{\sum_{K=0}^{M-1} b_k z^{-k}} = \frac{A(z)}{B(z)} \qquad (1.2)$$

Definition 2

A infinite response linear shift invariant 2-D system can be described by finite difference equation of the form

$$\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} a(n_1,n_2)\, x(m_1-n_1,m_2-n_2)$$

$$= \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} b(n_1,n_2)\, y(m_1-n_1,m_2-n_2) \qquad (1.3)$$

where $\{x(n_1,n_2)\}$ is the input array, $\{y(n_1,n_2)\}$ is the output array, and $\{a(n_1,n_2)\}$, $\{b(n_1,n_2)\}$ are the coefficient matrics which define the transfer function of the filter.

In the (z_1,z_2) plane the filter represented by (1.3) is describe by the transfer function

$$H(z_1,z_2) = \frac{\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} a(n_1,n_2)\, z_1^{-n_1}\, z_2^{-n_2}}{\sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} b(n_1,n_2)\, z_1^{-n_1}\, z_2^{-n_2}}$$

$$= \frac{A(z_1,z_2)}{B(z_1,z_2)} \qquad (1.4)$$

1.3 SOME ADVANTAGES OF 1-D SYSTEMS

In several interesting cases, some 1-D systems can represent special classes of two and higher dimensional systems. Some examples are

a) In some cases, 2-D arrays can be represented exactly by 1-D sequences [14]. This approach has great potential for 2-D digital signal processing, Since the resulting sequences can be processed by familiar 1-D techniques and the results can then be remapped to 2-D.

b) In some cases, some 1-D systems have the same magnitude characteristics as some class of 2-D systems. An 1-D system can be represented a 2-D lowpass filtering (LP) system with approximately circularly symmetric response. For this reason, those 2-D results can be regarded as straightforward extensions of 1-D results.

c) Because there are many operations which are more easily performed using 1-D mathematics, it is worthwhile to use 1-D filters to perform 2-D tasks.

1.4 SOME SIMILARITIES OF 1-D AND 2-D DIGITAL SYSTEMS

The following are some of the similarities of 1-D and 2-D recursive digital system:

a) Fundamental properties: For all stable 1-D and 2-D digital systems, it is desirable that both 1-D and 2-D digital systems are time-invariant and shift- invariant.

b) Pole location conditions for BIBO stability: In 1-D digital systems, The BIBO(Bounded-input,Bounded-output) stability condition can be related to the position of the poles in the z-domain, that is to the region of analyticity for the transfer function H(z). Such as

$$B(z) \neq 0 \qquad \text{for} \qquad |z| \leq 1 .$$

This can be extended to 2-D digital systems. As shown below

$$B(z_1, z_2) \neq 0 \qquad \text{for } |z_1| \leq 1 \cap |z_2| \leq 1 .$$

simultaneously.

c) Constraints on the coefficients: The stability of the 2-D digital system can also be determined by the coefficients of the denominator polynomial, $B(z_1, z_2)$. The general criterion for stability given by Shanks [6] is that for the coefficients of expansion of $1/B(z_1, z_2)$ in a positive power of z_1 and z_2 to converge absolutely, it is necessary and sufficient that the poles of $1/B(z_1, z_2)$ must satisfy the pole location conditions for BIBO stability. This is similar to the constraints on the 1-D case.

d) Magnitude response: Some 1-D digital systems have same magnitude response as some class of 2-D digital systems. For example, some 2-D digital filters, whose frequency response is approximately circularly symmetric. This can also be obtained by cascading a number of some classes of 1-D digital filters.

1.5 SOME DISTINCT DIFFERENCES BETWEEN 1-D AND 2-D DIGITAL SYSTEMS:

Some of the important difference between 1-D and 2-D digital systems are:

a) Factorization of the polynomial: The difficulties of 2-D digital systems of degree 2N are almost always related to the fact that there is no fundamental theorem of algebra for polynomials in two independent variables which allows us to factorize it into a product 2N polynomial factors of first degree [2]. On the other hand, there are some theorem and powerful techniques which allows us to factorize an 1-D polynomial of degree N into a product of N polynomial factors of first degree, thereby allowing us to find the roots of polynomials to check the pole location conditions for BIBO stability.

b) Singularities of rational functions: In 1-D digital systems, all poles are isolated. It is only necessary to locate a finite set of poles in z-plane, which require that the system function have no poles outside the unit circle. In 2-D digital systems, due to the factorization problem, there is no techniques for computing the continuum of poles for a general 2-D function. However, for a low-order 2-D function, we can estimate the region of poles, but we cannot guarantee stability. In general, it requires that the system function have no poles outside the unit sphere.

1.6 SCOPE OF THE REPORT:

The objective of this technical resport is to discuss some aspects of the mathematical framework underlying the 2-D digital filtering having approximately circularly symmetric response, and to design this type of 2-D digital filters.

The remainder of this report is divided into VII chapters. In chapter II, some properties of circularly symmetric 2-D system are presented. This will establish the objective to later chapters. In chapter III, we discussed the approximating technique by using Taylor Series Expansion. The representation of one-sided (1-s) approximation, two-sided (2-s) approximation of 1-D and 2-D continuous filter functions are also discussed here. In chapter IV, we discuss the digital filter function which can be obtained by using double bilinear z-transformation. Also, we discussed the stability testing procedure for the system function. In chapter V, we presented the design examples. In chapter VI, we discussed the computer-aided design of this type recursive digital filter. In chapter VII, we will review the design technique for 2-D digital low-pass filter with approximately circularly symmetric response.

CHAPTER II

SOME PROPERTIES OF CIRCULARLY SYMMETRIC 2-D SYSTEM

2.1 INTRODUCTION

In many applications, a 2-D digital filter is required to possess circular symmetry in its frequency response. Some typical applications exist in the processing of images and geoghysical data. When designing such filters it is desirable to know the class of transfer functions that possess the desired circularly symmetry, so that the approximation can be carried out in the domain of that class of filter functions.

In this chapter, we will examine some properties of 2-D systems whose magnitude response is approximately circularly symmetric.

2.2 PROPERTIES OF CIRCULAR SYMMETRY 2-D FUNCTION

Under circular symmetry the function is invariant on any path around the origin in the desired S- or Z-plane. The nature of the conditions imposed by circular symmetry on the analog and digital functions are derived as follow:

a) Analog function: A two-variable function $H(s_1,s_2)$ is said to possess circularly symmetric magnitude response, if this magnitude function satisfies following conditions[4].

1) Under the circular symmetry, the function magnitude is invariant on any circular path around the origin in S-plane. This can be written as

$$|H(s_1,s_2)|^2 = H_s(s_1^2 + s_2^2) \qquad (2.1)$$

2) $H(s_1,s_2)$ possesses circular symmetry in its magnitude response, iff its numerator and denominator possess the above symmetry property individually.

3) A two-variable polynomial $P(s_1,s_2)$ possesses circularly symmetric property in its magnitude response, iff each of its factors individually are

$$P(s_1,s_2) = P_s(s_1^2 + s_2^2) \qquad (2.2)$$

4) From the conditions (2) and (3), we then concluded that a two-dimensional rational function $H(s_1,s_2)$ possessing these properties can be expressed as

$$|H(s_1,s_2)|^2 = \frac{P_s(s_1^2 + s_2^2)}{Q_s(s_1^2 + s_2^2)} \qquad (2.3)$$

b) Digital function: Any digital transfer function $H(z_1,z_2)$ can be generated from analog function $H(s_1,s_2)$ by means of bilinear z-transformation. As the circularly symmetry of a function depends on s_1 and s_2 in S-domain, the circularly symmetry behaviour is unaffacted by bilinear z-transformation. Therefore, the conditions to be required by digital functions to possess circularly symmetry can be obtained from the corresponding conditions on the analog functions.

A two-variable function $H(z_1,z_2)$ is said to posses circularly symmetric magnitude response, iff this function satisfies the following conditions.

1) Under the circular symmetry, the function magnitude is invariant on any circular path around the origin in Z-plane. This can be written as

$$|H(z_1,z_2)|^2 = H_s\left(\left(\frac{z_1-1}{z_1+1}\right)^2 + \left(\frac{z_2-1}{z_2+1}\right)^2 \right) \qquad (2.4)$$

2) $H(z_1,z_2)$ possesses circular symmetry in its magnitude response, iff its numerator and denominator possess the above symmetry property individually.

3) A two-variable polynomial $P(z_1,z_2)$ possesses circularly symmetric property in its magnitude response, iff each of its factors individually are

$$P(z_1,z_2) = P_s\left(\left(\frac{z_1-1}{z_1+1}\right)^2 + \left(\frac{z_2-1}{z_2+1}\right)^2 \right) \qquad (2.5)$$

4) From the conditions (2) and (3), we then concluded that a two-dimensional rational function $H(z_1,z_2)$ possessing these properties can be expressed as

$$|H(z_1,z_2)| = \frac{P_s\left(\left(\frac{z_1-1}{z_1+1}\right)^2 + \left(\frac{z_2-1}{z_2+1}\right)^2 \right)}{Q_s\left(\left(\frac{z_1-1}{z_1+1}\right)^2 + \left(\frac{z_2-1}{z_2+1}\right)^2 \right)} \qquad (2.6)$$

2.3 SUMMARY AND DISCUSSION

In this chapter, we have examined the properties of 2-D system whose magnitude response is circular symmetry. To achieve a circularly symmetric response, the transfer function required its magnitude response is invariant on any circular path around the origin in the desired domain. In a similar manner, one other way to define a circularly symmetric function is that both numerator and denominator polynomials possess circular symmetry properties.

From the theory of multivariable polynomials, it is found that an exact circularly symmetric function which is required to satisfy all the above conditions cannot be obtained [8]. But it is possible to obtain a good approximating function which possesses circular symmetry in its magnitude response. One of this approximating problems will be examined in the following chapter.

CHAPTER III

APPROXIMATION

3.1 INTRODUCTION

From the theory of multivariable polynomials and the properties of circularly symmetric 2-D system, it can be shown that the relation in chapter II cannot be satisfied by any polynomial other than $H(s_1, s_2) = 1$, or $H(z_1, z_2) = 1$ [1]. This implies that an exact circularly symmetric filter cannot be obtained in any rational, causal and stable two variable function. However, we could obtain an approximated function which possesses very good approximately circularly symmetric response.

In this chapter, we will examine some rational functions which possess approximately circularly symmetry in its magnitude response.

3.2 CIRCULARLY SYMMETRIC FUNCTION APPROXIMATED BY TAYLOR SERIES

An exact circular symmetric 2-D function which requires that the filter frequency response $H(s_1, s_2)$ be a function of $s_1^2 + s_2^2$ in S-domain cannot be obtained [8]. This can be proved by the theory of multivariable polynomials. It can also be shown that there is no polynomial which can satisfy all the above conditions, other than $H(s_1, s_2) = 1$. But it is possible to obtain a good approximating function by Taylor Series which possesses circular symmetry. Also this 2-D function can be seperated into a two one-dimensional functions of the type $H_i(w_i)$; for $i = 1,2$. So that

$$H(w_1^2,w_2^2) = H_1(w_1^2) \cdot H_2(w_2^2) \qquad (3.1)$$

To satisfy condition (2.1) to (2.3), equation (3.1) can be expressed as

$$H_s(w_1^2 + w_2^2) = H_1(w_1^2) \cdot H_2(w_2^2) \qquad (3.2)$$

One of the solutions of this approximating problem is to approximate

$$H_i(w_i^2) = e^{-x_c w_i^2} \quad ; \text{ for } i = 1,2 \qquad (3.3)$$

where x_c is a scalar which gives the required response of $H_i(w_i^2)$ at w_c and its magnitude response can be shown in figure (3.1). Then, its 2-D filter function $H_s(w_1^2, w_2^2)$ will be approximated by

$$\begin{aligned} H_s(w_1^2,w_2^2) &= H_1(w_1^2) \cdot H_2(w_2^2) \\ &= e^{-x_c w_1^2} \cdot e^{-x_c w_2^2} \\ &= e^{-x_c(w_1^2 + w_2^2)} \qquad (3.4) \end{aligned}$$

and its magnitude response can be shown as in figure (3.2) for $x_c=1/2$, and its contour response is shown as in figure (3.3) Equation (3.4) will possesses required circular symmetric properties with $w_1^2 + w_2^2$ = constant.

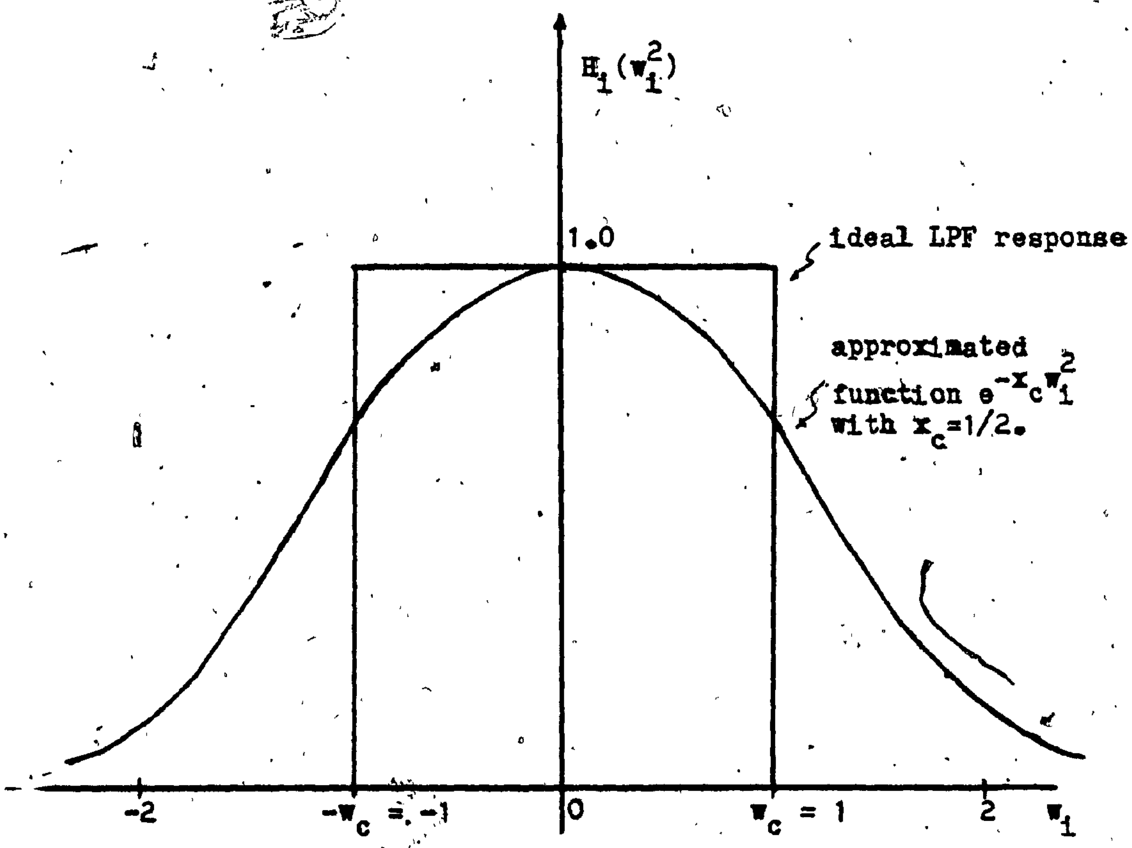

Figure 3.1 magnitude response of approximating function $e^{-x_c w_1^2}$.

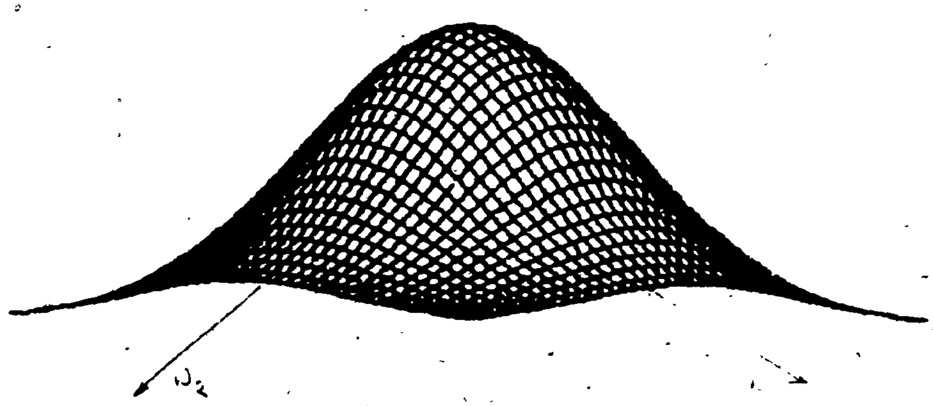

FIG 3.2 PERSPECTIVE VIEW OF MAGNITUDE RESPONSE OF THEORECTICAL FUNCTION

Y(X,Z) = EXP(-(X**2 + Z**2))

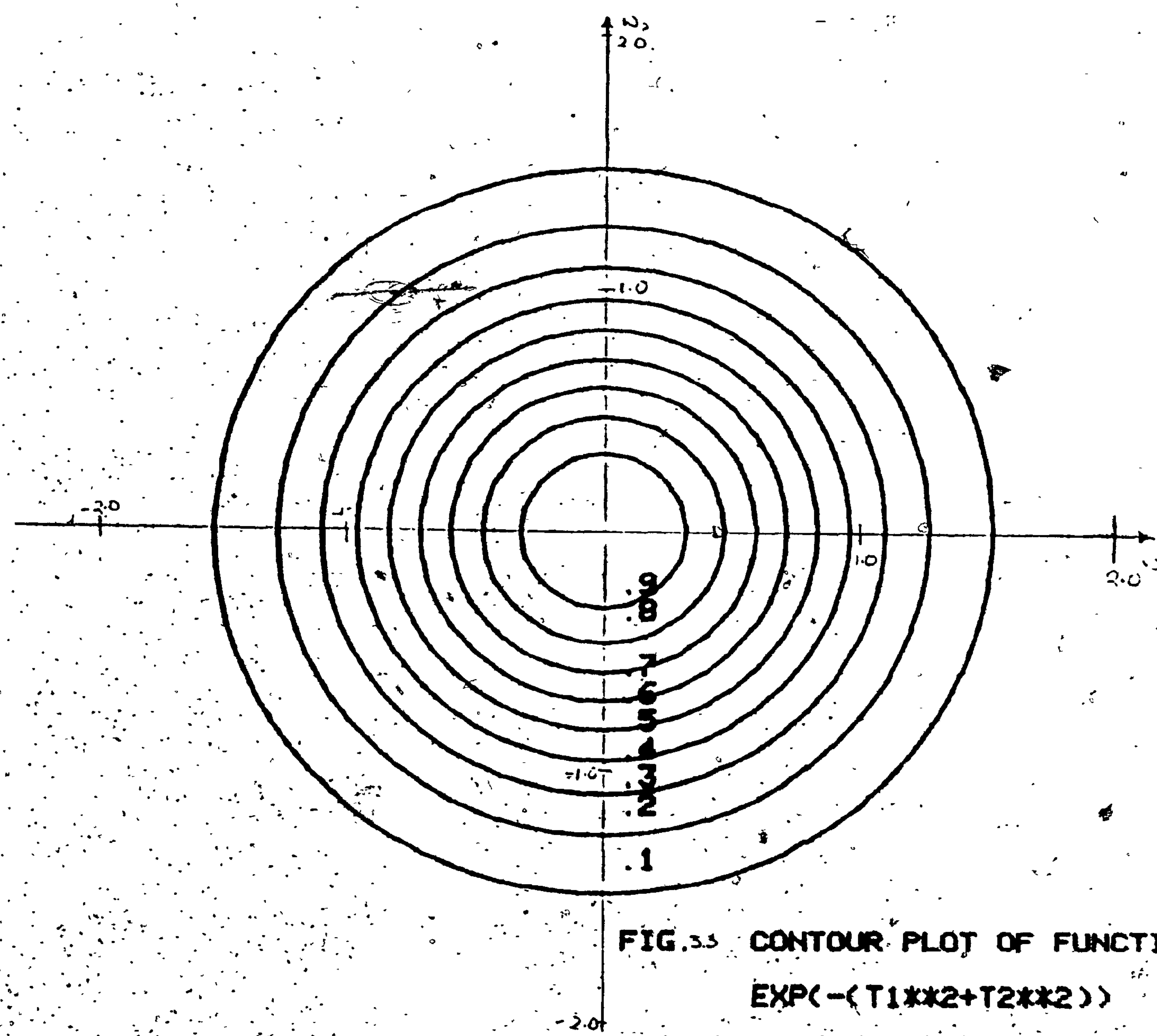

FIG. 3.3 CONTOUR PLOT OF FUNCTION
EXP(-(T1**2+T2**2))

It is impossible to obtain a rational function in $S-(s_1,s_2)$ domain whose frequency response is the same as in equation (3.4). However, there is a class of functions with its impulse response approximately circular symmetric and rational in s-domain. These can be obtained by expanding function $e^{-x_c t^2}$ in Taylor's Series about the origin in positive direction, or $t \geq 0$, as

$$h(t)=e^{-x_c t^2}$$

$$\cong 1 - x_c t^2 + \frac{x_c t^4}{2!} - \frac{x_c^3 t^6}{3!} + \frac{x_c^4 t^8}{4!} + \cdots\cdots$$

$$+ \frac{(-1)^n x_c^n t^{2n}}{n!} \qquad \text{as } t \to \infty \qquad (3.5)$$

$$= 0 \quad \text{for} \quad t < 0$$

and its Laplace Transformation will be

$$H(s) = \frac{1}{s} - \frac{2x_c}{s^3} + \frac{12x_c^2}{s^5} - \frac{120x_c^3}{s^7} + \frac{1680x_c^4}{s^9}$$

$$- \frac{30240x_c^4}{s^{11}} + \cdots\cdots$$

$$\text{as } N \to \infty \qquad (3.6)$$

It is possible to obtain some class of rational functions, whose continued division gives an approximation to equation (3.6). As

$$H_N(s) = \frac{\sum_{i=0}^{n-1} a_i s^{n-1-i}}{\sum_{j=0}^{n} b_j s^{n-j}} \tag{3.7}$$

where 1) N is the Nth (or s^{-N}) power approximation of equation (3.6)

2) $n \geq 1$; $n = \frac{N+1}{2}$ for $N \geq 1$

3) all a_i and b_j are positive ans real

4) $a_0 = b_0 = 1$

Then, long division of the numerator by denominator of H(s) of equation (3.7) , yields

$$H(s) = \frac{1}{s} + \frac{a_1 - b_1}{s^2} + \frac{(a_2 - b_2) - b_1(a_1 - b_1)}{s^3}$$

$$+ \frac{((a_3 - b_3) - b_2(a_1 - b_1)) - ((a_2 - b_2) - b_1(a_1 - b_1))}{s^4}$$

$$+ \frac{\begin{array}{l} (((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1))) \end{array}}{s^5}$$

$$+ \frac{\begin{array}{l} ((((a_5 - b_5) - b_4(a_1 - b_1)) - b_3((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_2(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \\ -b_1((((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \end{array}}{s^6}$$

$$+ \frac{\begin{array}{l} (((((a_6 - b_6) - b_5(a_1 - b_1)) - b_4((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_3(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \\ -b_2((((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1))))) \\ -b_1((((((a_5 - b_5) - b_4(a_1 - b_1)) - b_3((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_2(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \\ -b_1(((((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))))) \end{array}}{s^7}$$

$$+ \cdots\cdots\cdots \qquad \text{as } s \to \infty \qquad (3.8)$$

By comparing equation (3.8) and equation (3.6), we obtained the following relations

$$a_1 - b_1 = 0 \qquad (3.9)$$

$$(a_2 - b_2) - b_1(a_1 - b_1) = -2x_c$$

or

$$a_2 - b_2 = -2x_c \qquad (3.10)$$

$$((a_3-b_3)-b_2(a_1-b_1))-b_1((a_2-b_2)-b_1(a_1-b_1)) = 0$$

By simplifying, we obtained

$$a_3 - b_3 = -2x_c b_1 \qquad (3.11)$$

$$((a_4-b_4)-b_3(a_1-b_1))-b_2((a_2-b_2)-b_1(a_1-b_1))$$
$$-b_1(((a_3-b_3)-b_2(a_1-b_1))-b_1((a_2-b_2)-b_1(a_1-b_1)))=12x_c^2$$

and

$$a_4 - b_4 = 12x_c^2 - 2b_2x_c \qquad (3.12)$$

Similary, we obtained

$$a_5 - b_5 = 12x_c^2b_1 - 2x_cb_3 \qquad (3.13)$$

$$a_6 - b_6 = -120x_c^3 - 2x_cb_4 + 12x_c^2b_2 \qquad (3.14)$$

$$a_7 - b_7 = -120x_c^3b_1 + 12x_c^2b_3 - 2x_cb_5 \qquad (3.15)$$

$$a_8 - b_8 = 1680x_c^4 - 120x_c^3b_2 + 12x_c^2b_4 - 2x_cb_6 \qquad (3.16)$$

and so on for higer terms. By substituting equations(3.9) to (3.16) into equation (3.8) we get

$$H(s) = \frac{1}{s} + \frac{a_2 - b_2}{s^3} + \frac{a_4 - b_4 + 2b_2}{s^5}$$
$$+ \frac{a_6 - b_6 - 12b_2 + 2b_4}{s^7}$$
$$+ \frac{a_8 - b_8 + 2b_6 - 12b_4 + 120b_2}{s^9}$$
$$+ \dots\dots \quad \text{as } N \to \infty \qquad (3.17)$$

If we define $H_N(s)$ to be an Nth-order approximating function to equation (3.17), we obtained

$$
\begin{aligned}
H_3(s) &= \frac{1}{s} + \frac{a_2 - b_2}{s^3} \\
&\cong \frac{s + a_1}{s^2 + b_1 s + b_2} \\
&= \frac{s + a_1}{s^2 + a_1 s + 2x_c}
\end{aligned}
\qquad (3.18)
$$

$$
\begin{aligned}
H_5(s) &= \frac{1}{s} + \frac{a_2 - b_2}{s^3} + \frac{a_4 - b_4 + 2b_2}{s^5} \\
&\cong \frac{s^2 + a_1 s + a_2}{s^3 + b_1 s^2 + b_2 s + b_3} \\
&= \frac{s^2 + a_1 s + 4x_c}{s^3 + a_1 s^2 + 6x_c s + 2x_c a_1}
\end{aligned}
\qquad (3.19)
$$

$$
\begin{aligned}
H_7(s) &= \frac{1}{s} + \frac{a_2 - b_2}{s^3} + \frac{a_4 - b_4 + 2b_2}{s^5} \\
&\quad + \frac{a_6 - b_6 - 12b_2 + 2b_4}{s^7} \\
&\cong \frac{s^3 + a_1 s^2 + a_2 s + a_3}{s^4 + b_1 s^3 + b_2 s^2 + b_3 s + b_4} \\
&= \frac{s^3 + a_1 s^2 + 10x_c s + 4x_c a_1}{s^4 + a_1 s^3 + 12x_c s^2 + 6x_c a_1 + 12x_c^2}
\end{aligned}
\qquad (3.20)
$$

$$H_9(s) = \frac{1}{s} + \frac{a_2 - b_2}{s^3} + \frac{a_4 - b_4 + 2b_2}{s^5}$$

$$+ \frac{a_6 - b_6 - 12b_2 + 2b_4}{s^7}$$

$$+ \frac{a_8 - b_8 + 2b_6 - 12b_4 + 120\ b_2}{s^9}$$

$$\cong \frac{s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4}{s^5 + b_1 s^4 + b_2 s^3 + b_3 s^2 + b_4 s + b_5}$$

$$= \frac{s^4 + a_1 s^3 + 18x_c s^2 + 10x_c a_1 s + 32x_c^2}{s^5 + a_1 s^4 + 20x_c s^3 + 12x_c a_1 s^2 + 60x_c^2 s + 12x_c^2 a_1} \tag{3.21}$$

3.3 OBTAINING A TWO-SIDED ONE-DIMENSIONAL TRANSFER FUNCTION

In the preceding section, we have successfully obtained the Nth order rational, one-sided one-dimensional transfer function by Taylor Series, as

$$H_N(s) = \frac{\sum_{i=0}^{n-1} a_i\ s^{n-i-1}}{\sum_{j=0}^{n} b_j s^{n-j}}$$

where $n = (N + 1)/2$

From the properties of function $h(t) = e^{-x_c t^2}$, it shown the function symmetric to h(t) in both positive and negative t. Then its two-sided one-dimensional transfer function can be obtained in summation form, given as

$$H_{2N}(s) = H_N(s) + H_N(-s) \tag{3.22}$$

Equation (3.22) shows that this is the even part of $H_N(s)$ and the poles of Ev $H_N(s)$ occur in quadrantal symmetry. This two-sided one-dimensional transfer function can be written as

$$H_{2N}(s) = \frac{N(s)}{((s-p_1)(s-p_2)......)((s+p_1)(s+p_2)......)} \tag{3.23}$$

Since the numerator polynomial of $H_N(s)$ is one degree lower than the denominator polynomial, then the resulting numerator polynomial in two-sided one-dimensional transfer function is

$$N(s) = 2a_{n-1}b_n \tag{3.24}$$

After rejecting the poles in right half plane, we obtain the required low-pass transfer function, as

$$H_N(s) = \frac{(2a_{n-1}b_n)^{\frac{1}{2}}}{\sum_{j=0}^{n} b_j s^{n-j}} \tag{3.25}$$

$$= \frac{(2a_{n-1}b_n)^{\frac{1}{2}}}{(s+p_1)(s+p_2)....(s+p_n)} \tag{3.26}$$

3.4 OBTAINING A TWO-SIDED TWO-DIMENSIONAL TRANSFER FUNCTION

To obtain a rational, causal and stable 2-D analog transfer function whose magnitude response is approximately circular symmetry, we can cascade two identical two-sided one-dimensional transfer functions with two different variables s_1 and s_2 respectively. Hence, we can write

$$H_{2DN}(s_1,s_2) = H_{2N}(s_1) \cdot H_{2N}(s_2)$$

$$= A \prod_{i=1}^{\frac{(N+1)}{2}} \frac{1}{(s_1 + p_{i1})(s_2 + p_{i2})} \qquad (3.27)$$

where $A = 2a_{n-1}b_n$ and p_{i1} and p_{i2} are the ith poles of function in S_1 and S_2 direction.

Equation (3.27) can also be written in the following form.

$$H_{2DN}(s_1,s_2) = A \prod_{i=1}^{\frac{(N+1)}{2}} T_i(s_1,s_2) \qquad (3.28)$$

$$\text{where} \quad T_i(s_1,s_2) = \frac{a_{00}^{(i)} + a_{01}^{(i)}s_2 + a_{10}^{(i)}s_1 + a_{11}^{(i)}s_1s_2}{b_{00}^{(i)} + b_{01}^{(i)}s_2 + b_{10}^{(i)}s_1 + b_{11}^{(i)}s_1s_2} \qquad (3.29)$$

where

$$a_{00}^{(i)} = 1$$

$$a_{01}^{(i)} = a_{10}^{(i)} = a_{11}^{(i)} = 0$$

$$b_{00}^{(i)} = 1$$

$$b_{01}^{(i)} = p_{i1}$$

$$b_{10}^{(i)} = p_{i2}$$

$$b_{11}^{(i)} = p_{i1}p_{i2}$$

3.5 SOURCE OF ERROR

Normally the error in any approximation solution cannot be determined exactly (unless the exact solution can be found by another method, in which case the approximation technique is unnecessary). We can specify exact bounds on round off errors but not for truncation errors. However, a measure of the magnitude of a truncation error is provided by the (neglected) subsequent terms of the series. In fact, it is the order of error which is harmful to the magnitude response for the lower order of approximation (or system). This error will appear in each stage of the four filter section, and can accumulate as the 2-D signal passing throught it, and may swamp the true response.

For a Nth order approximation, we assumed all higher order coefficients to be zero. As a result of this, we know that the error in each section will no be greater than the (N+1)th order term. From this, it provided us a bound of the error, which may be helpful in improving the performance of designed system. Normally, this error can be reduced by varying the coefficient in the Nth order term (the variation should not be greater than the (N+1)th order term), and this will be discussed later.

3.6 SUMMARY AND DISCUSSION

From the theory of multivariable polynomials and the properties of circularly symmetric 2-D system, it can be shown that an exact circularly symmetric function cannot be obtained in any rational, causal and stable 2-D function. However, it is possible to obtain a good approximating function which possesses circular symmety by Taylor Series Expansion.

From the study of Taylor Series Expansion of a rational function $e^{-x_c t^2}$, and Laplace Transformation of this series, we obtained an one-sided approximating function which is also rational in s. Its two-sided approximating function can be obtained in summation form of two identical one-sided function. Finally, a rational, causal and stable 2-D function, whose magnitude response is approximately sircular symmetry can be achieved by cascade two identical two-sided one-dimensional function with two different variables S_1 and S_2 respectively.

The source of error in the approximating function is also studied. This error can be reduced by varying the coefficient in Nth order term.

CHAPTER 4

TWO-DIMENSIONAL DIGITAL TRANSFER FUNCTION

4.1 INTRODUCTION

The usual approach of designing of recursive digital filter is to perform a suitable transformation, such as bilinear z-transformation, of a given analogue filter. This technique provides good results for a filter having picewise constant. frequency response. In this chapter, we would like to discuss the design of circularly symmetric 2-D digital filters as a direct substitution into a continuous frequency response function $H(s_1,s_2)$.

The stability of a 2-D digital filter is one of the major problems in designing of digital filters. It is possible for a 2-D recursive filter to be unstable. That is, given a bounded-input array, the output array can be unbounded. Later in this chapter, we would like to able to place some constraints on the coefficients of the 2-D recursive filter to assure its stability.

4.2 BILINEAR Z-TRANSFORMATION

The bilinear z-transformation is a direct method to deriving a digital filter from an analog filter. This direct replacement in H(s) of the complex frequency variable S by

$$s \rightarrow \frac{2(1 - z^{-1})}{T(1 + z^{-1})} \tag{4.1}$$

or

$$s \rightarrow \frac{2(z - 1)}{T(z + 1)}$$

Equation (4.1) also known that

$$w = \frac{2}{T} \tan(\Omega T/2) \tag{4.2}$$

where w = frequency variable for analog filter

Ω = frequency variable for digital filter

T = sampling period

The nature of this mapping is that the entire jw axis in the s-plane is mapped onto the unit circle, and the entire left-half plane in s-domain is mapped into inside of the unit circle in Z-plane, as shown in figure 4.1

Let H(s) be the transfer function of a given prototype analog filter. Then its corresponding prototype digital filter can be obtained from bilinear z-transformation by making the algebraic substitution of equation (4.1); i.e.

$$H(z) = H(S)\Big|_{s = \frac{2(1-z^{-1})}{T(1+z^{-1})}} \tag{4.3}$$

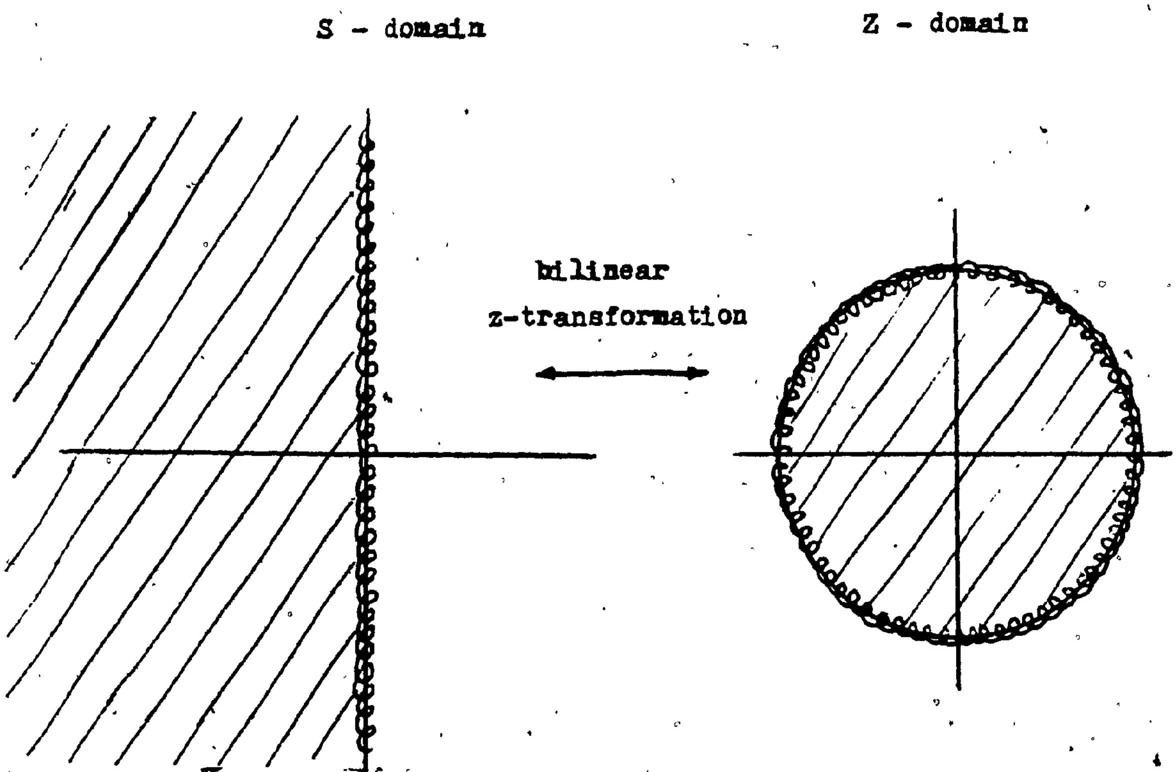

figure 4.1 The mapping from s-plane to z-plane by bilinear z-transformation

It is also shown that the orders of the denominators of H(z) is same as H(s). However, the order of numerator may be differ. Finally, if a given filter in analog domain is a realizable, stable and continuous system, it will result into a realizable, and stable digital system.

4.3 DIGITAL FILTER FUNCTION IN CANONIC FORM

From last section, it is easy to shown that a 2-D digital transfer function can be obtained from an approximately circular symmetric 2-D function with two variable in s by using double bilinear z-transformation. i.e.

$$s_1 \rightarrow \frac{2(1-z_1^{-1})}{T_1(1+z_1^{-1})} \tag{4.4}$$

or

$$s_1 \rightarrow \frac{2(z_1-1)}{T_1(z_1+1)}$$

and

$$s_2 \rightarrow \frac{2(1-z_2^{-1})}{T_2(1+z^{-1})} \tag{4.5}$$

or

$$s_2 \rightarrow \frac{2(z_2-1)}{T_2(z_2+1)}$$

where T_1, T_2 are sampling period in z_1, z_2 plane respectively.

The prototype digital filter with approximate circularly symmetric response corresponding to the analog function with same characteristic is the algebraic substitution of equations (4.4) and (4.5) in equation (3.29). i.e.

$$H(z_1,z_2) = H(s_1,s_2)\Big|_{s_1=\frac{2(1-z_1^{-1})}{T_1(1+z_1^{-1})},\ s_2=\frac{2(1-z_2^{-1})}{T_2(1+z_2^{-1})}}$$

$$= A\prod_{i=1}^{\frac{(N+1)}{2}} T_i(s_1,s_2)\Big|_{s_1=\frac{2(1-z_1^{-1})}{T_1(1+z_1^{-1})},\ s_2=\frac{2(1-z_2^{-1})}{T_2(1+z_2^{-1})}}$$

$$= A\prod_{i=1}^{\frac{(N+1)}{2}} T_i'(z_1,z_2) \qquad (4.6)$$

where

$$T_i'(z_1,z_2) = \frac{1+z_1^{-1}+z_2^{-1}+z_1^{-1}z_2^{-1}}{d^{(i)}+c^{(i)}z_1^{-1}+b^{(i)}z_2^{-1}+a^{(i)}z_1^{-1}z_2^{-1}}$$

$$= \frac{1+z_1+z_2+z_1z_2}{a^{(i)}+b^{(i)}z_1+c^{(i)}z_2+d^{(i)}z_1z_2} \qquad (4.7)$$

and

$$a^{(i)} = b_{00}^{(i)} - b_{01}^{(i)}\cdot\frac{2}{T_2} - b_{10}^{(i)}\cdot\frac{2}{T_1} + b_{11}^{(i)}\cdot\frac{4}{T_1T_2},$$

$$b^{(i)} = b_{00}^{(i)} - b_{01}^{(i)}\cdot\frac{2}{T_2} + b_{10}^{(i)}\cdot\frac{2}{T_1} - b_{11}^{(i)}\cdot\frac{4}{T_1T_2}$$

$$c^{(i)} = b_{00}^{(i)} + b_{01}^{(i)} \cdot \frac{2}{T_2} - b_{10}^{(i)} \cdot \frac{2}{T_1} - b_{11}^{(i)} \cdot \frac{4}{T_1 T_2}$$

$$d^{(i)} = b_{00}^{(i)} + b_{01}^{(i)} \cdot \frac{2}{T_2} + b_{10}^{(i)} \cdot \frac{2}{T_1} + b_{11}^{(i)} \cdot \frac{4}{T_1 T_2}$$

4.4 STABILITY OF THE DIGITAL FILTER

There are two major problems in design of digital filters; approximation and their stability. If a filter is unstable, any input including noise can cause the filter output to grow without bound. Thus, we would like to place some constraints on the coefficients of this 2-D recursive filter to assure its stability.

A 2-D digital recursive filter is characterized by the 2-D transfer function

$$H(z_1, z_2) = \frac{\sum_{m=0}^{p} \sum_{n=0}^{q} a_{mn} z_1^m z_2^n}{\sum_{m=0}^{p} \sum_{n=0}^{q} b_{mn} z_1^m z_2^n} \qquad (4.8)$$

where a_{mn} and b_{mn} are constant.

If we expand equation (4.8), $H(z_1, z_2)$ into a power series, we obtain

$$H(z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} h_{mn} z_1^{-m} z_2^{-n} \qquad (4.9)$$

where the coefficients h_{mn} represent the impulse response of the filter. To assure its stability, it can be easily show 5 that this requires that the impulse response of the filter h (m,n) should satisfy the inequality

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h(m,n)| \leqslant \infty \qquad (4.10)$$

The basic theorem for guaranteeing a system with BIBO stability is due to Shanks [5][6].

Testing stability of a given 2-D function by using Shanks theorem is very hard work, and tedious to apply. A simplified version of Shanks theorem done by Huang [7] is as follows.

A causal filter characterized by

$$H(z_1, z_2) = \frac{A(z_1, z_2)}{B(z_1, z_2)} \qquad (4.11)$$

is stable if

1) $b(z_1, z_2) \neq 0$ for $|z_1| = 1$ and $|z_2| \geqslant 1$

2) $b(z_1, +\infty) \neq 0$ for $|z_1| \geqslant 1$

According to this theorem and the theory of algebraic functions in $z_2 = f(z_1)$, we have following test procedure:

1) map $\partial d_1 = (z_1;\ |z_1| = 1)$ into the z_2 plane according to $B(z_1, z_2) = 0$.

2) check whether the image lies outside $d_2 = (z_2;\ |z_2| \leqslant 1)$

3) solve $B(z_1, 0) = 0$ to check whether there are any roots with magnitude smaller than 1 .

We next consider the ith section of 2-D filter function of

$$H(z_1,z_2) = A \prod_{i=1}^{\frac{(N+1)}{2}} T_i'(z_1,z_2)$$

where

$$T_i'(z_1,z_2) = \frac{A(z_1,z_2)}{B(z_1,z_2)}$$

$$= \frac{1 + z_1 + z_2 + z_1 z_2}{a^{(i)} + b^{(i)} z_1 + c^{(i)} z_2 + d^{(i)} z_1 z_2} \quad (4.12)$$

We first establish the stability conditions in terms of coefficients $a^{(i)}$, $b^{(i)}$, $c^{(i)}$ and $d^{(i)}$ using theorem 2 (Huang).

Let

$$B(z_1,z_2) = a^{(i)} + b^{(i)} z_1 + c^{(i)} z_2 + d^{(i)} z_1 z_2$$

$$= 0 \quad (4.13)$$

Thus, we obtained

$$z_2 = -\frac{a^{(i)} + b^{(i)} z_1}{c^{(i)} + d^{(i)} z_1} \quad (4.14)$$

Equation (4.14) is a new bilinear transformation mapping circle into circles and hence the image of the unit circle

$$\delta d_1 \equiv (z_1;\ |z_1|=1\ )$$

in the z_2 plane is a circle.

From équation (4.14), the center of this image circle is on the real axis and its intersection with real axis at

$$z_2 = - \frac{a^{(i)} + b^{(i)}}{c^{(i)} + d^{(i)}} \quad ; \quad \text{for} \quad z_1 = 1 \tag{4.15}$$

and

$$z_2 = - \frac{a^{(i)} - b^{(i)}}{c^{(i)} - d^{(i)}} \quad ; \quad \text{for} \quad z_1 = -1 \tag{4.16}$$

Again, from theorem 2 (Huang), it is easy to obtain the following constraints. In order to satisfy condition 1, that is if

$$\text{Constraint 1,} \quad \left| \frac{a^{(i)} + b^{(i)}}{c^{(i)} + d^{(i)}} \right| > 1 \tag{4.17}$$

and

$$\text{Constraint 2,} \quad \left| \frac{a^{(i)} - b^{(i)}}{c^{(i)} - d^{(i)}} \right| > 1 \tag{4.18}$$

In order to satisfy condition 2, we obtained following constraint

$$B(z_1,0) = a^{(i)} + b^{(i)} z_1$$
$$= 0$$

gives

$$z_1 = - \frac{a^{(i)}}{b^{(i)}}$$

Thus

$$\text{Constraint 3,} \quad \left| \frac{a^{(i)}}{b^{(i)}} \right| > 1 \qquad (4.19)$$

After repeating the above testing procedures for all ith section of $H(z_1,z_2)$, we can conclude:

The approximated function is stable, if and only if all ith section of $T_i'(z_1,z_2)$ are stable.

Proof:

Since $H(z_1,z_2)$ is a product of $(N+1)/2$ second order digital filter functions $T_i'(z_1,z_2)$, and if $T_i'(z_1,z_2)$ is stable; for all i, then

$$H(z_1,z_2) = A \prod_{i=1}^{\frac{(N+1)}{2}} T_i'(z_1,z_2)$$

also represents a stable filter

Q.E.D.

From the above constraints, which is in terms of coefficients of $a^{(i)}$, $b^{(i)}$, $c^{(i)}$ and $d^{(i)}$, it will be easy to obtain the corresponding constraints for a_{ij} and b_{ij}.

4.5 SUMMARY AND DISCUSSION

In this chapter, we discussed the digital filter function with circularly symmetric response which is obtainable by applying the double bilinear transformation to a continuous frequency response function $H(s_1,s_2)$. The resulting 2-D digital filter function consists of (N+1)/2 second order sections connected in cascade. The cascade form has the following advantages over the direct form [15].

1) The world length that guarantees the stability of the filter after the rounding of its coefficients is shorter. This due to the fact that sensitivity of the mapping of the unit circle of z_2 -plane into the z_1 -plane under the denominator of transfer function $T_1'(z_1,z_2)$ is lower [16].
2) The error caused by quantization effect and finite coefficient size is lower [17].
3) In 2-D filters, the number of multiplications needed for each output using the cascade form is less than that needed in direct form.
4) It is easier to check the stability of the cascade filter during the testing process.

We had established some constraints on the coefficients of the 2-D filter function. In order to check the stability of the cascade filter, we started from first section of the filter by substituting the coefficients of the denominator of $T_1'(z_1,z_2)$

into all three constraints, and for all sections of $T_i'(z_1, z_2)$. For a stable 2-D digital filter (in cascade form) all of its sections should satisfy these three constraints.

CHAPTER 5

DESIGN EXAMPLES

5.1 INTRODUCTION

In this chapter, the design techniques of 2-D recursive digital filter with approximately circularly symmetric magnitude response by using Taylor Series Expansion and double bilinear z-transformation are illustrated through the following examples:

1) A third-order approximation function.
2) A fifth-order approximation function.

5.2 DESIGN EXAMPLE 1

Design a third-order approximation lowpass 2-D digital filter with approximatly circularly symmetric response.

Solution : From chapter 3, we obtained the one-sided one-dimensional analog transfer function as

$$H_3(s) = \frac{s + a_1}{s^2 + a_1 s + 2x_c} \qquad (5.2.1)$$

Now, its zero and pole locations at

zero at $s = -a_1$

poles at $s = \dfrac{-a_1 + (a_1^2 - 8x_c)^{1/2}}{2}$

and

$$s = \frac{- a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$$

If p_1 and p_2 are the poles of $H_3(s)$, the one-sided one-dimensional transfer function can be written as

$$H_3(s) = \frac{s + a_1}{(s + p_1)(s + p_2)} \qquad (5.2.2)$$

and its two-sided one-dimensional transfer function will be

$$H_{23}(s) = \frac{4a_1 x_c}{(s+p_1)(s+p_1^*)(s+p_2)(s+p_2^*)} \qquad (5.2.3)$$

where p_1^* and p_2^* are the complex conjugates of p_1 and p_2 respectively.

Therefore, its two-sided two-dimensional analog transfer function will be

$$H_{2D3}(s_1,s_2) = H_3(s_1) \cdot H_3(s_2)$$

$$= \frac{4a_1 x_c}{(s_1+p_{11})(s_1+p_{11}^*)(s_1+p_{12})(s_1+p_{12}^*)} \cdot \frac{4a_1 x_c}{(s_2+p_{21})(s_2+p_{21}^*)(s_2+p_{22})(s_2+p_{22}^*)} \qquad (5.2.4)$$

where p_{11}, p_{12} are the poles at s_1 plane and

p_{11}^*, p_{12}^* are the complex conjugates of p_{11} and p_{12}.

p_{21} , p_{22} are the poles at s_2 plane and

p^*_{21} , p^*_{22} are the complex conjugates of p_{21} and p_{22} .

Equation (5.2.4) can also be written as

$$H_{2D3}(s_1,s_2) = A \prod_{i=1}^{4} \frac{1}{(s_1+p_{i1})(s_2+p_{i2})} \quad (5.2.5)$$

$$= A \prod_{i=1}^{4} \frac{1}{b^{(i)}_{oo}+b^{(i)}_{o1}s_2+b^{(i)}_{1o}s_1+b^{(i)}_{11}s_1s_2} \quad (5.2.6)$$

where $A = 16a_1^2x_c^2$

p_{i1} = ith complex pole in s_1 plane

p_{i2} = ith complex pole in s_2 plane

$b^{(i)}_{oo} = 1$

$b^{(i)}_{o1} = p_{i1}$

$b^{(i)}_{10} = p_{i2}$

$b^{(i)}_{11} = p_{i1}p_{i2}$

After applying double bilinear z-transformation to equation (5.2.6), we obtain

$$H_{2D3}(z_1,z_2) = A \prod_{i=1}^{4} \frac{1+z_1+z_2+z_1z_2}{a^{(i)} + b^{(i)}z_1 + c^{(i)}z_2 + d^{(i)}z_1z_2} \quad (5.2.7)$$

where
$$a^{(i)} = b_{00}^{(i)} - b_{01}^{(i)} \cdot \frac{2}{T_2} - b_{10}^{(i)} \cdot \frac{2}{T_1} + b_{11}^{(i)} \cdot \frac{4}{T_1 T_2}$$

$$b^{(i)} = b_{00}^{(i)} - b_{01}^{(i)} \cdot \frac{2}{T_2} + b_{10}^{(i)} \cdot \frac{2}{T_1} - b_{11}^{(i)} \cdot \frac{4}{T_1 T_2}$$

$$c^{(i)} = b_{00}^{(i)} + b_{01}^{(i)} \cdot \frac{2}{T_2} - b_{10}^{(i)} \cdot \frac{2}{T_1} - b_{11}^{(i)} \cdot \frac{4}{T_1 T_2}$$

$$d^{(i)} = b_{00}^{(i)} + b_{01}^{(i)} \cdot \frac{2}{T_2} + b_{10}^{(i)} \cdot \frac{2}{T_1} + b_{11}^{(i)} \cdot \frac{4}{T_1 T_2}$$

Now for $i = 1$, $p_{11} = \dfrac{-a_1 + (a_1^2 - 8x_c)^{1/2}}{2}$ and

$$p_{12} = \frac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$$

and thus $b_{00}^{(1)} = 1$

$$b_{01}^{(1)} = p_{11} = \frac{-a_1 + (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{10}^{(1)} = p_{12} = \frac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{11}^{(1)} = p_{11} p_{12} = 2x_c$$

$$a^{(1)} = 1 + a_1 + 2x_c$$

$$b^{(1)} = 1 - (a_1^2 - 8x_c)^{1/2} - 2x_c$$

$$c^{(1)} = 1 + (a_1^2 - 8x_c)^{1/2} - 2x_c$$

$$d^{(1)} = 1 - a_1 + 2x_c$$

For $i = 2$, we have $p_{21} = \dfrac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$

$$p_{22} = \frac{-a_1 + (a_1^2 - 8x_c)^{1/2}}{2}$$

and $b_{00}^{(2)} = 1$

$$b_{01}^{(2)} = p_{21} = \frac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{10}^{(i)} = p_{22} = \frac{-a_1 + (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{11}^{(i)} = p_{21}p_{22} = 2x_c$$

$$a^{(2)} = 1 - a_1 + 2x_c$$

$$b^{(2)} = 1 + (a_1^2 - 8x_c)^{1/2} - 2x_c$$

$$c^{(2)} = 1 - (a_1^2 - 8x_c)^{1/2} - 2x_c$$

$$d^{(2)} = 1 + a_1 + 2x_c$$

For $i = 3$, we have $p_{31} = \dfrac{-a_1 + (a_1^2 - 8x_c)}{2}$

$$p_{32} = \frac{-a_1 + (a_1^2 - 8x_c)}{2}$$

and $b_{00}^{(3)} = 1$

$$b_{01}^{(3)} = p_{31} = \frac{-a_1 + (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{10}^{(3)} = p_{32} = \frac{-a_1 + (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{11}^{(3)} = p_{31}p_{32} = \frac{a_1^2 - 4x_c - a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$a^{(3)} = 1 + a_1 - (a_1^2 - 8x_c)^{1/2} + \frac{a_1^2 - 4x_c - a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$b^{(3)} = 1 - \frac{a_1^2 - 4x_c - a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$c^{(3)} = 1 - \frac{a_1^2 - 4x_c - a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$d^{(3)} = 1 - a_1 + (a_1^2 - 8x_c)^{1/2} + \frac{a_1^2 - 4x_c - a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

For $i = 4$, we have $p_{41} = \dfrac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$

$$p_{42} = \frac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$$

and $b_{00}^{(4)} = 1$

$$b_{01}^{(4)} = p_{41} = \frac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{10}^{(4)} = p_{42} = \frac{-a_1 - (a_1^2 - 8x_c)^{1/2}}{2}$$

$$b_{11}^{(4)} = p_{41}p_{42} = \frac{a_1^2 - 4x_c + a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$a^{(4)} = 1 + a_1 + (a_1^2 - 8x_c)^{1/2} + \frac{a_1^2 - 4x_c + a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$b^{(4)} = 1 - \frac{a_1^2 - 4x_c + a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$c^{(4)} = 1 - \frac{a_1^2 - 4x_c + a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

$$d^{(4)} = 1 - a_1 - (a_1^2 - 8x_c)^{1/2} + \frac{a_1^2 - 4x_c + a_1(a_1^2 - 8x_c)^{1/2}}{2}$$

Now consider the stability test for the resulting transfer function

For i=1

From constraint 1, we have $\left|\frac{a^{(1)} + b^{(1)}}{c^{(1)} + d^{(1)}}\right| > 1$

By substituting the $a^{(1)}, b^{(1)}, c^{(1)}$ and $d^{(1)}$ values into constraint 1, we have,

$$\left|\frac{2 + a_1 - (a_1^2 - 8x_c)^{1/2}}{2 - a_1 + (a_1^2 - 8x_c)^{1/2}}\right| > 1$$

From contraint 2, we have

$$\left|\frac{a^{(1)} - b^{(1)}}{c^{(1)} - d^{(1)}}\right| > 1$$

By assigning $a^{(1)}$, $b^{(1)}$, $c^{(1)}$ and $d^{(1)}$ valuesinto constraint 2, we have

$$\left|\frac{a_1 + 4x_c + (a_1^2 - 8x_c)^{1/2}}{a_1 - 4x_c + (a_1^2 - 8x_c)^{1/2}}\right| > 1$$

From constraint 3, we have $\left|\frac{a^{(1)}}{b^{(1)}}\right| > 1$

By assigning $a^{(1)}$, $b^{(1)}$, $c^{(1)}$ and $d^{(1)}$ values into constraint 3, we have

$$\left|\frac{1 + a_1 + 2x_c}{1 - a_1 + 2x_c}\right| > 1$$

By applying the testing procedure for i=2,3 and 4, we obtain the similar results for each constraint.

From constraint 3, we can easily obtained the constraint for a_1, as $a_1 > 0$.

From constraints 1 and 2, we obtained the constraint for x_c, as $x_c \neq 0$ and $x_c > 0$.

For $x_c = 1$, $a_1 = 3$ and $T_1 = T_2 = 2$, we obtain the magnitude response for this third-order approximating function as shown in figure 5.1. The perspective view of this function is shown in figure 5.2, and the contour plot of this function is shown in figure 5.3.

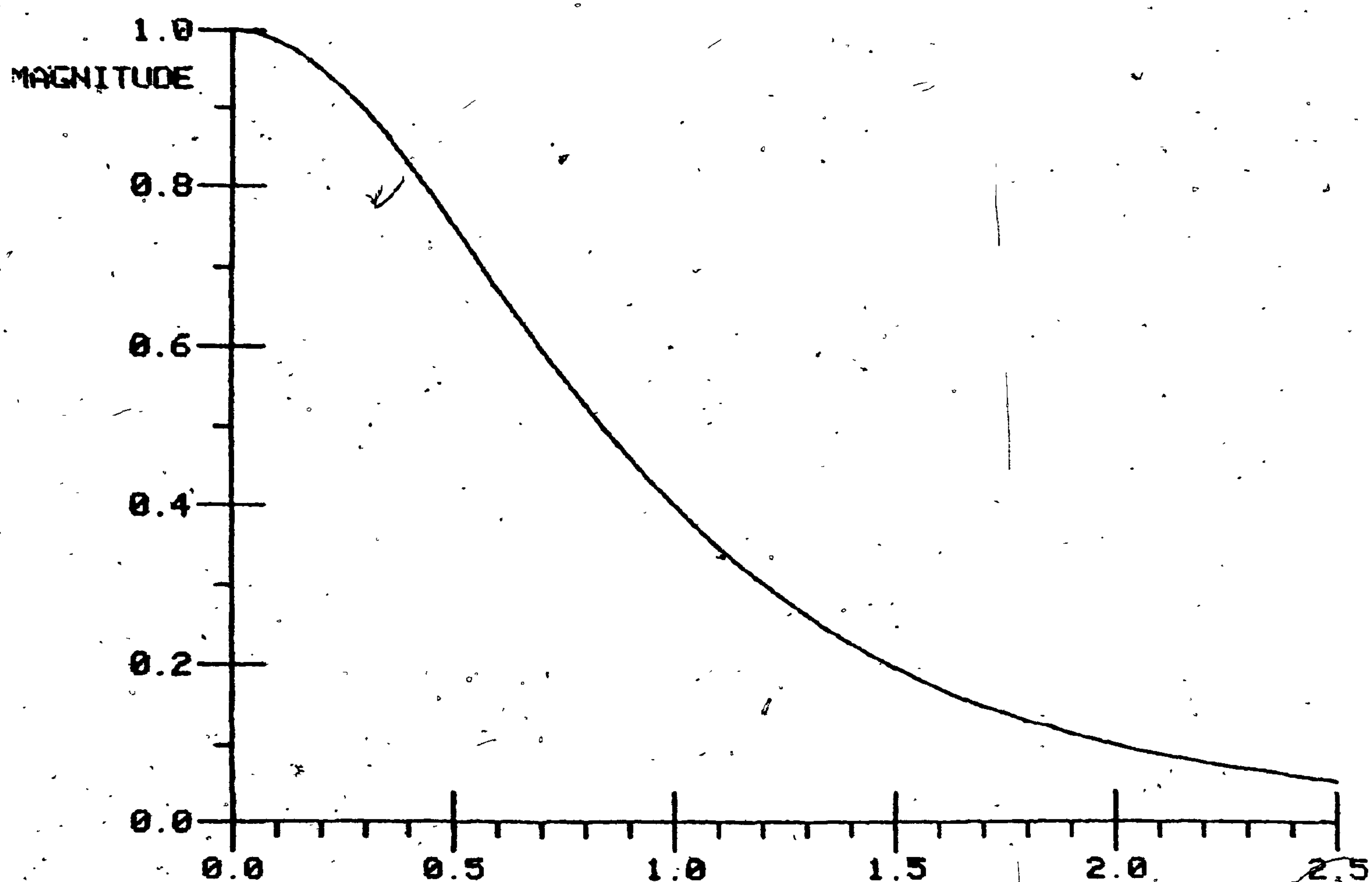

FIG. 5.1 MIGNITUDE RESPONSE OF 3RD ORDER FILTER.
WITH A1 = 3

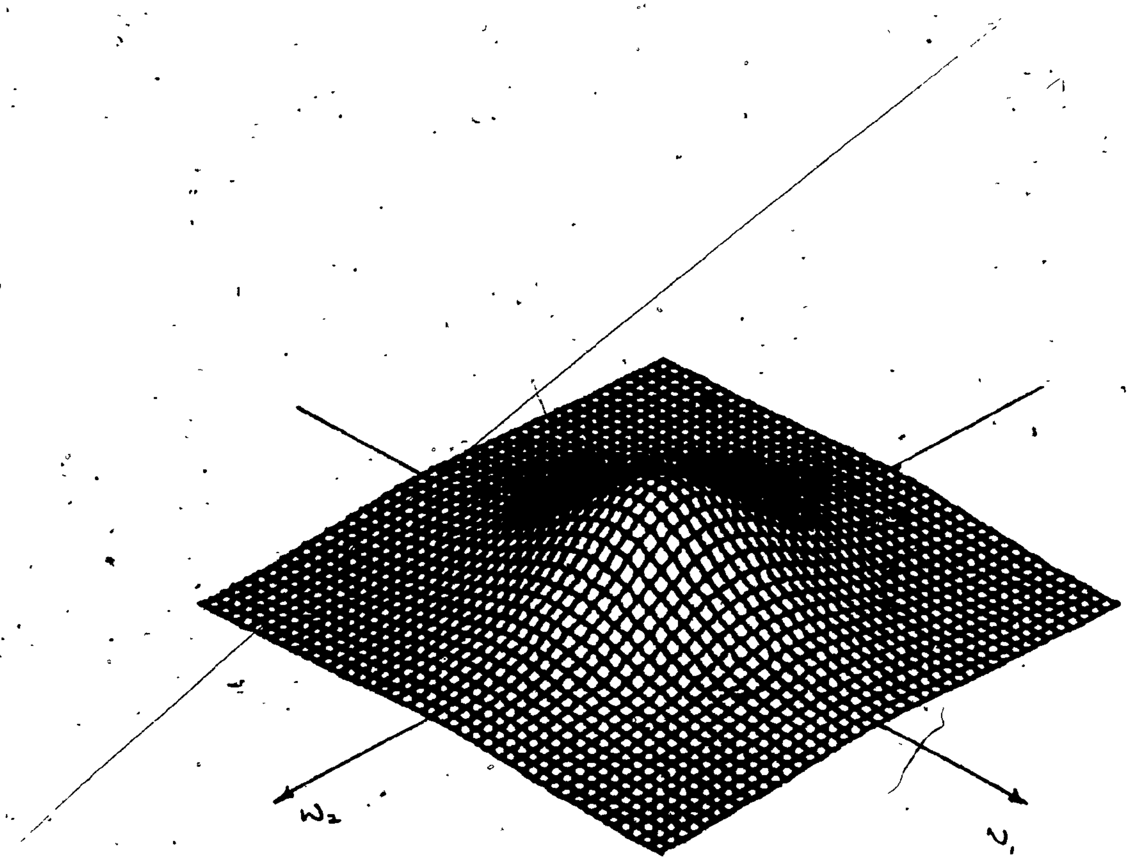

FIG. 5.2 PERSPECTIVE VIEW OF MAGNITUDE RESPONSE.

WITH ORDER N = 3
A1= 3
XC= 1

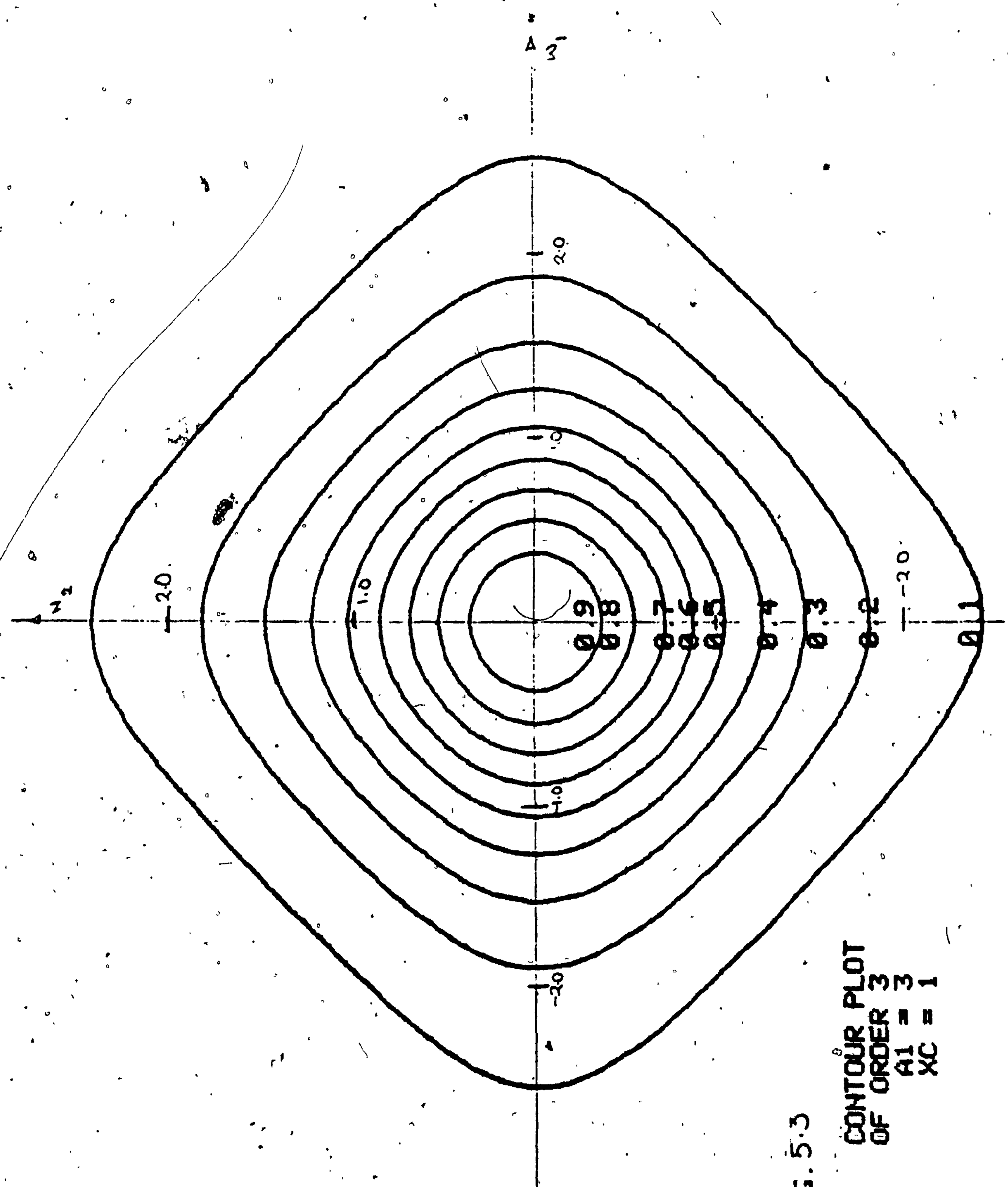

FIG. 5.3

CONTOUR PLOT
OF ORDER 3
A1 = 3
XC = 1

5.3 DESIGN EXAMPLE 2

Design a fifth-order approximation lowpass 2-D digital filter with approximately circularly symmetric response.

Solution: From chapter 3, we obtained the one-sided one-dimensional analog transfer function as

$$H_5(s) = \frac{s^2 + a_1 s + 4x_c}{s^3 + a_1 s^2 + 6x_c s + 2x_c a_1} \qquad (5.3.1)$$

After the error analysis, we obtained an optimum transfer function as

$$H_5(s) = \frac{s^2 + a_1 s + 4x_c}{s^3 + a_1 s^2 + 6x_c s + 0.5 a_1 x_c} \qquad (5.3.2)$$

If we choose $x_c = 1$

$a_1 = 5$

and $T_1 = T_2 = 2$

the corresponding transfer function will be

$$H_5(s) = \frac{s^2 + 5s + 4}{s^3 + 5s + 6s + 2.5} \qquad (5.3.3)$$

The pole and zero locations of the function as follow:

zeros at $s = -1.0$

and $s = -4.0$

poles at $s = -0.7582 - j0.3779$

$s = -0.7582 + j0.3779$

and $s = -3.4837 + j0.0$

Let p_1 , p_2 and p_3 be the poles of $H_5(s)$, the two-sided one-dimensional transfer function can be written as

$$H_{25}(s) = \frac{20}{(s+p_1)(s+p_1^*)(s+p_2)(s+p_2^*)(s+p_3)(s+p_3^*)} \tag{5.3.4}$$

where p_1^*, p_2^* and p_3^* are complex conjugates of p_1, p_2 and p_3 respectively.

Therefore, its two-sided two-dimensional analog transfer function will be

$$H_{2D5}(s_1,s_2) = H_{25}(s_1) \cdot H_{25}(s_2) \tag{5.3.5}$$

$$= \frac{20}{(s_1+p_{11})(s_1+p_{11}^*)(s_1+p_{21})(s_1+p_{21}^*)(s_1+p_{31})(s_1+p_{31}^*)} \cdot \frac{20}{(s_2+p_{12})(s_2+p_{12}^*)(s_2+p_{22})(s_2+p_{22}^*)(s_2+p_{32})(s_2+p_{32}^*)}$$

where p_{11}, p_{21}, p_{31}, p_{11}^*, p_{21}^* and p_{31}^* are poles at s_1-plane

p_{12}, p_{22}, p_{32}, p_{22}^*, and p_{32}^* are poles at s_2-plane.

Equation (5.3.5) can also be written as

$$H_{2D5}(s_1,s_2) = A\prod_{i=1}^{6}\frac{1}{(s_1+p_{i1})(s_2+p_{i2})} \qquad (5.3.6)$$

$$= A\prod_{i=1}^{6}\frac{1}{b_{00}^{(i)}+b_{01}^{(i)}s_2+b_{10}^{(i)}s_1+b_{11}^{(i)}s_1s_2} \qquad (5.3.7)$$

where $A = 400$

p_{i1} = ith pole in s_1-plane

p_{i2} = ith pole in s_2-plane

and

$$b_{00}^{(i)} = 1$$

$$b_{01}^{(i)} = p_{i1}$$

$$b_{10}^{(i)} = p_{i2}$$

$$b_{11}^{(i)} = p_{i1}p_{i2}$$

After applying double bilinear z-transformation to equation (5.3.7), we obtain

$$H_{2D5}(z_1,z_2) = A\prod_{i=1}^{6}\frac{1+z_1+z_2+z_1z_2}{a^{(i)}+b^{(i)}z_1+c^{(i)}z_2+d^{(i)}z_1z_2} \qquad (5.3.8)$$

where

$$a^{(i)} = b_{00}^{(i)} - b_{01}^{(i)}\cdot\frac{2}{T_2} - b_{10}^{(i)}\cdot\frac{2}{T_1} + b_{11}^{(i)}\cdot\frac{4}{T_1T_2}$$

$$b^{(i)} = b_{00}^{(i)} - b_{01}^{(i)}\cdot\frac{2}{T_2} + b_{10}^{(i)}\cdot\frac{2}{T_1} - b_{11}^{(i)}\cdot\frac{4}{T_1T_2}$$

$$c^{(i)} = b_{00}^{(i)} + b^{(i)}\cdot\frac{2}{T_2} - b^{(i)}\cdot\frac{2}{T_1} - b^{(i)}\cdot\frac{4}{T_1T_2}$$

$$d^{(i)} = b_{00}^{(i)} + b_{01}^{(i)} \cdot \frac{2}{T_2} + b_{10}^{(i)} \cdot \frac{2}{T_1} + b_{11}^{(i)} \cdot \frac{4}{T_1 T_2}$$

For i=1, we have

$$p_{11} = -0.7582 - j0.3779$$

$$p_{12} = -0.7582 + j0.3779$$

The corresponding digital function for this section is

$$T_1(z_1,z_2) = \frac{1 + z_1 + z_2 + z_1 z_2}{3.234+(0.2824+j0.7558)z_1+(0.2824-j0.7558)z_2+0.2013z_1z_2}$$

The stability test for this section is

constraint 1 :

$$\left|\frac{a^{(1)} + b^{(1)}}{c^{(1)} + d^{(1)}}\right| > 1$$

$$\left|\frac{3.234 + (0.2824 + j0.7558)}{0.2824-j0.7558 + 0.2013}\right| > 1$$

which is satisfied.

Constraint 2 :

$$\left|\frac{a^{(1)} - b^{(1)}}{c^{(1)} - d^{(1)}}\right| > 1$$

$$\left|\frac{3.234 - (0.2824+j0.7558)}{0.2824- j0.7558 - 0.2013}\right| > 1$$

which is satisfied.

and

constraint 3
$$\left|\frac{a^{(1)}}{b^{(1)}}\right| > 1$$

$$\left|\frac{3.234}{0.2824+j0.7558}\right| > 1$$

where constraint 3 also satisfied. Thus, section 1 is stable.

For i = 2, we have

$$p_{21} = -0.7582 + j0.3779$$

$$p_{22} = -0.7582 - j0.3779$$

The corresponding digital function for this section is.

$$T_2(z_1,z_2)= \frac{1+z_1+z_2+z_1z_2}{3.234+(0.2824-j0.7558)z_1+(0.2824+j0.7558)z_2+0.2013z_1z_2}$$

and the stability test for this section as

constraint 1 :

$$\left|\frac{a^{(2)} + b^{(2)}}{c^{(2)} + d^{(2)}}\right| > 1$$

$$\left|\frac{3.234+(0.2824-j0.7558)}{0.2824+j0.7558+0.2013}\right| > 1$$

which is satisfied.

constraint 2 :

$$\left|\frac{a^{(2)} - b^{(2)}}{c^{(2)} - b^{(2)}}\right| > 1$$

$$\left|\frac{3.234 - (0.2824 - j0.7558)}{0.2824 + j0.7558 - 0.2013}\right| > 1$$

which is satisfied.

constraint 3 :

$$\left|\frac{a^{(2)}}{b^{(2)}}\right| > 1$$

$$\left|\frac{3.234}{0.2824 - j0.7558}\right| > 1$$

where constraint 3 also satisfied. Thus, section 2 is stable.

For $i = 3$, we have

$$p_{31} = -3.34837 + j\,0.0$$

$$p_{32} = -3.34837 + j\,0.0$$

Then the corresponding digital function for this section is

$$T_3(z_1,z_2) = \frac{1 + z_1 + z_2 + z_1 z_2}{20.1 - 11.14z_1 - 11.14z_2 + 6.169z_1 z_2}$$

and the stability test for this section as follow:

constraint 1 :

$$\left|\frac{a^{(3)} + b^{(3)}}{c^{(3)} + d^{(3)}}\right| > 1$$

$$\left|\frac{20.1 - 11.14}{-11.14 + 6.169}\right| > 1$$

which is satisfied.

constraint 2 :

$$\left|\frac{a^{(3)} - b^{(3)}}{c^{(3)} - b^{(3)}}\right| > 1$$

$$\left|\frac{20.1+11.14}{-11.14-6.169}\right| > 1$$

which is satisfied.

constraint 3 :

$$\left|\frac{a^{(3)}}{b^{(3)}}\right| > 1$$

$$\left|\frac{20.1}{-11.14}\right| > 1$$

where constraint 3 also satisfied. Thus, section 3 is stable.

The poles for sections 4, 5 and 6 are complex conjugate of the poles of sections 1, 2 and 3. Since section 1, 2 and 3 are stable then section 4, 5 and 6 are also stable. Thus, this fifth-order appoximation filter is stable. The magnitude response of this fifth-order 2-D digital filter as shown in figure 5.4, 5.5 and 5.6.

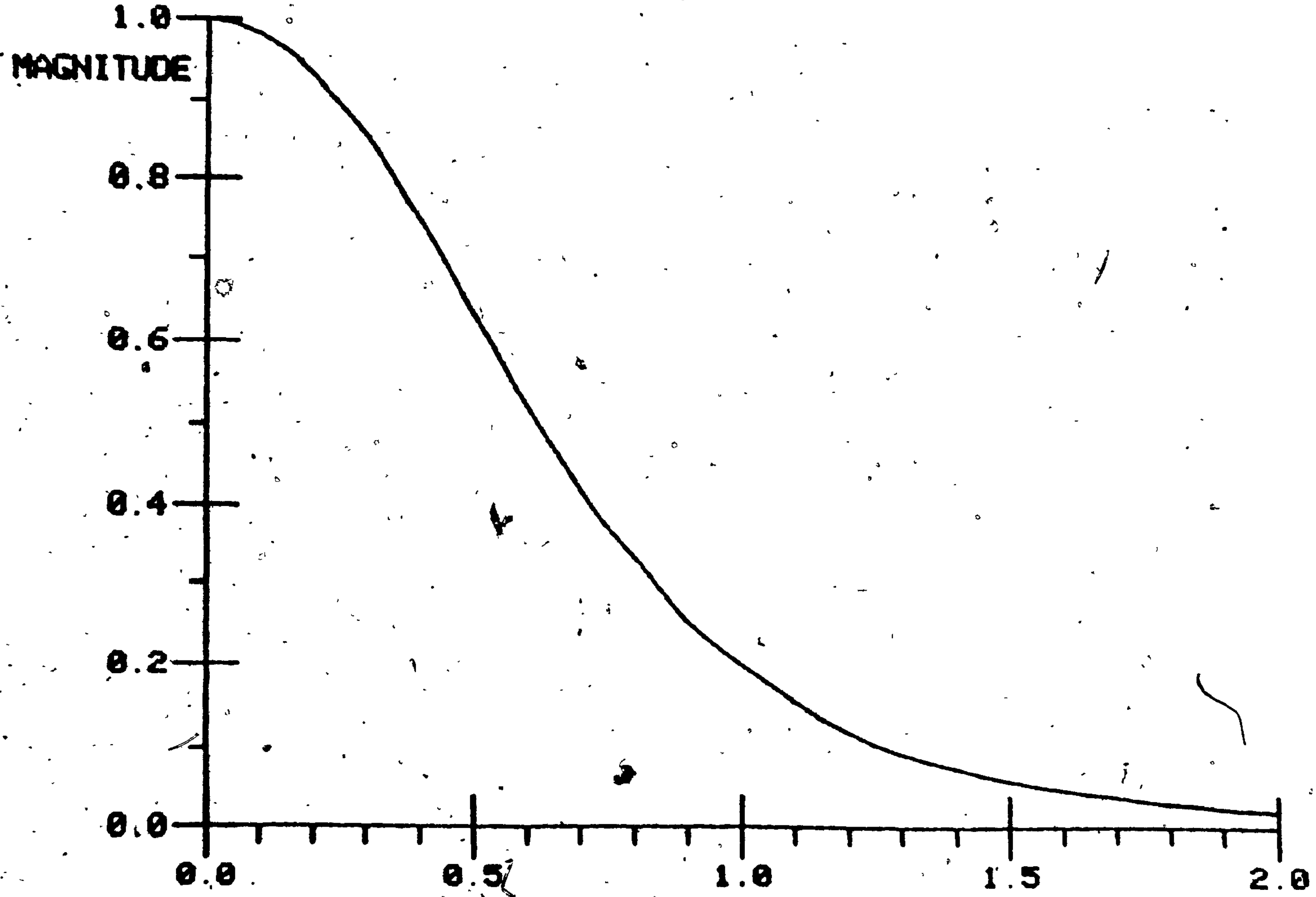

FIG.5.4 MAGNITUDE RESPONSE OF 5TH ORDER LOWPASS FILTER.
WITH A1 = 5

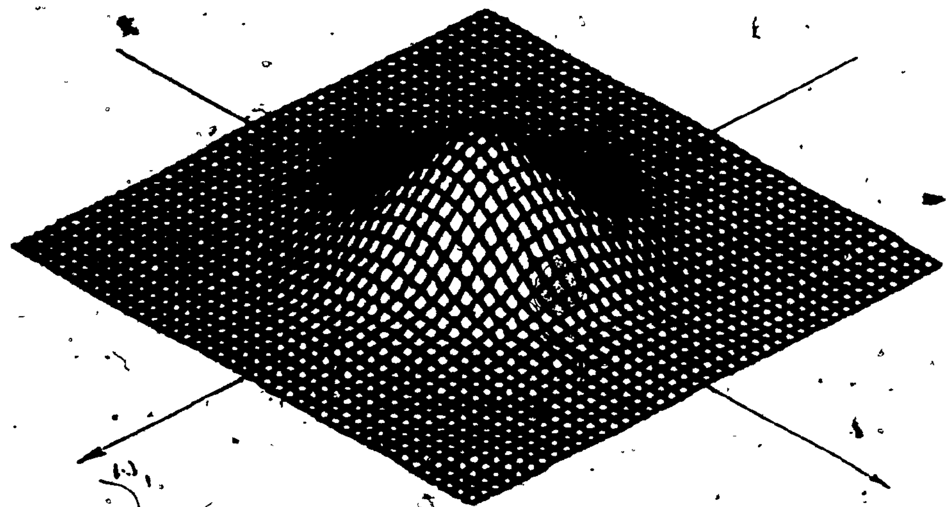

FIG.5.5 PERSPECTIVE VIEW OF MAGNITUDE RESPONSE.
WITH ORDER N = 5
A1= 5

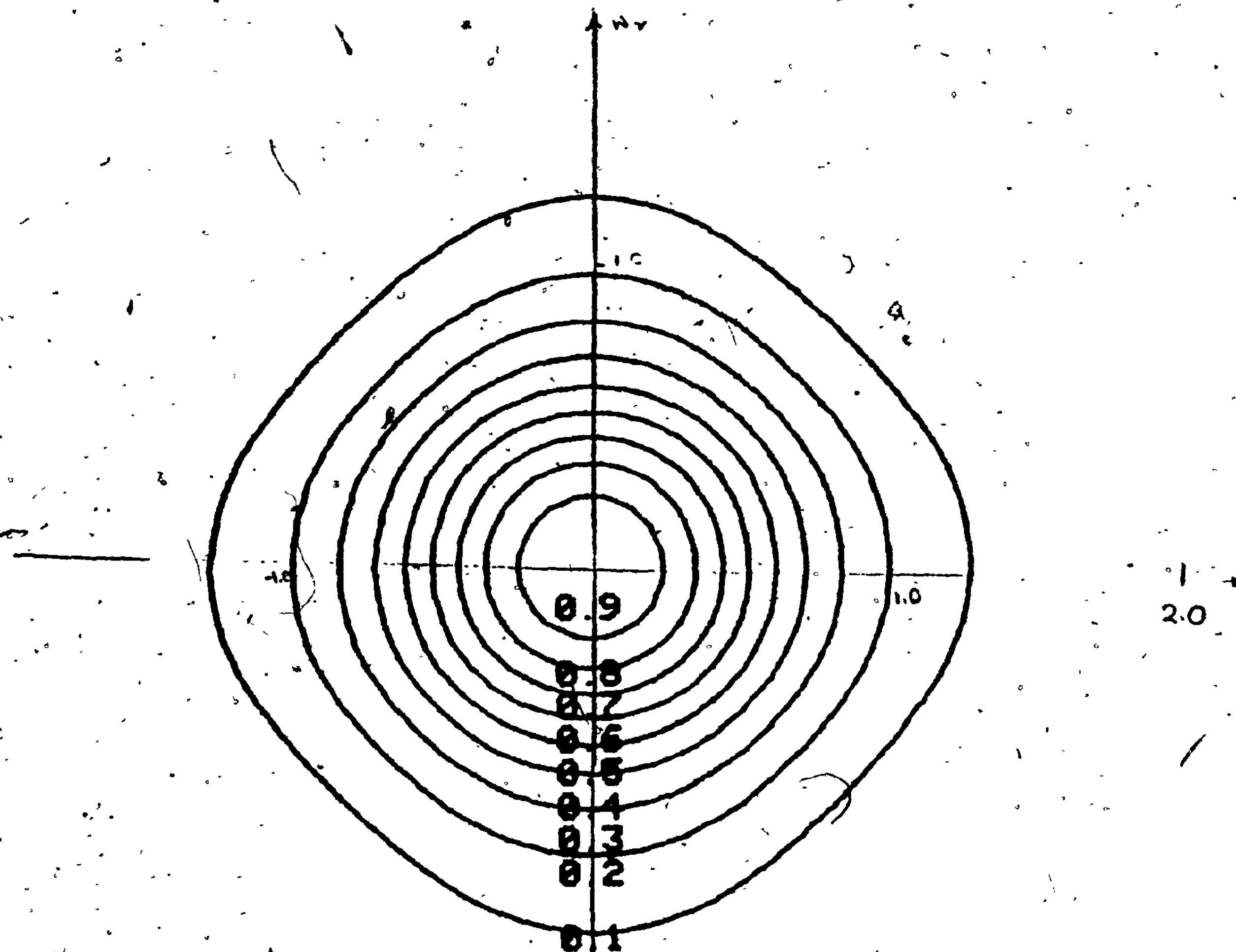

FIG.5·6 CONTOUR PLOT OF 5TH ORDER LOWPASS FILTER.

WITH A1 = 5

CHAPTER 6

COMPUTER-AIDED DESIGN OF RECURSIVE DIGITAL FILTERS

6.1 INTRODUCTION

As indicated earlier, the technique for the design of 2-D digital filters with approximately circularly symmetric magnitude response described above is straightforward and computationally efficient. What we propose in this chapter is to add to the normal design procedure, a computer analysis step for the sole purpose of the problems of solving simultaneous equations, finding the zero and pole patterns of the required function, performing bilinear z-transformation .... etc. We shall see that the addition of this computer analysis step is highly effective.

6.2 COMPUTER-AIDED DESIGN PROGRAM

From the previous design examples, it has been shown the computer-aided program is quite efficient and successful. It should be noted that the procedure suggested in this report, various levels of approximations are involved. In the first level, it generates a set of simultaneous equations for a given order of approximation, and this approximation always leads to a globally optimum solution with respect to a separable approximation of the 2-D magnitude specification (circulary symmetric). In the second level of approximation, a computer-aided technique is used for solving the simultaneous equations which are obtained in the previous step, and then, to obtain a one-dimensional transfer function which is "close" to the 1-D magnitude specifications. In the third level, a computer subroutine will compute the zero and pole patterns of the transfer function, which are obtained in the previous step, and then, to form a number of second-order elementary 2-D filters. In the fourth level, the computer subprogram will provide a) the magnitude response, b) contour plot, and c) perspective plot of the resulting filter function. The basic flowchart of program is shown in figure 6.1.

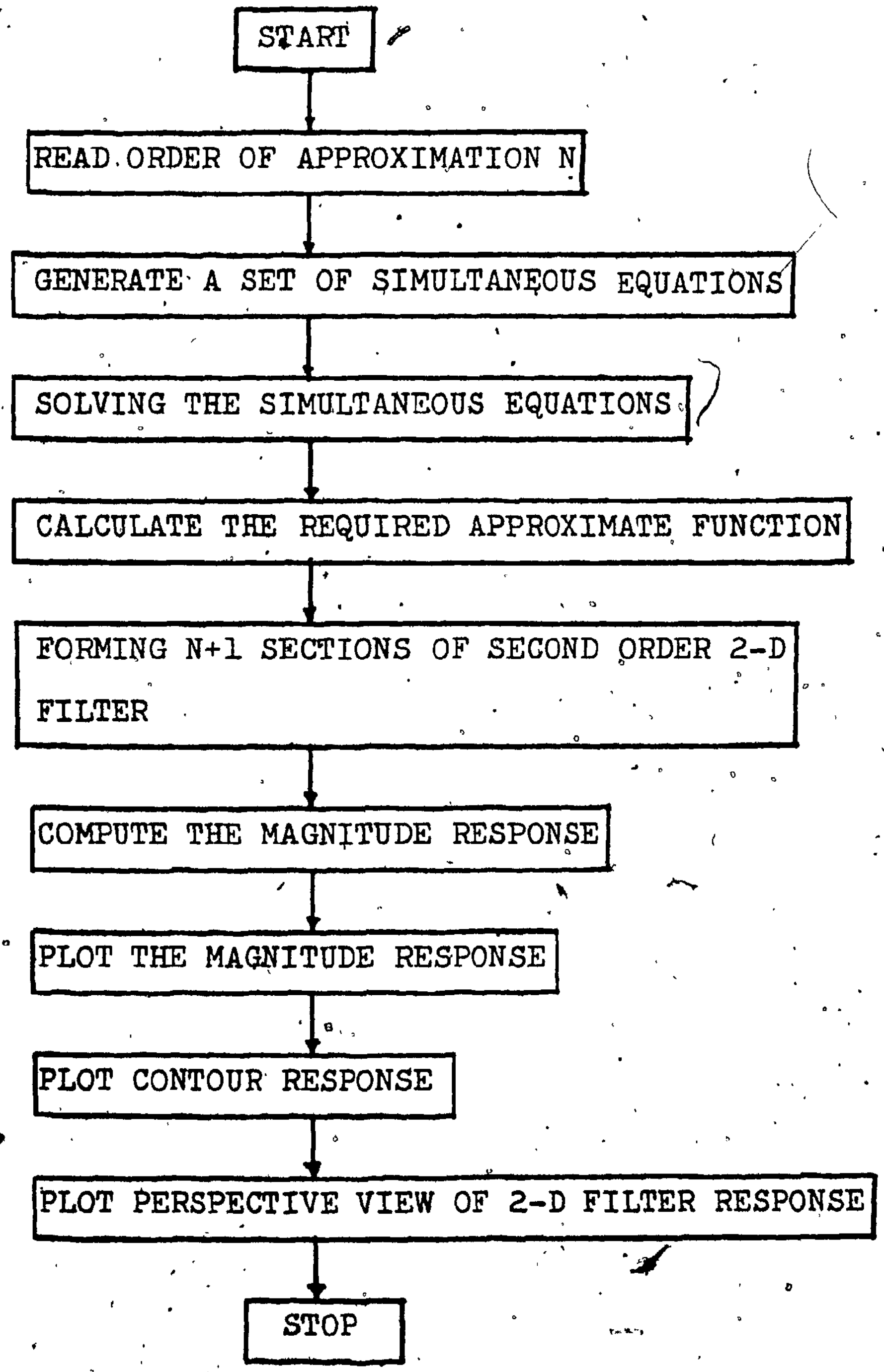

Figure 6.1 computer aided design flowchart.

CHAPTER 7

CONCLUSION

The work described previously concerns the design of stable two-dimensional recursive filters whose magnitude response approximates a circularly symmetric function. The Taylor-Series-Expansion about the origin of this recursive transfer function in time domain has been examined, and a direct and simple technique for expressing the general analog function in term of the coefficients of the Laplace transform of this approximating function. By cascading two such filters, one with variable s_1 and the other with variable s_2, we obtained a required 2-D transfer function with the required approximate magnitude response.

The method has been developed into a computer aided-design program. The structure and the use of these programs have been described. Some practical examples have been included to show the accuracy of this approximating technique. The principle advantages of the method are the following:

1) Stability can be guaranteed.

2) The filters are factorable into lower order filters, so that they can be easily realized.

3) The calculation of the coefficients of a filter using a digital computer is fast.

Finally, we have given some various computer programs for CDC-6600 at Concordia University which yield very good results. Further work could be done on frequency domain synthesis in order to control the desired cutoff frequency in each given direction.

REFERENCE

1. Karivaratharajan, P., and Swamy, M.N.S., "Approximation of Circularly Symmetric Two-dimensional Lowpass Recursive Digital Filters", Proceding of IEEE International Symposium on Circuits and Systems, New York, U.S.A. pp. 48-485, 1978.

2. Huang, T.S., Schreiber, W.F., and Tretiak O., "Image Processing," proceding of IEEE, vol.59, pp.1586-1609, Nov. 1971.

3. Crochiere, R.E., and Oppenheim, A.V., "Analysis of Linear Digital Networks," Proceding of IEEE, vol. pp.581-595, April, 1975.

4. Rajan, P.K., and Swamy, M.N.S., "Symmetry Constraints on Two-Dimensional Half-plane Digital Transfer Functions," IEEE Trans. on Acoustic Speech and Signal Proc., vol. Assp-27, no.5, pp.506-511, Oct.1979.

5. Shanks, J.L., "Two-Dimensional Recursive Filters," in 1969 Swieeco Rec. pp.19E1-19E8.

6. Shanks, J.L., and Justice, J.H., "Stability and Synthesis of Two-Dimensional Recursive Filters," IEEE Trans. Audio Electrocoast., vol. AU-20, pp.115-128, June, 1972.

7. Huang, T.S., "Stability of Two-Dimensional Recursive Filters," IEEE Trans. Audio Electrocoast., vol. AU-20, no.2, pp.158-163, June, 1972.

8. Rabiner, L.R., and Gold, B., "Theory and Applications of Digital Signal Processing," Prentice Hall Inc., Englewood Cliffs, N.J. 1975, CH.7.

9. Valkenburg, V., "Introduction to Modern Network Synthesis," John Wiley, New York, 1960.

10. Tuttle, D.F., "Network Synthesis," John Wiley, London, 1958.

11. Guillemin, E.A., "Synthesis of Passive Networks," John Wiley, New York, 1957.

12. Cappellini, V. and Constantinides, G., "Digital Filters and Their Applications," Academic press, London, 1978.

13. Karivaratharajan, P., and Swamy, M.N.S., "Some Results on the Nature of a Two-Dimensional Filter Function Possesing Certain Symmetry in its Magnitude Response," IEEE. Elect. and Circuit and Systems, vol. 2, no.5, pp.147-153, Sept. 1978.

14. Merserau, R.M., and Dudgeon, D.E., "The Representation of Two-Dimensional Sequences as One-Dimensional Sequences," IEEE Trans. Acoust., Speech, and Sig. Proc., vol. ASSP-22, no. 5, pp. 320-325, Oct. 1974.

15. Maria, G.A., and Fahmy, M.M., "An Lp Design Technique for Two-Dimensional Digital Recursive Filters," IEEE. Trans. on Acoustic, vol. ASSP-22, no. 1, pp. 15-21, Feb. 1974.

16. Maria, G.A., and Fahmy, M.M., "Effect of Rounding on the Stability of the Two-Dimensional Digital Filters," IEEE Trans. Circuits System, July, 1974.

17. Knowles, J.B., and Olcayto E.M., "Coefficient Accuracy and Digital Filters Response," IEEE Trans. Circuit Theory, vol. CT-14, pp. 31-41, March, 1968.

APPENDIX A

Definition 1: A Mth order approximation of Taylor Series is a approximation function which will give a result good through S^M terms in Taylor Series expansion. M is defined as

$$M = 2n - 1 \qquad \text{for } n \geq 2$$

or

$$n = (M + 1) / 2 \qquad \text{for } M \geq 3 \text{ and odd}$$

Definition 2: A rational function

$$H_M(s) = \frac{\sum_{i=0}^{n-1} a_i s^{n-1-i}}{\sum_{j=0}^{n} b_j s^{n-j}} \qquad \text{(A3.1)}$$

where a_0 and b_0 are unity whose long division is equivalent to the Taylor Series expansion of a given function only if the long division of the equation (A3.1) is

$$H(s) = \frac{1}{s} + \frac{a_1 - b_1}{s^2} + \frac{(a_2 - b_2) - b_1(a_1 - b_1)}{s^3}$$

$$+ \frac{(\,(a_3 - b_3) - b_2(a_1 - b_1)\,) - (\,(a_2 - b_2) - b_1(a_1 - b_1)\,)}{s^4}$$

$$+ \frac{\begin{array}{l} (((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1))) \end{array}}{s^5}$$

$$+ \frac{\begin{array}{l} ((((a_5 - b_5) - b_4(a_1 - b_1)) - b_3((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_2(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \\ -b_1((((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \end{array}}{s^6}$$

$$+ \frac{\begin{array}{l} (((((a_6 - b_6) - b_5(a_1 - b_1)) - b_4((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_3(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \\ -b_2((((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1))))) \\ -b_1(((((a_5 - b_5) - b_4(a_1 - b_1)) - b_3((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_2(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))) \\ -b_1((((a_4 - b_4) - b_3(a_1 - b_1)) - b_2((a_2 - b_2) - b_1(a_1 - b_1))) \\ -b_1(((a_3 - b_3) - b_2(a_1 - b_1)) - b_1((a_2 - b_2) - b_1(a_1 - b_1)))))) \end{array}}{s^7}$$

$$+ \cdots\cdots\cdots \qquad \text{as } s^N \to \infty \qquad (A3.3)$$

The Laplace transformation of Taylor Series of a continuous function $f(x) = e^{-x_c t^2}$ as follow

$$H(s) = \frac{1}{s} - \frac{2x_c}{s^3} + \frac{12x_c^2}{s^5} - \frac{120x_c^3}{s^7} + \frac{1680x_c^4}{s^9} - \frac{30240x_c^5}{s^{11}} + \dots +$$

$$\frac{N!x^{\frac{n-1}{2}}}{s^n} \quad \text{as} \quad n \rightarrow \infty \qquad (A3.4)$$

By comparing equation (A3.3) and (A3.4), we obtained following relations

$$a_1 - b_1 = 0 \qquad (A3.5)$$

$$a_2 - b_2 = -2x_c \qquad (A3.6)$$

$$a_3 - b_3 = -2x_c b_1 \qquad (A3.7)$$

$$a_4 - b_4 = 12x_c^2 - 2b_2 x_c \qquad (A3.8)$$

$$a_5 - b_5 = 12x_c^2 b_1 - 2x_c b_3 \qquad (A3.9)$$

$$a_6 - b_6 = -120x_c^3 - 2x_c b_4 + 12x_c^2 b_2 \qquad (A3.10)$$

$$a_7 - b_7 = -120x_c^3 b_1 + 12x_c^2 b_3 - 2x_c b_5 \qquad (A3.11)$$

$$a_8 - b_8 = 1680x_c^4 - 120x_c^3 b_2 + 12x_c^2 b_4 - 2x_c b_6 \qquad (A3.12)$$

$$\vdots \qquad \vdots$$

$$a_\infty - b_\infty = \dots\dots$$

The values of a_i's and b_j's of equation (A3.1) can be easily obtained through same procedures as following examples:

EXAMPLE 1 : Design a third-order approximation function which possesses approximately circularly symmetric response.

Solution For N=3, we have

$$n = \frac{N+1}{2} = 2$$

Then the required rational transfer function will be

$$H_3(s) = \frac{\sum_{i=}^{1} a_i s^{2-1-i}}{\sum_{j=0}^{2} b_j s^{2-j}}$$

$$= \frac{a_0 s + a_1}{b_0 s^2 + b_1 s + b_2}$$

$$= \frac{s + a_1}{s^2 + b_1 s + b_2}$$

From the relations of equations (A3.3) and (A3.4), we have following set of equations

$$a_1 - b_1 = 0$$

$$a_2 - b_2 = -2x_c$$

Since $a_2 = 0$, then

$$b_2 = 2x_c$$

and $a_1 = b_1$

The final required function is

$$H_3(s) = \frac{s + a_1}{s^2 + a_1 s + 2x_c}$$

This function will be generate a family of $H_3(s)$ for any positive value of parameter a_1.

EXAMPLE 2: Design a fifth-order approximation function which possesses approximately circularly symmetric response.

Solution For $N = 5$, we have

$$n = \frac{N+1}{2} = 3$$

Then the required rational transfer function will be

$$H_5(s) = \frac{\sum_{i=0}^{2} a_i s^{3-1-i}}{\sum_{j=0}^{3} b_j s^{3-j}}$$

$$H_5(s) = \frac{a_0s^2 + a_1s + a_2}{b_0s^3 + b_1s^2 + b_2s + b_3}$$

$$= \frac{s^2 + a_1s + a_2}{s^3 + b_1s^2 + b_2s + b_3}$$

From the relations of equations (A3.3) and (A3.4), we have following set of equations

$$a_1 - b_1 = 0$$

$$a_2 - b_2 = -2x_c$$

$$a_3 - b_3 = -2x_cb_1$$

$$a_4 - b_4 = 12x_c^2 - 2b_2x_c$$

Since $a_4 = 0$, $b_4 = 0$, $a_5 = 0$, and $b_5 = 0$, then

$$a_1 - b_1 = 0$$

$$a_2 - b_2 + 2x_c = 0$$

$$-b_3 + 2x_cb_1 = 0$$

$$2b_2x_c - 12x_c^2 = 0$$

Solving this set of simultaneous equations obtained

$$a_1 = b_1$$

$$b_2 = 6x_c$$

$a_2 = 4x_c$

$b_3 = 6b_1x_c$

finally obtain the required transfer function

$$H_5(s) = \frac{s^2 + a_1 s + 4x_c}{s^3 + a_1 s^2 + 6x_c s + 2a_1 x_c}$$

EXAMPLE 3 : Design a seventh-order approximation function which possesses approximately circularly symmetric response.

Solution For N=7, we have

$$n = \frac{N+1}{2} = 4$$

Then the required rational transfer function will be

$$H_7(s) = \frac{\sum_{i=0}^{3} a_i s^{4-1-i}}{\sum_{j=0}^{4} b_j s^{4-j}}$$

$$= \frac{a_0 s^3 + a_1 s^2 + a_2 s + a_3}{b_0 s^4 + b_1 s^3 + b_2 s^2 + b_3 s + b_4}$$

$$H_7(s) = \frac{s^3 + a_1 s^2 + a_2 s + a_3}{s^4 + b_1 s^3 + b_2 s^2 + b_3 s + b_4}$$

From the relations of equations (A3.3) and (A3.4), we have following set of equations

$$a_1 - b_1 = 0$$

$$a_2 - b_2 = -2x_c$$

$$a_3 - b_3 = -2x_c b_1$$

$$a_4 - b_4 = 12x_c^2 - 2b_2 x_c$$

$$a_5 - b_5 = 12x_c^2 b_1 - 2x_c b_3$$

$$a_6 - b_6 = -120x_c^3 - 2x_c b_4 + 12x_c^2 b_2$$

Since $a_4 = a_5 = b_5 = a_6 = b_6 = 0$

then

$$a_1 - b_1 = 0$$

$$a_2 - b_2 = -2x_c$$

$$a_3 + 2x_c b_1 - b_3 = 0$$

$$2b_2 x_c - b_4 = 12x_c^2$$

$$-12x_c^2 b_1 - 2x_c b_3 = 0$$

$$12x_c^2 b_2 - 2x_c b_4 = 120x_c^3$$

By solving this set of simultaneous equations, obtained

$a_1 = b_1$

$a_2 = 10x_c$

$b_2 = 12x_c$

$a_3 = 4x_c a_1$

$b_3 = 6x_c a_1$

$a_4 = 12x_c^2$

Finally, obtained the required transfer function

$$H_7(s) = \frac{s^3 + a_1 s^2 + 10x_c s + 4x_c a_1}{s^4 + a_1 s^3 + 12x_c s^2 + 6x_c a_1 s + 12x_c^2}$$

EXAMPLE 4 : Design a ninth-order approximation function which possesses approximately circularly symmetric response.

Solution For N=9, we have

$$n = \frac{N+1}{2} = 5$$

Then the required rational transfer function will be

$$H_9(s) = \frac{\sum_{i=0}^{4} a_i s^{5-1-i}}{\sum_{j=0}^{5} b_j s^{5-j}}$$

$$H_9(s) = \frac{a_0s^4 + a_1s^3 + a_2s^2 + a_3s + a_4}{b_0s^5 + b_1s^4 + b_2s^3 + b_3s^2 + b_4s + b_5}$$

$$= \frac{s^4 + a_1s^3 + a_2s^2 + a_3s + a_4}{s^5 + b_1s^4 + b_2s^3 + b_3s^2 + b_4s + b_5}$$

From the relations of equations (A3.3) and (A3.4), we have following set of equations

$a_1 - b_1 = 0$

$a_2 - b_2 = -2x_c$

$a_3 - b_3 = -2x_cb_1$

$a_4 - b_4 = 12x_c^2 - 2b_2x_c$

$a_5 - b_5 = 12x_c^2b_1 - 2x_cb_3$

$a_6 - b_6 = -120x_c^3 - 2x_cb_4 + 12x_c^2b_2$

$a_7 - b_7 = -120x_c^3b_1 + 12x_c^2b_3 - 2x_cb_5$

$a_8 - b_8 = 1680x_c^4 - 120x_c^3b_2 + 12x_c^2b_4 - 2x_cb_6$

Since, $a_5 = a_6 = b_c = a_7 = b_7 = a_8 = b_8 = 0$

then

$a_1 - b_1 = 0$

$a_2 - b2 = -2x_c$

$a3 + 2x_cb_1 - b_3 = 0$

$$a_4 + 2b_2x_c - b_4 = 12x_c^2$$

$$-12x_c^2b_1 + 2x_cb_3 - b_5 = 0$$

$$12x_c^2b_2 - 2x_cb_4 = 120x_c^3$$

$$120x_c^3b_1 - 12x_c^2b_3 + 2x_cb_5 = 0$$

$$120x_c^3b_2 - 12x_c^2b_4 = 1680x_c^4$$

Solving this set of equations, obtained

$$a_1 = b_1$$

$$a_2 = 18x_c$$

$$b_2 = 20x_c$$

$$a_3 = 10x_ca_1$$

$$b_3 = 12x_ca_1$$

$$a_4 = 32x_c^2$$

$$b_4 = 60x_c^2$$

$$b_5 = 12x_c^2a_1$$

Finally, the required transfer function is

$$H_9(s)= \frac{s^4 + a_1s^3 + 18x_cs^2 + 10x_ca_1s + 32x_c^2}{s^5 + a_1s^4 + 20x_cs^3 + 12x_ca_1s^2 + 60x_c^2s + 12x_c^2a_1}$$

APPENDIX B

Shank's theorem of stability:

A causal recursive filter with z-transform

$$H(z_1, z_2) = \frac{A(z_1, z_2)}{B(z_1, z_2)}$$

where A and B are polynomial of z_1 and z_2 is stable if and only if there are no value of z_1 and z_2 such that $B(z_1, z_2) = 0$ and $|z_1| \leq 1$ and $|z_2| \leq 1$.

APPENDIX C

Huang Stability Theorem: A simplified version of Shanks Theorem

A causal filter with a z transform

$$H(z_1, z_2) = \frac{A(z_1, z_2)}{B(z_1, z_2)}$$

where A and B are polynomials in z_1 and z_2 is stable if and only if

1) the map of $2d_1 \equiv (z_1: |z_1| = 1)$ in the z_2 plane, according to $B(z_1, z_2) = 0$ lies outside $d_2 \equiv (z_2: |z_2| \leq 1)$;

and

2) no point in $d_1 = (z_1: |z_1| \leq 1)$ maps into the point $z_2 = 0$ by the relation $B(z_1, z_2) = 0$.

APPENDIX D

COMPUTER-AIDED DESIGN PROGRAM

```
      PROGRAM CIRCULA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
*
*
*      PROGRAM CIRCULA IS A COMPUTER-AIDED FILTER DESIGN
*      PROGRAM.  WHICH DESIGN A NTH ORDER TWO DIMENSIONAL
*      DIGITAL FILTER WITH APPROXIMATLY CIRCULARLY SYMMETRIC
*      MAGNITUDE RESPONSE.
*
*
*
      DIMENSION POLE1R(40),POLE1I(40),POLE2R(40),POLE2I(40)
      DIMENSION XCOF(40),ROOTR(40),ROOTI(40),H(101,101),HPD(301)
      DIMENSION X(40),Y(40),HD(41,41)
      COMPLEX DS00,DS01,DS10,DS11,HSK1K2,S1,S2
      COMPLEX B00(40),B01(40),B10(40),B11(40)
      COMPLEX A(40),B(40),C(40),D(40)
      COMPLEX Z1,Z2,HZ(41,41),TZN(40),TZD(40),TZ(40)
*
*
*     INPUT ORDER OF APPROXIMATION IN TAYLOR SERIES
*     FOR THE DESIGN FILTER
*
*
*
      PRINT 10
   10 FORMAT(5X,*ENTER ORDER OF APPROXIMATION  M *)
      READ*,M
*      INPUT THE XC VALUE FOR THE FILTER
      PRINT 20
   20 FORMAT(5X,*ENTER THE XC VALUE *)
      READ*,XC
*
*
*
*      INPUT A1 VALUE    (NOTE: X(2)=A1)
      PRINT 25
   25 FORMAT(5X,*ENTER THE VALUE FOR A1*)
      READ*,X(2)
*
*      INPUT THE SAMPLING PERIOD FOR T1 AND T2
*
      PRINT 27
   27 FORMAT(5X,*ENTER THE SAMPLING PERIOD FOR T1 AND T2*)
      READ*,T1,T2
      PRINT 21
   21 FORMAT(5X,*ENTER FILTER OPTION 0=DIGITAL,1=ANALOG*)
      READ*,IDIGAN
      PRINT 22
   22 FORMAT(5X,*ENTER THE MAGNITUDE PLOT OPTION*)
      PRINT 23
```

```
23      FORMAT(5X,*1=MAGNITUDE PLOT*)
        PRINT 24
24      FORMAT(5X,*2=CONTOUR PLOT*)
        PRINT 26
26      FORMAT(5X,*3=PERSPECTIVE VIEW PLOT*)
        READ*,IMCPLOT
        PRINT 28
28      FORMAT(5X,*ENTER NUMBER OF POINTS *)
        READ*,NPOINT
        PRINT 29
29      FORMAT(5X,*ENTER FREQUENCY SCALE FACTOR*)
        READ*,SCALE
        IF(M.LE.1)STOP
        IF(MOD(M,2).EQ.0)STOP
        IF(M.EQ.3) GO TO 30
        IF(M.EQ.5) GO TO 50
        IF(M.EQ.7) GO TO 70
        IF(M.EQ.9) GO TO 90
30      X(1)=1.0
        X(2)=X(2)
        Y(1)=1.0
        Y(2)=X(2)
        Y(3)=2*XC
        G=2*X(2)*Y(3)
        GO TO 100
50      X(1)=1.0
        X(2)=X(2)
        X(3)=4*XC
        Y(1)=1.0
        Y(2)=X(2)
        Y(3)=6*XC
        Y(4)=0.5*XC*X(2)
        G=2*X(3)*Y(4)
        GO TO 100
70      X(1)=1.0
        X(2)=X(2)
        X(3)=10*XC
        X(4)=4*XC*X(2)
        Y(1)=1.0
        Y(2)=X(2)
        Y(3)=12*XC
        Y(4)=6.0*XC*X(2)
        Y(5)=6.0*XC**2
        G=2*X(4)*Y(5)
        GO TO 100
90      X(1)=1.0
        X(2)=X(2)
        X(3)=18*XC
        X(4)=10*XC*X(2)
        X(5)=32*XC**2
```

```
      Y(1)=1.0
      Y(2)=X(2)
      Y(3)=20.0*XC
      Y(4)=12*XC*X(2)
      Y(5)=65.0*XC**2
      Y(6)=4.0*XC*XC*X(2)
      G=2*X(5)*Y(6)
*
*
*
*      COMPUTE THE ZEROS OF THE 1-S-1-D APPROXIMATE FUNCTION
*
*
*
*
  100 I=(M+1)/2
      DO 110 K=1,I
      XCOF(I-K+1)=X(K)
  110 CONTINUE
      I=I-1
      CALL ROOT(XCOF,I,ROOTR,ROOTI,IER)
*
*      PRINT THE ZEROS OF 1-S-1-D APPROXIMATE FUNCTION
*
      PRINT 120
  120 FORMAT(5X,*THE ZEROS OF 1-S-1-D APPROXIMATE FUNCTION IS*,/)
      PRINT 130,(K,ROOTR(K),ROOTI(K),K=1,I)
  130 FORMAT(5X,I2,*  ZERO = *,F10.4,*  J *,F10.4)
*
*
*      COMPUTE THE POLES OF THE 1-S-1-D APPROXIMATE FUNCTION
*
      J=(M+1)/2+1
      DO 140 K=1,J
      XCOF(J-K+1)=Y(K)
  140 CONTINUE
      J=J-1
      CALL ROOT(XCOF,J,ROOTR,ROOTI,IER)
*
*      PRINT THE POLES OF 1-S-1-D APPROXIMATE FUNCTION
*
      PRINT 150
  150 FORMAT(5X,*THE POLES OF 1-S-1-D APPROXIMATE FUNCTION IS*,/)
      PRINT 160,(K,ROOTR(K),ROOTI(K),K=1,J)
  160 FORMAT(5X,I2,*  POLE = *,F10.4,*  J *,F10.4)
*
*      COMPUTE THE 2-S-1-D FUNCTION
*
      G=G
      DO 170 K=1,J
```

```
      ROOTR(J+K)=-ROOTR(K)
      ROOTI(J+K)=-ROOTI(K)
  170 CONTINUE
*
      JJ=2*J
*
*      PRINT THE REQUIRE 2-S-1-D APPROXIMATE FUNCTION
*
      PRINT 180
  180 FORMAT(5X,*THE REQUIRED 2-S-1-D APPROXIMATE FUNCTION IS*,/)
      PRINT 190,G
  190 FORMAT(5X,*SCALAR FACTOR = *,F15.4,//)
      PRINT 200,(K,ROOTR(K),ROOTI(K),K=1,JJ)
  200 FORMAT(5X,I2,*  POLE = *,F10.4,*  J *,F10.4)
*
*
*      COMPUTE THE 2-S-2-D APPROXIMATE FUNCTION
*
*
      HA=G*G
      DO 210 K=1,JJ
      POLE1R(K)=ROOTR(K)
      POLE1I(K)=ROOTI(K)
      POLE2R(K)=POLE1R(K)
      POLE2I(K)=-POLE1I(K)
  210 CONTINUE
*
*
*      PRINT THE REQUIRED 2-S-2-D APPROXIMATE FUNCTION
*
*
      PRINT 220
  220 FORMAT(5X,*THE REQUIRED 2-S-2-D APPROXIMATE FUNCTION IS*,/)
      PRINT 230,HA
  230 FORMAT(5X,*SCALAR FACTOR = *,F15.4,//)
      PRINT 235
  235 FORMAT(22X,*POLE AT S1 PLANE*,12X,*POLE AT S2 PLANE*,/)
      PRINT 240,(K,POLE1R(K),POLE1I(K),POLE2R(K),POLE2I(K),K=1,JJ)
  240 FORMAT(5X,I2,* POLE PAIR*,F9.4,* J *,F9.4,9X,F9.4,* J *,F9.4)
*
*      COMPUTE THE PRODUCT OF TWO FIRST ORDER FUNCTION
*      T(I)(S1,S2)=1/B00(I)+B01(I)S2+B10(I)S1+B11(I)S1S2
*
*
      DO 250 I=1,JJ
      B00(I)=CMPLX(1.0,0.0)
      B01(I)=CMPLX((-1.0*POLE1R(I)),(-1.0*POLE1I(I)))
      B10(I)=CMPLX((-1.0*POLE2R(I)),(-1.0*POLE2I(I)))
      B11(I)=B01(I)*B10(I)
  250 CONTINUE
```

```
*
*
*     PRINT ITH SECTION FUNCTION
*
*
*
      DO 260 I=1,JJ
      PRINT 270,I
 270  FORMAT(5X,*THE DENOMINATOR POLYNOMIAL OF *,I2,* SECTION IS*,/)
      PRINT 280,B00(I),B01(I),B10(I),B11(I)
 280  FORMAT(5X,2E11.4,* + *,2E11.4,*S2+ *,2E11.4,*S1+ *,2E11.4,*S1S2*)
 260  CONTINUE
*
*     COMPUTE THE COEFFICIENTS OF  T(I)(Z1,Z2)
*     AFTER THE DOUBLE BILINEAR TRANSFORMATION TO ITH SECTION
*     OF ANALOG FUNCTION
*
      DO 300 I=1,JJ
      A(I)=B00(I)-B01(I)*2.0/T1-B10(I)*2.0/T2+B11(I)*4.0/(T1*T2)
      B(I)=B00(I)-B01(I)*2.0/T1+B10(I)*2.0/T2-B11(I)*4.0/(T1*T2)
      C(I)=B00(I)+B01(I)*2.0/T1-B10(I)*2.0/T2-B11(I)*4.0/(T1*T2)
      D(I)=B00(I)+B01(I)*2.0/T1+B10(I)*2.0/T2+B11(I)*4.0/(T1*T2)
 300  CONTINUE
*
*     PRINT ITH SECTION DIGITAL FUNCTION
*
      DO 310 I=1,JJ
      PRINT 320,I
 320  FORMAT(5X,*THE *,I2,* SECTION DIGITAL FUNCTION IS*,/)
      PRINT 330
 330  FORMAT(5X,*NUMERATOR POLYNOMIAL     1+Z1+Z2+Z1Z2 *)
      PRINT 340,A(I),B(I),C(I),D(I)
 340  FORMAT(5X,2E11.4,* + *,2E11.4,*Z1+ *,2E11.4,*Z2+ *,2E11.4,*Z1Z2*
 310  CONTINUE
*
*
*     COMPUTE  THE DIGITAL RESPONSE OF  THE FILTER
*
*
      DO 350 K1=1,41
      OMEGA1=(K1-21)/1.0
      Z1=CEXP((0.0,1.0)*OMEGA1*T1)
      DO 360 K2=1,41
      OMEGA2=(K2-21)/1.0
      Z2=CEXP((0.0,1.0)*OMEGA2*T2)
      HZ(K1,K2)=CMPLX(1.0,0.0)
      DO 370 I=1,JJ
      TZN(I)=1.0+Z1+Z2+Z1*Z2
      TZD(I)=A(I)+B(I)*Z1+C(I)*Z2+D(I)*Z1*Z2
      TZ(I)=TZN(I)/TZD(I)
```

```
      HZ(K1,K2)=HZ(K1,K2)*TZ(I)
370   CONTINUE
      HD(K1,K2)=(-20.0)*ALOG10((CABS(HZ(K1,K2))*HA))
360   CONTINUE
350   CONTINUE
      CALL NORMAL(HD,NPOINT,H1D)
      IF(IDIGAN.EQ.1) GO TO 101
*
*
*      PRINT THE DIGITAL RESPONSE OF THE FILTER
*
*
      IF(IMCPLOT.EQ.1) CALL PLOT(H1D,41,SCALE)
       IF(IMCPLOT.EQ.2) CALL CONTOUR(HD,41)
      IF(IMCPLOT.EQ.3) CALL THREED(HD,41)
     STOP
101   N=1+(M+1)/2
      DO 900 K1=1,81
      S1=CMPLX(0.0,1.0)*(41.0-K1)/20.0
      DS00=CMPLX(0.0,0.0)
      DS01=CMPLX(0.0,0.0)
      DO 910 I=1,N
      DS00=DS00+Y(I)*S1**(N-I)
      DS01=DS01+(-1.0)**(N-I)*Y(I)*S1**(N-I)
910   CONTINUE
      DO 920 K2=1,81
      S2=CMPLX(0.0,1.0)*(41.0-K2)/20.0
      DS10=CMPLX(0.0,0.0)
      DS11=CMPLX(0.0,0.0)
      DO 930 I=1,N
      DS10=DS10+Y(I)*S2**(N-I)
      DS11=DS11+(-1.0)**(N-I)*Y(I)*S2**(N-I)
930   CONTINUE
      HSK1K2=1.0/((DS00*DS01*DS10*DS11)
      H(K1,K2)=CABS(HSK1K2)
```

```
      SUBROUTINE ROOT(XCOF,M,ROOTR,ROOTI,IER)
*      SUBROUTINE ROOT IS COMPUTING THE COMPLEX ROOTS
*      OF A GIVEN POLYNOMIAL AS
*      A(1) +A(2)X + ...... + A(N)X**(N-1) + A(N+1)X**(N) =0
*      XCOF(I)   = COEFFICIENTS OF POLYNOMIAL
*      M  = DEGREE OF POLYNOMIAL
*      ROOTR(I) = REAL PART OF ROOT(I)
*      ROOTI(I) = IMAGINARY PART OF ROOT(I)
*
*
      DIMENSION XCOF(30),COF(30),ROOTR(15),ROOTI(15)
      DOUBLE PRECISION XO,YO,X,Y,XPR,YPR,UX,UY,U,YT,XT,U
      DOUBLE PRECISION XT2,YT2,SUMSQ,DX,DY,TEMP,ALPHA
      IFIT=0
      N=M
      IER=0
      IF(XCOF(N+1))10,25,10
   10 IF(N)15,15,32
   15 IER=1
   20 RETURN
   25 IER=4
      GO TO 20
   30 IER=2
      GO TO 20
   32 IF(N-14)35,35,30
   35 NX=N
      NXX=N+1
      N2=1
      KJ1=N+1
      DO 40 L=1,KJ1
      MT=KJ1-L+1
   40 COF(MT)=XCOF(L)
   45 XO=0.00500101
      YO=0.01000101
       IN=0
   50 X=XO
      XO=-10.0*YO
      YO=-10.0*X
      X=XO
      Y=YO
      IN=IN+1
      GO TO 59
   55 IFIT=1
      XPR=X
      YPR=Y
   59 ICT=0
   60 UX=0.0
      UY=0.0
      V=0.0
      YT=0.0
```

```
      XT=1.0
      U=COF(N+1)
      IF(U)65,130,65
65    DO 70 I=1,N
      L=N-I+1
      TEMP=COF(L)
      XT2=X*XT-Y*YT
      YT2=X*YT+Y*XT
      U=U+TEMP*XT2
      V=V+TEMP*YT2
      FI=I
      UX=UX+FI*XT*TEMP
      UY=UY-FI*YT*TEMP
      XT=XT2
70    YT=YT2
      SUMSQ=UX*UX+UY*UY
      IF(SUMSQ)75,110,75
75    DX=(V*UY-U*UX)/SUMSQ
      X=X+DX
      DY=-(U*UY+V*UX)/SUMSQ
      Y=Y+DY
78    IF(DABS(DY)+DABS(DX)-1.0D-05)100,80,80
80    ICT=ICT+1
      IF(ICT-500)60,85,85
85    IF(IFIT)100,90,100
90    IF(IN-5)50,95,95
95    IER=3
      GO TO 20
100   DO 105 L=1,NXX
      MT=NXX-L+1
      TEMP=XCOF(MT)
      XCOF(MT)=COF(L)
105   COF(L)=TEMP
      ITEMP=N
      N=NX
      NX=ITEMP
      IF(IFIT)120,55,120
110   IF(IFIT)115,50,115
115   X=XPR
      Y=YPR
120   IFIT=0
122   IF(DABS(Y/X)-1.0D-4)135,125,125
125   ALPHA=X+X
      SUMSQ=X*X+Y*Y
      N=N-2
      GO TO 140
130   X=0.0
      NX=NX-1
      NXX=NXX-1
135   Y=0.0
```

```
      SUMSQ=0.0
      ALPHA=X
      N=N-1
140   COF(2)=COF(2)+ALPHA*COF(1)
145   DO 150 L=2,N
150   COF(L+1)=COF(L+1)+ALPHA*COF(L)-SUMSQ*COF(L-1)
155   ROOTI(N2)=Y
      ROOTR(N2)=X
      N2=N2+1
      IF(SUMSQ)160,165,160
160   Y=-Y
      SUMSQ=0.0
      GO TO 155
165   IF(N) 20,20,45
      END
```

```
      SUBROUTINE CONTOUR(Z,NPOINT)
      DIMENSION Z(101,101)
      CALL INITT(120,5,6,0,0)
      NXI=NPOINT
      NYI=NPOINT
      XA=-2.0
      YA=-2.0
      XB=2.0
      YB=2.0
      Z00=0.0
      DZ=0.1
      ZMIN=0.0
      ZMAX=1.0
      NS=1
      SIZE=0.15
      XAX=4
      YAX=4
      CALL CONTPLT(Z,NXI,NYI,XA,XB,YA,YB,Z00,DZ,ZMIN,ZMAX,NS)

      CALL FINITT(0,760)
      RETURN
      END
```

```
      SUBROUTINE CONTPLT(Z,NXI,NYI,XA,XB,YA,YB,Z00,DZ,ZMIN,ZMAX,NS)
*
*
*
*       SUBROUTINE CONTPLT IS A COMPUTER PROGRAM TO PLOT
*       NON-SMOTHING CONTOUR LINES FOR ANY TWO-DIMENSIONAL
*       XA AND XB ARE THE END POINTS OF THE X-AXIS, XA.LT.XB
*       NXI=NUMBER OF X-POINTS OF THE GRID Z(X,Y)
*       YA AND YB ARE THE END POINTS OF THE Y-AXIS, YA.LT.YB
*       NYI=NUMBER OF Y-POINTS IN THE GRID Z(X,Y)
*       Z00=ONE OF THE CONTOUR LINES TO BE PLOT
*       DZ=CONTOUR SPACING
*       ONLY CONTOURS Z=CONST WITH ZMIN.LE.Z.LE.ZMAX WILL BE PLOTTED
*       THE ROUTINE WILL PLOT CONTOUR LINE FOR THE LEVELS Z00
*       Z00+(-)DZ,Z00+(-)2DZ,....  UNLESSTHE LIMITS
*       ZMIN AND ZMAX OR THE MINIMIUM VALUE OF THE ARRAY Z IS EXCEEDED.
*
*
*
      DIMENSION Z(101,101),X(400),Y(400),IZ(400),KZ(400)
      DIMENSION NEXT(400,6),XLL(5),YLL(5),I4(4)
      X(1)=XA
      X(2)=XB
      XMIN=X(1)
      Y(1)=YA
      Y(2)=YB

      YMIN=Y(1)
      DX=(X(2)-XMIN)/FLOAT(NXI-1)
      DY=(Y(2)-YMIN)/FLOAT(NYI-1)
      Z0=Z00
      DZ0=DZ
      LZ0=-1
*      NEW LEVEL Z0
   55 Z0=Z0+DZ0
   56 NZ0=1
      IF(Z0.GT.ZMAX.OR.Z0.LT.ZMIN) GO TO93
*      SEARCH FOR Z(I,K).LT.Z0.LT.Z(I,K+1) OR VICE VERSA
      DO 80 I=1,NXI
      DO 80 K=1,NYI
      IF(K.LT.NYI) GO TO 59
      IF(Z0-Z(I,K))60,61,60
   59 IF((Z0-Z(I,K))*(Z0-Z(I,K+1)).GT.0..OR.Z0.EQ.Z(I,K+1))GO TO 60
      Y(NZ0)=YMIN+(FLOAT(K-1)+(Z(I,K)-Z0)/(Z(I,K)-Z(I,K+1)))*DY
      KZ(NZ0)=K-1
*      (XNZ0,YNZ0)=POINT WITH Z(XNZ0,YNZ0)=Z0:
      X(NZ0)=XMIN+FLOAT(I-1)*DX
      IZ(NZ0)=I-1
      NZ0=NZ0+1
```

```
      IF(NZO.GT.400) GO TO 82
*      SEARCH FOR Z(I,K).LT.ZO.LT.Z(I+1,K) OR VICE VERSA
 60   IF(I.LT.NXI)GO TO 70
      IF(ZO-Z(I,K))80,61,80
 61   X(NZO)=XMIN+FLOAT(I-1)*DX
      GO TO 77
 70   IF((ZO-Z(I,K))*(ZO-Z(I+1,K)).GT.0..OR.ZO.EQ.Z(I+1,K))GO TO 80
      X(NZO)=XMIN+(FLOAT(I-1)+(Z(I,K)-ZO)/(Z(I,K)-Z(I+1,K)))*DX
 77   IZ(NZO)=I-1
      Y(NZO)=YMIN+FLOAT(K-1)*DY
      KZ(NZO)=K-1
      NZO=NZO+1
      IF(NZO.GT.400) GO TO 82
 80   CONTINUE
      GO TO 90
 82   PRINT 85,ZO
 85   FORMAT(21H CONTOUR TOO LONG, Z= ,1PE15.7)
 90   NZO=NZO-1
      IF(NZO.NE.0) GO TO 100
*      NO CONTOUR ZO
 93   IF(DZO)95,96,96
 95   CONTINUE
      RETURN
*      LEVELS ZO.LE.ZOO
*      LEVELS ZO.LE.ZOO
 96   DZO=-DZO
      ZO=ZOO
      NS=-IABS(NS)
      GO TO 56
 100   NS=-NS
      INUM=1
*      PRELIMINARY ORDER OF CONTOUR POINT (SMALLEST DISTANCES);
      DO 160 I=1,NZO
      IF(I.GT.1) GO TO 130
```

```
 140   YX=X(I)
       X(I)=X(L)
       X(L)=YX
       YX=Y(I)
       Y(I)=Y(L)
       Y(L)=YX
       J=IZ(I)
       IZ(I)=IZ(L)
       IZ(L)=J
       J=KZ(I)
       KZ(I)=KZ(L)
 160   KZ(L)=J
*       POSSIBLE CONNECTIONS BETWEEN POINTS OF ONE COUNTOUR LINE:
       NZM1=NZO-1
       DO 220 I=1,NZM1
       DO 210 L=1,6
 210   NEXT(I,L)=0
       J=1
       IP1=I+1
       DO 220 K=IP1,NZO
      IF(IZ(I).NE.IZ(K).AND.(IABS(IZ(I)-IZ(K)).NE.1.OR.AMAX1(X(I),X(K)).
     *NE.XMIN+AMAX0(IZ(I),IZ(K))*DX)) GO TO 220
      IF(KZ(I).NE.KZ(K).AND.(IABS(KZ(I)-KZ(K)).NE.1.OR.AMAX1(Y(I),Y(K)).
     *NE.YMIN+AMAX0(KZ(I),KZ(K))*DY))GO TO220
       NEXT(I,J)=K
       J=J+1
 220   CONTINUE
       DO 360 I=1,NZM1
       DO 360 L=1,6
       IF(NEXT(I,L).EQ.0) GO TO 360
*      RODING OUT OF USELESS INTERSECTION
       IP1=I+1
       NEXTIL=NEXT(I,L)
       DO 300 K=IP1,NZO
       DO 300 J=1,6
*       FOR IDENTICAL POINTS:
       IF(I.EQ.NEXT(K,J).OR.K.EQ.NEXT(I,L).OR.NEXT(I,L).EQ.NEXT(K,J).OR.
     *NEXT(K,J).EQ.0) GO TO 300
*       FOR POINTS BELONGING TO DIFFERENT GRID CELLS:
       I4(1)=I
       I4(2)=NEXT(I,L)
       I4(3)=K
       I4(4)=NEXT(K,J)
       DO 250 M=1,3
       I4M=I4(M)
       MP1=M+1
       DO 250 N=MP1,4
       I4N=I4(N)
       IF(IZ(I4M).NE.IZ(I4N).AND.(IABS(IZ(I4M)-IZ(I4N)).NE.1.OR.AMAX1(X(
     *I4M),X(I4N)).NE.XMIN+AMAX0(IZ(I4M)-1,IZ(I4N)-1)*DX)) GO TO 300
```

```
      IF(KZ(I4M).NE.KZ(I4N).AND.(IABS(KZ(I4M)-KZ(I4N)).NE.1.OR.AMAX1(Y
     *I4M),Y(I4N)).NE.YMIN+AMAX0(KZ(I4M)-1,KZ(I4N)-1)*DY)) GO TO 300
  250 CONTINUE
      M=3
      N=4
      NEXTKJ=NEXT(K,J)
      IF(X(NEXTIL).EQ.X(I)) GO TO 270
      IF(X(NEXTKJ).EQ.X(K)) GO TO 290
      YX21=(Y(NEXTIL)-Y(I))/(X(NEXTIL)-X(I))
      YX43=(Y(NEXTKJ)-Y(K))/(X(NEXTKJ)-X(K))
      YX=YX21-YX43
*      PARALLEL LINES:
      IF(YX.EQ.0) GO TO 300
*      COORDINATES OF THE INTERSECTION:
      XS=(Y(K)-Y(I)+X(I)*YX21-X(K)*YX43)/YX
      YS=Y(I)+YX21*(XS-X(I))
*      INTERSECTION INSIDE THE GRID CELL
  265 IF(AMIN1(X(I),X(NEXTIL),X(K),X(NEXTKJ)).GE.XS.OR.XS.GE.AMAX1(X(I)
     *X(NEXTIL),X(K),X(NEXTKJ)).OR.AMIN1(Y(I),Y(NEXTIL),Y(K),Y(NEXTKJ))
     *GE.YS.OR.YS.GE.AMAX1(Y(I),Y(NEXTIL),Y(K),Y(NEXTKJ))) GO TO 300
      NEXT(K,J)=0
      GO TO 360
  270 XS=X(I)
      YS=Y(K)+(Y(NEXTKJ)-Y(K))/(X(NEXTKJ)-X(K))*(XS-X(K))
      GO TO 265
  290 XS=X(K)
      YS=Y(I)+(Y(NEXTIL)-Y(I))/(X(NEXTIL)-X(I))*(XS-X(I))
      GO TO 265
  300 CONTINUE
      K=NZO
      J=6
*      PLOT OF ONE CONTOUR LINE
      XLL(2)=X(I)
      XLL(1)=X(NEXTIL)
      YLL(2)=Y(I)
      YLL(1)=Y(NEXTIL)
      DO 310 N=3,5
      DO 310 M=1,2
      IF(XLL(M).EQ.XLL(N).AND.YLL(M).EQ.YLL(N)) GO TO 320
  310 CONTINUE
      N=5
      M=2
      IF(IABS(NS).EQ.3)INUM=1
*      (XS,YS) IS THE CURRENT PEN POSITION
  320 CONTINUE
      IF(XLL(M).NE.XS.OR.YLL(M).NE.YS) CALL PLOT(XLL(M),YLL(M),3)
*      PLOT(X,Y,N) MOVES THE PEN TO (X,Y)
*      PEN UP IF N=3,  PEN DOWN IF N=2
      M3M=3-M
      CALL PLOT(XLL(M3M),YLL(M3M),2)
```

```
      DO 330 NM=1,3
      N=4-NM
      NP2=N+2
      XLL(NP2)=XLL(N)
330   YLL(NP2)=YLL(N)
      NSP4=NS+4
      GO TO (350,360,340,360,340,340,350),NSP4
340   IF(NEXT(I,L).LT.NZO/2.AND.I.LT.NZO/2) GO TO 360
350   IF(INUM.EQ.2) GO TO  360
*      CURRENT PEN POSITION
      INUM=2
      CALL PLOT(XLL(3),YLL(3),3)
360   CONTINUE
*      NEXT CONTOUR LINE
      GO TO 55
      END
```

```
      SUBROUTINE PLOT(X,Y,J)
*      MOVE THE PEN TO THE POINT (X,Y)
*      WHERE
*         X--IS THE NUMBER OF PLOTTER INCREAMENTS
*            ALONG THE LENGTH OF THE PAPER FROM THE ORIGIN
*         Y--IS THE NUMBER OF PLOTER INCREAMENTS
*            ACROSS THE WIDTH OF THE PAPER FROM THE ORIGIN
*         J--IS THE PEN STATUS
*            2  LOWER PEN BEFORE MOVING
*            3  RAISE PEN BEFORE MOVING
*            IF J IS NEGATIVE THE ORIGIN WILL RESET AT (X,Y)
      IREFY=Y*190+398
      IREFX=X*190+512
      IF(J.LT.0) CALL MOVABS(IREFX,IREFY)
      IF(J.EQ.2) CALL DRWABS(IREFX,IREFY)
      IF(J.EQ.3) CALL MOVABS(IREFX,IREFY)
      RETURN
      END
```

```
      SUBROUTINE NORMAL(H,NPOINT)
      DIMENSION H(101,101),H1D(301)
      HMAX=H(1,1)
      DO 10 I=1,NPOINT
      DO 10 J=1,NPOINT
      IF(H(I,J).GT.HMAX) HMAX=H(I,J)
   10 CONTINUE
      N=(NPLOT+1)/2
      DO 20 I=1,NPOINT
      DO 20 J=1,NPOINT
      IF(I.EQ.N) H1D(J)=H(I,J)
      H(I,J)=H(I,J)/HMAX
   20 CONTINUE
      RETURN
       END
```

```
      SUBROUTINE THREED(H,LINEPON)
      DIMENSION MASK(2000),VERTEX(16),YDATA(41),ZDATA(41)
      DIMENSION XDATA(41),H(41,41)
      CALL INITT(120,5,6,0,0)
      ICROSS=0
      IVXYZ=10
      XSCALE=-0.1
      YSCALE=1.0
      ZSCALE=0.1
      NPNTS=LINEPON
      PHI=-45.0
      THETA=35.0
      XREF=5.0
      YREF=3.0
      XLENTH=10.0
      DO 11 J=1,16
      VERTEX(J)=0.0
11    CONTINUE
      DO 20 NLINE=1,LINEPON
      XDATA(NLINE)=0.0
      DO 10 NPOINT=1,LINEPON
      ZDATA(NPOINT)=0.0
      YDATA(NPOINT)=H(NLINE,NPOINT)
10    CONTINUE
      CALL PLOT3D(IVXYZ,XDATA,YDATA,ZDATA,XSCALE,YSCALE,ZSCALE,
     *NLINE,NPNTS,PHI,THETA,XREF,YREF,XLENTH,MASK,VERTEX,ICROSS)
20    CONTINUE
      ICROSS=1
      IVXYZ=1010
      XSCALE=-0.1
      ZSCALE=0.1
      DO 30 NLINE=1,LINEPON
      ZDATA(NLINE)=0.0
30    CONTINUE
      DO 50 NLINE=1,LINEPON
      XDATA(NLINE)=0.0

      DO 40 NPOINT=1,LINEPON
      YDATA(NPOINT)=H(NLINE,NPOINT)
40    CONTINUE
      CALL PLOT3D(IVXYZ,XDATA,YDATA,ZDATA,XSCALE,YSCALE,ZSCALE,
     *NLINE,NPNTS,PHI,THETA,XREF,YREF,XLENTH,MASK,VERTEX,ICROSS)
50    CONTINUE
      CALL FINITT(0,760)
      RETURN
      END
```

```
      SUBROUTINE PLOT3D(IVXYZ,XDATA,YDATA,ZDATA,XSCALE,YSCALE,ZSCALE,
     *NLINE,NPNTS,PHI,THETA,XREF,YREF,XLENTH,MASK,VERTEX,ICROSS)
*
*
*     SUBROUTINE PLOT3D IS A MASKED 3-DIMENSIONAL PLOT
*     PROGRAM WITH ROTATIONS
*     THIS ROUTINE WILL ACCEPT 3-DIMENSIONAL DATA IN VARIOUS FORMS
*     AS INPUT, ROTATE IT IN 3-SPACE TO ANY ANGLE AND
*     PLOT THE PROJECTION OF THE RESULTING FIGURE UNTO THE
*     XY PLANE.  LINEAR INTERPOLATION IS USED BETWEEN DATA
*     POINTS. THOSE LINES OF A FIGURE WHICH SHOULD BE HIDDEN BY
*     A PREVIOUS LINE ARE MASKED.
*
*     EACH CALL TO PLOT3D CAUSES ONE LINE OF A FIGURE TO BE
*     PLOTED.
*     TWO PARAMETERS OF THE PLOTTER ARE SET ON THE INITIAL CALL
*     FOR EACH FIGURE.
*     (PIPI) IS THE NUMBER OF PLOTTER INCREMENTS PER INCH.
*     (NYPI) IS THE NUMBER OF INCREMENTS AVAILABLE ACROSS THE
*     WIDTH OF THE PAPER ( Y-DIRECTION).
*     ICROSS=0 NO CROSS HATCH.
*     ICROSS=1 CROSSHATCH TO BE PLOTED.
*     (PHI) AND (THETA) ARE TWO ANGLES (IN DEGREES) USED TO
*     SPECIFY THE DESIRED 3-DIMENSIONAL ROTATION.
*     FOR THE INITIAL SYSTEM OF AXES XYZ
*     ANGLE (PHI) IS ROTATED COUNTERCLOCKWISE ABOUT Y-AXIS,
*     ANGLE (THETA) IS ROTATED COUNTERCLOCKWISE ABOUT X-AXIS.
*     (XREF) AND (YREF) ARE THE COORDINATES IN INCH
*     RELATIVES TO THE PLOTTER ORIGIN.
*     (XLENGTH) IS THE LENGTH, IN INCH, TO WHICH THE PLOT IS
*     RESTRICTED.
*     (MASK) IS AN INTEGER ARRAY OF 2*XLENGTH*PIPI ENTRIES WHICH
*     IS USED TO STORE THE MASK.
*     THE CONTENTS OF THIS ARRAY SHOULD NOT BE ALTERED AT ANY TIME
*
      INTEGER HIGH,OLDHI,OLDLOW
      DIMENSION XDATA(41),YDATA(41),ZDATA(41),MASK(2000),VERTEX(16)
      DATA INIT,JVXYZ,SPHI,STHETA/-1,-1,-1.0E99,-1.0E99/
      IF(NLINE.EQ.0) GO TO 550
      IF(NLINE.NE.1) GO TO 20
```

```
      I=0
20    IF(JVXYZ.EQ.IVXYZ) GO TO 70
      JVXYZ=IVXYZ
      INDZ=1
      INDY=1
      INDX=1
      INDV=1
      IF(JVXYZ.LT.1000) GO TO 30
      INDV=2
      JVXYZ=JVXYZ-1000
30    IF(JVXYZ.LT.100) GO TO 40
      INDX=2
      JVXYZ=JVXYZ-100
40    IF(JVXYZ.LT.10) GO TO 50
      INDY=2
      JVXYZ=JVXYZ-10
50    IF(JVXYZ.LT.1) GO TO 60
      INDZ=2
60    JVXYZ=IVXYZ
70    IF(PHI.EQ.SPHI.AND.THETA.EQ.STHETA) GO TO 80
      SPHI=SIN(0.0174532925*PHI)
      CPHI=COS(0.0174532925*PHI)
      STHETA=SIN(0.0174532925*THETA)
      CTHETA=COS(0.0174532925*THETA)
      A11=CPHI
      A13=-SPHI
      A21=STHETA*SPHI
      A22=CTHETA
      A23=STHETA*CPHI
      SPHI=PHI
      STHETA=THETA
80    INCI=-INCI
      IF(I.NE.0)I=NPNTS+1
      DO 530 K=1,NPNTS
      I=I+INCI
      GO TO (90,100),INDX
90    IF(ICROSS.EQ.0) X=XDATA(1)+(I-1)*XSCALE
      IF(ICROSS.EQ.1) X=XDATA(1)+(NLINE-1)*XSCALE
      GO TO 110
100   X=XDATA(I)*XSCALE
110   GO TO(120,130),INDY
120   Y=YDATA(1)+(I-1)*YSCALE
      GO TO 140
130   Y=YDATA(I)*YSCALE
140   GO TO (150,160),INDZ
150   IF(ICROSS.EQ.0) Z=ZDATA(1)+(NLINE-1)*ZSCALE
      IF(ICROSS.EQ.1) Z=ZDATA(1)+(I-1)*ZSCALE
      GO TO 170
160   Z=ZDATA(I)*ZSCALE
170   XXX=A11*X+A13*Z+XREF
```

```
      XX=XXX
      IX=IROUND(XX*PIPI)
      YYY=A21*X+A23*Z+YREF
      YY=YYY+A22*Y
      IY=IROUND(YY*PIPI)
      IF(IX.LE.0) IX=1
      IF(IX.GT.LIMITX) IX=LIMITX
      IF(IY.LT.10) IY=10
      IF(IY.GT.NYPI) IY=NYPI
      IF(K.NE.1) GO TO 250
      LOW=IX+IX
      HIGH=LOW-1
      MLOW=MASK(LOW)
      MHIGH=MASK(HIGH)
      IF(MHIGH-IY) 200,210,180
180   IF(MLOW-IY) 190,230,220
190   LOCOLD=0
      GO TO 240
200   MASK(HIGH)=IY
      IF(MLOW.EQ.-1) MASK(LOW)=IY
210   LOCOLD=+1
      GO TO 240
220   MASK(LOW)=IY
230   LOCOLD=-1
240   CALL IPLOT(ICROSS,IX,IY,3)
      JX=IX
      JY=IY
      IYREF=IY
      IF(INDV.EQ.1) GO TO 530
      INDEX=INCI+6
      VERTEX(INDEX)=XX
      VERTEX(INDEX+1)=YY
      VERTEX(INDEX+8)=XXX
      VERTEX(INDEX+9)=YYY
      IF(NLINE.NE.1) GO TO 530
      VERTEX(1)=XX
      VERTEX(2)=YY
      VERTEX(9)=XXX
      VERTEX(10)=YYY
      GO TO 530
250   IF(IX.NE.JX) GO TO 260
      JY=IY
      GO TO 280
260   YINC=FLOAT(IY-JY)/ABS(FLOAT(IX-JX))
      INCX=(IX-JX)/IABS(IX-JX)
      YJ=JY
270   JX=JX+INCX
      YJ=YJ+YINC
      JY=IROUND(YJ)
      LOW=JX+JX
```

```
      HIGH=LOW-1
      MLOW=MASK(LOW)
      MHIGH=MASK(HIGH)
280   IF(MHIGH-JY)300,300,290
290   IF(MLOW-JY)310,320,320
300   LOC=+1
      IF(LOCOLD)360,370,430
310   LOC=0
      IF(LOCOLD)340,350,330
320   LOC=-1
      IF(LOCOLD)510,450,440
330   IF(MHIGH.LE.IYREF)CALL IPLOT(ICROSS,JX,MHIGH,2)
      GO TO 350
340   IF(MLOW.GE.IYREF) CALL IPLOT(ICROSS,JX,MLOW,2)
350   CALL IPLOT(ICROSS,JX,JY,3)
      GO TO 520
360   IF(MLOW-IYREF)370,380,380
370   IF(MHIGH-IYREF)400,390,390
380   CALL IPLOT(ICROSS,JX,MLOW,2)
390   CALL IPLOT(ICROSS,JX,MHIGH,3)
      GO TO 430
400   IF(MHIGH.EQ.-1) GO TO 430
      OLDHI=HIGH-2*INCX
      IF(MASK(OLDHI)-JY)420,420,410
410   CALL IPLOT(ICROSS,JX,JY,3)
      GO TO 430
420   CALL IPLOT(ICROSS,JX-INCX,MASK(OLDHI),3)
430   MASK(HIGH)=JY
      IF(MLOW.EQ.-1) MASK(LOW)=JY
      CALL IPLOT(ICROSS,JX,JY,2)
      GO TO 520
440   IF(MHIGH-IYREF)460,460,450
450   IF(MLOW-IYREF)470,470,480
460   CALL IPLOT(ICROSS,JX,MHIGH,2)
470   CALL IPLOT(ICROSS,JX,MLOW,3)
      GO TO 510
480   OLDLOW=LOW-2*INCX
      IF(MASK(OLDLOW)-JY)490,500,500
490   CALL IPLOT(ICROSS,JX,JY,3)
      GO TO 510
500   CALL IPLOT(ICROSS,JX-INCX,MASK(OLDLOW),3)
510   MASK(LOW)=JY
      CALL IPLOT(ICROSS,JX,JY,2)
520   IYREF=JY
      LOCOLD=LOC
      IF(JX.NE.IX) GO TO 270
530   CONTINUE
      CALL IPLOT(ICROSS,JX,JY,3)
      IF(INDV.EQ.1) GO TO 540
      INDEX=-INCI+6
```

```
      VERTEX(INDEX)=XX
      VERTEX(INDEX+1)=YY
      VERTEX(INDEX+8)=XXX
      VERTEX(INDEX+9)=YYY
      IF(NLINE.NE.1) GO TO 540
      VERTEX(3)=XX
      VERTEX(4)=YY
      VERTEX(11)=XXX
      VERTEX(12)=YYY
540   I=I-1
      RETURN
550    INIT=0
      RETURN
      END
```

```
      FUNCTION IROUND(B)
      IROUND=B+0.5
      RETURN
      END
```

```
      SUBROUTINE IPLOT(ICROSS,IX,IY,J)
*      MOVE THE PEN TO THE POINT (IX,IY)
*      WHERE
*         IX--IS THE NUMBER OF PLOTTER INCREAMENTS
*             ALONG THE LENGTH OF THE PAPER FROM THE ORIGIN
*         IY--IS THE NUMBER OF PLOTER INCREAMENTS
*             ACROSS THE WIDTH OF THE PAPER FROM THE ORIGIN
*         J--IS THE PEN STATUS
*            2  LOWER PEN BEFORE MOVING
*            3  RAISE PEN BEFORE MOVING
*            IF J IS NEGATIVE THE ORIGIN WILL RESET AT (IX,IY)
      ITEMPX=IX
      ITEMPY=IY
      IF(J.LT.0) CALL MOVABS(IX,IY)
      IF(J.EQ.2) CALL DRWABS(IX,IY)
      IF(J.EQ.3) CALL MOVABS(IX,IY)
      IX=ITEMPX
       IY=ITEMPY
      RETURN
      END
```

```
      SUBROUTINE PLOT(H,NPOINT,SCALE)
      DIMENSION X(4),Y(301),H(301)
      CALL INITT(120,5,6,0,0)
      CALL BINITT
*
*
*      DATA IN XARRAY DEFINE AS FOLLOW
*      X(1)=-1       .....LINEAR
*      X(2)=NPOINT        .....NUMBER OF DATA POINT TO BE PLOT
*      X(3)=FIRST ELEMENT VALUE IN X AXIS OF FIRST DATA IN X
*      X(4)= INCREAMENT FOR EACH DATA POINT IN X AXIS
*
*
      X(1)=-1
      X(2)=NPOINT
      X(3)=0
      X(4)=SCALE
*
*      DATA IN YARRAY DEFINE AS FOLLOW
*      Y(1)=NPOINT        NUMBER OF DATA  TO BE PLOT
*      Y(I)=H(I-1)     (I-1) DATA VALUE IN Y AXIS
*
*
      Y(1)=NPOINT
      N=NPOINT+1
      DO 1 I=1,N
      Y(I+1)=H(I)
    1 CONTINUE
      CALL XFRM(4)
      CALL YFRM(4)
      CALL CHECK(X,Y)
      CALL DSPLAY(X,Y)
      CALL FINITT(0,700)
      RETURN
      END
```