A New z - Domain Continued Fraction Expansion and its Use in the Generation of Stable Transfer Functions

Ling Luo

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montréal, Québec, Canada

June 2003

Your file Votre référence
ISBN: 0-612-83871-4
Our file Notre référence
ISBN: 0-612-83871-4

ABSTRACT

A New z - Domain Continued Fraction Expansion and its Use in the Generation of Stable Transfer Functions

Ling Luo

A new z - domain continued fraction expansion (CFE) has been proposed. It is shown that the ratio of an anti-mirror - image polynomial to a mirror - image polynomial originating from the denominator of a transfer function can be expanded into this new type of z - domain CFE . Unlike the earlier types of z - domain continued fraction expansions, this new type is not unique for a given polynomial, and there will be a large number of possibilities. Algorithms have been proposed to obtain the various possible $CFEs$, which are implemented using MATLAB software. For the denominators of some well - known filters like Butterworth lowpass filters, Butterworth lowpass Complementary Pole-pair Filters (CPPF) and Chebyshev lowpass filters, the required coefficients of the $CFEs$ are obtained. Also, starting from the coefficients of the CFEs, algorithms have been presented to generate stable transfer functions for filters such as lowpass, highpass and bandpass in the z - domain.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis supervisor, Dr. Venkat Ramachandran, for his valuable guidance throughout the course of this thesis, for his extremely careful, critical and thorough review of my work. He introduced me to this exciting area of filters and was a constant source of inspiration. I also would like to thank the examiners Dr. C.S. Gargour and Dr. Weiping Zhu for their helpful comments in the final preparation of the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

1.1 General

The design of any filter requires that the resulting transfer function be stable. In $1-D$ discrete domain, it is required that the denominator polynomial shall contain its roots within the unit circle. For a $1-D$ IIR filter (Infinite-duration Impulse Response), a transfer function is given by

$$H(z) = \frac{N(z)}{D(z)} = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + ... + p_M z^{-M}}{d_0 + d_1 z^{-1} + d_2 z^{-2} + ... + d_N z^{-N}} \tag{1.1}$$

which can be put in factored forms as:

$$H(z) = \frac{p_0 \prod_{k=1}^{M}(1 - \xi_k z^{-1})}{d_0 \prod_{k=1}^{N}(1 - \lambda_k z^{-1})} \tag{1.2}$$

$$= \frac{p_0}{d_0} z^{(N-M)} \frac{\prod_{k=1}^{M}(z - \xi_k)}{\prod_{k=1}^{N}(z - \lambda_k)} \tag{1.3}$$

where $\xi_1, \xi_2..., \xi_M$ are the finite zeros of $H(z)$, and $\lambda_1, \lambda_2...\lambda_N$ are the finite poles of $H(z)$. For a causal IIR filter, the ROC (Region of Convergence, the set of values of z for which the z - transform converges) of the transfer function $H(z)$ is thus exterior to the unit circle going through the pole furthest from the origin, the ROC is given by

$$|z| > max_k |\lambda_k|$$

It implies that all poles of a stable and causal transfer function $H(z)$ must be strictly inside the unit circle. It means the zeros of $D(z)$ shall be contained within the unit circle.

If the transfer function is available in the factored forms as shown in [1], it is relatively easy to determine, whether it is stable or not.

However, when a digital filter is implemented, it is highly preferable that alternative conditions are desired so that one doesn't have to obtain the transfer function and then determine its stability. Also, when the denominator of the transfer function is not given in the factored forms, tests can be carried out, like Jury-Marden Stability Criterion [1], in order to determine the stability of the filter.

1.2 Mirror-image and Anti-mirror-image Polynomials and Some Important Properties

The authors of [2], [3] and[4] have established several useful properties of a stable polynomial $D(z)$ having all its roots inside the unit circle in the z-plane. Some of those properties are discussed below[2]. Let,

$$D(z) = a_n z^n + a_{n-1} z^{n-1} + ... + a_1 z + a_0 = \sum_{i=0}^{n} a_i z^i \tag{1.4}$$

with a_n is positive. $D(z)$ can be decomposed as the sum of mirror-image polynomial

$$F_1(z) = \frac{[D(z) + z^n D(z^{-1})]}{2} \tag{1.5}$$

and the anti-mirror-image polynomial

$$F_2(z) = \frac{[D(z) - z^n D(z^{-1})]}{2} \tag{1.6}$$

For $D(z)$ which has its zeros inside the unit circle, the necessary and sufficient conditions are [2],

(a) the zeros of $F_1(z)$ and $F_2(z)$ are located on the unit circle,

(b) they are simple,

(c) they separate each other and

(d) $|\frac{a_0}{a_n}| < 1$, [4]

In this paper, we affix the subscript "e" for all polynomials and functions when the order n is even, and the subscript 'o' when n is odd. Thus, $F_1(z)$ and $F_2(z)$ should be represented by $F_{1e}(z)$ and $F_{2e}(z)$ respectively when n is even; and by $F_{1o}(z)$ and $F_{2o}(o)$ respectively when n is odd.

It is also necessary to ensure that [1]

(a) $D(1) > 0$, and

(b) $(-1)^n D(-1) > 0$

these are necessary conditions for the zeros of $D(z)$ to be contained within the unit circle.

Some properties of the mirror-image and the anti-mirror-image polynomials are discussed below [5].

Result 1 If $D(z)$ contains a factor $d_1(z) = z^2 + a_1 z + 1$, $0 \leq |a_1| < 2$, then both $F_1(z)$ and $F_2(z)$ also contain $d_1(z)$.

Result 2 $(z + 1)$ is a factor of both $F_{2e}(z)$ and $F_{1o}(z)$.

Result 3 $(z - 1)$ is a factor of both $F_{2e}(z)$ and $F_{2o}(z)$.

From Schussler's theorem and Results 1, 2 and 3, it is concluded:

(a) All the zeros of F_{1e} are complex conjugate pairs and lie on the unit circle.

(b) A zero of F_{1o} is -1, other zeros are complex conjugate pairs and lie on the unit circle.

(c) ± 1 are zeros of F_{2e}, other zeros are complex conjugate pairs and lie on the unit circle.

(d) A zero of F_{2o} is 1, other zeros are complex conjugate pairs and lie on the unit circle.

Therefore, these polynomials can be expressed as follows.

Table 1.1: Expansions of $F_1(z)$ and $F_2(z)$ [5]

	F_1 and F_2
n even	$F_{1e} = k_{1e} \prod_{i=1}^{n/2} (z^2 + a_i z + 1), 0 \leq \|a_i\| < 2$
	$F_{2e} = (z^2 - 1) k_{2e} \prod_{i=1}^{(n-2)/2} (z^2 + \beta_i z + 1), 0 \leq \|\beta_i\| < 2$
n odd	$F_{1o} = (z+1) k_{1o} \prod_{i=1}^{(n-1)/2} (z^2 + \gamma_i z + 1), 0 \leq \|\gamma_i\| < 2$
	$F_{2o} = (z-1) k_{2o} \prod_{i=1}^{(n-1)/2} (z^2 + \delta_i z + 1), 0 \leq \|\delta_i\| < 2$

where k_{1e}, k_{2e}, k_{2o} and k_{2o} are real and positive constants.

1.3 Known Continued Fraction Expansions [3][6]

Based on the above properties, certain continued fraction expansions exist, which are discussed below:

The types of $CFEs$ are:

1) CFE1:

When n is even or odd,

$$\frac{F_1(z)}{F_2(z)} = ka(1)\frac{(z+1)}{(z-1)} + \frac{1}{ka(2)\frac{(z+1)}{(z-1)} + \frac{1}{ka(3)\frac{(z+1)}{(z-1)} + \frac{1}{\cdots\cdots}}} \quad (1.7)$$

where $ka(1), ka(2)...ka(n)$ are real and positive.

2) $CFE2$:

(a) $CFE\ 2a$:

For n is even,

$$\frac{F_{1e}(z)}{F_{2e}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\cdots\cdots}}} \quad (1.8)$$

where $kb(1), kb(2)...kb(n)$ are real and positive.

(b) $CFE\ 2b$:

or if n is odd,

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\cdots\cdots}}} \quad (1.9)$$

where $kb(1), kb(2)...kb(n)$ are real and positive.

3) $CFE\ 1a$

In addition to the above set of continued fraction expansions, there exist other forms, one of them (we call this as $CFE\ 1a$) is obtained by reformulating the mirror - image and anti - mirror - image polynomials, and they are briefly discussed below [7]:

If the degree is even,

$$\mathrm{F}_1(z) = \frac{zD(z) + z^n D(z^{-1})}{z+1} \quad (1.10)$$

and

$$\mathrm{F}_2(z) = \frac{D(z) - z^n D(z^{-1})}{z+1} \tag{1.11}$$

If the degree is odd,

$$\mathrm{F}_1(z) = \frac{D(z) + z^n D(z^{-1})}{z+1} \tag{1.12}$$

and

$$\mathrm{F}_2(z) = \frac{zD(z) - z^n D(z^{-1})}{z+1} \tag{1.13}$$

The CFE 1a is obtained as

$n =$ even

$$\frac{F_1(z)}{F_2(2)} = r_1(z-1) + \cfrac{1}{r_2(1-z^{-1}) + \cfrac{1}{r_3(z-1) + \cfrac{1}{r_4(1-z^{-1}) + \cfrac{1}{\cdots}}}} \tag{1.14}$$

$n =$ odd

$$\frac{F_2(z)}{F_1(2)} = r_1(z-1) + \cfrac{1}{r_2(1-z^{-1}) + \cfrac{1}{r_3(z-1) + \cfrac{1}{r_4(1-z^{-1}) + \cfrac{1}{\cdots}}}} \tag{1.15}$$

where the coefficients $r's$ are positive. It must be noted that the positivity of $r's$ will not be sufficient to prove that $D(z)$ contains all its zeros within the unit circle[8] and [9]. In [8], there are alternative forms of $CFE's$ based on the decompositions given in Eqs (1.10) , (1.11) and Eqs (1.12) , (1.13). They are not considered in this thesis, because the positiveness of the coefficients does not guarantee stability of the transfer function generated by these $CFEs$.

4) $CFE3$

It has been recently reported that another type of continued fraction expansion is possible, as given below [6].

For n is odd,

CFE 3a

$$\frac{F_{1o}(z)}{F_{2o}(z)} = kz(1)\frac{z+1}{z-1} + \frac{1}{kf(2)\frac{z-1}{z+1} + kz(2)\frac{z+1}{z-1} + \frac{1}{\cdots\cdots}} \tag{1.16}$$

For n is even,

CFE 3b

$$\frac{F_{1e}(z)}{F_{2e}(z)} = kf(1)\frac{z-1}{z+1} + kz(1)\frac{z+1}{z-1} + \frac{1}{kf(2)\frac{z-1}{z+1} + kz(2)\frac{z+1}{z-1} + \frac{1}{\cdots\cdots}} \tag{1.17}$$

1.4 Scope of the thesis

It is known that, for a stable $1-D$ IIR filter, the denominator of the transfer function, whose zeros lie inside the unit circle, can be decomposed as the mirror-image polynomial $F_1(z)$ and the anti-mirror-image polynomial $F_2(z)$. In addition, the function $\frac{F_1(z)}{F_2(z)}$ or its reciprocal has simple poles and zeros on the unit circle, and they interlace. The methods to generate continued fraction expansions from mirror-image polynomials $F_1(z)$ and the anti-mirror-image polynomials $F_2(z)$ are discussed in following chapters.

One of the objectives of this thesis is to expand the denominator of a stable transfer function into $CFEs$.

Chapter 1 povides an overview of the stability of the IIR filter, mirror-image polynomial, anti-mirror-image polynomial and some properties; it also presents an introduction to the field of the known $CFE1$, $CFE2$ and $CFE3$. In Chapter 2, the methods to generate the coefficients of continued fraction expansions are described, which are illustrated with MATLAB. Chapter 2 is concerned primarily with the new expansion type of $CFE4$, which is given below:

$CFE4$:

For n = odd,

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\frac{(z-1)}{(z+1)} + kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \cfrac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \cfrac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \cfrac{1}{\cdots\cdots}}}$$

where $kc(0), kc(1), kc(2)...kc(\frac{n-1}{2})$ are real and positive; $0 \leq |\gamma_i| < 2, i = 1, 2...\frac{n-1}{2}$.

For n = even,

$$\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \cfrac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \cfrac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \cfrac{1}{\cdots\cdots}}}$$

where $kc(0) = 0$; $kc(1), kc(2)...kc(\frac{n}{2})$ are real and positive; $0 \leq |\gamma_i| < 2, i = 1, 2...\frac{n}{2}$.

In addition, algorithms have been given to obtain the various coefficients of $CFE4$.

In Chapter 3, Chapter 4 and Chapter 5, a few applications are outlined. The most common types of filters, namely Butterworth lowpass filters, Butterworth lowpass CPPF (Complementary Pole-pair Filter) and Chebyshev lowpass filter, have been considered in detail with respect to the above mentioned theories. Chapter 3 establishes algorithms to produce the coefficients ka, kb, kc and γ of the continued fraction expansions ($CFEs$) using the concept of mirror-image polynomial and anti-mirror-image polynomial for Butterworth lowpass filters and Butterworth $CPPFs$ lowpass filters in z domain after bilinear transformation.

Chapter 4 is devoted to an algorithm which produces the coefficients ka, kb, kc and γ of the continued fraction expansion ($CFEs$) using the concept of mirror-image polynomial and anti-mirror-image polynomial for Chebyshev lowpass filters in z domain after bilinear transformation.

Another objective of this thesis is to produce stable lowpass filter transfer functions from coefficients kc and γ of continued fraction expansion ($CFE4$). To produce highpass and bandpass filters in z domain, first, highpass and bandpass filters in s domain are characterized; then, using bilinear transformation, the transfer functions in z domain are obtained, and are discussed in Chapter 5.

This thesis can be considered as a starting point for application of continued fraction expansions in filter design. The same methods can be considered as suggestions for any other stable filter designs.

Chapter 2

The New Type of Continued Fraction Expansion and the Determination of its Coefficients

First, we review some of the issues associated with $CFE1$, $CFE2$ and $CFE3$ techniques; then we produce a new type of $CFE4$. The sufficient and necessary condition are that the denominator of a stable IIR filter can be decomposed as the mirror-image polynomial $F_1(z)$ and the anti-mirror-image polynomial $F_2(z)$, and $\frac{F_1(z)}{F_2(z)}$(or $\frac{F_2(z)}{F_1(z)}$) can be expressed as continued fraction expansions of $CFE1$, $CFE2$, $CFE3$ and $CFE4$, where all coefficients of the continued fraction expansions are real and positive.

Secondly, the methods of coefficient determination for four kinds of continued fraction expansions are introduced.

Finally, there are programmes on the computer by using MATLAB to obtain the various coefficients.

2.1 The new type of z - domain CFE

We now introduce the new type of continued fraction expansion given as follows:

CFE 4a: For n odd,

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\frac{(z-1)}{(z+1)} + kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{\dots\dots}}} \tag{2.1}$$

CFE 4b: For n even,

$$\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{\dots\dots}}} \tag{2.2}$$

We shall now prove the existence of CFE 4a .

Since n is *odd*, there exists a zero of $F_{1o}(z)$ at $z=-1$, we can write

$$\frac{F_{20}(z)}{F_{10}(z)} = kc(0)\frac{(z-1)}{(z+1)} + \phi_1(z) \tag{2.3}$$

where $\phi_1(z)$ is of the type $\frac{F_{2e}(z)}{F_{1e}(z)}$; the degree of $\phi_1(z)$ is one less than that of $\frac{F_{2o}(z)}{F_{1o}(z)}$ and is of even order now.

It is known from [3] that $kc(0)$ is real and positive.

$\phi_1(z)$ can now be written as

$$\phi_1(z) = kc(1)\frac{z^2-1}{z^2+\gamma_1 z+1} + \phi_2(z) \tag{2.4}$$

This is possible, because $\phi_1(z)$ contains its poles and zeros located on the unit circle which are simple and interlacing [5], which means that $|\gamma_1| < 2$.

It is known that $kc(1)$ is real and positive [10]; also $\phi_2(z)$ is of similar form as $\phi_1(z)$. $\phi_2(z)$ contains poles and zeros on the unit circle which are simple and interlace. One of the poles of $\frac{1}{\phi_2(z)}$ is chosen and then expanded further. Therefore, $\frac{1}{\phi_2(z)}$ can now be expanded similar to the one given in Equ (2.4), and the process can be continued. This shows that CFE $4a$ given by Equ (2.1) and CFE $4b$ given by Equ (2.2) are always possible.

The steps for the process will be as follows:

(i) Starting from the given $D(z)$, formulate the mirror - image polynomial $F_1(z)$ as given by Eqs (1.5) and the anti - mirror - image polynomial $F_2(z)$ as given by Eqs (1.6).

(ii) Select one pair of roots on the unit circle from $F_1(z)$ and expand $\frac{F_2(z)}{F_1(z)}$ into CFE $4a$ or CFE $4b$, as the case maybe.

The following observations can be made:

(i) If the order of $D(z)$ is even, we start with the ratio $\frac{F_{2e}(z)}{F_{1e}(z)}$. The coefficient $kc(0)$ corresponding to the term $\frac{z-1}{z+1}$ doesn't need be taken out first. The same can be obtained in CFE $4b$ at any step during the expansion.

(ii) One can choose any of the zeros of $F_{1o}(z)$ or $F_{1e}(z)$ to start with. In fact, the same is possible at every step in CFE $4a$ or CFE $4b$. Therefore, a large number of $CFE's$ of this type will be possible. This is in sharp contract to the existence of $CFE1$, $CFE2$ and $CFE3$, which are unique.

2.2 Determination of the Coefficients of the Continued Fraction Expansion

1) $CFE1$:

For n even or odd,

$$\frac{F_1(z)}{F_2(z)} = ka(1)\frac{(z+1)}{(z-1)} + \frac{1}{\frac{F_3'(z)}{F_4'(z)}} \tag{2.5}$$

$$ka(1) = \lim_{z \longrightarrow 1} \frac{z-1}{z+1} \frac{F_1(z)}{F_2(z)} |_{z=1} \tag{2.6}$$

The above procedure is recursive, so,

$$\frac{F_3'(z)}{F_4'(z)} = ka(2)\frac{(z+1)}{(z-1)} + \frac{1}{\frac{F_5'(z)}{F_6'(z)}} \tag{2.7}$$

Using the same procedure, we can obtain $ka(1), ka(2), ka(3)...ka(n)$, where n is the degree of mirror-image polynomial or anti-mirror polynomial.

2)

(a) CFE $2a$:

For n even,

$$\frac{F_{1e}(z)}{F_{2e}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{\frac{F_3'(z)}{F_4'(z)}} \tag{2.8}$$

$$kb(1) = \lim_{z \longrightarrow -1} \frac{z+1}{z-1} \frac{F_{1e}(z)}{F_{2e}(z)} |_{z=-1} \tag{2.9}$$

The above procedure is recursive, so,

$$\frac{F_3'(z)}{F_4'(z)} = kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{\frac{F_5'(z)}{F_6'(z)}} \tag{2.10}$$

Using the same procedure, we can obtain $kb(1), kb(2), kb(3)...kb(n)$, where n is the degree of mirror-image polynomial or anti-mirror polynomial.

(b) $CFE2$ $2b$:

For n odd,

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{\frac{F_4'(z)}{F_3'(z)}} \tag{2.11}$$

$$kb(1) = \lim_{z \longrightarrow -1} \frac{z+1}{z-1} \frac{F_{2o}(z)}{F_{1o}(z)} |_{z=-1} \tag{2.12}$$

The above procedure is recursive, so,

$$\frac{F_4'(z)}{F_3'(z)} = kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{\frac{F_6'(z)}{F_5'(z)}} \tag{2.13}$$

Using the same procedure, we can obtain $kb(1), kb(2), kb(3)...kb(n)$, where n is the degree of mirror-image polynomial or anti-mirror polynomial.

3)

(a) $CFE\ 3a$:

For n is odd,

$$kz(1) = \lim_{z \longrightarrow 1} \frac{z-1}{z+1} \frac{F_{1o}(z)}{F_{2o}(z)} |_{z=1} \tag{2.14}$$

where $kf(1) = 0$.

(b) $CFE\ 3b$:

For n is even,

$$kf(1) = \lim_{z \longrightarrow -1} \frac{z+1}{z-1} \frac{F_{1e}(z)}{F_{2e}(z)} |_{z=-1} \tag{2.15}$$

$$kz(1) = \lim_{z \longrightarrow 1} \frac{z-1}{z+1} \frac{F_{1e}(z)}{F_{2e}(z)} |_{z=1} \tag{2.16}$$

4)

(a) $CFE\ 4a$:

For n odd,

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\frac{(z-1)}{(z+1)} + \frac{F'_{2e}(z)}{F'_{1e}(z)} \tag{2.17}$$

$$kc(0) = \lim_{z \longrightarrow -1} \frac{z+1}{z-1}\frac{F_{2o}(z)}{F_{1o}(z)} |_{z=-1} \tag{2.18}$$

Then, the degrees of $F'_{1e}(z)$ and $F'_{2e}(z)$ are even.

$$\frac{F'_{2e}(z)}{F'_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{\frac{F'_{4e}(z)}{F'_{3e}(z)}} \tag{2.19}$$

$$kc(1) = \lim_{z \longrightarrow root} \frac{z^2+\gamma_1+1}{z^2-1}\frac{F'_{2e}(z)}{F'_{1e}(z)} |_{z=z_1} \tag{2.20}$$

z_1 is one of the pair of roots of complex conjugate for $z^2 + \gamma_1 z + 1$.

Using the same procedure, we can obtain $kc(0), kc(1), kc(2)...kc(\frac{n-1}{2})$, where n is the degree of mirror-image polynomial or anti-mirror polynomial.

(b) $CFE\ 4b$:

For n even,

$kc(0) = 0$.

The remaining coefficients are obtained as in Eqs (2.19) and (2.20).

Remark:

The foregoing algorithm describes a way of obtaining kc and γ coefficients of a continued fraction expansion. However, for $CFE4$, as the degree of mirror-image polynomial and anti-mirror-image polynomial increases, the number of different terms in $kc(1)$, $kc(2)$... and $\gamma(1)$, $\gamma(2)$... increases correspondingly. $kc(1)$,

$kc(2)$...and $\gamma(1)$, $\gamma(2)$... are different values when we change the order of selecting subpolynomials of $F_{1e}(z)$ or $F'_{1e}(z)$, $F_{3e}(z)$ or $F'_{3e}(z)$, $F_{5e}(z)$ or $F'_{5e}(z)$...... as the denominators $z^2+\gamma_i z+1$ of the continued fraction expansion, $i=1,2,3...\frac{n}{2}$, $n = even$ or $i=1,2,3...\frac{n-1}{2}$, $n = odd$.

We know that $F_1(z)$ and $F_2(z)$ can be expressed as:

n even

$$F_{1e} = k_{1e}\prod_{i=1}^{n/2}(z^2+a_i z+1), 0 \leq |a_i| < 2 \tag{2.21}$$

$$F_{2e} = (z^2-1)k_{2e}\prod_{i=1}^{(n-2)/2}(z^2+\beta_i z+1), 0 \leq |\beta_i| < 2 \tag{2.22}$$

n odd

$$F_{1o} = (z+1)k_{1o}\prod_{i=1}^{(n-1)/2}(z^2+\gamma_i z+1), 0 \leq |\gamma_i| < 2 \tag{2.23}$$

$$F_{2o} = (z-1)k_{2o}\prod_{i=1}^{(n-1)/2}(z^2+\delta_i z+1), 0 \leq |\delta_i| < 2 \tag{2.24}$$

Where k_{1e}, k_{2e}, k_{1o} and $k_{2o}$are positive constants.

$kc(1)$, $kc(2)$...and $\gamma(1)$, $\gamma(2)$... in different groups are determined when we change the order of selecting subpolynomials of $F_{1e}(z)$ or $F'_{1e}(z)$, $F_{3e}(z)$ or $F'_{3e}(z)$, $F_{5e}(z)$ or $F'_{5e}(z)$...... as the denominators $z^2+\gamma_i z+1$ of the continued fraction expansion, which is illustrated by an example as follows, assuming $n=6$.

2.2.1 Example

$F_{1e} = (z^2 + a_1 z + 1)(z^2 + a_2 z + 1)(z^2 + a_3 z + 1)$, $0 \leq |a_i| < 2$

1) step 1, we first select $\gamma_1 = \alpha_1$, for $kc(1)$,

$\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1) \frac{(z^2-1)}{(z^2+\alpha_1 z+1)} + \frac{1}{\frac{F_{4e}(z)}{F_{3e}(z)}}$,

then, $F_{3e}(z) = (z^2 + \zeta_1 z + 1)(z^2 + \zeta_2 z + 1)$, $0 \leq |\zeta_i| < 2$,

step 2, we select $\gamma_2 = \zeta_1$, for $kc(2)$,

$\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1) \frac{(z^2-1)}{(z^2+\alpha_1 z+1)} + \frac{1}{kc(2)\frac{z^2-1}{z^2+\zeta_1 z+1} + \frac{1}{\frac{F_{6e}(z)}{F_{5e}(z)}}}$

then, $F_{5e}(z) = (z^2 + \sigma_1 z + 1)$, $0 \leq |\sigma_i| < 2$,

Step3,we select $\gamma_3 = \sigma_1$, for $kc(3)$,

2) In step2, if we select $\gamma_2 = \zeta_2$, intead of ζ_1, for $kc(2)$,

$\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1) \frac{(z^2-1)}{(z^2+\alpha_1 z+1)} + \frac{1}{kc(2)\frac{z^2-1}{z^2+\zeta_2 z+1} + \frac{1}{\frac{F_{6e}(z)}{F_{5e}(z)}}}$

then, $F_{5e}(z) = (z^2 + \varsigma_1 z + 1)$ $0 \leq |\varsigma_i| < 2$,

Then, in step3,we select $\gamma_3 = \varsigma_1$, for $kc(3)$

This example has given the values of γ_i respectively in two groups when we select $\gamma_2 = \zeta_1$ or $\gamma_2 = \zeta_2$ for $kc(2)$; that means that there are two options to select subpolynomial $(z^2+\zeta_1 z+1)$ or $(z^2+\zeta_2 z+1)$ of $F_{3e}(z)$ as the denominators $z^2+\gamma_2 z+1$ of the continued fraction expansion. If we select the first subpolynomial $(z^2+\zeta_1 z+1)$ of $F_{3e}(z)$ as the denominators $z^2 + \gamma_2 z + 1$ of the continued fraction expansion; then obtain γ_1, γ_2 and γ_3 in the first group as the following respectively,

$$\alpha_1, \zeta_1, \sigma_1$$

If we select the second subpolynomial $(z^2 + \zeta_2 z + 1)$ of $F_{3e}(z)$ as denominators $z^2 + \gamma_2 z + 1$ of the continued fraction expansion; then obtain γ_1, γ_2 and γ_3 in the second group as the following respectively,

$$\alpha_1, \zeta_2, \varsigma_1$$

3) If in Step1, we first select $\gamma_1 = \alpha_2$, for $kc(1)$; then, γ_2 and γ_3 will be different from the above values

$$\alpha_1, \zeta_1, \sigma_1$$

or

$$\alpha_1, \zeta_2, \varsigma_1$$

4) Similarly, we first select $\gamma_1 = \alpha_3$, for $kc(1)$, then, γ_2 and γ_3 will be different again.

Now, we get the conclusion, the values of γ_i are different in each group depending on the order of the subpolynomials which are selected as the denominators of continued fraction expansions.

α_1	ζ_1	σ_1
α_1	ζ_2	ς_1
	...	
	...	
	...	
	...	

Let $Ncoe = \frac{n}{2}$, when $n = even$

$= \frac{n-1}{2}$, when $n = odd$

where n is the degree of mirror-image polynomial or anti-mirror-image polynomial in Matlab function cfekc.m, we suggest that the coefficients of rootsele vector are the subscript values of α_1,ζ_1, $\sigma_1$and α_1, ζ_2,ς_1 and so on respectively.

For example, for the above example, one of $z^2+\gamma_i z+1$ terms is $z^2+\alpha_1 z+1$,$z^2+\zeta_1\mathrm{z}+1$ and $z^2+\sigma_1 z+1$ respectively, so

$$\alpha_1, \zeta_1, \sigma_1 \xrightarrow{subscript} 1, 1, 1,$$

then, rootsele=[1 1 1].

Another $z^2 + \gamma_i z + 1$ term is $z^2 + \alpha_1 z + 1$, $z^2 + \zeta_2 z + 1$, $z^2 + \varsigma_1 + 1$, so

$$\alpha_1, \zeta_2, \varsigma_1 \xrightarrow[\rightarrow]{subscript} 1, 2, 1,$$

then, rootsele=[1 2 1].

......

The maximum number of different terms in $kc(1)$, $kc(2)$... and $\gamma(1)$,$\gamma(2)$... $=$ $Ncoe \times (Ncoe - 1) \times (Ncoe - 2) \times ...3 \times 2 \times 1$

The table below gives the coefficients for each of rootsele vectors to obtain the different kc and γ terms for MATLAB function cfekc.m, and we can produce the different kc and γ vectors correspondingly when we run function cfekc.m.

Table 2.1: Rootsele Vector

rootsele	$Ncoe = 4$	$Ncoe = 3$	$Ncoe = 2$	$Ncoe = 1$
	1	1	1	1
	1	1	2	1
	1	2	1	1
	1	2	2	1
	1	3	1	1
	1	3	2	1
	2	1	1	1
	2	1	2	1
	2	2	1	1
	2	2	2	1
	2	3	1	1
	2	3	2	1
	3	1	1	1
	3	1	2	1
	3	2	1	1
	3	2	2	1
	3	3	1	1
	3	3	2	1
	4	1	1	1
	4	1	2	1
	4	2	1	1
	4	2	2	1
	4	3	1	1
	4	3	2	1
...				

From the table,

1) If $Ncoe = 1$,

The number of different terms in $kc(1), kc(2)...$ and $\gamma(1)$, $\gamma(2)...= Ncoe \times (Ncoe-1)\times(Ncoe-2)\times...3\times2\times1=1$. As a result, the only one is that $rootsele = [1]$. The length of $rootsele$ vector $= Ncoe = 1$.

2) If $Ncoe = 2$,

The number of different terms in $kc(1), kc(2)...$ and $\gamma(1)$, $\gamma(2)...= Ncoe \times (Ncoe - 1) \times (Ncoe - 2) \times ...3 \times 2 \times 1=2\times1$. As a result, the two $rootsele$s are [1 1] and [2 1]. The length of each $rootsele$ vector $=Ncoe= 2$.

3) If $Ncoe = 3$,

The number of different terms in $kc(1), kc(2)...$ and $\gamma(1)$, $\gamma(2)...= Ncoe \times (Ncoe - 1) \times (Ncoe - 2) \times ...3 \times 2 \times 1=3\times2\times1$. As a result, the six $rootsele$s are [1 1 1] ,[1 2 1],[2 1 1],[2 2 1],[3 1 1] and [3 2 1] respectively. The length of each $rootsele$ vector $=Ncoe= 3$

4) If $Ncoe = m$, m is an integer,

The total number of different terms in $kc(1), kc(2)...$ and $\gamma(1)$, $\gamma(2)...= Ncoe \times (Ncoe - 1) \times (Ncoe - 2) \times ...3 \times 2 \times 1 = m \times (m-1) \times (m-2) \times ...3 \times 2 \times 1$.

$rootsele(1) = 1, 2, 3...m$

$rootsele(2) = 1, 2, 3...(m - 1)$

$rootsele(3) = 1, 2, 3...(m - 2)$

...

$rootsele(m - 2) = 1, 2, 3$

$rootsele(m - 1) = 1, 2$

$rootsele(m) = 1$

The length of each $rootsele$ vector $=Ncoe=m$.

5) If $n = 1$, rootsele = any value.

But not all $rootsele$s can be determined as kc and γ. For example, assume,

$n = 6$, $D(z) = 33.7972z^6 + 26.2840z^4 + 3.8596z^2 + 0.0592$.

$F_{1e} = (z^2 + a_1 z + 1)(z^2 + a_2 z + 1)(z^2 + a_3 z + 1)$, $0 \leq |a_i| < 2$

$=(z^2 + 1.4524z + 1)(z^2 - 1.4524z + 1)(z^2 + 1)$

$F_{2e} = 16.8690z^6 + 11.2122z^4 - 11.2122z^2 - 16.8690$

and if we set *rootsele* =[1 2 1].

Step 1: because $rootsele(1) = 1$, so, we select the first subpolynomial $(z^2 + 1.4524z + 1)$ of $F_{1e}(z)$ as the denominator $z^2 + \gamma_1 z + 1$ of the continued fraction expansion, $\gamma_1 = 1.4524$.

$\frac{F_{2e}(z)}{F_{1e}(z)} = 0.4191 \frac{(z^2-1)}{(z^2+1.4524z+1)} + \frac{1}{\frac{F_{4e}(z)}{F_{3e}(z)}}$, and $k(1) = 0.4191$.

It is easy to find $F_{3e} = (z^2 - 0.3984z + 1)(z^2 - 1)$

$F_{4e} = 16.9282z^4 - 24.5877z^3 + 33.8564z^2 - 24.5877z + 16.9282$

Step 2: because $rootsele(2) = 2$, so, we select the second subpolynomial (z^2-1) of $F_{3e}(z)$ as the denominator $z^2 + \gamma_2 z + 1$ of the continued fraction expansion. As we have known before, this condition, the zeros of $z^2 + \gamma_i z + 1$ are strictly inside the unit circle, must be satisfied, so $(z^2 - 1)$ can't be as the denominator $z^2 + \gamma_2 z + 1$. As a result, there is no solution for *rootsele* = [1 2 1].

2.3 Implementation of Algorithms for Coefficient Determination

2.3.1 The M-file cfekakb.m is Employed to Compute Coefficients ka and kb of $CFE1$ and $CFE2$ Respectively

Algorithm 1: Matlab Function cfekakb.m

```
function [ka,kb] = cfekakb( d )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [ka,kb] = cfekakb(d) finds the coefficients ka and kb of the continued
```

```
% fraction expansion (CFE) for F1(z)/F2(z) or F2(z)/F1(z), where F1(z)
% is mirror-image polynomial; F2(z) is antimirror-image polynomial of D.
% F1(z) = [D(z)+z^n D(z^-1)]/2
% F2(z) = [D(z)-z^n D(z^-1)]/2
% d is the coefficients of polynomial respectively in descending powers of z ,
% and d(1) > 0.
%
% CFE1:
% For n = even or odd,
% F1(z)/F2(z) = ka(1)(z+1)/(z-1) + 1/(ka(2)(z+1)/(z-1) + 1/(ka(3)(z+1)/(z-1) + 1/......))
% CFE2:
% For n = even,
% F1e(z)/F2e(z) = kb(1)(z-1)/(z+1) + 1/(kb(2)(z-1)/(z+1) + 1/(kb(3)(z-1)/(z+1) + 1/......))
% Or if n = odd,
% F2o(z)/F1o(z) = kb(1)(z-1)/(z+1) + 1/(kb(2)(z-1)/(z+1) + 1/(kb(3)(z-1)/(z+1) + 1/......))
% d is coefficients of polynomial D respectively in descending powers of z.
%% Convert F1(z)/F2(z) or F2(z)/F1(z) to continued fraction expansions %%

% d(1) is non- negative
if( d(1) <= 0 )
  error('First coefficient in d vector must be non- negative.')
  return
end

% Obtain the mirror- image polynomial f1 and
% antimirror-image polynomial f2
f1=0.5*d+0.5*fliplr(d);
```

```
f2=0.5*d-0.5*fliplr(d);
LF1 = length(f1);
LF2 = length(f2);

if( LF2 == 1 )
   error('No continued-fraction expansion!')
   return
end

%  Check if 1 and -1 are the roots of f1 and f2
f1value=polyval(f1,1); % the value of f1 at z = 1
f2value=polyval(f2,1); % the value of f2 at z = 1
f1val=polyval(f1,-1);  % the value of f1 at z = -1
f2val=polyval(f2,-1);  % the value of f2 at z = 1

tol = 0.000001; % tolerance

% If f1/f2 can be expressed in (z+1)/(z-1) form
if abs(f2value)<=tol
   fnum=[1 1]; fden=[1,-1];
   rootfden=1;
   mir_poly = f1; antimir_poly = f2;
   ka=subcfekakb( mir_poly,antimir_poly,fnum,fden,rootfden);
else
   disp('No coefficients ka.')
end

% If f1/f2 or f2/f1 can be expressed in (z-1)/(z+1) form
```

```
if abs(f2val)<= tol
   fnum=[1 -1]; fden=[1,1];
   rootfden=-1;
   mir_poly = f1; antimir_poly = f2;
   kb=subcfekakb( mir_poly,antimir_poly,fnum,fden,rootfden);

% If -1 is a root of f1,exchange f1 and
% f2,f2/f1 can be expressed as (z-1)/(z+1)form
elseif (abs(f2val)> tol)&(abs(f1val)<= tol)
   Ncoe = LF2-1;
   mir_poly = f2; antimir_poly = f1;
   fnum=[1 -1]; fden=[1,1];
   rootfden=-1;
   kb=subcfekakb( mir_poly,antimir_poly,fnum,fden,rootfden);
else
   disp('No coefficients kb.')
end
```

Algorithm 2: Matlab Subfunction subcfekakb.m

```
function [ k ] = subcfekakb( mir_poly,antimir_poly,fnum,fden,rootfden)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is a subfunction of cfekakb
%% Convert F1(z)/F2(z) or F2(z)/F1(z) to continued fraction expansions %%

% Tolerance of poles
pole_tol=0.001;
Ncoe = length(antimir_poly)-1;
```

```
% Initial coefficients k
k=zeros(1,Ncoe);

% Obtain k
for j=1: Ncoe     %— number of k
    f4=deconv(antimir_poly,fden); %—
   k(j)=polyval(mir_poly,rootfden)./(polyval(f4,rootfden).*polyval(fnum,rootfden));
    ftemp = conv(mir_poly,fden)-k(j)*conv(antimir_poly,fnum);
    f3=deconv(ftemp,conv(fden,fden)); %—
    f3value=polyval(f3,rootfden);
% If rootfden is a root of f3, continue loop structure
    if (abs(f3value) <= pole_tol)
        mir_poly=f4;
        antimir_poly=f3;
    else
        break
    end
end
```

2.3.2 The M-file cfekeo.m is Employed to Compute Coefficients kf and kz of $CFE3$

Algorithm 3: Matlab Function cfekeo.m

```
function [kf,kz] = cfekeo(d)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [kf,kz] = cfekeo(d) finds the coefficients kf and kz of the continued
```

% *fraction expansion (CFE3) of* $F_1(z)/F_2(z)$ *, where* $F_1(z)$ *is mirror-image*

% *polynomial and* $F_2(z)$ *is antimirror-image polynomial of D.*

% *d is coefficients of polynomial D respectively in descending powers of z.*

%

% For n is odd,

% $CFE\ 3a$

% $\frac{F_{1o}(z)}{F_{2o}(z)} = kz(1)\frac{z+1}{z-1} + \frac{1}{kf(2)\frac{z-1}{z+1}+kz(2)\frac{z+1}{z-1}+\frac{1}{......}}$

% For n is even,

% $CFE\ 3b$

% $\frac{F_{1e}(z)}{F_{2e}(z)} = kf(1)\frac{z-1}{z+1} + kz(1)\frac{z+1}{z-1} + \frac{1}{kf(2)\frac{z-1}{z+1}+kz(2)\frac{z+1}{z-1}+\frac{1}{......}}$

%% *Convert* $F_1(z)/F_2(z)$ *to continued fraction expansions* %%

```
if( d(1) <= 0 )
    error('First coefficient in d vector must be non- negative.')
    return
end

% The coefficient of d vector must be real value
for ii=1:length(d)
  if( any(imag(d(ii))))
    error('The coefficient of d vector must be real value.')
    return
  end
end

% To obtain mirror-imagine poly f1 and antimirror-imagine polynomial
f1=0.5*d+0.5*fliplr(d);
f2=0.5*d-0.5*fliplr(d);
```

```
LF1 = length(f1);
LF2 = length(f2);

antimir_poly = f2;
mir_poly = f1;
numz=[1 1];
denf=[1 -1];
numf=[1 -1];
denz=[1 1];

% If degree of d is odd, To obtain coefficients kf and kz of continued-fractor
% expansion
if rem((length(mir_poly)-1),2)==1   %— degree of d is odd
   tempf1=deconv(antimir_poly,denf);  %—
   ko(1)=polyval(mir_poly,1)./(polyval(tempf1,1).*polyval(numz,1));
   tempantif2 = conv(mir_poly,denf)-ko(1)*conv(antimir_poly,numz);
   tempf2=deconv(conv(tempantif2,tempf1),conv(antimir_poly,denf));   %—
   mir_poly=tempf1;
   antimir_poly=tempf2;
   [knumberf,knumberz]=subcfekeo(mir_poly,antimir_poly);
   kz=[ko(1) knumberz];
   kf=[0 knumberf];

% If degree of d is even, To obtain coefficients kf and kz of continued-fraction
% expansion
elseif rem((length(mir_poly)-1),2)==0   %— degree of d is even
   [kf,kz]=subcfekeo(mir_poly,antimir_poly);
end
```

Algorithm 4: Matlab Function subckekeo.m

```
function [ knumberf,knumberz ] = subcfekeo( mir_poly,antimir_poly)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   This function is a subfunction of cfekeo
%% Convert F1(z)/F2(z) to continued fraction expansions %%

pole_tol=0.001; %  Tolerance of poles
numz=[1 1];
denf=[1 -1];
numf=[1 -1];
denz=[1 1];

if rem((length(mir_poly)-1),2)==1   %— degree of d is odd
   Ncoe = (length(antimir_poly)-2)/2;
elseif rem((length(mir_poly)-1),2)==0   %— degree of d is even
   Ncoe = (length(antimir_poly)-1)/2;
end

knumberf=zeros(1,Ncoe); %  Initial coefficients of vector
knumberz=zeros(1,Ncoe); %  Initial coefficients of vector

for j=1: Ncoe      %— number of k
    fden=conv(denz,denf);
    fden1=deconv(antimir_poly,denz);
    knumberf(j)=polyval(mir_poly,-1)./(polyval(fden1,-1).*polyval(numf,-1));
    fden2=deconv(antimir_poly,denf);
    knumberz(j)=polyval(mir_poly,1)./(polyval(fden2,1).*polyval(numz,1));
```

```
f3=deconv(antimir_poly,fden); %—
temp=knumberf(j)*conv(numf,denf)+ knumberz(j)*conv(numz,denz);
f4temp = mir_poly - conv(temp,f3);
f4=deconv(conv(f4temp,f3),antimir_poly); %—
f4valuez=polyval(f4,1);
f4valuef=polyval(f4,-1);

% If rootfden is a root of f3, continue loop structure
if (abs(f4valuez) <= pole_tol)& (abs(f4valuef) <= pole_tol)
antimir_poly=f4;
mir_poly=f3;
else
break
end
end
```

2.3.3 The M-file cfekc.m is Employed to Compute Coefficients kc and γ of $CFE4$

Algorithm 5: The Matlab Function cfekc.m

```
function [k0,kc,r] = cfekc(d,rootsele)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [k0,kc,r] = cfekc(d,rootsele) finds the coefficients kc and γ of the continued
% fraction expansion (CFE) of F2(z)/F1(z), where F1(z) is mirror-image
% polynomial and F2(z) is antimirror-image polynomial of D.
```

% $F_1(z) = \frac{[D(z)+z^n D(z^{-1})]}{2}$

% $F_2(z) = \frac{[D(z)-z^n D(z^{-1})]}{2}$

```
% d is the coefficients of polynomial respectively in descending powers of z ,
% and d(1) > 0.
%
% CFE4:
% For n = odd,
```

$$\% \frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\frac{(z-1)}{(z+1)} + kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{\cdots\cdots}}}$$

```
% CFE4:
% For n = even,kc(0)
```

$$\% \frac{F_{2e}(z)}{F_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{\cdots\cdots}}}$$

```
%% Convert F2(z)/F1(z) to continued fraction expansions %%

% format short;
if( d(1) <= 0 )
    error('First coefficient in d vector must be non- negative.')
    return
end

%  The coefficient of d vector must be real
for ii=1:length(d)
  if( any(imag(d(ii))))
    error('The coefficient of d vector must be real value.')
    return
  end
end

%  To obtain mirror-imagine poly f1 and antimirror-imagine polynomial
f1=0.5*d+0.5*fliplr(d);
```

```
f2=0.5*d-0.5*fliplr(d);
LF1 = length(f1);
LF2 = length(f2);

% Tolerance of imag of polynomial
imag_tol=0.000001;

antimir_poly = f2;
mir_poly = f1;

% If degree of d is odd, obtain coefficients k0 and kc of continued-fractor
% expansion and z^2+riz+1

if rem((length(mir_poly)-1),2)==1   % degree of d is odd
   tempden=[1 1];
   tempnum=[1 -1];
   f00=deconv(mir_poly,tempden);
   k0=polyval(antimir_poly,-1)./(polyval(f00,-1).*polyval(tempnum,-1));
   ftemp = conv(antimir_poly,tempden)-k0*conv(mir_poly,tempnum);
   f01=deconv(ftemp,conv(tempden,tempden));
   mir_poly=f00;
   antimir_poly=f01;
   [k,r]=subcfekc(mir_poly,antimir_poly,rootsele);
   kc=k;

% If degree of d is even, To obtain coefficients kc of continued-fractor
% expansion and z^2+r1Z+1
elseif rem((length(mir_poly)-1),2)==0   % degree of d is even
```

```
  [k,r]=subcfekc(mir_poly,antimir_poly,rootsele);
  k0 = 0;
  kc=k;
end
```

Algorithm 6: Matlab Subfunction subcfekc.m

```
function [k,r] = subcfekc(mir_poly,antimir_poly,rootsele)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function is a subfunction of cfekc.
% Find the coefficients of the continued-fractionexpansion of F2(z)/F1(z),
% for n = even; F1(z) is mirror-image polynomial and F2(z) is antimirror-
% image polynomial of D.
%
% CFE4 :
% For n = even,kc(0)
```

% $\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \cfrac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \cfrac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{\cdots\cdots}}}$

```
%% Convert F2(z)/F1(z) to continued fraction expansions %%

imag_tol = 0.000001;

if rem((length(mir_poly)-1),2)==0   % degree of d is even
  Ncoe = (length(mir_poly)-1)/2;
  k=zeros(1,Ncoe);  % initialize coefficients k
  fden = zeros(Ncoe,3); % initialize Z^2+r(n)z+1
  i=0;
  jj=0;
end
```

```
for j=1:Ncoe,       % number of k
    i=i+1;
    jj=jj+1;
    antimir_root = roots(antimir_poly);
    mir_root = roots(mir_poly);

 if (i <= Ncoe-1)
      % index z^2+r(n)z+1
      index=rootsele(jj);
    ind = find((imag(mir_root)<= 0)&(abs(real(mir_root)-1))>=imag_tol);

      if (abs(imag(mir_root(ind(index))))<=imag_tol)
         error('No solution.')
         return
      end
      fdenroot = [mir_root(ind(index)),conj(mir_root(ind(index)))];
      fden(j,:) = poly(fdenroot);  % set Z^2+r(n)+1
      % build subpolynomial of f1 for denominator of continued-fractor
      % expansion
      fnum=[1 0 -1];
      % build the numerator of continued-fractor poly
      rootfden = mir_root(ind(index)); % initial z = a root of z^2+r(n)z+1

      f4=deconv(mir_poly,fden(j,:));  %—
k(j)=real(polyval(antimir_poly,rootfden)./(polyval(f4,rootfden).*polyval(fnum,rootfden)))
      ftemp = antimir_poly-k(j)*conv(f4,fnum);
      f3=deconv(ftemp,fden(j,:));   %—
```

```
      if (abs(imag(f3)) <= imag_tol)  % If f3 is real, continue loop structure
        antimir_poly=real(f4);  % coefficients of antimir_poly are real
        mir_poly=real(f3);

      else
        break
      end
%  keyboard;
  elseif i==Ncoe
      % When degree is 6,10,14..., f4= z^2-1,
     % obtain the last continued-fractor poly
     if rem(Ncoe,2)==1
       fdenroot = [mir_root(1),conj(mir_root(1))];
       fden(j,:) = poly(fdenroot);
       k(j)= antimir_poly(1)/mir_poly(1);
     end

% When degree is 4,8,12...f3= z^2-1, obtain the last continued-fractor poly
     if  rem(Ncoe,2)==0
       fdenroot = [antimir_root(1),conj(antimir_root(1))];
       fden(j,:) = poly(fdenroot);
       k(j)= mir_poly(1)/antimir_poly(1);
     end
  end
end

r=fden(:,2);
r=r';
```

2.4 Summary and Discussion

In this chapter, a new type of $z - domain$ continued fraction expansion $CFE4$ has been introduced. It is known that this type of $CFE4$ is always possible for a rational function formulated by the ratio ($\frac{F_{2o}(z)}{F_{1o}(z)}$ or $\frac{F_{2e}(z)}{F_{1e}(z)}$) of anti - mirror - image polynomial to mirror - image polynomial, which are obtained by a Schur polynomial containing all its zeros within the unit circle. The coefficients of this CFE are of the type given in Eqs (2.18) and Eqs (2.20). It is also shown that this type of CFE is non-unique as compared with the other three types of $CFEs$ - $CFE1$, $CFE2$ and $CFE3$, which are unique. This is because the coefficients of $CFE4$ are dependent on the poles and zeros of the $\frac{F_{2o}(z)}{F_{1o}(z)}$ or $\frac{F_{2e}(z)}{F_{1e}(z)}$. Algorithms have been given in order to obtain all possible $CFE4's$.

Chapter 3

The New Continued Fraction Expansion for Butterworth Lowpass Filter and Butterworth Lowpass *CPPF* Filter in z Domain

Generally, Butterworth lowpass filter approximations are based in Laplace transform domain[11] and [12]. By applying the bilinear transformation $s = \frac{z-1}{z+1}$, the transfer functions in discrete domain are characterized. In this chapter, we shall obtain the denominator polynomials of Butterworth lowpass filters of different order and also the coefficients in the various continued fraction expansions.

In addition, we shall discuss the new family of transitional filters - Complementary Pole-pair Filters ($CPPFs$) [13], whose design is based on the judicious position of symmetry poles in the s-plane. For such filters also, we obtain the coefficients in the various continued fraction expansions.

3.1 The Application of the Continued Fraction Expansion for Butterworth Lowpass Filter in z domain

First we take the nth-order Butterworth lowpass filter polynomial in s domain, and by applying the bilinear transformation $s = k\frac{z-1}{z+1}$, we obtain the digital lowpass transfer functions

$$T_{Bd} = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_{Bd}(z)} \tag{3.1}$$

In this thesis, k is chosen to be unity. For each order of $D_{Bd}(z)$, the Butterworth polynomials in the z - domain are obtained which are listed in Table 3.1 up to order $n = 8$. For these polynomials, the mirror-image polynomial $F_{B1}(z)$ and the anti-mirror-image polynomials $F_{B2}(z)$ are obtained and listed in Table 3.2.

Table 3.1: The Butterworth Denominator Polynomials in the z - domain

	$D_{Bd}(z)$
2	$3.4142z^2 + 0.5858$
3	$6.0000z^3 + 2z$
4	$10.6405z^4 + 5.1716z^2 + 0.1880$
5	$18.9443z^5 + 12.0000z^3 + 1.0557z$
6	$33.7972z^6 + 26.2840z^4 + 3.8596z^2 + 0.0592$
7	$60.3672z^7 + 55.5362z^5 + 11.6328z^3 + 0.4638z$
8	$107.9065z^8 + 114.4818z^6 + 31.3887z^4 + 2.2045z^2 + 0.0185$

Table 3.2: The Mirror - image and Anti - mirror - image Polynomials of the z - domain Butterworth Denominator Polynomials Given in Table 3.1

n	$F_{B1}(z)$ and $F_{B2}(z)$
2	$F_{B1}(z) = 2.0000z^2 + 2.0000$ $F_{B2}(z) = 1.4142z^2 - 1.4142$
3	$F_{B1}(z) = 3.0000z^3 + z^2 + z + 3$ $F_{B2}(z) = 3.0000z^3 - z^2 + z - 3$
4	$F_{B1}(z) = 5.4142z^4 + 5.1716z^2 + 5.4142$ $F_{B2}(z) = 5.2263z^4 - 5.2263$
5	$F_{B1}(z) = 9.4721z^5 + 0.5279z^4 + 6.0000z^3 + 6.0000z^2 + 0.5279z + 9.4721$ $F_{B2}(z) = 9.4721z^5 - 0.5279z^4 + 6.0000z^3 - 6.0000z^2 + 0.5279z - 9.4721$
6	$F_{B1}(z) = 16.9282z^6 + 15.0718z^4 + 15.0718z^2 + 16.9282$ $F_{B2}(z) = 16.8690z^6 + 11.2122z^4 - 11.2122z^2 - 16.8690$
7	$F_{B1}(z) = 30.1836z^7 + 0.2319z^6 + 27.7681z^5 + 5.8164z^4 + 5.8164z^3 + 27.7681z^2 + 0.2319z + 30.1836$ $F_{B2}(z) = 30.1836z^7 - 0.2319z^6 + 27.7681z^5 - 5.8164z^4 + 5.8164z^3 - 27.7681z^2 + 0.2319z - 30.1836$
8	$F_{B1}(z) = 53.9625z^8 + 58.3431z^6 + 31.3887z^4 + 58.3431z^2 + 53.9625$ $F_{B2}(z) = 53.9440z^8 + 56.1387z^6 - 56.1387z^2 - 53.9440$

The following algorithm, called butterworthkakb.m, is employed to obtain ka and kb coefficients of continued fraction expansions for a nth-order denominator of Butterworth lowpass filter polynomial in z domain.

Algorithm 7: Matlab Function butterworthkakb.m

---

function [ka, kb] = butterworthkakb(d)

%%

% d is a vector which represents the coefficients of a denominator polynomial

% of Butterworth filter respectively in descending powers of s domain,

% applying bilinear transformation $s = \frac{z-1}{z+1}$, *we get* $T_{Bd} = k_B \frac{N_{Bd}(z)}{D_{Bd}(z)} = \frac{(z+1)^n}{D_{Bd}(z)}$,

% $D_{Bd}(z)$ *is a denominator polynomial of Butterworth filter in* z *domain.*

% *Find coefficients ka and kb of the continued-fractionexpansion (CFE) of*

% $F_{B1}(z)/F_{B2}(z)$ *or* $F_{B2}(z)/F_{B1}(z)$, *where* $F_{B1}(z)$ *is mirror-image*

% *polynomial, and* $F_{B2}(z)$ *is antimirror-image polynomial of* $D_{Bd}(z)$.

% $F_{B1}(z) = \frac{[D_{Bd}(z)+z^n D_{Bd}(z^{-1})]}{2}$

% $F_{B2}(z) = \frac{[D_{Bd}(z)-z^n D_{Bd}(z^{-1})]}{2}$

%

% *CFE1:*

% *For n = even or odd,*

% $\frac{F_{B1}(z)}{F_{B2}(z)} = ka(1)\frac{(z+1)}{(z-1)} + \frac{1}{ka(2)\frac{(z+1)}{(z-1)} + \frac{1}{ka(3)\frac{(z+1)}{(z-1)} + \frac{1}{\cdots\cdots}}}$

% *CFE2:*

% *For n = even,*

% $\frac{F_{B1e}(z)}{F_{B2e}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\cdots\cdots}}}$

% *Or if n = odd,*

% $\frac{F_{2o}(z)}{F_{1o}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\cdots\cdots}}}$

% *Convert* $F_{B1}(z)/F_{B2}(z)$ *or* $F_{B2}(z)/F_{B1}(z)$ *to continued fraction expansions*

```
% d(1) must be non-negative.
if( d(1) <= 0 )
    error('First coefficient in d vector must be non- negative.')
    return
end

% The coefficient of d vector must be real
for ii = 1:length(d)
  if( any(imag(d(ii))))
    error('The coefficient of d vector must be real value.')
```

```
    return
  end
end

% Get ka and kb coefficients of the continued-fraction expansion
[Nd,Dd]=bilinear(1,d,0.5);
denpoly=Dd/Nd(1);
[ka,kb]=cfekakb(denpoly);
```

The following table 3.3 gives the coefficients $ka's$ of $CFE1$ for denominators of Butterworth lowpass polynomials.

d is a vector which represents the coefficients of a denominator polynomial of Butterworth lowpass filter respectively in descending powers of s domain, in the form $s^n + a_1 s^{n-1} + a_2 s^{n-2} + ... + a_{n-2}s^2 + a_{n-1}s + 1$, for a Butterworth filter function of the order n, with pass-band from 0 to 1 rad/sec. By applying bilinear transformation $s = \frac{z-1}{z+1}$, we present $T_{Bd} = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_{Bd}(z)}$, $D_{Bd}(z)$ is a denominator polynomial of Butterworth lowpass filter in z domain. $ka's$ are coefficients of the continued-fraction expansion ($CFE1$) of $F_{B1}(z)/F_{B2}(z)$, where $F_{B1}(z)$ is mirror-image polynomial, and $F_{B2}(z)$ is antimirror-image polynomial of $D_{Bd}(z)$.

Table 3.3: The Values of $ka's$ in $CFE1$ for Denominators of Butterworth Lowpass Polynomials

n	$ka(1)$	$ka(2)$	$ka(3)$	$ka(4)$	$ka(5)$	$ka(6)$	$ka(7)$	$ka(8)$
2	0.7071	1.4142						
3	0.5000	1.3333	1.5000					
4	0.3827	1.0824	1.5772	1.5307				
5	0.3090	0.8944	1.3820	1.6944	1.5451			
6	0.2588	0.7579	1.2016	1.5529	1.7593	1.5529		
7	0.2225	0.6560	1.0550	1.3971	1.6589	1.7988	1.5576	
8	0.1950	0.5774	0.9367	1.2583	1.5277	1.7283	1.8252	1.5636

The following table 3.4 gives the coefficients of $kb's$ of $CFE2$ for denominators of Butterworth lowpass polynomials.

We present $T_{Bd} = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_{Bd}(z)}$. $kb's$ are coefficients of the continued-fraction expansion ($CFE2$) of $F_{B1}(z)/F_{B2}(z)$ or $F_{B2}(z)/F_{B1}(z)$, where $F_{B1}(z)$ is mirror-image polynomial, and $F_{B2}(z)$ is antimirror-image polynomial of $D_{Bd}(z)$.

Table 3.4: The Values of $kb's$ in $CFE2$ for Denominators of Butterworth Lowpass Polynomials

n	$kb(1)$	$kb(2)$	$kb(3)$	$kb(4)$	$kb(5)$	$kb(6)$	$kb(7)$	$kb(8)$
2	0.7071	1.4142						
3	0.5000	1.3333	1.5000					
4	0.3827	1.0824	1.5772	1.5307				
5	0.3090	0.8944	1.3820	1.6944	1.5451			
6	0.2588	0.7579	1.2016	1.5529	1.7593	1.5529		
7	0.2225	0.6560	1.0550	1.3971	1.6589	1.7988	1.5576	
8	0.1950	0.5774	0.9367	1.2583	1.5277	1.7283	1.8252	1.5636

Using matlab function cfekeo.m, we obtain the values of $kf's$ and $kz's$ in $CFE3$. The following table 3.5 gives the values of $kf's$ and $kz's$ in $CFE3$

Table 3.5: The Values of $kf's$ and $kz's$ in $CFE3$ for Denominators of Butterworth Lowpass Polynomials

n	$kf(1)$	$kz(1)$	$kf(2)$	$kz(2)$	$kf(3)$	$kz(3)$	$kf(4)$	$kz(4)$
2	0.7071	0.7071						
3	0	0.5000	0.6667	1.3333				
4	0.3827	0.3827	1.8478	1.8478				
5	0	0.3090	0.3416	0.8944	2.1180	2.6180		
6	0.2588	0.2588	0.9428	0.9428	2.8978	2.8978		
7	0	0.2225	0.2341	0.6560	0.9695	1.4010	3.0778	3.4161
8	0.1951	0.1951	0.6509	0.6509	1.4194	1.4194	3.9231	3.9231

The algorithm, called butterworthkc.m, is employed to compute kc or $k0$ and γ coefficients of $CFE4$ for a nth-order denominator of Butterworth lowpass filter polynomial.

Algorithm 8: Matlab Function butterworthkc.m

function [k0, kc, r] = butterworthkc(d,rootsele)

%%%

% d is a vector which represents the coefficients of a denominator polynomial

% of Butterworth filter respectively in descending powers of s domain,

% applying bilinear transformation $s = \frac{z-1}{z+1}$*,we get* $T_{Bd} = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_{Bd}(z)}$,

% $D_{Bd}(z)$ *is a denominator polynomial of Butterworth filter in z domain.*

% Find coefficients kc and r of the continued-fraction expansion (CFE) of

% $F_{B1}(z)/F_{B2}(z)$ *or* $F_{B2}(z)/F_{B1}(z)$*, where* $F_{B1}(z)$ *is mirror-image*

% polynomial, and $F_{B2}(z)$ is antimirror-image polynomial of $D_{Bd}(z)$.

% $F_{B1}(z) = \frac{[D_{Bd}(z)+z^n D_{Bd}(z^{-1})]}{2}$

% $F_{B2}(z) = \frac{[D_{Bd}(z)-z^n D_{Bd}(z^{-1})]}{2}$

%

% CFE4:

% For n = odd,

% $\frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\frac{(z-1)}{(z+1)} + kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)}+\frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)}+\frac{1}{\cdots\cdots}}}$

% CFE4:

% For n = even,$kc(0)$

% $\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)}+\frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)}+\frac{1}{\cdots\cdots}}}$

%% Convert $F_2(z)/F_1(z)$ to continued fraction expansions %%

```
% d(1) must be non-negative.
if( d(1) <= 0 )
    error('First coefficient in d vector must be non- negative.')
    return
end

% The coefficient of d vector must be real value
for ii = 1:length(d)
  if( any(imag(d(ii))))
    error('The coefficient of d vector must be real value.')
    return
  end
end

% Get kc and r coefficients of the continued-fraction expansion
```

```
[Nd,Dd]=bilinear(1,d,0.5);
denpoly=Dd/Nd(1);
[k0, kc, r]=cfekc(denpoly,rootsele);
```

The following table 3.6 gives the values of coefficients $kc's$ and $\gamma's$ in $CFE4$ for denominators of Butterworth lowpass filters.

We present $T_{Bd} = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_{Bd}(z)}$. kc and γ are coefficients of the continued-fraction expansion ($CFE4$) of $F_{B2}(z)/F_{B1}(z)$, where $F_{B1}(z)$ is mirror-image polynomial, and $F_{B2}(z)$ is antimirror-image polynomial of $D_{Bd}(z)$.

Table 3.6: The values of $kc's$ and $\gamma's$ for Denominators of Butterworth LP Filters

n	$kc(0)$	$kc(1)$	$kc(2)$	$kc(3)$	$kc(4)$		$\gamma(1)$	$\gamma(2)$	$\gamma(3)$	$\gamma(4)$
2		0.7071					0.0000			
3	0.5000	0.5000					-0.6667			
4		0.4826	0.4826				1.0222	-1.0222		
		0.4826	0.4826				-1.0222	1.0222		
5	0.3090	0.4578	0.2332				-1.2754	0.3311		
	0.3090	0.2332	0.4578				0.3311	-1.2754		
6		0.4191	0.1893	0.6483			1.4525	-0.3984	-1.2323	
		0.4191	0.1893	0.6483			-1.4525	0.3984	1.2323	
		0.1584	0.6293	1.7735			0.0000	0.0000	0.0000	
7	0.2225	0.3803	0.1489	0.4222			-1.5758	0.1566	0.4635	
	0.2225	0.2519	0.1939	0.5852			0.7951	-0.5887	-1.4015	
	0.2225	0.1453	0.5358	0.9560			-0.2116	-0.1497	-1.0310	
8		0.3454	0.2371	0.6809	0.0935		1.6637	-0.8770	-1.4063	0.2972
		0.3454	0.2371	0.0935	0.6809		1.6637	-0.8770	0.2972	-1.4063
		0.3454	0.0997	0.3366	0.3634		1.6637	0.0915	-0.1722	-1.6379
		0.3454	0.0997	0.3634	0.3366		1.6637	0.0915	-1.6379	-0.1722
		0.3454	0.2371	0.6809	0.0935		-1.6637	0.8770	1.4063	-0.2972
		0.3454	0.2371	0.0935	0.6809		-1.6637	0.8770	-0.2972	1.4063
		0.3454	0.0997	0.3366	0.3634		-1.6637	-0.0915	0.1722	1.6379
		0.3454	0.0997	0.3634	0.3366		-1.6637	-0.0915	1.6379	0.1722

Continued

n	$kc(0)$	$kc(1)$	$kc(2)$	$kc(3)$	$kc(4)$		$\gamma(1)$	$\gamma(2)$	$\gamma(3)$	$\gamma(4)$
8		0.1545	0.5177	1.1130	0.3897		0.3899	0.5695	0.9654	-1.5371
		0.1545	0.5177	0.3897	1.1130		0.3899	0.5695	-1.5371	0.9654
		0.1545	0.2493	0.6959	0.3749		0.3899	-0.8882	-1.3751	1.6173
		0.1545	0.2493	0.3749	0.6959		0.3899	-0.8882	1.6173	-1.3751
		0.1545	0.5177	1.1130	0.3897		-0.3899	-0.5695	-0.9654	1.5371
		0.1545	0.5177	0.3897	1.1130		-0.3899	-0.5695	1.5371	-0.9654
		0.1545	0.2493	0.6959	0.3749		-0.3899	0.8882	1.3751	-1.6173
		0.1545	0.2493	0.3749	0.6959		-0.3899	0.8882	-1.6173	1.3751

3.2 The Application of the Continued Fraction Expansion for Butterworth Lowpass *CPPF* Filter in z domain

3.2.1 Pole-parameter Representation [10]

In following section, we give a brief summary of the Pole-Parameter Representation and Complementary Pole -pair Filters. We will introduce the terminology, "pole-parameters". The line which joins any pole with the origin of the co-ordinates is referred to as the "pole-phasor". The magnitude ω_p and the angle θ_p of pole-phasor are shown in figure below, wherein the pole-parameters (ω_p,θ_p) corresponding to a negative real pole (s_r) and a complex conjugate pole-pair ($s, s*$) are defined. In addition, the pole-parameter ω_p is "pole-frequency", and the polar-angle θ_p is related to the pole-Q.

$$Q_p = \frac{1}{2\cos(\theta_p)}$$

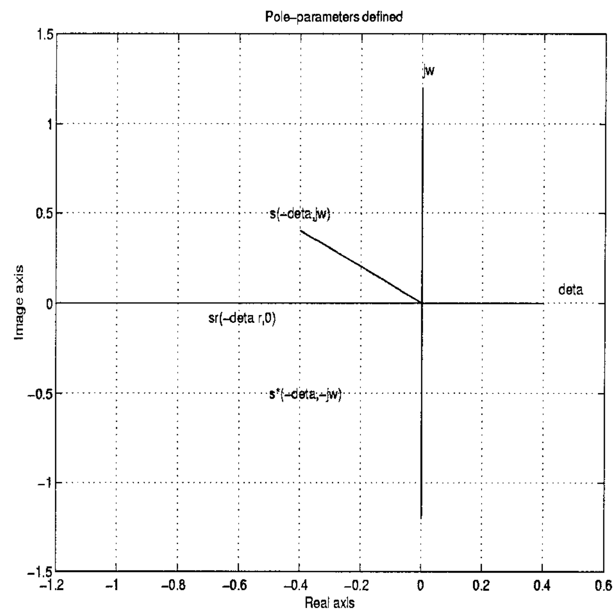

Figure 3.1: Pole Parameter Representation

3.2.2 Complementary Symmetry Among Butterworth Poles [10]

For a specified order Butterworth filter, the poles located uniformly on the left-half of s- plane, on second and third quadrant, on the unit circle, are symmetrical about the horizontal axis, and exhibit a mirror-image symmetry. Therefore all poles on the left-half plane of the s-plane can be expressed just by poles on the second-quadrant only. In addition, for a nth-order Butterworth filter, the pole-parameters corresponding to poles in the second-quadrant can be written as follows:

$$\omega_p = 1 \begin{cases} k = 1, 2, ...\frac{n}{2} & for \quad n \quad even \\ k = 0, 1, 2, ...\frac{n-1}{2} & for \quad n \quad odd \end{cases} \tag{3.2}$$

and

$$\theta_{pk} = \begin{cases} \frac{(2k-1)\pi}{2n} & k = 1, 2, ... \frac{n}{2}, \quad for \quad n \quad even \\ \frac{k\pi}{n} & k = 0, 1, 2, ... \frac{n-1}{2} \quad for \quad n \quad odd \end{cases} \tag{3.3}$$

The squared-magnitude of this Butterworth filter is

$A^2(\omega) = \frac{1}{1+\omega^{2n}}$

For the general case, all pole-phasors have magnitudes equal to ω_p (not necessarily unity), and the polar angles θ_{pk} are related at values given by the Equ(3.3). Furthermore, we normalize the d.c. values of $A(\omega)$ to unity at $\omega = 0$ (at DC), so that $A(\omega)$ at $\omega = 0$ is independent of ω_p; then, we can write the squared-magnitude of transfer function,

$A^2(\omega) = \frac{\omega_p^{2n}}{\omega_p^{2n}+\omega^{2n}}$

In general, for any odd or even order Butterworth filter, the poles exhibit half-plane symmetry properties with respect to both real and imaginary axis. Based on the fact that even order filters (order is 2^k, k is integers) exhibit additional symmetry properties, for example, $n = 16$, from Equ (3.3), we get

[$\theta_1 = \frac{\pi}{32}$ $\quad \theta_8 = \frac{15\pi}{32}$]

[$\theta_2 = \frac{3\pi}{32}$ $\quad \theta_7 = \frac{13\pi}{32}$]

[$\theta_3 = \frac{5\pi}{32}$ $\quad \theta_6 = \frac{11\pi}{32}$]

[$\theta_4 = \frac{7\pi}{32}$ $\quad \theta_5 = \frac{9\pi}{32}$]

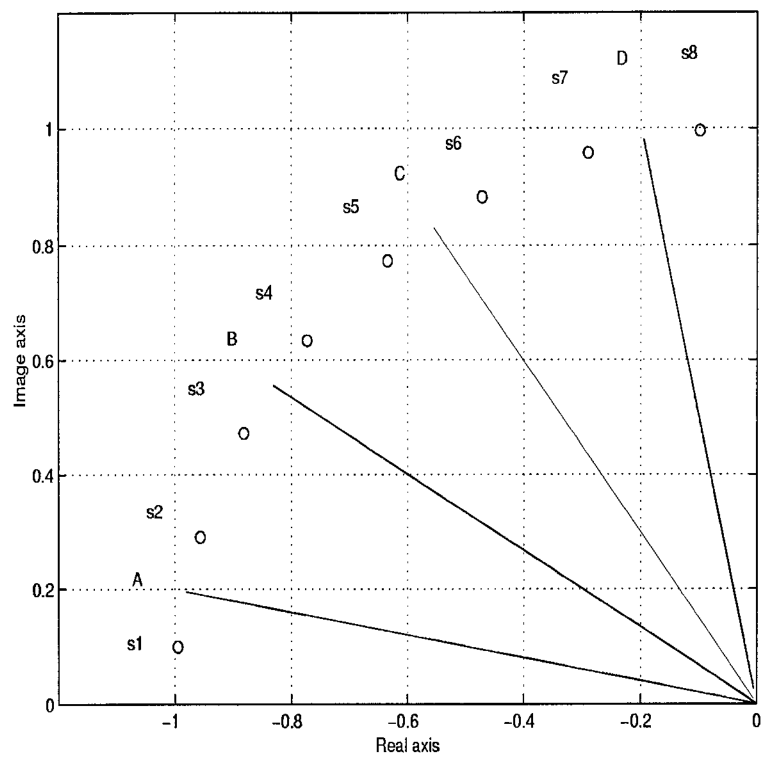

Figure 3.2: s-plane Pole-pattern of the Butterworth Filter of $n = 16$

We can easily find adjacent Butterworth poles in the second quadrant possess another symmetry - the adjacent pole-pairs (s_1, s_2), (s_3, s_4),(s_5, s_6),(s_7, s_8) are symmetrical about the lines OA, OB, OC and OD whose angles are $\frac{\pi}{16}, \frac{3\pi}{16}, \frac{5\pi}{16}, \frac{7\pi}{16}$ respectively, and each of the four symmetrical pairs adds up to $\frac{\pi}{2}$.

$(\theta_1 + \theta_8) = (\theta_2 + \theta_7) = (\theta_3 + \theta_6) = (\theta_4 + \theta_5) = \frac{\pi}{2}$

We designate such pole-pairs, whose θ_p values add up to $\frac{\pi}{2}$, as " Complementary Pole Pairs $(CPPs)$".

$(\theta_\Gamma + \theta_{(\frac{n}{2}-\Gamma+1)}) = \frac{\pi}{2}$, $\Gamma = 1,2,...,\frac{n}{4}$

3.2.3 Modification of Butterworth Pole-pattern Preserving the $CPPF$ Properties [10]

If we swing the pole-vectors of each adjacent pole-pair about Butterworth filters by equal angle θ_0 toward the respective axes of symmetry, the new set of polar angles will be

$$[\theta_1 = \tfrac{\pi}{32} + \theta_0 \quad \theta_8 = \tfrac{15\pi}{32} - \theta_0]$$
$$[\theta_2 = \tfrac{3\pi}{32} - \theta_0 \quad \theta_7 = \tfrac{13\pi}{32} + \theta_0]$$
$$[\theta_3 = \tfrac{5\pi}{32} + \theta_0 \quad \theta_6 = \tfrac{11\pi}{32} - \theta_0]$$
$$[\theta_4 = \tfrac{7\pi}{32} - \theta_0 \quad \theta_5 = \tfrac{9\pi}{32} + \theta_0\ ]$$

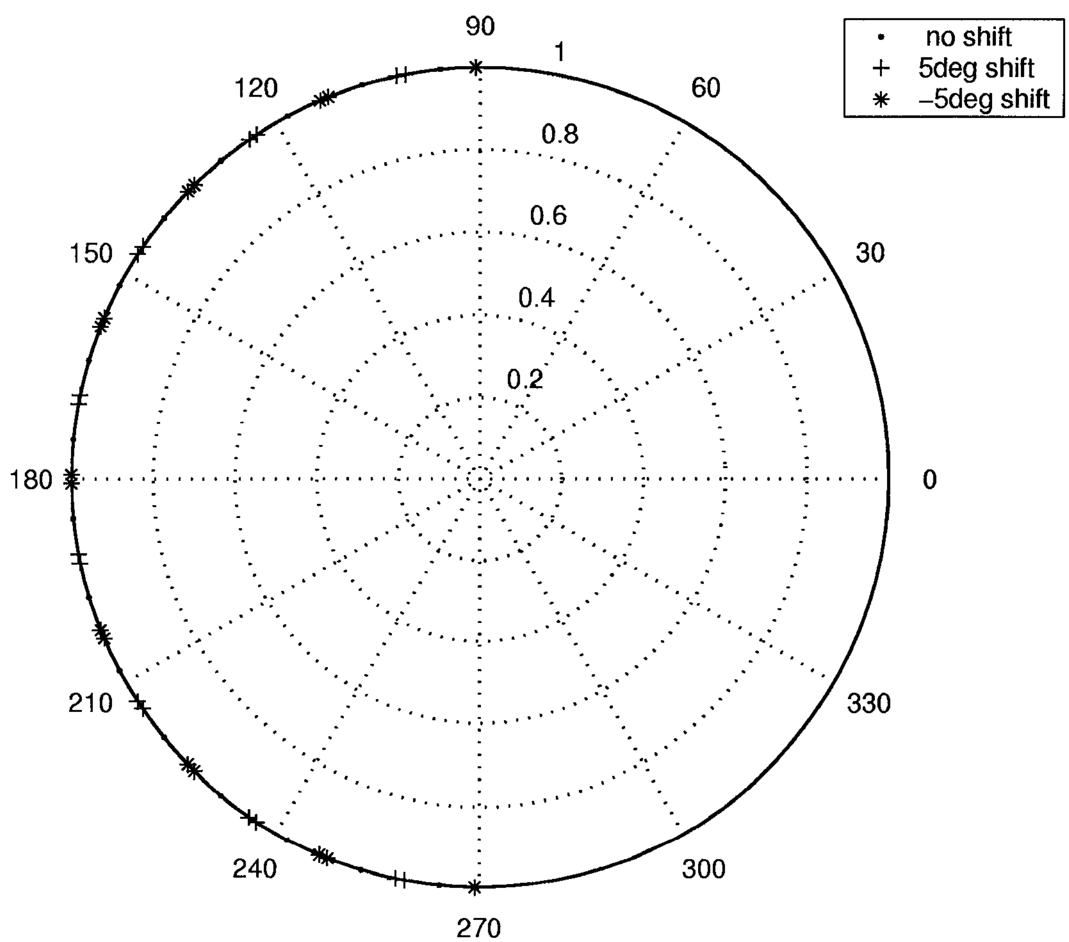

Figure 3.3: s-plane Pole-pattern of the $CPPF$ of $n = 16$

The reference pole-pattern of any specified order $n = 2^k$ Butterworth filter can be modified as:

$$\theta_k^{'} = \frac{(2k-1)\pi}{2n} + (-1)^{k-1}\theta_0, k = 1, 2, ...\frac{n}{2} \tag{3.4}$$

where $\theta_0 < |\frac{\pi}{2n}|$.

When $\theta_0 > 0$, modified transfer function is a low-Q filter (LQF), or $\theta_0 < 0$, it is a high-Q filter (HQF).

Then, we can see the second-order transfer function corresponding to a complex conjugate pole-pair with ω_{pk} as magnitude and $\pm\theta_{pk}$ as pole-parameters can be obtained as

$$T_k(s) = \frac{\omega_{pk}^2}{s^2 + 2\omega_{pk}\cos(\theta_{pk})s + \omega_{pk}^2} \tag{3.5}$$

Furthmore, it is easy to get a nth-order $T(s)$ transfer function,which is the product of all $T_k(s)$.

$$T(s) = \Pi T_k(s), k = 1, 2, ...\frac{n}{2} \tag{3.6}$$

The algorithm, called cppkakb.m, is employed to compute ka and kb coefficients of continued fraction expansions of CPP filters.

Algorithm 9: Matlab Function cppkakb.m

---

function [ka,kb] = cppkakb(n)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Continued fraction expansion for Low-Q Complementary Pole-Pair Filters

% (CPPF).

% n is the degree of CPPF in s domain

% $T_k = \frac{\omega L^2}{s^2+2\omega L\cos(\alpha k)s+\omega L^2}$, $\alpha = \theta_0 = 5^o$

% $D_d = s^2 + 2\omega L\cos(\alpha k)s + \omega L^2$, *k=1,2.... n/2,*

% d_0 *is the product of all terms of* D_k, *ka are coefficients of the continued*

% *fraction expansion (CFE) of* $F_1(z)/F_2(z)$, *where*$F_1(z)$ *is mirror-image*

% *polynomial,and* $F_2(z)$ *is antimirror-image polynomial of* d_0.

% $F_1(z) = \frac{[d_0(z)+z^n d_0(z^{-1})]}{2}$

% $F_2(z) = \frac{[d_0(z)-z^n d_0(z^{-1})]}{2}$

%

% $CFE1$*:*

% *For* n = *even or odd,*

% $\frac{F_1(z)}{F_2(z)} = ka(1)\frac{(z+1)}{(z-1)} + \frac{1}{ka(2)\frac{(z+1)}{(z-1)}+\frac{1}{ka(3)\frac{(z+1)}{(z-1)}+\frac{1}{......}}}$

% $CFE2$*:*

% *For* n = *even,*

% $\frac{F_{1e}(z)}{F_{2e}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)}+\frac{1}{kb(3)\frac{(z-1)}{(z+1)}+\frac{1}{......}}}$

% *Or if* n = *odd,*

% $\frac{F_{2o}(z)}{F_{1o}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)}+\frac{1}{kb(3)\frac{(z-1)}{(z+1)}+\frac{1}{......}}}$

%% *Convert* $F_1(z)/F_2(z)$ *or* $F_2(z)/F_1(z)$ *to continued fraction expansions* %%

```
format short;
wB=1;
a0= 5*pi/180;

% Get ωp of CPP
for j=1:n/2,
  a (j) = (2*j-1)*pi/(2*n) + (-1)^(j-1)*a0;
  lowq_wLN=[1 2*wB^(2*n)*cos(n*a(1)) -wB^(2*n)];
  rootlowq_wLN=roots(lowq_wLN);
```

```
  ind = find(rootlowq_wLN >= 0);
  rootlowq_wLN=rootlowq_wLN(ind);
  wL = (rootlowq_wLN).^(1/n);
end

d = zeros(n/2,3);
d0=1;

% To obtain denominator d0 of T
for j=1:n/2,
  d(j,:)= [1 2*wL*cos(a (j )) wL^2];
  d0= conv ( d0,d (j,:));
end

[ka,kb]=butterworthkakb (d0);
```

Continued fraction expansion for Low-Q Complementary Pole-Pair Filters ($LQFs$).

$T_k = \frac{\omega L^2}{s^2+2\omega L\cos(\alpha k)s+\omega L^2}$, $\alpha = \theta_0 = 5^o$

$D_k = s^2 + 2\omega L\cos(\alpha k)s + \omega L^2$, k=1,2.... $n/2$,

d_0 is the product of all terms of D_k, ,$F_1(z)$, $F_2(z)$ are mirror-image polynomial and antimirror-image polynomial of $d0$.

Table 3.7: The Denominator Polynomials of Butterworth $CPPFs$ in s- domain $(\theta_0 = 5^o)$

n	$d0 = s^n + a_{n-1}s^{n-1} + ... + a_1 s + a_0$
2	$s^2 + 1.4016s + 1.1886$
4	$s^4 + 2.9337s^3 + 4.3032s^2 + 3.4698s + 1.3989$
8	$s^8 + 5.6034s^7 + 15.6992s^6 + 28.4658s^5 + 36.2733s^4 + 33.1153s^3 + 21.2465s^2 + 8.8220s + 1.8316$

Remark: The numerator of Butterworth $CPPFs$ in s domain is same as the a_0.

The following table 3.8 gives the z - domain denominator polynomials of $CPPFs$ ($\theta_0 = 5^o$) .

Table 3.8: The Denominator Polynomials of Butterworth Lowpass $CPPFs$ in z-domain ($\theta_0 = 5^o$)

n	$D_{CH}(z)$
2	$3.0205z^2 + 0.3174z + 0.6621$
4	$9.3685z^4 + 1.9071z^3 + 4.1368z^2 + 0.3741z + 0.2135$
8	$83.0206z^8 + 31.3681z^7 + 87.7766z^6 + 22.6811z^5 + 25.3290z^4 + 3.9388z^3 + 1.7355z^2 + 0.1262z + 0.0241$

The following table 3.9 gives the mirror - image and anti - mirror - image polynomials of the z - domain polynomial of the $CPPFs$ ($\theta_0 = 5^o$) .

Table 3.9: The Mirror - image and Anti - mirror - image Polynomials of the z - domain Butterworth Denominator Polynomials Given in Table 3.8 ($\theta_0 = 5^o$)

	$F_{ccp1}(z)$ and $F_{cpp2}(z)$
2	$F_{cpp1}(z) = 1.8413z^2 + 0.3174z + 1.8413$
	$F_{cpp2}(z) = 1.1792z^2 - 1.1792$
4	$F_{cpp1}(z) = 4.7910z^4 + 1.1406z^3 + 4.1368z^2 + 1.1406z + 4.7910$
	$F_{cpp2}(z) = 4.5775z^4 + 0.7665z^3 - 0.7665z - 4.5775$
8	$F_{cpp1}(z) = 41.5214z^8 + 15.7468z^7 + 44.7551z^6 + 13.3096z^5 + 25.3284z^4 + 13.3096z^3 + 44.7551z^2 + 15.7468z + 41.5214$
	$F_{cpp2}(z) = 41.4973z^8 + 15.6206z^7 + 43.0196z^6 + 9.3709z^5 - 9.3709z^3 - 43.0196z^2 - 15.6206z - 41.4973$

$ka's$ are coefficients of the continued-fractionexpansion (CFE) of $F_1(z)/F_2(z)$, where $F_1(z)$ is mirror-image polynomial,and $F_2(z)$ is antimirror-image polynomial of d_0.

Table 3.10: The values of $CFE1$ for $CPPF$ Filters ($\theta_0 = 5^o$) - $ka's$

n	$ka(1)$	$ka(2)$	$ka(3)$	$ka(4)$	$ka(5)$	$ka(6)$	$ka(7)$	$ka(8)$
2	0.8481	1.4016						
4	0.4032	1.1120	1.7129	1.8217				
8	0.2076	0.6139	0.9928	1.3206	1.5914	1.8400	2.0469	1.8290

$kb's$ are coefficients of the continued-fraction expansion (CFE) of $F_1(z)/F_2(z)$ or $F_2(z)/F_1(z)$.

Table 3.11: The values of $CFE2$ for $CPPF$ Filters ($\theta_0 = 5^o$) - $kb's$

n	$kb(1)$	$kb(2)$	$kb(3)$	$kb(4)$	$kb(5)$	$kb(6)$	$kb(7)$	$kb(8)$
2	0.7135	1.1792						
4	0.3409	0.9401	1.4483	1.5402				
8	0.1785	0.5277	0.8534	1.1351	1.3679	1.5817	1.7595	1.5722

Using matlab function cfekeo.m, we obtain the values of $kf's$ and $kz's$ in $CFE3$.

Table 3.12: The values of $CFE3$ for $CPPF$ Filters ($\theta_0 = 5^o$) - $kf's$ and $kz's$

n	$kf(1)$	$kz(1)$	$kf(2)$	$kz(2)$	$kf(3)$	$kz(3)$	$kf(4)$	$kz(4)$
2	0.7134	0.8480						
4	0.3409	0.4032	1.5140	1.7907				
8	0.1785	0.2076	0.5926	0.6894	1.2682	1.4754	3.0394	3.5359

The algorithm, called cppkc.m, is employed to compute kc and γ coefficients of continued fraction expansions of $CPPF$ filters.

Algorithm 10: Matlab Function cppkc.m

---

function [k0,kc,r] = cppkc(n,rootsele)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Continued fraction expansion for Low-Q Complementary Pole-Pair Filters

% n is the degree of CPPF in s domain

% $T_k = \frac{\omega L^2}{s^2+2\omega L\cos(\alpha k)s+\omega L^2}$, $\alpha = \theta_0 = 5^o$

% $D_d = s^2 + 2\omega L\cos(\alpha k)s + \omega L^2$, *k=1,2.... n/2,*

% d_0 is the product of all terms of D_k, ka are coefficients of the continued

% fraction expansion (CFE) of $F_1(z)/F_2(z)$, where$F_1(z)$ is mirror-image

% polynomial,and $F_2(z)$ is antimirror-image polynomial of d_0.

% $F_1(z) = \frac{[d_0(z)+z^n d_0(z^{-1})]}{2}$

% $F_2(z) = \frac{[d_0(z)-z^n d_0(z^{-1})]}{2}$

% CFE3:

% For n = odd,

% $\frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\frac{(z-1)}{(z+1)} + kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{\cdots\cdots}}}$

```
% CFE3:
% For n = even,kc(0)
```

$$\% \ \frac{F_{2e}(z)}{F_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \frac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)}+\frac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)}+\frac{1}{\cdots\cdots}}}$$

```
%% Convert F2(z)/F1(z) to continued fraction expansions %%
format short;
wB=1;
a0= 5*pi/180;

% Get ωp of CPP
for j=1:n/2,
  a (j) = (2*j-1)*pi/(2*n) + (-1)^(j-1)*a0;
  lowq_wLN=[1 2*wB^(2*n)*cos(n*a(1)) -wB^(2*n)];
  rootlowq_wLN=roots(lowq_wLN);
  ind = find(rootlowq_wLN >= 0);
  rootlowq_wLN=rootlowq_wLN(ind);
  wL = (rootlowq_wLN).^(1/n);
end

d = zeros(n/2,3);
d0=1;

% To obtain denominator d0 of T
for j=1:n/2,
  d(j,:)= [1 2*wL*cos(a (j )) wL^2];
  d0= conv ( d0,d (j,:)); % d0 is the CPPF in s domain
end
[k0,kc,r]=butterworthkc (d0,rootsele);
```

The following table 3.13 gives the values of $kc's$ and $\gamma's$ coefficients of the continued-fraction expansion $CPPF$ Filters ($\theta_0 = 5^o$).

$kc's$ and $\gamma's$ are coefficients of the continued-fraction expansion (CFE) of $F_2(z)/F_1(z)$.

Table 3.13: The values of $CFE4$ for $CPPF$ Filters ($\theta_0 = 5^o$) - $kc's$ and $\gamma's$

n	$kc(1)$	$kc(2)$	$kc(3)$	$kc(4)$		$r(1)$	$r(2)$	$r(3)$	$r(4)$
2	0.6404					0.1724			
4	0.4993	0.4562				-0.9537	1.1917		
	0.4562	0.4993				1.1917	-0.9537		
8	0.3622	0.2234	0.6376	0.1053		-1.6239	1.0294	1.4911	-0.1925
	0.3622	0.2234	0.1053	0.6376		-1.6239	1.0294	-0.1925	1.4911
	0.3622	0.1161	0.3502	0.3379		-1.6239	0.0462	0.3450	1.6898
	0.3622	0.1161	0.3379	0.3502		-1.6239	0.0462	1.6898	0.3450
	0.3192	0.2393	0.7098	0.1027		1.7151	-0.7901	-1.3409	0.4858
	0.3192	0.2393	0.1027	0.7098		1.7151	-0.7901	0.4858	-1.3409
	0.3192	0.1152	0.3826	0.3554		1.7151	0.2550	-1.5914	-0.0459
	0.3192	0.1152	0.3554	0.3826		1.7151	0.2550	-0.0459	-1.5914
	0.1616	0.2330	0.6500	0.3910		-0.2847	1.0386	1.4675	-1.5761
	0.1616	0.2330	0.3910	0.6500		-0.2847	1.0386	-1.5761	1.4675
	0.1616	0.5474	1.1873	0.3603		-0.2847	-0.4329	-0.8244	1.6039
	0.1616	0.5474	0.3603	1.1873		-0.2847	-0.4329	1.6039	-0.8244
	0.1564	0.5241	1.1054	0.4050		0.5728	0.7101	1.0592	-1.4821
	0.1564	0.5241	0.4050	1.1054		0.5728	0.7101	-1.4821	1.0592
	0.1564	0.2498	0.7222	0.3457		0.5728	-0.8007	-1.3118	1.6779
	0.1564	0.2498	0.3457	0.7222		0.5728	-0.8007	1.6779	-1.3118

3.3 Summary and Discussion

This chapter gives the algorithms for obtaining the coefficients of the different continued fraction expansions for the z - domain denominator polynomials corresponding to Butterworth lowpass filters and complementary pole - pair lowpass filters.

Chapter 4

The Application of the Continued Fraction Expansion for Chebyshev Lowpass Filter in z Domain

Chebyshev lowpass filter approximations are based on Laplace domain. By applying the bilinear transformation $s = \frac{z-1}{z+1}$, the transfer functions in discrete domain are characterized.

The purpose of this chapter is to establish an algorithm to obtain the coefficients ka's , kb's and kc's and $\gamma's$ of continued fraction expansion (CFE) using the concept of mirror-image polynomial and anti-mirror-image polynomial for Chebyshev lowpass filters in z domain after bilinear transformation.

4.1 The Application of the Continued Fraction Expansion for Chebyshev Lowpass Filter in z domain

First we take the nth-order Chebyshev lowpass filter polynomial, and apply bilinear transformation $s = \frac{z-1}{z+1}$, then we will get lowpass transfer functions in discrete domain $T_{CHd}(z) = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_{CHd}(z)}$. For each order of $D_{CHd}(z)$, we can produce the mirror-image and anti-mirror-image polynomials.

Each order of $D_{CHd}(z)$, the denominator polynomial of Chebyshev lowpass filters in the z - domain, is obtained which is listed in Table 4.1 up to order $n = 6$. For these polynomials, the mirror-image polynomial $F_{CH1}(z)$ and the anti-mirror-image polynomials $F_{CH2}(z)$ are obtained and are listed in Table 4.2.

Table 4.1: The Denominator Polynomials of Chebyshev Lowpass Filters in the z - domain

passaband ripple A_p	n	$D_{CHD}(z)$
$A_p = 0.5dB$ $\varepsilon = 0.3493$	1	$1.3493z + 0.6507$
	2	$2.7539z^2 + 0.7213z + 0.7619$
	3	$6.2925z^3 - 0.7977z^2 + 3.2965z - 0.7912$
	4	$14.8632z^4 - 7.9019z^3 + 13.5269z^2 - 5.9800z + 2.4398$
	5	$35.4949z^5 - 33.4953z^4 + 51.1133z^3 - 34.1779z^2 + 18.8155z - 5.7505$
	6	$85.1736z^6 - 115.5648z^5 + 181.8225z^4 - 157.6631z^3 + 110.5424z^2 - 50.5708z + 14.0525$
$A_p = 1.0dB$ $\varepsilon = 0.5089$	1	$1.5088z + 0.4912$
	2	$3.2569z^2 + 0.2086z + 1.0226$
	3	$7.5677z^3 - 2.5972z^2 + 4.5739z - 1.5444$
	4	$18.0131z^4 - 13.5063z^3 + 19.3196z^2 - 10.0837z + 4.2097$
	5	$43.1888z^5 - 50.2440z^4 + 72.7789z^3 - 54.0022z^2 + 30.3400z - 10.0597$
	6	$103.8300z^6 - 163.7136z^5 + 256.1931z^4 - 238.6443z^3 + 172.4902z^2 - 82.7969z + 24.4508$
$A_p = 2.0dB$ $\varepsilon = 0.7648$	1	$1.7648z + 0.2352$
	2	$4.0180z^2 - 0.5413z + 1.5590$
	3	$9.4432z^3 - 5.3075z^2 + 6.7933z - 2.9291$
	4	$22.6086z^4 - 21.8775z^3 + 28.8882z^2 - 16.9972z + 7.5208$
	5	$54.3370z^5 - 75.1175z^4 + 107.2263z^3 - 86.5908z^2 + 50.2212z - 18.0762$
	6	$131.8641z^6 - 235.0385z^5 + 372.2532z^4 - 368.8968z^3 + 276.4932z^2 - 138.9171z + 43.8129$

Table 4.2: The Mirror - image and Anti - mirror - image Polynomials of Denominators of the z - domain Chebyshev Lowpass Polynomials Given in Table 4.1

A_p	n	$F_{CH1}(z)$ and $F_{CH2}(z)$
	1	$F_{CH1}(z) = 1.0000z + 1.0000$
		$F_{CH2}(z) = 0.3493z - 0.3493$
	2	$F_{CH1}(z) = 1.7579z^2 + 0.7213z + 1.7579$
		$F_{CH2}(z) = 0.9960z^2 - 0.9960$
	3	$F_{CH1}(z) = 2.7506z^3 + 1.2494z^2 + 1.2494z + 2.7506$
$A_p = 0.5dB$		$F_{CH2}(z) = 3.5419z^3 - 2.0471z^2 + 2.0471z - 3.5419$
$\varepsilon = 0.3493$	4	$F_{CH1}(z) = 8.6515z^4 - 6.9409z^3 + 13.5269z^2 - 6.9409z + 8.6515$
		$F_{CH2}(z) = 6.2117z^4 - 0.9609z^3 + 0.9609z - 6.2117$
	5	$F_{CH1}(z) = 14.8722z^5 - 7.3399z^4 + 8.4677z^3 + 8.4677z^2 - 7.3399z + 14.8722$
		$F_{CH2}(z) = 20.6227z^5 - 26.1554z^4 + 42.6456z^3 - 42.6456z^2 + 26.1554z - 20.6227$
	6	$F_{CH1}(z) = 49.6130z^6 - 83.0678z^5 + 146.1825z^4 - 157.6631z^3 + 146.1825z^2 - 83.0678z + 49.6130$
		$F_{CH2}(z) = 35.5605z^6 - 32.4970z^5 + 35.6401z^4 - 35.6401z^2 + 32.4970z - 35.5605$

A_p	n	$F_{CH1}(z)$ and $F_{CH2}(z)$
$A_p = 1.0dB$ $\varepsilon = 0.5089$	1	$F_{CH1}(z) = 1.0000z + 1.0000$ $F_{CH2}(Z) = 0.5088z - 0.5088$
	2	$F_{CH1}(Z) = 2.1397z^2 + 0.2086z + 2.1397$ $F_{CH2}(Z) = 1.1172z^2 - 1.1172$
	3	$F_{CH1}(Z) = 3.0117z^3 + 0.9883z^2 + 0.9883z + 3.0117$ $F_{CH2}(Z) = 4.5560z^3 - 3.5855z^2 + 3.5855z - 4.5560$
	4	$F_{CH1}(Z) = 11.1114z^4 - 11.7950z^3 + 19.3196z^2 - 11.7950z + 11.1114$ $F_{CH2}(Z) = 6.9017z^4 - 1.7113z^3 + 1.7113z - 6.9017$
	5	$F_{CH1}(Z) = 16.5603z^5 - 9.9484z^4 + 9.3881z^3 + 9.3881z^2 - 9.9484z + 16.5603$ $F_{CH2}(Z) = 26.6176z^5 - 40.2780z^4 + 63.3693z^3 - 63.3693z^2 + 40.2780z - 26.6176$
	6	$F_{CH1}(Z) = 64.1404z^6 - 123.2553z^5 + 214.3416z^4 - 238.6443z^3 + 214.3416z^2 - 123.2553z + 64.1404$ $F_{CH2}(Z) = 39.6896z^6 - 40.4583z^5 + 41.8514z^4 - 41.8514z^2 + 40.4583z - 39.6896$

A_p	n	$F_{CH1}(z)$ and $F_{CH2}(z)$
$A_p = 2.0dB$ $\varepsilon = 0.7648$	1	$F_{CH1}(z) = 1.0000z + 1.0000$ $F_{CH2}(Z) = 0.7648z - 0.7648$
	2	$F_{CH1}(Z) = 2.7885z^2 - 0.5413z + 2.7885$ $F_{CH2}(Z) = 1.2295z^2 - 1.2295$
	3	$F_{CH1}(Z) = 3.2571z^3 + 0.7429z^2 + 0.7429z + 3.2571$ $F_{CH2}(Z) = 6.1861z^3 - 6.0504z^2 + 6.0504z - 6.1861$
	4	$F_{CH1}(z) = 15.0647z^4 - 19.4374z^3 + 28.8882z^2 - 19.4374z + 15.0647$ $F_{CH2}(Z) = 7.5439z^4 - 2.4402z^3 + 2.4402z - 7.5439$
	5	$F_{CH1}(Z) = 18.1304z^5 - 12.4481z^4 + 10.3177z^3 + 10.3177z^2 - 12.4481z + 18.1304$ $F_{CH2}(Z) = 36.2066z^5 - 62.6693z^4 + 96.9086z^3 - 96.9086z^2 + 62.6693z - 36.2066$
	6	$F_{CH1}(Z) = 87.3385z^6 - 186.9778z^5 + 324.3732z^4 - 368.8968z^3 + 324.3732z^2 - 186.9778z + 87.3385$ $F_{CH2}(Z) = 43.5256z^6 - 48.0607z^5 + 47.8800z^4 - 47.8800z^2 + 48.0607z - 43.5256$

The following algorithm, called chebyshevkakb.m, is employed to obtain ka and kb coefficients of continued fraction expansions for a nth-order denominator polynomial of Chebyshev lowpass filter.

Algorithm 11: Matlab Function chebyshevkakb.m

---

function [ka, kb] = chebyshevkakb(d)

%%%

% *d is a denominator polynomial of Chebyshev filter in s domain, apply*

% *bilinear transformation* $s = \frac{z-1}{z+1}$, *we get* $T_{CHd}(z) = k_{CH}\frac{N_{CHd}(z)}{D_{CHd}(z)} = \frac{(z+1)^n}{D_{CHd}(z)}$

% $D_d(z)$ *is a denominator polynomial of Chebyshev filter in z domain.*

% *Find coefficients ka and kb of the continued-fractionexpansion (CFE) of*

% $F_1(z)/F_2(z)$ *or* $F_2(z)/F_1(z)$, *where* $F_1(z)$ *is mirror-image polynomial,*

% *and* $F_2(z)$ *is antimirror-image polynomial of* $D_d(z)$.

% $F_1(z) = \frac{[D_d(z)+z^n D_d(z^{-1})]}{2}$

% $F_2(z) = \frac{[D_d(z)-z^n D_d(z^{-1})]}{2}$

%

% *CFE1:*

% *For n = even or odd,*

% $\frac{F_1(z)}{F_2(z)} = ka(1)\frac{(z+1)}{(z-1)} + \cfrac{1}{ka(2)\frac{(z+1)}{(z-1)} + \cfrac{1}{ka(3)\frac{(z+1)}{(z-1)} + \frac{1}{\ldots\ldots}}}$

% *CFE2:*

% *For n = even,*

% $\frac{F_{1e}(z)}{F_{2e}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \cfrac{1}{kb(2)\frac{(z-1)}{(z+1)} + \cfrac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\ldots\ldots}}}$

% *Or if n = odd,*

% $\frac{F_{2o}(z)}{F_{1o}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \cfrac{1}{kb(2)\frac{(z-1)}{(z+1)} + \cfrac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\ldots\ldots}}}$

% *d is coefficients of polynomial D respectively in descending powers of z.*

%% *Convert* $F_1(z)/F_2(z)$ *or* $F_2(z)/F_1(z)$ *to continued fraction expansions* %%

```
% d(1) must be non-negative.
if( d(1) <= 0 )
    error('First coefficient in d vector must be non- negative.')
    return
end

% The coefficient of d vector must be real value
for ii = 1:length(d)
  if( any(imag(d(ii))))
    error('The coefficient of d vector must be real value.')
    return
  end
end

% Get ka and kb coefficients of the continued-fraction expansion
[Nd,Dd]=bilinear(1,d,0.5);
denpoly=Dd/Nd(1);
[ka,kb]=cfekakb(denpoly);
```

The following table 4.3 gives the coefficients $ka's$ of $CFE1$ for denominator polynomials of Chebyshev lowpass filters.

d is a vector which represents the coefficients of a denominator polynomial of Chebyshev filter respectively in descending powers of s domain, in the form $s^n + a_1 s^{n-1} + a_2 s^{n-2} + ... + a_{n-2}s^2 + a_{n-1}s + a_n$, for Chebyshev filter function of order n, with pass-band from 0 to 1 rad/sec. By applying bilinear transformation $s = \frac{z-1}{z+1}$,we get $T_d = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_d(z)}$, $D_d(z)$ is a denominator polynomial of Chebyshev lowpass filter in z domain. ka are coefficients of the continued-fractionexpansion (CFE) of $F_1(z)/F_2(z)$, where $F_1(z)$ is mirror-image polynomial,and $F_2(z)$ is antimirror-image polynomial of $D_d(z)$.

Table 4.3: The Values of $ka's$ in $CFE1$ for Denominator Polynomials of Chebyshev Lowpass Filters

passaband ripple A_p	n	$Ka(1)$	$ka(2)$	$ka(3)$	$ka(4)$	$ka(5)$	$ka(6)$
$A_p = 0.5dB$ $\varepsilon = 0.3493$	1	2.8628					
	2	1.0635	1.4256				
	3	0.4663	1.9512	0.7866			
	4	0.3696	0.8047	3.2454	0.3926		
	5	0.2378	0.8864	0.7656	6.5573	0.1691	
	6	0.2192	0.5251	1.4130	0.5703	16.0217	0.0638
$A_p = 1.0dB$ $\varepsilon = 0.5089$	1	1.9652					
	2	1.0044	1.0977				
	3	0.3967	2.0933	0.5916			
	4	0.3712	0.6749	3.9696	0.2779		
	5	0.2116	0.9408	0.6131	8.9738	0.1122	
	6	0.2244	0.4586	1.6245	0.4295	23.9259	0.0401
$A_p = 2.0dB$ $\varepsilon = 0.7648$	1	1.3076					
	2	1.0239	0.8038				
	3	0.3198	2.4453	0.4180			
	4	0.3982	0.5321	5.2744	0.1842		
	5	0.1779	1.0765	0.4585	13.2920	0.0700	
	6	0.2446	0.3759	2.0325	0.3014	38.1753	0.0239

The following table 4.4 gives the coefficients $kb's$ of $CFE2$ for denominator polynomials of Chebyshev lowpass filters.

$kb's$ are coefficients of the continued-fraction expansion (CFE) of $F_1(z)/F_2(z)$ or $F_2(z)/F_1(z)$.

Table 4.4: The Values of $kb's$ of $CFE2$ for Denominator Polynomials of Chebyshev Lowpass Filters

passaband ripple A_p	n	$Kb(1)$	$kb(2)$	$kb(3)$	$kb(4)$	$kb(5)$	$kb(6)$
$A_p = 0.5dB$ $\varepsilon = 0.3493$	1	0.3493					
	2	0.7014	0.9403				
	3	0.7981	1.3001	1.3465			
	4	0.8352	1.3916	1.7279	1.3138		
	5	0.8529	1.4291	1.8142	1.6426	1.5388	
	6	0.8627	1.4483	1.8494	1.7101	1.9018	1.4042
$A_p = 1.0dB$ $\varepsilon = 0.5089$	1	0.5088					
	2	0.9110	0.9957				
	3	1.0118	1.3332	1.5088			
	4	1.0495	1.4126	1.9093	1.2817		
	5	1.0674	1.4441	1.9938	1.5908	1.6652	
	6	1.0773	1.4601	2.0270	1.6507	2.0491	1.3457
$A_p = 2.0dB$ $\varepsilon = 0.7648$	1	0.7648					
	2	1.2441	0.9766				
	3	1.3553	1.2740	1.7717			
	4	1.3962	1.3389	2.2169	1.1727		
	5	1.4155	1.3640	2.3049	1.4468	1.9004	
	6	1.4261	1.3765	2.3383	1.4974	2.3304	1.2137

Using matlab function cfekeo.m, we obtain the values of $kf's$ and $kz's$ in $CFE3$. The following table gives the values of $kf's$ and $kz's$ in $CFE3$.

Table 4.5: The Values of $kf's$ and $kz's$ in $CFE3$ for Denominator Polynomials of Chebyshev Lowpass Filters

passaband ripple A_p	n	$Kf(1)$	$kz(1)$	$kf(2)$	$kz(2)$	$kf(3)$	$kz(3)$
$A_p = 0.5dB$ $\varepsilon = 0.3493$	1	0	2.8628				
	2	0.7014	1.0635				
	3	0	0.4663	1.2712	1.9512		
	4	0.8352	0.3696	2.8656	2.4541		
	5	0	0.2378	1.0698	0.8864	4.6600	4.2323
	6	0.8627	0.2192	2.1218	0.9599	4.9832	4.1085
$A_p = 1.0dB$ $\varepsilon = 0.5089$	1	0	1.9652				
	2	0.9110	1.0044				
	3	0	0.3967	1.6903	2.0933		
	4	1.0495	0.3712	2.9693	2.3143		
	5	0	0.2116	1.3788	0.9408	4.6590	3.9642
	6	1.0773	0.2244	2.1715	0.9064	5.4102	4.2876
$A_p = 2.0dB$ $\varepsilon = 0.7648$	1	0	1.3076				
	2	1.2441	1.0239				
	3	0	0.3198	2.3922	2.4453		
	4	1.3962	0.3982	2.8677	2.0693		
	5	0	0.1779	1.8920	1.0765	4.2884	3.4620
	6	1.4261	0.2446	2.0754	0.8103	6.1786	4.7452

The M-file chebyshevkc.m, is employed to compute kc or $k0$ and γ coefficients of continued fraction expansions for the denominator polynomial of Chebyshev Lowpass Filter.

Algorithm 12: Matlab Function chebyshevkc.m

```
function [k0, kc, r] = chebyshevkc(d,rootsele)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% d is a denominator polynomial of Chebyshev filter in s domain, apply
% bilinear transformation s = (z-1)/(z+1), we get Td = Nd(z)/Dd(z) = (z+1)^n/Dd(z),
% Dd(z) is a denominator polynomial of Chebyshev filter in z domain.
% Find coefficients kc/r of the continued-fraction expansion of F1(z)/F2(z)
% or F2(z)/F1(z), where F1(z) is mirror-image polynomial, and F2(z) is
% antimirror-image polynomial of Dd(z).
% F1(z) = [Dd(z)+z^n Dd(z^-1)]/2
% F2(z) = [Dd(z)-z^n Dd(z^-1)]/2
% CFE4:
% For n = odd,
% F2o(z)/F1o(z) = kc(0)(z-1)/(z+1) + kc(1)(z^2-1)/(z^2+γ1 z+1) + 1/(kc(2)(z^2-1)/(z^2+γ2 z+1) + 1/(kc(3)(z^2-1)/(z^2+γ3 z+1) + 1/......))
% CFE4:
% For n = even,kc(0)
% F2e(z)/F1e(z) = kc(1)(z^2-1)/(z^2+γ1 z+1) + 1/(kc(2)(z^2-1)/(z^2+γ2 z+1) + 1/(kc(3)(z^2-1)/(z^2+γ3 z+1) + 1/......))
%% Convert F1(z)/F2(z) or F2(z)/F1(z) to continued fraction expansions %%

% d(1) must be non-negative.
if( d(1) <= 0 )
    error('First coefficient in d vector must be non- negative.')
    return
end
```

```
% The coefficient of d vector must be real value
for ii = 1:length(d)
  if( any(imag(d(ii))))
     error('The coefficient of d vector must be real value.')
     return
  end
end

% Get ka and kb coefficients of the continued-fraction expansion
[Nd,Dd]=bilinear(1,d,0.5);
denpoly=Dd/Nd(1);
[k0, kc, r]=cfekc(denpoly,rootsele);
```

The following table 4.6 gives the coefficients $kc's$ and $\gamma's$ of $CFE4$ for denominator polynomials of Chebyshev lowpass filters.

$kc's$ and $\gamma's$ are coefficients of the continued-fraction expansion (CFE) of $F_2(z)/F_1(z)$.

Table 4.6: The Values of $kc's$ and $\gamma's$ of $CFE4$ for Denominator Polynomials of Chebyshev Lowpass Filters

passband ripple A_p	n	$kc(0)$	$kc(1)$	$kc(2)$	$kc(3)$		$\gamma(1)$	$\gamma(2)$	$\gamma(3)$
	1	0.3493							
	2		0.5666				0.4103		
	3	0.7981	0.4895				-0.5458		
	4		0.4735	0.2445			-1.1740	0.3718	
$A_p = 0.5dB$			0.2445	0.4735			0.3718	-1.1740	
$\varepsilon = 0.3493$	5	0.8529	0.4324	0.1014			-1.4502	-0.0434	
		0.8529	0.1014	0.4324			-0.0434	-1.4502	
	6		0.3382	0.0676	0.3885		-1.6437	-0.0590	0.0276
			0.2149	0.2185	0.5636		0.3074	-0.7638	-1.4619
			0.1637	0.4307	0.7261		-0.3380	-0.4508	-1.3033

passband ripple A_p	n	$kc(0)$	$kc(1)$	$kc(2)$	$kc(3)$		$\gamma(1)$	$\gamma(2)$	$\gamma(3)$
	1	0.5088							
	2		0.5221				0.0975		
	3	1.0118	0.5010				-0.6718		
	4		0.4298	0.1914			-1.2676	0.2061	
$A_p = 1.0dB$			0.1914	0.4298			0.2061	-1.2676	
$\varepsilon = 0.5089$	5	1.0674	0.4416	0.0982			-1.4881	-0.1127	
		1.0674	0.0982	0.4416			-0.1127	-1.4881	
	6		0.3076	0.0712	0.3183		-1.6737	-0.1080	-0.1456
			0.1472	0.2309	0.5292		0.1738	-0.8571	-1.4943
			0.1640	0.4301	0.5654		-0.4217	-0.4244	-1.4403

passband ripple A_p	n	$kc(0)$	$kc(1)$	$kc(2)$	$kc(3)$		$\gamma(1)$	$\gamma(2)$	$\gamma(3)$
	1	0.7648							
	2		0.4409				-0.1941		
	3	1.3553	0.5439				-0.7719		
	4		0.3644	0.1363			-1.3512	0.0610	
$A_p = 2.0dB$			0.1363	0.3644			0.0610	-1.3512	
$\varepsilon = 0.7648$	5	1.4155	0.4798	0.1017			-1.5182	-0.1684	
		1.4155	0.1017	0.4798			-0.1684	-1.5182	
	6		0.2632	0.0824	0.2398		-1.7011	-0.1471	-0.3013
			0.0877	0.2557	0.4588		0.0672	-0.9355	-1.5316
			0.1474	0.4328	0.4139		-0.5069	-0.3749	-1.5517

4.2 Summary and Discussion

This chapter gives the algorithms for obtaining the coefficients of the different continued fraction expansions CFE for the z - domain denominator polynomials corresponding to Chebyshev lowpass filters.

Chapter 5

The Generation of the Transfer Functions

The previous chapters dealt with the determination of the coefficients of continued fraction expansions using the concept of mirror-image polynomial and anti-mirror-image polynomial for Butterworth lowpass filter, Butterworth lowpass $CPPF$ filter and Chebyshev lowpass filter approximations in z domain after bilinear transformation.

It must be noted that the positivity of $kc's$ will not be sufficient to prove that $D(z)$ contains all its zeros within the unit circle.However, when CFEs in this thesis are considered, $kc's$ are positive, and magnitudes of γ_i's are less than 2. From these, stable transfer functions $D(z)$ can be obtained. The purpose of this chapter is to establish an algorithm to obtain stable transfer functions of the filters in z domain from coefficients kc and γ of the continued fraction expansions ($CFE4$). For example, we would like to use coefficients $k0$, kc and γ of continued fraction expansions for Butterworth lowpass filters, Butterworth $CPPFs$ filter and Chebyshev lowpass filter approximations in z domain, which are obtained in chapter 3 and chapter 4, to get the transfer functions of the filters.

5.1 Generation of the Stable Transfer Functions from Coefficients kc and γ of Continued Fraction Expansion

5.1.1 Lowpass Filter

a)

Given kc (and k_0) and γ coefficients of a continued fraction expansion for a lowpass filter, we can easily represent the denominator $D_d(z)$ of transfer function in z domain whose zeros are satisfied inside the unit circle.

$$T(z) = \frac{N_d(z)}{D_d(z)} = \frac{(z+1)^n}{D_d(z)} \tag{5.1}$$

We can normalize Equ (5.1) to make the maximum of $T_{ld}(z)$ unity.

For a nth-order lowpass filter transfer function, we will get digital transfer function

$$T_{ld} = \frac{(z+1)^n}{D_{ld}(z)} \tag{5.2}$$

b)

The code dlowpoly.m used to design the denominators of lowpass filter transfer functions is given below. Then we select 3 frequencies randomly in passband to examine the magnitude values of lowpass filter transfer functions. The range of 3 frequencies are [0, 2×arctan(1)].

The M-file dlowpoly.m is employed to examine the magnitude values of lowpass filter transfer functions coming from $k0$, kc and γ coefficients of continued fraction expansions.

Algorithm 13: Matlab Function dlowpoly.m

```
function [wl, Maglow] = dlowpoly(k0,kc,r)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% d(z) = F1(z) + F2(z)
% F2o(z)/F1o(z) = kc(0) (z-1)/(z+1) + kc(1) (z^2-1)/(z^2+γ1 z+1) + 1/(kc(2) (z^2-1)/(z^2+γ2 z+1) + 1/(kc(3) (z^2-1)/(z^2+γ3 z+1) + 1/......))
% k0, kc are coefficients of the continued-fraction expansion for a
% denominator of lowpass
% r are coefficients of z^2 + γ(n)z + 1 respectively
% Maglow is the vector of magnitudes values at wl in [0, 2*arctan(1)]
% randomly
%%%%% Obtain a denominator polynomial from coefficients of CFE %%%%%

k=kc;
LK = length(k);
LR = length(r);
w = -pi:0.001:pi;
if (LK >=2)
  for ii= 1:length(k)
    if( k(ii) <= 0 )
      error('Coefficient in k vector must be positive.')
      return
    end
    if( imag(k(ii)) ~= 0 )
      error('The coefficient in k vector must be real value.')
      return
    end
  end
```

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\,\frac{(z-1)}{(z+1)} + kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \cfrac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \cfrac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \cfrac{1}{\cdots\cdots}}}$$

```
for ii=1:length(r)
  if( imag(r(ii)) ~= 0 )
      error('The coefficient in a vector must be real value.')
      return
  end
end

% Obtain denominator of filter transfer function
   if (k0==0)   % if the degree of antimirror/mirror-image polynomial is even
      odddeg = 0;
      [m,a] = dpolyeven(k,r);
      d = m+a;
   else % When the degree of antimirror/mirror-image polynomial is odd
      odddeg = 1;
      [m,a] = dpolyeven(k,r);
      fdenodddeg = [1 1];
      fnumodddeg = [1 -1];
      modddeg = conv(m,fdenodddeg);
      aodddeg = conv(m,k0*fnumodddeg)+conv(a,fdenodddeg);
      d = modddeg+aodddeg;
   end

% Obtain numerator of lowpass filter transfer function
   numlowpass = 1;
   numlowpass0 = [1 1];
  for ii=1:(2*length(r)+odddeg),
     numlowpass = conv( numlowpass,numlowpass0);
  end
```

```
  attenlow = max(abs(freqz(numlowpass,d,w)));
  wl = 1.5708*rand(1,3);
  Maglowpass = freqz(numlowpass,d,wl)/attenlow;
  Maglow = abs(Maglowpass);
  Dlow=d*attenlow;
end

if (LK <=1)
   if (LK==1)&(LR==1)&(k0==0)&k~=0 % n=2
      a=k*[1 0 -1];
      m=[1 r 1];
      d = m+a;
      numlowpass = [1 2 1];
      attenlow = max(abs(freqz(numlowpass,d,w)));
      wl = 1.5708*rand(1,3);
      Maglowpass = freqz(numlowpass,d,wl)/attenlow;
      Maglow = abs(Maglowpass);
      Dlow=d*attenlow;
  end

  if (LK==1)&(LR==1)&(k0~=0)&k~=0 % n=3
     a=k*[1 0 -1];
     m=[1 r 1];
     fdenodddeg = [1 1];
     fnumodddeg = [1 -1];
     modddeg = conv(m,fdenodddeg);
     aodddeg = conv(m,k0*fnumodddeg)+conv(a,fdenodddeg);
     d = modddeg+aodddeg;
```

```
    numlowpass = [1 3 3 1];
    attenlow = max(abs(freqz(numlowpass,d,w)));
    wl = 1.5708*rand(1,3);
    Maglowpass = freqz(numlowpass,d,wl)/attenlow;
    Maglow = abs(Maglowpass);
    Dlow=d*attenlow;
  end

  if ((LK==0&LR==0)&k0~=0)|((k==0&r==0)&k0~=0) % n=1
    fdenodddeg = [1 1];
    fnumodddeg = [1 -1];
    modddeg=fdenodddeg;
    aodddeg=k0*fnumodddeg;
    d = modddeg+aodddeg;
    numlowpass = [1 1];
    attenlow = max(abs(freqz(numlowpass,d,w)));
    wl = 1.5708*rand(1,3)
    Maglowpass = freqz(numlowpass,d,wl)/attenlow;
    Maglow = abs(Maglowpass);
    Dlow=d*attenlow;
  end
end

w=0.01:0.05:pi;
lowpass = freqz(numlowpass,d,w)/attenlow;
%lowpass = freqz(numlowpass,Dlow,w);
plot(w,abs(lowpass));
```

```
%set(gca,'XTick',0:pi/2:pi)
%set(gca,'XTickLabel',{'0','pi/2','pi'})
xlabel('\omega');
ylabel('Frequency Response of Lowpass');
axis([0 pi 0 1.2])
```

Algorithm 14: Matlab Subfunction dpolyeven.m

```
function [m,a] =dpolyeven(k,r)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  This function is a subfunction of dpoly.
% From the coefficients of continued-fraction expansion of F2(z)/F1(z),
% find the denominator of transfer function
%% Convert F2(z)/F1(z) to continued fraction expansions %%

Ncoe = length(r);
fnum=[1 0 -1];
m = zeros(1,2*Ncoe);
a = zeros(1,2*Ncoe);

i= Ncoe;
f1=[1 r(i) 1];
f2=k(i)*fnum;
for ii=2:Ncoe
i=i-1;
f4=[1 r(i) 1];
f3=k(i)*fnum;
  if rem(Ncoe,2)==0   % number of k is even
```

```
      a =conv(f2,f4)+ conv(f1,f3);
      m=conv(f1,f4);
      f1=a;
      f2=m;
    end
    if rem(Ncoe,2)==1    % number of k is odd
      a =conv(f1,f4)+ conv(f2,f3);
      m=conv(f2,f4);
      f2=a;
      f1=m;
    end
  end
```

Table 5.1: The Selected Frequencies and Magnitude Values of the Transfer Functions for Butterworth Lowpass Filters in z domain

	n	$k0$ and kc	r	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	2	[0 0.7071]		1.4925	0.7601	0.3631	0.9994	0.9532	0.9663
	3	[0.50000 0.5000]	[-0.6667]	0.7634	0.9979	1.4001	0.8584	1.1971	0.9532
	4	[0 0.4826 0.4826]	[1.0222 -1.0222]	0.7170	0.9998	0.0291	1.0000	1.2903	0.9522
lp	5	[0.3090 0.4578 0.2332]	[-1.2754 0.3311]	0.6985	0.9999	0.9667	0.9991	1.2440	0.9824
	6	[0 0.4191 0.1893 0.6483]	[1.4525 -0.3984 -1.2323]	1.4480	0.9025	1.1596	0.9970	0.2769	0.9999
	7	[0.2225 0.3803 0.1489 0.4222]	[-1.5758 0.1566 0.4635]	0.6373	0.9999	1.4694	0.8975	1.4403	0.9285
	8	[0 0.3454 0.2371 0.6809 0.0935]	[1.6637 -0.8770 -1.4063 0.2972]	0.6445	0.9999	1.4037	0.9653	0.0909	1.0000

Table 5.2: The Selected Frequencies and Magnitude Values of the Transfer Functions for Butterworth $CPPF$ Lowpass Filters in z domain

	n	kc	r	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	2	[0 0.6404]	[0.1724]	0.5543	0.9943	1.2773	0.9592	0.0155	0.9848
lp	4	[0 0.4993 0.4562]	[-0.9537 1.1917]	0.9484	0.9823	0.4276	0.9995	0.3123	0.9998
	8	[0 0.3622 0.2234 0.6376 0.1053]	[-1.6239 1.0294 1.4911 -0.1925]	0.0240	1.0000	1.1731	0.9867	0.6992	0.9999

Table 5.3: The Selected Frequencies and Magnitude Values of the Transfer Functions for Chebyshev Lowpass Filters in z domain

	n	kc	r	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	1	[0.3493]		1.4925	0.9516	0.3631	0.9980	0.9532	0.9841
	2	[0 0.5664]	[0.4102]	0.7634	0.9730	1.4001	0.9895	1.1971	0.9997
$A_p = 0.5dB$	3	[0.7981 0.4894]	[-0.5455]	0.7170	0.9524	0.0291	0.9995	1.2903	0.9818
$\epsilon = 0.3493$	4	[0 0.4732 0.2445]	[-1.1742 0.3719]	0.6985	0.9996	0.9667	0.9790	1.2440	0.9440
	5	[0.8529 0.4324 0.1014]	[-1.4502 -0.0433]	1.4480	0.9663	1.1596	0.9895	0.2769	0.9757
	6	[0 0.3382 0.0675 0.3885]	[-1.6435 -0.0590 0.0276]	0.6373	0.9876	1.4694	0.9550	1.4403	0.9446
	1	[0.5089]		0.6445	0.9859	1.4037	0.9186	0.0909	0.9997
	2	[0 0.5221]	[0.0980]	0.5543	0.9198	1.2773	0.9987	0.0155	0.8913
$A_p = 1.0dB$	3	[1.0121 0.5010]	[-0.6721]	0.2182	0.9867	0.3185	0.9731	0.3122	0.9740
$\epsilon = 0.5089$	4	[0 0.4299 0.1913]	[-1.2667 0.2059]	0.9484	0.9609	0.4276	0.9481	0.3123	0.9225
	5	[1.0675 0.4416 0.0982]	[-1.4882 -0.1126]	0.0240	0.9995	1.1731	0.9717	0.6992	0.8990
	6	[0 0.3076 0.0712 0.3182]	[-1.6737 -0.1080 -0.1456]	1.4637	0.9063	0.7320	0.9399	0.6576	0.9691
	1	[0.7645]		1.3292	0.8579	0.8249	0.9483	0.3183	0.9926
	2	[0 0.4911]	[-0.4435]	1.0558	0.9989	1.3165	0.8899	0.0308	0.8704
$A_p = 2.0dB$	3	[1.3550 0.5436]	[-0.7718]	1.0702	0.8104	0.5961	0.8511	1.3066	0.9335
$\epsilon = 0.7648$	4	[0 0.3648 0.1360]	[-1.3502 0.0602]	0.7898	0.9886	1.1144	0.8207	0.6737	0.9890
	5	[1.4154 0.4798 0.1017]	[-1.5183 -0.1684]	0.4785	0.8110	0.2979	0.8861	0.3038	0.8830
	6	[0 0.2633 0.0823 0.2397]	[-1.7014 -0.1471 -0.3011]	0.2370	0.8656	1.0963	0.8803	0.5943	0.9742

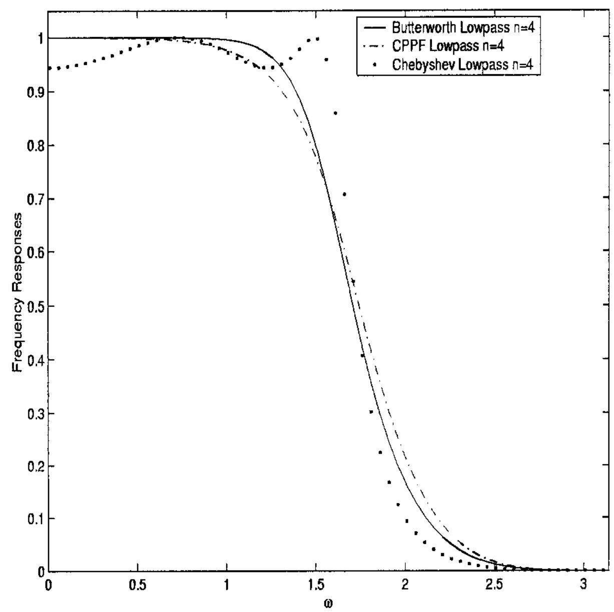

Figure 5.1: Graph of Lowpass Frequency Responses

5.1.2 Highpass Filter

a)

Using $s \rightarrow \frac{1}{s}$ transformation or m-file *lp2hp*, $\omega o = 1$, a highpass filter, whose cutoff frequency is 1, can be prooduced from a lowpass filter in s domain.

b)

First we take the nth-order highpass filter polynomial in s domain, and by applying the bilinear transformation $s = \frac{z-1}{z+1}$, we obtain a nth-order highpass filter transfer function in discrete domain.

$$T_{hd} = \frac{N_d(z)}{D_d(z)} = \frac{(z-1)^n}{D_{hd}(z)} \tag{5.3}$$

c)

The m-files of dhighbutter.m, dhighcpp.m and dhighcheby.m, which are used to examine the magnitude values of highpass filter transfer functions, are given below.

Then we select 3 frequencies randomly in passband, and the range of 3 frequencies is [$2\times\arctan(1)$, π].

The M-file dhighbutter.m is employed to examine the magnitude values of highpass filter transfer functions for Butterworth filters.

Algorithm 15: Matlab function dhighbutter.m

```
function [wh, Maghighbutter]= dhighbutter(N)
% N is the order of Butterworth filters
%  Maghighbutter is the vector of magnitudes values at wh in
% [ 2*arctan(1), π] randomly

[nums,dens]=butter(N,1,'high','s');
[NUMZ,DENZ]=bilinear(nums,dens,0.5);
wh = 4*atan(1)-2*atan(1)*rand(1,3);
Maghighbutter=abs(freqz(NUMZ,DENZ,wh));
w=0.01:0.01:pi;
highpass = freqz(NUMZ,DENZ,w);
plot(w,abs(highpass));
xlabel('\omega');
ylabel('Frequency Response of Highpass');
axis([0 pi 0 1.0])
```

Table 5.4: The Selected Frequencies and Magnitude Values of the Transfer Functions for Butterworth Highpass Filters in z domain

	n	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	2	2.3616	0.9860	1.7282	0.8081	1.8510	0.8700
	3	2.6044	0.9998	2.6865	0.9999	2.6056	0.9998
	4	1.8245	0.9413	2.2493	0.9986	2.5597	1.0000
hp	5	2.0506	0.9966	2.1656	0.9991	1.8931	0.9817
	6	2.8699	1.0000	1.6026	0.7709	2.7152	1.0000
	7	2.9271	1.0000	3.1231	1.0000	1.7375	0.9552
	8	2.6948	1.0000	2.4045	1.0000	3.0398	1.0000

The M-file dhighcpp.m is employed to examine the magnitude values of high-pass filter transfer functions for Butterworth $CPPF$ filters.

Algorithm 16: Matlab Function dhighcpp.m

```
function [wh, Maghighcpp]= dhighcpp(num,den)
% den is the coefficients of denominator for CPPF filters
% num is the numerator of CPPF filters
%  Maghighcheby is the vector of magnitudes values at wh in
% [ 2*arctan(1), π]
%  randomly

[nums,dens]=lp2hp(num,den,1);
[NUMZ,DENZ]=bilinear(nums,dens,0.5);
wh = 4*atan(1)-2*atan(1)*rand(1,3);
Maghighcpp=abs(freqz(NUMZ,DENZ,wh));
```

```
w=0.01:0.01:pi;
highpass = freqz(NUMZ,DENZ,w);
plot(w,abs(highpass));

%set(gca,'XTick',0:pi/2:pi)
%set(gca,'XTickLabel',{'0','pi/2','pi'})
xlabel('\omega');
ylabel('Frequency Response of Highpass');
axis([0 pi 0 1.0])
```

Table 5.5: The Selected Frequencies and Magnitude Values of the Transfer Functions for Butterworth $CPPF$ Highpass Filters in z domain

	n	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	2	3.1176	1.0000	1.9685	0.9955	2.4424	1.0134
hp	4	1.8124	0.8920	2.3167	0.9908	2.8233	0.9998
	8	2.0714	0.9947	2.5455	1.0000	1.8350	0.9592

The M-file dhighcheby.m is employed to examine the magnitude values of highpass filter transfer functions for Chebyshev filters.

Algorithm 17: Matlab Function dhighcheby.m

```
function [wh, Maghighcheby]= dhighcheby(N,A)
% N is the order of Chebyshev filters
% with A decibels of peak-to-peak ripple in the
% passband.
% Maghighcheby is the vector of magnitudes values at wh in
% [ 2*arctan(1), π]
```

```
% randomly

[nums,dens]=cheby1(N,A,1,'high','s');
[NUMZ,DENZ]=bilinear(nums,dens,0.5);
wh = 4*atan(1)-2*atan(1)*rand(1,3);
Maghighcheby=abs(freqz(NUMZ,DENZ,wh));

w=0.01:0.01:pi;
highpass = freqz(NUMZ,DENZ,w);
plot(w,abs(highpass));

%set(gca,'XTick',0:pi/2:pi)
%set(gca,'XTickLabel',{'0','pi/2','pi'})
xlabel('\omega');
ylabel('Frequency Response of Highpass');
axis([0 pi 0 1.0])
```

Table 5.6: The Selected Frequencies and Magnitude Values of the Transfer Functions for Chebyshev Highpass Filters in z domain

	n	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	1	2.6044	0.9954	2.6865	0.9967	2.6056	0.9954
	2	2.3027	0.9786	1.9994	0.9981	2.6558	0.9561
$A_p = 0.5dB$	3	1.8245	0.9869	2.2493	0.9443	2.5597	0.9639
$\varepsilon = 0.3493$	4	2.0377	0.9556	2.2830	0.9936	2.4428	0.9996
	5	2.0506	0.9991	2.1656	0.9931	1.8931	0.9656
	6	1.6386	0.9803	2.3207	0.9535	1.7591	0.9533
	1	1.5891	0.8946	2.2261	0.9700	2.4764	0.9849
	2	2.3318	0.9519	2.6170	0.9168	2.4616	0.9343
$A_p = 1.0dB$	3	2.7867	0.9673	2.2308	0.8914	1.9472	0.9291
$\varepsilon = 0.5089$	4	2.3093	0.9915	2.1355	0.9437	2.8132	0.9284
	5	2.5450	0.8913	1.9111	0.9412	2.0721	0.9999
	6	2.4173	0.9436	2.2496	0.8933	1.8940	0.9985
	1	2.4891	0.9681	2.6625	0.9830	1.7681	0.8472
	2	3.1180	0.7944	1.9353	0.9992	1.6166	0.8458
$A_p = 2.0dB$	3	1.5864	0.8344	1.9025	0.8849	2.4525	0.8262
$\varepsilon = 0.7648$	4	2.3588	0.9951	2.8055	0.8598	2.1308	0.8818
	5	2.6389	0.8051	1.6335	0.9906	2.0002	0.9684
	6	2.4945	0.9427	1.9720	0.9636	2.7207	0.9785

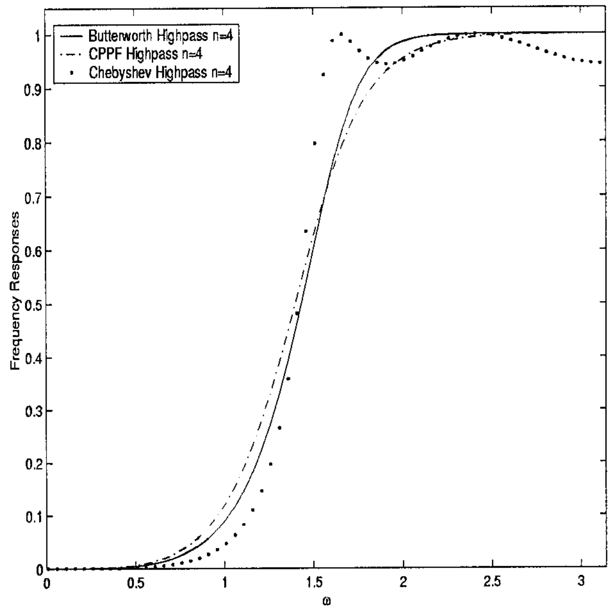

Figure 5.2: Graph of Highpass Frequency Responses

5.1.3 Bandpass Filter

a)

Using $s \rightarrow \frac{s^2+\Omega_0^2}{sB_\omega}$ transformation or m-file *lp2bp*, a bandpass filter can be prooduced from a lowpass filter in s domain; for example, center frequency $\Omega_0 = 1$, and bandwidth $B_\omega = 1$.

b)

First we take the nth-order bandpass filter polynomial in s domain, and by applying the bilinear transformation $s = \frac{z-1}{z+1}$, we obtain a nth-order bandpass filter transfer function in discrete domain.

$$T_{bd} = \frac{N_d(z)}{D_d(z)} = \frac{(z^2-1)^{\frac{n}{2}}}{D_{bd}(z)} \tag{5.4}$$

c)

The M-files of dbandbutter.m, dbandcpp.m and dbandcheby.m, which are used to examine the magnitude values of bandpass filter transfer functions, are given below. Then we select 3 frequencies randomly in passband, and the range of 3 frequencies is [$2\times\arctan(0.6180)$, $2\times\arctan(1.6180)$].

The M-file dbandbutter.m is employed to examine the magnitude values of bandpass filter transfer functions for Butterworth filters.

Algorithm 18: Matlab function dbandbutter.m

```
function [wb, Magbandbutter]= dbandbutter(N)
% N is the order of Butter filters
%  Magbandbutter is the vector of magnitudes values at wb in
% [ 2*arctan(0.6180),2*arctan(0.6180)] randomly

[b,a]=butter(N,1,'s');
[nums,dens]=lp2bp(b,a,1,1);
[NUMZ,DENZ]=bilinear(nums,dens,0.5);
wb = 2*atan(0.6180)+(2*atan(1.6180)-2*atan(0.6180))*rand(1,3);
Magbandbutter=abs(freqz(NUMZ,DENZ,wb));

w=0.01:0.01:pi;
bandpass = freqz(NUMZ,DENZ,w);
plot(w,abs(bandpass));

xlabel('\omega');
ylabel('Frequency Response of Bandpass');
axis([0 pi 0 1.0])
```

Table 5.7: The Selected Frequencies and Magnitude Values of the Transfer Functions for Butterworth Bandpass Filters in z domain

	n	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	2	1.7051	0.9973	1.8656	0.9382	1.7193	0.9960
	3	1.6024	1.0000	1.7814	0.9970	1.3939	0.9990
	4	1.7588	0.9998	1.6139	1.0000	1.5196	1.0000
bp	5	1.9944	0.8588	1.5917	1.0000	1.9233	0.9776
	6	1.3411	0.9999	1.9192	0.9894	1.7908	1.0000
	7	1.2918	0.9998	1.3841	1.0000	1.7205	1.0000
	8	2.0236	0.7795	1.6475	1.0000	1.4998	1.0000

The M-file dbandcpp.m is employed to examine the magnitude values of bandpass filter transfer functions for Butterworth $CPPF$ bandpass filters.

Algorithm 19: Matlab Function dbandcpp.m

```
function [wb, Magbandcpp]= dbandcpp(num, den)
% den is the coefficients of denominator for CPPF filters in s domain
% num is the numerator for CPPF filters in s domain
%  Magbandcpp is the vector of magnitudes values at wb in
% [ 2*arctan(0.6180),2*arctan(0.6180)] randomly

[nums,dens]=lp2bp(num,den,1,1);
[NUMZ,DENZ]=bilinear(nums,dens,0.5);
wb = 2*atan(0.6180)+(2*atan(1.6180)-2*atan(0.6180))*rand(1,3);
Magbandcpp=abs(freqz(NUMZ,DENZ,wb));

w=0.01:0.01:pi;
bandpass = freqz(NUMZ,DENZ,w);
```

```
plot(w,abs(bandpass));

xlabel('\omega');
ylabel('Frequency Response of Bandpass');
axis([0 pi 0 1.0])
```

Table 5.8: The Selected Frequencies and Magnitude Values of the Transfer Functions for Butterworth *CPPF* Bandpass Filters in z domain

	n	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
hp	2	1.9712	0.9308	1.5392	1.0006	1.4953	1.0032
	4	1.7304	0.9974	1.8843	0.9524	1.1253	0.7537
	8	1.5734	1.0000	1.7650	0.9998	1.5048	1.0000

The M-file dbandcheby.m is employed to examine the magnitude values of bandpass filter transfer functions for Chebyshev filters.

Algorithm 20: Matlab function dbandcheby.m

```
function [wb, Magbandcheby]= dbandcheby(N,A)
% N is the order of Chebyshev filters
% with A decibels of peak-to-peak ripple in the
% passband.
% Magbandcheby is the vector of magnitudes values at wb in
% [ 2*arctan(0.6180),2*arctan(0.6180)] randomly

[b,a]=cheby1(N,A,1,'s');
[nums,dens]=lp2bp(b,a,1,1);
[NUMZ,DENZ]=bilinear(nums,dens,0.5);
wb = 2*atan(0.6180)+(2*atan(1.6180)-2*atan(0.6180))*rand(1,3);
```

```
Magbandcheby=abs(freqz(NUMZ,DENZ,wb));

w=0.01:0.01:pi;
bandpass = freqz(NUMZ,DENZ,w);
plot(w,abs(bandpass));

xlabel('\omega');
ylabel('Frequency Response of Bandpass');
axis([0 pi 0 1.0])
```

Table 5.9: The Selected Frequencies and Magnitude Values of the Transfer Functions for Chebyshev Bandpass Filters in z domain

	n	ω_1	$mag1$	ω_2	$mag2$	ω_3	$mag3$
	1	1.5578	1.0000	1.9336	0.9666	1.8138	0.9853
	2	1.5304	0.9454	1.1243	0.9601	1.8688	0.9964
$A_p = 0.5dB$	3	1.5195	0.9944	1.6778	0.9784	1.8415	0.9460
$\varepsilon = 0.3493$	4	1.9619	0.9682	1.7917	0.9950	1.2706	0.9548
	5	1.4833	0.9655	1.9746	0.9524	1.9574	0.9442
	6	1.4876	0.9830	1.9358	0.9859	1.1608	0.9441
	1	1.4343	0.9904	1.8612	0.9567	1.1162	0.8954
	2	1.2359	0.9999	1.2951	0.9836	1.2914	0.9852
$A_p = 1.0dB$	3	1.6670	0.9630	1.3595	0.8965	1.2915	0.8975
$\varepsilon = 0.5089$	4	1.1213	0.9706	1.7996	0.9837	1.5198	0.9061
	5	1.9712	0.9013	1.5392	0.9877	1.4953	0.9438
	6	1.8918	0.9854	1.5941	0.8983	1.2950	0.9105
	1	1.7304	0.9710	1.8843	0.8959	1.1253	0.8075
	2	1.7389	0.8618	1.4590	0.8240	1.8784	0.9894
$A_p = 2.0dB$	3	1.5734	0.9999	1.7650	0.8129	1.5048	0.9588
$\varepsilon = 0.7648$	4	1.3896	0.9986	1.2830	0.8468	1.2865	0.8532
	5	1.7397	0.7986	1.3879	0.8096	1.6094	0.9608
	6	1.2470	0.9758	1.7543	0.8948	1.4580	0.9887

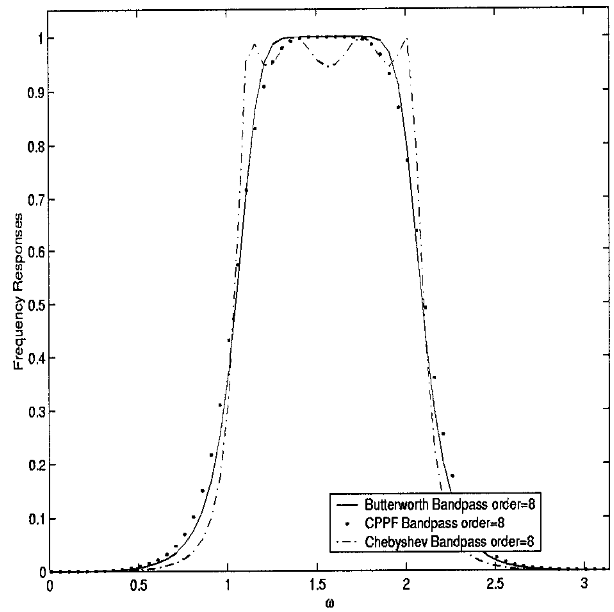

Figure 5.3: Graph of Bandpass Frequency Responses

5.2 Summary and Discussion

In this chapter techniques are developed for establishing algorithms to obtain stable transfer functions of filters in z domain from coefficients kc and γ of the continued fraction expansions $(CFE4)$.

The chapter has also considered the computer-aided design of filters.

Chapter 6

Summary and Discussion

In this thesis, the following three types of continued fraction expansions $CFE1$,$CFE2$ and $CFE3$ have been discussed.

1) $CFE1$:

For n is even or odd,

$$\frac{F_1(z)}{F_2(z)} = ka(1)\frac{(z+1)}{(z-1)} + \frac{1}{ka(2)\frac{(z+1)}{(z-1)} + \frac{1}{ka(3)\frac{(z+1)}{(z-1)} + \frac{1}{\cdots\cdots}}}$$

where $ka(1), ka(2)...ka(n)$ are real and positive.

2) $CFE2$:

(a) $CFE\ 2a$:

For n is even,

$$\frac{F_{1e}(z)}{F_{2e}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\cdots\cdots}}}$$

where $kb(1), kb(2)...kb(n)$ are real and positive.

(b) $CFE\ 2b$:

If n is odd,

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kb(1)\frac{(z-1)}{(z+1)} + \frac{1}{kb(2)\frac{(z-1)}{(z+1)} + \frac{1}{kb(3)\frac{(z-1)}{(z+1)} + \frac{1}{\cdots\cdots}}}$$

where $kb(1), kb(2)...kb(n)$ are real and positive.

3) $CFE\ 1a$

$$\frac{F_1(z)}{F_2(2)} = r_1(z-1) + \cfrac{1}{r_2(1-z^{-1}) + \cfrac{1}{r_3(z-1) + \cfrac{1}{r_4(1-z^{-1}) + \frac{1}{...}}}}$$

4) $CFE3$

For n is odd,

$CFE\ 3a$

$$\frac{F_{1o}(z)}{F_{2o}(z)} = ko(1)\frac{z+1}{z-1} + \cfrac{1}{ko(2)\frac{z+1}{z-1} + ko(3)\frac{z-1}{z+1} + \frac{1}{......}}$$

For n is even,

$CFE\ 3b$

$$\frac{F_{1e}(z)}{F_{2e}(z)} = ke(1)\frac{z-1}{z+1} + ke(2)\frac{z+1}{z-1} + \cfrac{1}{ke(3)\frac{z-1}{z+1} + ke(4)\frac{z+1}{z-1} + \frac{1}{......}}$$

where kf and kz are real and positive.

MATLAB programmes have been developed so that the various coefficients in each continued fraction expansion can be obtained.

5) We have also produced the new type of continued fraction expansion given as follows $CFE4$:

For n = odd,

$$\frac{F_{2o}(z)}{F_{1o}(z)} = kc(0)\frac{(z-1)}{(z+1)} + kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \cfrac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \cfrac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{......}}}$$

where $kc(0), kc(1), kc(2)...kc(\frac{n-1}{2})$ are real and positive; $0 \leq |\gamma_i| < 2, i = 1, 2...\frac{n-1}{2}$.

For n = even,

$$\frac{F_{2e}(z)}{F_{1e}(z)} = kc(1)\frac{(z^2-1)}{(z^2+\gamma_1 z+1)} + \cfrac{1}{kc(2)\frac{(z^2-1)}{(z^2+\gamma_2 z+1)} + \cfrac{1}{kc(3)\frac{(z^2-1)}{(z^2+\gamma_3 z+1)} + \frac{1}{......}}}$$

where $kc(0) = 0$; $kc(1), kc(2)...kc(\frac{n}{2})$ are real and positive;$0 \leq |\gamma_i| < 2, i = 1, 2...\frac{n}{2}$.

Unlike the $CFE1$, $CFE2$ and $CFE3$, $CFE4$ is non-unique for a given Schur polynomial $D(z)$. This is because the CFE is based on the zeros of a mirror-image or an anti-mirror image polynomial located on the unit circle. This shows that a large number of possibilities exist. Algorithms have been given to obtain the various possible $CFE4$'s for a given $D(z)$, and some have been written in MATLAB.

In addition, it is shown that given coefficients of a $CFE4$ have been shown, it is possible to obtain stable transfer functions so that filters can be designed from the coefficients of $CFE4$.

The three necessary frequency samples have been taken randomly for lowpass, highpass and bandpass respectively. The techniques produce good performance filters with good magnitude responses of lowpass, highpass and bandpass filters.

This thesis can be considered as a starting point for application of continued fraction expansions in filter design. The same methods are suggestions for any other kinds of stable filter design.

Bibliography

[1] A.Antoniou , *Digital Filters*, McGraw-Hill, Inc., PP. 76-78, 1993

[2] H.W.Schuessler, *A Stability Theorem for Discrete Systems*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-24, PP.87-89, February 1976

[3] J.SZCZUPAK, S.K.MITRA, and E.I.JURY, *Some New Results on Discrete System Stability* , IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-25, PP.101-102, February 1977

[4] R.GNANASEKARAN, *A Note on the new* $1-D$ *and* $2-D$ *Stability Theorems for Discrete Systems* , IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, No.6, PP.1211-1212, December 1981

[5] V.Ramachandran and C.S. Gargour, *Implementation of a Stability Test of* $1-D$ *Discrete System Based on Schussler's Theorem and Some Consequent Coefficient Conditions*, Journal of Franklin Institute, Vol.317, No.5, PP.341-358, May 1984

[6] V.Ramachandran, Ravi P. Ramachandran and C.S. Gargour, *Some Properties of the* $z-$ *domain Continued Fraction Expansion of* $1-D$ *Discrete Reactance Functions*, International Symposium on Circuits and Systems (ISCAS), Sydney, Australia, May 2001, PP II-841 to II-844

[7] A.M.Davis, *A new $z-$domain Continued Fraction Expansion* , IEEE Transactions on Circuits and Systems, Vol. CAS-29, No.10, PP 658-662, October 1982

[8] Y. Bistritz, *$z-$domain Continued Fraction Expansions for Stable Discrete Systems Polynomials* , IEEE Transactions on Circuits and Systems, Vol. CAS-32, No.11, PP 1162-1166, November 1985

[9] C. Hwang, J. Hwang and S.Y. Tsay, *A Stability Test for Discrete Systems Using Davis' $z-$domain Continued Fraction Expansion*, IEEE Transactions on Circuits and Systems - I, Fundamental Theory and Applications, Vol. 46, No.8, PP 1012-1018, February 1999

[10] V.Ramachandran, *Determination of Discrete Transfer Function From Its Real (or Imaginary) Part on the Unit Circle*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 37, No. 3, PP.440-442, March 1989

[11] M.E. Van Valkenburg, *Analog Filter Design*, Holt, Rinehart Winston, 1982

[12] K.L. Su, *Analog Filters* , Chapman and Hall, 1996

[13] K.R. Pai, K.V.V. Murthy and V.Ramachandran, *Complementary Pole$-$pair Filters and Pole$-$parameter Transformations* , Journal of Circuits, Systems and Computers, World Scientific Publishing Company, Vol. 6, No.4 (1996) 319-350

[14] V.Ramachandran and C.S. Gargour, *Generation of Very Strict Hurwitz Polynomials and Applications to $2-D$ Filter Design*, Academic Press, Inc., Control and Dynamic System, Vol. 69, 1995

[15] C.S. Gargour and V.Ramachandran, *Generation of Stable $2-D$ Transfer Functions Having Variable Magnitude Characteristics* , Academic Press, Inc., Control and Dynamic System, Vol. 69, 1995