2007

Transfer Function and Impulse Response Synthesis using Classical Techniques

Sonal S. Khilari
University of Massachusetts Amherst

TRANSFER FUNCTION AND IMPULSE RESPONSE SYNTHESIS USING CLASSICAL TECHNIQUES

A Thesis Presented

by

SONAL S. KHILARI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2007

Electrical and Computer Engineering

TRANSFER FUNCTION AND IMPULSE RESPONSE SYNTHESIS USING CLASSICAL TECHNIQUES

A Thesis Presented

by

SONAL S. KHILARI

Approved as to style and content by:

____
Dev Vrat Gupta, Chair

____
Patrick Kelly, Member

____
Weibo Gong, Member

____
Christopher V. Hollot, Department Head
Electrical and Computer Engineering

ABSTRACT

TRANSFER FUNCTION AND IMPULSE RESPONSE SYNTHESIS
USING CLASSICAL TECHNIQUES

SEPTEMBER 2007

SONAL S. KHILARI B.E., UNIVESITY OF MUMBAI

M.S.ECE., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Dev Vrat Gupta

This thesis project presents a MATLAB based application which is designed to synthesize any arbitrary stable transfer function. Our application is based on the Cauer synthesis procedure. It has an interactive front which allows inputs either in the form of residues and poles of a transfer function, in the form of coefficients of the numerator and denominator of the transfer impedance or in the form of samples of an impulse response. The program synthesizes either a single or double resistively terminated LC ladder network. Our application displays a chart showing the variation of stability of an impulse response with the addition of delay. An attempt is made to synthesize usually unstable impulse responses by calculating the delay that would make them stable.

TABLE OF CONTENTS

LIST OF TABLES

TABLE Page

LIST OF FIGURES

CHAPTER 1

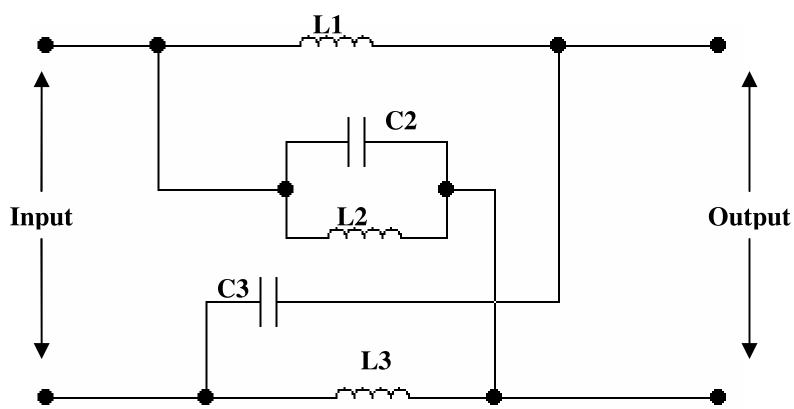
INTRODUCTION

Network synthesis involves the methods used to determine an electric circuit that satisfy certain specifications. Given an impulse response there are myriad techniques that can be used to synthesize a circuit with the specified response. Different methods may also be used to synthesize circuits, all of which may be optimal. Hence the solution to a network synthesis problem is never unique.

1.1. Why use analog passive, LC ladder networks?

Many applications today use digital processing in lieu of analog processing and the GHz spectrum is finding increasing use in applications such as wireless communications. However, operation at high frequencies requires analog filtering and processing circuits simply because using digital techniques is neither realistic nor economical. Another advantage that analog devices have over their digital counterparts is their ability to operate with wide instantaneous bandwidths and moderately high dynamic ranges at microwave frequencies [3]. Analog circuits with passive elements are generally preferred unlike active components, as they do not require an excitation source. Passive LC networks are also more advantageous as compared to active networks since they have a high tolerance to component variances and are simple. Also analog passive circuits can be used as prototypes for designing active networks, interface circuits, transmission lines and other complex networks with discrete components or on chips.

Most importantly, passive LC circuits generally operate in the range of 10^2 to 10^9 Hz [3]. As we will be dealing with high frequency applications (of the order of GHz) in this project, we felt that it was best to use analog passive LC circuits.



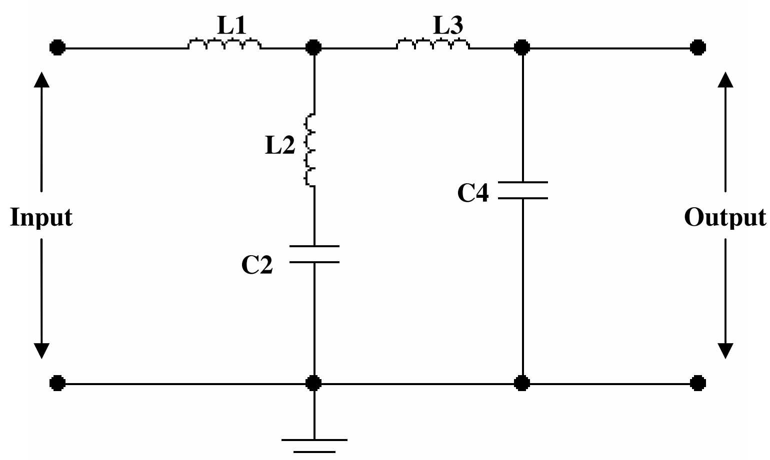


Figure 1.1 Examples of lattice (top) and ladder (below) networks

Two of the most commonly synthesized network structures are lattice networks and ladder networks as shown in Figure 1.1. Lattice structures are relatively simple but balanced circuits. This means that they do not have a common ground between input and output. Also because of tolerance requirements they are usable only when the specified transfer function has a zero on or near the '*jw*' axis [7]. Although this problem can be solved by applying balanced to unbalanced conversion methods using transformers (e.g. the Weinberg synthesis procedure), all of these techniques only lead to relatively complicated parallel networks [7].

On the other hand, ladders are popular structures for circuits because the shunt or series LC arms are directly related to the transmission zeros by $\omega_{transmission} = \frac{1}{\sqrt{LC}}$. This makes circuit tuning not only a simpler process but also making the loss poles relatively insensitive to element variations as compared to balanced networks [7]. Hence ladders are preferred over lattices.

1.2. Thesis objective and applications

There is no dearth of literature on the methods to synthesize transfer functions. The problem arises in using these methods to synthesize a specific function into realizable elements. Today there are numerous software applications for filter synthesis available in the market. However, all the available applications synthesize only filters of standard families such as Butterworth, Chebyshev, Elliptic and so on. We have not found any applications that can be used for the synthesis of any arbitrary transfer function. Also, in communications and signal processing it sometimes becomes necessary to synthesize impulse responses as well; an option that is unavailable in synthesis programs commercially available today.

Hence, the objective of this thesis is to develop a generic application in software (MATLAB ®) that will synthesize (using classical synthesis techniques) a lossless passive network for any arbitrary stable transfer function. This application will synthesize stable transfer functions with zeros located anywhere in the s-plane, instead of being limited to the imaginary axis. Furthermore, in cases where the input impulse response is not stable i.e. if the poles of the transfer function lie in the right hand side of the s-plane (for continuous case) or lie outside the unit circle (in the case of a discreet system), our program will calculate the delay (up to a specified granularity) required to make this impulse response stable after which it will proceed to synthesize this modified response; thus synthesizing a delayed stable version of the unstable impulse response.

Our main objective behind this project is to enhance the circuit design process and enable circuits designers synthesize a variety of arbitrary transfer functions and impulse responses. In other words, the motivation behind this project is to automate the process of network synthesis, making it simple, efficient and fast.

The outline of this thesis is as follows. In chapter 2, the various prerequisites for network synthesis and an overview on the various methods available to synthesize a transfer function are presented. We also mention briefly the most widely used commercially available filter synthesis applications. Chapter 3 describes the procedures used to synthesize single and double resistively terminated LC ladder networks. In chapter 4, some results for the synthesis of transfer functions are presented. Finally, in chapter 5, the synthesis method is extended to impulse responses and results showing the variation in the stability of impulse responses with delay are presented.

CHAPTER 2

BACKGROUND (OVERVIEW OF WORK DONE IN THE PAST)

In order to synthesize a driving point function into a passive network using resistors, inductors and capacitors, it must be positive real; a fact that was first demonstrated by Otto Brune [8]. This means that the following properties must be satisfied,

a. It must be a rational function of the complex frequency, *s*.
b. The poles must lie on the left hand side of the *jw* axis or on the imaginary axis (stable function).
c. The poles on the *jw* axis must be simple (multiplicity of 1). The denominator polynomial must be Hurwitz.
d. Complex poles and zeros must occur in conjugate pairs.

Most of the transfer function synthesis methods in literature can be considered to be realizations of driving point impedances [8]. Hence the same conditions of positive realness are applicable for transfer functions also. For a driving point function, the zeros should have negative or zero real parts. However in the case of synthesis of transfer functions, there is no restriction on the location of zeros. But to synthesize lossless circuits (those with only inductors and capacitors), the zeros must lie exclusively on the imaginary (*jw*) axis [11].

2.1. Theoretical synthesis of two-terminal networks

The most practical application of passive network synthesis is the synthesis of two-terminal transfer functions, which is what we will be focusing on throughout this thesis project. Today, there are various methods available for the synthesis of one-port to *n*-port networks. Due to the exhaustive literature present on the synthesis of two-terminal networks, we will not discuss every one of these techniques in detail. Instead we will focus only on a few important methods.

The synthesis of transfer impedance can be considered to be the realization of the associated driving point impedance at the zeros of transmission [7]. Also, since networks serve the purpose of point-to-point (or multipoint) transmission of information, two-port networks (or in general, n-port networks) are more practical. The design of these higher port networks has its roots in the synthesis of one-port networks. Notable among the driving point impedance synthesis methods are those by Brune, Bott Duffin, Darlington and Cauer.

The first method for the synthesis of passive networks was proposed by Brune. The main idea behind this method is the removal of the zeros located at the origin, infinity and on the *jw* axis. The Bott Duffin method is quite similar to Brune's method, but is more complex as it does not use transformers. Kuh and Miyata proposed transfer function simplification by splitting the input function into a sum of functions easier to realize. Darlington's method synthesizes resistively terminated reactive networks (containing inductors, capacitors and transformers). This method uses surplus factors and requires ideal components. Cauer's method also requires the use of transformers in case a negative inductance is encountered but is by far the easiest and simplest method to implement. This method is based on continued fraction expansion or in Foster's representation; the function may be split into partial fractions to realize a ladder structure.

In addition, there are many other techniques for network synthesis based on one or more of the earlier methods. They differ in the procedure followed to attain the impedance (or admittance) functions to be realized. Some of these techniques include Guillemin's transfer admittance synthesis (which realizes the impedance function as the summation of a series of functions each having a single numerator terms), Lucal's method (decomposition of driving point functions to realize the conditions for RC synthesis) and so on. A more detailed description on classical and modern synthesis methods is provided in literature [9].

2.2. Available commercial filter design applications (in software)

There are a large number of commercially available software in the market that can be used to design and synthesize filters. For example, MATLAB has a filter design tool called FDAtool which allows digital filter synthesis of standard filters. Filter Solutions (by Nuhertz Technologies), Filter Master (by Intusoft), S/FILSYN (by ALK Engineering) and others* design filters by scaling them from a large variety of normalized reference low pass filters. These filter design applications perform active and passive filter synthesis of all types of filters (high pass, low pass, band pass etc) in only the standard approximations (Chebyshev, Butterworth, elliptic etc).

Figure 2.1 shows the console of a commercially available filter synthesis application (S/Filsyn by ALK Engineering).

* www.circuitsage.com has a list of different filter synthesis programs available.

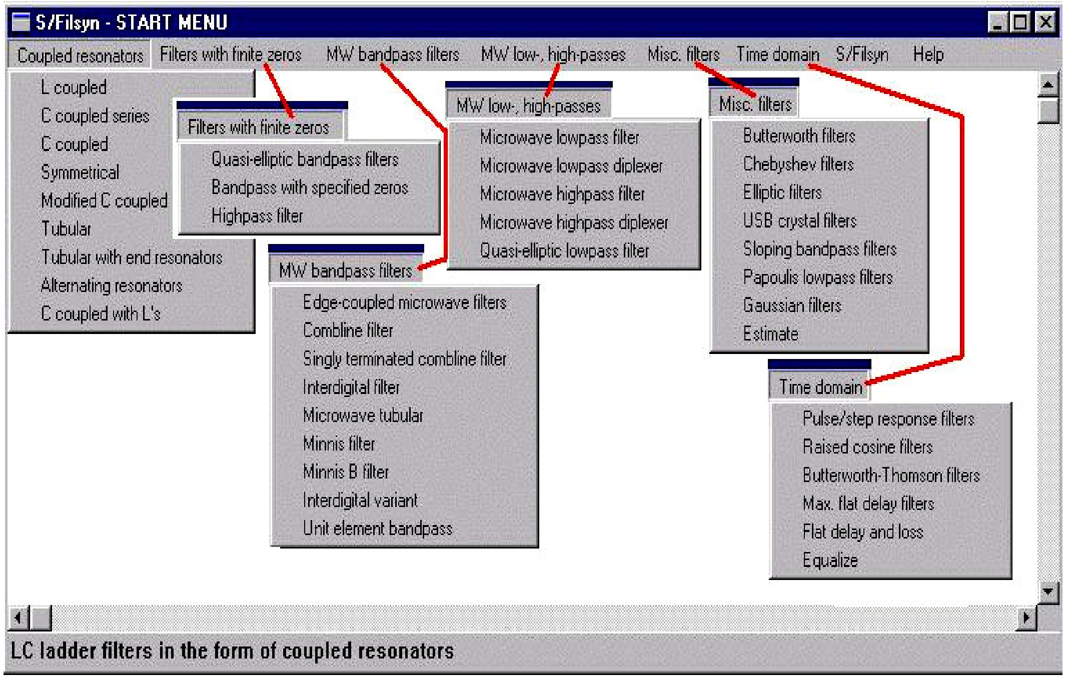


Figure 2.1 A commercially available filter synthesis application

In summary, this chapter describes in brief the various theoretical methods available for the synthesis of two-port lossless ladder networks. Although commercial network synthesis software exist, none of these applications allow the user to enter an arbitrary transfer function. They do not give the user the option to enter an arbitrary transfer function or impulse response to be synthesized. The main advantage of our application is that it offers the user both options.

CHAPTER 3

SYNTHESIS OF TRANSFER FUNCTIONS USING RESISTIVELY TERMINATED LADDER NETWORKS

3.1. The objective

In the synthesis of networks, the most commonly preferred synthesis methods are the two-element synthesis methods. In this chapter, the synthesis of transfer functions using single and double resistively terminated lossless ladder networks is considered. Synthesis is performed using the classical techniques proposed by Cauer, which is based on continued fraction expansion.

3.2. The algorithm (The Cauer-Guillemin synthesis technique)

The method for the synthesis of a transfer function as proposed by Cauer and Guillemin can be reduced to the problem of realizing an associated driving point function with a specific number of transmission zeros [8, 10]. This synthesis procedure describes a convenient way of splitting the given transfer function and we end up with two networks to synthesize, both of which have zeros exclusively on the imaginary axis. The resultant networks are not only simpler to compute but also easier to analyze since they are purely reactive ladder networks. The only components in the circuit are inductors and capacitors or combinations of the two in shunt or series branches, with the exception of the termination resistance and/or source resistance. The circuit thus synthesized is efficient since it generates a minimum number of elements, which is equal to the order of the numerator or the denominator (whichever is higher), to produce a given response.

The Cauer synthesis method is preferred because it is a simple technique based on continued fraction expansion, which results in an unbalanced ladder network. Another desirable feature about Cauer synthesized networks is that it is possible to synthesize a circuit such that no transformers are required. For synthesis, only two functions need be known: namely, the driving point impedance function and the transfer impedance function.

3.3. Synthesis of single resistively terminated lossless ladder networks

The general form of the transfer impedance function $H_{12}(s)$ can be represented as [10]

$$H_{12}(s) = Z_{12} = \frac{Z_L \cdot z_{12}}{Z_L + z_{22}}$$

where

Z_L = Load impedance

Z_{12} = Transfer impedance when the circuit is terminated with a load 'Z_L'

z_{12} = Backward open circuit transfer impedance

z_{22} = Open circuit input impedance (driving point impedance).

For the case of $Z_L = 1\Omega$ we get

$$H_{12} = \frac{z_{12}}{1 + z_{22}}$$

This impulse response can be written in the following form

$$H_{12}(s) = \frac{N(s)}{D(s)} = \frac{N_{even}(s) + N_{odd}(s)}{D(s)} = \frac{N_{even}(s)}{D(s)} + \frac{N_{odd}(s)}{D(s)}$$

where

$N_{even}(s)$ = Even part of the numerator of the transfer function.

$N_{odd}(s)$ = Odd part of the numerator of the transfer function.

$D(s)$ = Denominator of the transfer function.

An even polynomial is one which contains terms with only even powers of 's' (for example $a_0 + a_2s^2 + a_4s^4 + a_6s^6 + \ldots$). Similarly an odd polynomial is one which contains terms with only odd powers of 's' (for example $a_1s + a_3s^3 + a_5s^5 + \ldots$). $D(s)$ is a polynomial which contains both even and odd terms.

This network synthesis method is sensitive to the impedance level and hence when we split the numerator, we must account for a scale factor [10]. An inefficient solution is to use transformers with varying turn ratio. This can be avoided by adjusting the impedance level so that we have a 1:1 transformer, which means that we do not have to use a transformer at all.

In this case $H_{12}(s)$ must be a minimum phase function. Once separated the zeros of $H_{12,even}(s) = \frac{N_{even}(s)}{D(s)}$ and $H_{12,odd}(s) = \frac{N_{odd}(s)}{D(s)}$ lie exclusively at the origin, infinity or on the $j\omega$ axis. The two transfer impedances can be realized separately to yield lossless ladder circuits using the Cauer synthesis method and the zero shifting techniques described below.

Cauer Synthesis Technique

The Cauer synthesis methods (Cauer I and Cauer II) are based on the continued fraction expansion method and involve the removal of alternate series and shunt elements as needed.

The zeros at infinity are realized using the Cauer I synthesis method. The important point to note in the Cauer I synthesis method is that the starting driving point function should always be such that the degree of the numerator is greater than the degree of the denominator [12]. Let $Z_1(s)$ be the original driving point function. The numerator is divided by the denominator to yield

$$Z_1(s) = sL_1 + Z_2(s).$$

$Z_2(s)$ is the remainder and the order of this function is one less than $Z_1(s)$. The next step is to produce an admittance branch. This is done by

$$Y_2(s) = sC_2 + Y_3(s).$$

Here $Y_2(s) = 1/Z_2(s)$ and $Y_3(s)$ is the remainder driving point function. The synthesis is carried out by removing series inductors and shunt capacitors each element accounting for one zero at infinity [11].

On the other hand, the zeros at the origin are realized using the Cauer II method which is also a continued fraction expansion but in this case the polynomials are arranged in ascending order of their powers. In the case of the Cauer II synthesis technique the starting driving point function is always chosen such that the denominator is an odd polynomial [12]. The procedure to determine the series capacitors and shunt inductors is the same as that described for the Cauer I synthesis method. In this case, the synthesis is carried out by removing series capacitors and shunt inductors where each element accounts for one zero at the origin [11].

The Cauer I and Cauer II techniques, however, can not be used if there are finite zeros on the imaginary axis (transmission zeros). The continued fraction expansion is valid only for transfer functions with zeros at the origin or at infinity [11]. If the transmission zeros coincide with the poles of the driving point impedance, they can be realized as series impedance branches or shunt admittance branches [13].

Zero shifting procedure:

However, when the transmission zeros do not occur at the poles of the driving point impedance function or any of its remainders, the zero shifting technique must be used. The pole locations are created by introducing redundant elements [13]. In other words, the basic idea is to manipulate the driving point

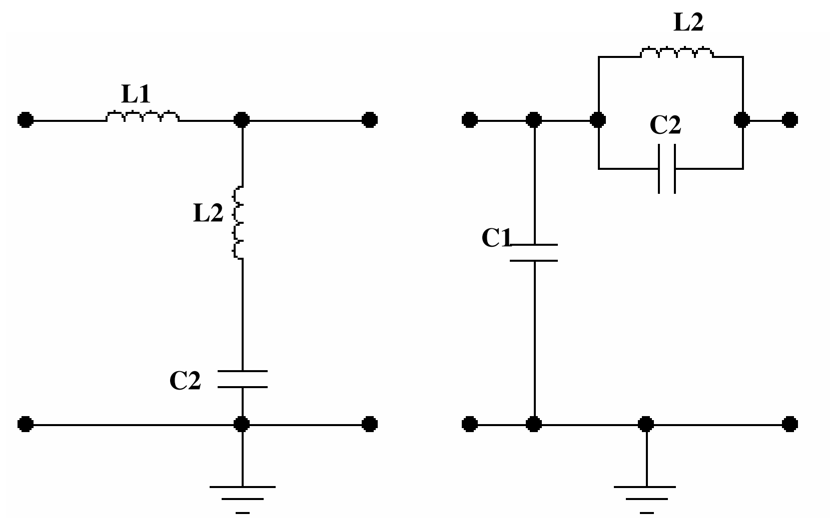


Figure 3.1 Shunt resonant and series anti-resonant sections

impedance function *(z_{22})* by pulling out a series inductance or shunt capacitance so that it exhibits a transmission zero at the frequency of the pole of the driving point function. This zero can then be removed as a shunt resonant or a series anti-resonant section while synthesizing the driving point function as shown in Figure 3.1

However, in the event that $H_{12}(s)$ is not minimum phase, it can be realized as a cascade of a minimum phase network and an all pass network as described below. Consider the response

$$H_{12}(s) = \frac{N(s)}{D(s)} = \frac{N`(s)\cdot(s-a)\cdot(s^2 - bs + c)}{D(s)} \qquad a > 0, b > 0$$

in which we have three zeros in the right hand side of the s-plane. They are, $s = a$ and $s = \frac{b \pm \sqrt{b^2 - 4c}}{2}$. If $N`(s)$ is the part of the numerator without right plane zeros, the original impulse response *H(s)* can be modified as

$$H_{12}(s) = \frac{N(s)}{D(s)} = \frac{N`(s)\cdot(s+a)\cdot(s^2 + bs + c)}{D(s)} \cdot \left(\frac{(s-a)}{(s+a)}\right) \cdot \left(\frac{(s^2 - bs + c)}{(s^2 + bs + c)}\right)$$

Substituting the realizable part of the impulse response with *H'(s)* we get,

$$H`_{12}(s) = \frac{N`(s)\cdot(s+a)\cdot(s^2 + bs + c)}{D(s)}$$

and $H_{12}(s)$ can be written in the form

$$H_{12}(s) = \frac{N(s)}{D(s)} = H`_{12}(s) \cdot \frac{(s-a)}{(s+a)} \cdot \frac{(s^2 - bs + c)}{(s^2 + bs + c)}$$

$$H_{12}(s) = \frac{N(s)}{D(s)} = H`_{12}(s) \cdot A_1(s) \cdot A_2(s)$$

$H`_{12}(s)$ can be synthesized using the method described previously in this section along with the Cauer continuous fraction expansion and zero shifting methods; Finally, this realization is cascaded with $A_1(s)$ as a first order all pass filter and $A_2(s)$ as a second order all pass filter.

Realization of All Pass Filters:

All pass filters, as the name suggests, are networks that have a flat frequency response but introduce a frequency phase shift. All pass filters are also known as delay equalizers or constant resistance networks as the input impedance has a constant value of *R* ohms throughout the frequency range. These constant resistance networks can be cascaded without loading. The transfer function of a first order all pass filter can be represented as $T(s) = \frac{s-a}{s+a}$. The magnitude is $|T(jw)| = \frac{|s-a|}{|s+a|} = \frac{\sqrt{w^2+a^2}}{\sqrt{w^2+a^2}} = 1$ and the phase shift is $\beta(w) = -2\tan^{-1}\left(\frac{w}{a}\right)$. The first order all pass section is relatively simple since it has only one parameter *(a)*. The first order LC all pass filter can be represented as shown in Figure 3.2. In the figure, values of the inductance and capacitance are $L = \frac{2R}{a}$ and $C = \frac{2}{R.a}$ respectively [15].

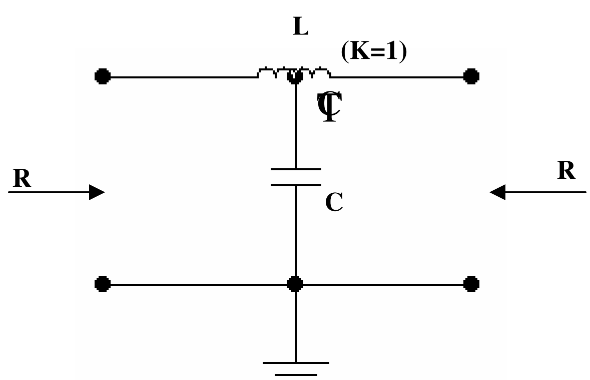


Figure 3.2 First order all pass filter

The second order all pass filter can be represented as

$$T(s) = \frac{s^2 - as + b^2}{s^2 + as + b^2} = \frac{s^2 - \left(\frac{w_r}{Q}\right)s + w_r^2}{s^2 + \left(\frac{w_r}{Q}\right)s + w_r^2}$$. On comparison, $w_r = b$ and $Q = \frac{b}{a}$. The design of a second order all pass filter is a little more complicated since it has two parameters *(w_r and Q)*. In this case the Q-factor needs to be taken into account and the circuit representation varies depending on whether Q is greater than 1 or less than 1

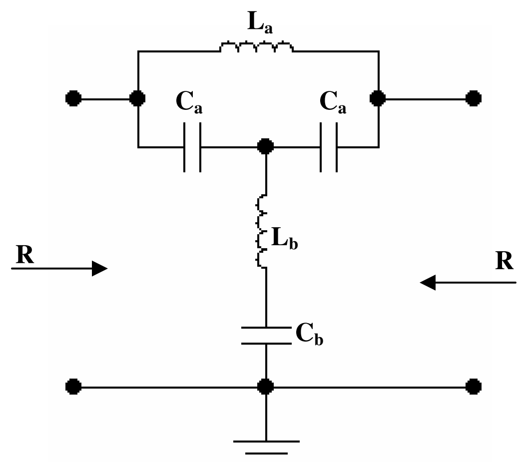


Figure 3.3 Second Order All Pass sections when Q > 1

When Q>1, the circuit in Figure 3.3 is used with values as follows [15].

$$L_a = \frac{2R}{w_r Q} \quad and \quad C_a = \frac{Q}{w_r R}$$
$$L_b = \frac{QR}{2w_r} \quad and \quad C_b = \frac{2Q}{w_r R \cdot (Q^2 - 1)}$$

When Q<1, the value of C_a becomes negative, hence a different circuit (shown in figure 3) is used. In this case the capacitance values are,

$$C_3 = \frac{Q}{2R \cdot w_r} \quad and \quad C_4 = \frac{2}{Q \cdot R \cdot w_r}$$

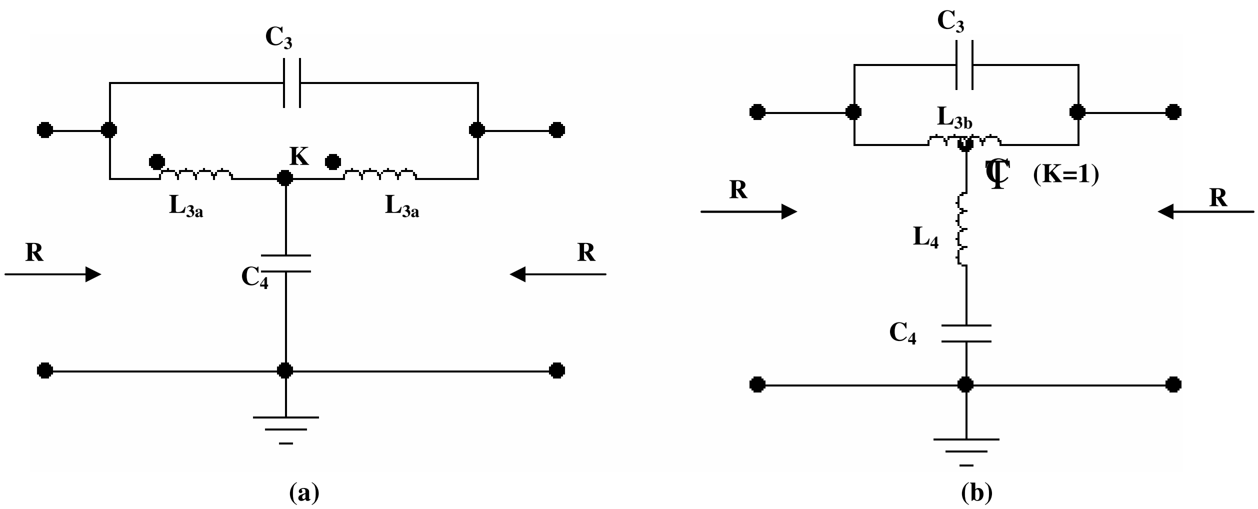


Figure 3.4 Second Order All Pass sections when Q < 1 (a) using coupled inductors; (b) Coupling coefficient =1

Although this circuit can be represented using coupled inductors (shown in Figure 3.4(a)), this is not a very convenient method. It is more practical to use a center tapped inductor with a couple coefficient of 1 (shown in Figure 3.4(b)). The values of the inductors L_{3b} and L_4 are found as follows [15]

Coupling coefficient: $K = \frac{1-Q^2}{1+Q^2}$

$$L_{3a} = \frac{(1+Q^2)R}{2Q \cdot w_r}$$

$$L_{3b} = 2(1+K)L_{3a} \quad and \quad L_4 = \frac{(1-K)L_{3a}}{2}$$

First order all pass sections are used when real roots (on the real axis) are to be compensated for; while, second order all pass filter sections are used when complex conjugate roots (located anywhere in the ‘s’ plane except the real axis) are to be compensated for. Any higher order all pass filter section can be synthesized as a cascade of one or more first and second order all pass filter structures.

Thus every realizable transfer function with zeros on the '*jw*' axis can be synthesized in the form of reactive ladders terminated in 1Ω resistances. This can be represented as shown in Figure 3.5.

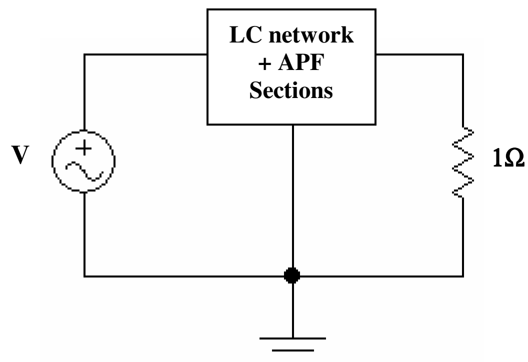


Figure 3.5. Single resistively terminated LC ladder network

Once the odd and even parts of the impulse response have been realized separately, they must be summed. This can be done using operational amplifiers (op-amps) as shown in Figure 3.6.

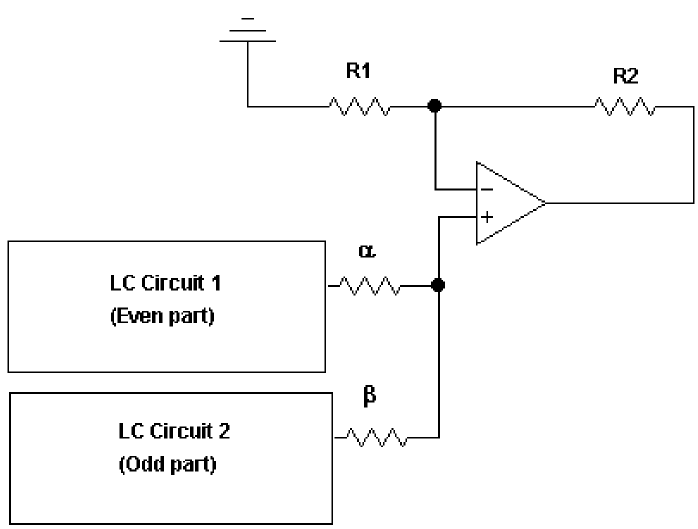


Figure 3.6 Connecting even and odd parts using an op-amp

The synthesis of a single resistively terminated LC ladder can be represented by a flowchart as shown in Figure 3.7

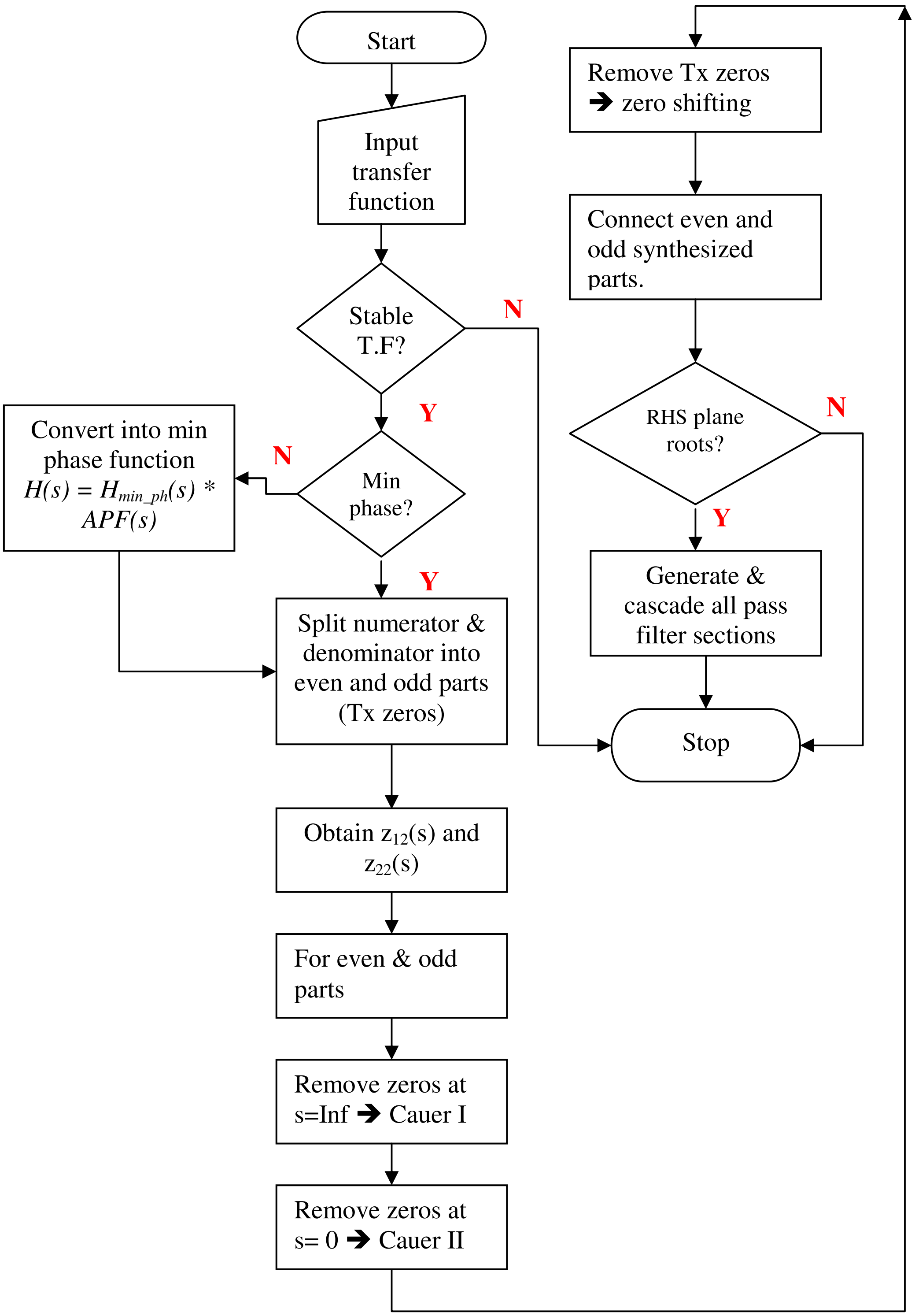


Figure 3.7 Flowchart for the synthesis of a single resistively terminated LC ladder network

3.4. Synthesis of doubly resistively terminated lossless ladder networks

Single resistively terminated networks are rarely used in practice mainly because of their high sensitivity to variation in component tolerances [3]. Another drawback of the single resistively terminated networks is resistive loading, because the network is always loaded by the element connected to it. If however the effective loading is lumped into the terminal resistance, the performance of the network is not affected [13]**.** In general a double resistively terminated reactance two port network (Figure 3.8) is the most widely used (and generally preferred) synthesis method. The reasons for this are its ability to produce any type of loss response and an optimal LC ladder realization with maximum power transfer resulting in low sensitivity to component variations [5].

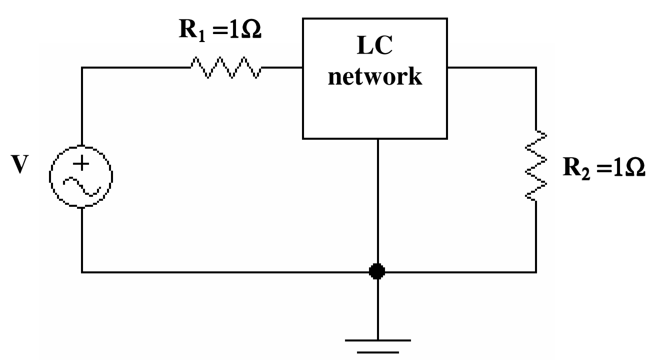


Figure 3.8 Doubly resistively terminated LC ladder network

According to Orchard [1,11], zero sensitivity of the loss to component variations (in the passband) can be achieved when a double resistively terminated ladder network is designed such that there is maximum power transfer at frequencies of minimum loss in the network. This property is exclusive to the double resistively terminated network which is why they are preferred over other types of networks.

The approach to finding a doubly terminated ladder network is a little different and more complicated as compared to the single terminated network [16]. The main difference lies in finding the driving point impedance function from the input transfer function. Once z_{11} and z_{22} have been determined, the network is synthesized with the zeros of transmission using the Cauer method described earlier.

The synthesis begins by assuming that the input transfer function is the voltage gain function, $T(s) = \dfrac{V_o(s)}{V_{in}(s)}$.

The transducer function *H(s)* is obtained by

$$H(jw) = \left(\sqrt{\frac{R_2}{4R_1}}\right) \cdot \frac{V_{in}(jw)}{V_o(jw)}$$

In order that this LC ladder realization have low sensitivity, *H(s)* must be scaled such that $20\log_{10}|H(jw)| = 0$ at the frequencies of loss minima. Next, the characteristic function *K(jw)* is evaluated [11].

$$|K(jw)|^2 = |H(jw)|^2 - 1$$

The transducer function is proportional to the loss of the network and the characteristic function indicates how close this loss is to 1. Now that we have *H(s)* and *K(s)*, the driving point impedance functions can be evaluated as follows [11].

$$z_{11} = R_1\left(\frac{H_{even}(s) - K_{even}(s)}{H_{odd}(s) - K_{odd}(s)}\right) \qquad z_{22} = R_2\left(\frac{H_{even}(s) + K_{even}(s)}{H_{odd}(s) - K_{odd}(s)}\right)$$

or alternatively,

$$z_{11} = R_1\left(\frac{H_{odd}(s) - K_{odd}(s)}{H_{even}(s) - K_{even}(s)}\right) \qquad z_{22} = R_2\left(\frac{H_{odd}(s) + K_{odd}(s)}{H_{even}(s) - K_{even}(s)}\right)$$

Along with the restriction imposed on transfer functions in section 0, it must also be noted that both *H(s)* and *K(s)* should have loss poles which lie exclusively on the imaginary axis.

One of the main problems faced in the synthesis of the double resistively terminated ladder networks is described for the particular transfer function that we have used. In the synthesis method, the transfer function was split into even and odd parts so that the transmission zeros are on the imaginary (*jw*) axis – a condition required for synthesis using lossless components. However, it was observed that this method produced accurate results only when the transfer functions either had maxima occurring at DC (0 Hz) or had zeros at the origin or at infinity. While the even part of our transfer function had a low pass response, the odd part of the transfer function did not. A solution was obtained when the odd part of the transfer function was further split into two parts such that we had a cascade of an even part and a simple transfer function with a zero at the origin of the form $\frac{s}{s^2 + as + b}$. Using Matthaei's procedure [9], it is possible to build a circuit that sees a 1Ω source and load and hence does not require a buffer. In this case, we use Matthaei's method to synthesize the simple function and this can be represented in the form of an *'L section'* composed of Z_a and Y_b as shown below in Figure 3.9. Multiple such sections can be formed (with different Z_a and Y_b) and cascaded together with a single source resistance and a single load resistance.

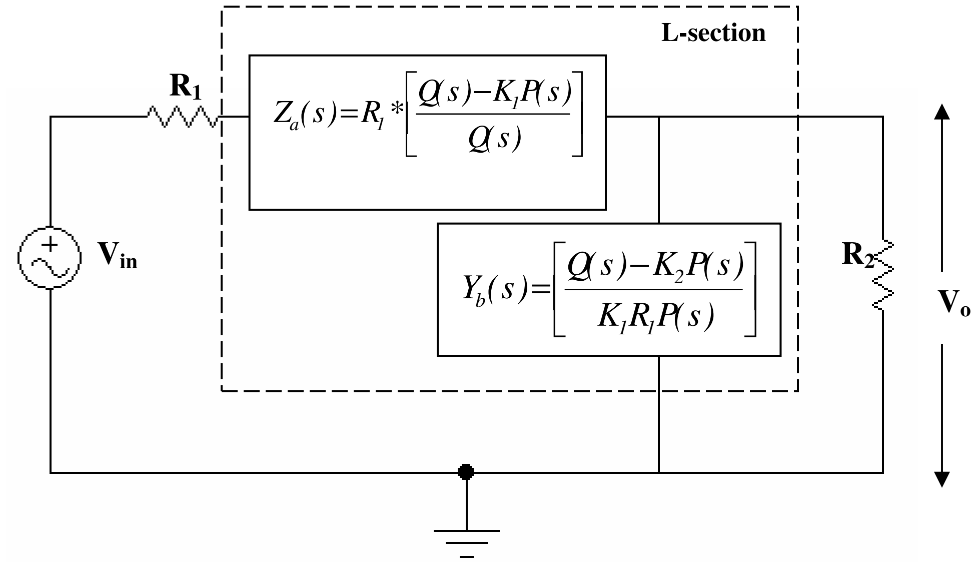


Figure 3.9 The basic L-section used in Matthaei's synthesis procedure

In our case, since this is being done for only the simple function of the odd part of the transfer function, synthesis by Matthaei's method becomes simpler. In this case we have,

$$\frac{V_o(s)}{V_{in}(s)}=\frac{P(s)}{Q(s)}=\frac{s}{s^2+as+b}$$

This function is then tested to ensure that it satisfies the conditions,

$$Re\left[\frac{P(jw)}{Q(jw)}\right]_{min}>0 \quad and \quad Re\left[\frac{Q(jw)}{P(jw)}\right]_{max}>0$$

The first equation formed that needs to be evaluated is $Z_a(s)=R_1\left[\frac{Q(s)-K_1P(s)}{Q(s)}\right]$. The constant K_1 must be chosen such that Z_a is a realizable driving point function. This means that the smallest degree of the numerator and denominator should differ by 1. The second equation evaluated is,

$$Y_b(s)=\left[\frac{Q(s)-K_2P(s)}{K_1R_1P(s)}\right]$$

K_2 is defined by the relation $K_2 = K_1 \frac{R_1}{R_2}$.

With the two functions Z_a and Y_b determined, a continued fraction expansion can be carried out to obtain the values of inductance, capacitance and resistance that should be connected in the series and shunt arms of the circuit.

In all examples that have been synthesized in the next chapter, the load and source resistances are 1Ω, normalized to a frequency of 1Hz. To implement circuits with different termination resistances and cut off frequencies, impedance and frequency scaling are necessary. This is performed as follows.

Impedance scale factor ➔ $\alpha_I = \frac{R_{new}}{R_{old}}$

Frequency scale factor ➔ $\alpha_F = \frac{f_{new}}{f_{old}}$

$$L_{new} = \frac{L_{old}}{\alpha_F} \times \alpha_I$$

$$C_{new} = \frac{C_{old}}{\alpha_F \times \alpha_I}$$

Thus, in this chapter, the procedures for the synthesis of single and double resistively terminated lossless ladder networks have been outlined. The next chapter shows examples of how these were used with the help of our network synthesis application.

CHAPTER 4

RESULTS FOR THE SYNTHESIS OF TRANSFER FUNCTIONS

4.1. Software description

The software to synthesize a lossless ladder circuit from a transfer function consists of 4 functions (functions for the cauer1, cauer2, zero shifting procedure , all pass filter generation, and the main function).The theory behind each of these algorithms has been explained in section 3. Later, chapter 5 discusses the synthesis of single or double resistively terminated ladder circuits when inputs are in the form of samples of an impulse response. The main function does not take any parameters and is independent of the type of function to be synthesized or the format of the inputs.

When the main function (*netSyn*) is called, the user is asked if a transfer function or an impulse response is to be synthesized and accordingly the inputs are accepted. In the case of transfer function synthesis, the program allows for the inputs to be either in the form of poles and residues of the transfer function or as coefficients of the numerator and denominator of the transfer function. Once these inputs are accepted, the user has a choice of synthesis using single resistive termination or double resistive termination.

The software then goes through stages to ensure that the inputted system is stable and realizable. If there are poles in the right half s-plane, the program prints an error synthesis stops. On the other hand, if zeros are found to exist on the RHS of the s-plane (non- minimum phase function), it replaces them with the corresponding minimum phase function in cascade with all pass filter sections.

The procedures described in section 3.3 and section 3.4 are then followed to produce values of inductance and capacitance for the obtained circuit.

The application for transfer function synthesis runs in the MATLAB® environment. In this chapter we show examples of how the network synthesis strategies discussed in chapter 3 are implemented.

4.2. Single resistively terminated LC ladder networks

4.2.1. Simple Example

We show a simple example in which the system has simple roots located only at the origin and at infinity [13]. The transfer function in this case is,

$$H(s) = Z_{21} = \frac{s}{s^3 + 2s^2 + 2s + 1}$$

The circuit obtained from the application with inputs as per the transfer function *H(s)* is shown in Figure 4.1

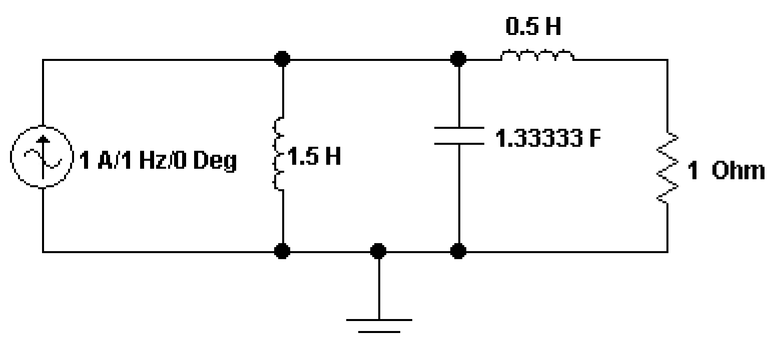


Figure 4.1 Circuit diagram using values obtained from application (Eg. 1)

A comparison of the ideal (from the transfer function) and obtained (using the synthesized circuit) frequency responses along with the error plot (using Electronic Workbench) is shown in Figure 4.2. It is evident that the error is very small (~10^{-6}). It is understood that this synthesis is exact and not an approximation. Hence all error must be attributable to round off errors.

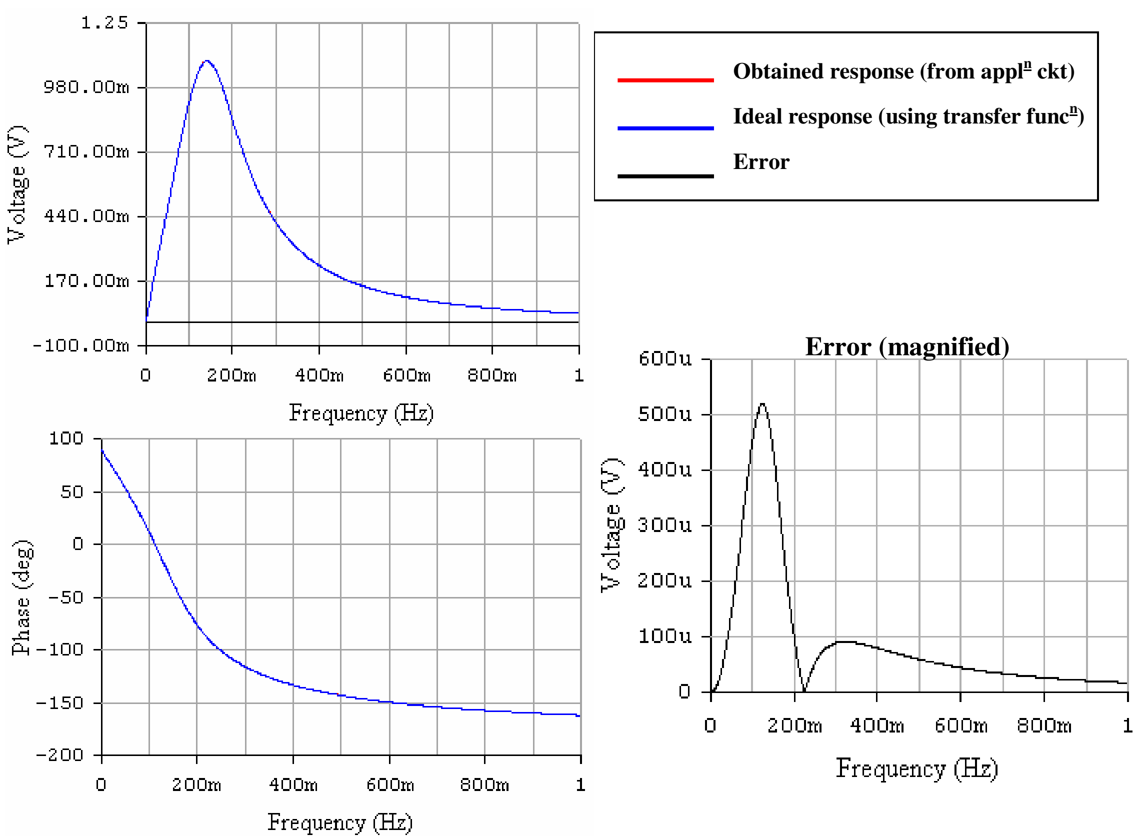


Figure 4.2 Obtained and ideal frequency responses with error plot (Eg. 1)

4.2.2. The Nyquist pulse

This example shows the single termination realization of a Nyquist pulse. [19] The inputs are the poles and residues and the output circuit is shown in Figure 4.3. As seen in the figure, the even and odd sections each have to be scaled by a different factor (obtained from the program). A comparison of the ideal and obtained AC frequency response at the output along with the corresponding error between the two responses is shown in Figure 4.4. Finally Figure 4.5 shows the transient responses along with the error plot

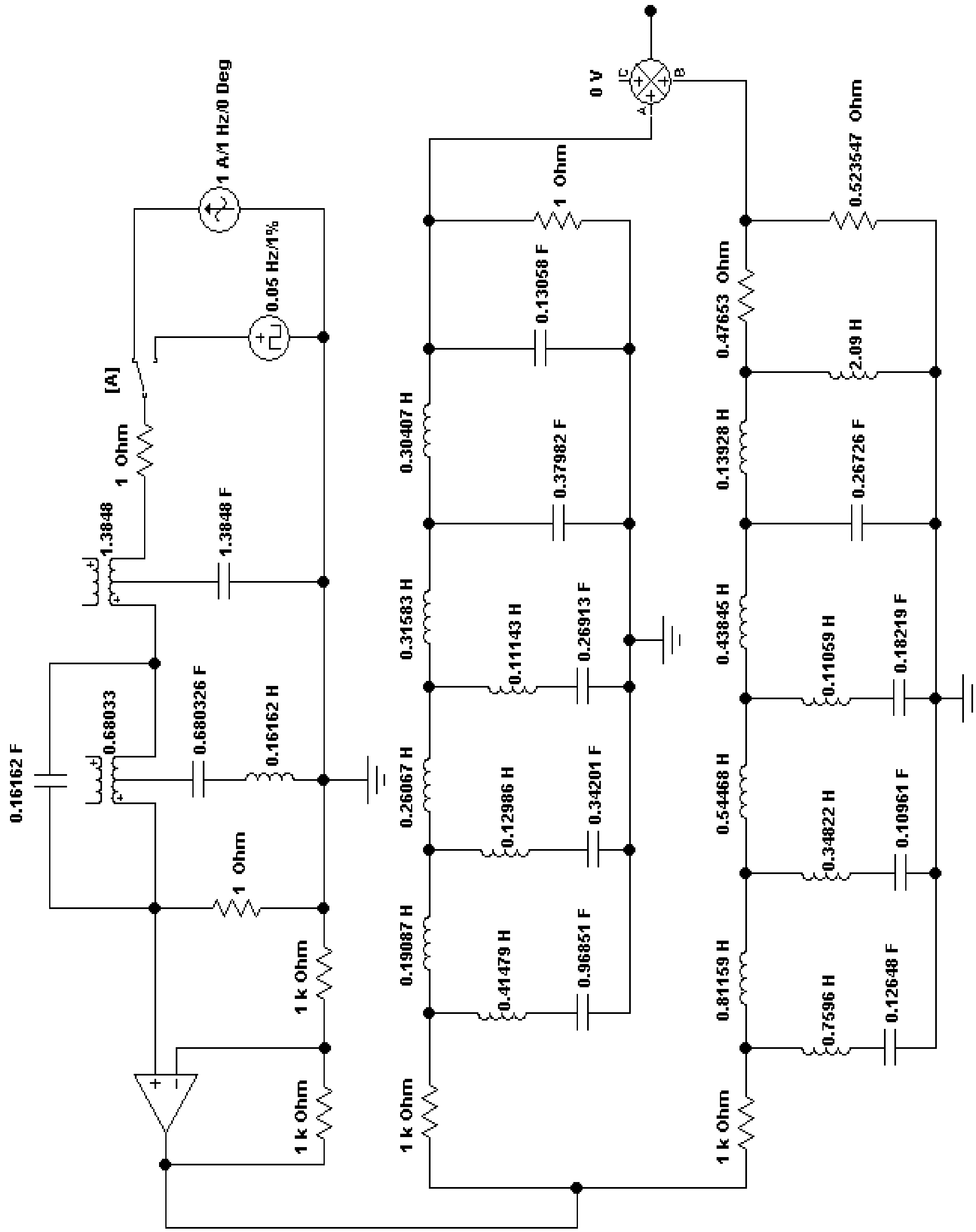


Figure 4.3 Circuit diagram for Nyquist pulse synthesis (obtained from application)

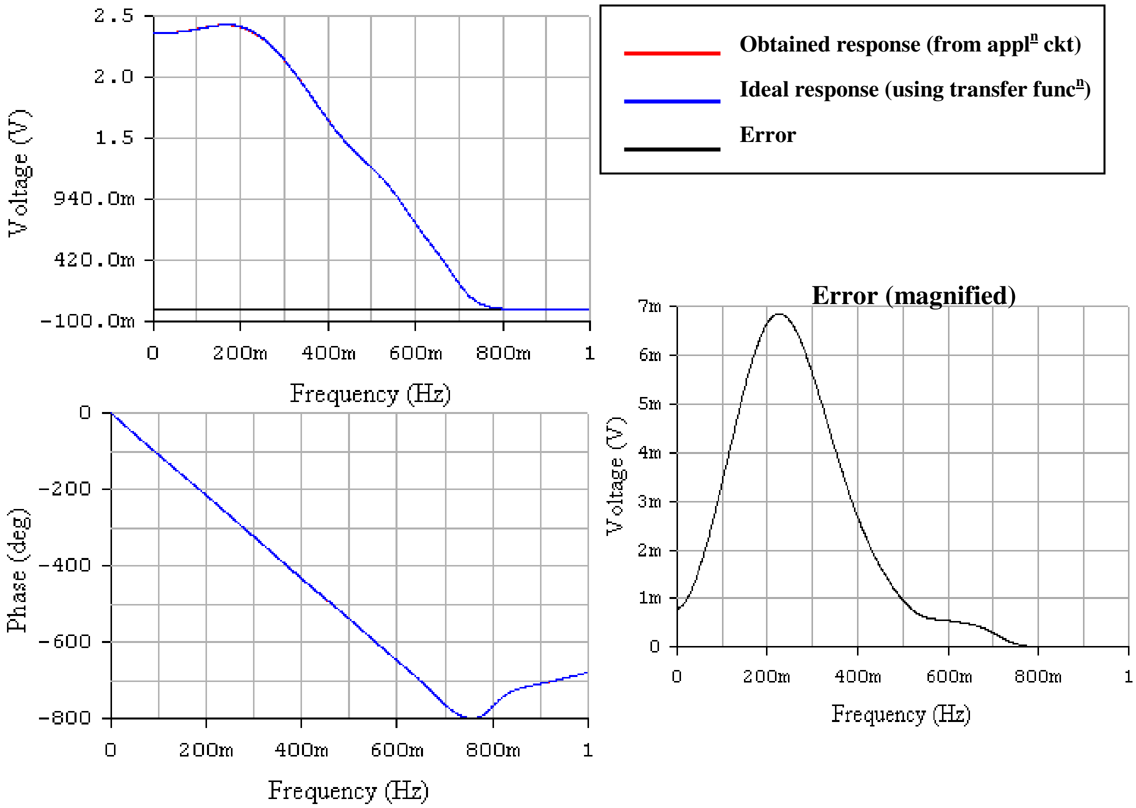


Figure 4.4 Superimposition of ideal and amplified circuit response (left) and error plot (right)

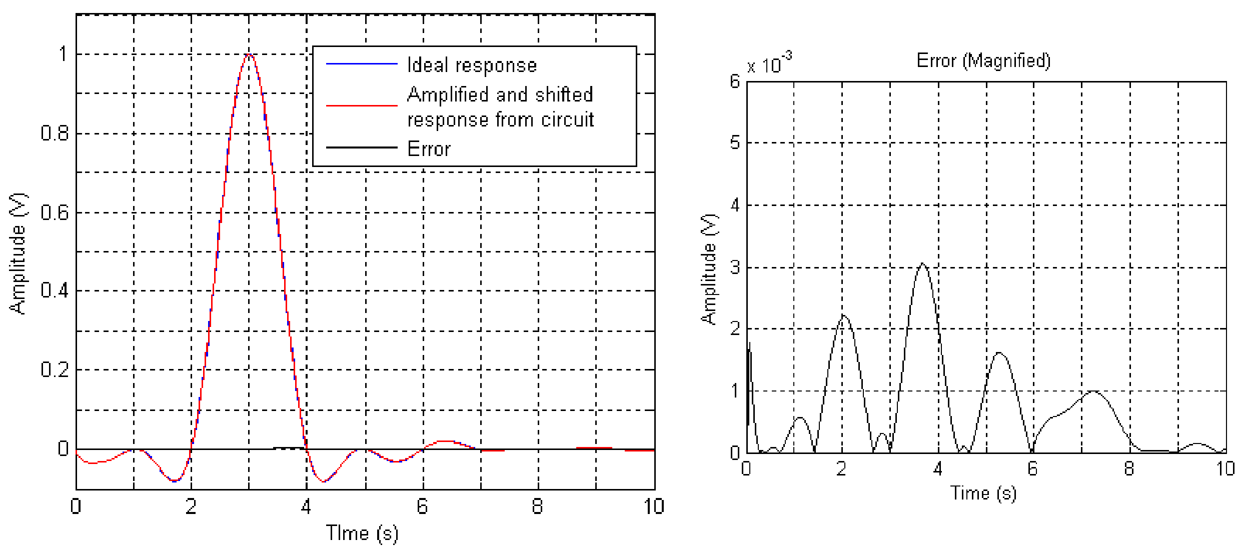


Figure 4.5 Superimposition of the ideal and obtained transient responses with the error

4.2.3. Square root raised cosine pulse

This example shows the single termination realization of a square root raised cosine pulse [19]. The output circuit is shown in Figure 4.6. The even and odd sections have been scaled separately and added by means of a summing element thereby avoiding the use of op-amps. A comparison of the ideal and obtained AC frequency response at the output is shown in Figure 4.7 along with the error between the two responses after the output of the circuit has been amplified. Finally Figure 4.8 shows the transient responses along with the error plot.

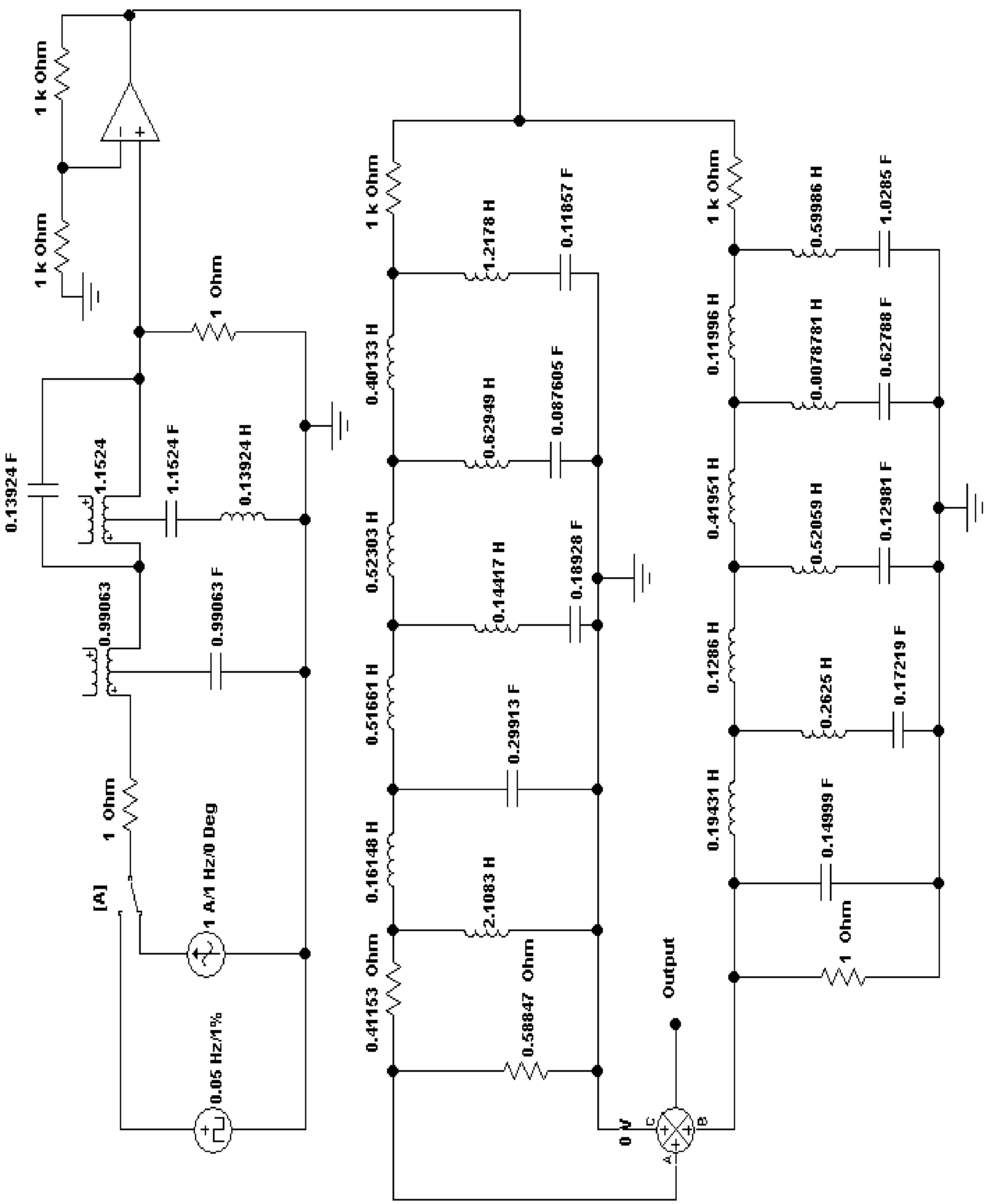


Figure 4.6 Single resistively terminated square root raised cosine pulse (obtained from application)

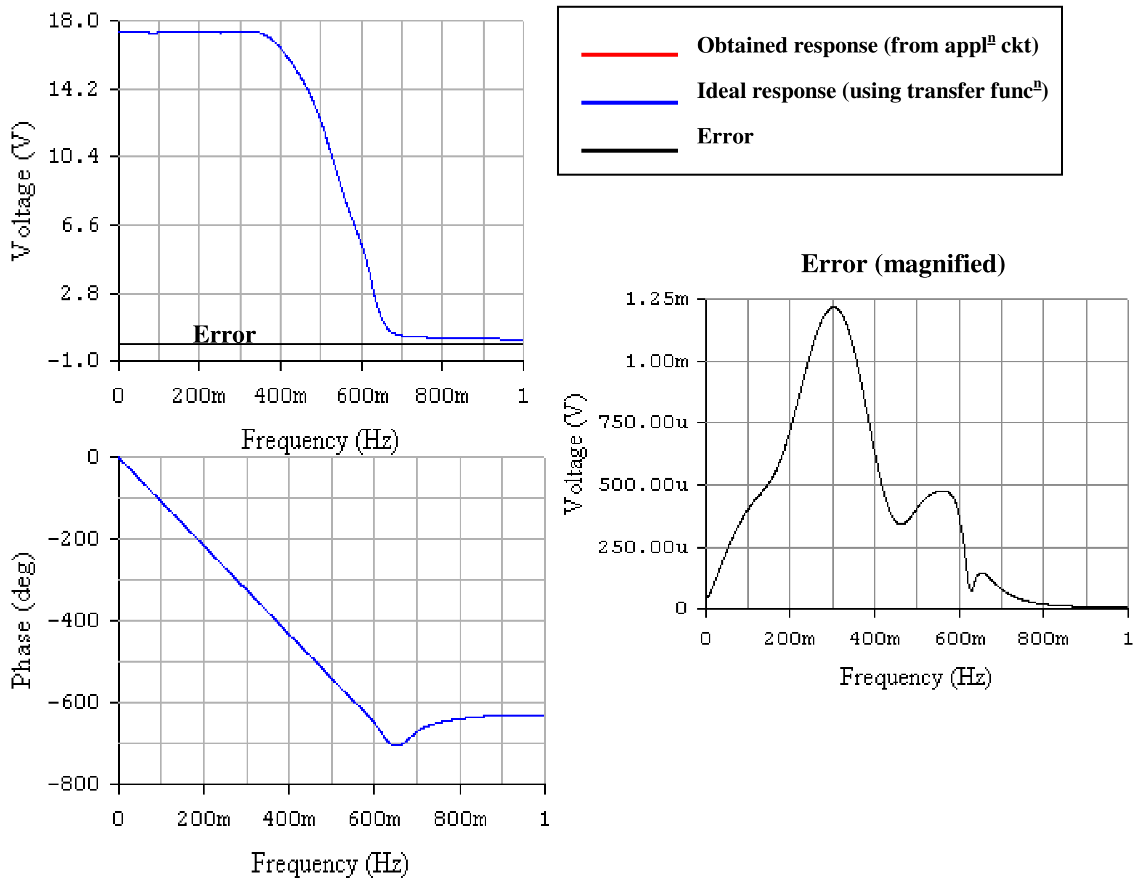


Figure 4.7 Superimposition of ideal and amplified circuit response (left) and error plot (right)

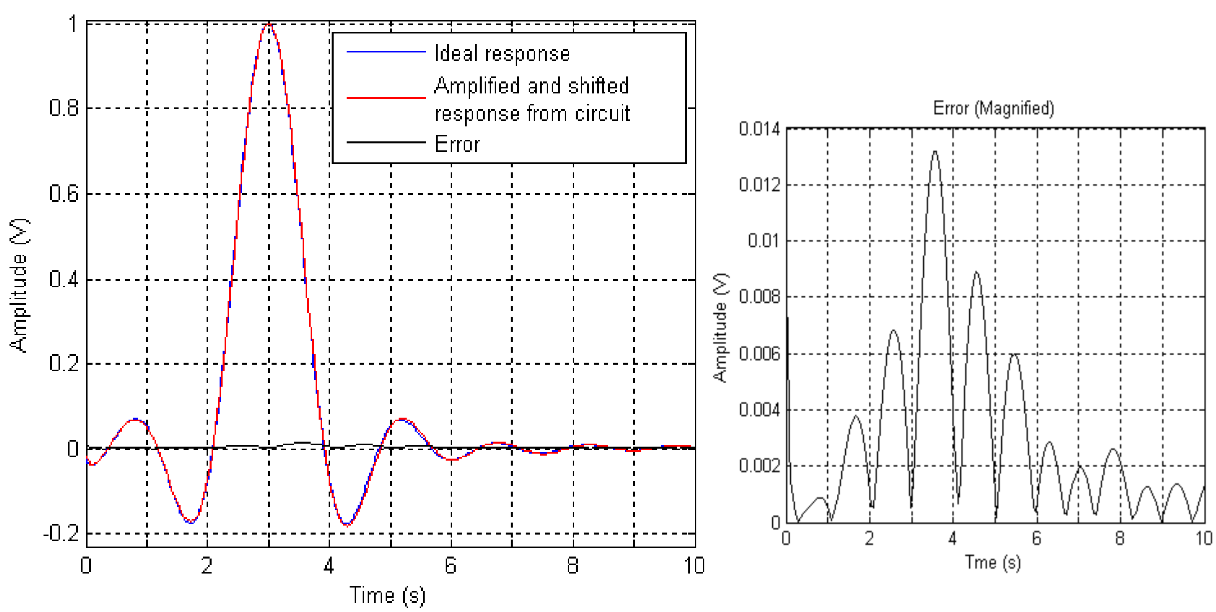


Figure 4.8 Superimposition of the ideal and obtained transient responses with the error

4.2.4. Butterworth filter

We also performed a comparison to test the component values obtained from this application against tabulated values from literature [20]. The equation form for the Butterworth filter is as follows

$$\left|Z(jw)\right|^2 = \frac{1}{1+\omega^{2N}} \quad \textit{where, N is the order of the system}$$

The circuit form is as shown in Figure 4.9

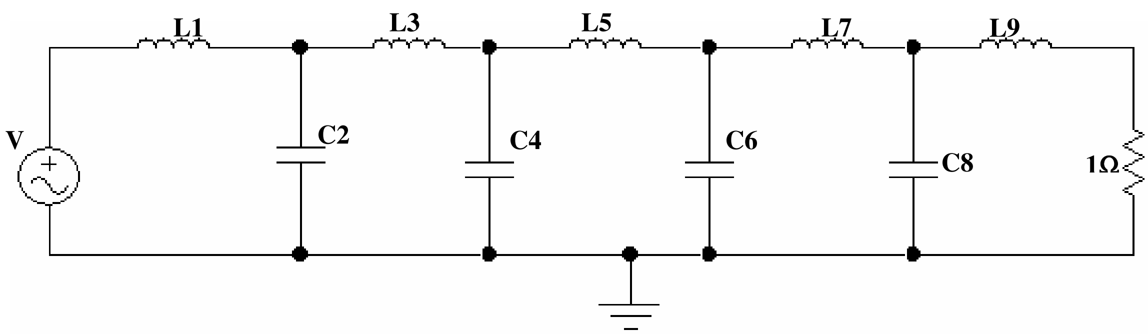


Figure 4.9 General form of a single resistively terminated Butterworth filter

The values tabulated for the Butterworth case are shown in Table 4.1. It can be observed that the values obtained using the application match closely with the values obtained from literature. In Table 4.1, *'prg'* refers to the values obtained from the synthesis application while *'tab'* refers to the tabulated values obtained from literature.

Table 4.1 Comparison of obtained and tabulated component values for Butterworth filter

N	L_1	C_2	L_3	C_4	L_5	C_6	L_7	C_8	L_9
2(prg)	0.7071	1.414							
2(tab)	0.7071	1.414							
4(prg)	0.3827	1.082	1.577	1.531					
4(tab)	0.3827	1.802	1.577	1.531					
7(prg)	0.2225	0.6560	1.055	1.397	1.659	1.799	1.588		
7(tab)	0.2225	0.6560	1.054	1.397	1.659	1.799	1.588		
9(prg)	0.1736	0.5155	0.8414	1.141	1.404	1.620	1.777	1.842	1.563
9(tab)	0.1736	0.5155	0.8414	1.141	1.404	1.620	1.777	1.842	1.563

The evaluation of error in these examples shows that the synthesized circuit reproduces the desired transfer function. The results obtained from our application are in excellent agreement with those obtained from literature for the Butterworth filter. Now examples of the synthesis of double resistively terminated lossless ladder networks are shown.

4.3. Double resistively terminated LC ladder networks

4.3.1. Synthesis of a low pass filter with finite transmission zeros.

Consider the following transfer function [11],

$$T(s) = \frac{(s^2 + 3.476896154)\cdot(s^2 + 8.2227391422)}{\left[\begin{array}{l}55.3858 * (s + 0.60913)\cdot(s^2 + 0.263147s + 1.166357185)\cdot \\ (s^2 + 0.85422659s + 0.7269594794)\end{array}\right]}$$

The circuit obtained using our application is shown in Figure 4.10. A comparison of the ideal and obtained frequency responses along with a plot of the difference between the two is shown in Figure 4.11

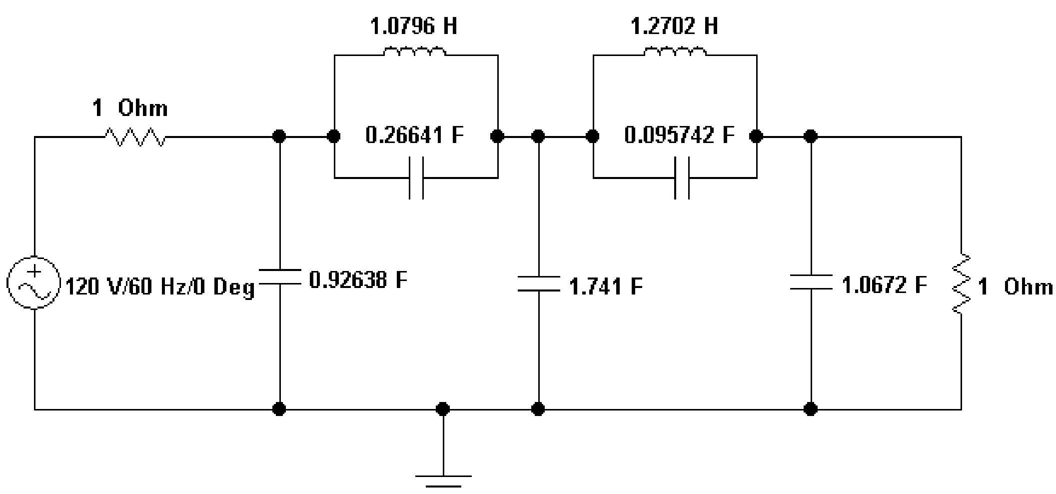


Figure 4.10 Circuit diagram using values obtained from application

(Example 1)

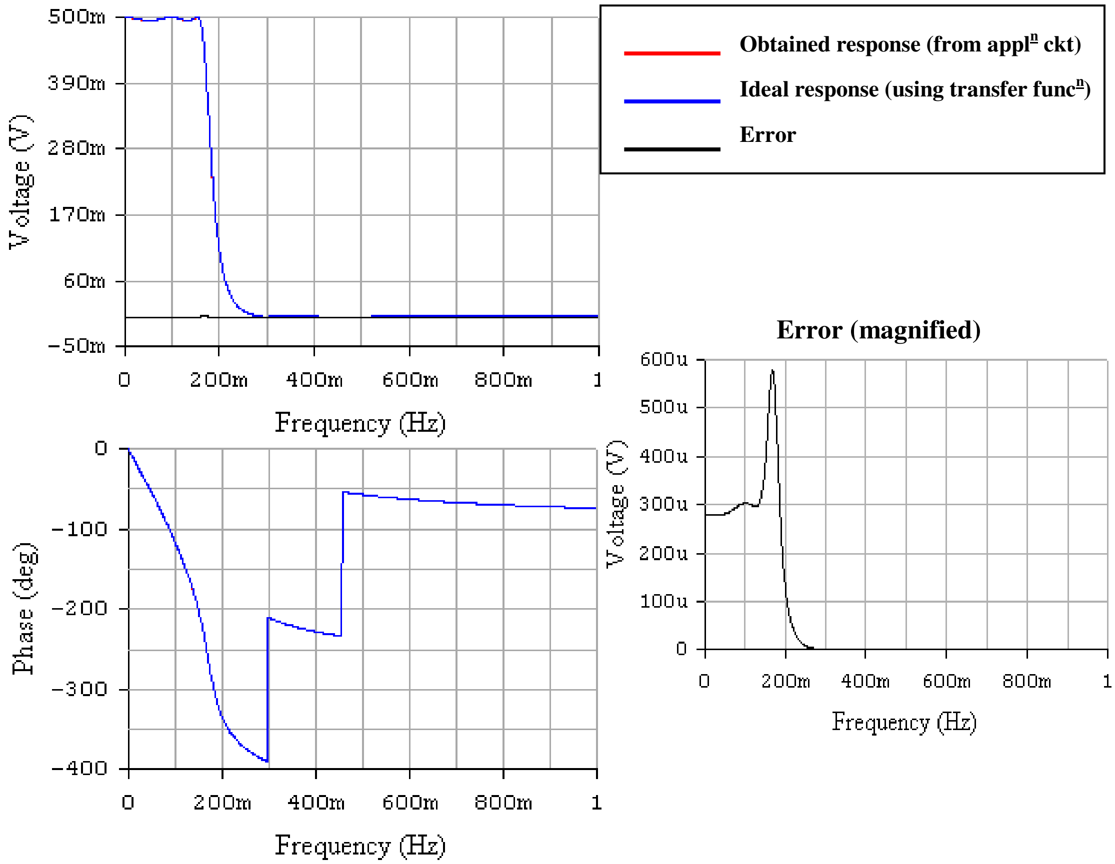


Figure 4.11 Ideal and Obtained frequency responses with error plot (Eg. 1)

4.3.2. The Nyquist pulse

In this example the Nyquist pulse [19] was synthesized to obtain a double resistively terminated LC ladder network. Figure 4.12 shows the circuit whose values were obtained using our filter synthesis application. The circuit is split into three parts – an even section, an odd section and the all pass filter section to compensate for the right hand s-plane zeros. The ideal and obtained AC frequency responses at the output were compared and the results along with an error plot are shown Figure 4.13 .The transient responses are compared in Figure 4.14.

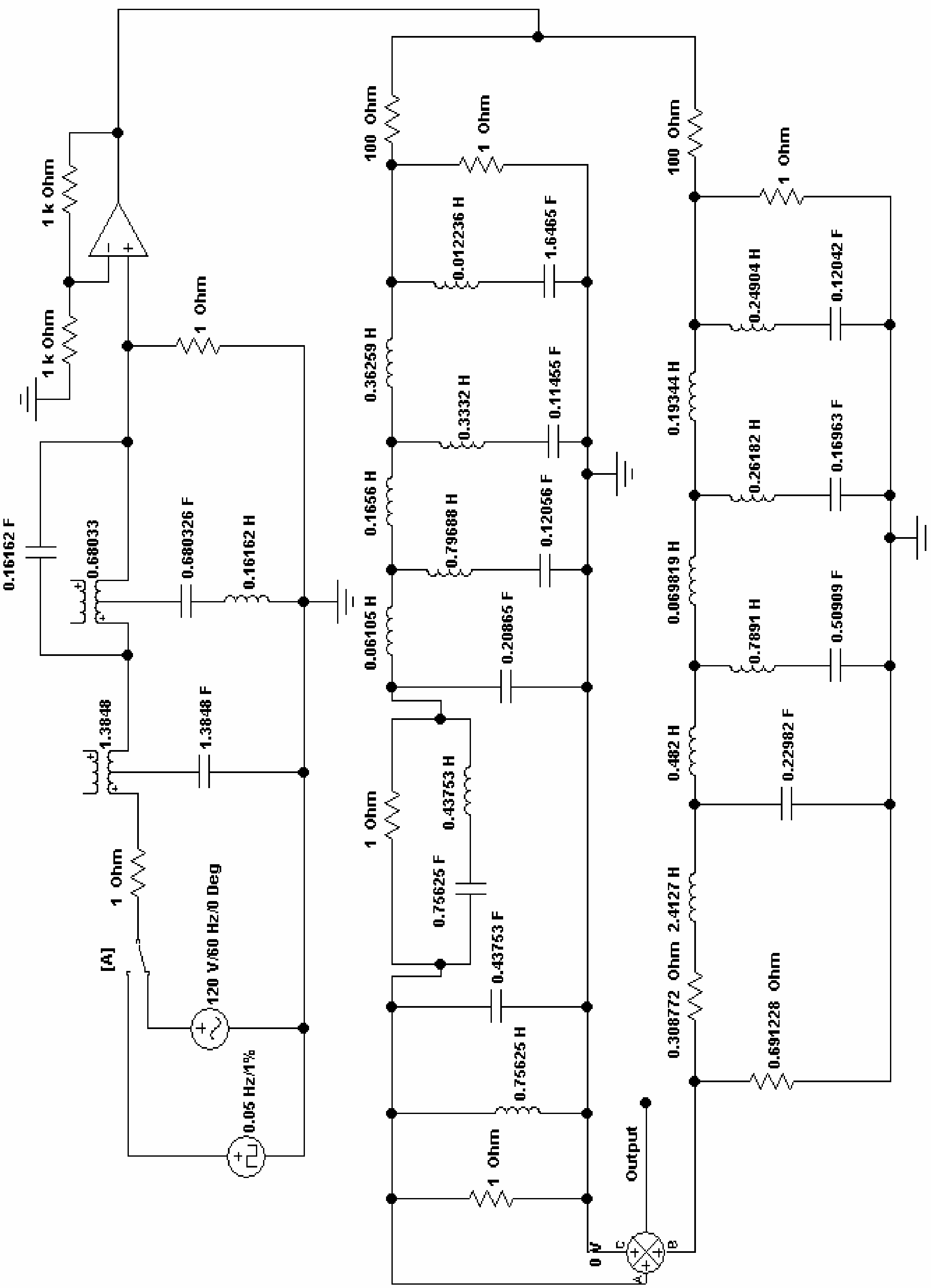


Figure 4.12 Double resistively terminated LC ladder realization for a Nyquist pulse

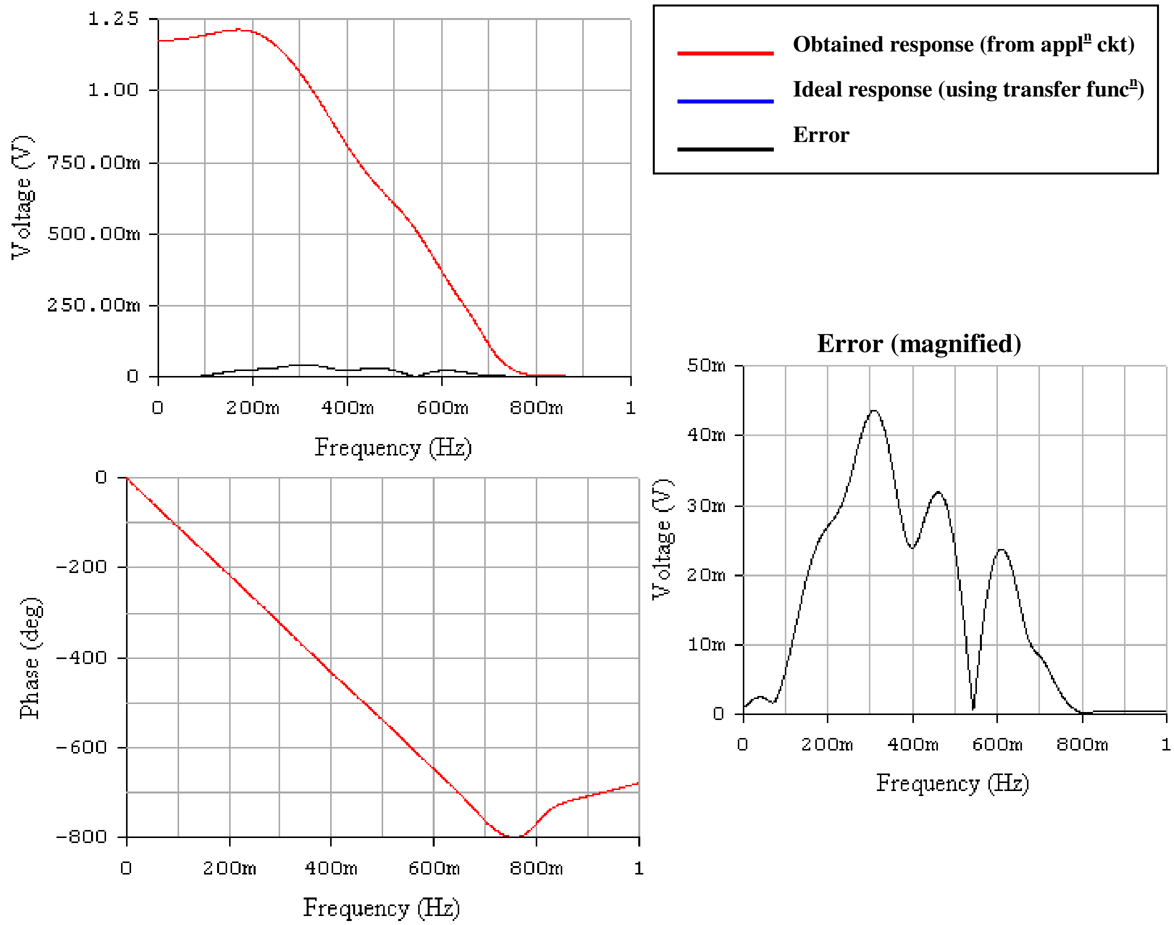


Figure 4.13 Superimposition of ideal and amplified circuit response (left) and error plot (right)

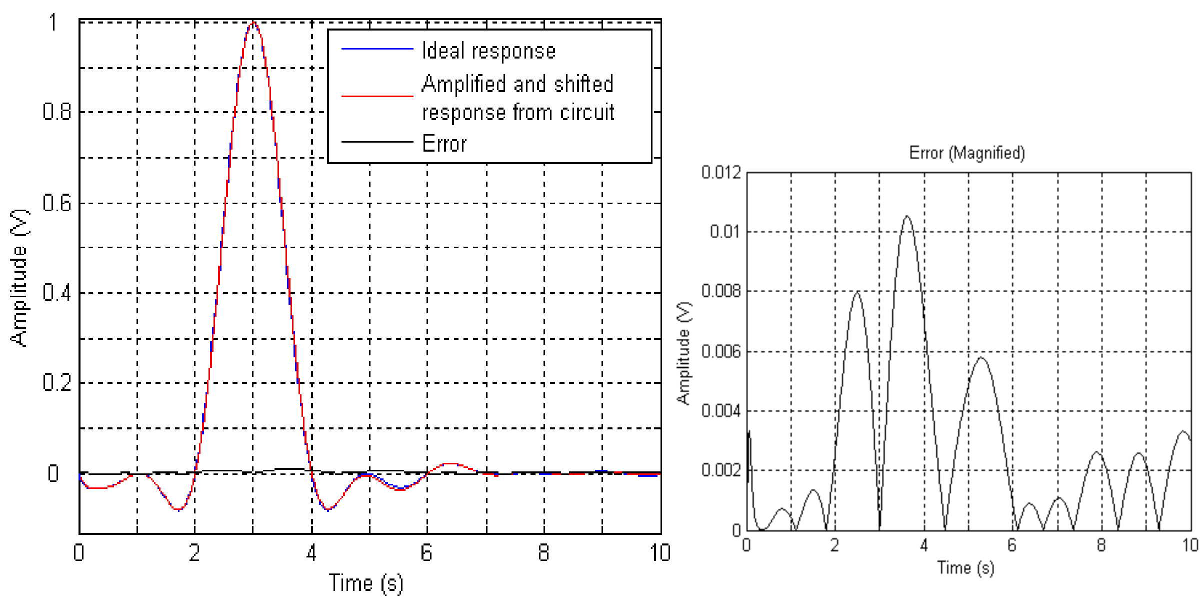


Figure 4.14 Superimposition of the ideal and obtained transient responses with the error

4.3.3. Prolate Spheroidal Wave Function

In this example a 9th order prolate spheroidal wave function (PSWF) was synthesized with a delay of 5 seconds to obtain a double resistively terminated LC ladder network. Figure 4.15 shows the circuit whose values were obtained using our filter synthesis application. The circuit is split into three parts – an even section, an odd section and the all pass filter section to compensate for the right hand s-plane zeros. The ideal and obtained transient responses along with a plot of the error are shown in Figure 4.16.

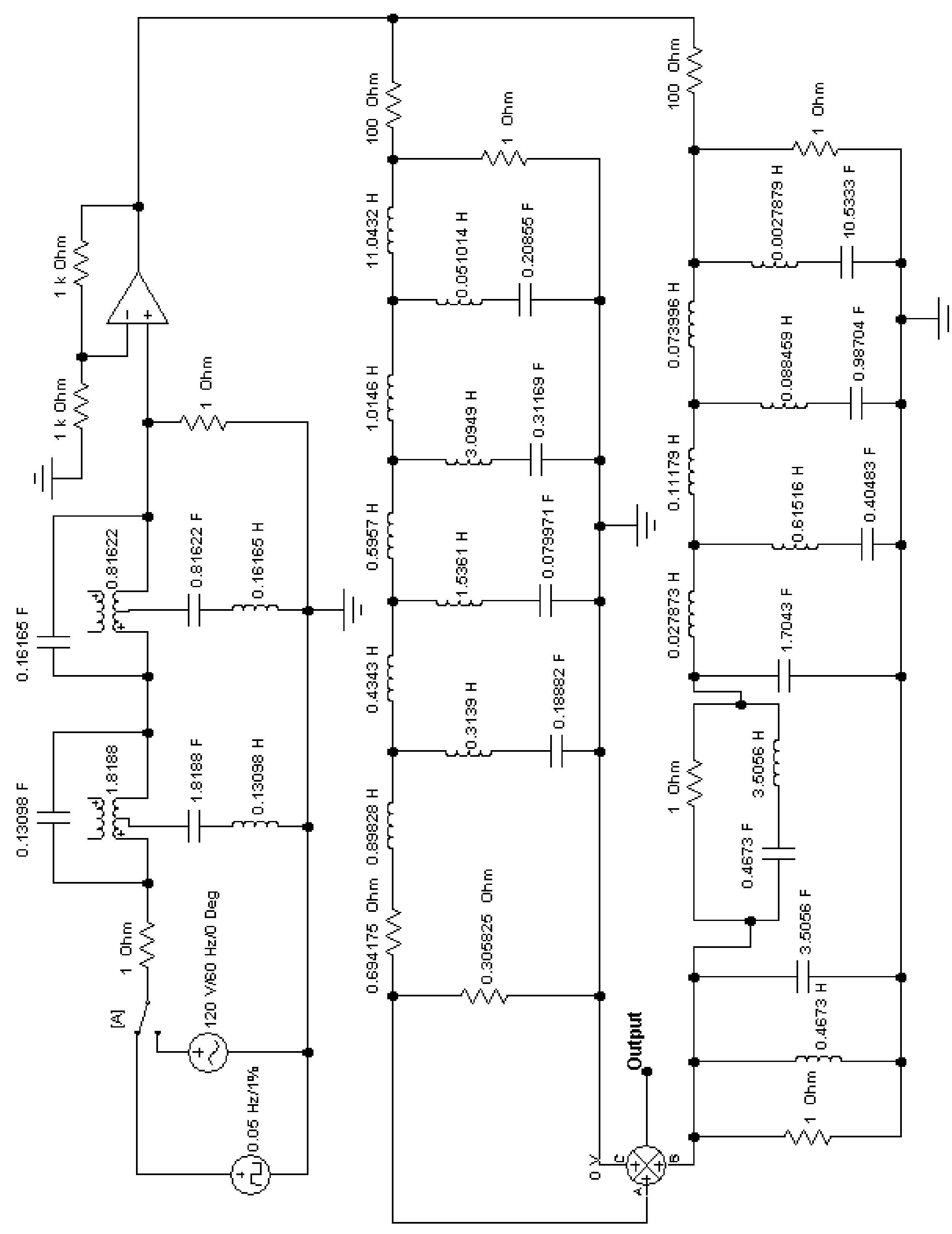


Figure 4.15 Double resistively terminated ladder representation for a 9th order PSWF

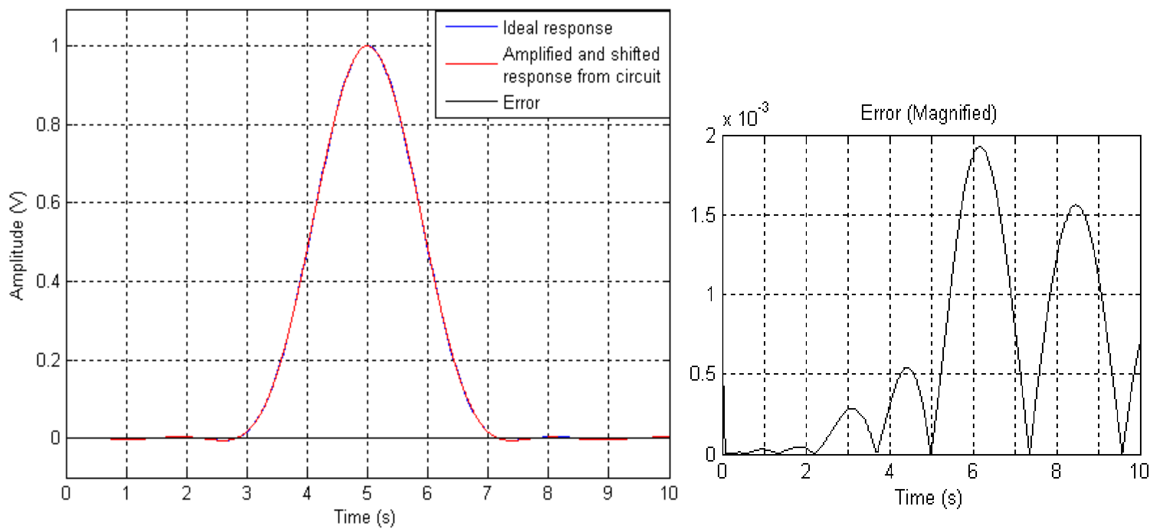


Figure 4.16 Superimposition of the ideal and obtained transient responses with the error (PSWF)

In summary, this chapter presents results obtained from our application for the synthesis of single and double resistively terminated ladder networks. It has been demonstrated that results obtained via the program presented in this thesis match very closely with the ideal response. The small error which is observed is most likely due to round off and truncation errors which arise during the process of calculations. Excellent agreement with literature in the case of the Butterworth filter was demonstrated.

CHAPTER 5

IMPULSE RESPONSE SYNTHESIS

In the areas of communication and signal processing, it is sometimes necessary to be able to synthesize impulse responses; for example in the synthesis of finite impulse response (FIR) and infinite impulse response (IIR) filters. We have found that this issue of impulse response synthesis is not addressed in the currently available software applications. The design and synthesis of impulse responses is done by creating a front end application for the Prony's method.

Prony's method is a method used to obtain a set of poles and residues from an impulse response. Given an impulse response *h(t)* we uniformly sample the signal at *2N* points to form the sample matrix and the sample vector.

$$\begin{bmatrix} h(0) & h(1) & \dots & h(N-1) \\ h(1) & h(2) & \dots & h(N) \\ \vdots & \vdots & \vdots & \vdots \\ h(N-1) & h(N-2) & \dots & h(2N-2) \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix} = \begin{bmatrix} h(N) \\ h(N+1) \\ \vdots \\ h(2N-1) \end{bmatrix}$$

The above system of linear equations is solved, from which the pole locations are obtained.

To synthesize the system represented by these poles and residues, its impulse response must be stable. The stability of such discrete time impulse responses can be tested using the Jury test. It comprises of four conditions that the characteristic function must satisfy in order that the impulse response be stable. Since the Jury test is a necessary and sufficient condition for stability, if the impulse response fails even one of these tests, we can be sure that the response is unstable.

Although not all impulse responses are stable, some of them are stable within specific intervals. Such impulse responses can be made stable by delaying the signal by a certain amount say 'δ'. This can be represented in terms of a *'stability chart'*. Once the delay (up to a user specified delay granularity) that makes the impulse response stable is found, the user will be notified of the modification to the input impulse response after which it will be synthesized. The flow of this program is further explained by the flowchart in Section 5.4 (Figure 5.1)

5.1. Prony's method

Prony's method is an algorithm for finding an IIR filter with a given time domain impulse response. The impulse response of a circuit can be obtained if the poles of the system in the *s*-plane and their corresponding residues are known. The impulse response can then be written as a summation of the residues, multiplied by exponentially damped functions. If '*N*' is the order of the system (number of poles in the system) then, the impulse response can be represented in the form [2, 17],

$$h(t) = \sum_{m=1}^{N} A_m e^{s_m t} = \sum_{m=1}^{N} \frac{A_m}{s - s_m}$$

Eq. 5.1

However, our impulse response is rarely in the continuous domain. Rather, it is in the form of a sampled signal. This sampled data can be expressed in the form of a linear combination of exponentials.

Hence the equation 5.1 can be modified as

$$h_{sampled}(n\Delta t) = \sum_{m=1}^{N} A_m e^{s_m(n\Delta t)} \quad n = 0,1\cdots 2N-1 \qquad \textbf{Eq. 5.2}$$

where,

A_m : Residues of the N^{th} order system.

s_m : Poles of the N^{th} order system.

Δt : Sampling interval.

N : Order of the system.

These equations represented in equation 5.2 are a set of *2N* non linear equations with *2N* unknowns. For any N^{th} order system, the first *2N* samples are independent and hence using Prony's method, the poles and residues for an N^{th} order system can be recovered. The equations also imply that the sampled impulse response can be reconverted into a continuous signal in the time domain using the first equality in equations 5.1 and can be represented in the frequency domain using the second equality in the same equation. This algorithm is also useful since it allows time domain to frequency domain interconversion, without the use of the Fourier transform, as shown below,

$$h_{sampled}(\Delta t) = \sum_{m=1}^{N} A_m e^{s_n(\Delta t)} \longleftrightarrow H(s) = \sum_{m=1}^{N} \frac{A_m}{(s - s_m)}$$

This impulse response is first converted into matrix form to find the poles,

$$\begin{bmatrix} h_{sampled}(0) & h_{sampled}(1) & \ldots & h_{sampled}(N-1) \\ h_{sampled}(1) & h_{sampled}(2) & \ldots & h_{sampled}(N) \\ \vdots & \vdots & \ldots & \vdots \\ h_{sampled}(N-1) & h_{sampled}(N-2) & \ldots & h_{sampled}(2N-2) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix} = \begin{bmatrix} h_{sampled}(N) \\ h_{sampled}(N+1) \\ \vdots \\ h_{sampled}(2N-1) \end{bmatrix}$$

Eq. 5.3

Equation 5.3 can also be represented in the form *Ax*=*B*. The matrix *A*, has a Toeplitz structure and a unique solution to equation 5.3 can be found for the vector $\begin{bmatrix} a_0 & a_1 & \dots & a_{N-1} \end{bmatrix}^T$. The characteristic equation is formed as,

$$B(\omega) = \omega^N - a_{N-1}\omega^{N-1} - \dots\dots - a_1\omega - a_0 = 0\,. \qquad \textbf{Eq. 5.4}$$

Equation 5.4 is then solved to get the *N* roots of the system$(w_N, w_{N-1}, \dots\dots, w_2, w_1)$ and the poles of the system (s_n) are obtained using,

$$s_n = \left(\frac{1}{\Delta t}\right)\cdot \log_e(\omega_n) \qquad \textbf{Eq. 5.5}$$

Now that the poles are known, the residues of this Nth order system can be found by making use of equation 5.2 in a similar fashion by solving a system of linear equations.

$$h_{sampled}(r) = \sum_{m=1}^{N}(A_m \cdot \omega_m^r) \qquad \text{for } r = 0,1 \dots N-1$$

Representing this as a system of N linear equations (in matrix form)

$$\begin{bmatrix} h_{sampled}(0) \\ h_{sampled}(1) \\ \vdots \\ h_{sampled}(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \omega_1 & \omega_2 & \dots & \omega_N \\ \vdots & \vdots & \dots & \vdots \\ \omega_1^{N-1} & \omega_2^{N-1} & \dots & \omega_N^{N-1} \end{bmatrix} * \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix} \qquad \textbf{Eq. 5.6}$$

This matrix is of the form *B* = *Ax* and the matrix *A*, is a Vandermonde matrix. A unique solution can be found for the vector of residues $\begin{bmatrix} A_0 & A_1 & \dots & A_{N-1} \end{bmatrix}^T$. Now that the poles and residues are known the impulse response is obtained in the frequency domain and can now be synthesized.

Thus to extract the poles and residues of an N^{th} order system using Prony's method, atleast *2N* samples are required. The algorithm involves the solution of two N^{th} order linear equations – one to obtain the poles and the other to obtain the residues. Once the poles and residues are known, the system transfer function can easily be determined using equation 5.1. Prony's method gives an approximate transfer function from the sampled transient response. For large order systems, the order of the transfer function produced by this method is generally much smaller than the actual order of the system. One drawback however is that Prony's method relies strongly on linearization and on the assumption of noiseless data. Hence, this might pose a problem for non-linear and/or noisy functions.

5.2. Jury Tests

The Jury test is a test that is generally used to test the stability of linear time invariant digital systems in the z- domain. The Jury test can also be used to test the stability of sampled data systems. It consists of a set of four criteria against which the characteristic equation of the system must be tested. The system is stable if and only if it passes all four conditions. In checking for stability these tests ensure that all the roots of the characteristic equation (in other words, the poles of the system) lie on or within the unit circle (region of stability in the z-domain).

Let the characteristic equation of the system be represented as

$$D = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 \qquad\qquad a_n > 0$$

then the four conditions for the Jury test are summarized as [18]

Test 1: The first and most important condition to be satisfied is that the coefficients of the characteristic equation be positive. In other words,

$$D(1) > 0 \qquad \text{for } z = 1 \qquad \textbf{Eq. 5.7}$$

Test 2:

$$(-1)^n D(-1) > 0 \qquad \text{for } z = -1 \qquad \textbf{Eq. 5.8}$$

Test 3:

$$|a_0| < a_n \qquad \textbf{Eq. 5.9}$$

Test 4 (Jury Array):

The Jury array is constructed as follows,

$$JuryArray = \begin{matrix} z^0 & z^1 & \cdots & z^{n-1} & z^n \\ a_0 & a_1 & \cdots & a_{n-1} & a_n \\ a_n & a_{n-1} & \cdots & a_1 & a_0 \\ b_0 & b_1 & \cdots & b_{n-1} & 0 \\ b_{n-1} & b_{n-2} & \cdots & b_0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_0 & m_1 & 0 & \cdots & 0 \\ m_1 & m_0 & 0 & \cdots & 0 \end{matrix}$$

where,

$$b_k = \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix} \qquad \forall k = 0,1 \cdots n-1$$

$$c_k = \begin{vmatrix} b_0 & b_{n-k-1} \\ b_{n-1} & b_k \end{vmatrix} \qquad \forall k = 0,1 \cdots n-2$$

and so on

The test conditions that must be satisfied are,

$$
\begin{aligned}
|b_0| &> |b_{n-1}| \\
|c_0| &> |c_{n-2}| \\
&\vdots \\
|m_0| &> |m_1|
\end{aligned}
\qquad \textbf{Eq. 5.10}
$$

Since equations 5.7 to 5.10 are a set of necessary and sufficient conditions, the system can be declared unstable as soon as any one test fails.

While checking for the stability of a system, a case might arise when the system has poles on the unit circle (the corresponding case in a continuous time system is the poles of the system lying on the imaginary axis). In this case the system is said to be marginally stable and $D(1) = 0$ and/or $D(-1) = 0$ (for the first and second Jury tests). This can be resolved, by removing the roots that lie on the unit circle ($z = 1$ and/or $z = -1$) and reconstructing the characteristic equation so as not to contain these roots and then checking for stability. The program to check for stability of digital systems using the Jury method can be found in the Appendix.

5.4. Synthesis of impulse responses (software description)

This program to synthesize impulse responses takes as inputs, samples from an impulse response. It also allows optional inputs such as the star and end sampling points and assumes uniform sampling. Prony's method (described in section 5.1) is used to generate a characteristic vector. The Jury test is then applied to the characteristic equation to test for stability.

If the original system is unstable but a delayed version of the system is found to be stable, the user is notified of this fact. Another thing to note about this program is that it is initially assumed that the order of the system (*N*) is half the number of the input samples (*2N).* However, to avoid cases where the user over-specifies the system order (For example, the actual order of the system may be 5, but the user may provide 20 samples and insist on a 10^{th} order system), the rank of the matrix generated is calculated. If the rank is less than the specified degree then, it means that the system order was too large to begin with and this is automatically changed and recalculated using the new order (rank).

Once a stable version of the response is found, the poles and residues are calculated from the impulse response and a single or double resistively terminated network can be synthesized as desired. On the other hand, if the system is unstable, a delay is added to the system until the system becomes stable. Then the shifted impulse response is synthesized. In addition, a 'stability chart' showing the delays for which the system is stable and unstable is plotted. The flowchart for this program is shown in Figure 5.1.

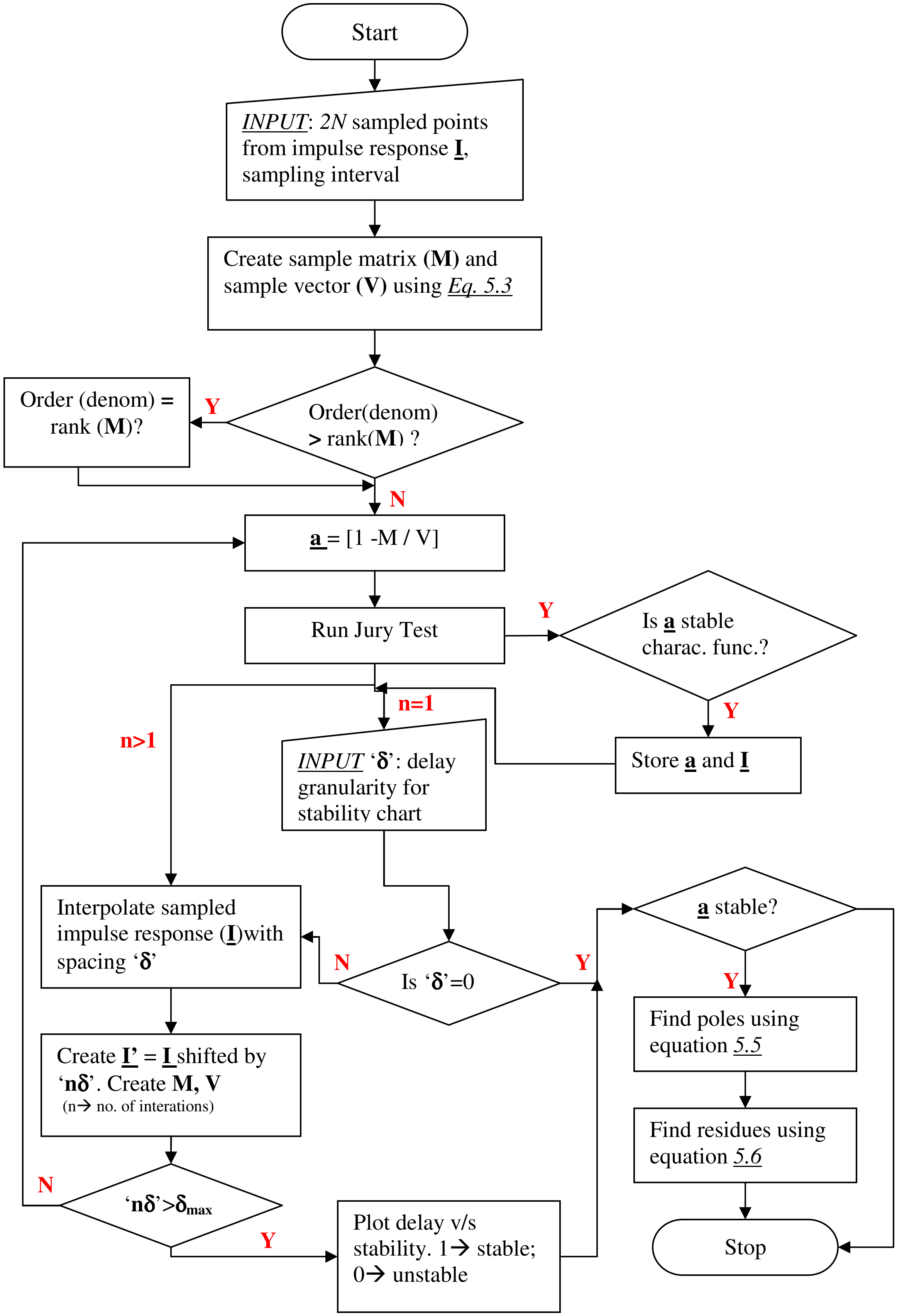


Figure 5.1 Flowchart depicting the synthesis of impulse responses

5.4. Stability as a function of delay

5.4.1. The Nyquist pulse

To test the stability of the Nyquist pulse, we took 18 samples starting from *t=0 sec* and at intervals of 0.5 sec. These samples were fed to the function. The delay was varied and a graph indicating the variation in stability in different intervals was plotted. From Figure 5.2, it is evident that by varying the delay of the system it is possible to make the system stable and unstable within certain intervals.

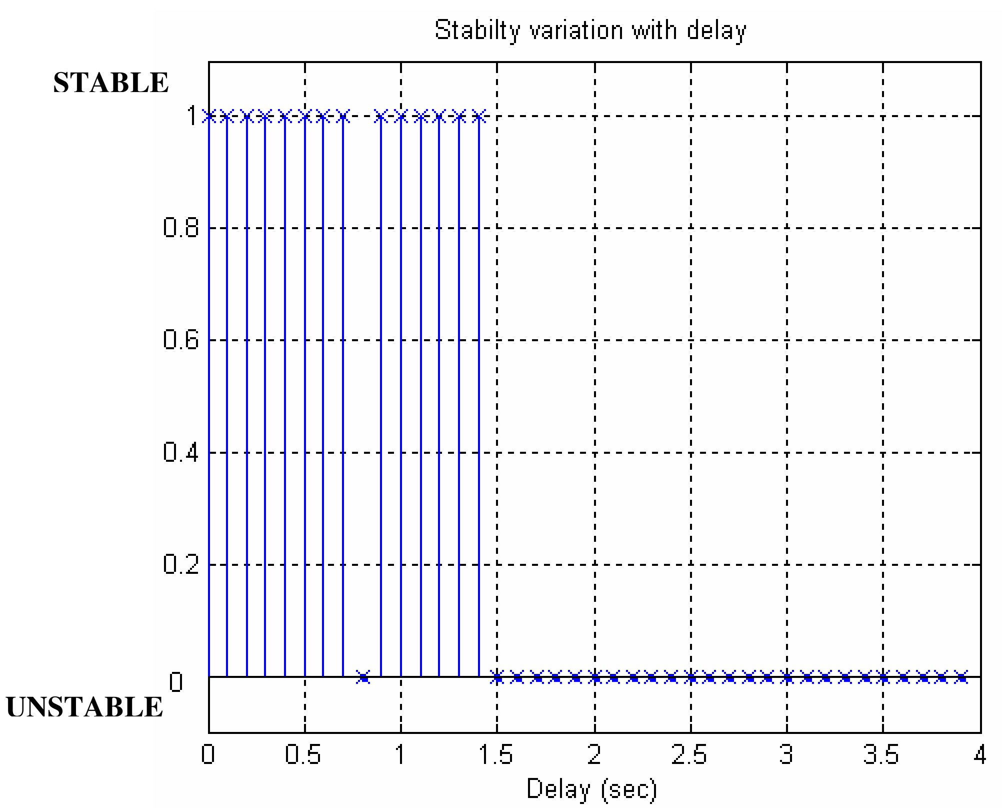


Figure 5.2 Changes in stability with delay (Nyquist pulse)

Of course the magnitude and phase response would vary because of the introduction of this delay, but if this is acceptable, then it might be possible to synthesize systems which were previously thought unstable just by introducing an additional delay element.

Figure 5.3 show the plots for the location of poles for the original system (no delay), system after a delay of 1sec (stable), a delay of 1.4 sec (stable) and after a delay of 1.5 sec (unstable).

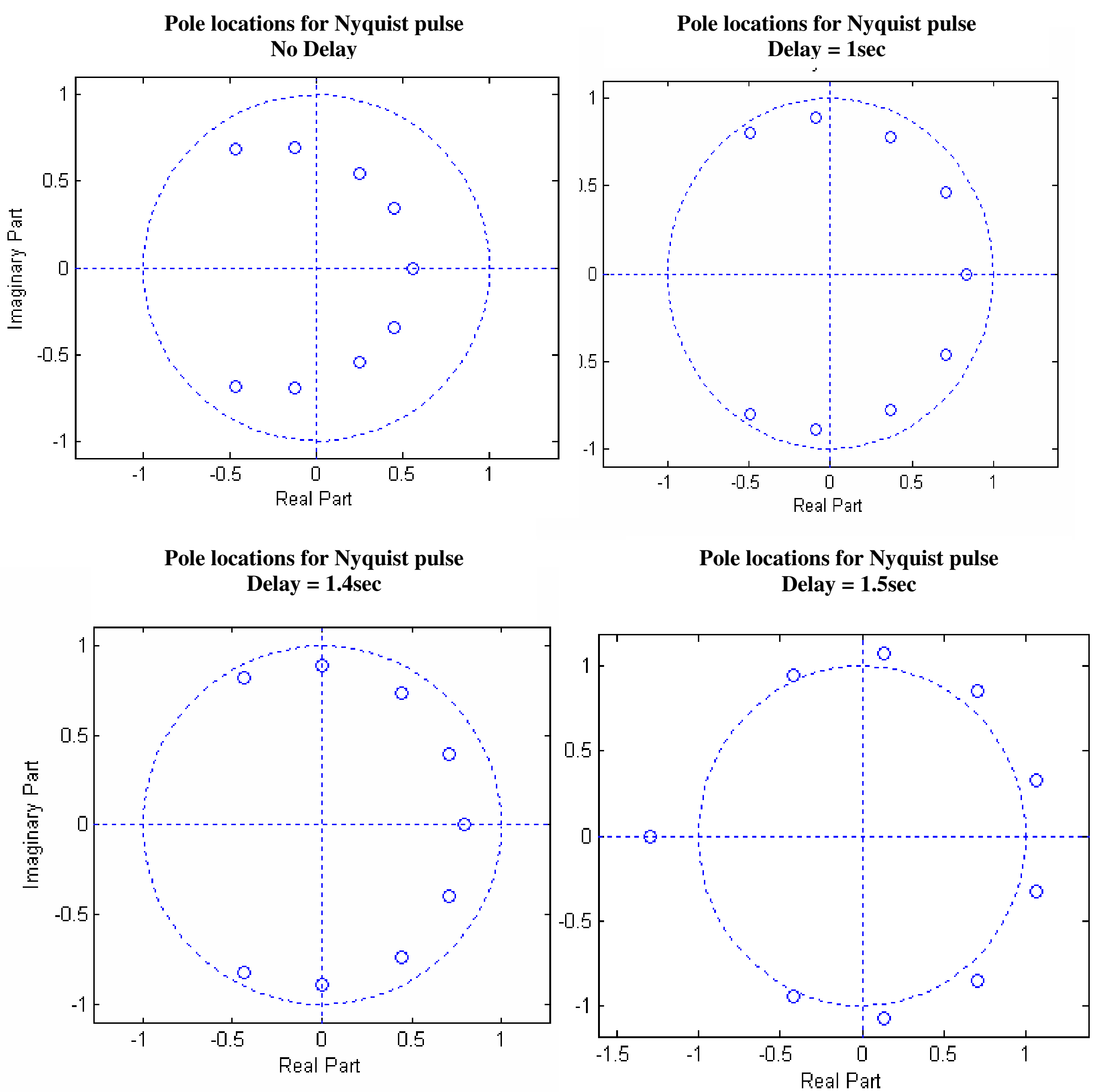


Figure 5.3 Pole locations for the Nyquist pulse at various stages of delay

Figure 5.4 shows a comparison of the normalized transient responses (all plotted in MATLAB) at zero delay, 1 sec, 1.4 sec and finally the unstable response at 1.5 seconds.

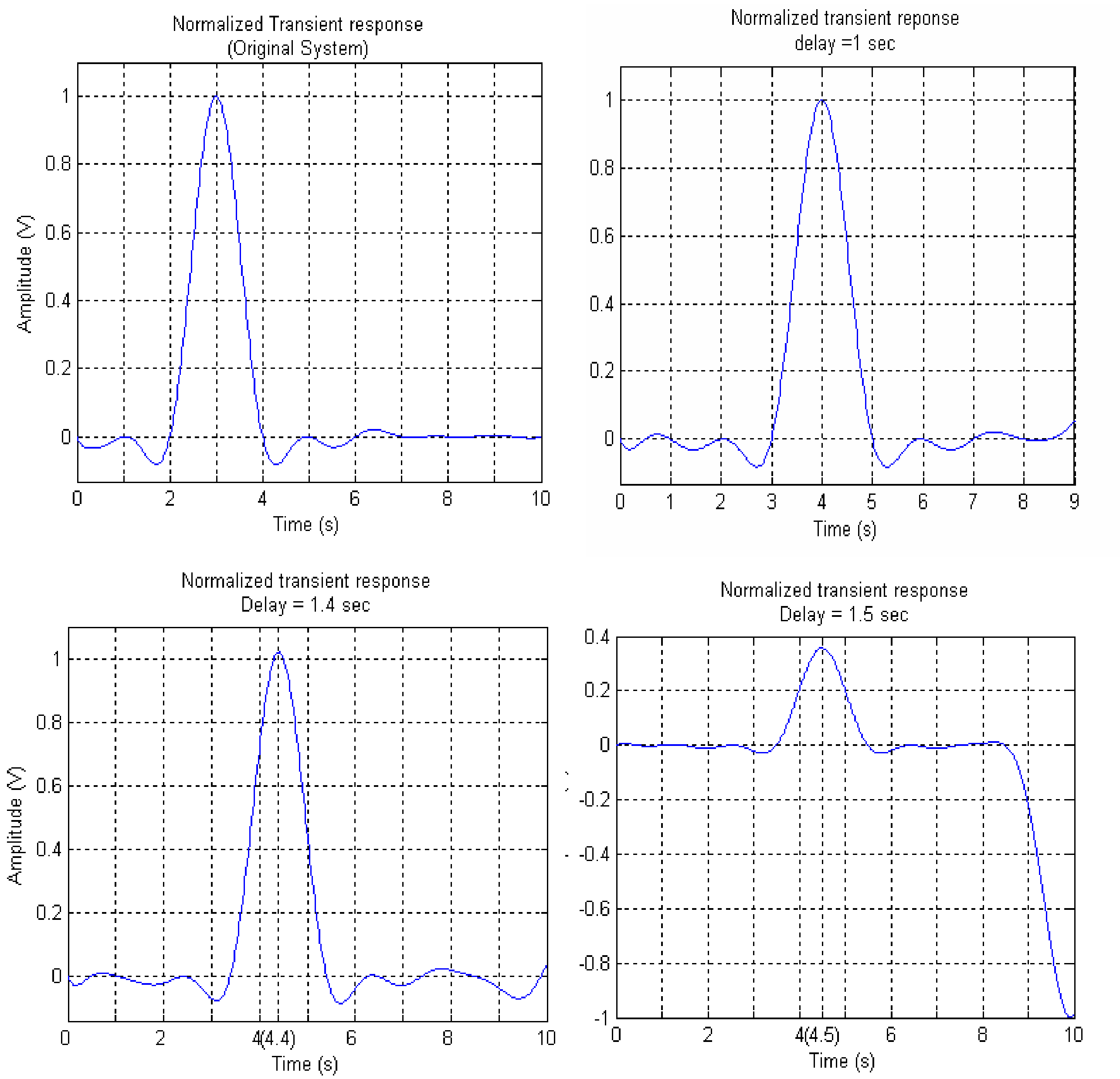


Figure 5.4 Transient response for the Nyquist pulse at various stages of delay

5.4.2. The Square root raised cosine pulse

The same procedure of obtaining 18 samples starting from *t=0 sec* at a sampling interval of *t=0.5 sec* was followed for the square root raised cosine pulse. These samples were fed to the function for the synthesis of impulse responses. The delay was varied and a graph indicating the variation in stability with delay was plotted. From Figure 5.5, it can be seen that by varying the delay of the system it is possible to make the system alternately stable and unstable within certain intervals.

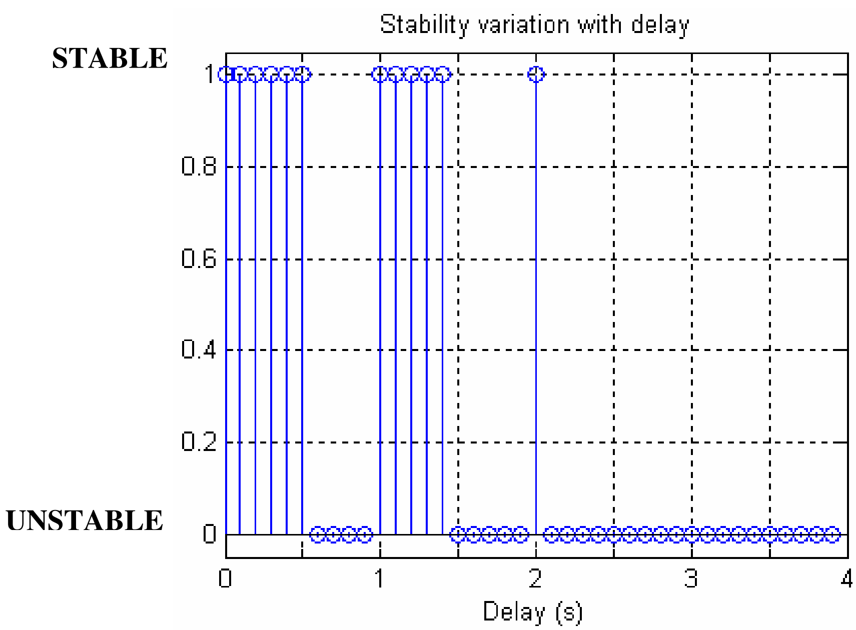


Figure 5.5 Variation in stability with delay (Square root raised cosine pulse)

Figure 5.6 show the plots for the location of poles for the original system (no delay), system after a delay of 0.5 and 0.55 sec (stable), a delay of 0.6 sec (unstable) and after a delay of 1 sec (stable) Figure 5.7 shows a comparison of the normalized transient responses (all plotted in MATLAB) at zero delay, 0.5 sec, 0.55 sec, 0.6 sec (unstable) and finally the stable response at 1 sec.

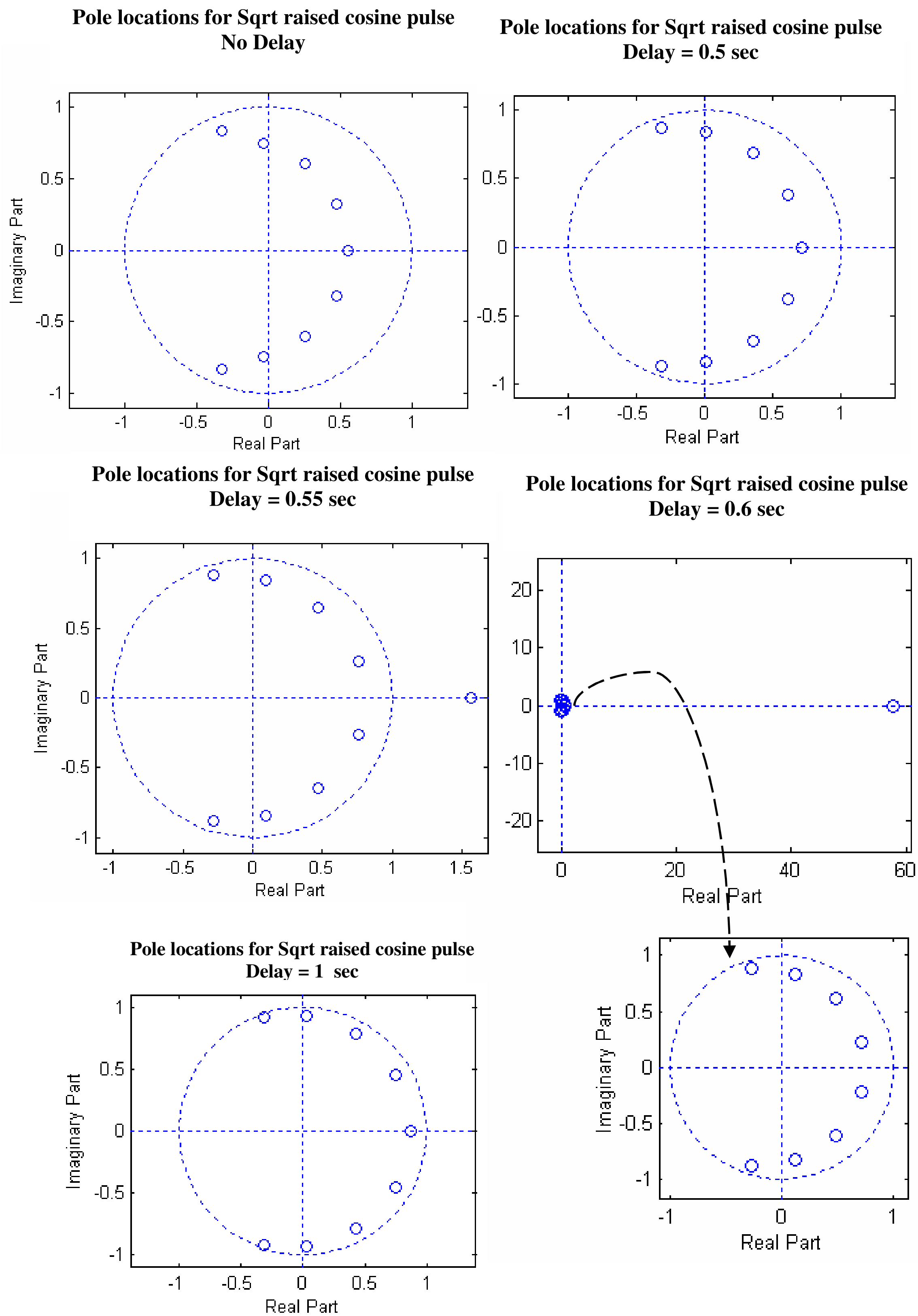


Figure 5.6 Pole locations for the Square root raised cosine pulse at various stages of delay

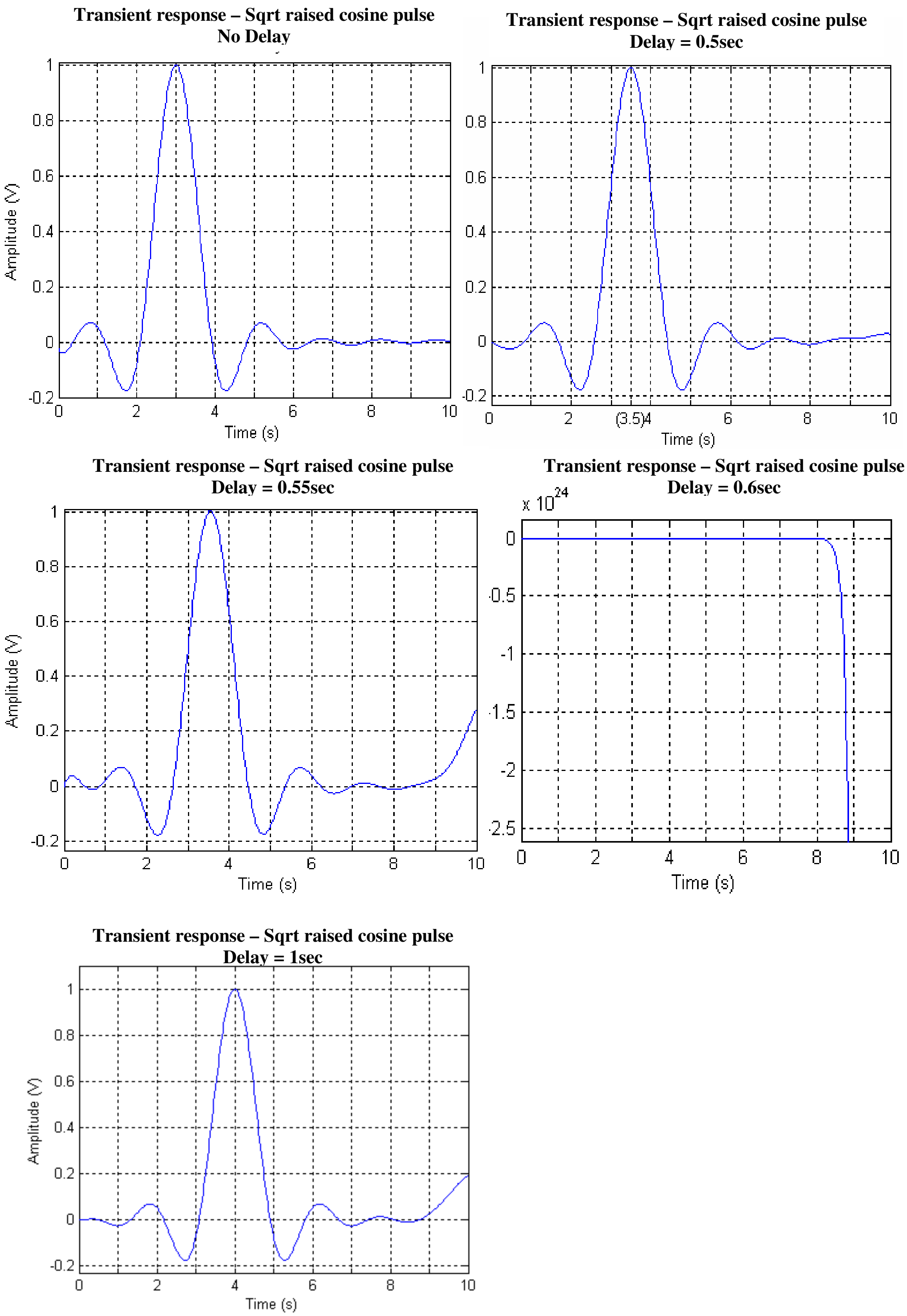


Figure 5.7 Transient response for the Square root raised cosine pulse at various stages of delay

In summary, the stability charts show that in some cases, it is possible to 'stabilize' a system if a delay in the response is acceptable. Prony's method was used for the synthesis of impulse responses. Also, with the introduction of delay, unstable impulse responses can in some cases be made stable

CHAPTER 6

CONCLUSION AND FUTURE WORK

In conclusion, this thesis offers an application written in MATLAB for the synthesis of transfer functions and impulse responses using passive ladder networks. Both single resistively terminated and double resistively terminated networks can be synthesized by this application. It was shown that this application was not only able to synthesize standard functions which have zeros on the imaginary axis, but also more complex functions, such as Nyquist and square root raised cosine pulses, that have zeros anywhere in the s-plane. It was also noticed that the presence of a single zero in the odd part of the split transfer function posed some problems in the synthesis of an acceptable circuit using the Cauer method. However, by splitting this odd section into a cascade of an even part and a simple function with a zero at the origin, it is possible to combine the Cauer technique and Matthaei's method so as to synthesize any arbitrary transfer function. It was also determined that delay has a significantly large effect on the stability of the system. Hence, by introducing some delay, it may be possible to stabilize an impulse response and hence synthesize a stable version of the originally unstable response. However, there appears to be a tradeoff between being able to synthesize a delayed version of the impulse response and the accuracy of the corresponding obtained pulse.

By splitting the original function into even and odd parts, a very simple approach has been adopted to synthesize the transfer function. However, if a certain amount of complexity and the use of transformers are permitted, it may be possible to synthesize the transfer function as one circuit instead of as a cascade of two to three circuits so that the circuit can be simplified. This could form a seed for the future work.

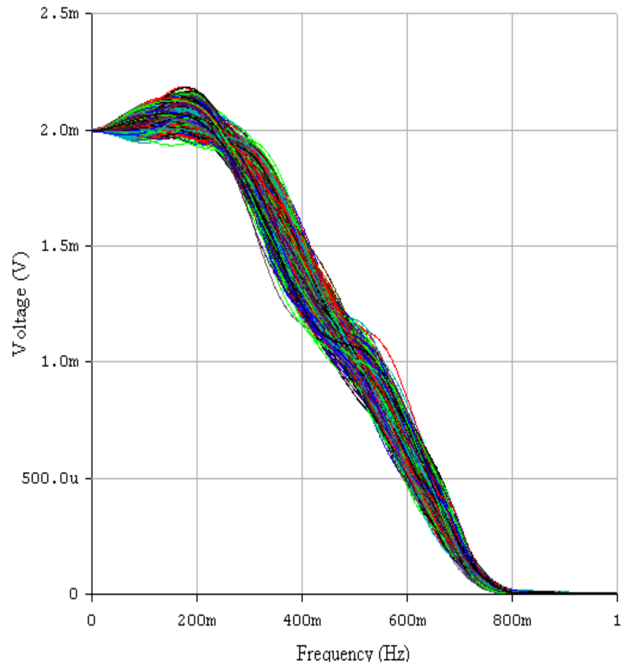
Other future work should address the issue of how the variation of individual component values affects the sensitivity of the synthesized networks. We also note that, while exact values were obtained from the application and used while comparing the obtained and ideal AC and transient responses, in practice, only specific values of components are available. Hence as part of future work, the sensitivity of the synthesized circuits could be examined in more detail. It would be interesting to see how the change in component values affects poles, zeros and stability of the original transfer function. To show just how important this sensitivity analysis is and to depict the advantage of double resistively terminated networks over their single resistively terminated counterparts, preliminary Monte Carlo simulations were run, the details of which can be found in the Appendix.

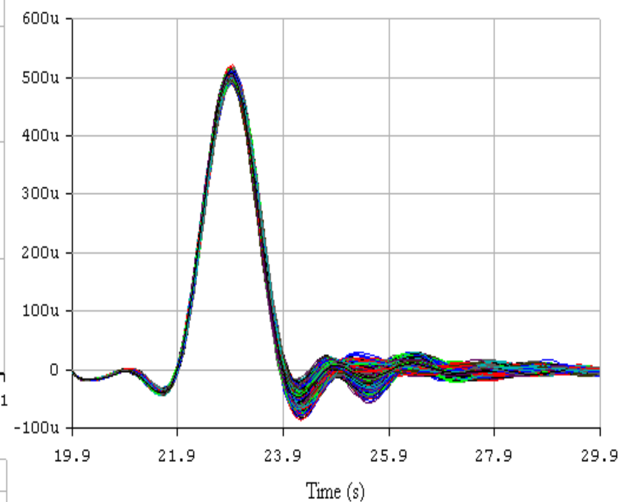
Lastly, a part of future work could also focus on an algorithm to dynamically tune the transmission zeros by tuning the LC resonant and anti-resonant sections. This would ensure a tighter, more accurate response.

APPENDIX

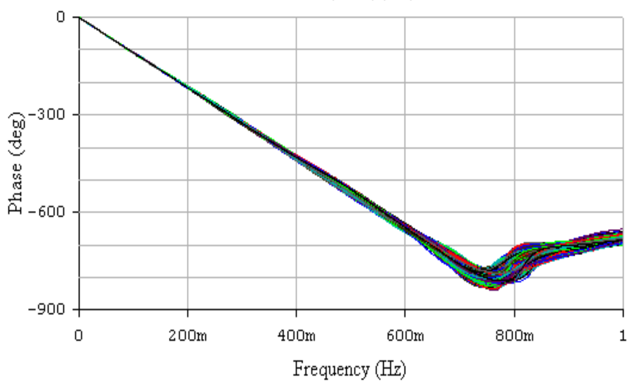
MONTE CARLO ANALYSIS

This section includes the details of a preliminary Monte Carlo analysis that was performed on the single and double resistively terminated ladder networks for the synthesis of a Nyquist pulse. This was done to estimate the sensitivity of frequency and transient analysis to component variations. The tolerances of the passive components were set in accordance with the standard component tolerance values available in the market- namely resistors with 0.1%, inductors with 5% and capacitors with 2% tolerance. 1000 runs were performed with uniform distribution. Figure A.1 shows the Monte Carlo simulations for the AC response and transient response for the single resistively terminated ladder network. Below each of the figures are the statistics in terms of mean, standard deviations and percentage of data within the standard deviations. Figure A.2 shows the Monte Carlo simulations for the AC and transient responses for the double resistively terminated circuit along with the corresponding statistics



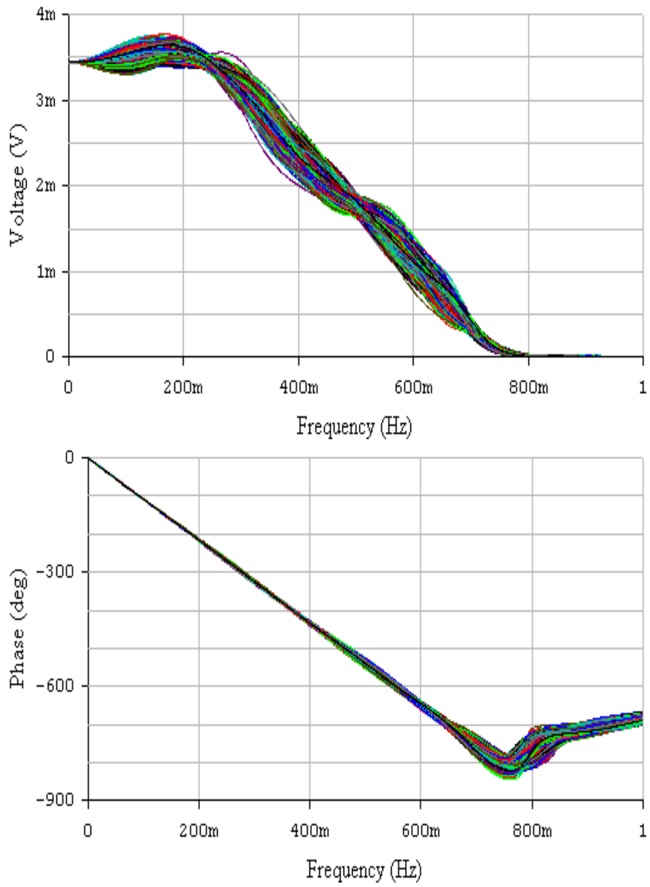


Mean	502.44496u
Standard deviation	5.37961u
Data within 1 std. dev. (%)	66.70000
Data within 2 std. dev. (%)	96.20000
Data within 3 std. dev. (%)	99.90000
Data within 4 std. dev. (%)	100.00000
Data within 5 std. dev. (%)	100.00000

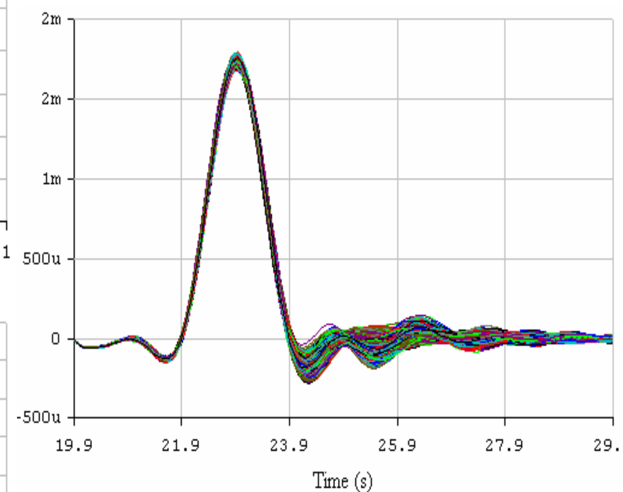


Mean	2.06982m
Standard deviation	34.48523u
Data within 1 std. dev. (%)	69.00000
Data within 2 std. dev. (%)	95.90000
Data within 3 std. dev. (%)	99.80000
Data within 4 std. dev. (%)	100.00000
Data within 5 std. dev. (%)	100.00000

Figure A.1 Monte Carlo simulations for AC response (left) and transient response (right) for single resistively terminated network for Nyquist pulse



Mean	3.58440m
Standard deviation	67.07326u
Data within 1 std. dev. (%)	66.90000
Data within 2 std. dev. (%)	96.00000
Data within 3 std. dev. (%)	100.00000
Data within 4 std. dev. (%)	100.00000
Data within 5 std. dev. (%)	100.00000



Mean	1.73546m
Standard deviation	21.40651u
Data within 1 std. dev. (%)	66.60000
Data within 2 std. dev. (%)	96.10000
Data within 3 std. dev. (%)	100.00000
Data within 4 std. dev. (%)	100.00000
Data within 5 std. dev. (%)	100.00000

Figure A.2 Monte Carlo simulations for AC response (left) and transient response (right) for double resistively terminated network for Nyquist pulse

It must be noted that the behavior of the response (frequency and transient) in the pass band must be as prescribed by the transfer function. The absolute level of the responses is not important [0, 0]. If component variations cause attenuation (at all frequencies) in the response, it is acceptable since the basic contour of the response is maintained. This can be compensated, if required, by means of an amplifier. On the other hand, errors in component values can also cause unwanted ripples or jumps in the response. It is difficult to compensate for such deviations [0, 0]. Comparing the statistics in Figures A.1 and A.2, it can be noted that the performance of a double resistively terminated ladder network is better as compared to the single terminated ladder network in the presence of component variation.

BIBLIOGRAPHY

1. H. J. Orchard, "Inductorless Filters", *Electronics Letters*, Vol. 2, September, 1966, pp.224-225

2. Van Blaricum, M. L., and R. Mittra, "A technique for extracting the poles and residues of a system directly from its transient response," *IEEE Trans. Antenna Propagat*ion., Vol. AP–23, No. 11, pp. 777-781, Nov. 1975

3. M. Piedade, M.Silva , "An improved method for the evaluation of filter sensitivity performance", *IEEE Trans. Circuits and Systems*., Vol. 33, No. 3, 332-335, Mar. 1986

4. H. J. Orchard, "Loss Sensitivities in singly and doubly terminated filters," *IEEE Trans. Circuits and Systems.*, vol. CAS-26, pp. 293-297, May 1979.

5. Gabor C. Temes and Sanjit K. Mitra, *Modern Filter Theory and Design*, John Wiley & Sons Inc, 1973.

6. R. Schaumann and E. Van Valkenburg, *Design of analog filters,* Oxford University Press, New York,2001

7. James E. Storer, *Passive network synthesis*, McGraw-Hill, New York, 1957.

8. L. Weinberg, *Network analysis and synthesis* McGraw-Hill, New York, 1962.

9. William C. Yengst, *Principles of modern network synthesis,* Macmillan, New York 1964

10. Ernst**.** A. Guillemin, *Synthesis of Passive Networks,* Wiley, New York, 1957

11. Gobind Daryanani, *Principles of active network synthesis and design*, Wiley, New York, 1976.

12. Harry Lam, *Analog and Digital filters*, Prentice Hall, 1979

13. Ernest. S. Kuh and Donald. O. Pederson, *Principles of Circuit Synthesis*, McGraw-Hill, New York, 1959.

14. Dov Hazony, *Elements of Network Synthesis*, Reinhold, New York, 1963.

15. A.B. Williams and F.J. Taylor *Electronic Filter Design Handbook*, 3rd Ed., McGraw-Hill, Inc, 1995

16. Van Valkenburg, *Introduction to Modern Network Synthesis*, Wiley, New York, 1959

17. S.L.Marple, *Digital spectral analysis: with applications*, Prentice Hall, New Jersey, 1987

18. Charles L. Philips and H. Troy Nagle, *Digital Control system analysis and design*, Prentice Hall , 1994

19. Divi R. Gupta, *Synthesis of continuous time Nyquist pulses,* M.S. Thesis, University of Massachusetts Amherst, 2006.

20. Van Valkenburg,M.E, *Analog Filter Design,* Holt Rinehart and Winston, 1982.