FILTER DESIGN SOFTWARE BY SYNTHESIS METHOD

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İHSAN YAYLA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2009

Approval of the thesis:

FILTER DESIGN SOFTWARE BY SYNTHESIS METHOD

submitted by **İHSAN YAYLA** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Nevzat Yıldırım _____
Supervisor, **Electrical and Electronics Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Canan Toker
Electrical and Electronics Engineering, METU _____

Prof. Dr. Nevzat Yıldırım
Electrical and Electronics Engineering, METU _____

Prof. Dr. Altunkan Hızal
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Şimşek Demir
Electrical and Electronics Engineering, METU _____

Dr. Yakup Şen
General Manager, Filkon Elektronik Tic. ve Pazarlama Ltd. Şti. _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: İHSAN YAYLA

Signature :

ABSTRACT

FILTER DESIGN SOFTWARE BY SYNTHESIS METHOD

Yayla, İhsan
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Nevzat Yıldırım

January 2009, 106 pages

In this study, Window-based computer program, named Synthesis Software, is developed for designing filters with equal-ripple or maximally flat passbands and general stopbands by using cascade synthesis technique in transformed frequency domain. Synthesis technique is applicable to lumped element and commensurate line distributed element filters with Lowpass, Highpass or Bandpass characteristics. Singly or Doubly terminated filters can be synthesized. This software is based on the previous softwares developed in EEE Department of Middle East Technical University. All the previous softwares were gathered in the well-known software Filpro, which is in DOS environment, in Pascal. Thus, the new software is actually a conversion of Synthesis part of Filpro from DOS environment into Windows environment in the language C#, with some improvements in root finding algorithms for numerical conditioning. Synthesis Software is has three parts. The first and main part is the implementation of synthesis technique by using object oriented programming technique. In this way, synthesis technique implementation is isolated from other parts of Synthesis Software and it can be used by other filter design programs as a module. The second part of the program is response-plotting section. In this part Insertion Loss, Return Loss, Time Delay, Phase and Smith Chart responses are calculated and displayed. The last part is User Interface, which provides user-

friendly environment for typing in the parameters of the filter to be designed. This part uses Synthesis and Plot parts as modules.

Keywords: Transformed variables, lumped and distributed element filters, cascade synthesis technique

ÖZ

SENTEZ YÖNTEMİ İLE SÜZGEÇ TASARIM PROGRAMI

Yayla, İhsan
Yüksek Lisans, Elektrik ve Elektronik Mühendisliğ Bölümü
Tez Yöneticisi : Prof. Dr. Nevzat Yıldırım

Ocak 2009, 106 sayfa

Bu çalışmada sentez yöntemi kullanılarak, geçirme bandında eş-salınımlı ve maximum düzleştirilmiş, durdurma bandında genel süzgeç tasarımı yapan Sentez Yazılımı isimli Windows tabanlı bilgisayar programı geliştirilmiştir. Bu yazılımla Toplu Parametreli veya Dağılmış Parametreli, alçak geçiren, yüksek geçiren veya band geçiren süzgeçler tasarımlanabilir. Tek ya da iki kapısı yükle sonlanmış süzgeçler tasarımlanabilir. Bu yazılım, ODTÜ Elektrik ve Elektronik Bölümünde süzgeç tasarım alanında geliştirilmiş olan yazılımlar temel alınmıştır. Bu yazılımların tümü, DOS ortamında çalışan ve Pascal dilinde yazılmış olan Filpro adlı yazılımda derlenmişti. Bu yazılım, Filpro'nun sentez bölümünün C# dili kullanılarak Windows ortamına aktarılmış halidir. Bu aktarma sırasında polinom köklerinin hesaplanma tekniği değiştirilerek geliştirilmiştir. Tez üç ana parçadan oluşmuştur. Birinci ve ana parçası sentez yönteminin nesneye yönelik programlama tekniği kullanılarak gerçekleştirilmesidir. Bu yolla sentez yöntemi yazılımının diğer parçalarından soyutlanmıştır ve diğer süzgeç tasarım programramları tarafından modül olarak kullanılabilir. Programın ikinci parçası süzgeç tepkisinin çizildiği bölümdür. Bu bölümde Araya Girme Kaybı, Yansıma Kaybı, Zaman Gecikmesi, Evre Gecikmesi ve Smith Haritası çizilir. Son parça ise süzgeç tasarımı için gereken parametrelerin girildiği, Sentez ve Çizim parçalarını modül olarak kullanan Kullanıcı Arayüzü'dür.

Anahtar Kelimeler: Dönüştürülmüş değişkenler, toplu ve dağıtılmış elemanlı süzgeçler, sentez yöntemi

To My Family

ACKNOWLEDGMENTS

I would like to express my gratitude to my thesis supervisor Prof. Dr. Nevzat Yıldırım for his encouragement, support and belief in my work.

In addition, I would like to thank Mehmet Karaaslan for giving his time to discuss various aspects of my thesis work.

Finally, I would like to thank my family for their support and understanding throughout my thesis study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The purpose of this thesis is to program synthesis of equal-ripple and maximally flat passband filters with general stopbands by using synthesis technique. The design process of filters using synthesis technique involve several stages and each of them includes many lengthy calculations, transformations and iterations. Optimal and realizable filters take too much time and are hard by hand calculations. In this thesis, these problems are eliminated by Windows program, named as Synthesis Software. By this software, user can specify the desired filter properties and start designing filter. Synthesis Software makes all of the calculations, transformations and iterations step by step and guides the user. Meantime filter responses are plotted and filter circuits are constructed and drawn. All of these steps can be applied to different types of filters. Also user can save and load designs by XML format. Synthesis implementation is programmed by object oriented programming technique and isolated of other parts of programs. Therefore, the synthesis module can be used by other filter design programs.

1.1 Organization of Thesis

Organization of the thesis is as follows.

In Chapter 2, the main definitions related to filter synthesis are given and the formation of transfer functions in S-domain and then in transformed frequency domain (Z-domain) are described.

In Chapter 3, Synthesis Software structure is described. In addition, usage of programs for filter design is discussed in this chapter.

In Chapter 4, some examples are given and designed step by step by using Synthesis Software.

Conclusion and future work is given in Chapter 5.

1.2 Previous Work

Although filter design is a very old problem, new design techniques are needed in parallel with continuously developing production technologies for new electronic and communication devices. Therefore, new softwares need be developed suiting to the new production technologies and devices. It is almost impossible to design the new filters without proper software. The old softwares must be updated in the new computer programming languages and new computer environments. A historic turning point for the synthesis of filters is the works of H.J. Orchard [1]. Prior to the works of Orchard, the synthesis softwares were severely limited by the numerical ill-conditioning caused by the clustering of roots of polynomials in certain frequency ranges. In his works, the synthesis problem is treated in Transformed Frequency domain, which overcame the numerical inaccuracy problems to a great extent. These accuracy enhancement techniques for direct design of equal-ripple passband general stopband filters are examined and programmed by Y. Sen and U. Koyaz [3], [4]. Another study is done by R. Uzun for the lumped and distributed parameter filters design and modification of the designed filter process and computer program is developed by using various modification techniques[5]. Filter design works are continued in the M. Sc. Thesis of M. Karaaslan. A DOS-based Computer program is developed named as FILPRO, which includes all prior works in this area besides the Synthesis technique [6]. In this thesis, the Synthesis part of Filpro is revised and transferred to Windows environment after some modifications in root finding algorithms and with a new User Interface.

CHAPTER 2

FILTER CONCEPTS AND DEFINITIONS

2.1 Introduction

In this section the basic definitions related to filter synthesis are given. The transmission zero and reflection zero concepts are defined and used in the formation of transfer functions. The transfer function is described first in s-domain and then in transformed variable z-domain for accuracy enhancement purposes. Finally, element extraction formulas are given in detail.

2.2 Filter Types

In synthesis stage, filters are divided two groups according to the element types. These are lumped element filters and distributed element filters which are defined as follows.

2.2.1 Lumped Filters

In the lumped element filters, the physical lengths of circuit element are much shorter than the wavelength in the range of frequency of interest and connecting wires are accepted lossless and zero length. The lumped elements used in synthesis are resistors, inductors, capacitors and transformers a ladder or cascade form.

2.2.2 Commensurate Length Distributed Element Filters

In distributed element filters, circuit element lengths are comparable to the wavelength at the frequency of operation. In this thesis, distributed element filters are made up of equal length (or commensurate length) transmission lines, open circuited stubs and short-circuited stubs. The frequency at which the common length of the elements becomes quarter wavelength is a critical frequency named as f_q. The response of the filter becomes periodic with a period of $2f_q$.

2.3 Passband Types

The frequency band occupied by the desired signals is named as the passband. In this band, the signal is transmitted with no attenuation and distortion. In the desired frequency range, filter transmits signals without loss and out of specified range signal does not transmit to load. In general, there are four passband types which lowpass (LP), highpass (HP), bandpass (BP) and bandstop (BS). In the synthesis algorithm presented in this thesis only LP, HP and BP filters are implemented. BS filters can be synthesized by first designing a prototype LP filter and then using LP-to-BS transformation.

2.4 Frequency Response Types

In this study, filters with equal ripple or maximally flat passband and general stopband filters are programmed. The stop bands are shaped by the transmission zeros at f=0, f=Infinity and Finite Transmission Zeros (FTZ) as shown in Figure 2.1

2.5 Scattering Parameters for Two Port Networks

In two port networks, scattering parameters (S Parameter) are used to describe filter characteristic. In high frequency practice, it is easy to distinguish and measure power and phase of waves instead of voltages and currents. Therefore, power wave concepts are the best suiting quantities for such applications. The power wave's a_1, b_1, a_2, b_2 on these lines can be defined

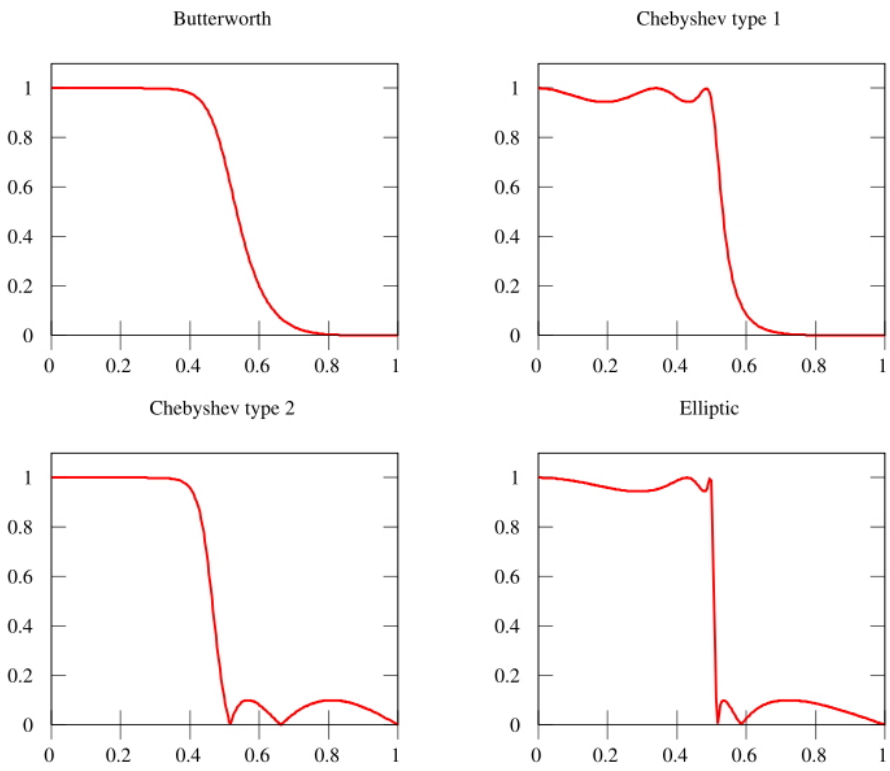


Figure 2.1: Frequency Filter Types [7]

by the port voltages and currents as shown in Figure 2.2, in accordance with equations 2.1.

$$a_1 = \frac{1}{2}\left[\frac{V_1}{\sqrt{R_1}} + \sqrt{R_1}I_1\right] \tag{2.1a}$$

$$b_1 = \frac{1}{2}\left[\frac{V_1}{\sqrt{R_1}} - \sqrt{R_1}I_1\right] \tag{2.1b}$$

$$a_2 = \frac{1}{2}\left[\frac{V_2}{\sqrt{R_2}} + \sqrt{R_2}I_2\right] \tag{2.1c}$$

$$b_2 = \frac{1}{2}\left[\frac{V_2}{\sqrt{R_2}} - \sqrt{R_2}I_2\right] \tag{2.1d}$$

Where R1 and R2 are the resistors terminating ports 1 and 2 (usually termed as reference resistors). From these equations it can be shown that,

$|a_1|^2$: is the maximum incident power signal.

$|b_1|^2$: Power reflected back to the source.

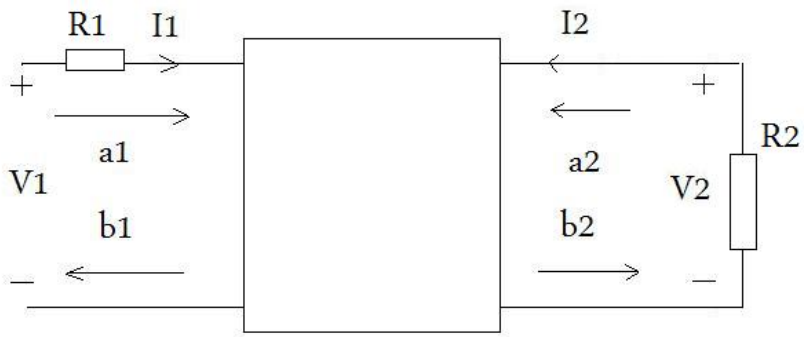


Figure 2.2: Two Port Network

$|b_2|^2$: Power transmitted to the load.

$|a_2|^2$: Power reflected from the source.

The scattering parameters are defined in terms of the incident and reflected signals as follows:

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{2.2}$$

When only port-1 is excited as shown in Figure 2.2, then the a and b parameters can be interpreted in terms of incident real power (maximum or available power) P_{max}, reflected real power $P_{reflected}$ and transmitted real power P_{trans} as follows:

$$|a_1|^2 = \frac{|V_1|^2}{4R_1} = P_{max} \tag{2.3a}$$

$$|a_2|^2 = 0 \tag{2.3b}$$

$$|b_1|^2 = P_{reflected} \tag{2.3c}$$

$$|b_2|^2 = P_{max} - P_{reflected} = P_{trans} \tag{2.3d}$$

Here $a_2 = 0$ means in the equation 2.3

(i) No source is connected to port-2

(ii) Port-2 is terminated in its (reference resistance R_2 (or R_L))

By using equation 2.2, magnitudes of S parameters can be calculated in terms of the magnitudes of a and b follows:

$$|s_{11}|^2 = \left.\frac{|b_1|^2}{|a_1|^2}\right|_{a_2=0} = \frac{P_{reflected}}{P_{max}} \tag{2.4a}$$

$$|s_{21}|^2 = \left.\frac{|b_2|^2}{|a_1|^2}\right|_{a_2=0} = \frac{P_{trans}}{P_{max}} \tag{2.4b}$$

For lossless and passive two ports, conservation of real power principle gives that,

$$|a_1|^2 = |b_1|^2 + |b_2|^2 \tag{2.5}$$

This equation can also be written in terms of the scattering matrix parameters, leading to the unitary property of s parameters as follows:

$$|s_{11}|^2 + |s_{21}|^2 = 1 \tag{2.6}$$

Insertion loss is defined in dB as shown in Equation 2.7

$$IL = -10log_{10}\left[s_{21}^2\right] \tag{2.7}$$

Return loss is defined in dB as shown in Equation 2.8

$$RL = -10log_{10}\left[s_{11}^2\right] \tag{2.8}$$

2.6 Approximation Theory

Filters are designed to approximate unity power gain in the passband:
$|S_{21}|^2 = 1$ for ω in passband
and to approximate zero gain in frequency band (s) called stopband.
$|S_{21}|^2 = 0$ for ω in stopband

Circuits made up of ideal, lossless elements like inductors, capacitors, transmission lines, stubs, coils, etc. can be used to approximate these properties. For lumped element circuits scattering matrix parameters come out to be rational functions of complex frequency $s = \sigma + j\omega$. Belevitch [9] had shown that scattering matrix parameters of two ports made up of lumped, lossless, reciprocal, passive elements could be described in terms of only three real polynomials, $f_e(s)$ or $f_o(s)$, h(s) and g(s) as:

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \left[\frac{1}{g(s)}\right] \begin{bmatrix} h(s) & f_e(s) \\ f_e(s) & -h(-s) \end{bmatrix} \tag{2.9}$$

or,

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \left[\frac{1}{g(s)}\right] \begin{bmatrix} h(s) & f_o(s) \\ f_o(s) & h(-s) \end{bmatrix} \tag{2.10}$$

Here, $f(s) = f_e(s)$ or $f(s) = f_o(s)$ is either even or odd polynomial, g(s) is a strictly Hurwitz polynomial (all roots in the left half of $s = \sigma + j\omega$ plane), while h(s) is a real polynomial linked to f(s) and g(s) through an equation called Feldtkeller Equation:

$$g(s)g(-s) = h(s)h(-s) + f(s)f(-s) \tag{2.11}$$

which is a generalization of the unitary property given in equation 2.6, to cover the whole $s = \sigma + j\omega$ plane. Thus, circuits formed by only lumped inductors and capacitors have scattering parameters of the form

$$S_{21}(s) = \frac{f(s)}{g(s)}, \; S_{11}(s) = \frac{h(s)}{g(s)} \tag{2.12}$$

from which the transducer gain of the two port can be obtained as

$$G(\omega^2) = S_{21}(s)S_{21}(-s)|_{s=j\omega} = \left[\frac{f(s)f(-s)}{g(s)g(-s)}\right]\Bigg|_{s=j\omega} \tag{2.13}$$

This relation can also be used in reverse form to obtain $S_{21}(s)$ from a properly specified power gain function $G(\omega^2)$ through

$$S_{21}(s)S_{21}(-s) = G(\omega^2)|_{\omega=s/j} \tag{2.14}$$

It is clear that the poles and zeros of $S_{21}(s)$ have the primary roles in shaping frequency response of the two-port. Therefore, let's write the product $S_{21}(s)S_{21}(-s)$ in more detailed form:

$$S_{21}(s)S_{21}(-s) = \frac{f(s)f(-s)}{g(s)g(-s)} = \frac{f(s)f(-s)}{f(s)f(-s) + h(s)h(-s)} = \frac{1}{1 + K(s)K(-s)} \tag{2.15}$$

where

$$K(s)K(-s) = \left[\frac{h(s)h(-s)}{f(s)f(-s)}\right] \tag{2.16}$$

2.7 Transmission Zeros

The frequencies which make $S_{21}(s)S_{21}(-s) = 0$ are called transmission zeros (TZ) because at these frequencies no signal can reach to output port. Similarly the frequencies which make $S_{21}(s)S_{21}(-s) = 1$ are termed as reflection zeros (RZ) of the two-port because at this frequency all the incident power is delivered to the load without any reflection. In order to inspect the nature of transmission and reflection zeros let's write f(s) and h(s) in expanded forms:

$$f(s) = f_m s^{m_0} \prod_{k=1}^{m-m_0} (s - s_k) \tag{2.17}$$

$$h(s) = h_n s^{n_0} \prod_{r=1}^{n-n_0} (s - s_r) \tag{2.18}$$

where f_m and h_n are constant coefficients, m_o and n_o are multiplicities of the roots of f(s) and h(s) at $s = 0$, s_k and s_r are the finite nonzero roots of f(s) and h(s). From equation 2.15 to 2.18 the following observations can be made:

(i) If $n > m$, then $S_{21}(s)S_{21}(-s) = 0$ at $s = \infty$. Therefore, the two-port has a transmission zero at $s = \infty$ of multiplicity $n_\infty = n - m$. The degree of g(s) is then set by h(s) (through Feldtkeller Equation) as n, which will be referred to as degree of the filter.

(ii) If $m > n$, then $S_{21}(s)S_{21}(-s) = 1$ at $s = \infty$ and it can be said that there is a reflection zero at $s = \infty$ with multiplicity $m_\infty = m - n$. Then the degree of g(s) is equal to m.

(iii) In equations 2.17 and 2.18 there is either $m_o = 0$ or $n_o = 0$ (cancellation in equation 2.16). If $m_o > 0$, then there will be a transmission zero at $s = 0$ of multiplicity m_o. If $n_o > 0$, then there will be a reflection zero at $s = 0$ with multiplicity n_o.

(iv) The "finite - nonzero" roots s_k of f(s)=0 are called "finite transmission zeros - FTZ" of the two port, because at these frequencies, $S_{21}(s_k)S_{21}(-s_k) = 0$. Similarly, the finite-nonzero roots s_r of h(s)= 0 at which $S_{21}(s_r)S_{21}(-s_r) = 1$ are called "finite reflection zeros (FRZ)" of the two-port.

(v) Total number of transmission zeros is equal to the total number of reflection zeros. Degree of a filter is equal to the larger one of degrees of f(s) and h(s). Therefore, degree of a filter is also equal to the total number of transmission zeros or reflection zeros.

Each transmission zero at $s = j\omega = 0$ or at $s = j\omega = \infty$ adds one degree while finite transmission zeros add double degrees.

Filter transmission and reflection characteristics are shaped by the multiplicities and positions of transmission and reflection zeros on $s = \sigma + j\omega$ plane. Since almost all signals of interest are on $s = j\omega$ axis (sinusoidal signals), they can be most efficiently filtered by choosing all transmission and reflection zeros on $s = j\omega$ axis (optimal amplitude filtering). The zeros at $s = \infty$ and s=0 are considered to be on $j\omega$ axis. Hence for optimal filtering finite transmission and reflection zeros s_k and s_r will have the forms

$$s_k = j\omega_k \; and \; s_r = j\omega_r \tag{2.19}$$

to be on $s = j\omega$ axis. They must also come in conjugate pairs to form real polynomials:

$$f(s) = f_m s^{m_0} \prod (s - j\omega_k)(s + j\omega_k) = f_m s^{m_0} \prod (s^2 + \omega_k^2) \tag{2.20}$$

$$h(s) = h_n s^{n_0} \prod (s - j\omega_r)(s + j\omega_r) = f_n s^{n_0} \prod (s^2 + \omega_r^2) \tag{2.21}$$

It is seen that f(s) is an even polynomial if m_0 is even and it is an odd polynomial if m_0 is odd. In addition, when all finite reflection zeros are chosen on $s = j\omega$ axis, h(s) becomes also an even or odd polynomial. h(s) is an even polynomial if n_0 is even and it is an odd polynomial if n_0 is odd. That is, the parity of transmission and reflection zeros at f=0 sets the parity of f(s) and h(s). Using the even / odd properties of f(s) and h(s) in equations 2.9 and2.10, the following observations can be made:

- For electrically symmetric two ports($S_{11} = S_{22}$), if f(s) is even, then h(s) is odd and vice versa.
- For electrically antimetric two ports($S_{11} = -S_{22}$), both f(s) and h(s) are even polynomials.

Both f(s) and h(s) cannot be odd, otherwise g(s) would also have a root at s=0, which is not allowed in Strictly Hurwitz polynomials. In addition, it is physically meaningless to place both a transmission zero and a reflection zero at the same frequency, s=0.
From 2.20 and 2.21 the following functions are observed.

$$f(s)f(-s) = f_m^2 s^{2m_0} \prod (s^2 + \omega_k^2)^2 \tag{2.22}$$

$$h(s)h(-s) = h_n^2 s^{2n_0} \prod (s^2 + \omega_r^2)^2 \tag{2.23}$$

which will in turn give power gain functions of the form

$$G(\omega) = S_{21}(j\omega)S_{21}(-j\omega) = \frac{1}{1 + K^2(\omega)} \tag{2.24}$$

where

$$K(\omega) = (\frac{h_n}{f_m})\omega^{(n_0-m_0)} \left[\frac{\prod (\omega^2 - \omega_r^2)}{\prod (\omega^2 - \omega_k^2)} \right] \tag{2.25}$$

with either n_o or m_o being zero. The power gain function $G(\omega)$ may also be written in the expanded form

$$G(\omega) = |S_{21}(j\omega)|^2 = \frac{\sum_{k=0}^{m} a_k \omega^{2k}}{\sum_{r=0}^{m} b_r \omega^{2r}} \tag{2.26}$$

Either of the functional forms given in equations 2.24 or 2.26 can be used in ”amplitude filtering” approximations. They can be tailored to various forms for lowpass (LP), highpass (HP), bandpass (BP) or bandstop (BS) filters by placing the zeros in proper positions and by adjusting the degrees n and m of polynomials. Clearly, all reflection zeros must be placed in passband(s) to facilitate transfer of power to output with minimal reflection and all transmission zeros must be placed in stopband(s) to maximize attenuation of signals in these frequency bands. The two classical techniques to approximate flat gain (unity or zero) within a finite range are as follows:

- **Maximally flat approximation:** If all zeros of a function is collected at a single point, then the function will be maximally flat around that point.
- **Equiripple approximation:** By properly locating zeros and poles of a function, the function can be made to deviate from the targeted constant value in an equal ripple manner within a certain range.

Both approaches can be used to approximate pass and stop bands of filters.
K(s) can be forced to have equiripple behavior in the passband by some suitable choices for K(s). One method is using the transformed frequency variable Z=X+jY which is related to frequency as follows:

$$Z^2 = \frac{s^2 + \omega_{p2}^2}{s^2 + \omega_{p1}^2} \text{ for BP} \tag{2.27}$$

This transformation maps the function f(s)f(-s) into Z-domain, as F(Z)F(-Z). Hence, the transformed version of the power gain function generalized to Z-plane can be written as;

$$S_{21}(Z)S_{21}(-Z) = \frac{1}{\epsilon^2 F(Z)F(-Z)} \tag{2.28}$$

In equation ϵ is passband ripple and F(Z)F(-Z) should have the following properties.

$$F(Z)F(-Z) = \begin{cases} F(jY)F(-jY) \leq 1 & \text{(in passband)} \\ F(X)F(X) \gg 1 & \text{(in stopband)} \end{cases} \tag{2.29}$$

Also TZs is should be used in transformed form. TZs in the s-plane, $s = \sigma + j\omega_i$, can be transformed to z-domain for different types of filters as follows.

$$Z_i(s_i) \begin{cases} \sqrt{\frac{1+\omega_p^2}{s_i^2}} & \text{(for LPF)} \\ \sqrt{\frac{1+s_i^2}{\omega_p^2}} & \text{(for HPF)} \\ \sqrt{\frac{\omega_{p2}^2+s_i^2}{\omega_{p1}^2+s_i^2}} & \text{(for BPF)} \\ \sqrt{\frac{-\omega_{p1}^2-s_i^2}{\omega_{p2}^2+s_i^2}} & \text{(for BSF)} \end{cases} \tag{2.30}$$

The equiripple passband and general stopband property of the gain function can be proved as follows:

Consider the gain function,

$$G(Z) = \prod_{(i=1)}^{n} \frac{Z_i + Z}{Z_i - Z} \tag{2.31}$$

formed by the transmission zeros $Z_i = X_i = X_i(\omega_i)$ such that, for each finite transmission zero pair $s = +j\omega_i$ and $s = -j\omega_i$ a pair of identical Z_i's occur.

K(Z) can be constructed as

$$K(Z) = \frac{1}{2}[1 + G(Z)] \tag{2.32}$$

which gives,

$$K(Z)K(-Z) = \frac{1}{4}[1 + G(Z)][1 + G(-Z)] \tag{2.33}$$

Equation 2.33 has the required properties as shown below:

The complex factors $(Z_i + Z)/(Z_i - Z)$ of G(Z) can be put into polar form as follows:

$$\frac{Z_i + Z}{Z_i - Z} = e^{2\gamma_i} = e^{2\alpha_i} e^{j2\beta_i} \tag{2.34}$$

where, $\gamma_i = \alpha_i + \beta_i$ can be extracted from 2.34 as follows:

$$cosh2\gamma_i = \frac{1}{2}(e^{2\gamma_i} + e^{-2\gamma_i}) = \frac{1}{2}(\frac{Z_i + Z}{Z_i - Z} + \frac{Z_i - Z}{Z_i + Z})$$

Using the identity,

$$cosh^2\gamma_i = \frac{1}{2}(1 + cosh(2\gamma_i))$$

the following equation is found,

$$cosh^2\gamma_i = \frac{Z_i^2}{Z_i^2 - Z^2}$$

or,

$$\gamma_i = cosh^{-1}\frac{Z_i}{\sqrt{Z_i^2 - Z^2}} \tag{2.35}$$

It can be show that for Z=X, equation 2.35 is real (in stopband) giving

$$\gamma_i = \alpha_i$$

and for Z=jY, γ_i becomes imaginary,

$$\gamma_i = \beta_i$$

Hence equation 2.35 becomes

$$\beta_i = cos^{-1}\frac{Z_i}{\sqrt{Z_i^2 + Y^2}} \text{ for } Z_i = jY_i \text{ in (passband)} \tag{2.36}$$

Using equation 2.34 in equation 2.31, G(Z) becomes

$$G(Z) = \prod_{(i=1)}^{n}\frac{Z_i + Z}{Z_i - Z} = \prod e^{2\gamma_i} = e^{\sum 2\gamma_i} = e^{2\gamma} \tag{2.37}$$

where,

$$\gamma = \sum_{i=1}^{n}\gamma_i \tag{2.38}$$

Then K(Z)K(-Z) can be written as

$$K(Z)K(-Z) = \frac{1}{4}[1 + G(Z)][1 + G(-Z)] \tag{2.39a}$$

$$= \frac{1}{4}(2 + e^{2\gamma} + e^{-2\gamma}) \tag{2.39b}$$

$$= \left(\frac{e^{\gamma} + e^{-\gamma}}{2}\right)^2 \tag{2.39c}$$

$$= cosh^2\gamma \tag{2.39d}$$

Hence,

$$K(Z)K(-Z) = cosh^2\left(\sum_{i=1}^{n}\gamma_i\right) \tag{2.40a}$$

$$= cosh^2\left[\sum_{i=1}^{n} cosh^{-1}\frac{Z_i}{\sqrt{Z_i^2 - Z^2}}\right] \tag{2.40b}$$

Analysis shows that the argument

$$\left.\frac{Z_i}{\sqrt{Z_i^2 - Z^2}}\right|_{Z=jY} = \frac{X_i}{\sqrt{X_i^2 + Y^2}} \leq 1$$

Hence in the passband (Z=jY), equation 2.40 becomes

$$K(jY)K(-jY) = cos^2\left[\sum_{i=1}^{n} cos^{-1}\frac{Z_i}{\sqrt{Z_i^2 + Y^2}}\right] \tag{2.41}$$

which has the desired property that it is an equal ripple polynomial oscillating between zero and one.

In the stopband Z=X,

$$K(X)K(-X) = cosh^2\left[\sum_{i=1}^{n} cosh^{-1}\frac{Z_i}{\sqrt{Z_i^2 - X^2}}\right] \tag{2.42}$$

It approaches to infinity about the transmission zero at $X = X_i$. Hence K(Z)K(-Z) satisfies the requirements to have an equal ripple characteristics in the passband, general behavior in the stopband. After showing the formulation of K(Z)K(-Z) which satisfies the requirements, K(s)K(-s) can be evaluated either by returning to s-plane to obtain the necessary polynomial-sor continuing the synthesis procedure in Z-domain as will be explained in section 2.10.

2.7.1 Transmission Zeros of Lowpass Filters

Lowpass filters must have at least one transmission zero at $\omega = \infty(n_\infty > 0)$. Therefore, $n > m$ and degree of filter will be equal to n. No transmission zero is allowed in the passband $\omega \leq \omega_p$. Therefore, no transmission zero exists at $\omega = 0(m_0 = 0)$. That is f(s) is always even for lowpass filters. So, symmetrical filters will be obtained if h(s) is an odd polynomial (odd degree, n) and antimetric filters will result for even h(s) (even degree, n). Finite transmission zeros, if there are, have to be placed at frequencies above the passband edge frequency ($\omega_k > \omega_p$). All reflection zeros lie in the range $0 \leq \omega_r \leq \omega_p$ and $n_0 > 0$. Since n_0 sets the parity of h(s) (odd n_0 means odd n), the parity of reflection zeros at s=0 sets the symmetry or antimetry of the resulting structure.

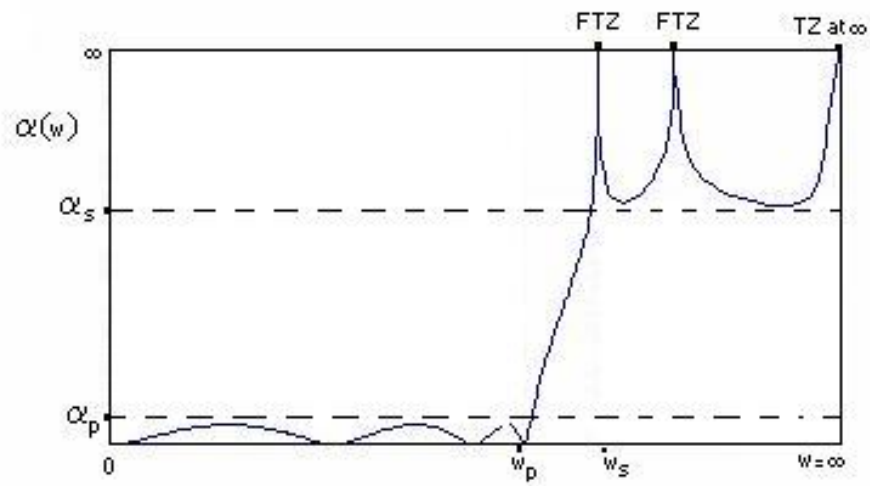


Figure 2.3: Lowpass Filter with Transmission Zeros

2.7.2 Transmission Zeros of Highpass Filters

Highpass filters must have at least one transmission zero at $\omega = 0$ ($m_0 > 0$). No transmission zero is allowed in the passband $\omega \geq \omega_p$. So, no transmission zero exists at $\omega = \infty$. Therefore, $m \geq n$ and degree of filter will be set by the degree m of f(s). Finite transmission zeros, if there are, must be placed below the passband edge frequency (f_p). Since all reflection zeros must be in the passband, $\omega \geq \omega_p$, no reflection zero exists at $\omega = 0$ ($n_0 = 0$). Hence h(s) is an even polynomial for highpass filters. For this reason, if f(s) is odd (odd degree m), symmetric filter is obtained and if f(s) is even (even degree, m), antimetric filter is obtained.

2.7.3 Transmission Zeros of Bandpass Filters

Usual bandpass filters have transmission zeros both at $\omega = 0$ and $\omega = \infty$. Since there are some transmission zeros at $\omega = \infty$, $n > m$ and degree of the filter is set by the degree n of h(s). If no transmission zero is placed at $\omega = 0$, then the filter may also be treated as a quasi-lowpass filter with some loss near $\omega = 0$. If no transmission zero is placed at $\omega = \infty$, then the filter may be treated as a quasi-highpass one, with some loss at high frequencies. Finite transmission zeros, if there are, have to be placed outside the passband. Bandpass filters have no reflection zeros at $\omega = 0$ and $\omega = \infty$. Therefore, $n_0 = 0$, giving even parity for h(s). Only

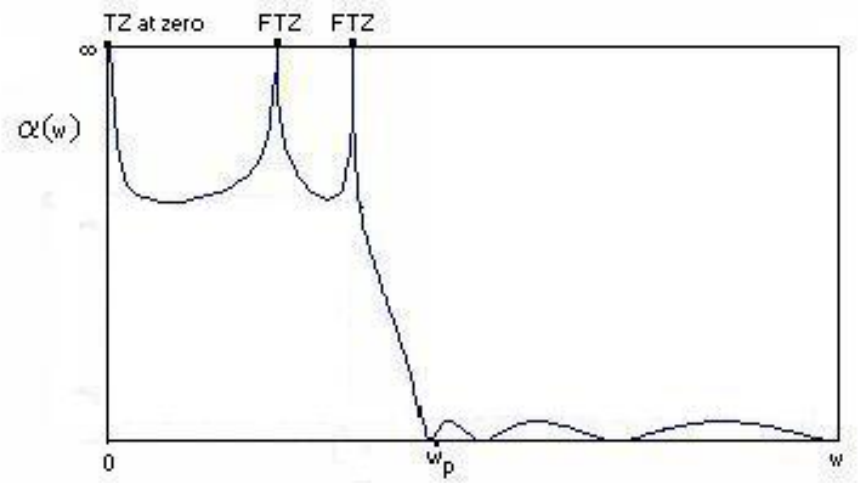


Figure 2.4: Highpass Filter with Transmission Zeros

finite reflection zeros are allowed, all of them being inside the pass band. Since h(s) is even, and since the degree of bandpass filters is set by h(s), bandpass filters will always have even degree. Symmetric filters will be obtained if f(s) is odd (odd number m_0 of transmission zeros at $\omega = 0$) and antimetric filters will be obtained for even f(s)(even number of transmission zeros at $\omega = 0$).

2.8 Element Extraction

Once the proper transducer power gain function $G(\omega)$ is formed to approximate the required specifications then the next stage is extraction of the parameters $S_{21}(s)$ and $S_{11}(s)$ using the relations

$$S_{21}(s)S_{21}(-s) = G(\omega)|_{\omega=-js} \tag{2.43}$$

$$S_{21}(s) = \frac{f(s)}{g(s)} \tag{2.44}$$

$$S_{11}(s) = \frac{h(s)}{g(s)} \tag{2.45}$$

In Equation 2.43 the zeros and poles of $S_{21}(s)S_{21}(-s)$ are shared between $S_{21}(s)$ and $S_{21}(-s)$ by observing the properties of polynomials f(s) and g(s) stated in Belevitch Representation Theorem. Then h(s) is solved from Feldtkeller Equation

$$h(s)h(-s) = g(s)g(-s) - f(s)f(-s) \tag{2.46}$$

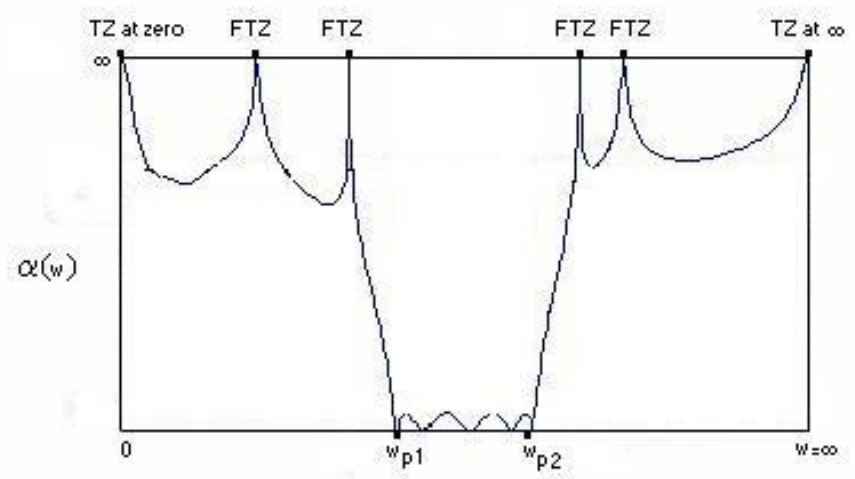


Figure 2.5: Bandpass Filter with Transmission Zeros

now by taking the analytical properties of h(s) into consideration. Once $S_{21}(s)$ and $S_{11}(s)$ are formed, all the other two-port parameters (Z, Y, ABCD, etc.) can be found using interrelations among them. In general the resulting h(s) (hence the parameters $S_{21}(s)$, $S_{11}(s)$, etc.) is nonunique because of freedom of choice in sharing the zeros of h(s)h(-s) between h(s) and h(-s). However, for the cases where all the roots of h(s) are on $s = j\omega$ axis (optimal filters), the roots of h(s)h(-s) are double roots, still on $s = j\omega$ axis. Therefore, h(s) and h(-s) have identical roots, simplifying the root sharing problem. Even in these simplest cases where no choice exist for zeros, the choice of + or - sign for h(s) yields two circuits which are called ”dual” circuits. The circuit corresponding to these parameters are found by extracting elements from driving point functions like $S_{11}(s)$, $Z_{in}(s), Y_{in}(s), Z_{11}(s), Y_{11}(s)$, etc. In this study, elements are extracted from either Z_{11} (s) or $Y_{11}(s)$, which from now on will be referred to as Z(s) and Y(s). They are reactance functions, because their output port is either open or short circuited, eliminating the only resistive component of the circuit, R_L. Therefore, the element extraction techniques described in following sections can be used. Element extraction is carried out in such a way as the resulting circuit will possess the transmission and reflection zeros immersed in the gain function $G(\omega)$. The simplest practical structures are obtained by extracting groups of elements called ”transmission zero sections” in cascaded form. Each section is responsible for creation of a transmission zero. A transmission zero is formed either by creating an open circuit in the series arm (Type-A TZ) or by creating a short circuit in the shunt arm (Type-B TZ) to stop signal flow between input and output. The three kinds of transmission zeros on

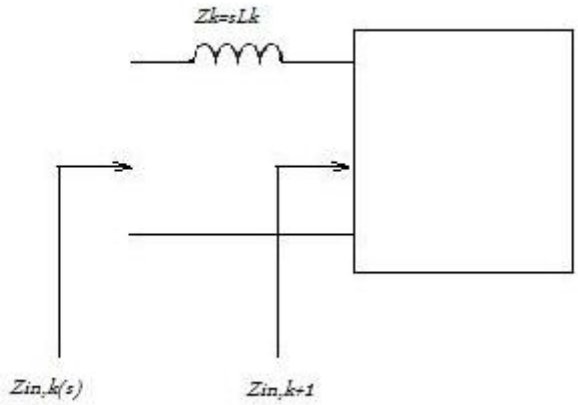


Figure 2.6: Transmission Zero at $s = \infty$ by Series Inductor

$s = j\omega$ axis are realized as follows:

2.8.1 Transmission Zeros at $s = \infty$

If Z_{in} has a pole at $s = \infty$ then it can be removed as a series inductor as shown in Figure 2.6.

The relationship between input impedances before and after extraction is

$$Z_{in,k} = Z_{in,k+1} + sL_k \tag{2.47}$$

from which the inductor is solved as

$$L_k = \left.\frac{Z_{in,k}(s)}{s}\right|_{s=j\omega_{k+1}} \tag{2.48}$$

where, ω_{k+1} is the frequency of the next transmission zero. If the next transmission zero is at finite frequency then the extraction is named as partial pole removal because the remaining impedance will still have a transmission zero at infinity. If the next transmission zero is also at $s = \infty$, then the pole removal process is termed as a full removal.

$$L_k = \left.\frac{Z_{in,k}(s)}{s}\right|_{s=\infty} \tag{2.49}$$

If input admittance has a transmission zero at $s = \infty$ then it is extracted as a shunt capacitor with a similar formulation as shown below (Figure 2.7)

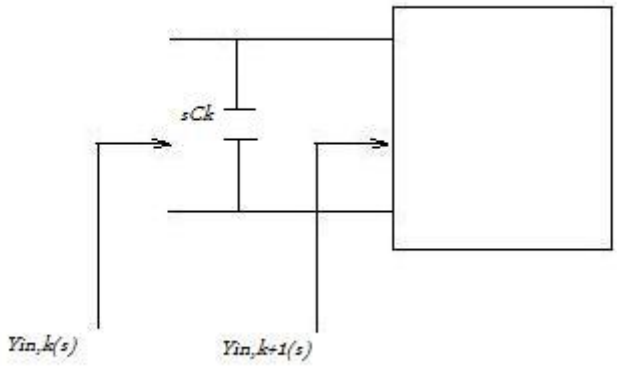


Figure 2.7: Transmission Zero at $s = \infty$ by Shunt Capacitor

$$Y_{in,k} = Y_{in,k+1} + sC_k \tag{2.50}$$

$$C_k = \left.\frac{Y_{in,k}(s)}{s}\right|_{s=j\omega_{k+1}} \tag{2.51}$$

If the next transmission zero is also at $s = \infty$, then pole remove process becomes full removal.

$$C_k = \left.\frac{Y_{in,k}(s)}{s}\right|_{s=\infty} \tag{2.52}$$

2.8.2 Transmission Zeros at $s = 0$

If Z_{in} has a pole at $s = 0$ it can be removed as a capacitor as shown in Figure 2.8.

The relationship can be defined as follows:

$$Z_{in,k} = Z_{in,k+1} + \frac{1}{sC_k} \tag{2.53}$$

where, ω_{k+1} is the frequency of the next transmission zero. If the next transmission zero is also at, the pole remove process becomes full removal.

$$C_k = \left.\frac{1}{Z_{in,k}(s)s}\right|_{s=j\omega_{k+1}} \tag{2.54}$$

If the input admittance has a transmission zero at s=0 the it is extracted as a shunt inductor as shown below (Figure 2.9).

$$C_k = \left.\frac{1}{Z_{in,k}(s)s}\right|_{s=0} \tag{2.55}$$

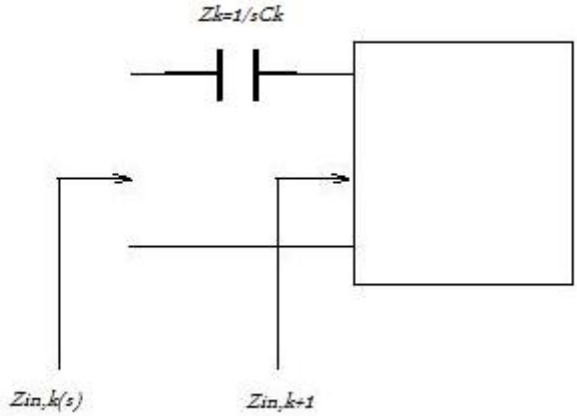


Figure 2.8: Transmission Zero at $s = 0$ by Series Capacitor

For the shunt inductor formulation process can be done as below.

$$Y_{in,k} = Y_{in,k+1} + \frac{1}{sL_k} \tag{2.56}$$

$$L_k = \left.\frac{1}{Y_{in,k}(s)s}\right|_{s=j\omega_{k+1}} \tag{2.57}$$

Again, if the next transmission zero is also at $s = 0$, the pole remove process becomes full removal.

$$L_k = \left.\frac{1}{Y_{in,k}(s)s}\right|_{s=0} \tag{2.58}$$

2.8.3 Transmission Zeros at a finite frequency

If the transfer function has FTZ's then they may be extracted either as series arm parallel LC resonators or shunt arm series LC resonators. In order to extract a series arm parallel LC resonator the input impedance function must have a pole at finite nonzero frequency $j\omega_k$. That is it should be in the following form:

$$Z_{in,k}(s) = \frac{1}{s^2 + \omega^2} \frac{\sum a_i s^i}{\sum b_i s^i} \tag{2.59}$$

Then a series arm parallel LC resonator can be extracted using the following relations (Figure 2.10):

$$L_k C_k = 1/\omega_k^2 \tag{2.60}$$

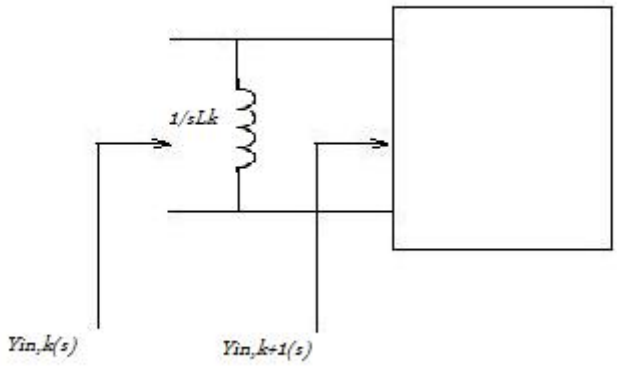


Figure 2.9: Transmission Zero at $s = 0$ by Shunt Inductor

$$Z_k = \frac{1}{C_k}\frac{s}{s^2 + \omega_k^2} \tag{2.61}$$

$$C_k = \left.\frac{s}{(s^2 + \omega_k^2)Z_{in,k}(s)}\right|_{s=j\omega_{k+1}} \tag{2.62}$$

Then using the resonance relation, the inductance can be found.

$$L_k = \frac{1}{\omega_k^2 C_k} \tag{2.63}$$

In order to extract a shunt arm series LC resonator, the input admittance function must have a pole at finite nonzero frequency jwk. That is it should be in the following form: Yin, k has the form

$$Y_{in,k}(s) = \frac{1}{s^2 + \omega^2}\frac{\sum a_i s^i}{\sum b_i s^i} \tag{2.64}$$

Then the series LC resonator in shunt arm can be extracted using the following relations:

$$Y_k = \frac{1}{L_k}\frac{s}{s^2 + \omega_k^2} \tag{2.65}$$

$$L_k = \left.\frac{s}{(s^2 + \omega_k^2)Y_{in,k}(s)}\right|_{s=j\omega_{k+1}} \tag{2.66}$$

Then the capacitor can be found using the resonance relation:

$$C_k = \frac{1}{\omega_k^2 L_k} \tag{2.67}$$

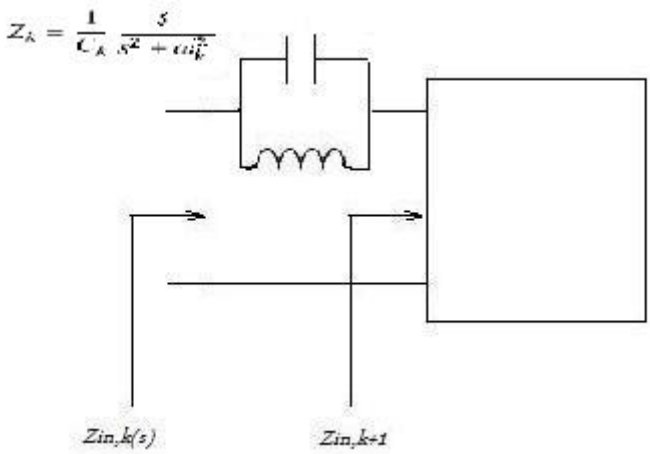


Figure 2.10: Transmission Zero at Nonzero Finite Frequency by Shunt Resonator

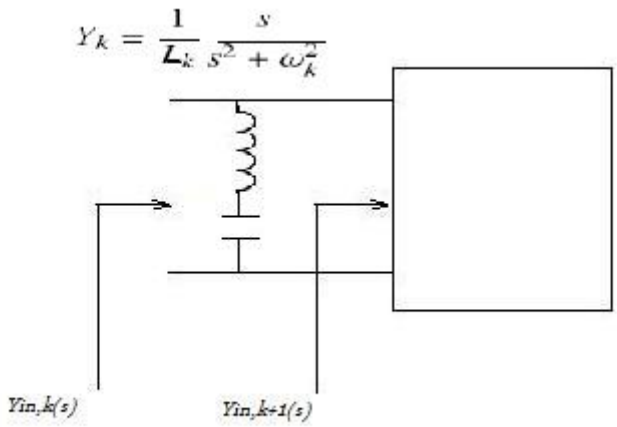


Figure 2.11: Transmission Zero at Nonzero Finite Frequency by Series Resonator

In order to put the input impedance or admittance functions into the forms shown in Equations 2.59 or 2.59, the partial pole removal processes described in the previous section are used.

2.8.4 Transmission zeros produced by Unit Elements

The distributed element filters made up of stubs only are usually unrealizable for several reasons:

- Some stub types may not be realizable with the technology of production used (e. g. series arm OC and SC stubs in stripline).
- Some stub types may be undesirable because of associated problems (e.g. shunt arm SC stubs in microstrip structures).
- Stubs require physical separation to avoid electromagnetic coupling. Furthermore, it is difficult (if not impossible) to connect more than two stubs at a node.
- Contrary to lumped elements, there are severe limitations on realizable element values. Typical dynamic ranges of realizable characteristic impedances are limited to roughly 1: 10.

Fortunately, most of these problems are solvable by equivalent circuit transformations. Some of these problems, like physical separation of adjacent stubs, conversion of series arm stubs to shunt form, etc. are solvable simply manner by inserting transmission line pieces (of the same length as stubs) into circuits by circuit transformations in a redundant manner (not contributing to response). Such transmission line pieces are termed as unit elements (UE). Unit elements have no counterparts in lumped element circuit theory. Though unit elements are (assumed to be) lossless, in reality they have nonnegligible losses and their use makes filters longer. Therefore, whenever possible, it is preferred to employ unit elements in a nonredundant manner (contributing to the response). This is possible by incorporating unit elements into the filter transfer function right at the beginning of synthesis procedure as a transmission zero section. Therefore, in distributed filter frequency variable should be replaced by Richard's variable $\lambda = \Sigma + j\Omega$ [3]. This can be related to the actual frequency variable s as follows:

$$\lambda = tanh\tau s \tag{2.68}$$

Here,

τ : l/v, one way delay time for the commensurate-length.

l : length of the commensurate-length.

v : velocity of propagation.

On the real frequency axis $\Sigma = 0$

$\lambda = j\Omega = jtan\theta$

where,

$$\theta = \frac{\omega l}{v} = 2\pi\frac{f}{v}l = 2\pi\frac{l}{\lambda} \tag{2.69}$$

The length can be written in term the special wavelength, λ_q, which is defined $l = \frac{\lambda_q}{4}$.

That is, λ_q is the wave length of the frequency, f_q where, l becomes quarter wavelength long.

So θ becomes

$$\theta = 2\pi\frac{\lambda_q}{4\lambda} = \frac{\pi}{2}\frac{\lambda_q}{\lambda} \tag{2.70}$$

Using $\lambda_q f_q = \lambda f = v_p$, θ is obtained as

$$\theta = \frac{\pi}{2}\frac{f}{f_q} \tag{2.71}$$

or

$$\theta = \frac{\pi}{2}\frac{\omega}{\omega_q} \tag{2.72}$$

Thus,

$$\Omega = tan\left(\frac{\pi}{2}\frac{f}{f_q}\right) \tag{2.73}$$

The scattering matrix of UE is given as;

$$[S(\lambda)] = \frac{1}{1+\lambda}\begin{bmatrix} 0 & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & 0 \end{bmatrix} = \frac{1}{g(\lambda)}\begin{bmatrix} h(\lambda) & f(\lambda)4 \\ f(\lambda) & \pm h(\lambda) \end{bmatrix} \tag{2.74}$$

$$S_{11} = S_{22} = 0 \tag{2.75}$$

$$S_{21} = S_{12} = \frac{\sqrt{1-\lambda^2}}{1+\lambda} \tag{2.76}$$

Because of the square root of the term in $f(\lambda)$, the UE creates half order TZs at $\lambda = \pm 1$. This transmission zero can be used to shape the response $\alpha(\omega)$ of the filter.

Synthesis of UE is performed using Richard's Theorem.

Richard's Theorem : If $Z(\lambda)$ is a positive real rational function and $\frac{Z_1(\lambda)}{Z_1(1)}$ is not identically equal to λ or $1/\lambda$ then,

$$Z_1'(\lambda) = \frac{Z_1(\lambda) - \lambda Z_1(1)}{Z_1(1) - \lambda Z_1(\lambda)} \tag{2.77}$$

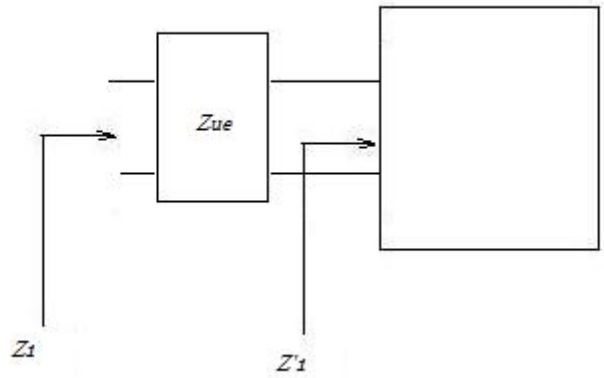


Figure 2.12: Extraction of a UE

is also a positive real rational function and a factor $(\lambda - 1)$ cancels in the numerator and denominator and $Z_1'(\lambda)$ is the same degree as $Z_1(\lambda)$. Further, if $Z_1(1) + Z_1(-1) = 0$, a second factor $(\lambda + 1)$ cancels in the numerator and denominator, and $Z_1'(\lambda)$ is lower degree than $Z_1(\lambda)$. Therefore, for an impedance function, $Z_1(\lambda)$ is odd and $Z_1(1) + Z_1(-1) = 0$ and $\frac{Z_1(\lambda)}{Z_1(1)}$ is not identically equal to λ or $1/\lambda$, $Z_1(1)$ is extracted from $Z_1(\lambda)$. The value of $Z_1'(\lambda)$ can be evaluated as follows

$$Z_1'(\lambda) = Z_1(1)\frac{Z_1(\lambda) - \lambda Z_1(1)}{Z_1(1) - \lambda Z_1(\lambda)} \tag{2.78}$$

2.9 The Transformation from S-Domain to Z-Domain

Filter elements can be extracted from any one of the input immitance parameters like Z_{11} or Y_{11} or Z_{in} where Z_{11} and Y_{11} are open and short circuit two port parameters and Z_{in} is the input impedance when output is loaded. In any case, the roots of numerator and denominator polynomials of these parameters have to be evaluated. However, in s-domain these roots very close to each other leading to great numerical inaccuracy problems and limiting the degree of synthesizable degrees to 5-10. In order to increase the numerical conditioning, transformation method from S-domain to Z-domain is used by using the following equation

$$Z^2 = \frac{s^2 + \omega_{p2}^2}{s^2 + \omega_{p1}^2} \tag{2.79}$$

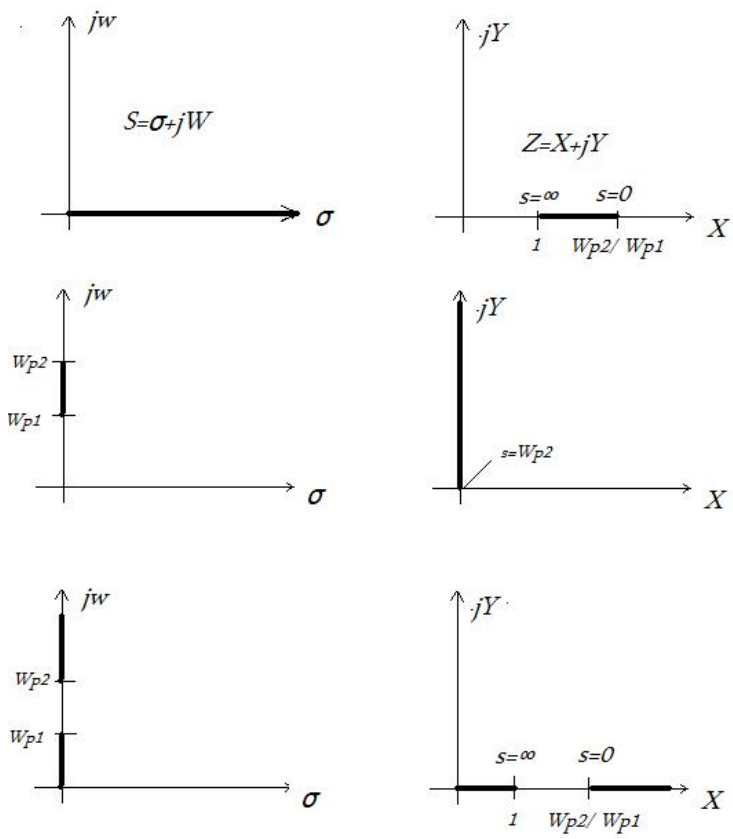


Figure 2.13: Transformation of Different Regions in S-plane to Z-plane

In the Equation 2.79 ω_{p1} and ω_{p2} are the passband edge frequencies. Equation 2.79 can be specialized for the lowpass, highpass and bandpass filter as below.

$$Z^2 = \frac{s^2 + \omega_{p2}^2}{s^2} \qquad \text{L}PF \tag{2.80a}$$

$$Z^2 = \frac{s^2 + \omega_{p2}^2}{s^2} \qquad \text{H}PF \tag{2.80b}$$

$$Z^2 = \frac{s^2 + \omega_{p2}^2}{s^2 + \omega_{p1}^2} \qquad \text{B}PF \tag{2.80c}$$

By this transformation, roots of polynomials in Z-domain are separated from each other more compared to the roots in s-domain, thus easing their solution. Numerical examples are given in detail in references [3] [1] [2].

2.10 Synthesis Formulation in Z-Domain

In order to take full advantage of numerical properties of the transformation from s-domain to Z-domain, the entire synthesis procedure will be performed in Z-domain [3], [1], [2].

The first step is the transformation of transmission zeros from S-domain to Z-domain. This is done by using equation 2.81:

$$Z^2 = \frac{s^2 + \omega_{p2}^2}{s^2 + \omega_{p1}^2} \tag{2.81}$$

Frequencies at infinity corresponds to Z=1,frequencies at zero corresponds to ω_1/ω_2.

After this transformation G(z) function can be formed.

$$G(z) = \prod_{n}^{i=1} \frac{(Z_i - Z)}{Z_i + Z)} = \frac{E_1 - zE_2}{E_1 + zE_2} \tag{2.82}$$

In the equation 2.82, E_1 and E_2 are even polynomial and zE_2 is the odd polynomial. Then F(Z) is formed;

$$F(z) = \frac{1}{2}\left[1 + G(z)\right] \tag{2.83}$$

$$\begin{aligned} F(z)F(-z) &= \frac{1}{4}\left[1 + \frac{E_1 - zE_2}{E_1 + zE_2}\right]\left[1 + \frac{E_1 + zE_2}{E_1 - zE_2}\right] \\ &= \frac{E_1^2}{E_1^2 - z^2E_2^2} \end{aligned} \tag{2.84}$$

After these steps, ϵ is added which is ripple in the passband and characteristic function is observed.

$$|K|^2 = \frac{\epsilon^2 E_1^2}{E_1^2 - z^2E_2^2} \tag{2.85}$$

In S-Domain, characteristic function is formed as shown below.

$$|K|^2 = \frac{h(s)h(-s)}{f(s)f(-s)} \tag{2.86}$$

So ϵE_1 is the z-plane equivalent of h(s) and denominator of characteristic function is equivalent of $f(s)^2$.

Next, the transducer function is obtained.

$$\begin{aligned} |H|^2 &= 1 + |K|^2 \\ &= \frac{(1+\epsilon^2)E_1^2 - z^2E_2^2}{E_1^2 - z^2E_2^2} \end{aligned} \tag{2.87}$$

Numerator of the $|H|^2$ is the z-plane equivalent of g(s)g(-s) and can be factored into two parts as follows.

$$|H|^2 = \left(E_1\sqrt{1+\epsilon^2} + zE_2^2\right)\left(E_1\sqrt{1+\epsilon^2} - zE_2^2\right) \tag{2.88}$$

Then, one of the two factors can be separated, $\left(E_1\sqrt{1+\epsilon^2} + zE_2^2\right)$, of the form $z^2 + mz + n$ and for the odd degree lowpass case z+l. The other part can be form of $z^2 - mz + n$ and for the odd degree lowpass case, z-l.

When the two factors are multiplied, the following form is obtained.

$$Z^4 + MZ^2 + N = (z^2 + mz + n)(z^2 - mz + n) \tag{2.89}$$

For the odd lowpass case (z-l) and (z+l) forms the $Z^2 - L$. The constants of equation 2.89, M, N and L can be defined as follows.

$$M = 2n - m^2 \tag{2.90a}$$

$$N = n^2 \tag{2.90b}$$

$$L = l^2 \tag{2.90c}$$

Then the numerator of the transformer transducer function can be written as:

$$g(Z)g(-Z) = \prod_{i=1}^{n/2}(Z^4 + M_iZ^2 + N_i) \qquad \text{for n even} \tag{2.91a}$$

$$g(Z)g(-Z) = (Z^2 - L)\prod_{i=1}^{(n-1)/2}(Z^4 + M_iZ^2 + N_i) \qquad \text{for n odd} \tag{2.91b}$$

Then, $Z^4 + MZ^2 + N$ is separated into quadratic forms as follows.

$$Z^4 + MZ^2 + N = \left[pZ^2 + q\sqrt{1-a^2Z^2}\sqrt{Z^2-1} + r\right]\left[pZ^2 - q\sqrt{1-a^2Z^2}\sqrt{Z^2-1} + r\right] \tag{2.92}$$

By using equation 2.92 p, q and r can be found as below.

$$R = \sqrt{1 + M + N} \tag{2.93a}$$

$$T = \sqrt{1 + a^2 M + a^4 N} \tag{2.93b}$$

$$p = \frac{1 + a^2(1 + M)}{T + a^2 R} \tag{2.93c}$$

$$q = \sqrt{\frac{r(r - Mp) + Np^2}{TR}} \tag{2.93d}$$

$$r = \frac{M + (1 + a^2 N)}{T + R} \tag{2.93e}$$

The relations of equation 2.93 is explained in detail [3]. Next the all factors in the form $pZ^2 + q\sqrt{1 - a^2 Z^2}\sqrt{Z^2 - 1} + r$ and for the odd degree lowpass $\sqrt{z^2 - 1} + \sqrt{l^2 - 1}$ are multiplied together to obtain expression;

$$A(Z^2) + B(Z^2)\sqrt{1 - a^2 Z^2}\sqrt{Z^2 - 1} \tag{2.94}$$

where A and B are polynomials in z^2. The equation 2.94 is the z-plane equivalent of g(s). When n is even A corresponds $g_e(s)$ and $B(Z^2)\sqrt{1 - a^2 Z^2}\sqrt{Z^2 - 1}$ corresponds to $g_o(s)$. Reversely when n is odd, A corresponds the $g_e(s)$ and $B(Z^2)\sqrt{1 - a^2 Z^2}\sqrt{Z^2 - 1}$ corresponds to $g_e(s)$. Also the constant factor $1 + \epsilon^2$ should be taken account.

2.11 Element Extraction in Z-Domain

After forming $Z_{in}(z)$, a ladder type network have to be synthesized, it's series and shunt arms must realize the assigned transmission zeros[3],[10].

In the synthesis process, impedance function may occur in four different forms.

Type-1 :poles at s=0 (Z=1/a) and $s = \infty$ (Z=1)

$$Z_{in} = \frac{N}{D\sqrt{1 - a^2 Z^2}\sqrt{Z^2 - 1}} \tag{2.95}$$

Type-2 : poles at $s = \infty$ (Z=1) and zero at s=0 (Z=1/a)

$$Z_{in} = \frac{N\sqrt{1 - a^2 Z^2}}{D\sqrt{Z^2 - 1}} \tag{2.96}$$

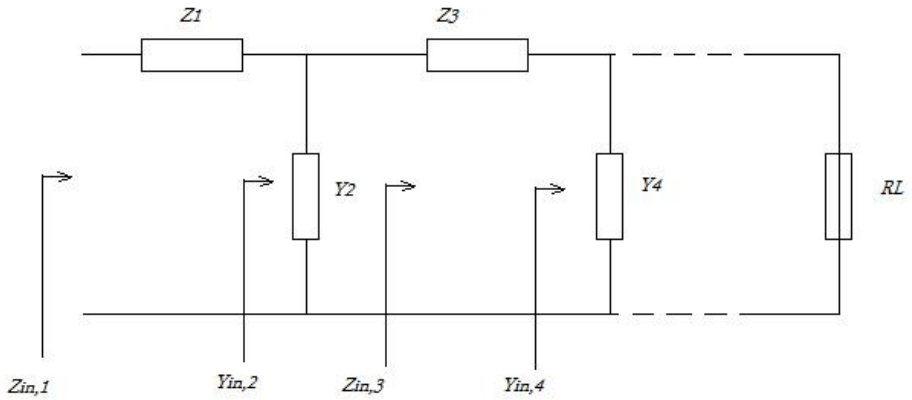


Figure 2.14: Realization of Transmission Zeros

Type-3 : poles at s=0 (Z=1/a) and zero at $s = \infty$ (Z=1)

$$Z_{in} = \frac{N\sqrt{Z^2 - 1}}{D\sqrt{1 - a^2Z^2}} \tag{2.97}$$

Type-4 :zeros at s=0 (Z=1/a) and $s = \infty$ (Z=1)

$$Z_{in} = \frac{N\sqrt{1 - a^2Z^2}\sqrt{Z^2 - 1}}{D} \tag{2.98}$$

When poles are removed partially from Z_{in} the remaining impedance will have a zero at the next transmission frequency. So that $Y_{in,k+1}$ will have a pole at $j\omega k + 1$. Also when poles are removed from Y_{in} the remaining impedance will have a zero at the next transmission frequency. So that $Z_{in,k+1}$ will have a pole at $j\omega k + 1$.

2.12 Generalization of Impedance Function by Element Extraction

For the circuit types given at the previous section can be realizable when the impedance function is formed at equations 2.95, 2.96, 2.97, 2.98. For all circuit types before and after extraction are listed in Table 2.1.

Table 2.1: Type of Impedance Function Before and After Extraction of Related Circuit Block

Circuit Block	Types of Z_1	Types of Z_2
Series Ind.	1	3
	2	4
Series Cap.	1	2
	3	4
Shunt Ind.	2	1
	4	3
Shunt Cap.	3	1
	4	2
Series Cap. Shunt SLC	1	1
	3	3
Series Ind. Shunt SLC	1	1
	2	2
Shunt Cap. Series PLC	3	3
	4	4
Shunt Ind. Series PLC	2	2
	4	4

And the impedances of an inductor, a capacitor, a series resonant circuit and parallel circuit as shown as below.

$$Z_L = \frac{\sqrt{1-a^2Z^2}}{\sqrt{Z^2-1}}L \quad \text{Inductor} \tag{2.99a}$$

$$Z_C = \frac{1}{C}\frac{\sqrt{Z^2-1}}{\sqrt{1-a^2Z^2}} \quad \text{Capacitor} \tag{2.99b}$$

$$Z_{sr} = \frac{(1-a^2)(Z^2-Z_s^2)}{C_s(1-Z_s^2a^2)\sqrt{Z^2-1}\sqrt{1-a^2Z^2}} \quad \text{Series Resonator} \tag{2.99c}$$

$$Z_{pr} = \frac{L_p(1-Z_p^2a^2)\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{(1-a^2)(Z^2-Z_p^2)} \quad \text{Parallel Resonator} \tag{2.99d}$$

Now general formulas can be found by using equations 2.99. For example, a series inductor can be extracted from impedance, which has poles at zero and infinity. It corresponds to Type1. After extraction Type3 circuit is obtained as shown table 2.1. It can be started by defining relationship of impedances.

$$Z_1 = Z_2 + Z_L \tag{2.100}$$

where, Z_1 = impedance before extraction

Z_2 = impedance after extraction

Z_L = impedance of inductor

Now Z_2 is found

$$\begin{aligned} Z_2 &= \frac{N}{D\sqrt{1-a^2Z^2}\sqrt{Z^2-1}} - \frac{\sqrt{1-a^2Z^2}}{\sqrt{Z^2-1}}L \\ &= \frac{N - LD(1-a^2Z^2)}{D\sqrt{1-a^2Z^2}\sqrt{Z^2-1}} \end{aligned} \tag{2.101}$$

When the equations 2.97 and 2.101 are equated, following equation is obtained;

$$\mathbf{N}(Z^2-1) = N - LD(1-a^2Z^2) \tag{2.102}$$

So the only one equation must be solved.

$$N = \mathbf{N}(Z^2-1) + LD(1-a^2Z^2) \tag{2.103}$$

Nonzero finite transmission zeros can be removed as series resonance circuit using the same approach. For example, there is a Type3 circuit and series resonance circuit has to be extracted. As mentioned before, Type3 circuit has pole at zero and zero at infinity. First of all, a pole is removed partially at zero to create a pole of admittance at $Z^2 = m^2$ and this pole is removed as series resonant circuit in shunt arm.

$$Z_1 = \frac{N\sqrt{Z^2-1}}{D\sqrt{1-a^2Z^2}} \tag{2.104}$$

$$\begin{aligned} Z_1 &= Z_2 + Z_{c1} \\ &= \frac{\mathbf{N}\sqrt{Z^2-1}(Z^2-m^2)}{D\sqrt{1-a^2Z^2}} + \frac{1}{C_1}\frac{\sqrt{Z^2-1}}{\sqrt{1-a^2Z^2}} \\ &= \frac{\left[\mathbf{N}(Z^2-m^2) + (1/C_1)D\right]\sqrt{Z^2-1}}{D\sqrt{1-a^2Z^2}} \end{aligned} \tag{2.105}$$

Then, the Z_1 and 2.105 are equated to get following equation;

$$N = \mathbf{N}(Z^2-m^2) + (1/C_1)D \tag{2.106}$$

$$1/C_1 = (N/D)|_{Z^2=m^2} \tag{2.107}$$

After removing the series capacitor, Z_2 and C_s are. Next series resonator circuit can be removed.

$$Y_2 = Y_3 + Ysr \tag{2.108}$$

Y_2 can be written as:

$$\begin{aligned} Y_2 &= \frac{D\sqrt{1-a^2Z^2}}{\mathbf{N}(Z^2-m^2)\sqrt{Z^2-1}} \\ &= \frac{D\sqrt{1-a^2Z^2}}{N\sqrt{Z^2-1}} + \frac{C_s(1-m^2a^2)\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{(1-a^2)(Z^2-m^2)} \end{aligned} \tag{2.109}$$

If α is defined as;

$$\alpha = C_s\frac{1-m^2a^2}{1-a^2} \tag{2.110}$$

then Y_3 is found like this,

$$Y_3 = \frac{\left[D-\alpha(Z^2-1)\mathbf{N}\right]\sqrt{1-a^2Z^2}}{\mathbf{N}(Z^2-m^2)\sqrt{Z^2-1}} \tag{2.111}$$

Now, the equation 2.109 and 2.111 can be equated.

$$D = \alpha(Z^2-1)\mathbf{N} + (Z^2-m^2)\mathbf{D} \tag{2.112}$$

When the equations 2.103, 2.105 and 2.112 are analyzed, a general formula can be found as below.

$$U(Z^2) = a_1(pZ^2+q)V(Z^2) + (Z^2-m^2)W(Z^2) \tag{2.113}$$

where the U, V, p, q and m are known and a_1 and W are unknowns. Now unknown coefficients can be solved by same algorithm as shown below.

$$a_1 = \frac{U}{(pZ^2+q)V} \tag{2.114}$$

Next W can be found as below,

$$W = \frac{U-a_1(pZ^2+q)V}{Z^2+-m^2} \tag{2.115}$$

The form of the impedance functions, before and after extraction as listed in Table 2.2

Table 2.2: Forms of te impedance function before and after extraction of circuit blocks [3]

CIRCUIT BLOCK	Impedance before extraction	PARTIAL REMOVAL		FULL REMOVAL	
		Eqn. to be solved	Impedance after zero shifting	Eqn. to be solved	Impedance after full removal
Series Inductor(L) $L = a_1$	$\frac{N}{D\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$	—-	—-	$N = a_1(1-a^2Z^2) + (Z^2-1)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{Z^2-1}}{D\sqrt{1-a^2Z^2}}$
	$\frac{N\sqrt{1-a^2Z^2}}{D\sqrt{Z^2-1}}$	—-	—-	$N = a_1D + (Z^2-1)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{D}$
Shunt Capacitor(C) $C = a_1$	$\frac{N\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{D}$	—-	—-	$D = a_1(1-a^2Z^2)N + (Z^2-1)\mathbf{D}$	$\frac{N\sqrt{1-a^2Z^2}}{\mathbf{D}\sqrt{Z^2-1}}$
	$\frac{N\sqrt{Z^2-1}}{D\sqrt{1-a^2Z^2}}$	—-	—-	$D = a_1N + (Z^2-1)\mathbf{D}$	$\frac{N}{\mathbf{D}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$
Shunt Inductor(L) $L = 1/a_1$	$\frac{N\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{D}$	—-	—-	$D = a_1(Z^2-1)N + (1-a^2Z^2)\mathbf{D}$	$\frac{N\sqrt{Z^2-1}}{\mathbf{D}\sqrt{1-a^2Z^2}}$
	$\frac{N\sqrt{1-a^2Z^2}}{D\sqrt{Z^2-1}}$	—-	—-	$D = a_1N + (1-a^2Z^2)\mathbf{D}$	$\frac{N}{\mathbf{D}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$
Series Capacitor(C) $C = 1/a_1$	$\frac{N}{D\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$	—-	—-	$N = a_1(Z^2-1)D + (1-a^2Z^2)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{1-a^2Z^2}}{D\sqrt{Z^2-1}}$
	$\frac{N\sqrt{Z^2-1}}{D\sqrt{1-a^2Z^2}}$	—-	—-	$N = a_1D + (1-a^2Z^2)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{D}$

Continued on next page

Table 2.2 Continued

Series Ind.(L_1) Shunt SLC(L_2C_2) $L_1 = a_1$ $C_2 = a_2/\alpha$	$\frac{N}{D\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$	$N = a_1(1-a^2Z^2)D + (Z^2-m^2)\mathbf{N}$	$\frac{\mathbf{N}(Z^2-m^2)}{D\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$	$D = a_2N + (Z^2-m^2)\mathbf{D}$	$\frac{\mathbf{N}}{\mathbf{D}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$
	$\frac{N\sqrt{1-a^2Z^2}}{D\sqrt{Z^2-1}}$	$N = a_1D + (Z^2-m^2)\mathbf{N}$	$\frac{\mathbf{N}(Z^2-m^2)\sqrt{1-a^2Z^2}}{D\sqrt{Z^2-1}}$	$D = a_2(1-a^2Z^2)\mathbf{N} + (Z^2-m^2)\mathbf{D}$	$\frac{\mathbf{N}\sqrt{1-a^2Z^2}}{\mathbf{D}\sqrt{Z^2-1}}$
Shunt Cap(C_1) Ser. PLC(L_2C_2) $C_1 = a_1$ $L_2 = a_2/\alpha$	$\frac{N\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{D}$	$D = a_1(1-a^2Z^2)N + (Z^2-m^2)\mathbf{D}$	$\frac{N\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{\mathbf{D}(Z^2-m^2)}$	$N = a_2\mathbf{D} + (Z^2-m^2)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{\mathbf{D}}$
	$\frac{N\sqrt{Z^2-1}}{D\sqrt{1-a^2Z^2}}$	$D = a_1N + (Z^2-m^2)\mathbf{D}$	$\frac{N\sqrt{Z^2-1}}{\mathbf{D}\sqrt{1-a^2Z^2}(Z^2-m^2)}$	$N = a_2(1-a^2Z^2)\mathbf{D} + (Z^2-m^2)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{Z^2-1}}{\mathbf{D}\sqrt{1-a^2Z^2}}$
Shunt Ind(L_1) Ser. PLC(L_2C_2) $L_1 = 1/a_1$ $C_2 = a_2/\alpha$	$\frac{N\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{D}$	$D = a_1(Z^2-1)N + (Z^2-m^2)\mathbf{D}$	$\frac{N\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{\mathbf{D}(Z^2-m^2)}$	$N = a_2\mathbf{D} + (Z^2-m^2)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}{\mathbf{D}}$
	$\frac{N\sqrt{1-a^2Z^2}}{D\sqrt{Z^2-1}}$	$D = a_1N + (Z^2-m^2)\mathbf{D}$	$\frac{N\sqrt{1-a^2Z^2}}{\mathbf{D}\sqrt{Z^2-1}(Z^2-m^2)}$	$N = a_2(Z^2-1)\mathbf{D} + (Z^2-m^2)\mathbf{N}$	$\frac{\mathbf{N}\sqrt{1-a^2Z^2}}{\mathbf{D}\sqrt{Z^2-1}}$
Series Cap(C_1) Shunt PLC (L_2C_2) $C_1 = 1/a_1$ $L_2 = a_2/\alpha$	$\frac{N}{D\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$	$N = a_1(Z^2-1)D + (Z^2-m^2)\mathbf{N}$	$\frac{\mathbf{N}(Z^2-m^2)}{D\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$	$D = a_2N + (Z^2-m^2)\mathbf{D}$	$\frac{\mathbf{N}}{\mathbf{D}\sqrt{Z^2-1}\sqrt{1-a^2Z^2}}$
	$\frac{N\sqrt{Z^2-1}}{D\sqrt{1-a^2Z^2}}$	$N = a_1D + (Z^2-m^2)N$	$\frac{\mathbf{N}\sqrt{Z^2-1}(Z^2-m^2)}{D\sqrt{1-a^2Z^2}}$	$D = a_2(Z^2-1)N + (Z^2-m^2)\mathbf{D}$	$\frac{\mathbf{N}\sqrt{Z^2-1}}{\mathbf{D}\sqrt{1-a^2Z^2}}$

$\alpha = \frac{1-a^2m^2}{1-a^2}$; $m^2 = \frac{f_i^2-f_{p2}^2}{f_i^2-f_{p1}^2}$; $a = \frac{f_{p1}}{f_{p2}}$

If the equations are rewrited in an open form, some terms cancel each other and better equations can be found. Now let's look how cancellations occur when the equations are written in an open form. [3] [1]

First of all U and V polynomials are written in an open form.

$$U(Z^2) = u_0 + u_1Z^2 + u_2Z^4 + u_3Z^6 + u_4Z^8 + u_5Z^{10} \tag{2.116a}$$

$$V(Z^2) = v_0 + v_1Z^2 + v_2Z^4 + v_3Z^6 + v_4Z^8 + v_5Z^{10} \tag{2.116b}$$

$$W(Z^2) = w_0 + w_1Z^2 + w_2Z^4 + w_3Z^6 + w_4Z^8 \tag{2.116c}$$

$W(Z^2)$ can be a polynomial of degree 8 when three different case occur as below.

(i) p=0, q=1

(ii) $p = 1, q = -1, v_5 = 0$

(iii) $p = -a^2, q = 1, v_5 = 0$

Now the $W(Z^2)$ can be evaluated as below;

$$\begin{aligned} W(Z^2 - m^2) = {} & [u_5 - a_1(pv_4 + qv_5)]Z^{10} + [u_4 - a_1(pv_3 + qv_4)]Z^8 \\ & + [u_3 - a_1(pv_2 + qv_3)]Z^6 + [u_2 - a_1(pv_1 + qv_2)]Z^4 \\ & + [u_1 - a_1(pv_0 + qv_1)]Z^2 + [u_0 - a_1(qv_0)] \end{aligned} \tag{2.117}$$

When the polynomial is divided by $(Z^2 - m^2)$ coefficient, the $W(Z^2)$ coefficients are found as below.

$$\begin{aligned} w_4 &= [u_5 - a_1(pv_4 + qv_5)] \\ w_3 &= [u_4 - a_1(pv_3 + qv_4)] + [u_5 - a_1(pv_4 + qv_5)]m^2 \\ w_2 &= [u_3 - a_1(pv_2 + qv_3)] + [u_4 - a_1(pv_3 + qv_4)]m^2 + [u_5 - a_1(pv_4 + qv_5)]m^4 \\ w_1 &= [u_2 - a_1(pv_1 + qv_2)] + [u_3 - a_1(pv_2 + qv_3)]m^2 + [u_4 - a_1(pv_3 + qv_4)]m^4 \\ &\quad + [u_5 - a_1(pv_4 + qv_5)]m^6 \\ w_0 &= [u_1 - a_1(pv_0 + qv_1)] + [u_2 - a_1(pv_1 + qv_2)]m^2 + [u_3 - a_1(pv_2 + qv_3)]m^4 \\ &\quad + [u_4 - a_1(pv_3 + qv_4)]m^6 + [u_5 - a_1(pv_4 + qv_5)]m^8 \end{aligned} \tag{2.118}$$

If a_1 is replaced by $\frac{U(m^2)}{(pm^2+q)V(m^2)}$ in the W coefficients and each side is multiplied by $(pm^2 + q)V(m^2)$, the following equation are found.

$$\begin{aligned} w_2(pm^2 + q)V(m^2) = {} & [u_3(pm^2 + q)V(m^2) - U(m^2)(pv_2 + qv_3)] \\ & + [u_4(pm^2 + q)V(m^2) - U(m^2)(pv_3 + qv_4)]m^2 \\ & + [u_5(pm^2 + q)V(m^2) - U(m^2)(pv_4 + qv_5)]m^4 \end{aligned} \tag{2.119}$$

and the equation is rewrited in an open form,

$$\begin{aligned} w_2(pm^2 + q)V(m^2) = {} & (pm^2 + q)(v_0 + v_1m^2 + v_2m^4 + v_3m^6 + v_4m^8 + v_5m^{10}). \\ & (u_3 + u_4m^2 + u_5m^4) \\ & - (u_0 + u_1m^2 + u_2m^4 + u_3m^6 + u_4m^8 + u_5m^{10}). \\ & [pv_2 + (pm^2 + q)(v_3 + v_4m^2 + v_5m^4)] \end{aligned} \tag{2.120}$$

The bold terms cancel each other. After the cancellation the w_2 coefficients are found as follows.

$$\begin{aligned} w_2(pm^2 + q)V(m^2) = {} & [(pm^2 + q)(v_0 + v_1m^2) + qv_2m^4)](u_3 + u_4m^2 + u_5m^4) \\ & - (u_0 + u_1m^2 + u_2m^4)[pv_2 + (pm^2 + q)(v_3 + v_4m^2 + v_5m^4)] \end{aligned} \tag{2.121}$$

Other coefficients can be found in a similar way. These coefficients listed as below.

$$
\begin{aligned}
\Delta &= (pm^2 + q)(v_0 + v_1m^2 + v_2m^4 + v_3m^6 + v_4m^8 + v_5m^{10}) \\
a_1\Delta &= (u_0 + u_1m^2 + u_2m^4 + u_3m^6 + u_4m^8 + u_5m^{10}) \\
w_0\Delta &= (u_1 + u_2m^2 + u_3m^4 + u_4m^6 + u_5m^8)qv_0 \\
&\quad - u_0[pv_0 + (pm^2 + q)(v_1 + v_2m^2 + v_3m^4 + v_4m^6 + v_5m^8)] \\
w_1\Delta &= (u_2 + u_3m^2 + u_4m^4 + u_5m^6)[(pm^2 + q)v_0 + qv_1m^2] \\
&\quad - (u_0 + u_1m^2)[pv_1 + (pm^2 + q)(v_2 + v_3m^2 + v_4m^4 + v_5m^6)] \\
w_2\Delta &= (u_3 + u_4m^2 + u_5m^4)[(pm^2 + q)(v_0 + v_1m^2) + qv_2m^4] \\
&\quad - (u_0 + u_1m^2 + u_2m^4)[pv_2 + (pm^2 + q)(v_3 + v_4m^2 + v_5m^4)] \\
w_3\Delta &= (u4 + u_5m^2)[(pm^2 + q)(v_0 + v_1m^2 + v_2m^4) + qv_3m^6] \\
&\quad - (u_0 + u_1m^2 + u_2m^4 + u_3m^6)[pv_3 + (pm^2 + q)(v_4 + v_5m^2)] \\
w_4\Delta &= (u5)[(pm^2 + q)(v_0 + v_1m^2 + v_2m^4 + v3m^6) + qv_4m^8] \\
&\quad - (u_0 + u_1m^2 + u_2m^4 + u_3m^6 + u_4m^8)[pv_4 + (pm^2 + q)(v_5)]
\end{aligned}
\tag{2.122}
$$

By using these general solutions, a new polynomial can be found after extraction.

2.13 Distributed Parameter Filters

For distributed filters, frequencies transformed using Richards' transformation. All considerations and derivations for $s = \sigma + j\omega$ are also valid for $\lambda = \Sigma + j\Omega$. For real frequencies Ω is expressed as below.

$\Omega = tan\left(\frac{\pi}{2}\frac{f}{f_q}\right)$

Bandpass edge frequencies can be transferred as below.

$$\Omega_{p1} = tan\left(\frac{\pi}{2}\frac{f_{p1}}{f_q}\right) \tag{2.123}$$

$$\Omega_{p2} = tan\left(\frac{\pi}{2}\frac{f_{p2}}{f_q}\right) \tag{2.124}$$

Ω_{p2} is not much greater than 1, generally less than 1, since f_q is chosen to be greater than $2xf_{p2}$.

Therefore, Z-transformation can be performed like this,

$$Z^2 = \frac{\Omega^2 - \Omega_{p2}^2}{\Omega^2 - \Omega_{p1}^2} \tag{2.125}$$

Also TZ frequencies f_i must be transformed by Richards' transformation.

$$\Omega_i = tan\left(\frac{\pi}{2}\frac{f_i}{f_q}\right) \tag{2.126}$$

TZs at s=0 maps to $\lambda = 0$, but TZs at infinity is indefinite. They are replaced by TZs $s = j\omega$. Synthesis procedure is almost same with the lumped element synthesis, while the inductors and capacitors are replaced by short and open circuited stubs respectively.

2.13.1 Synthesis UE in Z-Domain

Synthesis of a UE in Z-plane does not differ from the one in the λ-plane. Characteristic impedance of the UE is evaluated from the value of impedance function at Z=k, where k is the value at Z-plane corresponding the $\lambda = 1$ in λ-plane. k is evaluated as [3];

$$k = \sqrt{\frac{1 + \Omega_{p2}^2}{1 + \Omega_{p1}^2}} \tag{2.127}$$

After evaluation of the characteristic impedance of the UE and the remaining impedance function, for the four types of the impedance function given in 2.95-2.98.

$$Z_a = \sqrt{k^2 - 1} \tag{2.128}$$

$$Z_b = \sqrt{\Omega_{p2}^2 - \Omega_{p1}^2 k^2} \tag{2.129}$$

Type-1

$$Z_1 = \frac{N}{D\sqrt{Z^2 - 1}\sqrt{\Omega_{p2}^2 - \Omega_{p1}^2 Z^2}}$$

$$Z_{UE} = Z_1(Z)|_{Z=k} = \frac{N_k}{D_k Z_a Z_b}$$

N_k and D_k are the values of polynomials, N, D respectively at Z=k. Applying Richards' theorem to obtain the remaining impedance function,

$$Z_1' = Z_{UE}\frac{Z_1 - \lambda Z_{UE}}{Z_{UE} - \lambda Z_1}$$

After evaluating the equations, Z_1' is found,

$$Z_1' = \frac{\mathbf{N}\sqrt{Z^2-1}}{\mathbf{D}\sqrt{\Omega_{p2}^2-\Omega_{p1}^2 Z^2}}$$

where,

$$\mathbf{N} = Z_{UE}\left[ND_kZ_aZ_b - DN_k(\Omega_{p2}^2-\Omega_{p1}^2Z^2)\right]/(Z^2-k^2)$$

$$\mathbf{D} = [DN_k(Z^2-1) - ND_kZ_aZ_b]/(Z^2-k^2)$$

The remaining function is of the form Type-3.

Type-2

$$Z_1 = \frac{N\sqrt{\Omega_{p2}^2-\Omega_{p1}^2Z^2}}{D\sqrt{Z^2-1}}$$

$$Z_{UE} = Z_1(Z)|_{Z=k} = \frac{N_kZ_b}{D_kZ_a}$$

After evaluation, Z_1' is found,

$$Z_1' = \frac{\mathbf{N}\sqrt{Z^2-1}\sqrt{\Omega_{p2}^2-\Omega_{p1}^2Z^2}}{\mathbf{D}}$$

where,

$$\mathbf{N} = Z_{UE}[Z_{UE}D - N]/(Z^2-k^2)$$

$$\mathbf{D} = [N(\Omega_{p2}^2-\Omega_{p1}^2Z^2) - Z_{UE}D(Z^2-1)]/(Z^2-k^2)$$

Type-3

$$Z_1 = \frac{N\sqrt{Z^2-1}}{D\sqrt{\Omega_{p2}^2-\Omega_{p1}^2Z^2}}$$

$$Z_{UE} = Z_1(Z)|_{Z=k} = \frac{N_kZ_a}{D_kZ_b}$$

After evaluation, Z_1' is found,

$$Z_1' = \frac{\mathbf{N}}{\mathbf{D}\sqrt{Z^2-1}\sqrt{\Omega_{p2}^2-\Omega_{p1}^2Z^2}}$$

where,

$$\mathbf{N} = Z_{UE}[Z_{UE}D(\Omega_{p2}^2-\Omega_{p1}^2Z^2) - N(Z^2-1)]/(Z^2-k^2)$$

$$\mathbf{D} = (N - DZ_{UE})/(Z^2-k^2)$$

Type-4

$$Z_1 = \frac{N\sqrt{Z^2-1}\sqrt{\Omega_{p2}^2-\Omega_{p1}^2 Z^2}}{D}$$

$$Z_{UE} = Z_1(Z)|_{Z=k} = \frac{N_k}{D_k} Z_a Z_b$$

After evaluation, Z_1' is found,

$$Z_1' = \frac{\mathbf{N}\sqrt{\Omega_{p2}^2-\Omega_{p1}^2 Z^2}}{\mathbf{D}\sqrt{Z^2-1}}$$

where,

$$\mathbf{N} = Z_{UE}[Z_{UE}D - N(Z^2-1)]/(Z^2-k^2)$$

$$\mathbf{D} = (N(\Omega_{p2}^2-\Omega_{p1}^2 Z^2) - DZ_{UE})/(Z^2-k^2)$$

CHAPTER 3

IMPLEMENTATION

In this chapter, implementation details of the software will be described. First, the background of the program and development environment will be explained. Then design of filters in accordance with theory and formulations will be discussed step by step.

3.1 Synthesis Program

The synthesis program consists three parts. One of these parts is *Core Synthesis*, which is actually main part of program. In this part, the desired filter specifications are evaluated and circuit is constructed. Second part of this program is *User Interface*. It handles interactions with the user and other sub-programs. Last part is *Plot Diagrams*, which is used to plot filter responses. In the following sections, all of these parts will be described in details.

3.1.1 Core Synthesis

Core Synthesis part of program, is the main part of the program. It obtains the specified filter parameter from the user through the *User Interface* and analyzes the specifications. After analysis, the circuit is constructed and circuit elements values are calculated. The formulations described in Chapter 2 are used in the formation of the transfer function and in element extraction parts of the software. *Core Synthesis* is written by C# programming languages. C# is object oriented programming language and provides speed, generic and hierarchic programming features. By these features, *Core Synthesis* is programmed independent from other parts of program.

In addition to *User Interface* and *Plot Diagrams* parts, another computer program can use

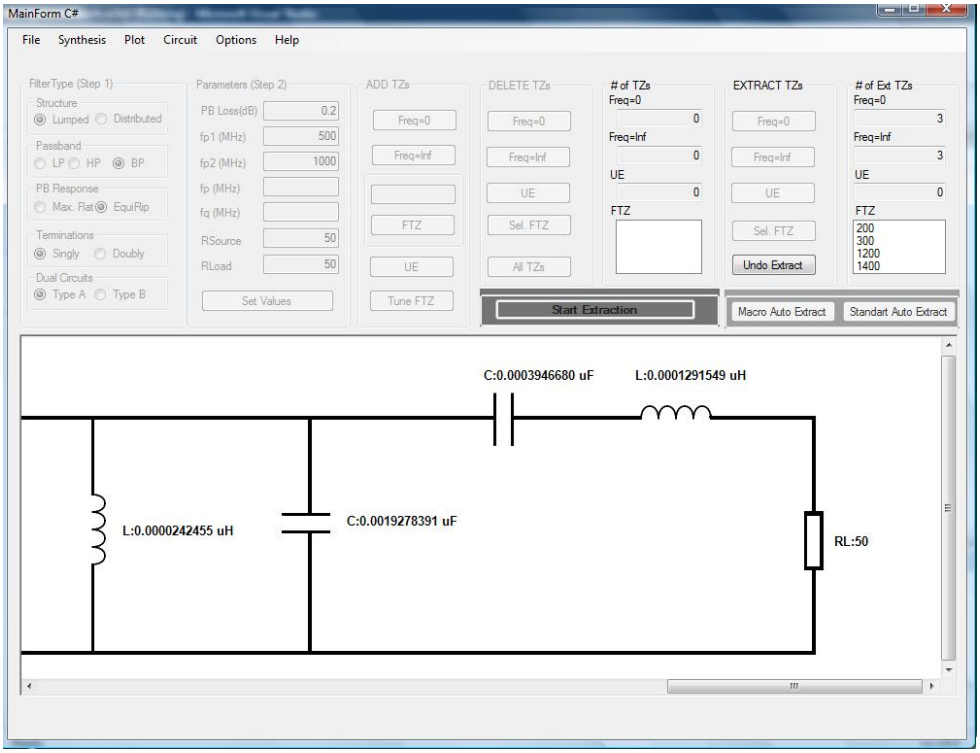


Figure 3.1: Synthesis Program Main User Interface

Core Synthesis part as library. This is the most powerful features of *Core Synthesis* part. It provides necessary interface to other programs and return generated circuits details and filter responses. It can also be generated in *.dll* (Dynamic Link Library) format and used by the other programs, which may be written in different programming languages.

3.1.2 User Interface

User Interface provides interface between user and *Core Synthesis*. Desired filter specifications are entered by this part and results of filters is shown to user. It is written in C# .NET language by using *Microsoft Visual Studio .NET 2005* development tool. Also GDI (Graphic Device Interface) is used to draw circuits. It handles user data as input data for the *Core Synthesis* and executes it. After the execution of the *Core Synthesis*, it shows results, returns back from *Core Synthesis*, to user and obtains next inputs to complete the filter design process. All specified filter properties and designed filters can be saved and loaded by XML (Extensible Markup Language) files.

3.1.3 Plot Diagrams

Last part of program is *Plot Diagrams*. It is written in C# .NET programming language and developed in *Microsoft Visual Studio* development tools. It also uses the *Graphics Device Interface (GDI)*, which is one of the three core components or ”subsystems”, together with the kernel and the Windows API for the user interface of *Microsoft Windows*. All calculations and necessary information are obtained from *Core Synthesis*. After extraction of all elements, *Core Synthesis* can evaluate seven different responses. These are *Insertion Loss, Return Loss, Delay-S21, Phase-S21, Delay-S11, Phase-S11, Smith Chart*. *Plot Diagrams* section can plot these responses in any frequency and magnitude range. In Figure 3.2, *Plot Diagrams* user interface is shown. Like other filter responses, Smith Chart data is also evaluated by the *Core Synthesis* and the Smith Chart diagram is drawn by *Plot Diagrams*. Besides Smith Chart diagram, the parameters like VSWR, Zin, Yin, S11 are also calculated at relevant frequencies.

3.2 Filter Types Section

Designing filter starts by defining filter properties. In this section desired filter properties entered by the user. The user can select one of displayed options or type in the necessary parameters. Figure 3.3 is the data input section of filter properties.

3.2.1 Structure Command

In this command user selects either lumped element or distributed element option.

The other synthesis procedure is the almost same with the lumped element synthesis. The relation between lumped and distributed elements is shown in figure 3.4.

3.2.2 Passband Command

By this command, user can choice filter passband types. There are three different passband types can be selected: Lowpass Filter, Highpass Filter and Bandpass Filter.

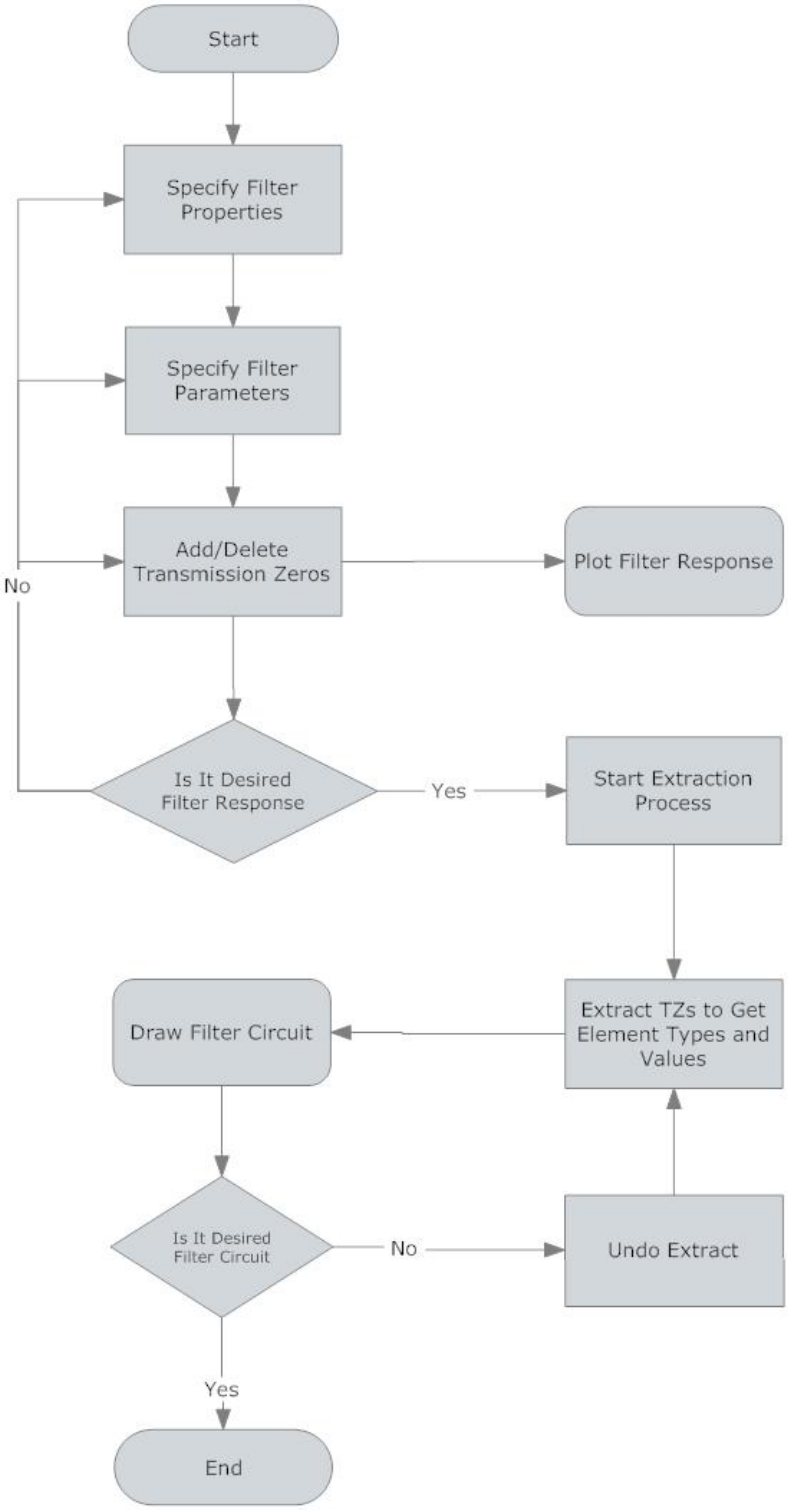


Figure 3.2: General Flow Diagram of Synthesis Program

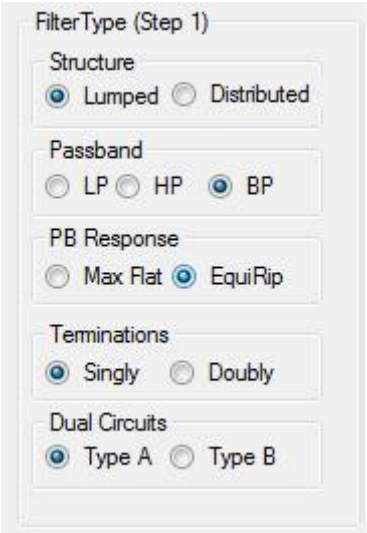


Figure 3.3: Filter Properties

When user chooses lowpass filter or highpass filter, only one passband edge frequency can be typed in. For bandpass filters, two nonzero frequencies must be entered as the passband edge frequencies. For distributed filter option, user must also enter the quarter wave frequency and we need to calculate normalized frequencies in a different way from equation 2.79.

$$N_f = tan\left(\frac{\pi}{2}\frac{f}{f_q}\right) \tag{3.1}$$

3.2.3 Passband Response Command

In synthesis program, two different filters can be designed according to passband responses. These are Maximally Flat and Equiripple. The information about this filter types described in section 2.4. When the user chooses one of the two options the rest of the synthesis formulation calculated according to this selection.

3.2.4 Terminations Command

User can select either singly terminated or doubly terminated options. These options are described in figures 3.2.4 and 3.6.

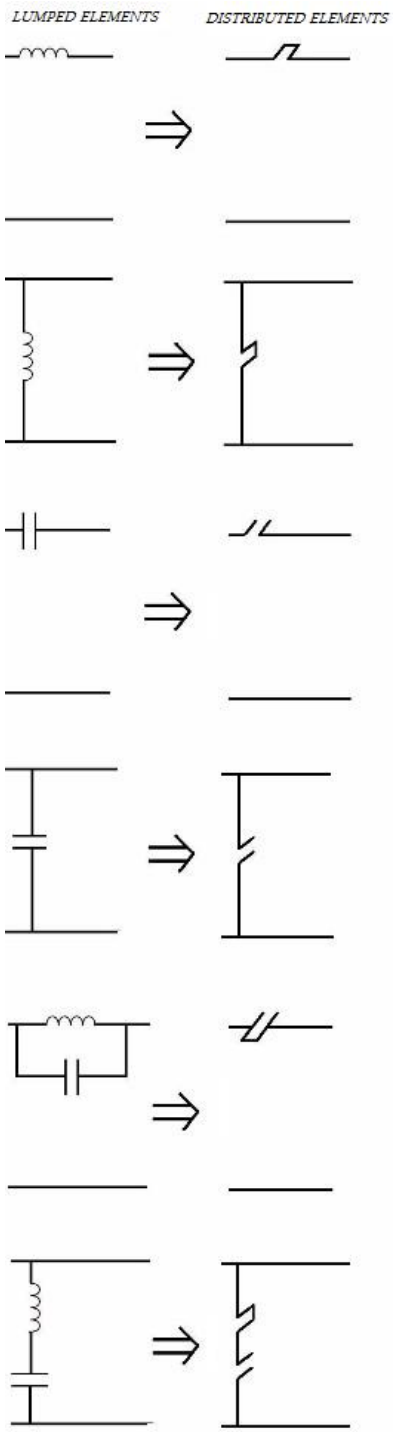


Figure 3.4: Lumped and Distributed Elements

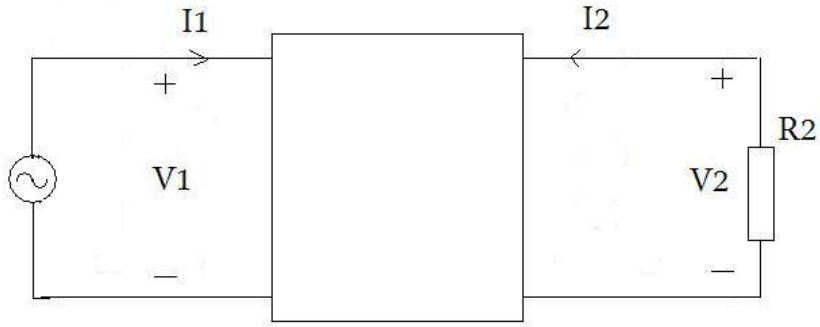


Figure 3.5: A Singly Terminated Two Port.

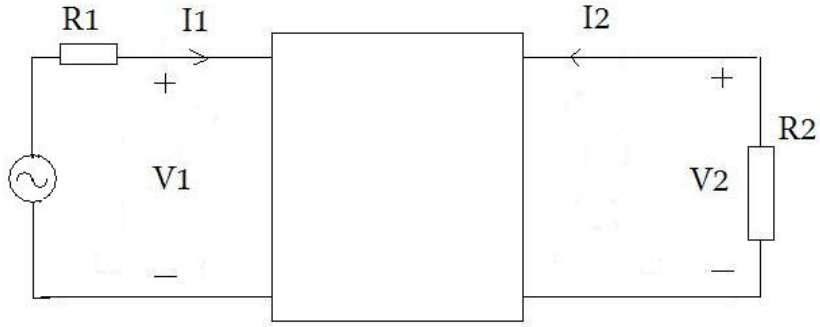


Figure 3.6: A Doubly Terminated Two Port.

3.2.5 Dual Circuits Command

Transmission zeros can be realized by two different sub networks. These are listed as below.

3.2.5.1 Type A Networks

Type A networks are formed by starting the circuit with a series element at source end. Each series element creates a transmission zero by being open circuit at relevant frequency. Open circuited behavior can be obtained by;

(i) Series inductor for transmission zero at $\omega = \infty$

(ii) Series capacitors for transmission zero at $\omega = 0$

(iii) Parallel resonators in series arms for finite transmission zeros

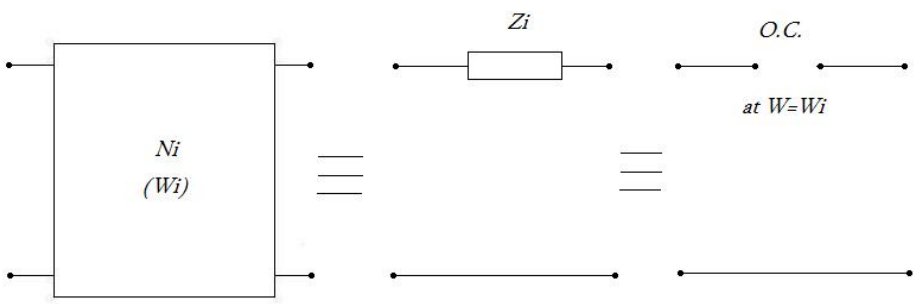


Figure 3.7: Type A Elements Create Transmission Zeros by Being Open Circuit

For example, if there are more than one transmission zero at $\omega = 0$, then they are realized alternately by series capacitors and shunt inductors because, as it was stated in studying lossless one-ports that, if a pole of input impedance (admittance) at $\omega = 0$ is extracted, then a zero appears at that frequency automatically, acting as the pole of the remaining input admittance (impedance).

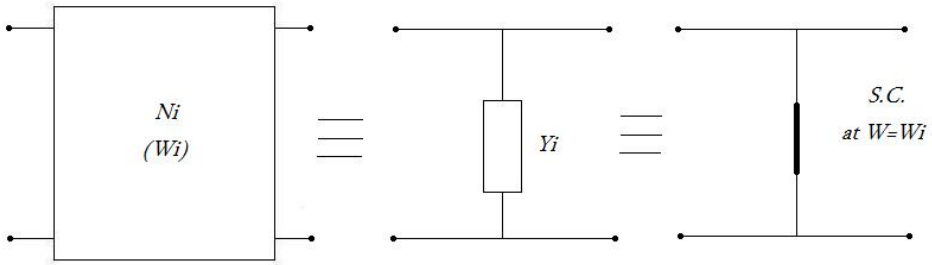


Figure 3.8: Type B Elements Create Transmission Zeros by Being Short Circuit

3.2.5.2 Type B Networks

Type B networks start with a shunt element at source end. Each shunt element creates a transmission zero by being short circuit at relevant frequency. Short circuited behavior can be obtained by;

(i) Shunt inductor for transmission zero at $\omega = 0$

(ii) Shunt capacitors for transmission zero at $\omega = \infty$

(iii) Series resonators in shunt arms for finite transmission zeros

3.3 Filter Parameters Section

In this part, according to filter type's selection, filter parameters inputs either enabled or disabled. Now the filter parameters can be explained in a detail.

3.3.1 Passband Ripple/Passband Edge Loss Inputs

After filter types selection, user must enter the maximum deviation from the zero insertion loss level. For Maximally Flat filter type, the maximum deviation occurs at passband edge. It is usually specified as 3 dB. For Equiripple passband filters desired passband insertion loss ripple must be specified between 0.0005 dB and 20 dB.

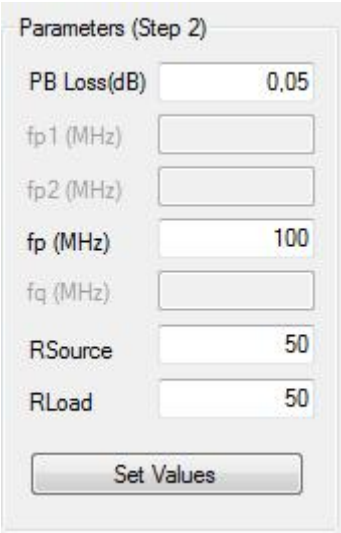


Figure 3.9: Filter Properties

3.3.2 Passband Edge Frequencies Inputs

Like passband ripple value, according to the filter types selection, user must specify the passband edge frequencies. For LP or HP filters, only one the passband edge

3.3.3 Source and Load Resistance Inputs

User can enter the source and load resistors. These values are used in calculating and scaling the element values and in plotting filter responses. In addition to filter responses, in doubly terminated BP filters, these resistor values are used in calculating the transformers needed at load ends to complete the synthesis by checking the output port impedance.

3.4 Set Values Command

After user specified the filter types and filter parameters respectively, Set Values command is used to store the filter parameters. The other synthesis steps are constructed by using these parameters. If the user entered wrong parameters, program checks these values and returns with information message to the user. This continues until all the filter parameters are correct. These controls guide the user for designing the targeted filter. If all parameters are correct then

the section related to filter type and parameters is disabled by the program and transmission zero specification section is enabled.

3.5 Specify Transmission Zeros Section

In synthesis method to design desired filter user must add transmission zeros to shape the desired response. At this stage, the transmission zeros at zero, infinite and non-zero finite frequencies are enabled or disabled according to filter types, as described below.

3.5.1 Adding Transmission Zeros

3.5.1.1 Add TZ at Zero Frequency Command

When user invokes this command, a transmission zero is added at $\omega = 0$ to the transfer function of the filter circuit. These transmission zeros can be added only to highpass and bandpass filters. When the number of transmission zeros is increased at $\omega = 0$, slope (selectivity) of the filter increases in the lower stopband. This type of transmission zeros adds one degree to the filter. Synthesis of program prevent adding transmission zeros at $\omega = 0$ for lowpass filters.

3.5.1.2 Add TZ at Infinite Frequency Command

When user invokes this command, a transmission zero is added at $\omega = \infty$ to the transfer function of the filter circuit. These transmission zeros can be added only to lowpass and bandpass filters. When the number of transmission zeros is increased at $\omega = \infty$, the slope (selectivity) of the filter increases in upper stopband. This type of transmission zeros add one degree to the filter. Synthesis of program prevent adding transmission zeros at $\omega = \infty$ for highpass filters.

3.5.1.3 Add TZ at Finite Frequency Command

When user invokes this command, a transmission zero is added at nonzero finite frequency to the transfer function of the filter circuit. These transmission zeros can be added to low-

pass, highpass and bandpass filters. When the number of transmission zeros are increased in the lower/upper stopband frequency, slope (selectivity) of the filter increases in lower/upper stopband. This type of transmission zeros add two degrees to the filter. Synthesis of program prevent adding transmission zeros inside the passband.

3.5.1.4 Add Unit Element Command

Unit Element (Transmission Line piece) command is enabled only for distributed element filters and can be added to lowpass, highpass and bandpass filters. Unit Element is extracted as a transmission line segment, which is quarter wavelength long at the specified frequency fq. This element adds also a single degree to the filter.

3.5.1.5 Tune Finite Transmission Zero Frequency Command

Using this command, it is possible to adjust the frequencies of finite transmission zeros while inspecting the response. To activate this command, first finite transmission zero is selected and *Tune FTZ* command is clicked. Then, the finite transmission zero to be tuned and increment for tuning frequency are asked. After these information are typed in, the *Plot Window* is shown. Now the response is ready for adjustment by increasing or decreasing the frequency of finite transmission zero using *Increase-Decrease* button. The tuned FTZ Frequency and increment frequency value are displayed at the right top of the Plot Window. Also new finite transmission zero frequency is set by *Set New FTZ* button after entering the value to *New FTZ Value textbox*. When tuning of the FTZ is finished, the *Save Last Value* button is pressed to save the last frequency.

3.5.2 Deleting Transmission Zeros

The user can control the number of transmission zeros by deleting transmission zeros as follows.

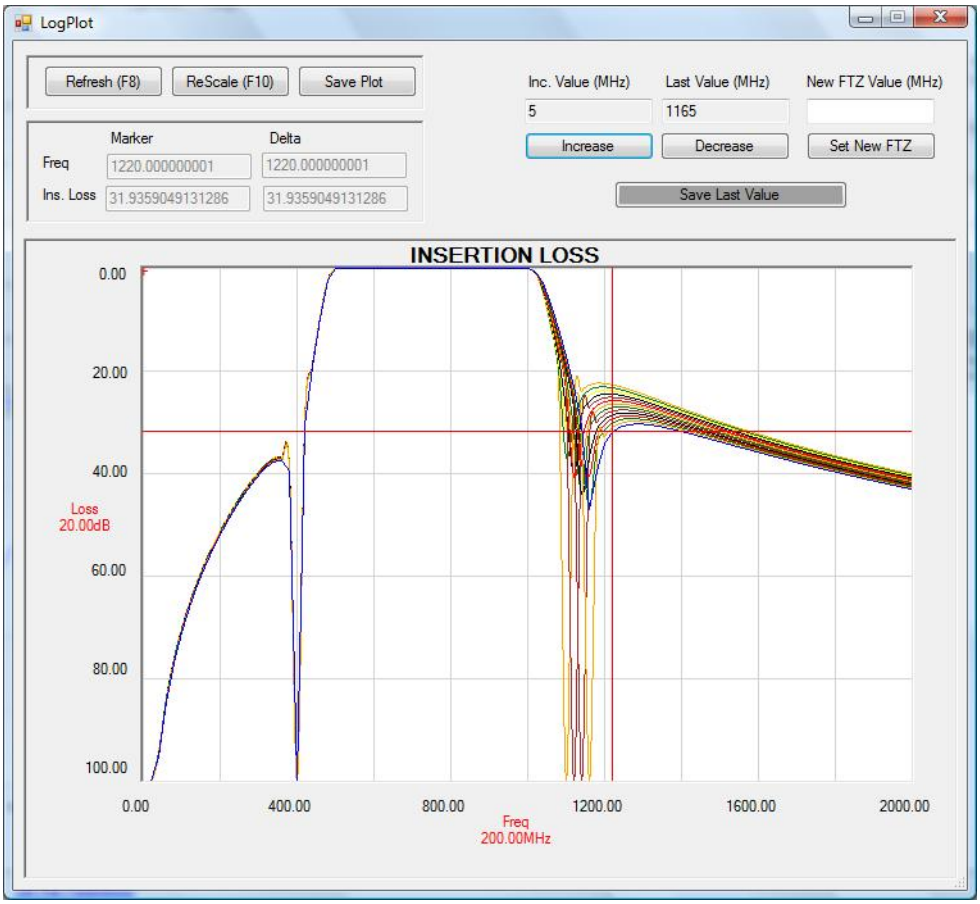


Figure 3.10: Tune Finite Transmission Zero Frequency

3.5.2.1 Delete TZ at Zero Frequency Command

For highpass or bandpass filters, added transmission zeros at $f = 0$ can be deleted one by one by clicking f = 0 button at *Delete TZs* section. If there is no transmission zeros at f = 0, delete command of the transmission zeros is disabled. When this command invoked, degree of filter decreases one.

3.5.2.2 Delete TZ at Infinite Frequency Command

For lowpass or bandpass filters, the added transmission zeros at $f = \infty$ can be deleted one by one by clicking f = Inf button at *Delete TZs* section. If there is no transmission zeros at $f = \infty$, delete command of the transmission zeros is disabled. When this command invoked one transmission zero is deleted and the degree of filter decreases by one.

3.5.2.3 Delete TZ at Finite Frequency Command

User can delete added transmission zeros at nonzero finite frequency by selecting frequency from the list and clicking the *Sel. FTZ* button at *Delete TZs* section. If there is no transmission zeros at nonzero finite frequency, delete command of the transmission zeros section is disabled. When this command is invoked, one transmission zero is deleted and the degree of the filter decreases two. If user clicks the delete button without selecting frequency, synthesis program returns with information message, which warns the user to select the finite transmission zero frequency.

3.5.2.4 Delete Unit Element Command

For distributed element filters, the added Unit Elements can be deleted one by one by clicking UE button at *Delete TZs* section. If there is no Unit Element, delete command of the Unit Element is disabled. When this command invoked, one Unit Element is deleted and the degree of the filter decreases by one.

3.5.2.5 Delete All TZs Command

By this command all added transmission zeros and Unit Elements can be deleted at once. This button is enabled only if there are any transmission zeros or Unit Elements.

3.5.3 Showing Transmission Zeros

Whenever user add or delete transmission zeros, the number of transmission zeros are listed in # section of the TZs section. Also transmission zeros at nonzero finite frequencies are listed by frequency.

3.6 Start Extraction Command

Start Extraction command is used to prepare filter properties and parameters section of the program for element extraction processes. When necessary conditions are met this command is enabled and extraction process is can be started. For example, there must be no transmission zeros within passband or there must be transmission zeros at $\omega = \infty$ and no transmission zeros at $\omega = 0$ for lowpass filters. By invoking this command, filter properties and parameters are evaluated and synthesis formulation starts. Main goal of this command is forming the A and B polynomials in equation 2.94.

Figure 3.11 shows the initialization of extraction. All of the processes given in the figure are implemented according to described formulas in the previous chapter. The following stages are carried out.

3.6.1 Normalize Frequencies

In this section, the passband edge frequencies are mapped from S-domain to Z-domain.

3.6.2 Normalize Transmission Zeros

In this part, the transmission zero frequencies are mapped to Z-domain.

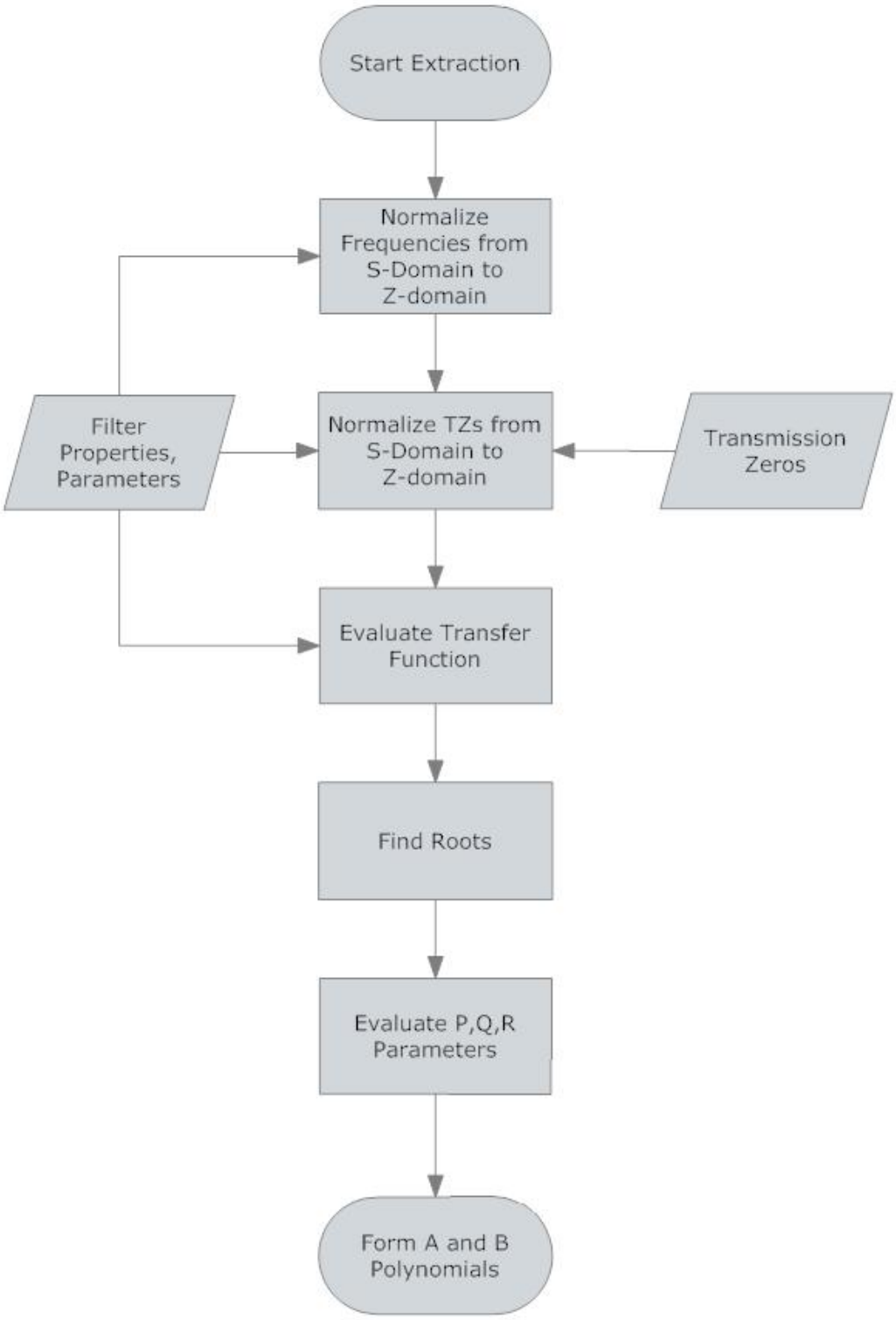


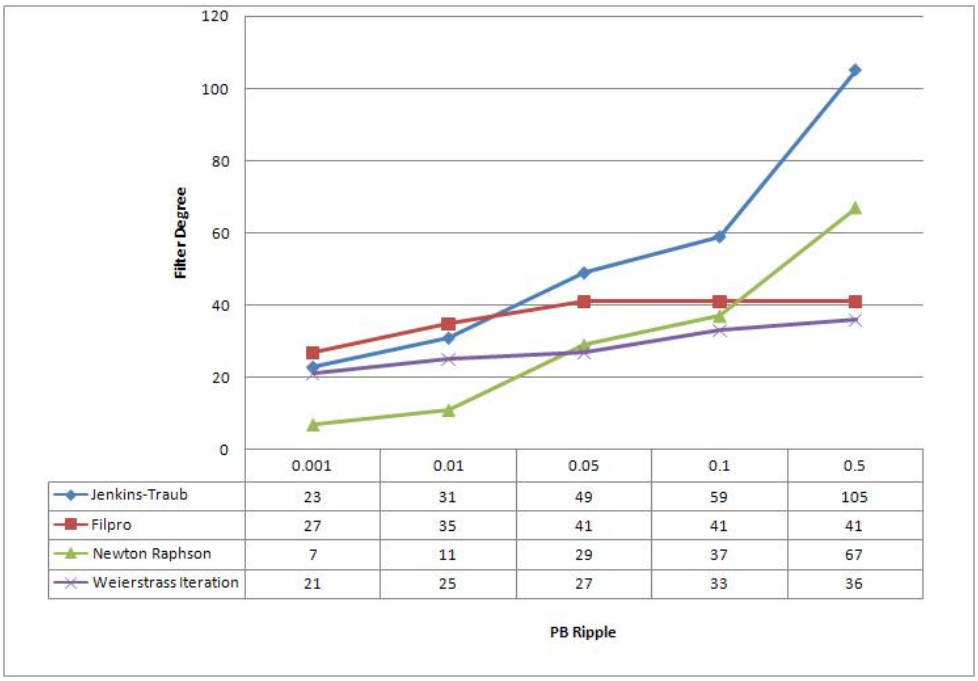
Figure 3.11: Start Extraction Process

3.6.3 Evaluate Transfer Function

After transforming the passband edge frequencies and the transmission zeros to Z-domain, the transfer function is formed in accordance with the specified filter properties. The transfer functions are also used to plot filter response while specifying transmission zeros. This process explained in a detail in the followed sections.

3.6.4 Find Roots

The most cumbersome process in filter synthesis is finding the roots of polynomials forming the transfer function. When filter degree is increased, the roots of polynomials become extremely close to each other. Especially if the element extraction is carried out in s-domain, isolating the roots from each other and calculating them becomes difficult. A small error in the roots leads to great errors in the calculated element values. This is the main factor limiting the degree of filters. This problem is severe if calculations are performed in s-domain. Degree of filters is limited to below 10. The development of Transformed Frequency Domain technique eased this problem considerably. Transformation from s-domain to z-domain separates the problematic roots from each other thus increasing the maximum degree a lot. In this software both formation of the transfer function and element extraction processes are carried out in Transformed Frequency domain. The maximum degree is now about 30-to-100, depending on the specifications of the filters. In synthesis program, different root-finding algorithms are implemented to increase accuracy. First Newton-Raphson algorithm is used which is implemented 3rd party library which is named as *Extreme Optimizations Numerical Component* [14]. Then Weierstrass Iteration is implemented [15]. Finally, Jenkins-Traub Algorithm is tested [11], [12], [13]. All algorithms are tested for the same filter specifications. Comparisons are made for lowpass, lumped element, chebyshev, doubly terminated filters with only odd number of transmission zeros at $\omega = \infty$. The best results are obtained when Jenkins-Traub algorithm is used. Hence, it is used in the new software. The table shown in Figure 3.12 shows some comparative results.



	0.001	0.01	0.05	0.1	0.5
Jenkins-Traub	23	31	49	59	105
Filpro	27	35	41	41	41
Newton Raphson	7	11	29	37	67
Weierstrass Iteration	21	25	27	33	36

Figure 3.12: Maximum Degrees of Filter Calculated By Algorithms for Different Passband Ripple

3.6.5 Evaluate P, Q, R Values

After finding roots of polynomials P, Q and R values must be evaluated in equation 2.93. These process are required to form A and B polynomials in equation 2.94.

3.6.6 Formation A and B Polynomials

Last step of initializing of synthesis is forming A and B polynomials. These polynomials are used in extraction process. After every element extraction, these polynomials are changed and used in the next element extraction.

3.7 Extraction of Transmission Zeros

After *Start Extraction* command is invoked, all initialization process is done and transmission zeros specification section is disabled. In order to extract the transmission zeros, extraction section is enabled. When these commands are invoked, required calculations are done and

circuit elements type and values are found. These elements are added to form the cascaded element filter. Also the complete circuit is drawn. Commands in this section are described in following sections and flow diagrams shown in 3.13.

3.7.1 Extract a Transmission Zero at $f = 0$

As mentioned before if user chose highpass or bandpass filter, synthesis program allows to add transmission zero at f=0 . If user added this type of transmission zero to design filter in specification section, the extraction command at f=0 is enabled. If there is a finite transmission zero exists in a LPF or in the lower stopband of a BPF then the last TZ at f=0 can't be extracted before the extraction of those finite transmission zeros. This is because such finite transmission zeros are extracted by partial extraction of poles at f=0.

3.7.2 Extract Transmission Zero at $f = \infty$

As mentioned before if user chose lowpass or bandpass filter, synthesis program allows to add transmission zeros at $f = \infty$. If user added this type of transmission zero to design filter in specification section, the extraction command at $f = \infty$ is enabled. Also user can not extract the last one of this type of TZs if there are nonzero finite TZs in lowpass filter or upper stopband nonzero finite frequency TZs in bandpass filter.

3.7.3 Extract Transmission Zero at NonZero Finite Frequency

As mentioned before if user chose lowpass, highpass or bandpass filter, synthesis program allows to add transmission zero at non-zero finite frequency. If user added this type of transmission zero to design filter in specification section, the extraction command at nonzero finite frequency is enabled. In order to extract specified frequency user must choose the TZ frequency from the list of TZ frequencies.

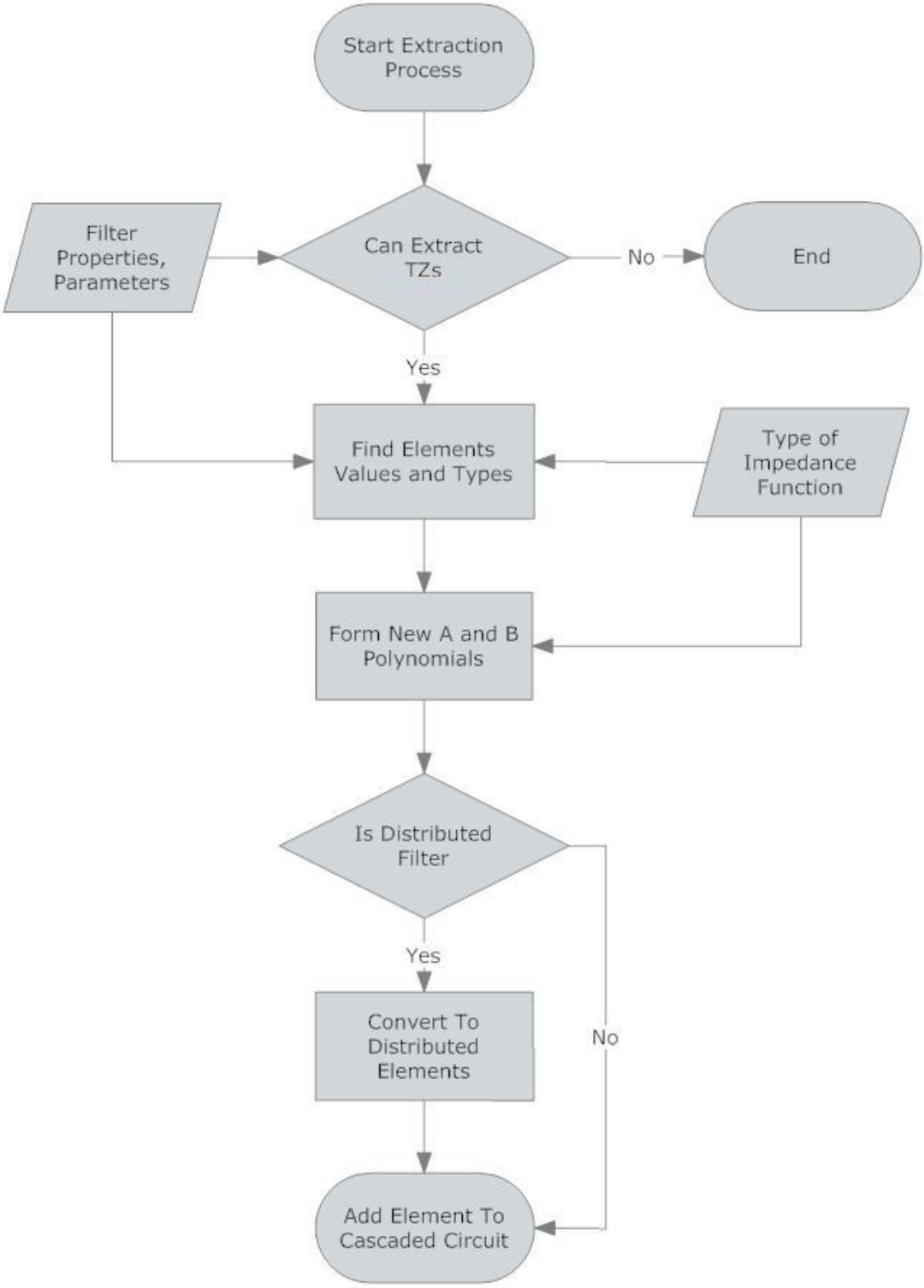


Figure 3.13: Process of Extraction Elements

3.7.4 Extract Transmission Zero Unit Element

As mentioned before if user chose distributed filter, synthesis program allows to add Unit Element. If user added Unit Element to design filter in specification section, the extraction command UE is enabled.

3.7.5 Undo Extraction Command

After extracting any transmission zero, user can undo this extraction by Undo Extraction command. When this command is invoked found element removed from filter circuit and the transmission zero is added to extraction list.

3.7.6 Start Auto Extraction Command

By this command predescribed extraction order is executed and all transmission zeros are converted to circuit elements and they are added to filter circuits. This command is enabled only when Start Extraction command is invoked and the specified transmission zeros and filter types match the

3.7.7 Macro Auto Extraction Command

By this command, the previous extraction order is executed automatically. If any filter type or transmission zeros are changed extraction order is cleaned. By this properties user can change some filter parameters like ripple, finite frequency transmission zero and band edge frequencies, etc.

3.8 Plot Response While Specifying Transmission Zeros

Filter response is plotted while user specifies transmission zeros. Insertion loss and return loss are calculated after user added any transmission zeros. By this property user can see the formation of the response.

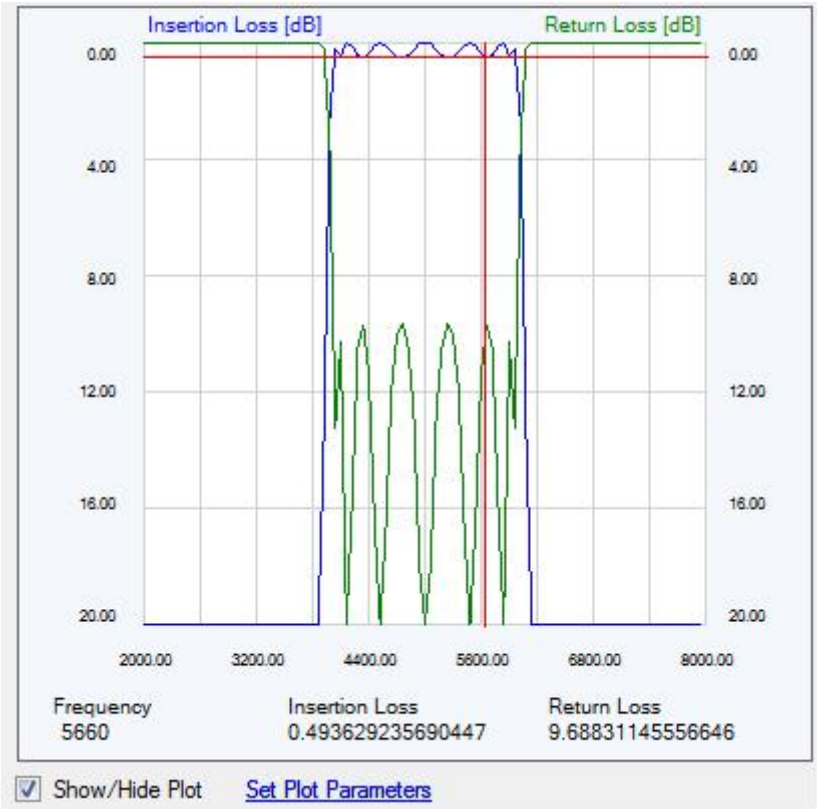


Figure 3.14: Filter Responses While Specifying Transmission Zeros

In figure 3.14 Insertion Loss is plotted by blue line and Return Loss is plotted by green line. Vertical left line shows Insertion Loss magnitude and vertical right side shows Return Loss magnitude. Horizontal line shows frequency values.

3.8.1 Mouse Move Command

When user moves the cursor over the filter response graphic, the magnitude and frequency values are shown at the bottom of the graphic. Also the point which is pointed by cursor is shown by red cross lines. This cross lines move with cursor when user moves the cursor.

3.8.2 Mouse Click Command

By default the red cross line moves over the insertion loss response with cursor. If user clicks once on the graphic, the red cross line moves over the return loss response with cursor. Whenever user clicks on the graphic it toggles the red cross line path.

3.8.3 Set Plot Parameters Command

By this command user can change magnitudes and frequency range, which is shown at horizontal and vertical lines of filter response graphic. When user clicks this command, new setting window is opened. This window is shown in figure 3.15.

In this setting window, following parameters are set.

- Minimum Frequency Value in MHz
- Maximum Frequency Value in MHz
- Minimum Insertion Loss Value in dB
- Maximum Insertion Loss Value in dB
- Minimum Return Loss Value in dB
- Maximum Return Loss Value in dB

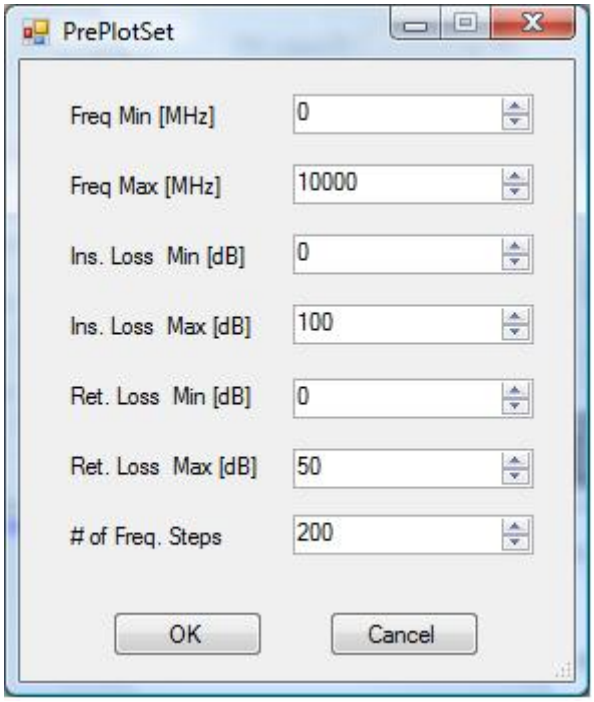


Figure 3.15: Set Plot Parameters

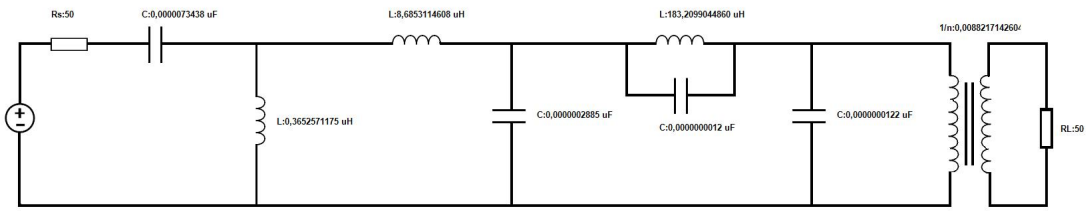
If user changes any of these parameters and then clicks *OK* button, filter responses are redrawn with new parameters and setting window is closed automatically. If user clicks the Cancel button, the setting window is closed without changing filter parameters.

3.8.4 Save Plot Command

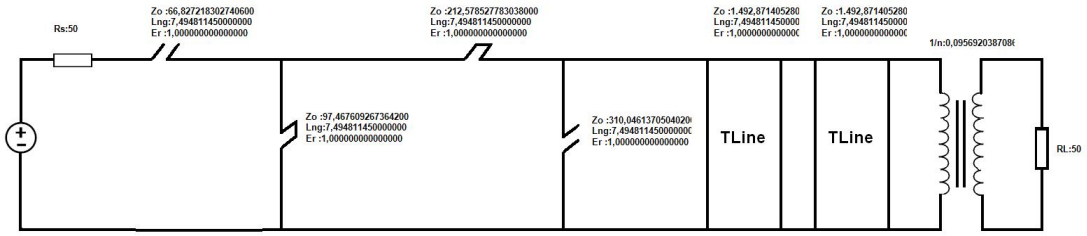
If user wants to save filter responses in a Bitmap format then he goes to *Circuit* menu and clicks the *Save Plot* command. By this command, *Save File Dialog* window is opened and user can save the filter response image to specified location.

3.9 Filter Circuit

After every extraction of the transmission zeros, synthesis program finds new filter elements and display the schematics of the filter circuit. The filter circuits include filter elements and values. In figure 3.9 shows sample filter circuits for lumped and distributed filters.



(a) Lumped Filter Circuit



(b) Distributed Filter Circuit

Figure 3.16: Filter Circuit Samples

3.9.1 Save Plot Command

If user wants to save filter circuit in a Bitmap format then he goes to *Circuit* menu and clicks the *Save Circuit* command. By this command, *Save File Dialog* window is opened and user can save the filter circuit image to specified location.

3.9.2 Change Unit Command

Change Units Window can be used to set the units of inductors and capacitors in micro, nano and pico scale. User clicks Options menu and then clicks Change Units command. After selection of the units user must click OK button and the window is closed automatically then the filter circuit is redrawn with new parameters.

3.10 Synthesis Menu

In synthesis process user is guided by program. In the first step as mentioned before, user set filter properties and parameters, the second step user specifies transmission zeros and finally

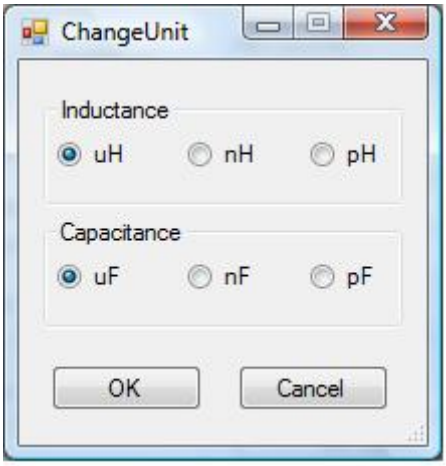


Figure 3.17: Change Unit Window

user extracts transmission zeros respectively. In every step, other steps are disabled and user cannot change any parameters. Following commands are used to activate specified steps.

3.10.1 Reset Properties

By this command user reactivates filter properties and parameters section and the other windows are disabled. This command is activated only when specification of transmission zeros is active. If the extraction of transmission zeros section is enabled and user wants to change any filter properties and parameters, first user must activate specification of transmission zeros section and then he must click *Reset Properties* menu. When user invokes this command the specified transmission zeros are not changed.

3.10.2 Respecify TZ's

By this command user reactivate specification of transmission zeros section and the other windows are disabled. This command is activated only when extraction of transmission zeros section is active. If the extraction of transmission zeros section is enabled and user wants to specify new transmission zeros, user must click *Respecify TZ's* menu. When user invokes this command the extracted transmission zeros are cleared and user can continue to specifying transmission zeros. Also the displayed filter circuit is cleared.

3.10.3 Reextract TZ's

By this command user reactivates extraction of the transmission zeros section and the other windows are disabled. This command is activated only when extraction of transmission zeros section is active. If the extraction of transmission zeros section is enabled and user wants to restart extraction process, user must click Reextract TZ's menu. When user invokes this command extracted transmission zeros are cleared and user can start extracting the transmission zeros again. Also the displayed filter circuit is cleared.

3.11 Plot Filter Responses

In synthesis program, after filter is constructed, six different types of filter responses can be plotted in addition to Smith Chart plot. These are listed as below.

- Insertion Loss
- Return Loss
- Delay S_{21}
- Phase S_{21}
- Delay S_{11}
- Phase S_{11}

These plots can be displayed by using *Plot Menu* in figure 3.18. When user click one of the Plot menu items, *PlotSet Window* is opened to specify frequency and magnitude range for the specified type of plots.

In the *PlotSet Window* there are five parameters can be set these are listed as below.

- Minimum Magnitude of Specified Type of Plots(Insertion,Return etc.)
- Maximum Magnitude of Specified Type of Plots
- Minimum Frequency in MHz

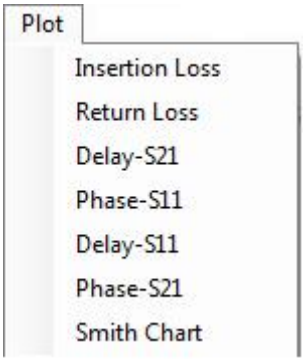


Figure 3.18: Plot Menu Window

- Maximum Frequency in MHz
- # of Frequency Steps

# of Frequency Steps defines the number of sampling of plot. If user set 200, response of filter calculated 200 times in different frequencies between minimum and maximum frequency. If user increases the number of steps, more precise data can be calculated.

After user specifies plot parameters, click OK button then the window is closed automatically and main *Plot Window* is opened with specified parameters. This main window is shown in figure 3.20.

3.11.1 Insertion Loss Command

When user invokes *Insertion Loss Command*, synthesis program calculates and plots insertion loss versus frequency. This response is evaluated using $S21$ which is calculated with ABCD parameters and its magnitude is displayed in dB [16].

3.11.2 Return Loss Command

When user invokes *Return Loss Command*, synthesis program calculates and plots return loss versus frequency. This response is evaluated using $S11$ and its magnitude is displayed in dB.

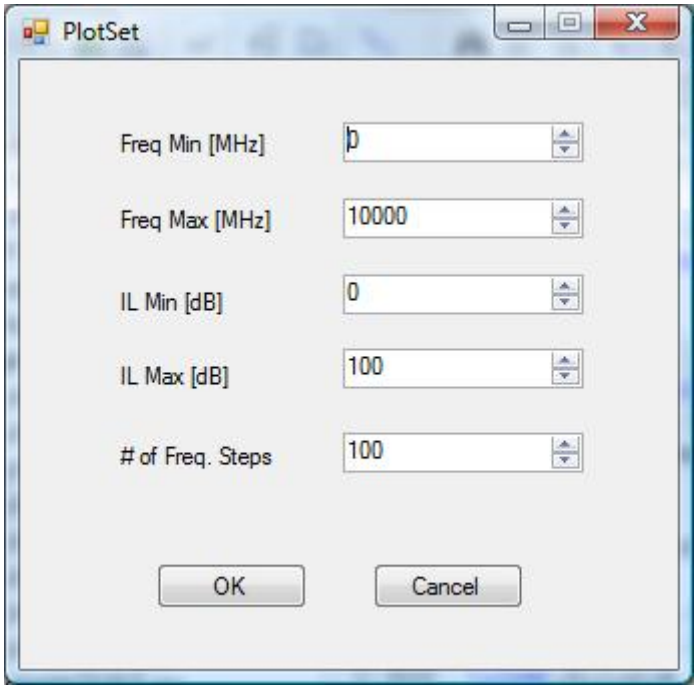


Figure 3.19: Plot Set Window

3.11.3 Delay S_{21} Command

When user invokes *Delay* $S21$ *Command*, synthesis program calculates and plots Delay of $S21$ versus frequency. This response is evaluated using S21 and its magnitude is displayed in nanoseconds

3.11.4 Phase S_{21} Command

When user invokes *Phase* S_{21} *Command*, synthesis program calculates and plots Phase S_{21} versus frequency. This response is evaluated using $S21$ and its magnitude represented in degrees.

3.11.5 Delay S_{11} Command

When user invokes *Delay* S_{11} *Command*, synthesis program calculates and plots Delay S_{11} versus frequency. This response is evaluated using $S11$ and its magnitude represented in nanoseconds.

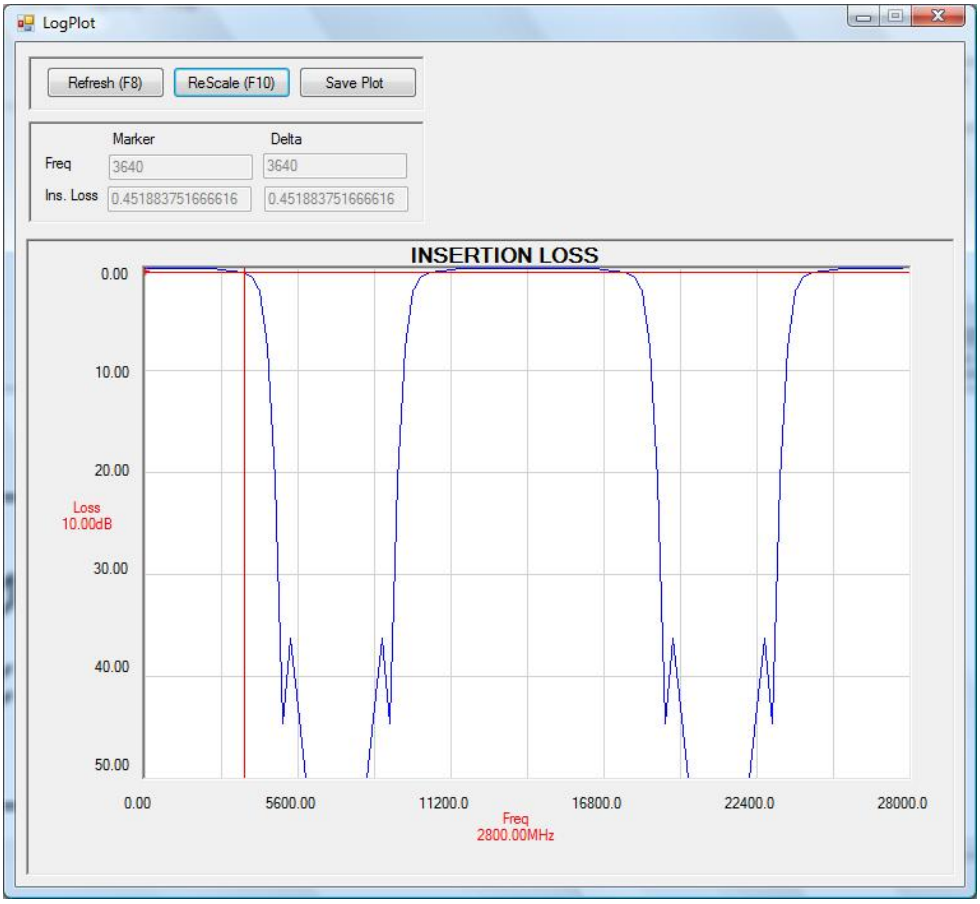


Figure 3.20: Main Plot Window

3.11.6 Phase S_{11} Command

When user invokes *Phase* S_{11} *Command*, synthesis program calculates and plots Phase S_{11} versus frequency. This response is evaluated using $S11$ and its magnitude represented in degrees.

3.11.7 Main Plot Window Commands

In the main plot window, six different plots can be calculated and drawn. In addition to these plots user can use some functional command to analyze the displayed plots. The plot window has three different parts. In the first part, frequency and magnitude values are shown at the specified frequency which is pointed by mouse. In the second part, there are some buttons

to manipulate plot parameters and save drawn filter response. In the last part, filter response is drawn. Magnitude and frequency values are shown on the vertical and horizontal lines respectively.

3.11.7.1 Mouse Move Command

When user moves the cursor over the filter response graphic, response's magnitude and frequency values are shown at the top of the graphic. Also point which is selected by cursor is shown by red cross lines. This cross lines move with cursor when user moves the cursor.

3.11.7.2 Mouse Click Command

When user clicks the graphic, specified point saved and whenever mouse cursor is moved, differences between the saved and current location values are calculated and displayed at the top of the page. If user clicks again saved point is reset and new location is saved to calculate new differences. This property is useful when user wants to see difference between two locations. The save point also is shown by little red crossline.

3.11.7.3 Mouse Drag and Drop Command

At the beginning of the plot responses section, user can specify the magnitude and frequency ranges. But after this processes, user can change the ranges with mouse. In order to invoke this command user must click the left mouse button and drags the any location of image and then release the mouse button. After that if user moves the cursor to left side, right side of the plot is shown. If the cursor is moved to right, left side of the plot is shown. If it is moved up then bottom side of the plot is shown and if it is moved down then the upper side of the plot is shown. Also minimum and maximum values of ranges are changed according to the mouse action.

3.11.7.4 Refresh Command

By this command, saved delta location and red cross lines are cleared. Also plot of response is redrawn.

3.11.7.5 Rescale Command

By this command *Rescale Window* is opened and plot types and parameters can be changed. In figure 3.21 plot types listed as an option and filter parameters related to selected option is listed below the options. Whenever plot type is changed, filter parameters changes according to this selection and user can fill the parameters. After that if user clicks *OK* button *Rescale Window* is closed automatically and new plot is drawn in the main plot window.

3.11.8 Save Plot Command

If user wants to save filter responses in a Bitmap format then he clicks the *Save Plot* command. By this command, *Save File Dialog* window is opened and user can save the filter response image to specified location.

3.11.9 Exit Command

By this command, *Plot Window* is closed and user returns the *Main Window*.

3.11.10 Smith Chart

In addition to the other plots (Insertion, Return Loss, etc.), Smith Chart is also drawn. In order to draw smith Chart, user clicks the Smith Chart from the Plot menu. After this, new dialog is opened and user specifies the minimum, maximum frequencies and number of frequency steps. After that new Smith Chart window is opened as shown in figure 3.22.

In this window has three parts. The first part is marker values, which shows the parameters of the selected location. The second part is includes buttons to save plot or change plot

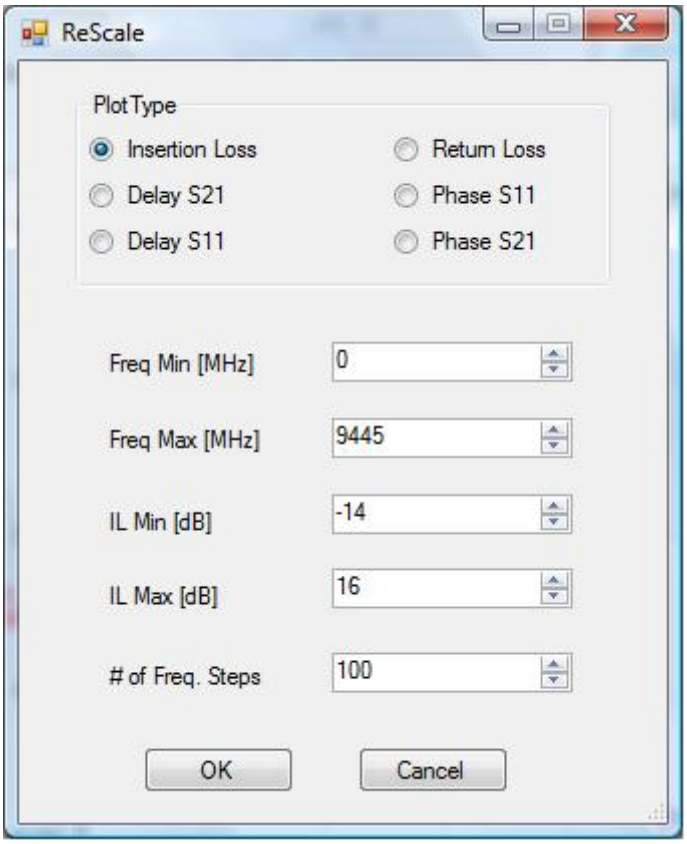


Figure 3.21: Rescale Window

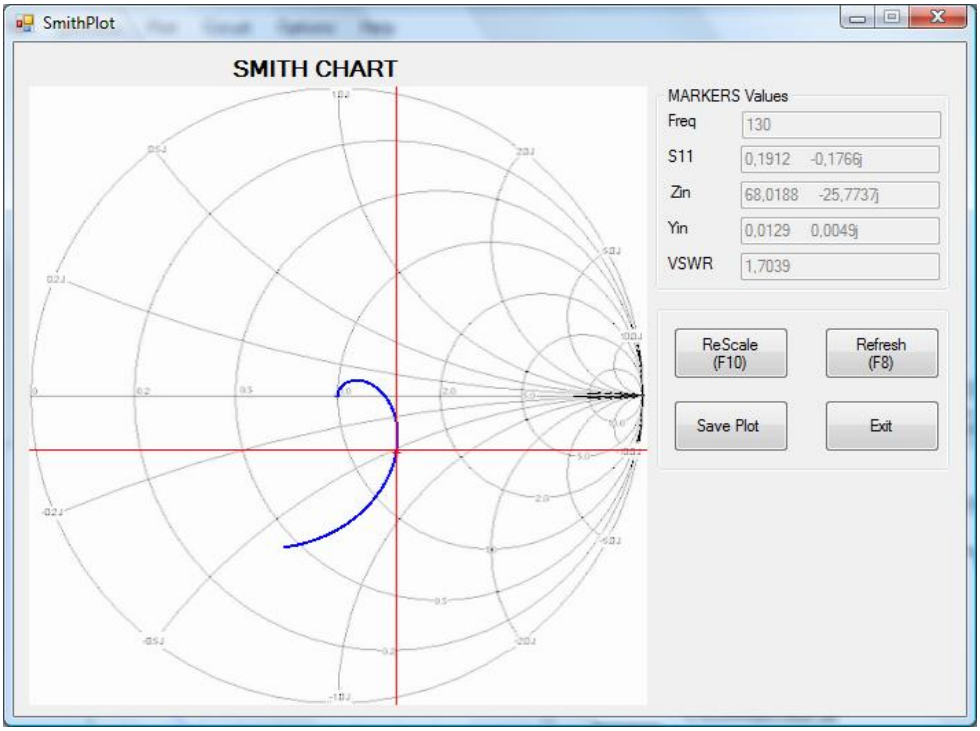


Figure 3.22: Smith Chart Window

parameters. In the last part Smith chart is drawn.

3.11.10.1 Mouse Move Command

When user moves the cursor over the Smith Chart graphic, specified values by cursor are shown at the right top of the graphic. Also point which is selected by cursor is shown by red cross lines. This cross lines move with cursor when user moves the cursor. In the markers values can be listed as below.

- Frequency
- S_{11}
- Z_{in}
- Y_{in}
- $VSWR$

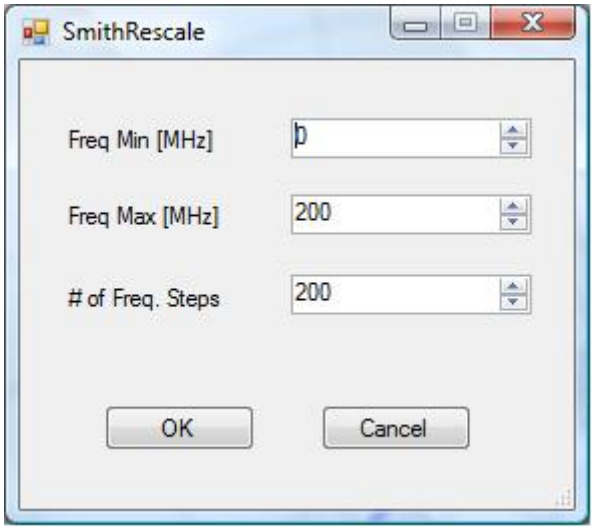


Figure 3.23: Smith Chart Rescale Window

3.11.10.2 Mouse Click Command

By this command, mouse move action stops and current values stay in the markers value section. If user clicks the Smith Chart again, mouse move process starts again.

3.11.10.3 Refresh Command

By this command, red cross lines are cleared. Also plot of response is redrawn.

3.11.10.4 Rescale Command

By this command, *Smith Rescale Window* is opened in figure 3.23 and Smith Chart parameters can be changed. After that if user clicks *OK* button *Smith Rescale Window* is closed automatically and new plot is drawn in the Smith Chart window.

3.11.11 Save Plot Command

If user wants to save Smith Chart in a Bitmap format then he clicks the *Save Plot* command. By this command, *Save File Dialog* window is opened and user can save the Smith Chart

image to specified location.

3.11.12 Exit Command

By this command, *Smith Chart Window* is closed and user returns the *Main Window*.

3.12 File Menu Commands

In all programs user can use some file operations like saving, loading etc. In synthesis program designed filter properties, parameters and specified transmission zeros can be saved and loaded. Following sections describe these commands in a detail.

3.12.1 New Command

By this command, all changed or specified parameters are cleared and new filter designing starts and filter properties and parameters section is enabled and other sections are disabled. Default filter properties and parameter can be listed as below.

- Lumped Elements
- LowPass
- Maximally Flat
- Singly Terminated
- Type-A
- $Ripple = 0.05dB$ $f_p = 100MHz$
- $R_{source} = 50$ $R_{load} = 50$

3.12.2 Save Command

By this command, all filter properties, parameters, specified transmission zeros and design steps are saved to XML file. By using XML format, filter parameters can be obtain by another programs or new versions of synthesis programs. In many other programs binary objects

are converted into stream and written in a file. When the objects structure is changed, saved files parameters cannot be retrieved successfully. But in synthesis program XML file and included data can be converted easily even objects structure or program are changed. When user invokes this command, all parameters are saved predefined location and named as *Default.xml*. If user invokes *Save As* command and specified new file, *Save* command now save parameters to new file. The sample XML file is shown in figure 3.24

3.12.3 Save As Command

By this command, all filter properties, parameters, specified transmission zeros and design steps are saved to XML file like *Save Command* but in every time when user invoke this command, *Save File Dialog* is opened and XML file saved to specified location as any desired name. After this process, if user invokes *Save Command* new changes saved specified file.

3.12.4 Load Command

By this command user can load saved file to program and continue to design filter. When user invokes this command, *Load File Dialog* is opened and user choices the saved XML file. After that, synthesis program parses the file and load all parameters and states to editor. This property user can save and load anytime his designed filter.

3.12.5 Exit Command

When user invokes this command, synthesis program is closed.

```xml
<?xml version="1.0" encoding="utf-8" ?>
- <Synthesis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <zeroSpec>0</zeroSpec>
  <infSpec>3</infSpec>
  <ueSpec>0</ueSpec>
- <SpecifiedTZ>
    <SpecifiedTZ>200</SpecifiedTZ>
    <SpecifiedTZ>200</SpecifiedTZ>
  </SpecifiedTZ>
- <ExtractedTZ>
  - <ExtractedTZ>
      <Tztype>INF</Tztype>
      <Specified>3</Specified>
      <Extracted>2</Extracted>
      <Ftzvalue>0</Ftzvalue>
    </ExtractedTZ>
  - <ExtractedTZ>
      <Tztype>INF</Tztype>
      <Specified>3</Specified>
      <Extracted>2</Extracted>
      <Ftzvalue>0</Ftzvalue>
    </ExtractedTZ>
  </ExtractedTZ>
  <passband>0</passband>
  <fp>100</fp>
  <fmin>100</fmin>
  <fmax>120</fmax>
  <ripple>0.05</ripple>
  <rs>50</rs>
  <rl>50</rl>
  <reflectionZero>1E-09</reflectionZero>
  <structure>0</structure>
  <response>0</response>
  <termination>0</termination>
  <status>2</status>
</Synthesis>
```


Figure 3.24: Sample XML File Includes Filter Parameters

CHAPTER 4

DESIGN EXAMPLES

In previous chapters, main filter concepts are described, synthesis formulation and synthesis program implementation. As mentioned before, main goal is designing filters by specifying transmission zeros and filter properties. In this chapter some filters will be shown by using synthesis program and analyze the results.

4.1 Result

By synthesis program, many different filter types can be designed. Therefore, in this time only three different types of filters will be designed.

4.1.1 Lumped Element LowPass Filter Example

At the first example, lumped element lowpass filter is designed. The filter properties specified as below.

- Lumped Structure
- Lowpass Filter
- Equiripple Filter
- Doubly Termination
- Type-A Circuit
- $Ripple = 0.1$, $f_p = 200$, $R_{source} = 50$,$R_{load} = 50$

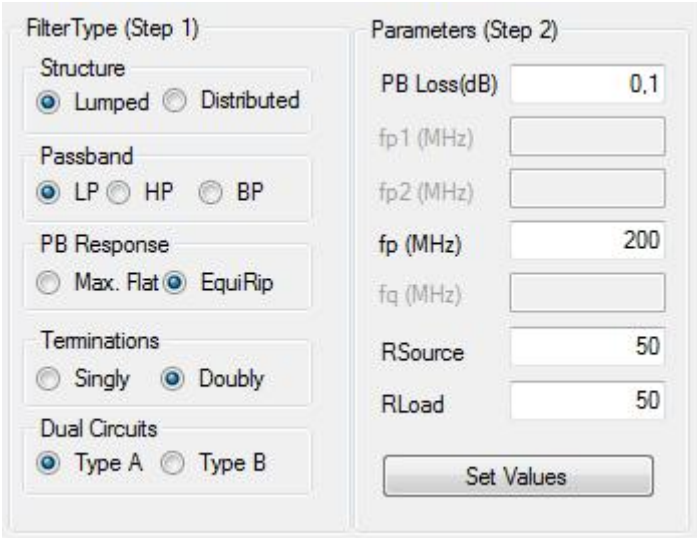


Figure 4.1: Setting Parameter for Lumped Element Lowpass Filter

- 4 TZs at $f = \infty$ and TZs at $f = 300MHz$, $f = 400MHz$, $f = 600MHz$

First filter properties and parameters must be specified like in figure 4.1

After this filter specification, *Set Values* button is invoked *Transmission Zeros Specification Section* is opened and the TZs are added as listed in figure 4.2

While transmission zeros being added to filter, filter responses are plotted at the plot section. The final responses are plotted in figure 4.3.

After the specification process, the transmission zeros are extracted in the following order

- TZ at ∞
- TZ at ∞
- TZ at ∞
- TZ at $f = 300MHz$
- TZ at $f = 400MHz$
- TZ at $f = 600MHz$

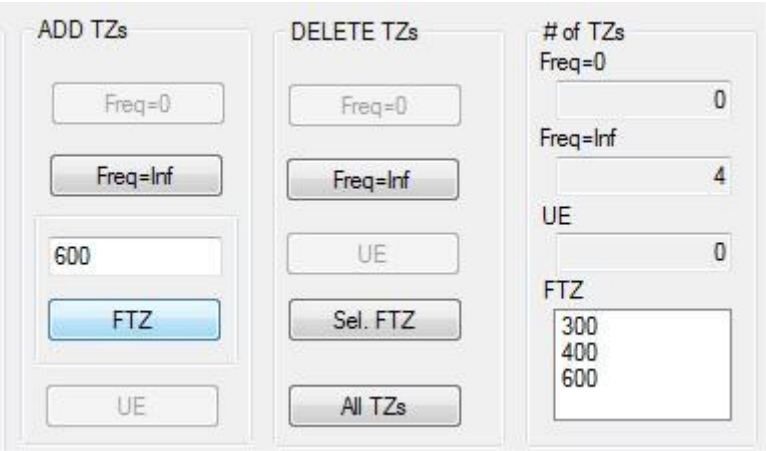


Figure 4.2: Setting Transmission Zeros for Lumped Element Lowpass Filter

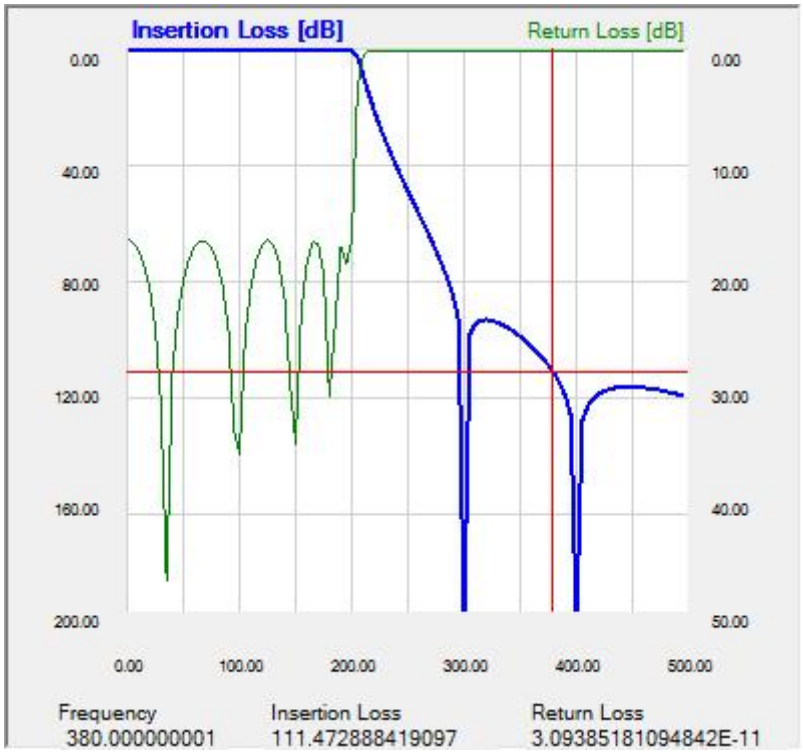


Figure 4.3: Plot for Lumped Element Lowpass Filter

- TZ at ∞

After every extraction of transmission zeros, calculated element types and values are added to filter circuit. The filter circuit is shown in figure 4.4.

4.1.2 Distributed Element LowPass Filter Example

In this example, distributed element lowpass filter is designed. The filter properties specified as below.

- Distributed Structure
- Lowpass Filter
- Maximally Filter
- Singly Termination
- Type-B Circuit
- $Ripple = 0.5$, $f_p = 3000$, $f_q = 6000$, $R_{source} = 50$,$R_{load} = 50$
- 3 TZs at $f = \infty$, TZs at $f = 4000MHz$, $f = 5000MHz$ and 2 Unit Element

First filter properties and parameters must be specified like in figure 4.5

After this filter specification, *Set Values* button is invoked *Transmission Zeros Specification Section* is opened and the TZs are added as listed in figure 4.6

While transmission zeros being added to filter, filter responses are plotted at the plot section. The final responses are plotted in figure 4.7.

After the specification process, the transmission zeros are extracted in the following order

- TZ at ∞
- TZ at ∞

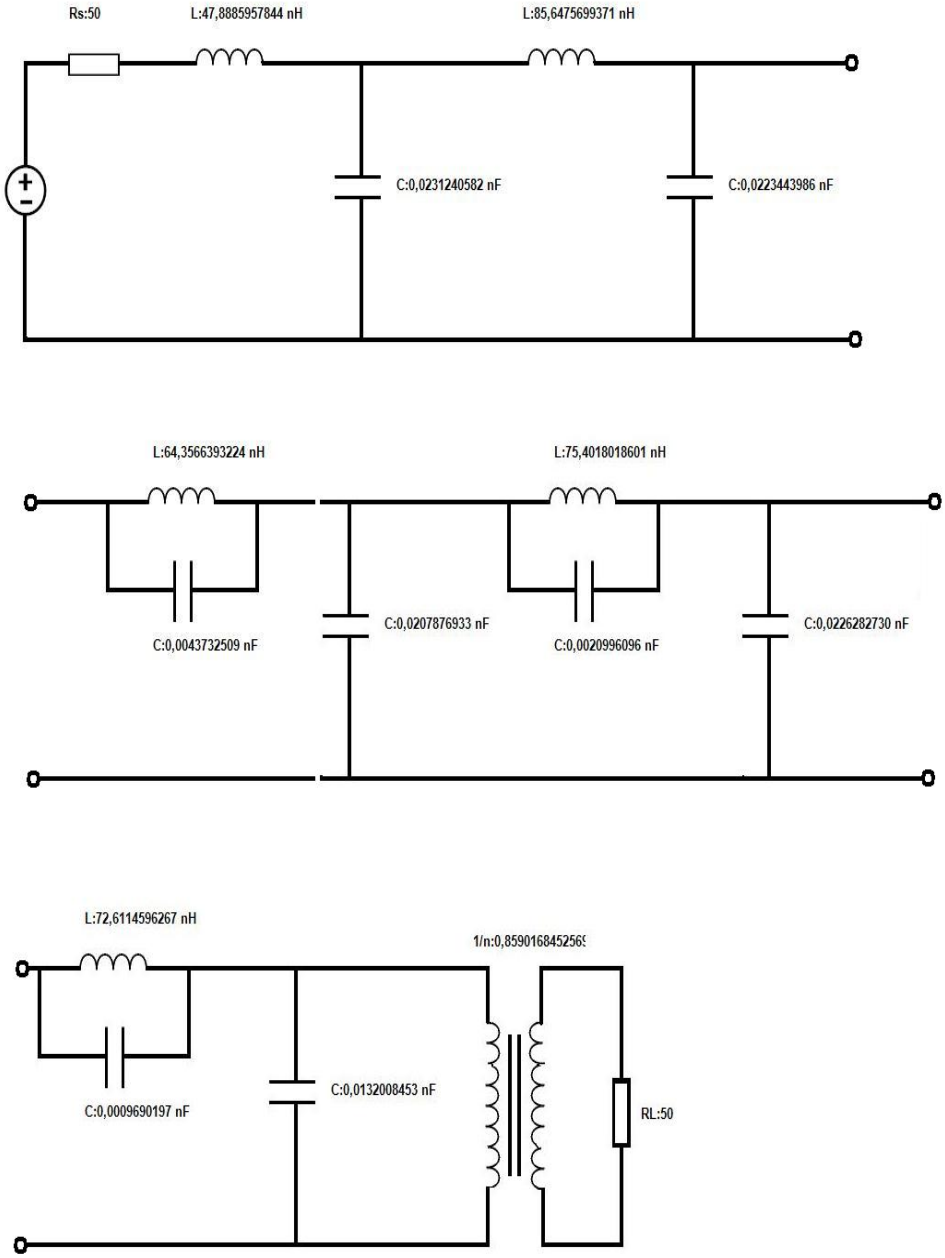


Figure 4.4: Circuit for Lumped Element Lowpass Filter

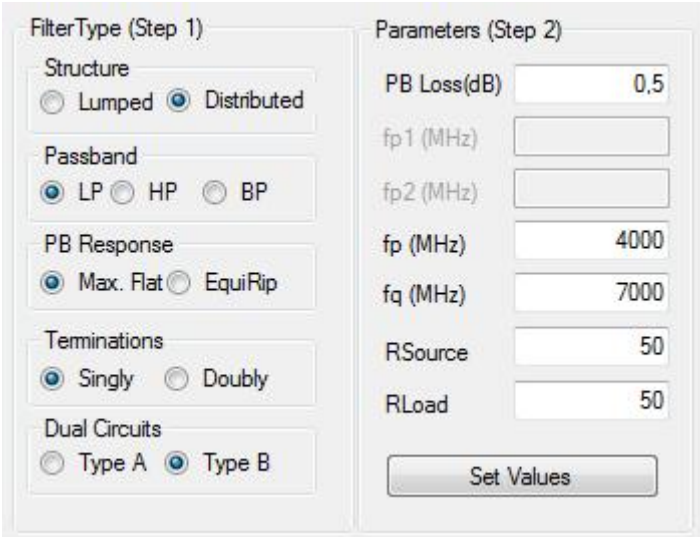


Figure 4.5: Setting Parameter for Distributed Element Lowpass Filter

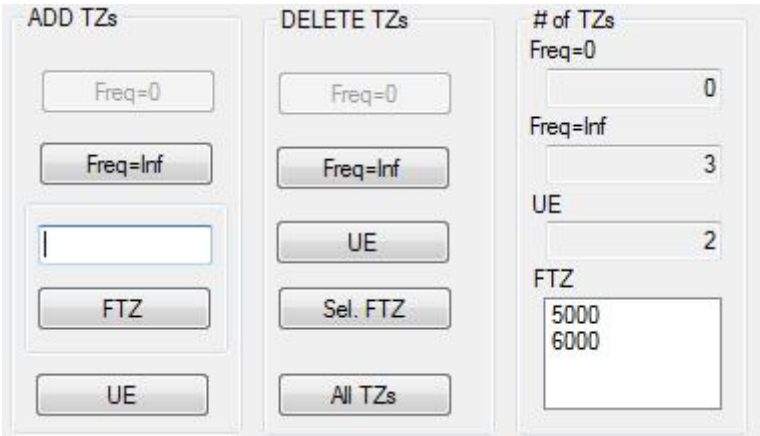


Figure 4.6: Setting Transmission Zeros for Distributed Element Lowpass Filter

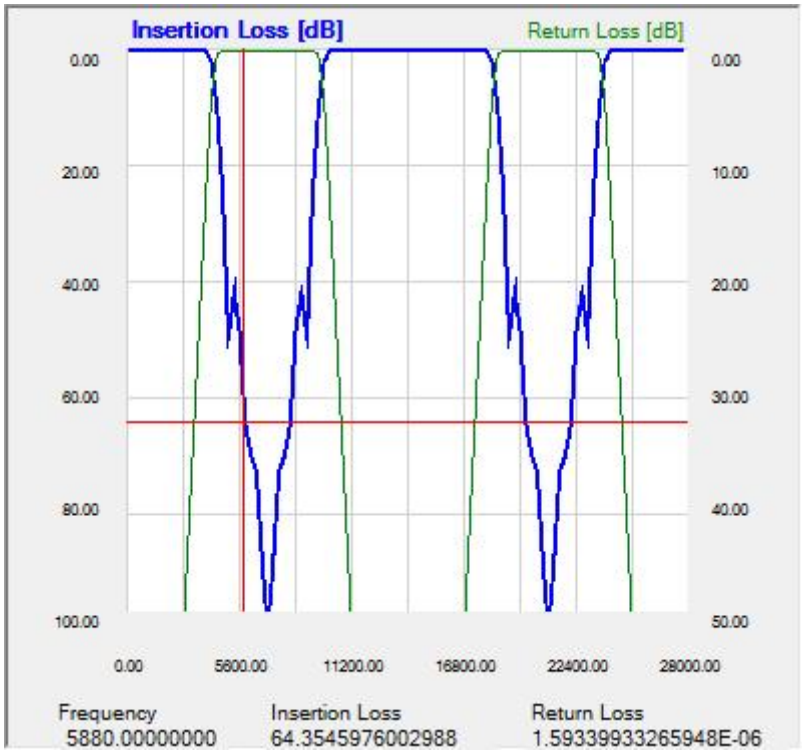


Figure 4.7: Plot for Distributed Element Lowpass Filter

- TZ at $f = 4000MHz$
- Unit Element
- TZ at $f = 5000MHz$
- Unit Element
- TZ at ∞

After every extraction of transmission zeros, calculated element types and values are added to filter circuit. The filter circuit is shown in figure 4.8.

4.1.3 Lumped Element BandPass Filter Example

In this example, lumped element bandpass filter is designed. The filter properties specified as below.

- Lumped Structure
- Bandpass Filter
- Equiripple Filter
- Singly Termination
- Type-B Circuit
- $Ripple = 0.2$, $f_{p1} = 500$, $f_{p2} = 1000$, $R_{source} = 50$, $R_{load} = 50$
- 3 TZ at $f = 0$, 3 TZs at $f = \infty$ and TZs at $f = 200MHz$, $f = 300MHz$, $f = 1200MHz$, $f = 1400MHz$

First filter properties and parameters are specified like in figure 4.9

After this filter specification, *Set Values* button is invoked, *Transmission Zeros Specification Section* is opened and TZs are added as listed in figure 4.10

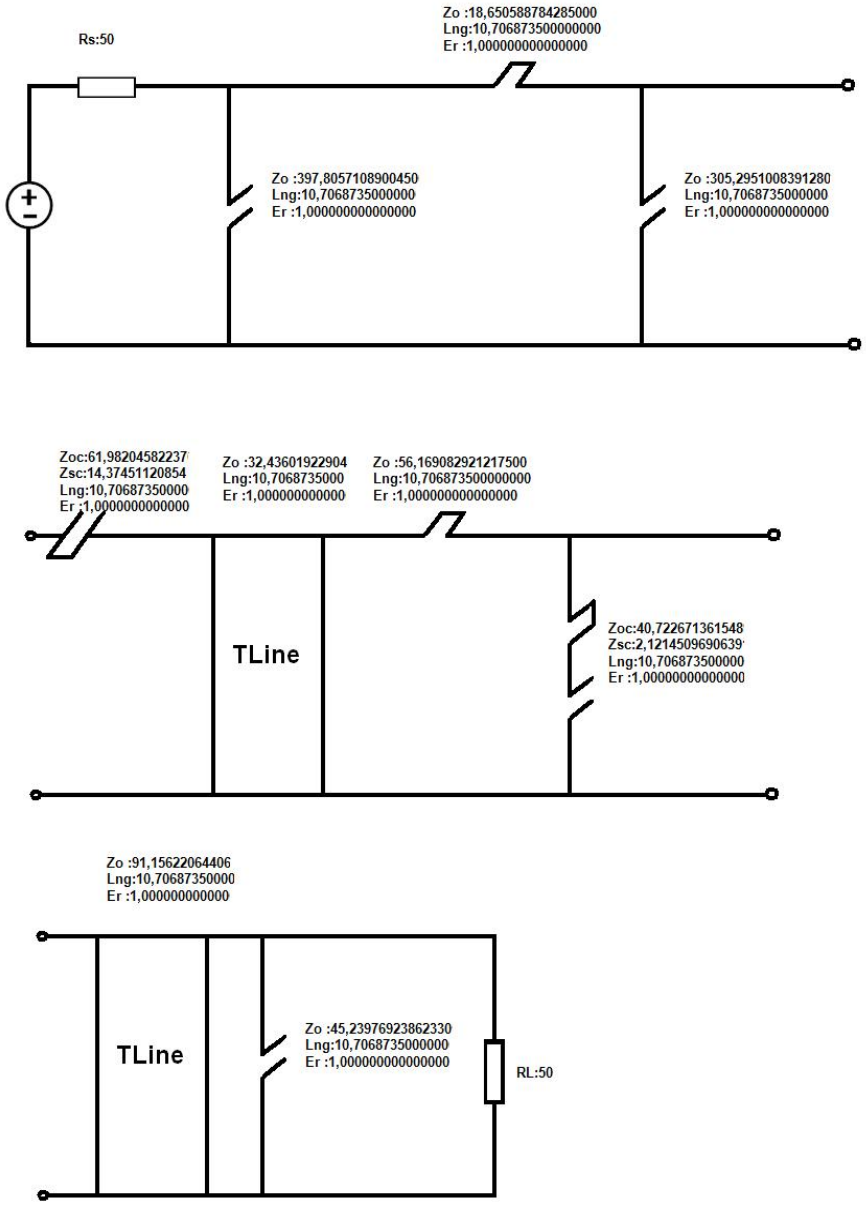


Figure 4.8: Circuit for Distributed Element Lowpass Filter

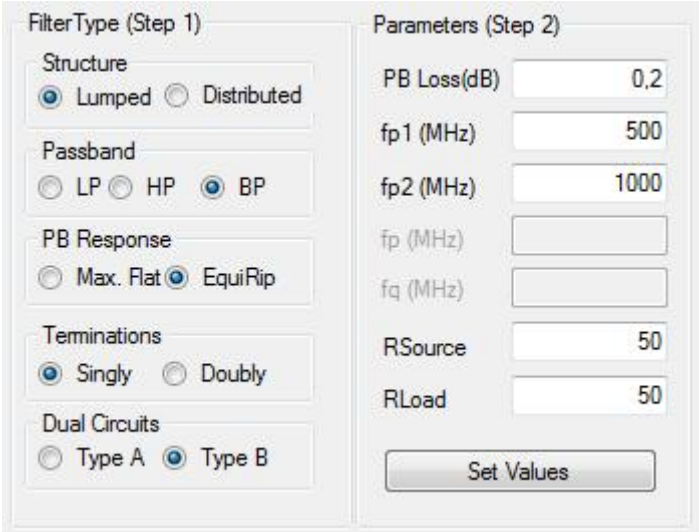


Figure 4.9: Setting Parameter for Lumped Element Bandpass Filter

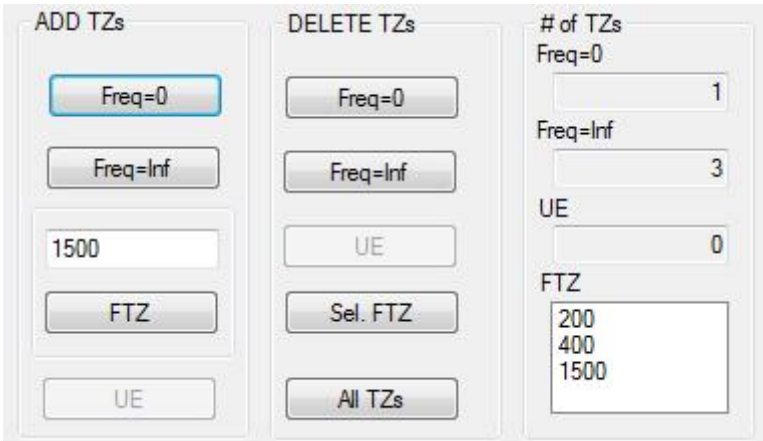


Figure 4.10: Setting Transmission Zeros for Lumped Element Bandpass Filter

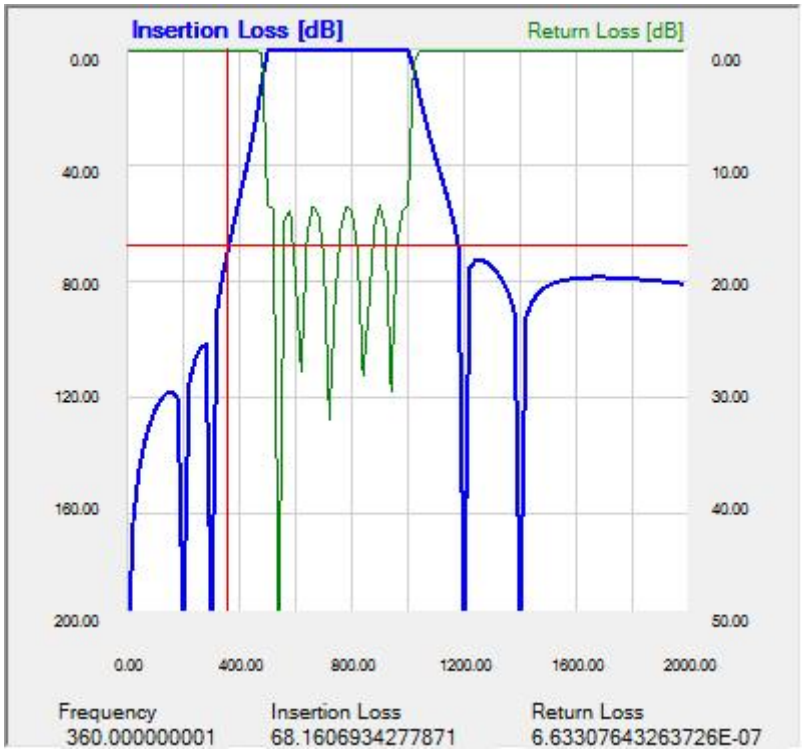


Figure 4.11: Plot for Lumped Element Bandpass Filter

While transmission zeros being added to filter, filter responses are plotted at the plot section. The final responses are plotted in figure 4.11.

The filter circuit is shown in figure 4.12.

After the specification process, the transmission zeros are extracted in the following order

- TZ at $f = 0$
- TZ at ∞
- TZ at $f = 200MHz$
- TZ at $f = 1400MHz$
- TZ at $f = 300MHz$
- TZ at $f = 1200MHz$

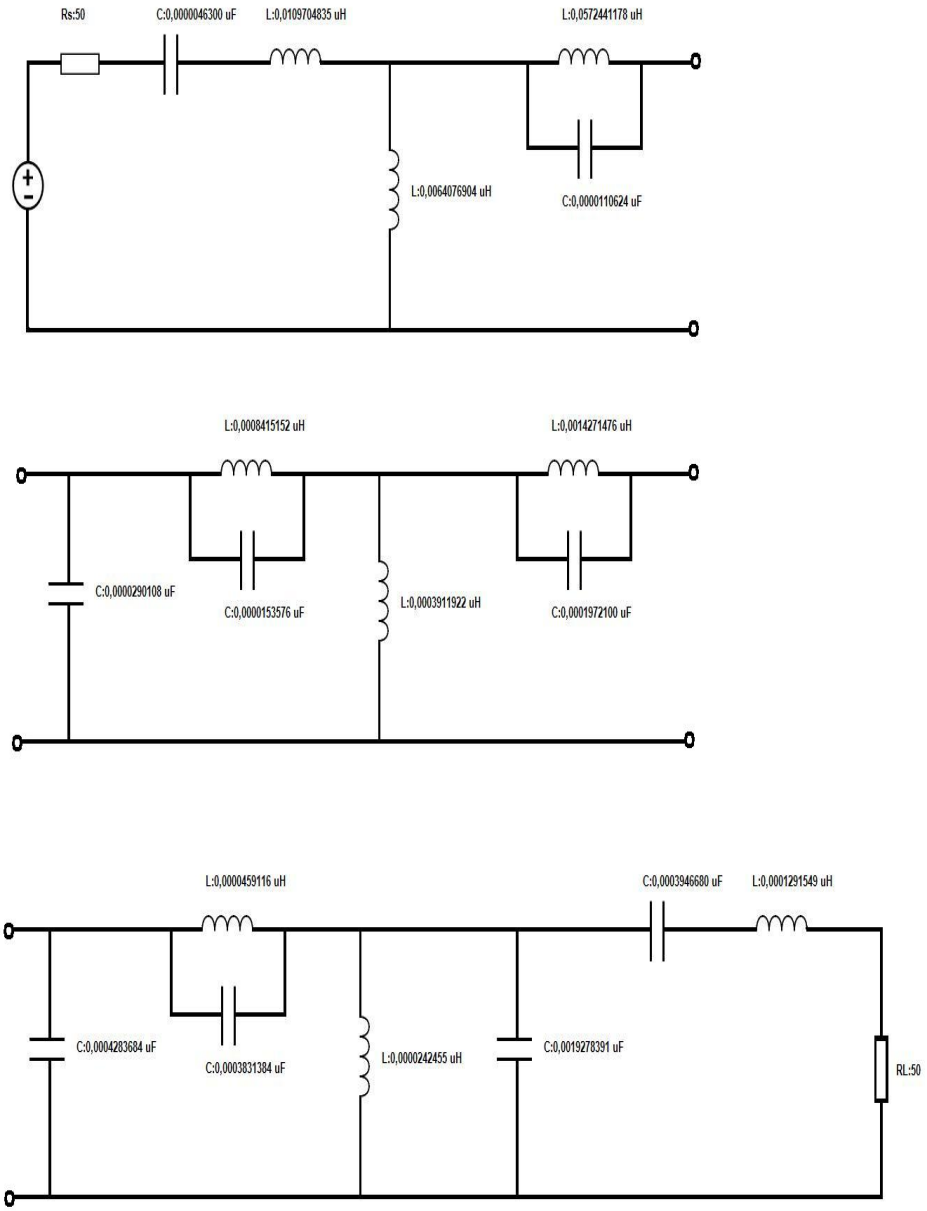


Figure 4.12: Circuit for Lumped Element Bandpass Filter

- TZ at $f = 0$
- TZ at ∞
- TZ at $f = 0$
- TZ at ∞

4.1.4 Distributed Element BandPass Filter Example

In this example, distributed element bandpass filter is designed. The filter properties specified as below.

- Distributed Structure
- Bandpass Filter
- Equiripple Filter
- Doubly Termination
- Type-A Circuit
- $Ripple = 0.5$, $f_{p1} = 4000$, $f_{p2} = 6000$, $f_q = 10000$, $R_{source} = 50$, $R_{load} = 50$
- 2 TZ at $f = 0$, 2 TZs at $f = \infty$, 2 Unit Element and TZs at $f = 3000MHz$, $f = 3500MHz$, $f = 6500MHz$, $f = 7000MHz$

First filter properties and parameters are specified like in figure 4.13

After this filter specification, *Set Values* button is invoked, *Transmission Zeros Specification Section* is opened and TZs are added as listed in figure 4.14

While transmission zeros being added to filter, filter responses are plotted at the plot section. The final responses are plotted in figure 4.15.

The filter circuit is shown in figure 4.16.

After the specification process, the transmission zeros are extracted in the following order

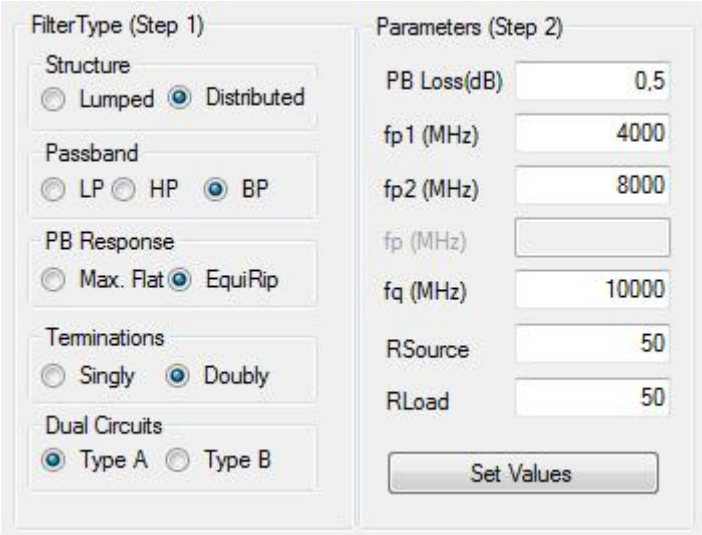


Figure 4.13: Setting Parameter for Distributed Element Bandpass Filter

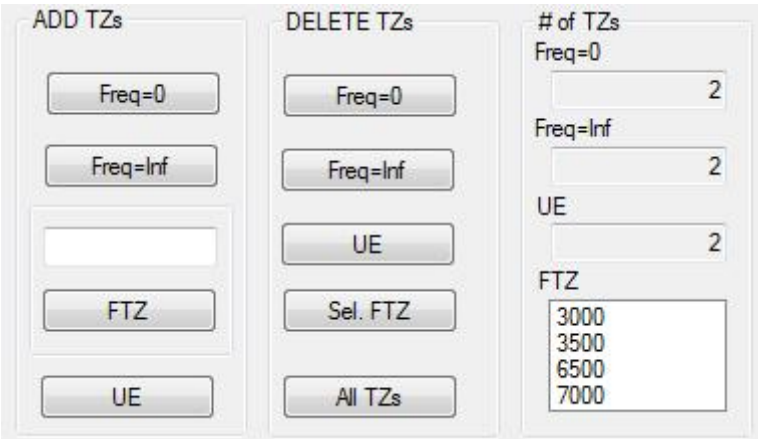


Figure 4.14: Setting Transmission Zeros for Distributed Element Bandpass Filter

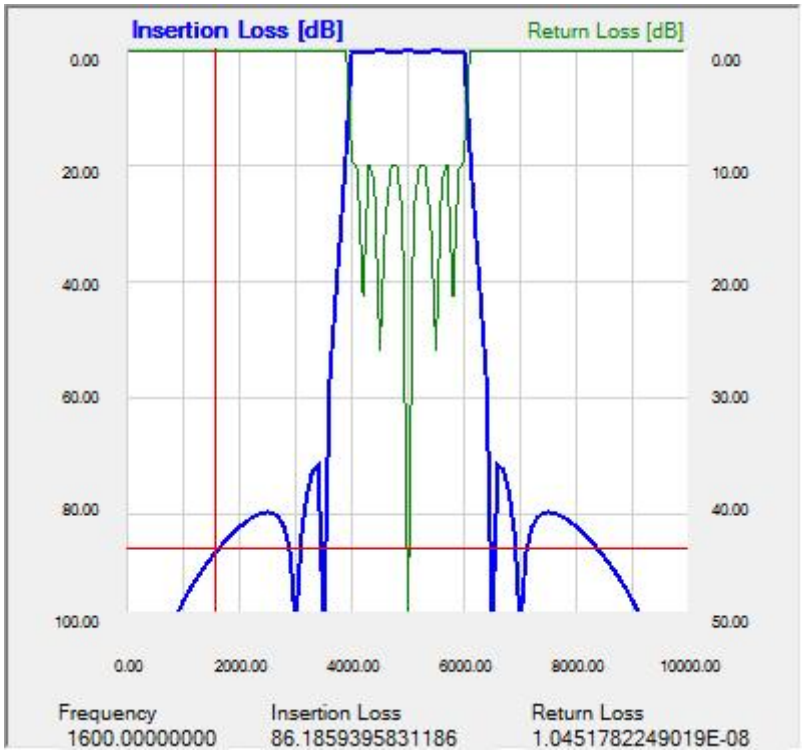


Figure 4.15: Plot for Distributed Element Bandpass Filter

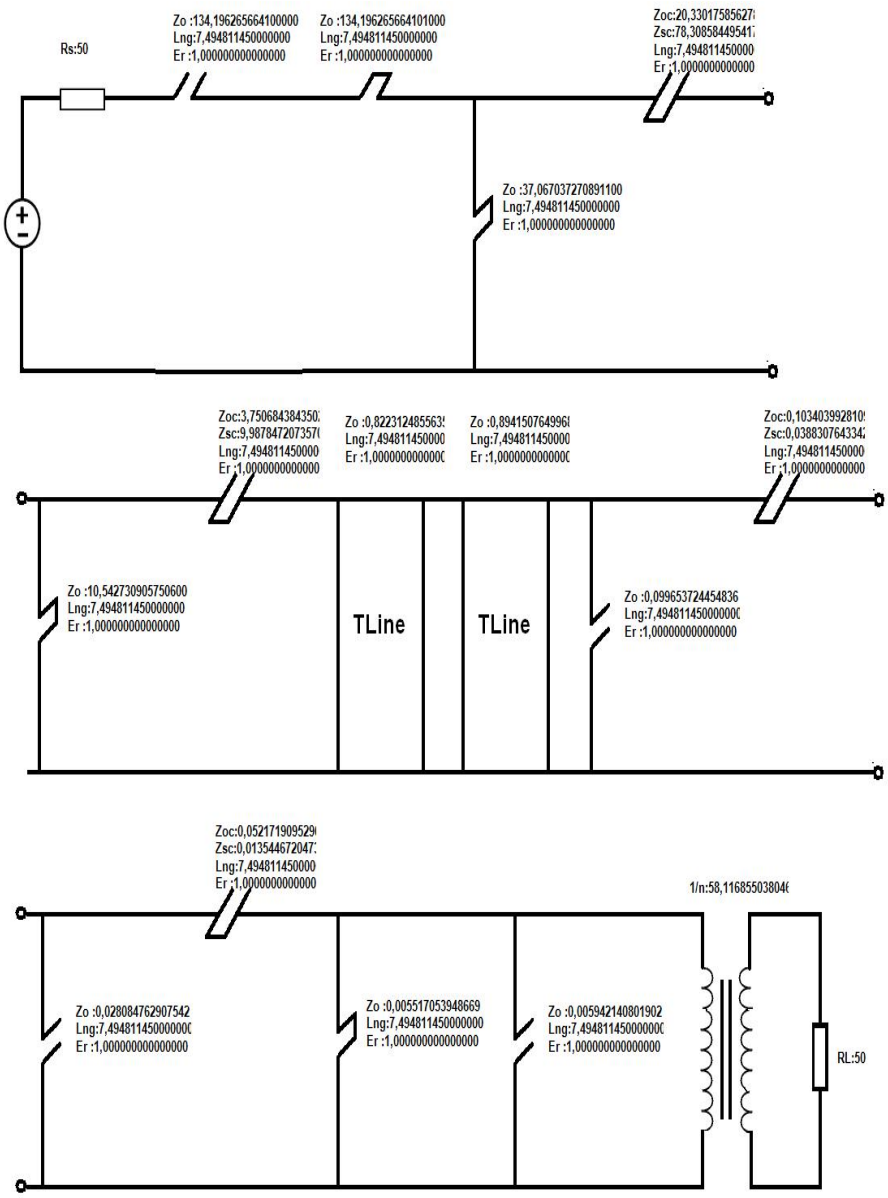


Figure 4.16: Circuit for Distributed Element Bandpass Filter

- TZ at $f = 0$
- TZ at ∞
- TZ at $f = 3000MHz$
- TZ at $f = 3500MHz$
- Unit Element
- Unit Element
- TZ at $f = 6500MHz$
- TZ at $f = 7000MHz$
- TZ at $f = 0$
- TZ at ∞

4.1.5 Distributed Element HighPass Filter Example

In this example, a distributed element highpass filter is designed. The filter properties specified as below.

- Distributed Structure
- Highpass Filter
- Maximally Filter
- Singly Termination
- Type-A Circuit
- $Ripple = 0.05$, $f_p = 4000$, $f_q = 6000$, $R_{source} = 50$, $R_{load} = 50$
- 5 TZ at $f = 0$, 3 Unit Element and TZs at $f = 3000MHz$, $f = 3500MHz$, $f = 3700MHz$, $f = 3800MHz$

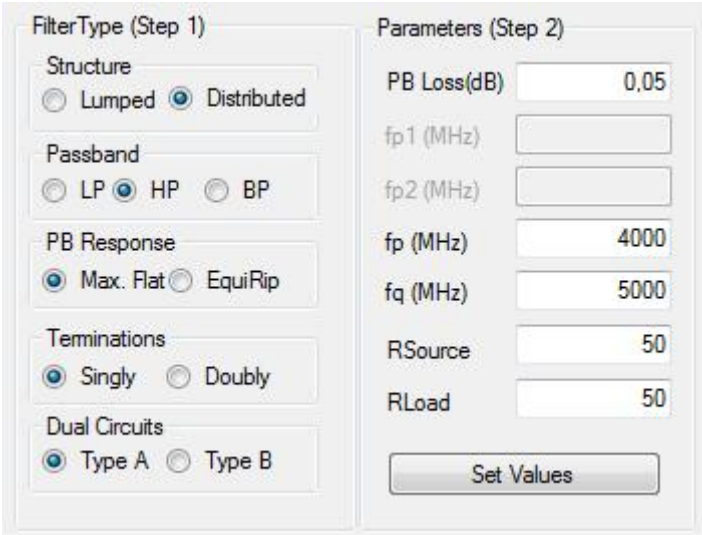


Figure 4.17: Setting Parameter for Distributed Element Highpass Filter

First, filter properties and parameters are specified like in figure 4.17

After this filter specification, *Set Values* button is invoked, *Transmission Zeros Specification Section* is opened and the TZs are added as listed in figure 4.18

While transmission zeros being added to filter, filter responses are plotted at the plot section. The final responses are plotted in figure 4.19.

After the specification process, the transmission zeros are extracted in the following order

- TZ at $f = 0$
- Unit Element
- TZ at $f = 0$
- Unit Element
- TZ at $f = 3000MHz$
- TZ at $f = 3500MHz$
- Unit Element

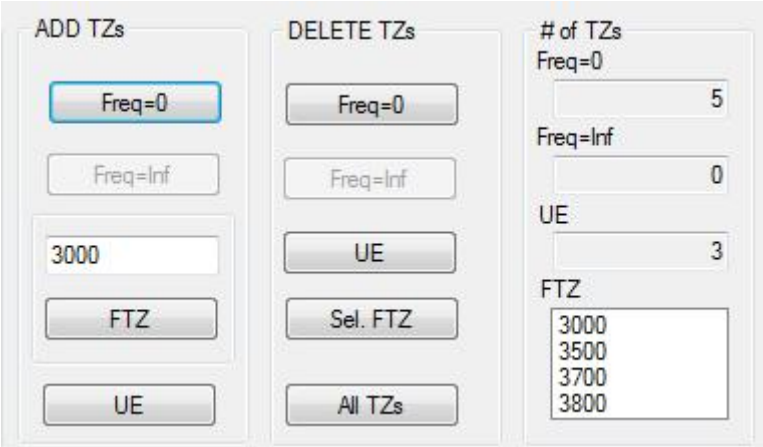


Figure 4.18: Setting Transmission Zeros for Distributed Element Highpass Filter

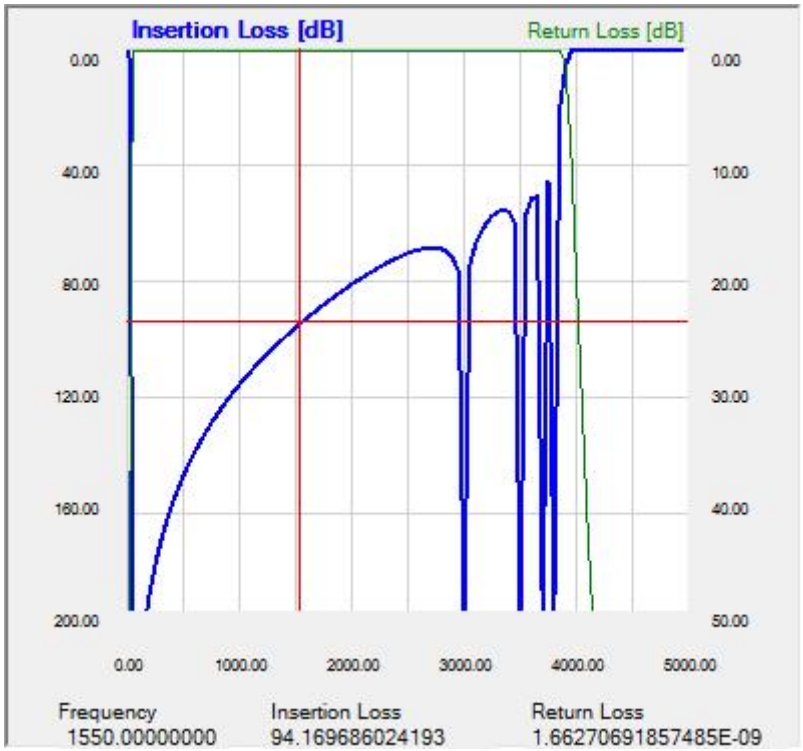


Figure 4.19: Plot for Distributed Element Highpass Filter

- TZ at $f = 3700MHz$
- TZ at $f = 3800MHz$
- TZ at $f = 0$
- TZ at $f = 0$
- TZ at $f = 0$

The filter circuit is shown in figure 4.20.

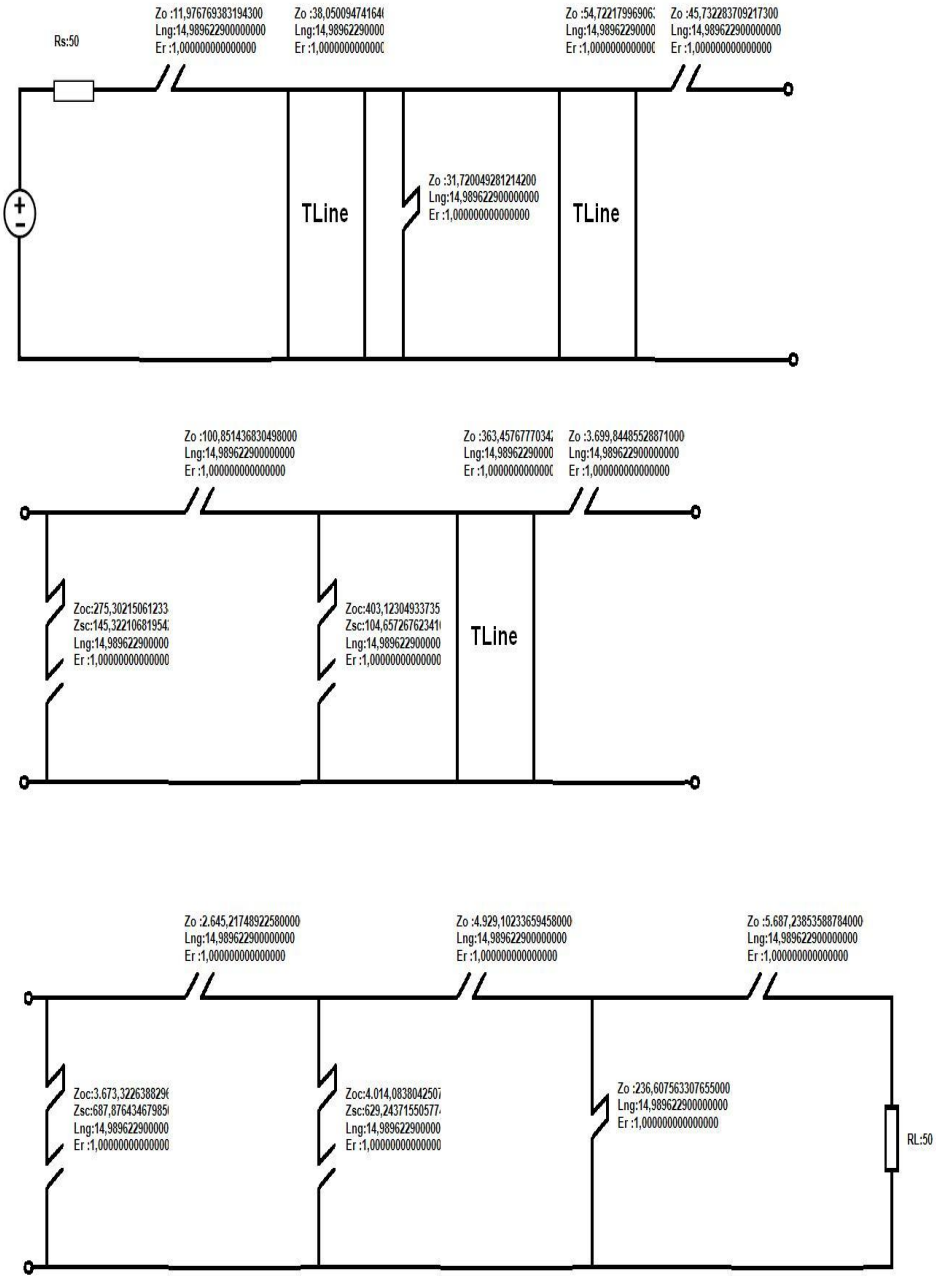


Figure 4.20: Circuit for Distributed Element Highpass Filter

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion and Future Works

In this study, Window-based computer program, named Synthesis Software, is developed for designing equal-ripple and maximally flat passband filter with general stopbands by using cascade synthesis technique. Different types of filter can be designed like Lumped element, Distributed element, Lowpass, Highpass, Bandpass, Singly terminated or Doubly terminated filters. The software is based on the synthesis part of the well-known DOS based software Filpro. It is transformed into C# with a new and improved algorithm for root finding. Synthesis Software is constructed from three parts. The first and main part is implementation of synthesis technique by using object oriented programming technique. In this way, synthesis technique implementation is isolated from other parts of Synthesis Software and it can be used by other filter design programs as a module. The second part of program is plotting filter response section. In this part Insertion Loss, Return Loss, Time Delay and Phase Delay are plotted. The last part is User Interface, which provides user-friendly environment for designing filter and uses Synthesis and Plot part as a module. The software is readily usable as engine part of more general software, which would include circuit transformations for transforming the synthesized filters into targeted topologies with realizable element values, like mixed lumped and distributed element filters, direct coupled and cross-coupled resonator filters, planar filters, dual and triple band filters, etc. These topics may form the subject of future works.

REFERENCES

[1] H. J. Orchard and G. C. Temes, ”Filter Design Using Transformed Variables”, IEEE Trans. CT, vol. CT -15, No.5, pp. 385-408, Dec. 1968.

[2] G. C. Temes and S. K. Mitra, ”Modern Filter Theory and Design”, J.Wiley and Sons, NY, 1973.

[3] Y. Sen, ”Direct Design of Lumped and Distributed Filters Using Transformed Variable Techniques”, M. Sc. Thesis, EEE Dept. Middle East Technical University, Ankara, Turkey, 1989.

[4] U. Koyaz, ”Synthesis of Lumped and Distributed Filters and Diplexers”, Ph. D. Thesis, EEE Dept. Middle East Technical University, Ankara, Turkey, 1994

[5] R. Uzun, ”Computer Aided Microwave Filter Design”,M. Sc. Thesis, EEE Dept. Middle East Technical University, Ankara, Turkey, 1990

[6] M. Karaaslan, ”Network Transformations for Realization of Lumped and Distributed Filters”, M. Sc. Thesis, EEE Dept. Middle East Technical University, Ankara, Turkey, 1994.

[7] Electronic filter, `http://en.wikipedia.org/wiki/Electronic_filter`, Retrieved Date 11 October 2008

[8] Butterworth filter, `http://en.wikipedia.org/wiki/Butterworth_filter`, Retrieved Date 11 October 2008

[9] V. Belevitch, ”Classical Network Theory”, San Francisco, Calif.Holden-Day, 1968.

[10] R. W. Daniels, ”Approximation Methods for Electronic Filter Design”, Mc Graw Hill, NY, 1974.

[11] Jenkins-Traub algorithm, `http://en.wikipedia.org/wiki/Jenkins-Traub_Algorithm_for_Polynomial_Zeros`, Retrieved Date 11 October 2008

[12] Jenkins-Traub $C++$ Implementation, `http://www.netlib.org/toms/493`, Retrieved Date 11 October 2008

[13] Polynomial Root-finding with the Jenkins-Traub Algorithm, `http://math-blog.com/?s=algorithm`, Retrieved Date 11 October 2008

[14] Extreme Optimization Numerical Libraries for .NET, `http://www.extremeoptimization.com/`, Retrieved Date 11 October 2008

[15] Weierstrass Iteration, `http://www.codeproject.com/`, Retrieved Date 11 October 2008

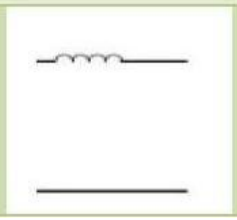
[16] Dean A. Frickey, ”Conversions Between S, Z, Y, H, ABCD,and T Parameters which are Valid for Complex Source and Load Impedances”, IEEE Trans. on microvawe theory and techniques. Vol 42, No.2, pp. 205-211, Feb. 1994

[17] H. Baher, ”Synthesis of Electrical Networks”, J. Wiley and Sons, NY, 1984.

APPENDIX A

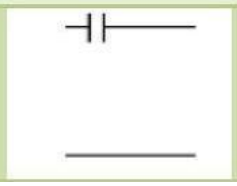
ABCD MATRICES OF SOME ELEMENTS

Series Inductor



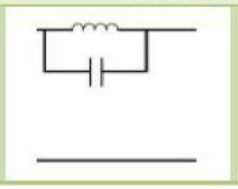
$$\begin{bmatrix} 1 & j\omega L \\ 0 & 1 \end{bmatrix} \quad \text{(A-1)}$$

Series Capacitor



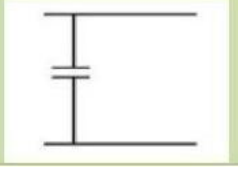
$$\begin{bmatrix} 1 & j\frac{-1}{\omega C} \\ 0 & 1 \end{bmatrix} \quad \text{(A-2)}$$

Series Connected Parallel Inductor-Capacitor



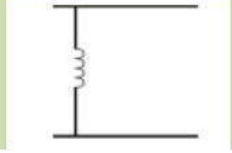
$$\begin{bmatrix} 1 & j\frac{\omega L}{1-\omega^2 LC} \\ 0 & 1 \end{bmatrix} \quad \text{(A-3)}$$

Shunt Capacitor



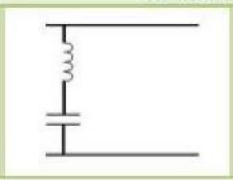
$$\begin{bmatrix} 1 & 0 \\ j\omega C & 1 \end{bmatrix} \quad \text{(A-4)}$$

Shunt Inductor



$$\begin{bmatrix} 1 & 0 \\ j\frac{-1}{\omega L} & 1 \end{bmatrix} \quad \text{(A-5)}$$

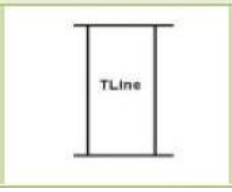
Shunt Connected Series Inductor-Capacitor



$$\begin{bmatrix} 1 & 0 \\ j\dfrac{\omega C}{1-\omega^2 LC} & 1 \end{bmatrix}$$

(A-6)

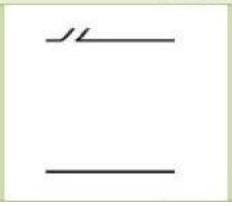
Transmission Line



$$\begin{bmatrix} \cos\theta & jZ_o\sin\theta \\ j\dfrac{\sin\theta}{Z_o} & \cos\theta \end{bmatrix}$$

(A-7)

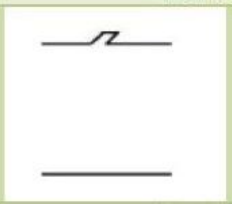
Series Connected Open-Circuited Stub



$$\begin{bmatrix} 1 & j(-Z_o\cot\theta) \\ 0 & 1 \end{bmatrix}$$

(A-8)

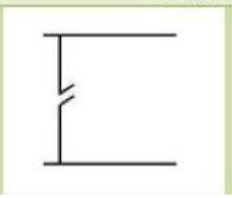
Series Connected Short-Circuited Stub



$$\begin{bmatrix} 1 & j(Z_o\tan\theta) \\ 0 & 1 \end{bmatrix}$$

(A-9)

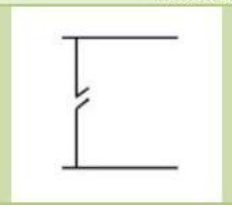
Shunt Connected Short-Circuited Stub



$$\begin{bmatrix} 1 & 0 \\ j\dfrac{-1}{\tan\theta Z_o} & 1 \end{bmatrix}$$

(A-10)

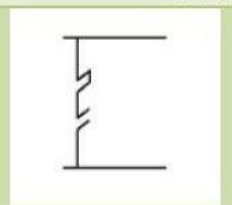
Shunt Connected Open-Circuited Stub



$$\begin{bmatrix} 1 & 0 \\ j\dfrac{\tan\theta}{Z_o} & 1 \end{bmatrix}$$

(A-11)

Shunt Connected Series Resonator



$$\begin{bmatrix} 1 & 0 \\ j\dfrac{\Omega}{Z_{oc}\Omega^2 Z_{sc}} & 1 \end{bmatrix}$$

(A-12)

Series Connected Parallel Resonator			
			$\begin{bmatrix} 1 & j\dfrac{\Omega Z_{oc} Z_{sc}}{Z_{oc} - \Omega^2 Z_{sc}} \\ 0 & 1 \end{bmatrix}$

(A-13)

Transformer			
			$\begin{bmatrix} \frac{1}{n} & 0 \\ 0 & n \end{bmatrix}$

(A-14)