

**Designing LC Ladder Filters with Lossy Elements**

by

Timothy A Brown

A thesis submitted to the Graduate Faculty of  
Auburn University  
in partial fulfillment of the  
requirements for the Degree of  
Master of Science

Auburn, Alabama  
May 5<sup>th</sup>, 2013

Keywords: ladder, filters, lossy, synthesis

Copyright 2013 by Timothy A Brown

Approved by

Bogdan M. Wilamowski, Chair, Professor of Electrical and Computer Engineering  
Michael Baginski, Professor of Electrical and Computer Engineering  
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering

## Abstract

Many publications and many books have been written on LC Ladder filters, however, most of the material in circulation only deals with ideal (lossless) LC ladder filters. It is assumed that in the lossless design, all of the reactance values are ideal ones. However, since reactance elements are actually lossy, the purpose of this thesis is to show how to design LC ladder filters with lossy components, and achieve optimal responses. Three different methods are presented to correct the lossy ladder filter. The first method uses the Sum Squared Error (SSE) to minimize the difference between the poles of the ideal filter and the lossy filter. The second method uses the SSE to minimize the difference between the coefficients of the ideal filter's transfer function and the lossy filter's transfer function. The final method uses the SSE to minimize the difference between magnitude plot of the ideal filter and lossy filter. The first and second methods both use the Nelder-Mead algorithm for the optimization scheme, and the third method uses the Levenberg-Marquardt algorithm. The first two methods work well but only for low-pass filters. In the case of more complex designs, the number of roots and the filters order in the ideal filter and lossy filter do not match, so this method cannot be used. In conclusion, the third method where the frequency magnitude plots of the ideal and lossy filter are being matched, provide the best solution. As a result, this method allows one to design a ladder filter with lossy components, while achieving ideal-like results.

## Acknowledgments

I would like to thank Dr. Bogdan Wilamowski, for his support and encouragement during my time here at Auburn, and for letting me work under him as a graduate student. I would also like to thank my committee members, Dr. Michael Baginski, and Dr. Thaddeus Roppel for being on my committee. I also would like to thank all of my colleagues in Room 409 in Broun Hall, as well as Michael Pukish and Xing Wu for helping me on certain projects in MATLAB. Last but not least, I would like to thank my parents who installed in me the value of education, without their guidance, I would not be where I am today.

## Table of Contents

Abstract.....	ii
Acknowledgments.....	iii
Table of Figures.....	vi
List of Tables .....	ix
1 Introduction .....	1
1.1 Thesis Outline.....	1
2 Low-pass Filters.....	2
2.1 Low Pass Filter Response .....	2
2.2 Simple low-pass filter circuit.....	4
2.3 2 <sup>nd</sup> Order Low-pass Filter .....	5
3 Synthesis of lossless LC Ladder Filters .....	7
3.1 Background .....	7
3.2 Properties of lossless ladders.....	7
3.3 A synthesis strategy .....	9
3.3.1 Ladder Synthesis Example.....	11
3.4 Calculate the Transfer Function of Ladder Circuit .....	12
3.5 Frequency Transformation from low-pass to other types of filters .....	16
4 Synthesis of lossy LC Ladder Filters.....	18
4.1 Introduction .....	18
4.2 Background .....	18
4.3 Procedure of designing Lossy Ladder Filters.....	20
4.3.1 Calculating the transfer function of a lossy ladder filter .....	21
4.3.2 Calculating the Error Function .....	26
4.3.3 Optimization Routine - Nelder Mead Method.....	27
5 Nelder-Mead Simulation Results – Error Calculated From Roots.....	31
5.1 3 <sup>rd</sup> Order Chebyshev Low-pass Filter .....	31
5.2 5 <sup>th</sup> Order Butterworth Low-pass filter .....	34

5.3	8 <sup>th</sup> Order Chebyshev Band-pass Filter.....	37
6	Nelder-Mead Simulation Results – Error Calculated From Coefficients.....	41
6.1	3 <sup>rd</sup> Order Chebyshev Low-pass Filter .....	41
6.2	5 <sup>th</sup> Order Butterworth Low-pass Filter.....	45
6.3	8 <sup>th</sup> Order Chebyshev Band-pass Filter.....	49
6.4	MATLAB Algorithm Limitations.....	51
7	Levenberg – Marquardt (LM) Simulation Results.....	53
7.1	3 <sup>rd</sup> Order Chebyshev Low-pass Filter .....	57
7.2	5 <sup>th</sup> Order Butterworth Low-Pass Filter.....	59
7.3	4 <sup>th</sup> Order Chebyshev Low-Pass Filter .....	61
7.4	8 <sup>th</sup> Order Chebyshev Band-pass Filter.....	63
7.5	8 <sup>th</sup> Order Butterworth Band-pass Filter .....	65
8	Conclusion.....	67
9	References .....	69
10	Appendices.....	71
A	Matlab input function: main.m.....	71
B	Matlab Impedance Function: impedance.m.....	72
C	Matlab Error and Plotting Function: error.m.....	73
D	Matlab Transfer Function: Ttra_poly.m.....	76
E	2 <sup>nd</sup> Order low-pass filter symbolic transfer function .....	76
F	3 <sup>rd</sup> Order low-pass filter symbolic transfer function.....	76
G	4 <sup>th</sup> Order low-pass filter symbolic transfer function.....	77
H	5 <sup>th</sup> Order low-pass filter symbolic transfer function.....	77
I	6 <sup>th</sup> Order low-pass filter symbolic transfer function.....	77
J	7 <sup>th</sup> Order low-pass filter symbolic transfer function.....	78
K	8 <sup>th</sup> Order low-pass filter symbolic transfer function.....	78
L	9 <sup>th</sup> Order low-pass filter symbolic transfer function.....	79

## Table of Figures

Figure 2-1: Brickwall Low-Pass Filter Amplitude Response .....	2
Figure 2-2: More realistic low-pass filter .....	3
Figure 2-3: Simple Low-pass Filter .....	4
Figure 2-4: Frequency Response of simple low-pass filter .....	5
Figure 2-5: 2 <sup>nd</sup> Order low-pass filter .....	5
Figure 2-6: Voltage Divider Circuit .....	6
Figure 3-1 Doubly Terminated lossless ladder:.....	8
Figure 3-2: Ladder Synthesis Example .....	12
Figure 3-3: 5 <sup>th</sup> order Ideal Chebyshev low-pass filter .....	13
Figure 3-4: 5 <sup>th</sup> order ladder with series impedances and shunt admittances .....	13
Figure 3-5: 5 <sup>th</sup> order Chebyshev low-pass filter .....	16
Figure 3-6: Filter transformation from low-pass to other types of filters .....	17
Figure 4-1: Basic Structure of Ladder Filter .....	18
Figure 4-2: Models of Real Capacitors and Real Inductors .....	19
Figure 4-3: Movement of Poles due to lossy elements .....	20
Figure 4-4: 3 <sup>rd</sup> order lossy ladder filter .....	21
Figure 4-5: Stages of 3 <sup>rd</sup> order low-pass lossy filter .....	23
Figure 4-6: Pole locations for 3 <sup>rd</sup> order ideal and lossy low-pass filter.....	26
Figure 4-7: The Triangle ABC, the midpoint M and the Reflected point R .....	29
Figure 4-8: The Triangle BGW and point R with extended point E .....	29

Figure 4-9: The Contraction point $C_1$ or $C_2$ .....	30
Figure 4-10: Shrinkage toward the Triangle B .....	30
Figure 5-1: 3 <sup>rd</sup> order ideal Chebyshev low-pass filter .....	32
Figure 5-2: 3 <sup>rd</sup> order lossy Chebyshev low-pass filter .....	32
Figure 5-3: 3rd order Chebyshev Pole plot and Frequency Response – Before Optimization .....	33
Figure 5-4: 3 <sup>rd</sup> order Chebyshev Pole plot and Frequency Response – After Optimization .....	33
Figure 5-5: 5 <sup>th</sup> order ideal Butterworth low-pass filter.....	35
Figure 5-6: 5 <sup>th</sup> order lossy Butterworth low-pass filter.....	35
Figure 5-7: 5 <sup>th</sup> order Butterworth Pole plot and Frequency Response – Before Optimization .....	36
Figure 5-8: 5 <sup>th</sup> order Butterworth Pole plot and Frequency Response – After Optimization.....	36
Figure 5-9: 8 <sup>th</sup> order Chebyshev band-pass filter.....	38
Figure 5-10: 8th order Lossy Chebyshev band-pass filter .....	38
Figure 5-11: 8 <sup>th</sup> order Band-pass Pole plot and Frequency Response – Before Optimization .....	40
Figure 5-12: 8 <sup>th</sup> order Band-pass Pole plot and Frequency Response – After Optimization .....	40
Figure 6-1: 3 <sup>rd</sup> order ideal Chebyshev low-pass filter .....	41
Figure 6-2: 3 <sup>rd</sup> order lossy Chebyshev low-pass filter .....	41
Figure 6-3: 3 <sup>rd</sup> order Chebyshev Coefficient Value Vs Order Plot – Before Optimization .....	43
Figure 6-4: 3 <sup>rd</sup> order Chebyshev Coefficient Value Vs Order Plot – After Optimization.....	43
Figure 6-5: 3 <sup>rd</sup> order Chebyshev filter - Before Optimization .....	44
Figure 6-6: 3 <sup>rd</sup> order Chebyshev filter- After Optimization.....	44
Figure 6-7: 5 <sup>th</sup> order ideal Butterworth low-pass filter.....	45
Figure 6-8: 3 <sup>rd</sup> order lossy Chebyshev low-pass filter .....	45
Figure 6-9: 5 <sup>th</sup> order Butterworth Coefficient Value Vs Order Plot – Before Optimization.....	47
Figure 6-10: 5 <sup>th</sup> order Butterworth Coefficient Value Vs Order Plot – After Optimization .....	47
Figure 6-11: 5 <sup>th</sup> order Butterworth filter- Before Optimization.....	48
Figure 6-12: 5 <sup>th</sup> order Butterworth filter- After Optimization .....	48

Figure 6-13: 8 <sup>th</sup> order Chebyshev Band-pass filter.....	49
Figure 6-14: 8 <sup>th</sup> order lossy Chebyshev Band-pass filter.....	49
Figure 6-15: 8 <sup>th</sup> order Band-pass Coefficient Value Vs Order Plot – Before Optimization .....	50
Figure 6-16: 8 <sup>th</sup> order Band-pass Frequency Response – Before Optimization.....	50
Figure 6-17: Example of Ideal Bandpass Filter Impedance .....	51
Figure 6-18: Example of Lossy Bandpass Filter Impedance .....	52
Figure 7-1: Block Diagram and Pseudocode for LM Algorithm.....	56
Figure 7-2: 3 <sup>rd</sup> Order Chebyshev Low-pass Filter .....	57
Figure 7-3: 3rd Order Chebyshev Low-Pass Filter Frequency Plot – Before Optimization .....	58
Figure 7-4: 3rd Order Chebyshev Low-Pass Filter Frequency Plot – After Optimization.....	58
Figure 7-5: 5 <sup>th</sup> Order Butterworth Low-pass Filter .....	59
Figure 7-6: 5 <sup>th</sup> Order Butterworth Low-Pass Filter Frequency Plot – Before Optimization.....	60
Figure 7-7: 5 <sup>th</sup> Order Butterworth Low-Pass Filter Frequency Plot – After Optimization .....	60
Figure 7-8: 4 <sup>th</sup> Order Chebyshev Low-Pass Filter .....	61
Figure 7-9: 4 <sup>th</sup> Order Chebyshev Low-Pass Filter Frequency Plot – Before Optimization .....	62
Figure 7-10: 4 <sup>th</sup> Order Chebyshev Low-Pass Filter Frequency Plot – After Optimization.....	62
Figure 7-11: 8 <sup>th</sup> Order Chebyshev Band-pass Filter .....	63
Figure 7-12: 8 <sup>th</sup> Order Chebyshev Band-pass Filter Frequency Plot – Before Optimization.....	64
Figure 7-13: 8 <sup>th</sup> Order Chebyshev Band-pass Filter Frequency Plot – After Optimization .....	64
Figure 7-14: 8 <sup>th</sup> Order Butterworth Band-pass Filter.....	65
Figure 7-15: 8 <sup>th</sup> Order Butterworth Band-pass Filter Frequency Plot – Before Optimization .....	66
Figure 7-16: 8 <sup>th</sup> Order Butterworth Band-pass Filter Frequency Plot – After Optimization.....	66



## List of Tables

Table 2-1 Classic Analog Filters .....	4
Table 3-1: Polynomials D(s) for Butterworth Filter.....	10
Table 3-2: Ladder Synthesis Division and Element Extraction .....	12
Table 3-3: Element Values for series and shunt ladder .....	13
Table 3-4: 5 <sup>th</sup> order transfer function .....	14
Table 3-5: Coefficients of 5 <sup>th</sup> Chebyshev Transfer Function .....	14
Table 3-6: Element Values for 5th Order Low-pass Filter .....	15
Table 4-1: 3 <sup>rd</sup> Order Lossy Element Values .....	21
Table 4-2: 3 <sup>rd</sup> Order lossy transfer function.....	22
Table 4-3: 3 <sup>rd</sup> Order Lossy Transfer Function - Mathematical .....	22
Table 4-4: 3 <sup>rd</sup> Order ideal low-pass filter with symbolic coefficients – Calculated in MATLAB .....	24
Table 4-5: 3 <sup>rd</sup> Order lossy low-pass filter with Symbolic coefficients – Calculated in MATLAB.....	25
Table 4-6: Ideal Roots and Lossy roots for 3 <sup>rd</sup> order low-pass filter .....	27
Table 5-1: 3 <sup>rd</sup> Order Chebyshev low-pass filter specifications.....	31
Table 5-2: 3 <sup>rd</sup> Order Chebyshev low-pass filter element values .....	32
Table 5-3: Optimized values for 3 <sup>rd</sup> order Chebyshev.....	33
Table 5-4: 5 <sup>th</sup> Order Butterworth low-pass filter specifications .....	34
Table 5-5: 5 <sup>th</sup> Order Butterworth low-pass filter element values.....	35
Table 5-6: Optimized values for 5 <sup>th</sup> order Butterworth .....	36
Table 5-7: 8 <sup>th</sup> Order Chebyshev Band-pass low-pass filter specifications .....	37

Table 5-8: 8 <sup>th</sup> Order Chebyshev Band-pass low-pass filter element values with Q=10% .....	38
Table 5-9: Optimized values for 8 <sup>th</sup> order Band-pass filter .....	39
Table 6-1: 3 <sup>rd</sup> Order Chebyshev Filter with Ideal and Optimized Values .....	42
Table 6-2: 5 <sup>th</sup> Order Butterworth Filter with Ideal and Optimized Values .....	46
Table 6-3: Coefficients for the Transfer Function for 4 <sup>th</sup> order Ideal Band-pass Filter with Q = 10% .....	53
Table 7-1 LM Algorithm Basic Steps.....	57
Table 7-2: 3 <sup>rd</sup> Order Chebyshev Low-Pass Filter with Ideal, Lossy and Optimized Values .....	57
Table 7-3: 5 <sup>th</sup> Order Butterworth Low-Pass Filter with Ideal, Lossy and Optimized Values .....	59
Table 7-4: 4th Order Chebyshev Low-Pass Filter with Ideal, Lossy and Optimized Values .....	61
Table 7-5: 8th Order Chebyshev Band-pass Filter with Ideal, Lossy and Optimized Values.....	63
Table 7-6: 8th Order Butterworth Band-pass Filter with Ideal, Lossy and Optimized Values .....	66

# 1 Introduction

LC (or RLC) ladder filters play a role in many electrical systems. They are used in many types of oscillator circuits and they are also very important when it comes to tuning, such as radio receivers and television sets, where they are used to select a narrow range of frequencies from a broad spectrum. An LC ladder filter can be used as a band-pass filter, low pass-filter, high-pass filter, or a band-stop filter. The frequency response of each one of these types of filters can greatly impact the significance/performance of the whole system. This is a bad thing and can definitely decrease performance on the whole system. Since communication systems demand high-performance filters, filters need to have characteristics as close to ideal as possible.

Since classical filter designs are done with ideal (lossless) components, ideal-like responses are not possible due to the fact that real inductors and capacitors have losses (reactance). The greater the losses of the reactance elements used in the filter, the greater the difference between the frequency response of the filter, and the frequency response that the filter was intended to have. The lack of ideal reactance elements makes all of the classical design methods inefficient for practical applications. The purpose of this thesis is to show how to design LC ladder filters, to achieve ideal-like responses, with lossy components. Three different ways are presented in this thesis to help minimize the error between the ideal filter and lossy filter, as well as two different optimization schemes to aid in the calculation of the new lossy component values. Using the algorithms proposed in this thesis allows one to design a ladder filter with lossy components, while achieving ideal-like results.

## 1.1 Thesis Outline

A basic overview of low-pass filters is given in Chapter 2. It includes a 2<sup>nd</sup> order filter and explains how this basic filter is the basis for all ladder filters. Chapter 3 explains the background of the LC ladder filter as well as the properties of lossless ladder filters. This chapter also presents a unique synthesis strategy to design ladder filters, as well as how to calculate the transfer function for an ideal filter, and how to transform low-pass filters into other types of filters. Chapter 4 explains how to design a lossy ladder filter, as well as how to determine the lossy transfer function mathematically, and in MATLAB. How to calculate the

error function and how the Nelder-Mead optimization function works is also presented in Chapter 3.

Chapter 4 shows the simulation results with the error calculated from the roots of the filter, and Chapter 5 shows the simulation results with the error calculated from the coefficients of the lossy and ideal transfer functions. This chapter also explains the MATLAB algorithm limitations for more complex filters. The Levenberg-Marquardt algorithm and simulation results are given in Chapter 6. Finally, the conclusion is given in Chapter 7, References are in Chapter 8, and the Appendices follow.

## 2 Low-pass Filters

Low-pass filters have been around for a long time and are perhaps the most well-known filter when it comes to LC passive ladder filters. However, what most people do not know is that the low pass filter design approach is the basic prototype design when it comes to other types of filter responses. Some of these responses include the high-pass filter, the band-pass filter, and the band-stop filter. Since the low-pass filter is the basic building block for all other filter designs, I will discuss how the LC low-pass filter operates.

### 2.1 Low Pass Filter Response

The brickwall low-pass frequency response is shown in Figure 2-1. The vertical axis is attenuation, and increases downwards. As long as the frequency is below the cut-off frequency,  $\omega_c$ , inputs

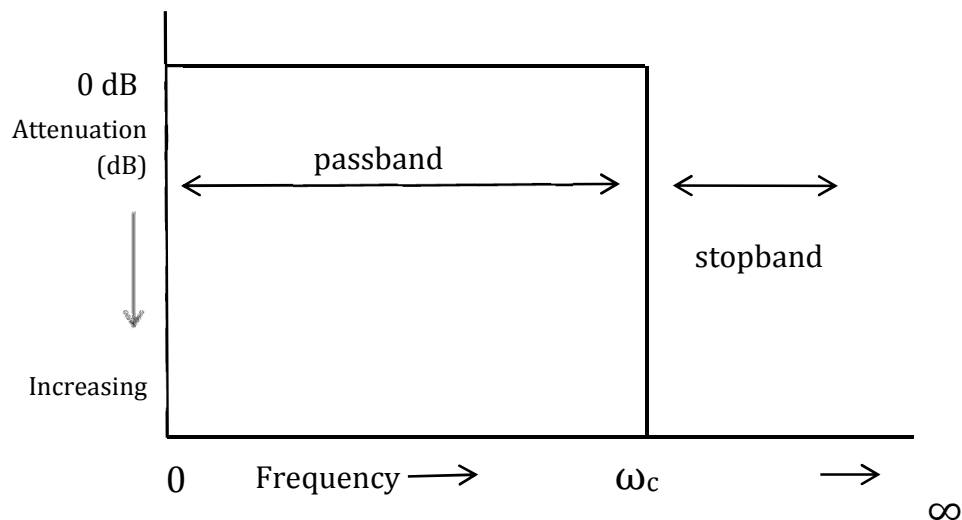
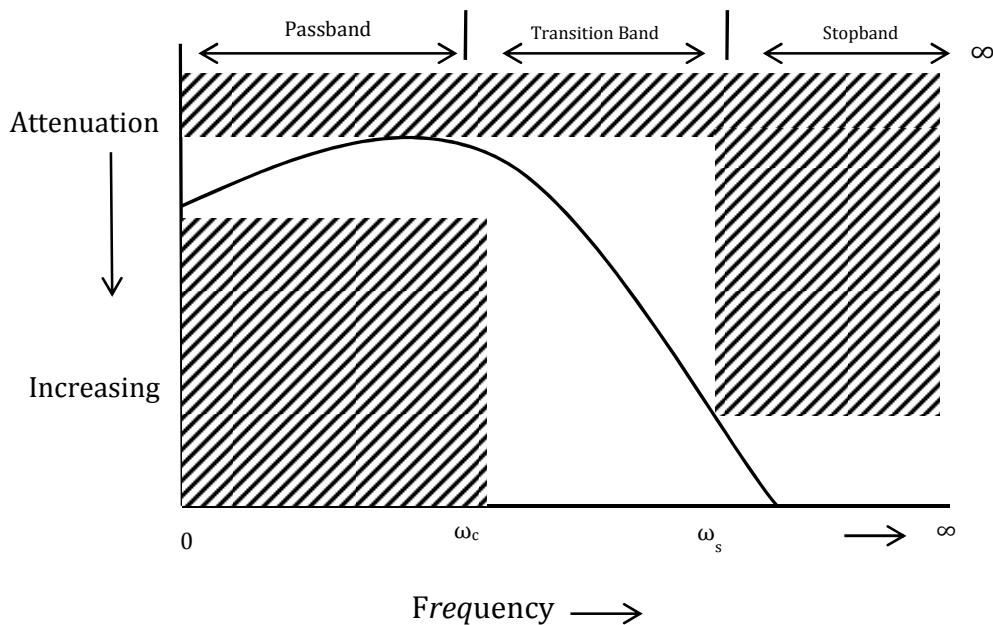


Figure 2-1: Brickwall Low-Pass Filter Amplitude Response

are passed in with no attenuation. However, inputs above  $\omega_c$  are attenuated. Frequencies up to  $\omega_c$  are called the “passband”, while frequencies after  $\omega_c$  are called the “stopband”. The passband is the frequency band of the wanted signal, while the stopband is the frequency band of the un-wanted signal.

As can be seen from Figure 2-1 the transition from passband to stopband is infinitesimal. This type of narrow response is unachievable in practice, but it represents the ideal response of a low pass filter. The more realistic response of a low-pass filter is shown in Figure 2-2. The un-shaded areas show



**Figure 2-2: More realistic low-pass filter**

the degree of freedom that the designer has for the frequency response. As can be seen in Figure 2-2, the passband does not have to be perfectly flat. There can be many variations in the passband, from perfectly flat, to ripples in the passband. There also can be variations in the stopband, from a perfectly flat response, to ripples in the stopband. There are four main types of classic analog low pass filters that create these different kinds of responses. A brief explanation of each type of filter is given in Table 2-1

Filter	Frequency Response
Butterworth	Flattest pass-band - poor roll-off rate
Chebyshev	Pass-band ripple - better roll-off rate
Elliptic	Pass- and -stop band ripple – steepest roll off rate
Inverse Chebyshev	Ripple in stop band

Table 2-1 Classic Analog Filters

## 2.2 Simple low-pass filter circuit

A simple low-pass filter is shown in Figure 2-3. It contains one component which is the capacitance, whose reactance varies with the frequency, which in turn causes the output voltage to change as the frequency changes. The reactance is given by  $X_c = 1/\omega C$ , where  $\omega$  is the angular frequency, and decreases as the frequency increases, causing a voltage divider action with the resistor (R1). The

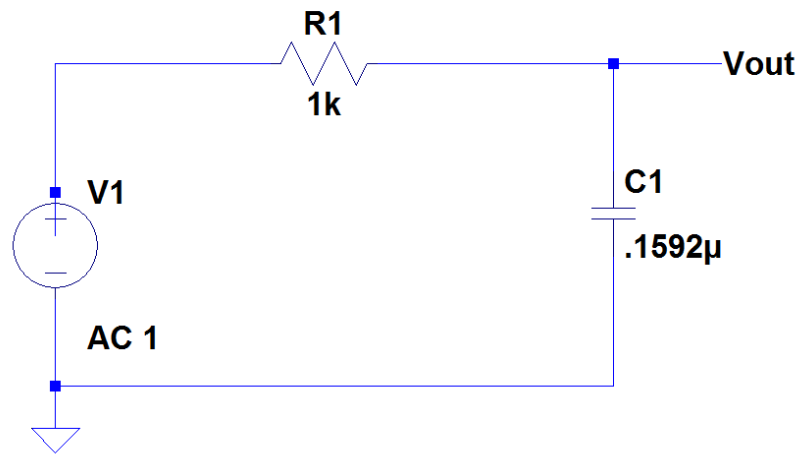
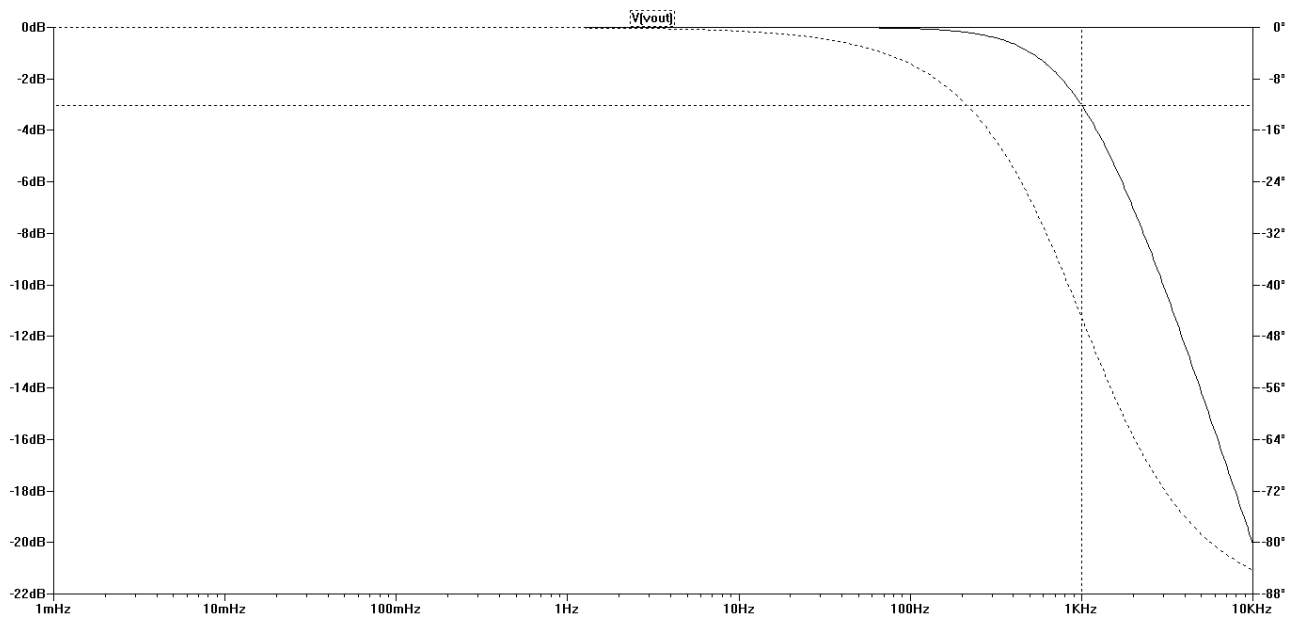


Figure 2-3: Simple Low-pass Filter

capacitor has reactance and blocks low frequency signals allowing them to go to the load, but at higher frequencies, the reactance drops and the capacitor acts like a short circuit. The frequency response of the circuit is shown in Figure 2-4.

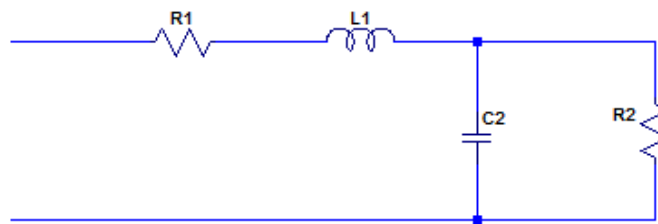


**Figure 2-4: Frequency Response of simple low-pass filter**

The cutoff frequency is defined by  $\omega_c = 1/RC$ , where  $\omega_c$  is in rad/s, R is in ohms, and C in Farads. This configuration has a single pole and a 3-dB cutoff frequency of 1kHz as shown in the figure above.

### 2.3 2<sup>nd</sup> Order Low-pass Filter

A simple low pass filter like this is mostly used in decoupling, or to prevent AC interference generated by power supplies. However, a single pole circuit may be inadequate. A second pole can be added by



**Figure 2-5: 2<sup>nd</sup> Order low-pass filter**

adding a series inductor to the circuit, as shown in Figure 2 This circuit now has two components to stop AC from passing from the input to the output. The increasing reactance of the inductor,  $X_i = \omega L$ , and the decreasing reactance of the capacitor,  $X_c = 1/\omega_c$ , make this a two-pole or 2<sup>nd</sup> order circuit. Once again at low frequencies the inductor is a short and the capacitor is an open, this couples the signal at the output. At resonance (when the voltage across the inductor is equal to the voltage across the capacitor), the 3-dB

cutoff frequency is set. Once again at high frequencies, the inductors reactance increases, and the capacitors reactance decreases, decouple the input from the output, which shorts the output signal to ground.

The transfer function for this circuit is seen as a voltage divider, where  $Z_1 = sL_1 + R_1$ , and  $Y_2 = sC_2 + G_2$ . The transfer function is shown in Equation (2.1).

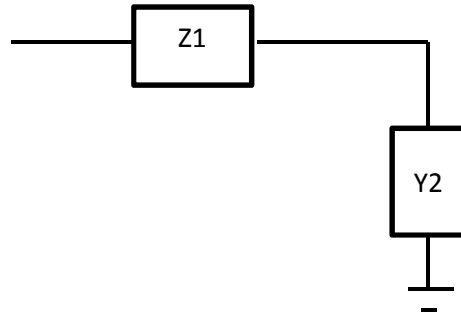


Figure 2-6: Voltage Divider Circuit

$$\frac{V_{out}}{V_{in}} = \frac{\frac{1}{Y_2}}{\frac{1}{Y_2} + Z_1} \quad (2.1)$$

Then if we plug in and simplify:

$$\frac{V_{out}}{V_{in}} = \frac{1}{s^2(L_1C_2) + s(L_1G_2 + C_2R_1) + 1 + R_1G_2} \quad (2.2)$$

The simplified transfer function is shown in equation (2.2). Since the variable “s” is raised to the second order, this is proof that this is a 2<sup>nd</sup> order filter, and has two complex poles. Also, since this is a 2<sup>nd</sup> order filter, it will roll off at 40 dB/dec and the steepness and bandwidth of the filter is controlled by the quality factor (Q). The 2<sup>nd</sup> order ladder filter is the proto-type design for all higher order ladder filters. To achieve an even higher order filter, all one must do is keep adding inductors in series, and capacitors in parallel; this is the beauty of the LC Ladder network circuit. However, the inductors and capacitors must be chosen so that the filter has the correct cutoff frequency at  $\omega_c$ , and is matched with the load.



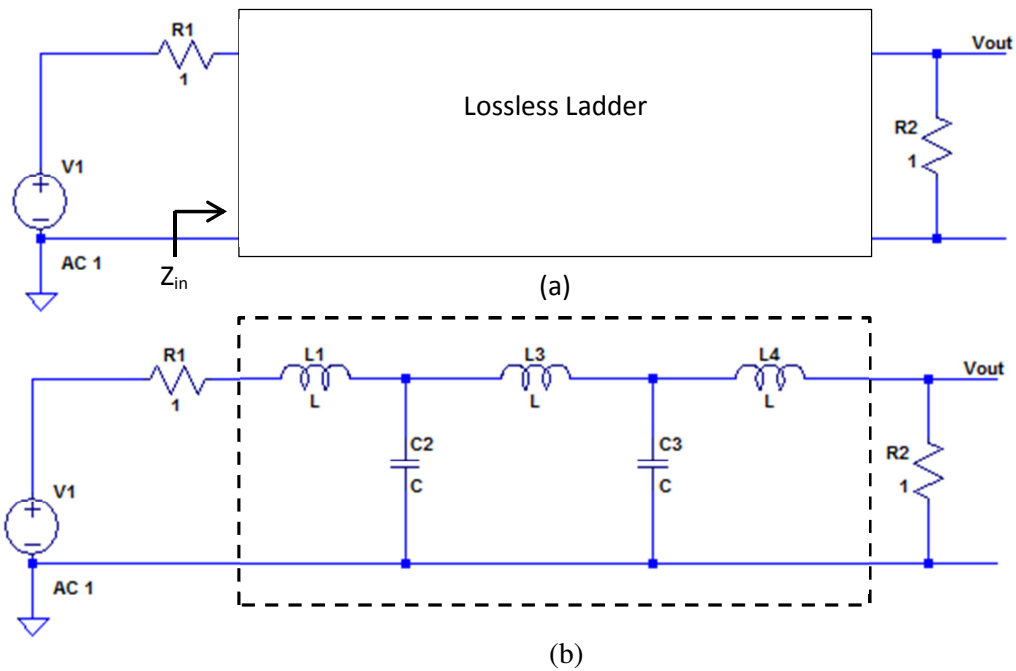
## 3 Synthesis of lossless LC Ladder Filters

### 3.1 Background

Many methods exist for synthesizing low-pass filters. One method is software based. The user selects what kind of filter to implement (low-pass, high-pass, bandpass, etc), and then selects the appropriate attenuation in the pass-band ( $\alpha_p$ ), attenuation in the stopband ( $\alpha_s$ ), the passband frequency ( $\omega_p$ ), and the stopband frequency ( $\omega_s$ ). After these values are selected, the type of approximation for the filter (Butterworth, Chebyshev, inverse Chebyshev, etc), and the order “N,” of the filter are selected. Then finally, the program would calculate all of the results, including the pole locations and zeros, the transfer function, as well as the frequency and phase response. One such program that does this is Professor Wilamowski’s LADDER software, which has been used in this thesis. The other method to synthesize LC Ladder filters is the mathematical method. For passive ladder filters, multiplication, division, design equations, and the use of look up tables, is all one needs to design a filter. This chapter explains how to calculate transfer functions mathematically, and also introduces another unique method to synthesize ladder filters.

### 3.2 Properties of lossless ladders

Lossless ladders have the form of the circuit shown in Figure 3-1. Since the signal is supplied to the ladder by a resistive source, and since the load is resistive as well, the ladder is said to be doubly terminated (the source and the load resistance are both the same). Doubly terminated ladder filters have a lower sensitivity to component variations.



**Figure 3-1 Doubly Terminated lossless ladder:**

**a) General Representation; b) 5th order low-pass**

As can be seen in Figure 3-1b, this is a low-pass ladder filter. At low frequencies the inductors will be shorts and the capacitors will be open, so the filter simply reduces down to two parallel wires, as shown in Figure 3-1a. In general, if the ladder has  $n$  branches, with one inductor in each series branch, and one capacitor in each shunt branch, the transfer function is an  $n$ th-order polynomial  $D(s)$ , and the transfer function once again is of order  $n$ :

$$T(s) = \frac{V_{out}}{V_{in}} = \frac{K}{D(s)} = \frac{K}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} \quad (3.1)$$

At high frequencies, the inductors act like opens, and capacitors are short circuits, which means the signal is grounded, and this is exactly what was explained in Chapter 1. No matter the order of the filter, all low-pass ladder circuits reduce to Equations (3.2) at dc and at high frequencies.

$$T(0) = \frac{K}{a_0} = \frac{R_2}{R_1 + R_2} \quad \text{and} \quad T(s)|_{s \rightarrow \infty} = \frac{K}{s^n} = 0 \quad (3.2)$$

### 3.3 A synthesis strategy

For the circuit shown in Figure 3-1, we have identified the current from the source  $I_1$ , and its direction, we have also identified  $Z_{in}$  as the input impedance of the RLC circuit, which is made up of the lossless ladder terminated into  $R_2$ . Just like the poles of the circuit, the input impedance has a real and imaginary components (see equation 3.3)

$$Z_{in} = R_{in} + jX_{in} \quad (3.3)$$

and the current at the input is shown below (see equation 3.4)

$$I_1 = \frac{V_s}{R_1 + Z_{in}} \quad (3.4)$$

Since this is a lossless LC circuit, we can assume that the power at the input is equal to the power at the load ( $P_1 = P_2$ ). Then,

$$P_1 = \frac{|V_1(j\omega)|^2}{4R_1} \quad (3.5)$$

$$P_2 = R_{in} * |I_1(j\omega)|^2 = \frac{|V_2(j\omega)|^2}{R_2} \quad (3.6)$$

After substituting Equation 3.4 into Equation 3.6 and after proper manipulation, the desired transfer function for the ladder circuit is shown below (see Equation 3.7)

$$|T(s)|^2 = \frac{R_2 R_{in}}{|R_1 + Z_{in}|^2} \quad (3.7)$$

Once again  $P_1 = P_2$ , because there is no power loss in this lossless LC circuit, if we take the ratio of both power expressions, we have the proposed transfer function for LC filters (see Equation 3.8), which is defined with Equations (3.5 - 3.7).

$$|H(j\omega)|^2 = \frac{4R_1}{R_2} \frac{|V_2(j\omega)|^2}{|V_1(j\omega)|^2} = \frac{4R_1 R_{in}}{|R_1 + Z_{in}|^2} \leq 1 \quad (3.8)$$

If we substitute the impedance (Equation 3.3), into Equation 3.8,  $H(j\omega)$  takes on the form

$$|H(j\omega)|^2 = 1 - \frac{|R_1 - Z_{in}|^2}{|R_1 + Z_{in}|^2} = 1 - |\rho(j\omega)|^2 \quad (3.9)$$

Where  $\rho(s)$  is called the *auxiliary function*, which is also known as the reflection coefficient. Like in electromagnetic wave theory,  $\rho(s)$  is a measure of how well  $R_1$  and  $Z_{in}$  are matched. After some algebra it can be shown that Equation 3.9 can be solved for  $Z_{in}$ , in terms of  $\rho(s)$  and  $R_1$ .

$$Z_{in} = R_1 \frac{1 - \rho(s)}{1 + \rho(s)} \quad (3.10)$$

Since  $\rho$ , is determined from the prescribed transfer function  $H$ , we have reduced the problem to determining a lossless ladder circuit terminated into  $R_2$ , from a specified  $Z_{in}$ . It is known that the Butterworth response is:

$$|H(j\omega)|^2 = \frac{1}{1 + \omega^{2n}} \quad (3.11)$$

then from equation 3.1 we have

$$|H(j\omega)|^2 = \frac{K_H}{D(s)D(-s)} \quad (3.12)$$

Where  $K_H$  is equal to  $K$  from Equation 3.1, and  $D(s)$  is the known denominator from the Butterworth function. The polynomials  $D(s)$  for  $n = 1, 2$  and  $3$  are shown in Table 3-1

Order	$ D(j\omega) ^2$	$D(s)$
1	$1 + \omega^2$	$s + 1$
2	$1 + \omega^4$	$s^2 + \sqrt{2}s + 1$
3	$1 + \omega^6$	$s^3 + 2s^2 + 2s + 1$

**Table 3-1: Polynomials  $D(s)$  for Butterworth Filter**

Since  $H(s)$  is known (Equation 3.8), we can rewrite the auxiliary function using Equation 3.9 as

$$|\rho(s)|^2 = 1 - \frac{1}{1 + \omega^{2n}} = \frac{\omega^{2n}}{1 + \omega^{2n}} = \frac{s^n}{D(s)} \quad (3.13)$$

Finally, if we substitute Equation 3.11, into Equation 3.9 we obtain Equation 3.14, which allows us to use this unique synthesis to calculate the elements for a ladder filter by using polynomial division.

$$Z_{in} = R_1 \left[ \frac{1 - \rho(s)}{1 + \rho(s)} \right]^{\pm 1} = R_1 \left[ \frac{D(s) - s^n}{D(s) + s^n} \right]^{\pm 1} \quad (3.14)$$

### 3.3.1 Ladder Synthesis Example

If we let  $n=3$ , and by setting  $R_1 = 1$ , and by using Equation 3.14 and Table 3-1, we find that the auxiliary function is

$$\rho(s) = \frac{s^3}{s^3 + 2s^2 + 2s + 1} = \quad (3.15)$$

and then we find that  $Z_{in}$

$$Z_{in} = \frac{2s^2 + 2s + 1}{2s^3 + 2s^2 + 2s + 1} = \quad (3.16)$$

In order to determine the ladder circuit,  $Z_{in}$  needs to be expanded. To expand the circuit, this is done by dividing the polynomial of higher degree by the polynomial of the lower degree. This is done for one step, then you have to invert the polynomial and divide again, then you have to invert the polynomial again, and once again divide again. This process is repeated over and over until there is no remainder. After each division, the first term of each quotient is an element value for the ladder circuit. An example of the polynomial division and the element extraction is shown below.

Division	Quotient	Element
$2s^2 + 2s + 1 \sqrt{s^3 + 2s^2 + 2s + 1} = s + 1$	$s$	$sL_1$
$s + 1 \sqrt{2s^2 + 2s + 1} = 1$	$2s$	$sC_2$
$1 \sqrt{s + 1} = 1$	$s$	$sL_3$

$1\sqrt{1} = 0$	1	$R_2$
-----------------	---	-------

Table 3-2: Ladder Synthesis Division and Element Extraction

or

$$Z_{in} = \frac{1}{s + \frac{1}{2s + \frac{1}{s + \frac{1}{1}}}} = \frac{1}{sL_1 + \frac{1}{sC_2 + \frac{1}{sL_3 + \frac{1}{G_2}}} \quad (3.17)$$

Now that the polynomial division is complete, the circuit realization is extracted from quotient, and the corresponding circuit is shown in Figure 3-2. The normalized element values can then be found from ladder tables. Ladder circuits can be developed this way for any  $n$ th order low pass filter. In the next section, the more traditional mathematical approach to designing  $n$ th order filters will be explained.

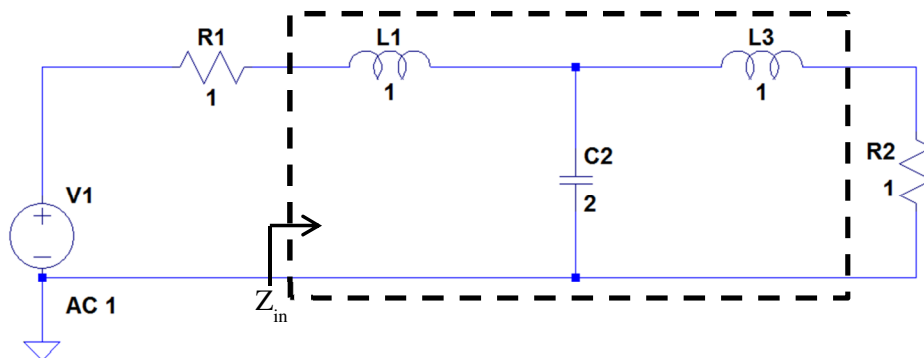


Figure 3-2: Ladder Synthesis Example

### 3.4 Calculate the Transfer Function of Ladder Circuit

To calculate the transfer function of a ladder filter, consider the 5<sup>th</sup> order low pass filter shown in Figure 3-3. In order to analyze this circuit, I will introduce the notation  $Y_j$  (admittance), for the parallel

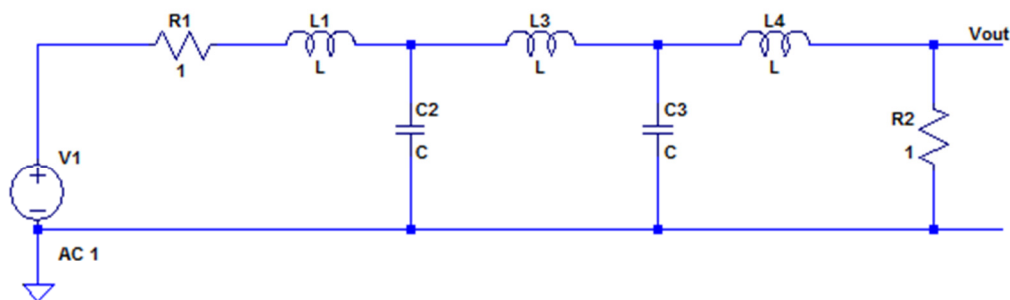


Figure 3-3: 5<sup>th</sup> order Ideal Chebyshev low-pass filter

branches, and  $Z_i$  (impedance), for the series branches as shown in Figure 3-4 and in Table 3-3. Then using Kirchoff's voltage laws, KCL and KVL, and also Ohm's law, I can calculate the transfer function by starting at  $V_{out}$ , and working back to  $V_s$ .

$Z1 = sL1$
$Y2 = sC2$
$Z3 = sL3$
$Y4 = sC4$
$Z5 = sL5$

Table 3-3: Element Values for series and shunt ladder

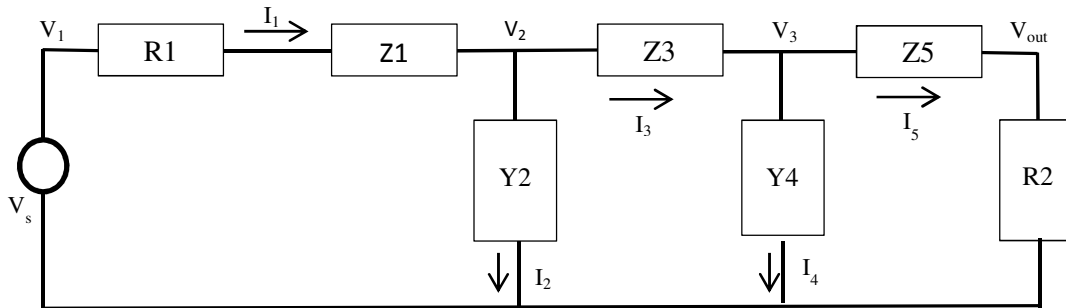


Figure 3-4: 5<sup>th</sup> order ladder with series impedances and shunt admittances

Ohm' s Law:	$I_6 = I_5 = V_{out} * R_2$	(3.18)
KVL	$V_3 = V_{out} * (1 + s * L_5 * R_2)$	(3.19)
KCL:	$I_3 = I_4 + I_5 = sC_4 * V_3 + V_{out} * R_2$ $I_3 = V_{out} [s * C_4 (1 + sL_5 * R_2) + R_2]$	(3.20)
KVL	$V_2 = I_3 * sL_3 + V_3$ $V_2 = V_{out} [s * C_4 (1 + sL_5 * R_2) + R_2] * sL_3 + V_{out} * (1 + s * L_5 * R_2)$	(3.21)

	$V_2 = V_{out}[sL_3[s^*C_4(1+sL_5^*R_2) + R_2] + (1 + s^*L_5^*R_2)]$	
KCL:	$I_1 = I_2 + I_3$ $I_1 = sC_2^*V_2 + V_{out}[s^*C_4(1+sL_5^*R_2) + R_2]$ $I_1 = V_{out}^*\{sC_2^*[sL_3[s^*C_4(1+sL_5^*R_2) + R_2] + (1 + s^*L_5^*R_2)] + [s^*C_4(1+sL_5^*R_2) + R_2]\}$	(3.22)
KVL	$V_s = I_1R_1 + sL_1I_1 + V_2$	(3.23)

Table 3-4: 5<sup>th</sup> order transfer function

After expansion and simplification of Equation (3.24), the coefficients of the 5<sup>th</sup> order transfer function are shown in Table 3-5.

S <sup>5</sup>	
L1*C2*L3*C4*L5*R2	
S <sup>4</sup>	
L1*C2*L3*C4+R1*C2*L3*C4*L5*R2	
S <sup>3</sup>	
L3*C4*L5*R2+L1*C4*L5*R2+L1*C2*L5*R2+L1*C2*L3*R2+R1*C2*L3*C4	
S <sup>2</sup>	
L3*C4+L1*C4+L1*C2+R1*C4*L5*R2+R1*C2*L5*R2+R1*C2*L3*R2	
S <sup>1</sup>	
L5*R2+L3*R2+L1*R2+R1*C4+R1*C2	
S <sup>0</sup>	
1+R1*R2	

Table 3-5: Coefficients of 5<sup>th</sup> Chebyshev Transfer Function



If we consider a 5<sup>th</sup> order low pass filter with the following characteristics  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec and designed with the following elements (see Table 3-6 ), and then plugging these values into the coefficients of the transfer function shown in Table 3-5, the transfer function of the circuit can be calculated.

Element	Value
L1	3.481829
C2	0.761919
L3	4.537546
C4	0.761919
L5	3.481829
R1	1
R2	1

**Table 3-6: Element Values for 5th Order Low-pass Filter**

Transfer function is shown in Equation 3.25:

$$TF = \frac{1}{31.93s^5 + 18.42s^4 + 45.18s^3 + 17.53s^2 + 13.02s + 2} \quad (3.25)$$

The frequency response is show in Figure 3-55, this plot was generated in MATLAB by plotting the transfer function (Equation 3.8), and as you can see from the plot, there are ripples in the passband.

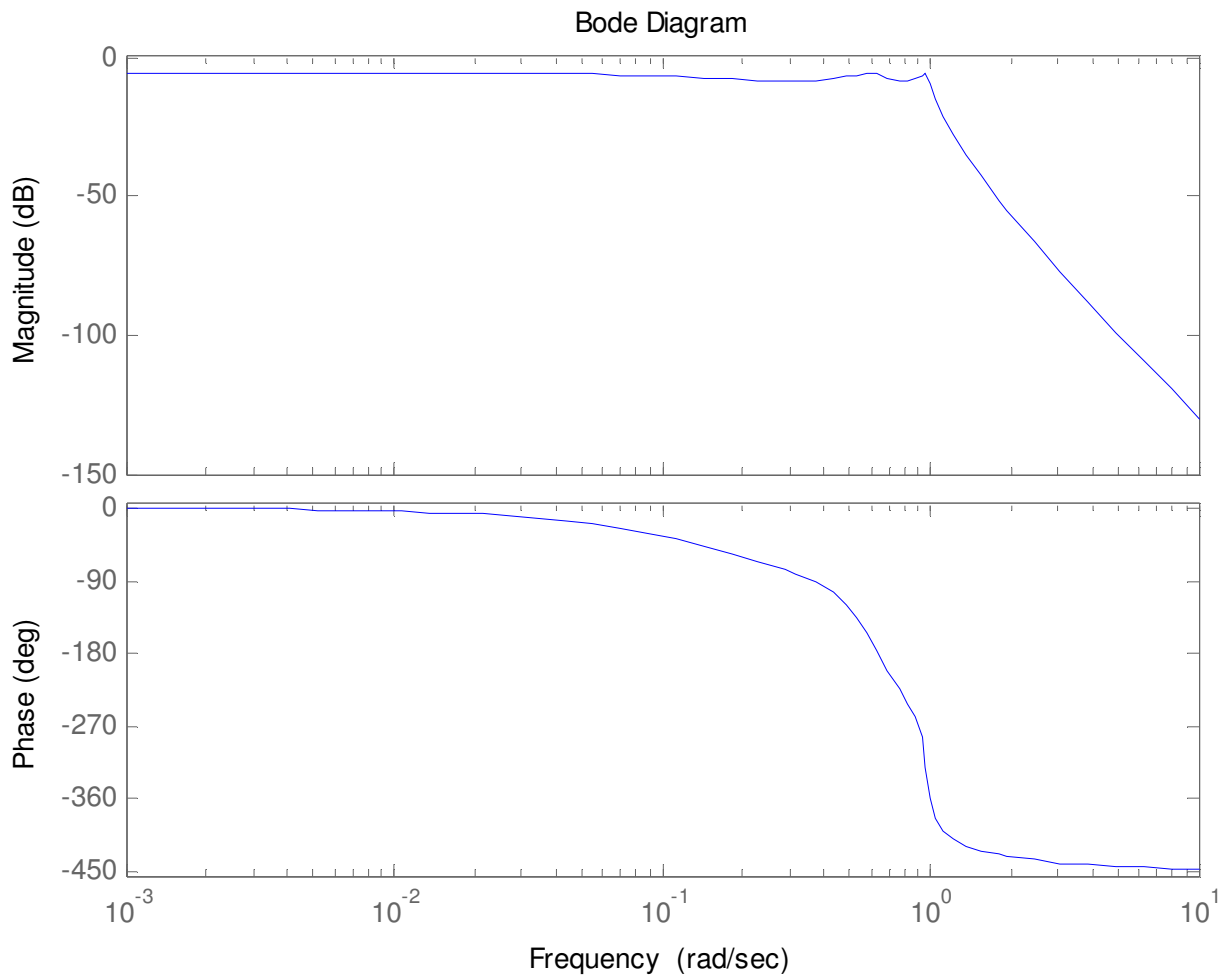


Figure 3-5: 5<sup>th</sup> order Chebyshev low-pass filter

### 3.5 Frequency Transformation from low-pass to other types of filters

The low-pass filter is a prototype filter for other types of filters. There is no need to calculate the transfer function of the desired filter, it can all be done from the low-pass filter. One method is to use the Foster Reactance Function. To use the Foster Function, the user substitutes the appropriate Foster Function into the transfer function of the low-pass filter, this intern transforms the filter into a high-pass, band-pass or any other type of filter. The Foster Function is mostly used in active filters. Another approach to filter transformation is element substitution. In this approach, individual elements of the original low-pass filter are replaced with sub-circuits (see Figure 3-6).

After the sub circuits have been inserted, the new element values are then re-scaled to the appropriate values by the equations shown in Figure 3-6 where:




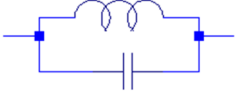


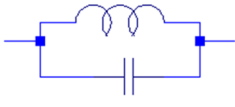

$L = \text{Inductor Value}$

$C = \text{Capacitance Value}$

$\omega_c = \text{Center Frequency}$

$\omega_0 = \text{Cut-off Frequency}$

$B = \text{Bandwidth of Band-pass filter}$

	Low-pass	Highpass	Bandpass	Bandstop
Z	$\frac{L}{\omega_c}$ 	$\frac{1}{L\omega_c}$ 	$\frac{L}{B} \quad \frac{B}{L\omega_0^2}$ 	$BL/\omega_0^2$  $1/BL$
Y	$\frac{C}{\omega_c}$ 	$\frac{1}{C\omega_c}$ 	$B/C\omega_0^2$  $C/B$	$\frac{1}{BC} \quad \frac{BC}{\omega_0^2}$ 

**Figure 3-6: Filter transformation from low-pass to other types of filters**

The two main reasons for scaling are to produce more practical/ feasible values for the elements e.g. load resistance (magnitude scaling), and/or to move the filter to the appropriate 3-dB cutoff point (frequency scaling). This form of frequency transformation from low-pass filters is quick and efficient, and is mostly used in passive designs

# 4 Synthesis of lossy LC Ladder Filters

## 4.1 Introduction

For almost all prototype low-pass filters, the values of the elements have already been derived and put into tables. While this type of design from tables works fine, it represents an ideal response, which is insufficient for the real world or for practical applications. In practical applications, the inductors and capacitors will each introduce losses into the circuit. These losses will affect the frequency response of the filter significantly, and if the frequency response changes, this can greatly impact the performance of the whole system. This next section describes how to design filters with lossy components, and how to optimize the filter to achieve ideal like results.

## 4.2 Background

The basic structure of the lossless ladder is shown in Figure 4-1, where the parallel branches are

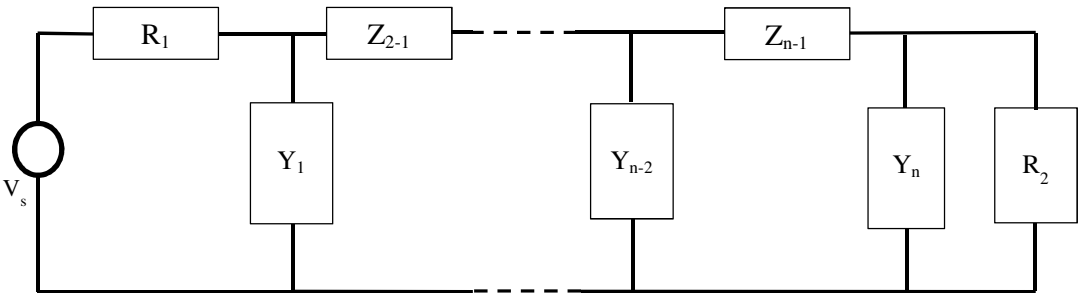
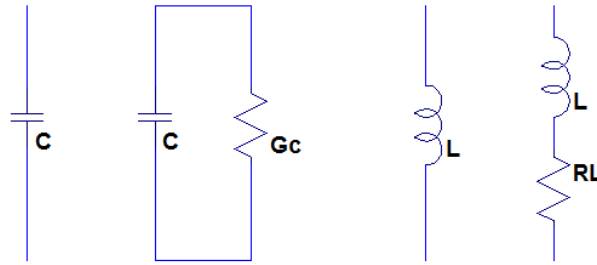


Figure 4-1: Basic Structure of Ladder Filter

the admittances ( $Y$ ), and the series branches are the impedances ( $Z$ ), which are purely imaginary. However, this time the filter includes the losses. In order to include the losses in the filter, all of the inductors and capacitors in the circuit must be replaced with their lossy counterpart, see Figure 4-2.



**Figure 4-2: Models of Real Capacitors and Real Inductors**

As can be seen from Fig. 4-2, each lossy inductor can be represented by an ideal inductor  $L$ , in series with  $R_L$  (which is measured in ohms), and each lossy capacitor can be represented by an ideal capacitor  $C$ , in parallel with a conductance  $G_c$  (which is measured in siemens). The resistance  $R_L$ , and the conductance  $G_c$ , are assumed to be constants, i.e. they do not depend on the frequency  $\omega$ . This is equivalent to saying that the quality factor ( $Q$ ) of each inductor and capacitor varies linearly with frequency. The complex impedance  $Z(s)$  of the lossy inductor is shown in Equation 4.1, and

$$Z(s) = sL + r \quad (4.1)$$

the complex admittance  $Y(s)$  of the lossy capacitors is shown in Equation 3.2

$$Y(s) = sC + g \quad (4.2)$$

In ideal filters, the lower the  $Q$ , the less sharper the response, and the higher the  $Q$ , the more idealistic the response. The same thing goes for the lossy filter, the higher the losses of the reactance elements used in the design, the greater the difference between the lossy frequency response of the filter, and the ideal frequency response of the filter.

The difference in the ideal response and lossy frequency response of the circuit is due to the reactance of the elements, which internally change the location of the poles on the complex plane (see Figure 4-3). In order to gain ideal like characteristics, one must minimize the distance between the poles of the

ideal filter and the lossy filter, this is done optimizing the reactance elements of the lossy filter, so that they match the ideal filter. The different methods of optimization will be explained in the following chapters.

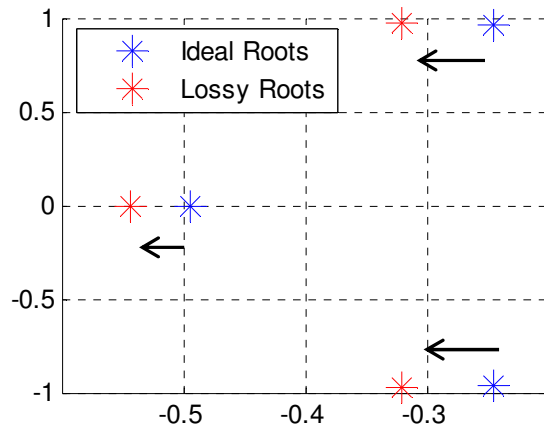


Figure 4-3: Movement of Poles due to lossy elements

### 4.3 Procedure of designing Lossy Ladder Filters

This chapter describes how to design passive low-pass filter ladder filter, with lossy elements. Results are included for the Butterworth, Chebyshev, and Bandpass ladder filters. Just like in the ideal case, one can use software, or tables to design the filter, however, for each ideal element, it must be replaced with its lossy component type (see Figure 4-2). The three main steps in designing a lossy ladder filter are listed below.

1. Determine the transfer function for the ideal and lossy ladder filter
2. Calculate the error function
3. Minimize the error function using optimization techniques

Determining the transfer function mathematically for the filter can be done as explain in Chapter 2.

However, for higher order  $N$  filters, and for band-pass filters, determining the transfer function

mathematically can be rather complex. In this Chapter, the method to calculate the transfer function by

hand, and in MATLAB, both in the symbolic form and in the coefficient form will be explained. After the

transfer function is calculated, the next step is to minimize the error between the ideal filter response and

the lossy filter response. To minimize the error between both responses, the least squares method is used.

Finally the error function is minimized using different optimization techniques.

#### 4.3.1 Calculating the transfer function of a lossy ladder filter

Consider a 3<sup>rd</sup> order lossy low-pass filter show in Figure 4-4, where  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec, designed using ideal reactance elements,  $L_1=2.023593$ ,  $C_2=0.994102$ ,  $L_3=2.023593$ ,  $R_{in}=R_{out}=1$  and with lossy elements  $R_1 = G_2 = R_3 = 0.1$ .

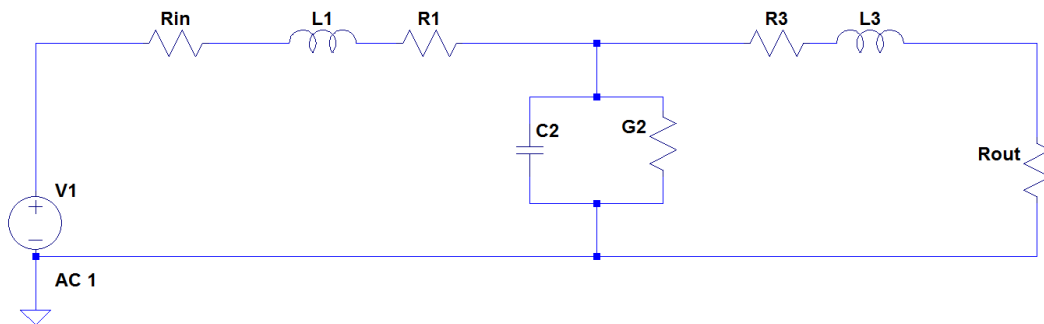


Figure 4-4: 3<sup>rd</sup> order lossy ladder filter

The impedance and admittance equations are shown in Table 4-1

$Z_1 = sL_1 + R_1$
$Y_2 = sC_2 + G_2$
$Z_3 = sL_3 + R_3$
$R_{in} = R_{out} = 1$

Table 4-1: 3<sup>rd</sup> Order Lossy Element Values

The mathematically calculated lossy transfer function is shown below:

Ohm's Law:	$I_4 = I_3 + 0 = V_{out} * R_{out}$	(4.3)
KVL:	$V_2 = V_{out} + I_3 Z_3$ $V_2 = V_{out} + V_{out} * R_{out} * Z_3$ $V_2 = V_{out} (1 + Z_3 * R_{out})$	(4.4)

KCL	$I_1 = I_2 + I_3$ $I_1 = Y_2 * [V_{out} * (1 + Z_3 * R_{out})] + V_{out} * R_{out}$ $I_1 = V_{out} [Y_2 (1 + Z_3 * R_{out}) + R_{out}]$	(4.5)
KVL	$V_{in} = I_1 R_1 + Z_1 I_1 + V_2$ $V_{in} = (V_{out} [Y_2 (1 + Z_3 * R_{out}) + R_{out}]) R_1 +$ $Z_1 * V_{out} [Y_2 (1 + Z_3 * R_{out}) + R_{out}] + V_{out} (1 + Z_3 * R_{out})$	(4.6)

**Table 4-2: 3<sup>rd</sup> Order lossy transfer function**

After simplification, and substituting the equations from Table 4-1 into Equation 4.6, the final transfer function is shown in Table 4-3. Equation 4.7 shows transfer function, with the coefficients a,b,c & d, representing the elements for each power “s”, in Table 4-1.

<b>S<sup>3</sup></b>
$a = L1 * C2 * L3$
<b>S<sup>2</sup></b>
$b = R1 * C2 * L3 + L1 * G2 * L3 + L1 * C2 + L1 * C2 * R3 + C2 * L3$
<b>S<sup>1</sup></b>
$c = L3 + R1 * G2 * L3 + R1 * C2 + R1 * C2 * R3 + L1 + L1 * G2 + L1 * G2 * R3 + G2 * L3 + C2 + C2 * R3$
<b>S<sup>0</sup></b>
$d = 2 + R3 + R1 + R1 * G2 + R1 * G2 * R3 + G2 + G2 * R3$

**Table 4-3: 3<sup>rd</sup> Order Lossy Transfer Function - Mathematical**

$$TF = \frac{1}{as^3 + bs^2 + cs + d} \quad (4.7)$$

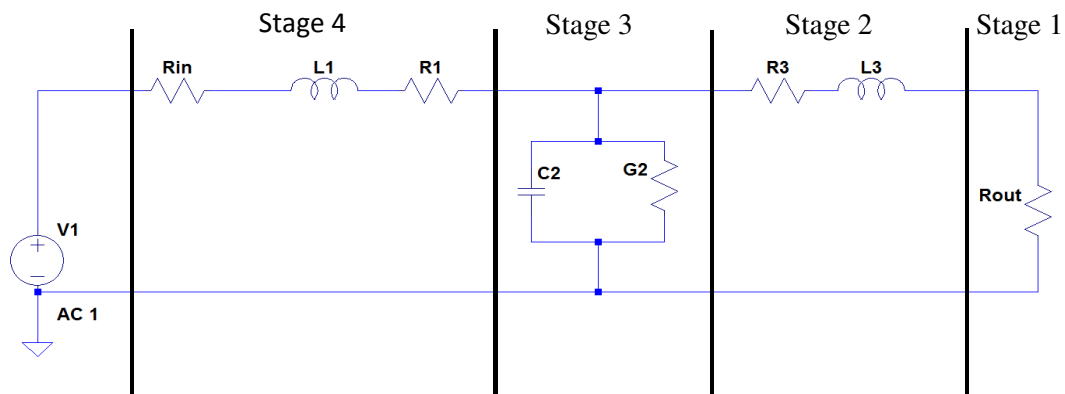


The final transfer function with all values substituted into a, b c, and d is show in in Equation 4.8

$$TF = \frac{1}{4.062s^3 + 4.019s^2 + 5.037s + 2} \quad (4.8)$$

Calculating the lossy transfer function mathematically like this is straight forward for small order N filters, but for higher order filters, calculating the transfer function mathematically can be extremely tedious, especially when it comes to band-pass filters. Band-pass filters introduce a complex numerator into the transfer function, which make calculating the transfer function a lot more intense. Another way to calculate the transfer function is to use MATLAB, which is explained below.

Every ladder filter has several reoccurring stages as shown in Figure 4-5. If we assume the output of Stage 1 is 1V ( $V_o = 1V$ ), and then if we work backwards through each stage, we can develop the transfer function for any type of filter, whether it be lossy or ideal. However upon completion, we must remember that the final transfer function will be the inverse of  $V_{in}$ .



**Figure 4-5: Stages of 3<sup>rd</sup> order low-pass lossy filter**

Consider the 3rd order low-pass lossy ladder filter shown in Figure 4-5. For the lossy ladder filter shown above, we regard the voltage or current of each stage, as the generic variable State ( $U$ ), as well as the resistance of each stage as a generic resistance, Coefficients ( $C_o$ ). Where Coefficients ( $C_o$ ) is going to be the impedance of a series branch, as well as the admittance of a shunt branch. Now by definition, each stage has

one State (U) variable, and one Coefficients (Co) variable, and every stage of the transfer function can be calculated by adding up all of the previous stages, as shown in Equation 4.9

$$U_n = U_{n-2} + U_{n-1} * C_{0n-1} \quad (4.9)$$

This method is good for calculating the transfer function numerically, and is also very generic. No matter the order of the filter, this method works to calculate the transfer function. However, it can be shown that MATLAB can also calculate the transfer function symbolically. The symbolic version follows the same method as described above, except you have to modify the MATLAB code to handle all of the functions symbolically. Once this is done, the coefficients of your transfer function are simply the elements as shown in the previous transfer function calculations (see Table 4-3 for example).

If we consider the same 3<sup>rd</sup> order low pass filter with  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec, designed using ideal reactance elements,  $L1=2.023593$ ,  $C2=0.994102$ ,  $L3=2.023593$ ,  $R_{in}=R_{out}=1$  and with lossy elements  $R1 = G2 = R3 = 0.1$  The ideal transfer function calculated in MATLAB is shown in Table 4-4, and the lossy transfer function calculated in MATLAB is shown in Table 4-5.

<b>S<sup>3</sup></b>
L1*C2*L3
<b>S<sup>2</sup></b>
L1*C2+C2*L3
<b>S<sup>1</sup></b>
L3+L1+C2
<b>S<sup>0</sup></b>
2

**Table 4-4: 3<sup>rd</sup> Order ideal low-pass filter with symbolic coefficients – Calculated in MATLAB**

$S^3$
$L1*C2*L3$
$S^2$
$R1*C2*L3+L1*G2*L3+L1*C2+L1*C2*R3+C2*L3$
$S^1$
$L3+R1*G2*L3+R1*C2+R1*C2*R3+L1+L1*G2+L1*G2*R3+G2*L3+C2+C2*R3$
$S^0$
$2+R3+R1+R1*G2+R1*G2*R3+G2+G2*R3$

**Table 4-5: 3<sup>rd</sup> Order lossy low-pass filter with Symbolic coefficients – Calculated in MATLAB**

After plugging in the right values into the correct parameter, it can be shown that both the symbolically calculated transfer function and the numerically calculated transfer function in MATLAB both match.

MATLAB Symbolically calculated transfer function (Equation 4.10) – Ideal

$$TF = \frac{1}{4.062s^3 + 4.019s^2 + 5.037s + 2} \quad (4.10)$$

MATLAB Numerically calculated transfer function (Equation 4.11) – Ideal

$$TF = \frac{1}{4.070s^3 + 4.023s^2 + 5.041s + 2.000} \quad (4.11)$$

MATLAB symbolically calculated transfer function (Equation 4.12) – Lossy

$$TF = \frac{1}{4.062s^3 + 4.829s^2 + 5.69s + 2.321} \quad (4.12)$$

$$TF = \frac{1}{4.070s^3 + 4.835s^2 + 5.695s + 2.321} \quad (4.13)$$

The MATLAB symbolic method and numerical method are both proven to calculate the transfer function correctly for any order (N), of a low pass filter.

### 4.3.2 Calculating the Error Function

The basic theory behind the error function is to reduce the distance between the poles of the ideal filter and the lossy filter. To calculate the error function, the Error Sum of Squares (SSE) is used, see Equation 4.14.

$$SSE = \sum_{i=1}^n (a0 - b0)^2 + (a1 - b1)^2 + (an - bn)^2 \quad (4.14)$$

There are two different ways to calculate the error, one way is to calculate the errors between the poles of the lossy filter and the ideal filter, and the other way is to calculate the error between the coefficients of the lossy filter and the ideal filter. Each technique will be presented in the following chapters.

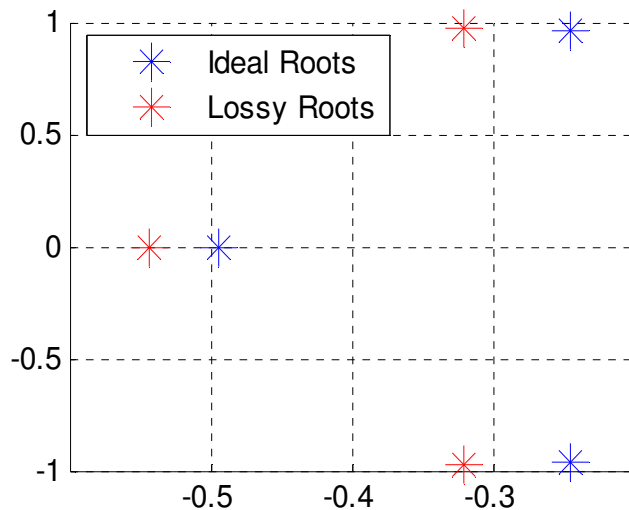


Figure 4-6: Pole locations for 3<sup>rd</sup> order ideal and lossy low-pass filter

For the 3<sup>rd</sup> order filter mentioned at the beginning of this chapter, the roots are shown in Table 4-6 and the pole locations are shown in Figure 4-6. To calculate the error, we substitute the ideal roots and lossy roots into Equation 4.14.

Ideal Roots	Lossy Roots
-0.2537 + 0.9513i	-0.3229 + 0.9583i
-0.2537 - 0.9513i	-0.3229 - 0.9583i
-0.4974	-0.5461

**Table 4-6: Ideal Roots and Lossy roots for 3<sup>rd</sup> order low-pass filter**

For example, if we let  $a_0 = -0.1437 + 0.8513i$ ,  $b_0 = -0.2229 + 0.8583i$ ,  $a_1 = -0.1437 - 0.8513i$ ,  $b_2 = -0.2229 - 0.8583i$ ,  $a_3 = -0.2874$ , and  $b_3 = -0.3161$ , and substitute these values in Equation 4.14, the error is calculated between the lossy roots and the ideal roots. This SSE value is then passed on into the Nelder-Mead optimization function which is a technique for minimizing the distance between the ideal roots and the lossy roots. This optimization routine will be explained in the next section.

Another way to calculate the error is to find the difference between the coefficients of the lossy filter and the ideal filter. Once again, Equation 4.14 is used to calculate the error, however this time, the coefficients of the transfer function are substituted into  $a_0$ ,  $b_1$ , etc.

For example, from Equation 4.11, the coefficients for the ideal filter would be  $a_0 = 4.070$ ,  $a_1 = 4.023$ ,  $a_3 = 5.041$ , and  $a_4 = 2.000$ , and the coefficients of the lossy filter would come from Equation 4.13, where  $b_0 = 4.070$ ,  $b_1 = 4.835$ ,  $b_3 = 5.695$ ,  $b_4 = 2.321$ . Once again, this error value is then calculated and passed onto the optimization routine.

### 4.3.3 Optimization Routine - Nelder Mead Method

The first optimization routine used to optimize the lossy ladder circuits was the Nelder-Mead Method. The Nelder-Mead algorithm is designed to solve the classical unconstrained optimization problem of minimizing a given nonlinear function. This is a fast algorithm which searches for local minimum without

the use of derivatives. Using the simplex method, the Nelder-Mead Method finds the minimum point of error for the lossy ladder filter. The term *simplex* is a generalized triangle of N dimensions, which is formed by (N+1) vertices (where N is the number of variables of a function). The worst vertex (which is the largest), is rejected and replaced with a new vertex, then a new triangle is created and the search continues on. This process continues on and on and creates a sequence of triangles, and at the same time, the function values vertices are getting smaller and smaller until the desired error value is found. During the search, the algorithm performs four routines: Reflection, Expansion, Contraction, and Shrinkage, which is explained below.

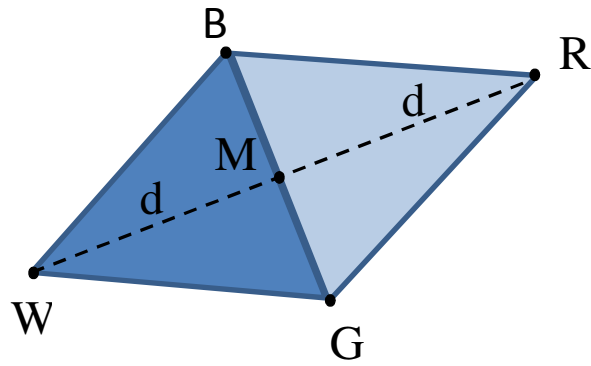
The initial triangle is formed by a initial set of points which represent solutions estimates. If  $f(x,y)$  is the function to be minimized, to start we are given three vertices of a triangle:  $V_k = (X_k, Y_k)$ , where  $k = 1, 2, 3$ . The function  $f(x,y)$  is then evaluated at each of the three points:  $z_k = f(X_k, Y_k)$  for  $k = 1, 2, 3$ . The subscripts are then reordered so that  $z_1 \leq z_2 \leq z_3$ . Equation 4.14 shows the notation where

$$B = (x_1, y_1) \qquad G = (x_2, y_2) \qquad W = (x_3, y_3) \qquad (4.14)$$

**B** is the best vertex, **G** is a good vertex (next to best) and **W** is the worst vertex. The first step in the algorithm to find the midpoint of the **B** & **G** line which is done by averaging the coordinates as shown in Equation 4.15

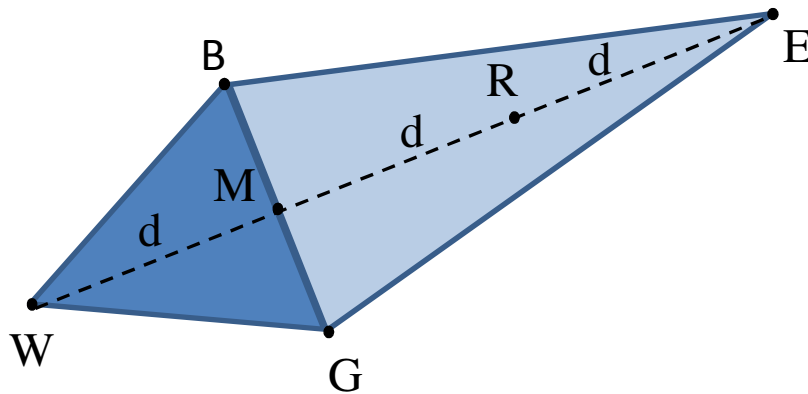
$$M = \frac{B + G}{2} = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \qquad (4.15)$$

Reflection: The function decreases as we move along the side of the triangle from **W** to **B**, and it also decreases as we move from **W** to **G**. The test point **R** is obtained by reflecting the triangle through the side **BG**. **R** is determined by multiplying the midpoint **M** of the side **BG**, then drawing the line segment from **W** to **M** and calling this length  $d$ . Finally, extend the distance  $d$  through **M**, to locate **R** (see Figure 4-7).



**Figure 4-7: The Triangle  $ABC$ , the midpoint  $M$  and the Reflected point  $R$**

Expansion: If the function value  $R$  is smaller than the function value at  $W$ , then we have moved in the correct direction towards the minimum. However, if the minimum is farther than the point  $R$ , the line segment through  $M$  and  $R$  is extended to form the new point  $E$  (see Figure 4-8). If the function value at  $E$  is less than the function value of  $R$ , a better vertex than  $R$  has been found.



**Figure 4-8: The Triangle  $BGW$  and point  $R$  with extended point  $E$**

Contraction: If the function values at  $R$  and  $W$  are the same, another point must be tested. Consider the two midpoints  $C_1$  and  $C_2$  of the line segments  $WM$  and  $MR$  (see Figure 4-9). The point with the smaller function value is called  $C$ , and the new simplex (triangle)  $BGC$  is created if the function value at  $C$  is smaller than the function value at  $W$ .

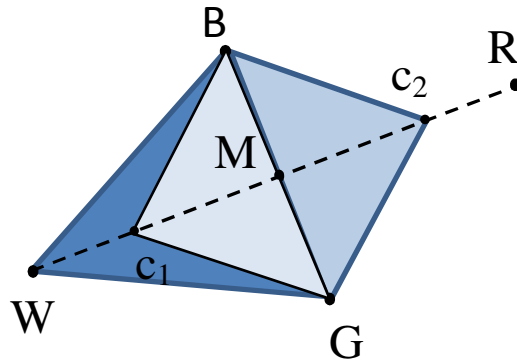


Figure 4-9: The Contraction point  $C_1$  or  $C_2$

Shrinkage: If the function values at  $C$  is not less than the function values at  $W$ , the points  $G$  and  $W$  will be shrunk towards  $B$  (see Figure 4-9).  $G$  will then move toward  $M$ , and  $W$  is replaced with  $S$ , which are midpoints of the line segments joining  $B$  with  $W$ .

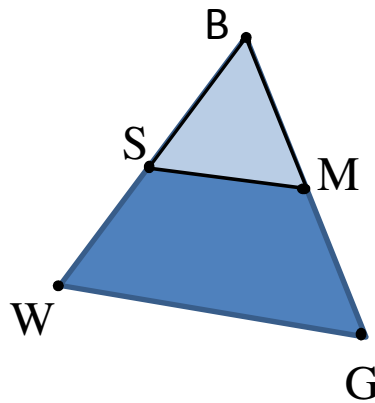


Figure 4-10: Shrinkage toward the Triangle B

The algorithm will continue to repeat until the desired minimum is found. This Nelder-Mead algorithm is popular in the fields of chemistry, medicine and engineering and gives significant improvements in the first few iterations and provides suitable results. However, for larger order filters and for the band-pass filter, the algorithm does not have the capability to optimize the filters completely. Simulation results using the Nelder-Mead Algorithm are in the next chapter.



## 5 Nelder-Mead Simulation Results – Error Calculated From Roots

In this section, simulation results are calculated by trying to reduce the distance between the poles of the ideal filter and the lossy filter. As explained earlier in the chapter, this error calculation is calculated from the roots of each transfer function, and then this error value is passed on into the optimization routine.

### 5.1 3<sup>rd</sup> Order Chebyshev Low-pass Filter

For our first example let us take into consideration a 3<sup>rd</sup> order Chebyshev low-pass filter with the following specifications given in Table 5-1

Element	Values
$\alpha_p$	1dB
$\alpha_s$	30dB
$\omega_p$	1rad/sec
$\omega_s$	2rad/sec

Table 5-1: 3<sup>rd</sup> Order Chebyshev low-pass filter specifications

Using the LADDER software, which uses the synthesis algorithm discussed in Chapter 3.1, the element values are shown in Table 5-2 for the ideal filter, and the ladder circuit for the ideal circuit is shown in Figure 5-1.

Element	Values
$C_1$	2.0236
$L_2$	0.9941
$C_3$	2.0236
$R_1 = R_2$	1

Table 5-2: 3<sup>rd</sup> Order Chebyshev low-pass filter element values

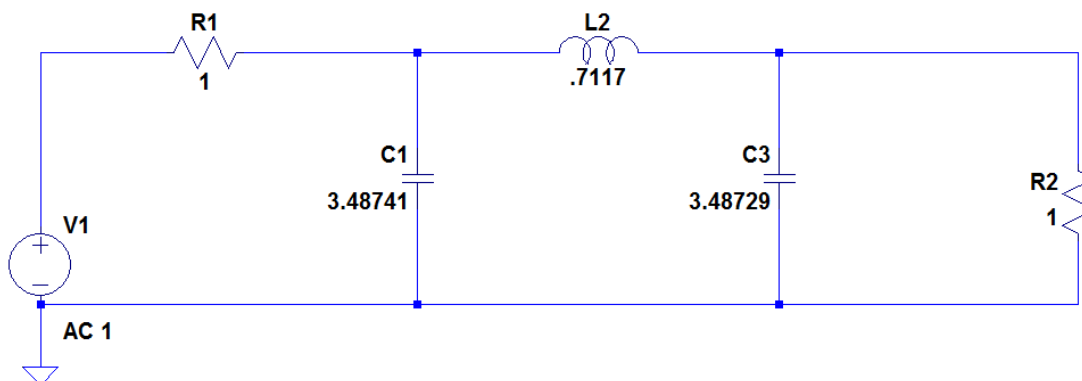


Figure 5-1: 3<sup>rd</sup> order ideal Chebyshev low-pass filter

In order to make the ladder filter lossy, all of the inductors and capacitors in the circuit must be replaced with their lossy filter counter parts, as explained earlier in the chapter. If we let  $GC_1 = 0.1$ ,  $RL_2 = 0.1$  and  $GC_3 = 0.1$ , the lossy filter is shown in Figure 5-2. Using the Nelder-Mead Algorithm as the optimization routine, the new element values for the lossy filter are shown in Table 5-3, and the simulations results are shown in the figures below.

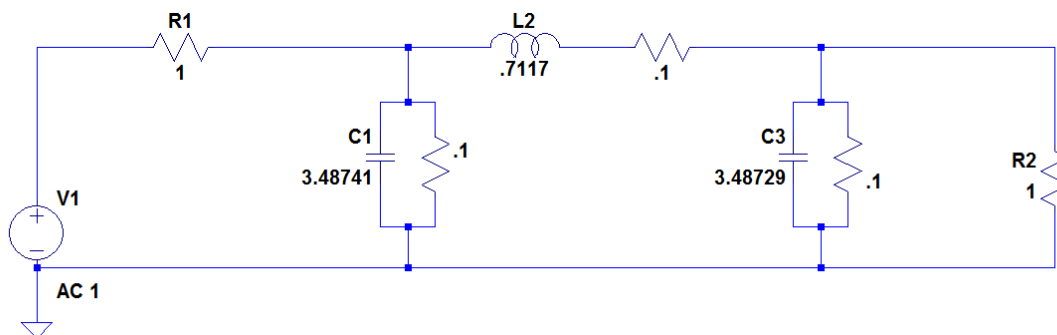


Figure 5-2: 3<sup>rd</sup> order lossy Chebyshev low-pass filter

Element	Values
$C_1$	4.0653
$L_2$	0.6163
$C_3$	4.0653
$R_1 = R_2$	1

Table 5-3: Optimized values for 3<sup>rd</sup> order Chebyshev

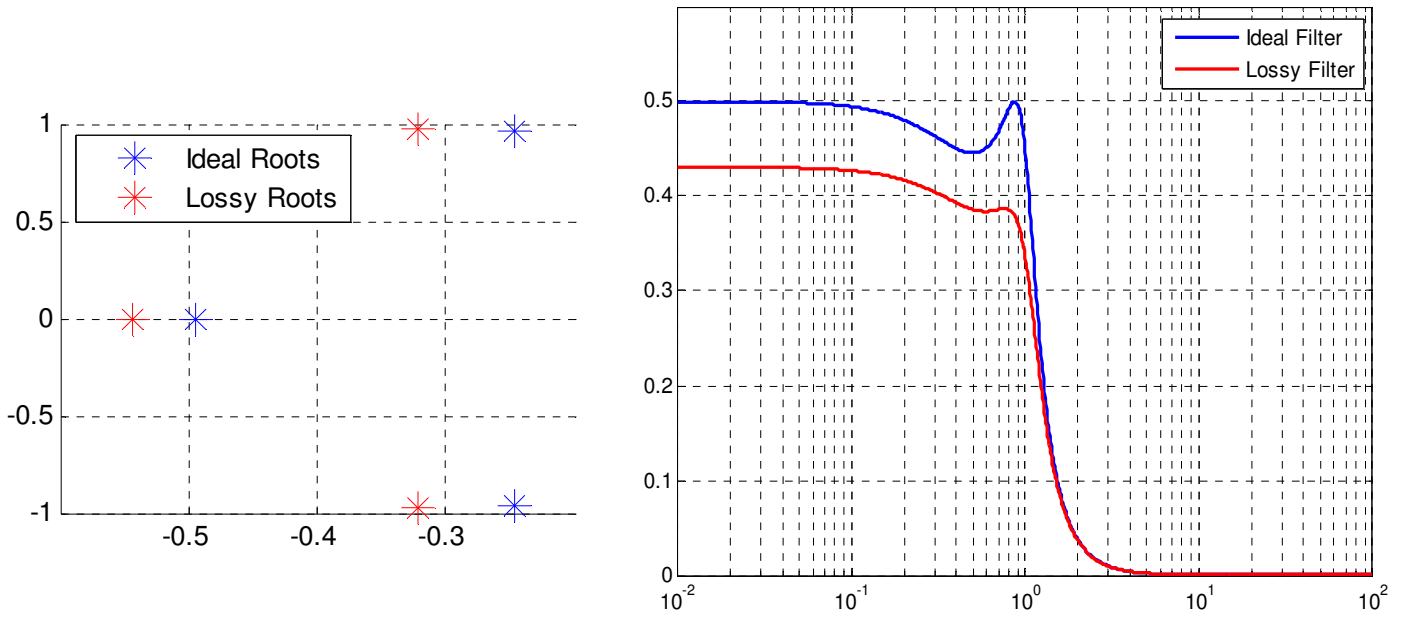


Figure 5-3: 3rd order Chebyshev Pole plot and Frequency Response – Before Optimization

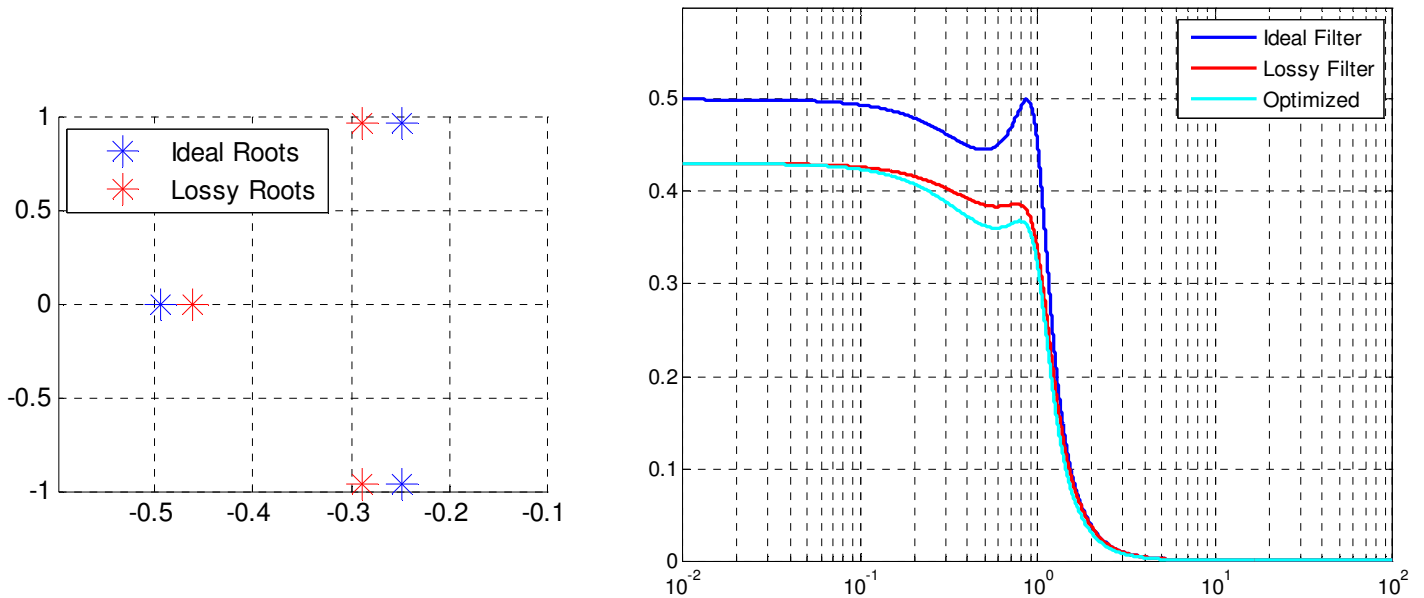


Figure 5-4: 3<sup>rd</sup> order Chebyshev Pole plot and Frequency Response – After Optimization

As you can see from the plots above, after the optimization routine, the roots of the lossy and ideal filter are a great deal closer than before optimization. However, they are not matched 100%. The frequency response plots have also been optimized. Apart from the bigger attenuation in the optimized response, the shape of the filter is different than the lossy response, which is of most importance in this routine. The

optimized response has a smoother and more uniform response than the lossy response, and this optimized response resembles the ideal response better than the lossy response, except for added attenuation. Since this optimization routine did not optimize the gain of the filter, the optimized response is acceptable with this optimization scheme.

## 5.2 5<sup>th</sup> Order Butterworth Low-pass filter

For our next example let us take into consideration a 5th order Butterworth low-pass filter with the following specifications given in Table 5-4.

Element	Values
$\alpha_p$	1dB
$\alpha_s$	30dB
$\omega_p$	1rad/sec
$\omega_s$	2rad/sec

**Table 5-4: 5<sup>th</sup> Order Butterworth low-pass filter specifications**

Using the LADDER software, which uses the synthesis algorithm discussed in Section 3.1, the element values are shown in Table 5-5 for the ideal filter, and the ladder circuit for the ideal circuit shown in Figure 5-5

Element	Values
$C_1$	0.617741
$L_2$	1.617266
$C_3$	1.99056
$L_4$	1.617266
$C_5$	0.617741

$R_1 = R_2$	1
-------------	---

Table 5-5: 5<sup>th</sup> Order Butterworth low-pass filter element values

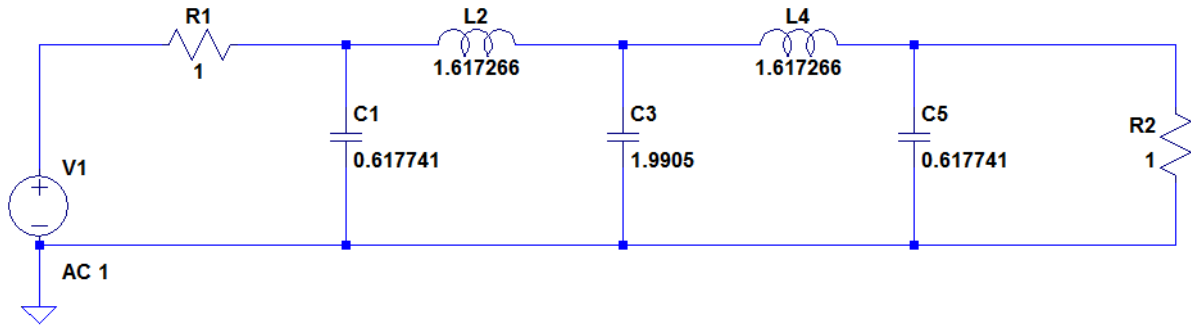


Figure 5-5: 5<sup>th</sup> order ideal Butterworth low-pass filter

If we let  $GC_1 = 0.1$ ,  $RL_2 = 0.1$ ,  $GC_3 = 0.1$ ,  $RL_4 = 0.1$ . and  $GC_5 = 0.1$ , the lossy filter is shown in Figure 6-5 5-6.

Using the Nelder-Mead Algorithm as the optimization routine, the new element values for the lossy filter are shown in Table 5-6, and the simulations results are shown below.

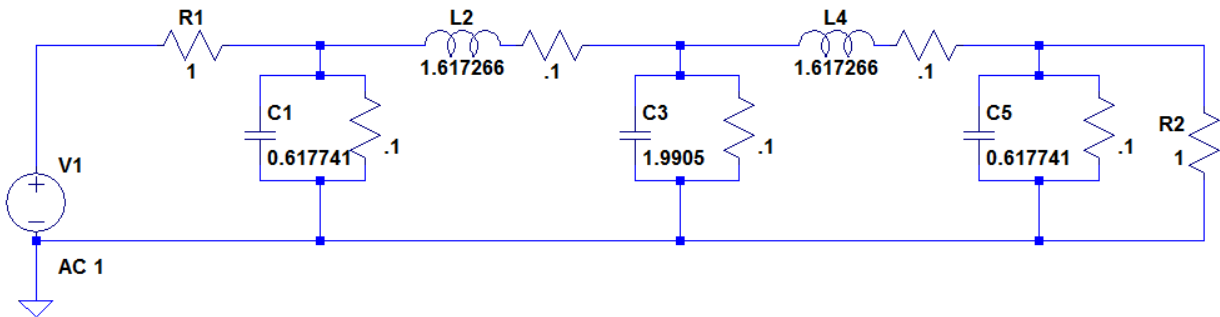


Figure 5-6: 5<sup>th</sup> order lossy Butterworth low-pass filter

Element	Values
$C_1$	0.71178
$L_2$	1.59611
$C_3$	1.98569

$L_4$	1.59611
$C_5$	0.71178
$R_1 = R_2$	1

Table 5-6: Optimized values for 5<sup>th</sup> order Butterworth

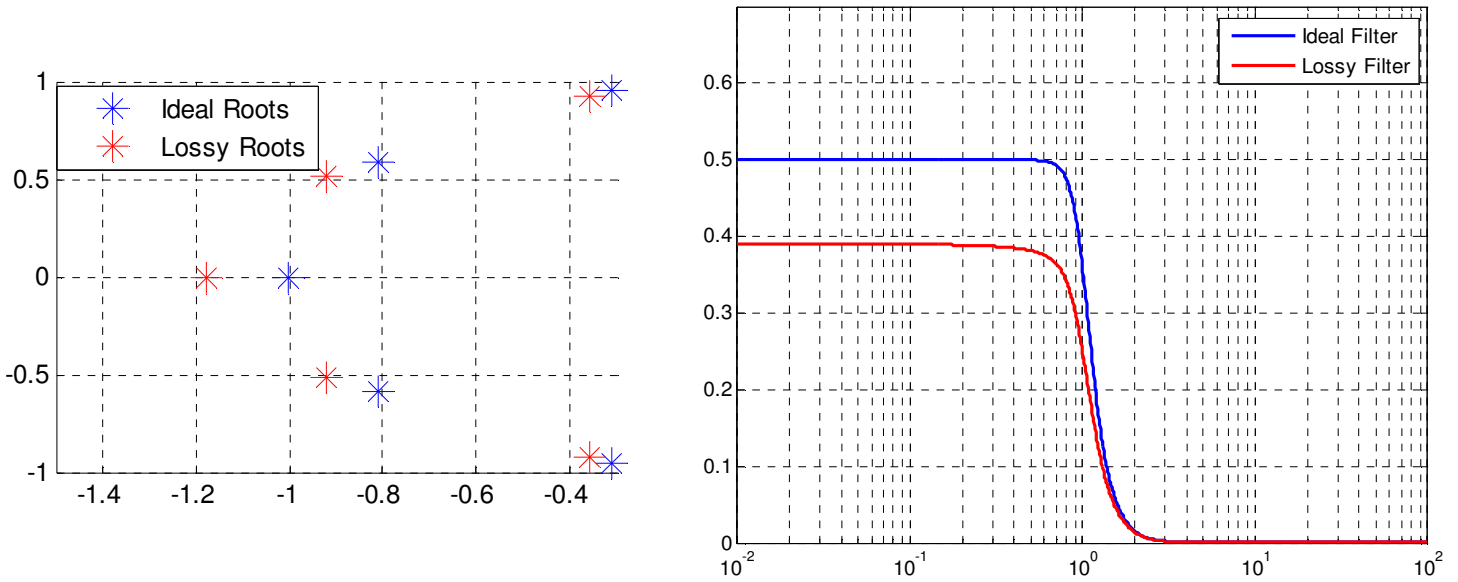


Figure 5-7: 5<sup>th</sup> order Butterworth Pole plot and Frequency Response – Before Optimization

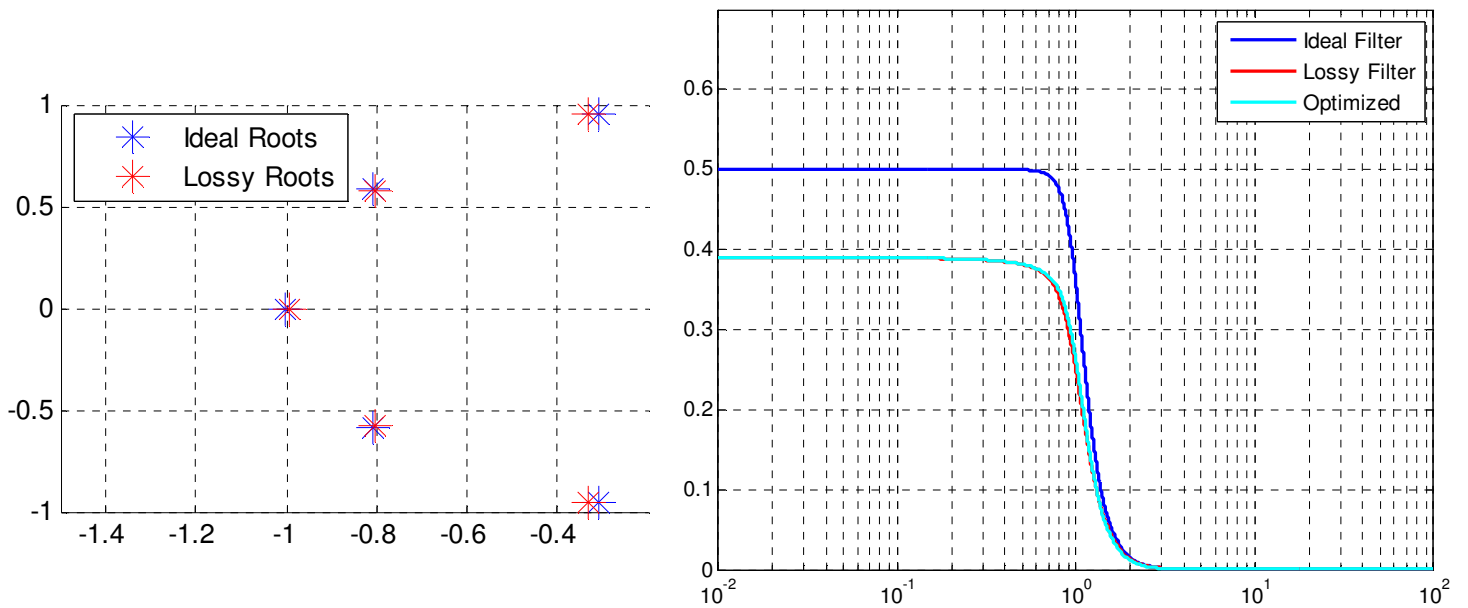


Figure 5-8: 5<sup>th</sup> order Butterworth Pole plot and Frequency Response – After Optimization

When it comes to optimization, the Butterworth filter clearly optimizes better than the Chebyshev filter as show in Figure 5-7 and Figure 5-9. This is due to the fact that the Butterworth filter is flat in the pass-band, while the Chebyshev has ripples. These ripples make it harder to optimize. After the optimization routine, the roots of the lossy and ideal filter are well-matched. The optimized response closely resembles the lossy response in this filter, however this optimized response is still better than the lossy response. This routine does not match the gain between the optimized filter and the ideal filter, if the gain was matched, both the ideal response and optimized response would be almost identical.

### 5.3 8<sup>th</sup> Order Chebyshev Band-pass Filter

For our next example let us take into consideration a 4<sup>th</sup> order band-pass low-pass filter with the following specifications given in Table 5-7

Element	Values
$\alpha_p$	1dB
$\alpha_s$	30dB
$\omega_p$	1rad/sec
$\omega_s$	2rad/sec

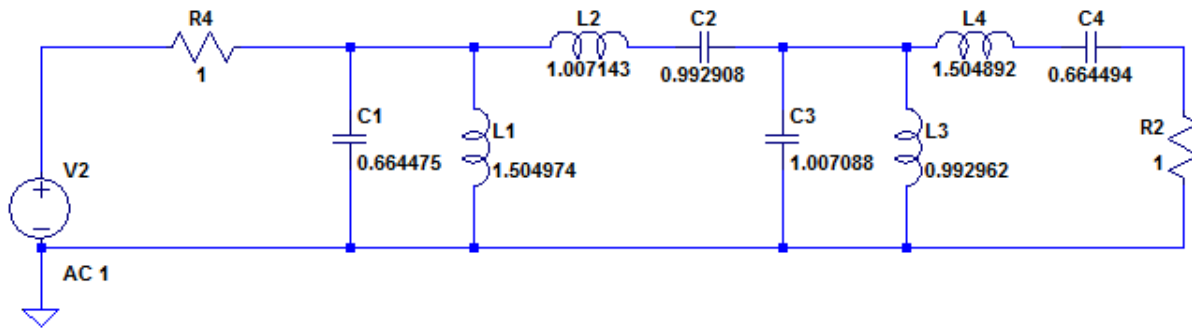
Table 5-7: 8<sup>th</sup> Order Chebyshev Band-pass low-pass filter specifications

Element	Values
$C_1$	0.664475
$L_1$	1.504947
$C_2$	0.992908
$L_2$	1.007143
$C_3$	1.007088
$L_3$	0.992962

$C_4$	0.664499
$L_4$	1.504892
$R_1 = R_2$	1

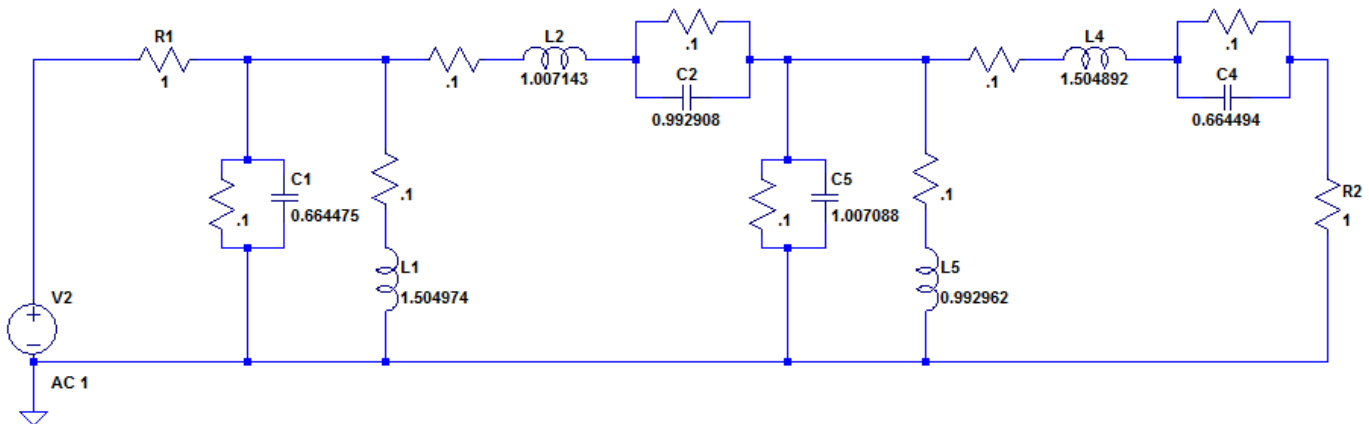
**Table 5-8: 8<sup>th</sup> Order Chebyshev Band-pass low-pass filter element values with  $Q=10\%$**

Using the LADDER software, which uses the synthesis algorithm discussed in Section 3.1, the element values are shown in Table 5-8 for the ideal filter, and the ladder circuit for the ideal filter is shown in Figure 5-9.



**Figure 5-9: 8<sup>th</sup> order Chebyshev band-pass filter**

If we let  $GC_1 = 0.1$ ,  $RL_1 = 0.1$ ,  $GC_2 = 0.1$ ,  $RL_2 = 0.1$ ,  $GC_3 = 0.1$ ,  $RL_3 = 0.1$ , and  $GC_4 = 0.1$ ,  $RL_4 = 0.1$ , the lossy filter is shown in Figure 5-10. Using the Nelder-Mead Algorithm as the optimization routine, the new element values for the lossy filter are shown in Table 5-9, and the simulations results are shown below.



**Figure 5-10: 8th order Lossy Chebyshev band-pass filter**

Element	Values
$C_1$	0.664475



L <sub>1</sub>	1.504947
C <sub>2</sub>	0.992908
L <sub>2</sub>	1.007143
C <sub>3</sub>	1.007088
L <sub>3</sub>	0.992962
C <sub>4</sub>	0.664499
L <sub>4</sub>	1.504892
R <sub>1</sub> = R <sub>2</sub>	1

**Table 5-9: Optimized values for 8<sup>th</sup> order Band-pass filter**

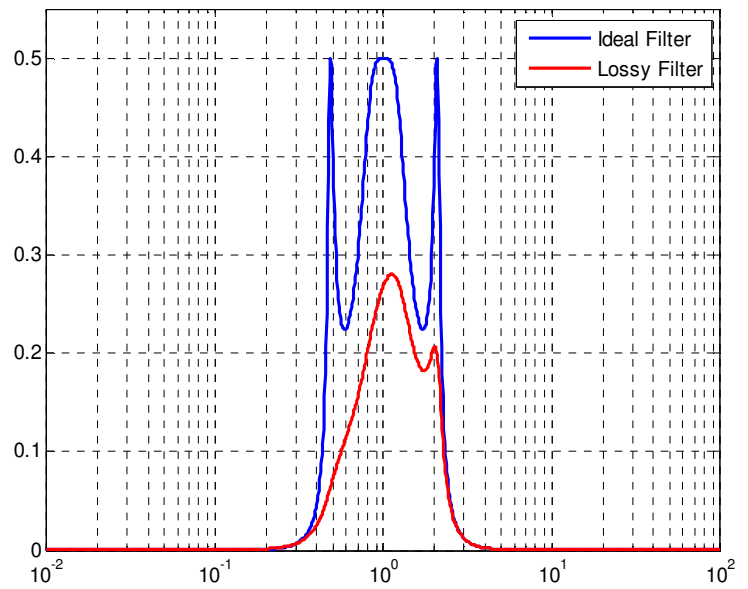
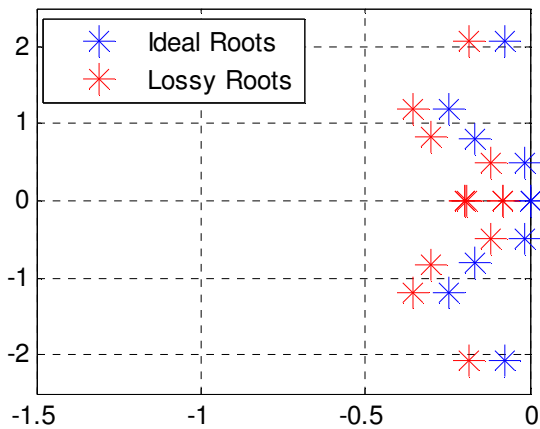


Figure 5-11: 8<sup>th</sup> order Band-pass Pole plot and Frequency Response – Before Optimization

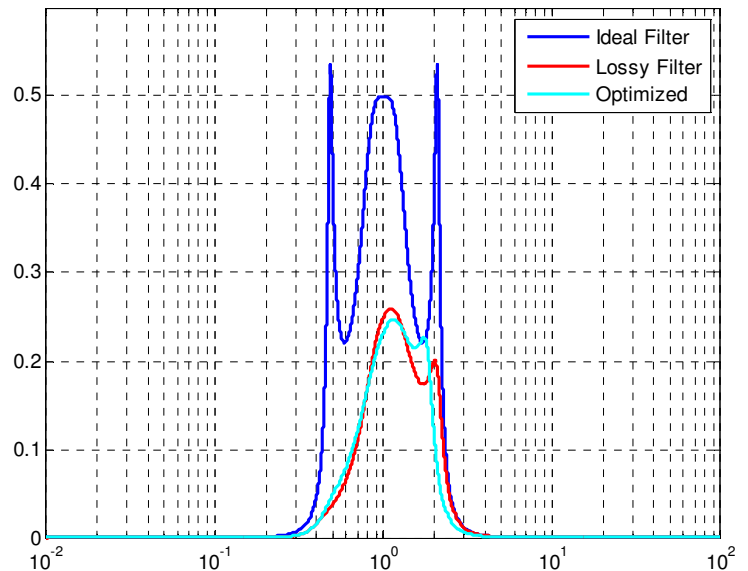
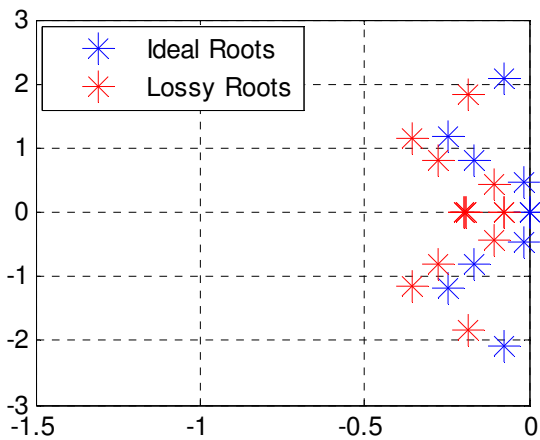


Figure 5-12: 8<sup>th</sup> order Band-pass Pole plot and Frequency Response – After Optimization

For this optimization routine, the band-pass filter is harder to optimize than the low-pass filter which is evident from Figure 5-11 and Figure 5-12. This filter is also harder to optimize because it is an 8<sup>th</sup> order filter, so there are more roots to match up. These results are worse than expected, the filter has been optimized, but another optimization scheme needs to be used for the 8<sup>th</sup> order band-pass filter. Other methods are introduced in Chapter 6 and Chapter 7.

## 6 Nelder-Mead Simulation Results – Error Calculated From Coefficients

In this section, simulation results are calculated by trying to match the coefficients of the lossy transfer function, with the coefficients of the ideal transfer function.

This procedure is explained earlier in the Chapter 3. Once this Sum of Squares Error is calculated (see Equation 6.1), it is then passed on into the Nelder-Mean optimization routine (see Chapter 4) which in turn calculates the optimized element values.

$$SSE = \sum_{i=1}^n (a_i - b_i)^2 + (a_0 - b_0)^2 + (a_n - b_n)^2 \quad (6.1)$$

I will re-simulate all of the circuits in Chapter 5, here in chapter 6, using the matching coefficients method.

### 6.1 3<sup>rd</sup> Order Chebyshev Low-pass Filter

The 3<sup>rd</sup> order ideal and lossy Chebyshev filters shown below,

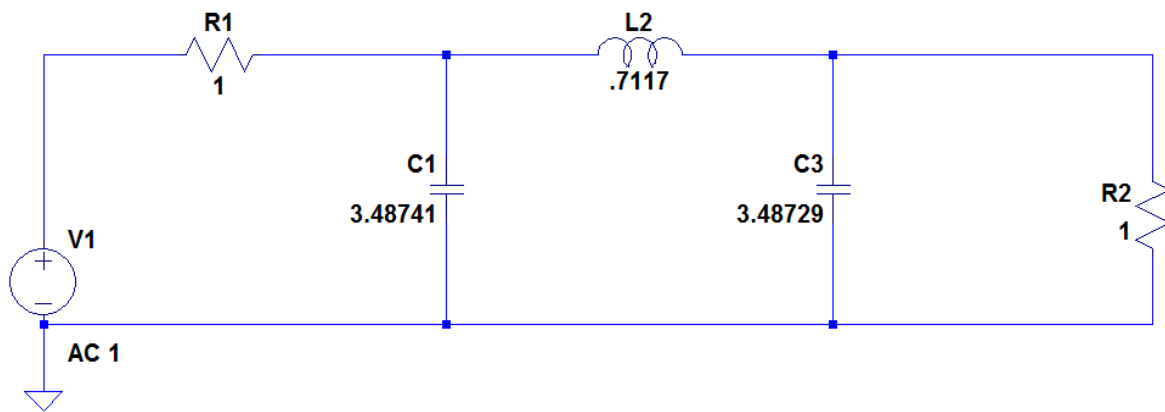


Figure 6-1: 3<sup>rd</sup> order ideal Chebyshev low-pass filter

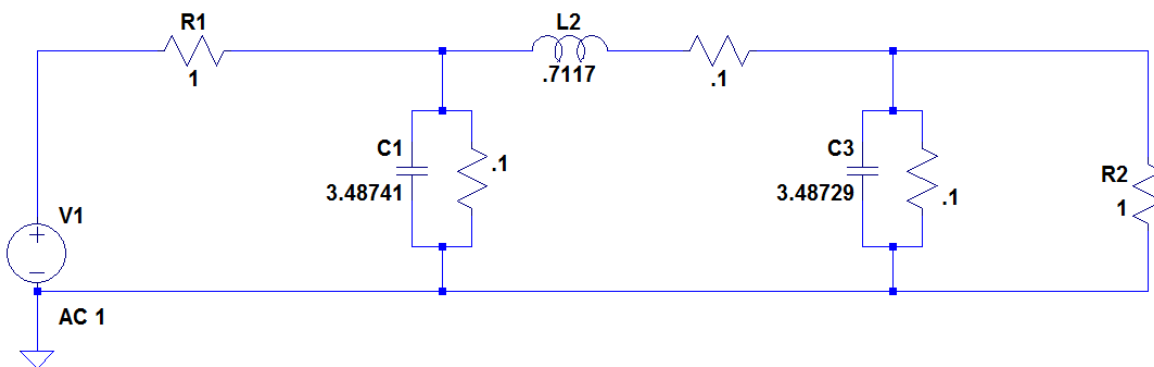


Figure 6-2: 3<sup>rd</sup> order lossy Chebyshev low-pass filter

The transfer functions for the *ideal* circuit is:

$$TF = \frac{.1253}{1s^3 + 0.5972 + 0.9284s + .2506} \quad (6.2)$$

The transfer circuit for the *lossy* circuit is:

$$TF = \frac{.1253}{1s^3 + 0.7975 + 1.0394s + 0.2908} \quad (6.3)$$

Using the coefficients from the ideal and lossy transfer function, and plugging them into the mean square value error equation, the error was calculated and then this error was passed on into the optimization routine.

The transfer circuit for the *optimized* circuit is:

$$TF = \frac{.0586}{1s^3 + 0.5989 + .9278s + 0.1319} \quad (6.4)$$

The optimized element values for the lossy filter are shown in Table 6-1, and the simulations results are shown below.

	Ideal Values	Optimized Values
C <sub>1</sub>	3.48741	7.16487
L <sub>2</sub>	.7117	0.34265
C <sub>3</sub>	3.48729	7.16487
R <sub>1</sub> = R <sub>2</sub>	1	1

Table 6-1: 3<sup>rd</sup> Order Chebyshev Filter with Ideal and Optimized Values

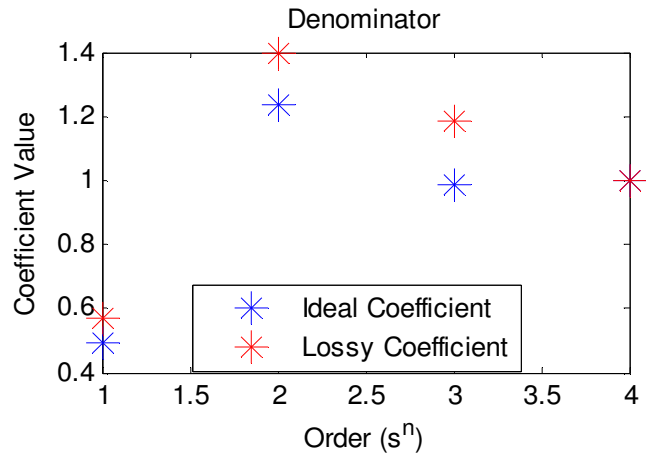


Figure 6-3: 3<sup>rd</sup> order Chebyshev Coefficient Value Vs Order Plot – Before Optimization

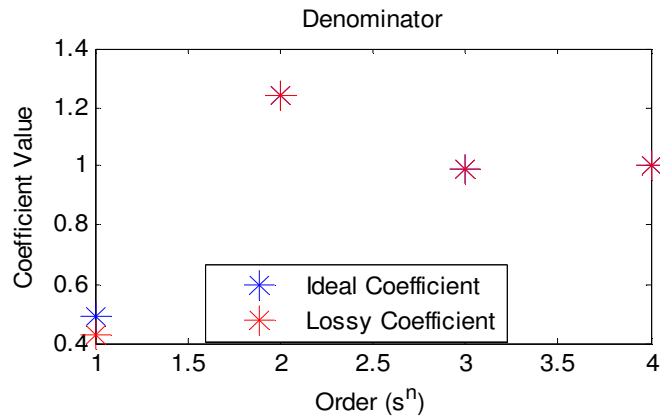


Figure 6-4: 3<sup>rd</sup> order Chebyshev Coefficient Value Vs Order Plot – After Optimization

After the optimization routine, the coefficients of the lossy and ideal filter nearly match, except for the numerator and the last coefficients value. The frequency responses of the ideal, lossy and optimized filter are shown below.

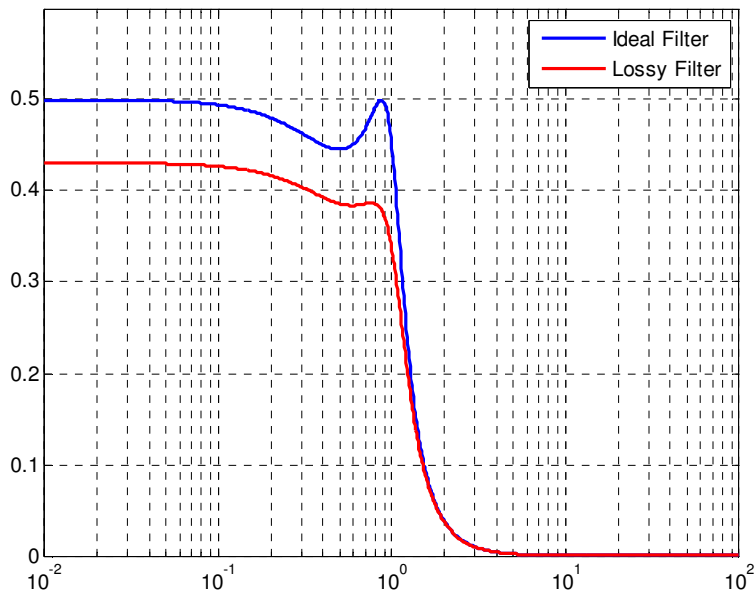


Figure 6-5: 3<sup>rd</sup> order Chebyshev filter - Before Optimization

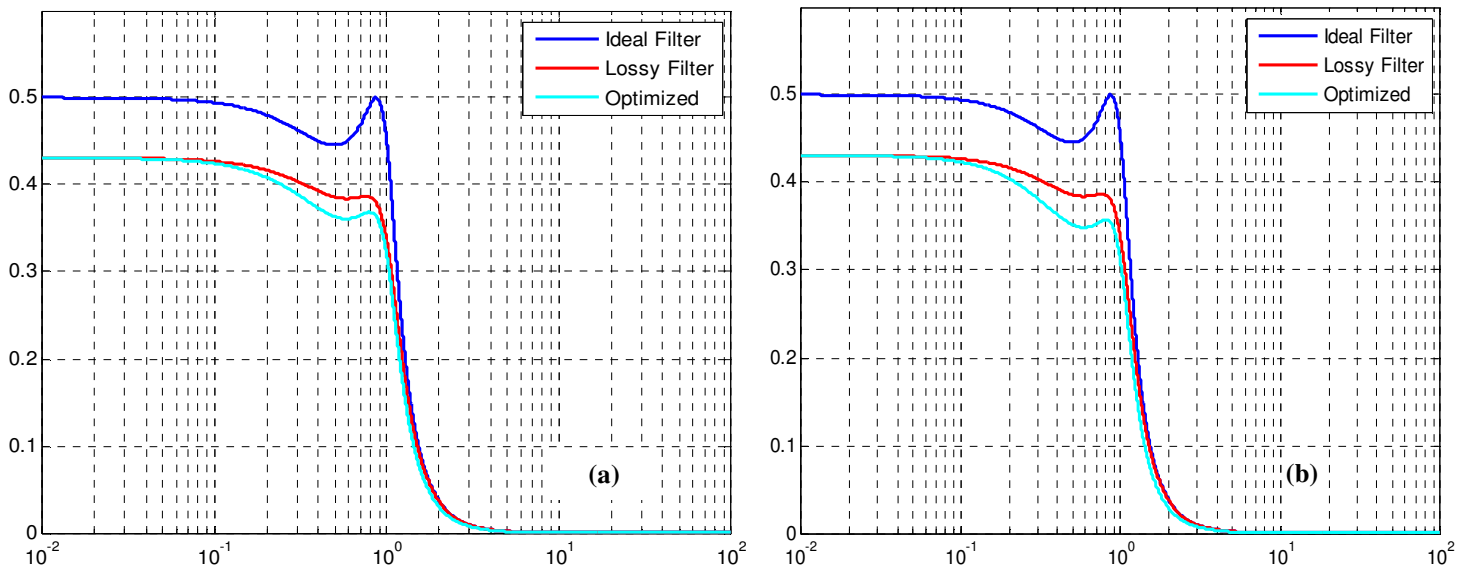


Figure 6-6: 3<sup>rd</sup> order Chebyshev filter- After Optimization

a) Roots Used to calculate error; b) Coefficients used to calculate error

Figure 6-6 shows the frequency response for two types of error calculations. Figure 6-6a shows the results after optimization, when the error is calculated from the roots (this was presented in section 5.1), and Figure 6-6b shows frequency response after optimization, when the error has been calculated from the coefficient difference (explained in this chapter). Both are valid methods, but results are slightly better when

the error is calculated from the roots, which is evident in the optimized response. Calculating the error from the roots gives a more uniform response for the optimized plot, which closely resembles the ideal filter, except it has more attenuation. While calculating the error from the coefficients, the optimized plot starts out just like the roots optimized plot, but the response falls off to some extent when it comes to replicating the ripple. This routine does not match the gain between the optimized filter and the ideal filter, which is the cause for a higher attenuation in the optimized plot.

## 6.2 5<sup>th</sup> Order Butterworth Low-pass Filter

The 5<sup>th</sup> order ideal and lossy Butterworth filters shown below,

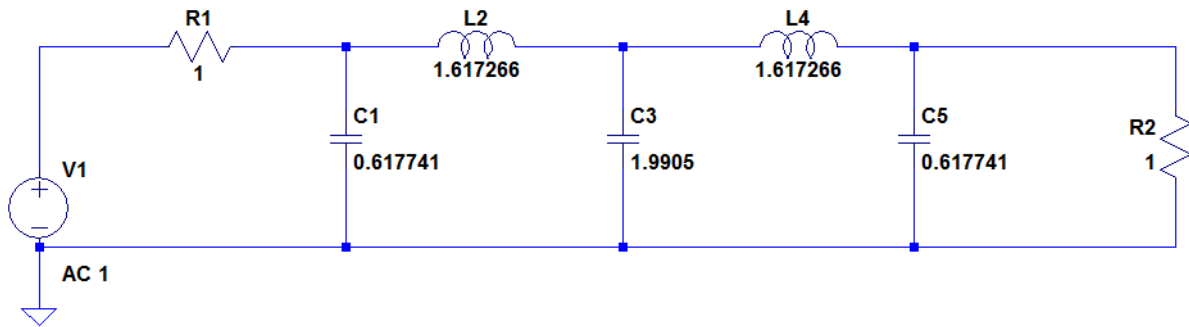


Figure 6-7: 5<sup>th</sup> order ideal Butterworth low-pass filter

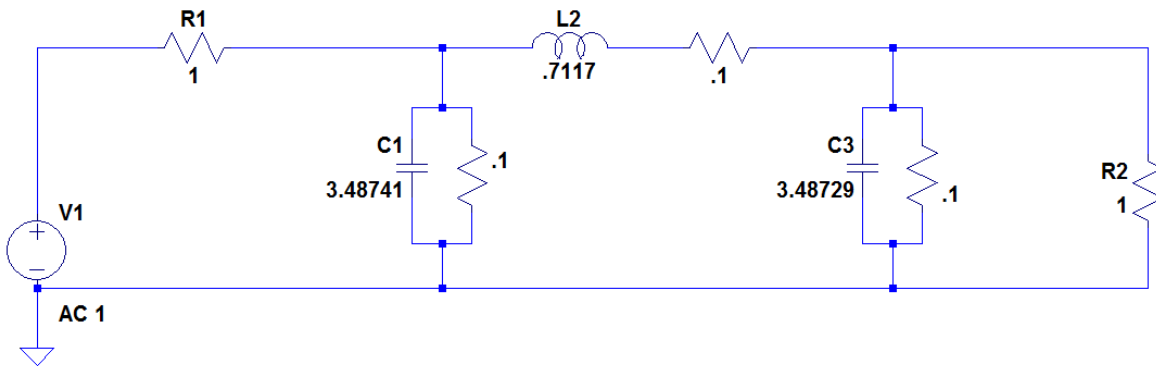


Figure 6-8: 3<sup>rd</sup> order lossy Chebyshev low-pass filter

The transfer functions for the *ideal* circuit is:

$$TF = \frac{0.5012}{1s^5 + 3.237s^4 + 5.241s^3 + 5.243s^2 + 3.2422s + 1.0024} \quad (6.5)$$

The transfer circuit for the *lossy* circuit is:

$$TF = \frac{0.5012}{1s^5 + 3.735s^4 + 6.419s^3 + 6.1666s^2 + 4.1562s + 1.2857} \quad (6.6)$$

Using the coefficients from the ideal and lossy transfer function, and plugging them into the mean square value error equation, the error was calculated and then this error is passed on into the optimization routine.

The transfer circuit for the *optimized* circuit is:

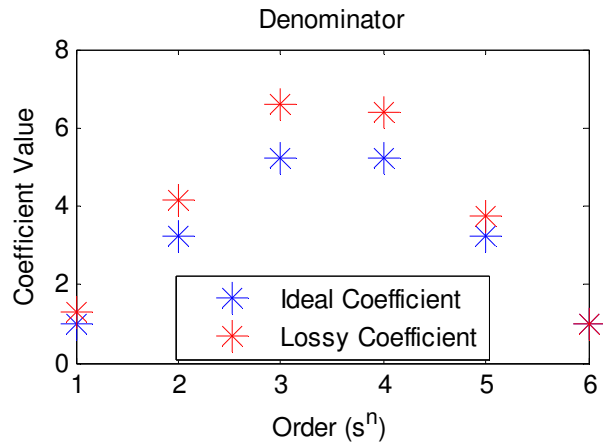
$$TF = \frac{0.3822}{1s^5 + 3.239s^4 + 5.234s^3 + 5.254s^2 + 3.234s + 0.9805} \quad (5.7)$$

The new element values for the lossy filter are shown in Table 6-2, and the simulations results are shown below.

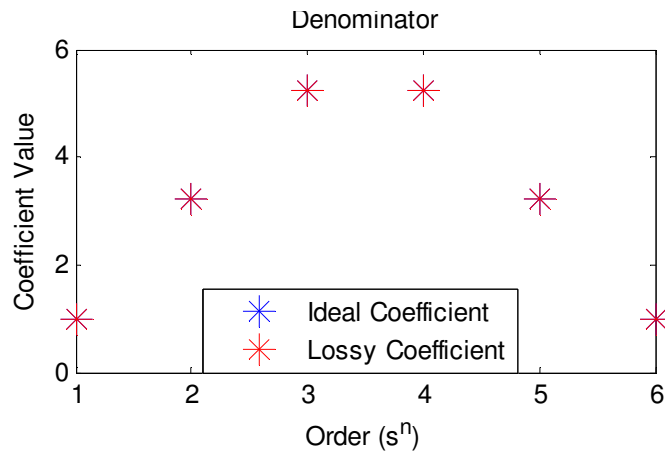
	Ideal Values	Optimized Values
C <sub>1</sub>	0.71178	0.71781
L <sub>2</sub>	1.59611	1.64704
C <sub>3</sub>	1.98569	1.87059
L <sub>4</sub>	1.59611	1.64776
C <sub>5</sub>	0.71178	0.71796
R <sub>1</sub> = R <sub>2</sub>	1	1

**Table 6-2: 5<sup>th</sup> Order Butterworth Filter with Ideal and Optimized Values**





**Figure 6-9: 5<sup>th</sup> order Butterworth Coefficient Value Vs Order Plot – Before Optimization**



**Figure 6-10: 5<sup>th</sup> order Butterworth Coefficient Value Vs Order Plot – After Optimization**

After the optimization routine, the coefficients of the lossy and ideal filter nearly match, except for the numerator and the last coefficient value. However, these values are a lot closer to each other than the results given from the 3<sup>rd</sup> order Chebyshev filter which, was evaluated in Section 6.1. The frequency responses of the ideal, lossy and optimized filter are shown below.

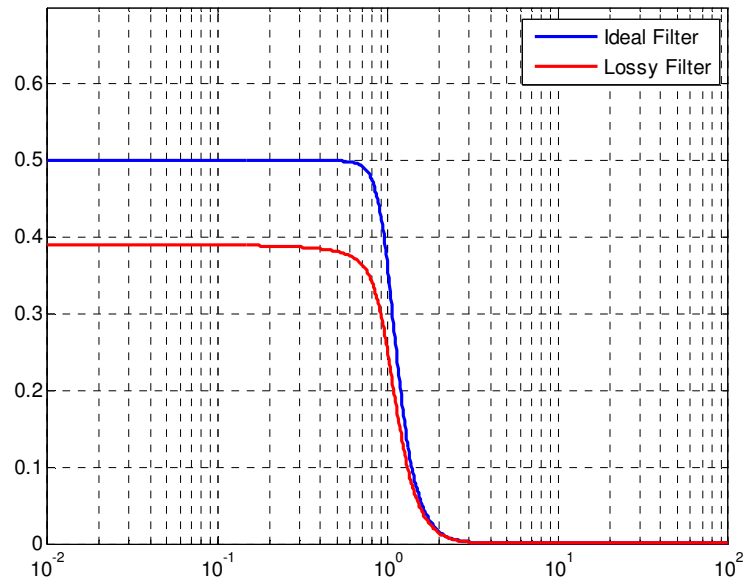


Figure 6-11: 5<sup>th</sup> order Butterworth filter- Before Optimization

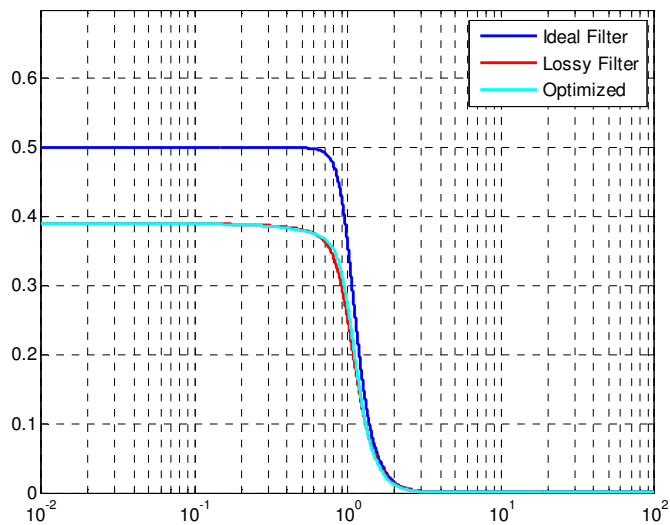


Figure 6-12: 5<sup>th</sup> order Butterworth filter- After Optimization

Figure 6-12 shows the frequency response after optimization when the error has been calculated from the coefficient difference. For the Butterworth filter, both error calculating methods have produced identical results after optimization. This is not the case for Chebyshev filters. The ripples make the Chebyshev filter harder to optimize. In order to achieve ideal-like results for Chebyshev filters, it is best to calculate the error from the difference between the roots, and not from the difference between the coefficients, or use the optimization method presented in Chapter 7.

### 6.3 8<sup>th</sup> Order Chebyshev Band-pass Filter

The 4<sup>th</sup> order ideal and lossy Chebyshev filters shown are below,

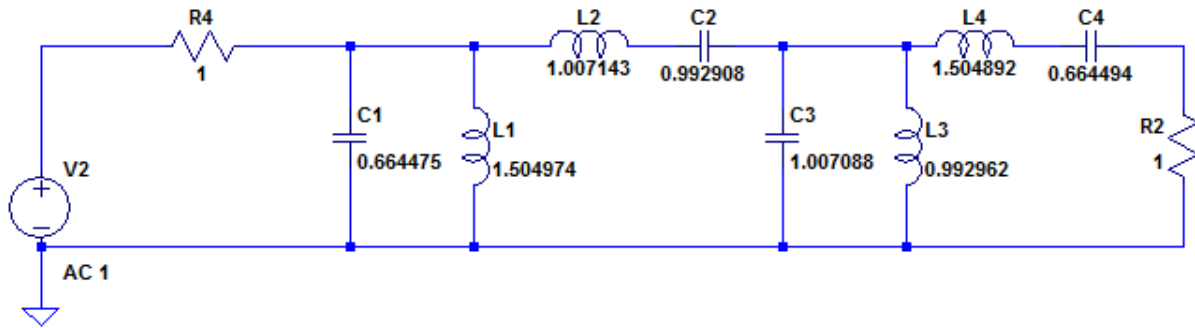


Figure 6-13: 8<sup>th</sup> order Chebyshev Band-pass filter

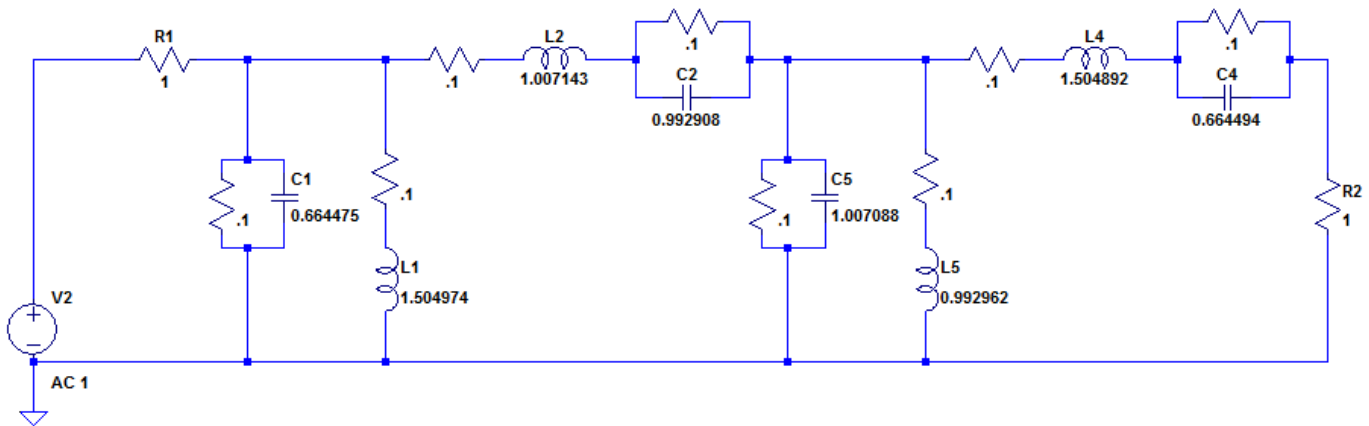


Figure 6-14: 8<sup>th</sup> order lossy Chebyshev Band-pass filter

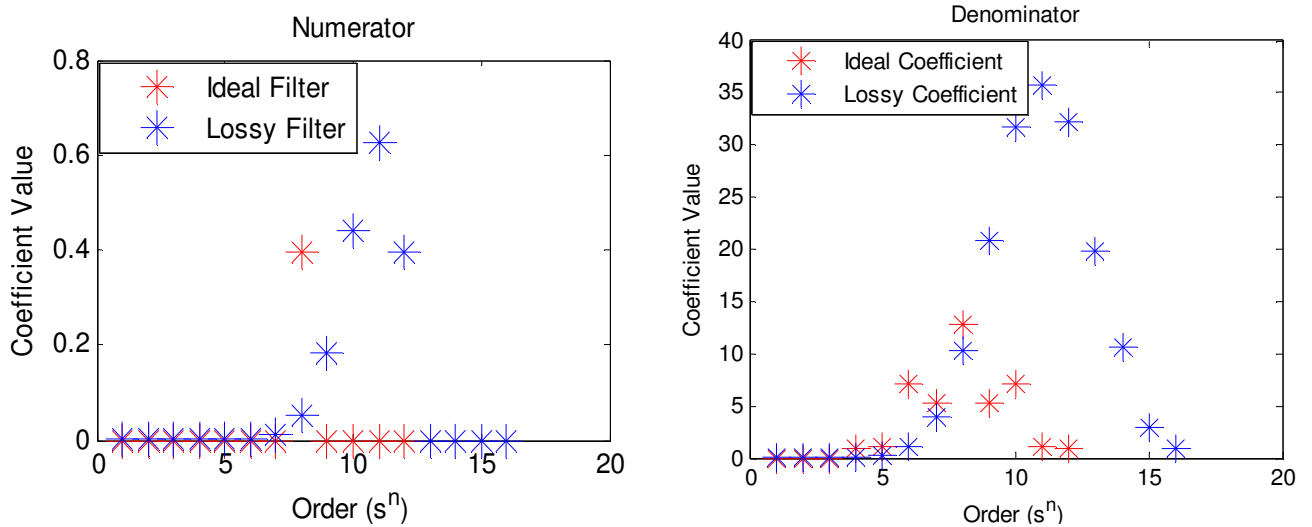
The transfer functions for the *ideal* circuit is:

$$TF = \frac{0.3941s^7}{1s^{11} + 1.019s^{10} + 7.0312s^9 + 5.254s^8 + 12.850s^7 + 5.245s^6 + 7.031s^5 + 1.019s^4 + 1.000s^3} \quad (6.8)$$

The transfer circuit for the *lossy* circuit is:

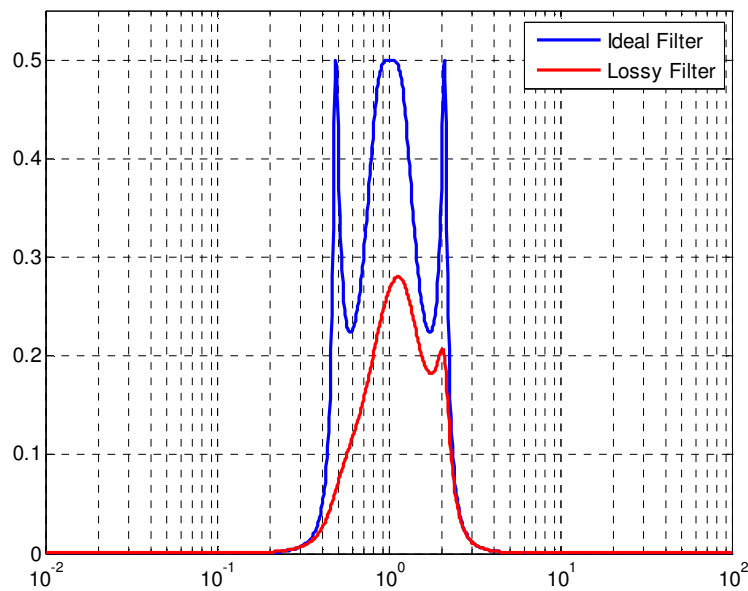
$$TF = \frac{0.3941s^{12} + 0.6238s^{11} + 0.4417s^{10} + 0.1845s^9 + 0.0505s^8 + 0.0095s^7 + 0.0012s^6 + 0.0001s^5}{1s^{15} + 2.950s^{14} + 10.663s^{13} + 19.88s^{12} + 32.142s^{11} + 35.715s^{10} + 31.702s^9 + 20.812s^8 + 10.354s^7 + 3.870s^6 + 1.028s^5 + 0.1848s^4 + 0.0215s^3 + 0.0015s^2 + 0.001s^1} \quad (6.9)$$

As you can see from the transfer functions above, the ideal transfer function and the lossy transfer function are both of different orders even though this is a 8<sup>th</sup> band-pass order filter. The reason for this difference is due to the MATLAB algorithm and is discussed in the next section.



**Figure 6-15: 8<sup>th</sup> order Band-pass Coefficient Value Vs Order Plot – Before Optimization**

Figure 6-15 shows graphically how there are a different number of coefficients for the ideal and for the lossy filter, and the frequency response for the filter in Figure 6-16.



**Figure 6-16: 8<sup>th</sup> order Band-pass Frequency Response – Before Optimization**

## 6.4 MATLAB Algorithm Limitations

As you can see from the transfer functions and from the plots above, the algorithm has introduced extra poles and zeroes into both transfer function. This is a 8<sup>th</sup> order band pass filter which should have a maximum power of  $s^8$ , but yet both transfer functions introduce powers which exceed  $s^8$ . This is due to the way that MATLAB handles rational fractions. In this algorithm, when the impedances is calculated, even though the denominator for both rational fractions may be the same, when adding the fractions, the algorithm will multiply the denominator and ignore the common roots. This is the cause for the higher order roots for the band-pass filter. The algorithm is designed to remove the roots with the small coefficients, but the order is still higher than the original order. These higher orders of roots do not occur in the low-pass filters because these type of filters to not have a complex numerator as well as a complex denominator. The complex numerator and denominator is introduced in the band-pass filter by the series and parallel combinations formed by the inductors and capacitors show in Figures 6:17-18.

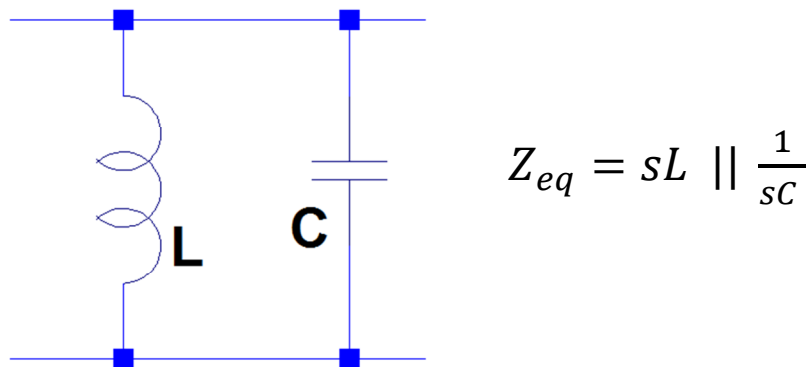


Figure 6-17: Example of Ideal Bandpass Filter Impedance

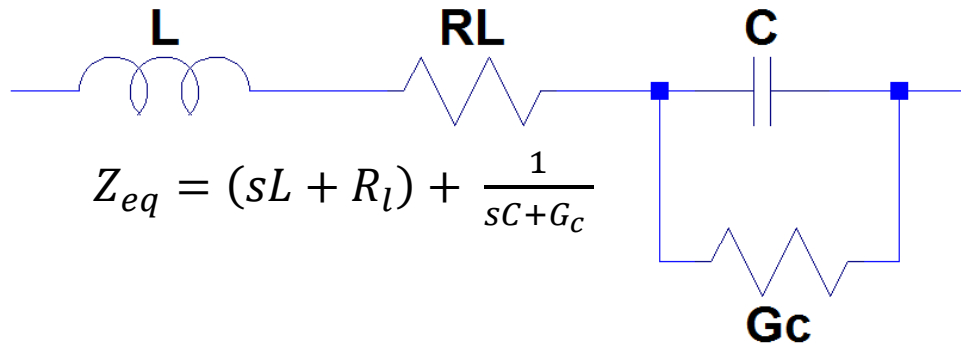


Figure 6-18: Example of Lossy Bandpass Filter Impedance

To try to eliminate the common roots, I calculated the transfer function symbolically so that like terms/roots may cancel out, but I still received the same results. The symbolically calculated transfer function for the 8<sup>th</sup> order ideal band-pass filter is shown is below.

Numerator
<b>S11</b>
$C7 * L6^2 * C3^3 * L2^5$
Denominator
<b>S15</b>
$C7 * L8 * L6^2 * C5 * C3^3 * L4 * L2^5 * C1$
<b>S14</b>
$C7 * L8 * L6^2 * C5 * C3^3 * L4 * L2^5 + L6^2 * C5 * C3^3 * L4 * L2^5 * C1 * C7$
<b>S13</b>
$L6^2 * C3^3 * L2^5 * C5 * L4 * C1 + L6 * C3^3 * L2^5 * C7 * L8 * L4 * C1 + L6^2 * C3^2 * L2^5 * C7 * L8 * C5 * C1 + L6^2 * C3^3 * L2^4 * C7 * L8 * C5 * L4 + L6^2 * C3^3 * L2^5 * C7 * L8 * C5 + L6^2 * C3^3 * L2^5 * C7 * L8 * C1 + L6^2 * C3^3 * L2^5 * L4 * C1 * C7 + L6^2 * C5 * C3^3 * L4 * L2^5 * C7$
<b>S12</b>
$L6^2 * C3^3 * L2^5 * C5 * L4 + L6 * C3^3 * L2^5 * C7 * L8 * L4 + L6^2 * C3^2 * L2^5 * C7 * L8 * C5 + L6^2 * C3^3 * L2^5 * C7 * L8 + L6^2 * C3^3 * L2^5 * L4 * C7 + C7 * L6 * C3^3 * L2^5 * L4 * C1 + C7 * L6^2 * C3^2 * L2^5 * C5 * C1 + C7 * L6^2 * C3^3 * L2^4 * C5 * L4 + C7 * L6^2 * C3^3 * L2^5 * C5 + C7 * L6^2 * C3^3 * L2^5 * C1$
<b>S11</b>
$L6 * C3^3 * L2^5 * L4 * C1 + L6^2 * C3^2 * L2^5 * C5 * C1 + L6^2 * C3^3 * L2^4 * C5 * L4 + L6^2 * C3^3 * L2^5 * C5 + L6^2 * C3^3 * L2^5 * C1 + L6 * C3^2 * L2^5 * C7 * L8 * C1 + L6 * C3^3 * L2^4 * C7 * L8 * L4 + L6 * C3^3 * L2^5 * C7 * L8 + L6^2 * C3^2 * L2^4 * C7 * L8 * C5 + L6^2 * C3^3 * L2^4 * C7 * L8 + L6^2 * C3^2 * L2^5 * C7 * C1 + L6^2 * C3^3 * L2^4 * C7 * L4 + 2 * C7 * L6^2 * C3^3 * L2^5 + C7 * L6 * C3^3 * L2^5 * L4 + C7 * L6^2 * C3^2 * L2^5 * C5$
<b>S10</b>
$L6 * C3^3 * L2^5 * L4 + L6^2 * C3^2 * L2^5 * C5 + L6^2 * C3^3 * L2^5 + L6 * C3^2 * L2^5 * C7 * L8 + L6^2 * C3^2 * L2^5 * C7 + C7 * L6 * C3^2 * L2^5 * C1 + C7 * L6 * C3^3 * L2^4 * L4 + C7 * L6 * C3^3 * L2^5 + C7 * L6^2 * C3^2 * L2^4 * C5 + C7 * L6^2 * C3^3 * L2^4$
<b>S9</b>

$L6^2 * C3^2 * L2^5 * C1 + L6 * C3^3 * L2^4 * L4 + L6 * C3^3 * L2^5 + L6^2 * C3^2 * L2^4 * C5 + L6^2 * C3^3 * L2^4 + L6 * C3^2 * L2^4 * C7 * L8 + L6^2 * C3^2 * L2^4 * C7 + L2^5 * C7 * L6 * C3^2$
$S^9$
$L2^5 * C3^2 * L6 + C7 * L6 * C3^2 * L2^4$
$S^7$
$C3^2 * L2^4 * L6$

**Table 6-3: Coefficients for the Transfer Function for 4<sup>th</sup> order Ideal Band-pass Filter with Q = 10%**

After subbing the ideal component values into the symbolic transfer function above, the ideal transfer function is:

$$TF = \frac{0.04973s^{11}}{.1262s^{15} + .1287s^{14} + .8872s^{13} + 0.6619s^{12} + 1.621s^{11} + 0.6619s^{10} + 0.8872s^9 + 0.1287s^8 + 0.1262s^7} \quad (6.10)$$

When I tried to calculate the lossy transfer function, MATLAB truncated the symbolic results saying that the “text exceeds maximum line length”, so I did not include these results, but just like the ideal response, the lossy response had higher order roots than the order of the filter. Since there is a different number of coefficients for the ideal and lossy transfer function, this method of optimization is not suitable for complex filters, another method is introduced in Chapter 7.

## 7 Levenberg – Marquardt (LM) Simulation Results

The Levenberg-Marquardt (LM) algorithm, also known as the damped least square method is another optimization method that provides solutions to the problem of minimizing a non-linear function. This algorithm is fast and stable, and is used a lot in the field of neural networks. This algorithm is also very popular when it comes to curve fitting. Curve fitting is the process of constructing a curve that has the best fit line to a series of data points. The LM algorithm is a combination of two minimization methods: the Steepest Descent method and the Gauss-Newton method. In the Steepest Decent method, the sum of the sum of the squared errors is reduced by updating the direction of the greatest reduction of the least squares objective. This is an iterative method that is given an initial point and follows the negative of the gradient in order to move the point toward a critical point, which is the desired local minimum. This method is very popular for optimization problems, but its main disadvantage is that it can take a long time to converge. There are two main reasons for the slow convergence: the first reason is the step size is small and constant, so in a place where the gradient is gentle, the training process is very slow. The second reason is that the

curvature of the error surface may not be the same in all directions, so the classic “error valley” problem may exist.

In the Gauss-Newton method, the sum of the squared errors is reduced by assuming the least squares function is locally quadratic, and finding the minimum of the quadratic. The slow convergence of the steepest descent method is greatly improved in the Gauss-Newton method. Using the second order-derivatives of the error function to “naturally” evaluate curvature of the error surface, this algorithm can find the proper step sizes for each direction and can converge very fast. However, this only happens when the quadratic approximation of the error function is reasonable. This algorithm does require the calculation of the second order derivative of the function which can be very complicated, but it uses the Hessian matrix  $\mathbf{H}$ , which is equivalent to the second order derivative though. The Hessian matrix  $\mathbf{H}$  gives the proper evaluation of the change of the gradient vector  $\mathbf{g}$ . In order to calculate the Hessian matrix  $\mathbf{H}$ , the Jacobian matrix  $\mathbf{J}$  also needs to be calculated. The Jacobian matrix is a matrix of all first order partial derivatives of the function. This Jacobian matrix is related to the gradient vector  $\mathbf{g}$ , and to the error vector  $\mathbf{e}$ , which is shown in Equation 7.1

$$\mathbf{g} = \mathbf{J}\mathbf{e} \quad (7.1)$$

After some integration and substitutions, the relationship between the Hessian matrix  $\mathbf{H}$  and the Jacobian matrix  $\mathbf{J}$  can be written rewritten as the relationship shown in Equation 7.2.

$$\mathbf{H} \approx \mathbf{J}^T \mathbf{J} \quad (7.2)$$

The basic idea of the Levenberg-Marquardt algorithm is that it performs a combined training process around the area with a complex curvature. The Levenberg-Marquardt algorithm switches to the steepest descent algorithm, until the local curvature is proper to make a quadratic approximation, this it becomes the Gauss-Newton method which can speed up the convergence. The update rule for the LM algorithm is shown in Equation 7.3:



$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \quad (7.3)$$

where

- $w$  is the weight vector
- $e$  is the error vector
- $J$  is the Jacobian matrix
- $\mu$  is the combination coefficient
- $I$  is the identity matrix

As the combination coefficient increases the Hessian matrix  $H$  will be factored out of the above equation and the Steepest Descent method is used. As the combination coefficient decreases, the Hessian matrix  $H$  becomes more significant, and the Gauss-Newton method is used. This allows the training algorithm to interpolate between the Steepest Descent method and the Gauss Newton method. Higher combination coefficient favors Steepest Descent, and lower combination coefficient favors Gauss-Newton.

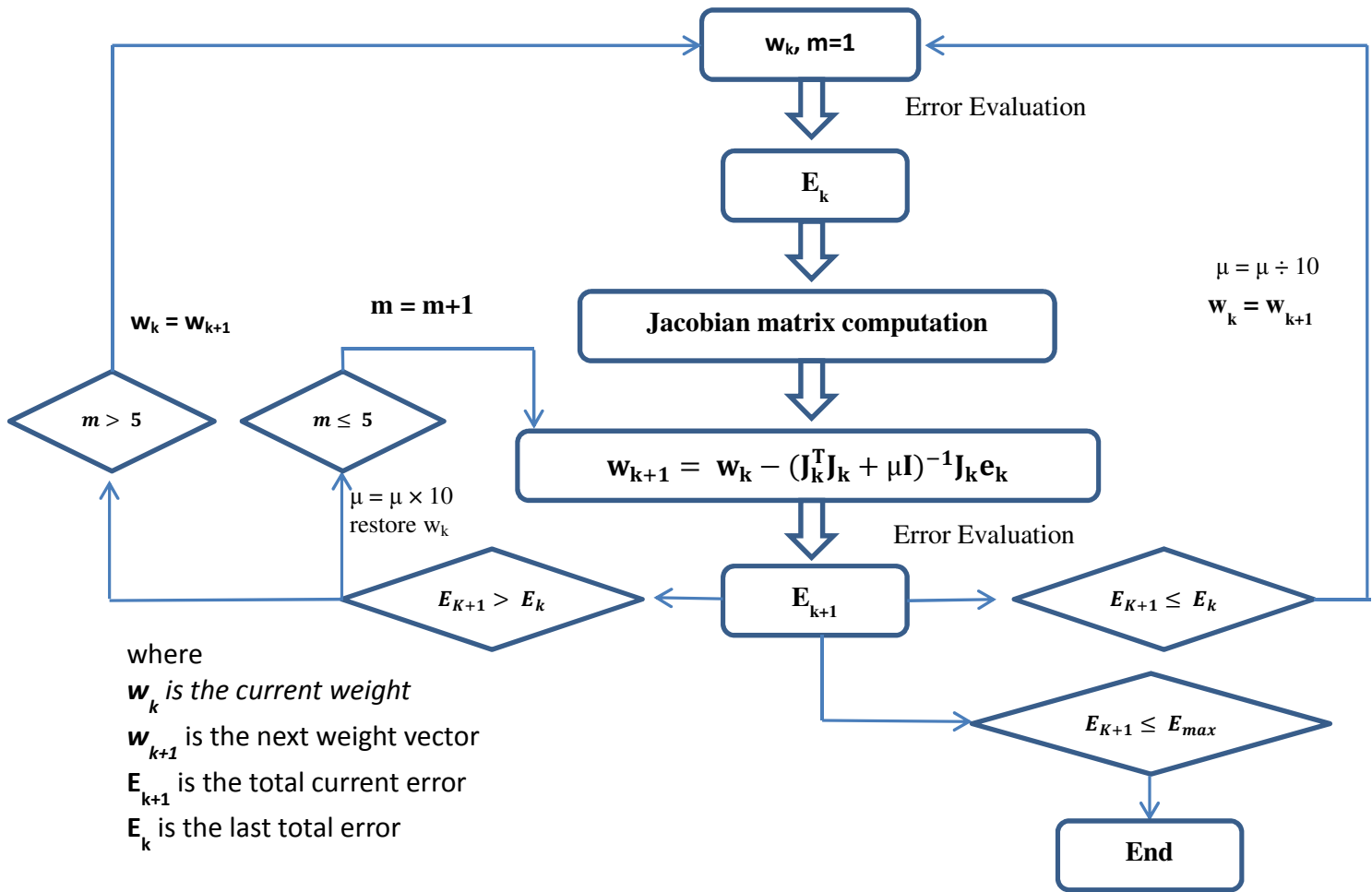
This LM Algorithm is used to optimize the amplitude characteristics of the filter. This method uses the transfer function just like the Nelder-Mead method, however, " $j\omega$ " is substituted for " $s$ " in the transfer function. Then for different " $\omega$ ", the bode plot will have different amplitudes, and then the LM algorithm is used to optimize the bode plot characteristics. *Pseudocode* and a block diagram for the LM algorithm is shown in Figure 7-1

```

for iter=1:maxiter
    calculate [num_id,den_id] % ideal transfer function
    calculate [num_ls,den_ls] % lossy transfer function
    for sample w
        substitute 's' with jw for num_id/den_id
        substitute 's' with jw for num_ls/den_ls
        calculate error
    end
    calculate Jacobian matrix with numeric way
    gradient=Jacobian'*error
    hessian=Jacobian'*Jacobian
    del=inv(hessian+mu*I)*gradient % change of elements
    use LM algorithm to adjust mu
    ...
    elements=elements+del % update elements
    ...
end

```

(a) Pseudocode



(b) Block Diagram

Figure 7-1: Block Diagram and Pseudocode for LM Algorithm

The block diagram shows that as the Error  $E_{k+1}$  decreases, (smaller than the last error), it implied a quadratic approximation is used on the total error function, and the combination coefficient  $\mu$  needs to be reduced, as long as the new error is less than or equal to the max error  $E_{max}$ . (As  $\mu$  decreases, the Gauss-Newton method is used) If the error  $E_{k+1}$  increases, (larger than the last error), this means that it is necessary to follow the gradient more to look for a quadratic approximation, so the combination component  $\mu$  needs to be increased. (As  $\mu$  increases, the Steepest Descent method is used). The basic steps of the LM Algorithm are explained in Table 7-1 and the results are in the next section.

1) With the initial weights (randomly generated) evaluate the total error (SSE).	3) With new weights evaluate the total error (SSE)	5. If error is decreased, then accept step (keep new weight vector) and decrease $\mu$ by a factor of 10
2) Do an update to adjust weights: $w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k$	4) If error increased, retract the step (keep new weight vector as current one) and increase $\mu$ by a factor of 10. →Go to Step 2:	6. →Go to Step 2 with new weights until the current total error is smaller than required value.

Table 7-1 LM Algorithm Basic Steps

### 7.1 3<sup>rd</sup> Order Chebyshev Low-pass Filter

Consider the generic 3<sup>rd</sup> order Chebyshev low-pass filter shown in Figure 7-5 with  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec. The element values for the ideal filter, lossy filter, and optimized filter are shown in Table 7-3, and the simulation results are shown in the figures below.

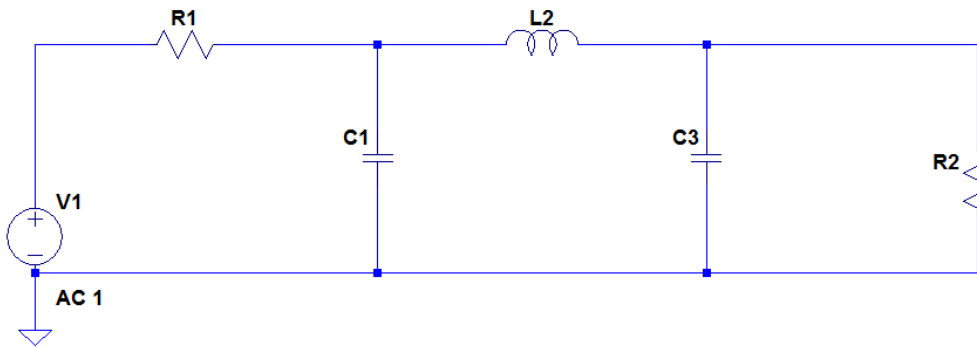


Figure 7-2: 3<sup>rd</sup> Order Chebyshev Low-pass Filter

	Ideal Values	Lossy Values	Optimized Values
C <sub>1</sub>	3.4823	3.4823	5.4045
L <sub>2</sub>	0.7714	0.7714	0.5402
C <sub>3</sub>	3.4823	3.4823	4.4299
R <sub>1</sub>	1	1	1
R <sub>2</sub>	1	1	1.5507
Q	∞	20	20

Table 7-2: 3<sup>rd</sup> Order Chebyshev Low-Pass Filter with Ideal, Lossy and Optimized Values

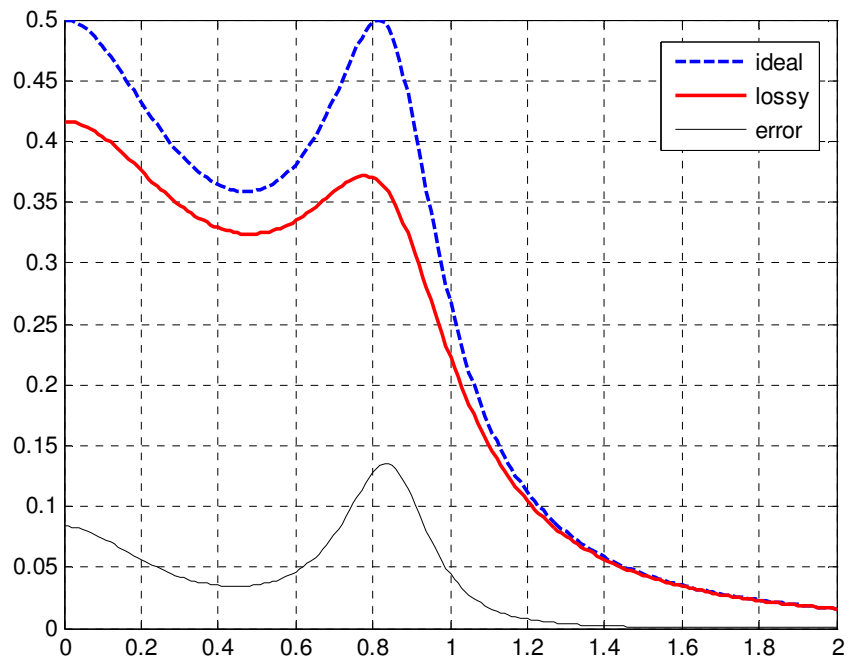


Figure 7-3: 3rd Order Chebyshev Low-Pass Filter Frequency Plot – Before Optimization

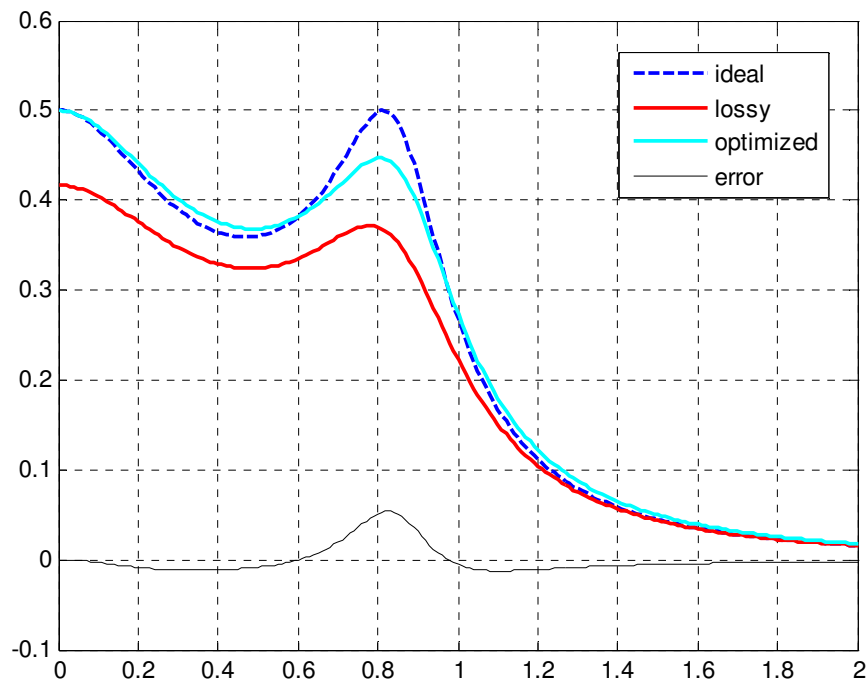


Figure 7-4: 3rd Order Chebyshev Low-Pass Filter Frequency Plot – After Optimization

Figure 7-4 shows how the LM Algorithm has provided better results than the Nelder-Mead method when optimizing the chebyshev filter. The optimized plot does not match the ideal plot completely, but the LM Algorithm can reproduce the ripple better than the Nelder-Mead algorithm. The LM Algorithm also

matches the gain between the ideal response and the lossy response, which is the reason why both the ideal plot and lossy plot have equal gain.

## 7.2 5<sup>th</sup> Order Butterworth Low-Pass Filter

Consider the generic 5<sup>th</sup> order Butterworth low-pass filter shown in Figure 7-5 with  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec. The element values for the ideal filter, lossy filter, and optimized filter are shown in Table 7-3, and the simulation results are shown in the figures below.

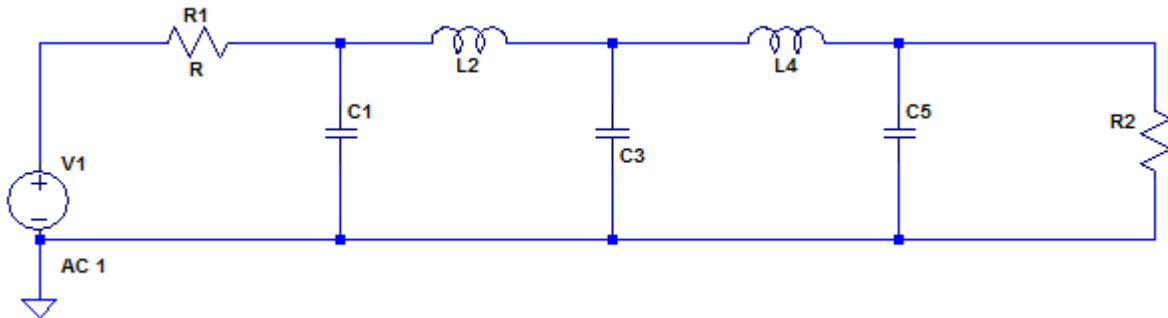


Figure 7-5: 5<sup>th</sup> Order Butterworth Low-pass Filter

	Ideal Values	Lossy Values	Optimized Values
C <sub>1</sub>	0.617741	0.617741	1.4654
L <sub>2</sub>	1.617266	1.617266	1.3223
C <sub>3</sub>	1.99905	1.99905	2.1814
L <sub>4</sub>	1.617266	1.617266	1.3777
C <sub>5</sub>	0.617741	0.617741	0.5208
R <sub>1</sub>	1	1	1
R <sub>2</sub>	1	1	1.4741
Q	$\infty$	20	20

Table 7-3: 5<sup>th</sup> Order Butterworth Low-Pass Filter with Ideal, Lossy and Optimized Values

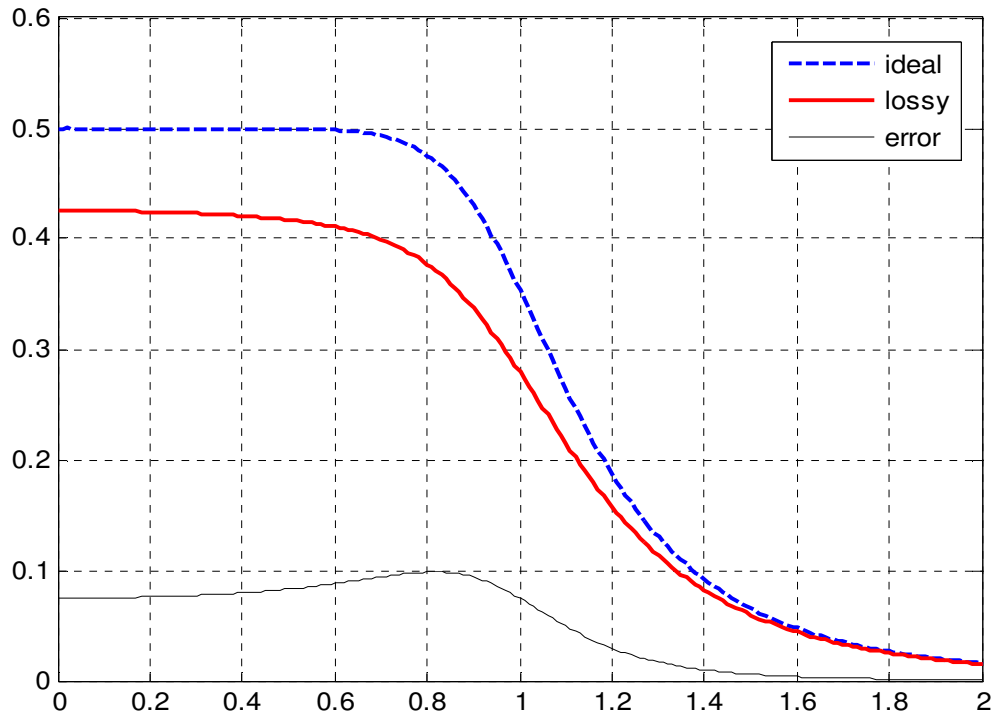


Figure 7-6: 5<sup>th</sup> Order Butterworth Low-Pass Filter Frequency Plot – Before Optimization

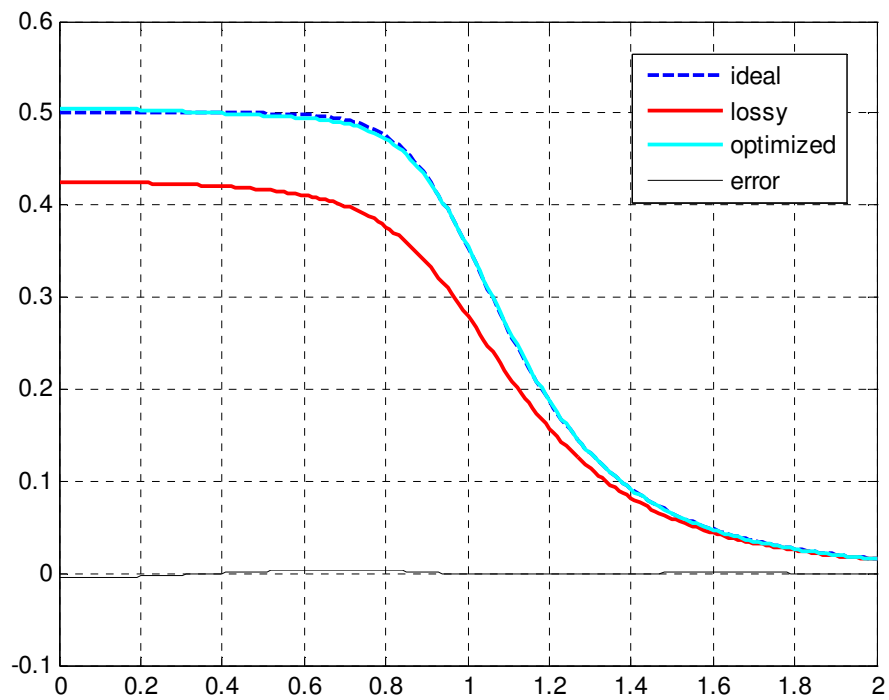


Figure 7-7: 5<sup>th</sup> Order Butterworth Low-Pass Filter Frequency Plot – After Optimization

Figure 7-7 shows how the LM Algorithm has provided the best results when it comes to optimizing lossy ladder filters. The ideal response and the lossy response are pretty much identical, which is evident from the “error” on the plot. The LM Algorithm also matches the gain between the ideal response and the lossy response, which is the reason why both the ideal plot and lossy plot have equal gain.

### 7.3 4<sup>th</sup> Order Chebyshev Low-Pass Filter

Consider the generic 4<sup>th</sup> order Chebyshev low-pass filter shown in Figure 7-8 with  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec. The element values for the ideal filter, lossy filter, and optimized filter are shown in Table 7-4, and the simulation results are shown in the figures below.

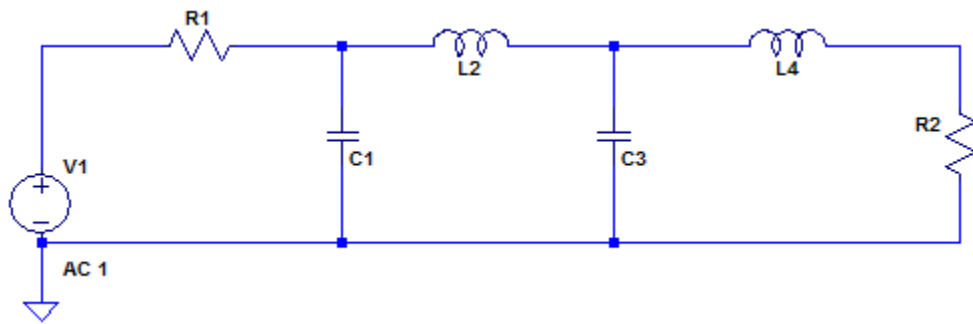


Figure 7-8: 4<sup>th</sup> Order Chebyshev Low-Pass Filter

	Ideal Values	Lossy Values	Optimized Values
C <sub>1</sub>	2.25742	2.25742	3.1759
L <sub>2</sub>	1.5107	1.5107	1.1393
C <sub>3</sub>	1.5106	1.5106	1.7863
L <sub>4</sub>	2.2573	2.2573	2.0249
R <sub>1</sub>	1	1	1
R <sub>2</sub>	1	1	1.5702
Q	∞	20	20

Table 7-4: 4th Order Chebyshev Low-Pass Filter with Ideal, Lossy and Optimized Values

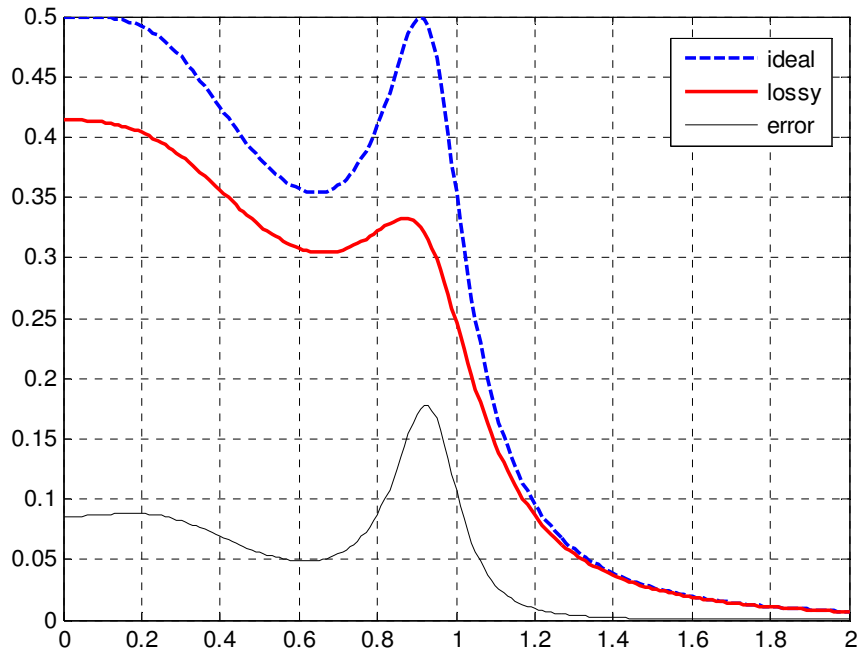


Figure 7-9: 4<sup>th</sup> Order Chebyshev Low-Pass Filter Frequency Plot – Before Optimization

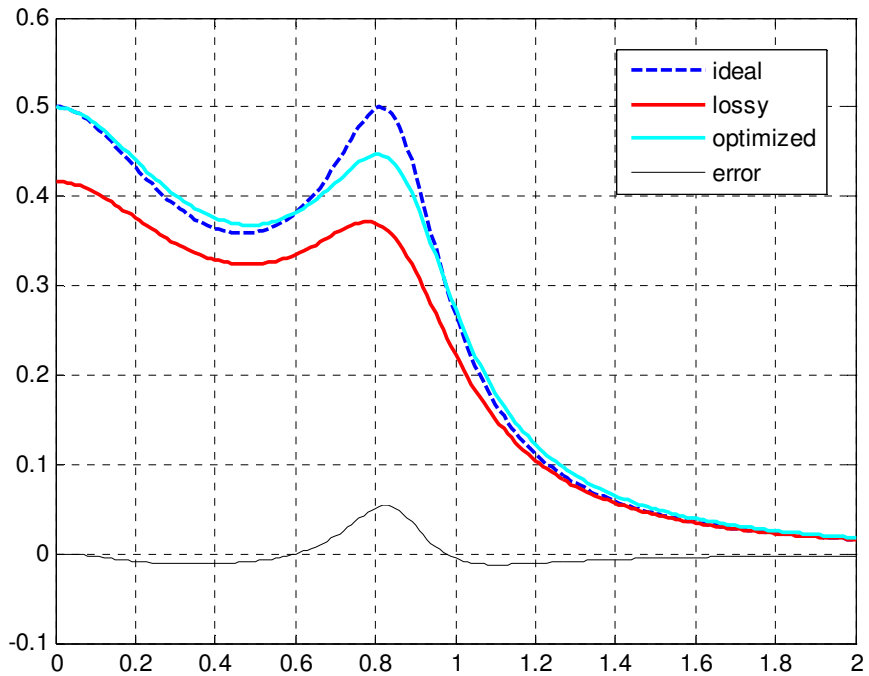


Figure 7-10: 4<sup>th</sup> Order Chebyshev Low-Pass Filter Frequency Plot – After Optimization



Figure 7-10 shows that the LM Algorithm has provided better results than the Nelder-Mead method when it comes to optimizing the chebyshev filter. However, better optimized responses can be obtained with butterworth filters than with chebyshev filters, for both the LM and Nelder-Mead optimization scheme

### 7.4 8<sup>th</sup> Order Chebyshev Band-pass Filter

Consider the generic 4<sup>th</sup> order Chebyshev Bandpass filter shown in Figure 7-11 with  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec. The element values for the ideal filter, lossy filter, and optimized filter are shown in Table 7-4, and the simulation results are shown in the figures below.

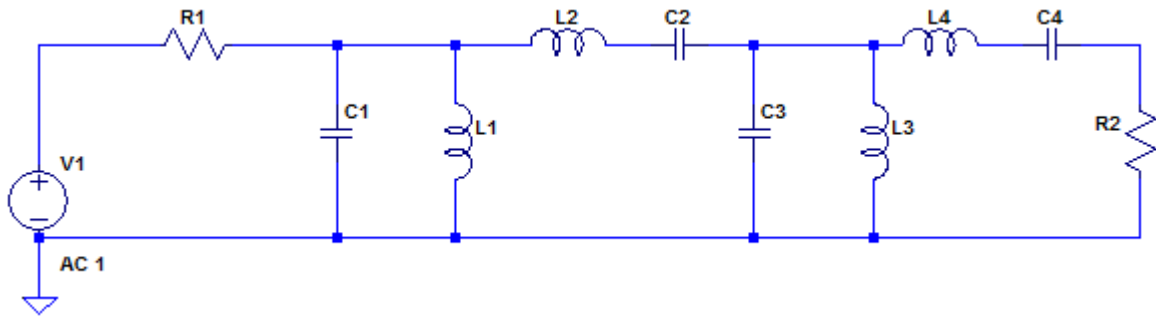


Figure 7-11: 8<sup>th</sup> Order Chebyshev Band-pass Filter

Element	Ideal Values	Lossy Values	Optimized Values
C <sub>1</sub>	4.5146	4.5146	4.0534
L <sub>1</sub>	0.2215	0.2215	0.2467
C <sub>2</sub>	0.3310	0.3310	0.2802
L <sub>2</sub>	3.0212	3.0212	3.5694
C <sub>3</sub>	3.0214	3.0214	2.2806
L <sub>3</sub>	0.3310	0.3310	0.4385
C <sub>4</sub>	0.2215	0.2215	0.1575
L <sub>4</sub>	4.5148	4.5148	6.3476
R <sub>1</sub>	1	1	1
R <sub>2</sub>	1	1	1
Q	$\infty$	20	20

Table 7-5: 8th Order Chebyshev Band-pass Filter with Ideal, Lossy and Optimized Values

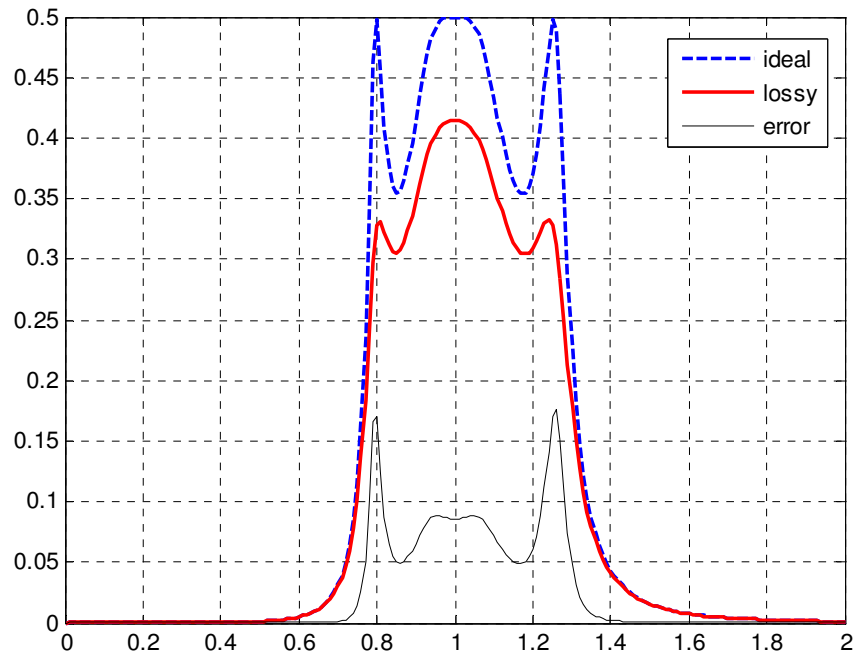


Figure 7-12: 8<sup>th</sup> Order Chebyshev Band-pass Filter Frequency Plot – Before Optimization

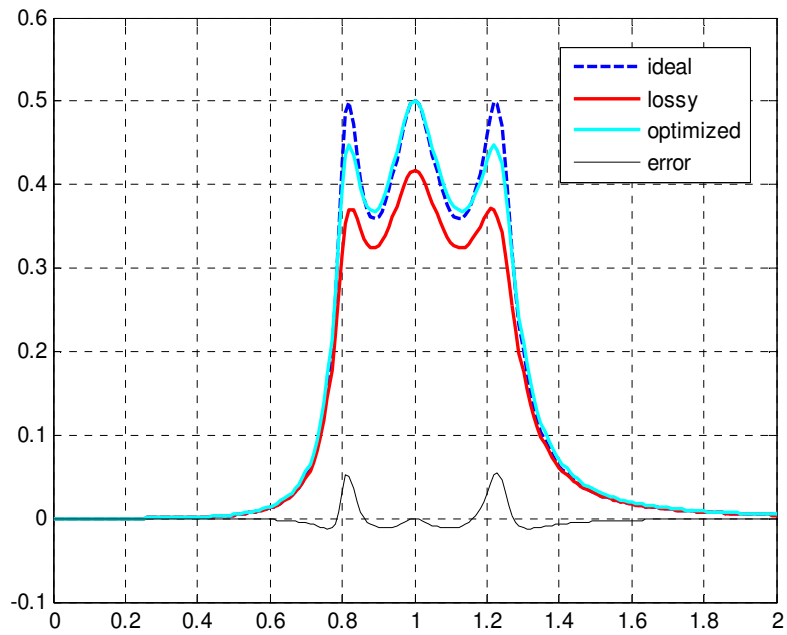


Figure 7-13: 8<sup>th</sup> Order Chebyshev Band-pass Filter Frequency Plot – After Optimization

Once again the LM Algorithm has provided better results, than the Nelder Mead method, as shown in Figure 7-13. The optimized response still has small losses at the peaks of the chebyshev filter which is evident in the “error” plot, but these results are far superior to any Nelder-Mead method.

## 7.5 8<sup>th</sup> Order Butterworth Band-pass Filter

Consider the generic 8<sup>th</sup> order Butterworth Bandpass filter shown in 13 with  $\alpha_p=1$  dB,  $\alpha_s=30$  dB,  $\omega_p=1$ rad/sec,  $\omega_s=2$ rad/sec. The element values for the ideal filter, lossy filter, and optimized filter are shown in Table 7-5, and the simulation results are shown in the figures below.

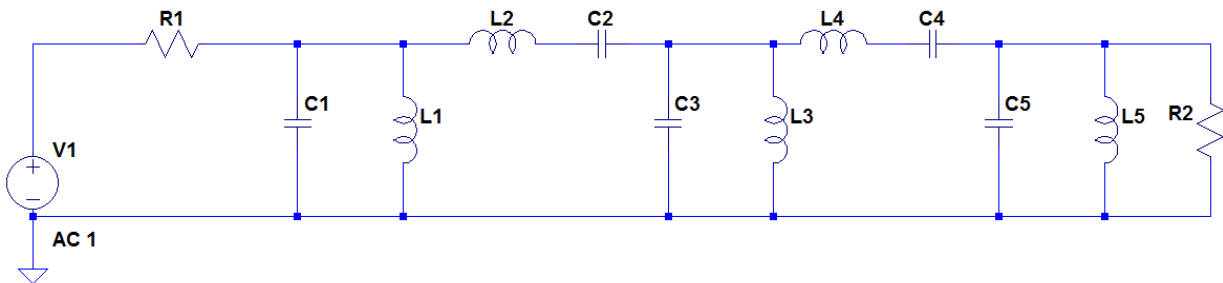


Figure 7-14: 8<sup>th</sup> Order Butterworth Band-pass Filter

Element	Ideal Values	Lossy Values	Optimized Values
C <sub>1</sub>	0.8094	0.8094	0.9639
L <sub>1</sub>	1.2355	1.2355	1.0374
C <sub>2</sub>	3.2345	3.2345	2.7620
L <sub>2</sub>	0.3092	0.3092	0.3621
C <sub>3</sub>	0.2501	0.2501	0.2297
L <sub>3</sub>	3.9981	3.9981	4.3528
C <sub>4</sub>	3.2345	3.2345	2.6492
L <sub>4</sub>	0.3092	0.3092	0.3775,
C <sub>5</sub>	0.8094	0.8094	0.3423
L <sub>5</sub>	1.2355	1.2355	2.9210
R <sub>1</sub>	1	1	1
R <sub>2</sub>	1	1	1
Q	$\infty$	20	20

Table 7-6: 8th Order Butterworth Band-pass Filter with Ideal, Lossy and Optimized Values

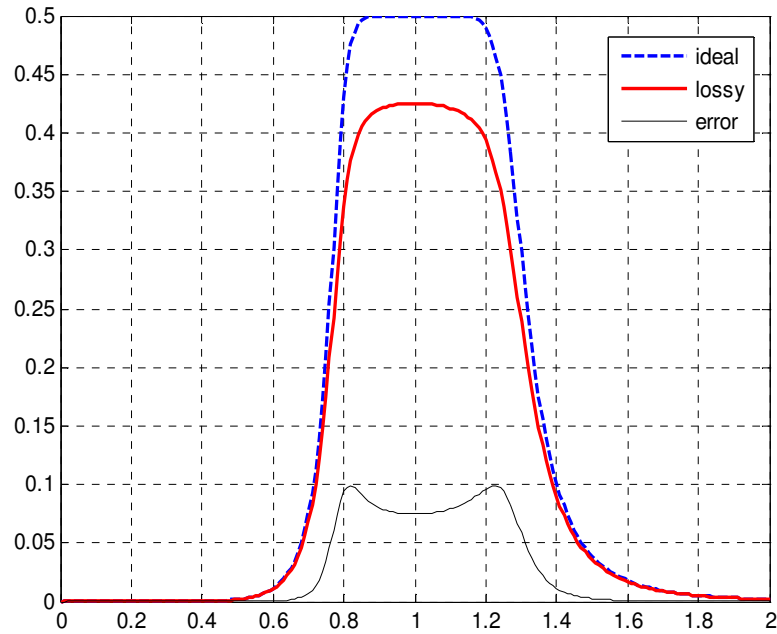


Figure 7-15: 8<sup>th</sup> Order Butterworth Band-pass Filter Frequency Plot – Before Optimization

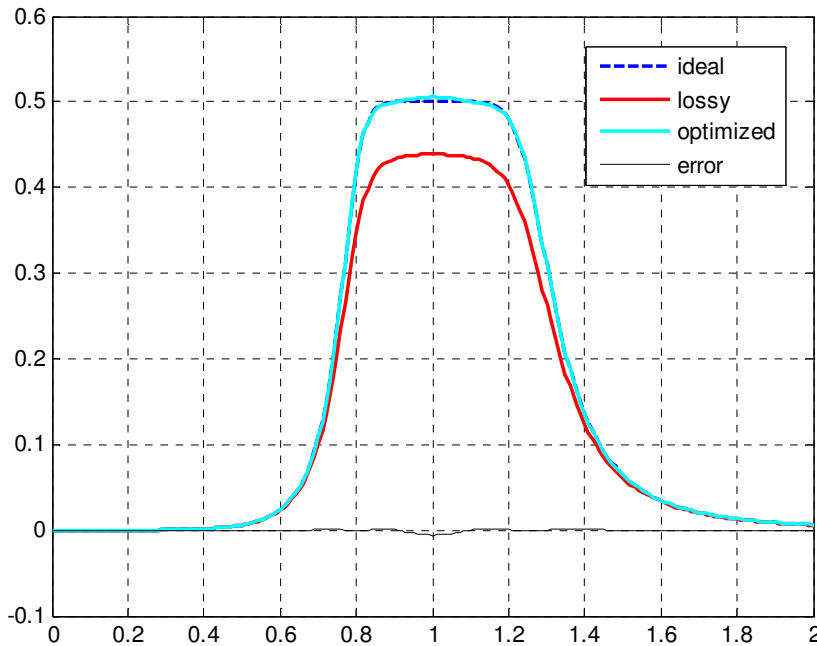


Figure 7-16: 8<sup>th</sup> Order Butterworth Band-pass Filter Frequency Plot – After Optimization

Once again as expected the LM Algorithm has provided better results, than the Nelder-Mead method, as shown in Figure 7-16. Also since this is a butterworth filter, as expected, better characteristics

are obtained when compared to the Bandpass filter because this filter does not have ripples, which makes it easier to optimize. The LM Algorithm has also matched the gain between the ideal response and the optimized response.

## 8 Conclusion

Three optimization methods have been presented. The first method used the SSE to minimize the difference between the poles of the ideal filter and the lossy filter. This procedure used the Nelder-Mead optimization routine and did produce results which were suitable for the Butterworth response, but for the Chebyshev filter, the results were impracticable. The second optimization method once again used the Nelder-Mead optimization routine, however the SSE was calculated from the difference between the coefficients of the ideal and lossy transfer functions. Both of these methods produced fitting results for low-pass Butterworth filters, but for the low-pass Chebyshev filter, the ripple was too hard to reproduce for both of these routines. In the case of more complex designs such as a Band-pass filter, the order of the filters between the ideal and lossy filters do not match, so these first two optimization methods can only be used for low-pass filters. The third optimization method uses the LM algorithm for the optimization, and the SSE was calculated from the difference between the ideal filter and lossy filters magnitude plot. The simulation results have shown that minimizing the differences between the filters' magnitude plots has been effective in evaluating the error, when used with the LM algorithm. This method has produced superb results for the low-pass and band-pass Butterworth filter, and more than adequate results when it comes to band-pass Chebyshev filters. The LM algorithm also matched the gain between the ideal response and the optimized response which also has made this algorithm superior than the Nelder-Mead optimization scheme.

In conclusion the filter synthesis algorithm and the optimization methods introduced in this thesis make it possible for a person to design a filter with non-ideal components, while achieving ideal-like characteristics. This thesis has also shown another unique synthesis algorithm showing how to design a ladder filter without the use of look up tables, and also how to calculate the transfer function both mathematically and symbolically by the use of MATLAB. Since most of the existing ladder synthesis

algorithms are based upon ideal components for the design, they are impractical for real world applications. Using the techniques developed in this thesis allows for one to design a low-pass Butterworth, Chebyshev, or Band-pass filter, with the use of lossy reactance elements, but yet achieve ideal-like results, which is more practical for real world applications.

## 9 References

1. A. M. Niknejad, "*Ladder Filter Design, Fabrication, and Measurement*", Berkeley Wireless Research Center, September 14, 2010
2. Wai-Kai Chen, "*Passive and Active Filters Theory and Implementations*" John Wiley and Sons, 1986
3. Rolf Schaumann and Mac E. Van Valkenburg "*Design of Analog Filters*" Oxford University Press, 2001
4. M. Jagiela and B.M. Wilamowski, "*A methodology of synthesis of lossy ladder filters*", University of Information Technology and Management, Autonomous Department of Electronics and Telecommunications, Rzeszow, Poland
5. John H. Mathews and Kurtis K. Fink, "*Numerical Methods Using Matlab*", 4th Edition, Prentice Hall, 2004
6. Karel Hajek and Jiri Sedlacek, "*GOAL-LOSSY RLC LADDER PROTOTYPES FOR SYNTHESIS OF ARC AND ASC FILTERS*" Military Academy Brno, Dpt. of Electrical engineering, Kounicova 65, 623 00 Brno, Czech Rep
7. Fuchang Gao and Lixing Han "Implementing the Nelder-Mead simplex algorithm with adaptive parameters", Received: 13 January 2010 © Springer Science+Business Media, LLC 2010
8. Pradeep Dandamudi, Timothy A. Brown, Xing Wu, Marcin Jagiela, and Bogdan M. Wilamowski, "*A Design Methodology of Lossy Transconductance Filters*", IECON 2012 – 38<sup>th</sup> Annual Conference on IEEE Industrial Electronics Society
9. Robert D. Koller and Bogdan Wilamowski, "*LADDER A Microcomputer Tool for Passive Filter Design and Simulation*", IEEE TRANSACTIONS ON EDUCATION, VOL. 39, NO. 4, NOVEMBER 1996
10. Nam Pham and Bogdan M. Wilamowski, "*Improved Nelder Mead's Simplex Method and Applications*" Electrical and Computer Engineering, Auburn University, Alabama, US
11. Koller, R. and B. M. Wilamowski, "*A Ladder Prototype Synthesis Algorithm*," proceedings of 35<sup>th</sup> IEEE Midwest Symposium on Circuits and Systems, Washington DC, USA, vol. 1, pp. 730-732, August 9-12, 1992
12. Bradley Bell and Alan Schmitzky, "*A generalization of the Gauss-Newton Method that solves extended Least Square Problems*", Department of Mathematics, University of Southern California, Los Angeles, CA, and Center for Bioengineering, University of Washington, Seattle, WA 98195
13. Ananth Ranganathan , "*The Levenberg-Marquardt Algorithm*", 8<sup>th</sup> June 2004
14. Bogdan M. Wilamowski and J. David Irwin "*The Industrial Electronics Handbook – Intelligent Systems*" CRC Press 2011
15. Alexander J. Casson and Esther Rodriguez-Villegas "*A Review and Modern Approach to LC Ladder Synthesis*", Circuits and Systems Research Group, Electrical and Electronic Engineering Department,

16. Stefan Niewiadomski, "*Filter Handbook*", Heinemann Newnes, 1989
17. Bogdan M. Wilamowski and J. David Irwin "*The Industrial Electronics Handbook – Fundamentals of Industrial Electronics*" CRC Press 2011
18. Henri Gavin, "*The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems*", Department of Civil and Environmental Engineering Duke University September 28, 2011
19. Gabor C. Themes and Sanjit K. Mitra, "*Modern Filter Theory and Design*", John Wiley and Sons 1973
20. J.A. Nelder and R Mead, "*A simplex method for function minimization*", The Computer Journal (1965) 7(4): 308-313



## 10 Appendices

### A Matlab input function: main.m

```
%% circuit 1 3rd Order Butterworth
% ideal=[0.999209;1.998418;0.999209];
% lossy=[0.999209;1.998418;0.999209];
% k=ones(1,length(ideal)); k=.1*k;
% perr=cc2zp(ideal,lossy,k);
% [x,fval,exitflag, output]=fminsearch(@(lossy) cc2zp(ideal,lossy,k),lossy,
optimset('TolFun',0.001,'MaxIter',1000))

%% circuit 2 3rd Order Chebyshev
ideal=[3.348729; 0.7117;3.348729];
lossy=[3.348729; 0.7117;3.348729];
k=ones(1,length(ideal)); k=.1*k;
perr=cc2zp(ideal,lossy,k);
[x,fval,exitflag, output]=fminsearch(@(lossy) cc2zp(ideal,lossy,k),lossy,
optimset('TolFun',0.001,'MaxIter',1000))

%% circuit 3 5th Order Chebyshev
% ideal=[3.481289; 0.761919; 4.537546; 0.761919; 3.481287];
% lossy=[3.481289; 0.761919; 4.537546; 0.761919; 3.481287];
% k=ones(1,length(ideal)); k=.1*k;
% perr=cc2zp(ideal,lossy,k);
% [x,fval,exitflag, output]=fminsearch(@(lossy) cc2zp(ideal,lossy,k),lossy,
optimset('TolFun',0.001,'MaxIter',1000))

%% circuit 4 5th Order Butterworth
% ideal=[0.617741; 1.617266; 1.99905; 1.617266; 0.617741];
% lossy=[0.617741; 1.617266; 1.99905; 1.617266; 0.617741];
% k=ones(1,length(ideal)); k=.1*k;
% perr=cc2zp(ideal,lossy,k);
% [x,fval,exitflag, output]=fminsearch(@(lossy)
cc2zp(ideal,lossy,k),lossy,optimset('TolFun',0.001,'MaxIter',1000))

%% circuit 5 4th Order Bandpass
% ideal=[1.961008; 0.509942;1.231108; 0.812276; 0.812279; 1.231108; 0.509942;
1.961008];
% lossy=[1.961008; 0.509942;1.231108; 0.812276; 0.812279; 1.231108; 0.509942;
1.961008];
% k=ones(1,length(ideal)); k=.1*k;
% perr=cc2zp(ideal,lossy,k);
% [x,fval,exitflag, output]=fminsearch(@(lossy)
cc2zp(ideal,lossy,k),lossy,optimset('TolFun',0.1,'MaxIter',1000))
```

## B Matlab Impedance Function: impedance.m

```
function [num,den]=Impedence(param)
type=param(1);
switch type
    case 0 % pure resistor
        R=param(2);
        num=[0 R];
        den=[0 1];
    case 1 % L with lossy resistor
        L=param(2);      Rl=param(3);
        num=[L Rl];
        den=[0 1];
    case 2 % C with lossy conductor
        C=param(2);      Gc=param(3);
        num=[0 1];
        den=[C Gc];
    case 3 % L series with C
        L=param(2);      Rl=param(3);
        C=param(4);      Gc=param(5);
        Lnum=[L Rl];     Lden=[0 1];
        Cnum=[0 1];     Cden=[C Gc];
        [num,den]=parallel(Lnum,Lden,Cnum,Cden);
    case 4 % L parallel with C
        L=param(2);      Rl=param(3);
        C=param(4);      Gc=param(5);
        Lnum=[L Rl];     Lden=[0 1];
        Cnum=[0 1];     Cden=[C Gc];
        [den,num]=parallel(Lden,Lnum,Cden,Cnum);
    %%%%% other cases
    %%%%% .....
end;
return;
```

## C Matlab Error and Plotting Function: error.m

```
function [err]=cc2zp(ideal,lossy,k)
format compact;

%% 3rd order filter
[num1,den1]=Impedence([1,ideal(1),0]);
[den2,num2]=Impedence([2,ideal(2),0]);
[num3,den3]=Impedence([1,ideal(3),0]);

nX_id=[0 1;num1;num2;num3;0 1];
dX_id=[0 1;den1;den2;den3;0 1];

[num1,den1]=Impedence([1,lossy(1),k(1)]);
[den2,num2]=Impedence([2,lossy(2),k(2)]);
[num3,den3]=Impedence([1,lossy(3),k(3)]);
nX_ls=[0 1;num1;num2;num3;0 1];
dX_ls=[0 1;den1;den2;den3;0 1];

%% 5th order filter
% [num1,den1]=Impedence([1,ideal(1),0]);
% [den2,num2]=Impedence([2,ideal(2),0]);
% [num3,den3]=Impedence([1,ideal(3),0]);
% [den4,num4]=Impedence([2,ideal(4),0]);
% [num5,den5]=Impedence([1,ideal(5),0]);
%
% nX_id=[0 1;num1;num2;num3;num4;num5;0 1];
% dX_id=[0 1;den1;den2;den3;den4;den5;0 1];
%
%
% [num1,den1]=Impedence([1,lossy(1),k(1)]);
% [den2,num2]=Impedence([2,lossy(2),k(2)]);
% [num3,den3]=Impedence([1,lossy(3),k(3)]);
% [den4,num4]=Impedence([2,lossy(4),k(4)]);
% [num5,den5]=Impedence([1,lossy(5),k(5)]);
%
% nX_ls=[0 1;num1;num2;num3;num4;num5;0 1];
% dX_ls=[0 1;den1;den2;den3;den4;den5;0 1];

%% 4th order BandPass
% [den1,num1]=Impedence([4,ideal(2),0,ideal(1),1]);
% [num2,den2]=Impedence([3,ideal(4),0,ideal(3),0]);
% [den3,num3]=Impedence([4,ideal(6),0,ideal(5),0]);
% [num4,den4]=Impedence([3,ideal(8),0,ideal(7),0]);
% nX_id=[num1;num2;num3;num4;0 0 1];
% dX_id=[den1;den2;den3;den4;0 0 1];
%
% [den1,num1]=Impedence([4,lossy(2),k(2),lossy(1),1+k(1)]);
% [num2,den2]=Impedence([3,lossy(4),k(4),lossy(3),k(3)]);
% [den3,num3]=Impedence([4,lossy(6),k(6),lossy(5),k(5)]);
% [num4,den4]=Impedence([3,lossy(8),k(8),lossy(7),k(7)]);
%
% nX_ls=[num1;num2;num3;num4;0 0 1];
% dX_ls=[den1;den2;den3;den4;0 0 1];
%% Plotting Roots

[num_id,den_id]=Ttra_poly(nX_id,dX_id);
[num_ls,den_ls]=Ttra_poly(nX_ls,dX_ls);

%Display Numerator and Denminator
```

```

dispf(num_id,9,4,'num_id');
dispf(den_id,9,4,'den_id');
disp(' ');

dispf(num_ls,9,4,'num_ls');
dispf(den_ls,9,4,'den_ls');

sys_id=tf(num_id,den_id);
sys_ls=tf(num_ls,den_ls);
[z_id,p_id,k_id]=tf2zp(num_id,den_id);
[z_ls,p_ls,k_ls]=tf2zp(num_ls,den_ls);
% Plots
figure(1); clf;
rz_id=real(z_id);rp_id=real(p_id);
rz_ls=real(z_ls);rp_ls=real(p_ls);
iz_id=imag(z_id);ip_id=imag(p_id);
iz_ls=imag(z_ls);ip_ls=imag(p_ls);

plot(rz_id,iz_id,'b*','MarkerSize',12); hold on
plot(rp_id,ip_id,'r*','MarkerSize',12); hold on
plot(rz_ls,iz_ls,'r*','MarkerSize',12); hold on
plot(rp_ls,ip_ls,'b*','MarkerSize',12); hold on
ax=axis; ax(1)=-1.5; axis(ax); grid on
legend ('Ideal Roots', 'Lossy Roots')

res=256;
cof=10^(1/res);

for i=1:4*res+1,
    if i==1, w(1)=0.01*j; else    w(i)=w(i-1)*cof; end
    y_id(i)=polyval(num_id,w(i))/polyval(den_id,w(i));
    y_ls(i)=polyval(num_ls,w(i))/polyval(den_ls,w(i));
end;
figure(2); clf;
semilogx(abs(w),abs(y_id),'b','LineWidth',2.2); hold on;
semilogx(abs(w),abs(y_ls),'r','LineWidth',2.2); hold on;
ax=axis; ax(4)=0.55; axis(ax); grid on
legend ('Ideal Filter', 'Lossy Filter')

figure(10); clf;
loglog(abs(w),abs(y_id),'b','LineWidth',2.2); hold on;
loglog(abs(w),abs(y_ls),'r','LineWidth',2.2); hold on;
ax=axis; ax(4)=0.55; axis(ax); grid on
legend ('Ideal Filter', 'Lossy Filter')

% bode(sys_id,'b',sys_ls,'r',P);

[z_id,p_id,k_id]=tf2zp(num_id,den_id);
[z_ls,p_ls,k_ls]=tf2zp(num_ls,den_ls);
nz_id=length(z_id);    np_id=length(p_id);
nz_ls=length(z_ls);    np_ls=length(p_ls);
if np_id < np_ls
    z_id=[zeros(np_ls-np_id,1);z_id];
    p_id=[zeros(np_ls-np_id,1);p_id];
end;

```

```

%% Plot Coefficients
figure(7); clf;
plot(num_id(end:-1:1), 'r*', 'MarkerSize', 12); hold on
plot(num_ls(end:-1:1), 'b*', 'MarkerSize', 12); hold on
title('Numerator');
xlabel('Order (s^n)');
ylabel('Coefficient Value');
legend('Ideal Filter', 'Lossy Filter')

figure(6); clf;
plot(den_id(end:-1:1), 'r*', 'MarkerSize', 12); hold on
plot(den_ls(end:-1:1), 'b*', 'MarkerSize', 12); hold on
title('Denominator');
xlabel('Order (s^n)');
ylabel('Coefficient Value');
legend('Ideal Coefficient', 'Lossy Coefficient')

%% Error calculated from Coef

for n=1:nz_ls
    err=err+abs(num_id(n)-num_ls(n))^2;
end;
for n=1:np_ls
    err=err+abs(den_id(n)-den_ls(n))^2;
end;
err
%% Error calculated from Roots
% for n=1:nz_ls
%     err=err+abs(z_id(n)-z_ls(n))^2;
% end;
% for n=1:np_ls
%     err=err+abs(p_id(n)-p_ls(n))^2;
% end;
% err

return;

```

## D Matlab Transfer Function: Ttra\_poly.m

```
function [num,den]=Ttra_poly(nX,dX)
```

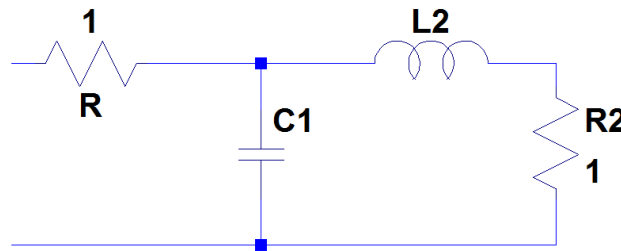
```
[ne,k]=size(nX);

nUpp=1; dUpp=1;
[nUp,dUp]=mulRat(nUpp,dUpp,nX(1,:),dX(1,:));
for i=3:ne+1
    [ntemp,dtemp]=mulRat(nUp,dUp,nX(i-1,:),dX(i-1,:));
    [nU,dU]=addRat(nUpp,dUpp,ntemp,dtemp);
    nUpp=nUp;dUpp=dUp;nUp=nU; dUp=dU;
end
num=dU; den=nU;
[num,den]=remrt(num,den);
return;
```

```
function [n,d]=mulRat(n1,d1,n2,d2)
n=conv(n1,n2);
d=conv(d1,d2);
d=rem0Poly(d); n=rem0Poly(n);
```

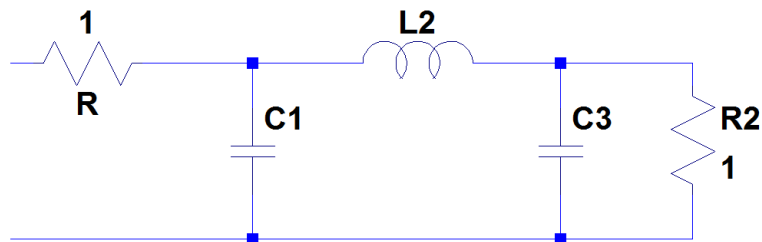
```
function [n,d]=addRat(n1,d1,n2,d2)
% add two rational functions
% n=d1*n2+d2*n1    d=d1*d2
% conv - multiplication of two polynomials
n=addPoly(conv(d1,n2),conv(d2,n1));
d=conv(d1,d2);
```

## E 2<sup>nd</sup> Order low-pass filter symbolic transfer function



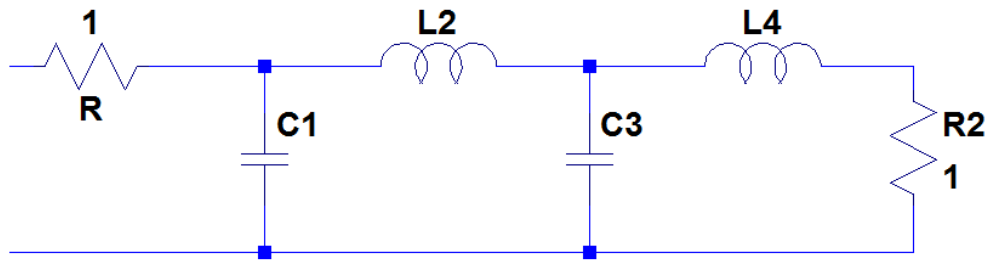
$s^2$	$s$	1
$L2 \cdot C1$	$L2 + C1$	2

## F 3<sup>rd</sup> Order low-pass filter symbolic transfer function



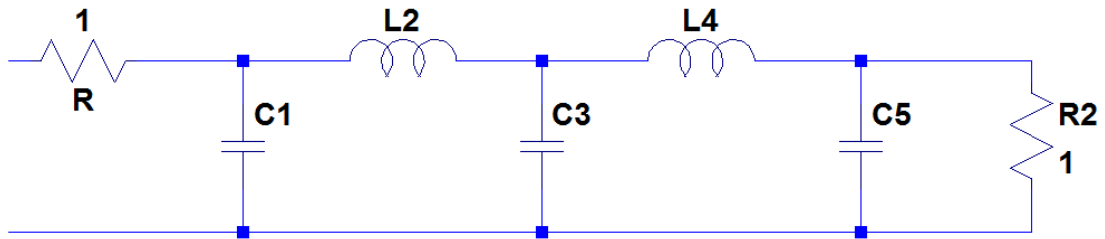
$s^3$	$s^2$	$s$	1
$C3 \cdot L2 \cdot C1$	$C3 \cdot L2 + L2 \cdot C1$	$C1 + C3 + L2$	2

### G 4<sup>th</sup> Order low-pass filter symbolic transfer function



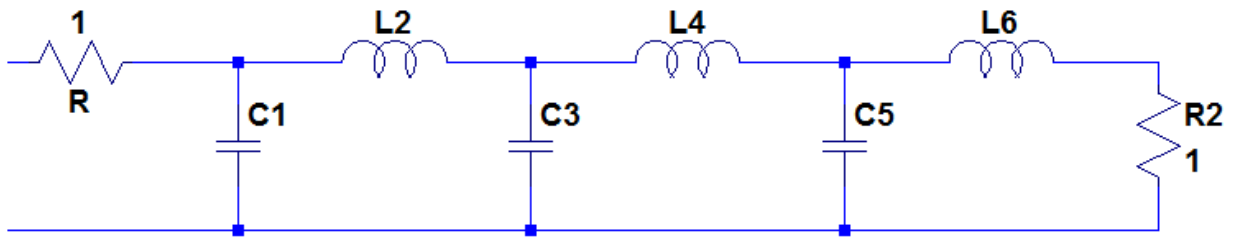
$s^4$	$s^3$	$s^2$	$s$	1
$L4 * C3 * L2 * C1$	$L4 * C3 * L2 + C3 * L2 * C1$	$L2 * C1 + L4 * C1 + L4 * C3 + C3 * L2$	$L2 + L4 + C1 + C3$	2

### H 5<sup>th</sup> Order low-pass filter symbolic transfer function



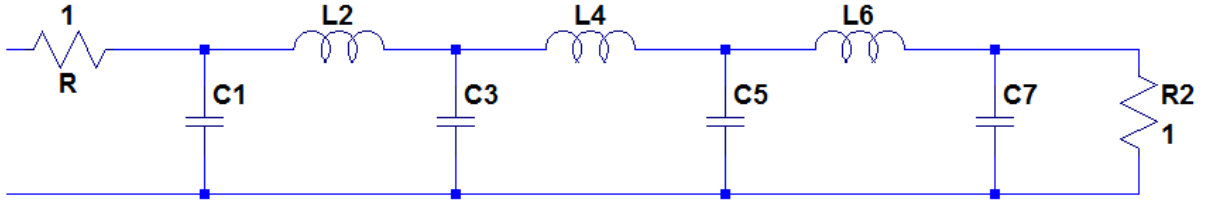
$s^5$	$s^4$	$s^3$	$s^2$	$s$	1
$C5 * L4 * C3 * L2 * C1$	$C5 * L4 * C3 * L2 + L4 * C3 * L2 * C1$	$C3 * L2 * C1 + C5 * L2 * C1 + C5 * L4 * C1 + C5 * L4 * C3 + L4 * C3 * L2$	$C3 * L2 + C5 * L2 + C5 * L4 + L2 * C1 + L4 * C1 + L4 * C3$	$C1 + C3 + C5 + L2 + L4$	2

### I 6<sup>th</sup> Order low-pass filter symbolic transfer function



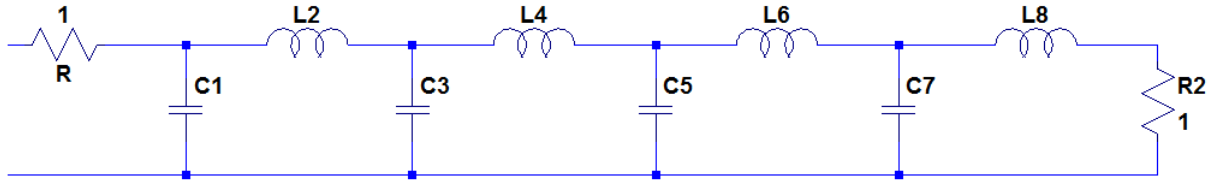
$s^6$	$s^5$	$s^4$	$s^3$	$s^2$	$s$	1
$L6 * C5 * L4 * C3 * L2 * C1$	$L6 * C5 * L4 * C3 * L2 + C5 * L4 * C3 * L2 * C1$	$L4 * C3 * L2 * C1 + L6 * C3 * L2 * C1 + L6 * C5 * L2 * C1 + L6 * C5 * L4 * C1 + L6 * C5 * L4 * C3 + C5 * L4 * C3 * L2$	$L4 * C3 * L2 + L6 * C3 * L2 + L6 * C5 * L2 + L6 * C5 * L4 + C3 * L2 * C1 + C5 * L2 * C1 + C5 * L4 * C1 + C5 * L4 * C3$	$L2 * C1 + L4 * C1 + L4 * C3 + L6 * C1 + L6 * C3 + L6 * C5 * C3 * L2 + C5 * L2 + C5 * L4$	$L2 + L4 + L6 + C1 + C3 + C5$	2

### J 7<sup>th</sup> Order low-pass filter symbolic transfer function



$s^7$	$s^6$	$s^5$	$s^4$	$s^3$	$s^2$	$s$	1
$C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + C7 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + C7 \cdot L6 \cdot C3 \cdot L2 \cdot C1 + L2 \cdot C1 + C7 \cdot L6 \cdot C3 \cdot L2 \cdot C1 + C7 \cdot L6 \cdot C3 \cdot L2 \cdot C1$	$C5 \cdot L4 \cdot C3 \cdot L2 + C7 \cdot L4 \cdot C3 \cdot L2 + C7 \cdot L6 \cdot C3 \cdot L2 + C7 \cdot L6 \cdot C5 \cdot L2 + C7 \cdot L6 \cdot C5 \cdot L4 + L4 \cdot C3 \cdot L2 + L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2$	$C3 \cdot L2 \cdot C1 + C5 \cdot L2 \cdot C1 + C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + C7 \cdot L2 \cdot C1 + C7 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + C7 \cdot L6 \cdot C5 \cdot L2 \cdot C1 + C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$C3 \cdot L2 + C5 \cdot L2 + C5 \cdot L4 + C7 \cdot L2 + C7 \cdot L4 + C7 \cdot L6 + L2 \cdot C1 + L4 \cdot C3 + L4 \cdot C3 \cdot L2 + L6 \cdot C3 + L6 \cdot C5$	$C1 + C3 + C5 + C7 + L2 + L4 + L6$	2

### K 8<sup>th</sup> Order low-pass filter symbolic transfer function



$s^8$	$s^7$	$s^6$	$s^5$	$s^4$	$s^3$	$s^2$	$s$	1
$L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2$	$L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$L4 \cdot C3 \cdot L2 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$L2 \cdot C1 + L4 \cdot C3 \cdot L2 \cdot C1 + L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1 + L8 \cdot C7 \cdot L6 \cdot C5 \cdot L4 \cdot C3 \cdot L2 \cdot C1$	$L2 + L4 + L6 + L8 + C1 + C3 + C5 + C7$	2



