A Predistortion Technique for OTA-C Filter Design

by

Pradeep Dandamudi

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
December 13, 2010

Keywords: ladder filters, synthesis, lossy, predistortion

Approved by

Bogdan M. Wilamowski, Chair, Professor of Electrical and Computer Engineering
Fa Foster Dai, Professor of Electrical and Computer Engineering
Thomas Baginski, Professor of Electrical and Computer Engineering

Abstract

The growing demand of continuous filters in portable electronic equipment, wireless receivers and continuous-time (CT) analog-to-digital converters (ADCs) has spawned a strong research interest from industry and academia. Gm -C filters are one of the most widely used continuous time filters because Gm -C filters are one of the fastest active integrators and enable low-power operation and tuning of the filter characteristics at higher frequencies. However issues concerning non-idealities which are parasitic conductance and capacitance in OTA and lossiness in reactance elements are less addressed. In this thesis a novel predistortion technique for synthesizing OTA-C filters is presented. The synthesis algorithm makes it possible to design a Butterworth or Chebychev filter which offsets non-idealities of the OTA. This approach substantially relieves the complexity of existing lossy ladder synthesis algorithms, and makes the synthesis of lossy ladder filters more reliable for practical applications.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr. Bogdan Wilamowski for his encouragement and guidance during the completion of this thesis. I would also like to thank my committee members, Dr. Foster Dai and Dr. Thomas Baginski for providing valuable feedback on my work. Finally, I would like to thank my parents Charles and Kezia Dandamudi for their love, encouragement and support over the years throughout my education, without which I would have likely never begun this journey.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Design of filters using passive ladder filters as a prototype for active filters, such as operational transconductance amplifiers and capacitors (OTA-C), has become more and more popular in recent years. OTA-C integrators are one of the fastest active integrators, with minimal or even zero number of internal nodes [23]. Therefore they are a suitable option in realizing high bandwidth integrators. The conventional way to design OTA-C filters is to design a passive ladder filter prototype and using signal flow graph methodology, transforming the prototype filter into GmC filter is easily obtainable. However, it is assumed that the reactance elements used to build passive ladder prototype and active OTA-C filter are lossless, i.e. Q (quality factor=0), and it is not possible to achieve this requirement. Every reactance element has considerable amount of lossy (finite vale of the Q-factor) and this lossiness significantly affects the frequency response of the filter. Existing classical synthesis algorithms [1, 3, 6, 9, 10, and 11] of ladder reactance filter do not consider lossiness of reactance elements. As a result frequency response of real filter differs from ideal prototype filter. Until recently [2] an algorithm has been proposed which considers the lossiness of reactance elements but it has experienced limited success due to computation complexity with respect to order of the filter and is the case with many other existing lossy synthesis algorithms [23,27, 28 and 29]. It is the purpose of this work to reduce the computation complexity when solving the problem of lossiness in the reactance element of ladder circuits using an algorithm proposed in chapter 3.

Although the proposed algorithm solves the lossy ladder circuit element values to achieve the prototype filter frequency response, when these ladder circuits are implemented using transconductance amplifiers, due to non-idealities associated with the Gm amplifiers such as finite output resistance, intrinsic input, output and Miller capacitances limits the performance of the filter at higher frequencies and high DC gain of the filter is difficult to achieve. Using the predistortion technique proposed in this work, it is however possible to map those non-idealities of the transconductance amplifier to achieve accurate DC gain of the filter.

1.1 Thesis Outline:

An overview of existing classical ladder synthesis algorithms is offered in Chapters 2. It includes a review of relevant concepts and brief look at most widely used algorithm. A novel algorithm for synthesizing lossy ladder filters is introduced in Chapter 3. The details of the algorithm are presented along with some experimental data. Transforming ladder circuits into OTA-C implementation using state equations is described in Chapter 4. The transconductance amplifier design process and its limitations is presented in Chapter 5. The OTA design will be developed in SPICE using CMOS 0.5 μm technology. The analysis of the filter is presented in Chapter 6, taking into account the non-ideal effects caused by parasitic capacitances and finite output impedances of the OTA. Finally, the thesis is concluded in Chapter 7.

Chapter 2

Ladder Prototype Synthesis

2.1 Background

Analog filters play a very significant role in every electronic system. There is a growing interest in continuous time filters, applications include hard disk read write systems [19, 20], digital video applications [21] and ultra wide band devices [22]. There are two different ways to implement analog filters. The first one uses cascade connection of second-order circuits. These types of filters are easy to design, but they are relatively sensitive to the tolerance of elements. The second method uses passive ladder prototypes and practical VLSI implementation of these prototypes need a series of integrators. This type of design is much sensitive to the tolerance of elements, but also it is more difficult to design.

Ladder filter structures are important for filter designs with modern circuit elements. Modern filter designs such as OTA-C filters, Switched Capacitor filters and Switched Current filters use LC ladder prototypes. An LC ladder prototype filter is build up of inductors and capacitors arranged series and shunt alternately, or of series or parallel tuned LC circuits. Predominantly, filters with Butterworth, Chebychev, inverse Chebychev and Cauer frequency responses are used. The classical synthesis process of ladder reactance filter begins with designing a low-pass prototype. For the most popular sorts of prototype low-pass filters, the values of elements have been derived and put into tables [7]. Thus, when designing a filter, there is no need to start from the desired frequency response and to recalculate them anew. Therefore, synthesis of a ladder reactance filter is very simple and requires relatively less computation, particularly when using tables of element values to obtain the prototype filter. There are several ways of implementation of such ladder filters.

The implementation of analog filter design requires calculation of poles, zeros and transfer function for that filter. All the necessary equations for calculating poles, zeros and transfer function for low-pass prototype filters are explained in the Chapter 2. For each type of approximation, poles and zeros are placed differently in s-plane. For high pass, band pass and band stop filters, given specifications are transformed to low-pass and low-pass prototype filter is again transformed to its original type. Quality factor and natural frequency of each complex conjugate poles are calculated.

2.2 Design of Low-Pass Prototype without Zeros:

The synthesis method begins by first transforming the filter specifications to a low-pass prototype, then synthesizing the low-pass prototype, and finally doing element transformations to get the proper filter form. Because low-pass transfer functions that have equal numerator and denominator orders are not implementable in a ladder form, special frequency transformations are added to the transfer function development. In order for a filter to be realizable, the order of the denominator must be greater than the order of the numerator. A special frequency transformation transforms the largest conjugate zero pair to infinity, thus reducing the order of the numerator by two and allowing the simulation of even order filters. For transfer functions without zeros, such as Butterworth, Chebychev and Bessel-Thompson, the approach used to determine the element values is the classical continued fraction expansion. This procedure is well documented [1] and is straight forward.

The algorithm follows a classical approach for Butterworth and Chebychev type filters. This involves using an auxiliary function [1] to find input impedance and then continued fractions to find the values of the elements.

Active circuit design of filters gives the transfer function of the required filter. Let T(s) be the transfer function. Auxiliary function [A(s)] found from transfer function and following sequence steps gives impedance function [Z(s)] of the circuit:

$$T(s) = \frac{A(s)A(-s)}{T_d(-s)}$$

1. Find $T_d(s)T_d(-s)$ and $T_n(s)T_n(-s)$

2. Find Auxiliary function A(s)A(-s)

$$A(s)A(-s) = 1 - 4\frac{R_{in}R_{out}}{(R_{in} + R_{out})^2}T(s)T(-s)$$

Where, R_{in} = input resistance and

R_{out} = output resistance

3. Find A(s) from A(s)A(-s)

$$A(s) = \frac{A_n(s)}{T_d(s)}$$

$A_n(s)$ is found by taking roots of A(s) A(-s) which lie in left half of plane.

4. Find Z(s)

$$Z(s) = R_{in}\frac{1 - A(s)}{1 + A(s)}$$

The impedance function $Z(s)$ gives the total impedance looking from the input terminal of the ladder circuit. Once impedance function is calculated, L and C values can be derived using classical continued fraction method.

The above procedure of designing low-pass prototype is implemented in MATLAB (code attached in appendix) and the results are shown below in various steps of the synthesis.

Example: Design of 5th order Chebychev filter:

In order to design a 5th order Chebychev filter ladder prototype, transfer function of the filter is calculated when necessary input parameters, as mentioned in Chapter 2, are provided.

With α_p =3dB, ω_p= 1rad/sec, α_s =50dB and ω_s= 2rad/sec, the transfer function of 5th order low pass filter is,

$$T(S) = \frac{0.06264858}{S^5 + 0.57450003S^4 + 1.41502514S^3 + 0.54893711S^2 + 0.40796631S^1 + 0.06264858}$$

And the impedance function: obtained from the simulation is,

$$Z(S) = \frac{9.1694S^4 + 2.6346S^3 + 8.7619S^2 + 1.5242S^1 + 1}{31.924S^5 + 9.171S^4 + 42.539S^3 + 8.7625S^2 + 11.5S^1 + 1}$$

Once calculating the impedance function, classical continued fraction is used to calculate the ladder element values. Element values can be scaled when frequency scaling is needed. The values obtained from the simulation are,

Element	Capacitance
L1	3.4816 F
C2	0.7619 F
L3	4.5375 F
C4	0.7618 F
L5	3.4814 F
Rin	1
Rout	1

Table 2.1: Element values for a 5th order low-pass Chebychev filter

The circuit implementation in SPICE is shown in Figure 2.1.

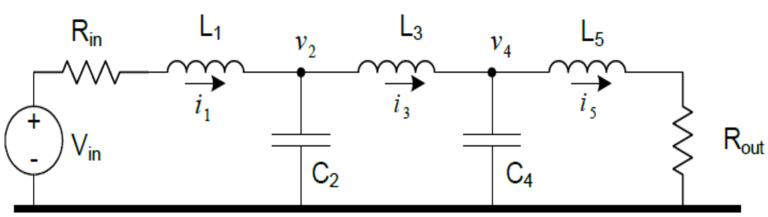

Figure 2.1: Ladder circuit implementation of 5th order low-pass Chebychev filter

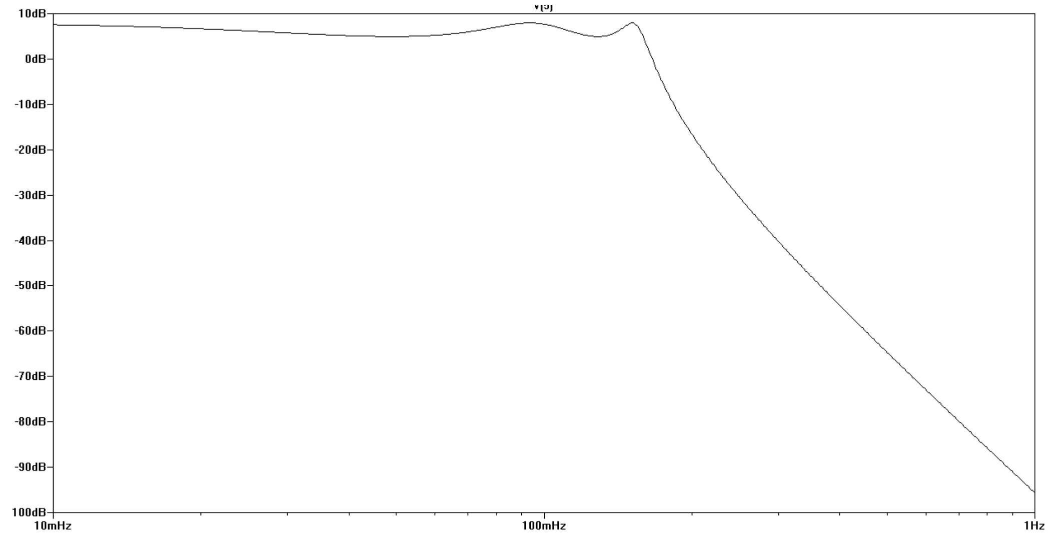

Figure 2.2: Frequency response of 5th order low-pass Chebychev filter

2.3 Design of Low-Pass Prototype with Zeros:

In the presence of zero, another transformation is introduced for filter without a maximum at zero, for example even order Cauer and Chebychev filters. For these filters, the first maximum, on the frequency scale is transformed to zero. The second transformation allows the synthesis of even-order Cauer and Chebychev filters with equal terminating resistance. The importance of this transformation comes from the fact that ladders with equal termination resistances are least sensitive to individual tolerances.

A special frequency transformation will transform the largest conjugate zero pair, $\mathrm{C_{a1}} = 1/\mathrm{sZ_A(s)}$ for $s=jz$ to infinity, thus reducing the order of the numerator by 2, while still maintaining the same basic response. The normalization frequency ω_N maps to the same frequency on both the ω –axis and the Ω-axis;

$$\Omega^2 = \frac{{\omega_N}^2 - {\omega_\infty}^2}{\omega^2 - {\omega_\infty}^2} \tag{2.1}$$

In the case of even-order Cauer and Chebychev filters, this transformation moves, eqn(next) the frequency of the first maximum (ω_0) to zero and allows the design of even-order Cauer and Chebychev filter with equal terminating resistances:

$$\Omega^2 = \frac{{\omega_N}^2 - {\omega_0}^2}{\omega^2 - {\omega_0}^2} \tag{2.2}$$

A generalized ladder circuit in presence of zeros is shown in Figure 2.3. Each zero pair (z) is represented by a resonant circuit consisting of inductor and capacitor in parallel. Resonant circuit with shunt capacitor (C_{a1}) constitutes complex conjugate pole and conjugate zero pair.

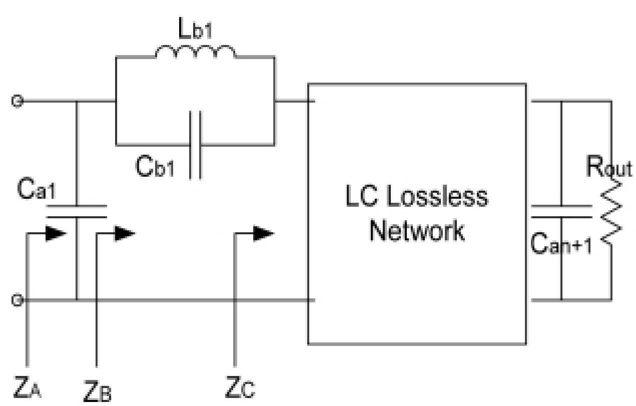

Figure 2.3: A generalized ladder circuit in presence of zeros

For extracting ladder circuit elements, following procedure to calculate elements from the impedance function.

1. Choose a value of zero (z) to be eliminated from impedance function.
2. Find value of C_{a1} solving equation,

$$Z_B(s) = \frac{N_A}{D_A - sC_{a1}N_A}$$
$$= \frac{a_{N-1}s^{N-1} + a_{N-2}s^{N-2} + \cdots + a_1 s + a_0}{k_N s^N + k_{N-1}s^{N-1} + k_{N-2}s^{N-2} + \cdots + k_1 s + k_0}$$
$$k_i = b_i - C_{a1}a_{i-1} \text{ for } (1 \le i \le N)\ (k_0 = b_0)$$
$$C_{a1} = \frac{b_1 z^{N-1} - b_3 z^{N-3} + b_5 z^{N-5} - \cdots}{a_0 z^{N-1} - a_2 z^{N-3} + a_4 z^{N-5} - \cdots}$$
$$C_{a1} = \frac{-(b_0 z^{N-1} - b_2 z^{N-3} + b_4 z^{N-5} - \cdots}{a_1 z^{N-3} - a_3 z^{N-5} + a_5 z^{N-7}}$$
$$z^2 = [L_{b1}C_{b1}]^{-1}$$

3. Choose a different zero and repeat steps 1 and 2 if the value of C_{a1} is not practically feasible.
4. Remove shunt capacitor and obtain Z_B.
5. Determine values D_C and L_{b1} from equations below.

$$Z_b = \frac{N_B}{D_B} = \frac{N_C}{D_C} + \frac{L_{b1}s}{z^2 s^2 + 1}$$
$$D_B = D_C(z^2 s^2 + 1)$$
$$L_{b1} = \frac{N_B}{D_C s}\Big|_{s \to \frac{j}{z}}$$

6. Find value of C_{b1} and Z_C.

7. Above steps are repeated for all resonant circuits (all zeros) and remaining circuit elements (like C_{aN+1}) are found using classical continued fraction method.

Example: Low-Pass Elliptic filter with α_p =1dB, ω_p= 1rad/sec, α_s =40dB and ω_s= 2rad/sec,

L&C element values obtained from the simulation are,

Element	Capacitance
L1	3.4816 F
C2	0.7619 F
L3	4.5375 F
C4	0.7618 F
L5	3.4814 F
Rin	1
Rout	1

Table 2.2: Element values for a 5th order low-pass Elliptic filter

The circuit implementation in SPICE is shown in Figure 2.4.

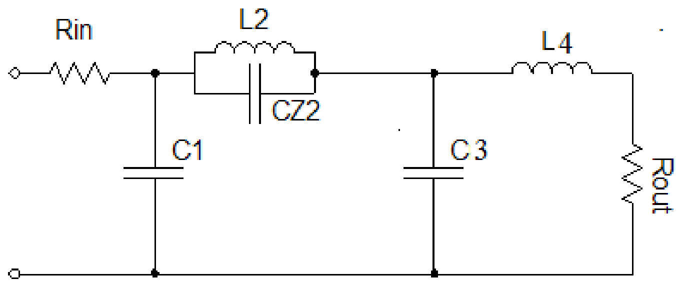

Figure 2.4: Ladder circuit implementation of 5th order low-pass Elliptic filter

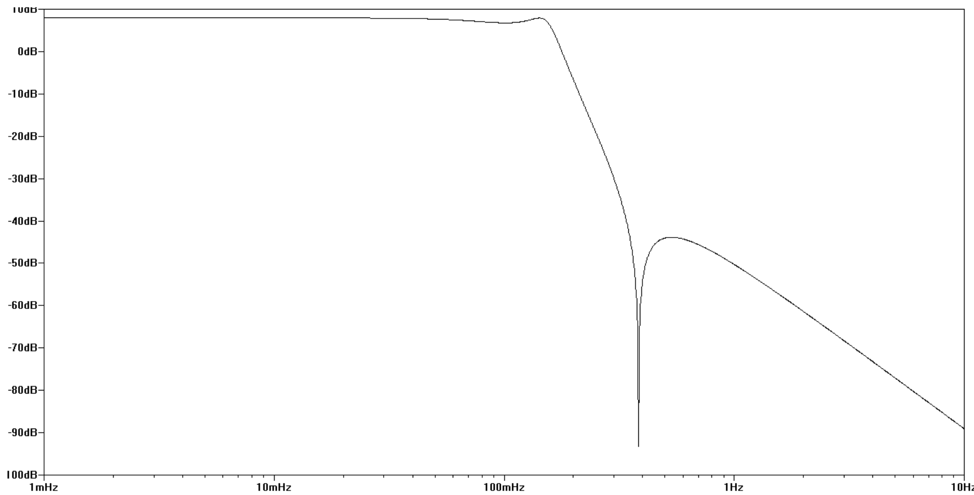

Figure 2.5: Frequency response of 5th order low-pass Elliptic filter

2.4 Transformation from Low-Pass to other type filters:

To obtain a high-pass, bandpass, or bandstop filter function from a low-pass prototype, the individual circuit elements of the prototype are replaced by other elements or sub-circuits as shown in Figure 2.6. Next, the values of all filter elements are scaled. There are two main purposes of scaling: to produce more practical values of filter elements, e.g. load resistance (magnitude scaling) and/or to move the filter passband towards the required range of frequencies (frequency scaling). This method is more useful in passive filter designs.

	LP	HP	BP	BS
Z	$\frac{L}{_0}$	$\frac{1}{L_0}$	$\frac{L}{B}$, $\frac{B}{L\,^2_0}$	$\frac{BL}{^2_0}$, $\frac{1}{LB}$
Y	$\frac{C}{_0}$	$\frac{1}{C_0}$	$\frac{B}{C\,^2_0}$, $\frac{C}{B}$	$\frac{1}{CB}$, $\frac{BC}{^2_0}$

Figure 2.6: Filter transformation from low-pass to other type filters

Another approach of filter transformation is by transforming transfer function of the low-pass prototype into the required form of the desired filter. Then a circuit is chosen to realize the new filter function. The approach is straight forward and explained detail in [9].

Chapter 3

Synthesis of Lossy Ladder Filters

3.1 Predistorting LC Ladder Networks

In all the existing algorithms of ladder filter synthesis [1, 3, 6, 9, 10, and 11], it is assumed that the reactance elements used to build passive ladder prototype are lossless, and it is not possible to achieve this requirement since actual reactance elements are lossy. Every reactance element has considerable amount of lossy and this lossiness significantly affects the frequency response of the filter. To determine the lossiness of the reactance element, the quality factor (Q-factor) is used. The use of the term lossiness means series resistance, measured in ohms, for inductors and leakage conductance, measured in siemens, for capacitors, as shown in Figure 3.1. The greater the lossiness of reactance elements used to build a filter, the greater the difference between the frequency response of the filter and the frequency response that the filter was intended to have.

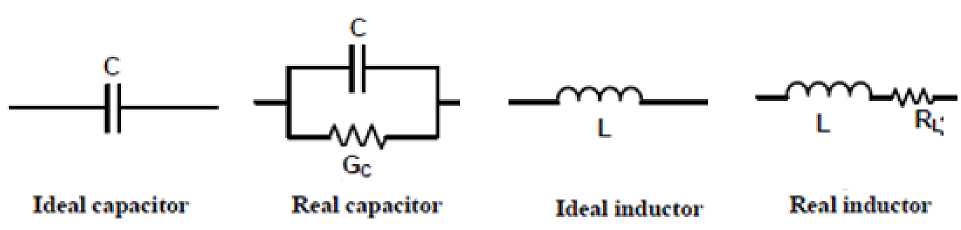

Figure 3.1: Models of real capacitor and real inductor.

A bulk of research has been dedicated to algorithms of ladder filter synthesis [1, 3, 6, and 9]. This research is done with the idea that reactance elements are ideal and lossiness of the reactance have no effect on the frequency response of the filter. One problem with this assumption is that it is impossible to accomplish ideal reactance elements. Another problem with this approach is that the lossiness of the reactance elements changes the location of poles on the complex plane, and this implies the deformation of the magnitude response of the filter. As a result the frequency response of the designed filter differs from the frequency response of the real filter. The greater the lossiness of reactance elements used to build a filter, the greater the difference between the frequency response of the filter and the frequency response that the filter was intended to have. Figure 3.2 and Figure 3.3 shows the comparison of magnitude Bode plots and poles location on the complex plane for a seventh-order Chebychev low-pass prototype filter and its realization with real, lossy elements.

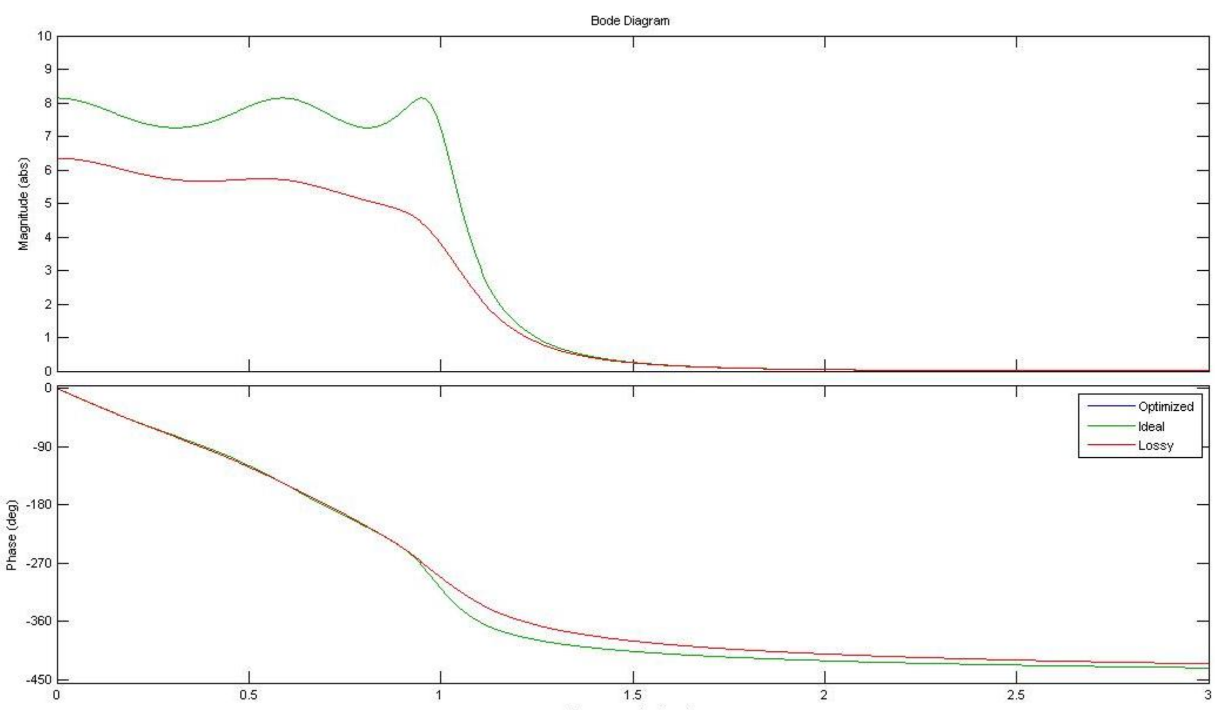

Figure 3.2: Magnitude Bode plots of the fifth-order low-pass Chebychev ladder filter

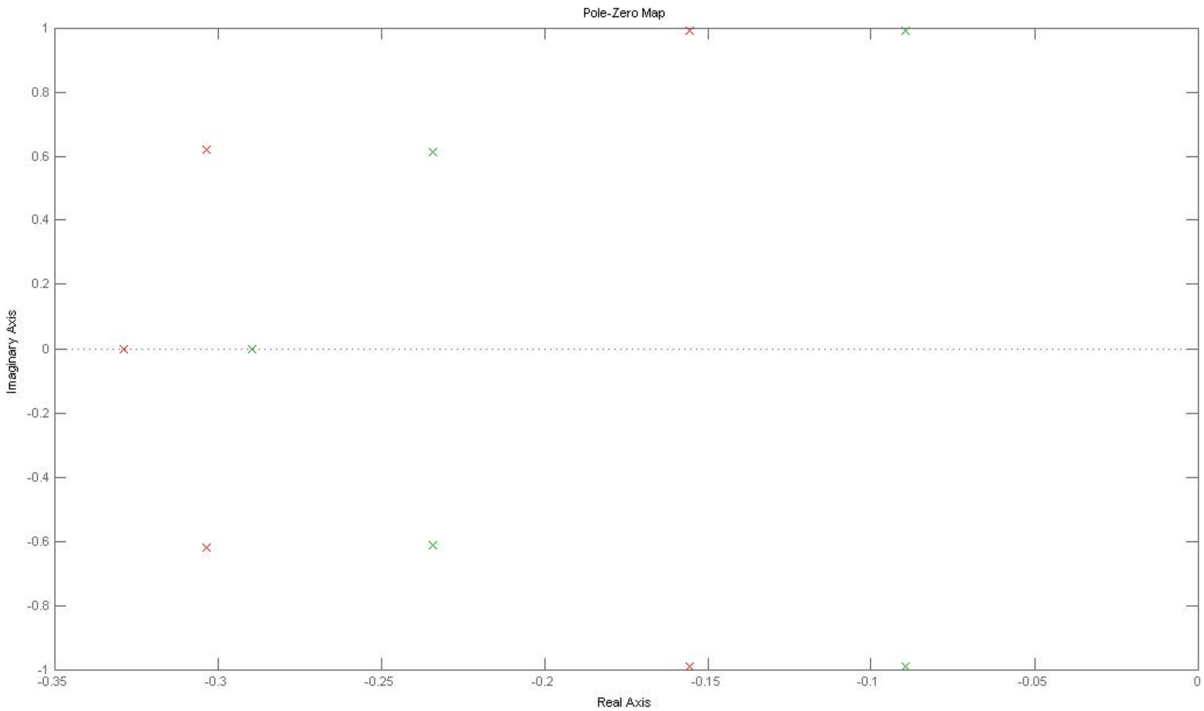

Figure 3.3: Pole location of the fifth-order low-pass Chebychev ladder filter transfer function

With some additional assumptions it is however possible to modify the values of capacitance and inductance of the reactance elements so as to move the poles back into place. A new method of predistorting LC ladder networks to offset the effect of resistive losses occurring in real components is described in this research work. Less research is devoted to this concept, and as such it is the crux of this work.

3.2 Methodology for Lossy ladder filter synthesis

The synthesis algorithm introduced in this chapter makes it possible to design a passive ladder, Butterworth or Chebychev filter with the use of lossy reactance elements. The filter obtained in this way can have exactly the same frequency response as the non-feasible prototype filter.

The lossy ladder synthesis algorithm follows three steps,

1. Determining the transfer functions for ideal, real and lossy ladder filters.
2. Estimating the error function.
3. Minimizing error function using optimization techniques.

One can easily write the transfer function of the circuit shown in Figure 3.4 using symbolic variables or in some other form in MATLAB and get the resulting transfer function coefficients at the output. The disadvantage using this approach is that the symbolic variable in state equations (3.4) affects the performance and as a result the time taken for computation process is increased. This case might even worsen as we go for higher order filters as more symbolic variables are used for the state equations. A new approach to write the transfer function without using symbolic variables or writing state equations is illustrated in this chapter.

After the transfer function of the ladder circuit is calculated, one can easily observe that the coefficients of the transfer function of any ladder prototype filter are made up of L&C elements of that circuit. Therefore, any change in L&C element values changes the transfer function of the ladder filter which in turn changes the location of poles of that transfer function. As shown in Figure 3.3, the poles of real filter do not match with that of the poles of ideal filter. If the poles of the real filter can be pushed towards and matched closely with the poles of prototype filter, then ideal filter response can be achieved in real filters. This concept is the backbone for the algorithm developed for lossy ladder synthesis. The optimization method described later in this chapter tries to reduce the error between the poles of real and prototype filters and gives the optimized L&C values that can be used in real filters to obtain ideal frequency response.

3.2.1 Calculating the transfer function of the ladder circuit

To calculate the transfer function of the real filter, consider a third order low-pass Chebychev filter shown in Figure 3.4.

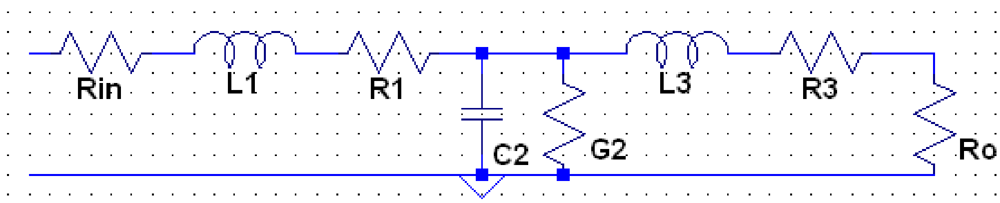

Figure 3.4: A third-order low-pass Chebychev ladder filter with real reactance elements.

State equations for the above circuit are,

$$I_3 = V_{out} / Ro$$

(3.1)

$$\begin{aligned} V_2 &= I_3(sL_3 + R_3) + V_{out} \\ &= (sL_3 / Ro + 1 + R_3 / Ro)V_{out} \end{aligned}$$

$$\begin{aligned} I_1 &= V_2(sC_2 + G_2) + I_3 \\ &= (s^2 L_3 C_2 / Ro + s[(1 + R_3 / Ro)C_2 + L_3 G_2 / Ro] + 1 + (1 + R_3 / Ro)G_2)V_{out} \end{aligned}$$

$$V_{in} = I_1 + I_1(sL_1 + R_1) + V_2 = I_1 + I_1 sL_1 + I_1 R_1 + V_2$$

Equation set (3.1) gives the relationship between state variables and reactance elements. These equations are arranged in a table such that each state variable is related to variable V_{out} as shown in table 3.1. The coefficients of the transfer function are calculated using tri-diagonal matrix system shown in table 3.1, 3.2 and 3.3.

	S^3	S^2	S^1	S^0
I_3	0	0	0	1/Ro
V_2	0	0	L_3/Ro	1+R_3/Ro
I_1	0	L_3C_2/Ro	(1+R_3/Ro)C_2+L_3G_2/Ro	1/Ro +(1+R_3/Ro)G_2
V_{in}	$L_3C_2 L_1$ /Ro	((1+R_3/Ro)C_2+L_3 G_2) L_1 + (L_3C_2/Ro)R1+ (L_3C_2Rin/Ro)	(1+(1+R_3/Ro)G_2) L_1 + ((1+R_3/Ro)C_2+L_3G_2/Ro)RinR1 + L3/Ro+Rin((1+R_3/Ro)C_2+ L_3G_2/Ro)	(1/Ro+(1+R_3/Ro)G_2)RinR1 + (1+R_3/Ro)+ 1/Ro +(1+R_3/Ro)G_2

Table 3.1: Relationship between the state variables I_3, V_2, I_1 and V_{in} with V_{out}

From Table 3.1, we find a relationship between the coefficients of one state variable with the coefficients of its previous variable. By using this relationship between the coefficients, a matrix can be formed with values same as coefficients shown in Table 3.1. Using this approach, state variables use for calculating transfer function is avoided and this approach has proved faster than other methods and efficient way when used in optimization process.

It can be observed that each row after the first two rows depends on the previous rows and the relationship between these rows can be derived as follows.

The process consists of three steps,

Step 1:

Initialize first two rows of the matrix with the coefficients of I3, V2, as shown below,

$$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{Ro} \\ 0 & 0 & \frac{L3}{Ro} & 1+\frac{R3}{Ro} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

And from the third row to end of the row, the process contains two steps.

Step 2:

From third row to the end of the matrix, any coefficient of I_1 is equal to the sum of,

1. Shifted coefficient of the previous variable (i.e. V_2) multiplied by active element of the previous variable (i.e. C_2 in this case) and,

2. Coefficient of variable before the previous variable (I_3) and coefficient of the previous variable (i.e. V_2) multiplied by lossy element of the previous variable (i.e. G_2 in this case).

i.e.,

First case shown in Table 3.2,

$$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{Ro} \\ 0 & 0 & \frac{L3}{Ro} & 1+\frac{R3}{Ro} \\ 0 & \frac{L3C2}{Ro} & \left(1+\frac{R3}{Ro}\right)C2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 3.2: Shows the relationship between third row and second row.

and second case is shown in Table 3.3,

$$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{Ro} \\ 0 & 0 & \frac{L3}{Ro} & 1+\frac{R3}{Ro} \\ 0 & \frac{L3C2}{Ro} & \left(1+\frac{R3}{Ro}\right)C2+\frac{L3G2}{Ro} & \frac{1}{Ro}+\left(1+\frac{R3}{Ro}\right)G2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 3.3: Shows the relationship between third row and first and second row.

step 2 is evaluated until all the row elements of the matrix are calculated.

Step 3:

After all the element values of the matrix are calculated using step 2, additional coefficients (Rin) need to be added to the last row of the matrix so that the coefficients accurately match with the table 3.1. The code for the transfer function calculation is written in MATLAB. See appendix A for the code.

Experimental Results:

For a third order Chebychev ladder filter, a 4 by 4 matrix is obtained (Table 3.4).

active_element = [3.348741 0.7117 3.348741]; r_loss=0.1; Rin=1; Ro=1

0	0	0	1.0000
0	0	3.3487	1.1000
0	2.3833	1.1177	1.1100
7.9811	6.3647	8.2954	2.3210

Table 3.4: Coefficients of state variables

As from the table 3.1, last row gives the coefficients of transfer function for Figure 3.1. Hence, last row of the matrix gives the coefficients of transfer function.

This process can be used for any higher order filter and transfer function coefficients can be easily obtained with less computation time.

3.2.2 Error function

As stated before, the concept behind the algorithm is to reduce the distance between poles of real and poles of prototype filter so that poles of the real filter will be very close to the prototype filter. For the lossy ladder synthesis program, the error function measures the difference of distance between each of the actual and real poles and finds the mean square value of these differences (as shown in Equation (3.2 & 3.3)) and passes this value to optimization technique for minimizing the distance between the poles of actual and real filter.

As seen from the Figure 3.6, poles of real filter are wide apart from prototype filter. Aim of the lossy ladder synthesis algorithm is to push the poles of the real filter close to the prototype filter. To monitor the distance between these poles, an error function is written that computes the difference of poles between real and prototype filters and passes the mean square value of the poles to nelder-mead method for optimization process.

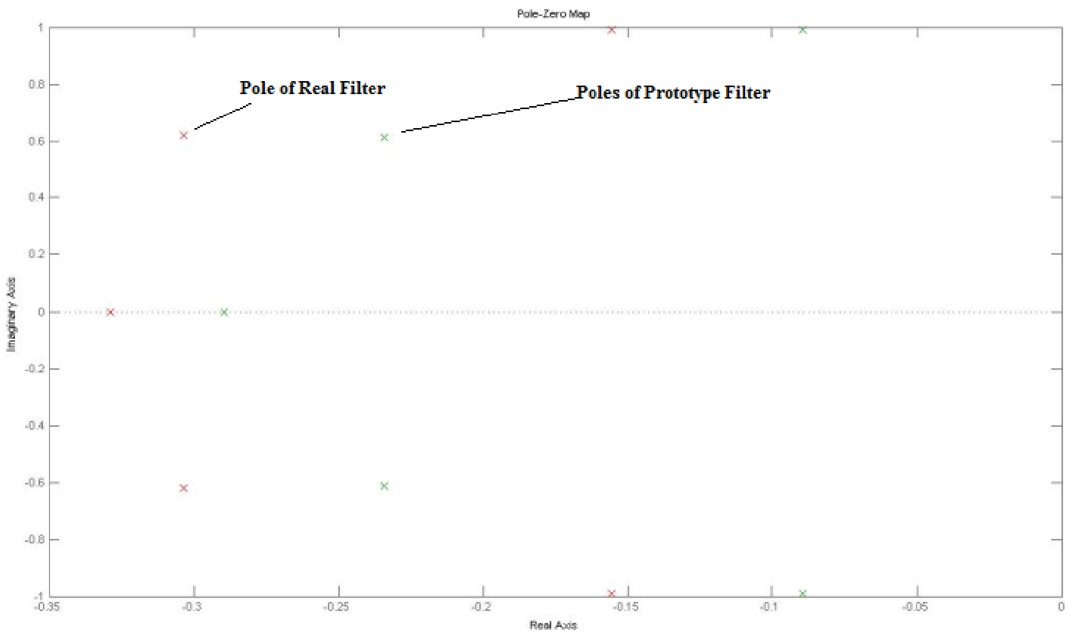

Figure 3.5: Pole locations of 5^{th} order real and prototype Chebychev filter

Error between actual and displaced poles can be found by,

Assume pl2 – poles of real filter and

pl1 – poles of prototype filter.

e= (pl2-pl1) (3.2)

And sum of mean squares of the error is,

E= (e'*e) (3.3)

The value E is passed as weight input to nelder-mead function which is an optimization technique that minimizes the error value, E, by choosing different possible values of L&C elements. See appendix B for the code.

3.2.3 A Numerical Optimization Technique: Nelder-Mead method

The Nelder-Mead finds a local minimum of a function (in this case error function) of several variables and gives the optimized values of variables that contributed to minimum value of a function. For lossy ladder synthesis program, nelder-mead finds the minimum point of error function using simplex method. The Nelder_Mead [4] algorithm seeks to solve the classical unconstrained optimization problem of a nonlinear function of several variables without derivatives. The algorithm is easy to visualize. The user supplies an initial set of points that represent solution estimates. The number of points supplied is one greater than the spatial dimension, so they form a "simplex" - in 2D, this is simply a triangle. Simplex is defined as a polyhedron which has one vertex more than the number of variables, which used in optimization. Each simplex vertex, correspond with a possible answer. Simplex algorithm tries to replace the worst vertex (vertex with greatest objective function value), with a better vertex which must be found. During the search, simplex performs four possible movements: reflection, expansion, contraction and shrinkage to find a vertex with a better value. New iterations continue until a termination criterion is obtained. Termination criterion is a simplex with a sufficient value or sufficiently small variations in objective function value.

Initial triangle is formed by evaluating given three vertices of a triangle from the function g(p,q) and reordering the subscripts such that x_0=(p1,q1) is worst vertex, x_1=(p2,q2) is best vertex and x_2=(p3,q3) is next best vertex. As shown in fig 3.6 a test point x' that is obtained by reflecting the triangle through the side $\overline{x_1x_2}$ is chosen. To determine x', the midpoint of the side $\overline{x_1x_2}$ is found and a line segment from x_0 to x' is drawn. If the function value at x' is smaller than the function value at x_0 then we have moved in the correct direction toward the minimum. The minimum may be a bit farther than the point x'. To find the new minimum, extend the line segment through ‘mid’ and x’ to the point x’’. The distance from x’ to x’’ is same as ‘mid’ to x’. If the function value at x’’ is less than the function value at x’, then a new vertex x’’ better tha x’ gives the minimum of the function.

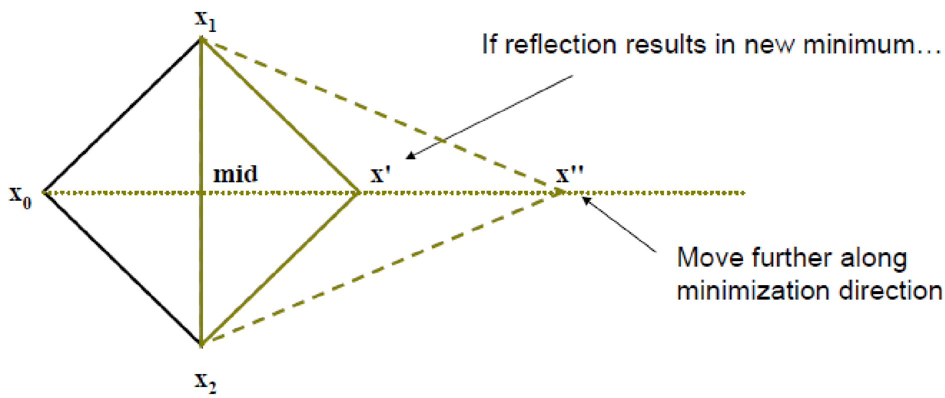

Figure 3.6: Reflection and Expansion: Triangle $x_0x_1x_2$ and point x’ and extended point x’’

Another point to be tested if the function values at R and W are same is by considering the midpoint of the line segment $\overline{x_0mid}$ as shown in figure3.7.

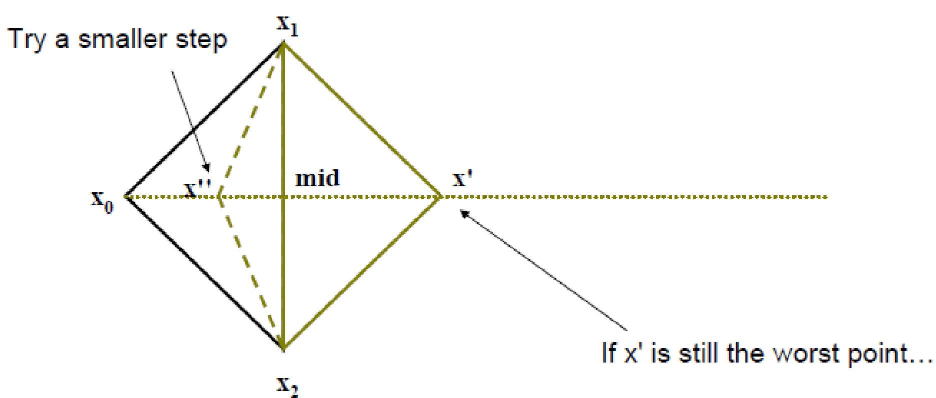

Figure 3.7: Contraction point for Nelder-Mead method

The Nelder-Mead algorithm is easy to understand and is well documented [5]. The Nelder-Mead method frequently gives significant improvements in the first few iterations and quickly produces quite satisfactory results. Also, the method typically requires only one or two function evaluations per iteration, except in shrink transformations. The method is often faster than other methods, especially those that require at least n function evaluations per iteration. See appendix C for nelder-mead optimization technique.

3.3 Simulation Results

3.3.1 Third Order Chebychev filter

Let us take into consideration a third order Chebychev low-pass prototype filter with specifications given in Table 3.5,

Element	Values
α_p	1 dB
α_s	30 dB
ω_p	1rad/sec
ω_s	2rad/sec

Table 3.5: 3rd order low-pass filter specifications

Having the filter designed using the classical synthesis algorithm without considering the reactance elements' lossiness, schematics as in Fig. 3.7 can be drawn with element values shown in table 3.6.

Element	Values
$L1$	2.023593
$C2$	0.994102
$L3$	2.023593
Rin	1
$Rout$	1

Table 3.6: 3rd order low-pass filter element values

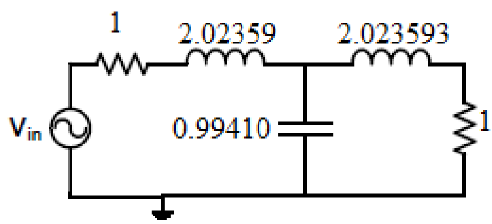

Figure 3.8: 3rd order low-pass ladder filter prototype

In order to include the lossiness, all of the inductors and capacitors in the schematic have to be replaced by their real models from Figure 3.8. In this way a circuit shown in Figure 3.9 can be obtained. Let us now set the example value of the lossiness RL1= RL2=GC2=0.1, although this value can be different for each element.

Element	Values
$L1$	2.0133
$C2$	1.5656
$L3$	0.97438

Rin	1
$Rout$	0.26841

Table 3.7: Lossy element values of 3rd order LPF with Q=10%

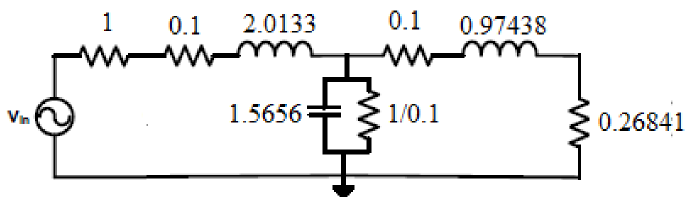

Figure 3.9: Ladder circuit of Fig. 3.5 with lossy elements

As shown in figure 3.10, using the proposed algorithm, one can easily obtain lossy element values for ladder circuits which gives a frequency response that matches exactly with that of ideal frequency response. Figure 3.11 shows the pole locations of ideal and real filters. The poles of the real filter are pushed towards the ideal filter until the optimization technique finds the minimum distance between poles. New element values obtained (table 3.7) using this method replaces ideal element values with lossy element values.

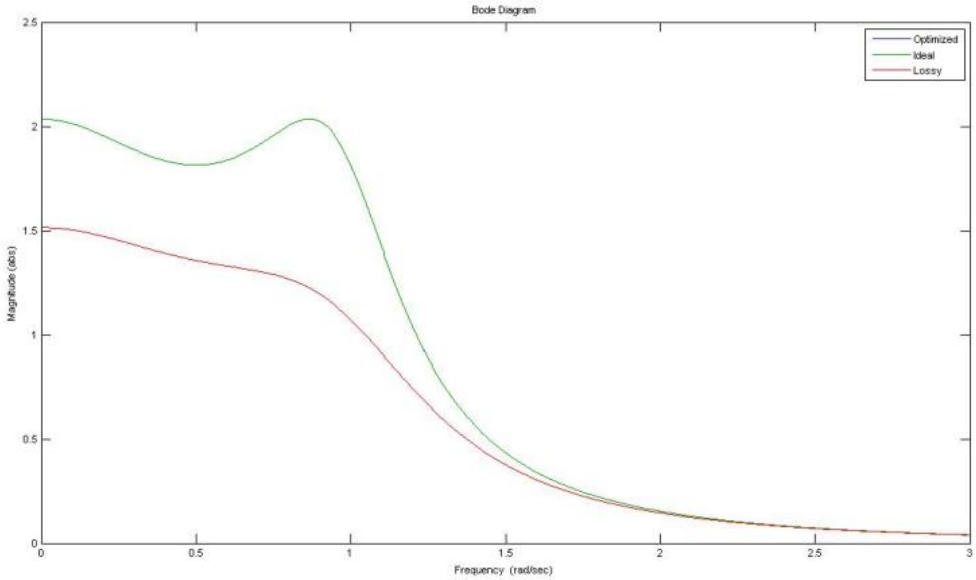

Figure 3.10: Magnitude Bode plots of 3rd order low-pass Chebychev ladder filter

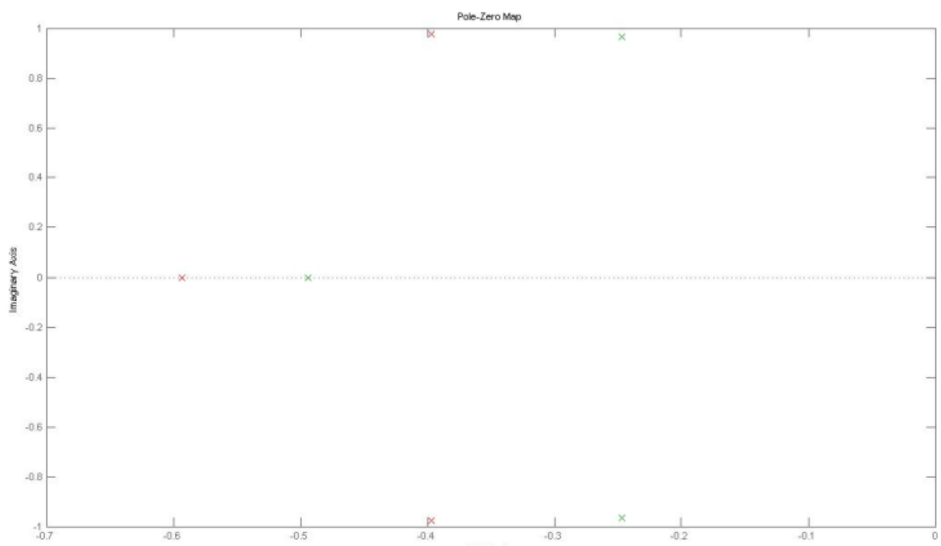

Figure 3.11: Pole locations of 3rd order real and prototype Chebychev filter

3.3.2 Fifth Order Chebychev Filter (1db pass-band ripple)

Another example of lossy ladder synthesis of a 5th order chebychev filter. The value of the lossiness chosen is *RL1=RL3=RL5=GC2=GC4=0.1*and specifications of the filter are shown in table 3.8.

Element	Values
α_p	1 dB
α_s	30 dB
ω_p	1rad/sec
ω_s	2rad/sec

Table 3.8: 5th order low-pass filter specifications

Ideal prototype ladder filter is shown in figure 3.12. The element values are obtained using the classical ladder synthesis approach (see appendix)

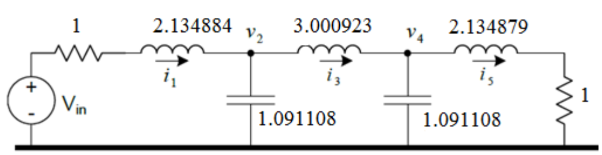

Figure 3.12: 5th order low-pass ladder filter prototype

Element	Value 1	Value 2	Value 3
$L1$	*1.6864*	*1.6445*	*1.6583*
$C2$	*1.5955*	*1.5530*	*1.5634*

$L3$	*2.5121*	*2.4023*	*2.4293*
$C4$	*1.6376*	*1.5695*	*1.5821*
$L5$	*1.0003*	*1.1405*	*1.1065*
Rin	1	1	1
$Rout$	*0.0210*	*0.0120*	*0.0163*

Table 3.9: Lossy element values of 5th order LPF with Q=10%

Different element values are obtained each time the program is executed in MATLAB, as shown in table 3.9 and these different values give the same magnitude response as of ideal filter response. The magnitude response for figure 3.13 is shown in figure 3.14 and figure 3.15 shows the pole locations of real and prototype filter.

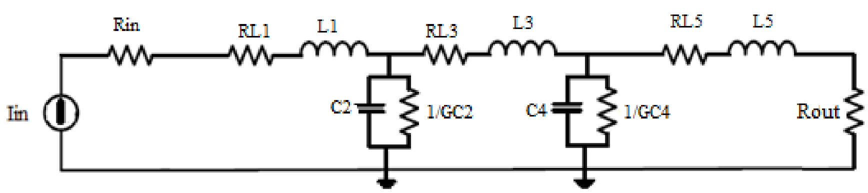

Figure 3.13: Ladder circuit of Fig. 3.11 with lossy elements

Very few algorithms take the lossiness introduced by the reactive elements into consideration for the synthesis process. Most of these algorithms are very complex to understand and these synthesis techniques become less effective as order of the filter is increased. Recently, [2] proposed a method to solve the problem of lossiness and derive the lossy element values. The method uses numerical analysis to solve the set of non linear equations (3.4) in order to determine the transfer function of the ladder circuit and tries to reduce the difference between coefficients of ideal and lossy transfer function. Disadvantage using that method is that as order of the filter increases it is difficult to solve those set of equations and obtain transfer function as a result complexity and computation time is increased. Using the algorithm proposed in this thesis one can easily obtain the lossy element values without regard to the order of the filter as a result complexity and computation time is improved.

$$i1 = \frac{1}{SL1 + 1/Rin + 1/RL1}(Vin - v2)$$

$$v2 = \frac{1}{SC2 + GC2}(i1 - i3)$$

$$i3 = \frac{1}{SL3+1/RL3}(v2 - v4) \tag{3.4}$$

$$v4 = \frac{1}{SC4 + GC4}(i3 - i5)$$

$$i5 = \frac{1}{SL5 + 1/Rout + 1/RL5}(v5)$$

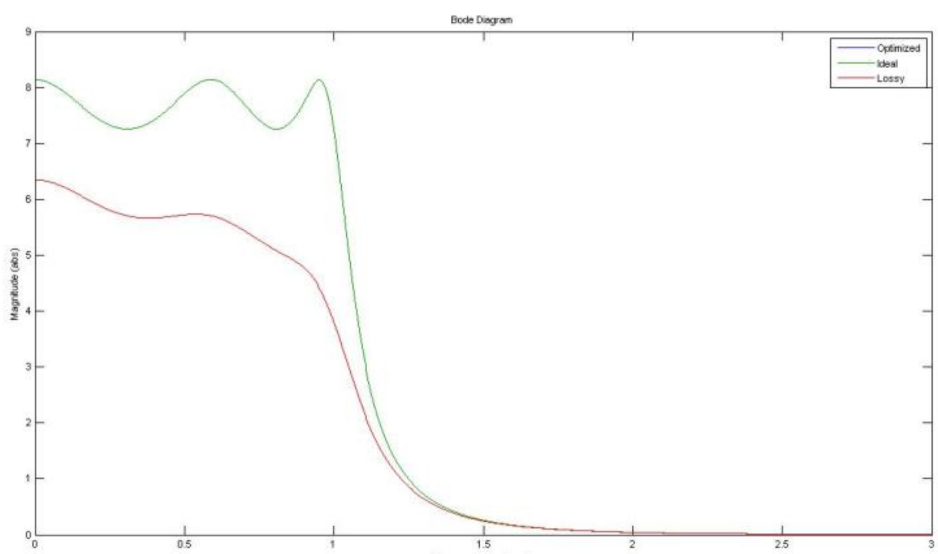

Figure 3.14: Magnitude Bode plots of 5th order low-pass Chebychev ladder filter

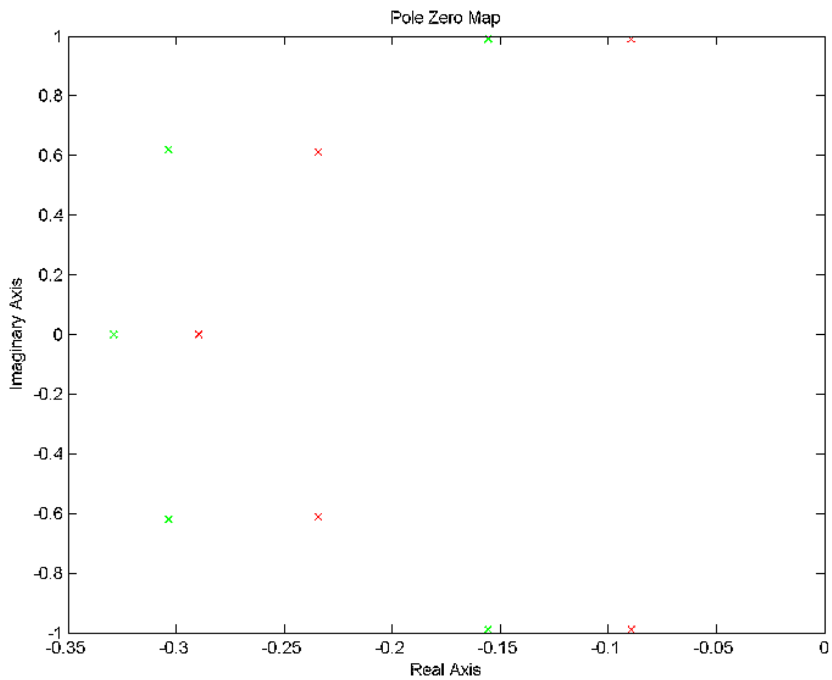

Figure 3.15: Pole locations of 5th order real and prototype Chebychev filter

3.4 The relationship between the amount of lossiness and positive value of resistance R2:

The limitation to the algorithm is the amount of lossiness that can be introduced in the synthesis process. Table 3.10 shows the relationship between order of the filter and amount of lossiness that can be introduced in the ladder circuit. For example for a 3rd order filter with 0.1 dB pass band ripple maximum amount of lossiness that can be introduced is Q=0.3. Beyond this value the lossy synthesis shows negative resistances. In order to get positive element values one has to limit their lossiness as given in table 3.10.

	3^{rd} (order)	4^{th}	5^{th}	7^{th}
0.1dB (pass band ripple)	30%	25%	16%	10%
1dB	20%	16%	10%	6%
3dB	15%	10%	6%	1%

Table 3.10: Relationship showing order of filter and amount of lossiness

From table 3.10, the value of Q decreases as the order of filter increases and similarly Q decreases as the value of pass band ripple is increased.

Chapter 4

Synthesis of OTA-C Filters Using Ladder Filter Prototype

4.1 Background

Transconductance filters (OTA-C) gained wide popularity due to its wide range of uses in the field of analog filters and the need for high quality analog filters implemented on standard CMOS integrated circuit made them have an edge over classical amplifier design. Because of its ease of use [11, 23] and advantages over classical amplifier design, given the ladder circuit of any order, an equivalent OTA-C filter can be easily synthesized.

There are two main techniques to transform a ladder circuit into its OTA-C implementation. The first technique known as element replacement, involves using capacitors directly and substituting inductors with an equivalent circuit thus has inductive input impedance. The other method is referred to as operational simulation, or signal flow graph method. In operational simulation, the node voltages and mesh currents of the ladder circuit are simulated by summations and the ladder elements are simulated by integration operators.

4.2 Low Pass filter Synthesis

4.2.1 LC Ladder Circuit

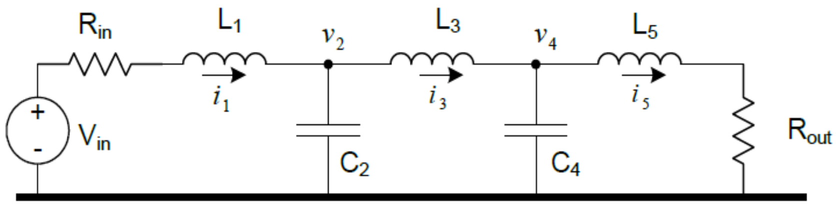

Figure 4.1: Low pass filter LC implementation

The implementation of a fifth order Chebychev filter designed for minimum capacitance is shown in Figure 4.1. For simplicity values for the passive elements are chosen to set the cut-off frequency at 1Hz and are shown in table 4.1

Element	Capacitance
L1	3.4816 F
C2	0.7619 F
L3	4.5375 F
C4	0.7618 F
L5	3.4814 F

Table 4.1: Capacitor values for low pass filter prototype

The element values were calculated using the software mentioned in Chapter 2.

4.2.2 OTA-C Transformation:

From Figure 4.1, one can easily write the state space equation for this particular circuit using current and voltage conventions defined in the figure.

$$i1 = \frac{1}{SL1 + Rin}(Vin - v2)$$

$$v2 = \frac{1}{SC2}(i1 - i3)$$

$$i3 = \frac{1}{SL3}(v2 - v4) \qquad (4.1)$$

$$v4 = \frac{1}{SC4}(i3 - i5)$$

$$i5 = \frac{1}{SL5 + Rout}v5$$

If the equations set (4.1) are used to implement an OTA-C filter, the circuit would be as shown in Figure 4.2.

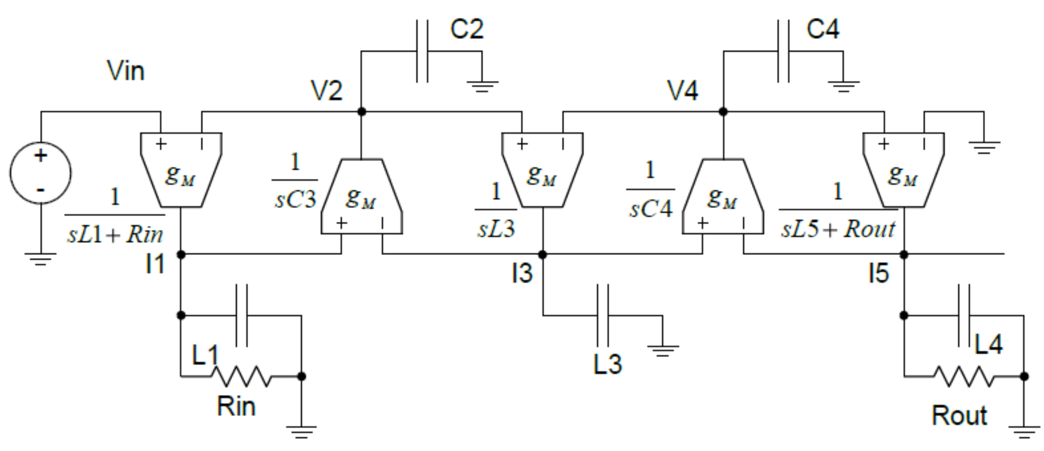

Figure 4.2: OTA-C filter implementation

Analyzing the circuit in Figure 4.2 with the OTA output voltages as state variables, the state equations become those shown in equation set (4.2) assuming an equal transconductance for each OTA.

$$v1 = \frac{gm}{SL1 + 1/Rin}(Vin - v2)$$

$$v2 = \frac{gm}{SC2}(v1 - v3)$$

$$v3 = \frac{gm}{SL3}(v2 - v4) \tag{4.2}$$

$$v4 = \frac{gm}{SC4}(v3 - v5)$$

$$v5 = \frac{gm}{SL5 + 1/Rout}v5$$

From comparing equations (4.1) and (4.2), the passive component values (capacitors) must be scaled by the value of gm and the resistive component values (Rin and Rout) must be scaled by the value of 1/gm to achieve the same filter critical frequency as shown in (4.3).

$$R = R*rm \tag{4.3}$$

$$C = C*gm$$

Figure 4.2 shows the OTA-C circuit. Using ideal components in SPICE, the frequency response of the circuit is exactly same as the LC ladder response shown in Figure 4.3.

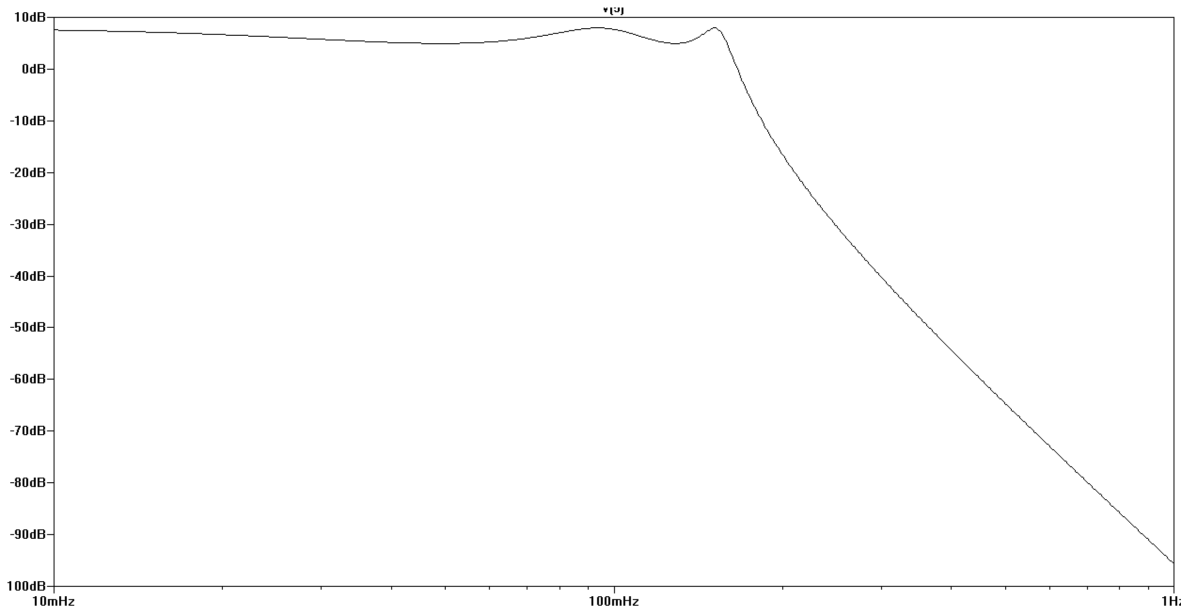

Figure 4.3: Frequency response of 5th order prototype Chebychev LPF using ideal OTA's.

Chapter 5

Implementing OTA's in VLSI

5.1 Background

The OTA capacitor (OTA-C) filters have several advantages over the more typical operational amplifier configurations. These advantages include: electronic filter tuning, high frequency of operation and fully integrated filter design. Electronic tuning is necessary to overcome process variations. A generally filter design can be used in many applications if the filter's center frequency can be varied over a wide band of frequencies without actually scaling the reactance element values. In OTA-C filters, electronic filter tuning can be achieved by changing the parameter transconductance (gm) of an OTA. Since transconductance of the filter is proportional to current or voltage bias level, frequency response can be modified by changing the bias level of the OTA.

High frequency OTA design can be achieved more easily than a high frequency operational amplifier. This is due to the output stage of a typical operational amplifier is not required in OTA. However, high frequency filter tuning in OTA-C filter is not achieved due to limitation in the OTA design and also due to the reactance elements are lossy. Additional research needs to be done in this area to analyze the effect of these limitations in order to achieve high frequency tuning.

Despite their advantages, OTA-C filters are often perceived as hard to design because literature on the subject is often unnecessarily complicated. A bulk of research has been dedicated to lower order OTA-C filter structures [12] – [13]. Due to loss of generality and sensitivity of element values, this approach will have serious impacts on the filter response. A better approach is to develop equations for the circuit to be implemented and match those equations with only the OTA and capacitors as building blocks. Less research is devoted to this concept [14] – [15], until recently a simplified approach to design any order filters has been proposed [16].

5.2 Approach

In practical applications, OTA have non-ideal characteristics which in turn affects the performance of the filter. In order to decrease the non-ideal characteristics of the OTA, it is important to consider the end use of the circuit. In the case of the filter application, the finite output resistance of the OTA can seriously degrade the shape of the desired filter, especially at higher quality factors [17]. Due to this fact, high output impedance must be a focus in the OTA design process.

5.3 Implementation

5.3.1 OTA with current mirrors:

A cascode structure gives higher output impedance without sacrificing transconductance when compared to that of a two input differential pair OTA. The OTA is shown in Figure 5.1. The current, i, at the ouput node C depends on the difference in the two input node voltages and the transconductance is determined by transistor M1 or M2. Equivalent model of Figure 5.1 is shown in Figure 5.2 with ideal OTA, parasitic capacitace and output resistance.

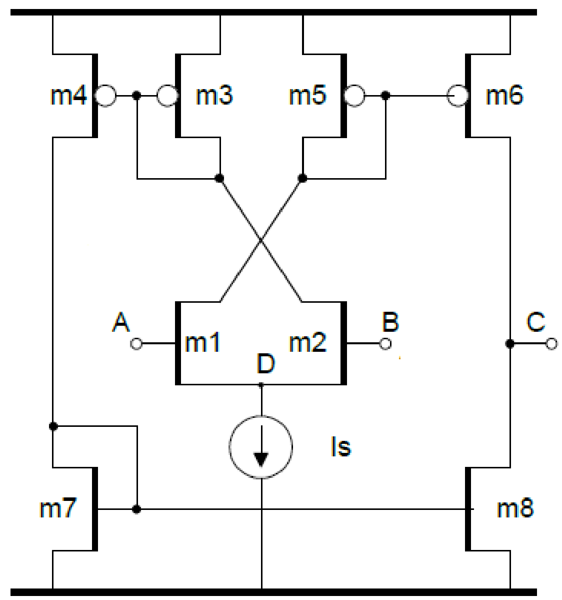

Figure 5.1: Simple OTA implementation

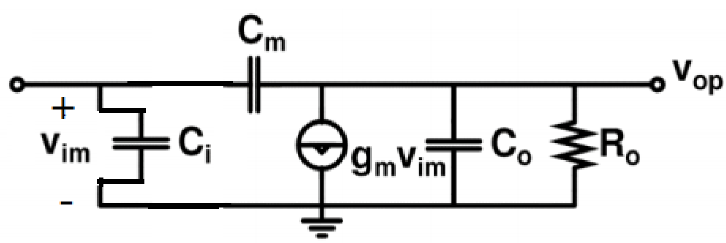

Figure 5.2: Equivalent of Fig.5.1 with ideal OTA, parasitic capacitances and output resistance.

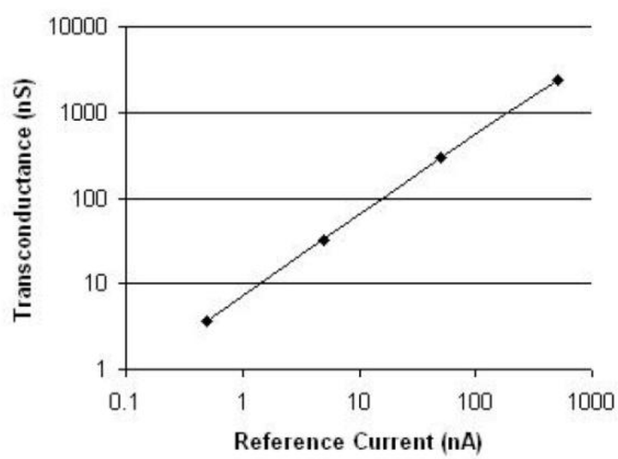

Figure 5.3: Transconductance vs. reference current for the OTA shown in Figure 5.1

Simulating in SPICE using 0.5um process models, transconductance varies with current as shown in Figure 5.3. The input transistors are operated in subthreshold conduction. This is done to keep transconductance linearly tunable and to achieve larger output resistances. If the amplifier were operated outside the subthreshold conduction region, the transconductance would be described by equation (5.1).

$$g_m = \sqrt{2k_n I_d} \tag{5.1}$$

However, in subthreshold conduction the transconductance follows equation (5.2), where η in this case is around two.

$$g_m = \frac{I_d}{\eta V_t} \tag{5.2}$$

The drawback is that the amplifier will not work as well at higher frequencies. However, to achieve higher quality factors the tradeoff is necessary. For filter design, a large transconductance value is not as important as a large output resistance. Figure 5.4 shows how the output resistance of the differential pair varies with bias current which follows equation (5.3).

$$R_{out} = r_{o-pmos} || r_{o-nmos} \tag{5.3}$$

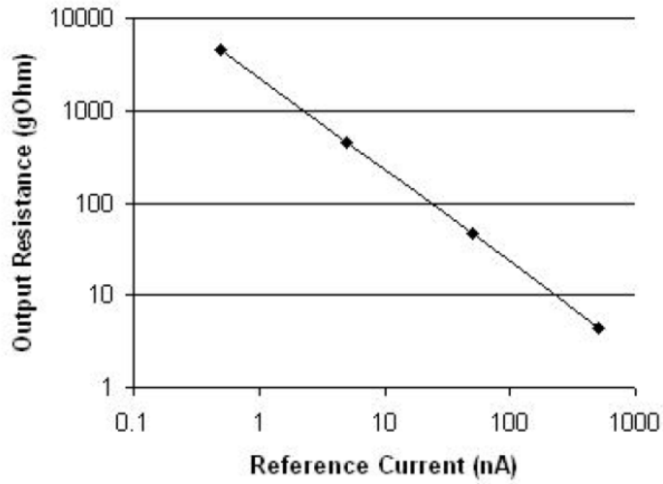

Figure 5.4: Output Resistance vs. reference current for the OTA shown in Figure 5.1

The drawback in OTA design comes from the impact of the non-dominant poles. In this case, the non-dominant poles of interest are located at the input side of the current mirrors [18]. If the non-dominant poles are too close to the dominant pole, the filter transfer function can be distorted at higher quality factors or higher frequencies [17]. Figure 5.5 shows a plot of the transconductance verses frequency. The constant roll off shown indicates the non-dominant poles are very close to the dominant one.

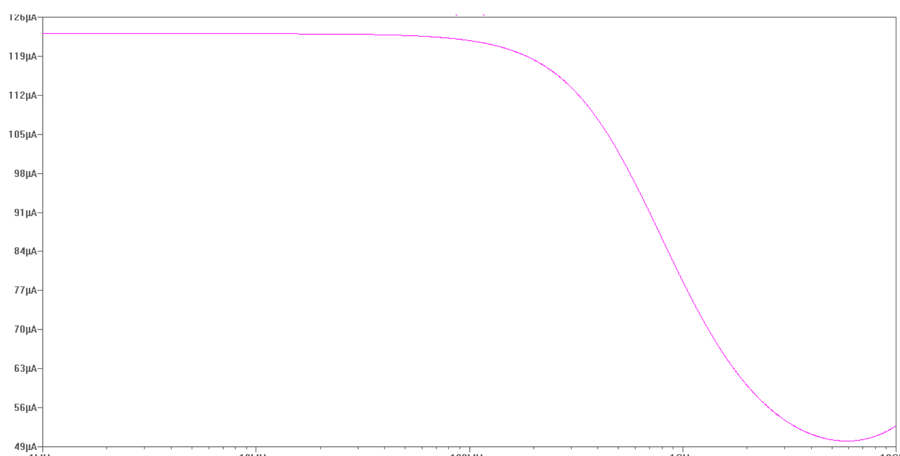

Figure 5.5: AC sweep of transconductance for OTA in Figure 5.1

5.4 Non-idealities in OTA

The small-signal model of the transconductance element used in this design example is shown in Fig. 5.6. The model includes intrinsic input, output and Miller capacitors and also finite output resistance. The losses in GmC-filters are caused by the finite output resistances of the transconductors. The component values in the model scale linearly with the value of the transconductance. The OTA has no internal nodes in order to simplify the analysis and mapping of the GmC-filter [24].

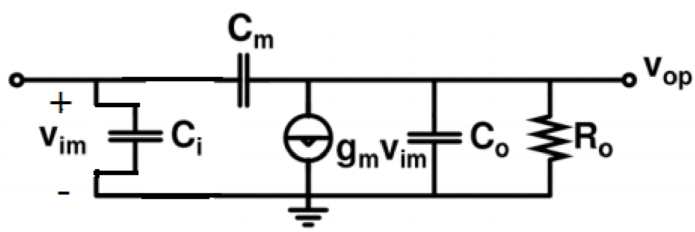

Figure 5.6 OTA small signal model.

Among these non- idealities, a Gain Bandwidth product generally has the most drastic effect on frequency characteristics, an effect due to an excess phase shift of an integrator [25]. And also the finite output resistance deeply effects the DC gain of the transconductance as shown in (5.4)

$$Q_{OTA} = A_{DC} = g_m R_o = R_o / r_m \qquad (5.4)$$

As shown in [26], non-ideal amplifier input causes parasitic zeros in the filter transfer function, and thus imposes fundamental limitations on the realizable pole frequency.

All the existing predistortion techniques use complex numerical computation procedure to solve a system of state equations in order to find the values for reactance elements of a particular filter. As the order of filter increases, these computation time involved in solving these state equations increases as the equations become more and more complex and limitations on the lossy values increases. As a result most of the existing procedures are not efficient when it comes to finding the reactance elements values of the GmC filters.

The predistortion technique presented in later chapters employ an optimization technique to obtain the active element for lossy ladder filters and by mapping lossy active element values with OTA losses, the synthesis procedure tries to compensate the non-idealities in the actual filter realization as precisely as possible. As a result an accurate predistortion result can be achieved with the techniques proposed in this research work.

Chapter 6

Predistorting OTA-C Filters

As illustrated in previous chapters that the lossy elements in LC ladder networks significantly degrades the performance of the filter prototype and as a result the frequency response of the real filter differs widely from frequency response that the filter was intended to have. The proposed (in previous chapter) lossy filter synthesis algorithm tries to offset the effect of resistive losses occurring in reactive components by taking lossiness of the reactance elements into the filter synthesis. A filter obtained this way can have exactly the same frequency response as the non-feasible filter designed using the classical algorithm and built with ideal reactance elements.

The conventional way to implement any Gm-C filter is to design an active element (transconductance amplifier) with desired gain and bandwidth that the frequency response degradation is negligible. This way the filter is directly synthesized using the usual prototype synthesis methods. When implementing filters with bandwidths of hundreds of megahertz or in gigahertz, the requirements for active components become stringent and high DC gain is difficult to achieve at low supply voltage of the modern CMOS processes.

6.1. Mapping of Non-Idealities

The losses in Gm-C filters are caused by the finite output resistances, intrinsic input, output and Miller capacitors of the transconductors (see section 5.4). These losses can be compensated by designing an active component with high output resistance and high quality reactance elements. Designing high quality active and reactance elements takes more chip area and power consumption. The transconductance element has no internal nodes in order to simplify the analysis and mapping of the Gm-C filter. However, the small signal model of the transconductance element used in the design example shows the internal nodes of the transconductance elements and from the transfer function of the active element, mapping of the Gm-C filter analysis can be simplified. Transfer function and DC gain of the transconductance element is shown in (6.1 & 6.2)

$$\frac{Vo}{Vin} = \frac{gm + scm}{scp + scm + go} \tag{6.1}$$

and DC gain,

$$Q_{gm} = A_{dc} = \frac{Ro}{Rm} \tag{6.2}$$

Where Cm and Co are intrinsic Miller and output capacitances and Go is the output resistance of the transconductance element.

To compensate the non-idealities of the actual filter realization as precisely as possible, accurate modeling and mapping of the different intrinsic components that influence the response of the filter is required.

6.2. Output resistance of OTA as lossy element for capacitor:

It is well known that the OTA output resistance introduce positive excess phase in the integrators, leading to Q degradation, and hence limiting the maximum achievable Q to be below the OTA dc gain. The DC gain of the active element can be improved by minimizing all other losses in the filter. Assuming that the DC gain of the transconductor is set and the only way to improve the integrator Q-values is through scaling of source (Rin) and load (Rout) resistances.

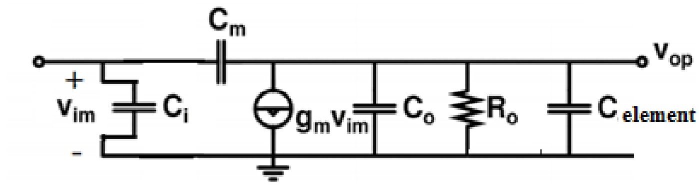

Figure 6.1: Macro model for OTA-C integrator.

In order to compensate the effect of output resistance on DC gain of the transconductance element, the output resistance of OTA can be used as lossy value for the capacitor.

i.e.,

$$Rc=Ro*gm \tag{6.3}$$

$$Gc=1/Rc$$

where Gc is the amount of lossiness that can be introduced in the ladder circuit elements.

The integrator equations corresponding to the passive filter equations shown in (5.2) is derived in (6.4)

$$v1 = \frac{Gm}{sc1 + Gin + Go}(Vin - v2)$$

$$v2 = \frac{Gm}{sc2 + Go}(v1 - v3)$$

$$v3 = \frac{Gm}{sc3+Go}(v2 - v4) \tag{7.4}$$

$$v4 = \frac{Gm}{sc4 + Go}(v3 - v5)$$

$$v5 = \frac{Gm}{sc5 + Gout + Go}(v4)$$

Another way to increase the OTA dc gain is by using cascades structures. The drawback with the topology is that it introduces at least one non-dominant high frequency pole, limiting the OTA's frequency response. For very high frequencies, these extra poles introduce negative excess phase on the basic integrator's transfer function.

Thus far only the effect of transconductor finite DC gain has been analyzed. When realizing active filters the Miller capacitor of the Gm-element affects the frequency response of the filter by adding high-frequency zeros to the integrator. By using the Miller-theorem, the effect of Miller-capacitor to the integrator can be compensated.

6.3. Effect of Intrinsic and Miller Capacitors:

The grounded parasitic capacitors increase the load, decreasing both center frequency and filter bandwidth. The overlapping capacitors introduce right-hand side zeros which causes peaking effects [17]. The degrading effect of Miller-capacitors is largest at the pass-band corner frequency.

The effect of intrinsic input, output and Miller capacitances can be compensated by subtracting parasitic capacitor values from the predistorted component values.

$$v2 = \frac{Gm + scm}{sc2 + scm + scp + Go}(v1 - v3)$$

Results: Non-idealities of the OTA's are mapped accordingly and reactive elements obtained for a 5th order chebychev filter are shown in table 6.1 and figure 6.2 shows the frequency response of the OTA-C filter with ideal and lossy reactance elements. As shown, by using ideal element values, the magnitude response offset from the response the filter intend to have.

Element	Value 1
c1	*2.18128*
c2	*0.90191*
c3	*3.37457*
c4	*0.90191*
c5	*2.18128*
Rin	10
Rout	*1*

Table 6.1: Lossy element values for 5th order OTA-C chebychev filter

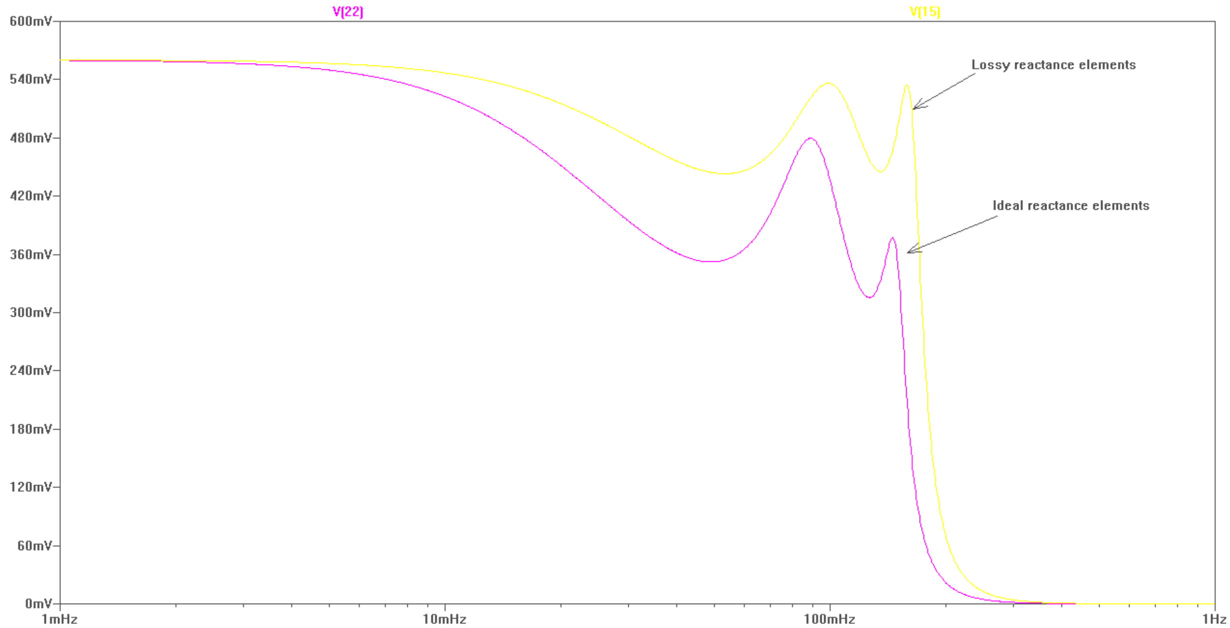

Figure 6.2: Magnitude bode plot of 5th order OTA-C chebychev filter

Chapter 7

Conclusion

In this thesis, a novel algorithm for pre-distorting OTA-C filters is presented. It is shown that the filter response can be predistorted through the mapping of losses and Miller-capacitors in the active components.

Many of the existing ladder synthesis algorithms [1, 3, 6, 9, 10, and 11] are based upon lossless prototype networks. However in reality inherent dissipation loss of reactance elements with finite ‘Q’ factors manifests itself as a degradation of achievable selectivity. This case is apparent in a roll off of insertion loss towards the pass band edges. However the effect of loss can be compensated by predistorting the synthesis of ladder filters which takes into consideration the lossiness introduced by the reactance elements. The synthesis algorithm makes it possible to design a passive ladder, Butterworth or Chebychev filter with the use of lossy reactance elements to provide better solutions as compared to existing predistortion algorithms. The algorithm is demonstrated by design examples. Major existing lossy ladder synthesis algorithms [2, 23, 27, 28, and 29] reported in the literature is summarized and advantages and disadvantages in terms of accuracy and practicality are also provided.

Using SPICE a prototype OTA is developed in 0.5 μm technology which is used as a building block for OTA-C filter design. Special care is taken to address those issues most important to filtering applications. The effect of OTA output resistance on filter’s quality factor, compensating techniques to increase the filter's quality factor and the effects of the non-linear capacitors are also discussed. Applying the OTA as a building block, the design of a fifth- order low-pass chebychev filter is presented.

This approach substantially relieves complexity of existing lossy ladder synthesis algorithms, and makes the synthesis of lossy ladder filters more reliable for practical applications. Further research can be done to extend the proposed synthesis algorithm and predistortion technique to design low-pass filters with zeros and for high-frequency filter applications.

Bibliography

1. Koller, R. and B. M. Wilamowski, "A Ladder Prototype Synthesis Algorithm," proceedings of *35th IEEE Midwest Symposium on Circuits and Systems*, Washington DC, USA, vol. 1, pp. 730-732, August 9-12, 1992
2. Marcin Jagiela and B. M. Wilamowski "A methodology of synthesis of lossy ladder filters" accepted for *13-th IEEE Intelligent Engineering Systems Conference, INES 2009*, Barbados, April 16-18., 2009
3. B. M. Wilamowski and R. Gottiparthy "Active and Passive Filter Design with MATLAB," *International Journal on Engineering Educations*, vol. 21, No 4, pp. 561-571, 2005
4. J.A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal* 7,1995,pp. 308-313.
5. John H. Mathews and Kurtis K. Fink, "Numerical Methods Using Matlab," 4th Edition, 2004
6. T. C. Fry, "The Use of Continued Fractions in the Design of Electrical Networks," *American Mathematical Society*, pp. 463-498, July-August 1929.
7. Steve Winder, "Analog and digital filter design," Newnes 2002.
8. Artur Lewinski "Analog filter techniques", VDM Verlag 2008.
9. Steadman, J. W. and B. M. Wilamowski, "Active Filters - Realization," chapter 29.2, pp. 750-757, *Electrical Engineering Handbook* second edition, R. Dorf - editor, CRC Press - 1997.
10. Rolf Schaumann, "Simulating Lossless Ladders with Transconductance-C Circuits," *IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing*, vol. 45, no. 3, Mar. 1998.
11. Robert D. Koller and Bogdan Wilamowski, "LADDER–A Microcomputer Tool for Passive Filter Design and Simulation," *IEEE Transactions on Education*, vol. 39, no. 4, Nov. 1996.
12. Edgar Sanchez-Sinencio, Randall L. Geiger, and H. Nevarez-Lozano, "Generation of Continuous-Time Two Integrator Loop OTA Filter Structures," *IEEETransactions on Circuits and Systems*, vol. 35, no. 8, Aug. 1988.
13. Mehmet A. Tan and Rolf Schaumann, "Design of a General Biquadratic Filter Section with Only Transconductances and Grounded Capacitors," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 4, Aug. 1988.
14. Yichuang Sun and J. Kel Fidler, "Structure Generation and Design of Multiple Loop Feedback OTA-Grounded Capacitor Filters," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 44, no. 1, Jan. 1997.
15. Yichuang Sun, "Synthesis of Leap-Frog Multiple-Loop Feedback OTA-C Filters," *IEEE Transactions on Circuits and Systems Express Briefs*, vol. 53, no. 9, Sept. 2006

16. W. M. Anderson, B. M. Wilamowski, and G. Dundar, "Wide Band Tunable Filter Design Implemented in CMOS", *11th INES 2007 -International Conference on Intelligent Engineering Systems,* Budapest, Hungary, June 29 2007-July 1 2007, pp. 219-223.
17. Silva-Martinez, J., "Design issues for UHF OTA-C filter realizations," *Mixed-Signal Design, 2001. SSMSD. 2001 Southwest Symposium*, 25-27 Feb. 2001 Page(s):93 – 98.
18. Adut, J. and Silva-Martinez, J., "Cascode transconductance amplifiers for HF switched-capacitor applications," *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium,* Volume 1, 25-28 May 2003, Page(s):I-365 - I-368 vol.1
19. **G.** A. DeVeirman and R. Yamasaki, "Fully-integrated 5 to 15 MHz programmable Bessel lowpass filter," *Proc. IEEE In?. Symp. Circ. Syst*., 1990, pp. 1155-1158.
20. -----, "Design of bipolar IO-MHz programmable continuous time 0.05' equiripple linear phase filter," *IEEE J. Solid-state Circuits*, vol. 27, pp. 324-331, 1992.
21. V. Gopinathan, Y. P. Tsividis, K.-S. Tan, and R. K. Hester, "Design considerations for high-frequency continuous-time filters and implemen- tation of an antialiasing filter for digital video," *IEEE J. Solid-state Circuits*, vol. 26, pp. 1368-1378, 1990.
22. Federal Communications Commision, "First Report and Order, Revision of Part 15 of the Commision's Rules Regarding UltraWideband Transmission Systems," ET Docket 98-153, February, 2002.
23. M. Kaltiokallio, S. Lindfors, V. Saari, J. Ryynanen, "Design of Precise Gain GmC-leapfrog Filters," *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium*, pp. 3534 – 3537
24. B. Nauta, E. Seevinck, "Linear CMOS Transconductance Element for VHF Filters," *IEEE Electronic Letters*, Volume 25, Issue 7, pp. 448450, March 1989
25. K. Wada, S. Takagi, Z. Czarnul, N. Fujii, "Design automation for integrated continuous-time filters using integrators," *Design Automation Conference, 1995. Proceedings of the ASP-DAC '95/CHDL '95/VLSI '95., IFIP International Conference on Hardware Description Languages; IFIP International Conference on Very Large Scale Integration., Asian and South Pacific*, 995 , Page(s): 435 - 439
26. H. Schmid, G.S. Moschytz, "Fundamental frequency limitations in current-mode Sallen-Key filters," *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium*, 1998 , Page(s): 57 - 60 vol.1
27. MacDonald, J., Temes, G. "A Simple Method for the Predistortion of Filter Transfer Functions," *Circuit Theory, IEEE Transactions*, vol. 10, 1963 , Page(s): 447 - 450
28. Natarajan, S., "A simple predistortion technique for active filters," Circuits and Systems, IEEE Transactions, vol. 31, 1984 , Page(s): 398 – 402
29. Fathelbab, W.M., Hunter, I.C., "A predistortion technique for microwave highpass prototype filters," *RF and Microwave Components for Communication Systems (Digest No.: 1997/126), IEE Colloquium*, 1997 , Page(s): 2/1 - 212

Appendices

A filtcoef.m

```
function [coefficient] = filtcoef(active_element,r_loss)

Rin=1;
Ro=active_element(end)+1e-9;
n = length(active_element)-1;
lossy_element = r_loss*ones(1,n);
coefficient = zeros(n+1,n+1);

for i = 1 : n + 1
   if i == 1
      coefficient(i,n+1) = 1/Ro;
   elseif i == 2
      coefficient(i,n) = active_element(n)/Ro;
      coefficient(i,n+1) = 1 + lossy_element(n)/Ro;
   else
      for j = 1 : n
         coefficient(i,j) = coefficient(i-1,j+1)*active_element(n+2-i);
      end;
      for j = 1: n + 1
         coefficient(i,j) = coefficient(i,j) + coefficient(i-2,j) + coefficient(i-1,j)*lossy_element(n+2-i);
      end;
      if i == n + 1
         for j = 1: n + 1
            coefficient(i,j) = coefficient(i,j) + Rin*coefficient(i-1,j);
         end;
      end;
   end;
end;

return
```

B filterr.m

```
function [E]=filterr(active_element,r_loss,pl1)
%% loss situation
% Ladder model
% ---Rin---L1--R1--|--|--L3--R3--|--|--L5--R5--|
%                |  |          |  |          |
%                C2 G2         C4 G4         Ro
%                |  |          |  |          |
% ---------------  |--|--------  |--|----------|
%
% active_element = [C1,L2,C3,L4,Rin,Ro]

%% calculation
coefficient = filtcoef(active_element,r_loss);

%% results
%normalize parameters
coefficient = coefficient./coefficient(end,1);
pl2=roots(coefficient(end,:));
e=(pl2-pl1);
E=e'*e;
return
```

C nelder_mead.m

```
function [BEST,f_BEST,SIMPLEX,f]=nelder_mead()
fid = fopen('C:\filter2.txt');                    % input data
tline = fgetl(fid);
[inp,tline]=strtok(tline,' ');
[d_SIM,tline]=strtok(tline,' ');
[df_min,tline]=strtok(tline,' ');
[ite_max,tline]=strtok(tline,' ');
fclose(fid);
i=1;
if ischar(inp)
   inp=str2num(inp);
end
if ischar(d_SIM)
   d_SIM=str2num(d_SIM);
end
if ischar(df_min)
   df_min=str2num(df_min);
end
if ischar(ite_max)
   ite_max=str2num(ite_max);
end
for i=1:3
obj = @filterr2;
x0 = inp(1,:);
xi = inp(2,1:end-1);
r_loss = x0(end);
x0(end)=[];
n=length(x0);
X0=ones(n,1)*x0;
SIMPLEX=[X0+diag(d_SIM*(rand(1,n)));x0]; % create simplex vertices
coefficient = filtcoef2(xi,0);
coefficient = coefficient./coefficient(end,1);
pl1=roots(coefficient(end,:));

for init=1:n+1
   f(init)=feval(obj,SIMPLEX(init,:),r_loss,pl1);
end
init=0;
SIMPLEX(:,end+1)=f';
SIMPLEX=sortrows(SIMPLEX,n+1);
f=SIMPLEX(:,end)';
SIMPLEX(:,end)=[];
for ite=1:ite_max
   M=sum(SIMPLEX(1:n,:))/n;
   R=2*M-SIMPLEX(end,:);
   f_R=feval(obj,R,r_loss,pl1);
   if f_R<f(end-1)
      if f(1)<f_R
         f(end)=f_R;
         SIMPLEX(end,:)=R;
      else
         E=2*R-M;
         f_E=feval(obj,E,r_loss,pl1);
         if f_E<f(1)
            f(end)=f_E;
            SIMPLEX(end,:)=E;
```

```
        else
           f(end)=f_R;
           SIMPLEX(end,:)=R;
        end
     end
  else
     if f_R<f(end)
        f(end)=f_R;
        SIMPLEX(end,:)=R;
     end
     C=(M+SIMPLEX(end,:))/2;
     f_C=feval(obj,C,r_loss,pl1);
     C2=(R+M)/2;
     f_C2=feval(obj,C2,r_loss,pl1);
     if f_C2<f_C
        C=C2;
        f_C=f_C2;
     end
     if f_C<f(end)
        f(end)=f_C;
        SIMPLEX(end,:)=C;
     else
        for i=2:n+1
           mid(i,:)=(SIMPLEX(1,:)-SIMPLEX(i,:))/2;
           f_mid(i)=feval(obj,mid(i,:),r_loss,pl1);
        end
        SIMPLEX=[SIMPLEX(1,:);mid];
        f=[f(1),f_mid];
     end

  end
  SIMPLEX(:,end+1)=f';
  SIMPLEX=sortrows(SIMPLEX,n+1);
  f=SIMPLEX(:,end)';
  SIMPLEX(:,end)=[];
  out=sprintf(['%d <== current best |',...
     '%d <== current worst| ite: %d'],f(1),f(end),ite);
 disp(out);
  if f(end)-f(1)<df_min, break; end
end
BEST=SIMPLEX(1,:);
f_BEST=f(1);
x0=BEST;
i=i+1;
if i==2,
  inp = [x0 r_loss;xi 0];
  d_SIM=2;
  df_min=1e-15;
  ite_max=500;
else
  inp = [x0 r_loss;xi 0];
  d_SIM=2;
  df_min=1e-17;
  ite_max=500;

end
end
x0=BEST;
```

```
xfer_calc;
disp('Optimized L C values:');
return
```

D xfercalc.m

```
%% transfer function calculation (optimized)

fid2 = fopen('output.txt','w');
active_element = x0;
Rin=active_element(end-1);
Ro=active_element(end);
n = length(active_element)-2;
lossy_element = r_loss*ones(1,n);
coefficient = zeros(n+1,n+1);
%% calculation
for i = 1 : n + 1
   if i == 1
      coefficient(i,n+1) = 1/Ro;
   elseif i == 2
      coefficient(i,n) = active_element(n)/Ro;
      coefficient(i,n+1) = 1 + lossy_element(n)/Ro;
   else
      for j = 1 : n
         coefficient(i,j) = coefficient(i-1,j+1)*active_element(n+2-i);
      end;
      for j = 1: n + 1
         coefficient(i,j) = coefficient(i,j) + coefficient(i-2,j) + coefficient(i-1,j)*lossy_element(n+2-i);
      end;
      if i == n + 1
         for j = 1: n + 1
            coefficient(i,j) = coefficient(i,j) + Rin*coefficient(i-1,j);
         end;
      end;
   end;
end;
%% results
%normalize parameters
fprintf(fid2,'\nOptimized Filter Characteristics:\n');
coefficient = coefficient./coefficient(n+1,1)
fprintf(fid2,['\n   coefficients: ',num2str(coefficient(n+1,:)),'\n']);
transfer_function = tf([1],coefficient(n+1,:))
pole(transfer_function)
figure(1); clf;
pzmap(transfer_function);
hold on;
figure(2); clf;
OPT=bodeoptions;
OPT.FreqScale='linear';
OPT.MagScale='linear';
OPT.MagUnits='abs';
bode(transfer_function,linspace(0,3,1000),OPT);
hold on;

active_element = xi(1:end-2);
Rin=xi(end-1);
Ro=xi(end);
n = length(active_element);
lossy_element = zeros(1,n);
coefficient = zeros(n+1,n+1);
%% calculation
```

```
for i = 1 : n + 1
   if i == 1
      coefficient(i,n+1) = 1/Ro;
   elseif i == 2
      coefficient(i,n) = active_element(n)/Ro;
      coefficient(i,n+1) = 1 + lossy_element(n)/Ro;
   else
      for j = 1 : n
         coefficient(i,j) = coefficient(i-1,j+1)*active_element(n+2-i);
      end;
      for j = 1: n + 1
         coefficient(i,j) = coefficient(i,j) + coefficient(i-2,j) + coefficient(i-1,j)*lossy_element(n+2-i);
      end;
      if i == n + 1
         for j = 1: n + 1
            coefficient(i,j) = coefficient(i,j) + Rin*coefficient(i-1,j);
         end;
      end;
   end;
end;
%% results
%normalize parameters
fprintf(fid2,'\nIdeal Filter Characteristics:\n')
coefficient = coefficient./coefficient(n+1,1)
fprintf(fid2,['\n   coefficients: ',num2str(coefficient(n+1,:)),'\n']);
pl1=roots(coefficient(n+1,:));
transfer_function = tf([1],coefficient(n+1,:))
pole(transfer_function)
figure(1);
pzmap(transfer_function);
figure(2);
bode(transfer_function,linspace(0,3,1000),OPT);


active_element = xi(1:end-2);
Rin=xi(end-1);
Ro=xi(end);
n = length(active_element);
lossy_element = r_loss*ones(1,n);
coefficient = zeros(n+1,n+1);
%% calculation
for i = 1 : n + 1
   if i == 1
      coefficient(i,n+1) = 1/Ro;
   elseif i == 2
      coefficient(i,n) = active_element(n)/Ro;
      coefficient(i,n+1) = 1 + lossy_element(n)/Ro;
   else
      for j = 1 : n
         coefficient(i,j) = coefficient(i-1,j+1)*active_element(n+2-i);
      end;
      for j = 1: n + 1
         coefficient(i,j) = coefficient(i,j) + coefficient(i-2,j) + coefficient(i-1,j)*lossy_element(n+2-i);
      end;
      if i == n + 1
         for j = 1: n + 1
            coefficient(i,j) = coefficient(i,j) + Rin*coefficient(i-1,j);
         end;
```

```
      end;
    end;
end;
%% results
%normalize parameters
fprintf(fid2,'\nLossy Filter Characteristics:\n')
coefficient = coefficient./coefficient(n+1,1)
fprintf(fid2,['\n   coefficients: ',num2str(coefficient(n+1,:)),'\n']);
transfer_function = tf([1],coefficient(n+1,:))
pole(transfer_function)
figure(1);
pzmap(transfer_function);
hold on;
print -dpng 'pzmap.png';
figure(2);
bode(transfer_function,linspace(0,3,1000),OPT);
legend('Optimized','Ideal','Lossy')
print -dpng 'bode.png';
fclose(fid2);
```