Complex Ladder Filter Realizations

Kenneth Martin
Granite SemiCom Inc.
Toronto ON, Canada
martin@granitesemi.com

***Abstract*—Passive real ladder synthesis is extended to synthesize ladder filters realizing transfer functions with non-complex-conjugate loss-poles. The normal requirement on ladder filters of having lossless two-port impedance (or admittance) transfer functions being positive-real-functions and purely even or odd is over-restrictive and unnecessary. For ladder filters realizing complex transfer functions, this is replaced by the requirement that two-port transfer functions are reactive functions where numerator and denominator polynomials have s-term coefficients alternating between being purely real or purely imaginary for even and odd s-terms. This preserves maximum power transfer and good sensitivity properties. Design procedures are given; examples are used to demonstrate the procedures.**

***Index Terms*—complex filters, cascade filters, pole-zero pairing, filter scaling, Monte Carlo sensitivity analysis**

I. Introduction

The synthesis of passive ladder filters having only inductors or capacitors (and possibly transformers), excepting resistive terminating components, has seen little activity since the 1970s. References describing how to design passive ladder filters are [1] and [2]. Passive ladder filters are the basis for many active filter realizations, such as active RC filters, and integrated active filters. The computer programs for designing passive ladder filters are mostly unavailable at this time; this paper describes a Matlab toolbox that is freely available and useful for the realization of ladder filters, both real and complex. The author has published techniques extending the approximation of band-pass filters to non-complex-conjugate complex transfer functions in [3] and Matlab programs for these techniques are available at [4]. This publication extends the work to ladder realization techniques suitable for realizing complex transfer functions. Much of this work is based on techniques described in [5] which introduced many of the original ideas, but does not give complete algorithms and generally only describes simple examples, and the programs of [5] are not currently available whereas the programs described here-in can be freely downloaded [4].

The author proposes the paper has merit for two reasons: a) it extends signal processing techniques in the area of complex filter realizations that could be useful in low-IF wireless applications, and b) it illustrates new techniques and programs for using Matlab for filter design and linear-system signal processing in general. [4].

Lossless one-port and two-port theory for real filters is extended to the case where complex-conjugate symmetry is not imposed. The impedances and admittances are still purely reactive (that is they have 90°I-V relationships in the frequency domain). However, the polynomial numerators and denominators of reactive complex impedances and admittances are not either purely even (that is have only terms of s^k where k is even) or purely odd (that is all even terms of s are zero and only odd terms of s exist). In complex lossless impedances and admittances, for even polynomials, the coefficients of s terms, where k is even, are still real. However, the coefficients of odd s terms are not zero; rather, they are purely imaginary. For odd polynomials the opposite is true: odd powers of s have real coefficients and even powers of s are not zero, but are purely imaginary. For real filters, this implies lossless one- and two-port impedances have $z(s) - z(-s) = 0$, whereas lossless complex one and two-port impedances have $z(s)-z^*(-s) = 0$ where the * superscript indicates all complex coefficients need to be conjugated. To the extent that size limitations permit, proofs for many of these changed properties will be given or outlined. Next, the paper continues with extending the procedures for realizing ladder filters using real components to realizations using complex components. Most of the time, the complex components used consist of just Ls and Cs as for real ladders, or an L in series with an imaginary resistance jx, or a capacitance in parallel with an imaginary admittance jy, or simply lone imaginary resistances or admittances. These complex passive elements can be realized using gyrators. The procedures start with how to produce the one-port impedance used for the removal process . The paper continues with describing how to remove complex components while ensuring the correct complex (non-conjugate) loss poles are realized. The approach is similar to that used for real filters, but new removal operations are presented. The fourth section presents examples.

II. Lossless Complex One and Two-Ports

One and two-port theory for real lossless networks normally starts with stating that a one-port impedance $Z(s)$, having reactance $X(s)$, and a one-port admittance having susceptance $B(s)$, must be positive-real-functions. For example, see pg 36 of [1] and pg 286 of [2]. This is not the case for complex passive networks. For a complex lossless passive network, one can write the one-port impedance as:

$$Z(s) = R_{-1}s + R_0 + \frac{R_1}{s-P_1} + \frac{R_2}{s-P_2} + \frac{R_3}{s-P_3} + \ldots \quad (1)$$

where R_{-1} if not 0, must be real, and R_0 if not equal to 0 must be imaginary. The residues $R_1 \ldots R_M$, when not zero, must be real and positive (similar to PRF impedances). The poles $P_1 \ldots P_M$ must be imaginary. Note that when s is replaced by $j\omega$, the impedance is reactive (or imaginary). Everything so far is identical for real lossless one-port impedances, the only difference is that whereas for real impedances the loss-poles must occur in complex-conjugate pairs, this is not the case for complex lossless one-port impedances (although nothing says it can‘t occur). Consider a simple example:

$$Z(s) = \frac{r_1}{s-jp_1} + \frac{r_2}{s-jp_2} \quad (2)$$

where all coefficients are real and r_1 and r_2 are positive. We have:

$$Z(s) = \frac{s(r_1 + r_2) - j(r_1 p_2 + r_2 p_1)}{s^2 + sj(-p_1 - p_2) + p_1 p_2} \tag{3}$$

Notice that: a) numerator and denominator differ in order of s by only one, b) the coefficients of the the highest terms in s are real, and c) as the power of s decreases, the coefficients switch between being imaginary and real. One can easily see that when s is replaced by $j\omega$, the impedance is purely reactive. It is straightforward, using a proof by induction, to show that these properties are in general true for lossless complex impedances of any order. Continuing the example, we can write (1) as:

$$Z(s) = \frac{sn_1 + jn_0}{s^2 + sjd_1 + d_0} \tag{4}$$

where all the numerator and denominator coefficients are now real. Also, we have $Z^*(-s)$ is given by

$$Z^*(-s) = \frac{-sn_1 - jn_0}{s^2 + sjd_1 + d_0} = -Z(s) \tag{5}$$

and we have

$$Z(s) + Z^*(-s) = 0 \tag{6}$$

This again is true in general. Whichever polynomial has its highest s term even order will not change when s is replace by -s and its coefficients are conjugated. The odd order polynomial stays the same except for its sign changing. This property is used to derive equations for the two-port parameters of the lossless portion of complex ladder filters.

III. Complex Ladder Filter Realizations

Realizing complex ladder filters is similar to realizing real ladder filters in that the first step is finding an impedance (or admittance) that can be used to remove the ladder sections. The removal process is done in such a way that transmission zero loss is guaranteed infinite (i.e. no transmission) at the desired loss frequencies while preserving maximal power transfer of the ladder to the termination resistor. Currently, three different alternatives are supported. One procedure involves first finding the two port parameters of the reactive section of the ladder filter; that is the ladder without the termination resistors. A second alternative is to first find the input impedance of the ladder to the right of R_S, but including R_L (the terminating load resistor), and then use this impedance for section removals. The first approach has the advantage that the zeros and poles are exactly on the imaginary axis, and this fact can be used to improve numerical accuracy. It has the disadvantage that for high-order monotonic passband filters with large attenuation in the stop band, finding the two-port reactive impedances is ill-conditioned. The conditioning is not as bad for filters having elliptic pass-bands. A third alternative is to find the removal impedances for singly terminated ladders, as opposed to doubly terminated ladders. The terminations can be either at the source or the output, with the procedure slightly different for the two cases. Numerical ill-conditioning seems less of a problem for this procedure even for monotonic pass-bands having numerous equal transmission zeros. We will first consider a doubly-terminated ladder where the removal is done from the two-port impedance input impedance z_{11}. This impedance is found using a derivation similar to pp. 308-315 of [2], but the final equation now becomes

$$z_{11}(s) = R_2 \left[\frac{N(s) + N^*(-s)}{N(s) - N^*(-s)} \right] \tag{7}$$

where we have now added the conjugate operation based on the discussion of the previous section. This modifies Table 7.3 on pg. 329 of [2] so that $E_{ev}(s)$ is replaced by $(E(s) + E^*(-s))/2$ and $E_{odd}(s)$ is replaced by $(E(s) - E^*(-s))/2$ [1]. Similar equations are used for $F_{ev}(s)$ and $F_{odd}(s)$. Also, either $z_{11}(s)$ or $z_{22}(s)$ should be chosen for the removals depending on which has the highest order. When realizing singly-terminated ladders, the equations for determining $z_{11}(s)$ or $z_{22}(s)$ are the same as for real ladder filters, but again with the difference that $E_{ev}(s)$ and $E_{odd}(s)$ are replaced using the definitions based on conjugation. If $P(s)$ is even then

$$z_{11}(s) = \frac{E_{ev}(s)}{E_{odd}(s)} \tag{8}$$

whereas if $P(s)$ is odd then $z_{11}(s) = \frac{E_{odd}(s)}{E_{ev}(s)}$. In addition, if the singly terminated side is at he load side, then the final ladder needs to be in reversed order as the equations are now for $z_{22}(s)$ rather than $z_{11}(s)$.

IV. Removal Operations

The largest differences between realizing complex ladder filters, as compared to real filters, has to do with the removal operations; there is considerably more flexibility with complex ladder filters for two reasons: first, negative elements are allowed (these are easily realized using differential circuits) and, second, a new element, namely an imaginary resistor (or conductance) can be removed [5]. This component can occur by itself as a partial removal, or often in parallel with series capacitors (or inductors) or in series with with shunt inductors (or capacitors). It allows the realization of transmission zeros that do not have conjugate-matched partners. The removal operations have been considerably extended from those proposed in [5]; for example, two non-conjugate-matched transmission zeros can be removed in one operation. Also, one structure the author often prefers is a ladder filter having only capacitors and imaginary resistors; restricting ladders in this manner allows for a large degree of automation in the design routines. When physically realizing the complex ladder filter using resistors (possibly imaginary), capacitors, and inductors only, one implements two signal paths, one for the real signal components, and one for the imaginary signal components. Cross-coupled components between the two paths realize imaginary components; imaginary resistors are realized using cross-coupled gyrators [5]. Alternatively, in a signal-flow-graph simulation of a ladder filter, the imaginary resistors can be realized by using cross-coupled gain paths in the feedback or feed-forward paths of integrator pairs (one integrator for the real signal component, and one integrator for the imaginary component). There are many possible removal operations for complex filters; paper size limitations limit this discussion to outlining just a couple of procedures.; others are found in the Matlab code [4] and will be described in the presentation. The Matlab code makes extensive use of Matlabs

[1] Note that definitions for $E(s)$, $F(s)$, and $P(s)$ are the same as in [2] and [3] and that they are related using Feldtkeller's equation

system objects and operations. In this publication we will describe a couple of selected removal operations. Consider the complete removal of a loss-pole at infinity. This can be done by first having the highest order in the numerator, which might first require inverting the current function (our Matlab procedures arbitrarily start with impedances) . If the removal function is an admittance, then a shunt capacitor with a parallel imaginary conductance is removed; otherwise a series inductor and imaginary resistor are removed. The Matlab built in *residue()* function can be used for these removals. The *remainder* portion of the output corresponds to the elements being removed, and the residues minus the reminder portion correspond to the remaining transfer function left after the removal. Another example of a removal operation is removing two loss poles in one operation by using a partial removal $Prmvl_i$ of the form

$$Prmvl_i = Ks + jy \tag{9}$$

where K and y are real. Let X_1 be the original removal function having its numerator order larger than its denominator order (the numerator and denominator order always differ by exactly 1) and arbitrarily assume X_1 is an admittance. Also, let $P1$ and $P2$ be the loss poles to be removed; note they are imaginary. If we take

$$\begin{aligned} X1P1 &= evalfr(X1, P1); \\ X1P2 &= evalfr(X1, P2); \\ K &= real(\frac{X1P2 - X1P1}{P1 - P2}); \\ X &= imag((-X1P1 - K \cdot P1)j) \end{aligned} \tag{10}$$

then taking

$$X2 = tf(X1) + tf([K, X], [1]); \tag{11}$$

changes the zeros of X2 so there are zeros exactly at the specified loss poles. The function *evalfr()* is Matlabs built in function for evaluating a system object (in this case $X1$) at a single frequency. The next step is to invert $X2$ and remove the poles $P1$ and $P2$; these can be removed using system object operations, or by again using Matlab's *residue()* function. The residues being removed must be converted to appropriate ladder components. Since $X1$ was an admittance, $X2$ is an impedance and the removals are series components. For example, they could be a series capacitor in parallel with an imaginary conductance. Alternatively, they could be the parallel combination of: a capacitor, an imaginary conductance, and an inductor having a series imaginary resistor; these component values can be found by solving a quadratic equation (using the completion of squares method) where the sum of the residues being removed is set equal to the total series impedance. It is also possible to remove single poles with the partial removal being a pole at infinity (i.e. Ks) or an imaginary conductance (or resistance, i.e. jy or jx) or partially removing a complex pole. The task of finding all valid complex removal operations, and determining which are best in which situations is on-going.

V. MATLAB CONSIDERATIONS

Using Matlab‘s built in functions and capabilities considerably simplifies obtaining ladder realizations compared to the tools and programs available in the 1970s when most of the original filter design programs were written. Still, Matlab has many limitations that should be considered. The control system objects mostly used for our programs are the *zpk()* system object (zero, pole, gain constant) and the *tf()* transfer-function system object. A major limitation is Matlab does not support adding or subtracting *zpk()* objects having complex coefficients. When adding or subtracting system objects, they must first be converted to *tf()* objects; unfortunately adding and subtracting *tf()* objects is ill-conditioned. We have successfully alleviated this problem to some degree in some of our operations by using Mullers method [6] for finding the zeros of the resulting transfer function; this is an area of ongoing research. Also, when adding or subtracting objects, often spurious (closely matched) poles and zeros can occur; the Matlab *minreal()* function is extensively used to remove them. For more complicated filters, numerical inaccuracies often degrade the calculations; to help eliminate these inaccuracies, pruning functions were developed; these help, but are not perfect. When designing a demanding high-order filter, the use of the debugger to identify and correct where the realization algorithms fail, was found to be essential. At the current time, the programs work well on some highly selective high-order filters, but not for all realizations. We hope to further improve numerical robustness as time goes on.

VI. EXAMPLES

In the first (simple) example, a real third-order elliptic low-pass transfer function, with a 0.25dB passband ripple, a pass-band edge of 1 rad., and having a complex conjugate loss-pole was approximated. This was then frequency shifted to +1Hz and the filter was realized in the complex domain using the removals just described. The resultant filter is shown in Fig. 1. A plot of the ladder response superimposed on a plot of

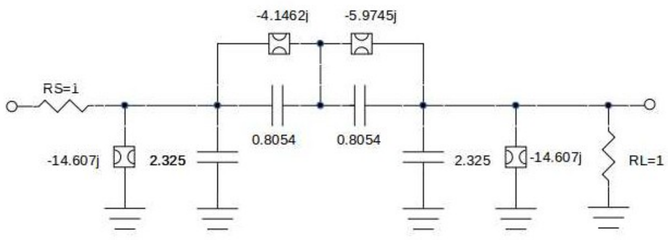

Figure 1. A third-order complex band-pass filter having a center frequency of 2π and a bandwidth of $1/(2\pi)$
.

the transfer function is shown in Fig. 2; the differences are almost indiscernible. The filter was also simulated in Spice where the imaginary conductances (equivalent to gyrators) were simulated using back-to-back voltage-controlled current sources, with the controlled source from the imaginary to real path being inverted. The simulated ac response (for positive frequencies only – spice normally can only simulate positive or negative frequencies, not both at the same time) were again effectively identical.

A second more interesting example is a filter that implements the magnitude transfer function shown in 3. The top level code for realizing this filter is:

```
p = [−5, −3, −1, 3.0, 5.0]; % initial guess at finite loss poles
px =[0.0]; % A fixed pole at dc
ni=1; % one loss pole at infinity
wp(1) = 0.5; % lower passband edge
wp(2) = 1.5; % upper passband edge
```

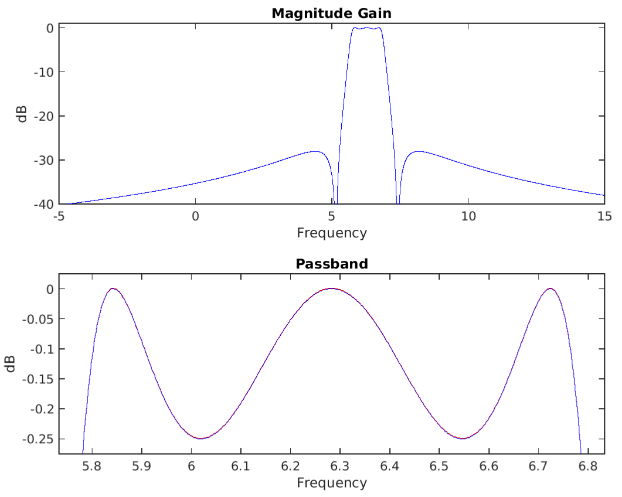

Figure 2. A plot of the ladder filter response superimposed on a plot of the transfer function

.

```
ws = [0.2  1.8];  % Stop−band spec
as = [20  20];  % Stop−band attenuation; not  important  unless  unequal

Ap = 0.02;  % the passband ripple  in  dB
ONE_STP = 1; % Consider negative and positive stop bands as a  single  sto
% A continuous−time  filter  with an equi−ripple pass−band
[H, E, F, P] =  design_ctm_filt (p,px,ni,wp,ws,as,Ap,' elliptic  ');
hndl(3)  =  figure(' Position ',[300  200 500  600]);

 plot_crsps (H,wp,ws,'b',[−10 10  −120 0.5]);

Etf  =  tf(zpk(E,   [],   1));
Ftf  =  tf(zpk(F,   [],   1));
Z1 = (Etf  − Ftf)/(Etf  + Ftf);
[Ks Pls  Rem] = getRes(Z1);
if  abs(min(Ks)) <  1e−5
     disp(' Filter ␣Design␣is␣ Ill −Conditioned')
end

Zin  = mkZin(E, F);  % In  this  design  remove from Zin

lddr  =  ladderClass ();  % A matlab object
% remove two poles at  same time  using  caps and complex impedances
[X1, elem1, elem2, elem3] = rmv2XPoles(Zin, P(2),  P(6),  lddr );
[X2, elem4, elem5, elem6] = rmv2XPoles(X1, P(1), P(3),  lddr );
[X3, elem7, elem8, elem9] = rmv2XPoles(X2, P(4), P(5),  lddr );
% remove final  pole  at   infinity
[X4, elem10] = rmvSCmplx(X3, lddr);
lddr.R2 = X4.K;  % the remainder should be simply  real  (the  load  resistor )

lddr2  =  ladderClass(lddr);
[X5, elem14, elem15] = rmvUsingS2(X2o, P(1), lddr2);

dispLddr(lddr);

lim  = [−10 10 −140 5];
% plot_lddr() uses  the  built  in  ladder  frequency  analysis:
% "lddr.freqEval(s)"
[gn,  db] =  plot_lddr (H,  lddr,wp,ws,'b',lim,lddr.R2);
```

Fig. 3 is a plot of the transfer function obtained from the approximation process superimposed on the frequency response of the realized ladder filter; it is difficult to see any differences. The realized filter response was obtained from a ladder analysis program written using a new Matlab object. This filter has four loss poles in the lower stop-band, three of which are movable, and one which is fixed to occur exactly at dc, to remove potential offsets and 1/f noise. The upper stop-band has two movable loss poles. The passband is from 0.5 rad to 1.5 rad and has 0.02dB ripple. The stop-bands are not symmetric. This filter was realized again using capacitors and parallel imaginary conductances only. The filter values are shown in Fig. 4. It was simulated using Spice. In addition, a signal-flow-graph (SFG) realization was designed that used first-order complex integrator sections only. Simulating the SFG filter in spice gave an identical response. Details of these realizations will be described at the conference.

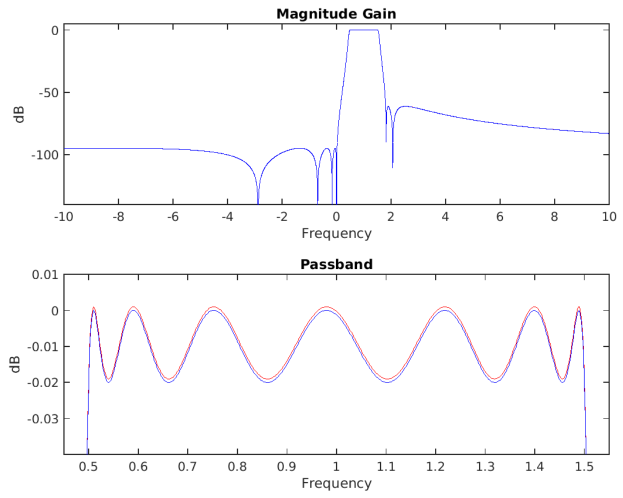

Figure 3. The response of a complex filter approximation superimposed on the response of the realized ladder filter based on capacitors and complex impedances only.

.

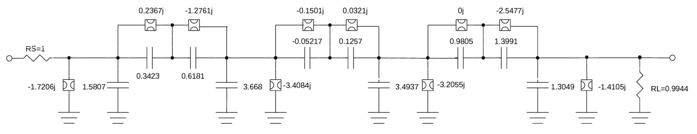

Figure 4. A seventh order non-symmetric complex filter

.

VII. Conclusion

The realization of complex ladder filters has been extended using Matlab; many new removal procedures were developed.

References

[1] G. Temes and W. LaPatr, *Introduction to Circuit Synthesis and Design*. McGraw Hill, 1977.

[2] A. Sedra and P. Brackett, *Filter Theory and Design: Active and Passive*. Matrix Publishers, 1978.

[3] K. Martin, "Approximation of complex iir bandpass filters without arithmetic symmetry," *IEEE Trans. Circuits and Systems I*, vol. 52, pp. 794–803, April 2005.

[4] K. Martin, "KM_ComplexFilterToolbox." https://github.com/kwmartin/KM_ComplexFilterToolbox.

[5] W. Snelgrove, A. Sedra, G. Lang, and P. O. Brackett, "Complex analog filters," *unpublished*, 1981. https://github.com/kwmartin/KM_ComplexFilterToolbox/blob/master/doc/snelgrove_cmplx.pdf.

[6] R. Burden and J. Faires, *Numerical Analysis*, pp. 79–81. PWS-Kent, 4 ed., 1989. coding by John Matthews, MatlabCentral/FileExchange.