Passive Ladder Synthesis in filtorX

B.M. Silveira‡, C. Ouslis, A.S. Sedra
University of Toronto, Toronto, Canada, M5S 1A4
fax: (416)978-7423 e-mail: ouslis@eecg.toronto.edu
‡Bell Northern Research, P.O. Box 3511 Station C, Ottawa, Canada, K1Y 4H7

Abstract — Passive ladder theory is a mature topic in filter design, yet there are few software tools to synthesise passive ladders. With the inclusion of ladder synthesis primitives into the filtorX language, filter designers can use a simple graphical interface to obtain passive ladder prototypes. These design techniques can also be programmed to obtain a means of automated ladder synthesis. A ladder structure derived from a bandpass transfer function highlights the ease of design and illustrates the potential for automation.

1 Introduction

filtorX[2] now incorporates the ability to perform passive ladder synthesis as an integral part of its language capabilities. Through its graphical interface, a designer can quickly and easily generate the many ladder realizations of a given transfer function and compare their relative merits.

Passive ladder filters, when designed for maximal power transfer from source to load, exhibit low sensitivity to component variations. Although passive ladders are no longer in widespread use, many of the desirable properties they exhibit are inherited by active circuit topologies based on ladders as prototypes; topologies such as active-RC[4], switched-capacitor[7], transconductance-C[8], and wave-digital[1] filters.

Ladder synthesis is an aspect of filter design which requires a significant degree of knowledge, and consequently ladder design has been limited to experts. By employing the flow control capabilities of filtorX, it is possible to capture the techniques and heuristic rules used by experts and apply these to automatically synthesise ladders.

2 Background

Because ladder synthesis entails numerous complex manipulations of rational functions, computer software is essential to ladder design. Currently available software[6][5], however, does not relieve the designer from possessing the extensive expert knowledge required. Thus, while the theory is mature[3], passive ladder synthesis remains the realm of more experienced designers.

The first step in the synthesis is to obtain a suitable transfer function which meets the requirements for a doubly-terminated lossless ladder. Among these requirements are that the maximum gain be unity (or 0dB), that there be at least one root at infinity for lowpass or bandpass filters, and at least one root at zero for bandpass and highpass filters. (With the removals shown here, all zeros must also lie on the imaginary axis in the s-plane, and hence must be at physical frequencies.)

From this transfer function, an impedance function is obtained, which corresponds to the network shown in figure 1, and its associated source and load resistance.

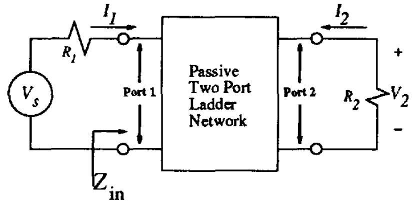

Fig.1 General Ladder Structure

The network is defined entirely by its impedance function and the location of the system zeros (or loss poles)[3]. In a ladder network, each arm realises either one zero, at zero or infinite frequency, or a pair of complex conjugate zeros at a finite frequency on the imaginary axis. A ladder is thus composed of the structures shown in figure 2.

The first two arms realise transmission zeros at DC, thus implementing high-pass sections, while the next two arms realise zeros at infinity, implementing low-pass characteristics. The arms numbered 5 and 6 implement high-pass sections which realise a pair of complex conjugate zeros at a finite frequency, but have finite transmission at infinite frequency. Sections 7 and 8 realise low-pass characteristics, also placing conjugate zeros at a finite frequency, and exhibiting finite transmission at DC. These last four sections will realise two zeros using three components; despite the presence of the single component preceding the resonant-tank circuits, no zero at DC or infinity is realised.

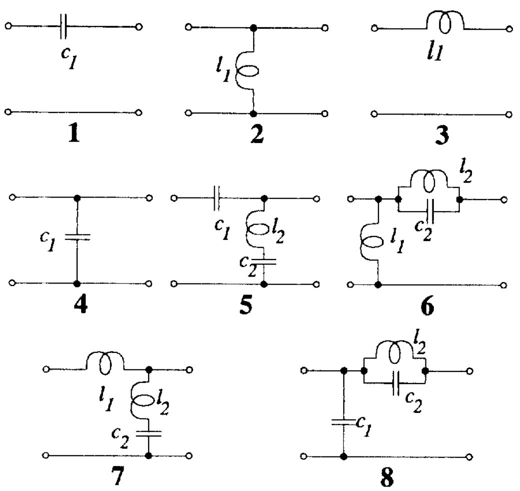

Fig.2 Possible Ladder Sections

After obtaining the impedance function, a structure which can realise all the zeros must be determined. This is conceptually as simple as cascading the sections depicted in figure 2. Practically speaking, the difficulty of passive ladder design rests with determining a sequence of ladder sections which realise the desired transfer function while exhibiting a low spread in component values. It should be noted that arbitrary sequences will not yield valid impedances.

Having decided upon a circuit configuration, the actual component values can then be determined using the method of "pole removals". This procedure extracts the component from the impedance function yielding the component value and a new impedance function of lower order than the original. The final result will be the values of all the components and a zero remainder impedance.

3 Synthesis

The procedure outlined above can be carried out manually using filtorX to perform the relevant operations. filtorX provides commands corresponding to each of the necessary actions in the synthesis of a ladder structure, thus minimizing the amount of manual work.

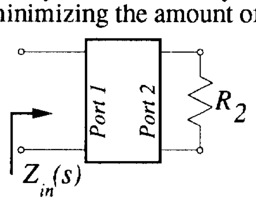

Fig.3 Input Impedance

To synthesise the network, an impedance function is first derived from the desired transfer function. As an example, consider performing the synthesis using the input impedance of the filter network, Z_{in} of figure 3, which would be obtained from filtorX as

```
impedance = zin( tx )
```

where this impedance would be obtained for the transfer function **tx**, and stored in the variable **impedance**.

Next, assuming that the designer knew the order of the sections, she would enter the sequence of removals as:

```
ladder=removal1 (impedance) *
removal2 (impedance)*
removal7 (impedance)*
removal3 (impedance)
```

where the removal numbers correspond to the ladder sections shown in figure 2, while '*', the filtorX ladder concatenation operator, is used to "connect" the sections together. The removal sequence operates on the **impedance** to yield the 5^{th} order ladder filter depicted in figure 4. This is merely one possible circuit realization

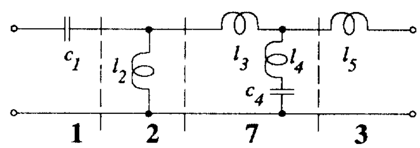

Fig.4 Example Ladder Filter

for the transfer function.

Instead of manually typing, the graphical menu shown in figure 5 could be used to perform the same sequence of removals. (The menu labels describe the ladder sections of figure 2.)

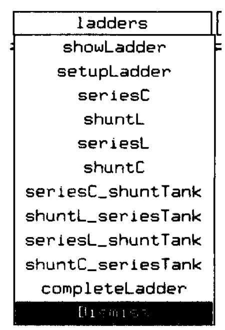

Fig.5 Graphical Ladder Menu.

For improved accuracy, a designer may choose to use the reactance functions of the network instead of the input impedance. The reactance functions define the

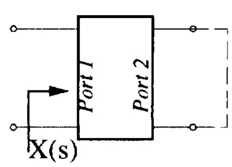

Fig.6 Input Reactance

input reactance looking into one port with the opposite port open or short circuited as denoted by the dashed line across port-2 in figure 6. Note that the reactance does not contain the termination resistance, R_2 (unlike Z_{in} where it was included).

The multitude of options makes it difficult to select a good impedance or reactance function; furthermore, the ordering of the sections provides a myriad of choices for the designer, each affecting the performance of the circuit. Typically, designers employ heuristic rules, or rules-of-thumb, to synthesise a ladder filter, yet there is no methodology to yield an optimal solution.

4 An Example

A complete example using the ladder primitives in filtorX will further illustrate the concepts. It involves an 8th order bandpass filter, with the attenuation response given in figure 7.

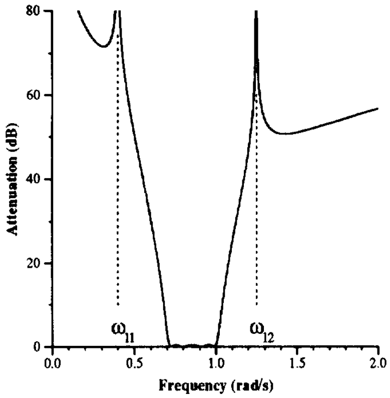

Fig.7 Response of 8th Order Example

This filter has two zeros at DC, two zeros at infinity, and two pairs of finite zeros(ω_{l1} and ω_{l2}). It is assumed that the transfer function exists in filtorX and is called **txBandpass**.

To obtain the input impedance, Z_{in}, the designer would type.

```
txImpedance = zin(txBandpass)
```

Given this impedance function, a circuit would next be sketched so that the removal operations could be used to obtain the component values. One possible circuit realization is shown in figure 8.

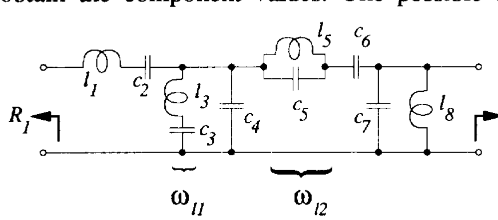

Fig.8 Example Circuit Realization

This realization was obtained using some common heuristic rules[4][3]: the sections involving zeros at extreme frequencies (s=0 and s=∞), are placed at the ends of the network; the sections involving finite zeros are located in the middle such that the zeros located closest to the passband edge are closest to the centre of the network; sections 5 and 6 are used for finite zeros which are located in the lower stopband; sections 7 and 8 are used for finite zeros which are located in the upper stopband. These rules-of-thumb generally provide reasonable filters which exhibit low component spread, minimise the presence of negative components, and have good sensitivity.

From this circuit, the sections can be ordered as 3-5-8-1-4-2, with section 5 realizing ω_{l1}, and section 8 realizing ω_{l2}. In filtorX, this would be accomplished by typing:

```
ExampleLadder =
removal3( impedance ) *
removal5(impedance, omegaL1)*
removal8(impedance, omegaL2)*
removal1( impedance ) *
removal4( impedance ) *
removal2( impedance )
```

where **omegaL1** and **omegaL2** are variables for the locations of the zero's frequencies. The filter can then be viewed by typing:

```
print ExampleLadder
```

producing the result:

```
ladder ExampleLadder (
seriesR 1.0000000000e+00;
seriesL 5.5068718184e+00;
seriesC 2.5942252752e-01;
shuntLC 5.6114878331e-01,
1.1160175271le+01;
shuntC 3.1821631906le+00;
seriesLC 2.0348942738e+00,
3.1446233865 4e-01;
seriesC 3.3418801030e-01;
shuntC 2.7625647438 0e+00;
shuntL 4.8129030575 0e-01;
shuntG 5.2965251047 7e-01)
```

Having obtained the ladder, it is prudent to check its transfer response for accuracy with the original transfer function. This ensures that there were no numerical problems with the impedance-generating or removal functions. This can be easily accomplished in filtorX by employing the function, **lad2ratfn**, which determines the transfer function corresponding to the given (8th-order) ladder as,

```
txObtained =
lad2ratfn(ExampleLadder,8)
```

Performing the simulation and checking these results with the original transfer function, **txBandpass**, showed that the relative accuracy between the original and simulated natural modes was on the order of 10^{-10}.

5 Implementation

Although it would seem that the heuristic rules might best be coded in C, a greater benefit would be gained by using the filtorX language. This choice will put the algorithm and specific solution at the user's level, empowering her to alter the functionality if desired. The loss in speed is minimal and would not cause any significant wait or delay for designers.

6 Automation

Due to the nature of ladder synthesis, many choices exist at each step of the procedure. This sequence of actions yields many valid circuits as well as many invalid ones.

In generating ladder structures automatically, knowledge typically used by the designer to direct choices in the design process is captured and documented as programming steps. Although many possible ladder configurations can realise a given transfer function, ladders can be broken down into basic sections (illustrated in figure 2). By nature of their construction, each section realises only certain root values. This relationship indicates that the order in which the roots are chosen for removal operations dictates the structure of the ladder. For any given system zero, and any given impedance function, only one of the two possible removals can be performed. For example, when performing the removal of a root at infinity, only one of either removal 3 or removal 4 can be performed on the given impedance function. Thus, a series of roots and an impedance function defines a specific ladder. Based on this principle, and the heuristic rules outlined in section 4, an algorithm can be created which orders the system zeros of a transfer function to be combined with an associated impedance function, in turn creating a ladder structure. Variations in the order of the system zeros will then yield various ladder structures.

The steps which would comprise such an algorithm are:

```
separate system zeros
order zeros
determine impedance function
for each zero {
   locate appropriate removal pair
   determine feasible removal
   reduce impedance function
}
```

Initial results of procedures based on this algorithm have successfully yielded ladders from arbitrary transfer functions without user interaction.

7 Conclusions

The functions required for ladder synthesis have been included as integral components of the filtorX software package. The simplicity and versatility of ladder design using filtorX was illustrated by means of a bandpass ladder example, which was synthesised and simulated to indicate accuracy.

An algorithm was devised, based on properties exhibited by ladders which could be related to the system zeros, by which a series of different ladder structures can be obtained from a single transfer function. By creating a variety of ladder structures, using the automatic generation algorithm, the relative merits of each may be compared using some figure of merit, such as component spread. Early results have shown success with the technique, for which data must now be compiled to determine its efficiency and efficacy.

8 Acknowledgements

This work was supported in part by: ITRC, a provincial centre of excellence funded by the government of Ontario; Micronet, a federally funded network of centres of excellence; and NSERC, the Natual Sciences and Engineering Research Coucil.

9 References

[1] A. Fettweis, *Canonic Realization of Ladder Wave Digital Filters* , International Journal of Circuit Theory and Applications, Vol. **3** No. 4, pp 321-332, December 1971.

[2] C. Ouslis, W.M. Snelgrove, A.S. Sedra, *filtorX: An interactive Design Language for Filters,* Advances in Electrical Engineering Software: Proceedings of the First international Conference On Electrical Engineering Analysis and Design. Computational Mechanics/Springer Verlag, Lowell, Massachuetts , August 1990.

[3] R. Saal, E. Ulbrich, *On the Design of Filters by Synthesis* , IRE Transactions on Circuit theory, Vol. **CT-5** No. 4, pp 284-317, December 1958.

[4] A.S. Sedra, P.O. Brackett, Filter theory and Design: Active and Passive , Matrix Publishers inc., Beaverton, Oregon, 1978.

[5] W.M. Snelgrove. *FILTOR2 -- A Computer-Aided Filter Design Package,* University of Toronto, Toronto, Ontario, 1981.

[6] G. Szentermai, *FILSYN — A General-Purpose Filter Synthesis Program* , Proceedings of IEEE, Vol. **65** No. 10, 1977.

[7] Y.P. Tsividis, P. Antognetti, VLSI Circuits for Telecommunications , Prentice-Hall, 1985.