

One port Network Synthesis

C. J. Smartt

November 13, 2018

1 Section

The network synthesis process implemented in this work aims to develop a one port passive electrical circuit which reproduces a specified frequency dependent impedance function. The impedance function may be represented in a rational function form or in a pole-residue form and the resulting network is developed as a Spice sub-circuit model.

The software is written in Fortran and includes a Makefile to allow the software to be compiled under unix. Windows users may install Cygwin (which is effectively a unix type environment) and then compile the software under cygwin.

This document describes the theory which underlies the modelling of frequency dependent transfer functions as passive circuits in Spice before describing the software which implements the process and presenting some examples of the process in action.

1.1 License

There are 2 license types used in the NETWORK_SYNTHESIS project.

GPL - GNU General Public License v.3 See COPYING in root

LGPL - GNU Lesser General Public License v.3 See COPYING.LESSER in root

These licenses are applied as follows:

GPL: Applies to all source coded with the exception of the eispack library in /SRC/EISPACK

LGPL Applies to the eispack library in /SRC/EISPACK

2 Network synthesis for specified frequency dependent transfer functions

In this section we describe how frequency dependent transfer functions represented as rational functions in the laplace variable, s , may be simulated using passive circuits and hence modelled in Spice. We assume that the rational functions represent stable systems i.e. that their impulse response does not diverge as $t \rightarrow \infty$. This implies that the poles of the transfer function lie in the left hand side of the s plane.

We consider two types of transfer functions according to their properties

1. Positive-real transfer functions which may be represented as a one port network (simple impedance)
2. non-positive-real transfer functions which may be represented as two port networks

The first type of transfer function arises in this work in the propagation correction functions. The second type may arise in transfer impedance models where the transfer resistance and/or inductance may be negative.

The form of the transfer functions that we wish to implement as a passive circuit are rational functions as in equation 1

$$H(s) = \frac{a_0 + a_1 \left(\frac{s}{\omega_0}\right) + a_2 \left(\frac{s}{\omega_0}\right)^2 + \dots}{b_0 + b_1 \left(\frac{s}{\omega_0}\right) + b_2 \left(\frac{s}{\omega_0}\right)^2 + \dots} \quad (1)$$

where ω_0 is a frequency normalisation factor. The methods applied make use of the work of Foster [4], Cauer[3] and Brune [5].

2.1 One port impedance models

A passive impedance function must be 'positive-real' to constitute a physical impedance which can be synthesised from a network of inductors, capacitors, resistors and transformers. A positive-real rational function is defined by the following conditions:

1. The real part of the function $Z(s)$ must be positive for all $s = \alpha + j\omega, \alpha > 0$
2. The imaginary part of the function $Z(s)$ must be zero for all $s = \alpha + j\omega, \omega = 0$

Positive real rational functions have the following properties:

1. The number of poles and zeros differ by at most 1
2. The coefficients of the rational function are all real and positive
3. The poles and zeros of the function must lie in the left hand side of the s -plane or on the $s - j\omega$ axis
4. If a function $Z(s)$ is positive-real then so is $Y(s) = \frac{1}{Z(s)}$

In this section we describe how an impedance represented as a rational function in s may be synthesised by a passive one port circuit consisting of resistors, inductors, capacitors and transformers i.e. spice R, L, C and K elements.

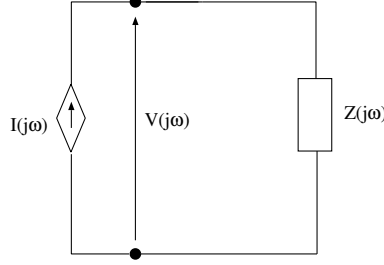


Figure 1 One port network model of a positive-real transfer function

Figure 1 shows a one port network driven by a current source $I\{j\omega\}$ whose port voltage is given by $V\{j\omega\} = Z\{j\omega\} I\{j\omega\}$. The impedance can be synthesised as a ladder network as shown in figure 2. The impedance of the ladder network may be

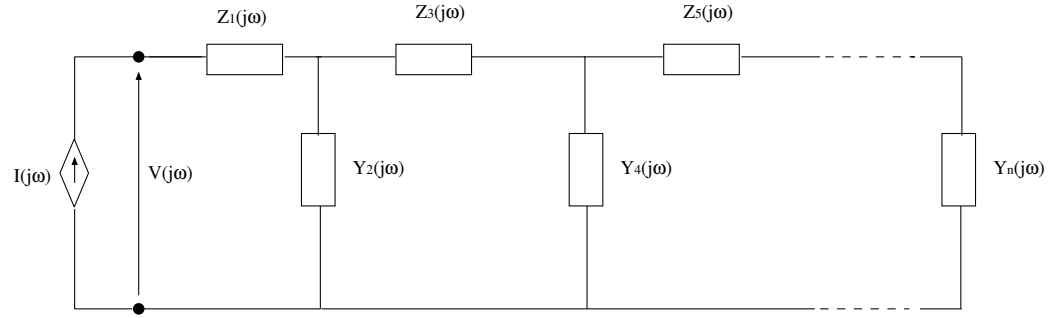


Figure 2 Ladder network synthesis of an impedance

expressed as the sum of the first series impedance, $Z_1(s)$ and a remainder, $Z_{r1}(s)$ i.e.

$$Z(s) = Z_1(s) + Z_{r1}(s) \quad (2)$$

The circuit is shown in figure 3 where $Z_1(s)$ is the impedance of a combination of R, L, C, K elements and $Z_{r1}(s)$ is a stable positive real function. The remainder impedance, $Z_{r1}(s)$, is then synthesised as the parallel combination of the admittance function $Y_2(s)$ and a remainder, $Y_{r2}(s)$ as seen in figure 4

$$Z_{r1}(s) = \frac{1}{Y_2(s) + Y_{r2}(s)} \quad (3)$$

thus

$$Z(s) = Z_1(s) + \frac{1}{Y_2(s) + Y_{r2}(s)} \quad (4)$$

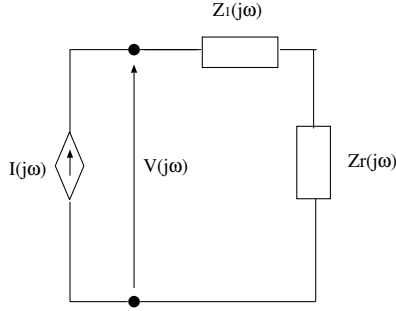


Figure 3 Initial stage of ladder network synthesis

The admittance $Y_{r2}(s)$ is then synthesised as the sum of two series impedances i.e.

$$Z(s) = Z_1(s) + \frac{1}{Y_2(s) + \frac{1}{Z_3(s) + Z_{r3}(s)}} \quad (5)$$

It is seen from this that the impedance function of the ladder network is expressed in a continued fraction form with alternating impedance and admittance terms. This synthesis procedure in which admittance and impedance elements of the ladder network are identified in turn continues until the remainder term is reduced to zero. This leads to a relatively straightforward synthesis process for impedance functions.

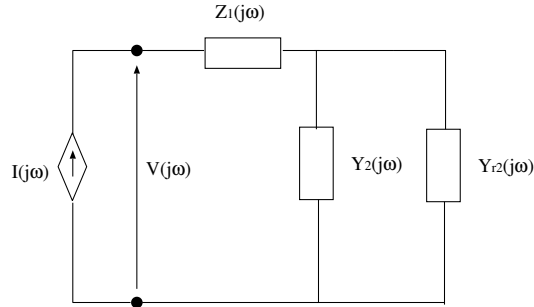


Figure 4 Second stage of ladder network synthesis

2.2 Algorithm for impedance synthesis

The starting point for the algorithm for impedance synthesis is a rational function in s which must be stable and positive-real. The impedance is synthesised by extracting series impedances and parallel admittances in turn until the remainder is zero. The impedance/ admittance extraction makes use of a pole-residue representation of the transfer function in order to identify branches which are combinations of R, L and C

elements. This method can get stuck in that a viable branch cannot be identified while leaving a stable positive-real remainder function. In this case a technique developed by Brune [5] is used to continue the synthesis. Brune's method makes use of transformers (K elements in Spice).

An outline of the algorithm is as follows:

1. Attempt to identify a viable series impedance branch i.e. a branch which may be synthesised with R, L, and C elements and which leaves a remainder $Z_r(s)$ which is stable and positive-real. If a viable branch is found then this process can be repeated until no further viable series impedance branches can be found.
2. Once no further viable series impedance contributions can be found, calculate the admittance $Y_r(s) = 1/Z_r(s)$ and attempt to identify a viable parallel admittance branch i.e. a branch which may be synthesised with R, L, and C K elements and which leaves a remainder $Y_r(s)$ which is stable and positive-real. If a viable branch is found then this process can be repeated until no further viable parallel admittance branches can be found.
3. If no viable series impedance branches or parallel admittance branches can be found then use the Brune synthesis method and implement the resulting circuit with R, L, C and K elements
4. Return to step 1 and repeat until the remainder impedance/ admittance is zero.

2.3 Identification of series impedance branches

A viable series impedance branch can be one of the following:

1. RLC in parallel branch
2. LC in parallel branch
3. RC in parallel branch
4. RL in parallel branch
5. C branch
6. L branch
7. R branch (which may be identified in two different ways)

These branches are shown in figure 5

Viable branches are found by first forming a pole-residue expansion of the impedance function where poles and their corresponding residues may be real or may be complex conjugate pairs i.e.

$$Z(s) = R + sL + \sum_{i=1}^{n_{real.poles}} \frac{r_i}{s - p_i} + \sum_{i=1}^{n_{complex.pole.pairs}} \frac{r_i}{s - p_i} + \frac{r_i^*}{s - p_i^*} \quad (6)$$

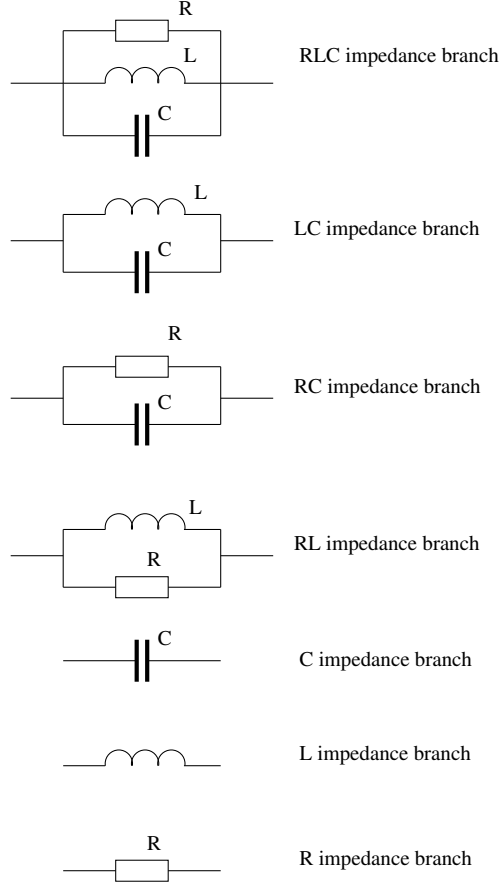


Figure 5 Series Impedance Branches

Then each of the branch types is looked for in the pole-residue expansion in the order of the above list. Each of the following subsections shows how each of the above branch types may be identified from the pole-residue expansion.

2.3.1 RLC in parallel branch

The RLC parallel branch has an impedance function which can be expressed in rational function as

$$Z(s) = \frac{\frac{s}{C}}{\frac{1}{LC} + \frac{s}{CR} + s^2} \quad (7)$$

This may be expressed in pole-residue form as the complex pole-residue pair

$$Z(s) = \frac{r}{s-p} + \frac{r^*}{s-p^*} = \frac{(rp^* - r^*p) + (r + r^*)s}{pp^* - (p + p^*)s + s^2} \quad (8)$$

Where the poles are complex (and not purely imaginary). Equating coefficients of the rational functions a requirement for a viable RLC parallel branch can be established i.e. the numerator constant term is zero:

$$rp^* - r^*p = 0 \quad (9)$$

In addition we require that all the component values of the parallel RLC circuit be positive i.e.

$$C = \frac{1}{r + r^*} > 0 \quad (10)$$

$$L = \frac{r + r^*}{pp^*} > 0 \quad (11)$$

$$R = \frac{r + r^*}{p + p^*} > 0 \quad (12)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the impedance function describe a physical impedance. If these requirements are satisfied then a parallel RLC branch can be extracted from the impedance function and implemented as a series impedance branch in the ladder network.

2.3.2 LC in parallel branch

The LC parallel branch has an impedance function which can be expressed in rational function as

$$Z(s) = \frac{\frac{s}{C}}{\frac{1}{LC} + s^2} \quad (13)$$

This is a special case of the RLC branch in which $R \rightarrow \inf$ may be expressed in pole-residue form as the complex pole-residue pair

$$Z(s) = \frac{r}{s - p} + \frac{r^*}{s - p^*} = \frac{(rp^* - r^*p) + (r + r^*)s}{pp^* - (p + p^*)s + s^2} \quad (14)$$

Where the poles are purely imaginary. Equating coefficients of the rational functions a requirement for a viable LC parallel branch can be established i.e. the poles are imaginary:

$$p^* + p = 0 \quad (15)$$

The numerator constant term is zero:

$$rp^* - r^*p = 0 \quad (16)$$

In addition we require that all the component values of the parallel LC circuit be positive i.e.

$$C = \frac{1}{r + r^*} > 0 \quad (17)$$

$$L = \frac{r + r^*}{pp^*} > 0 \quad (18)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the impedance function describe a physical impedance. If these requirements are satisfied then a parallel RLC branch can be extracted from the impedance function and implemented as a series impedance branch in the ladder network.

2.3.3 RC in parallel branch

The RC parallel branch has an impedance function which can be expressed in rational function as

$$Z(s) = \frac{\frac{1}{C}}{\frac{1}{RC} + s} \quad (19)$$

This may be expressed in pole-residue form as

$$Z(s) = \frac{r}{s - p} \quad (20)$$

Where the pole and residue are both real. By equating coefficients of the functions, values of R and C can be identified We require that all the component values of the parallel RC circuit be positive i.e.

$$C = \frac{1}{r} > 0 \quad (21)$$

$$R = \frac{-r}{p} > 0 \quad (22)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the impedance function describe a physical impedance. If these requirements are satisfied then a parallel RC branch can be extracted from the impedance function and implemented as a series impedance branch in the ladder network.

2.3.4 RL in parallel branch

The RL parallel branch has an impedance function which can be expressed in rational function as

$$Z(s) = \frac{sR}{\frac{R}{L} + s} = R - \frac{\frac{R^2}{L}}{s + \frac{R}{L}} \quad (23)$$

As for the parallel RC branch, this may be expressed in pole-residue form, including a constant term as

$$Z(s) = K + \frac{r}{s - p} \quad (24)$$

Where the pole and residue are both real. By equating coefficients of the functions, values of R and L can be identified We require that all the component values of the parallel RL circuit be positive i.e.

$$R = \frac{r}{p} > 0 \quad (25)$$

$$L = \frac{-r}{p^2} > 0 \quad (26)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the impedance function describe a physical impedance. If these requirements are satisfied then a parallel RC branch can be extracted from the impedance function and implemented as a series impedance branch in the ladder network.

2.3.5 Capacitance branch

The capacitance branch has an impedance function which consists of a pole at zero i.e.

$$Z(s) = \frac{1}{sC} \quad (27)$$

Thus in the pole-residue expansion we require a pole at zero. In addition we require that capacitance be positive i.e.

$$C = \frac{1}{r} > 0 \quad (28)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the impedance function describe a physical impedance. If these requirements are satisfied then a capacitance branch can be extracted from the impedance function and implemented as a series impedance in the ladder network.

2.3.6 Inductance branch

The inductance branch has an impedance given by

$$Z(s) = sL \quad (29)$$

Thus in the pole-residue expansion we require a sL term (pole at infinity in the impedance function) with positive inductance, L . The final requirement is that the remainder function be positive real i.e. the remaining terms of the impedance function describe a physical impedance. If these requirements are satisfied then an inductance branch can be extracted from the impedance function and implemented as a series impedance in the ladder network.

2.3.7 Resistance branch, type 1

The resistance branch of type 1 is identified by a positive constant term in the pole-residue expansion of the impedance. If the remainder function is positive real i.e. the remaining terms of the impedance function describe a physical impedance then a resistance branch can be extracted from the impedance function and implemented as a series impedance in the ladder network.

2.3.8 Resistance branch, type 2

The resistance branch of type 2 may be found from the rational function form of the impedance i.e. if

$$Z(s) = \frac{a_0 + a_1 \left(\frac{s}{\omega_0}\right) + a_2 \left(\frac{s}{\omega_0}\right)^2 + \dots}{b_0 + b_1 \left(\frac{s}{\omega_0}\right) + b_2 \left(\frac{s}{\omega_0}\right)^2 + \dots} \quad (30)$$

then we can calculate the resistance, R as

$$R = \frac{a_0}{b_0} \quad (31)$$

as opposed to the pole-residue form whose constant (resistance) term is found as the ratio of the numerator and denominator coefficients of the highest power of s subtracting the resistance as defined in 31 results in a remainder impedance function with a pole at infinity. As for all the other branch identification methods, this resistance branch can only be extracted if the remainder impedance is a stable physical impedance function.

2.4 Identification of admittance branches

A variable admittance branch can be one of the following:

1. RLC in series branch
2. LC in series branch
3. RC in series branch
4. RL in series branch
5. C branch
6. L branch
7. R branch (which may be identified in two different ways)

These branches are shown in figure 6 and are seen to be the duals of the corresponding impedance branches in figure 5. The dual circuits are found by the transformations

$$\begin{array}{lcl} Z & \leftrightarrow & Y \\ L & \leftrightarrow & C \\ \frac{1}{C} & \leftrightarrow & \frac{1}{L} \\ R & \leftrightarrow & G \end{array} \quad (32)$$

The admittance branches are found in the same way as for the series impedance branches i.e. viable branches are found by first forming a pole-residue expansion of

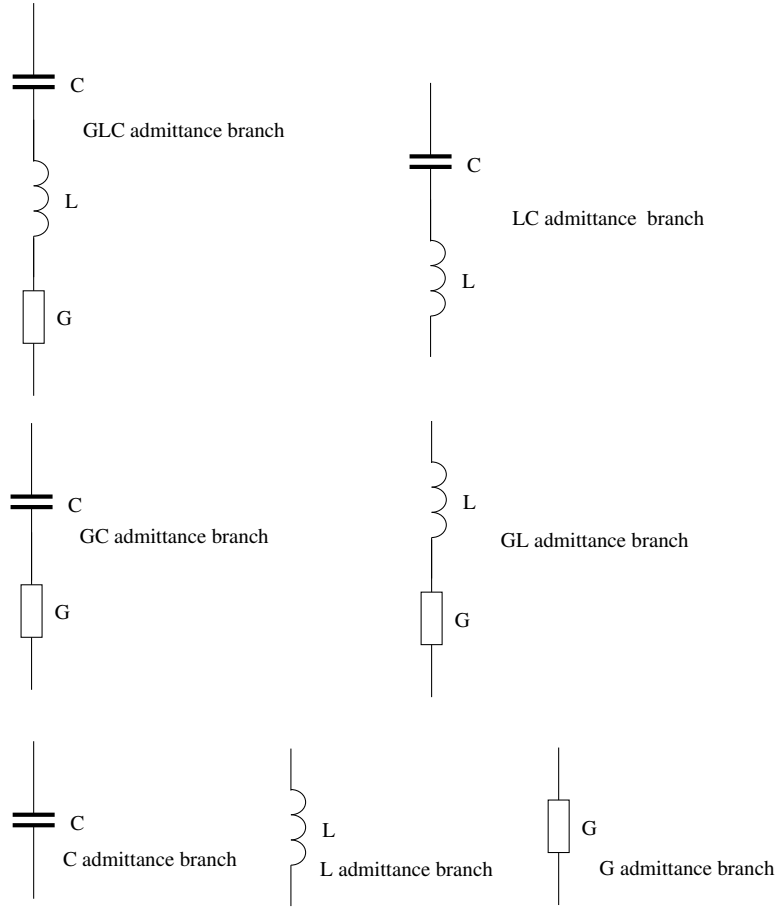


Figure 6 Parallel Admittance Branches

the admittance function where poles and their corresponding residues may be real or may be complex conjugate pairs i.e.

$$Y(s) = G + sC + \sum_{i=1}^{n_{real.poles}} \frac{r_i}{s - p_i} + \sum_{i=1}^{n_{complex.pole.pairs}} \frac{r_i}{s - p_i} + \frac{r_i^*}{s - p_i^*} \quad (33)$$

For completeness a brief summary of the admittance branch identification processes is given here. The admittances may have inductance, capacitance or conductance though a non-zero conductance is implemented as a resistance $R = 1/G$ in Spice.

2.4.1 GCL in series branch

The GCL series branch has an admittance function which can be expressed in rational function as

$$Y(s) = \frac{\frac{s}{L}}{\frac{1}{CL} + \frac{s}{LG} + s^2} \quad (34)$$

This may be expressed in pole-residue form as the complex pole-residue pair

$$Y(s) = \frac{r}{s-p} + \frac{r^*}{s-p^*} = \frac{(rp^* - r^*p) + (r + r^*)s}{pp^* - (p + p^*)s + s^2} \quad (35)$$

The poles are complex (and not purely imaginary). Equating coefficients of the rational functions a requirement for a viable GCL series branch can be established i.e. the numerator constant term is zero:

$$rp^* - r^*p = 0 \quad (36)$$

In addition we require that all the component values of the series GCL circuit be positive i.e.

$$L = \frac{1}{r + r^*} > 0 \quad (37)$$

$$C = \frac{r + r^*}{pp^*} > 0 \quad (38)$$

$$G = \frac{r + r^*}{p + p^*} > 0 \quad (39)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the admittance function describe a physical admittance. If these requirements are satisfied then a series GCL branch can be extracted from the admittance function and implemented as a parallel admittance branch in the ladder network.

2.4.2 LC in series branch

The LC series branch has an admittance function which can be expressed in rational function as

$$Y(s) = \frac{\frac{s}{L}}{\frac{1}{CL} + s^2} \quad (40)$$

This is a special case of the GCL branch in which $G \rightarrow \inf (R \rightarrow 0)$ may be expressed in pole-residue form as the complex pole-residue pair

$$Y(s) = \frac{r}{s-p} + \frac{r^*}{s-p^*} = \frac{(rp^* - r^*p) + (r + r^*)s}{pp^* - (p + p^*)s + s^2} \quad (41)$$

Where the poles are purely imaginary. Equating coefficients of the rational functions a requirement for a viable LC series branch can be established i.e. the poles are imaginary:

$$p^* + p = 0 \quad (42)$$

The numerator constant term is zero:

$$rp^* - r^*p = 0 \quad (43)$$

In addition we require that all the component values of the series LC circuit be positive i.e.

$$L = \frac{1}{r + r^*} > 0 \quad (44)$$

$$C = \frac{r + r^*}{pp^*} > 0 \quad (45)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the admittance function describe a physical admittance. If these requirements are satisfied then a series GCL branch can be extracted from the admittance function and implemented as a parallel admittance branch in the ladder network.

2.4.3 GL in series branch

The GL series branch has an admittance function which can be expressed in rational function as

$$Y(s) = \frac{\frac{1}{L}}{\frac{1}{GL} + s} \quad (46)$$

This may be expressed in pole-residue form as

$$Y(s) = \frac{r}{s - p} \quad (47)$$

Where the pole and residue are both real. By equating coefficients of the functions, values of G and L can be identified. We require that the component values of the series GL circuit be positive i.e.

$$L = \frac{1}{r} > 0 \quad (48)$$

$$G = \frac{-r}{p} > 0 \quad (49)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the admittance function describe a physical admittance. If these requirements are satisfied then a series GL branch can be extracted from the admittance function and implemented as a parallel admittance branch in the ladder network.

2.4.4 GC in series branch

The GC series branch has an admittance function which can be expressed in rational function as

$$Y(s) = \frac{sG}{\frac{G}{C} + s} = G - \frac{\frac{G^2}{C}}{s + \frac{G}{C}} \quad (50)$$

As for the series GL branch, this may be expressed in pole-residue form, including a constant term as

$$Y(s) = K + \frac{r}{s-p} \quad (51)$$

Where the pole and residue are both real. By equating coefficients of the functions, values of G and C can be identified. We require that all the component values of the series GC circuit be positive i.e.

$$G = \frac{r}{p} > 0 \quad (52)$$

$$C = \frac{-r}{p^2} > 0 \quad (53)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the admittance function describe a physical admittance. If these requirements are satisfied then a series GL branch can be extracted from the admittance function and implemented as a parallel admittance branch in the ladder network.

2.4.5 Inductance branch

The inductance branch has an admittance function which consists of a pole at zero i.e.

$$Y(s) = \frac{1}{sL} \quad (54)$$

Thus in the pole-residue expansion we require a pole at zero. In addition we require that inductance be positive i.e.

$$L = \frac{1}{r} > 0 \quad (55)$$

The final requirement is that the remainder function be positive real i.e. the remaining terms of the admittance function describe a physical admittance. If these requirements are satisfied then an inductance branch can be extracted from the admittance function and implemented as a parallel admittance in the ladder network.

2.4.6 Capacitance branch

The capacitance branch has an admittance given by

$$Y(s) = sC \quad (56)$$

Thus in the pole-residue expansion we require a sC term (pole at infinity in the admittance function) with positive capacitance, C. The final requirement is that the remainder function be positive real i.e. the remaining terms of the admittance function describe a physical admittance. If these requirements are satisfied then a capacitance branch can be extracted from the admittance function and implemented as a parallel admittance in the ladder network.

2.4.7 Conductance (Resistance) branch, type 1

The resistance branch of type 1 is identified by a positive constant term in the pole-residue expansion of the admittance. If the remainder function is positive real i.e. the remaining terms of the admittance function describe a physical admittance then a resistance branch can be extracted from the admittance function and implemented as a parallel admittance in the ladder network.

2.4.8 Conductance (Resistance) branch, type 2

The resistance branch of type 2 may be found from the rational function form of the admittance i.e. if

$$Y(s) = \frac{a_0 + a_1 \left(\frac{s}{\omega_0}\right) + a_2 \left(\frac{s}{\omega_0}\right)^2 + \dots}{b_0 + b_1 \left(\frac{s}{\omega_0}\right) + b_2 \left(\frac{s}{\omega_0}\right)^2 + \dots} \quad (57)$$

then we can calculate the conductance, G as

$$G = \frac{a_0}{b_0} \quad (58)$$

2.5 Brune Synthesis

There are circumstances when the ladder network synthesis procedure described above fails i.e. a physical impedance/ admittance function results for which no viable series impedance or parallel admittance branch can be found. In this case the method described by Brune [5] may be applied to allow the process to proceed. Brune's method involves the use of a transformer which can be included in Spice simulations using the K element. The basic process is described here however the associated proofs of the properties of the functions at each stage will not be given. For further details see reference [5].

The ladder network synthesis procedure ensures that poles at zero, poles at infinity, zeros at infinity, zeros at zero, poles on the $s = \pm j\omega$ axis and zeros on the $s = \pm j\omega$ axis have been removed from the impedance function. Brune's method then operates on this remainder function, $Z_r(s)$, as follows:

Stage 1. Find ω_0 such that the real part of the impedance $Z_r(s = j\omega_0)$ is a minimum. The minimum resistance value at this frequency is R_{min} .

Stage 2. Subtract R_{min} from $Z_r(s)$ to give the minimum resistance impedance function

$$Z_1(s) = Z_r(s) - R_{min} \quad (59)$$

i.e. R_{min} is extracted as a series resistance.

The function $Z_1(s)$ is now purely reactive at ω_0 i.e. $Z_1(\omega_0) = jX$. This reactance is set to be the reactance of an inductor, L_A where $L_A = \frac{X}{\omega_0}$. Note that this inductance may be negative.

This series inductance is subtracted from $Z_1(s)$ to give

$$Z_2(s) = Z_1(s) - sL_A \quad (60)$$

$Z_2(s)$ has a zero of order 2 at $s = j\omega_0$ and we can write the admittance $Y_2(s) = \frac{1}{Z_2(s)}$ as

$$Y_2(s) = \frac{1}{Z_1(s) - sL_A} = \frac{\alpha}{s^2 + \omega_0^2} + Y_3(s) \quad (61)$$

From this we can identify a LC admittance branch where $L_B = \frac{1}{\alpha}$ and $C_B = \frac{1}{\omega_0^2 L_B}$

The remainder impedance $Z_3(s) = \frac{1}{Y_3(s)}$ after subtraction of the LC admittance branch has a first order pole at infinity and therefore a series inductance can be extracted i.e.

$$Z_r(s) = Z_3(s) - sL_C \quad (62)$$

where L_C is negative if L_A is positive and vice versa. In addition to this it can be shown that the remainder impedance, $Z_r(s)$ is positive-real.

This series of component identification leads to the circuit shown in figure 7 however this is not a physical circuit in that one of the series inductors L_A or L_B is negative.

The three inductances may be combined into two positive inductors with a mutual inductance term as seen in figure 8. In this circuit the component values can be shown to be

$$\begin{aligned} L_1 &= L_A + L_0 \\ L_2 &= L_B + L_0 \\ K &= 1 \end{aligned} \tag{63}$$

The inductances in the final Brune synthesis circuit are all positive and may therefore be simulated in Spice using L, C R and K elements. Since the remainder impedance is positive-real, the ladder network synthesis procedure can proceed.

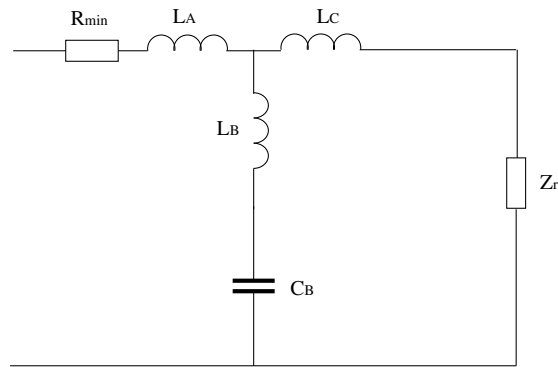


Figure 7 Initial Brune Synthesis Circuit

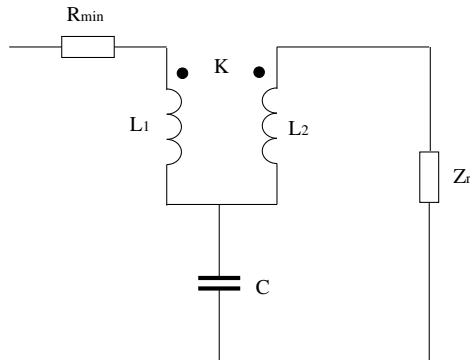


Figure 8 Final Brune Synthesis Circuit

3 Network Synthesis Software

The software may be compiled in a unix type operating system by going into the SRC/ directory and using the command

```
make
```

The makefile may need to be altered for different compilers or to apply specific compilation flags for example. The Makefile in SRC/ is designed to use the f95 gfortran compiler. The executable file, network_synthesis, is moved to the bin directory.

3.1 Running network_synthesis

The network synthesis process required a command line argument which is the file-name of the input impedance function. It is run with the command

```
network_synthesis filename
```

The input impedance function file may take one of two forms; a pole-residue format or a rational function format as described below.

3.1.1 Pole-residue format

The pole-residue input file format contains the pole-residue representation of the input function for the specified model order. The format of the function uses a normalised angular frequency where the angular frequency normalisation (wnorm) is often based on the maximum frequency in the input data used in a curve fitting process (for example the VECTOR_FIT software (https://github.com/chris-smartt/vector_fit)). The function is therefore of the form

$$Z(s) = d + \frac{s}{wnorm}h + \sum_{n=1}^N \frac{c_n}{\frac{s}{wnorm} - a_n} \quad (64)$$

At the end of the file is the (normalised) frequency range and number of samples in the input data.

Below is an example input function in pole-residue form:

```
Vfit filter output
      1 # order
628318530.71795857      # wnorm
8.5000000899999470E-002 # d
27.017695590661077      # h
      pole (a)      residue (c)
-1.12842870712061E-021 0.000000000000000 1.59154949383951E-003 0.000000000000000
      wnorm_min      wnorm_max      nw
1.0000000000000001E-005 1.000000000000000 200
```

3.1.2 Rational function format

The rational function impedance takes the form:

$$Z(s) = \frac{a_0 + a_1 \frac{s}{w_{norm}} + a_2 \frac{s}{w_{norm}}^2 + \dots}{b_0 + b_1 \frac{s}{w_{norm}} + b_2 \frac{s}{w_{norm}}^2 + \dots} \quad (65)$$

The file contains the frequency range of validity of the model (often the frequency range for which data is available for a curve fitting process applied to derive the rational function model e.g. GGI.TLM_filter.fit from the GGI.TLM project (<https://github.com/ggiemr/GGI.TLM>)). Following this is the angular frequency normalisation constant then in turn the numerator order, numerator coefficients, denominator order and denominator coefficients.

Below is an example input function in rational function form:

```
# rational function impedance representation
0.01 1.0 # fmin, fmax (not normalised)
# Impedance filter
1d0 # wnorm
2 # a order, a coefficients follow below:
2d0 2d0 1d0
2 # b order, b coefficients follow below:
1d0 1d0 1d0
```

4 Examples

The TEST_DATA directory contains some examples of the application of the network_synthesis process. These examples consist of impedance functions which may be in rational function form (as produced by GGI_TLM_filter_fit from the GGI_TLM project (https://github.com/ggiemr/GGI_TLM)) or alternatively in pole-residue form as produced by the Vector_fit process (https://github.com/chris-smartt/vector_fit)

The script run_test runs the network synthesis process. The argument of run_test is the filename for the test data. The script runs the network synthesis process and produces a circuit file (ngspice_circuit.cir) suitable for running in ngspice. The script runs ngspice and the output of the ngspice circuit gives the frequency domain voltage due to a unit current source and hence gives the (complex) impedance.

A process 'compare_results' provides a measure of the difference between the complex spice model impedance and the impedance calculated by evaluating the input rational function or pole-zero impedance function. Gnuplot then plots the comparison between the evaluation of the input function and the ngspice result.

Examples:

1. Rational function describing the impedance of a ladder network with R in series with a parallel combination of a C branch and a RL branch.

```
run_test test_RCRL_ladder
```

2. Rational function fit to measured impedance data. The filter fit was produced by GGI_TLM_filter_fit from the GGI_TLM project (https://github.com/ggiemr/GGI_TLM)

```
run_test test_GGI_TLM_filter_fit_Z
```

3. Pole-zero representation of a real capacitor generated by the VECTOR_FIT software (https://github.com/chris-smartt/vector_fit) from measured data

```
run_test test_Vfit.filter
```

References

- [1] https://github.com/chris-smartt/vector_fit
- [2] https://github.com/chris-smartt/Spice_impedance_synthesis
- [3] W. Cauer *Synthesis of Linear Communication Networks* vols 1 and 2, 2nd ed. McGraw-Hill New York, 1958.
- [4] R. M. Foster *A reactance Theorem* Bell System Tech. J, vol 3 1924, pp 259-267
- [5] O. Brune, *Synthesis of a finite two-terminal network whose driving-point impedance is a prescribed function of frequency* J. of Math. Phys., Vol 10, 1931, pp 191-236.
- [6] F. Mukhtar, Y. Kuznetsov, P. Russer *Network modelling with Brune's synthesis* Advances in Radio Science., 9, 2011, pp 91-94