Electrical Two-ports; A Maple Approach

Amir H. Khanshan
Islamic Azad University (IAU) of Iran
khanshan@yahoo.com

Abstract — **Electrical two-ports are an important subject in the course of circuit theory and form a basis for other topics such as electronics or communications circuit design. However, problems involving two-ports almost have a large amount of computational burden. In this paper we present a software package based on Maple to deal with these types of problems. The package can extract various parameters of a two-port network symbolically, directly from its Ladder or Spice-alike description. The package is especially appropriate for educational purposes and contributes to a computer-aided learning environment.**

Index Terms — E-learning, computer-aided instruction, Maple, symbolic circuit analysis, Two-port networks

INTRODUCTION

For circuit theory as a fundamental course in electrical engineering, a symbolic analyzer can be helpful in many regards[1-3]. At present, there are many symbolic manipulation software tools available, none of which have a provision for treating two-port networks. On the other hand, for students and teachers alike, it is more efficient to use general-purpose tools based on a computer algebra system (CAS)[3]. Students may learn these tools beginning in the first semester when they learn math, itself a prerequisite for circuit theory. The evolution of computer algebra systems (see for example [4]), has provided a useful environment for symbolic analysis of electrical circuits. In this scenario, a CAS provides tools to help the user overcome the computational burden that is incidental to implementing or interpreting a theory. Thus, more time and energy can be devoted to the main concept rather than to mathematical manipulations. Problem can be solved step-by-step with user assistance. An apparent feature of this environment is that it will be of no use if the student does not have proper understanding of the underlying concepts.

An important feature of the problems in circuit theory is that many of them can be solved in more than one way. Going through different ways can augment insight into the problem and strengthen conceptual understanding. Our experience shows that many students hesitate to solve a problem in more than one way because of the computational burden this would entail. This burden can be considerably reduced with the aid of an appropriate tool.

In this work, the focus is on Maple[4] and we specifically address two-port networks. Two-port networks are especially important in the sense that a two-port problem can encompass many types of former topics in the course, such as coupled inductors, transformers, filters, transfer functions, etc. Problems involving two-ports, usually impose a large amount of computational burden. The TwoPorts package that has been developed can be used to extract various parameters of two-ports symbolically. It consists of several commands to calculate two-port's characterizations directly from the circuit description. The circuit description can be in a spice-like netlist, or it can be in a simpler "list" structure in Maple for ladder networks. The latter simplifies and speeds up the circuit description for ladder networks, which are frequently used especially for educational purposes. The TwoPorts package can help students easily access the correct answers and verify their own answers. Many students request that the final answer be provided for problems and hesitate to solve problems for which answers are not available. An appropriate tool helps students to be confident of their answers and if not, search for possible mistakes. Otherwise they have to wait until the next session. Specifically if the problem consists of several steps, any mistake can propagate to the next steps and waste lots of time and energy.

The second point is that the teacher is not constrained to pre-selected examples. Such a symbolic analyzer allows the teacher to go through user-requested ones. Also the instructor can use the Two-Port package to design new problems or tune the parameters of a problem so that it leads to the intended answers.

Maple 11 itself acts as a presenter. It differs from other presenters in that each object is still active; commands, animations, etc., are "live," and connected to Maple's math engine.

As analysis is done in a mathematically-based environment, the students can bring other Maple packages and routines to bear on the problem. Not only can Maple be used as an equation-solver, there are many commands for implementing integral transforms, Fourier and Laplace analysis, the computations of linear algebra, matrix manipulation, etc., all available in an integrated system.

In the sequel, we introduce the TwoPorts package for Maple and show its contribution to computer-aided instruction. In the first section, we introduce the package and the usage for its commands. Some simple examples are provided to show applications. These examples are selected from those that are used in class for teaching or are given to students as practice.

PACKAGE'S DESCRIPTION

The Package is loaded through the "with" command and consists of six Maple procedures.

```
> with(TwoPorts);
   [Hparam, Zparam, Yparam, ZtoT, ZtoH, HtoT, HtoZ,
       TtoZ, TtoH, FactorMatrix]
```

The functionality of each command is summarized in Table I. The commands Zparam, Hparam, etc., are used to extract

the parameters of the network symbolically. The parameters are returned as a Matrix (not matrix, note the difference between Matrix and matrix in Maple, see [8]). The usage is as follows:

```
> Zparam(NET, a,b,c,d);
```

where NET is the name of the network described in either Ladder or Spice-like format [4]. The letters *a* and *b* represent input or primary port nodes and *c* and *d* represent output or secondary ports of the network. The order is important as the first node represents the positive voltage. Example 1 elaborates the application.

FIGURE II
EQUIVALENT CIRCUIT FOR A BJT BASED ON Π MODEL

These parameters can be converted to hybrid parameters by ZtoH and to transmission parameters by ZtoT. The Y-matrix can be obtained by inverting the Z-matrix. Symbolic use of each of these procedures would result in the conversion formula. For example:

```
> Z:=<<z11 | z12> , <z21 | z22>>;
```

$$Z := \begin{bmatrix} z11 & z12 \\ z21 & z22 \end{bmatrix}$$

```
> ZtoH(Z);
```

$$\begin{bmatrix} \dfrac{z11\,z22 - z12\,z21}{z22} & \dfrac{z12}{z22} \\ -\dfrac{z21}{z22} & \dfrac{1}{z22} \end{bmatrix}$$

This simple example shows how the extraction of these parameters might be simple and efficient.

TABLE I
POINT SIZES AND TYPE STYLES

Command	Description
Zparam	Return open circuit Impedance parameters
Hparam	Return Hybrid parameters
HtoZ	Converting Hybrid to Impedance parameters
HtoT	Converting Hybrid to Transmission parameters
ZtoH	Converting Hybrid to Impedance parameters
ZtoT	Converting Hybrid to Impedance parameters
TtoZ	Converting Hybrid to Impedance parameters

Example 1. Suppose a simple Π network is given by Figure I. The Network can be simply described in ladder form as :

```
> N:=[0,R1,R3,R2]:
```

The draw command can be used to generate a descent graph of the ladder network:
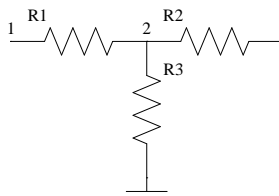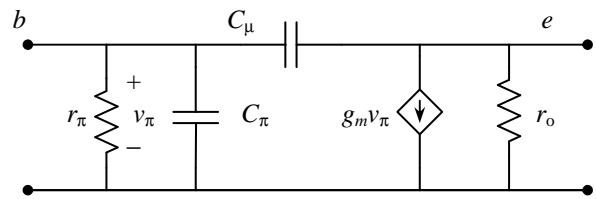
```
> draw(N);
```


FIGURE I
A SIMPLE Π NETWORK FOR EXAMPLE 1.

It warns the user of any possible mistake in circuit description. Also, it helps to find the node numbering. Then the Z parameters are easily obtained via:

```
> Zparam(N,1,0,3,0);
```

$$\begin{bmatrix} R1 + R3 & R3 \\ R3 & R3 + R2 \end{bmatrix}$$

Example 2. As a little more complex network, consider Figure II, which is the equivalent model for a Bipolar Junction Transistor (BJT). As the network has a ladder structure, it can be defined as:

```
> BJT:=[C[pi],[R(r[pi]),C[mu],
G(1,0,g[m])],Ro]:
```

or

```
> BJT2:=[C[pi]&//R(r[pi]),C[mu],
G(1,0,g[m])&//Ro]:
```

which are identical, but generate different schematics under draw. The hybrid parameters obtained by Hparam are shown in the bottom of the page. FactorMatix is simply used to extract the common factors of the entries. Using indexed names generates the response in the format that is popular in textbooks and hand-written materials. This is an important feature, helping students avoid having to concentrate on relating symbols.

```
> H:=simplify(Hparam(BJT,1,0,2,0));
```

$$H := \begin{bmatrix} \dfrac{r_\pi}{s\,C_\pi\,r_\pi + 1 + s\,C_\mu\,r_\pi} & \dfrac{r_\pi\,s\,C_\mu}{s\,C_\pi\,r_\pi + 1 + s\,C_\mu\,r_\pi} \\ -\dfrac{r_\pi\,(s\,C_\mu - g_m)}{s\,C_\pi\,r_\pi + 1 + s\,C_\mu\,r_\pi} & \dfrac{Ro\,s^2\,C_\mu\,C_\pi\,r_\pi + s\,C_\mu\,Ro + s\,C_\pi\,r_\pi + 1 + s\,C_\mu\,r_\pi + g_m\,Ro\,s\,C_\mu\,r_\pi}{Ro\,(s\,C_\pi\,r_\pi + 1 + s\,C_\mu\,r_\pi)} \end{bmatrix}$$

```
> FactorMatrix(H);
```

$$\dfrac{1}{s\,C_\pi\,r_\pi + s\,C_\mu\,r_\pi + 1}\ \&*\ \begin{bmatrix} r_\pi & s\,C_\mu\,r_\pi \\ -r_\pi\,(s\,C_\mu - g_m) & \dfrac{r_\pi\,Ro\,g_m\,s\,C_\mu + s\,C_\pi\,r_\pi + s\,C_\mu\,r_\pi + 1 + s^2\,C_\mu\,Ro\,C_\pi\,r_\pi + s\,C_\mu\,Ro}{Ro} \end{bmatrix}$$

A specific parameter can be extracted easily. For example h_{fe} of the transistor based on the model in Example 2 is

```
> h[fe]=H[2,1];
```

$$h_{fe} = -\frac{r_\pi(-g_m + s\,C_\mu)}{s\,C_\mu\,r_\pi + 1 + s\,C_\pi\,r_\pi}$$

A very simple model can be adopted when we omit C_μ, C_π and R_O. This can be done directly by setting the value 0 for capacitors and "infinity" for R_O. In Maple, "infinity" is a symbol that states the mathematical infinity.

```
> BJT:=[C[pi](0),[R(r[pi]),C[mu](0),
G(1,0,g[m])],Ro(infinity)]:
> H:=simplify(Hparam(BJT,1,0,2,0));
syrup:   There may be an unconnected component.
The following component(s) have zero current:
{Ro, `C[mu]`, `C[pi]`}.
```

$$H := \begin{bmatrix} r_\pi & 0 \\ r_\pi\,g_m & 0 \end{bmatrix}$$

The returned text preceding the response is a warning, not an error. It helps the user to detect any possible cut-off in the circuit due to mistyping.

It should be mentioned that direct insertion of infinity is not identical to taking the limit and in some cases may lead to incorrect results. The better solution is simply to omit R_O from the list.

Example 3. Δ-Y transformation

A Y circuit may be considered as a T network, and a Δ circuit may be considered as a Π network. Define:

```
> N[T]:=[0,R1,R2,R3];
> N[Delta]:=[Ra,Rb,Rc]:
```

For which have the Z characterizations are:

```
> Z1:=Zparam(N[T],1,0,3,0);
```

$$Z1 := \begin{bmatrix} R2 + R1 & R2 \\ R2 & R3 + R2 \end{bmatrix}$$

```
> Z2:=Zparam(N[Delta],1,0,2,0);
```

$$Z2 := \begin{bmatrix} \dfrac{Ra\,(Rb + Rc)}{Rb + Ra + Rc} & \dfrac{Ra\,Rc}{Rb + Ra + Rc} \\ \dfrac{Ra\,Rc}{Rb + Ra + Rc} & \dfrac{Rc\,(Rb + Ra)}{Rb + Ra + Rc} \end{bmatrix}$$

Comparing these matrices leads to the following familiar equations:

```
> solve({Z1[1,1]=Z2[1,1],Z1[1,2]=Z2[1,2]
,Z1[2,1]=Z2[2,1],Z1[2,2]=Z2[2,2]},
{R1,R2,R3});
```

$$R1 = \frac{Ra\,Rb}{Rc + Rb + Ra}, R2 = \frac{Ra\,Rc}{Rc + Rb + Ra}, R3 = \frac{Rb\,Rc}{Rc + Rb + Ra}$$

CALCULATING TWO-PORT PARAMETERS IN A STEP-BY-STEP MANNER

For the network shown in Figure III, we want to find admittance parameters. The main idea is to let the students formulate the problem and use Maple for manipulating symbolic expressions. This gives the student a complete understanding of the analysis process from beginning to end.
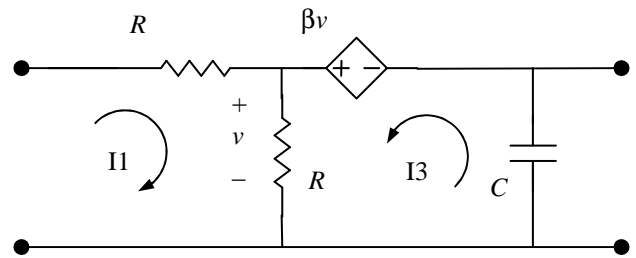

FIGURE III

For this example, inserting two voltage sources at the ports and writing circuit equations leads to the system of equations

```
> eq1:=-V1+R*I1+R*(I1+I3)=0:
> eq2:=-(beta-1)*R*(I1+I3)-V2=0:
> eq3:=V2=(I2-I3)*1/(C*s):
```

which can be solve as

```
> solve({eq1,eq2,eq3},{I1,I2,I3});
```

$$I1 = \frac{-V1 + V2 + \beta\,V1}{R\,(\beta - 1)}, I3 = -\frac{\beta\,V1 - V1 + 2\,V2}{R\,(\beta - 1)},$$

$$I2 = \frac{-\beta\,V1 + V1 - 2\,V2 + R\,\beta\,V2\,C\,s - R\,V2\,C\,s}{R\,(\beta - 1)}$$

The desired parameters are obtained by arranging the above equations. The command `jacobian` in the `linalg` package can be used to arrange the equations. In admittance representation, I1 and I2 are represented by V1 and V2.

```
> assign(%);
> Y:=linalg[jacobian]([I1,I2],[V1,V2]);
```

$$Y := \begin{bmatrix} \dfrac{1}{R} & \dfrac{1}{R\,(\beta - 1)} \\ -\dfrac{1}{R} & \dfrac{-2 + R\,\beta\,C\,s - R\,C\,s}{R\,(\beta - 1)} \end{bmatrix}$$

Rearranging the above equation can lead to other two-port representations. The students can verify the correctness of their answer by inserting the two simple commands:

```
> net:=[0,R,R,E(2,0,beta),C];
> Yparam(net,1,0,3,0):
```

The network can be defined as

```
> net:=[0,R,R,V(beta*v[2]),C];
```

The difference is that the former uses Spice syntax for representing a dependent source, whereas the latter replaces it with an independent source with a symbolic referenced value.

TWO-PORTS CONCATENATION

Consider the circuit in Figure IV, where the two networks are interconnected in a series-series form. The hybrid parameters of each netwok are given as

```
> H1:=<<s | alpha> , <1 | 1>>;
```

$$H1 := \begin{bmatrix} s & \alpha \\ 1 & 1 \end{bmatrix}$$

```
> H2:=<<1 | alpha> , <1 | -1>>;
```

$$H2 := \begin{bmatrix} 1 & \alpha \\ 1 & -1 \end{bmatrix}$$

We are supposed to find the impedance parameters of the augmented network. The students may consider several ways. One way is to apply two current sources at the ports, writing KCL or KVL equations and solving them by `solve` as we did in the preceding section. Arranging the equations with `jacobian` would give the desired matrix. Another idea is first converting H parameters to Z parameters and adding them to find the impedance parameters of the series-series connection. The resulting two-port is in parallel (shunt-shunt configuration[5]) with a network containing R1, R2, C. Thus, the Z matrix is inverted to obtain Y and added to Y-matrix of the second two-port. The resultant Y-matrix is inverted to yield the Z-matrix of the overall network.

```
> Z1:=HtoZ(H1);
```

$$Z1 := \begin{bmatrix} s - \alpha & \alpha \\ -1 & 1 \end{bmatrix}$$

```
> Z2:=HtoZ(H2);
```

$$Z2 := \begin{bmatrix} 1 + \alpha & -\alpha \\ 1 & -1 \end{bmatrix}$$

```
> Z1+Z2;
```

$$\begin{bmatrix} s + 1 & 0 \\ 0 & 0 \end{bmatrix}$$

In fact, the above network is identical to the network in Figure VI in which $Z_i=1+s$. This is a network that does not have any other characterizations. Its impedance matrix can be found by `Zparam([R],1,0,0,0)`. Since the output nodes are short-circuited, C, R1 and R2 are in parallel with $Z_i=1+s$, resulting in $Z_i=1/2\Omega$.
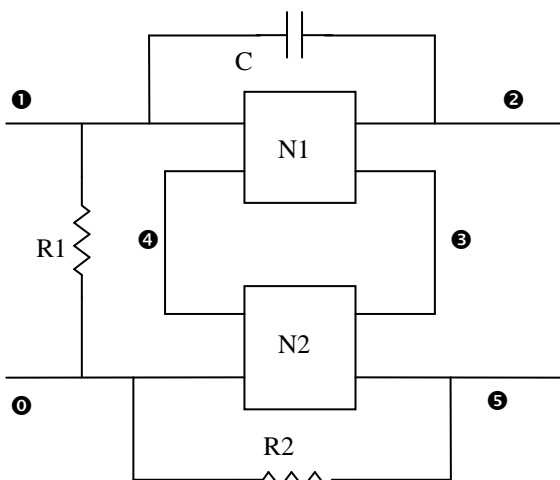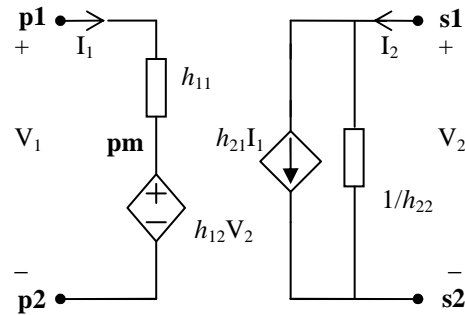
FIGURE IV

FIGURE V
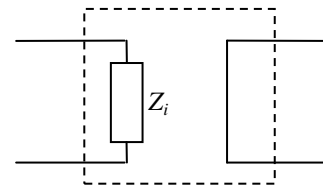EQUIVALENT MODEL FOR A TWO-PORT BASED ON HYBRID CHARACTERIZATION

FIGURE VI

Using TwoPort, the problem can be solved as follows.

```
> Ex:="Series Concatenation
R1 1 0 1
C 1 2 1
X1 1 4 2 3 HE(h11=s,h12=alpha,h21=1,h22=1)
X2 4 0 3 5 HE(h11=1,h12=alpha,h21=1,h22=-1)
R2 0 5 1
.subckt HE p1 p2 s1 s2
R1 p1 pm h11
V1 pm p2 h12*(v[s1]-v[s2])
R2 s1 s2 1/h22
I2 s1 s2 h21*i[V1]
.ends
.end":
```

The sub-circuit HE defines a two-port based on its hybrid equivalent model as shown in Figure V. The parameters are defined symbolically and can be valued when referenced in parentheses after the name HE.

The impedance parameters of the extended circuit are:

```
> Zparam(Ex,1,0,2,5);
```

parsedeck: Analyzing SPICE deck "Series Concatination" (ignoring this line)

$$\begin{bmatrix} \dfrac{1}{2} & 0 \\ 0 & 0 \end{bmatrix}$$

To see if the series-series connection of N1 and N2 is a two-port, two transformers can be set to isolate these two-ports at the output. Ideal transformers can be defined in the netlist by their hybrid representation and no extra subcircuit is needed.

IDEAL OPAMPS

In numerical analyzers, an ideal Opamp is approximated with a voltage-controlled voltage source (VCVS) with a high gain. In a symbolic environment, the response can be found as a function of Opamp gain (A) by taking limit as

A approaches infinity[7]. Figure VII shows a two-port network with an Opamp in it. The network is represented as:

```
> P:="
Rf 1 3 R
R1 1 0 R
RL 3 2 R
X 2 1 3 opamp
.subckt  opamp noninv inv out
V out 0 A*(v[noninv]-v[inv])
.ends
.end":
```
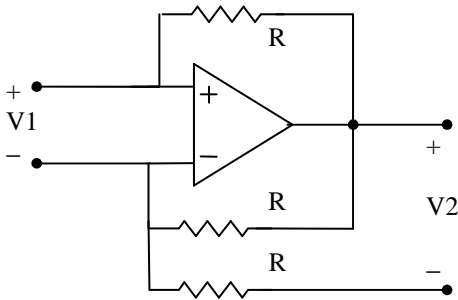

FIGURE VII
A TWO-PORT NETWORK WITH AN IDEAL OPAMP.

The impedance parameters can be obtained through

```
> Z:=Zparam(p36,1,2,3,0);
```

If Opamp is considered to be ideal, the Z-parameters are

```
> Z1:=map(limit,Z,A=infinity);
```

$$Z1 := \begin{bmatrix} 0 & 0 \\ 3\,R & 0 \end{bmatrix}$$

Transmission parameters can be obtained by ZtoT. But this network does not have other characterizations.

CONCLUSION

With the growth in the course content of electrical engineering degree programs, traditional pedagogies should be superseded by computer-aided ones. New educational systems such as e-learning or distance learning are also demanded by these trends. From an educational point of view, it is more efficient for both students and instructors to use general-purpose tools that can be used in different courses. In this paper, we addressed two-port characterization, an important topic in the course of circuit theory and showed how Maple can be used for symbolic manipulation of a prototype problem regarding two-port networks in a step-by-step manner. We also introduced the TwoPorts package, which provides tools for symbolic extraction of two-ports' parameters. The package can be seen as a useful contribution to a computer-aided learning.

REFERENCES

[1] A. I. Beltzer and A. L. Shenkman, "Use of symbolic computation in engineering education", *IEEE Transactions on Education*, vol. 38, no. 2, pp. 177-184, May 1995.

[2] L. P. Huelsman, "Symbolic analysis-a tool for teaching undergraduate circuit theory", *IEEE Transactions on Education*, vol. 39, no. 2, pp. 243-250, May 1996.

[3] Hemanshu R. Pota, "Computer-Aided Analog Electronics Teaching", *IEEE Transactions On Education*, vol. 40, no. 1, February 1997.

[4] Waterloo Maple homepage: http://www.maplesoft.com

[5] Charles A Desoer & E.S.Kuh , Basic Circuit Theory, McGraw-Hill,1998.

[6] Joseph S. Riel, "Syrup 9 Symbolic Circuit Analysis for Maple 9", *Proc. Of MSW2004*, Waterloo, ON, Canada, July 2004.

[7] Amir H. Khanshan, "A Maple approach to electrical engineering problems", *Proc. Of MSW2004*, Waterloo, ON, Canada, July 2004.

[8] Frank Gravan, "The Maple Book", Chapman & Hall/CRC, USA, 2002.