Representation of Microwave Circuits in *SchematicSolver*

Dejan V. Tošić

Abstract—**A new concept of computer representation of microwave circuits is presented and explained by an illustrative example. A software package *SchematicSolver*, written in *Mathematica*, is proposed for representation and symbolic analysis of general finite, linear, time-invariant microwave circuits. The new concept and the corresponding software are intended for practitioners, researchers, educators and students, who design or evaluate microwave circuits.**

Index Terms—**microwave circuit, SchematicSolver, Mathematica, symbolic analysis, signal flow graph**

I. Introduction

With the wireless revolution, brought on mostly by cellular radio technologies, commercial RF and microwave applications have come to dominate the industry. The rapid changes in technological knowledge base make RF and microwave engineering courses important in electrical engineering education and the corresponding syllabi form a solid ground for grasping concepts and phenomena of contemporary RFIC design, communication systems, defense systems, or radar systems.

At RF and microwave frequencies, circuit elements and electromagnetic structures can be viewed as multiport networks. Therefore, a microwave circuit can be represented as an interconnection of multiport components characterized by suitable parameters determined analytically or experimentally (by measurement). Lumped element modeling is not adequate because elements' dimensions are comparable to the signal wavelength. The logical choice of circuit variables to use at these frequencies are traveling waves, referred to as wave signals, rather than total voltages and currents. The most suitable (and from the measurement viewpoint the only possible) device characterization are scattering parameters (S-parameters). [1]-[5]

Microwave circuit analysis is based on formulation of the relevant equations and on their solutions. The microwave circuit equations can be directly formulated from the block diagram (circuit schematic) by writing (a) the element equations and (b) the junction equations. Next, the equations are solved to obtain the circuit response.

Signal flow graphs provide an alternative way of representing and analyzing circuits using scattering waves. In signal flow graphs, each node is represented by a particular wave name and lines connecting these nodes represent interactions between the relative waves. Consequently, signal flow graphs provide much insight to actual physical interactions through the circuits. [3]-[5]

A signal flow graph (SFG) is a graphical representation of the relationships between a set of independent input variables that are linearly related to a set of dependent output variables.

Figure 1 represents a microwave generator connected to a load through a two-port network and the signal-flow-graph representation of this linear system.

The scattering-matrix description of a two-port network $b_1 = s_{11}a_1 + s_{12}a_2$, $b_2 = s_{21}a_1 + s_{22}a_2$ may be represented by a signal flow graph by drawing four nodes to represent the variables a_1, a_2, b_1, and b_2 and connecting lines having transmission factors that show how the inputs a_1 and a_2 feed signals to the output nodes labeled b_1 and b_2.

If we connect a generator to the input, then one additional equation $b_{1g} = \Gamma_{gen}a_{1g} + B_{gen}$ is imposed on the system. To represent this equation on the graph, a node labeled B_{gen} is added along with a directed line segment from this node to a_1 with unity transmission factor. Also an additional line segment from node b_1 to node a_1 with transmission factor Γ_{gen} must be added. If we connect a load to the output, then the load equation $b_{1L} = \Gamma_{load}a_{1L}$ is represented in a similar way. (We assume that the nominal impedances of all ports are Z_0 so $a_{1g} = b_1$, $b_{1g} = a_1$, $a_{1L} = b_2$, and $b_{1L} = a_2$ hold.)

Generally, each port, i, of a microwave network has two nodes, a_i and b_i. Node a_i is identified with a wave entering port i, while node b_i is identified with a wave reflected from port i. The wave signal at a node is equal to the sum of all wave signals entering that node. A branch is a directed path between two nodes, representing signal flow from one node to another. Every branch has an associated s-parameter or reflection coefficient.

The signal flow graph gives an intuitive graphical illustration of the network.

This work was supported in part by the Serbian Ministry of Science and Technological Development under Grant TR-11021.

Dejan V. Tošić is with the University of Belgrade, School of Electrical Engineering, Bulevar kralja Aleksandra 73, PO Box 35-54, 11120 Belgrade, SERBIA (phone: ++381.11.337.007.4; fax: ++381.11.324.86.81; e-mail: tosic@etf.rs; http://kondor.etf.rs/~tosic/).

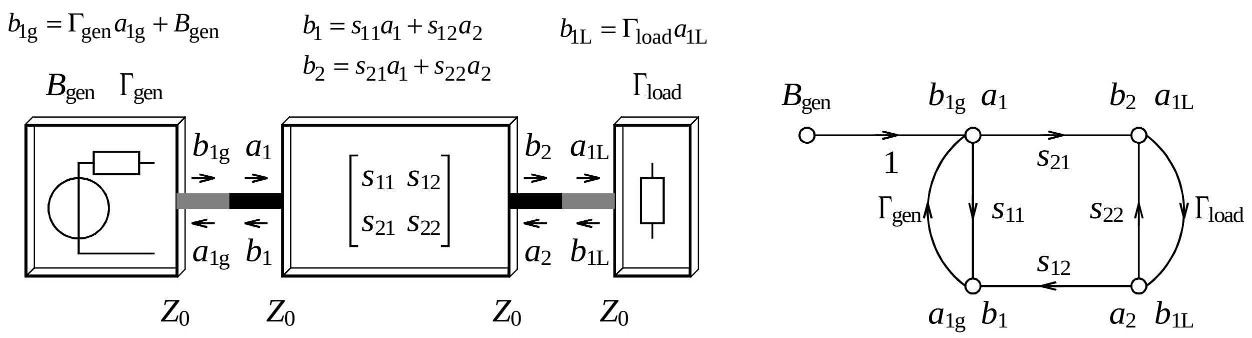

Fig. 1. Model (example) microwave circuit, the element equations, and the corresponding signal flow graph. Nominal impedances of ports are equal. The circuit represents a generator connected to a load through a two-port network.

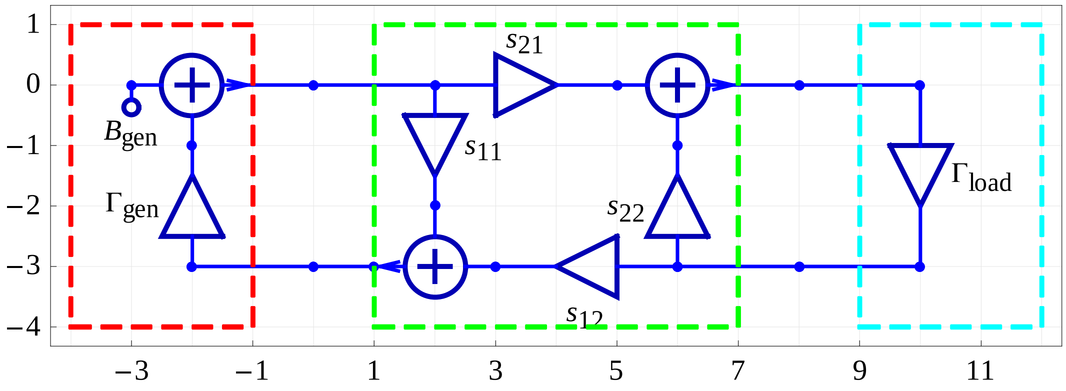

Fig. 2. *SchematicSolver* representation of microwave circuit shown in Fig 1. The schematic generated with *SchematicSolver* directly corresponds to the signal flow graph. The SFG nodes correspond to the *SchematicSolver* adders and take-off points. The SFG branches correspond to the *SchematicSolver* amplifiers; the branch transmission factor corresponds to the amplifier gain.

II. *SCHEMATICSOLVER* REPRESENTATION OF SFG

SchematicSolver [6] is a *Mathematica* [7]-[9] application package that allows you to create symbolic representations of systems. It provides functionality for system drawing, solving, simulating, processing, and implementation. The knowledge embedded in the representation can be used to generate implementation code or to analytically derive system properties, such as transfer functions or impulse responses.

SchematicSolver also automatically generates software implementations of linear and nonlinear discrete systems. This function can process symbolic samples: for a symbolic input sequence, you can compute the symbolic output sequence with both the system parameters and the states specified by symbols. Similarly, the transfer function of a complex multiple-input multiple-output (MIMO) system can be derived in terms of system parameters kept as symbols. [10]-[12]

Symbolic signal processing, an innovative feature of *SchematicSolver* not available in other software, brings you computation of transfer functions as closed-form expressions in terms of symbolic system parameters and can find the closed-form response of schematics. The derived result is the most general because all system parameters, inputs, and initial conditions (states) can be given by symbols.

SchematicSolver can be invoked in a *Mathematica* notebook with the command

```
Needs["SchematicSolver`"]
```

Microwave circuits and elements, the generator (source), the two-port network (e.g., an amplifier model), and the load, can be represented by the *SchematicSolver* elements (Adder, Amplifier, Input, and Line), Fig 2. Specifications of particular microwave elements can be encapsulated in modules as follows:

```
MicrowaveSource[{x_, y_}, S11_, Bg_] :=
 Module[{},
  {{"Input", {x - 3, y}, Bg, "",
    TextOffset → {0, 1}},
   {"Adder", {{x - 3, y}, {x - 2, y - 1},
     {x, y}, {x - 2, y + 2}}, {1, 1, 2, 0}},
   {"Amplifier",
    {{x - 2, y - 3}, {x - 2, y - 1}}, S11},
   {"Line", {{x, y - 3}, {x - 2, y - 3}}},
   {"Polyline", {{x - 4, y - 4}, {x - 1, y - 4},
     {x - 1, y + 1}, {x - 4, y + 1},
     {x - 4, y - 4}},
    PlotStyle → {{Red}, {Red}}}}]
```

```
MicrowaveTwoport[{x_, y_},
   {{S11_, S12_}, {S21_, S22_}}] :=
  Module[{},
   {{"Adder", {{x - 3, y}, {x - 2, y - 1},
      {x, y}, {x - 2, y + 2}}, {1, 1, 2, 0}},
    {"Amplifier",
     {{x - 2, y - 3}, {x - 2, y - 1}}, S22},
    {"Amplifier",
     {{x - 2, y - 3}, {x - 5, y - 3}}, S12},
    {"Line", {{x, y - 3}, {x - 2, y - 3}}},
    {"Adder", {{x - 7, y - 3}, {x - 6, y - 4},
      {x - 5, y - 3}, {x - 6, y - 2}},
     {2, 0, 1, 1}},
    {"Line", {{x - 8, y - 3}, {x - 7, y - 3}}},
    {"Amplifier", {{x - 6, y}, {x - 6, y - 2}},
     S11}, {"Amplifier",
     {{x - 6, y}, {x - 3, y}}, S21},
    {"Line", {{x - 8, y}, {x - 6, y}}},
    {"Polyline", {{x - 7, y - 4}, {x - 1, y - 4},
      {x - 1, y + 1}, {x - 7, y + 1},
      {x - 7, y - 4}},
     PlotStyle → {{Green}, {Green}}}}];
```

```
MicrowaveLoad[{x_, y_}, S11_] :=
 Module[{},
  {{"Amplifier", {{x + 2, y}, {x + 2, y - 3}},
     S11}, {"Line", {{x, y}, {x + 2, y}}},
   {"Line", {{x, y - 3}, {x + 2, y - 3}}},
   {"Polyline", {{x + 1, y - 4}, {x + 4, y - 4},
     {x + 4, y + 1}, {x + 1, y + 1},
     {x + 1, y - 4}},
    PlotStyle → {{Cyan}, {Cyan}}}}]
```

The microwave circuit of Fig. 2 can be assembled from the modules with

```
microwaveCircuit =
  Join[MicrowaveSource[{0, 0}, Γgen, Bgen],
   MicrowaveTwoport[{8, 0},
    {{s11, s12}, {s21, s22}}],
   MicrowaveLoad[{8, 0}, Γload]];
```

and displayed as follows:

```
typeset[S_] := Module[{t = ToString[S]},
  S → Subscript[StringTake[t, 1] //
     ToExpression, StringDrop[t, 1]]]
```

```
microwaveSymbols =
  {a1, a1g, a1L, a2, b1, b2, b1g, b1L, Bgen,
   s11, s12, s21, s22, Γgen, Γin, Γload};
```

```
microwaveCircuit /.
  typeset /@ microwaveSymbols //
 ShowSchematic
```

The response of the circuit from Fig. 2 is computed by the *SchematicSolver* **ContinuousSystemResponse** function:

```
{microwaveResponse, microwaveSignals} =
  ContinuousSystemResponse[
   microwaveCircuit, b];
```

All variables are designated by **b** and a particular variable is identified by its coordinates on the schematic. For example, the input reflection coefficient is computed as follows:

```
inputReflection =
 b[{0, -3}]/b[{0, 0}] /. microwaveResponse //
   Apart[#, s11] & // Simplify
```

```
Γin == b1/a1 == inputReflection /.
   (typeset /@ microwaveSymbols) //
 TraditionalForm
```

$$\Gamma_{\text{in}} = \frac{b_1}{a_1} = s_{11} + \frac{s_{12}\, s_{21}\, \Gamma_{\text{load}}}{1 - s_{22}\, \Gamma_{\text{load}}}$$

The wave signal incident to the load can be found analytically in terms of the circuit parameters and the excitation:

```
loadIncidentWave =
 b[{8, 0}] /. microwaveResponse
```

```
a1L == loadIncidentWave /.
   (typeset /@ microwaveSymbols) // TraditionalForm
```

$$a_{1\text{L}} = -\frac{s_{21}\, B_{\text{gen}}}{-s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}} + s_{11}\, \Gamma_{\text{gen}} + s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}} + s_{22}\, \Gamma_{\text{load}} - 1}$$

which, in the case of a unilateral two-port network, simplifies to

```
loadIncidentWave /. {s12 → 0} /.
    (typeset /@ microwaveSymbols) // Simplify //
 TraditionalForm
```

$$\frac{s_{21}\, B_{\text{gen}}}{(s_{11}\, \Gamma_{\text{gen}} - 1)\,(s_{22}\, \Gamma_{\text{load}} - 1)}$$

For complex-conjugate matching at both ports, the unilateral amplification becomes

```
loadIncidentWaveAmplifier =
 (loadIncidentWave /.
      {s12 → 0, Γgen → Conjugate[s11],
       Γload → Conjugate[s22]}) //.
    {x_. * z_ * Conjugate[z_] → x * Abs[z]^2} //
  Simplify
```

```
loadIncidentWaveAmplifier /.
  (typeset /@ microwaveSymbols) //
 TraditionalForm
```

$$\frac{s_{21}\, B_{\text{gen}}}{\left(\left|s_{11}\right|^2-1\right)\left(\left|s_{22}\right|^2-1\right)}$$

```
unilateralAmplification =
 Coefficient[loadIncidentWaveAmplifier,
  Bgen]
```

```
unilateralAmplification /.
  (typeset /@ microwaveSymbols) //
 TraditionalForm
```

$$\frac{s_{21}}{\left(\left|s_{11}\right|^2-1\right)\left(\left|s_{22}\right|^2-1\right)}$$

The full symbolic response includes wave signals at all nodes and can be obtained in the following way:

```
(Sort[microwaveResponse] /. (typeset /@ microwaveSymbols) /.
      b[i_] → Subscript[b, i]) /. Rule[x_, y_] → x == y /.
    b[i_] → Subscript[b, i] // Together // Column // TraditionalForm
```

$$b_{\{-3,0\}} = B_{\text{gen}}$$

$$b_{\{-2,-1\}} = \frac{-s_{11}\, s_{22}\, B_{\text{gen}}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{11}\, B_{\text{gen}}\, \Gamma_{\text{gen}}+s_{12}\, s_{21}\, B_{\text{gen}}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}}{s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{11}\, \Gamma_{\text{gen}}-s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{22}\, \Gamma_{\text{load}}+1}$$

$$b_{\{0,-3\}} = \frac{-s_{11}\, s_{22}\, B_{\text{gen}}\, \Gamma_{\text{load}}+s_{11}\, B_{\text{gen}}+s_{12}\, s_{21}\, B_{\text{gen}}\, \Gamma_{\text{load}}}{s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{11}\, \Gamma_{\text{gen}}-s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{22}\, \Gamma_{\text{load}}+1}$$

$$b_{\{0,0\}} = -\frac{B_{\text{gen}}\left(s_{22}\, \Gamma_{\text{load}}-1\right)}{s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{11}\, \Gamma_{\text{gen}}-s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{22}\, \Gamma_{\text{load}}+1}$$

$$b_{\{2,-2\}} = -\frac{s_{11}\, B_{\text{gen}}\left(s_{22}\, \Gamma_{\text{load}}-1\right)}{s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{11}\, \Gamma_{\text{gen}}-s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}-s_{22}\, \Gamma_{\text{load}}+1}$$

$$b_{\{3,-3\}} = -\frac{s_{12}\, s_{21}\, B_{\text{gen}}\, \Gamma_{\text{load}}}{-s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{11}\, \Gamma_{\text{gen}}+s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{22}\, \Gamma_{\text{load}}-1}$$

$$b_{\{5,0\}} = \frac{s_{21}\, B_{\text{gen}}\left(s_{22}\, \Gamma_{\text{load}}-1\right)}{-s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{11}\, \Gamma_{\text{gen}}+s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{22}\, \Gamma_{\text{load}}-1}$$

$$b_{\{6,-1\}} = -\frac{s_{21}\, s_{22}\, B_{\text{gen}}\, \Gamma_{\text{load}}}{-s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{11}\, \Gamma_{\text{gen}}+s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{22}\, \Gamma_{\text{load}}-1}$$

$$b_{\{8,-3\}} = -\frac{s_{21}\, B_{\text{gen}}\, \Gamma_{\text{load}}}{-s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{11}\, \Gamma_{\text{gen}}+s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{22}\, \Gamma_{\text{load}}-1}$$

$$b_{\{8,0\}} = -\frac{s_{21}\, B_{\text{gen}}}{-s_{11}\, s_{22}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{11}\, \Gamma_{\text{gen}}+s_{12}\, s_{21}\, \Gamma_{\text{gen}}\, \Gamma_{\text{load}}+s_{22}\, \Gamma_{\text{load}}-1}$$

III. Conclusion

A novel approach to representing and solving microwave circuits is presented. *SchematicSolver*, a software package written in *Mathematica*, is used to represent signal flow graphs that correspond to microwave circuits. A microwave circuit is treated as a linear continuous-time system and is solved by the corresponding *SchematicSolver* functions. The solution is a set of wave signals at all nodes of the schematic, which represents the microwave circuit. Each wave signal is a symbolic closed-form expression in terms of symbolic scattering parameters, reflection coefficients, and excitations.

The *SchematicSolver* representation of a microwave circuit is a list of elements, such as adders, amplifiers, and inputs, that directly map the circuit signal flow graph into the *SchematicSolver* netlist – specification. The visual appearance of the schematic clearly gives insight into the flow of the incident and reflected waves at circuit ports.

References

[1] V. V. Petrović, D. V. Tošić, and A. R. Đorđević, *Microwave Passive Circuits*. Belgrade, Serbia: School of Electrical Engineering, University of Belgrade, 2010. (in Serbian)

[2] A. R. Đorđević and D. V. Tošić, *Microwave Engineering*. Belgrade, Serbia: Academic Mind, 2006. (in Serbian)

[3] D. M. Pozar, *Microwave engineering*, 3rd ed. New York, NY: John Wiley, 2005.

[4] R. E. Collin, *Foundations for Microwave Engineering*. New York, NY: McGraw-Hill, 1992; reprint J. Wiley - IEEE Press, 2001.

[5] N. Kinayman and M. I. Aksun, *Modern Microwave Circuits*. Norwood, MA: Artech House, 2005.

[6] M. D. Lutovac and D. V. Tošić, *SchematicSolver*, [online] http://www.wolfram.com/products/applications/schematicsolver/ Version 2.2 (for *Mathematica* 7) released August 18, 2009.

[7] S. Wolfram, *The Mathematica Book*. Cambridge, MA: Cambridge University Press, Wolfram Media, 2003.

[8] *Mathematica*, http://www.wolfram.com/, Wolfram Research, Inc., USA, Version 7.0.1 released Feb. 18, 2009.

[9] M. Trott, *The Mathematica GuideBook for Symbolics*. New York, NY: Springer Science+Business Media, 2006.

[10] D. V. Tošić and M. D. Lutovac, "Symbolic Simulation of Engineering Systems," In *Proc. 4th IEEE European Conference on Circuits and Systems for Communications* (ECCSC'08), Politehnica University, Bucharest, Romania, July 10-11, 2008, plenary lecture, doi: 10.1109/ECCSC.2008.4611647, pp. 59–68.

[11] M. D. Lutovac and D. V. Tošić, "Symbolic analysis and design of control systems using *Mathematica*," *International Journal of Control*, Special Issue on the Use of Computer Algebra Systems for Computer Aided Control System Design, vol. 79, no. 11, pp. 1368–1381, Nov. 2006.

[12] M. D. Lutovac, D. V. Tošić, and B. L. Evans, *Filter Design for Signal Processing using MATLAB and Mathematica*. Upper Saddle River, NJ: Prentice Hall, 2001. Translated in Chinese, Publishing House of Electronics Industry, PHEI, Beijing, P. R. China 2004.