A CAD Program for the Symbolic and Numerical Analysis of Microwave Electronic Circuits

Alfy Riddle

Macallan Consulting
1583 Pinewood Way
Milpitas, CA 95035

ABSTRACT

A new circuit analysis program, capable of both symbolic and numerical calculations, is described. The ability to derive equations describing a circuit's operation is a great leap forward for engineering analysis. This program, *nodal*, can calculate voltages, currents, S–parameters, and noise coefficients. Since *nodal* uses the *Mathematica*® engine, it takes advantage of the capabilities of *Mathematica* as well as its ability to run on various hardware platforms. A detailed description of this program and examples of its use is given below.

INTRODUCTION

The design of circuit analysis programs is well known [1,2]. To date commercial programs for circuit analysis have performed only numerical analysis. Although symbolic analysis has been described [1,3] the techniques for implementing it have not been widely available until now. Since symbolic analysis derives equations instead of numbers engineers can understand how certain components affect the yield, flatness, temperature stability or noise figure of the circuit they are designing. Engineers can solve for formulas as well as numbers and plots. This is a great leap forward in making circuit theory and analysis more intuitive.

The *nodal* program performs circuit and noise analysis using Y matrix techniques, hence the name *nodal*. This program uses the *Mathematica* program to perform all of the analysis [4]. Since *Mathematica* is capable of symbolic and numerical analysis, the *nodal* program inherits these capabilities. Since the nodal program exists on top of the *Mathematica* engine all of the plots, analysis, and word processing features of *Mathematica* are available to the engineer. This means that a circuit design and its supporting analysis can all be done in a single file. This reduces documentation time and reduces the chances that critical design data will be missing from the collection of plots and paper the engineer has been accumulating. The following sections describe how *nodal* is written and show examples of its use.

THEORY

The simplest way to create a nodal analysis program is to implement methods discussed in the literature directly [1]. We begin with the Y matrix because it is so easy to create, and solve for the Z matrix because all nodes besides the input ports will have zero current (this allows them to be ignored). Although Y matrix analysis is not the most robust for numerical work, the symbolic capability of *Mathematica* will allow variables to be introduced which make normally singular matrices invertable. These variables can then be eliminated in the final equations. An outline of the program flow is (see Fig. 1):

i) Set up an indefinite Y matrix of order equal to the number of circuit nodes plus 1 [1]. The extra node takes care of the ground terminal. This is done by the `initCir[nodes]` function. The indefinite Y matrix is used because it is far easier to set up than a definite Y matrix, but is easily reduced to a definite Y matrix (this reduction will remove the extra node). See Figure 1.

ii) Use function calls to initialize the various components at the desired nodes in the matrix. Actually the components will be placed at the desired nodes plus 1 and then when the matrix is reduced to a definite Y matrix they will be in the expected location. The zeroth node is considered to be ground by the user.

iii) Reduce the indefinite Y matrix to a definite Y matrix by canceling the first row and column (which is the ground node), and moving the rest of the elements down one row and column. This now has the elements properly numbered. This is done by the `defY[]` function.

CH2848-0/90/0000-0355$01.00 © 1990 IEEE

iv) Solve the Y matrix by inverting it into a Z matrix. The Z matrix allows us to specify the terminal current on the desired input port and calculate the voltage anywhere in the network. This is done by the `solveY[]` function.

v) Solve for the noise parameters of two ports [6-8]. The `noiseY[nodes]` function does this (see Fig. 1).

vi) Convert the Z-parameters to S-parameters for the desired ports and port normalization impedances. This calculation is done using matrix techniques as described in [5]. The `defPorts[ nodes, impedances ]` function does this (see Fig. 1).

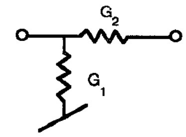
vii) Plot the results using the `smithPlot` function or *Mathematica*'s own 2 and 3D plotting routines.

A set of examples may be seen in Figure 1 corresponding to the above program flow. Passive networks, such as that used in Figure 1i), have noise correlation matrices which may be deduced directly from their **Y** or **Z** matrices. Active networks have more interesting noise correlation matrices.

The noise analysis technique may not be obvious even though it uses very old methods [6]. Therefore, a short derivation follows. The noise voltage at each port may be calculated from the impedance matrix (**Z**) as $\mathbf{V} = \mathbf{Z}\,\mathbf{I_n}$, where $\mathbf{I_n}$ represents the noise currents entering each node. Since $\mathbf{I_n}$ is a random variable in the frequency domain, so is **V**. Since the noise currents at various nodes may be related, there will be some correlation between the various currents and between the resulting voltages. A complete description of our noisy n-port is given by the correlation matrix $\mathbf{C_z} = \langle \mathbf{V}\,\mathbf{V}^\dagger \rangle$, where the $\langle\rangle$ denote averaging and the $\dagger$ denotes the Hermitian conjugate (transpose conjugate). Note that **V** is a column vector and $\mathbf{V}^\dagger$ is a row vector. The resulting product (outer product) yields a matrix.

As a result of the averaging $\mathbf{C_z}$ is filled with numbers rather than random variables. $\mathbf{C_z}$ is easily calculated by realizing $\langle \mathbf{V}\,\mathbf{V}^\dagger \rangle = \langle \mathbf{Z}\,\mathbf{I_n}\,\mathbf{I_n}^\dagger\,\mathbf{Z}^\dagger \rangle = \mathbf{Z} \langle \mathbf{I_n}\,\mathbf{I_n}^\dagger \rangle \mathbf{Z}^\dagger$, and $\langle \mathbf{I_n}\,\mathbf{I_n}^\dagger \rangle = \mathbf{C_y}$, the correlation matrix for the original admittance matrix. $\mathbf{C_y}$ is easily calculated for passive circuits as it is 4kT Re{Y} [8]. This follows directly from the fact that the conductances generate the noise currents, $\mathbf{I_n}$. Active networks must construct $\mathbf{C_y}$ from formulas for noise currents [7]. The noise matrix $\mathbf{C_y}$ is built up the same way that the indefinite Y matrix is built, by adding in the effective conductance of each element. Note that since only two nodes will be used for noise figure calculations, the **Z** matrix used in obtaining $\mathbf{C_z}$ need only have the two rows pertaining to those nodes. The resulting 2x2 matrix $\mathbf{C_z}$ can easily be transformed to a form where the noise figure, optimum source admittance, and effective noise, resistance can be calculated [8].

i)



Linear | Noise

ii)
$$Y_i = \begin{bmatrix} G_1 & -G_1 & 0 \\ -G_1 & G_1+G_2 & -G_2 \\ 0 & -G_2 & G_2 \end{bmatrix} \qquad C_{yi} = 4kT \begin{bmatrix} G_1 & -G_1 & 0 \\ -G_1 & G_1+G_2 & -G_2 \\ 0 & -G_2 & G_2 \end{bmatrix}$$

iii)
$$Y_d = \begin{bmatrix} G_1+G_2 & -G_2 \\ -G_2 & G_2 \end{bmatrix} \qquad C_{yd} = 4kT \begin{bmatrix} G_1+G_2 & -G_2 \\ -G_2 & G_2 \end{bmatrix}$$

iv)
$$Z = Y_d^{-1} = \begin{bmatrix} \frac{1}{G_1} & \frac{1}{G_1} \\ \frac{1}{G_1} & \frac{1}{G_1}+\frac{1}{G_2} \end{bmatrix}$$

v)
$$C_z = Z\,C_{yd}\,Z^\dagger = 4kT \begin{bmatrix} \frac{1}{G_1} & \frac{1}{G_1} \\ \frac{1}{G_1} & \frac{1}{G_1}+\frac{1}{G_2} \end{bmatrix}$$

vi)
$$S = \begin{bmatrix} \frac{1-50\,G_1-2500\,G_1G_2}{1+50\,G_1+100\,G_2+2500\,G_1G_2} & \frac{100\,G_2}{1+50\,G_1+100\,G_2+2500\,G_1G_2} \\ \frac{100\,G_2}{1+50\,G_1+100\,G_2+2500\,G_1G_2} & \frac{1+50\,G_1-2500\,G_1G_2}{1+50\,G_1+100\,G_2+2500\,G_1G_2} \end{bmatrix}$$

Fig 1. Example of the matrices used within the *nodal* package. The roman numerals correspond to the program flow description above.

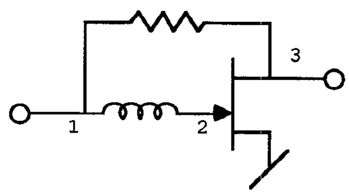
The FET and bipolar models were constructed by building up a circuit equivalent to the device and then using *Mathematica*'s symbolic abilities to create the formulas for directly filling the indefinite Y matrix. The FET and BJT noise models are valid for a broad range of frequencies [9-10]. The FET noise model includes the usual parameters, and allows for gate-drain capacitance as well as for transconductance delay, both of which are very important in modern FETs.

Examples

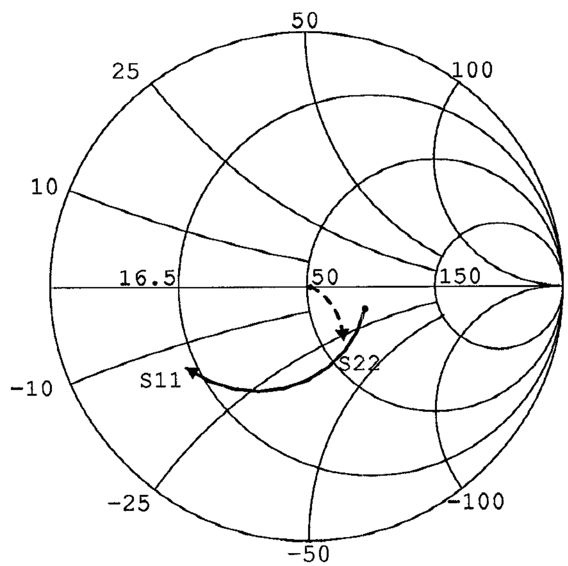
Figure 2 shows some results of an interactive session with the *nodal* package. A simple FET amplifier is defined. The latter part of the FET definition contains the noise coefficients: P,R, and C.

```
initCir[3];
ind[{1,2}, 0.25];
fet[{2,0,3},8.5,180,0.4,0.04,0.15,0.05,
            5.0, {0,1.0,0.5,0.65}];
res[{1,3},200];
defY[];
smithPlot[{1,3}, Range[1,10],
          zLabels->50, LineDashed->True ]
```

a)



b)



c)

Fig 2. Circuit analysis example for *nodal* package: a) input file for a circuit; b) schematic for the circuit specified in a); c) Smith Chart output of S_{11} and S_{22} for the above circuit.

The next example shows the most powerful ability of this package. Circuits may be analyzed symbolically. Figure 4 contains the circuit description and *Mathematica* commands for analyzing a voltage controlled current source, feedback resistor, and input shunt resistor. The circuit analysis returns the transfer function (S_{21}) for the symbolic parameters given (`gm`, `Rin` and `Rfb`). The -3db frequency for the voltage controlled current source was declared to be `Infinity`. The `defPorts[]` command is part of the *nodal* package and is used to specify which nodes, 1 and 2 in this case, are used for the S-parameter analysis and the characteristic impedances for those ports. The number of ports is arbitrary. The `Simplify[]` command is part of *Mathematica* and is used to force symbolic expressions into a more compact form. While the symbolic S_{21} analysis took only seconds to perform on a MacII computer, the minimum noise figure equation took almost a minute to solve.

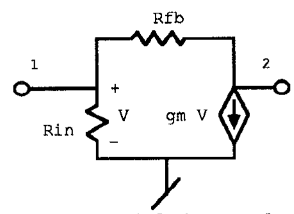


Figure 3. Circuit being analyzed.

```
In[1]:=
    initCir[2];
    vccs[{1,0,0,2},gm,Infinity];
    res[{1,2},Rfb];
    res[{1,0},Rin];
    defY[];
    solveY[];
    spar = defPorts[{1,2},{50,50}];
    Simplify[ spar[[2,1]] ]

Out[1]=
          -100 Rin (-1 + Rfb gm)
---------------------------------------------
2500 + 50 Rfb + 100 Rin + Rfb Rin + 2500 Rin gm

In[2]:=
     Simplify[ noise[{1,2}][[1]] ]
```

```
Out[2]=
                        2        2                                              2
      2 Rfb + Rin + Rfb  Rin gm  - 2 Sqrt[Rfb] Sqrt[Rfb + Rin] Sqrt[1 + Rfb Rin gm ]
(10 Log[ ----------------------------------------------------------------------------- ]) / Log[10]
                                              2
                                  Rin (-1 + Rfb gm)
```

Fig. 4 Input commands and *Mathematica* output for the symbolic analysis of a circuit. The transfer function (S_{21}) is returned. The second output is the minimum noise figure. The *Log* function is a natural log, hence the division by Log[10].

Noise analysis has become standard in microwave software. Crucial to the accuracy of the noise analysis is the model the analysis is based on. *nodal* couples the basic theory of FET noise current sources [9] with such important terms as feedback capacitance and transconductance delay in order to give a truly accurate FET noise model. The P, R, and C noise coefficients are used in nodal to define the noise currents for the FET [9]. Typically the *nodal* FET model shows more variation over frequency than FET models based on approximate formulas. The FET below (see Fig. 5) comes from an example in the literature [12].

In[3] :=

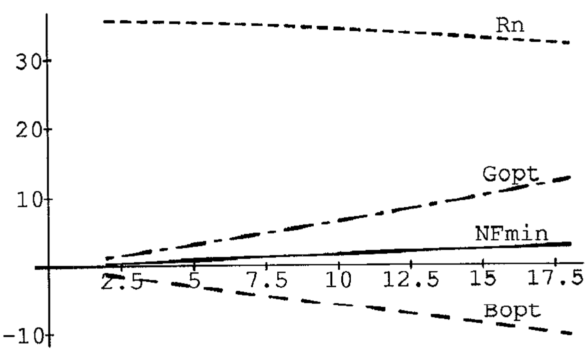
```
initCir[6];
z = 0.63;   (* 190μ Wg, normalized to 300μ *)
res[{1,2}, 4.0/z];  (* Rg *)
fet[{2,3,4},4.0/z,1.0/(0.0029 z),0.3 z,0.02 z,
    0.085 z,0.045 z,3,{0,0.65,0.25,0.55}];
res[{3,6}, 3.5/z];  (* Rs *)
res[{4,5}, 3.8/z];  (* Rd *)
ind[{6,0}, 0.02];   (* Ls *)
defY[];
sparOutput[{1,5}, Range[2,18,2]];
```

f	|S11|	<S11	|S12|	<S12	|S21|	<S21	|S22|	<S22	K
2.	0.99	-13.9	0.0162	82.6	2.2_	166.	0.853	-5.5	0.128
4.	0.962	-27.2	0.0313	75.7	2.13	152.	0.844	-10.8	0.255
6.	0.922	-39.7	0.0443	69.6	2.01	139.	0.831	-15.8	0.382
8.	0.876	-51.2	0.055	64.5	1.88	127.	0.817	-20.5	0.508
10.	0.83	-61.5	0.0636	60.4	1.75	117.	0.803	-24.9	0.632
12.	0.786	-70.7	0.0703	57.3	1.62	107.	0.791	-29.	0.755
14.	0.748	-79.	0.0756	55.	1.49	97.7	0.78	-33.	0.877
16.	0.713	-86.4	0.0799	53.5	1.38	89.5	0.77	-36.8	0.995
18.	0.684	-93.1	0.0834	52.6	1.28	82.	0.762	-40.5	1.11

a)

```
noiseYPlot[{1,5}, Range[2,18,2],
                   LineDashed->True];
```

b)



c)

Fig 5. GaAs FET model taken from the literature [12]. a) nodal circuit file and S-parameter analysis; b) noise analysis command; c) resulting plot of noise parameters versus frequency.

CONCLUSIONS

A new circuit and noise analysis program has been described. This program is unique in its capabilities for both symbolic and numerical analysis. The symbolic capability allows the user to derive both transfer functions and formulas. State of the art models for both FETs and BJTs are included in the program. Because this program uses the *Mathematica* engine, all of the features and benefits of *Mathematica* are available to the user. Version 1.2 of this program is being distributed at cost for research purposes. The complete source code for *nodal* can be viewed and modified by the user. This makes the program useful in education and research as well as in applied engineering.

REFERENCES

[1] Vlach, J., and Singhal, K., *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, New York, 1983.

[2] Monaco, V.A., and Tiberio, P.,"Computer-Aided Analysis of Microwave Circuits," IEEE Trans MTT, pp. 249 - 263, Mar 1974.

[3] Liberatore, A. , and Manetti, S. , "SAPEC - A Personal Computer Program for the Symbolic Analysis of Electric Circuits," IEEE ISCAS, pp. 897 - 900, 1988

[4] Wolfram Research Incorporated, P.O. Box 6059, Champaign, IL. 61821, 1988.

[5] Ha,T.T., *Solid-State Microwave Amplifier Design*, Wiley, 1981.

[6] Haus, H.A., and Adler, R.B., *Circuit Theory of Linear Noisy Networks*, MIT Press, 1959.

[7] Ambrózy, A., *Electronic Noise*, McGraw-Hill, New York, 1982.

[8] Hillbrand, H., and Russer, P.H., "An Efficient Method for Computer Aided Noise Analysis of Linear Amplifier Networks," IEEE Trans CAS, pp. 235 - 238, Apr 1976.

[9] Pucel, R.A., Haus, H.A., and Statz, H., "Signal and Noise Properties of Gallium Arsenide Field-Effect Transistors," *Advances in Electronics and Electron Physics*, vol. 38, pp. 195 - 265, 1974.

[10] Hawkins, R.J., "Limitations of Nielsen's and Related Noise Equations Applied to Microwave Bipolar Transistors, and a New Expression for the Frequency and Current Dependent Noise Figure," Solid-State Electronics, pp. 191 - 196, 1977.

[11] Ghausi, M.S., *Principles and Design of Linear Active Circuits*, McGraw-Hill, New York, 1965.

[12] Rohde, U.L., Pavio, A.M., and Pucel, R.A., "Accurate Noise Simulation of Microwave Amplifiers Using CAD," Microwave Journal, pp. 130 - 141, Dec. 1988.