Using Maxima to solve symbolic circuit equations

December 6, 2017

J.L.

1 Introduction

Electric circuits are typically modelled using a set of linear equations, which need to be solved either numerically or symbolically to find out voltages or currents in specific part of a circuit. For numerical analysis one can use SPICE simulators or numerical software like `Octave`, for example.

Circuit equations in symbolic form are typically solved by hand calculations using pen and paper, either from matrix models or simply using the laws of electromagnetism directly. Although the matrix model itself is a symbolic equation of the circuit, detailed simplified expressions for some specific quantities often need to be solved from the matrix equation to examine the behaviour of a certain node voltage or branch current. Symbolic equations serve their purpose best when they are short and compact, therefore one seldom describes complex circuits by symbolic equations.

In some cases it is beneficial to solve symbolic equations even for a bit more complex circuits, because this can save time in following numerical analysis. Lenghty pen and paper calculations are prone to errors and take a considerable amount of time to simplify to the most compact or clearest form. Therefore it would be nice if there would exist some tool or a process which could assist in evaluating the symbolic equations for more complex circuit models.

This document describes how a symbolic computer algebra system (CAS) can be applied for solving or simplifying circuit matrix equations in symbolic form. In this context `Maxima` is used, since it is available for free for everyone using Linux or Windows operating system.

More background information about `Maxima` can be found from the official project website `http://maxima.sourceforge.net/`. Nowadays `Maxima` is completely an open-source project for which everyone interested can take part on the development.

2 First example: the common-emitter BJT amplifier

Let's roll up our sleeves and start with an example of the basic common-emitter transistor amplifier circuit, for which the small-signal transfer function will be evaluated symbolically using the `Maxima` computer algebra system. The circuit and its small-signal model are shown in Figures 1 and 2 respectively.

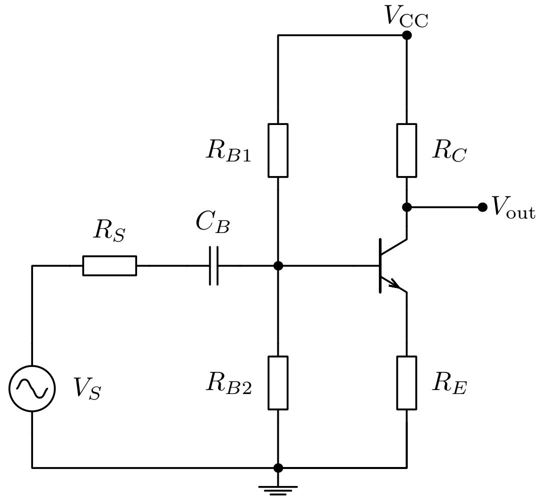

Figure 1: A common-emitter BJT amplifier

The base voltage divider consisting of resistors R_{B1} and R_{B2} is combined as a single base resistor R_B in the small-signal model. In the small-signal model the voltage divider resistors appear in parallel, so the basic parallel resistance formula can be used.

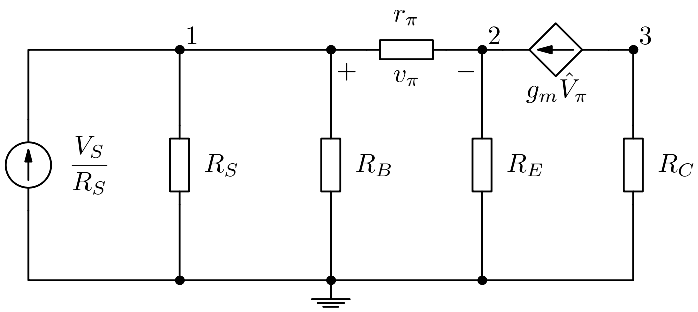

Figure 2: A small-signal model of the common-emitter BJT amplifier

The matrix equation for the small-signal circuit is given in Equation (1). The transfer function $\frac{V_{\text{IN}}}{V_{\text{OUT}}}$ will be solved using Cramer's rule of dividing two determinants.

$$\begin{bmatrix} \frac{1}{R_S} + \frac{1}{R_B} + \frac{1}{r_\pi} & -\frac{1}{r_\pi} & 0 \\ -\frac{\beta_F + 1}{r_\pi} & \frac{\beta_F + 1}{r_\pi} + \frac{1}{R_E} & 0 \\ \frac{\beta_F}{r_\pi} & -\frac{\beta_F}{r_\pi} & \frac{1}{R_C} \end{bmatrix} \times \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} \frac{V_S}{R_S} \\ 0 \\ 0 \end{bmatrix}. \tag{1}$$

From the matrix equation (1) one can write a `Maxima` batch file (a normal text file), which can be run under the `Maxima` command line interpreter using the command `batch("commonemitter.txt");`. The contents of the batch file is listed below.

```
stardisp: true$
ratmx: true$
ratfac: false$
matrix([1/RS+1/RB+1/rpi,-1/rpi,0],
       [-(BF+1)/rpi,(BF+1)/rpi+1/RE,0],
       [BF/rpi,-BF/rpi,1/RC]);
DET_den: determinant(%);

matrix([1/RS+1/RB+1/rpi,-1/rpi,1/RS],
       [-(BF+1)/rpi,(BF+1)/rpi+1/RE,0],
       [BF/rpi,-BF/rpi,0]);
DET_num: determinant(%);
DET_num/DET_den;
```

Since the Maxima notation is shown here the first time, some clarifications need to be given for the functions used in the batch file: The `stardisp` variable is set to true, to show an asterisk * as a multiplication symbol. The `ratmx` true means that matrix operations for polynomials are performed in Canonical Rational Expressions (CRE). The `matrix([],[],[]);` command defines the matrix and the `determinant()` command evaluates the determinant of the provided matrix expression. The '%' mark refers to the result of previous line. For solving the voltage V_3 using Cramer's rule, two determinants need to be evaluated and then divided.

The output from running the script is

```
                        BF*RB*RC
- -----------------------------------------------------
  (rpi + RB + (BF + 1)*RE)*RS + (1 + BF)*RB*RE + rpi*RB
```

This is the correct result, but it also reveals that the simplification algorithms in CAS software do not always manage to perform the most optimal solution. The same expression can be simplified further to the form

$$-\frac{\beta_F R_B R_C}{(R_S+R_B)[r_\pi+R_E(1+\beta_F)]+R_S R_B}. \tag{2}$$

Well, on the other hand it is still nice to have the brain defeat the machine.

3 Case example: BJT amplifier bias and transfer function

As a second example we try to analyse the collector-to-base feedback bias circuit with emitter resistor included. The complete circuit schematic for the circuit under analysis is drawn in Figure 3.

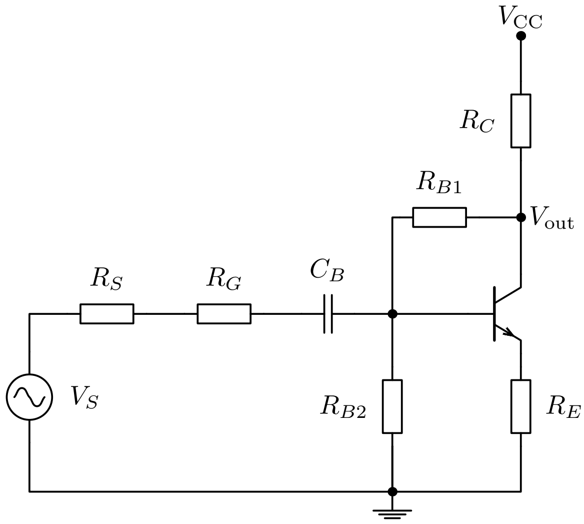

Figure 3: A collector-to-base bias arrangement with voltage divider at the base and emitter resistor included

Let's analyse the DC model of the circuit first. The DC model configuration is drawn to Figure 4, which includes the notations for all the necessary currents and voltage nodes. Before the node voltage expressions can be solved with `Maxima`, the circuit model needs to be written in matrix form. For that purpose, some pre-analysis is required.

According to the Kirchhoff's current rule, the current equation for the voltage node V_B is:

$$I' = I + I_B = I + \frac{I_E}{\beta_F + 1}, \tag{3}$$

and for the voltage node V_C:

$$I'_C = I_C + I' = I_E \frac{\beta}{\beta_F + 1} + I'. \tag{4}$$

After the current equations are written down for each voltage node, the next step is to express the currents using the supply voltage and the node voltages. The currents appearing in the current equations can be expressed with

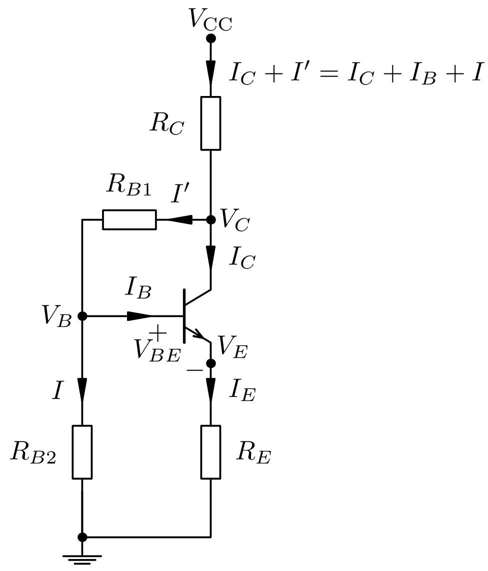

Figure 4: The DC model of the collector-to-base bias circuit

respect to the node voltage as:

$$I'_C = \frac{V_{CC} - V_C}{R_C} \quad ; \quad I_E = \frac{V_B - V_{BE}}{R_E} \quad ; \quad I' = \frac{V_C - V_B}{R_{B1}} \quad ; \quad I = \frac{V_B}{R_{B2}}$$

and after substituting these voltage equations to the current equations,

$$\frac{V_{CC} - V_B}{R_C} = \frac{V_B - V_{BE}}{R_E}\frac{\beta}{\beta+1} + \frac{V_C - V_B}{R_{B1}}$$
$$\frac{V_C - V_B}{R_{B1}} = \frac{V_B}{R_{B2}} + \frac{V_B - V_{BE}}{R_E}\frac{1}{\beta+1}.$$

These two equations can be rearranged into a matrix equation from where the node voltages can be solved systematically:

$$\begin{bmatrix} \frac{\beta}{\beta+1}\frac{1}{R_C} + \frac{1}{R_{B1}} & \frac{\beta}{\beta+1}\frac{1}{R_E} - \frac{1}{R_{B1}} \\ -\frac{1}{R_{B1}} & \frac{1}{R_{B1}} + \frac{1}{R_{B2}} + \frac{1}{\beta+1}\frac{1}{R_E} \end{bmatrix} \times \begin{bmatrix} V_C \\ V_B \end{bmatrix} = \begin{bmatrix} \frac{V_{CC}}{R_C} + \frac{\beta}{\beta+1}\frac{V_{BE}}{R_E} \\ \frac{1}{\beta+1}\frac{V_{BE}}{R_E} \end{bmatrix}.$$

Now that the matrix equation is ready, one can write a `Maxima` batch file to solve the unknown node voltages from the matrix. The batch script below shows an example of how the collector voltage V_C is solved from the matrix equation, again utilizing the Cramer's rule.

```
stardisp: true$
ratmx: true$
ratfac: false$
matrix([(BF/(BF+1))*(1/RC) + 1/RB1, (BF/(BF+1))*(1/RE) - 1/RB1],
```

```
       [-1/RB1, 1/RB1 + 1/RB2 + (1/(BF+1))*(1/RE)]);
DET_den: determinant(%);

matrix([VCC/RC + (BF/(BF+1))*(VBE/RE), (BF/(BF+1))*(1/RE) - 1/RB1],
       [(1/(BF+1))*(VBE/RE), 1/RB1 + 1/RB2 + (1/(BF+1))*(1/RE)]);
DET_num: determinant(%);
DET_num/DET_den;
```

After running the batch script in `Maxima`, the result is printed out to the console as follows:

```
           2
((1+2*BF+BF)*(((1+BF)*RB2+BF*RB1)*RC*VBE+(((BF+1)*RB1+(BF+1)*RB2)*RE+RB1*RB2)*VCC))
              2              2        2                         2
/((BF+1)*(((BF+BF)*RB1+(BF+BF)*RB2+(BF+2*BF+1)*RC)*RE+(1+2*BF+BF)*RB2*RC+BF*RB1*RB2))
```

This does not look very elegant. The thing that bothers the most is the polynomial expansions of $(\beta_F + 1)^2$, which cannot be prevented (at least to the authors knowledge). By taking $(\beta_F + 1)^2$ as a common factor in numerator and denominator and simplifying manually, the expression becomes

```
((1 + BF)*RB2 + BF*RB1)*RC*VBE + ((BF + 1)*(RB1 + RB2)*RE + RB1*RB2)*VCC
/((BF*(RB1 + RB2) + (BF+1)*RC)*RE + (BF+1)*RB2*RC + BF/(BF+1)*RB1*RB2)
```

This is better, but still not the optimal form of solution. Taking the simplification one step further, we finally get

$$V_C = \frac{V_{CC}[R_E(\beta_F+1)(R_{B1}+R_{B2}) + R_{B1}R_{B2}] + V_{BE}R_C[R_{B2}(\beta_F+1)+\beta_F R_{B1}]}{\frac{\beta_F}{\beta_F+1}[R_E(\beta_F+1)(R_{B1}+R_{B2}) + R_{B1}R_{B2}] + R_C(\beta_F+1)(R_E+R_{B2})}. \tag{5}$$

Here we did not get much help from `Maxima`, but it lead us to the right direction at least.

Next one can try to tame the AC model of the same circuit. The small-signal model of the circuit shown in Figure 3 is drawn in Figure 5.

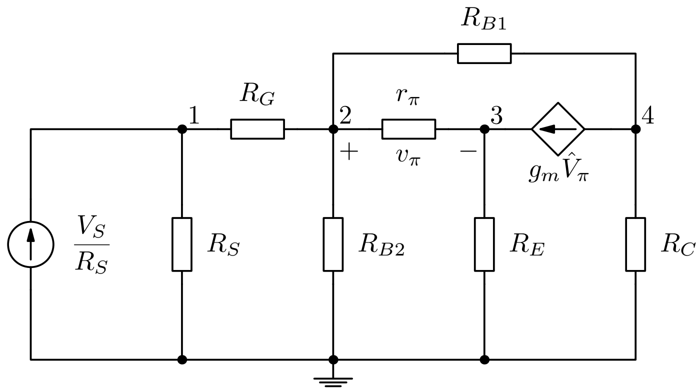

Figure 5: Small-signal model of the collector-to-base bias circuit

The matrix equation describing this circuit is

$$\begin{bmatrix} \frac{1}{R_S}+\frac{1}{R_G} & -\frac{1}{R_G} & 0 & 0 \\ -\frac{1}{R_G} & \frac{1}{R_G}+\frac{1}{r_\pi}+\frac{1}{R_{B1}}+\frac{1}{R_{B2}} & -\frac{1}{r_\pi} & -\frac{1}{R_{B1}} \\ 0 & -\frac{\beta_F+1}{r_\pi} & \frac{\beta_F+1}{r_\pi}+\frac{1}{R_E} & 0 \\ 0 & \frac{\beta_F}{r_\pi}-\frac{1}{R_{B1}} & -\frac{\beta_F}{r_\pi} & \frac{1}{R_{B1}}+\frac{1}{R_C} \end{bmatrix} \times \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} \frac{V_S}{R_S} \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

For evaluating the gain transfer function $\frac{V_{\text{IN}}}{V_{\text{OUT}}}$ from this matrix equation using Cramer's rule, the following `Maxima` batch script can be written:

```
stardisp: true$
ratmx: true$
ratfac: false$
matrix([1/RS+1/RG,-1/RG,0,0],
       [-1/RG,1/RG+1/rpi+1/RB1+1/RB2,-1/rpi,-1/RB1],
       [0,-(BF+1)/rpi,(BF+1)/rpi+1/RE,0],
       [0,BF/rpi-1/RB1,-BF/rpi,1/RB1+1/RC]);
DET_den: determinant(%);

matrix([1/RS+1/RG,-1/RG,0,1/RS],
       [-1/RG,1/RG+1/rpi+1/RB1+1/RB2,-1/rpi,0],
       [0,-(BF+1)/rpi,(BF+1)/rpi+1/RE,0],
       [0,BF/rpi-1/RB1,-BF/rpi,0]);
DET_num: determinant(%);
DET_div: DET_num/DET_den;
num(DET_div);
```

```
collectterms(denom(DET_div),RE);
```

After running the script, the expression for the numerator is printed as

```
RB2*RC*((1 + BF)*RE - BF*RB1 + rpi)
```

and for the denominator the printed expression is

```
(rpi*RB1 + (RB1 + rpi)*RB2 + ((1 + BF)*RB2 + rpi)*RC
 + ((1 + BF)*RC + (1 + BF)*RB2 + (1 + BF)*RB1)*RE)*(RG + RS)
 + ((BF + 1)*RB1*RB2 + (BF + 1)*RB2*RC)*RE + rpi*RB2*RC + rpi*RB1*RB2
```

When continuing the simplification process manually, this equation simplifies to

$$\frac{R_{B2}R_C[r_\pi + (\beta_F + 1)R_E - \beta_F R_{B1}]}{(R_G + R_S)\{(R_C + R_{B2} + R_{B1})[r_\pi + R_E(\beta_F + 1)] + R_{B2}[R_{B1} + (\beta_F + 1)R_C]\} + [r_\pi + R_E(\beta_F + 1)](R_{B1} + R_C)R_{B2}}$$

So again `Maxima` failed to produce the optimal form of the solution, but if the complete transfer function would have been evaluated from scratch using pen and paper, it would have taken several hours to complete the task.

4 Filter applications

Transfer functions for passive RLC filter circuits are typically good candidates to be solved by `Maxima`, because there the terms can be grouped according to the powers of Laplace variable s.

For a real life application, we will analyse the transfer function of the classic Baxandall tone stack, or actually the passive version of it, which is also known as the James tone stack. The circuit for analysis is shown in Figure 6, with added input and load resistances R_S and R_L and a signal source V_S. For the analysis, the potentiometers R_B and R_T are split into two separate resistors for each R_{B1}, R_{B2}, R_{T1} and R_{T2}.

Unlike the other examples given in this context, this circuit is analysed using the current-loop method instead of the more common nodal analysis. The current loops are marked in Figure 6, and because there is six current loops

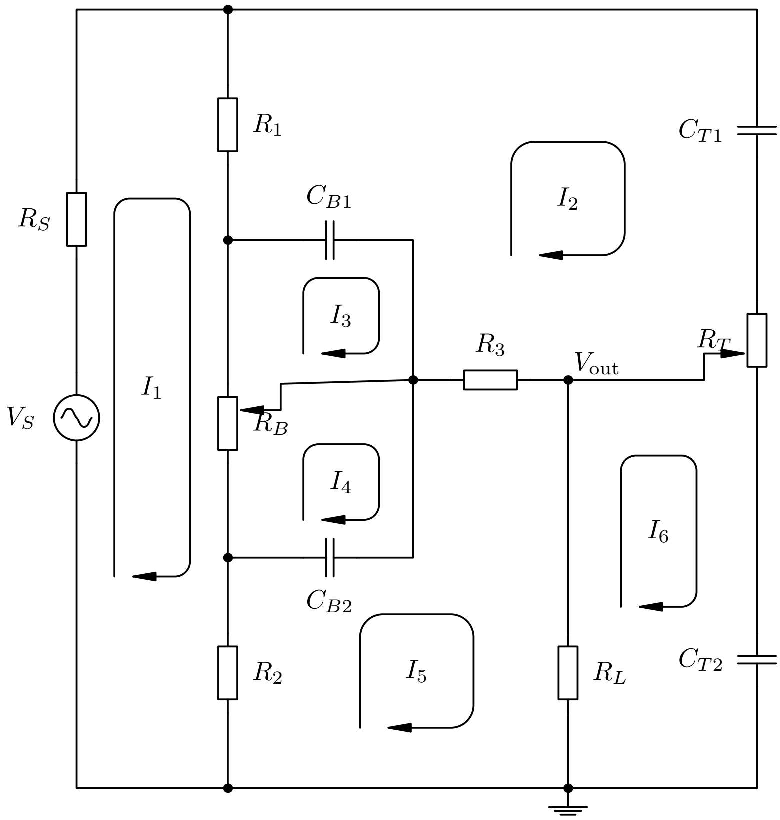

Figure 6: James tone control circuit

identified, the resulting matrix equation will be of size 6x6. The output current in this case will be the difference $I_5 - I_6$.

The circuit matrix equation is too large to be written here so the matrix is only shown in the `Maxima` batch file, which is written as follows:

```
stardisp: true$
ratmx: true$
/* matrix obtained from current loops in the circuit */
matrix([RIN+R1+R2+RB1+RB2,-R1,-RB1,-RB2,-R2,0],
       [-R1,R1+R3+RT1+1/(s*CB1)+1/(s*CT1),-1/(s*CB1),0,-R3,0],
       [-RB1,-1/(s*CB1),RB1+1/(s*CB1),0,0,0],
       [-RB2,0,0,RB2+1/(s*CB2),-1/(s*CB2),0],
       [-R2,-R3,0,-1/(s*CB2),RL+R3+R2+1/(s*CB2),-RL],
       [0,0,0,0,-RL,RL+RT2+1/(s*CT2)])$
determinant(%)$
DEN_fs: factor(%)$
DEN_den: denom(DEN_fs)$
pld: num(DEN_fs)$
/* extract numerator coefficients in the powers of s to designated variables */
DEN_XRe: coeff(pld,s,4);
```

```
DEN_AIm: coeff(pld,s,3);
DEN_BRe: coeff(pld,s,2);
DEN_CIm: coeff(pld,s,1);
DEN_DRe: coeff(pld,s,0);

/* matrix to solve Iout using Cramer's rule */
matrix([RIN+R1+R2+RB1+RB2,-R1,-RB1,-RB2,1,0],
       [-R1,R1+R3+RT1+1/(s*CB1)+1/(s*CT1),-1/(s*CB1),0,0,0],
       [-RB1,-1/(s*CB1),RB1+1/(s*CB1),0,0,0],
       [-RB2,0,0,RB2+1/(s*CB2),0,0],
       [-R2,-R3,0,-1/(s*CB2),0,-RL],
       [0,0,0,0,0,RL+RT2+1/(s*CT2)])$
IIIII: determinant(%)$
matrix([RIN+R1+R2+RB1+RB2,-R1,-RB1,-RB2,-R2,1],
       [-R1,R1+R3+RT1+1/(s*CB1)+1/(s*CT1),-1/(s*CB1),0,-R3,0],
       [-RB1,-1/(s*CB1),RB1+1/(s*CB1),0,0,0],
       [-RB2,0,0,RB2+1/(s*CB2),-1/(s*CB2),0],
       [-R2,-R3,0,-1/(s*CB2),RL+R3+R2+1/(s*CB2),0],
       [0,0,0,0,-RL,0])$
IIIIII: determinant(%)$
IIIII-IIIIII$
NUM_fs: factor(%)$
NUM_den: denom(NUM_fs)$
pln: expand(num(NUM_fs)*DEN_den/NUM_den)$
/* extract numerator coefficients in the powers of s to designated variables */
NUM_XRe: coeff(pln,s,4);
NUM_AIm: coeff(pln,s,3);
NUM_BRe: coeff(pln,s,2);
NUM_CIm: coeff(pln,s,1);
NUM_DRe: coeff(pln,s,0);
DEN_den;
NUM_den;
DEN_den/NUM_den;
```

The general format of the James tone control circuit transfer function will be

$$\frac{A_n s^4 + B_n s^3 + C_n s^2 + D_n s + E_n}{A_d s^4 + B_d s^3 + C_d s^2 + D_d s + E_d}, \tag{6}$$

where the capitalized coefficients will be obtained from the `Maxima` symbolic evaluation. The result will be too long to be written here.

5 Conclusions

All in all, in many cases `Maxima` will be a good aid when circuit equations with lots of symbolic variables are to be solved in symbolic form.

The most suitable applications for `Maxima` appear to be passive filter circuits, where the coefficients of the Laplace variable s can be listed individually for each power. From the individual coefficients it is relatively simple to construct the complete transfer function.

When using `Maxima` for transistor circuit analysis, is seems to save a lot of work in the small-signal transfer functions, but often the results are not optimally simplified.

The least favourable task for `Maxima` is the transistor circuit DC analysis, where almost the same work needs to be done in paper regardless of using `Maxima` or not.