Literature Review

SHI Yuchen

2018/11/23

Literatures

[1] Alexander Dreyer, “Combination of Symbolic and Interval-Numeric Methods for Analysis of Analog Circuits,”

[2] Guoyong Shi, “Topological Approach to Symbolic Pole–Zero Extraction Incorporating Design Knowledge,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*

Circuits are representable using a system of equations.

$$\vec{f}(\vec{x}) = \vec{0}$$

Difficulties in symbolic simulation

- No way to work out closed-form solution if too many non-linear equations
- Solution becomes *very* complex even if all, but many, linear equations

Numeric simulation is the only way out.

Numeric simulation is not perfect either.

- Floating-point operations lead to error accumulation $\rightarrow$ use rational operations
- Model does not perfectly symbolize real-world transistors $\rightarrow$?

Take model imperfection into consideration in advance using interval-arithmetic-based simulation

Define object *interval* $[\boldsymbol{x}] := [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ as the interval for variable $\boldsymbol{x}$.

Introduce the interval arithmetic.

Define basic operations $\circ \in \{+, -, \times, /\}$ for intervals $[x], [y]$:

$$[x] \circ [y] = [\underline{z}, \overline{z}] = [z]$$

where

$$\begin{cases} \underline{z} = \min\left\{\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y}\right\} \\ \overline{z} = \max\left\{\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y}\right\} \end{cases}$$

If function $f(x, y)$ is independently monotonous for x, y, then the rules above applies too for $f([x], [y])$.

Problem alert

What should $[x] - [x]$ be?

- From intuition: $[0, 0]$
- By definition: $[\underline{x}^2, \overline{x}^2]$

Namely, *the dependence problem.*

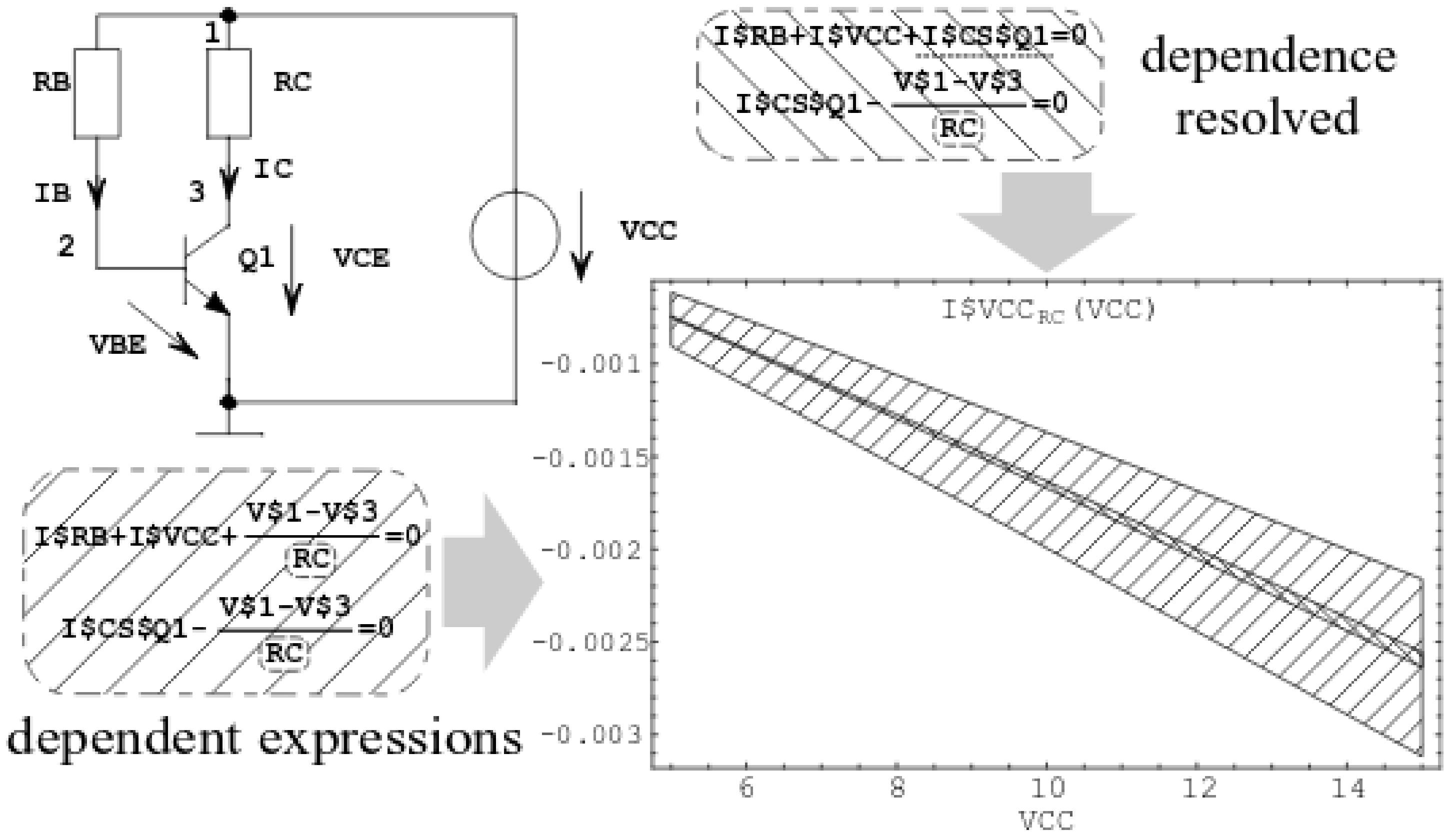


Figure 1: Dependence problem exemplified, in a non-linear circuit

Dependence problem

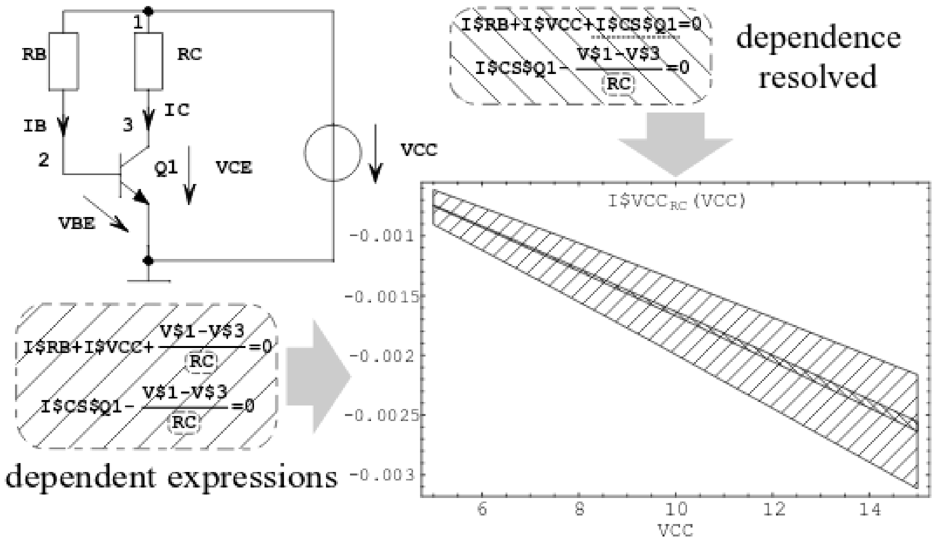
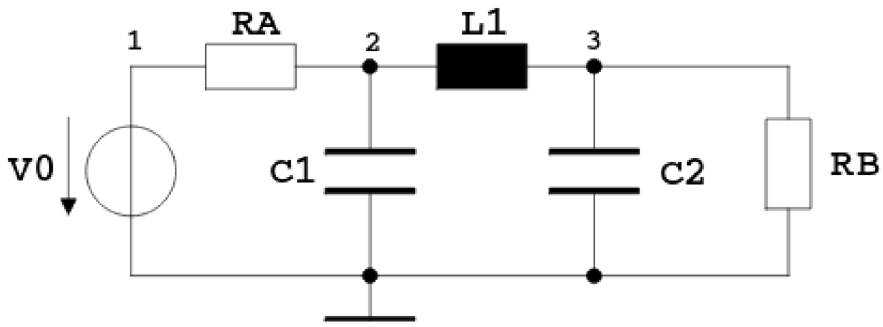


Figure 2: Dependence problem exemplified, in a non-linear circuit

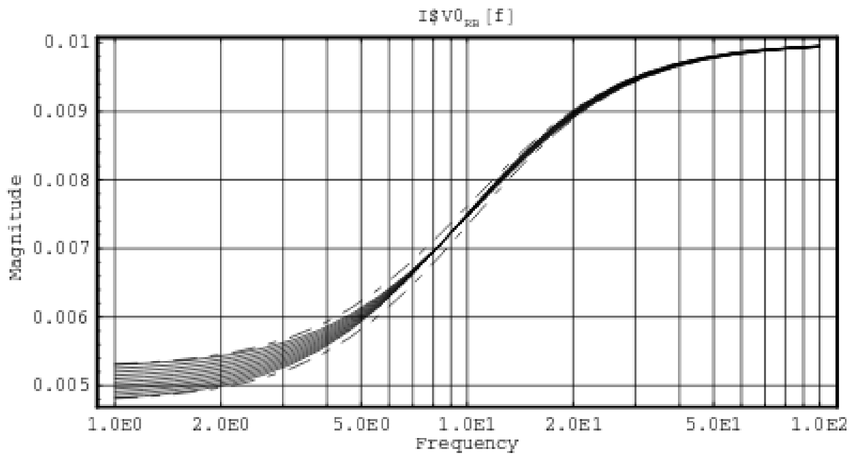
If I am to symbolize this circuit...(in the hope of not letting *any* variable occur twice)

$$\begin{cases} \dfrac{V_{\mathrm{C}} - V_{\mathrm{DD}}}{R_{\mathrm{C}}} + \underbrace{I_{\mathrm{CE}}(V_{\mathrm{C}}, V_{\mathrm{B}})}_{\text{non-linear}} = 0 \\ \dfrac{V_{\mathrm{B}} - V_{\mathrm{DD}}}{R_{\mathrm{B}}} + \underbrace{I_{\mathrm{BE}}(V_{\mathrm{C}}, V_{\mathrm{B}})}_{\text{non-linear}} = 0 \end{cases}$$

I avoided R_{C}, but could not avoid V_{DD}.



(a) RLC-filter schematics



(b) Absolute values of output current: interval AC analysis (dashed lines) and parameter sweeps (solid lines)

Figure 3: Interval arithmetic exemplified, in a linear circuit

We would like to know how the current flowing past V_0 will react to R_B within an interval $[R_\mathrm{B}]$.

That is, with different R_B, what interval will $\frac{V_0 - V_2}{R_\mathrm{A}}$ be inside?

[1] Example: linear

$$\begin{cases} \dfrac{V_2 - V_0}{R_\mathrm{A}} + V_2 sC_1 + \dfrac{V_2 - V_3}{sL_1} = 0 \\ \dfrac{V_3 - V_2}{sL_1} + V_3 sC_2 + \dfrac{V_3}{R_\mathrm{B}} = 0 \end{cases}$$

$$\downarrow$$

$$\underbrace{\begin{pmatrix} \frac{1}{R_\mathrm{A}} + sC_1 + \frac{1}{sL_1} & -\frac{1}{sL_1} \\ -\frac{1}{sL_1} & \frac{1}{sL_1} + sC_2 + \frac{1}{R_\mathrm{B}} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} V_2 \\ V_3 \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} \frac{V_0}{R_\mathrm{A}} \\ 0 \end{pmatrix}}_{B}$$

$$\downarrow$$

$$\begin{aligned} |V_2(j\omega)| \in [&\min\{A(\underline{R_\mathrm{B}})^{-1}B)_1(j\omega), (A(\overline{R_\mathrm{B}})^{-1}B)_1(j\omega)\}, \\ &\max\{A(\underline{R_\mathrm{B}})^{-1}B)_1(j\omega), (A(\overline{R_\mathrm{B}})^{-1}B)_1(j\omega)\}] \end{aligned}$$

What is this??

$$|V_2(j\omega)| \in [\min\{A(\underline{R_\mathrm{B}})^{-1}B)_1(j\omega), (A(\overline{R_\mathrm{B}})^{-1}B)_1(j\omega)\},$$
$$\max\{A(\underline{R_\mathrm{B}})^{-1}B)_1(j\omega), (A(\overline{R_\mathrm{B}})^{-1}B)_1(j\omega)\}]$$

It means,

1. For every Hz, you take $\underline{R_\mathrm{B}}$ and $\overline{R_\mathrm{B}}$ into A
2. Work out X by $A^{-1}B$, but since you have $\underline{R_\mathrm{B}}$ and $\overline{R_\mathrm{B}}$, you will get 2 copies of X
3. Work out your target function I with 2 copies of X, and get $\underline{I}, \overline{I}$

$[\underline{I}, \overline{I}]$ will be I's interval if R_B is in $[\underline{R_\mathrm{B}}, \overline{R_\mathrm{B}}]$

Time complexity $O(2^n)$ alert

If you have n independent variables that have their own intervals, you have to

1. Solve $AX = B$ by applying $\mathbf{2}^n$ times, as for each variable you try twice
2. As you solve, find the minimum and the maximum

...and remember it's just linear. Non-linear solving takes much more time.

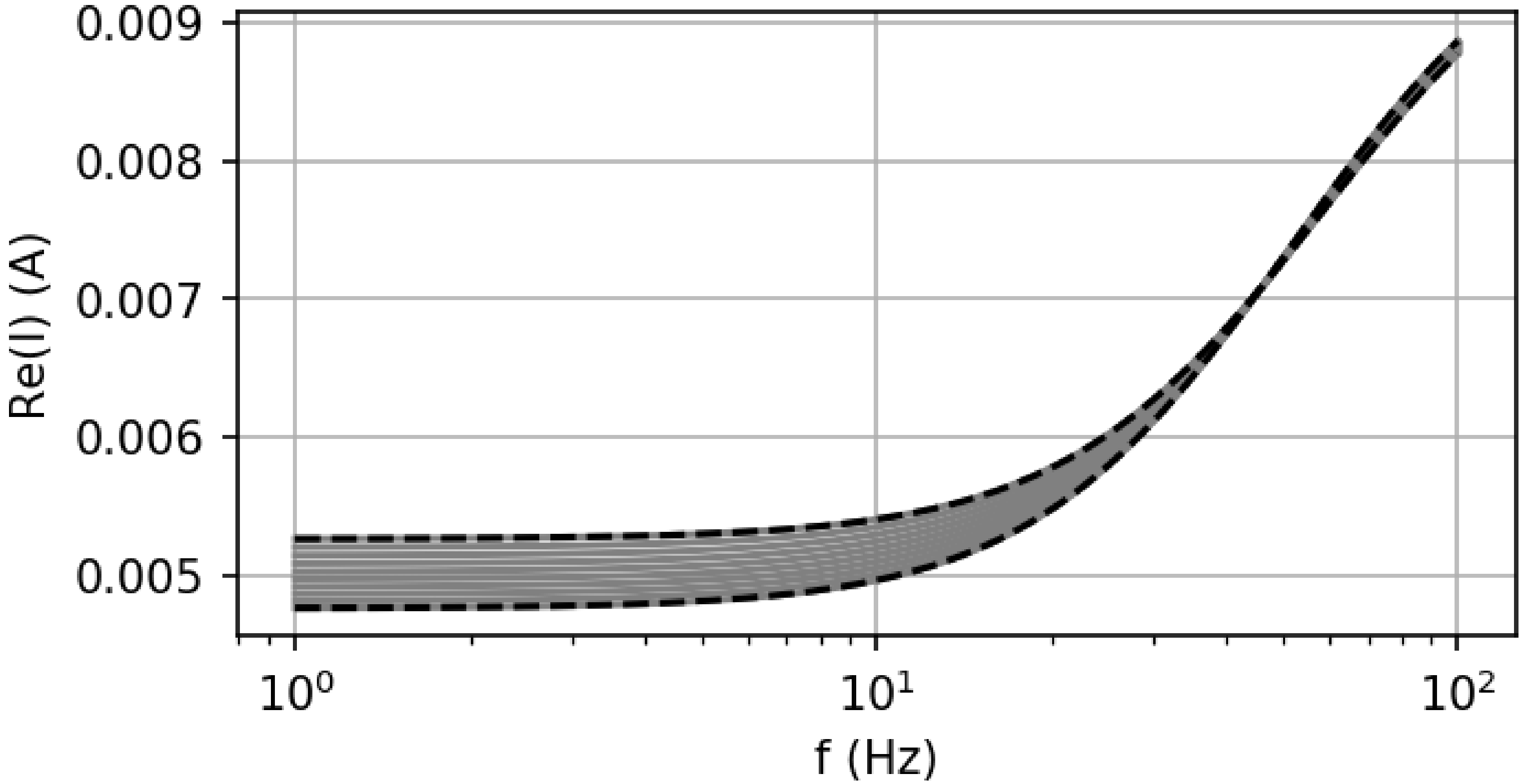


Figure 4: Current frequency response, with $R_{\mathrm{B}} \in [\mathbf{0.9}\boldsymbol{e}\mathbf{3}, \mathbf{1.1}\boldsymbol{e}\mathbf{3}]$

Problems

- $O(2^n)$ time complexity
- I did not get his result

Inspiration

- Random variable

  possibility density for each instance inside interval

- Monte-Carlo method

Automatic extraction of *human-readable* poles and zeros in symbolic form

Why?

- Why automatic? Opamps are getting very complicated.
- Why symbolic? Numeric simulation provides no guidance in *design* stage (but in verification stage)
- Why human-readable? Design needs interpretable information.

How do we reduce a transfer function $H(s)$ until it becomes human-readable?

1. Write $H(s)$ in the form of

$$H(s) = \frac{\sum_{k=0}^{m} b_k s^k}{\sum_{k=0}^{n} a_k s^k}$$

and every b_k, a_k should be a sum of several multiplication, e.g.,

$$b_0 = \underbrace{R_2 R_3 R_1 g_{\mathrm{m1}} g_{\mathrm{m2}} g_{\mathrm{m3}}}_{\text{multiplication term 1}} + \underbrace{R_3 R_1 g_{\mathrm{m1}} g_{\mathrm{mf}}}_{\text{multiplication term 2}}$$

2. Apply rules for every b_k, a_k
   1. If any multiplication term does not contain compensation resistance R_c, keep those multiplications with the **most** occurrence of g_m.

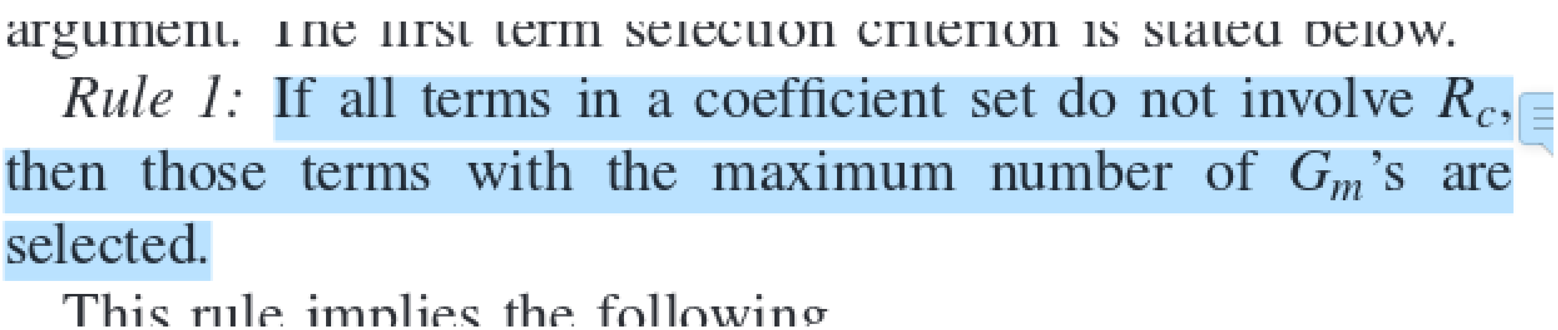
   e.g., before,

$$b_0 = \underbrace{R_2 R_3 R_1 g_{\mathrm{m}1} g_{\mathrm{m}2} g_{\mathrm{m}3}}_{3g_\mathrm{m}:)} + \underbrace{R_3 R_1 g_{\mathrm{m}1} g_{\mathrm{mf}}}_{2g_\mathrm{m}:(}$$

   after,

$$b_0 = R_2 R_3 R_1 g_{\mathrm{m}1} g_{\mathrm{m}2} g_{\mathrm{m}3}$$

[2] Grammatical error



argument. The first term selection criterion is stated below.

Rule 1: If all terms in a coefficient set do not involve R_c, then those terms with the maximum number of G_m's are selected.

This rule implies the following

Figure 5: Ambiguity in paper[2]

This sentence can mean

- If **no** terms in a coefficient set involves R_c ...
- If **not all** term in a coefficient set involves R_c ...

I deduce that the 1st is right, because the 2nd contradicts with rule 2.

...rules continued. For simplicity, we define each multiplication term as $b_{k,i}, a_{k,i}$.

2. Count the occurrence of g_{m} in each term $b_{k,i}, a_{k,i}$ that does not contain R_{c}, and get the maximum of them $\#g_{\mathrm{m,max}}$. For each term $b_{k,i}, a_{k,i}$, keep them only if they have the occurrence of g_{m} bigger than the sum of the occurrence of R_{c} and $\#g_{\mathrm{m,max}}$, i.e., keep terms where
$$\#g_{\mathrm{m}} \geq \#g_{\mathrm{m,max}} + \#R_{\mathrm{c}}$$
is satisfied.

...e.g.,

$$
\begin{aligned}
b_1 = {} & R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{mf}} C_{\mathrm{c2}} + R_{\mathrm{c}} R_1 R_3 g_{\mathrm{m1}} g_{\mathrm{f}} C_{\mathrm{c2}} \\
& + R_{\mathrm{c}} R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} g_{\mathrm{m3}} C_{\mathrm{c2}} \\
& + R_{\mathrm{c}} R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} g_{\mathrm{m3}} C_{\mathrm{c1}} \\
& + R_{\mathrm{c}} R_1 R_3 g_{\mathrm{m1}} g_{\mathrm{mf}} C_{\mathrm{c1}} - R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} C_{\mathrm{c2}} \\
& - R_1 R_3 g_{\mathrm{m1}} C_{c1}
\end{aligned}
$$

rule 1 does not apply, because it contains terms with R_{c}.

With rule 2, $\#g_{\mathrm{m,max}} = 2$,

$$
\begin{aligned}
b_1 = {} & R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{mf}} C_{\mathrm{c2}} && \rightarrow \#g_m = 2, \#R_c = 0 \\
& + R_{\mathrm{c}} R_1 R_3 g_{\mathrm{m1}} g_{\mathrm{mf}} C_{\mathrm{c2}} && \rightarrow \#g_m = 2, \#R_c = 1 \\
& + R_{\mathrm{c}} R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} g_{\mathrm{m3}} C_{\mathrm{c2}} && \rightarrow \#g_m = 3, \#R_c = 1 \\
& + R_{\mathrm{c}} R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} g_{\mathrm{m3}} C_{\mathrm{c1}} && \rightarrow \#g_m = 3, \#R_c = 1 \\
& + R_{\mathrm{c}} R_1 R_3 g_{\mathrm{m1}} g_{\mathrm{mf}} C_{\mathrm{c1}} && \rightarrow \#g_m = 2, \#R_c = 1 \\
& - R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} C_{\mathrm{c2}} && \rightarrow \#g_m = 2, \#R_c = 0 \\
& - R_1 R_3 g_{\mathrm{m1}} C_{c1} && \rightarrow \#g_m = 1, \#R_c = 0
\end{aligned}
$$

so

$$
\begin{aligned}
b_1 = {} & R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{mf}} C_{\mathrm{c2}} + R_{\mathrm{c}} R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} g_{\mathrm{m3}} C_{\mathrm{c2}} \\
& + R_{\mathrm{c}} R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} g_{\mathrm{m3}} C_{\mathrm{c1}} - R_1 R_2 R_3 g_{\mathrm{m1}} g_{\mathrm{m2}} C_{\mathrm{c2}}
\end{aligned}
$$

...rules continued.

3. If all terms contain $\boldsymbol{R}_\mathbf{c}$ and for all terms, $\#\boldsymbol{g}_\mathrm{m} < \#\boldsymbol{R}_\mathbf{c}$, then keep all terms.

[2]

Conclusion

- Very explicit rules that are suitable for programming implementation. I like it
- Over-exaggerated, misleading title

[3] Frederico A.E. Rocha, et al., “Electronic Design Automation of Analog ICs Combining Gradient Models with Multi-Objective Evolutionary Algorithms,”

[4] Rodney Phelps, et al., “Anaconda: Simulation-Based Synthesis of Analog Circuits Stochastic Pattern Search,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*

[5] Bo Liu, et al., “Analog circuit optimization system based on hybrid evolutionary algorithms,” *INTEGRATION, the VLSI journal*

- Write a program that implements the rules described in [2]