

符号网络分析算法及计算机实现

张 蓉* 兰家隆

(电子科技大学信息工程系 成都 610054)

【摘要】 提出了一种分析电路符号网络函数的新编码算法 NSCA 它能有效降低基码,防止过早发生整数溢出,从而扩大分析电路的规模。提出了灵敏度分析算法和化简算法,并可对模块电路进行处理,增强了编码算法的适用范围。根据该算法开发了软件系统 SN SPC2.0 实例运行结果证明了算法理论的正确性和可行性。

关键词 编码; 反码; 基码; 灵敏度; 模块电路; 网络函数; 符号分析; 伪码
中图分类号 TN711.1

电路的符号网络分析是指用自变量(时间或频率)、应变量(电压和电流)以及部分或全部表示为符号的电路元件来描述电路特性的一种电路分析方法。这种分析方法与数值分析相对应,是一种定性分析。符号网络函数是指有元件参数和频率等变量的网络转移函数,利用它们可以方便地实现各种特性参数的转换^[1],可以清楚地洞察出电路的内部特性,看出某个元件对电路功能实现的贡献,以及当电路功能没有得到满足时该采取何种解决办法的启示。从 60 年代起,许多学者就致力于这一研究,先后产生了矩阵运算法、信号流图法、树权举法、参数提取法、数值插入法等算法^[2]。在这些方法中,大部分过程都用于字符串的处理,有些方法还需对电路进行某种特殊变换,结果难以化简。针对这一情况,产生一种符号编码算法 SCA^[3],属于矩阵运算范畴。它将符号进行准数值化编码,形成编码单元,按照改进节点法填入矩阵,最后利用克莱姆法则求解,反码后得出结果。

对于线性、集总、时不变的电路,在复频域上得到的符号网络函数是有理函数,可以表示为

$$H(s) = \frac{N(S, P_1, \dots, P_m)}{D(S, P_1, \dots, P_m)} = \frac{\sum_i S^i a_i(P_1, \dots, P_m) d_{in}}{\sum_j S^j b_j(P_1, \dots, P_m) d_{jd}} \quad (1)$$

其中 $a_i(\dots)$ 和 $b_j(\dots)$ 是对每个 S 幂次的关于符号 $P_j (j=1, 2, \dots, m)$ 的矢量函数, d_{in} 和 d_{jd} 是系数。因此分子、分母可以看成是三部分:数值部分、符号部分以及复频率部分。在 SCA 中,正是利用这一性质将任何一个电路元件参数用三部分表示,形成了编码单元。实际上,编码单元可以进一步简化由数值部分和编码部分。复频率幂次部分可以由编码部分获得。当编码部分选为符号的电容和电感的编码时,可知对应的复频率幂次分别为 +1 和 -1。但是,当电容和电感没被选为符号时,编码部分为零,无法恢复出复频率幂次,因此增设两个伪码 C 和 L ,表示没有被选为符号的电容和电感,从而恢复出 S 幂次。这就是改进符号编码算法 MSCA,它有效地节约了存储空间。

但是, MSCA 在编码上存在不足,基码的设定并不是最低,且适用范围也不够广。本文提出的 NSCA 克服了 MSCA 和 SCA 算法的不足,采用更简洁的编码法,进一步降低了基码,从而扩大分析电路的规模,并新增灵敏度分析算法和化简算法,且能对模块电路进行处理,较大地提高了这一

① 1996 年 6 月 17 日收稿,1996 年 7 月 9 日修改定稿
* 女 24 岁 硕士生

算法的有效性.

1 编码算法原理

定义两种运算如下:

1) 乘法 两项相乘,对应的数值项 N_d 相乘,编码项 N_c 相加

2) 加法 两项相加,若对应的编码项 N_c 相同,则对应的数值项 N_d 相加,否则不进行操作。下面介绍编码算法原理并论证其正确性。

下面介绍编码算法原理并论证其正确性。

定理 1 设电路中某一符号对应的元件有 n 个,则在该电路所对应的符号网络函数的分子和分母的每一项中,该符号的幂次不超过 n 次

证明见文献 [4]

定理 2 设电路中具有相同符号的元件的最大个数为 N ,则可以定义基码 $C_b = N + 1$,以基码为底,对各符号进行编码,其编码值分别为 $C_b^0, C_b^1, C_b^2, \dots$,那么,编码矩阵按照上述定义的运算准则和行列式递归算法计算的值可以正确反码出原始信息

证明 设一电路有 n 个符号 x_1, x_2, \dots, x_n ,并且具有相同符号的元件的最大个数为 N ,则由题设 $C_b = N + 1$ 设所有符号编码值从小到大排列,依次为 $C_b^0, C_b^1, \dots, C_b^{n-1}$,其中 C_b^{m-1} 为第 m 个符号的编码值 由定理 1,第 m 个符号在分子或分母中任一项中给出的最大幂次为 N ,由运算规则可知其编码值最大为 $N C_b^{m-1}$ 。所以前 m 个符号相乘的值最大为

$$\sum_{i=1}^m (N C_b^{i-1}) = N \frac{1 - C_b^m}{1 - C_b} = (C_b - 1) \frac{1 - C_b^m}{1 - C_b} = C_b^m - 1 < C_b^m$$

而 C_b^m 为第 $m+1$ 个符号的编码值,因此前任意 m 个符号相乘的值不会超过第 $m+1$ 个符号的编码值,故不会产生混码失真

证毕

特殊地,当只有一种符号所表示的元件个数为最大个数 N 时,只要采取一定的编码规则,可以进一步降低基码

定理 3 设电路中具有相同符号的元件的最大个数为 N ,次最大个数为 N' ,且只有一种符号所表示的元件个数为 N ,则基码 C_b 可定义为

$$C_b = N' + 1$$

证明 设一电路中有 n 个符号,将这 n 个符号按其所表示的元件个数多少从大到小排列,且满足题设,即

$$n_1 > n_2 \geq n_3 \geq \dots \geq n_n$$

其中 $n_i (i = 1, 2, \dots, n)$ 表示第 i 个符号所表示的元件个数。

由题意, $C_b = n_2 + 1$ 我们将编码值越大的赋给 n_i 值越大的符号,即对应于 n_i 的编码值为

$$n_1 \rightarrow C_b^{n-1}, n_2 \rightarrow C_b^{n-2}, \dots, n_n \rightarrow C_b^0$$

1) 对于符号 n_2, n_3, \dots, n_n ,由于 $C_b = n_2 + 1$,根据定理 2 可以证明 $n-1$ 个符号编码不会产生混码失真。

2) 证明 n_2, n_3, \dots, n_n 产生的符号编码项不会与 n_1 的混淆

由定理 1 可知,第 i 个符号在分子或分母任一项中给出的最大幂次为 n_i ,因此第 i 个符号在任一项中给出的最大编码值为 $n_i C_b^{n-i}$,所以 n_2, n_3, \dots, n_n 相乘的编码值最大为

$$n_2 C_b^{n-2} + n_3 C_b^{n-3} + \dots + n_{n-1} C_b + n_n C_b^0 \leq n_2 C_b^{n-2} + n_2 C_b^{n-3} + \dots + n_2 C_b^0 =$$

$$n_2 \frac{C_b^{n-1} - 1}{C_b - 1} = (C_b - 1) \frac{C_b^{n-1} - 1}{C_b - 1} = C_b^{n-1} - 1 < C_b^{n-1}$$

而 C_b^{n-1} 为第 1 个符号 (表示的元件个数最大) 的编码值, 因此不会产生混码失真。

下面对三种算法最多可容纳的符号数作一个简单比较:

设 m 为电路中具有相同符号的元件的最大个数, n 为最多可容纳的符号数, C_b 为基码, 比较结果见表 1

表 1 三种算法最多容纳符号数比较表

SCA		MSCA			NSCA	
m	C_b	n	C_b	n	C_b	n
1	4	15	2	30	2	30
2	8	10	4	15	3	19
3	8	10	4	15	4	15
4	16	8	8	10	5	13
5	16	8	8	10	6	12
6	16	8	8	10	7	11
7	16	8	8	10	8	10
8	32	6	16	8	9	10
...

注 当只有一种符号所表示的元件个数为最大个数 N 时, NSCA 算法中的 C_b, n 取决于具有相同符号的元件的次最大个数 m' 。

从上面的分析可知, NSCA 确实可以降低基码, 增加可容纳的符号数, 防止整数过早溢出, 使这一算法适用范围更大。特别是当只有一种符号所表示的元件为最大个数 N 时, 编码还可进一步降低为具有相同符号的元件的最大个数加 1, 从而基码进一步降低。由于在一般的电路, 电抗元件中电感出现次数较少, 电容出现概率较大, 由算法原理可知伪码 C 的次数是比较高的, 而且一般是最高, 其他的符号的次数都远远小于它, 因而定理 3 在这种情况下更具有优势。

2 灵敏度算法

由灵敏度定义

$$S_x^T = \frac{\partial T}{\partial x} \frac{x}{T} \quad (2)$$

设 $T = N/D$, 则

$$S_x^T = \frac{\partial(N/D)}{\partial x} \frac{x}{T} = \frac{x D \frac{\partial N}{\partial x} - x N \frac{\partial D}{\partial x}}{DN} \quad (3)$$

由于 N 和 D 中每一项均可写为 $A_i X^i$, 其中, x 为灵敏度运算所针对的元件参数, i 为元件参数 x 的幂次, A_i 为该项中不含有 x 的乘积部分, 但有可能含有其他元件参数或 S 幂次以及系数的乘积

$$\frac{\partial(A_i X^i)}{\partial x} = i A_i x^{i-1} \quad i \neq 0 \quad (4)$$

由此可见,只要在计算时,将含有 x 项的码值部分减去 x 的编码值,数值部分乘以原该项表达式中 x 的幂次,即完成了求导。

同时,由以上分析可知,由于存在两个编码多项式的乘积,所以若原来符号的幂次最大为 n 次 ($Th 1$),则现在可能达到 $2n$ 次,因此应相应增大基码,以保证结果不失真。由于这种情况相当于该符号对应的元件有 $2n$ 个,因此若原来基码为 O ,则现在应设为 $2O - 1$ 。

3 化简算法

定理 4 在加减法中,结果的绝对误差界由操作数的绝对误差界的和给出。

定理 5 在乘除法中,结果的相对误差界由操作数的相对误差界的和给出。

证明见文献 [5]。

设要求化简结果的最大误差为 $\max \epsilon\%$,则根据定理 5,可设分子、分母的化简结果的最大误差分别为 $(\max \epsilon / 2)\%$ 。设分子或分母多项式的实际结果(取模)为 M ,则其绝对误差为 $M(\max \epsilon\%) / 2$ 。又根据定理 4,设多项式有 n 项,则每一项的绝对误差平均为 $M(\max \epsilon\%) / (2^n - n)$,因此凡是绝对误差小于这个平均值的项可去掉。由于定理 4 给出的是对实际误差的粗糙的超估,所以还可进一步消掉某些项,使误差更接近于最大误差,可分两步进行化简:

1) 将符号化器件的原始值代入多项式,将每个和项的模与多项式的模比,若小于平均最大误差,则置标志位为 0,否则为 1。

2) 将标志位为 1 的项选出,并按模的大小进行排序,从模小的项开始,若去掉一项后没有超过最大误差限,则设标志位为 0,否则设为 1 后,结束化简过程。

4 模块电路处理

将完成一定功能的模块电路当成是一种特殊的器件来使用,而不用重复在电路中输入这些模块电路内部细节,具有很大的现实意义。建立模块电路的规则如下:

在主电路中,加入模块电路的格式为

$S^* \text{ Model-name } n_1 n_2 n_3 \dots$

其中 S 表示该器件是模块电路; $*$ 为序号。

Model-name 模块电路名。

n_1, n_2, n_3, \dots : 与模块电路接口节点。

由于该器件的特殊性,故定义一新的结构单元来存贮这种器件。

```
typedef struct mo {
    char* name;
    char* model_name;
    int* nade;
    struct mo* next;
}MODEL;
```

模块电路采用文件形式存放,为使用户一目了然,文件名即取为模块电路名。文件中第一行非注释语句应列出输出节点,相当于集成电路中的管脚。然后列出模块电路中所含元器件及其连接关系,格式同 SPICE 输入格式。

5 SN SPC 2.0 软件系统简介及举例

5.1 SN SPC 2.0 软件系统简介

本软件系统是在 SN SPC 1.0 基础上开发,除具有原来的优点,如友好的图形界面,还具有以下特点: 1) 采用 NSCA 算法,可输出五种网络函数 (V_0/V_i , I_0/I_i , P_0/P_i , V_0/I_i , V_i/I_i),具有更快的运算速度,更节约存贮空间; 2) 可以处理并联元件; 3) 对设有输入值的元件也可进行处理; 4) 增加了模块电路的处理功能; 5) 可进行灵敏度分析和更灵活的化简运算,并可输出图形,对于结果的分析十分直观。本软件系统是在 AST486, DOS5.0 环境下开发,可以处理 31 个符号同时 45 个节点的功能电路(当没有元件采用同一种符号表示时),同时允许一个符号代表多个元件,在这种情况下,可同时分析的符号个数参见表 1。

5.2 举例

下面的例子均在 AST486 微机上进行,工作频率为 33 MHz,内存为 8 M。

例 1 电路如图 1 所示,为 n 阶梯形电阻网络,

取 R_{2n-4} , R_{2n-6} , R_{2n-8} , R_{2n-10} , R_{2n-12} 为符号,其余电阻值为 1Ω 。表 2 列出了 SN SPC 1.0 和 SN SPC 2.0 两个版本和采用经典法开发的软件 SNAP 的运行时间比较 (SNAP 运行时间在 CDC6500 上测得)。

表 2 三种软件运行时间比较

节点数	SNAP	SN SPC 1.0	SN SPC 2.0
11	3.27 s	1.14 s	1.06 s
13	19.8 s	1.27 s	1.15 s
15	140 s	2.64 s	2.47 s
17	16 min	6.24 s	5.93 s

例 2 电路如图 2 所示,给出它的电压传输函数及对元件 g_3 的灵敏度分析及小于 20% 误差内化简表达式,其输出结果如下。

$$V_0/V_i = \frac{-g_{1S_2}^* g_{2S_1}^* g_4}{c_{2S_1}^* g_{3S_2}^* (g_1 + c_{1S}^* s)^* s + g_{1S_2}^* g_{2S_1}^* g_3}$$

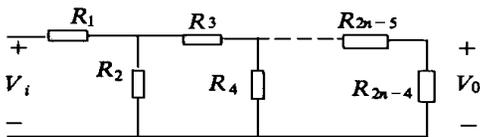


图 1 n 阶梯形网络

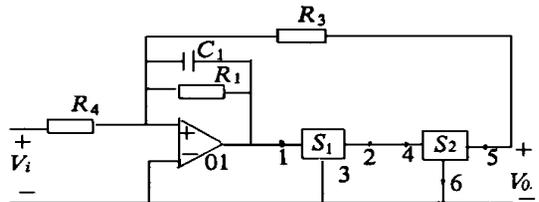


图 2 带模块电路的主电路

6 结束语

本文讨论的 NSCA 算法适用于用任何方法建立的电路方程,并具有快捷、有效,同时节省空间的特点,提高了分析电路的规模,在此基础上开发的 SN SPC 2.0 软件包在实际运行中的效果令人满意。能否将编码单元进一步简化成只用一个部分组成,并且采用有效的电路分解技术以大幅度增大分析电路规模,是今后研究符号编码分析的一个方向。

参 考 文 献

- 1 Gielen G E, Herman C C, Walscharts Willy. ISAAG a symbolic simulator for analog integrated circuits. IEEE Solid-state Circuits, 1989, 24(6): 1 587~ 1 597
- 2 Gielen Georges, Wambacq Piet, Sansen Willy M. Symbolic analysis methods and applications for analog circuits a tutorial overview. Proc of IEEE, 1994, 82(2): 287~ 303
- 3 蒲立新, 兰家隆. 计算机实现符号网络的新算法. 电子科技大学学报, 1994, 23, (2): 132~ 137
- 4 Lin P M. A survey of applications of symbolic network functions. IEEE Trans Circuit Theory, 1973, CT-16: 509~ 517
- 5 王尊正. 数值分析教程. 哈尔滨: 哈尔滨工业大学出版社, 1993

New Algorithm of Generating Symbolic Network Functions by Computer

Zhang Rong Lan Jialong

(Dept. of Information Engineering, UEST of China Chengdu 610054)

Abstract A new symbolic coded algorithm NSCA of generating symbolic network functions is presents. In NSCA, the base code of coded cell prescribed each component is much lower, so it prevents early overflow of integer. The scale of the circuit analyzed then can be larger. It also provides sensitivity algorithm and simplify algorithm, and processes a circuit with definite function as a model circuit, which enlarge the application. A computer program SNSPC2. 0 (symbolic network simulation by parameter coded) which is based on NSCA is developed, and its effectiveness is illustrated with examples.

Key words code; decode; base code; sensitivity; model circuit; network functions; symbolic analysis; pseudo code

编辑 徐培红