

Title	Tolerance Analysis of Linear Resistive Network by interval Mathematics
Author(s)	OKUMURA, Kohsi; SAKANASHI, Koh
Citation	数理解析研究所講究録 (1993), 831: 48-52
Issue Date	1993-04
URL	http://hdl.handle.net/2433/83374
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

Tolerance Analysis of Linear Resistive Network by Interval Mathematics

Kohshi OKUMURA and Koh SAKANASHI
奥村 浩 士 坂 梨 興

Department of Electrical Engineering II
Kyoto University, Kyoto

1. Introduction

In reference [2], the application of the interval Gaussian algorithm to the tolerance analysis, particularly, the worst case analysis of the linear resistive network has been presented. The interval hybrid equation is proposed to formulate the network equation. Hansen's method is applied to the interval hybrid equation to perform the interval Gaussian algorithm. In order to have a good estimation of the interval solutions, we proposed to use the maximally distant tree pair for the formulation of the network equation and take the intersection of the solutions of the hybrid equation for each tree. The maximally distant tree pair is introduced under assumption that the tree pair covers all the resistive branches.

In this report, we try to deal with the network of a little larger scale where the maximally distant tree pair can not cover the network. The interval Gaussian algorithm is compared with the iteration method. Finally, in order to have a better estimated solution, we describe a method for solving the interval hybrid equations with the subdivided interval parameters and taking the union of the interval solutions.

2. Interval Hybrid Equation

The network is assumed to have a single resistive element between nodes. We suppose that every branch of the network has connected across it an ideal current source and inserted into it an ideal voltage source. The hybrid equation of the linear resistive time-invariant network becomes

$$\mathbf{H}\mathbf{x} = \mathbf{a} \quad (1)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{G} & \mathbf{Q}_t \\ \mathbf{B}_t & \mathbf{R} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \mathbf{V}_t \\ \mathbf{I}_l \end{bmatrix}, \mathbf{a} = \begin{bmatrix} \mathbf{J}_s \\ \mathbf{E}_s \end{bmatrix}, \mathbf{J}_s = -\mathbf{j}_{st} + \mathbf{G}\mathbf{e}_{st}, \mathbf{E}_s = -\mathbf{e}_{st} + \mathbf{R}\mathbf{j}_{st} \quad (2)$$

The matrix \mathbf{G} is the diagonal branch conductance matrix of dimension ρ and \mathbf{R} is the diagonal branch resistance matrix of dimension μ , where ρ and μ are the rank and nullity of the simple graph associated with the network with the voltage sources short and current sources open. The submatrices \mathbf{Q}_t and \mathbf{B}_t are the principal parts of the cutset and tieset matrices, respectively. The vectors \mathbf{V}_t and \mathbf{I}_l are the tree branch voltage and link current vectors, respectively. The vectors \mathbf{J}_s and \mathbf{E}_s are the cutset current source and loop voltage source vectors, respectively. The vectors \mathbf{j}_t and \mathbf{j}_l are the current source vectors associated with the tree and the link, respectively. The vectors \mathbf{e}_t and \mathbf{e}_l are the voltage source vectors associated with the tree and the link, respectively.

3. Selection of Trees

We pick up a tree T_1 which contains as many conductors as possible and select the maximally distant tree T_2 from T_1 . At this stage, we check whether the pair of the trees T_1, T_2 covers all the resistive branches of the network. If not, we choose another tree T_3 and check whether the tree set T_1, T_2, T_3 all the resistive branches. We repeat this procedure until the set of the trees T_1, T_2, \dots, T_m covers all the tree branches where the integer m is the minimum number of the trees.

For the tree T_i ($i = 1, 2, \dots, m$) we formulate the hybrid equation

$$\mathbf{H}_i \mathbf{x}_i = \mathbf{a}_i \quad i = 1, 2, \dots, m. \quad (3)$$

We assume that the matrix \mathbf{H}_i ($i = 1, 2, \dots, m$) is nonsingular. Using Hansen's method, we transform Eq.(3) into

$$\tilde{\mathbf{H}}_i \mathbf{x}_i = \tilde{\mathbf{a}}_i \quad i = 1, 2, \dots, m \quad (4)$$

where

$$\tilde{\mathbf{H}}_i = m(\mathbf{H}_i)^{-1} \mathbf{H}_i, \quad \tilde{\mathbf{a}}_i = m(\mathbf{H}_i)^{-1} \mathbf{a}_i \quad i = 1, 2, \dots, m. \quad (5)$$

The interval Gaussian algorithm or the iteration method is applied to Eq.(5) for each i to have the interval branch voltage solution $\mathbf{V}^{(i)}$ ($i = 1, 2, \dots, m$) and the branch current solution $\mathbf{I}^{(i)}$ ($i = 1, 2, \dots, m$). We represent the true branch voltage and current solutions as \mathbf{V}_{true} and \mathbf{I}_{true} , respectively. Hence the relations

$$\mathbf{V}_{true} \subseteq \mathbf{V}^{(i)}, \quad \mathbf{I}_{true} \subseteq \mathbf{I}^{(i)} \quad i = 1, 2, \dots, m \quad (6)$$

Clearly, we have

$$\mathbf{V}_{true} \subseteq (\cap_{i=1}^m \mathbf{V}^{(i)}) \subseteq \mathbf{V}^{(i)} \quad i = 1, 2, \dots, m \quad (7)$$

$$\mathbf{I}_{true} \subseteq (\cap_{i=1}^m \mathbf{I}^{(i)}) \subseteq \mathbf{I}^{(i)} \quad i = 1, 2, \dots, m \quad (8)$$

Hence we take $\mathbf{V} = \cap_{i=1}^m \mathbf{V}^{(i)}$ and $\mathbf{I} = \cap_{i=1}^m \mathbf{I}^{(i)}$ as the nearest interval branch voltage and current solutions to the true solutions.

4. Partitioning Interval Parameters

If the tolerance of the resistive parameter is larger, partitioning the interval parameters is effective to have a good estimation of the interval solutions. Let the network have $b = \rho + \mu$ resistive parameters. Partitioning i -th resistive parameters into b_i ($i = 1, 2, \dots, b$) subintervals provides us with $\prod_{i=1}^b b_i$ hybrid equations for each tree. Associated with the tree T_i , let $\mathbf{V}_k^{(i)}$ and $\mathbf{I}_k^{(i)}$ be the interval branch voltage and current solutions of the hybrid equation for the k -th combination of each partitioned parameter. For the tree T_i , we have

$$\mathbf{V}^{(i)} = \cup_{k=1}^K \mathbf{V}_k^{(i)}, \quad \mathbf{I}^{(i)} = \cup_{k=1}^K \mathbf{I}_k^{(i)}, \quad K = \prod_{i=1}^b b_i \quad (9)$$

Solving K hybrid equations for the tree T_i , we have the interval solutions which are of the form

$$\mathbf{V} = \cap_{i=1}^m \mathbf{V}^{(i)} = \cap_{i=1}^m (\cup_{k=1}^K \mathbf{V}_k^{(i)}) \quad (10)$$

$$\mathbf{I} = \cap_{i=1}^m \mathbf{I}^{(i)} = \cap_{i=1}^m (\cup_{k=1}^K \mathbf{I}_k^{(i)}). \quad (11)$$

If b interval parameters take part in partitioning, Km combinations of the intervals involves the solution of the hybrid equations. For example, by each halving of the interval parameters, we have to solve $2^b m$ hybrid equations.

5. Numerical Examples

This section demonstrates the worst-case analysis of a linear resistive network with rather large number of branches. Hybrid equation (3) is solved by the interval Gaussian algorithm(abbreviated as IGA) as well as symmetric single step Method(abbreviated as SS method)[1]. The latter is one of the iteration methods in which the hybrid matrix \tilde{H} is decomposed into a diagonal, a strictly lower and a strictly upper triangular matrix. Both methods are compared. Further, the validity of partitioning of the interval parameters is shown.

5.1 Comparison of IGA with SS Method

Fig.1 shows a linear resistive network the branch voltages of which are analysed. Each resistor has its own tolerance. The resistors and voltage sources assign respectively the center values g_i ($i = 1, 2, \dots, 26$) as conductances and numerical values E_i ($i = 1, 2$) in table 1. We give 10 percents tolerance of the center values for each resistor. Table 2 shows the branch numbers of the trees chosen. The distance between T_1 and T_2 is maximal. The set of the trees T_1, T_2, T_3 covers all the resistive branches.

Table 3 shows the branch voltages computed by IGA and SS method as well as Monte Carlo method(abbreviated as MC method). The name of each column indicates the method as its name shows. The interval numbers in MONTE CARLO are computed by MC method. The numbers in GAUSS and SS are the ratios of the widths of the voltages by IGA and SS method to the widths by MC method. The branch voltages are well estimated except $V_9, V_{12}, V_{13}, V_{19}$ and V_{24} . The four times of iterations are enough to converge the procedures of SS method. The starting interval are the point interval numbers which are determined by the solutions of the network equation without the tolerances of resistances.

5.2 Validity of Partitioning

We deal with the ladder network which was used as an example in Ref.[2]. The resistive parameters are the same as given in Ref.[2]. As a typical case, the interval of each resistive parameter is halved($K = 2^9$). The maximally distant tree pair covers all the resistive branches($m = 2$). Table 4 shows the results computed by IGA and SS method as well as MC method. The numbers in each column have the same meaning as in Table 3. Clearly the interval branch voltages by both methods are best estimated in comparison with MC method. However, IGA and SS methods require longer CPU time than MC method. Hence the number of interval parameters subject to partitioning should not be excessive in practice.

6. Conclusion

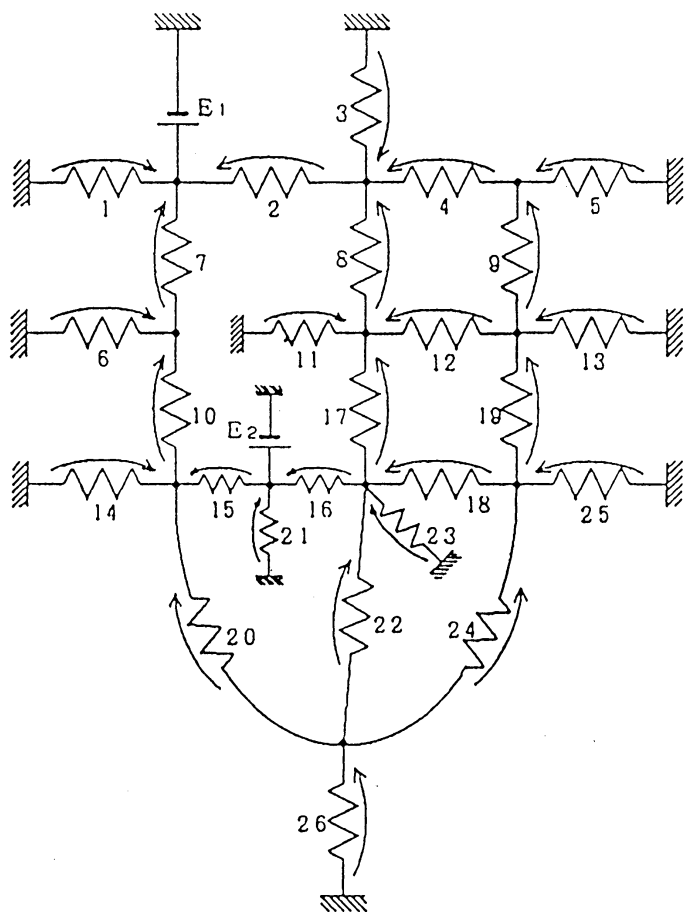
We have presented methods for better estimation of the interval solutions of linear resistive network. In order to reduce the computational time, choosing the minimum number of trees to cover all the resistive branches is of importance. Partitioning of the interval parameters leads us to having a good estimation of the solutions although the number of hybrid equations to be solved increases excessively. The methods described here will be effective when a parallel machine will be more common.

The author expresses his sincere thanks to Professor Emeritus of Kyoto University, Akira Kishima who has led the way for him.

References

- [1] G. Alefeld and J. Herzberger;"Introduction to Interval Computations",translation by J. Rokne, Academic Press, New York, New York, 1983.
- [2] K. Okumura;"Several Applications of Interval Mathematics to Electrical Network Analysis", submitted to RIMS Kokyuroku of Kyoto University, 1993.

Table 1 Center values of conductances.



G_1	0.400
G_2	0.013
G_3	0.800
G_4	0.038
G_5	0.320
G_6	0.928
G_7	0.019
G_8	0.031
G_9	0.019
G_{10}	0.013
G_{11}	0.800
G_{12}	0.063
G_{13}	0.448
G_{14}	0.928
G_{15}	0.063
G_{16}	0.031
G_{17}	0.013
G_{18}	0.031
G_{19}	0.013
G_{20}	0.063
G_{21}	1.280
G_{22}	0.031
G_{23}	0.800
G_{24}	0.038
G_{25}	0.768
G_{26}	1.440
E_1	0.500
E_2	1.050

Fig. 1 Resistive network. Arrow shows direction of branch voltage.

Table 2 Trees and their branch numbers.

	Branch number
T_1	1, 3, 5, 6, 11, 13, 14, 21, 23, 25, 26
T_2	1, 2, 4, 7, 9, 10, 12, 16, 18, 19, 22
T_3	2, 3, 5, 7, 8, 13, 15, 17, 20, 21, 24

Table 3 Interval solutions of branch voltages by MC method,
IGA and SS method.

	MONTE CARLO		GAUSS	SS
V1	[0.500000	0.500000]	1.0000	1.0000
V2	[0.490969	0.493896]	1.0755	1.0763
V3	[0.006104	0.009031]	1.0258	1.0666
V4	[0.005377	0.008249]	1.0939	1.1080
V5	[0.000524	0.001084]	1.1392	1.1520
V6	[0.008742	0.013217]	1.0393	1.0411
V7	[0.486783	0.491258]	1.0812	1.0824
V8	[0.005119	0.008351]	1.1505	1.1585
V9	[0.000331	0.000933]	1.3198	1.3306
V10	[-0.065901	-0.038925]	1.1266	1.1295
V11	[0.000543	0.001122]	1.1903	1.2002
V12	[0.000412	0.000938]	1.3521	1.3751
V13	[0.000098	0.000240]	1.4369	1.4569
V14	[0.051612	0.075056]	1.0697	1.0878
V15	[-0.998388	-0.974944]	1.0741	1.0775
V16	[1.006290	1.020232]	1.0792	1.0811
V17	[-0.042951	-0.028955]	1.1441	1.1473
V18	[0.028348	0.042195]	1.0876	1.0949
V19	[-0.001947	-0.000848]	1.3302	1.3353
V20	[0.048242	0.071966]	1.0879	1.0978
V21	[1.050000	1.050000]	1.0000	1.0000
V22	[0.025721	0.041037]	1.1178	1.1237
V23	[0.029768	0.043709]	1.0741	1.0834
V24	[-0.003205	-0.000524]	1.2613	1.2698
V25	[0.001010	0.002110]	1.1673	1.1758
V26	[0.002273	0.004589]	1.1572	1.1642

Table 4 Interval solution of ladder network[2].

	MONTE CARLO		GAUSS	SS
V1	[6.4248	7.8960]	1.0441	1.0503
V2	[3.6234	4.8511]	1.0640	1.0723
V3	[4.9306	6.0860]	1.0755	1.0821
V4	[1.8081	2.6369]	1.0812	1.0903
V5	[0.8304	1.4132]	1.0709	1.0827
V6	[2.3123	3.5820]	1.0546	1.0663
V7	[-1.8596	-0.7401]	1.0746	1.0878
V8	[2.7787	3.8542]	1.0636	1.0732
V9	[0.8399	1.4170]	1.0871	1.0991