Chapter 2

BASICS OF CIRCUIT ANALYSIS

There has been striking progress in linear circuit reduction since the last two decades or so. Linear circuits that we will discuss throughout the book refer to parasitic RC and RLC circuits that represent interconnects on metal layers in modern ICs.

As we are entering the very deep submicron era with the smallest CMOS transistor less than 65nm in width, interconnects start playing a significant role than ever before. Interconnects are important to designers in terms of timing, crosstalk, electro-migration, power, process variation, etc.

Having stressed on the importance of interconnects, analyzing the effects caused by congested interconnects are still the bottleneck in state-of-the-art commercial EDA softwares. The reason is very simple: there are too many to analyze. A typical interconnect connecting two gates in one design module is about a few microns long. To accurately model the interconnect, a parasitic circuit with tens of RC or RLC segments shown in has to be used to model the interconnects' electrical characteristics. For all the RC and RLC circuits that we are interested, each node has one or more resistive path to others. It is an important assumption. All the reduction methods that will be introduced in this chapter take it as a prerequisite.

This chapter reviews some, but not all, of the representative work that is well established in theory and widely used in circuit reduction and simulation. A trend in linear circuit reduction is that reduced circuits are preferably passive or realizable, so that they can be simulated in standard circuit simulators such as SPICE [63].

We start the chapter by introducing time and s domain analysis methods used in circuit simulation. Then we review linear circuit reduction techniques, which can be classified into three categories: moment-matching reduction via Padé

approximation, passive reduction, and realizable reduction, with each more sophisticated than the previous one.

1. Time Domain Analysis

In this section, we describe RC and RLC network analysis in time domain progressively in three steps. Our first step is to present an approach to analyzing RC circuits. And then the second step is to formulate RLC circuits with certain special structure. Our last step is to introduce the formulation of RLC circuits of general structure. Each of them begins with a simple example and are presented formally in matrix terminologies afterwards.

1.1 *RC* Interconnect Circuit Formulation

RC interconnect circuits can be formulated using nodal analysis formulation. Nodal analysis is a classical circuit analysis method based on Kirchhoff's Current Law[1] (KCL) and branch constitutive equations[2]. For a given RC linear circuit with $n+1$ nodes, nodal analysis formulates the problem in the following two steps:

step 1. choose a ground or reference node, which usually is taken to be at a potential of zero volt. All other node voltages constitute n unknowns[3];

step 2. establish KCL equations for all the n nodes by representing branch currents in terms of node voltages using branch constitutive equations.

Example 2.1. Refer to the RC tree in Fig. 2.1. In nodal analysis formulation of the circuit, we first determine the unknowns. Since v_1 is equal to v_s which

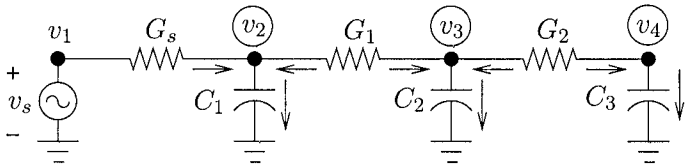

Figure 2.1. A RC tree demonstrating nodal analysis formulation: current flowing out of a node is equal to currents flowing into the node

is the given input, we use node voltages v_2, v_3, and v_4 as unknown variables to

[1] Kirchhoff's Current Law: for lumped circuits, the algebraic sum of the currents entering (leaving) a node is zero.

[2] A branch constitutive equations are i-v relationships for circuit elements such as resistors, capacitors, inductors, dependent and controlled sources, etc. For example, the branch constitutive equation for a resistor of value r is $i = v/R$.

[3] The n node voltages are independent because they linearly represent n independent voltage drops on tree trunks on any tree of the circuit.

write the three KCL equations

$$C_1\dot{v}_2 = G_s(v_s - v_2) + G_1(v_3 - v_2) \tag{1.1}$$
$$C_2\dot{v}_3 = G_1(v_2 - v_3) + G_2(v_4 - v_3) \tag{1.2}$$
$$C_3\dot{v}_4 = G_2(v_3 - v_4), \tag{1.3}$$

or if we order the unknown variables on the right-hand side of the equations, we may have

$$C_1\dot{v}_2 = -(G_s + G_1)v_2 + G_1v_3 + G_sv_s \tag{1.4}$$
$$C_2\dot{v}_3 = G_1v_2 - (G_1 + G_2)v_3 + G_2v_4 \tag{1.5}$$
$$C_3\dot{v}_4 = G_2v_3 - G_2v_4 \tag{1.6}$$

The matrix form of the above three simultaneous equations would be

$$\begin{bmatrix} C_1 & & \\ & C_2 & \\ & & C_3 \end{bmatrix} \begin{bmatrix} \dot{v}_2 \\ \dot{v}_3 \\ \dot{v}_4 \end{bmatrix} = - \begin{bmatrix} G_s + G_1 & -G_1 & 0 \\ -G_1 & G_1 + G_2 & -G_2 \\ 0 & -G_2 & G_2 \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} G_sv_s \\ 0 \\ 0 \end{bmatrix}. \tag{1.7}$$

Note that we have put a minus sign outside the square matrix.

Without loss of generality, we assume that the voltage drop and the current from ⓥ2 to ⓥ3 are two output variables that we are interested,i. e., i_x ans v_x are circuit outputs:

$$\begin{bmatrix} i_x \\ v_x \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ G_1 & -G_1 & 0 \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \\ v_4 \end{bmatrix}. \tag{1.8}$$

As the reader can imagine, we can use some linear combination of the unknowns to obtain voltages between any two nodes or currents on any branch in the circuit.

In (1.7), each row is derived from KCL for each node in Fig. 2.1. For example, the first equation states that the current flowing out of node ⓥ2 through C_1, i. e., $C_1\dot{v}_1$, is equal to the currents flowing into the node through G_s and G_1, which are $G_s(v_s - v_2)$ and $G_1(v_3 - v_2)$, respectively.

In general, RC circuit formulation can be expressed as

$$C\dot{\boldsymbol{V}}(t) = -G\boldsymbol{V}(t) + P\boldsymbol{U}(t) \tag{1.9}$$
$$\boldsymbol{Y}(t) = Q\boldsymbol{V}(t) \tag{1.10}$$

where V denotes the n unknown nodal voltages in RC circuits. In (1.9), the matrices $G \in \Re^{n\times n}$ and $C \in \Re^{n\times n}$ represent the conductance and capacitance elements, respectively.

Please note that matrix C may be singular, i. e., some rows in C may be zero. It happens when the corresponding node does not connect to any capacitor. It is

worth noting as well that the G matrix in (1.7) is non-singular if and only if the RC interconnect circuits meet the requirement that each node has one or more resistive path to some other nodes. Being a non-singular matrix is a necessary condition for the most linear reduction techniques.

1.2 *RLC* Interconnect Circuit Formulation

Formulation of a type of RLC circuits can be easily obtained by augmenting the RC formulation we have introduced. The RLC circuits in this class require that every inductor must be in series with a resistor. In fact when L is considered, the parasitics of an interconnect segment typically is modeled as a RLC branch as shown in Fig. 2.2. The branch constitutive equation of L in Fig. 2.2 is given

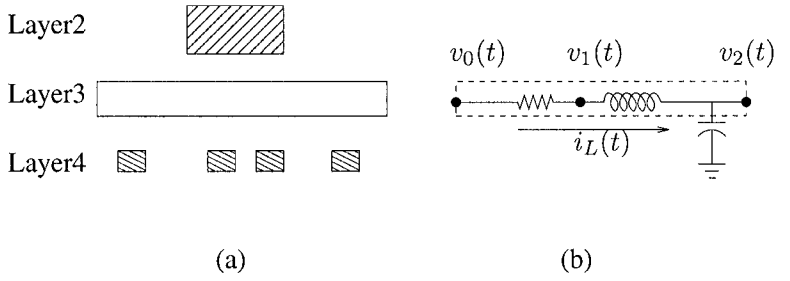

Figure 2.2. RLC parasitics of a segment of interconnect on metal layers:(a) illustration of orthogonal interconnect layers (b) RLC parasitic model of an interconnect segment in layer 3.

by

$$v_1(t) - v_2(t) = L\dot{i}_L(t), \tag{1.11}$$

where $v_1(t)$ and $v_2(t)$ are the two nodal voltages of the inductor.

The formulation method that we described for RC circuits can be used to formulate RLC circuits with minor modifications. We have known that each row in the formulation is constituted by KCL. KCL equations are established in terms of nodal voltages and their derivatives as unknowns. If we want to keep the formulation, $i_L(t)$ has to be represented with nodal voltages.

In general, $i_L(t)$ can be calculated by

$$i_L(t) = \frac{1}{L}\int_{t_0}^{t}(v_1(t) - v_2(t))\mathrm{d}t + i_L(t_0), \tag{1.12}$$

This integral equation, however, is apparently not a fit to our RC formulations because only nodal voltages and their derivatives can be used as unknowns.

Fortunately, provided that the inductor is in series with a resistor in our RLC circuits, $i_L(t)$ is equal to the current flowing through the resistor as well, i.e.,

$$i_L(t) = G\left(v_0(t) - v_1(t)\right), \tag{1.13}$$

where $v_0(t)$ and $v_1(t)$ are the two nodal voltages of the resistor. Therefore, insert (1.13) into (1.11), we can rewrite (1.11) into the form of

$$\begin{aligned} v_1(t) - v_2(t) &= L\dot{i}_L(t) \\ &= LG\left(\dot{v}_0(t) - \dot{v}_1(t)\right). \end{aligned} \tag{1.14}$$

Essentially we have used the nodal voltages of the resistor to represent the current of the inductor.

RLC circuit formulation can be augmented based on *RC* formulation as follows:

step 1. choose a ground or reference node, which usually is taken to be at a potential of zero volt. All other node voltages constitute n unknowns;

step 2. establish KCL equations for all the *inter-branch nodes (those on the joints of branches)* by representing branch currents in terms of node voltages using branch constitutive equations. *For branch that is an inductor, use (1.13) to represent the current in the inductor.*

step 3. establish (1.14) for all intra-branch nodes (those inside branches between resistors and inductors).

Example 2.2. We change the circuit in Fig. 2.1 to the one in Fig. 2.3 by adding two inductors L_1 and L_2 in series with G_1 and G_2, respectively. This circuit structure meets our requirement: L_1 is in series with G_1, and L_2 is in series with G_2.

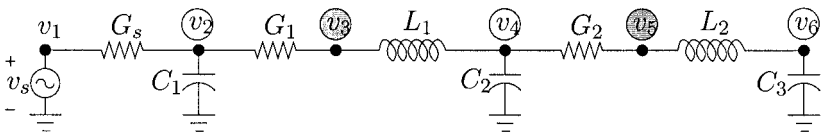

Figure 2.3. *RLC* circuit meeting the two prerequisites. Shaded ones are the so-called intra-branch nodes

The circuit can be formulated as (1.15). The first three equations are established based on Step 2. While the last two are based on Step 3. In terms of nodes, the first three rows are derived from KCL for three inter-branch nodes. The last two rows are modified branch constitutive equations (1.14) for two

intra-branch nodes.

$$\begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_3 \\ -G_1L_1 & G_1L_1 & 0 & 0 & 0 \\ 0 & 0 & -G_2L_2 & G_2L_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_2 \\ \dot{v}_3 \\ \dot{v}_4 \\ \dot{v}_5 \\ \dot{v}_6 \end{bmatrix} =$$
$$- \begin{bmatrix} G_s+G_1 & -G_1 & 0 & 0 & 0 \\ -G_1 & G_1 & G_2 & -G_2 & 0 \\ 0 & 0 & -G_2 & G_2 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + \begin{bmatrix} G_s v_s \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (1.15)$$

Letting the branch current and voltage of L_2 be the outputs, we have

$$\begin{bmatrix} v_{L_2} \\ i_{L_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & G_2 & -G_2 & 0 \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix}. \quad (1.16)$$

In general, RLC circuit formulation can be expressed as

$$C\dot{\boldsymbol{v}}(t) = -G\boldsymbol{v}(t) + P\boldsymbol{u}(t) \quad (1.17)$$
$$\boldsymbol{y}(t) = Q\boldsymbol{v}(t) \quad (1.18)$$

Similar to the RC formulation, V denotes the n unknown nodal voltages in RLC circuits, and matrices $G \in \Re^{n\times n}$ and $C \in \Re^{n\times n}$ represent the conductance and capacitance elements. In addition, the two matrices contain other elements such as 1 and GL due to the introduction of modified branch constitutive equations (1.14).

Same as the RC formulation, matrix C may be singular. We assume that this will not happen in RLC circuits throughout our discussion. The same assumption for RC circuits applies to the RLC circuits to ensure that matrix G is non-singular.

1.3 General RLC Interconnect Circuit Formulation

For more general RLC circuits of which our assumption to the topology of L does not hold, a more general formulation is needed. Modified nodal analysis is yet another classical circuit formulation method which improves nodal analysis method by adding currents in inductors as unknown variables. The introduction of the inductance current variables would help keep modified nodal analysis formulation in the differential form.

For example, the branch constitutive equation of an inductor is

$$v_L = L\frac{\mathrm{d}i_L}{\mathrm{d}t}, \tag{1.19}$$

where L is the inductance value. If we had to use nodal analysis, in the KCL equations involving the inductor, i_L has to be represented in terms of v_L, i. e., $i_L = \frac{1}{L}\int_{t_0}^{t} v_L \mathrm{d}t + i(t_0)$. While in the modified version, introducing i_L into the unknowns would keep the KCL equations in the differential form. The cost, however, is an additional equation (1.19).

Our general RLC circuit formulation can take one step further from modified nodal analysis, additional inductance current variables can be removed from the formulation by block Gauss elimination. However, this can be done only in s domain.

step 1. choose a ground or reference node, which usually is taken to be at a potential of zero volt. All other node voltages constitute n unknowns;

step 2. create a current variable for each inductor with certain direction defined;

step 3. establish KCL equations for all the n nodes by representing branch currents of RC elements in terms of node voltages and current variables pre-defined in step 2;

step 4. establish the branch constitutive equation of inductance in differential form of (1.19) using pre-defined current and nodal voltage variables;

step 5 (optional). remove current variables using block Gauss elimination in s domain.

Step 1$\tilde{4}$ are the procedure of modified nodal analysis on general RLC circuits. Step 5 is the post-procedure for removal of current variables.

Example 2.3. Fig. 2.4 shows an RLC circuit with a mutual inductance M between L_1 and L_2. Note that L_2 in the circuit does not meet our assumption in 1.2, i. e., it does not run in series with any resistor.

In order to formulate the circuit using modified nodal analysis, i_1 and i_2 are two current variables in addition to the four nodal voltages. The modified nodal

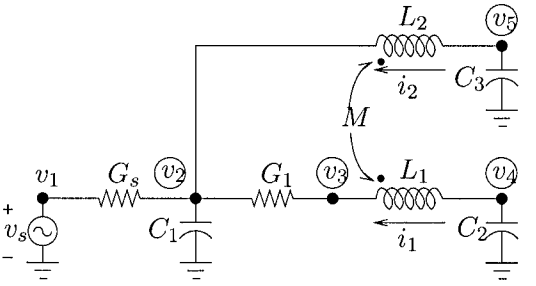

Figure 2.4. A RLC tree demonstrating modifi ed nodal analysis formulation.

analysis formulation of the circuit is given by

$$\left[\begin{array}{cccc|cc} C_1 & & & & 0 & 0 \\ & 0 & & & 0 & 0 \\ & & C_2 & & 0 & 0 \\ & & & C_3 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & L_1 & M \\ 0 & 0 & 0 & 0 & M & L_2 \end{array}\right] \left[\begin{array}{c} \dot{v}_2 \\ \dot{v}_3 \\ \dot{v}_4 \\ \dot{v}_5 \\ \dot{i}_1 \\ \dot{i}_2 \end{array}\right] =$$

$$-\left[\begin{array}{cccc|cc} G_s+G_1 & -G_1 & 0 & 0 & 0 & -1 \\ -G_1 & G_1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} v_2 \\ v_3 \\ v_4 \\ v_5 \\ i_1 \\ i_2 \end{array}\right] + \left[\begin{array}{c} G_s v_s \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right]. \quad (1.20)$$

Let v_6 be the output voltage; then we have

$$[v_6] = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] \begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ i_1 \\ i_2 \end{bmatrix}. \quad (1.21)$$

In (1.20), the first four rows are derived from KCL for the five circled nodes. The last two rows are branch constitutive equations of the two inductors and the mutual one, which are added because of the two extra variables, i_1 and i_2.

In general, modified nodal analysis formulation can be expressed as

$$M\dot{\boldsymbol{X}}(t) = -G\boldsymbol{X}(t) + P\boldsymbol{U}(t) \quad (1.22)$$

$$\boldsymbol{Y}(t) = Q\boldsymbol{X}(t) \quad (1.23)$$

where

$$\boldsymbol{x}(t) \equiv \begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{i}(t) \end{bmatrix} \quad M \equiv \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix} \quad G \equiv \begin{bmatrix} G_1 & W^T \\ E & 0 \end{bmatrix} \tag{1.24}$$

where V and I are the modified nodal analysis variables (yielding a total number of n unknowns in (1.22)) corresponding to the node voltages and the branch currents for floating voltage sources and inductors. In (1.22), the matrices $G \in \Re^{n\times n}$ and $M \in \Re^{n\times n}$ represent the conductance and susceptance matrices (except that the rows corresponding to the current variables are negated). In (1.24), C and L are generally capacitance and inductance matrices of the circuit. However, please note that C may be singular, i. e., some rows in C may be zero. It happens when the corresponding node does not connect to any capacitor. Similarly, L may be singular too, and it happens when the corresponding branch is a floating voltage source.

To go one step further to remove the extra variables in s domain, (1.21) can be symbolically represented by

$$\begin{bmatrix} Cs & 0 \\ 0 & Ls \end{bmatrix} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{I} \end{bmatrix} = - \begin{bmatrix} G_1 & W^T \\ -W & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix}, \tag{1.25}$$

or

$$\begin{bmatrix} G_1 + Cs & W^T \\ -W & Ls \end{bmatrix} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix}. \tag{1.26}$$

Using block Gauss elimination, $\boldsymbol{I}$ can be first written as

$$\boldsymbol{I} = (Ls)^{-1} W. \tag{1.27}$$

Replace $\boldsymbol{I}$ in (1.26) with (1.27),

$$\left(\left[G_1 + Cs \right] + \left[W^T (Ls)^{-1} W \right] \right) \left[\boldsymbol{V} \right] = \left[\boldsymbol{B} \right] \tag{1.28}$$

1.4 Remarks

We reviewed three kinds of circuit analysis approaches: 1) RC formulation, 2) RLC formulation, and 3) general RLC formulation. The first one, also known as nodal analysis, is widely used in circuit simulation tools such as SPICE[63] for its robustness and simplicity in implementation. The second formulation method is augmented based on the RC formulation and it is different from the well-known modified nodal analysis, for it does not introduce current variables into the formulation. Therefore, our RLC formulation is more compact, and has the nice property that it guarantees the non-singularity of matrix M in (1.22) under certain assumptions. The last one is to be used on general RLC circuits which may also contain mutual inductance. Further simplification can be done to reduce the matrix size in s domain.

In the next subsection, we will give a closed form for circuit responses to RC and RLC formulations that we have introduced. In some ill cases, matrix M in (1.22) may be singular. When it happens, the closed form solution will not be available. Therefore we reiterate our assumptions to the RC and RLC circuits of our interest: when an inductor is present in circuits, there has to be a series resistor with it; each inter-node has a coupling or ground capacitor. Under these assumptions, M is non-singular.

2. Responses in Time Domain

2.1 Responses in Closed Form

Before proceeding to s domain analysis, we discuss the time domain solution of linear networks derived from RC and RLC formulations in the previous subsection. Let us pre-multiply matrix M^{-1} on both sides of (1.22); then we have

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= -M^{-1}G\boldsymbol{x}(t) + M^{-1}P\boldsymbol{u}(t) \\ &\equiv A\boldsymbol{x}(t) + B\boldsymbol{u}(t), \end{aligned} \tag{2.1}$$

where $A \equiv -M^{-1}G$ and $B \equiv M^{-1}P$. Pre-multiply e^{-At} on both sides, (2.1) can be written as

$$\begin{aligned} e^{-At}\dot{\boldsymbol{x}}(t) - e^{-At}A\boldsymbol{x}(t) &= e^{-At}B\boldsymbol{u}(t) \\ \frac{\mathrm{d}\left[e^{-At}\boldsymbol{x}(t)\right]}{\mathrm{d}t} &= e^{-At}B\boldsymbol{u}(t) \end{aligned} \tag{2.2}$$

The solution to the above differential equation is the time-domain response on circuit nodes:

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_{t_0}^{t} e^{A(t-\tau)}B\boldsymbol{u}(\tau)\mathrm{d}\tau \tag{2.3}$$

where $\boldsymbol{x}_0$ is the initial condition, i. e., $\boldsymbol{x}_0 = \boldsymbol{x}(t)|_{t=t_0}$.

The output response in time domain is derived from (2.3) by pre-multiplying matrix Q:

$$\boldsymbol{y}(t) = Q\boldsymbol{x}(t) = Q\boldsymbol{x}_0 + Q\int_{t_0}^{t} e^{A(t-\tau)}B\boldsymbol{u}(\tau)\,\mathrm{d}\tau \tag{2.4}$$

The first term is the output at time $t = t_0$. The second term is the convolution of the impulse response and the input waveform. The result can be verified by Laplace and inverse Laplace transformations.

2.2 Taylor Expansion in Time Domain

The matrix exponential e^{At} is defined by Taylor expansion:

$$e^{At} = 1 + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots + \frac{(At)^n}{n!} + \cdots. \tag{2.5}$$

Therefore, the solution (2.3), with $t_0 = 0$ without losing any generality, can be rewritten as:

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{x}_0 + \int_0^t e^{A(t-\tau)} B\boldsymbol{u}(\tau)\,\mathrm{d}\tau \\ &= \boldsymbol{x}_0 + \int_0^t \sum_{i=0}^{\infty} \frac{(A(t-\tau))^i}{i!} B\boldsymbol{u}(\tau)\,\mathrm{d}\tau \\ &= \boldsymbol{x}_0 + \sum_{i=0}^{\infty} \frac{A^i B}{i!} \int_0^t (t-\tau)^i \boldsymbol{u}(\tau)\,\mathrm{d}\tau. \end{aligned} \tag{2.6}$$

If we approximate $\boldsymbol{x}(t)$ by first k terms:

$$\boldsymbol{x}(t) \cong \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \frac{A^i B}{i!} \int_0^t (t-\tau)^i \boldsymbol{u}(\tau)\,\mathrm{d}\tau \tag{2.7}$$

Define

$$a_i \equiv \frac{A^i B}{i!} \tag{2.8}$$

$$\boldsymbol{x}_i(t) \equiv \int_0^t (t-\tau)^i \boldsymbol{u}(\tau)\,\mathrm{d}\tau \tag{2.9}$$

Therefore,

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \left[a_i \boldsymbol{x}_i(t)\right]. \tag{2.10}$$

We can get all the a_i by k Matrix-Vector multiplications. If the matrix is in Harwell-Boeing Format, the complexity of Matrix-Vector multiplication grows linearly with the number of non-zero elements in Matrix.

The evaluation of $\boldsymbol{x}_i(t)$ needs to take the source vector $U(t)$ into account. For a vector of constant sources, $\boldsymbol{u}(t) = \boldsymbol{\alpha}$,

$$\boldsymbol{x}_i(t) = \frac{t^{i+1}}{i+1}\boldsymbol{\alpha}. \tag{2.11}$$

For a vector of linear sources, $\boldsymbol{u}(t) = \boldsymbol{\alpha} t$,

$$\boldsymbol{x}_i(t) = \frac{t^{i+2}}{(i+1)(i+2)}\boldsymbol{\alpha}. \tag{2.12}$$

Sources with classical waveforms, such as exponential or sinusoidal function, have also closed form representation of (2.10). Some sources, on the other hand, are combinations of different ones mentioned above. One of its kind is piece-wise linear voltage or current source, which is a combination of a series of timed

ramp inputs. Because system analyzed here is a linear network, if system has different kinds of source, we can calculate $x_i(t)$ for each independent source alone and sum the response together.

Please note that a_i and b_i have no relation to source $u(t)$ and time t, thus a_i and b_i need to be computed only once. For each interested time point t, calculate the $x_i(t)$, substitute into (2.10), we can get the result value.

The choice of k will greatly affect the accuracy of this method. For a given error tolerance, we want to find out the smallest k that satisfies the accuracy requirement. Since the value of $u(t)$ is bounded in real circuit(e. g., less than 5V), We consider $u(t)$ is constant α. Substitute (2.11) into (2.10), local truncation error LTE can be approximated as:

$$\begin{aligned} LTE &\cong \sum_{i=k}^{\infty} \frac{(\| A \|)^i \| B\alpha \| t^{i+1}}{(i+1)!} \\ &< \frac{\| A \|^k \| B\alpha \| t^{k+1}}{(k+1)!(1 - \frac{\|A\|t}{k+2})}. \end{aligned} \tag{2.13}$$

Here $\| \cdot \|$ is 1-norm($\| \cdot \|_1$) or ∞-norm($\| \cdot \|_\infty$) of matrix. $\frac{\|A\|t}{k+2}$ must be smaller than 1, otherwise the local truncation error does not converge. Because $\| A \|$ is a fixed value for a given circuit, k and t can be adjusted mutually to satisfy the convergence condition. Specifically, if t is equal to T, the time point when circuit response is desired, then k has to be large enough such that $k+2$ is greater than $\| A \| T$. On the other hand, if k is set to a fixed value, e. g., 100, T may have to be time stepped such that individual time step t is small enough to make $\| A \| t$ smaller than $k+2$. In summary, the smaller t is, the smaller k could be.

For a given time point T, the absolute truncation error $ATE = \frac{T}{t} LTE$, i. e.,

$$ATE < \frac{\| A \|^k \| B\alpha \| t^k}{(k+1)!(1 - \frac{\|A\|t}{k+2})}. \tag{2.14}$$

Theoretically because self-multiplication of matrix A is more expensive than matrix-vector multiplication in (2.10), and A^iB is much smaller than the square matrix A in terms of dimension, we can select time step t as small as possible. Thus for the same absolute truncation error, smaller k is allowed.

For non-stiff systems, Taylor expansion method can approach the accuracy of SPICE with about one or two order less computing time. For stiff systems, this method requires small time steps compared to the interested time interval. It will generate extremely long simulation time. Practically, $k = 10$ is a reasonable number for most of systems.

3. s Domain Analysis

In this section, we discuss how to obtain the transfer function matrix of a linear network from the modified nodal analysis formulation in s domain, and how to convert the responses in s domain to the time domain.

3.1 Transfer Function

The Laplace transformation of the modified nodal analysis equations (1.22) and (1.23) is given by

$$sM\boldsymbol{X}(s) - M\boldsymbol{X}_0 = -G\boldsymbol{X}(s) + P\boldsymbol{U}(s) \tag{3.1}$$

$$\boldsymbol{Y}(s) = Q\boldsymbol{X}(s). \tag{3.2}$$

Recall that $\boldsymbol{X}_0$ is the initial condition of the time domain vector $\boldsymbol{X}(t)$. Pre-multiply G^{-1} on both sides of (3.1) and obtain

$$(I + sG^{-1}M)\boldsymbol{X}(s) = G^{-1}P\boldsymbol{U}(s) + G^{-1}M\boldsymbol{X}_0, \tag{3.3}$$

or

$$(I - sA)\boldsymbol{X}(s) = B\boldsymbol{U}(s) + C\boldsymbol{X}_0, \tag{3.4}$$

where

$$A \equiv -G^{-1}M \qquad B \equiv G^{-1}P \qquad C \equiv G^{-1}M\boldsymbol{X}_0. \tag{3.5}$$

Therefore, we can derive the solution to (3.1) as

$$\boldsymbol{X}(s) = (I - sA)^{-1}B\boldsymbol{U}(s) + (I - sA)^{-1}C\boldsymbol{X}_0. \tag{3.6}$$

Insert (3.6) into (3.2); the output, $\boldsymbol{Y}(s)$, can be represented by

$$\boldsymbol{Y}(s) = Q(I - sA)^{-1}B\boldsymbol{U}(s) + Q(I - sA)^{-1}C\boldsymbol{X}_0. \tag{3.7}$$

So the transfer function matrix defining the relationship between the input $\boldsymbol{U}(s)$ and the output $\boldsymbol{Y}(s)$ is given by

$$H(s) = Q(I - sA)^{-1}B. \tag{3.8}$$

Example 2.4. To interpret the definition of the transfer function matrix $H(s)$, let us consider a linear network with two input terminals and three output terminals shown in Fig. 2.5. The I/O terminals are related by the transfer function matrix below:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \\ H_{31}(s) & H_{32}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}. \tag{3.9}$$

In the transfer function matrix, $H_{ij}(s)$ is the impulse response at output Y_i when U_j has an impulse input and the other input terminal is off[4].

[4]To turn off the input terminal, if it is connected to a voltage source, it has to be grounded; if connected to a current source, it has to be disconnected.

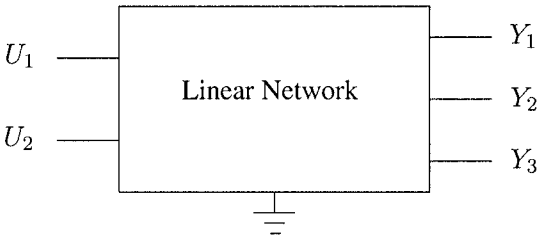

Figure 2.5. A linear network with two inputs and three outputs.

3.2 Responses from s Domain to Time Domain

In general, matrix $H(s)$ in (3.8) would be a $M \times N$ matrix, where M and N are the numbers of inputs and outputs of the system, respectively. Due to term $(I - sA)^{-1}$ in (3.8), each entry in $H(s)$ is a real rational function of s, i. e.,

$$H_{ij}(s) = \frac{Y_i(s)}{U_j(s)} \tag{3.10}$$

$$= \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_m s^m}{b_0 + b_1 s + b_2 s^2 + \cdots + b_n s^n} \tag{3.11}$$

$$= \frac{a_m (s - z_1)(s - z_2) \cdots (s - z_m)}{b_m (s - p_1)(s - p_2) \cdots (s - p_n)}, \tag{3.12}$$

where a_i and b_i are real coefficients of the polynomial expressions of s, and z_i and p_i are the *zeros* and *poles* of the transfer function, respectively.

Furthermore, all the entries share the same denominator. In fact, the partial fraction decomposition of $H_{ij}(s)$ is given by

$$H_{ij}(s) = \frac{k_1}{s - p_1} + \frac{k_2}{s - p_2} + \cdots + \frac{k_n}{s - p_n}, \tag{3.13}$$

or

$$H_{ij}(s) = \frac{r_1}{1 + s\lambda_1} + \frac{r_2}{1 + s\lambda_2} + \cdots + \frac{r_n}{1 + s\lambda_n}, \tag{3.14}$$

where λ_i is the i-th eigenvalue of square matrix $A_{n \times n}$. It is worth noting that $p_i = -1/\lambda_i$, which is the relationship between system poles and eigenvalues.

Particularly, from the expression in partial fraction decomposition, the time domain impulse response at output terminal Y_j to the impulse input at input terminal U_i can be obtained via inverse Laplace transformation:

$$y_j(t) = k_1 e^{p_1 t} + k_2 e^{p_2 t} + \cdots + k_n e^{p_n t}. \tag{3.15}$$

For an arbitrary input signal at U_i, performing convolution on the impulse response and the signal gives us the time domain response at Y_j. For arbitrary input signals at all input terminals, time domain output responses at Y_j can be obtained via principle of superposition.

4. Preliminaries of Symbolic Analysis

In this section, we briefly review some mathematic notations and theories relevant to the graph-based symbolic analysis techniques to be discussed in details in the later chapters.

4.1 Matrix, Determinant, and Cofactors

Let $e = \{1, ..., n\}$ be a set of integers. Let A denote a set of m elements, called *symbolic parameters* or simply *symbols*, $\{a_1, ..., a_m\}$, where $1 \leq m \leq n^2$ and each symbol is labeled by a unique pair (r, c), where $r \in e$ and $c \in e$. Often, we write A as an $n \times n$ (square) *matrix*, denoted by $\mathbf{A}$, and use $a_{r,c}$ to denote the element of matrix $\mathbf{A}$ at row r and column c. We sometimes use $r(a)$ and $c(a)$ to denote, respectively, the row and column indices of element a.

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & ... & a_{1,n} \\ a_{2,1} & a_{2,2} & ... & a_{2,n} \\ ... & ... & ... & ... \\ a_{n,1} & a_{n,2} & ... & a_{n,n} \end{bmatrix}.$$

If $m = n^2$ the matrix is said to be *full*. If $m << n^2$ the matrix is said to be *sparse*. The *determinant* of $\mathbf{A}$, denoted by $det(\mathbf{A})$, is defined by

$$\begin{vmatrix} a_{1,1} & a_{1,2} & ... & a_{1,n} \\ a_{2,1} & a_{2,2} & ... & a_{2,n} \\ ... & ... & ... & ... \\ a_{n,1} & a_{n,2} & ... & a_{n,n} \end{vmatrix} = \sum_{j_1 \neq j_2 \neq ... \neq j_n} (-1)^p \cdot a_{1,j_1} \cdot a_{2,j_2} ..... a_{n,j_n}. \quad (4.1)$$

Here $(j_1, j_2, ..., j_n)$ is a permutation of e, and p is the number of permutations needed to make the sequence $(j_1, j_2, ..., j_n)$ monotonically increasing. The right hand side of (4.1) is a symbolic expression of $det(\mathbf{A})$ in the *expanded* form, more precisely, the *sum-of-product* form, where each *term* is an algebraic product of n symbolic parameters. We note that each symbol can be assigned a real or complex value for analog circuit simulation.

Let p, $p \subseteq e$, and q, $q \subseteq e$ such that $|p| = |q|$. The square matrix obtained from the matrix $\mathbf{A}$ by deleting those rows not in p and columns not in q forms a *submatrix* of $\mathbf{A}$, and is represented by $\mathbf{A}(p, q)$. It has dimension $|p|$ by $|q|$.

Let $a_{r,c}$ be the element of $\mathbf{A}$ at row r and column c. Let $\mathbf{A}_{a_{r,c}}$ be the $(n-1) \times (n-1)$-matrix obtained from the matrix $\mathbf{A}$ by deleting row r and column c, and let $\mathbf{A}_{\overline{a}_{r,c}}$ be the $n \times n$-matrix obtained from $\mathbf{A}$ by setting $a_{r,c} = 0$. Then, the determinant of matrix $\mathbf{A}$ can be *expanded* as below in a way similar to Shannon expansion for Boolean functions:

$$det(\mathbf{A}) = a_{r,c}(-1)^{r+c} det(\mathbf{A}_{a_{r,c}}) + det(\mathbf{A}_{\overline{a}_{r,c}}), \quad (4.2)$$

where $(-1)^{r+c} det(\mathbf{A}_{a_{r,c}})$ is referred to as the *cofactor* of $det(\mathbf{A})$ with respect to $a_{r,c}$, and $det(\mathbf{A}_{\overline{a}_{r,c}})$ as the *remainder* of $det(\mathbf{A})$ with respect to $a_{r,c}$. The

determinant $det(\mathbf{A}_{a_{r,c}})$ is called the *minor* of $det(\mathbf{A})$ with respect to $a_{r,c}$. We note that the following two special cases of the expansion above are well known as *Laplace expansions* along row r and column c, respectively:

$$det(\mathbf{A}) = \sum_{r=1}^{n} a_{r,c}(-1)^{r+c} det(\mathbf{A}_{a_{r,c}}), \tag{4.3}$$

$$det(\mathbf{A}) = \sum_{c=1}^{n} a_{r,c}(-1)^{r+c} det(\mathbf{A}_{a_{r,c}}). \tag{4.4}$$

4.2 Cramer's Rule

Cramer's rule gives the explicit solution of a system of linear equations which is foundation for all the symbolic analysis. Give a $n \times n$ system of a linear equations $A_{n\times n}x_n = b_n$:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ b_n \end{bmatrix}. \tag{4.5}$$

Assuming $det(A_{n\times n}) \neq 0$, based on the Cramer's rule, the unknown x_k can be written as

$$x_k = \frac{det(A_k)}{det(A)}. \tag{4.6}$$

Where A_k is a $n \times n$ matrix defined as

$$A_k = \begin{bmatrix} a_{1,1} & \dots & a_{1,k-1} & b_1, & a_{1,k+1} & \dots & a_{1,n} \\ a_{2,1} & \dots & a_{2,k-1} & b_2, & a_{2,k+1} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & \dots & a_{n,k-1} & b_n, & a_{n,k+1} & \dots & a_{n,n} \end{bmatrix}. \tag{4.7}$$

With the Cramer's rule, we can solve for any any unknowns $x_1, ...x_n$ explicitly. The solution can be represented by the ratio of two determinants as shown in (4.6).

If we determine $det(A_k)$ along the kth column, the Cramer's rule for the unknown x_k can be written in the following form:

$$x_k = \frac{\sum_{i=1}^{n} b_i \, (-1)^{i+k} \, det(A_{a_{i,k}})}{det(A)}. \tag{4.8}$$

Where $det(A_{a_{i,k}})$ is the minor of $det(A)$ with respect to element $a_{i,k}$. As a result, symbolic simulators are targeted at finding various network functions, each being defined as the ratio of an output unknown from x to a input from

b. These are special cases of (4.8) or (4.6). The critical issue left is how to efficiently represent the determinants symbolically given the fact the number of symbolic product terms will grow exponentially with the size of the determinant.

Springer

http://www.springer.com/978-0-387-23904-0

Symbolic Analysis and Reduction of VLSI Circuits

Qin, Z.; Cheng, C.-K.

2005, XXIV, 284 p., Hardcover

ISBN: 978-0-387-23904-0