

Simulation of RF integrated circuits

Dr. Emad Gad

2015

Contents

1	Formulation of Circuit Equations	1
1.1	Modified Nodal Analysis	1
1.1.1	Resistor Stamps	4
1.1.2	Stamps of Independent Current Sources	5
1.1.3	Capacitor Stamp	5
1.1.4	Stamp of Independent Voltage Source (IVS)	9
1.1.5	Inductor Stamp	10
1.1.6	Stamp of nonlinear conductance	13
1.1.7	Stamp of nonlinear capacitors and inductors	15
2	Harmonic Balance for Circuits with Single-Tone Excitation	21
2.1	Background on Fourier Transform for Single-Tone Signals	22
2.1.1	Discrete Fourier Transformation	23
2.1.2	Fast Fourier Transformation	25
2.1.3	Numerical Example	26
2.2	Harmonic Balance for Linear Circuits	27
2.2.1	Steady-state response for a simple circuit	27
2.2.2	Steady-state response for general linear circuits using HB	29
2.2.3	Matrix-Vector formulation for HB of Linear Circuits	31
2.3	Harmonic Balance for Nonlinear Circuits	32
2.3.1	General nonlinear circuits	34
2.4	Solving the HB equations using N-R method	39
2.4.1	Computing $\Phi(\bar{\mathbf{X}}^{(j)})$	40
2.4.2	Formulation of $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ Based on the Kronecker Product	45
2.4.3	Computing the HB Jacobian $\partial\Phi(\bar{\mathbf{X}})/\partial\bar{\mathbf{X}} _{\bar{\mathbf{X}}=\bar{\mathbf{X}}^{(j)}}$	47
2.5	Structure of the HB Jacobian Matrix	50

2.6	Error Dynamics	52
3	Harmonic Balance for Circuits with Multi-Tone Excitation	59
3.1	Multi-tone and Almost-Periodic Waveforms	59
3.2	Response of a Nonlinear Circuit to Almost-Periodic Excitations	61
3.3	Truncating the Spectrum of Almost-Periodic signals	63
3.3.1	Box truncation for two-tone waveform	64
3.3.2	Diamond truncation for two-tone waveforms	66
3.3.3	Box and diamond truncation for multi-tone waveforms	67
3.4	Harmonic Balance Analysis	68
3.5	Generalized Fourier Transform	71
3.5.1	Multi-Dimensional Fourier Transform (MDFT)	75
3.5.2	Almost-Periodic Fourier Transform (APFT)	77
3.5.3	Frequency Mapping Techniques (FMT)	78
3.5.4	Implementation Procedures for FMT	85
4	Time Domain Methods	89
4.1	Steady-State Analysis using IVP approaches	90
4.1.1	Illustrative Examples	93
4.1.2	Transient Analysis of Circuits using BE and TR	100
4.1.3	Steady-State Analysis via IVP	101
4.2	Steady-State Analysis using Boundary-Value Problem (BVP)	103
4.2.1	Shooting Method	103
4.2.2	Newton Shooting Method (NSM)	104
5	Volterra Series	111
5.1	Linear Systems and Transfer Function	111
5.1.1	Steady-State Response in Linear Systems via (TF)	112
5.2	Volterra-Series and Small-Signal Analysis	115
5.2.1	First-Order Volterra Operator	118
5.2.2	Second-Order Volterra Operator	118
5.2.3	Third-Order Volterra Operator	119
5.2.4	General n^{th} -order Volterra Kernel	119
5.3	Constructing a Response to a Sinusoidal input using Volterra Series	119
5.3.1	The Fourier Representation of the Volterra Kernels	120
5.3.2	Response in Terms of Fourier-Based kernels	121
5.3.3	Small-Signal Response using Volterra-Series	122
5.4	The Method of Nonlinear Currents	124
5.4.1	Analytical Expressions for the Frequency-domain kernels	126
5.5	Extension to Multi-Tone Analysis	136
5.5.1	Computing $v_{o1}(t)$	138
5.5.2	Discussion on the Method of Nonlinear Currents	140
5.6	Application of the Method of Nonlinear Currents to General Circuits	140

CONTENTS

Appendices	143
A The Newton-Raphson Iterative Method	143
B The Kronecker Matrix Product	147

CONTENTS

Chapter **1**

Formulation of Circuit Equations

The main objective in this chapter is to introduce the basic idea used to automatically represent general nonlinear circuits in the mathematical domain. This representation is the very first step needed to apply almost all of the simulation techniques currently implemented in commercial circuit simulators (e.g., frequency-domain, time-domain, steady-state, distortion analysis, etc.).

The techniques of representing a general circuit in the mathematical domain have matured over two or three decades ago, and all of modern circuit simulators such as HSPICE, ADS have adopted them. The basic approach used to represent circuits mathematically is known as the Modified Nodal Admittance (MNA) approach is the main subject of this chapter.

1.1 Modified Nodal Analysis

Consider the circuit shown in Figure 1.1. We will first derive the nodal equations for this circuit. Using these derivations we will be able to deduce the roles of each element in the general formulation. The knowledge of each element's role will then enable us to derive the systematic approach employed in the MNA technique to represent general circuits.

By writing Kirchhoff Current Law (KCL) at each node in the circuit, we obtain 4 differ-

1.1. MODIFIED NODAL ANALYSIS

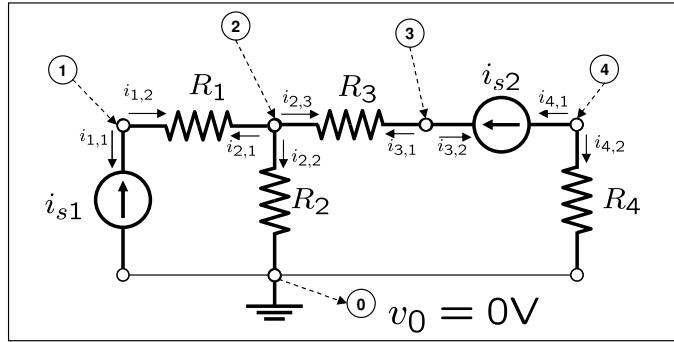


Figure 1.1: A simple resistive circuit.

ent equations,

$$\sum \text{Currents @ node 1} = 0$$

$$\sum \text{Currents @ node 2} = 0$$

$$\sum \text{Currents @ node 3} = 0$$

$$\sum \text{Currents @ node 4} = 0$$

Note that we did not include a nodal equation at the reference since the voltage at this node is assumed to be zero by convention. Using the currents designated at each node and substituting in the above equations yield,

$$\sum_{k=1}^2 i_{1,k} = 0$$

$$\sum_{k=1}^3 i_{2,k} = 0$$

$$\sum_{k=1}^2 i_{3,k} = 0$$

$$\sum_{k=1}^2 i_{4,k} = 0$$

The currents in the above equations can be expressed in terms of the nodes voltages using

simple Ohm's law. For example,

$$i_{1,2} = \frac{1}{R_1} (v_1 - v_2)$$

while $i_{1,1}$ is related to the current from the independent current source,

$$i_{1,1} = -i_{S_1}$$

Substituting for the currents in terms of the nodes voltages gives us

$$\begin{aligned} \frac{1}{R_2} (v_1 - v_2) - i_{S_1} &= 0 \\ \frac{1}{R_1} (v_2 - v_1) + \frac{v_2}{R_2} + \frac{v_2 - v_3}{R_3} &= 0 \\ \frac{1}{R_3} (v_3 - v_2) - i_{S_2} &= 0 \\ \frac{1}{R_4} v_4 + i_{S_2} &= 0 \end{aligned}$$

The above equations can be arranged in matrix/vector form as follows,

$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & 0 \\ -\frac{1}{R_1} & \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}\right) & -\frac{1}{R_3} & 0 \\ 0 & -\frac{1}{R_3} & \frac{1}{R_3} & 0 \\ 0 & 0 & 0 & \frac{1}{R_4} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} i_{S_1} \\ 0 \\ i_{S_2} \\ -i_{S_2} \end{bmatrix} \quad (1.1)$$

or more succinctly,

$$\mathbf{Gx} = \mathbf{b} \quad (1.2)$$

where

$$\mathbf{G} = \begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & 0 \\ -\frac{1}{R_1} & \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}\right) & -\frac{1}{R_3} & 0 \\ 0 & \frac{1}{R_3} & -\frac{1}{R_3} & 0 \\ 0 & 0 & 0 & \frac{1}{R_4} \end{bmatrix} \quad (1.3)$$

1.1. MODIFIED NODAL ANALYSIS

and

$$\mathbf{x} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -i_{S_1} \\ 0 \\ i_{S_2} \\ -i_{S_2} \end{bmatrix} \quad (1.4)$$

Obviously, solving for the unknown variables \mathbf{x} will give us all nodes voltages, which in turn will enable computing all branch currents.

We would like now to examine the role of each element in the final formation of the circuit equations in (1.1). This will enable us to infer the “stamp” that each element leaves on the equations formulations.

1.1.1 Resistor Stamps

Consider the resistor R_1 in the circuit of Figure 1.1. Such a resistor is connected between nodes 1 and 2. The effect that this resistor has on the formulation is having $1/R_1$ added to the entries at (1, 1) and (2, 2) of the \mathbf{G} matrix. Also the existence of this resistor caused a value of $1/R_1$ to be subtracted from the entries at (1, 2) and (2, 1).

Similarly resistor R_3 (connected between nodes 2 and 3) influenced the \mathbf{G} matrix by adding $1/R_3$ to the entries at (2, 2) and (3, 3) and subtracting $1/R_3$ from the entries at (2, 3) and (3, 2).

This observation can be further generalized to any resistor having a resistance $R \Omega$ and connected between nodes i and j . The stamp that results from this general resistor is the *addition* of $1/R$ to the entries at (i, i) and (j, j) and the *subtraction* of $1/R$ from the entries of (i, j) and (j, i) , respectively. Figure 1.2 illustrates this notion graphically.

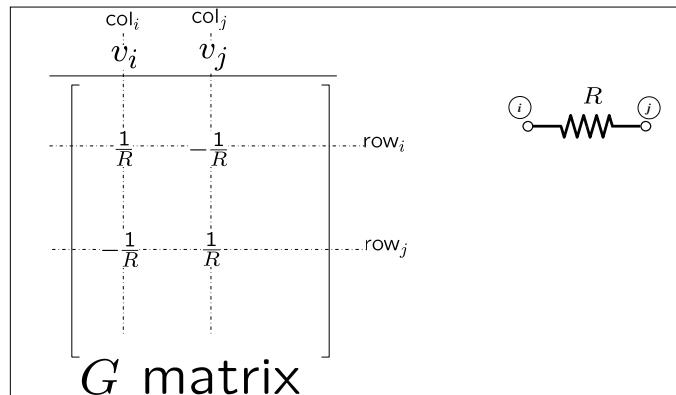


Figure 1.2: Resistor stamp.

However, if one of the nodes happened to be the reference node, such as in the case of the resistor R_4 , then the stamp reduces to only adding $1/R$ to the (k, k) entry of the \mathbf{G} matrix, as shown in Figure 1.3

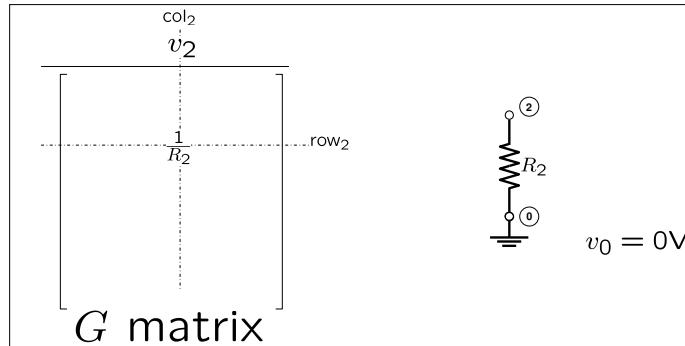


Figure 1.3: Resistor stamp, where one node is the reference node.

1.1.2 Stamps of Independent Current Sources

By examining Eq. (1.1), we can also deduce the “stamp” that an independent current source leaves on the MNA formulation. This stamp appears only on the right-side vector b , and therefore this vector is usually referred to as the *source vector*. As can be observed from Eq. (1.1), the current source i_{S_2} , which is connected between nodes 3 and 4 has left its Ampere value of i_{S_2} at the 3rd and 4th entries of the source vector b . Figure 1.4 demonstrates the general rule for including a general independent current source connected between two general nodes. Figure 1.5 also shows the stamp of an independent current source when one node is the reference node.

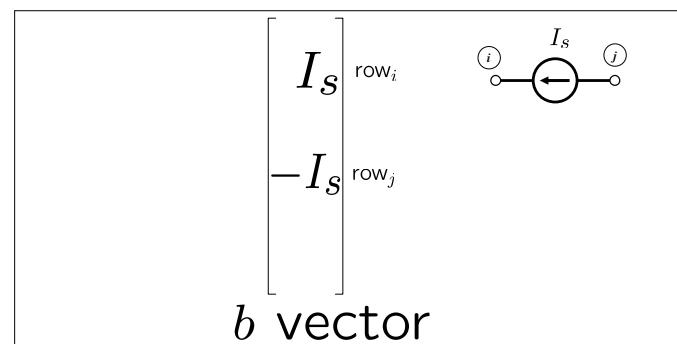


Figure 1.4: Stamp of independent current sources.

1.1.3 Capacitor Stamp

We now derive the stamp of a general capacitor in a similar manner by first formulating the circuit equations for the circuit shown in Figure 1.6. Writing KCL at each node gives,

1.1. MODIFIED NODAL ANALYSIS

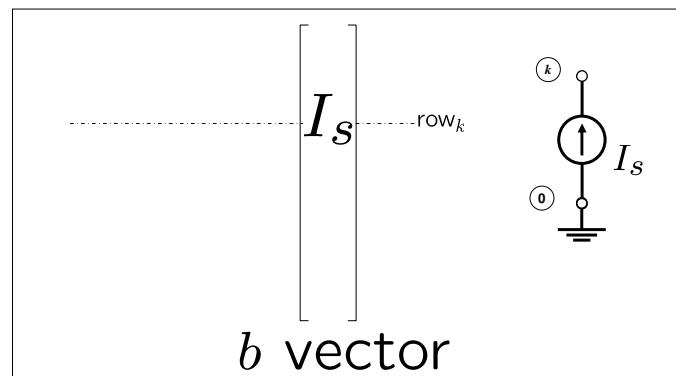


Figure 1.5: Stamps of independent current sources, where one of the nodes is a reference node.

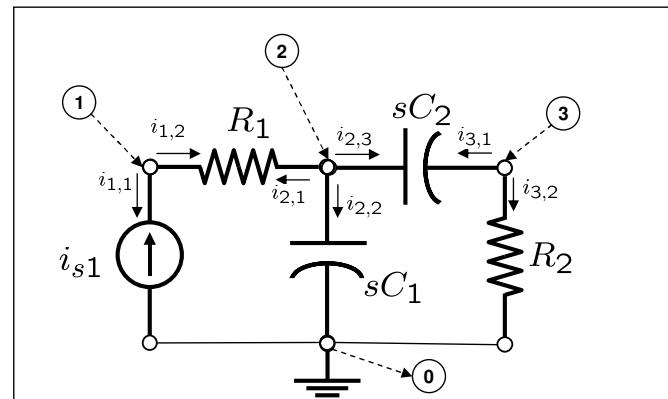


Figure 1.6: A circuit with a capacitive element.

$$\begin{aligned} i_{1,1} + i_{1,2} &= 0 \\ i_{2,1} + i_{2,2} + i_{2,3} &= 0 \\ i_{3,1} + i_{3,2} &= 0 \end{aligned} \quad (1.5)$$

Substituting for each current in terms of the nodes voltages, we get

$$\begin{aligned} i_{1,1} &= -I_{s_1} \\ i_{1,2} &= \frac{1}{R_1} (v_1 - v_2) \\ i_{2,1} &= \frac{1}{R_1} (v_2 - v_1) \\ i_{2,2} &= C_1 \frac{dv_2}{dt} \\ i_{2,3} &= C_2 \left(\frac{dv_2}{dt} - \frac{dv_3}{dt} \right) \\ i_{3,1} &= C_2 \left(\frac{dv_3}{dt} - \frac{dv_2}{dt} \right) \\ i_{3,2} &= \frac{v_3}{R_2} \end{aligned}$$

Substituting from the above equations into Eqs. (1.5) and arranging in matrix form, we get

$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} & 0 \\ 0 & 0 & \frac{1}{R_2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_1 + C_2 & -C_2 \\ 0 & -C_2 & C_2 \end{bmatrix} \begin{bmatrix} \frac{dv_1}{dt} \\ \frac{dv_2}{dt} \\ \frac{dv_3}{dt} \end{bmatrix} = \begin{bmatrix} i_{s_1} \\ 0 \\ 0 \end{bmatrix} \quad (1.6)$$

or in a more compact form

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} = \mathbf{b}(t) \quad (1.7)$$

where

$$\mathbf{x}(t) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} & 0 \\ 0 & 0 & \frac{1}{R_2} \end{bmatrix} \quad (1.8)$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_1 + C_2 & -C_2 \\ 0 & -C_2 & C_2 \end{bmatrix} \quad \mathbf{b}(t) = \begin{bmatrix} i_{s_1} \\ 0 \\ 0 \end{bmatrix} \quad (1.9)$$

One obvious conclusion from Eq. (1.6), is that capacitors do not have any effect on the

1.1. MODIFIED NODAL ANALYSIS

G matrix. Instead they appear in another matrix, denoted by C matrix, which is multiplied by the **derivatives** of the nodes voltages. It is also easy to see from the structure of the C matrix that capacitor C_2 , which is connected between nodes 2 and 3, was *added* to the entries (2, 2) and (3, 3) and *subtracted* from the entries at (2, 3) and (3, 2) of the C matrix. We can generalize this observation to any general capacitor as shown in Figures 1.7 and 1.8.

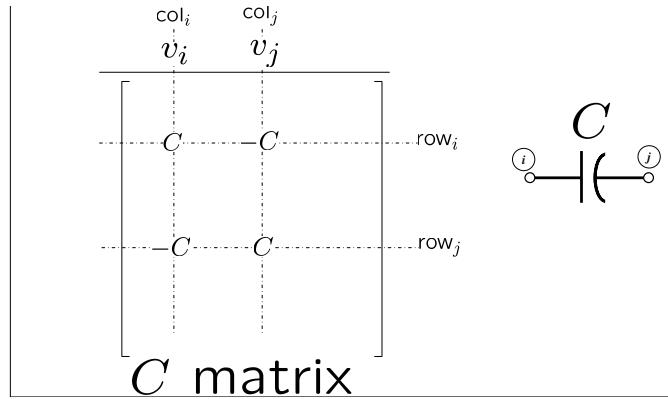


Figure 1.7: Illustrating the stamp of a capacitor on the MNA formulation.

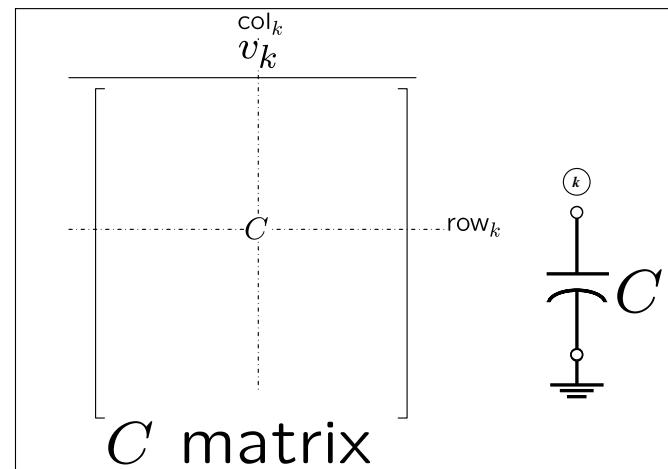


Figure 1.8: Illustrating the stamp of a capacitor on the MNA formulation (One node is the reference node).

1.1.4 Stamp of Independent Voltage Source (IVS)

Independent Voltage Sources (IVS) require special attention in the MNA formulation. To illustrate how IVS's are handled in the MNA formulation, consider the circuit shown in 1.9. Writing KCL at each node gives us

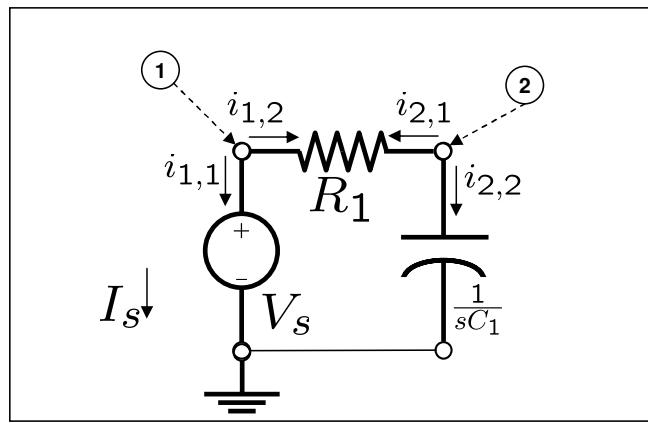


Figure 1.9: A circuit example with a independent voltage source.

$$\begin{aligned} i_{1,1} + i_{1,2} &= 0 \\ i_{2,1} + i_{2,2} &= 0 \end{aligned} \quad (1.10)$$

Substitution using Ohm's law,

$$i_{1,1} = I_s \quad (1.11)$$

$$i_{1,2} = (v_1 - v_2) \times \frac{1}{R} \quad (1.12)$$

$$i_{2,1} = (v_2 - v_1) \times \frac{1}{R} \quad (1.13)$$

$$i_{2,2} = C_1 \frac{dv_2}{dt} \quad (1.14)$$

Hence,

$$i_{1,1} + \frac{v_1}{R} - \frac{v_2}{R} = 0 \quad (1.15)$$

$$-\frac{v_1}{R} + \frac{v_2}{R} + C_1 \frac{dv_2}{dt} = 0 \quad (1.16)$$

1.1. MODIFIED NODAL ANALYSIS

Arranging in matrix form gives us

$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R} & 1 \\ -\frac{1}{R} & \frac{1}{R} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ I_s \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} dv_1/dt \\ dv_2/dt \\ dI_s/dt \end{bmatrix} = \mathbf{0} \quad (1.17)$$

The above system of equations is unbalanced, in the sense that there are more unknowns ($v_1, v_2, i_{1,1}$) than equations. To overcome this problem, we need one more equation, which can be obtained by noting that

$$v_1 = V_S \quad (1.18)$$

Adding Eq. (1.18) to the system of equations in (1.17) yields

$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R} & 1 \\ -\frac{1}{R} & \frac{1}{R} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ I_s \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} dv_1/dt \\ dv_2/dt \\ dI_s/dt \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_S \end{bmatrix} \quad (1.19)$$

which is a *square* system, i.e., a system in which the number of unknowns is equal to the number of equations.

Using the above example, we would like to derive the effect that an IVS has on the general MNA formulation. The first observation is that having an IVS increments the number of equations by 1. In addition, the value of the voltage source appears in the right hand side vector. Figures 1.10 and 1.11 summarizes the influences of IVS on the MNA formulation. Note the $+1$ and -1 entries in the G matrix.

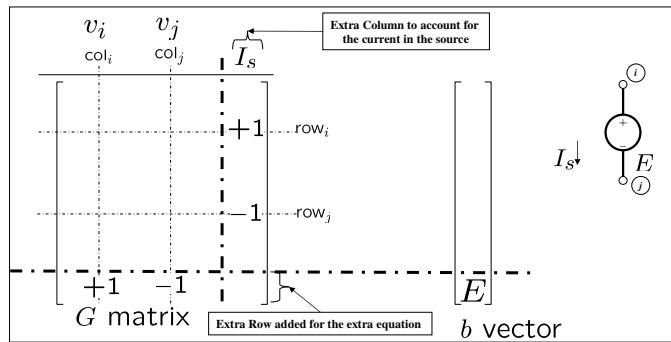


Figure 1.10: Incorporating independent voltage source in the MNA formulation. The voltage source is connected between two general nodes.

1.1.5 Inductor Stamp

To derive the inductor's stamp, consider the circuit shown in Figure 1.12. Applying KCL

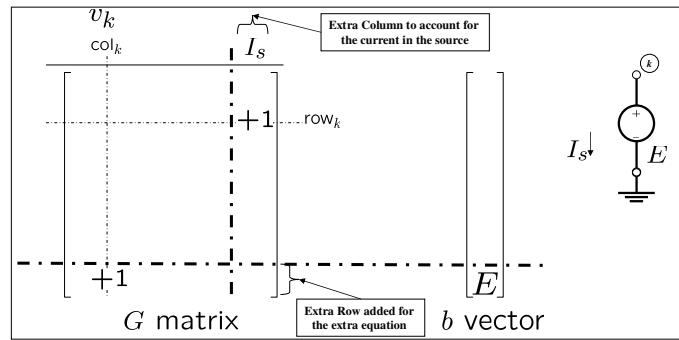


Figure 1.11: Incorporating independent voltage source in the MNA formulation. The voltage source is connected between a general node and the reference node.

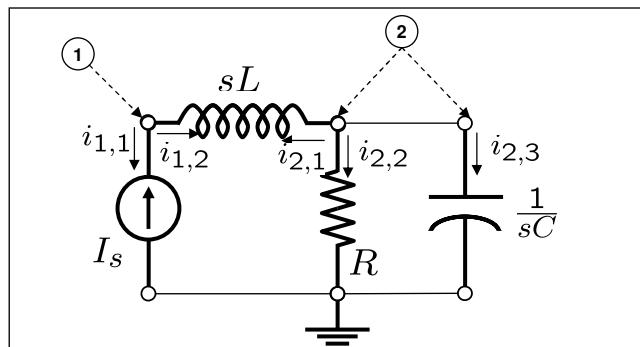


Figure 1.12: A circuit with an inductor.

1.1. MODIFIED NODAL ANALYSIS

at nodes 1, and 2 results in the following two equations

$$i_{1,1} + i_{1,2} = 0 \quad (1.20)$$

$$i_{2,1} + i_{2,2} + i_{2,3} = 0 \quad (1.21)$$

Given that using Laplace elements

$$\begin{aligned} I_{1,1} &= -I_S \\ I_{1,2} &= \frac{1}{sL}(V_1 - V_2) \\ I_{2,1} &= \frac{1}{sL}(V_2 - V_1) \\ I_{2,2} &= \frac{1}{R}V_2 \\ I_{2,3} &= sCV_2 \end{aligned}$$

Substituting in (1.20), rearranging in matrix form results in the following system of equations

$$\begin{bmatrix} 0 & 0 \\ 0 & 1/R \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + s \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \frac{1}{s} \begin{bmatrix} 1/L & -1/L \\ -1/L & 1/L \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} I_s \\ 0 \end{bmatrix} \quad (1.22)$$

The above formulation presents a particular problem when transformed to the time-domain,

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} + \mathbf{L} \int_0^t \mathbf{x}(\tau) d\tau = \mathbf{0}, \quad (1.23)$$

since the resulting system is of the integro-differential equations, which is more difficult to deal with than a system of differential equations. Therefore, it would be much better if we can find an alternative method to incorporate the inductor while keeping the differential equations form.

To this end, we add the current in the inductor as an additional unknown variable. With this modification we have

$$I_{1,2} = I_L, \quad I_{2,1} = -I_L \quad (1.24)$$

Substituting from (1.24) into (1.20), rearranging in matrix form gives us

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1/R_1 & -1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_L \end{bmatrix} + s \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_1 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_L \end{bmatrix} = \begin{bmatrix} I_s \\ 0 \\ 0 \end{bmatrix} \quad (1.25)$$

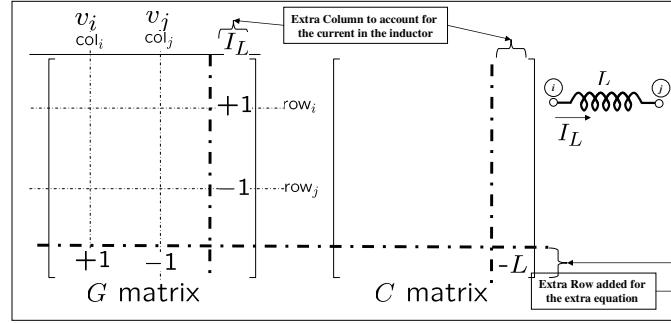


Figure 1.13: Inductor's stamp. The inductor is connected between two general nodes in the circuit.

Since we have one less equation than the number of unknowns, we will need one more equation. That equation can be obtained through applying Ohm's law on the inductor as follows,

$$V_1 - V_2 = sI_L L \quad (1.26)$$

Adding Eq. (1.26) to the system of Eqs. in (1.25) results in the following system of equations

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1/R_1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_L \end{bmatrix} + s \begin{bmatrix} 0 & 0 & 0 \\ 0 & C & 0 \\ 0 & 0 & -L \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_L \end{bmatrix} = \begin{bmatrix} I_s \\ 0 \\ 0 \end{bmatrix} \quad (1.27)$$

in which the number of equations is equal to the number of unknowns. It can also be observed that transforming the system in (1.27) to the time-domain results in the following system of differential equations.

We are now in a position to deduce the general technique that enables automatic incorporation of the inductor in the general MNA formulation. First, it is clear that having the inductor led to adding one more unknown (the inductor's current) with an additional equation. Furthermore, the inductance value appeared in the extra entry added to the **C** matrix. Figures illustrate the general stamp of an inductor.

1.1.6 Stamp of nonlinear conductance

Figure 1.15 shows a circuit with a nonlinear conductance whose current is controlled by a quadratic nonlinearity function of the applied voltage. Writing KCL at nodes 1 and 2 gives

$$\begin{aligned} i_{1,1} + i_{1,2} &= 0 \\ i_{2,1} + i_{2,2} + i_{2,3} &= 0 \end{aligned}$$

1.1. MODIFIED NODAL ANALYSIS

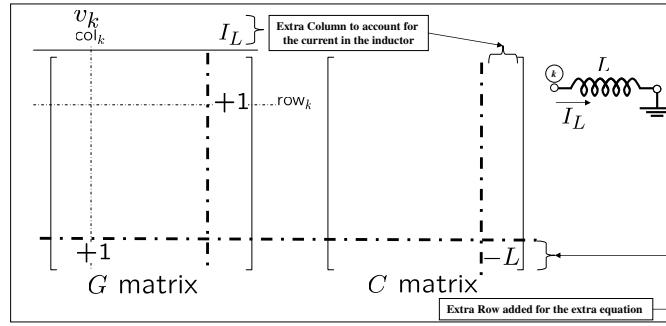


Figure 1.14: Inductor's stamp. The inductor is connected between one node and the reference node.

Substituting for the currents using

$$\begin{aligned}
 i_{1,1} &= -I_S \\
 i_{1,2} &= -i = -f(v_2 - v_1) = -(v_2 - v_1)^2 \\
 i_{2,1} &= i = f(v_2 - v_1) = (v_2 - v_1)^2 \\
 i_{2,2} &= \frac{v_2}{R} \\
 i_{2,3} &= C \frac{dv_2}{dt}
 \end{aligned}$$

and rearranging in matrix form produces the following system of equations

$$\left[\begin{array}{cc} 0 & 0 \\ 0 & 1/R \end{array} \right] \left[\begin{array}{c} v_1 \\ v_2 \end{array} \right] + \left[\begin{array}{cc} 0 & 0 \\ 0 & C \end{array} \right] \left[\begin{array}{c} \frac{dv_1}{dt} \\ \frac{dv_2}{dt} \end{array} \right] + \left[\begin{array}{c} -(v_2 - v_1)^2 \\ (v_2 - v_1)^2 \end{array} \right] = \left[\begin{array}{c} I_s \\ 0 \end{array} \right] \quad (1.28)$$

It is straightforward to see from Eq. (1.28) to see that the presence of a nonlinear conductance can be accounted for by adding a new vector whose entries are dedicated to nonlinear functions. This vector will be denoted henceforth by $\mathbf{f}(\mathbf{x}(t))$, where $\mathbf{x}(t)$ is the vector of the MNA variables. Figure 1.16 shows the stamp that a nonlinear conductance leaves on the MNA formulation. In general having a nonlinear conductance characterized by an analytic function $g(\dots)$ connected between nodes i and j results in having $g(\cdot)$ added to/subtracted from the i -th entry and subtracted to/added to the j -th entry of the nonlinear function vector $\mathbf{f}(\mathbf{x}(t))$. Notice that the stamp developed here can be used to represent a large class of two-terminal devices. One notable example is the diode whose

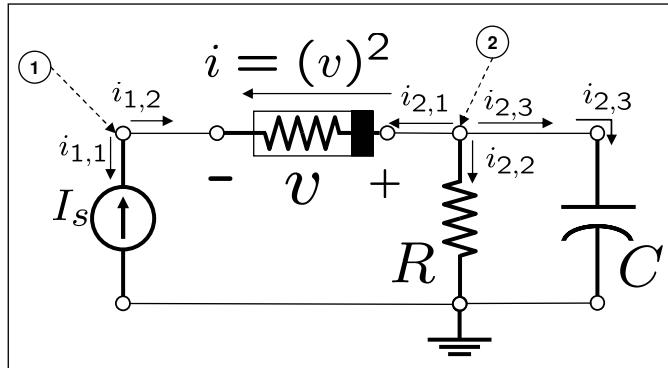


Figure 1.15: A circuit with nonlinear conductance.

current is controlled by the applied voltage through the exponential function,

$$i = I_o (e^{v/V_T} - 1) \quad (1.29)$$

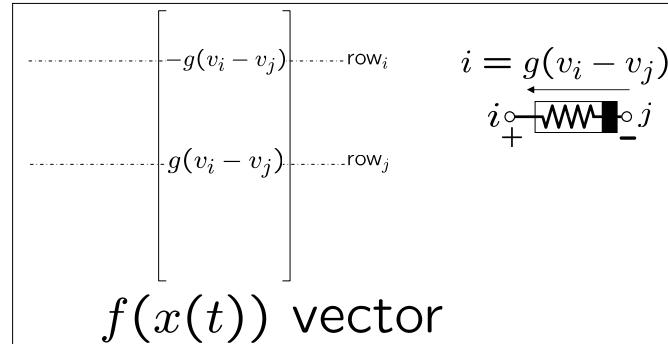


Figure 1.16: The MNA stamp of nonlinear conductance.

The general formulation developed so far for a general nonlinear circuit takes the following form

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b}(t) \quad (1.30)$$

1.1.7 Stamp of nonlinear capacitors and inductors

Figure 1.17 shows the circuit symbol that is often used to represent a nonlinear capacitor. A nonlinear capacitor is a capacitor in which the charge is a nonlinear function of the applied

1.1. MODIFIED NODAL ANALYSIS

voltage,

$$q = g(v) \quad (1.31)$$

where $g(v)$ can be any arbitrary nonlinear function. This is to be contrasted to a linear capacitor whose charge is linearly related to the applied voltage via the capacitance.

$$q = Cv \quad (1.32)$$

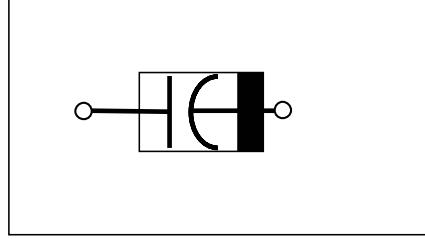


Figure 1.17: Circuit symbol for nonlinear capacitor

Since the current in the capacitor is equal to the rate of flow charge, the i - v relation for the nonlinear capacitor should be given by

$$i = \frac{dq}{dt} = \frac{d}{dt}g(v) \quad (1.33)$$

To show how a nonlinear capacitor is incorporated in the general MNA formulation, we consider the circuit shown in Figure 1.18.

Applying KCL at each node results in the following two equations,

$$I_s + i = 0 \quad (1.34)$$

$$\frac{v_2}{R} + C \frac{dv_2}{dt} + f(v_2) - i = 0 \quad (1.35)$$

where i is the current flowing in the nonlinear capacitor. Substituting i with $\frac{d}{dt}(g(v_1 - v_2))$ in the above two equations and arranging them in matrix form, we obtain

$$\begin{bmatrix} 0 & 0 \\ 0 & 1/R \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \frac{dv_1}{dt} \\ \frac{dv_2}{dt} \end{bmatrix} + \begin{bmatrix} 0 \\ f(v_2) \end{bmatrix} + \frac{d}{dt} \left(\begin{bmatrix} g(v_1 - v_2) \\ -g(v_1 - v_2) \end{bmatrix} \right) = \begin{bmatrix} I_s \\ 0 \end{bmatrix} \quad (1.36)$$

or in a more concise form,

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} + \mathbf{f}(\mathbf{x}(t)) + \frac{d}{dt} \mathbf{g}(\mathbf{x}(t)) = \mathbf{b} \quad (1.37)$$

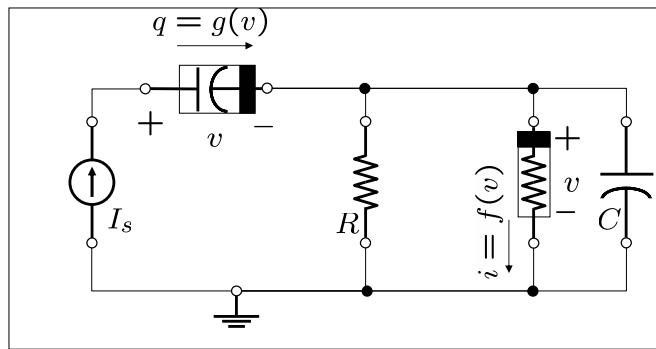


Figure 1.18: A circuit with nonlinear capacitor

It can be seen from the above equations that the presence of a nonlinear capacitor has forced one more vector of nonlinear functions, whose components are to be differentiated with respect to time t . That may not be a desirable feature, especially for some of the techniques used in Chapter 3, where the nonlinearities will be required to appear in pure algebraic rather than the differential form of the fourth term (1.37).

Overcoming this difficulty is typically done by treating the charge in the nonlinear capacitor as an additional unknown variable in the circuit, and adding one more equation to the system. The addition equation is basically the constitutive equation for the nonlinear capacitor given in (1.31). Thus, for the example circuit of Figure 1.18, we should have three equations given by

$$I_s + \frac{dq}{dt} = 0 \quad (1.38)$$

$$\frac{v_2}{R} + C \frac{dv_2}{dt} + f(v_2) - \frac{dq}{dt} = 0 \quad (1.39)$$

$$q - g(v_2) = 0 \quad (1.40)$$

Arranging in matrix form results in the following system of equations

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/R & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 & +1 \\ 0 & C & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{dv_1}{dt} \\ \frac{dv_2}{dt} \\ \frac{dq}{dt} \end{bmatrix} + \begin{bmatrix} 0 \\ f(v_2) \\ -g(v_2) \end{bmatrix} = \begin{bmatrix} I_s \\ 0 \\ 0 \end{bmatrix} \quad (1.41)$$

1.1. MODIFIED NODAL ANALYSIS

which is equivalent to

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b} \quad (1.42)$$

where $\mathbf{x}(t)$ is the vector of the MNA variables which contains the charge waveform in the nonlinear capacitor in addition to other unknown variables. Based on the above analysis, it is possible to deduce the stamp that a nonlinear capacitor will leave on the general MNA formulation as shown in Figures 1.21 and 1.22.

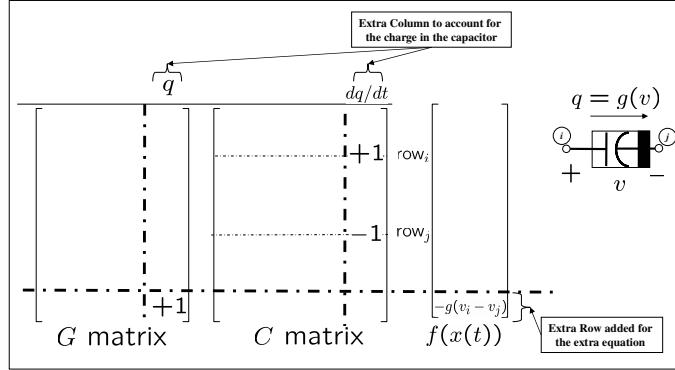


Figure 1.19: Stamp of a nonlinear capacitor connected between nodes i and j .

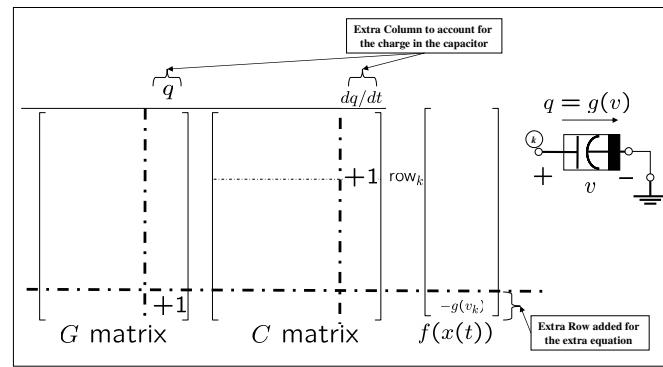


Figure 1.20: Stamp of a nonlinear capacitor connected between node k and the reference node.

The above analysis can be used in an analogous manner to deduce the MNA stamp of a nonlinear inductor. In a nonlinear inductor, the flux is a nonlinear function of the applied voltage,

$$\phi = u(v) \quad (1.43)$$

where u can be any arbitrary nonlinear function. Using the fact that the current in the inductor is the rate of flux change, the i - v relation in the inductor is given by

$$i = \frac{d}{dt}u(v) \quad (1.44)$$

The MNA stamp for the inductor is shown in Figures 1.21 and 1.22.

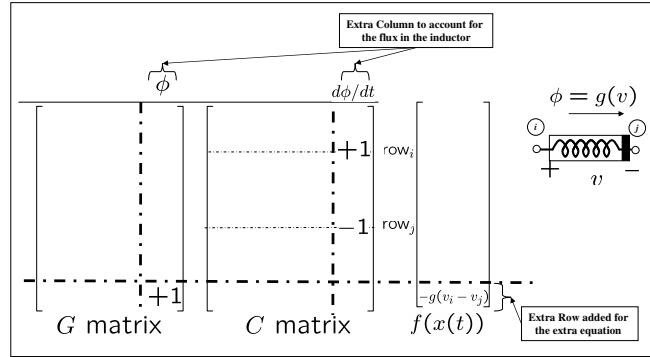


Figure 1.21: Stamp of a nonlinear inductor connected between nodes i and j .

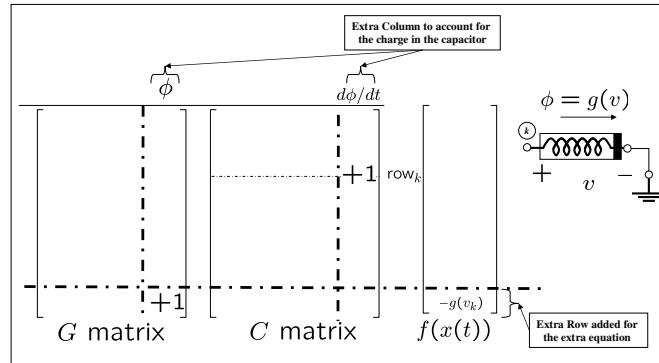


Figure 1.22: Stamp of a nonlinear inductor connected between node k and the reference node.

1.1. MODIFIED NODAL ANALYSIS

Chapter **2**

Harmonic Balance for Circuits with Single-Tone Excitation

This chapter introduces the Harmonic Balance (HB) approach for finding the steady-state response of circuits with single-tone excitations. The term “single-tone” excitation here refers to the fact that the stimulus excitation waveforms considered in this chapter contain a single angular frequency and integer multiples of it. It is well known that such a waveform can be represented by an infinite Fourier series expansion of the form,

$$u(t) = \sum_{k=-\infty}^{\infty} \mathcal{U}_k e^{jk\omega_0 t} \quad (2.1)$$

where ω_0 is known as the fundamental (angular) frequency, and \mathcal{U}_k is the coefficient of the k -th harmonic. The fundamental period T is the period of time for which $u(t) = u(t + T)$, and is given in terms of the fundamental angular frequency by $T = \frac{2\pi}{\omega_0}$. If $u(t)$ is a real signal, it is easy to show that $\mathcal{U}_m = \mathcal{U}_{-m}^*$, where $*$ represents the complex conjugate operation.

Another equivalent representation for single-tone periodic waveforms can be provided using sinusoidal waveforms as follows,

$$u(t) = U_0 + \sum_{k=1}^{\infty} U_k^C \cos(k\omega_0 t) + U_k^S \sin(k\omega_0 t) \quad (2.2)$$

where U_0 , U_k^C and U_k^S are all real coefficients. It is well-known that $U_k^C = 2\Re(\mathcal{U}_k)$, $U_k^S = -2\Im(\mathcal{U}_k)$, and $U_0 = \mathcal{U}_0$. Typically in finding the steady-state response of a gen-

eral circuit, one always assumes that all independent sources in the circuit exhibit a certain periodic waveform with a certain period, say T , and the main objective is then focused on computing the time-domain circuit response at steady-state, i.e., as $t \rightarrow \infty$, where by “response” here we mean node voltages, branch currents, capacitor charges, inductor fluxes or a combination thereof. Given that the excitation sources are periodic at steady-state, it is intuitive to expect that the response will also be periodic, with the same period, T . This basic knowledge about the general nature of the response at steady-state represents the main key idea in the HB technique and in other steady-state analysis approaches.

This chapter starts first by offering a brief background material in Section 2.1 on the Fourier transformation techniques used to convert a periodic signal, such as $u(t)$, from the time-domain to its Fourier domain representation, and vice-versa. The notation and mathematical operations laid out in this section are central to the derivation of the HB approach in the following sections.

2.1 Background on Fourier Transform for Single-Tone Signals

Let $x(t)$ ba an arbitrary periodic signal with period T having a Fourier series representation similar to (2.2). To make the process of computing its Fourier series representation tractable, we will have to truncate the infinite series to a limited number of terms, say K terms, and rewrite (2.2) as follows

$$x(t) = X_0 + \sum_{k=1}^K X_k^C \cos(k\omega_0 t) + X_k^S \sin(k\omega_0 t) \quad (2.3)$$

The above truncation implicitly assumes that the effect of coefficients corresponding to terms of order higher than K are negligibly small, and would therefore add no further accuracy should $x(t)$ be reconstructed using the Fourier series expansion. The value of integer K deemed sufficient for a good approximation usually depends on the shape of the waveform $x(t)$. For example, in the extreme case where $x(t)$ is a smooth cosinusoid, we would have $K = 1$ with $X^0 = X_m^C = X_m^S = 0$ for $m > 1$, while, $X_1^C = 1$ and $X_1^S = 0$. On the other hand, if $x(t)$ is a square-wave with sharp bends and turning points, K will range from few hundreds to few thousands of terms.

Assume that $x(t)$ is sampled at equally spaced H time points, where $H = 2K + 1$, at the time instants t_0, t_1, \dots, t_{2K} in the interval $[0, T]$ as shown in Figure 2.1, with $t_n = n\frac{T}{H}, n = 0, \dots, 2K$. Generally, there are two main techniques that are typically used to compute the Fourier Coefficients for a periodic waveform given its time-domain samples (or vice versa, obtain the time-domain samples from a given Fourier series representation). We start with most basic one of them, the Discrete Fourier Transform , DFT (or its inverse known as Inverse Fourier Transform, the IDFT) in the following subsection.

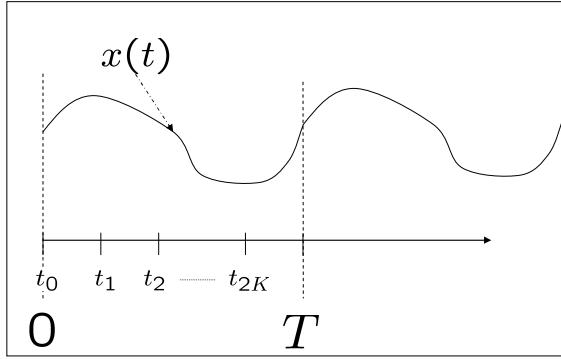


Figure 2.1: A periodic waveform sampled at $2K + 1$ time points.

2.1.1 Discrete Fourier Transformation

Writing the Fourier series at an arbitrary time instant t_i gives us,

$$x(t_i) = X_0 + \sum_{k=1}^K X_k^C \cos(k\omega_0 t_i) + X_k^S \sin(k\omega_0 t_i) \quad (2.4)$$

Repeating the above equation for all time points enables us to write the following matrix equations,

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{2K} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \cos(\omega_0 t_0) & \sin(\omega_0 t_0) & \cdots & \cos(K\omega_0 t_0) & \sin(K\omega_0 t_0) \\ 1 & \cos(\omega_0 t_1) & \sin(\omega_0 t_1) & \cdots & \cos(K\omega_0 t_1) & \sin(K\omega_0 t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\omega_0 t_{2K}) & \sin(\omega_0 t_{2K}) & \cdots & \cos(K\omega_0 t_{2K}) & \sin(K\omega_0 t_{2K}) \end{bmatrix}}_{\Gamma^{-1}} \begin{bmatrix} X_0 \\ X_1^C \\ X_1^S \\ \vdots \\ X_K^C \\ X_K^S \end{bmatrix} \quad (2.5)$$

where $x_i = x(t_i)$. Eq. (2.5) can be represented more compactly as follows,

$$\mathbf{x} = \Gamma^{-1} \mathbf{X} \quad (2.6)$$

where $\mathbf{x} = [x_0, \dots, x_{2K}]^T$ and $\mathbf{X} = [X_0, X_1^C, X_1^S, \dots, X_K^C, X_K^S]^T$, and Γ^{-1} is the matrix shown in Eq. (2.5).

A closer inspection of the elements in the matrix Γ^{-1} reveals that they are independent of the fundamental angular frequency ω_0 as can be seen from the argument of any co- or

sinusoidal function,

$$k\omega_0 t_i = k \left(\frac{2\pi}{T} \right) \left(i \frac{T}{H} \right) = ki \frac{2\pi}{H}, \quad (1 \leq k \leq K, 0 \leq i \leq H-1) \quad (2.7)$$

Thus the only condition to make Γ^{-1} independent of the fundamental angular frequency is to have the time samples equally spaced in the period $[0, T]$. In that sense, we can see that the matrix Γ^{-1} acts as an operator that converts a vector of $2K + 1$ Fourier-domain coefficients to its time-domain representation. The essence of the operation that needs to be performed is a matrix-vector multiplication and results in $H = 2K + 1$ time-domain samples of $x(t)$ at equally spaced time points. The matrix Γ^{-1} is therefore termed the Inverse Discrete Fourier Transformation (IDFT) operator.

Let Γ denote the matrix inverse of Γ^{-1} , assuming it is nonsingular. Then Eq. (2.6) can be written as

$$\mathbf{X} = \Gamma \mathbf{x} \quad (2.8)$$

Hence, the matrix Γ functions as an operator that, when presented with a vector of time-domain samples of a periodic waveform, produces the corresponding coefficients of its harmonic content. Based on this notion, Γ is termed the Discrete Fourier Transformation (DFT) operator.

The IDFT operator matrix Γ^{-1} has another nice property; its columns are orthogonal and have a norm equal to $H/2$, except for the first column which has a norm equal to H . Thus if γ_m and γ_n are two columns in Γ^{-1} , then their inner-product is given by

$$\gamma_m^T \gamma_n = \begin{cases} 0 & m \neq n \\ \frac{H}{2} & m = n \neq 1 \\ H & m = n = 1 \end{cases} \quad (2.9)$$

This property makes the matrix product $(\Gamma^{-1})^T \Gamma^{-1}$ a diagonal matrix,

$$(\Gamma^{-1})^T \Gamma^{-1} = \begin{bmatrix} H & 0 & \cdots & 0 \\ 0 & H/2 & 0 & \cdots & 0 \\ \vdots & \ddots & \cdots & & 0 \\ 0 & \cdots & & & H/2 \end{bmatrix} \quad (2.10)$$

Post-multiplying both sides of (2.10) by the DFT operator matrix Γ shows that it is possible to write the rows of Γ in terms of the columns of Γ^{-1} . If the row vectors of Γ are denoted

by $\rho_1, \rho_2, \dots, \rho_H$, then

$$\rho_m = \begin{cases} \frac{1}{H} \gamma_m^T & m = 1 \\ \frac{2}{H} \gamma_m^T & m = 2, 3, \dots, H \end{cases} \quad (2.11)$$

That shows that there is no need to perform the cumbersome matrix inversion to obtain Γ .

2.1.2 Fast Fourier Transformation

As can be seen from Eq.(2.5) the matrices Γ^{-1} and Γ are full matrices with dimensions $H \times H$, where H is the number of time-domain sampling points. It is well-known that multiplying an $H \times H$ matrix by an $H \times 1$ vector requires performing H^2 multiplications and H^2 additions. This fact is expressed by stating that the computational complexity associated with a DFT (or IDFT) is of the order of H^2 , or $O(H^2)$. The second order growth of the computational complexity with the number of time-domain samples can represent a significant burden on the computational resources for a large number of time samples, especially if the DFT (IDFT) operation has to be performed repeatedly, such as in the course of HB analysis as will be shown in this chapter.

Nevertheless, the very structures of the matrices Γ^{-1} and Γ offer a special advantage that can be used to reduce the computational complexity. In fact, the characterization of this structure led to an efficient implementation of the DFT (IDFT) known as Fast Fourier Transform or FFT (or IFFT for the inverse implementation). A complete description of the FFT (IFFT) is beyond the scope of this text. However, it would be sufficient to state some basic facts about the FFT (or IFFT). The most important fact is related to the computational complexity which is equivalent to $O(H \log_2 H)$ if the number of time samples, H , is a power of 2. Other implementations can also be performed in $O(H \log H)$ for all H . The reader is referred to the link in [1] for a C library that implements the state-of-the-art in FFT. In fact, this is the version upon which Matlab functions `fft` and `ifft` are built. For the purpose of this course, we use the DFT (IDFT) for its convenience in the mathematical description of the HB analysis, where it will be implicitly understood that the FFT (IFFT) can be used in the actual implementation.

It is worth noting that the above discussion implicitly assumed that the number of time-domain samples is always odd ($H = 2K + 1$). Yet, this should by no means be always the case, because this number can be made even (or order of 2) by including the cosinusoidal coefficient of the next high order while neglecting the associated sinusoidal coefficient. In this case the Fourier expansion takes the following unbalanced form

$$x(t) = X_0 + \sum_{k=1}^K (X_k^C \cos(k\omega_0 t) + X_k^S \sin(k\omega_0 t)) + X_{K+1}^C \cos((K+1)\omega_0 t) \quad (2.12)$$

2.1. BACKGROUND ON FOURIER TRANSFORM FOR SINGLE-TONE SIGNALS

which yields an even number of time-domain samples ($2K + 2$).

2.1.3 Numerical Example

We consider here a simple numerical example to illustrate the usage of FFT and DFT techniques. For that purpose, we consider a simple sinusoidal signal given by,

$$x(t) = \sin(10t) \quad (2.13)$$

The Fourier coefficients of this signal are trivial and can be written as follows,

$$X_1^S = 1, X_1^C = 0, X_m^C = X_m^S = X_0 = 0 \quad \text{for } m > 1 \quad (2.14)$$

Another illustrative example can be obtained by considering a square wave with 50% duty cycle. Program 1 is a Matlab script that generates a square wave and computes its Fourier series coefficients using `fft` function. Also note that because Matlab `fft` function uses the complex form of the Fourier transformation (i.e., (2.1)), some additional steps had to be done to obtain the corresponding real form.

Program 1 A Matlab script that generates a square-wave and compute its Fourier series.

```
clear all, close all
nTimePoints = 512;
T = 10;
Time = linspace(0, T-T/nTimePoints, nTimePoints);
x = 2*((Time <= T/2)-0.5)';
plot(Time, x)
% Use fft to find the Fourier Coefficients
Xscaled = fft(x);
% rescale the Fourier coefficients
X = Xscaled/nTimePoints';
% Extract the DC coefficient
XDC = X(1);
% Extract the cos and sin coefficients
if (rem(nTimePoints, 2) == 0)
    % Even number of points
    Xc = 2*real(X([2:nTimePoints/2]));
    Xs = -2*imag(X([2:nTimePoints/2]));
    figure, semilogy([1:(nTimePoints/2-1)], abs(Xc), 'o', ...
                    [1:(nTimePoints/2-1)], abs(Xs), '*');
    grid, xlabel('k Harmonic number')
else
    % Odd number of points
    Xc = 2*real(X([2:(nTimePoints-1)/2+1]));
    Xs = 2*imag(X([2:(nTimePoints-1)/2+1]));
end
```

2.2 Harmonic Balance for Linear Circuits

Although the HB technique is mainly used for steady-state analysis of nonlinear circuits, this section considers its application for linear circuits to highlight the basic ideas that will be used extensively in nonlinear circuits.

2.2.1 Steady-state response for a simple circuit

Consider the simple circuit shown in Figure 2.2. The independent source in this circuit is a voltage-source whose voltage is a co-sinusoid signal having an amplitude of 2V and an angular frequency of 10rad/Sec. We now seek to find the steady-state response of this circuit, i.e., the voltages at nodes 1 and 2, and the currents through resistors R_1 and R_2 , as $t \rightarrow \infty$.

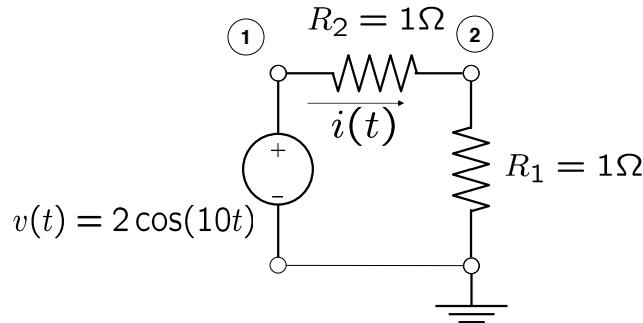


Figure 2.2: A simple resistive circuit with a periodic (co-sinusoid) stimulus.

This problem can be solved easily by noting that the circuit functions mainly as a voltage divider, where

$$v_1(t) = 2 \cos(10t) \quad \text{V} \quad (2.15)$$

$$v_2(t) = \frac{R_1}{R_1 + R_2} \times 2 \cos(10t) = \cos(10t) \quad \text{V} \quad (2.16)$$

2.2. HARMONIC BALANCE FOR LINEAR CIRCUITS

and the current through R_1 and R_2 is given by

$$i(t) = \cos(10t) \quad \text{A} \quad (2.17)$$

The responses obtained above are obviously the steady-state responses, since they will maintain the above functional forms no matter how time t grows large (i.e., as $t \rightarrow \infty$). It is needless to say that finding the steady-response of the above circuit was easy due to the extreme simplicity of the circuit. However, in more complicated circuits, this manual approach is unlikely to be affordable, and one will eventually have to resort to computer-based techniques to find the steady-state response. Yet, despite the simplicity of the above circuit, it provides us with a chance to draw a certain conclusion that will prove useful when tackling more general linear circuits. The main conclusion that we wish to highlight here is that the response of a *linear* circuit contains harmonics at **the same** angular frequencies as that of the driving input source. This concept is illustrated in Figure 2.3.

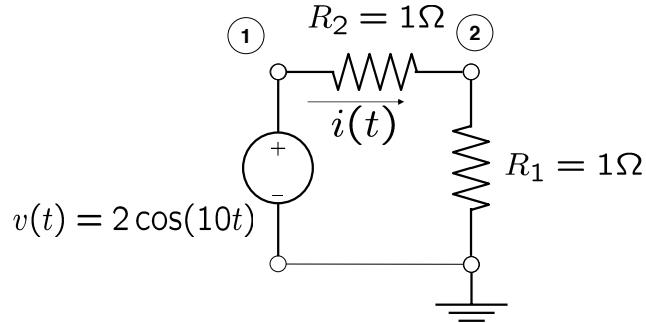


Figure 2.3: A linear circuit responds to a periodic input with a periodic response that has the same frequency components at that of the input.

2.2.2 Steady-state response for general linear circuits using HB

As discussed in the previous chapter, any general linear circuit can be described in the time-domain using the MNA formulation as follows,

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} = \mathbf{b}(t) \quad (2.18)$$

Assuming that all independent sources in the circuit are periodic with a period T , then all components of the vector $\mathbf{b}(t)$ are periodic functions with period T , and therefore can be represented by a Fourier series expansion. This means that the vector $\mathbf{b}(t)$ can also be represented by a Fourier Series expansion as shown next,

$$\mathbf{b}(t) = \mathbf{B}_0 + \sum_{k=1}^K \mathbf{B}_k^C \cos(k\omega_0 t) + \mathbf{B}_k^S \sin(\omega_0 t) \quad (2.19)$$

using sufficiently large integer K , where $\omega_0 = \frac{2\pi}{T}$ and $\mathbf{B}_0, \mathbf{B}_k^C, \mathbf{B}_k^S$ are vectors collecting the k -th Fourier series coefficients of each independent source.

Using the conclusion drawn from the simple example of the previous subsection, we know that the response vector $\mathbf{x}(t)$ will also be periodic with the same period having the harmonic components at the same frequencies as that in the driving input sources. Hence $\mathbf{x}(t)$ can be represented by a Fourier series expansion similar to that in Eq. (2.2),

$$\mathbf{x}(t) = \mathbf{X}_0 + \sum_{k=0}^K \mathbf{X}_k^C \cos(k\omega_0 t) + \mathbf{X}_k^S \sin(k\omega_0 t) \quad (2.20)$$

However, the Fourier vector coefficients $\mathbf{X}_0, \mathbf{X}_k^C, \mathbf{X}_k^S$ are not yet known, and we seek to compute them in order to construct the steady-state response. To proceed with that objective, we substitute from (2.20) and (2.19) into (2.18) to obtain, after taking the derivative,

$$\begin{aligned} & \mathbf{G} \left(\mathbf{X}_0 + \sum_{k=1}^K \mathbf{X}_k^C \cos(k\omega_0 t) + \mathbf{X}_k^S \sin(k\omega_0 t) \right) \\ & + \mathbf{C} \left(\sum_{k=1}^K -k\omega_0 \mathbf{X}_k^C \sin(k\omega_0 t) + k\omega_0 \mathbf{X}_k^S \cos(k\omega_0 t) \right) \\ & = \mathbf{B}_0 + \sum_{k=1}^K \mathbf{B}_k^C \cos(k\omega_0 t) + \mathbf{B}_k^S \sin(k\omega_0 t) \end{aligned} \quad (2.21)$$

2.2. HARMONIC BALANCE FOR LINEAR CIRCUITS

Note that in Eq. (2.21), \mathbf{B}_0 , \mathbf{B}_k^C and \mathbf{B}_k^S are known, since the waveforms driving the circuit are periodic functions and we can compute their Fourier coefficients using Fourier transform.

Solving for \mathbf{X}_0 , \mathbf{X}_k^C , \mathbf{X}_k^S in (2.21) is performed by taking advantage of the fact that the functions $\cos(k\omega_0 t)$, $\sin(k\omega_0 t)$ (where k is an integer) are orthonormal in the period $0 \leq t \leq T$. The orthonormality property is usually expressed by the following integrals

$$\frac{2}{T} \int_0^T \cos(m\omega_0 t) \sin(n\omega_0 t) dt = 0 \quad (2.22)$$

$$\frac{2}{T} \int_0^T \cos(m\omega_0 t) \cos(n\omega_0 t) dt = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases} \quad (2.23)$$

$$\frac{2}{T} \int_0^T \sin(m\omega_0 t) \sin(n\omega_0 t) dt = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases} \quad (2.24)$$

To take advantage of the property in (2.22), we multiply both sides of (2.21) by $\cos(p\omega_0 t)$ (p is an integer), integrate from 0 to T and then multiply by $2/T$. This process will yield the following system of equations

$$\mathbf{G}\mathbf{X}_p^C + p\omega_0 \mathbf{C}\mathbf{X}_p^S = \mathbf{B}_p^C \quad (2.25)$$

We repeat the above process again, but using $\sin(p\omega_0 t)$ instead of $\cos(p\omega_0 t)$ to obtain

$$\mathbf{G}\mathbf{X}_p^S - p\omega_0 \mathbf{C}\mathbf{X}_p^C = \mathbf{B}_p^S \quad (2.26)$$

The two systems in (2.25) and (2.26) can be combined in a single system of equations as follows,

$$\begin{bmatrix} \mathbf{G} & p\omega_0 \mathbf{C} \\ -p\omega_0 \mathbf{C} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{X}_p^C \\ \mathbf{X}_p^S \end{bmatrix} = \begin{bmatrix} \mathbf{B}_p^C \\ \mathbf{B}_p^S \end{bmatrix} \quad (2.27)$$

Since there was no restriction on the integer p used to reach the system in (2.27), this index can run over values from 1 to K , thus producing K systems of equations similar to (2.27), and whose solutions yield the values of the unknown vectors \mathbf{X}_p^C and \mathbf{X}_p^S for $1 \leq p \leq K$. Obtaining the DC coefficients vector \mathbf{X}_0 can be achieved by simply integrating the system (2.21) from 0 to T , and dividing by T to obtain the following system,

$$\mathbf{G}\mathbf{X}_0 = \mathbf{B}_0 \quad (2.28)$$

whose solution provides the DC coefficients vector.

2.2.3 Matrix-Vector formulation for HB of Linear Circuits

It is possible to cast the HB problem for general linear circuits in a compact matrix form by collecting all of \mathbf{X}_m^C , \mathbf{X}_m^S , \mathbf{X}_0 in a single vector and denote this vector by $\bar{\mathbf{X}}$, i.e.,

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1^C \\ \mathbf{X}_1^S \\ \vdots \\ \mathbf{X}_K^C \\ \mathbf{X}_K^S \end{bmatrix} \quad (2.29)$$

where $\bar{\mathbf{X}} \in \mathbb{R}^{N \times (2M+1)}$ and N is the total number of variables in the MNA formulation.

The HB problem for linear circuit , which was described by (2.27) and (2.28) can be made in the following form

$$\bar{\mathbf{Y}} \bar{\mathbf{X}} = \bar{\mathbf{B}} \quad (2.30)$$

where the matrix $\bar{\mathbf{Y}} \in \mathbb{R}^{N(2K+1) \times N(2K+1)}$ is a block diagonal matrix given by,

$$\bar{\mathbf{Y}} = \begin{bmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G} & \omega_0 \mathbf{C} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & -\omega_0 \mathbf{C} & \mathbf{G} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G} & 2\omega_0 \mathbf{C} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -2\omega_0 \mathbf{C} & \mathbf{G} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{G} & K\omega_0 \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & -K\omega_0 \mathbf{C} & \mathbf{G} & \mathbf{0} \end{bmatrix} \quad (2.31)$$

and the vector $\bar{\mathbf{B}} \in \mathbb{R}^{N(2K+1)}$ is given by

$$\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1^C \\ \mathbf{B}_1^S \\ \vdots \\ \mathbf{B}_K^C \\ \mathbf{B}_K^S \end{bmatrix} \quad (2.32)$$

One can also represent the matrix $\bar{\mathbf{Y}}$ more compactly by using the Kronecker product

2.3. HARMONIC BALANCE FOR NONLINEAR CIRCUITS

operator \otimes (See Appendix A) as follows,

$$\bar{\mathbf{Y}} = \begin{bmatrix} \mathbf{G} & \bar{\mathbf{O}} \\ \bar{\mathbf{O}}^T & \sum_{m=1}^K \mathbf{L}_{m,K} \otimes (\mathbf{I}_2 \otimes \mathbf{G} + m\omega_0 \mathbf{T}_2 \otimes \mathbf{C}) \end{bmatrix} \quad (2.33)$$

where

- \mathbf{I}_2 is an identity matrix of size 2×2 ,
- \mathbf{T}_2 is a matrix given by

$$\mathbf{T}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (2.34)$$

- $\bar{\mathbf{O}}$ is matrix of size $N \times 2KN$ with all zero elements,
- $\mathbf{L}_{m,K} = \mathbf{e}_{m,K} \mathbf{e}_{m,K}^T$, and
- $\mathbf{e}_{m,K}$ is the m^{th} column in an identity matrix of size $K \times K$,
- $\mathbf{e}_{m,K}^T$ is the m^{th} row in an identity matrix of size $K \times K$.

2.3 Harmonic Balance for Nonlinear Circuits

We start the derivation of the HB approach for nonlinear circuits by considering the simple illustrative example shown in Figure 2.4.

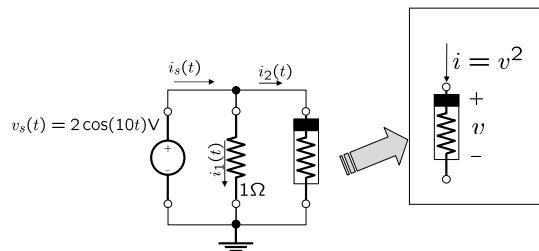


Figure 2.4: A simple nonlinear circuit with periodic excitation.

Assume that our objective in this circuit is to compute the steady-state waveform for the

current in the independent voltage source, $i_s(t)$. It is obvious that this current is given by

$$\begin{aligned}
 i_s(t) &= i_1(t) + i_2(t) \\
 &= \frac{v_s(t)}{1} + (v_s(t))^2 \\
 &= 2 \cos(10t) + (2 \cos(10t))^2 \\
 &= 2 \cos(10t) + 4 (\cos(10t))^2
 \end{aligned} \tag{2.35}$$

Noting that $\cos^2(\theta) = \frac{1}{2} (\cos(2\theta) + 1)$, then $i_s(t)$ in *steady-state* will be given by

$$i_s(t) = 3 + 2 \cos(10t) + 2 \cos(20t) \tag{2.36}$$

The above outcome for the steady-state of $i_s(t)$ allows us to draw the central conclusion pertinent to the steady-state response in nonlinear circuits. In this conclusion we note that, although the input signal contains only one harmonic component at frequency $\omega_0 = 10\text{rad/sec}$, the steady-state response comes at the same and twice the excitation frequency of the input (i.e., at $\omega = 20\text{rad/sec}$.).

The above conclusion can also be further verified by considering another simple nonlinear circuit, but this time with a cubic nonlinearity, as shown in Figure 2.5. In this case the

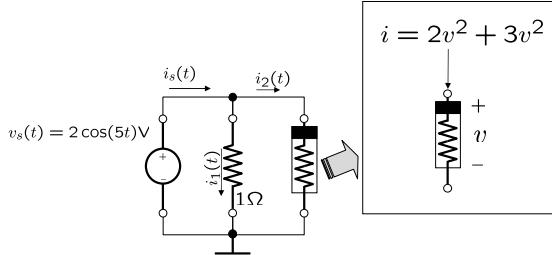


Figure 2.5: A nonlinear circuit with cubic nonlinearity.

steady-state response for $i_s(t)$ will be given by,

$$i_s(t) = 1 + \frac{17}{8} \cos(5t) + \cos(10t) + \frac{3}{8} \cos(15t) \tag{2.37}$$

As can be observed from the above expression, the steady-state current (response) contains harmonic components at multiples of the excitation frequency up to order 3, i.e. at $\omega = 5, 10, 15\text{rad/sec}$.

It is easy to see that for circuits with higher order nonlinearity, the response will contain frequency components that are at higher multiples of the excitation frequency. It is also fair to conclude that in the case of general nonlinear functions, such as the exponential function

2.3. HARMONIC BALANCE FOR NONLINEAR CIRCUITS

of the diode, which can be expanded as infinite Taylor series,

$$\sum_{i=0}^{\infty} a_i v^i \quad (2.38)$$

the response will have an infinite multiples of the excitation frequency.

The case of having a steady-state response containing infinite multiples of the excitation frequency is not restricted to having general nonlinearity with infinite order power expansion. It also arises in circuit with finite-order nonlinearity provided there is feedback path. The circuit shown in Figure 2.6 is an example that illustrates this fact. The presence of a feedback resistor produces a voltage drop proportional to the current, which will ultimately influence the voltage applied across the nonlinearity.

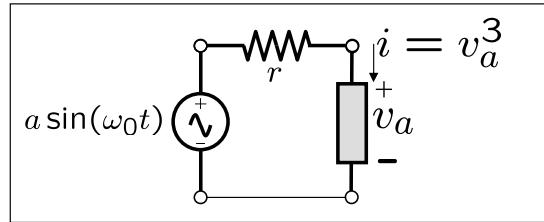


Figure 2.6: A nonlinear circuit with feedback.

Hence, it is safe to conclude that in generic nonlinear circuits excited by a single-tone periodic input, the response will be periodic with the same period of the input. In addition, it will generally have an infinite multiples of the excitation frequency, and therefore can be represented using an infinite Fourier expansion of the form shown in (2.1) or (2.2).

2.3.1 General nonlinear circuits

Let us put the conclusion drawn from the previous example to work in a general nonlinear circuit with periodic excitation. From our discussion in Chapter 1, we know that any general nonlinear circuit can be represented in the time-domain using the MNA formulation as follows,

$$G\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b}(t) \quad (2.39)$$

We assume that all source excitations in $\mathbf{b}(t)$ are periodic having a period T (or angular frequency $\omega_0 = \frac{2\pi}{T}$), with a Fourier series expansion of K harmonics as given by Eq. (2.19).

Using the conclusion established above, the response vector $\mathbf{x}(t)$ will have to be repre-

sented by an infinite Fourier series of the form

$$\mathbf{x}(t) = \mathbf{X}_0 + \sum_{m=1}^{\infty} \mathbf{X}_m^C \cos(m\omega_0 t) + \mathbf{X}_m^S \sin(m\omega_0 t) \quad (2.40)$$

However, to provide practical computations, we will have to truncate the series expansion to a reasonably large number M ,

$$\mathbf{x}(t) = \mathbf{X}_0 + \sum_{m=1}^M \mathbf{X}_m^C \cos(m\omega_0 t) + \mathbf{X}_m^S \sin(m\omega_0 t) \quad (2.41)$$

In truncating the infinite spectrum response, we are basically assuming that terms of order higher than M are very negligible to add noticeable accuracy to the steady-state response. The issue of whether M is enough to ensure accuracy will be further elaborated on later in this chapter. Note here that the number of harmonics in the response vector, M , is typically larger than those in the input, K , to reflect the fact that the circuit responds at multiples of *each* excitation frequency in the input.

To compute the Fourier coefficients in the response vector, $\mathbf{X}_m^{C,S}$, we substitute from (2.41) and (2.19) into (2.39) to obtain

$$\begin{aligned} & \mathbf{G} \left(\mathbf{X}_0 + \sum_{m=1}^M \mathbf{X}_m^C \cos(m\omega_0 t) + \mathbf{X}_m^S \sin(m\omega_0 t) \right) \\ & + \mathbf{C} \left(\sum_{m=1}^M -m\omega_0 \mathbf{X}_m^C \sin(m\omega_0 t) + m\omega_0 \mathbf{X}_m^S \cos(m\omega_0 t) \right) \\ & + \mathbf{f} \left(\mathbf{X}_0 + \sum_{m=1}^M \mathbf{X}_m^C \cos(m\omega_0 t) + \mathbf{X}_m^S \sin(m\omega_0 t) \right) \\ & = \mathbf{B}_0 + \sum_{k=1}^K \mathbf{B}_k^C \cos(k\omega_0 t) + \mathbf{B}_k^S \sin(k\omega_0 t) \end{aligned} \quad (2.42)$$

We need to focus first on the nonlinear term

$$\mathbf{f} \left(\mathbf{X}_0 + \sum_{m=1}^M \mathbf{X}_m^C \cos(m\omega_0 t) + \mathbf{X}_m^S \sin(m\omega_0 t) \right).$$

We note that any nonlinear function whose argument is periodic w.r.t. some parameter,

2.3. HARMONIC BALANCE FOR NONLINEAR CIRCUITS

say t , will also be periodic w.r.t. the same parameter t . A notable example is a scalar quadratic function given by $f(x(t)) = (x(t))^2$; if $x(t)$ happened to be periodic in t with a period T , then $f(x(t))$ will also be periodic in t repeating itself every T . With that notion in mind, we can see that every component in the nonlinear function vector, $\mathbf{f}(\mathbf{x}(t))$, will be periodic due to the fact that every component in the vector $\mathbf{x}(t)$ is periodic. To account for general nonlinear functions with arbitrary high-order nonlinearity, $\mathbf{f}(\mathbf{x}(t))$ is expanded as infinite Fourier series,

$$\mathbf{f}(\mathbf{x}(t)) = \mathbf{F}_0 + \sum_{m=1}^{\infty} \mathbf{F}_m^C \cos(m\omega_0 t) + \mathbf{F}_m^S \sin(m\omega_0 t) \quad (2.43)$$

where $\mathbf{F}_m^C, \mathbf{F}_m^S, \mathbf{F}_0$ are vectors that contain the m -th Fourier coefficients of the nonlinear functions in $\mathbf{f}(\mathbf{x}(t))$. Since $\mathbf{F}_m^C, \mathbf{F}_m^S, \mathbf{F}_0$ depend on the shape of the time-domain periodic waveform of each component in $\mathbf{f}(\mathbf{x}(t))$, and given that this shape is dictated directly by the time-domain waveforms of the components in $\mathbf{x}(t)$, whose waveforms in turn are determined by the value of all $\mathbf{X}_m^C, \mathbf{X}_m^S, \mathbf{X}_0$, then we may conclude that each one of the $\mathbf{F}_m^C, \mathbf{F}_m^S, \mathbf{F}_0$ is ultimately a function of *all* of the $\mathbf{X}_m^C, \mathbf{X}_m^S, \mathbf{X}_0$. Figure 2.7 illustrates this dependence cycle graphically.

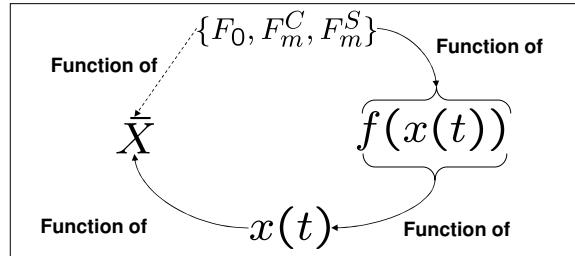


Figure 2.7: Illustrating the dependence of \mathbf{F}_m on all of \mathbf{X}_m .

To emphasize the notion that each \mathbf{F}_m is a function of all of \mathbf{X}_m 's, we rewrite the Fourier expression in (2.43) as follows,

$$\mathbf{f}(\mathbf{x}(t)) = \mathbf{F}_0(\bar{\mathbf{X}}) + \sum_{m=1}^{\infty} \mathbf{F}_m^C(\bar{\mathbf{X}}) \cos(m\omega_0 t) + \mathbf{F}_m^S(\bar{\mathbf{X}}) \sin(m\omega_0 t) \quad (2.44)$$

where we express $\mathbf{F}_0, \mathbf{F}_m^C, \mathbf{F}_m^S$ as explicit functions of $\bar{\mathbf{X}}$. Substituting from (2.44) into

(2.42) we obtain

$$\begin{aligned}
 & \mathbf{G} \left(\mathbf{X}_0 + \sum_{m=1}^M \mathbf{X}_m^C \cos(m\omega_0 t) + \mathbf{X}_m^S \sin(m\omega_0 t) \right) \\
 & + \mathbf{C} \left(\sum_{m=1}^M -m\omega_0 \mathbf{X}_m^C \sin(m\omega_0 t) + m\omega_0 \mathbf{X}_m^S \cos(m\omega_0 t) \right) \\
 & + \mathbf{F}_0(\bar{\mathbf{X}}) + \sum_{m=1}^{\infty} \mathbf{F}_m^C(\bar{\mathbf{X}}) \cos(m\omega_0 t) + \mathbf{F}_m^S(\bar{\mathbf{X}}) \sin(m\omega_0 t) \\
 & = \mathbf{B}_0 + \sum_{k=1}^K \mathbf{B}_k^C \cos(k\omega_0 t) + \mathbf{B}_k^S \sin(k\omega_0 t) \quad (2.45)
 \end{aligned}$$

Using the orthonormality property expressed by Eqs. (2.22)-(2.24), through multiplying both sides by $\cos(p\omega_0 t)$ (p is an arbitrary integer), integrating from 0 to T , and multiplying by $2/T$ we obtain,

$$\mathbf{G}\mathbf{X}_p^C + p\omega_0 \mathbf{C}\mathbf{X}_p^S + \mathbf{F}_p^C(\bar{\mathbf{X}}) = \mathbf{B}_p^C \quad (2.46)$$

If the above process is repeated using $\sin(p\omega_0 t)$ we obtain,

$$\mathbf{G}\mathbf{X}_p^S - p\omega_0 \mathbf{C}\mathbf{X}_p^C + \mathbf{F}_p^S(\bar{\mathbf{X}}) = \mathbf{B}_p^S \quad (2.47)$$

Since the choice of p was arbitrary, we can let p take indices ranging over $\{1, 2, \dots, \infty\}$ and thereby yielding an infinite number of systems. An additional system can be obtained by integrating both sides of (2.45) from 0 to T and dividing by T , yielding,

$$\mathbf{G}\mathbf{X}_0 + \mathbf{F}_0(\bar{\mathbf{X}}) = \mathbf{B}_0 \quad (2.48)$$

The systems of equations in (2.48) and (2.46), (2.47) for values of $\{p = 1, 2, \dots, \infty\}$

can be written in the following form

$$\begin{aligned}
 \mathbf{G}\mathbf{X}_0 + \mathbf{F}_0(\bar{\mathbf{X}}) &= \mathbf{B}_0 \\
 \mathbf{G}\mathbf{X}_1^C + \omega_0 \mathbf{C}\mathbf{X}_1^S + \mathbf{F}_1^C(\bar{\mathbf{X}}) &= \mathbf{B}_1^C \\
 -\omega_0 \mathbf{C}\mathbf{X}_1^C + \mathbf{G}\mathbf{X}_1^S + \mathbf{F}_1^S(\bar{\mathbf{X}}) &= \mathbf{B}_1^S \\
 &\vdots = \vdots \\
 \mathbf{G}\mathbf{X}_K^C + K\omega_0 \mathbf{C}\mathbf{X}_K^S + \mathbf{F}_K^C(\bar{\mathbf{X}}) &= \mathbf{B}_K^C \\
 -K\omega_0 \mathbf{C}\mathbf{X}_K^C + \mathbf{G}\mathbf{X}_K^S + \mathbf{F}_K^S(\bar{\mathbf{X}}) &= \mathbf{B}_K^S \\
 \mathbf{G}\mathbf{X}_{K+1}^C + (K+1)\omega_0 \mathbf{C}\mathbf{X}_{K+1}^S + \mathbf{F}_{K+1}^C(\bar{\mathbf{X}}) &= \mathbf{0} \\
 -(K+1)\omega_0 \mathbf{C}\mathbf{X}_{K+1}^C + \mathbf{G}\mathbf{X}_{K+1}^S + \mathbf{F}_{K+1}^S(\bar{\mathbf{X}}) &= \mathbf{0} \\
 &\vdots = \vdots \\
 \mathbf{G}\mathbf{X}_M^C + M\omega_0 \mathbf{C}\mathbf{X}_M^S + \mathbf{F}_M^C(\bar{\mathbf{X}}) &= \mathbf{0} \\
 -M\omega_0 \mathbf{C}\mathbf{X}_M^C + \mathbf{G}\mathbf{X}_M^S + \mathbf{F}_M^S(\bar{\mathbf{X}}) &= \mathbf{0} \\
 \mathbf{F}_{M+1}^C(\bar{\mathbf{X}}) &= \mathbf{0} \\
 \mathbf{F}_{M+1}^S(\bar{\mathbf{X}}) &= \mathbf{0} \\
 &\vdots = \vdots
 \end{aligned} \tag{2.49}$$

Note that the size of each subsystem in (2.49) is N , where N is assumed to be the total number of variables in the MNA formulation. The number of equations in (2.49) is much larger than the number of unknowns. To facilitate solving for the $2M + 1$ unknowns $\mathbf{X}_0, \mathbf{X}_m^{(C,S)}, m = 1, \dots, M$, we will have to truncate the number of equations to $2M + 1$ equations. This is equivalent to truncating the spectral representation of the waveforms in $\mathbf{f}(\mathbf{x}(t))$ to only M terms,

$$\mathbf{f}(\mathbf{x}(t)) = \mathbf{F}_0(\bar{\mathbf{X}}) + \sum_{m=1}^M \mathbf{F}_m^C(\bar{\mathbf{X}}) \cos(m\omega_0 t) + \mathbf{F}_m^S(\bar{\mathbf{X}}) \sin(m\omega_0 t) \tag{2.50}$$

Thus $(2M + 1)$ systems result from truncating (2.49) to $2M + 1$. Those systems can be described in a more compact form as a large system whose size is $N \times (2M + 1)$ shown next,

$$\bar{\mathbf{Y}}\bar{\mathbf{X}} + \bar{\mathbf{F}}(\bar{\mathbf{X}}) = \bar{\mathbf{B}} \tag{2.51}$$

The vector $\bar{\mathbf{F}}(\bar{\mathbf{X}})$ is an $(N \times (2M + 1))$ vector of Fourier coefficients,

$$\bar{\mathbf{F}}(\bar{\mathbf{X}}) = \begin{bmatrix} \mathbf{F}_0(\bar{\mathbf{X}}) \\ \mathbf{F}_1^C(\bar{\mathbf{X}}) \\ \mathbf{F}_1^S(\bar{\mathbf{X}}) \\ \vdots \\ \mathbf{F}_M^C(\bar{\mathbf{X}}) \\ \mathbf{F}_M^S(\bar{\mathbf{X}}) \end{bmatrix} \quad (2.52)$$

The matrix $\bar{\mathbf{Y}} \in \mathbb{R}^{N(2M+1) \times N(2M+1)}$ is similar in structure to the one given in (3.34) but is extended to M frequency components.

It is obvious at this point that obtaining the Fourier coefficients of the response vector $\mathbf{x}(t)$ requires solving a **nonlinear** system of $N \times (2M + 1)$ equations. This is to be contrasted with the situation in linear circuits, where computing the response required solving $2M + 1$ **linear** systems of N equations. This fact makes using the HB more difficult in nonlinear circuits than in the linear ones. More specifically the presence of the nonlinear elements couples the different Fourier coefficients in one big system.

Solving a nonlinear system of equations such as the HB equations in (2.51), can only be done using iterative techniques. The simplest form of those techniques is based on the direct implementation of the Newton-Raphson (N-R) method. A brief illustration of the N-R method is given in Appendix A.

2.4 Solving the HB equations using N-R method

Let us now apply the N-R method in solving the HB equations in (2.51). In this context we seek a vector $\bar{\mathbf{X}}$ that satisfies (2.51), which may be rewritten as

$$\bar{\mathbf{Y}}\bar{\mathbf{X}} + \bar{\mathbf{F}}(\bar{\mathbf{X}}) - \bar{\mathbf{B}} = \mathbf{0} \quad (2.53)$$

Denoting the whole of the left-side of (2.53) by $\Phi(\bar{\mathbf{X}})$, we can see that the main steps needed in applying the N-R method is the computation of $\Phi(\bar{\mathbf{X}})$ and its Jacobian matrix at $\frac{\partial \Phi(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}}$ at the j -th trial vector $\bar{\mathbf{X}}^{(j)}$. This is the main focus in the rest of this section.

2.4.1 Computing $\Phi(\bar{\mathbf{X}}^{(j)})$

At the j -th trial vector, we have

$$\Phi(\bar{\mathbf{X}}^{(j)}) = \bar{\mathbf{Y}}\bar{\mathbf{X}}^{(j)} + \bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)}) - \bar{\mathbf{B}} \quad (2.54)$$

The first term on the right side can be computed easily using simple matrix-vector multiplication, which results in a single vector from which the vector $\bar{\mathbf{B}}$ can be subtracted. The main significant computational difficulty is actually encountered in computing the nonlinear term $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$. The absence of direct analytic expressions for $\bar{\mathbf{F}}$ in terms of $\bar{\mathbf{X}}$ will leave us with no other option but to use numerical computation schemes. We consider this computation next, where we summarize the basic ideas and then give a detailed description for a step-by-step process that results in computing $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$.

The starting point in computing $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ is assigning some values for the components of the j -th trial vector, $\bar{\mathbf{X}}^{(j)}$. Those values are typically taken from the values of the previous ($j - 1$) trial vector, or they could be based on some educated guess, if we happened to be at the very first trial vector, ($j = 0$). More on choosing the initial trial vector will be discussed later. Next, recall from Eq. (2.52) that $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ comprises the Fourier coefficients of the nonlinear functions in $\mathbf{f}(\mathbf{x}(t))$. Therefore, obtaining those Fourier coefficients requires the knowledge of the periodic time-domain waveform of each one of those components. However, obtaining the time-domain waveform for a given component in $\mathbf{f}(\mathbf{x}(t))$ requires computing the waveform of its arguments, which are typically some selected components of $\mathbf{x}(t)$. Thus we need to compute the time-domain waveform for each variable in the MNA formulation. Given that we have the Fourier coefficients at the j -th trial vector, we can make use of those Fourier coefficients to obtain the waveforms for each component in $\mathbf{x}(t)$. We can now describe a five-step process to compute $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ implementing the above procedural description but in the reverse order.

- **STEP 1. Preprocessing of the j -th trial vector.** This step is done in preparation for the following steps. In the next step, the steady-state periodic waveform for each variable in the MNA formulation is computed through IFFT. To perform the IFFT, we first need to collect the harmonics of each variable from $\bar{\mathbf{X}}^{(j)}$. As shown in Eq. (2.29), and more clearly in Figure 2.8, the harmonics of each variable are scattered in the vector $\bar{\mathbf{X}}^{(j)}$.

The ordering configuration shown in Figure 2.8 is known as **Harmonic-major/Node-minor** ordering in an indication to the fact the components are ordered according to their Harmonic index first and then according to their MNA index, where changes in

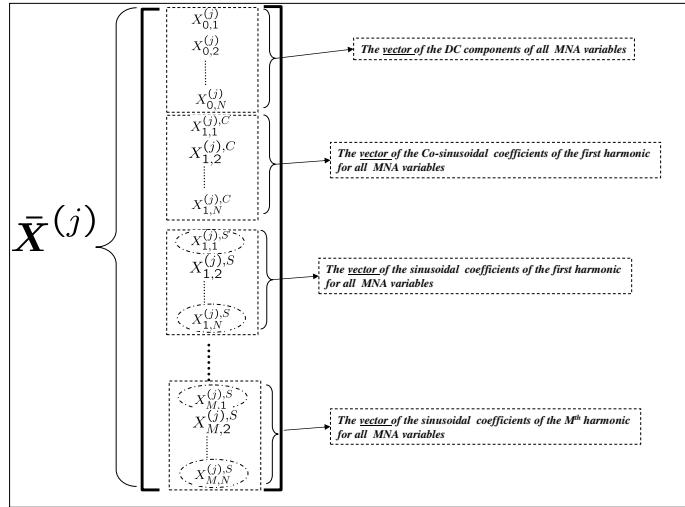


Figure 2.8: The initial ordering of the j -th trial vector $\bar{X}^{(j)}$.

the Harmonic index run slower than changes in the MNA variable index. Figure 2.9 illustrates these two indices.

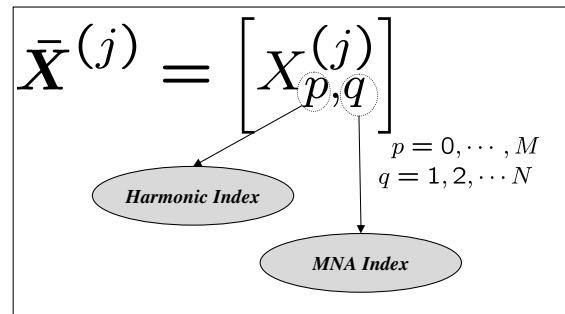


Figure 2.9: Illustrating the two main indices used to index the components of $\bar{X}^{(j)}$.

To collect the harmonics of each variable, we will need to reorder the components in $\bar{X}^{(j)}$ in Node-major/Harmonic-minor, as shown in Figure 2.10. We denote this new vector by $\bar{X}_{\text{node}}^{(j)}$ to emphasize the ordering mode. Vector reordering is usually implemented through swapping memory pointers. A simpler ordering procedure, which is more convenient for the sake of mathematical derivation, is based on the multiplication with a suitable permutation matrix. A permutation matrix is an orthonormal matrix with *at most* one entry equal to (1) in each row, and (0)'s in the rest of row's

2.4. SOLVING THE HB EQUATIONS USING N-R METHOD

entries. Assume that the required permutation matrix needed to obtain $\bar{\mathbf{X}}_{\text{node}}^{(j)}$ from $\bar{\mathbf{X}}^{(j)}$ is denoted by \mathcal{P} , then,

$$\bar{\mathbf{X}}_{\text{node}}^{(j)} = \mathcal{P} \bar{\mathbf{X}}^{(j)} \quad (2.55)$$

and by the orthonormality of \mathcal{P} we have

$$\mathcal{P}^T \mathcal{P} = \text{identity matrix} \quad (2.56)$$

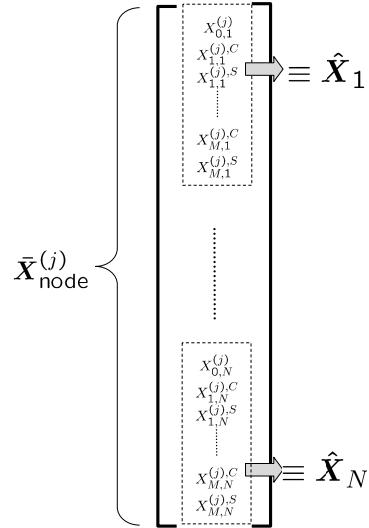


Figure 2.10: Reordering the components of $\bar{\mathbf{X}}^{(j)}$ in Node-major/Harmonic-minor.

- **STEP 2. Performing IFFT.** In this step the harmonics of every MNA variable is passed through IFFT to obtain the corresponding periodic time-domain waveform. From the discussion of Section 2, this process is equivalent to multiplying each one of the boxed subvectors (shown in Figure 2.10) of $\bar{\mathbf{X}}_{\text{node}}^{(j)}$ by the IDFT operator Γ^{-1} . The resulting time-domain waveforms are represented by $(2M + 1)$ sample time points at the time instants t_0, t_1, \dots, t_{2M} . We can collect those time samples for each MNA variable in a single vector and denote that vector by $\bar{\mathbf{x}}_{\text{node}}^{(j)}$ ¹. Figure 2.11 displays the structure of $\bar{\mathbf{x}}_{\text{node}}^{(j)}$. Note that the operation performed in this step can be described mathematically as follows

$$\bar{\mathbf{x}}_{\text{node}}^{(j)} = \bar{\Gamma}^{-1} \bar{\mathbf{X}}_{\text{node}}^{(j)} \quad (2.57)$$

¹The subscript ‘‘node’’ highlights that the vector is in the Node-major/Harmonic-minor ordering mode, and the lowercase ‘‘ $\bar{\mathbf{x}}$ ’’ is used to denote the time-domain waveform content.

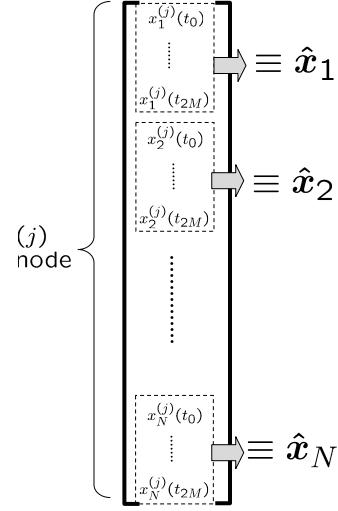


Figure 2.11: The structure of $\bar{\mathbf{x}}_\text{node}^{(j)}$

where $\bar{\Gamma}^{-1}$ is an $N(2M + 1) \times N(2M + 1)$ block diagonal matrix having the matrix Γ^{-1} as its diagonal blocks.

- **STEP 3. Computing the nonlinear components of $f(\mathbf{x}(t))$ at the sampling time points.** This step is straightforward and is simply performed by computing the values of each nonlinear component, $f_i(\mathbf{x}(t))$, at the time samples t_0, t_1, \dots, t_{2M} . This process is carried out by substituting from the time samples available in the vector $\bar{\mathbf{x}}_\text{node}^{(j)}$ into the nonlinear expressions in $f(\mathbf{x}(t))$. The result of this step is a vector, denoted by $\bar{\mathbf{f}}_\text{node}^{(j)}$, which contains the time-domain samples of each component $f_i(\mathbf{x}(t))$ as shown in Figure 2.12.
 - **STEP 4. Performing FFT on $\bar{\mathbf{f}}_\text{node}^{(j)}$.** This step is carried out to obtain the Fourier coefficients for each nonlinear component in the vector $f(\mathbf{x}(t))$. The results of this step is a vector of harmonics, and therefore is denoted by an uppercase $\bar{\mathbf{F}}$, and subscripted by “node” as a reminder that the vector is ordered in Node-Major/Harmonic-minor indexing scheme, $\bar{\mathbf{F}}_\text{node}^{(j)}$. The structure of this vector is shown in Figure 2.13
- Similar to STEP 2, this step also allows a mathematical description of the form

$$\bar{\mathbf{F}}_\text{node}^{(j)} = \bar{\Gamma} \bar{\mathbf{f}}_\text{node}^{(j)} \quad (2.58)$$

where $\bar{\Gamma}$ is an $N(2M + 1) \times N(2M + 1)$ block diagonal matrix having the matrix Γ

2.4. SOLVING THE HB EQUATIONS USING N-R METHOD

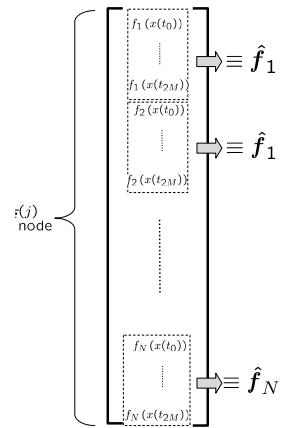


Figure 2.12: Structure of $\bar{f}_{\text{node}}^{(j)}$.

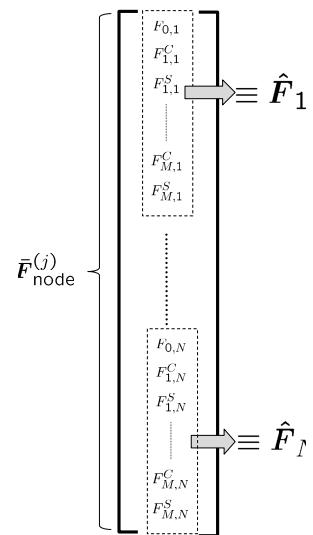


Figure 2.13: Structure of $\bar{F}_{\text{node}}^{(j)}$.

as its diagonal blocks.

- **STEP 5. Reordering in Harmonic-Major/Node-Minor indexing mode.** This is the final step and its objective is to restore the vector to the conventional ordering mode to be able to use it in Eq. (2.53). Such a step can be expressed in a mathematical notation using the permutation matrix \mathcal{P} that was used earlier,

$$\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)}) = \mathcal{P}^T \bar{\mathbf{F}}_{\text{node}}^{(j)} \quad (2.59)$$

Figure 2.14 presents a brief summary for the above steps.

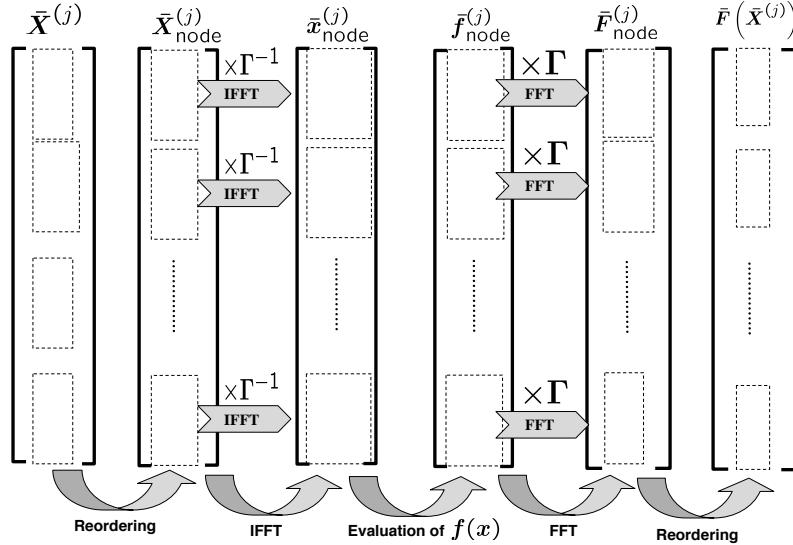


Figure 2.14: A schematic description of the 5 steps used to compute $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ starting from the initial trial vector $\bar{\mathbf{X}}^{(j)}$.

2.4.2 Formulation of $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ Based on the Kronecker Product

The above steps for computing $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ can be cast HB problem can be cast using the Kronecker matrix product matrix operator \otimes , see Appendix B. This will also facilitate the formulation of the Jacobian matrix.

The above 5 steps suggest that $\bar{\mathbf{F}}(\bar{\mathbf{X}})$ can be expressed using the following nested steps.

2.4. SOLVING THE HB EQUATIONS USING N-R METHOD

$$\bar{\mathbf{X}}_{\text{node}}^{(j)} = \mathcal{P} \bar{\mathbf{X}}^{(j)} \quad (2.60)$$

$$\bar{\mathbf{x}}_{\text{node}}^{(j)} = \bar{\boldsymbol{\Gamma}}^{-1} \bar{\mathbf{X}}_{\text{node}}^{(j)} \quad (2.61)$$

$$\mathbf{x}^{(j)} = (\mathbf{I}_N \otimes \mathbf{e}_{k+1,H}^T) \bar{\mathbf{x}}_{\text{node}}^{(j)} \quad (2.62)$$

$$\bar{\mathbf{f}}_{\text{node}}^{(j)} = \sum_{k=0}^{H-1} \mathbf{f} \left(\mathbf{x}^{(j)} \right) \otimes \mathbf{e}_{k+1,H} \quad (2.63)$$

$$\bar{\mathbf{F}}_{\text{node}}^{(j)} = \bar{\boldsymbol{\Gamma}} \bar{\mathbf{f}}_{\text{node}}^{(j)} \quad (2.64)$$

$$\bar{\mathbf{F}} = \mathcal{P}^T \bar{\mathbf{F}}_{\text{node}}^{(j)} \quad (2.65)$$

where,

- (2.60) is the equivalent of Step 1,
- (2.61) is the equivalent of Step 2,
- (2.62) and (2.63) are both equivalent to Step 3,
- (2.63) is equivalent to Step 4,
- (2.64) is equivalent to Step 5.

In a more compact and general way, the above equalities can be combined to yield the following form for $\bar{\mathbf{F}}(\bar{\mathbf{X}})$,

$$\bar{\mathbf{F}}(\bar{\mathbf{X}}) = \mathcal{P}^T \bar{\boldsymbol{\Gamma}} \underbrace{\left(\sum_{k=0}^{H-1} \mathbf{f} \left(\underbrace{\begin{array}{c} \overbrace{(\mathbf{I}_N \otimes \mathbf{e}_{k+1,H}^T)}^{\in \mathbb{R}^{N \times NH}} \bar{\boldsymbol{\Gamma}}^{-1} \underbrace{\mathcal{P} \bar{\mathbf{X}}}_{\bar{\mathbf{X}}_{\text{node}} \in \mathbb{R}^{NH}} \\ \underbrace{\mathbf{x} \in \mathbb{R}^N}_{\bar{\mathbf{x}}_{\text{node}} \in \mathbb{R}^{NH}} \end{array}} \right) \otimes \mathbf{e}_{k+1,H} \right)}_{\bar{\mathbf{f}}_{\text{node}} \in \mathbb{R}^{NH}} \underbrace{\left(\bar{\mathbf{f}}_{\text{node}} \in \mathbb{R}^{NH} \right)}_{\bar{\mathbf{F}} \in \mathbb{R}^{NH}} \quad (2.66)$$

2.4.3 Computing the HB Jacobian $\partial\Phi(\bar{\mathbf{X}})/\partial\bar{\mathbf{X}}|_{\bar{\mathbf{X}}=\bar{\mathbf{X}}^{(j)}}$

The HB Jacobian matrix is obtained by taking the partial derivatives of Φ w.r.t. $\bar{\mathbf{X}}$. Taking the partial derivative of the right side of (2.54) yields

$$\frac{\partial\Phi}{\partial\bar{\mathbf{X}}} = \bar{\mathbf{Y}} + \frac{\partial\bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial\bar{\mathbf{X}}} \quad (2.67)$$

While the first term is readily available from Eqs. (2.31), the main computational effort is spent in computing the second term $\partial\bar{\mathbf{F}}/\partial\bar{\mathbf{X}}$ at the j -th trial vector, $\bar{\mathbf{X}}^{(j)}$. The computation of this term will be derived next using the Kronecker product formulation.

Using the notion of chain differentiation, we can use the following sequence to obtain $\partial\bar{\mathbf{F}}/\partial\bar{\mathbf{X}}$,

$$\frac{\partial\bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial\bar{\mathbf{X}}} = \frac{\partial\bar{\mathbf{F}}}{\partial\bar{\mathbf{F}}_{\text{node}}} \cdot \frac{\partial\bar{\mathbf{F}}_{\text{node}}}{\partial\bar{\mathbf{f}}_{\text{node}}} \cdot \frac{\partial\bar{\mathbf{f}}_{\text{node}}}{\partial\mathbf{x}} \cdot \frac{\partial\mathbf{x}}{\partial\bar{\mathbf{x}}_{\text{node}}} \cdot \frac{\partial\bar{\mathbf{x}}_{\text{node}}}{\partial\bar{\mathbf{X}}_{\text{node}}} \cdot \frac{\partial\bar{\mathbf{X}}_{\text{node}}}{\partial\bar{\mathbf{X}}} \quad (2.68)$$

The matrices of partial derivatives in (2.68) can be computed from Eqs. (2.60)-(2.65) as follows

$$\frac{\partial\bar{\mathbf{X}}_{\text{node}}}{\partial\bar{\mathbf{X}}} = \mathcal{P} \quad (2.69)$$

$$\frac{\partial\bar{\mathbf{x}}_{\text{node}}}{\partial\bar{\mathbf{X}}_{\text{node}}} = \bar{\Gamma}^{-1} \quad (2.70)$$

$$\frac{\partial\mathbf{x}}{\partial\bar{\mathbf{x}}_{\text{node}}} = (\mathbf{I}_N \otimes \mathbf{e}_{k+1,H}^T) \quad (2.71)$$

$$\frac{\partial\bar{\mathbf{f}}_{\text{node}}}{\partial\mathbf{x}} = \sum_{k=0}^{H-1} \frac{\partial\mathbf{f}(\mathbf{x})}{\partial\mathbf{x}} \otimes \mathbf{e}_{k+1,H} \quad (2.72)$$

$$\frac{\partial\bar{\mathbf{F}}_{\text{node}}}{\partial\bar{\mathbf{f}}_{\text{node}}} = \bar{\Gamma} \quad (2.73)$$

$$\frac{\partial\bar{\mathbf{F}}}{\partial\bar{\mathbf{F}}_{\text{node}}} = \mathcal{P}^T \quad (2.74)$$

The matrix $\frac{\partial\mathbf{f}(\mathbf{x})}{\partial\mathbf{x}}$ in (2.72) is a matrix of size $N \times N$ and is given by the partial derivatives

2.4. SOLVING THE HB EQUATIONS USING N-R METHOD

of the circuit nonlinear equations, that is

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \ddots & & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \dots & \frac{\partial f_N}{\partial x_N} \end{bmatrix} \quad (2.75)$$

Substitution from (2.69) to (2.74) into (2.68) gives us,

$$\frac{\partial \bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}} = \mathcal{P}^T \cdot \bar{\Gamma} \cdot \sum_{k=0}^{H-1} \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \otimes \mathbf{e}_{k+1,H} \right) (\mathbf{I}_N \otimes \mathbf{e}_{k+1,H}^T) \cdot \bar{\Gamma}^{-1} \cdot \mathcal{P} \quad (2.76)$$

Using the following property of the Kronecker product

$$(\mathbf{A}_1 \otimes \mathbf{A}_2)(\mathbf{B}_1 \otimes \mathbf{B}_2) = \mathbf{A}_1 \mathbf{B}_1 \otimes \mathbf{B}_1 \mathbf{B}_2 \quad (2.77)$$

in (2.78) simplifies the expression of the nonlinear part of the Jacobian matrix to

$$\frac{\partial \bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}} = \mathcal{P}^T \cdot \bar{\Gamma} \cdot \sum_{k=0}^{H-1} \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \otimes \mathbf{e}_{k+1,H} \mathbf{e}_{k+1,H}^T \right) \cdot \bar{\Gamma}^{-1} \cdot \mathcal{P} \quad (2.78)$$

Since we are interested in computing $\frac{\partial \bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}}$ at $\bar{\mathbf{X}}$ equal to the j -th trial vector $\bar{\mathbf{X}}^{(j)}$, then we have from (2.61) and (2.62)

$$\mathbf{x}^{(j)} = (\mathbf{I}_N \otimes \mathbf{e}_{k+1,H}^T) \bar{\Gamma}^{-1} \bar{\mathbf{X}}_{\text{node}}^{(j)} \quad (2.79)$$

Therefore,

$$\begin{aligned} \left. \frac{\partial \bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}} \right|_{\bar{\mathbf{X}}=\bar{\mathbf{X}}^{(j)}} &= \\ \mathcal{P}^T \cdot \bar{\Gamma} \cdot \sum_{k=0}^{H-1} \left(\left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=(\mathbf{I}_N \otimes \mathbf{e}_{k+1,H}^T) \bar{\Gamma}^{-1} \bar{\mathbf{X}}_{\text{node}}^{(j)}} \otimes \mathbf{e}_{k+1,H} \mathbf{e}_{k+1,H}^T \right) \cdot \bar{\Gamma}^{-1} \cdot \mathcal{P} \end{aligned} \quad (2.80)$$

The above expression for the nonlinear part of the Jacobian matrix can be further sim-

plified by defining a vector $\mathbf{z}_k \in \mathbb{R}^N$

$$\mathbf{z}_k^{(j)} := (\mathbf{I}_N \otimes \mathbf{e}_{k+1,H}^T) \bar{\boldsymbol{\Gamma}}^{-1} \bar{\mathbf{X}}_{\text{node}}^{(j)} \quad (2.81)$$

and rewriting both of $\bar{\boldsymbol{\Gamma}}$ and $\bar{\boldsymbol{\Gamma}}^{-1}$ in the Kronecker product formulation

$$\bar{\boldsymbol{\Gamma}} = \mathbf{I}_N \otimes \boldsymbol{\Gamma} \quad (2.82)$$

$$\bar{\boldsymbol{\Gamma}}^{-1} = \mathbf{I}_N \otimes \boldsymbol{\Gamma}^{-1} \quad (2.83)$$

giving us

$$\left. \frac{\partial \bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}} \right|_{\bar{\mathbf{X}}=\bar{\mathbf{X}}^{(j)}} = \mathcal{P}^T \cdot \underbrace{\mathbf{I}_N \otimes \boldsymbol{\Gamma} \cdot \sum_{k=0}^{H-1} \left(\left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{z}_k^{(j)}} \otimes \mathbf{e}_{k+1,H} \mathbf{e}_{k+1,H}^T \right) \cdot \mathbf{I}_N \otimes \boldsymbol{\Gamma}^{-1} \cdot \mathcal{P}}_{\text{can be simplified using (2.77)}} \quad (2.84)$$

As indicated in the above equation, the property of Kronecker product in (2.77) may be used to further compactify the right side in (2.84) as follows

$$\left. \frac{\partial \bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}} \right|_{\bar{\mathbf{X}}=\bar{\mathbf{X}}^{(j)}} = \mathcal{P}^T \cdot \underbrace{\sum_{k=0}^{H-1} \left(\left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{z}_k^{(j)}} \otimes [\boldsymbol{\Gamma} \mathbf{e}_{k+1,H} \mathbf{e}_{k+1,H}^T \boldsymbol{\Gamma}^{-1}] \right)}_{\text{matrix of size } NH \times NH} \cdot \mathcal{P} \quad (2.85)$$

It is not difficult to see that the summation in (2.85) is a matrix of size $NH \times NH$ which may be represented as follows,

$$\begin{aligned} & \sum_{k=0}^{H-1} \left(\left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{z}_k^{(j)}} \otimes [\boldsymbol{\Gamma} \mathbf{e}_{k+1,H} \mathbf{e}_{k+1,H}^T \boldsymbol{\Gamma}^{-1}] \right) = \\ & \left[\boldsymbol{\Gamma} \text{ diag} \left(\left. \frac{\partial f_p}{\partial x_q} \right|_{\mathbf{x}=\mathbf{z}_0^{(j)}}, \left. \frac{\partial f_p}{\partial x_q} \right|_{\mathbf{x}=\mathbf{z}_1^{(j)}}, \dots, \left. \frac{\partial f_p}{\partial x_q} \right|_{\mathbf{x}=\mathbf{z}_{H-1}^{(j)}} \right) \boldsymbol{\Gamma}^{-1} \right] p = 1, \dots, N, \\ & q = 1, \dots, N \end{aligned} \quad (2.86)$$

where,

- “diag (a_1, a_2, \dots, a_H) ” denotes the $H \times H$ diagonal matrix whose diagonal elements are given by a_1, \dots, a_H ,

2.5. STRUCTURE OF THEHB JACOBIAN MATRIX

- $\frac{\partial f_p}{\partial x_q} \Big|_{\mathbf{x}=\mathbf{z}_m^{(j)}}$ denotes the partial derivative of the p -th component of the nonlinear function vector $\mathbf{f}(\mathbf{x})$ with respect to the q -th component in the vector of the MNA variables \mathbf{x} computed for $\mathbf{x} = \mathbf{z}_m^{(j)}$, and
- the notation of $[L_{p,q}]_{p=1,\dots,N,q=1,\dots,N}$, where $L_{p,q} \in \mathbb{R}^{H \times H}$ is used in (2.86) to denote a block-structured matrix, with $N \times N$ blocks with size $H \times H$ each.

Let us define the matrix in (2.86) as $\mathcal{J}_{\text{node}}$. Thus, we have

$$\left. \frac{\partial \bar{\mathbf{F}}(\bar{\mathbf{X}})}{\partial \bar{\mathbf{X}}} \right|_{\bar{\mathbf{X}}=\bar{\mathbf{X}}^{(j)}} = \mathbf{P}^T \mathcal{J}_{\text{node}} \mathbf{P} \quad (2.87)$$

Therefore, the final expression for the total Jacobian matrix is given by

$$\frac{\partial \Phi}{\partial \bar{\mathbf{X}}} = \bar{\mathbf{Y}} + \mathbf{P}^T \mathcal{J}_{\text{node}} \mathbf{P} \quad (2.88)$$

2.5 Structure of theHB Jacobian Matrix

The structure of the jacobian matrix plays an essential role in determining the computational complexity involved in solving the HB problem numerically. Such structure can be studied by analyzing the structures of its two main components, namely $\bar{\mathbf{Y}}$ and $\mathbf{P}^T \mathcal{J}_{\text{node}} \mathbf{P}$.

The structure of $\bar{\mathbf{Y}}$ is generally a block-diagonal structure as shown by Equation (2.31). This block structure form is generally problem or circuit-dependent and is associated with linear components of the circuit.

On the other hand, $\mathbf{P}^T \mathcal{J}_{\text{node}} \mathbf{P}$ is obtained from $\mathcal{J}_{\text{node}}$ using similarity transformation with the permutation matrix \mathbf{P} . Therefore, it suffices to analyze the matrix $\mathcal{J}_{\text{node}}$ to gain some insights in the matrix $\mathbf{P}^T \mathcal{J}_{\text{node}} \mathbf{P}$.

As illustrated by (2.86) $\mathcal{J}_{\text{node}}$ is a block-structured matrix, where it is composed of $N \times N$ blocks, each of size $H \times H$. Let us use $\mathcal{J}_{\text{node}}^{p,q}$ to denote block at, say, (p, q) block entry, where $p, q = 1, \dots, N$. Hence,

$$\mathcal{J}_{\text{node}}^{p,q} = \Gamma \begin{bmatrix} \frac{\partial f_p}{\partial x_q} \Big|_{\mathbf{x}=\mathbf{z}_0^{(j)}} & 0 \dots & 0 & 0 \\ 0 & \frac{\partial f_p}{\partial x_q} \Big|_{\mathbf{x}=\mathbf{z}_1^{(j)}} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & & 0 \\ 0 & 0 & \dots & 0 & \frac{\partial f_p}{\partial x_q} \Big|_{\mathbf{x}=\mathbf{z}_{H-1}^{(j)}} \end{bmatrix} \Gamma^{-1} \quad (2.89)$$

Several remarks can be made based on the form of the block shown in (2.89).

1. A necessary condition to have a non-zero $\mathcal{J}_{\text{node}}^{p,q}$ block is that f_p is a function of x_q .
2. $\mathcal{J}_{\text{node}}^{p,q}$ is a diagonal block if $\frac{\partial f_p}{\partial x_q}$ is a constant at all the sampling points. For example assume that

$$\left. \frac{\partial f_p}{\partial x_q} \right|_{x=\mathbf{z}_0^{(j)}} = \left. \frac{\partial f_p}{\partial x_q} \right|_{x=\mathbf{z}_1^{(j)}} = \cdots = \left. \frac{\partial f_p}{\partial x_q} \right|_{x=\mathbf{z}_{H-1}^{(j)}} = \beta \quad (2.90)$$

where β is some constant value, then

$$\begin{aligned} \mathcal{J}_{\text{node}}^{p,q} &= \Gamma \begin{bmatrix} \beta & 0 & \cdots & 0 & 0 \\ 0 & \beta & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & 0 \\ 0 & 0 & \cdots & 0 & \beta \end{bmatrix} \Gamma^{-1} \quad (2.91) \\ &= \beta \Gamma \mathbf{I}_H \Gamma^{-1} \quad (2.92) \end{aligned}$$

$$= \beta \mathbf{I}_H \text{ (diagonal block)} \quad (2.93)$$

where \mathbf{I}_H is an identity matrix of size H .

3. Using the above remark, one can deduce further that $\mathcal{J}_{\text{node}}^{p,q}$ is a diagonal block if the function represented by the partial derivative $\frac{\partial f_p}{\partial x_q}$ is independent of the time t . This could occur if all the independent MNA variables of which $\frac{\partial f_p}{\partial x_q}$ is function, happened to be constant, i.e. do not change with time. An example of such situation is the case where the MNA variable represents the voltage of node attached to a DC source that is grounded on the other end. Another situation, can be encountered in the case where all the sources in the circuit are dc sources, with no time-dependent excitations in the circuit.
4. If the block $\mathcal{J}_{\text{node}}^{p,q}$ is a non-zero block, then in general it is a full block, i.e., it should have all of its H^2 entries as nonzero. However, if the f_p behaves in a mildly nonlinear manner, then this block becomes diagonally dominant, where the magnitude of the entries rapidly decrease one moves further away from the diagonal.

To illustrate this idea further, assume that f_p is given as a function of the MNA variables \mathbf{x} as follows

$$f_p = 3 + 0.1x_p + 0.0001x_p^2 \quad (2.94)$$

where it is assumed to be a function of only the q th MNA variable. This particular form is chosen to show a mild nonlinear behaviour since the nonlinear term is much

2.6. ERROR DYNAMICS

smaller than the linear terms. It should be obvious that

$$\frac{\partial f_p}{\partial x_q} = 0.1 + 0.0002x_p \quad (2.95)$$

In this situation, the $\mathcal{J}_{\text{node}}^{p,q}$ block is given by

$$\mathcal{J}_{\text{node}}^{p,q} = 0.1\mathbf{I}_H + 0.0002 \times \Gamma \begin{bmatrix} x_q(t_0) & 0 \dots & 0 & 0 \\ 0 & x_q(t_1) & 0 & \dots \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & x_q(t_{H-1}) \end{bmatrix} \Gamma^{-1} \quad (2.96)$$

where $x_q(t_i)$ is the i th sample in the time-domain waveform of the q th MNA variable. The above expression indicates a strong diagonal entries (proportional to 0.1) with much smaller off-diagonal entries that are mainly scaled by 0.0002.

2.6 Error Dynamics

Apart from the round-off errors that results from using finite machine precision numbers, there are three sources of errors that arise in the course of HB analysis. The first source of error can be encountered if the numerical technique used to solve HB equations does not converge completely. This error can be controlled if the N-R method is used as the solution algorithm and the tolerance stopping criteria is made arbitrarily small. The other two sources are more dominant and may lead to inaccurate results if not considered carefully. These two sources result from truncating the infinite spectrum of $\mathbf{x}(t)$ to a finite spectrum with M harmonics. In this truncation process, harmonics of order higher than M masquerade as lower than M frequency components and end up corrupting the truncated spectrum. This type of error is known as the aliasing error. The only remedy to this error is to use larger values for the truncation index M to ensure that harmonics of order higher than M are negligibly small to contribute noticeable error to the harmonics inside the truncated spectrum.

Even though aliasing error can be driven to small levels via increasing the truncation index M to large values, this truncation will still generate a third source of error due to truncating the spectral components of waveforms in $\mathbf{f}(\mathbf{x}(t))$ to the same number of harmonics M . Recall that this truncation was performed to force the number of equations to be equal to the number of unknowns. There are two aspects to the third error source. The first is an error due to truncation and the second one is due to another aliasing error. We consider the error due to truncation first.

In fact, by truncating waveforms in $\mathbf{f}(\mathbf{x}(t))$ to only M harmonics, the steady-state solution obtained upon convergence is not the steady-state solution of the original problem (2.39). Instead, this solution represents the steady-state operating point of a slightly per-

turbed version of (2.39). This fact is illustrated through the simple example circuit of Figure 2.6. Clearly, this circuit can be described using simple Kirchhoff Voltage Law (KVL), without the need for the sophisticated formulation techniques of Chapter 1. Using KVL on this circuit we get,

$$v_a + v_a^3 - a \sin(\omega_0 t) = 0 \quad (2.97)$$

As noted earlier $v_a(t)$ has infinite spectrum. Assuming that $v_a(t)$ can be adequately truncated to a finite spectrum with $M = 1$,

$$v_a(t) = K_0 + K_1^C \cos(\omega_0 t) + K_1^S \sin(\omega_0 t) \quad (2.98)$$

where K_1^C , K_1^S and K_0 are unknowns. The Fourier representation for $v_a(t)^3$ thus takes the form

$$v_a^3(t) = T_0 + \sum_{m=1}^3 T_m^C \cos(m\omega_0 t) + T_m^S \sin(m\omega_0 t) \quad (2.99)$$

where

$$\begin{aligned} T_0 &= K_0^3 + \frac{3}{2}K_0 K_1^{C^2} + \frac{3}{2}K_0 K_1^{S^2} \\ T_1^C &= 3K_1^C K_0^2 + \frac{9}{4}K_1^{S^2} K_1^C + \frac{3}{4}K_1^S C^3 \\ T_1^S &= 3K_1^S K_0^2 + \frac{3}{4}K_1^{C^2} K_1^S + \frac{1}{4}K_1^{S^3} \\ T_2^C &= \frac{3}{2}K_1^{C^2} K_0 + \frac{3}{2}K_1^{S^2} K_0 \\ T_2^S &= 3K_1^C K_1^S K_0 \\ T_3^C &= \frac{3}{4}K_1^{S^2} K_1^C + \frac{1}{4}K_1^{C^3} \\ T_3^S &= \frac{3}{4}K_1^{C^2} K_1^S + \frac{1}{4}K_1^{S^3} \end{aligned}$$

Substituting from (2.99) and (2.98) into (2.97) and using the orthogonality property of the

2.6. ERROR DYNAMICS

sines and cosines gives us a system of 7 equations

$$\begin{aligned} K_0 + T_0 &= 0 \\ K_1^C + T_1^C &= 0 \\ K_1^S + T_1^S &= a \\ T_2^C &= 0 \\ T_2^S &= 0 \\ T_3^C &= 0 \\ T_3^S &= 0 \end{aligned}$$

Since we have only three unknowns and 7 equations, an exact solution satisfying all of the 7 equations is not possible. Thus a solution for K_0, K_1^C, K_1^S satisfying the first three equations will leave the last four equations with an error term. In fact, this solution is the exact solution for a slightly perturbed problem given by

$$v_a(t) + rv_a^3(t) = a \sin(\omega_0 t) + \underbrace{r \left(T_2^C \cos(2\omega_0 t) + T_2^S \sin(2\omega_0 t) + T_3^C \cos(3\omega_0 t) + T_3^S \sin(3\omega_0 t) \right)}_{\substack{\text{perturbation terms} \\ \text{Terms of order higher than } M=1 \text{ in } v_a^3(t)}} \quad (2.100)$$

This indicates that by merely truncating the waveforms in the nonlinear term $v_a^3(t)$ to the same number of terms of $v_a(t)$, we end up solving for the steady-state of another slightly perturbed problem. It can be observed from (2.100) that the magnitude of the perturbation terms is proportional to the amplitude of the harmonics included in the spectrum K_0 , and K_1 . Furthermore, the perturbing terms have frequency components $(2\omega_0, 3\omega_0)$ higher than the highest component in the presumed solution, ω_0 . In fact, closer inspection of the perturbed circuit equation (2.100) reveals that it represents the same circuit as the original one, but with additional voltage sources that are equal to the perturbing terms, as shown in Figure 2.15. It is not difficult to notice also that the perturbation terms (or sources) are those terms of $v_a^3(t)$ with order higher than the truncation index $M = 1$.

If $v_a(t)$ is represented by $M = 2$ Fourier series,

$$v_a(t) = K_0 + K_1 \sin(\omega_0 t + \phi_1) + K_2 \sin(2\omega_0 t + \phi_2) \quad (2.101)$$

then it can be shown that the perturbation terms will include only terms with magnitudes proportional to $K_0 K_1 K_2$, $K_2^2 K_0$, $K_1^2 K_2$, $K_2^2 K_1$, and K_2^3 , and frequency components **higher** than $2\omega_0$. Hence if K_2 was negligibly small, then including it in $v_a(t)$ should further reduce the perturbation terms, and make the obtained steady-state solution closer to the so-

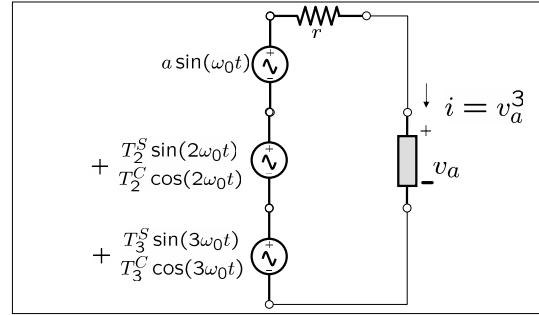


Figure 2.15: The equivalent circuit of the perturbed problem in (2.100).

lution of the problem under analysis. Thus including higher order terms in the truncated spectrum (i.e., increasing M) will tend to minimize the effect of the perturbation terms and the solution obtained will tend to align with the real steady-state operating point of the circuit.

The observations made for the simple circuit of Figure 2.6 hold also true for general nonlinear circuits. As a matter of fact, the solution satisfying the first $2M + 1$ systems in (2.49) is the steady-state operating point for a slightly modified circuit. This is a direct result of the fact that higher order systems ($\geq M+1$) in (2.49) are not solved exactly by that solution. Instead, they are left with some error terms $\mathcal{E}_n^{(C,S)}$, for $n = M+1, M+2, \dots, \infty$. In other words the solution obtained by considering the first $2M + 1$ systems only is a

2.6. ERROR DYNAMICS

solution to the following infinite-dimensional system,

$$\begin{aligned}
& \mathbf{G}\mathbf{X}_0 + \mathbf{F}_0(\bar{\mathbf{X}}) = \mathbf{B}_0 \\
& \mathbf{G}\mathbf{X}_1^C + \omega_0 \mathbf{C}\mathbf{X}_1^S + \mathbf{F}_1^C(\bar{\mathbf{X}}) = \mathbf{B}_1^C \\
& -\omega_0 \mathbf{C}\mathbf{X}_1^C + \mathbf{G}\mathbf{X}_1^S + \mathbf{F}_1^S(\bar{\mathbf{X}}) = \mathbf{B}_1^S \\
& \quad \vdots = \vdots \\
& \mathbf{G}\mathbf{X}_K^C + K\omega_0 \mathbf{C}\mathbf{X}_K^S + \mathbf{F}_K^C(\bar{\mathbf{X}}) = \mathbf{B}_K^C \\
& -K\omega_0 \mathbf{C}\mathbf{X}_K^C + \mathbf{G}\mathbf{X}_K^S + \mathbf{F}_K^S(\bar{\mathbf{X}}) = \mathbf{B}_K^S \\
& \mathbf{G}\mathbf{X}_{K+1}^C + (K+1)\omega_0 \mathbf{C}\mathbf{X}_{K+1}^S + \mathbf{F}_{K+1}^C(\bar{\mathbf{X}}) = \mathbf{0} \\
& -(K+1)\omega_0 \mathbf{C}\mathbf{X}_{K+1}^C + \mathbf{G}\mathbf{X}_{K+1}^S + \mathbf{F}_{K+1}^S(\bar{\mathbf{X}}) = \mathbf{0} \\
& \quad \vdots = \vdots \\
& \mathbf{G}\mathbf{X}_M^C + M\omega_0 \mathbf{C}\mathbf{X}_M^S + \mathbf{F}_M^C(\bar{\mathbf{X}}) = \mathbf{0} \\
& -M\omega_0 \mathbf{C}\mathbf{X}_M^C + \mathbf{G}\mathbf{X}_M^S + \mathbf{F}_M^S(\bar{\mathbf{X}}) = \mathbf{0} \\
& \mathbf{F}_{M+1}^C(\bar{\mathbf{X}}) = \mathcal{E}_{M+1}^C \\
& \mathbf{F}_{M+1}^S(\bar{\mathbf{X}}) = \mathcal{E}_{M+1}^S \\
& \mathbf{F}_{M+2}^C(\bar{\mathbf{X}}) = \mathcal{E}_{M+2}^C \\
& \mathbf{F}_{M+2}^S(\bar{\mathbf{X}}) = \mathcal{E}_{M+2}^S \\
& \quad \vdots = \vdots
\end{aligned} \tag{2.102}$$

The error terms $\mathcal{E}_n^{(C,S)}$, are those spectral components of waveforms in $\mathbf{f}(\mathbf{x}(t))$ not included in the truncation. Those terms are typically characterized by decreasing magnitudes and frequency components higher than $(M+1)\omega_0$. The solution obtained through considering the first $2M+1$ systems appears as the steady-state periodic condition of a slightly modified circuit, which is similar to the original circuit except that it has additional high frequency sources ($\geq (M+1)\omega_0$) whose power is proportional to $\mathcal{E}_n^{(C,S)}$.

The above discussion basically showed that higher values of M are likely to decrease both of the truncation and aliasing errors. Increasing M , however, increases the size of the nonlinear problem significantly leading to higher computational time. In general there is no accurate way of knowing the most suitable value for M beforehand, but it is always possible to have an idea about its range based on the nature of the input excitation, and further refine its value based on the progress of the iterative process or using the final results once a solution has been reached. For example, a large value of M will be needed if the input stimulus is sinusoidal with an amplitude large enough to make the circuit behave strongly nonlinearly. The input stimuli in such a circuit are likely to switch on and off devices at

a fast rate and make the waveforms alternate between the highest and lowest value of the DC supply values in square-wave-like manner with sharp bends and fast transitions. These waveforms can only be captured by a large number of harmonics. If input stimulus is sinusoidal with relatively small magnitude, the circuit will tend to behave mildly nonlinearly, and the waveforms of the circuit variables will be almost sinusoidal with few higher order harmonics at much lower values than the fundamental. Another instance, where large valued M will be needed, will occur if the input stimulus is square wave and requires large number of harmonics K . In this case, even though the input amplitude might be small, M has to be increased to higher values than the number of harmonics in the input to capture the waveforms accurately.

It is also possible to detect, once a final steady-state solution has been obtained, if the value of M used in that solution is adequate in minimizing the truncation error. A steady-state solution that is heavily contaminated by truncation error will show in the time-domain with high frequency small ripples. This is commonly known as the Gibb's phenomena, and comes as consequence of the fact that the error terms, acting as parasitic sources, are significant enough to contribute high-frequency components to the response.

Truncation error can also be numerically estimated upon convergence to the steady-state solution, through estimating the additional residual error terms $\mathcal{E}_n^{(C,S)}$. This is typically done through **oversampling** the solution waveforms $\mathbf{x}(t)$ at twice the sampling rate, i.e., with $4M + 1$ sampling points. The nonlinearities in $\mathbf{f}(\mathbf{x}(t))$ are then computed at those $2(2M + 1)$ time points and then converted to the frequency-domain using DFT or FFT techniques, resulting in $4M + 1$ Fourier coefficients, where components higher than or equal to $2M$ represent error terms $\mathcal{E}_n^{(C,S)}$, $n = M + 1, M + 2, \dots, 2M$.

The final error component to be considered here is the error due to aliasing encountered in converting $\mathbf{f}(\mathbf{x}(t))$ to its Fourier spectrum in each iteration. This error, similar to the aliasing error in $\mathbf{x}(t)$, occurs if significant spectral components of $\mathbf{f}(\mathbf{x}(t))$ are not included in the truncation set. The higher frequency components outside the truncation set $\mathbf{F}_m^{(C,S)}$, $m \geq M + 1$ corrupt the values computed for the low frequency components $m \leq M$. The idea of oversampling described above can be used to reduce this error. It should be noted though, that this oversampling will have to be done in each iteration, and not only once upon convergence as in the case of estimating the truncation error. In that case the size of the nonlinear problem will still remain the same, where only the first $2M + 1$ frequency components will be selected from the oversampled frequency components. Nonetheless, there will be considerable computation cost added as a result of running the DFT or the FFT at twice the sampling frequency.

2.6. ERROR DYNAMICS

Chapter **3**

Harmonic Balance for Circuits with Multi-Tone Excitation

The previous chapter showed how the HB approach can be used to find the steady-state operating point of nonlinear circuits excited by a single-tone large signal. Analysis of single-tone circuits, however, bears little practical value, since most communication circuits are designed to handle multi-tone excitations. One example is found in up-converter mixer circuits which are fed with a local oscillator and another baseband signal with much lower frequency. Both the oscillator and the base band frequencies operate at totally different frequencies. To determine various basic figures-of-merits, this circuit will have to be simulated in the steady-state under two-tone periodic excitations. The response in this case appears at multiples of the excitation frequencies as well as at mixing products between them. Although conceptually similar to single-tone, multi-tone steady-state analysis usually presents unique computational challenges. In particular, the problem of developing Fourier transforms suitable for multi-tone signals comes at the forefront of those challenges and will preoccupy most of the discussion in this chapter.

3.1 Multi-tone and Almost-Periodic Waveforms

The term “multi-tone” excitations generally refers to signals having more than one *non-commensurate* frequencies. Generally two frequencies f_1 and f_2 are said to be non-commensurate if they **can not** be related in the following form,

$$mf_1 = nf_2 \quad (3.1)$$

3.1. MULTI-TONE AND ALMOST-PERIODIC WAVEFORMS

using **any** nonzero positive integers m and n . Likewise, two frequencies f_1 and f_2 are said to be commensurate if they can be related as in (3.1). Note that by this definition, single-tone signals can be considered a special case of commensurate signals for which m (or n) = 1. A well-known class of commensurate signals are those given by rational numbers, e.g., u/v where both u and v are integers.

Further examples of commensurate waveforms can still be constructed. Consider for example a signal having two frequency components $f_1 = 10\text{KHz}$ and $f_2 = 10.1\text{KHz}$. In this case we can write

$$\underbrace{101}_{m} f_1 = \underbrace{100}_{n} f_2 \quad (3.2)$$

Another example can be constructed by setting $f_1 = 10.2\text{KHz}$ and $f_2 = 10.345\text{KHz}$ for which we can write,

$$\underbrace{2069}_{m} f_1 = \underbrace{2040}_{n} f_2 \quad (3.3)$$

One last example, that serves to illustrate a general trend, can be obtained for $f_1 = 1\text{GHz}$ and $f_2 = 1.00001\text{GHz}$ for which the relation (3.1) holds if

$$m = 2^{53}, \quad \text{and} \quad n = 9007109183649156.$$

From the above examples, it may seem plausible at first thought to conclude that any two arbitrary frequencies can be represented in the relation (3.1)¹, and can therefore be considered commensurate. It should be noted, however, that this would be the case only if both of those frequencies are truncated to a finite precision, such as in the situation where both f_1 and f_2 are stored in a computer memory that can only allow finite precision. This is a consequence of the fact that any finite precision numbers can be represented as rational numbers. Nonetheless, if both frequencies are strict real numbers with infinite precision, the corresponding integers (m, n) that make Eq. (3.1) hold will have to grow unbounded that they may ultimately be deemed non-existing.

Given that we are mainly concerned with computer-based techniques where frequencies are represented using finite precision, we may feel justified to assume that all frequencies in the scope of this text are commensurate. It is always possible for waveforms having two commensurate tones to be represented by a periodic waveform with a common fundamental frequency. For example, a two-tone waveform having two commensurate components f_1 and f_2 such that $mf_1 = nf_2$, can be represented by a periodic waveform with common fundamental f_0 , given by f_2/m or f_1/n . This observation may lead us to the conclusion that all waveforms considered in this text may be treated as periodic waveforms provided that the common fundamental frequency can be deduced. This conclusion, however, is not correct. This is because for any general two frequencies, the common period, which is given by the reciprocal of the common fundamental, is typically very large, where the waveforms go over millions of cycles before a periodic pattern can be observed. For instance, a non-

¹Of course this is possible provided that we have enough storage to hold the corresponding large integers (m, n)

periodic waveform may be considered periodic with a common period equal to infinity.

This fact suggests that waveforms with general two-tones should be given a different names to distinguish them from purely periodic single-tone waveforms encountered in Chapter 2. In fact, the common name used to denote waveforms having two general tones is “**almost-periodic**” (AP) or “**Quasi Periodic**” (QP).

Most of the discussions and analysis presented in this chapter will focus initially on handling situations involving almost-periodic signals having only *two-tones*. We aim by that at simplifying the introduction of new concepts. Nonetheless, occasionally we will present generalization to the case of multi-tone almost-periodic signals wherever this becomes convenient.

For instance, the formal definition of non-commensurate signals having d fundamental or base tones f_1, f_2, \dots, f_d can be generalized from (3.1) by stating that f_1, f_2, \dots, f_d are non-commensurate if

$$\sum_{l=1}^d f_l k_l = 0, \quad k_l \in \mathbb{Z} \quad \iff k_1 = k_2 = \dots = k_d = 0 \quad (3.4)$$

where \mathbb{Z} is the set of all integers. The expression (3.4) is merely stating that f_1, f_2, \dots, f_d are non-commensurate if they are linearly independent, where no single f_i can be written as a linear combination of the rest of f_i 's using integers.

Steady-state analysis of nonlinear circuits with almost-periodic excitations is challenging on various levels. We begin investigating those challenges by examining how a nonlinear circuit responds to an almost-periodic excitation.

3.2 Response of a Nonlinear Circuit to Almost-Periodic Excitations

Assume that a nonlinear conductance whose $i - v$ relationship is governed by a third-order polynomial, given by,

$$i = a_0 + a_1 v + a_2 v^2 + a_3 v^3 \quad (3.5)$$

is being excited by a two-tone voltage signal given by

$$v(t) = \sin(\omega_1 t) + \sin(\omega_2 t) \quad (3.6)$$

3.2. RESPONSE OF A NONLINEAR CIRCUIT TO ALMOST-PERIODIC EXCITATIONS

Then the current steady-state waveform can be written, after some manipulation and rearrangements of terms using basic trigonometric identities, as follows,

$$\begin{aligned}
i(t) = & a_0 + a_2 + \left(a_1 + \frac{1}{2} + \frac{2}{4}a_3 + \frac{3}{2}a_3 \right) \sin(\omega_1 t) + \left(a_1 + \frac{1}{2} + \frac{2}{4}a_3 + \frac{3}{2}a_3 \right) \sin(\omega_2 t) \\
& - \frac{1}{2}a_2 \cos(2\omega_1 t) - \frac{1}{2}a_2 \cos(2\omega_2 t) \\
& + a_2 \cos((\omega_2 - \omega_1)t) - a_2 \cos((\omega_1 + \omega_2)t) \\
& + \frac{3}{4}a_3 \sin((2\omega_2 - \omega_1)t) + \frac{3}{4}a_3 \sin((2\omega_1 - \omega_2)t) \\
& - \frac{3}{4}a_3 \sin((2\omega_2 + \omega_1)t) - \frac{3}{4}a_3 \sin((2\omega_1 + \omega_1)t) \\
& - \frac{1}{4}a_3 \sin(3\omega_1 t) - \frac{1}{4}a_3 \sin(3\omega_2 t)
\end{aligned} \tag{3.7}$$

The foremost observation that we can harvest from this skeletal example is that the current (or the response in this case) appears not only at multiples of the excitation frequencies ($2\omega_1, 2\omega_2, 3\omega_1, 3\omega_2$), as one would expect in a third-order nonlinearity, but also at combinations of those multiples (e.g., $2\omega_1 \pm \omega_2$). If the nonlinearity was of the general form allowing an infinite Taylor Series, i.e.,

$$i = \sum_{k=0}^{\infty} \frac{1}{k!} \frac{d^k i}{dv^k} v^k \tag{3.8}$$

then the current will be expected to have frequency components at given by

$$k_1\omega_1 + k_2\omega_2 \quad \begin{cases} k_1 = \pm 1, \pm 2, \dots \\ k_2 = \pm 1, \pm 2, \dots \end{cases} \tag{3.9}$$

If the nonlinearity is embedded within a bigger circuit with some feedback, then the resulting current will ultimately influence other voltages and currents making their frequency content appear at the frequency combinations of (3.9).

In that sense, we can expect the vector of unknown waveforms in a general formulation such as the MNA formulation for a general nonlinear circuit to have an almost periodic

waveform with an infinite expansion of the form

$$\mathbf{x}(t) = \mathbf{X}_0 + \sum_{\substack{k_1 = -\infty \\ k_1 \neq 0}}^{\infty} \sum_{\substack{k_2 = -\infty \\ k_2 \neq 0}}^{\infty} \mathbf{X}_{k_1, k_2}^C \cos((k_1 \omega_1 + k_2 \omega_2)t) + \mathbf{X}_{k_1, k_2}^S \sin((k_1 \omega_1 + k_2 \omega_2)t) \quad (3.10)$$

From that point onward, the above series expansion will be referred to as the generalized Fourier expansion. This is to be distinguished from the special case of Fourier transform of periodic single-tone waveforms, where the frequency components are harmonically related. We also refer to the basic tones in the excitation as the base or fundamental frequencies.

Furthermore, if the excitation has multiple base frequencies $\omega_1, \omega_2, \dots, \omega_d$, the above expansion generalizes further to

$$\begin{aligned} \mathbf{x}(t) = \mathbf{X}_0 + \sum_{\substack{k_1, k_2, \dots, k_d = -\infty \\ k_i \neq 0 \ i = 1, \dots, d}}^{\infty} & \mathbf{X}_{k_1, k_2, \dots, k_d}^C \cos \left(\left(\sum_{l=1}^d k_l \omega_l \right) t \right) \\ & + \mathbf{X}_{k_1, k_2, \dots, k_d}^S \sin \left(\left(\sum_{l=1}^d k_l \omega_l \right) t \right) \end{aligned} \quad (3.11)$$

3.3 Truncating the Spectrum of Almost-Periodic signals

The first step that we have to take in order to make the task of computing the Fourier coefficients amenable to computer-based techniques is to truncate the generalized Fourier series to some suitable finite series. This decision was actually very straightforward for single-tone periodic representation. In the more general cases of multi-tone signals the issue of truncating the infinite spectrum is quite involved.

Explaining truncation schemes can be best approached by first representing the spectrum of the two-tone signal using a two-dimensional grid as shown in Figure 3.1.

Each point on the grid corresponds to a unique combination of ω_1 and ω_2 using the associated integer indices k_1 and k_2 . Points shown by a square mark correspond to multiples of either ω_1 or ω_2 , while points marked by a circle correspond to a given combination of those multiples, or the so-called “mixing products” between ω_1 and ω_2 .

In truncating the spectrum, it is essential to have a distinct set of frequencies in the truncated spectrum. For example, a point on the grid having $k_1 = 1$ and $k_2 = -2$ corresponds to the same frequency associated with the point at $k_1 = -1$ and $k_2 = 2$; the frequency points at those locations only differ by their signs, and when factored in the general Fourier expansion appear as the same term. To assist in the process of identifying those frequencies that differ by their signs only, the grid in Figure 3.1 shows two types of points with “plus” and “minus” signs, where it is to be understood that frequency points associated with “plus” are the negative of those associated with the minus signs. Thus one only needs to consider

3.3. TRUNCATING THE SPECTRUM OF ALMOST-PERIODIC SIGNALS

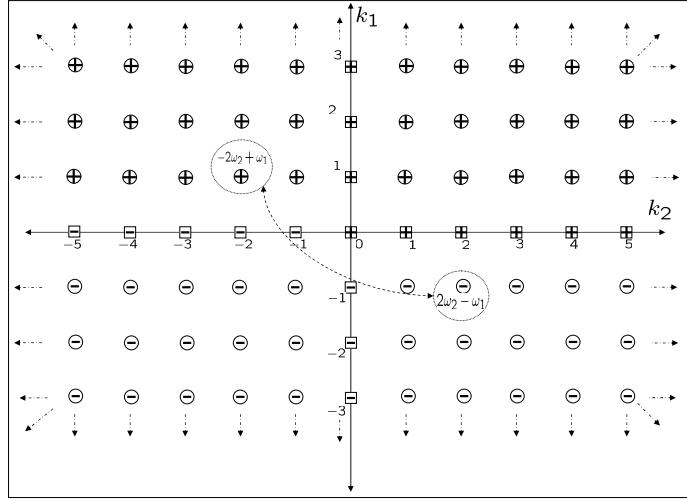


Figure 3.1: Representing the spectrum of a two-tone signal by a two-dimensional grid.

points with the same sign to filter out points corresponding to similar frequencies that differ by their signs only.

It should be noted though that different points with the same sign can still correspond to the same frequency if ω_1 and ω_2 are commensurate. For example, if $\omega_1 = 1/2\text{rad/Sec.}$ and $\omega_2 = 4/3\text{rad/Sec.}$, then a point at $k_1 = 8, k_2 = 9$ represents the same frequency as a point at $k_1 = 40, k_2 = -3$ although both points appear with the same sign on the grid. Yet this situation is very unlikely to occur in practice since the typically values for ω_1 and ω_2 encountered make the similar points practically too separate on the grid to be included in the same truncation set. Therefore, it is safe to assume that by including only points with the same sign, one is *almost* assured that the resulting frequencies are distinct. Two general schemes can be used to provide suitable truncation that complies with the above requirement.

3.3.1 Box truncation for two-tone waveform

The first truncation scheme is the box truncation which is illustrated by Figure 3.2. Box truncation usually works by specifying two integer values K_1 and K_2 and choosing all the frequencies λ given by

$$\lambda = |k_1\omega_1 + k_2\omega_2| \quad (3.12)$$

such that

$$(0 \leq k_1 \leq K_1, |k_2| \leq K_2) \text{ and, } (k_1 \neq 0 \text{ if } k_2 < 0) \quad (3.13)$$

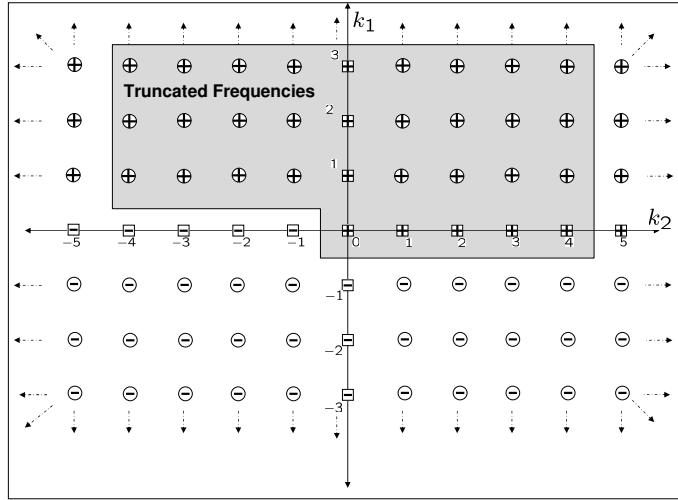


Figure 3.2: An example of box truncation with $K_1 = 3$ and $K_2 = 4$.

The set of truncated frequencies according to this scheme is usually denoted by Λ_M and formally described by

$$\begin{aligned} \Lambda_M = \{ \lambda : \lambda = |k_1\omega_1 + k_2\omega_2| \quad &|k_1, k_2 \in \mathbb{Z}; \\ &(0 \leq k_1 \leq K_1, |k_2| \leq K_2,) \text{ and } (k_1 \neq 0 \text{ if } k_2 < 0) \} \end{aligned} \quad (3.14)$$

where M here refers to the number of nonzero frequencies included in the truncation. For a box truncation with truncation indices K_1 and K_2 , it is easy to see that M will be given by

$$M = \frac{1}{2} ((2K_1 + 1)(2K_2 + 1) - 1) \quad (3.15)$$

The box truncation is useful in truncating two-tone signals, where one of the tones (say associated with ω_1) is known a-priori to be of low power and therefore generates a limited number of harmonics, while the other tone (ω_2) is stronger and might require a relatively larger number of harmonics. In this case box truncation offers a significant degree of flexibility if a larger integer value is used for K_2 while a smaller value is assigned for K_1 . A major disadvantage with box truncation, however, is that it does not allow sufficient flexibility in selecting mixing products.

3.3.2 Diamond truncation for two-tone waveforms

The diamond truncation provides better flexibility in choosing the mixing products. An example of diamond truncation is shown in Figure 3.3. Diamond truncation is typically

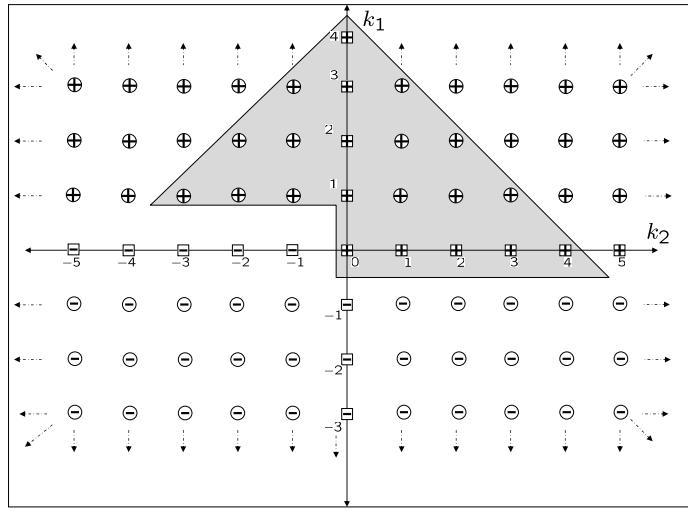


Figure 3.3: An example of diamond truncation with $K = 4$.

realized by specifying a maximum integer value K and then including in the truncated set all frequencies λ given by

$$\lambda = |k_1\omega_1 + k_2\omega_2| \quad (3.16)$$

such that k_1 and k_2 satisfy the following requirements

$$(0 \leq |k_1| + |k_2| \leq K, k_1 \geq 0) \text{ and } (k_2 \geq 0 \text{ if } k_1 = 0) \quad (3.17)$$

Formally the set of diamond truncated frequencies are given by

$$\begin{aligned} \Lambda_M = \{ \lambda: \lambda = |k_1\omega_1 + k_2\omega_2| \quad &|k_1, k_2 \in \mathbb{Z}; \\ &(|k_1| + |k_2| \leq K, k_1 \geq 0) \text{ and } (k_1 \neq 0 \text{ if } k_2 < 0) \} \end{aligned} \quad (3.18)$$

where M is the total number of nonzero frequencies in Λ_M and is given by

$$M = K^2 + K \quad (3.19)$$

3.3.3 Box and diamond truncation for multi-tone waveforms

Generalization of box and diamond truncation to the case of multi-tone signals is straightforward. For a multi-tone signal with d base frequencies $\omega_1, \omega_2, \dots, \omega_d$, the box truncation is usually represented by the following set

$$\Lambda_M = \left\{ \lambda: \lambda = \left| \sum_{l=1}^d k_l \omega_l \right|; k_l \in \mathbb{Z}; |k_l| \leq K_l, l = 1, \dots, d; \text{first nonzero } k_l \text{ positive} \right\} \quad (3.20)$$

where K_1, \dots, K_d are the truncation indices corresponding to each base frequency. The number of nonzero frequency components M in Λ_M is given by

$$M = \frac{1}{2} \left(\left(\prod_{l=1}^d (2K_l + 1) - 1 \right) \right) \quad (3.21)$$

The diamond truncation for a multi-tone waveform with d base frequencies are given by

$$\Lambda_M = \left\{ \lambda: \lambda = \left| \sum_{l=1}^d k_l \omega_l \right|; k_l \in \mathbb{Z}; \sum_{l=1}^d |k_l| \leq K; \text{first nonzero } k_l \text{ positive} \right\} \quad (3.22)$$

and the number of nonzero frequency components M in Λ_M is given by

$$M = M_{d-1} + \sum_{l_1=1}^K \sum_{l_2=0}^{K-l_1} \sum_{l_3=0}^{K-(l_1+l_2)} \dots \\ \dots \sum_{l_{d-2}=0}^{K-\sum_{u=1}^{d-3} l_u} \left(2^{\sum_{u=2}^{d-2} \delta(l_u)} \left[2 \left(K - \sum_{u=1}^{d-2} l_u \right)^2 + 2 \left(K - \sum_{u=1}^{d-2} l_u \right) + 1 \right] \right) \quad (3.23)$$

where M_{d-1} is the number of non-zero frequency components corresponding to a diamond truncation for $(d-1)$ -tone waveform with truncation index K , and

$$\delta(x) = \begin{cases} 1 & x \neq 0 \\ 0 & x = 0 \end{cases}$$

3.4. HARMONIC BALANCE ANALYSIS

Thus for $d = 2$ we get the expression given by (3.19) and for $d = 3$ (3.23) reduces to

$$M = K^2 + K + \sum_{l_1=1}^K [2(K - l_1)^2 + 2(K - l_1) + 1] \quad (3.24)$$

and for $d = 4$ (3.23) becomes

$$\begin{aligned} M = K^2 + K + \sum_{l_1=1}^K [2(K - l_1)^2 + 2(K - l_1) + 1] + \\ \sum_{l_1=1}^K \sum_{l_2=0}^{K-l_1} 2^{\delta(l_2)} [2(K - (l_1 + l_2))^2 + 2(K - (l_1 + l_2)) + 1] \end{aligned} \quad (3.25)$$

Although diamond and box truncations are general enough to handle any general situation, the total number of frequencies M can grow to exceedingly large numbers when $d = 3$ or greater. For this reason specialized truncation schemes may be advisable if not mandatory in some problems.

Given any appropriate truncation scheme, the infinite generalized Fourier representation in Eq. (3.10) can be cast in the following finite form

$$\mathbf{x}(t) = \mathbf{X}_0 + \sum_{\lambda_m \in \Lambda_M, \lambda_m \neq 0} \mathbf{X}_m^C \cos(\lambda_m t) + \mathbf{X}_m^S \sin(\lambda_m t) \quad (3.26)$$

where the summation in (3.26) is over the discrete frequencies that are elements in the set Λ_M .

3.4 Harmonic Balance Analysis

Recall that our ultimate goal in finding the steady-state operating is to calculate the Fourier series coefficients that appear in right-side of (3.26) for a general nonlinear circuits. Given that Chapter 1 established that any general nonlinear circuit can be described in the time-domain in the following set of nonlinear algebraic-differential equations,

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b}(t) \quad (3.27)$$

we can substitute from (3.26) into (3.27) and proceed to compute the coefficients $\mathbf{X}_0, \mathbf{X}_m^C, \mathbf{X}_m^S$.

Assuming that each independent source in the circuit has a multi-tone almost-periodic

waveform with a generalized Fourier series expansion

$$\mathbf{b}(t) = \mathbf{B}_0 + \sum_{\lambda_k \in \Lambda_K, \lambda_k \neq 0} \mathbf{B}_k^C \cos(\lambda_k t) + \mathbf{B}_k^S \sin(\lambda_k t) \quad (3.28)$$

over a truncated set of frequencies, Λ_K with K discrete frequency components, then we obtain after substituting from (3.28) and (3.26) into (3.27),

$$\begin{aligned} & \mathbf{G} \left(\mathbf{X}_0 + \sum_{\lambda_m \in \Lambda_M, \lambda_m \neq 0} \mathbf{X}_m^C \cos(\lambda_m t) + \mathbf{X}_m^S \sin(\lambda_m t) \right) \\ & + \mathbf{C} \left(\sum_{\lambda_m \in \Lambda_M, \lambda_m \neq 0} -\mathbf{X}_m^C \lambda_m \sin(\omega_m t) + \mathbf{X}_m^S \lambda_m \cos(\lambda_m t) \right) \\ & + \mathbf{f} \left(\mathbf{X}_0 + \sum_{\lambda_m \in \Lambda_M, \lambda_m \neq 0} \mathbf{X}_m^C \cos(\lambda_m t) + \mathbf{X}_m^S \sin(\lambda_m t) \right) \\ & = \mathbf{B}_0 + \sum_{\lambda_k \in \Lambda_K, \lambda_k \neq 0} \mathbf{B}_k^C \cos(\lambda_k t) + \mathbf{B}_k^S \sin(\lambda_k t) \end{aligned} \quad (3.29)$$

It is implicitly understood here that both Λ_K and Λ_M are constructed using the same base frequencies with $M > K$. The vector of algebraic nonlinearities can also be expanded as a generalized Fourier series

$$\begin{aligned} & \mathbf{f} \left(\mathbf{X}_0 + \sum_{\lambda_m \in \Lambda_M, \lambda_m \neq 0} \mathbf{X}_m^C \cos(\lambda_m t) + \mathbf{X}_m^S \sin(\lambda_m t) \right) = \\ & \mathbf{F}_0 + \sum_{\lambda_m \in \Lambda_M, \lambda_m \neq 0} \mathbf{F}_m^C \cos(\lambda_m t) + \mathbf{F}_m^S \sin(\lambda_m t) \end{aligned} \quad (3.30)$$

Hence if $\bar{\mathbf{X}} \in \mathbb{R}^{N(2M+1)}$ is a vector that collects all the Fourier coefficients of $\mathbf{x}(t)$,

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1^C \\ \mathbf{X}_1^S \\ \vdots \\ \mathbf{X}_M^C \\ \mathbf{X}_M^S \end{bmatrix} \quad (3.31)$$

3.4. HARMONIC BALANCE ANALYSIS

we would expect that $\mathbf{F}_0, \mathbf{F}_m^C, \mathbf{F}_m^S, m = 1, \dots, M$ are all functions of $\bar{\mathbf{X}}$. Substitution from (3.30) into (3.29) and taking advantage of the orthogonality property of the sines and cosines functions of different frequencies under adequate inner-products, yields $2M + 1$ systems of equations,

$$\begin{aligned}
& \mathbf{G}\mathbf{X}_0 + \mathbf{F}_0(\bar{\mathbf{X}}) = \mathbf{B}_0 \\
& \mathbf{G}\mathbf{X}_1^C + \lambda_1 \mathbf{C}\mathbf{X}_1^S + \mathbf{F}_1^C(\bar{\mathbf{X}}) = \mathbf{B}_1^C \\
& -\lambda_1 \mathbf{C}\mathbf{X}_1^S + \mathbf{G}\mathbf{X}_1^C + \mathbf{F}_1^S(\bar{\mathbf{X}}) = \mathbf{B}_1^S \\
& \quad \vdots \\
& \mathbf{G}\mathbf{X}_K^C + \lambda_K \mathbf{C}\mathbf{X}_K^S + \mathbf{F}_K^C(\bar{\mathbf{X}}) = \mathbf{B}_K^C \\
& -\lambda_K \mathbf{C}\mathbf{X}_K^S + \mathbf{G}\mathbf{X}_K^C + \mathbf{F}_K^S(\bar{\mathbf{X}}) = \mathbf{B}_K^S \\
& \mathbf{G}\mathbf{X}_{K+1}^C + \lambda_{K+1} \mathbf{C}\mathbf{X}_{K+1}^S + \mathbf{F}_{K+1}^C(\bar{\mathbf{X}}) = \mathbf{0} \\
& -\lambda_{K+1} \mathbf{C}\mathbf{X}_{K+1}^S + \mathbf{G}\mathbf{X}_{K+1}^C + \mathbf{F}_{K+1}^S(\bar{\mathbf{X}}) = \mathbf{0} \\
& \quad \vdots \\
& \mathbf{G}\mathbf{X}_M^C + \lambda_M \mathbf{C}\mathbf{X}_M^S + \mathbf{F}_M^C(\bar{\mathbf{X}}) = \mathbf{0} \\
& -\lambda_M \mathbf{C}\mathbf{X}_M^S + \mathbf{G}\mathbf{X}_M^C + \mathbf{F}_M^S(\bar{\mathbf{X}}) = \mathbf{0}
\end{aligned} \tag{3.32}$$

which can be formulated as big system of $N(2M + 1)$ nonlinear equations in the form

$$\bar{\mathbf{Y}}\bar{\mathbf{X}} + \bar{\mathbf{F}}(\bar{\mathbf{X}}) = \bar{\mathbf{B}} \tag{3.33}$$

where $\bar{\mathbf{Y}}$ is an $N(2M + 1) \times N(2M + 1)$ matrix

$$\bar{\mathbf{Y}} = \left[\begin{array}{ccccccccc}
\mathbf{G} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{G} & \lambda_1 \mathbf{C} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\
\mathbf{0} & -\lambda_1 \mathbf{C} & \mathbf{G} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G} & \lambda_2 \mathbf{C} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & -\lambda_2 \mathbf{C} & \mathbf{G} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{G} & \lambda_M \mathbf{C} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & -\lambda_M \mathbf{C} & \mathbf{G} & \mathbf{G}
\end{array} \right] \tag{3.34}$$

and the vector $\bar{\mathbf{F}}(\bar{\mathbf{X}})$ and $\bar{\mathbf{B}}$ are given by

$$\bar{\mathbf{F}}(\bar{\mathbf{X}}) = \begin{bmatrix} \mathbf{F}_0(\bar{\mathbf{X}}) \\ \mathbf{F}_1^C(\bar{\mathbf{X}}) \\ \mathbf{F}_1^S(\bar{\mathbf{X}}) \\ \vdots \\ \mathbf{F}_M^C(\bar{\mathbf{X}}) \\ \mathbf{F}_M^S(\bar{\mathbf{X}}) \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1^C \\ \mathbf{B}_1^S \\ \vdots \\ \mathbf{B}_M^C \\ \mathbf{B}_M^S \end{bmatrix}, \quad \mathbf{B}_m^{(C,S)} = 0 \quad \text{if } m > K \quad (3.35)$$

Hence, the steady-state operating point can be computed by solving the above system for $\bar{\mathbf{X}}$.

The similarity between the new system in (3.33) for multi-tone excitation and the single-tone system derived in (2.51) suggests that (3.33) can be solved using similar steps to that used in finding the solution for the single-tone case. That statement is only partially true. In fact the basic computational steps of the N-R method will remain the same in the multi-tone system (3.33). Specifically, during any iteration of the N-R method, the vector of nonlinear Fourier coefficients $\bar{\mathbf{F}}(\bar{\mathbf{X}})$ and its Jacobian matrix w.r.t. $\bar{\mathbf{X}}$ will have to be computed at a given trial vector $\bar{\mathbf{X}}^{(j)}$. The absence of a direct analytic expressions for $\mathbf{F}_m^{C,S}$ in terms of $\mathbf{X}_m^{C,S}$, however, necessitates using numerical approach to carry out this computation. In the case of the single-tone periodic excitation, the numerical approach used was based on using the Direct Fourier Transformation (DFT) (or its faster variation FFT). In that approach, $\bar{\mathbf{X}}^{(j)}$ was first converted to its time-domain version represented as time samples to evaluate the vector of nonlinearities at those time points and then convert that to its Frequency coefficients. In the general case of multi-tone excitation, the numerical approach remains essentially the same. Nonetheless the major difference between the multi-tone and single-tone emerges in the Fourier transformation techniques. This is a direct consequence of having the waveforms in the circuit represented by the generalized Fourier expansion in (3.26), whose frequency components are not harmonically related. Understanding the current state-of-the-art related to Fourier transforms used in multi-tone signals will take us a little deeper in the theory of Fourier transform and will dominate the forthcoming discussions in most of this chapter.

3.5 Generalized Fourier Transform

A direct, and perhaps a brute-force, approach to generalize Fourier transform to an almost-periodic signal, $x(t)$, is to deal with the almost-periodic waveforms as periodic waveforms, through figuring out the global period T_{global} for which $x(t) = x(t + T_{\text{global}})$. Typically the global period is related to the smallest frequency component in the truncated set Λ_M over

3.5. GENERALIZED FOURIER TRANSFORM

which $x(t)$ is defined, i.e.

$$T_{\text{global}} = \frac{2\pi}{\lambda_{\min}} \quad (3.36)$$

where λ_{\min} is the smallest in Λ_M ,

$$\lambda_{\min} = \inf_m \{\lambda_m \in \Lambda_M, \lambda_m \neq 0\} \quad (3.37)$$

To derive the IDFT (DFT) operator matrices for almost-periodic signals in a manner analogous to that used in Section 2.1, we will have to sample the almost-periodic waveforms, $x(t)$, (as represented by (3.26)) at H equally spaced time points. Writing this expression at those H sampling points leads to the following system of equations,

$$\begin{bmatrix} x(t_0) \\ x(t_1) \\ \vdots \\ x(t_{H-1}) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \cos(\lambda_1 t_0) & \sin(\lambda_1 t_0) & \cdots & \cos(\lambda_M t_0) & \sin(\lambda_M t_0) \\ 1 & \cos(\lambda_1 t_1) & \sin(\lambda_1 t_1) & \cdots & \cos(\lambda_M t_1) & \sin(\lambda_M t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\lambda_1 t_{H-1}) & \sin(\lambda_1 t_{H-1}) & \cdots & \cos(\lambda_M t_{H-1}) & \sin(\lambda_M t_{H-1}) \end{bmatrix}}_{\mathbf{\Gamma}_G^{-1}} \begin{bmatrix} X_0 \\ X_1^C \\ X_1^S \\ \vdots \\ X_M^C \\ X_M^S \end{bmatrix} \quad (3.38)$$

We refer to the matrix on the right-side of (3.38) as the Generalized IDFT (G-IDFT) operator matrix ^{II}. In order to make the above system square, the number of sampling points will have to be given by

$$H = 2M + 1 \quad (3.39)$$

Nevertheless, there is one more important requirement that H to has to satisfy. This requirement arises from having to avoid the aliasing error. In order to avoid the aliasing error, $x(t)$ will have to be sampled at a *rate* higher than or equal to twice the largest frequency component in Λ_M . Thus if λ_{\max} is the largest frequency component, i.e.,

$$\lambda_{\max} = \sup_m \{\lambda_m \in \Lambda_M\} \quad (3.40)$$

then the sampling step size must be

$$\Delta t_i = t_i - t_{i-1} \leq \frac{2\pi}{\lambda_{\max}} \quad (3.41)$$

^{II}Note that $\mathbf{\Gamma}_G^{-1}$ on the right side of (3.38), is the generalized form of the IDFT operator matrix, $\mathbf{\Gamma}$, used for periodic single-tone waveform. It is easy to verify that $\mathbf{\Gamma}_G^{-1} = \mathbf{\Gamma}$ if $\lambda_m = m\lambda_0$, and $H = 2M + 1$

To satisfy the above requirement H will have to be given by

$$H \geq 2 \frac{\lambda_{\max}}{\lambda_{\min}} \quad (3.42)$$

Naturally to satisfy (3.42) and still keep the system (3.38) square, we would have to increase (or decrease) M to H as specified by (3.39). Note that the requirement on H was automatically satisfied in the case of periodic single-tone signals just by setting $H = 2M + 1$, since in that case we had $\lambda_{\min} = \lambda_0$, and $\lambda_{\max} = M\lambda_0$ for some fundamental λ_0 . The situation in almost-periodic waveforms is quite different. For example, consider an almost-periodic waveform with only two base frequencies ω_1 and ω_2 , where it can be easily shown that $\lambda_{\min} = |\omega_2 - \omega_1|$. Here H must be

$$H \geq 2 \frac{\lambda_{\max}}{|\omega_2 - \omega_1|} \quad (3.43)$$

This shows that the closer the base frequencies to each other, the higher the number of sampling points will be, and consequently the larger the number of Fourier coefficients will have to be taken into account. In practice, though, circuit designers are typically interested in calculating few low frequency components.

Nonetheless, this is not the only difficulty that arises from that brute-force approach to generalizing the Fourier transform to almost-periodic signals. There is another more subtle and numerically more devastating aspect to this approach that can potentially make all the computations totally inaccurate and practically useless.

Before embarking on illustrating this aspect, we put the system in (3.38) in a more compact form

$$\mathbf{x} = \boldsymbol{\Gamma}_G^{-1} \mathbf{X} \quad (3.44)$$

where $\mathbf{x} = [x_0, x_1, \dots, x_{H-1}]^T$, $\mathbf{X} = [X_0, X_1^C, X_1^S, \dots, X_M^C, X_M^S]^T$, and $x_k = x(t_k)$. If we let $\boldsymbol{\Gamma}_G$ be the inverse of $\boldsymbol{\Gamma}_G^{-1}$, then (3.44) is invariably given by,

$$\mathbf{X} = \boldsymbol{\Gamma}_G \mathbf{x} \quad (3.45)$$

It is perhaps tempting, looking at (3.44) and (3.45), to take the matrices $\boldsymbol{\Gamma}_G^{-1}$ and $\boldsymbol{\Gamma}_G$, once they are properly constructed with sufficient sampling as the G-IDFT and G-DFT operator matrices, where conversion between time- and frequency-domain is simply a matter of matrix multiplication with frequency- and time-domain vectors. This is a rather simplistic view, as a huge source of error sets in. Studying this source of error requires further investigation into the aliasing error.

The aliasing error results essentially from truncating the spectrum of signal with infinite spectrum. Notice this is typically the situation we have in HB analysis, where an infinite spectral signal is represented by a truncated set of frequencies. Therefore, it is inevitable

that $x(t)$ will have frequency components outside Λ_M . In truncating an infinite spectrum signal to a limited band, frequencies that lie outside the truncated spectrum masquerade as frequencies within Λ_M , and thereby end up corrupting the truncated spectrum. Usually this error can be controlled if the magnitude of the frequencies not in Λ_M are negligibly small. Yet this small error, can potentially undergo an amplification process each time a transformation is performed. The amplification mechanism that kicks in during the transformation is powerful enough to make the originally unnoticeable error due to aliasing a very troublesome source of error.

Interestingly enough, this amplification mechanism is kept at minimum if the signal is periodic, i.e., when the matrix Γ^{-1} is being used in the transformation. For almost-periodic waveforms, however, it can grow unbounded due to certain properties of the transformation matrix Γ_G^{-1} . A deeper insight into this issue require scratching the surface of some ideas in matrix theory related to the so-called condition number of a matrix. Before delving into this issue, let us denote the error due to aliasing in the vector of time points by $\delta\mathbf{x}$. If \mathbf{x} is contaminated by $\delta\mathbf{x}$, then \mathbf{X} obtained through the transformation (3.45) would also be contaminated by another error $\delta\mathbf{X}$, so that,

$$(\mathbf{X} + \delta\mathbf{X}) = \Gamma_G (\mathbf{x} + \delta\mathbf{x}) \quad (3.46)$$

The relation between $\delta\mathbf{X}$ and $\delta\mathbf{x}$ goes through the condition number of the matrix Γ_G^{-1} and is given by theorem that shows that

$$\frac{\|\delta\mathbf{X}\|_2}{\|\mathbf{X}\|_2} \leq \kappa(\Gamma_G^{-1}) \frac{\|\delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \quad (3.47)$$

where $\|\cdot\|_2$ is the 2-norm and $\kappa(\Gamma_G^{-1})$ is the condition number of the matrix Γ_G^{-1} defined by

$$\kappa(\Gamma_G^{-1}) = \|\Gamma_G^{-1}\|_2 \|\Gamma_G\|_2 \quad (3.48)$$

It can be shown that the minimum value for the condition number of a matrix is attained when the columns of the matrix are orthogonal with unity norm. If the columns are orthogonal but with different norms, κ will still be given by a certain constant value. On the other hand, it can grow unbounded for general matrices, and can reach infinity, if at least two columns are linearly dependent. Given that the columns of the IDFT operator matrix Γ^{-1} of periodic waveforms are orthogonal, the condition number $\kappa(\Gamma^{-1})$ is given by a well-defined *constant* number, given by,

$$\kappa(\Gamma^{-1}) = \sqrt{2} \quad (3.49)$$

which makes the error $\delta\mathbf{X}$ resulting from a given transformation strictly proportional to $\delta\mathbf{x}$, the original aliasing error. Unfortunately, for almost-periodic waveforms, the transformation operator Γ_G^{-1} does not have this nice property. In fact the condition number can

reach very high levels. The reason for this is a result of the fact that the columns of the matrix Γ_G^{-1} become almost linearly-dependent if the sampling points are equally spaced and at least two base frequencies are close to each other. This property Γ_G^{-1} is frequently encountered in simulation of RF circuits, and causes a huge error especially that Fourier transformation is invoked repeatedly in the course of HB analysis. This problem in the Fourier transformation of almost-periodic waveforms represents one of the major source of difficulties in HB analysis of circuits with multi-tone excitation. Several approaches have been proposed to address this issue. Those approaches can be classified under three main categories,

- Multi-dimensional Fourier Transformation
- Almost-Periodic Fourier transformation
- Frequency-Mapping techniques

The following text will summarize the basic ideas behind the first two approaches and present a detailed description for the third one.

3.5.1 Multi-Dimensional Fourier Transform (MDFT)

MDFT techniques are only applicable to multi-dimensional waveforms. Given that all waveforms in the HB analysis are single dimensional in the variable t , a preprocessing step is necessary to adapt those waveforms for MDFT techniques. The adaptation process is simply carried out through replacing the HB waveform, which is essentially multi-tone waveform defined over d base tone frequencies with generalized Fourier expansion of the form (3.26), by an auxiliary d -dimensional function $\hat{x}(\tau_1, \dots, \tau_d)$ such that

$$\hat{x}(\tau_1, \dots, \tau_d)|_{\tau_1=\tau_2=\dots=\tau_d=t} = x(t) \quad (3.50)$$

The above condition can be satisfied only if $x(t)$ is defined over a box truncation set, in which case $x(t)$ is explicitly given by

$$x(t) = \sum_{|\sum_{l=1}^d k_l \omega_l| \in \Lambda_M} \left[X_{k_1, k_2, \dots, k_d}^C \cos \left(\left(\sum_{l=1}^d k_l \omega_l \right) t \right) \right. \\ \left. X_{k_1, k_2, \dots, k_d}^S \sin \left(\left(\sum_{l=1}^d k_l \omega_l \right) t \right) \right] \quad (3.51)$$

$|k_l| \leq K_l; l = 1, \dots, d; \quad \text{first nonzero } k_l \text{ positive}$

3.5. GENERALIZED FOURIER TRANSFORM

$\hat{x}(\tau_1, \dots, \tau_d)$ can be defined as

$$\begin{aligned}\hat{x}(\tau_1, \dots, \tau_d) = & \sum_{\left| \sum_{l=1}^d k_l \omega_l \right| \in \Lambda_M} \left(X_{k_1, k_2, \dots, k_d}^C \cos \left(\sum_{l=1}^d k_l \omega_l \tau_l \right) \right. \\ & \left. X_{k_1, k_2, \dots, k_d}^S \sin \left(\sum_{l=1}^d k_l \omega_l \tau_l \right) \right) \\ & |k_l| \leq K_l; l = 1, \dots, d;\end{aligned}\quad (3.52)$$

Clearly the choice of $\hat{x}(\tau_1, \dots, \tau_d)$ in (3.52) satisfies (3.50). $\hat{x}(\tau_1, \dots, \tau_d)$ is a multi-dimensional function and its Fourier coefficients can be obtained using the efficient MDFT techniques. Furthermore, those Fourier coefficients are the same Fourier coefficients of $x(t)$. Therefore, MDFT techniques can be used as the Fourier transform vehicle to convert from time- to frequency-domain and vice-versa.

MDFT techniques have the same attractive features of DFT techniques for one-dimensional single-tone periodic waveforms. In particular, an Inverse MDFT (I-MDFT) matrix operator with orthogonal columns can be constructed, similar to the matrix Γ^{-1} , leading to minimum error in the transformation between time- and frequency-domains. Furthermore, efficient FFT techniques can be extended to the multi-dimensional setting to avoid the matrix-vector multiplication used in classical DFT.

The main disadvantage, however, is the higher computational cost which comes as a consequence of having to use multi-dimensional discretization grid for the auxiliary function $\hat{x}(\tau_1, \dots, \tau_d)$. For example, in the special case of a 2-d grid, and assuming that the number of discretization points is S_1 and S_2 in each dimension, the corresponding I-MDFT matrix will be $(S_1 S_2) \times (S_1 S_2)$ matrix, thus requiring a $(S_1 S_2)^2$ complex multiplications in matrix-vector operation each time a transformation is performed. For $S_1 = S_2 = 32$, this equivalent to 1,048,567.

More efficient techniques have also been used proposed to reduce the computational cost for MDFT techniques. The approach presented in [2] is based on the observation that the multi-dimensional auxiliary $\hat{x}(\tau_1, \dots, \tau_d)$ is periodic in τ_i with period $2\pi/\omega_i$ if all other τ_l 's are held constant. With this techniques significant saving in computational cost is achieved. For example, in the case of a 2-d MDFT with S_1 and S_2 sampling points, the number of multiplications is reduced to $S_1 S_2 (S_1 + S_2)$. For $S_1 = S_2 = 32$ that number becomes 65,536. A further substantial improvement is also obtained if the efficient techniques of FFT are employed. In a 2-d MDFT using FFT technique, the number of multiplications is reduced to $(S_1 S_2 / 2) \log_2 (S_1 S_2)$. This yields 5,120 for $S_1 = S_2 = 32$. Nevertheless the computational cost increases exponentially for MDFT techniques for higher dimensional.

3.5.2 Almost-Periodic Fourier Transform (APFT)

The most common characteristic that APFT techniques share is that they aim at improving the condition number of the G-IDFT operator matrix Γ_G^{-1} through special numerical algorithms. Recall from the above discussion that the condition number worsens when at least two base frequencies are closely spaced, while the sampling time points are equally spaced. The central question that all APFT-based techniques try to answer is the following “*Given that one has no control over selecting the base frequencies, which are dictated by the sources in the circuit, is it possible to select a different set of sampling time points to improve the condition number?*”. It turned out that this question can be best addressed through using a set of nonuniformly distributed sampling points.

The earliest approach in this category appeared in 1984 by Chua and Uhida [3], where the problem of high condition number is addressed by using extra time-domain sampling points than required. In this case $H > 2M + 1$ and the G-IDFT operator matrix Γ_G^{-1} becomes a tall rectangular matrix. To make the system (3.44) square again, both sides are multiplied by $(\Gamma_G^{-1})^T$ which results in

$$(\Gamma_G^{-1})^T \mathbf{x} = (\Gamma_G^{-1})^T (\Gamma_G^{-1}) \mathbf{X} \quad (3.53)$$

and the Fourier spectrum \mathbf{X} is thus obtained from

$$\mathbf{X} = \left((\Gamma_G^{-1})^T (\Gamma_G^{-1}) \right)^{-1} (\Gamma_G^{-1})^T \mathbf{x} \quad (3.54)$$

The solution obtained above is not the exact solution but it is the optimum in the least-squares-sense.

The technique presented by Sorkin and Kundert [4–6] approaches the task of constructing the G-IDFT operator matrix through generating a pool of sample points and selecting the best $2M + 1$ points that make the columns of Γ_G^{-1} nearly orthogonal. The algorithm used to achieve this goal is called “Near Orthogonal APFT”. Its basic idea is to initially generate the matrix Γ_G^{-1} by choosing twice as many time points as needed. In this way Γ_G^{-1} becomes $2(2M + 1) \times (M + 1)$ ($\in \mathbb{R}^{2(2M+1) \times (M+1)}$) matrix with $2(2M + 1)$ rows generated at time points selected randomly in a period of time $0 \leq t \leq 3T_{\text{global}}$. The algorithm then processes those rows using the Modified Gram-Schmidt process to select the most orthogonal $(2M + 1)$ rows. This leads to a significant improvement in the condition number. Another technique was proposed in [7] to generate an exactly orthogonal Γ_G^{-1} . This algorithm, called “Orthogonal APFT” can achieve the theoretically lowest condition number. The reader is referred to [8] for a list of other APFT-based techniques and comparison between their performances.

The basic advantage found in APFT techniques is the simplicity of implementation. However, the main disadvantage is that they can not be implemented using the more efficient FFT-based techniques.

3.5.3 Frequency Mapping Techniques (FMT)

Frequency mapping techniques are perhaps the most efficient and commonly used in the HB analysis of multi-tone circuits. The main idea behind the mapping techniques stems from two basic observations. The first observation is that in HB analysis, we only need to compute the Fourier spectra of the nonlinear function vector $f(\mathbf{x}(t))$ ($\mathbf{F}_0, \mathbf{F}_m^{(C,S)}, m = 1, \dots, M$) given the spectra of the MNA variables $\mathbf{x}(t)$ ($\mathbf{X}_0, \mathbf{X}_m^{(C,S)}, m = 1, \dots, M$); the actual time-domain waveforms for $\mathbf{x}(t)$ and $f(\mathbf{x}(t))$ are inconsequential, except perhaps at the end of the analysis when a solution has been reached and the user may be interested in visualizing the time-domain waveform^{III}. Apart from that, computing the time-domain waveforms in $\mathbf{x}(t)$ and $f(\mathbf{x}(t))$ is only performed as an intermediate step to reach $\mathbf{F}_0, \mathbf{F}_m^{(C,S)}, m = 1, \dots, M$ starting from $\mathbf{X}_0, \mathbf{X}_m^{(C,S)}, m = 1, \dots, M$.

The second observation is that the Fourier coefficients of any algebraic nonlinearity in the vector $f(\mathbf{x}(t))$ are independent of the frequency components of the set Λ_M over which $\mathbf{x}(t)$ is being defined. We illustrate the latter observation with an example, where we consider a scalar $f(x(t))$ given by

$$f(x(t)) = x(t) + (x(t))^2 \quad (3.55)$$

We then let $x(t)$ be an almost-periodic waveform with only two base frequencies ω_1 and ω_2 , and for further simplicity assume it is given by

$$x(t) = a_1 \cos(\omega_1 t) + a_2 \cos(\omega_2 t) \quad (3.56)$$

Substituting from (3.56) into (3.55) and expanding we get

$$\begin{aligned} f(x(t)) &= \frac{1}{2} (a_1^2 + a_2^2) + a_1 \cos(\omega_1 t) + a_2 \cos(\omega_2 t) \\ &\quad + \frac{a_1^2}{2} \cos(2\omega_1 t) + \frac{a_2^2}{2} \cos(2\omega_2 t) \\ &\quad + a_1 a_2 \cos((\omega_1 + \omega_2)t) + a_1 a_2 \cos((\omega_1 - \omega_2)t) \end{aligned} \quad (3.57)$$

Thus the set Λ_M for $f(x(t))$ is given by

$$\Lambda_M = \{ \omega_1, \omega_2, |\omega_1 - \omega_2|, \omega_1 + \omega_2, 2\omega_1, 2\omega_2 \} \quad (3.58)$$

^{III}At that point the one can use some expensive and accurate technique such as MDFT.

and $M = 6$.^{IV} The Fourier coefficients for $f(x(t))$ are thus given by,

$$\begin{aligned}
 F_0 &= \frac{1}{2} (a_1^2 + a_2^2) \\
 F_1^C &= a_1 \quad F_1^S = 0 \\
 F_2^C &= a_1 \quad F_2^S = 0 \\
 F_3^C &= a_1 a_2 \quad F_3^S = 0 \\
 F_4^C &= a_1 a_2 \quad F_4^S = 0 \\
 F_5^C &= \frac{a_1^2}{2} \quad F_5^S = 0 \\
 F_6^C &= \frac{a_1^2}{2} \quad F_6^S = 0
 \end{aligned} \tag{3.59}$$

The important point to highlight in (3.59) is that all Fourier coefficients of $f(x(t))$ are independent of ω_1 and ω_2 and depend only on the values of the Fourier coefficients of $x(t)$. This result shows that we can change the values of ω_1 and ω_2 with no effect on the Fourier coefficients of $f(x(t))$. The only parameters that would change in this process is the actual waveforms for both of $x(t)$ and $f(x(t))$. In fact, both waveforms will not resemble any physical waveform in the circuit. But, given that we are not concerned with the temporal shapes of $x(t)$ or $f(x(t))$, as it changes during the HB iteration, then why not use the following mapping and replace ω_1 and ω_2 using

$$\omega_1 \rightarrow 2\lambda_0 \tag{3.60}$$

$$\omega_2 \rightarrow 3\lambda_0 \tag{3.61}$$

for some “artificial” frequency $\lambda_0 > 0$. The main advantage of this mapping will become clear shortly. Clearly, the above mapping will transform the original waveforms to new waveforms, which will be referred to henceforth by $x_m(t)$ and $f_m(x_m(t))$, given by

$$x_m(t) = a_1 \cos(2\lambda_0 t) + a_2 \cos(3\lambda_0 t) \tag{3.62}$$

$$\begin{aligned}
 f_m(x_m(t)) &= \frac{1}{2} (a_1^2 + a_2^2) + a_1 \cos(2\lambda_0 t) + a_2 \cos(3\lambda_0 t) \\
 &\quad + \frac{a_1^2}{2} \cos(4\lambda_0 t) + \frac{a_2^2}{2} \cos(6\lambda_0 t) \\
 &\quad + a_1 a_2 \cos(5\lambda_0 t) + a_1 a_2 \cos(\lambda_0 t)
 \end{aligned} \tag{3.63}$$

^{IV}Notice here that Λ_M represents a diamond truncation set with $K = 2$.

3.5. GENERALIZED FOURIER TRANSFORM

It is easily seen from (3.62) and (3.63) that $x_m(t)$ and $f_m(x_m(t))$ are periodic, with period $2\pi/\lambda_0$.

The equivalence between the Fourier coefficients of $f(x(t))$ and $f_m(x_m(t))$, coupled with the fact that both $x_m(t)$ and $f_m(x_m(t))$ are single-tone periodic waveforms, suggests that we can compute the Fourier coefficients of $f(x(t))$ via one-dimensional simple DFT techniques or using its faster version of FFT version. The only requirement we need to fulfill is to be able to recover those coefficients from the Fourier coefficients of $f_m(x_m(t))$. This should be achieved easily if we keep in mind the inverse mapping between the artificial frequency-domain and the original multi-tone domain. In this simple example, the inverse mapping recovers the Fourier coefficients of the original domain from the mapped domain as follows,

Mapped Spectrum	Original Spectrum
Fourier coefficient of $\{0\}$	\rightarrow Fourier coefficient of $\{0\}$
Fourier coefficient of $\{\lambda_0\}$	\rightarrow Fourier coefficient of $\{ \omega_1 - \omega_2 \}$
Fourier coefficient of $\{2\lambda_0\}$	\rightarrow Fourier coefficient of $\{\omega_1\}$
Fourier coefficient of $\{3\lambda_0\}$	\rightarrow Fourier coefficient of $\{\omega_2\}$
Fourier coefficient of $\{4\lambda_0\}$	\rightarrow Fourier coefficient of $\{2\omega_1\}$
Fourier coefficient of $\{5\lambda_0\}$	\rightarrow Fourier coefficient of $\{\omega_1 + \omega_2\}$
Fourier coefficient of $\{6\lambda_0\}$	\rightarrow Fourier coefficient of $\{2\omega_2\}$

We summarize the central steps to the frequency mapping technique as follows.

1. Find an artificial frequency λ_0 such that each element in Λ_M corresponds to a **unique integer** multiple of λ_0 . Refer to the mapped spectrum (the multiples of λ_0) as Ω_M , where

$$\Omega_M = \{\omega : \omega = h\lambda_0; h = 0, 1, 2, \dots, M\} \quad (3.64)$$

Thus for every $\lambda_m \in \Lambda_M$ there is a unique positive integer p , such that

$$\lambda_m \Leftrightarrow p\lambda_0 \quad (3.65)$$

2. Associate with each element in Ω_M the Fourier coefficients of the corresponding element in Λ_M . The waveform constructed from frequencies in Ω_M and those Fourier coefficients constitute a periodic waveform with the period $2\pi/\lambda_0$, and is denoted by $x_m(t)$.

3. Using IFFT, compute the time-domain samples of $x_m(t)$ at $2M + 1$ time points.
4. Evaluate the nonlinearities in the vector $f(x_m(t))$ at each of the above points. Denote the computed waveform by $f_m(t)$.
5. Use FFT to compute the Fourier coefficients of $f_m(t)$.

6. Using the inverse mapping between Ω_M and Λ_M recover the Fourier coefficients of $f(x(t))$ from those of $f_m(t)$.

The above steps describe in a very generic way the essential methodology of the frequency mapping technique. Nonetheless, it comes short on many of the specific implementation issues. It leaves the reader entertaining the following questions.

- What is the artificial frequency λ_0 that can produce the one-to-one mapping between elements of Λ_M and elements of Ω_M ?
- Is it always possible to derive the required mapping for every type of truncated set (diamond or box truncation), and for any number of base frequency?
- How can we derive the inverse mapping which will be used to recover the desired multi-tone spectrum from the artificial one?
- How can the mapping technique described above be used to yield, not only the values of the desired spectrum ($F_0, F_m^{(C,S)}$), but also the Jacobian matrices of those spectral coefficients w.r.t. the spectral coefficients of $x(t)$ ($X_0, X_m^{(C,S)}$)

We address the above concerns through showing how to adapt different truncation schemes to FMT.

Box truncation with two base frequencies

Consider a box-truncated set Λ_M with two base frequencies, ω_1 and ω_2 and truncation indices K_1 and K_2 . This set was described earlier by (3.14). Now consider the following mapping for the base frequencies

$$\begin{aligned}\omega_1 &\rightarrow \lambda_0 \\ \omega_2 &\rightarrow (1 + 2K_1)\lambda_0\end{aligned}\tag{3.66}$$

for some arbitrary $\lambda_0 > 0$. It is easy to verify that under this mapping, each element in Λ_M is mapped to a unique integer multiple of λ_0 . Thus each element in the original spectrum Λ_M is mapped to a unique element in the artificial spectrum Ω_M . The uniqueness property is illustrated further in Figure 3.4.

The relation between any element in Λ_M and the corresponding element in Ω_M can be derived as follows. Any element in Λ_M is described by $|k_1\omega_1 + k_2\omega_2|$, where $0 \leq k_1 \leq K_1; |k_2| \geq K_2$ and $k_2 \geq 0$ if $k_1 = 0$. Thus using the mapping of (3.66), an arbitrary element in Λ_M can be mapped as shown next,

$$|k_1\omega_1 + k_2\omega_2| \rightarrow \underbrace{|k_1 + k_2(1 + 2K_1)|}_{\text{integer } p} \lambda_0\tag{3.67}$$

3.5. GENERALIZED FOURIER TRANSFORM

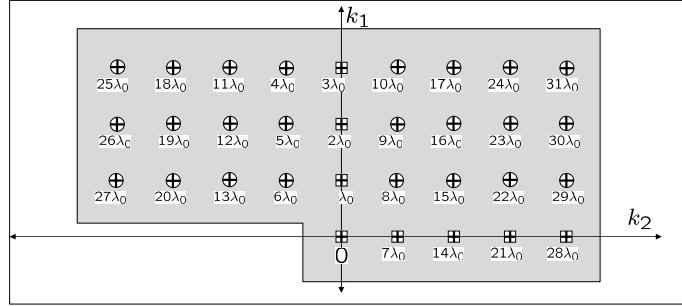


Figure 3.4: Illustrating the mapping technique for a box-truncated set with two base frequencies and truncation indices $K_1 = 3$ and $K_2 = 4$. Note that each element in the truncated set Λ_M corresponds to a unique element in the mapped spectrum Ω_M .

Since the value of λ_0 is arbitrary, we can choose it to be $\lambda_0 = 1\text{rad/sec.}$, for simplicity. Therefore, the relation between p , k_1 , and k_2 is given by

$$p = |k_1 + k_2 (2K_1 + 1)| \quad (3.68)$$

Table 3.1 uses the above relation to list elements in Ω_M and the corresponding elements in Λ_M .

Box truncation with multiple base frequencies

The frequency mapping technique can be extended to a general box-truncated Λ_M with d base frequencies $\omega_1, \omega_2, \dots, \omega_d$. This is achieved by mapping the base frequencies as follows,

$$\begin{aligned} \omega_1 &\rightarrow \lambda_1 \\ \omega_2 &\rightarrow \lambda_2 \\ &\vdots \\ \omega_d &\rightarrow \lambda_d \end{aligned} \quad (3.69)$$

where λ_1 is an arbitrary *positive* number, and

$$\begin{aligned} \lambda_2 &\rightarrow (1 + 2K_1)\lambda_1 \\ \lambda_3 &\rightarrow (1 + 2K_2)\lambda_2 \\ &\vdots \\ \lambda_d &\rightarrow (1 + 2K_{d-1})\lambda_{d-1} \end{aligned} \quad (3.70)$$

Table 3.1: Illustrating the individual mapping between elements in the artificial spectrum Ω_M and the original spectrum Λ_M , of a box truncated set with truncation indices $K_1 = 3$ and $K_2 = 4$.

index	Ω_M		Λ_M		
	p	ω	k_1	k_2	ω
0	0	0	0	0	0
1	1	λ_0	1	0	ω_1
2	2	$2\lambda_0$	2	0	$2\omega_1$
3	3	$3\lambda_0$	3	0	$3\omega_1$
4	7	$7\lambda_0$	0	1	ω_2
5	8	$8\lambda_0$	1	1	$\omega_1 + \omega_2$
6	9	$9\lambda_0$	2	1	$2\omega_1 + \omega_2$
7	10	$10\lambda_0$	3	1	$3\omega_1 + \omega_2$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
31	31	$31\lambda_0$	3	4	$3\omega_1 + 4\omega_2$

The following relation is used to move between the original and mapped spectra

$$k = \frac{1}{\lambda_1} \left| \sum_{l=1}^d k_l \omega_l \right| \quad (3.71)$$

Diamond truncation with two base frequencies

Consider a diamond-truncated set Λ_M with two base frequencies ω_1 and ω_2 and truncation index K . Elements in such a set have been described earlier in (3.18). Now consider the following mapping

$$\begin{aligned} \omega_1 &\rightarrow K\lambda_0 \\ \omega_2 &\rightarrow (K+1)\lambda_0 \end{aligned} \quad (3.72)$$

for some arbitrary $\lambda_0 > 0$. The reader can verify that the above mapping ensures a one-to-one correspondence between elements in Λ_M and the set Ω_M defined as

$$\Omega_M = \{\omega : p\lambda_0; p = 0, 1, 2, \dots, M\} \quad (3.73)$$

3.5. GENERALIZED FOURIER TRANSFORM

Figure 3.5 demonstrates this uniqueness for a diamond truncation with index $K = 4$, which induces the mapping

$$\begin{aligned}\omega_1 &\rightarrow 4\lambda_0 \\ \omega_1 &\rightarrow 5\lambda_0\end{aligned}\tag{3.74}$$

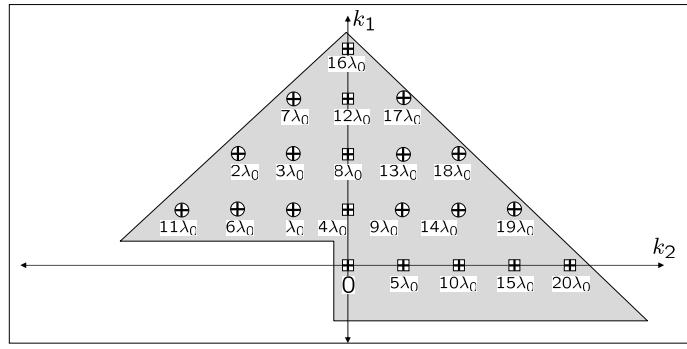


Figure 3.5: Illustrating the mapping technique for a diamond-truncated set with two base frequencies and truncation index $K = 4$. Note that each element in the truncated set Λ_M corresponds to a unique element in the mapped spectrum Ω_M .

The relation between the artificial and original spectra can be derived by noting that an arbitrary element in the original spectrum (diamond-truncated set) Λ_M is given by $|k_1\omega_1 + k_2\omega_2|$ where k_1 and k_2 satisfy (3.17). Thus using the mapping proposed in (3.72) shows that,

$$|k_1\omega_1 + k_2\omega_2| \rightarrow \underbrace{|k_1K + k_2(K+1)|}_{\text{integer } p} \lambda_0\tag{3.75}$$

and the desired relation follows directly as follows

$$p = |k_1K + k_2(K+1)|\tag{3.76}$$

Diamond truncation with multiple base frequencies

There does not seem to be explicit solutions for a diamond truncated set with more than two base frequencies. It was suggested in [9] that artificial base frequencies can be determined numerically for each value of M when there is more than two base frequencies.

3.5.4 Implementation Procedures for FMT

Having established the main underlying concept behind the FMT, we now turn to describe a step-by-step implementation procedure to compute $\bar{\mathbf{F}}(\bar{\mathbf{X}})$ and $\partial\bar{\mathbf{F}}/\partial\bar{\mathbf{X}}$ at given trial vector, $\bar{\mathbf{X}} = \bar{\mathbf{X}}^{(j)}$ during the j -th NR iteration.

Computing $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$

To begin with, we note that $\bar{\mathbf{X}}^{(j)}$ is a vector ordered by a harmonic-major indexing scheme, and therefore the Fourier coefficients of the different MNA variables are scattered in $\bar{\mathbf{X}}^{(j)}$. To collect the Fourier coefficients of every MNA variable, we will need to reorder the components of $\bar{\mathbf{X}}^{(j)}$ in a node-major indexing whereby all the Fourier coefficients of every MNA variable will be grouped together in subvectors of size $(2M + 1)$, where M is the number of non-zero frequencies in Λ_M . The resulting vector is therefore denoted by $\bar{\mathbf{X}}_{\text{node}}^{(j)}$, and can be mathematically described by

$$\bar{\mathbf{X}}_{\text{node}}^{(j)} = \mathcal{P}\bar{\mathbf{X}}^{(j)} \quad (3.77)$$

where \mathcal{P} is the permutation matrix used in Chapter 2 to convert from harmonic-major to node-major indexing modes. Thus $\bar{\mathbf{X}}_{\text{node}}^{(j)}$ contains the $2M + 1$ Fourier coefficients for the n -th MNA variable as its n -th subvector. It is to be stressed that those coefficients represent the Fourier coefficients of the original frequency spectrum in Λ_M , i.e., those that are composed from the linear combinations of the base frequencies.

To implement the idea of FMT, those $(2M + 1)$ coefficients (for each MNA variable) will have to be reordered such that their new ordering scheme reflects the artificial frequency mapping. The reordering process may be described concisely through multiplying each subvector in $\bar{\mathbf{X}}_{\text{node}}^{(j)}$ by an appropriate $(2M + 1) \times (2M + 1)$ permutation matrix, denoted here by \mathcal{P}_F . The transformation needed for that purpose can be described through multiplying $\bar{\mathbf{X}}_{\text{node}}^{(j)}$ by a block diagonal matrix that has the matrix \mathcal{P}_F as its diagonal blocks. If that block diagonal matrix is denoted by $\bar{\mathcal{P}}_F$ and the resulting vector is denoted by $\bar{\mathbf{X}}_{\text{node, map}}^{(j)}$, then

$$\bar{\mathbf{X}}_{\text{node, map}}^{(j)} = \bar{\mathcal{P}}_F \bar{\mathbf{X}}_{\text{node}}^{(j)} \quad (3.78)$$

Hence, $\bar{\mathbf{X}}_{\text{node, map}}^{(j)}$ has its subvectors ordered according to the artificial frequency mapping, i.e. corresponding to harmonically related frequencies, and thus are ready to be acted upon through the IDFT operator discussed earlier in Chapter 2, using the IDFT operator matrix

3.5. GENERALIZED FOURIER TRANSFORM

Γ^{-1} . The result of this operation can be described collectively by

$$\bar{\mathbf{x}}_{\text{node, map}}^{(j)} = \bar{\Gamma}^{-1} \bar{\mathbf{X}}_{\text{node, map}}^{(j)} \quad (3.79)$$

In (3.79), $\bar{\mathbf{x}}_{\text{node, map}}^{(j)}$ is a vector that contains $N(2M + 1)$ time-domain samples, where N is assumed to be the number of MNA variables in the circuit. In other words, the n -th subvector (whose size is $2M + 1$) within $\bar{\mathbf{x}}_{\text{node, map}}^{(j)}$ collects the $2M + 1$ time samples for the n -th MNA periodical waveform. The matrix $\bar{\Gamma}^{-1}$ has been described earlier in Chapter 2 as a block diagonal matrix having the matrices Γ^{-1} as its diagonal blocks.

Substituting the time samples of $\bar{\mathbf{x}}_{\text{node, map}}^{(j)}$ in the nonlinear function vector $\mathbf{f}(\mathbf{x}(t))$ results in another vector of time samples that describe the waveforms of the components of $\mathbf{f}(\mathbf{x}(t))$. We denote the resulting vector by $\bar{\mathbf{f}}_{\text{node, map}}^{(j)}$, and emphasize that it is ready to be processed using the DFT operator matrix Γ . This operation is described through a multiplication with the DFT block-diagonal matrix $\bar{\Gamma}$ as follows

$$\bar{\mathbf{F}}_{\text{node, map}} = \bar{\Gamma} \bar{\mathbf{f}}_{\text{node, map}}^{(j)} \quad (3.80)$$

Note that the Fourier coefficients in $\bar{\mathbf{F}}_{\text{node, map}}$ are ordered according to the artificial frequencies in Ω_M . Therefore, we need to restore their ordering based on the original frequency spectrum Λ_M . This process is performed by *undoing* the (original-to-artificial) ordering performed in (3.78) through multiplying $\bar{\mathbf{F}}_{\text{node, map}}$ by $\bar{\mathcal{P}}_F^T$,

$$\bar{\mathbf{F}}_{\text{node}} = \bar{\mathcal{P}}_F^T \bar{\mathbf{F}}_{\text{node, map}} \quad (3.81)$$

The final step needed to complete the computation of $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ is just to put $\bar{\mathbf{F}}_{\text{node}}$ in harmonic-major ordering mode through multiplying it by \mathcal{P}^T . The entire process for computing $\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)})$ can be described in a more compact form as follows

$$\bar{\mathbf{F}}(\bar{\mathbf{X}}^{(j)}) \equiv \mathcal{P}^T \bar{\mathcal{P}}_F^T \bar{\Gamma} \bar{\mathbf{f}}(\bar{\Gamma}^{-1} \bar{\mathcal{P}}_F \mathcal{P} \bar{\mathbf{X}}^{(j)}) \quad (3.82)$$

Computing the Jacobian term $\partial \bar{\mathbf{F}} / \partial \bar{\mathbf{X}}$

Calculating the Jacobian term $\partial \bar{\mathbf{F}} / \partial \bar{\mathbf{X}}$ through following the chain rule of differentiation and using the intermediate vectors employed above can proceed as follows,

$$\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\mathbf{X}}} = \left(\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\mathbf{F}}_{\text{node}}} \right) \left(\frac{\partial \bar{\mathbf{F}}_{\text{node}}}{\partial \bar{\mathbf{F}}_{\text{node, map}}} \right) \left(\frac{\partial \bar{\mathbf{F}}_{\text{node, map}}}{\partial \bar{\mathbf{X}}_{\text{node, map}}} \right) \left(\frac{\partial \bar{\mathbf{X}}_{\text{node, map}}}{\partial \bar{\mathbf{X}}_{\text{node}}} \right) \left(\frac{\partial \bar{\mathbf{X}}_{\text{node}}}{\partial \bar{\mathbf{X}}} \right) \quad (3.83)$$

Using (3.77), (3.78), (3.81) and (3.82), we can simplify (3.83) as follows

$$\frac{\partial \bar{\mathbf{F}}}{\partial \bar{\mathbf{X}}} = (\mathcal{P})^T \ (\bar{\mathcal{P}}_F)^T \left(\frac{\partial \bar{\mathbf{F}}_{\text{node, map}}}{\partial \bar{\mathbf{X}}_{\text{node, map}}} \right) (\bar{\mathcal{P}}_F) \ (\mathcal{P}) \quad (3.84)$$

Noting that $\bar{\mathbf{F}}_{\text{node, map}}$ and $\bar{\mathbf{X}}_{\text{node, map}}$ are the Fourier coefficients of a harmonically-related artificial frequency spectrum in Ω_M , it can be seen that computing $\partial \bar{\mathbf{F}}_{\text{node, map}} / \partial \bar{\mathbf{X}}_{\text{node, map}}$ can be implemented in the same manner described in Chapter 2 for circuits with single-tone excitation.

3.5. GENERALIZED FOURIER TRANSFORM

Chapter **4**

Time Domain Methods

The previous chapters presented steady-state analysis techniques based on pure frequency-domain approaches. In those approaches, the steady-state periodic waveforms in the circuit are represented by their Fourier series expansion and the problem is subsequently formulated as a nonlinear algebraic problem in the Fourier series coefficients. This type of approach becomes computationally demanding if the periodic waveforms exhibit sharp transitions and abrupt variations, for in this case capturing the waveforms accurately requires a huge number of Fourier coefficients. This fact may be viewed as a consequence of using uniformly distributed time points as the sampling scheme for the waveforms. If certain portions of the waveforms exhibit very sharp transitions while other portions show a relatively slower behavior, then both portions will have eventually to be sampled with the same density under the uniform sampling scheme. Naturally, a large density of sampling points is needed in the faster portions. However, the uniform sampling scheme will also dictate sampling the slower portions with the same density. This might lead to unnecessarily large number of sampling points with an equal number of Fourier coefficients resulting in a large nonlinear problem.

One might observe that using a uniform equidistant sampling scheme to sample a waveform with few sharp turning points is an over-exhaustive sampling paradigm. A more efficient way is to try to follow the waveform as they evolve in the time-domain, through marching in time, starting from $t = 0$, with time steps whose sizes are adapted according to the local conditions of the fastest waveform in the circuit. Ideally such a time-stepping technique should offer the following features,

- If one of the circuit waveforms is going through a very fast transition, the step size will have to be made smaller;
- If all the circuit waveforms show periods of slower activity, the time-stepping technique should try to make wider strides;

4.1. STEADY-STATE ANALYSIS USING IVP APPROACHES

- If the waveforms exhibit a significant period of transient response due to having some high-Q circuit, then they must be allowed to proceed in time until all their transients have died out and a steady-state periodic forms has been reached for all circuit waveforms.

A time-stepping algorithm with the above features is usually referred to as an **Initial-Value-Problem (IVP)** approach to steady-state analysis. This terminology is an obvious consequence of starting the time marching from $t = 0$. IVP approaches represent the simplest and most direct steady-state analysis methods based on pure time-domain techniques. Yet, as simple as they may seem at a first glance, they suffer certain difficulties that render them very computationally cumbersome for many of RF circuits. The foremost difficulty in IVP-based approaches arises from having to follow the waveforms in their transient phase. This can consume days, if not weeks, of computation time especially for large circuits having a high-Q filter tanks and excited by a periodical fast input source. Another problem that arises inherently from any IVP-based approach is the error introduced in the times-stepping process. This error comes as a direct result of the fact that available IVP-techniques are forced to use a small order polynomial to approximate the circuit waveforms locally in order to preserve the stability of time-stepping. Although IVP techniques are rarely used in steady-state analysis, they represent an essential component in understanding more advanced time-domain approaches, and therefore will be taken as our starting point for the exposition in this chapter.

4.1 Steady-State Analysis using IVP approaches

To simplify illustrating IVP approaches to steady-state analysis, we will consider first their application for solving scalar differential equations, and then demonstrate how they can be adapted to handle the differential-algebraic equations arising from the circuit MNA formulation. Consider a simple differential equations described by

$$\frac{dx}{dt} = f(x(t), t) \quad (4.1)$$

where $x(t)$ is a scalar unknown function of time and $f(x(t), t)$ is a an analytic function that depends on t . The goal of the IVP approaches is to compute $x(t)$ at discrete time points. To achieve that, the initial condition of $x(t)$ at $t = t_0 = 0$, (i.e., $x(0)$) is assumed to be known, and denoted by x_0 ¹, and the values of $x(t)$ at future time points t_1, t_2, \dots, t_n are sought. Since $x(t_0)$ is known, the value of the derivative, $x' \equiv dx/dt$, can also be computed at $t = 0$ through direct substitution in (4.1). Figure 4.1 shows that x' at $t = 0$ is equivalent to the slope of a line drawn tangent to $x(t)$ at at $t = 0$. If h is the time step from t_0 to t_1

$$h = t_1 - t_0 \quad (4.2)$$

¹Initial conditions for circuit variables can be found out through DC analysis.

and x_1 is the value of the tangent line at $t = t_1$, the derivative x' at $t = t_0$ would then be given by

$$x'|_{t=t_0} \equiv x'_0 = \frac{x_1 - x_0}{h} \quad (4.3)$$

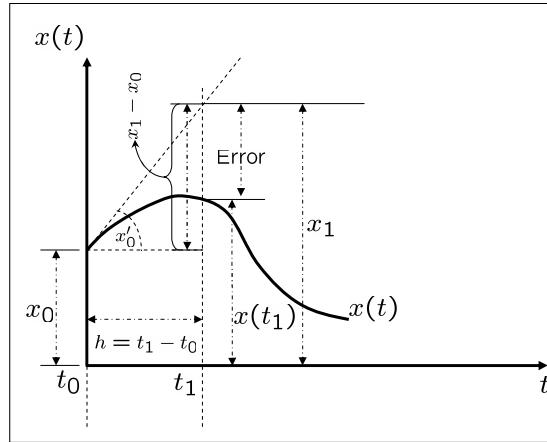


Figure 4.1: Approximation of x' by the slope of the tangent.

If h is sufficiently small, the the value of $x(t)$ at $t = t_1$ can be approximated by x_1 , and using (4.3) we obtain,

$$x(t_1) \approx x_1 = x_0 + h x'_0 \quad (4.4)$$

The above relation is known as the **forward Euler (FE) method** and can be used to compute $x(t)$ progressively in time. As seen from Figure 4.1, the approximation error will be larger if h is larger.

Another approach to approximate $x(t)$ at future time points is to assume that an initial *estimate* value is available at the next time instant, e.g. $t = t_1$, and then represent the derivative at a future time instant ($t = t_1$) by the slope of tangent line,

$$\left. \frac{dx}{dt} \right|_{t=t_1} = \frac{x_1 - x_0}{h} \quad (4.5)$$

and thus we have,

$$x_1 = x_0 + h \left. \frac{dx}{dt} \right|_{t=t_1} \quad (4.6)$$

If x_1 is taken as the approximate value for $x(t)$ at $t = t_1$, then after substituting $t = t_1$ in

4.1. STEADY-STATE ANALYSIS USING IVP APPROACHES

(4.1) and using its right-hand side in (4.6), we obtain

$$\begin{aligned} x(t_1) \approx x_1 &= x_0 + h f(x(t_1), t_1) \\ &\approx x_0 + h f(x_1, t_1) \end{aligned} \quad (4.7)$$

Thus x_1 is obtained as the solution of the following nonlinear equation

$$x_1 = x_0 + h f(x_1, t_1) \quad (4.8)$$

As indicated earlier, solving nonlinear equations is typically done using an iterative technique such as the NR method for which some starting initial guess is required. For that purpose, the initial estimate may be used as the starting guess solution. For example, one possible choice for the initial guess is to use the value of $x(t)$ at the previous time point $t = t_0$, or x_0 . Formula (4.8) is called **backward Euler (BE) method**, and may be used to compute $x(t)$ at future time points.

Another possibility to generate another formula to approximate $x(t_1)$, is to take the *average* of the derivatives at the two time points t_0 and t_1 as the slope of line tangent at $t = t_0$,

$$\frac{1}{2} \left(\frac{dx}{dt} \Big|_{t=t_1} + \frac{dx}{dt} \Big|_{t=t_0} \right) = \frac{x_1 - x_0}{h} \quad (4.9)$$

Putting $t = t_1$ and $t = t_2$ in (4.1) and substituting the right sides in (4.9) we get,

$$\frac{1}{2} (f(x(t_1), t_1) + f(x(t_0), t_0)) = \frac{x_1 - x_0}{h} \quad (4.10)$$

Taking x_1 as the approximate value for $x(t_1)$ and rearranging terms in (4.10) we get,

$$x_1 = x_0 + \frac{h}{2} (f(x_1, t_1) + f(x_0, t_0)) \quad (4.11)$$

Again the above formula is a nonlinear equation that needs to solved for x_1 . Formula (4.11) is known as the **Trapezoidal rule (TR) method**.

FE, BE and TR can all be cast in a general formulation for deducing the value of $x(t)$ at the future time point, t_{n+1} , given that its value at present time, t_n , is available. Those formulations are given by

$$x_{n+1} = x_n + h_n f(x_n, t_n) \quad \text{FE method} \quad (4.12)$$

$$x_{n+1} = x_n + h_n f(x_{n+1}, t_{n+1}) \quad \text{BE method} \quad (4.13)$$

$$x_{n+1} = x_n + \frac{h_n}{2} (f(x_{n+1}, t_{n+1}) + f(x_n, t_n)) \quad \text{TR method} \quad (4.14)$$

where

$$h_n = t_{n+1} - t_n \quad (4.15)$$

4.1.1 Illustrative Examples

We use simple examples in this section to illustrate the basic computational steps involved in using FE, BE and TR methods in the numerical solution of differential equations.

Example 1

Consider the differential equations defined by,

$$\frac{dx}{dt} = -e^{-x(t)} + 3 \quad (4.16)$$

where $x(t = 0) = 0$, and our goal is to estimate $x(t)$ at $t = 0.1$ Sec.. It is obvious here that $f(x(t), t) \equiv -e^{-x(t)} + 3$.

Using FE to compute $x(t = 0.1)$ is a straightforward application of the corresponding formula in (4.12), where $h = 0.1$. Hence, we have,

$$\begin{aligned} x(0.1) \approx x_{1,\text{FE}} &= x(0) + h \times (-e^{-x(0)} + 3) \\ &= 0 + 0.1 \times 2 \\ &= 0.2 \end{aligned} \quad (4.17)$$

Applying BE to predict $x(0.1)$ requires solving the following nonlinear equation,

$$x_{1,\text{BE}} = x_0 + 0.1 \times (-e^{-x_1} + 3) \quad (4.18)$$

for x_1 , where $x_1 = x(0.1)$ and $x_0 = x(0) = 0$. Using NR method to solve for x_1 in (4.18) requires first an initial guess point, which can be taken as the value of $x(t)$ at the previous time point $t = 0$, 0. Letting $\phi(x_1) := x_1 - 0.1 \times (-e^{-x_1} + 3)$, Table 4.1 displays the iterative progress of the NR iteration. After 10 iterations, the solution was approximated to $x_{1,\text{BE}} = 0.21973$.

Similar steps can be used to implement the TR, where in this case $x(0.1)$ follows from substituting in (4.14) and solving the resulting nonlinear equation,

$$x_1 = x_0 + \frac{0.1}{2} (-e^{-x_1} + 3 - e^{-x_0} + 3) \quad (4.19)$$

for x_1 . Table 4.2 summarizes the results of applying the NR to solve (4.19) for x_1 . After 7 iterations, the solution reached is $x_{1,\text{TR}} = 0.20945$.

Table 4.1: Results of applying NR iterations for solving (4.18) starting from the initial guess of $x_1 = 0$.

iteration	ϕ	x_1
0	0	0.2
1	0.002296	0.22222
2	-0.00011729	0.2196
3	5.932510^{-06}	0.21973
4	-3.002110^{-07}	0.21973
5	1.519210^{-08}	0.21973
6	-7.687610^{-10}	0.21973
7	3.890210^{-11}	0.21973
8	-1.968610^{-12}	0.21973
9	9.96710^{-14}	0.21973

Table 4.2: Results of applying NR iterations for solving (4.19) starting from the initial guess of $x_1 = 0$.

iteration	ϕ	x_1
0	0	0.2
1	0.0010342	0.21053
2	-2.335210 ⁻⁰⁵	0.20942
3	5.250110 ⁻⁰⁷	0.20945
4	-1.180510 ⁻⁰⁸	0.20945
5	2.654210 ⁻¹⁰	0.20945
6	-5.967810 ⁻¹²	0.20945
7	1.341710 ⁻¹³	0.20945

4.1. STEADY-STATE ANALYSIS USING IVP APPROACHES

The previous example may have created the impression that implementing the FE is much easier than the BE and TR since it does not require solving nonlinear equations. But this would be a hasty conclusion if the approximation error is not thoroughly investigated.

Example 2

The goal of this example is to shed the light on the approximation error produced from using FE, BE and TR. To this end, we consider the following differential equation

$$\frac{dx}{dt} = -Gx(t) + 5 + \cos(t) \quad (4.20)$$

where $G = 0.0001$. The above differential equation is linear in $x(t)$ and therefore has the closed-form solution,

$$x(t) = e^{-Gt}C + \frac{5G^2 + 5 + G^2 \cos(t) + \sin(t)G}{G(G^2 + 1)} \quad (4.21)$$

where $C = -50000$. The availability of the exact analytic solution for (4.20) in (4.21) will allow us to gain some insight and draw basic conclusion with regard to the error performance of the three methods. Table 4.1.1 summarizes the results obtained after running the three methods using a constant step size $h = 0.1$ and up to time $t = 2\text{Sec}$. To further illustrate the error patterns, the three methods were run up to a longer time, $t = 125\text{ Sec}$. and the corresponding percentage error results were plotted in Figure 4.2. Figures 4.3 and 4.4 depict the results obtained by running the three methods using two different step sizes of $h = 0.01$ and $h = 0.5\text{ Sec.}$, respectively.

It is obviously clear from Figures 4.2- 4.4 that regardless of the step size used, the error obtained from FE just keeps on growing unboundedly with time. In addition, the following observation can be made easily from the obtained figures,

- The TR has *consistently* performed better than the BE for all step sizes with regard to the approximation error,
- Both of the BE and TR tend to produce less error if smaller step sizes are used.

It is quite natural at this point to see that there are several issues related to any given integration method. The first issue is related to the potential unbounded error growth such as in the case of FE method. An integration technique that results in an error that keeps on growing with each step is called unstable and has a limited practical value. The second important issue pertinent to the integration method is related to the magnitude of the approximation error and its relation to the size of the time step. This issue is commonly known as the integration order of the method, where methods with higher integration order tend to have less approximation error. Thus, as observed form the results of the previous example, it is possible to state that TR is of higher order than BE. Another central issue in

Table 4.3: A numerical comparison between FE, BE and TR

Time	Exact	FE	BE	TR	% Error in BE	% Error in BE	% Error in TR
0.1	0.59983	0.6	0.59949	0.59975	0.028272	0.056015	0.013872
0.2	1.1987	1.1994	1.1975	1.1985	0.06533	0.097462	0.013814
0.3	1.7955	1.7973	1.793	1.7952	0.10214	0.13859	0.013718
0.4	2.3894	2.3927	2.3851	2.389	0.13856	0.17927	0.013583
0.5	2.9794	2.9845	2.9728	2.979	0.17444	0.21935	0.013411
0.6	3.5645	3.572	3.5553	3.5641	0.20964	0.2587	0.013202
0.7	4.1441	4.1542	4.1318	4.1435	0.24402	0.29718	0.012956
0.8	4.7172	4.7303	4.7014	4.7166	0.27744	0.33464	0.012674
0.9	5.2831	5.2995	5.2635	5.2824	0.30974	0.37096	0.012357
1	5.8412	5.8611	5.8175	5.8405	0.3408	0.40599	0.012006
1.1	6.3909	6.4145	6.3628	6.3901	0.37046	0.43959	0.011621
1.2	6.9316	6.9592	6.8989	6.9308	0.39859	0.47164	0.011206
1.3	7.4631	7.4948	7.4256	7.4623	0.42504	0.50199	0.010759
1.4	7.9849	8.0208	7.9425	7.9841	0.44969	0.53051	0.010285
1.5	8.4968	8.537	8.4495	8.496	0.4724	0.55708	0.0097828
1.6	8.9988	9.0432	8.9465	8.998	0.49304	0.58157	0.0092562
1.7	9.4908	9.5394	9.4335	9.49	0.51149	0.60387	0.0087067
1.8	9.9729	10.026	9.9107	9.9721	0.52764	0.62386	0.0081367
1.9	10.445	10.502	10.378	10.444	0.54137	0.64144	0.0075488
2	10.908	10.968	10.837	10.907	0.5526	0.65652	0.0069456

4.1. STEADY-STATE ANALYSIS USING IVP APPROACHES

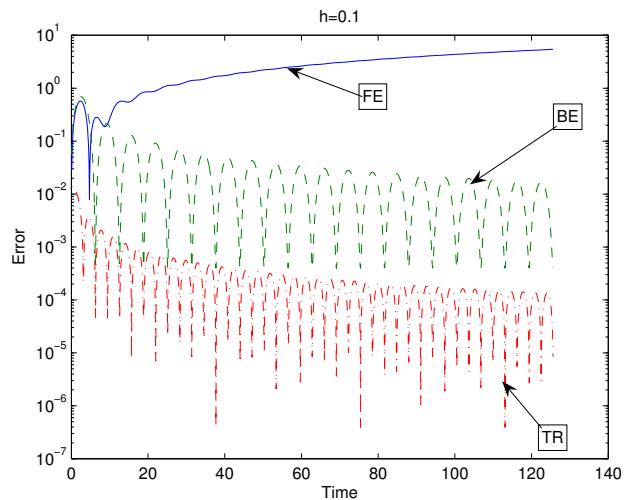


Figure 4.2: A comparison between the % Error produced from FE, BE and TR for $h = 0.1$ Sec.

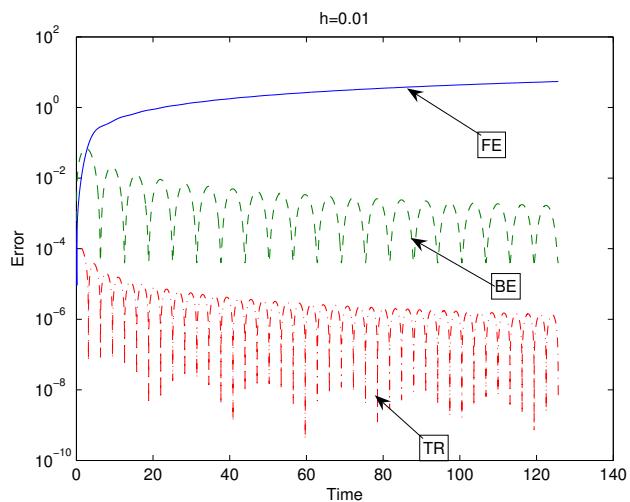


Figure 4.3: A comparison between the % Error produced from FE, BE and TR for $h = 0.01$ Sec.

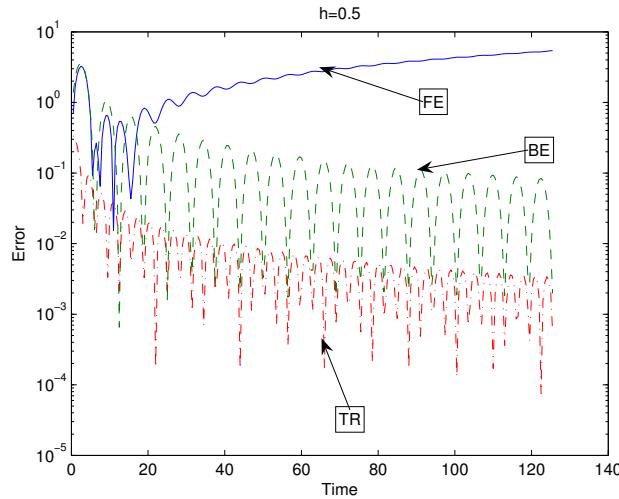


Figure 4.4: A comparison between the % Error produced from FE, BE and TR for $h = 0.5$ Sec.

performing time-domain integration is the problem of *estimating* the approximation error, since in general nonlinear problems one usually does not have an access to the exact solution to compare it against the approximate one. Furthermore the role of the step size in the approximation error is another pivotal issue in implementing various integration methods. Normally, dynamic control and adjustment of the step size is crucial to maintain the error estimate within a prescribed bound. Unfortunately, an in-depth theoretical exposition for the above issues and their characterization in different integration techniques is beyond the limited scope of this chapter. The interested reader may refer to [10] for a more thorough analytical discussion of those topics. For the purpose of this course, however, it will be sufficient to know that both of BE and TR, thanks to their absolute stability properties, represent the simplest and best candidates to tackle the issue of time-domain simulation of general nonlinear circuits. In addition, the problems of adjusting the step size to control the error and estimating that error will not be addressed further, and when desired, those tasks can be performed using commercial special software packages such as the Matlab ODE toolbox. Thus a plain implementation of BE and TR with a constant step size will represent our main tool for handling a nonlinear circuit.

4.1.2 Transient Analysis of Circuits using BE and TR

A general nonlinear circuit is described in the time-domain using the MNA formulation as a set of algebraic-differential equations in the form

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C} \frac{d\mathbf{x}(t)}{dt} + \mathbf{f}(\mathbf{x}(t)) = \mathbf{b}(t) \quad (4.22)$$

or equivalently in a more convenient form

$$\mathbf{C} \frac{d\mathbf{x}(t)}{dt} = -\mathbf{G}\mathbf{x}(t) - \mathbf{f}(\mathbf{x}(t)) + \mathbf{b}(t) \quad (4.23)$$

The main objective here is to show how simple integration methods such as BE and TR can be employed to compute the vector of unknown waveforms $\mathbf{x}(t)$ at various time points. We start with BE, where we consider Eq. (4.6) as the basic update rule. According to Eq. (4.6), the vector of the values of the MNA waveforms at time t_1 , $\mathbf{x}(t_1)$, can be approximated by \mathbf{x}_1 where,

$$\mathbf{x}_1 = \mathbf{x}_0 + h \left. \frac{d\mathbf{x}}{dt} \right|_{t=t_1} \quad (4.24)$$

where $\mathbf{x}_0 = \mathbf{x}(t_0)$ and $h = t_1 - t_0$. To obtain \mathbf{x}_1 , we premultiply both sides of (4.24) by the matrix \mathbf{C} to get,

$$\mathbf{C}\mathbf{x}_1 = \mathbf{C}\mathbf{x}_0 + h\mathbf{C} \left. \frac{d\mathbf{x}(t)}{dt} \right|_{t=t_1} \quad (4.25)$$

Writing the MNA equations at $t = t_1$ yields,

$$\mathbf{C} \left. \frac{d\mathbf{x}(t)}{dt} \right|_{t=t_1} = -\mathbf{G}\mathbf{x}(t_1) - \mathbf{f}(\mathbf{x}(t_1)) + \mathbf{b}(t_1) \quad (4.26)$$

Substituting from (4.25) into (4.26) and rearranging we get,

$$(\mathbf{C} + h\mathbf{G})\mathbf{x}_1 + h\mathbf{f}(\mathbf{x}_1) = \mathbf{C}\mathbf{x}_0 + h\mathbf{b}(t_1) \quad (4.27)$$

The above system of equations represents a system of nonlinear equations in \mathbf{x}_1 . Thus, computing \mathbf{x}_1 can be achieved through using some iterative technique such as the N-R method. A good initial guess for \mathbf{x}_1 , $\mathbf{x}_1^{(0)}$, is usually taken as \mathbf{x}_0 .

Similar steps can also be used to adapt the TR to compute \mathbf{x}_1 . In this case Eq. (4.9) is taken as the basic update rule, where substitution from the MNA equations (4.25) at $t = t_1$

results in the following system of nonlinear equations

$$\left(\mathbf{C} + \frac{h}{2}\mathbf{G}\right)\mathbf{x}_1 + \frac{h}{2}\mathbf{f}(\mathbf{x}_1) = \left(\mathbf{C} - \frac{h}{2}\mathbf{G}\right)\mathbf{x}_0 - \frac{h}{2}\mathbf{f}(\mathbf{x}(t_0)) + \frac{h}{2}(\mathbf{b}(t_0) + \mathbf{b}(t_1)) \quad (4.28)$$

Notice that the right side in both of (4.27) and (4.28) is already known.

Generalization of the BE and TR to evaluate \mathbf{x}_n , the values of $\mathbf{x}(t)$ at time $t = t_1$, can be established in a straightforward manner method given the values at the previous time point t_{n-1} . The corresponding systems of equations in the general case thus becomes,

$$(\mathbf{C} + h_n\mathbf{G})\mathbf{x}_n + h_n\mathbf{f}(\mathbf{x}_n) = \mathbf{C}\mathbf{x}_{n-1} + h_n\mathbf{b}(t_n) \quad (4.29)$$

for the BE method and,

$$\left(\mathbf{C} + \frac{h_n}{2}\mathbf{G}\right)\mathbf{x}_n + \frac{h_n}{2}\mathbf{f}(\mathbf{x}_n) = \left(\mathbf{C} - \frac{h_n}{2}\mathbf{G}\right)\mathbf{x}_{n-1} - \frac{h_n}{2}\mathbf{f}(\mathbf{x}(t_{n-1})) + \frac{h_n}{2}(\mathbf{b}(t_{n-1}) + \mathbf{b}(t_n)) \quad (4.30)$$

for the TR method, where $h_n = t_n - t_{n-1}$.

4.1.3 Steady-State Analysis via IVP

It was demonstrated above that integration methods such as BE and TR can be used to follow the circuit waveforms as they evolve with time starting from an initial time point. For circuits excited by periodic sources, the circuit waveforms will eventually settle into steady-state periodic waveforms. Finding the steady-state operating point for these circuits can be carried out using integration methods tailored for IVP such as BE and TR, through following the waveforms until all transients disappear and a periodic pattern emerges for all waveforms. Once a periodic pattern has been reached, a Fourier transformation technique such as FFT can be used to compute the spectrum of the desired signal. Such an approach to steady-state analysis is known as IVP-based steady-state analysis. Given that an IVP-based steady-state analysis techniques are purely grounded in time-domain, they can offer a promising remedy for the difficulties encountered in methods based on pure frequency-domain such as the Harmonic Balance, especially in circuits with strongly nonlinear behavior where the waveforms can not be captured by a limited number of harmonical components (or Fourier terms). Nonetheless, IVP-based techniques can suffer a major disadvantage for circuits with high-Q. This is because in these circuits, the transient phase can last for many cycles of the excitation sources, and thus end up forcing the numerical integrator to take too many time steps and solve a nonlinear system of equations at each step. The huge CPU cost is not the only problem encountered in using IVP approaches to find the steady-state. In fact, the numerical inaccuracy introduced in each step due to the truncation error presents another source of difficulty that is likely to corrupt the computed spectrum.

4.1. STEADY-STATE ANALYSIS USING IVP APPROACHES

The example circuit shown in Figure 4.5 illustrates using the IVP-based techniques to compute the steady-state point. This circuit is excited by a sinusoidal source having a frequency of 10kHz, and therefore has a periodic steady-state response as $t \rightarrow \infty$. The circuit has been deliberately designed to have a high-Q tank circuit to demonstrate the practical difficulties in using IVP-based technique in computing the steady-state response. The transient phase of the circuit can be estimated based on the time constant, which is given by

$$\tau = 0.5R = 0.1\text{Sec}$$

Thus the transient phase would last around 1000 cycles of the periodic source. The numerical integrator used to reach the steady-state is based on the Matlab ODE toolbox, which can take a variable time step size to maintain the estimated error within some prescribed acceptable bounds. Table 4.4 summarizes the number of time steps that the integrator needed before reaching the steady-state for three sets of values of initial conditions x_0 .

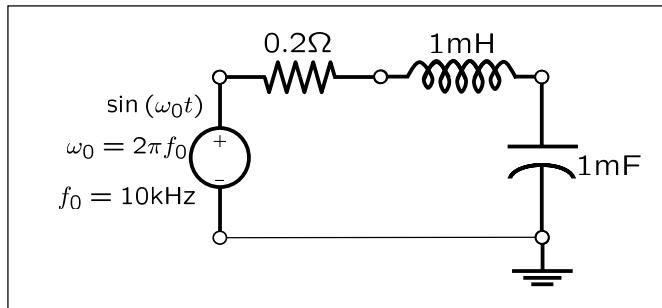


Figure 4.5: A circuit with high-Q that produces a long transient phase in the time-domain.

Table 4.4: Results for steady-state analysis using IVP approach for the circuit of Figure 4.5.

Initial value x_0	Number of time steps.
{ 0.9501 0.2311 0.6068 0.4860 0.8913 }	274981
{ 0 0 0 0 0 }	264981
{ 0 0.0032 0 -0.0159 0.0159 }	3

The results shown in Table 4.4 demonstrate that the choice of the initial conditions can lead to shorter transient phase. That observation suggests that with a suitably chosen initial values for the MNA waveforms can bypass long transients.

4.2 Steady-State Analysis using Boundary-Value Problem (BVP)

The issue of *searching* for a proper set of initial conditions that can bypass long transients in the circuit response is the basic objective underlying the BVP approaches to steady-state analysis. The optimum initial conditions sought will be a vector \mathbf{x}_0^* which when assigned as initial values for $\mathbf{x}(t)$ at $t = t_0$, no transient phase will be encountered and the circuit will directly enter into the periodic steady-state operation. In other words, if transient analysis is performed starting from initial conditions of $\mathbf{x}(0) = \mathbf{x}_0^*$ and continued for a period of time up to $t = T$, where T is the common period of the circuit stimulus (input sources), then, ideally, we would have

$$\mathbf{x}(T) = \mathbf{x}(0) = \mathbf{x}_0^*.$$

The search for \mathbf{x}_0^* that satisfies the above equation is the main theory behind the so-called shooting method (SM) which is one of the main BVP techniques used to find the steady-state response.

4.2.1 Shooting Method

The central and enabling idea behind the SM is built around the concept of State-Transition Function (STF), which is defined for any system described by differential equations, as a function that takes as input the values of the system state at certain time, say $t = t_0$, i.e., $\mathbf{x}(t_0)$, and a period of time ΔT and produces at its output the system state at time $t_0 + \Delta T$, where the system state in the present context refers to the MNA variables. Figure 4.6 depicts the notion of STF graphically.

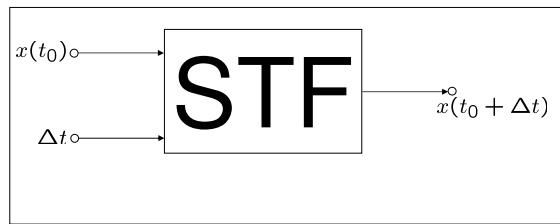


Figure 4.6: Schematic description for the concept of State-Transition Function (STF).

We use the notation $\phi(\mathbf{x}_0, t_0, \Delta T)$ to denote the STF, where $\mathbf{x}_0 = \mathbf{x}(t_0)$. Thus by the very definition of STF, we may write

$$\mathbf{x}(t_0 + \Delta T) = \phi(\mathbf{x}_0, t_0, \Delta T). \quad (4.31)$$

To have a better grasp of STF, consider a simple system described by the following scalar differential equation

$$\dot{x} = -ax(t) + b \cos(\omega_0 t) \quad (4.32)$$

4.2. STEADY-STATE ANALYSIS USING BOUNDARY-VALUE PROBLEM (BVP)

where a is a positive constant. The above differential equation is of the linear type and therefore its solution function can be written analytically as,

$$x(t) = e^{-a(t-t_0)}x_0 + \int_{t_0}^t be^{-a(\tau-t_0)} \cos(\omega_0\tau) d\tau \quad (4.33)$$

The STF for the system in (4.32) can be deduced by inspection,

$$\phi(x_0, t_0, \Delta T) = e^{-a(\Delta T)}x_0 + \int_{t_0}^{t_0+\Delta T} be^{-a(\tau-t_0)} \cos(\omega_0\tau) d\tau \quad (4.34)$$

Unfortunately, obtaining an analytical expression for the STF of a general systems described by a nonlinear algebraic-differential equations is not possible. Instead, one will have to use numerical techniques to represent STF. We next show how the concept of STF can be employed to search for the optimum initial conditions.

As mentioned earlier, the basic objective in SM is to search for a vector \mathbf{x}_0^* that when used as an initial condition for $\mathbf{x}(0)$, we would get

$$\mathbf{x}(0) = \mathbf{x}(T) = \mathbf{x}_0^* \quad (4.35)$$

after transient analysis has been performed for a period of time T . Using the notion of the STF, (4.35) be formulated as

$$\phi(\mathbf{x}_0^*, 0, T) = \mathbf{x}_0^* \quad (4.36)$$

Thus, \mathbf{x}_0^* , the optimum initial condition, may be viewed as the solution vector for \mathbf{x}_0 satisfying the *algebraic* nonlinear system of equations,

$$\psi(\mathbf{x}_0) \equiv \phi(\mathbf{x}_0, 0, T) - \mathbf{x}_0 = \mathbf{0} \quad (4.37)$$

A variety of techniques can be deployed to solve the system of equations in (4.37). Newton-Raphson (NR) method is the most commonly used in solving nonlinear system of equations, its application in solving (4.37) represents one of the popular techniques for implementing the SM. Solving (4.37) using NR method is known as the Newton shooting and will be detailed in the next subsection.

4.2.2 Newton Shooting Method (NSM)

Using the NR method to solve (4.37) requires the iterative computation of the residual error $\psi(\mathbf{x}_0^{(j)})$ and its Jacobian matrix at successive trial vectors $\mathbf{x}_0^{(j)}$, starting from a suitably chosen initial trial vector $\mathbf{x}_0^{(0)}$. The i^{th} trial vector is obtained from the $(i-1)^{\text{th}}$ vector via

the NR update rule,

$$\mathbf{x}_0^{(i)} = \mathbf{x}_0^{(i-1)} - \Delta \mathbf{x}_0^{(i-1)} \quad (4.38)$$

where

$$\Delta \mathbf{x}_0^{(i-1)} = (\mathcal{J}_{i-1})^{-1} \psi(\mathbf{x}_0^{(i-1)}) \quad (4.39)$$

and \mathcal{J}_{i-1} is the Jacobian matrix of ψ w.r.t. \mathbf{x}_0 computed at $\mathbf{x}_0 = \mathbf{x}_0^{(i-1)}$, i.e.,

$$\begin{aligned} \mathcal{J}_{i-1} &= \left. \frac{\partial \psi}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0=\mathbf{x}_0^{(i-1)}} \\ &= \left. \frac{\partial \phi}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0=\mathbf{x}_0^{(i-1)}} - \mathbf{I} \end{aligned} \quad (4.40)$$

As noted earlier, the STF $\phi(x_0, t_0, T)$ can not be described in an analytic form for general nonlinear systems. This fact entails that computing the STF and its Jacobian matrix can be achieved only through numerical means. Evaluating $\phi(\cdot, 0, T)$ and its Jacobian matrix $\partial \phi / \partial \mathbf{x}_0$ at any given trial vector $\mathbf{x}_0^{(i)}$ represents the central issues at the core of implementing the Newton shooting method. These issues will be discussed next.

Computing the STF $\phi(\mathbf{x}_0^{(i)}, 0, T)$.

From the very basic definition of the STF, $\phi(\mathbf{x}_0^{(i)}, 0, T)$ is the numerical value of the system state (or the MNA variables $\mathbf{x}(t)$) after a period of time T assuming that the system initial state at $t = 0$ was given by $x(0) = \mathbf{x}_0^{(i)}$. In other words,

$$\phi(\mathbf{x}_0^{(i)}, 0, T) = \mathbf{x}(T)|_{x(0)=\mathbf{x}_0^{(i)}} \quad (4.41)$$

Thus the task of computing $\phi(\mathbf{x}_0^{(i)}, 0, T)$ is equivalent to computing $x(T)$ using the set of prescribed initial conditions $\mathbf{x}(0) = \mathbf{x}_0^{(i)}$. This task is an example of a classical IVP, and therefore any of the approaches discussed in Section 4.1 can be employed for this purpose. Hence, computing the STF $\phi(\mathbf{x}_0^{(i)}, 0, T)$ can be computed by launching an IVP method such as BE or TR using the initial condition $\mathbf{x}(0) = \mathbf{x}_0^{(i)}$ and proceeding forward in time for a period of time T . The value of $\mathbf{x}(t)$ reached at $t = T$ is $\phi(\mathbf{x}_0^{(i)}, 0, T)$. It is worth noting that the ability of IVP approaches to adapt the time step size according to the local condition of the waveforms represents a very desirable feature required in handling

arbitrary waveforms, especially those waveforms with sharp spikes whose representation using Fourier series would require a large number of terms.

Computing the Jacobian matrix of the STF, $\partial\phi/\partial\mathbf{x}_0$

Note that by the very definition of the STF, we can write

$$\frac{\partial\phi(\partial\mathbf{x}_0, 0, T)}{\partial\mathbf{x}_0} \Big|_{\mathbf{x}_0=\mathbf{x}_0^{(i)}} = \left(\frac{\partial\mathbf{x}(T)}{\partial\mathbf{x}_0} \right) \Big|_{\mathbf{x}_0=\mathbf{x}_0^{(i)}} \quad (4.42)$$

The matrix $\partial\mathbf{x}(T)/\partial\mathbf{x}_0$ is typically known as the sensitivity of the final state with respect to the initial state. Recall that the absence of an analytic form for $\phi(\cdot, 0, T)$ will force us to use only numerical approximation techniques to compute its Jacobian matrix. It is usually difficult to find a relation relating the final state ($t = T$) to the initial state ($t = 0$). However, it is always possible to find a relation between the final state and the system state at an earlier state, say at $t = T - h_m$, for a sufficiently small step h_m . For example, if BE method has been used to reach $\mathbf{x}(T)$, then we must have

$$(\mathbf{C} + h_m \mathbf{G}) \mathbf{x}(T) + h_m \mathbf{f}(\mathbf{x}(T)) = \mathbf{C}\mathbf{x}(T - h_m) + h_m \mathbf{b}(T) \quad (4.43)$$

and therefore, the matrix of partial derivatives of $\mathbf{x}(T)$ with respect to $\mathbf{x}(T - h_m)$ will be given as the solution to the following system,

$$\left(\mathbf{C} + h_m \mathbf{G} + h_m \frac{\partial\mathbf{f}}{\partial\mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}(T)} \right) \frac{\partial\mathbf{x}(T)}{\partial\mathbf{x}(T - h_m)} = \mathbf{C} \quad (4.44)$$

It is important to note that term between brackets in (4.44) represents the Jacobian matrix of the MNA equations in the time-domain. It is worth noting that the LU factors of this matrix have been generated during the transient simulation to obtain $\mathbf{x}(T)$. Those factors can be stored and reused to solve the linear system (4.44).

To use the matrix $\partial\mathbf{x}(T)/\partial\mathbf{x}(T - h_m)$ towards computing the Jacobian matrix of the STF, the chain rule of differentiation is applied to write the right side of (4.42) as follows

$$\frac{\partial\mathbf{x}(T)}{\partial\mathbf{x}_0} = \left(\frac{\partial\mathbf{x}(T)}{\partial\mathbf{x}(T - h_m)} \right) \left(\frac{\partial\mathbf{x}(T - h_m)}{\partial\mathbf{x}_0} \right) \quad (4.45)$$

In fact, the chain rule of differentiation can be applied repeatedly to expand $\partial\mathbf{x}(T)/\partial\mathbf{x}_0$ as follows,

$$\frac{\partial\mathbf{x}(T)}{\partial\mathbf{x}_0} = \left(\frac{\partial\mathbf{x}(T)}{\partial\mathbf{x}(t_m)} \right) \left(\frac{\partial\mathbf{x}(t_m)}{\partial\mathbf{x}(t_{m-1})} \right) \dots \left(\frac{\partial\mathbf{x}(t_1)}{\partial\mathbf{x}_0} \right) \left(\frac{\partial\mathbf{x}_0}{\partial\mathbf{x}_0} \right) \quad (4.46)$$

assuming that t_1, t_2, \dots, t_m is the sequence of time steps used to reach $\mathbf{x}(T)$ starting from $\mathbf{x}(0)$, and $h_k = t_{k+1} - t_k$. Noting that the last matrix on the right side of (4.47) is equivalent to the identity matrix, the Jacobian of the STF reduces to

$$\frac{\partial \mathbf{x}(T)}{\partial \mathbf{x}_0} = \left(\frac{\partial \mathbf{x}(T)}{\partial \mathbf{x}(t_m)} \right) \left(\frac{\partial \mathbf{x}(t_m)}{\partial \mathbf{x}(t_{m-1})} \right) \cdots \left(\frac{\partial \mathbf{x}(t_1)}{\partial \mathbf{x}_0} \right) \quad (4.47)$$

which is a sequence of m matrix multiplications where the k^{th} matrix, $\partial \mathbf{x}(t_{k+1})/\partial \mathbf{x}(t_k)$, is given by the matrix-valued solution to the following linear system of equations

$$\left(\mathbf{C} + h_k \mathbf{G} + h_k \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}(t_{k+1})} \right) \frac{\partial \mathbf{x}(t_{k+1})}{\partial \mathbf{x}(t_k)} = \mathbf{C} \quad (4.48)$$

if the BE formulation is used in the IVP phase. On the other hand, if the TR is used to compute $\mathbf{x}(T)$, the k^{th} matrix would be given by the matrix-valued solution to the following linear system,

$$\left(\mathbf{C} + \frac{h_k}{2} \left(\mathbf{G} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}(t_{k+1})} \right) \right) \frac{\partial \mathbf{x}(t_{k+1})}{\partial \mathbf{x}(t_k)} = \left(\mathbf{C} - \frac{h_k}{2} \left(\mathbf{G} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}(t_k)} \right) \right) \quad (4.49)$$

Computational complexity of the NSM

Given that the main solution approach to the NSM is an iterative one in which the number of iterations is expected to vary from a problem to problem, a good way to estimate the computational complexity of the NSM would be to derive the computational cost per single iteration. The above discussion showed that a single iteration involves the following three components,

- Computing $\mathbf{x}(T)$ using IVP approaches starting from a given initial condition $\mathbf{x}_0^{(i)}$;
- Computing the Jacobian (or sensitivity) matrix of the final state $\mathbf{x}(t)$ w.r.t. the chosen initial condition $\mathbf{x}_0^{(i)}$; and finally
- Solving a linear system of equations in which the Jacobian matrix is the coefficients matrix.

The computational complexity of the first component is determined by the total number of time steps needed to reach $\mathbf{x}(T)$ and the computational cost associated with each step. It is obvious that each time step requires performing NR iteration if either BE or the TR method is used. The convergence of NR iteration at each step is typically fast since the initial guess is very close to the final solution and hence few NR iterations are required at

4.2. STEADY-STATE ANALYSIS USING BOUNDARY-VALUE PROBLEM (BVP)

each time step. Every NR iteration requires solving a linear system of equations whose matrix is the Jacobian matrix of the MNA formulation. The Jacobian matrix of the MNA formulation is typically very sparse since its sparsity pattern is identical to the sparsity pattern of the circuit MNA formulation. Thus the computational complexity of solving the system at each step can be approximated by $O(N^\alpha)$ where $\alpha \approx 1.1 - 1.2$. Therefore the computational complexity of the step is described by $O(mN^\alpha)$, assuming that m is the required number of time steps.

The computational cost of the second component is the major dominant component in the overall computational complexity of a single iteration. The reason for that is simply due to the fact computing the sensitivity matrix requires m matrix-matrix multiplications, as can be seen from (4.47). Unfortunately, those matrices have no sparse structure with few nonzero entries which can be taken advantage of in the multiplication to reduce the computations. Rather, those matrices are typically full matrices with no structurally zero entries. This is a consequence of the fact that those matrices are resulting as a matrix-valued solution for linear systems of equations described by (4.48) or (4.49). Since a full matrix-matrix multiplication is an operation of complexity $O(N^3)$, we expect that the computational complexity of the second component to reach $O(mN^3)$, which is much higher than the first component.

The last component to the overall computational complexity of a single NSM iteration is the linear system solution required at to update the current trial vector, $x_0^{(i)}$, for the initial condition. This is the step where the Jacobian matrix \mathbf{J} is to be factorized. As noted above, the final form for \mathcal{J} is full matrix and therefore its factorization is a process which can only be performed using classical techniques that are $O(N^3)$.

Appreciating the full magnitude of the computational complexity of the NSM still requires estimating the number of time steps m required during the course of numerical integration as this reflects directly on the overall complexity. First recall that both of BE and TR are just examples of general numerical integration for solving differential equations, that enjoy robust stability criteria. Also note that the integration step size is usually dictated by the acceptable error bound prescribed prior to launching the integration process. To keep the numerical error within the acceptable bounds, the numerical integrator will need to follow the waveform closely. This might require reducing the step size if the waveforms exhibit fast variation, and it also implies increasing the step size if the waveform experiences periods of slowing activities. If we denote the average step size by δt , we may produce a rough estimate for δt by considering it inversely proportional to the largest frequency component in the waveform. The rationale behind this is based on the fact the highest frequency components in a waveform are responsible for its showing fast variations. Thus as a first order approximation, we can state

$$\delta t = \frac{\beta}{\omega_{\max}} \quad (4.50)$$

where β is some constant. Thus the number of steps m taken through the period T is expected to be

$$m = \frac{T}{\delta t} = \frac{T\omega_{\max}}{\beta} \quad (4.51)$$

Thus far, we have not specified the type of excitation and whether it is periodic or quasi-periodic. This issue is central to gauging the growth of time steps, m , as will be shown next. We consider this issue by examining the periodic case first.

For periodic excitations, the circuit waveforms are periodic with frequency components at multiples of a single fundamental, say ω_0 . The period, T , in this case is well defined and is given by $T = 2\pi/\omega_0$. Also the maximum frequency content in the waveform is given by the highest harmonic, i.e. $\omega_{\max} = M\omega_0$, where M is a given integer. Thus the number of steps, m , for periodic waveforms is given roughly by,

$$m = \frac{2\pi M}{\beta} \quad (4.52)$$

This result shows that for periodic waveforms, the number of time steps is proportional to the largest harmonic index corresponding to the highest frequency component in the signal spectrum.

We now consider the case of quasi-periodic excitation, where we assume that the input excitation has only two tones ω_1 and ω_2 . The highest frequency component is expected to be given by

$$\omega_{\max} = K_1\omega_1 + K_2\omega_2 \quad (4.53)$$

for some positive integers K_1 and K_2 . Furthermore the common period T_{common} is usually associated with the lowest nonzero frequency which is given by

$$\omega_{\min} = \frac{2\pi}{|\omega_2 - \omega_1|} \quad (4.54)$$

Thus the number of time steps should be given by

$$m = \frac{T_{\text{common}}\omega_{\max}}{\beta} = \frac{K_1\omega_1 + K_2\omega_2}{\beta |\omega_2 - \omega_1|} \quad (4.55)$$

The above key result shows that a huge number of time steps will be needed if the two tones are closely spaced, which is a common occurrence in RF circuit design. Thus the computational cost is expected to become impractical for circuit with quasi periodic excitation.

This point shows that despite the fact that NS is a pure time-domain approach that is supposedly insensitive to the number of harmonics in the waveforms, and is therefore ideal to handle circuits with strong as well as mild nonlinearity, it fails if the circuits is under quasi-tone stimulus.

Chapter **5**

Volterra Series

5.1 Linear Systems and Transfer Function

A quick review of the concept of the transfer function in linear circuits is useful as it will highlight some basic principles that will be discussed for nonlinear circuits. It is well-known that one can always describe a transfer function between the input and the output using the concepts of the Laplace and Fourier transforms. Consider as an example the circuit shown in Figure 5.1. This circuit is a low-pass filter, with an input $v_i(t)$ and an output $v_o(t)$. Using the Laplace transformation, the ratio between the Laplace-domain output to the Laplace-domain input voltage can be described as follows

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{1}{1 + sRC} \quad (5.1)$$

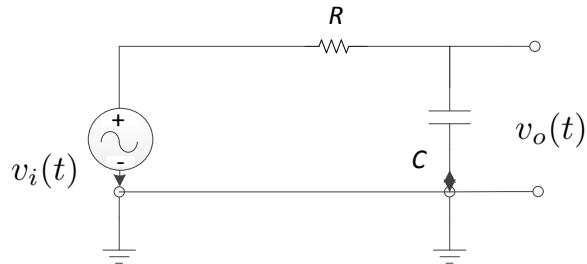


Figure 5.1: A low-pass filter circuit

It is well-known that the inverse Laplace transform of $H(s)$ is the impulse response of

the system. In other words, if $h(t)$ is defined as

$$h(t) = \mathcal{L}^{-1}\{H(s)\} \quad (5.2)$$

then $h(t)$ is the response at the output, in this case $v_o(t)$, if $v_i(t) = \delta(t)$, where $\delta(t)$ is the Dirac delta function. By the same token we have,

$$H(s) = \mathcal{L}^{-1}\{H(s)\} = \int_0^\infty h(\tau)e^{-s\tau}d\tau \quad (5.3)$$

5.1.1 Steady-State Response in Linear Systems via (TF)

It is an interesting exercise to see how one can obtain the steady-state periodical response for periodical input through utilizing the concept of the circuit TF. For that purpose, assume that the input source is a sinusoidal single-tone source given by

$$v_i(t) = A \cos(\lambda_0 t) \quad (5.4)$$

whose Laplace-domain version is given by

$$V_i(s) = A \frac{s}{s^2 + \lambda_0^2} \quad (5.5)$$

The responses (output) voltage in the Laplace-domain is thus given by

$$V_o(s) = H(s) \times V_i(s) \quad (5.6)$$

$$= \frac{1}{1 + sRC} \times A \frac{s}{s^2 + \lambda_0^2} \quad (5.7)$$

Using the idea of partial fractions, one can obtain $v_o(t)$ through the inverse operation of the Laplace transformation, as follows

$$v_o(t) = \frac{A}{1 + \lambda_0^2 R^2 C^2} e^{-t/RC} + \frac{A}{1 + \lambda_0^2 R^2 C^2} \cos(\lambda_0 t) - \frac{A \lambda_0 R C}{1 + \lambda_0^2 R^2 C^2} \sin(\lambda_0 t) \quad (5.8)$$

It is clear that the first term is exponentially decaying term and therefore disappears as

$t \rightarrow \infty$. Thus the steady-state response for $v_o(t)$ is given by

$$v_o(t)|_{\text{steady-state}} = A \left[\underbrace{\frac{1}{1 + \lambda_0^2 R^2 C^2} \cos(\lambda_0 t)}_{H(s)|_{s=j\lambda_0}} - \underbrace{\frac{\lambda_0 R C}{1 + \lambda_0^2 R^2 C^2} \sin(\lambda_0 t)}_{H(s)|_{s=-j\lambda_0}} \right] \quad (5.9)$$

Noting the expression of $H(s)$ in Equation (5.1), we can express $v_o(t)$ at the steady-state by the following expression

$$v_o(t)|_{\text{steady-state}} = \frac{A}{2} \left[H(s)|_{s=j\lambda_0} e^{j\lambda_0 t} + H(s)|_{s=-j\lambda_0} e^{-j\lambda_0 t} \right] \quad (5.10)$$

Thus finding the steady-state response of the circuit in Figure 5.1 required only the knowledge of the transfer path between the input and output ports, $H(s)$, and a simple substitution with the input excitation angular frequency, λ_0 , in $H(s)$ at its both negative and positive values.

It is also important here to remember that the values that $H(s)$ takes on the $j\omega$ axis of the s complex plan can be obtained using the Fourier transformation of the impulse response $h(t)$. Thus

$$H(j\omega) = \int_{-\infty}^{\infty} h(\tau) e^{-j\omega\tau} d\tau \quad (5.11)$$

Given that $h(\tau)$ is always a real waveform, we should have that

$$H(j\omega) = H^*(-j\omega) \quad (5.12)$$

where (\star) is the complex-conjugate operator. Finding $H(j\omega)$ is also done in a similar manner to finding $H(s)$ for linear circuit, through replacing each capacitor C by its phasor impedance $1/j\omega C$, and each inductor L by its phasor impedance $j\omega L$. Figure 5.2 summarizes graphically the basic idea involved in using the circuit TF to compute the steady-state response.

The above discussion made it clear that having a transfer function for a circuit can provide significant savings in computations provided that this transfer function is available in a symbolic form, i.e. in a closed-form expression for general ω . Such an expression can be used to find the steady-state response for any input source as long as it is sinusoidal. This is because all the computations that are needed is the substitution for ω in the TF with the input sinusoidal oscillation frequency $s = \pm j\lambda_0$. In addition, using the fact that the circuit is linear, one can apply the superposition principal to find the steady-state responses

5.1. LINEAR SYSTEMS AND TRANSFER FUNCTION

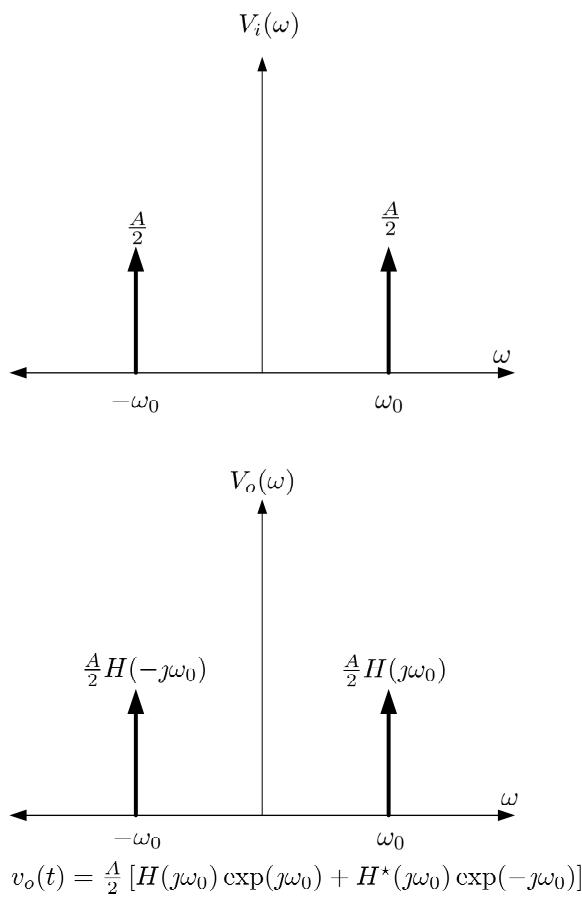


Figure 5.2: Illustrating the role of the circuit TF, $H(s)$ in obtaining the steady-state output of the circuit in response to a sinusoidal input.

for a general periodic input having the form of a summation of sinusoids shown next

$$v_i(t) = V_{i,0} + \sum_{k=1}^K V_{i,k}^C \cos(k\lambda_0 t) + V_{i,k}^S \sin(k\lambda_0 t) \quad (5.13)$$

by finding the response to each term in the above Fourier expansion, using its frequency $k\lambda_0$, and then summing all the responses to form the entire circuit response.

It is desirable to extend the idea of transfer function to nonlinear circuits. Setting aside that obtaining such a transfer function in a symbolic closed-form expression is not a trivial task, even for linear circuits, this notion of transfer function can present us with great advantage. One advantage becomes obvious when the designer is interested in repeating the simulation, in particular in the situations where a certain design parameter needs to be swept over a certain range. Having a transfer function in this case can save the designer having to launch a full simulation process each time the design parameter is changed.

Unfortunately, the concept of TF does not exist for nonlinear circuits. The foremost reason for that is that both the Laplace and Fourier transforms are only applicable to linear circuits.

Another difficulty for extending the concept of transfer function to nonlinear circuits is that the circuit generates additional spectral components to the output that are not present in the input signal, as shown by Figure 5.3. In this situation there is no one-to-one correspondence between the spectral components in the input and output waveforms. Rather, there is a one-to-many transfer paths from the input spectral component at λ_0 to its multiples $k\lambda_0$ that exist at the output.

However, capturing those multiple paths between the input and output is still possible in certain situations using the Volterra and Weiner pioneering work in describing nonlinear systems. This is the subject of the following section.

5.2 Volterra-Series and Small-Signal Analysis

Volterra series is the natural analogue of Taylor series in describing nonlinear systems. Taylor series is used to represent a general nonlinear function in the form of a power series expansion around a certain point of the independent variable. For example if $f(x)$ is a differentiable function at a given point, say $x = a$, then $f(x)$ can be represented by

$$f(x) = f(a) + \frac{\partial f}{\partial x} \Big|_{x=a} (x-a) + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2} \Big|_{x=a} (x-a)^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial x^3} \Big|_{x=a} (x-a)^3 + \dots \quad (5.14)$$

Volterra series can be described in a similar manner, to show that the output of a nonlinear system to a general input signal can be constructed in the form of a series. In fact, if the circuit is memoryless, i.e. without any memroy elements such capacitors or induc-

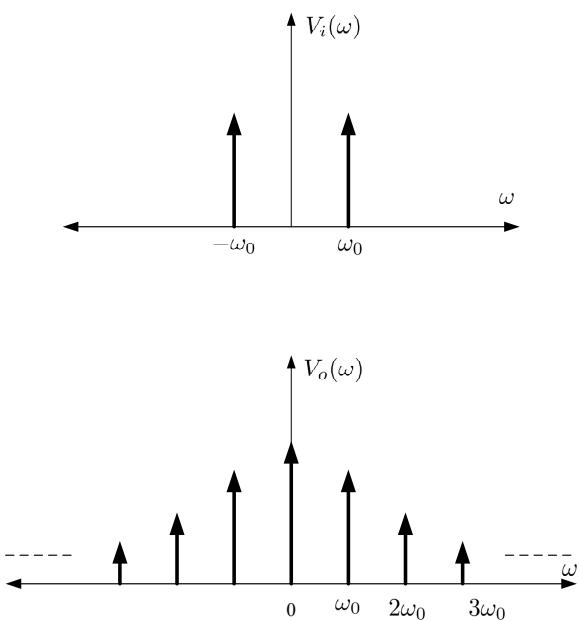


Figure 5.3: Illustrating the response of nonlinear circuits to single-tone sinusoidal input. Here we show that the response of the circuit includes multiples of the frequency spectral components that did not exist in the input.

tors, then its Volterra Series reduce to a power series. In this case, the circuit output can be represented in the form of a power series similar to the one given in (5.14). To illustrate this point, consider the circuit shown in Figure 5.4, where the dependent source is a voltage-controlled current source whose current is given by

$$i = f(v) \quad (5.15)$$

and $f(v)$ is assumed to have the following Taylor series expansion around $v = 0$

$$f(v) = a_1 v + a_2 v^2 + a_3 v^3 + \dots \quad (5.16)$$

The output voltage for this circuit is given by

$$\begin{aligned} v_o(t) &= -R_0 i \\ &= -R_0 [a_1 v + a_2 v^2 + a_3 v^3 + \dots] \\ &= -R_0 a_1 v_i(t) + a_2 v_i(t)^2 + a_3 v_i(t)^3 + \dots \end{aligned} \quad (5.17)$$

which is obviously a power series in the input voltage $v_i(t)$.

If the circuit contains memory elements (e.g. inductances and capacitors), a power series description for the relation between the output and input becomes inadequate, and thus Volterra series becomes the natural extension of the notion of power series to memory circuits.

Volterra series presents a tool to model the transfer path between the input and output of general nonlinear systems in the form of an infinite series of operators that operate on the input signal.

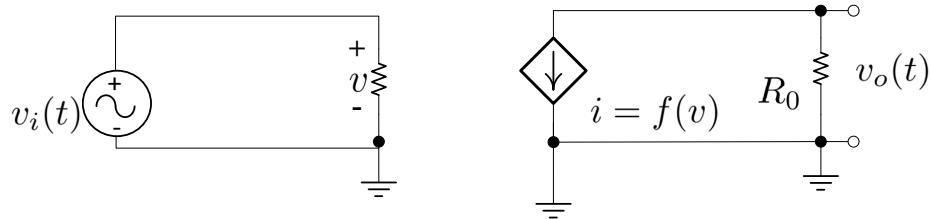


Figure 5.4: An example of memoryless circuit.

Figure 5.5 explains this idea using a block diagram representation. Thus describing the output waveform in terms of the input waveform in the Volterra-series framework is carried out by writing the following infinite series

$$v_o(t) = \mathcal{H}_1 [v_i(t)] + \mathcal{H}_2 [v_i(t)] + \mathcal{H}_3 [v_i(t)] + \dots + \mathcal{H}_n [v_i(t)] + \dots \quad (5.18)$$

where $\mathcal{H}_i [\cdot]$ is understood to be *an operation of the i^{th} order that is performed on the input*.

5.2. VOLTERRA-SERIES AND SMALL-SIGNAL ANALYSIS

Let us explain the concept of the Volterra operators by describing the first three Volterra operators.

5.2.1 First-Order Volterra Operator

The first-order Volterra operator is defined as follows

$$\mathcal{H}_1 [v_i(t)] = \int_{-\infty}^{\infty} h_1(\tau) v_i(t - \tau) d\tau \quad (5.19)$$

which, when compared with the linear system output in response to an input $v_i(t)$, proves to be identical to a linear system response to an the circuit input. This is indeed nothing more than the convolution of the input signal with the impulse response of a system represented by $h_1(\tau)$. $h_1(\tau)$ is, therefore, termed the first-order Volterra kernel.

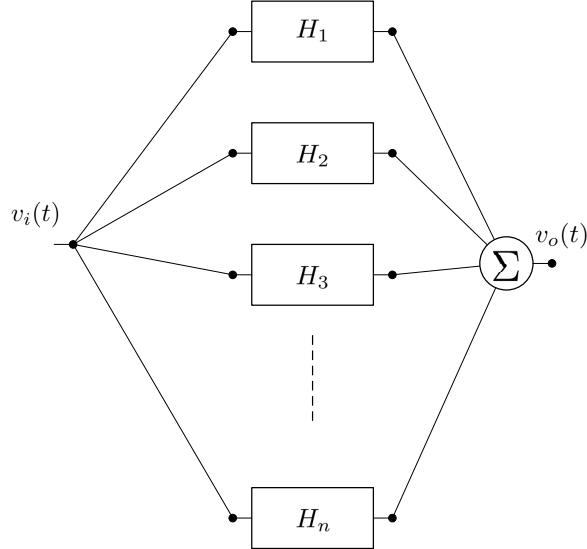


Figure 5.5: Representing the idea of Volterra Series in block diagram.

5.2.2 Second-Order Volterra Operator

The second-order volterra operator is defined using a two-dimensional kernel, $h_2(\tau_1, \tau_2)$

$$\mathcal{H}_2 [v_i(t)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) v_i(t - \tau_1) v_i(t - \tau_2) d\tau_1 d\tau_2 \quad (5.20)$$

5.2.3 Third-Order Volterra Operator

In a similar manner the third-order volterra operator is defined using a three-dimensional kernel, $h_3(\tau_1, \tau_2, \tau_3)$

$$\mathcal{H}_3 [v_i(t)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_3(\tau_1, \tau_2, \tau_3) v_i(t - \tau_1) v_i(t - \tau_2) v_i(t - \tau_3) d\tau_1 d\tau_2 d\tau_3 \quad (5.21)$$

5.2.4 General n^{th} -order Volterra Kernel

In general the n^{th} -order Volterra kernel is given by

$$\begin{aligned} \mathcal{H}_n [v_i(t)] = & \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \tau_3, \dots, \tau_n) \\ & v_i(t - \tau_1) v_i(t - \tau_2) v_i(t - \tau_3) \cdots v_i(t - \tau_n) \\ & d\tau_1 d\tau_2 d\tau_3 \cdots d\tau_n \end{aligned} \quad (5.22)$$

5.3 Constructing a Response to a Sinusoidal input using Volterra Series

Given that our main objective is centered around finding the periodical steady-state response arising from a periodical input, we then assume that our input signal $v_i(t)$ is given by a sinusoidal input with amplitude A and angular frequency λ_0 rad/sec, i.e.

$$v_i(t) = A \cos(\lambda_0 t) \quad (5.23)$$

We next proceed to show how the Volterra series approach may be used to construct the periodical steady-state output response. First the input in Equation (5.23) is expressed in the complex exponential form as follows,

$$v_i(t) = \frac{A}{2} [e^{j\lambda_0 t} + e^{-j\lambda_0 t}] = \sum_{k=-1, k \neq 0}^{k=1} e^{jk\lambda_0 t} \quad (5.24)$$

Next we substitute from (5.24) into the Volterra operators of all orders. Starting with the first-order we substitute (5.24) in (5.19) to obtain,

$$\mathcal{H}_1 [v_i(t)] = \frac{A}{2} \sum_{k=-1}^{k=1} \int_{-\infty}^{\infty} h_1(\tau) e^{jk\lambda_0(t-\tau)} d\tau \quad (5.25)$$

5.3. CONSTRUCTING A RESPONSE TO A SINUSOIDAL INPUT USING VOLTERRA SERIES

and for the second-order operator we get upon substitution into (5.20)

$$\mathcal{H}_2 [v_i(t)] = \frac{A^2}{4} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) e^{jk_1 \lambda_0(t-\tau_1)} e^{jk_2 \lambda_0(t-\tau_2)} d\tau_1 d\tau_2 \quad (5.26)$$

and similarly for the third-order term we have

$$\mathcal{H}_3 [v_i(t)] = \frac{A^3}{8} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} \sum_{k_3=-1}^{k_3=1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_3(\tau_1, \tau_2, \tau_3) e^{jk_1 \lambda_0(t-\tau_1)} e^{jk_2 \lambda_0(t-\tau_2)} e^{jk_3 \lambda_0(t-\tau_3)} d\tau_1 d\tau_2 d\tau_3 \quad (5.27)$$

Finally, the n^{th} -order term is given by

$$\mathcal{H}_n [v_i(t)] = \frac{A^n}{2^n} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} \sum_{k_3=-1}^{k_3=1} \cdots \sum_{k_n=-1}^{k_n=1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \tau_3, \dots, \tau_n) e^{jk_1 \lambda_0(t-\tau_1)} e^{jk_2 \lambda_0(t-\tau_2)} e^{jk_3 \lambda_0(t-\tau_3)} \cdots e^{jk_n \lambda_0(t-\tau_n)} d\tau_1 d\tau_2 d\tau_3 \cdots d\tau_n \quad (5.28)$$

5.3.1 The Fourier Representation of the Volterra Kernels

At this point, it becomes more convenient to introduce the multi-dimensional frequency-domain representation of the Volterra kernels.

The definition of the single-dimensional frequency-domain kernel, $H_1(\omega)$, is the Fourier transformation applied to the first order Volterra kernel $h_1(\tau)$

$$H_1(j\omega) = \int_{-\infty}^{\infty} h_1(\tau) e^{-j\omega\tau} d\tau \quad (5.29)$$

The definition of the two-dimensional frequency-domain kernel $H_2(j\omega_1, j\omega_2)$ is obtained by applying a two-dimensional Fourier transformation to the second-order Volterra kernel in the following manner

$$H_2(j\omega_1, j\omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) e^{-j\omega_1 \tau_1} e^{-j\omega_2 \tau_2} d\tau_1 d\tau_2 \quad (5.30)$$

In a similar manner, the three-dimensional frequency-domain kernel $H_3(j\omega_1, j\omega_2, j\omega_3)$

is obtained from applying a three-dimensional Fourier transform

$$H_3(j\omega_1, j\omega_2, j\omega_3) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_3(\tau_1, \tau_2, \tau_3) e^{-j\omega_1\tau_1} e^{-j\omega_2\tau_2} e^{-j\omega_3\tau_3} d\tau_1 d\tau_2 d\tau_3 \quad (5.31)$$

Finally, and more generally one can use a n -dimensional Fourier transform to define the n -dimensional frequency-domain kernel $H_n(j\omega_1, j\omega_2, \dots, j\omega_n)$ as follows

$$\begin{aligned} H_n(j\omega_1, j\omega_2, j\omega_3, \dots, j\omega_n) &\equiv \\ &\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \tau_3, \dots, \tau_n) \times e^{-j(\tau_1\omega_1 + \tau_2\omega_2 + j\tau_3\omega_3 + \dots + \tau_n\omega_n)} \\ &\quad d\tau_1 d\tau_2 d\tau_3 \dots d\tau_n \end{aligned} \quad (5.32)$$

5.3.2 Response in Terms of Fourier-Based kernels

In this section, we express the output response of the circuit $v_o(t)$ in terms of the Fourier-domain multi-dimensional Volterra kernels introduced in the above section.

Using (5.18), (5.25)- (5.28) the output response of the circuit is written as

$$v_o(t) = \sum_{n=1}^{\infty} w_n(t) \quad (5.33)$$

where

$$\begin{aligned} w_n(t) &= \frac{A^n}{2^n} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} \sum_{k_3=-1}^{k_3=1} \dots \sum_{k_n=-1}^{k_n=1} \\ &\quad \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \tau_3, \dots, \tau_n) e^{-j(k_1\tau_1 + k_2\tau_2 + jk_3\tau_3 + \dots + k_n\tau_n)\lambda_0} \\ &\quad e^{j(k_1+k_2+jk_3+\dots+k_n)\lambda_0 t} d\tau_1 d\tau_2 d\tau_3 \dots d\tau_n \end{aligned} \quad (5.34)$$

The above operators can be written in terms of the multi-dimensional Fourier kernels by taking the terms of $e^{jk_1\lambda_0 t} e^{jk_2\lambda_0 t} e^{jk_3\lambda_0 t} \dots e^{jk_n\lambda_0 t}$ outside the multiple integrals since they are independent of $\tau_1, \tau_2, \tau_3, \dots, \tau_n$. With that operation, the output $v_o(t)$ can be written as

5.3. CONSTRUCTING A RESPONSE TO A SINUSOIDAL INPUT USING VOLTERRA SERIES

follows,

$$w_n(t) = \frac{A^n}{2^n} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} \sum_{k_3=-1}^{k_3=1} \cdots \sum_{k_n=-1}^{k_n=1} e^{j(k_1+k_2+jk_3+\cdots+k_n)\lambda_0 t} \times \\ \underbrace{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \tau_3, \dots, \tau_n) e^{-j(k_1\tau_1+k_2\tau_2+jk_3\tau_3+\cdots+k_n\tau_n)\lambda_0} d\tau_1 d\tau_2 d\tau_3 \cdots d\tau_n}_{n\text{-dimensional Fourier domain kernel } H_n(j\omega_1, j\omega_2, \dots, j\omega_n) \text{ at } \omega_1=k_1\lambda_0, \omega_2=k_2\lambda_0, \dots, \omega_n=k_n\lambda_0} \quad (5.35)$$

Using the definitions of the Fourier-domain Volterra kernels enables us to represent $v_o(t)$ in the following form

$$w_n(t) = \frac{A^n}{2^n} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} \sum_{k_3=-1}^{k_3=1} \cdots \sum_{k_n=-1}^{k_n=1} H_n(k_1\lambda_0, k_2\lambda_0, k_3\lambda_0, \dots, k_n\lambda_0) e^{j(k_1+k_2+jk_3+\cdots+k_n)\lambda_0 t} \quad (5.36)$$

5.3.3 Small-Signal Response using Volterra-Series

In the following, we add the assumption that the amplitude of the small-signal input, A is small enough. Given that the amplitude of each term $w_n(t)$ is proportional to powers of A , then we may justifiably consider only the first three order kernel responses as a good approximation to the entire output response. Thus, for such small A , we express $v_o(t)$ by its first three kernel responses,

$$\begin{aligned} v_o(t) &\approx w_1(t) + w_2(t) + w_3(t) \\ &\approx \frac{A}{2} \sum_{k_1=-1}^{k_1=1} H_1(jk_1\lambda_0) e^{jk_1\lambda_0 t} \\ &+ \frac{A^2}{4} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} H_2(jk_1\lambda_0, jk_2\lambda_0) e^{j(k_1+k_2)\lambda_0 t} \\ &+ \frac{A^3}{8} \sum_{k_1=-1}^{k_1=1} \sum_{k_2=-1}^{k_2=1} \sum_{k_3=-1}^{k_3=1} H_3(jk_1\lambda_0, jk_2\lambda_0, jk_3\lambda_0) e^{j(k_1+k_2+k_3)\lambda_0 t} \end{aligned}$$

Our next goal is to break the response into elementary components that can be easily identified. To this end, we substitute with the different values of $k_1, k_2, k_3 = \pm 1$ and collect

those terms having identical exponential terms, i.e., e^0 , $e^{\pm j\lambda_0 t}$ and $e^{\pm 2j\lambda_0 t}$. this results in the following expression where similiar terms (i.e., those multiplied by e^0 , $e^{\pm j\lambda_0 t}$ and $e^{\pm 2j\lambda_0 t}$) are grouped together.

$$\begin{aligned}
 v_o(t) &= \frac{A^2}{4} [H_2(-j\lambda_0, j\lambda_0) + H_2(j\lambda_0, -j\lambda_0)], \rightarrow \text{ This is a pure DC term.} \\
 &+ \frac{A}{2} [H_1(j\lambda_0)e^{j\lambda_0 t} + H_1(-j\lambda_0)e^{-j\lambda_0 t}] \\
 &+ \frac{A^3}{8} [H_3(-j\lambda_0, j\lambda_0, j\lambda_0) + H_3(j\lambda_0, -j\lambda_0, j\lambda_0) + H_3(j\lambda_0, j\lambda_0, -j\lambda_0)] e^{j\lambda_0 t} \\
 &+ \frac{A^3}{8} [H_3(-j\lambda_0, -j\lambda_0, j\lambda_0) + H_3(j\lambda_0, -j\lambda_0, -j\lambda_0) + H_3(-j\lambda_0, j\lambda_0, -j\lambda_0)] e^{-j\lambda_0 t} \quad \left. \right\} \text{First-order harmonic, } \lambda_0 \\
 &+ \frac{A^2}{4} [H_2(-j\lambda_0, -j\lambda_0)e^{-2j\lambda_0 t} + H_2(j\lambda_0, j\lambda_0)e^{2j\lambda_0 t}] \quad \left. \right\} \text{Second-order harmonic, } 2\lambda_0 \\
 &+ \frac{A^3}{8} [H_3(-j\lambda_0, -j\lambda_0, -j\lambda_0)e^{-3j\lambda_0 t} + H_3(j\lambda_0, j\lambda_0, j\lambda_0)e^{3j\lambda_0 t}] \quad \left. \right\} \text{Third-order harmonic, } 3\lambda_0
 \end{aligned} \tag{5.37}$$

The above expression can be further simplified by taking into consideration two important properties of the Frequency-domain multi-dimensional kernels $H_n(\cdot)$. Those properties are

- **Complex-Conjugateness.** Flipping the sign of any single argument produces the complex-conjugate. For example, $H_2(j\lambda_1, j\lambda_2) = H_2^*(-j\lambda_1, j\lambda_2)$. This property follows from the fact that the time-domain multi-dimensional kernels $h_n(\tau_1, \dots, \tau_n)$ are real.
- **Symmetry with respect to the frequency arguments.** This property is described briefly by simply stating that swapping any two frequency arguments does not change the numerical value of the frequency-domain kernels. Thus, for example

$$\begin{aligned}
 H_5(j\omega_1, j\omega_2, j\omega_3, j\omega_4, j\omega_5) &= H_5(j\omega_2, j\omega_1, j\omega_3, j\omega_4, j\omega_5) \\
 &= H_5(j\omega_3, j\omega_2, j\omega_1, j\omega_4, j\omega_5) \\
 &= H_5(j\omega_4, j\omega_2, j\omega_3, j\omega_1, j\omega_5) \\
 &= H_5(j\omega_5, j\omega_2, j\omega_3, j\omega_4, j\omega_1) \\
 &= H_5(j\omega_1, j\omega_3, j\omega_2, j\omega_4, j\omega_5) \\
 &= H_5(j\omega_1, j\omega_4, j\omega_3, j\omega_2, j\omega_5) \\
 &\vdots
 \end{aligned} \tag{5.38}$$

This property can be proved easily from the definition of the frequency-domain kernels in (5.32).

5.4. THE METHOD OF NONLINEAR CURRENTS

The above two properties can be used to simplify the output response in (5.37) considerably. To see how this is done, recall that for any complex number K , it is always the case that $K + K^* = 2\Re\{K\}$ and that $Ke^{j\omega t} + K^*e^{-j\omega t} = 2|K|\cos(\omega t + \angle K)$, where $\angle A = \tan^{-1}\left(\frac{\Im\{A\}}{\Re\{A\}}\right)$. Therefore, the final form for $v_o(t)$ is given by

$$\begin{aligned} v_o(t) &= \frac{1}{2}A^2\Re\{H_2(j\lambda_0, -j\lambda_0)\} \\ &+ A|H_1(j\lambda_0 t)|\cos(\lambda_0 t + \angle H_1(j\lambda_0)) \\ &+ \frac{3}{4}A^3|H_3(j\lambda_0, j\lambda_0, -j\lambda_0)|\cos(\lambda_0 t + \angle H_3(j\lambda_0, j\lambda_0, -j\lambda_0)) \\ &+ \frac{1}{2}A^2|H_2(j\lambda_0, j\lambda_0)|\cos(2\lambda_0 t + \angle H_2(j\lambda_0, j\lambda_0)) \\ &+ \frac{1}{4}A^3|H_3(j\lambda_0, j\lambda_0, j\lambda_0)|\cos(3\lambda_0 t + \angle H_3(j\lambda_0, j\lambda_0, j\lambda_0)) \end{aligned} \quad (5.39)$$

Figure 5.6 shows the components of the spectrum in the output signal and provides the amplitudes obtained from the above Volterra-based analysis in (5.37).

It is obvious from this figure, and from the final simplified response in (5.39) that if we have access to the three first kernels of the system in the frequency-domain, i.e., $H_1(j\omega)$, $H_2(j\omega_1, j\omega_2)$ and $H_3(j\omega_1, j\omega_2, j\omega_3)$ that we can easily compute the spectral components in the output response $v_o(t)$, by simple substitution for $\omega_1, \omega_2, \omega_3$ with appropriate values of $\pm\lambda_0$. Furthermore, if those kernels are available in closed-form expressions, i.e. in the form of multi-dimensional analytical expressions in ω_1, ω_2 and ω_3 then we would be able to compute the output for any excitation with an arbitrary angular frequency ω .

However, given that this task of finding analytical expressions in which the frequency variable appears symbolically, is far from being trivial even for linear circuits of modest size, we anticipate that the complications for nonlinear circuits to be horrendous.

Nonetheless, there is a way to approach this task from a numerical standpoint. The most common and classical approach to handle this task is known as the method of nonlinear currents. For simple circuits, this task can also yield the analytical expression for the first three frequency-domain multi-dimensional kernels, H_1 , H_2 and H_3 . This approach is considered in the following text.

5.4 The Method of Nonlinear Currents

In this section, we illustrate the immense difficulties involved in deriving closed-form analytical expressions for the first three Fourier-domain kernels, by considering this task in the context of a very simple circuit. The method used for this purpose is known as the method of nonlinear currents.

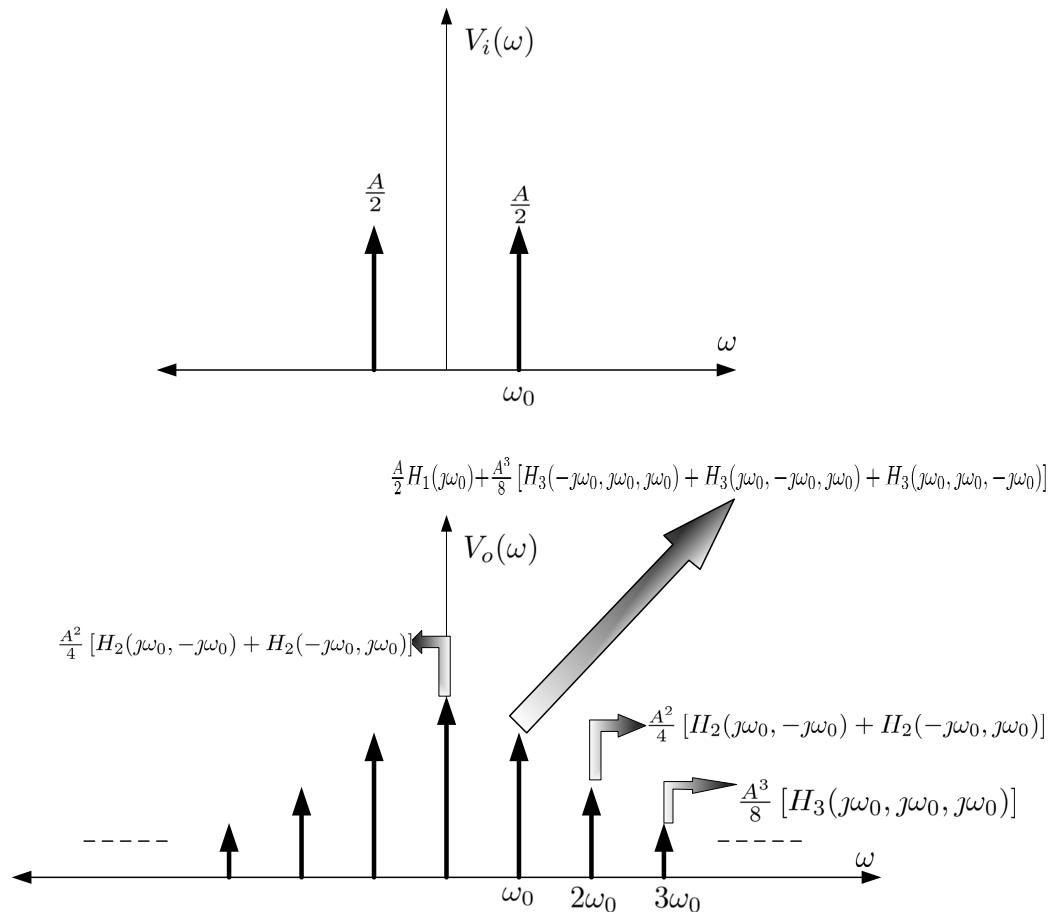


Figure 5.6: Illustrating the low-order spectral components in the output signal and showing their amplitudes using the multi-dimensional frequency-domain Volterra kernels. The information in this figure is taken from Equation (5.37).

5.4.1 Analytical Expressions for the Frequency-domain kernels

As a first step to understand the basic concept in the method of nonlinear currents, we will use it to derive the analytical closed-form expression for the frequency-domain multi-dimensional kernels for a simple circuit. For this purpose we consider the circuit shown in Figure 5.6, where the input signal $v_i(t)$ is assumed to be sinusoidal, i.e.

$$v_i(t) = A \cos(\lambda_0 t) \quad (5.40)$$

Here we assume that the amplitude A is relatively small to justify writing $v_o(t)$ using the only first three kernels. Thus all what we need right now is derive the first three frequency-domain multi-dimensional kernels $H_1(\cdot)$, $H_2(\cdot, \cdot)$, and $H_3(\cdot, \cdot, \cdot)$. The first step to this end is to expand the nonlinear conductance in this circuit $f(v)$ in a Taylor series around the DC biasing voltage, i.e.

$$\begin{aligned} i = f(v) = f(v_{\text{DC}}) + \frac{\partial f}{\partial v} \Big|_{v=v_{\text{DC}}} (v - v_{\text{DC}}) + \frac{1}{2!} \frac{\partial^2 f}{\partial v^2} \Big|_{v=v_{\text{DC}}} (v - v_{\text{DC}})^2 \\ + \frac{1}{3!} \frac{\partial^3 f}{\partial v^3} \Big|_{v=v_{\text{DC}}} (v - v_{\text{DC}})^3 + \dots \end{aligned} \quad (5.41)$$

As can be seen from Figure 5.7, there is no DC source in the circuit, and thus

$$v_{\text{DC}} = 0.$$

Also, assuming that the nonlinear conductance draws zero current when there is zero voltage applied to it, i.e., $f(0) = 0$, simplifies the current in the nonlinear element to the following, upon truncating the infinite Taylor series after the first three terms

$$i = f(v) = g_1 v + g_2 v^2 + g_3 v^3 \quad (5.42)$$

where

$$g_1 = \frac{\partial f}{\partial v} \Big|_{v=0} \quad (5.43)$$

$$g_2 = \frac{\partial^2 f}{\partial v^2} \Big|_{v=0} \quad (5.44)$$

$$g_3 = \frac{\partial^3 f}{\partial v^3} \Big|_{v=0} \quad (5.45)$$

The fact that we can represent the nonlinear current i as a summation of three terms as

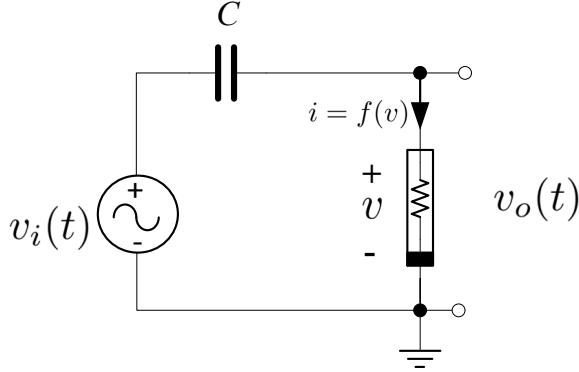


Figure 5.7: A simple nonlinear circuit with a memory element.

shown in Equation (5.42), and the fact that $v = v_o(t)$ for this particular example suggests that we can modify the circuit as shown in Figure 5.8.

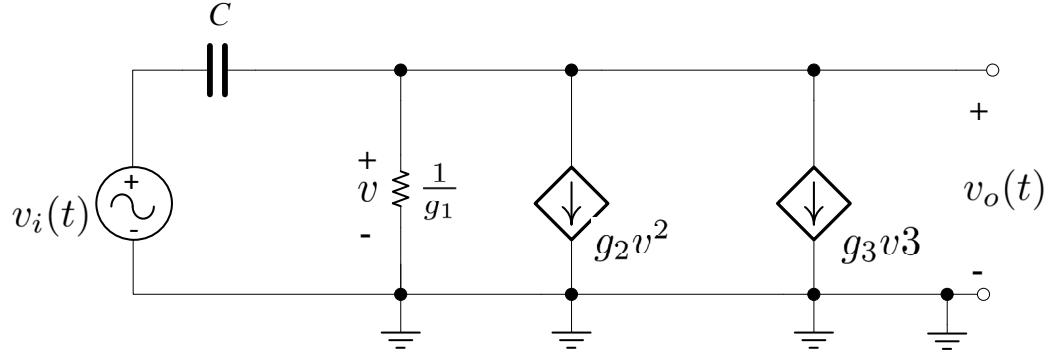


Figure 5.8: Modifying the circuit in Figure 5.7 by replacing the nonlinear current drawn from the nonlinear conductance by three dependent sources in parallel using the simplified series in Equation (5.42)

Now let us write the output voltage, $v_o(t)$, i.e. the response of the circuit that we seek to compute, using the form we anticipate it, i.e. based on its Volterra-series representation as derived earlier in Equation (5.37). For that purpose, we express $v_o(t)$ as follows

$$v_o(t) = u_1(t) + u_2(t) + u_3(t) \quad (5.46)$$

where $u_i(t)$ represent the terms the resulting from i^{th} -order Volterra kernel. For example, $u_1(t)$ represents those terms that are proportional to $H_1(\cdot)$ and hence

$$u_1(t) = \frac{1}{2}A [H_1(-j\lambda_0)e^{-j\lambda_0 t} + H_1(j\lambda_0)e^{j\lambda_0 t}] \quad (5.47)$$

5.4. THE METHOD OF NONLINEAR CURRENTS

Similarly, $u_2(t)$ groups all the terms that arise from $H_2(\cdot, \cdot)$, which makes it given by

$$\begin{aligned} u_2(t) &= \frac{1}{4}A^2 [H_2(-\jmath\lambda_0, \jmath\lambda_0) + H_2(\jmath\lambda_0, -\jmath\lambda_0)] \\ &+ \frac{1}{4}A^2 [H_2(-\jmath\lambda_0, -\jmath\lambda_0)e^{-2\jmath\lambda_0 t} + H_2(\jmath\lambda_0, \jmath\lambda_0)e^{2\jmath\lambda_0 t}] \end{aligned} \quad (5.48)$$

Notice here that $u_2(t)$ includes a DC term and a term proportional to twice the input angular frequency $2\lambda_0$.

Finally, $u_3(t)$ is the term that groups all the terms arising from the third-order kernel $H_3(\cdot, \cdot, \cdot)$.

$$\begin{aligned} u_3(t) &= \frac{3}{8}A^3 [H_3(-\jmath\lambda_0, -\jmath\lambda_0, \jmath\lambda_0)e^{-\jmath\lambda_0 t} + H_3(\jmath\lambda_0, \jmath\lambda_0, -\jmath\lambda_0)e^{\jmath\lambda_0 t}] \\ &+ \frac{1}{8}A^3 [H_3(-\jmath\lambda_0, -\jmath\lambda_0, -\jmath\lambda_0)e^{-3\jmath\lambda_0 t} + H_3(\jmath\lambda_0, \jmath\lambda_0, \jmath\lambda_0)e^{3\jmath\lambda_0 t}] \end{aligned} \quad (5.49)$$

which has two spectral components at λ_0 and $3\lambda_0$. Figure 5.9 presents the spectral components for each of the signals $u_1(t)$, $u_2(t)$ and $u_3(t)$.

Next, we use the fact that the voltage v for the nonlinear element is the same output voltage, $v = v_o(t)$ for this particular circuit example, to rewrite the current i as follows

$$i(t) = g_1v_o(t) + g_2v_o(t)^2 + g_3v_o(t)^3 \quad (5.50)$$

Now let us substitute from (5.46) into (5.50),

$$\begin{aligned} i(t) &= g_1 \times (u_1(t) + u_2(t) + u_3(t)) + g_2 \times (u_1(t) + u_2(t) + u_3(t))^2 \\ &\quad + g_3 \times (u_1(t) + u_2(t) + u_3(t))^3 \end{aligned} \quad (5.51)$$

The above expression will surely have terms multiplied by powers of A that can reach up to order 9, i.e. A^9 . Since, we have previously assumed that the amplitude A is small, we may retain only those terms multiplied by those powers of A that are equal to or less than 3, and consider those terms adequate to capture the current $i(t)$ with acceptable accuracy.

Thus neglecting the terms in (5.51) which would be responsible for generating terms with powers of A higher than 3 in results in the following expression for the current

$$i(t) = \underbrace{g_1 \times (u_1(t) + u_2(t) + u_3(t))}_{\text{a linear component} \equiv g_1v_o(t)} + \underbrace{g_2 \times u_1(t)^2 + 2g_2 \times u_1(t)u_2(t) + g_3 \times u_1(t)^3}_{\text{nonlinear components}} \quad (5.52)$$

Here we can make an observation using the insights revealed by Equation (5.52). A

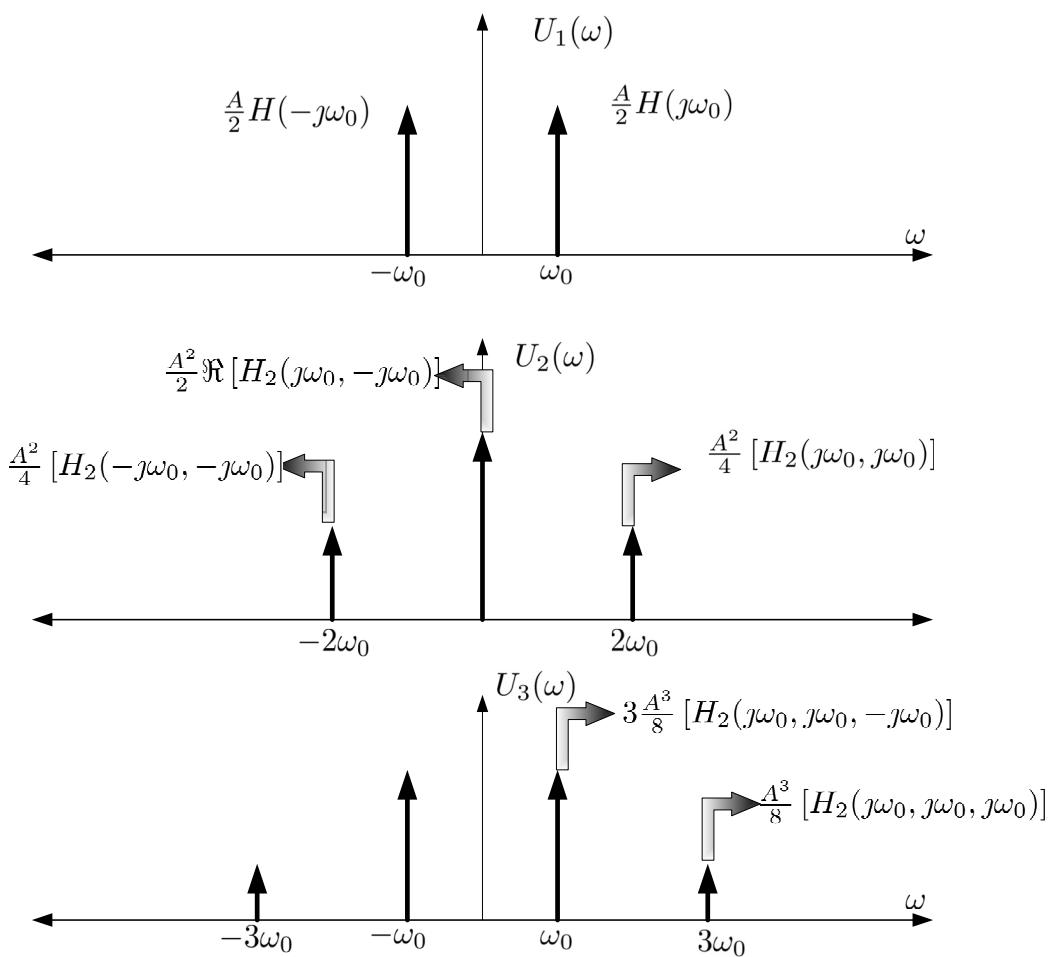


Figure 5.9: Illustrating the spectral components of the three components $U_1(\omega)$, $U_2(\omega)$ and $U_3(\omega)$

5.4. THE METHOD OF NONLINEAR CURRENTS

closer inspection of Equation (5.52) reveals that the current in the nonlinear conductance, $i(t)$ has indeed four components. The first component is proportional to the the voltage $v_o(t)$, i.e., it is linear in the applied voltage $v = v_o(t)$ and therefore can be represented by a simple resistor whose value is $\frac{1}{g_1}$. The other three components are nonlinear dependent voltage-controlled current sources. This observation is best represented by the redrawing the equivalent circuit as shown in Figure 5.10.

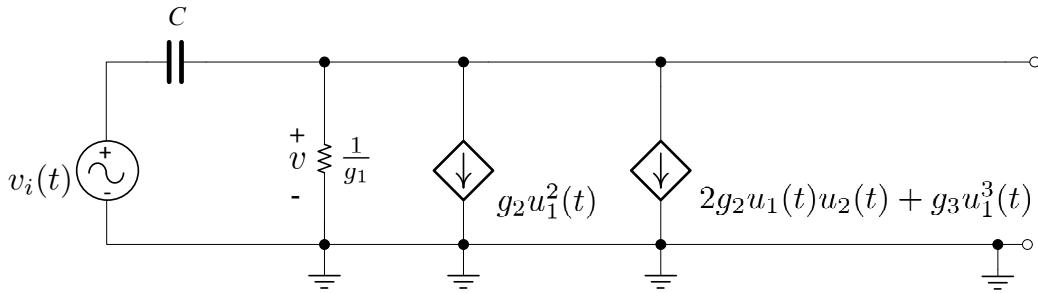


Figure 5.10: Redrawing the equivalent circuit of Figure 5.8 using the insight provided by Equation (5.52).

It is very important to note here that the circuit in Figure 5.10 is now a linear circuit, but with multiple dependent *nonlinear sources*. Although, it is not possible, in principle, to apply the principle of superposition to nonlinear circuits, the nature of dependent sources in this circuit will enable us to obtain the response $v_o(t)$ using the principle of superposition. To illustrate the main idea, let us proceed to compute the response pretending for the moment that the superposition principle is applicable. Thus we split the circuit into three circuits that are shown in Figures 5.11, 5.12 and 5.13, where the response in each circuit is denoted by $v_{o1}(t)$, $v_{o2}(t)$ and $v_{o3}(t)$, respectively. According to principle of superposition, the response voltage $v_o(t)$ of the circuit in Figure 5.10 can be obtained by adding those three responses from the three circuits in Figures 5.11, 5.12 and 5.13. Thus,

$$v_o(t) = v_{o1}(t) + v_{o2}(t) + v_{o3}(t) \quad (5.53)$$

Before we proceed to compute $v_{o1}(t)$, $v_{o2}(t)$ and $v_{o3}(t)$, we remark that it appears that $v_{o1}(t)$ is the only one that can be presently computed since it is the response of a linear circuit whose input source $v_i(t)$ is the only *known* source, while the other two responses $v_{o2}(t)$ and $v_{o3}(t)$ in the other two circuits are responses of the same circuit but to two sources that are hitherto unknown, since they depend on $u_1(t)$, $u_2(t)$. However, this problem will be resolved as shown shortly. We now focus on computing $v_{o1}(t)$.

Computing $v_{o1}(t)$

Being a response of a linear circuit due to the sinusoidal stimulus $v_i(t)$, see Figure 5.11, the linear system theory based on Laplace-transform can be invoked to express $v_{o1}(t)$ using the circuit transfer function. The Laplace-domain transfer function in this circuit is given by

$$\begin{aligned} H(s) &= \frac{sRC}{1+sRC} \\ &= \frac{s\frac{1}{g_1}C}{1+s\frac{1}{g_1}C} \\ &= \frac{sC}{g_1+sC} \end{aligned} \quad (5.54)$$

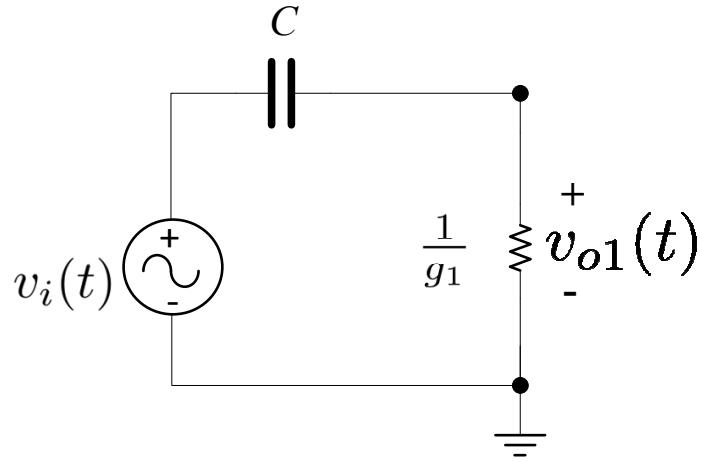


Figure 5.11: Applying the superposition principle to the circuit in Figure 5.10: Circuit 1.

Using an approach similar to the one given in Section 5.1 for linear circuits, we can write the output of the circuit in Figure 5.11 due to an input $v_i(t) = A \cos(\lambda_0 t)$ as follows

$$\begin{aligned} v_{o1}(t) &= \frac{1}{2}A \left[H(s)|_{s=-j\lambda_0} e^{-j\lambda_0 t} + H(s)|_{s=j\lambda_0} e^{j\lambda_0 t} \right] \\ &= \frac{1}{2}A \left[\frac{j\lambda_0 C}{g_1 + j\lambda_0 C} e^{-j\lambda_0 t} + \frac{-j\lambda_0 C}{g_1 - j\lambda_0 C} e^{j\lambda_0 t} \right] \end{aligned} \quad (5.55)$$

Let us stop here for a moment and make the interesting observation that $v_{o1}(t)$ shown above is nothing other than $u_1(t)$ itself!

To see that this is true indeed, consider Equations (5.46)-(5.49) and notice that it is

5.4. THE METHOD OF NONLINEAR CURRENTS

only $u_1(t)$ that is responsible for producing a spectral components at $\pm\lambda_0$ in $v_o(t)$. On the other hand, we can rest assured that it is only $v_{o1}(t)$ that has a spectral component at $\pm\lambda_0$. This observation follows immediately when we see that both $v_{o2}(t)$ and $v_{o3}(t)$ are responses of a linear circuit due to the sources that have spectral components at $\pm 2\lambda_0$ and $\pm 3\lambda_0$, or $u_1(t)^2, u_2(t), u_3(t)$. Hence, it is impossible that $v_{o2}(t)$ or $v_{o3}(t)$ will have spectral components at $\pm\lambda_0$. This observation allows us to assert that

$$v_{o1}(t) = u_1(t) \quad (5.56)$$

Hence the solution that is obtained from the circuit in Figure 5.11 $v_{o1}(t)$ is nothing but the source used in the second circuit in Figure 5.12 (of course after squaring and multiplying it by g_2 , i.e., $g_2u_1(t)^2$). Thus solving the first linear circuit of Figure 5.11 provides us with the key we need to proceed to analyze the next circuit of Figure 5.12.

Using Equation (5.56) into the left-side of Equation (5.55), and comparing the result with Equation (5.47) leads to the important conclusion regarding the first-order kernel $H_1(\cdot)$, which is

$$H_1(j\omega) = \frac{j\lambda_0 C}{g_1 + j\lambda_0 C} \quad (5.57)$$

This indeed the transfer function of the so-called linearized circuit, i.e. the same nonlinear circuit but after replacing the nonlinear conductance with a linear resistor having a resistance equal to the reciprocal of the derivative of the current with respect to the voltage, calculated at the DC operating voltage of the circuit.

Therefore, we conclude from the above discussion an essential fact and that is: the first-order frequency-domain Volterra kernel, $H_1(j\omega)$ of a nonlinear circuit is simply the frequency-domain transfer function of the linearized circuit, i.e. the circuit in which the nonlinear conductance(s) have been replaced by a resistor, whose resistance is equal to the reciprocal of the derivative of this nonlinear conductance calculated at the DC operating point.

Let us now carry the result we obtained in Equation (5.56) to move and find the response of the second circuit in Figure 5.12.

Computing $v_{o2}(t)$

By the same token, we see that in Figure 5.12, $v_{o2}(t)$ is the response of the same linear circuit, composed of the resistor $\frac{1}{g_1}$ and capacitor C , to a the input source $g_2u_1(t)^2$.

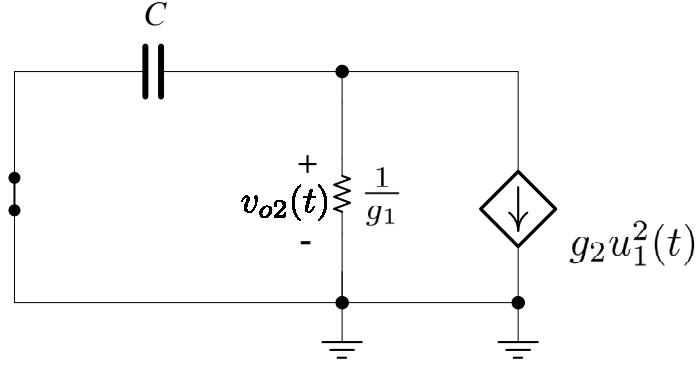


Figure 5.12: Applying the superposition principle to the circuit in Figure 5.10: Circuit 2.

First, we expand $u_1(t)^2$ from (5.47) as follows

$$\begin{aligned} u_1(t)^2 = & \frac{A^2}{4} H_1(-j\lambda_0)^2 e^{-2j\lambda_0 t} \\ & + \frac{A^2}{2} H_1(-j\lambda_0) H_1(j\lambda_0) \\ & + \frac{A^2}{4} H_1(j\lambda_0)^2 e^{2j\lambda_0 t} \end{aligned} \quad (5.58)$$

Therefore, we see that the current source $g_2 u_1(t)^2$ that stimulates the linear circuit has a spectral component at the DC, i.e. $\omega = 0$ and the frequencies $\omega = \pm 2\lambda_0$. To find the response $v_{o2}(t)$, we will need to compute the transfer function between the current source and the voltage across the resistor. The transfer function from a current to a voltage is an impedance transfer function, and for this circuit is given by the parallel combination of $1/g_1$ and $1/(sC)$, i.e.

$$Z(s) = \frac{\frac{1}{sC} \times \frac{1}{g_1}}{\frac{1}{sC} + \frac{1}{g_1}} = \frac{1}{g_1 + sC} \quad (5.59)$$

Using $Z(s)$, the Laplace-domain of $v_{o2}(t)$, is given by multiplying the Laplace-domain of the current stimulus $g_2 u_1(t)^2$ by $Z(s)$. Since the current stimulus has the components given by (5.58), the concept of linear system utilized in Section 5.1 can be used again to write

$v_{o2}(t)$ directly as

$$\begin{aligned} v_{o2}(t) = & Z(-2\jmath\lambda_0) \times \frac{A^2}{4} H_1(-\jmath\lambda_0)^2 e^{-2\jmath\lambda_0 t} \\ & + Z(0) \times \frac{A^2}{2} H_1(-\jmath\lambda_0) H_1(\jmath\lambda_0) \\ & + Z(2\jmath\lambda_0) \times \frac{A^2}{4} H_1(-\jmath\lambda_0)^2 e^{2\jmath\lambda_0 t} \end{aligned} \quad (5.60)$$

We can use the same approach that we used in computing $v_{o1}(t)$ above to prove that

$$v_{o2}(t) = u_2(t) \quad (5.61)$$

Thus, substituting from Equation (5.60) and Equation (5.48) into Equation (5.61) will lead to the following result

$$\begin{aligned} H_2(\jmath\lambda_0, \jmath, \lambda_0) &= Z(2\jmath\lambda_0) \times H_1(\jmath\lambda_0)^2 \\ &= \frac{1}{g_1 + 2\jmath\lambda_0 C} H_1(\jmath\lambda_0)^2 \end{aligned} \quad (5.62)$$

and

$$\begin{aligned} H_2(\jmath\lambda_0, -\jmath\lambda_0) &= Z(0) \times H_1(-\jmath\lambda_0) H_1(\jmath\lambda_0) \\ &= \frac{1}{g_1} H_1(-\jmath\lambda_0) H_1(\jmath\lambda_0) \end{aligned} \quad (5.63)$$

This is indeed the closed-form analytical expression that we seek for $H_2(\jmath\omega_1, \jmath\omega_2)$ obtained at $\omega_1 = \lambda_0$ and $\omega_2 = -\lambda_0$.

Computing $v_{o3}(t)$

Now having found both $u_1(t)$ and $u_2(t)$ we have all we need to compute the response $v_{o3}(t)$ of circuit the third in Figure 5.13. To this end, we expand $u_1(t)^3$ and $u_1(t)u_2(t)$ using their expressions in terms of the second- and third-order Volterra kernels in Equations (5.48) and (5.49). We commence with $u_1(t)^3$.

$$\begin{aligned} u_1(t)^3 = & \frac{A^3}{8} \left[\right. \\ & \left. H_1(+\jmath\lambda_0)^3 e^{+3\jmath\lambda_0 t} + 3H_1(\jmath\lambda_0)^2 H_1(-\jmath\lambda_0) e^{+\jmath\lambda_0 t} \right. \\ & \left. + 3H_1(+\jmath\lambda_0) H_1(-\jmath\lambda_0)^2 e^{-\jmath\lambda_0 t} + H_1(\jmath\lambda_0)^3 e^{-3\jmath\lambda_0 t} \right] \end{aligned} \quad (5.64)$$

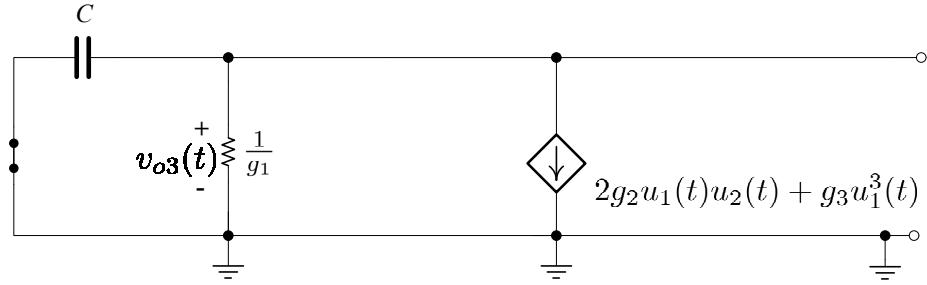


Figure 5.13: Applying the superposition principle to the circuit in Figure 5.10: Circuit 3.

Next we consider expanding $u_1(t)u_2(t)$, which results

$$\begin{aligned}
 u_1(t)u_2 = & \frac{A^3}{8} \left[H_1(+j\lambda_0)H_2(+j\lambda_0, -j\lambda_0)e^{+j\lambda_0 t} \right. \\
 & + H_1(-j\lambda_0)H_2(+j\lambda_0, +j\lambda_0)e^{+j\lambda_0 t} \\
 & + H_1(+j\lambda_0)H_2(-j\lambda_0, +j\lambda_0)e^{+j\lambda_0 t} \\
 & + H_1(-j\lambda_0)H_2(+j\lambda_0, -j\lambda_0)e^{-j\lambda_0 t} \\
 & + H_1(-j\lambda_0)H_2(-j\lambda_0, +j\lambda_0)e^{-j\lambda_0 t} \\
 & + H_1(+j\lambda_0)H_2(-j\lambda_0, -j\lambda_0)e^{-j\lambda_0 t} \\
 & + H_1(+j\lambda_0)H_2(+j\lambda_0, +j\lambda_0)e^{+3j\lambda_0 t} \\
 & \left. + H_1(-j\lambda_0)H_2(-j\lambda_0, -j\lambda_0)e^{-3j\lambda_0 t} \right] \quad (5.65)
 \end{aligned}$$

The input stimulus in the third circuit is therefore given by $2g_2u_1(t)u_2(t) + g_3u_1(t)^3$. To avoid cluttering the presentation with the large number of terms that result from (5.64) and (5.65), we will represent the components in the stimulus in the third circuit $2g_2u_1(t)u_2(t) + g_3u_1(t)^3$ by the frequency components in the following form

$$2g_2u_1(t)u_2(t) + g_3u_1(t)^3 = L_{-3}e^{-3j\lambda_0 t} + L_{-1}e^{-j\lambda_0 t} + L_{+1}e^{+j\lambda_0 t} + L_{+3}e^{+3j\lambda_0 t} \quad (5.66)$$

where $L_{\pm 3}$ and $L_{\pm 1}$ group all positive and negative terms that are multiplied by 3rd and 1st

5.5. EXTENSION TO MULTI-TONE ANALYSIS

harmonics, respectively. For example,

$$L_{+3} = \frac{A^3}{8} [H_1(+j\lambda_0)^3 + H_1(+j\lambda_0)H_2(+j\lambda_0, +j\lambda_0)] \quad (5.67)$$

$$L_{-3} = \frac{A^3}{8} [H_1(-j\lambda_0)^3 + H_1(-j\lambda_0)H_2(-j\lambda_0, -j\lambda_0)] \quad (5.68)$$

Now the Laplace-domain transform $v_{o3}(t)$ can be obtained by multiplying the Laplace-domain of the current source $2g_2u_1(t)u_2(t) + g_3u_1(t)^3$ by the Laplace-domain of impedance $Z(s)$. Since the current source is sinusoidal with spectral components at $\pm\lambda_0$ and $\pm 3\lambda_0$, $v_{o3}(t)$ can be written directly in the time-domain using the technique given in Section 5.1 for linear circuits. Thus, we have

$$\begin{aligned} v_{o3}(t) = & Z(-3j\lambda_0)L_{-3}e^{3j\lambda_0 t} + Z(-j\lambda_0)L_{+1}e^{-j\lambda_0 t} \\ & + Z(+j\lambda_0)L_{+1}e^{j\lambda_0 t} + Z(+3j\lambda_0)L_{+1}e^{+3j\lambda_0 t} \end{aligned} \quad (5.69)$$

A similar argument can be made to show that

$$v_{o3}(t) = u_3(t) \quad (5.70)$$

Substituting from (5.49) and (5.69) into (5.70) reveals the closed-form analytical expression for $H_3(j\omega_1, j\omega_2, j\omega_3)$ that we set out to find for the simple circuit in Figure 5.7

$$H_3(j\lambda_0, j\lambda_0, j\lambda_0) = L_3Z(+3j\lambda_0) \quad (5.71)$$

$$H_3(j\lambda_0, j\lambda_0, -j\lambda_0) = L_1Z(j\lambda_0) \quad (5.72)$$

$$H_3(-j\lambda_0, -j\lambda_0, j\lambda_0) = L_{-3}Z(-3j\lambda_0) \quad (5.73)$$

5.5 Extension to Multi-Tone Analysis

The reader might object that the Volterra kernels obtained using the above analysis are only obtained at diagonal points. Taking as an example to this objection the frequency-domain second-order Volterra kernel obtained for the circuit in Figure 5.7 and given in Equation (5.62), we can see that the expression for $H_2(j\omega_1, j\omega_2)$ is only given when both arguments are both equal to λ_0 . We might have hoped to derive a general expression for $H_2(j\omega_1, j\omega_2)$ in which ω_1 and ω_2 appear explicitly as two independent variables.

This objection is only a result of having decided to use a simplified analysis by considering only a single-tone input excitation. In this section, we show the more general expression in which the arguments of the Fourier Volterra kernel are totally independent by starting the derivation with multi-tone input.

Towards this goal, we now restart the analysis by a m -tone quasi-periodic input, for the

same circuit, of the form,

$$\begin{aligned} v_i(t) &= \sum_{k=1}^m A_k \cos(\lambda_k t) \\ &= \sum_{\substack{k=-m, \\ k \neq 0}}^m \frac{A_k}{2} e^{-j\lambda_k t} \end{aligned} \quad (5.74)$$

where $\lambda_k, k = 1 \dots, m$ are the angular frequencies in rad/sec, and $\lambda_{-p} = -\lambda_p, A_{-p} = A_p$ for any positive integer $p = 1, \dots, m$.

Next we substitute from (5.74) into (5.19), (5.20), and (5.21), to get the first three Volterra kernels as follows

$$\begin{aligned} w_1(t) &= \mathcal{H}_1[v_i(t)] \\ &= \sum_{\substack{k=-m, \\ k \neq 0}}^m A_k H_1(-j\lambda_k) e^{-j\lambda_k t} \\ w_2(t) &= \mathcal{H}_2[v_i(t)] \\ &= \sum_{\substack{k_1=-m, \\ k_1 \neq 0}}^m \sum_{\substack{k_2=-m, \\ k_2 \neq 0}}^m A_{k_1} A_{k_2} H_2(j\lambda_{k_1}, j\lambda_{k_2}) e^{+j\lambda_{k_1} t} e^{+j\lambda_{k_2} t} \end{aligned} \quad (5.75)$$

$$\begin{aligned} w_3(t) &= \mathcal{H}_3[v_i(t)] \\ &= \sum_{\substack{k_1=-m, \\ k_1 \neq 0}}^m \sum_{\substack{k_2=-m, \\ k_2 \neq 0}}^m \sum_{\substack{k_3=-m, \\ k_3 \neq 0}}^m A_{k_1} A_{k_2} A_{k_3} H_3(j\lambda_{k_1}, j\lambda_{k_2}, j\lambda_{k_3}) \\ &\quad \times e^{+j\lambda_{k_1} t} e^{+j\lambda_{k_2} t} e^{+j\lambda_{k_3} t} \end{aligned} \quad (5.76)$$

(5.77)

We can move much faster this time having studied the derivation methodologies for the single-tone case. In our approach this time, we derive the $v_{o1}(t), v_{o2}(t)$ and $v_{o3}(t)$ and then equate them with $u_1(t), u_2(t)$, and $u_3(t)$, respectively, to reach the shape and form of the multi-dimensional Volterra kernels.

5.5.1 Computing $v_{o1}(t)$

Taking advantage that the linearized circuit in Figure 5.11 is linear, and that the input excitation, $v_i(t)$ is multi-tone sinusoidal gives us the form of the response as follows

$$v_{o1}(t) = \frac{A_1}{2} H(-j\lambda_1) e^{-j\lambda_1 t} + \frac{A_1}{2} H(j\lambda_1) e^{+j\lambda_1 t} + \frac{A_2}{2} H(-j\lambda_2) e^{-j\lambda_2 t} + \frac{A_2}{2} H(j\lambda_2) e^{+j\lambda_2 t} \quad (5.78)$$

where $H(j\omega)$ is given by

$$H(j\omega) = \frac{1}{g_1 + j\omega C} \quad (5.79)$$

We can clarify matters much more by summarizing the steps that we took to find the Volterra kernels above and include discussion for more general cases.

1. **Step 1: Linearizing the circuit.** In this step the DC operating point of the circuit is first computed and every nonlinear conductance in the circuit is replaced by a resistor. The numerical Ohmic value for this resistance is the reciprocal of the first derivative of the current with respect to the terminal voltage of the nonlinear conductance. Such a derivative must be computed at the DC value of the terminal voltage.
2. **Step 2: Computing the response of the linearized circuit to the independent sources.** Here the response of the linear circuit obtained from the above linearization step is computed under the independent stimulus to the circuit $v_i(t)$. This response is called the first-order response and denoted by $u_1(t)$. Given that the linearized circuit is linear, then $u_1(t)$ will have spectral components at exactly the same frequency values of the input stimulus $v_i(t)$ but with different amplitudes.

Suppose for example that our independent stimulus had two tones λ_1 and λ_2 with four spectral lines at $\omega = \pm\lambda_1, \pm\lambda_2$, then $u_1(t)$ will have the same spectral lines, but only with different amplitudes. It is those amplitudes that do represent the in $u_1(t)$ that will give us the reading for the first-order Volterra kernel, at those spectral lines, i.e., $H_1(j\lambda_1)$, $H_1(-j\lambda_1)$, $H_1(j\lambda_2)$ and $H_1(-j\lambda_2)$.

Suppose that the circuit was simple enough to allow us to do the derivations by hand and arrive at the amplitude of $u_1(t)$ in the form of an analytical expression that includes λ_1 or λ_2 in its closed-form such as in the simple circuit used above. Then we would have found the first-order kernel $H_1()$ in its most ideal form as we did in the same circuit example considered above. However, the circuit may be too complex to let us find analytically an expression for the amplitudes of the spectral components of $u_1(t)$. In this case, we could only compute numerically the amplitudes of the spectral component in $u_1(t)$. In that case, we will still be able to “read off” the value that $H_1(j\omega)$ take when $\omega = \pm\lambda_1, \pm\lambda_2$ by simply using the value of those amplitudes in $u_1(t)$. We will see examples for this situation later on.

3. **Step 3: Computing the response of $u_2(t)$.** In this step, we deactivate the independent sources of the linearized circuit, where an independent voltage source is deactivated

by replacing it by a short-circuit and an independent current source is deactivated by replacing by an open-circuit. Next connect a current source between the nodes where the nonlinear conductance was connected to. the value of this current source should be $g_2 u_1(t)^2$ where g_2 is the second-order derivative of the nonlinear conductance computed at the DC operating point, i.e.

$$g_2 = \left. \frac{\partial^2 f}{\partial v^2} \right|_{v_{DC}} \quad (5.80)$$

Finally compute the response of the resulting circuit and call this response $u_2(t)$. Given that $u_1(t)$ has discrete spectral components, then $u_1(t)$ will have discrete components at mixing products of those components. To illustrate that, assume as before that $u_1(t)$ had two spectral components at $\pm\lambda_1$ and $\pm\lambda_2$ that initially resulted from a two-tone in $v_i(t)$. Then $u_1(t)^2$ will have spectral components at

$$\begin{aligned} & -\lambda_1 - \lambda_2 \\ & -\lambda_1 + \lambda_2 \\ & +\lambda_1 - \lambda_2 \\ & +\lambda_1 + \lambda_2 \end{aligned}$$

Given that the linearized circuit is linear, then the response $u_2(t)$ due to the input $u_1(t)^2$ will have four spectral lines at the exactly the same frequencies indicated above. Those frequencies are usually called mixing product, since they appear as mixture between the input frequencies. Now, it is the amplitudes of those frequencies in $u_2(t)$ that will give us the second-order Volterra kernel. For example, the amplitude of $u_2(t)$ at $-\lambda_1 + \lambda_2$ is the value of Second-order Volterra kernel $H_2(j\omega_1, j\omega_2)$, when $\omega_1 = -\lambda_1$ and $\omega_2 = \lambda_2$.

Again if the circuit was simple enough to allow us to do this calculation of $u_2(t)$ by hand and express the amplitudes of its different spectral components as analytical expressions in terms of symbolical variables λ_1, λ_2 then we will have reached the analytical expression for $H_2(j\lambda_1, j\lambda_2)$, which is an ideal situation. However, in most cases it is only possible to compute numerical values for the spectral components in $u_2(t)$, which would give us the values of the $H_2(j\omega_1, j\omega_2)$ when $\omega_1 = \pm\lambda_1$ and $\omega_2 = \pm\lambda_2$.

However, it is obvious the increasing amount of complexity if attempt to do that. For example, when we started with single-tone we had two exponential terms in $v_{o1}(t)$, 4 terms in $v_{o2}(t)$ ¹ and 8 terms in $v_{o3}(t)$. On the other hand, if we choose to start the analysis with the three-tone stimulus, then we will have in $v_{o1}(t)$ 6 terms, 36 terms in $v_{o2}(t)$ and $6^3 = 216$ terms in $v_{o3}(t)$, which adds up to a total of 258 terms.

¹In fact there are two terms at DC in (5.60) that were combined in one term.

5.5.2 Discussion on the Method of Nonlinear Currents

The general idea that emerges from the previous developments is that analysis based on the concepts of Volterra kernels is a possible option for general multi-tone steady-state analysis if the amplitude of each tone in the input stimulus to the circuit is sufficiently small. Indeed, the only assumption that we used to truncate the Volterra series after the third term is that magnitude of the input signal is sufficiently small.

However, the only troubles that seems to be looming if we adopt the Volterra kernel is the enormous complexity of deriving the Volterra kernels, even if we attempt to that only up to order 3. We have seen how this can become more complex for a very simple circuit consisting of only two nodes.

Nonetheless, generalizing the method of nonlinear current to find the first three Volterra kernels to more general and complex circuit is possible if we willing to sacrifice its elegant advantage of having the Volterra kernels in closed-form analytical expressions which, enables us to simply plug in numbers for the excitation frequencies and get back the amplitude of the spectral components of the output waveform, which is . The generalization to large circuits, and the sacrifice needed to attain it, is discussed in the next section.

5.6 Application of the Method of Nonlinear Currents to General Circuits

The application of the method of nonlinear currents to general nonlinear circuit would be practically possible if, instead of attempting to find the closed-form expressions, we only seek the numerical (complex) values of the Volterra kernels at specific frequency values.

Appendices

Appendix **A**

The Newton-Raphson Iterative Method

The main objective in the N-R method is to compute the vector \mathbf{x} that satisfies the following system of equations

$$\Psi(\mathbf{x}) = \mathbf{0} \quad (\text{A.1})$$

where \mathbf{x} is a vector with, say, L components and $\Psi(\mathbf{x})$ is a system of L nonlinear multi-variable functions. In other words, the N-R method seeks the solution vector

$$\mathbf{x} = [x_1 \quad \cdots \quad x_L]^T \quad (\text{A.2})$$

that satisfies the L nonlinear equations

$$\begin{aligned} \psi_1(x_1, x_2, \dots, x_L) &= 0 \\ \psi_2(x_1, x_2, \dots, x_L) &= 0 \\ &\vdots \\ \psi_L(x_1, x_2, \dots, x_L) &= 0 \end{aligned} \quad (\text{A.3})$$

The starting point in the N-R method is to come up with a good initial guess for the solution vector which is denoted by $\mathbf{x}^{(0)}$. Naturally, it is highly unlikely that this initial trial vector is the accurate solution and therefore is not expected to satisfy the nonlinear system in (A.3). Hence substituting $\mathbf{x} = \mathbf{x}^{(0)}$ in (A.1) should give us

$$\Psi(\mathbf{x}^{(0)}) = \Delta^{(0)} \quad (\text{A.4})$$

where $\Delta^{(0)}$ is an error term. The N-R method then proceeds to modify the initial trial vector $\mathbf{x}^{(0)}$ based on the error term in Eq. (A.4) and the Jacobian matrix of $\psi(\mathbf{x})$ w.r.t \mathbf{x}

computed at $\mathbf{x} = \mathbf{x}^{(0)}$. Assume that the correction term is denoted by δ_x , then

$$\delta_x = (\mathbf{J}|_{\mathbf{x}=\mathbf{x}^{(0)}})^{-1} \Delta^{(0)} \quad (\text{A.5})$$

The Jacobian matrix \mathbf{J} is the matrix of partial derivatives defined by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \psi_1}{\partial x_1} & \frac{\partial \psi_1}{\partial x_2} & \dots & \frac{\partial \psi_1}{\partial x_L} \\ \frac{\partial \psi_2}{\partial x_1} & \frac{\partial \psi_2}{\partial x_2} & \dots & \frac{\partial \psi_2}{\partial x_L} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial \psi_L}{\partial x_1} & \frac{\partial \psi_L}{\partial x_2} & \dots & \frac{\partial \psi_L}{\partial x_L} \end{bmatrix} \equiv \frac{\partial \Psi}{\partial \mathbf{x}} \quad (\text{A.6})$$

and the notation of $\mathbf{J}|_{\mathbf{x}=\mathbf{x}^{(0)}}$ means that all the partial derivatives $\frac{\partial \psi_i}{\partial x_j}$ are computed at $\mathbf{x} = \mathbf{x}^{(0)}$, while $(\cdot)^{-1}$ denotes the matrix inversion operation. We then modify the initial trial vector using δ_x to obtain the next trial vector

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \delta_x \quad (\text{A.7})$$

We substitute $\mathbf{x} = \mathbf{x}^{(1)}$ in (A.1),

$$\Psi(\mathbf{x}^{(1)}) = \Delta^{(1)} \quad (\text{A.8})$$

If the vector-norm of the error term is below an acceptable threshold, then we can take $\mathbf{x}^{(1)}$ as the final solution. However, if this error term is unacceptable, the above calculation of another modification term as in (A.5) is repeated iteratively. The theory behind the N-R method assures us that this iterative process will eventually converge provided that the initial guess is sufficiently close to the final solution. Figure A.1 presents a pseudo-code implementation of the N-R method.

Algorithm 1: Solving a system of nonlinear equations using the N-R method.

```

input : A system of nonlinear equations  $\Psi(\mathbf{x})$ 
output : A solution vector  $\mathbf{x}^{(\cdot)}$  satisfying  $\Psi(\mathbf{x}^{(\cdot)}) = \mathbf{0}$ 
begin
     $\epsilon \leftarrow$  some small positive umber representig an acceptable error threshold;
     $\mathbf{x}^{(0)} \leftarrow$  An initial guess trial vector;
    comment: Calculate the error due to this initial vector.
    ;
    ;
     $\Delta^{(0)} \leftarrow \Psi(\mathbf{x}^{(0)})$ ;
    ;
    if  $\|\Delta^{(0)}\| > \epsilon$  then
        comment: Error is higher than an acceptable threshold.
        ;
        Convergence = false
    else
        comment: Error is lower than an acceptable threshold.
        ;
        Convergence = true;
         $\mathbf{x}^{(\cdot)} \leftarrow \mathbf{x}^{(0)}$ 
     $j \leftarrow 0$ ;
    comment: Repeat until converging to a solution
    while Convergence = false do
        comment: Compute the Jacobian matrix and the nonlinear function vector at
         $\mathbf{x} = \mathbf{x}^{(j)}$ 
        ;
         $\mathbf{J} \leftarrow \frac{\partial \Psi}{\partial \mathbf{x}} @ \mathbf{x} = \mathbf{x}^{(j)}$ ;
         $\Delta^{(j)} \leftarrow \Psi(\mathbf{x}) @ \mathbf{x} = \mathbf{x}^{(j)}$ ;
        comment: Compute the correction step for  $\mathbf{x}^{(j)}$ 
        ;
         $\delta_x \leftarrow \mathbf{J}^{-1} \Delta^{(j)}$ ;
        comment: Obtain the next trial vector.
         $\mathbf{x}^{(j+1)} \leftarrow \mathbf{x}^{(j)} - \delta_x$ ;
        comment: Compute the error at the next trial vector.
         $\Delta^{(j)} \leftarrow \Psi(\mathbf{x}) @ \mathbf{x} = \mathbf{x}^{(j)}$ ;
        if  $\|\Delta^{(j)}\| < \epsilon$  then
            comment: Error is higher than an acceptable threshold.
            ;
            Convergence = false
        else
            comment: Error is lower than an acceptable threshold.
            ;
            Convergence = true;
             $\mathbf{x}^{(\cdot)} \leftarrow \mathbf{x}^{(j)}$ 
         $j \leftarrow j + 1$ ;
    return  $\mathbf{x}^{(\cdot)}$ 

```

Figure A.1: A pseudocode description of the N-R method.

Appendix **B**

The Kronecker Matrix Product

The Kronecker product matrix operator \otimes is defined as follows. Let \mathbf{A} and \mathbf{B} be matrixces of sizes $n \times m$ and $p \times q$, respectively. Then the Kronecker peroduct of $\mathbf{A} \otimes \mathbf{B}$ is a matrix of size $nm \times mq$ defined as follows.

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1m}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2m}\mathbf{B} \\ \vdots & & \ddots & \vdots \\ a_{n1}\mathbf{B} & a_{n2}\mathbf{B} & \cdots & a_{nm}\mathbf{B} \end{bmatrix} \quad (\text{B.1})$$

Bibliography

- [1] <http://www.fftw.org/>.
- [2] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes in C. New York: Cambridge University Press, 1988.
- [3] L. O. Chua and A. Ushida, “Algorithms for computing almost periodic steady-state response of nonlinear systems to multiple input frequencies,” vol. 28, pp. 953–971, 1981.
- [4] K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, Steady-State Methods for Simulating Analog and Microwave Circuits. Boston: Kluwer Academic, 1990.
- [5] K. S. Kundert, G. B. Sorkin, and A. Sangiovanni-Vincentelli, “Applying harmonic balance to almost-periodic circuits,” IEEE Trans. Microwave Theory Tech., vol. 36, pp. 366–378, 1988.
- [6] G. B. Sorkin, K. S. Kundert, and A. Sangiovanni-Vincentelli, “An almost-periodic Fourier transform for use with harmonic balance,” in IEEE MTT-S Intl. Microwave Symp. Dig., 1987, pp. 717–220.
- [7] P. J. C. Rodrigues, “An orthogonal almost-periodic Fourier transform for use in nonlinear circuit simulation,” IEEE Microwave and Guided Wave Letters, vol. 4, pp. 74–76, 1994.
- [8] ——, Computer-Aided Analysis of Nonlinear Microwave Circuits. Boston: Artech House, 1997.
- [9] D. Hente and R. H. a. Jansen, “Frequency domain continuation method for the analysis and stability investigation of nonlinear microwave circuits,” IEE Proceedings, vol. 133, no. 351-362, 1986.
- [10] J. Vlach and K. Singhal, Computer Methods for Circuit Analysis and Design. New York: Van Nostrand Reinhold, 1983.