CHAPTER TWO

Transmission network modelling

This chapter sets out the simulation model of the backbone of the power system, the transmission network. An important part of this simulation program is the power flow (load flow) calculation to obtain the operating parameters of a transmission network such as magnitude and angle of bus voltages and active and reactive power flows across all branches. The chapter starts with the formulation of the admittance matrix, which relates bus voltages to bus current injections. The matrix is required for the computation of the power flow solution. Mathematical formulation of the power flow and an example solution using a three-bus system are illustrated next. The programming aspects of the power flow in computers and the code to obtain power flow solutions are elaborated at the end.

Once the power flow solution is obtained, the initial steady state operating points for other network components, such as synchronous machines, for the dynamic simulation can be easily calculated. The steady state defines the initial operating point at which the time derivative of all system states is zero. Time domain simulation usually starts from a steady-state operating condition. In this book, we will follow the steps presented in Fig. 2.1.

This chapter serves as guidance for those less familiar on the use of Matlab and Simulink, as it provides more details than later chapters. By the end of this chapter, the reader will be able to formulate an admittance matrix, find power flow solutions for small networks and be able to run a power flow simulation program and explain the results. Furthermore, the reader will know how to create and run a Simulink time domain simulation and validate the result. The network representation implemented in this chapter is the cornerstone to all the simulations in the later chapters.

2.1 Admittance matrix

Let us consider a simple network in Fig. 2.2, in which two transmission lines connect three buses. The lines are represented using the nominal π section model capturing the electrical properties of an overhead line such as resistance, inductance and capacitance. The shunt elements at Bus 2, shunt susceptance and shunt conductance, are represented using

susceptance is the imaginary part of admittance. the real part is conductance. Hence susceptance + conductance = admittance

Simulation of Power System with Renewables
ISBN: 978-0-12-811187-1
https://doi.org/10.1016/B978-0-12-811187-1.00002-0

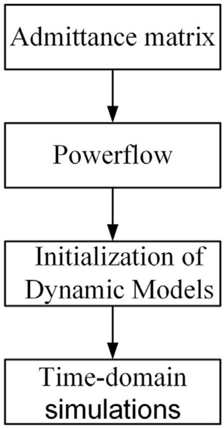

Figure 2.1 Power system simulation procedure.

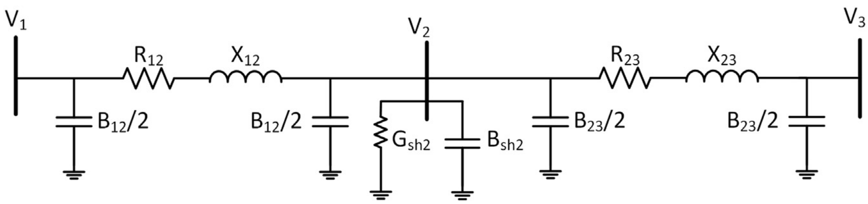

Figure 2.2 Representation of line and bus parameters.

B_{sh2} and G_{sh2}, respectively. The relationship between current injection at Bus 2 and the bus voltages is given by

$$I_2 = V_2\left(G_{sh2} + j\left(B_{sh2} + \frac{B_{12}}{2} + \frac{B_{23}}{2}\right)\right) + \frac{(V_2 - V_1)}{(R_{12} + jX_{12})} + \frac{(V_2 - V_3)}{(R_{23} + jX_{23})} \tag{2.1}$$

Self-admittance Y_{22} at Bus 2 is defined as

$$Y_{22} = \left(G_{sh2} + j\left(B_{sh2} + \frac{B_{12}}{2} + \frac{B_{23}}{2}\right)\right) + \frac{1}{(R_{12} + jX_{12})} + \frac{1}{(R_{23} + jX_{23})} \tag{2.2}$$

In addition, the mutual admittances Y_{21} and Y_{23} between buses 1 and 2 and buses 1 and 3, respectively, are defined as

$$Y_{21} = \frac{-1}{(R_{12} + jX_{12})} \tag{2.3}$$

$$Y_{23} = \frac{-1}{(R_{23} + jX_{23})} \tag{2.4}$$

Substituting (2.2–2.4) in (2.1), the equation for current injection at the Bus 2 becomes

$$I_2 = V_2 Y_{22} + V_1 Y_{21} + V_3 Y_{23} \tag{2.5}$$

Eq. (2.5) can be generalized for an n bus system as

$$\begin{bmatrix} \widetilde{I}_1 \\ \widetilde{I}_2 \\ \vdots \\ \widetilde{I}_n \end{bmatrix} = \begin{bmatrix} \widetilde{Y}_{11} & \widetilde{Y}_{12} & \cdots & \widetilde{Y}_{1n} \\ \widetilde{Y}_{21} & \widetilde{Y}_{22} & \cdots & \widetilde{Y}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{Y}_{n1} & \widetilde{Y}_{n2} & \cdots & \widetilde{Y}_{nn} \end{bmatrix} \begin{bmatrix} \widetilde{V}_1 \\ \widetilde{V}_2 \\ \vdots \\ \widetilde{V}_2 \end{bmatrix} \tag{2.6}$$

$$\widetilde{\boldsymbol{I}} = \widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{V}} \tag{2.7}$$

where $\widetilde{\boldsymbol{Y}}$ is the admittance matrix, $\widetilde{\boldsymbol{V}}$ is bus voltage vector and $\widetilde{\boldsymbol{I}}$ is a current vector representing current injection at all buses. All the three quantities are complex values.

The $\widetilde{\boldsymbol{Y}}$ matrix can be formed by inspection from the line and bus parameters. The diagonal element $\widetilde{Y}_{kk}$ is the sum of admittances of all the elements connected at Bus k.

$$\widetilde{Y}_{kk} = \sum_{l=1}^{n} \widetilde{y}_{kl} + \sum_{l=1}^{n} \widetilde{y}_{Bkl} + \widetilde{y}_k \tag{2.8}$$

In the admittance matrix, the susceptance is added to the diagonal term of the bus it is connected to, leading to the term $\widetilde{y}_{Bkl}$. Buses may further have shunt elements installed, which are accounted for by the $\widetilde{y}_k$ term.

$$\widetilde{y}_{Bkl} = 0.5\, jB_{kl}, \widetilde{y}_k = G_{shk} + jB_{shk} \tag{2.9}$$

The off-diagonal element $\widetilde{Y}_{kl}$ is the negative value of the sum of admittances of all lines connecting Bus k and Bus l. The negative sign accounts for the negative sign in the receiving end voltage for every branch current calculation as seen in Eq. (2.1–2.5).

$$\widetilde{Y}_{kl} = -\widetilde{y}_{kl}, \widetilde{y}_{kl} = \frac{1}{R_{kl} + jX_{kl}} \tag{2.10}$$

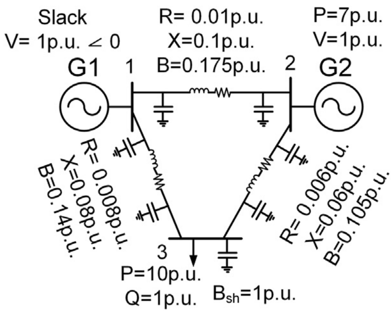

Figure 2.3 Two-machine, three-bus system.

2.2 Example

Let us understand the procedure of calculating an admittance matrix by inspection, using the three-bus system example shown in Fig. 2.3. Using the equations described above, find the admittance matrix for this system, checking your results as you go along with the provided solution.

The individual admittance values are

$$\tilde{y}_{12} = \tilde{y}_{21} = 0.9901 - j9.9010, \tilde{y}_{13} = \tilde{y}_{31} = 1.2376 - j12.3762,$$

$$\tilde{y}_{23} = \tilde{y}_{32} = 1.6502 - j16.5017$$

$$\tilde{y}_{B12} = \tilde{y}_{B21} = j0.0875,\ \tilde{y}_{B13} = \tilde{y}_{B31} = j0.07, \tilde{y}_{B23} = \tilde{y}_{B32} = j0.0525,$$
$$\tilde{y}_3 = j1$$

The admittance matrix, containing the appropriate diagonal and off-diagonal terms, as shown below, leads to the following numerical end result:

$$Y = \begin{bmatrix} y_{12} + y_{13} + y_{B12} + y_{B13} & -y_{12} & -y_{13} \\ -y_{21} & y_{21} + y_{23} + y_{B21} + y_{B23} & -y_{23} \\ -y_{31} & -y_{32} & y_{31} + y_{32} + y_{B31} + y_{B32} + y_3 \end{bmatrix}$$
$$= \begin{bmatrix} 2.2277 - j22.1197 & -0.9901 + j9.901 & -1.2376 + j12.3762 \\ -0.9901 + j9.901 & 2.6403 - j26.2626 & -1.6502 + j16.5017 \\ -1.2376 + j12.3762 & -1.6502 + j16.5017 & 2.8878 - j27.7554 \end{bmatrix} \tag{2.11}$$

2.3 Power flow computation

The objective of the power flow program is to find the magnitude and angle of bus voltages given the active and reactive power load and active power generation at all buses. As losses are not known a priori, active power generation at one (sometimes more than one) of the buses, called slack bus or swing bus, is not specified. Instead, the magnitude and angle of voltage at the slack bus is specified usually (not necessarily) as 1 pu and zero degrees, respectively. There are four variables in each bus, voltage magnitude, voltage angle, active power and reactive power, and two of them are always specified. Accordingly, the buses are classified into the following three types:

Slack bus: This bus is the reference point for the rest of the AC network and is frequently chosen to be one of the generator buses. The bus voltage magnitude at this bus is commonly set to 1pu while the bus angle is set to 0 degree. As voltage magnitude and angle are provided as knowns, both the real power and reactive power at the bus are variables.

PV bus: All generator buses apart from the slack bus are commonly modelled as PV buses; however this is not a fixed rule. In the PV bus, the real power and voltage magnitude are known, while the reactive power and voltage angle are unknown.

PQ bus: A bus where active power and reactive power are fixed is called PQ bus. The magnitude and angle of voltage are unknowns. Buses that neither contain generation nor load naturally have zero entries for P and Q.

This section focuses on essential steps to obtain a power flow solution without going into details of the method. The three-bus, two-generator system shown in Fig. 2.3 is used as an example. With the admittance matrix at hand, which was calculated previously, we can find the load flow solution. Let us first go through a small derivation to understand the power flow equations.

The power injection at a bus is given by

$$P_k + jQ_k = \widetilde{V}_k \widetilde{I}_k^* \tag{2.12}$$

Multiplying out the matrix equation $\widetilde{\boldsymbol{I}} = \widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{V}}$ and taking the conjugate on both sides leads to

$$\widetilde{I}_k^* = \sum_{l=1}^{n} \widetilde{Y}_{kl}^* \widetilde{V}_l^* \tag{2.13}$$

Combining (2.12) and (2.13), where $\widetilde{Y} = G + jB$, gives

$$P_k + jQ_k = \widetilde{V}_k \sum_{l=1}^{n} (G_{kl} - jB_{kl}) \widetilde{V}_l^* \tag{2.14}$$

Converting the phasors $\widetilde{V}_k$ and $\widetilde{V}_l$ to polar form, we get

$$\widetilde{V}_k \widetilde{V}_l^* = V_k V_l (\cos \theta_{kl} + j \sin \theta_{kl}) \tag{2.15}$$

where $\theta_{kl} = \theta_k - \theta_l$.

Eq. (2.14) becomes

$$P_k + jQ_k = V_k \sum_{l=1}^{n} V_l (\cos \theta_{kl} + j \sin\theta_{kl})(G_{kl} - jB_{kl}) \tag{2.16}$$

The real and imaginary parts of (2.16) are

$$P_k = f_p(V, \theta) = V_k \sum_{l=1}^{n} V_l (G_{kl} \cos \theta_{kl} + B_{kl} \sin \theta_{kl}) \tag{2.17}$$

$$Q_k = f_q(V, \theta) = V_k \sum_{l=1}^{n} V_l (G_{kl} \sin \theta_{kl} - B_{kl} \cos \theta_{kl}) \tag{2.18}$$

Eqs. (2.17) and (2.18) can be written in the form given below:

$$\begin{bmatrix} P_k \\ Q_k \end{bmatrix} = \begin{bmatrix} f_p(V_1, \ V_2 \cdots V_n, \ \theta_1, \ \theta_2 \cdots \theta_n) \\ f_q(V_1, \ V_2 \cdots V_n, \ \theta_1, \ \theta_2, \cdots \theta_n) \end{bmatrix} \tag{2.19}$$

Let V and θ be the estimate of magnitude and angle of voltage at the beginning of an iteration, respectively, and ΔV and $\Delta\theta$ be the correction required to improve the solution. Taylor's theorem can be used to obtain the following result. Here, the details of derivation are skipped as the focus of the chapter is on implementing a power flow solution. Readers are advised to refer (Arrillaga, 2001) for more details.

$$\begin{bmatrix} P_k \\ Q_k \end{bmatrix} = \begin{bmatrix} f_p(V, \theta) \\ f_q(V, \theta) \end{bmatrix} = \underbrace{\begin{bmatrix} \dfrac{\partial f_p}{\partial \theta} & \dfrac{\partial f_p}{\partial V} \\ \dfrac{\partial f_q}{\partial \theta} & \dfrac{\partial f_q}{\partial V} \end{bmatrix}}_{j} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} \tag{2.20}$$

where J is called the Jacobian matrix. The Newton Raphson method using polar coordinates uses the inverse Jacobian matrix to update angles and voltage elements. Fig. 2.4 shows a flow chart for finding a power flow solution. The major step in the power flow calculation is the formulation of Jacobian matrix. Let us understand the formulation of this matrix. Once it is ready, we will return to the simple three-bus, two-generator system to

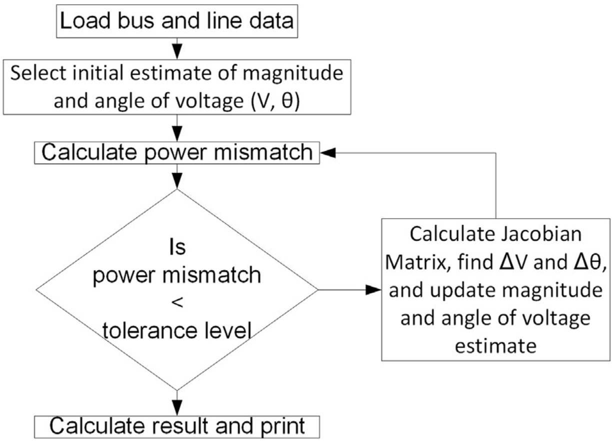

Figure 2.4 Program flow diagram of power flow solver.

practise the formulation of a Jacobian matrix and the procedure for the power flow solution.

2.4 Formulation of jacobian

The off-diagonal elements of J obtained using (2.17 and 2.18) are

$$\frac{\partial P_k}{\partial \theta_l} = V_k V_l (G_{kl} \sin \theta_{kl} - B_{kl} \cos \theta_{kl}) \tag{2.21}$$

$$V_l \frac{\partial P_k}{\partial V_l} = V_k V_l (G_{kl} \cos \theta_{kl} + B_{kl} \sin \theta_{kl}) \tag{2.22}$$

$$\frac{\partial Q_k}{\partial \theta_l} = - V_k V_l (G_{kl} \cos \theta_{kl} + B_{kl} \sin \theta_{kl}) \tag{2.23}$$

$$V_l \frac{\partial Q_k}{\partial V_l} = V_k V_l (G_{kl} \sin \theta_{kl} - B_{kl} \cos \theta_{kl}) \tag{2.24}$$

The diagonal elements of J are

$$\frac{\partial P_k}{\partial \theta_k} = - B_{kk} V_k^2 - Q_k \tag{2.25}$$

$$V_k \frac{\partial P_k}{\partial V_k} = P_k + G_{kk} V_K^2 \tag{2.26}$$

$$\frac{\partial Q_k}{\partial \theta_k} = P_k - G_{kk} V_k^2 \tag{2.27}$$

$$V_k \frac{\partial Q_k}{\partial V_k} = -B_{kk} V_k^2 + Q_k \tag{2.28}$$

An additional element, V_k or V_l, present in the left-hand side (LHS) of Eqs. (2.22), (2.24), (2.26) and (2.28) makes the right-hand side (RHS) of the equations similar, which makes calculation of J easier. The J matrix has to be calculated at every iteration, which demands an easy approach. A simple way of finding the elements of J matrix is to compute K using (2.29) for given estimate of magnitude and angle of voltage.

$$K = diag(V) Y^* diag(V^*) \tag{2.29}$$

Expanding (2.29), the off-diagonal and diagonal elements of K are obtained as (2.30) and (2.31), respectively.

$$K_{kl} = V_k V_l [(G_{kl} \cos \theta_{kl} + B_{kl} \sin \theta_{kl}) + j(G_{kl} \sin \theta_{kl} - B_{kl} \cos \theta_{kl})] \tag{2.30}$$

$$K_{kk} = V_k^2 (G_{kl} - jB_{kl}) \tag{2.31}$$

The real and imaginary parts of the off-diagonal elements of $\boldsymbol{K}$ are directly related to off-diagonal elements of J. A similar relation is visible with the diagonal elements of K and J.

For example, $\frac{\partial P_k}{\partial \theta_l} = imag(K_{kl})$ and $\frac{\partial P_k}{\partial \theta_k} = imag(K_{kk}) - Q_k$.

In this chapter, we will calculate the matrix K and pick elements of J through comparison.

2.5 Example of three-bus system

Let us apply the method in detail using the three-bus system example, which contains one PV bus, one PQ bus and one slack bus. By definition, PV buses have fixed active power generation and voltage magnitude. At these buses, terms corresponding to ΔV and ΔQ are absent. The Jacobian has two rows for each PQ bus and one row for each PV bus. No term appears for the slack bus, as both the voltage magnitude and angle are known.

As shown in Fig. 2.3, Bus 1 is the slack bus, where both the angle and voltage magnitude are known. This bus does not appear in the Jacobian. Bus 2 is a generator or PV bus, where the voltage angle is unkown. Only one row corresponding to ΔP appears for this bus. Bus 3 is a PQ bus, where neither the voltage nor angle is known, and two rows represent this bus in the Jacobian matrix.

The Jacobian for the study system is defined as

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_3 \end{bmatrix} = \begin{bmatrix} \partial P_2/\partial\theta_2 & \partial P_2/\partial\theta_3 & \partial P_2/\partial V_3 \\ \partial P_3/\partial\theta_2 & \partial P_3/\partial\theta_3 & \partial P_3/\partial V_3 \\ \partial Q_3/\partial\theta_2 & \partial Q_3/\partial\theta_3 & \partial Q_3/\partial V_3 \end{bmatrix} \begin{bmatrix} \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta V_3 \end{bmatrix} \tag{2.32}$$

The LHS is the real and reactive power mismatch vector for a given assumed voltage magnitude and angle. The vector at the RHS represents the required correction in the assumed voltage magnitude and angle.

Iteration 1:

Let the initial estimates for the three-bus system example be $V^0 = [1.0\ 1.0\ 1.0]$, $\theta^0 = [0.0\ 0.0\ 0.0]$, $P_2 = 7$, $P_3 = 10$ and $Q_3 = 1$, and the admittance Y has been calculated in (2.11). The essential calculations in the iteration are

$$\widetilde{I^0} = Y\widetilde{V^0} = \begin{bmatrix} 0.0 & - & 0.1575i \\ 0.0 & - & 0.1400i \\ 0.0 & - & 1.1225i \end{bmatrix}$$

$$P^0 + jQ^0 = \widetilde{V^0}\widetilde{I^{0*}} = \begin{bmatrix} 0.0 & - & 0.1575i \\ 0.0 & - & 0.1400i \\ 0.0 & - & 1.1225i \end{bmatrix}$$

Using (2.29),

$$K = \begin{bmatrix} 2.2277 + 22.1197i & -0.9901 - 9.9010i & -1.2376 - 12.3762i \\ -0.9901 - 9.9010i & 2.6403 + 26.2626i & -1.6502 - 16.5017i \\ -1.2376 - 12.3762i & -1.6502 - 16.5017i & 2.8878 + 27.7554i \end{bmatrix}$$

Comparing (2.21)−(2.28) with (2.30)−(2.31),

$$J = \begin{bmatrix} 26.4026 & -16.5017 & -1.6502 \\ -16.5017 & 28.8779 & 2.8878 \\ 1.6502 & -2.8878 & 26.6329 \end{bmatrix}$$

The readers are advised to compare the K and J matrix with Eqs. (2.21)−(2.31) to understand the formulation. At this point, $\Delta P_2 = P_2^0 - P_2 = -7$, $\Delta P_3 = P_3^0 - P_3 = -10$ and $\Delta Q_3 = Q_3^0 - Q_3 = 0.1225$. The corrections required in the initial estimate using (2.32) are $\Delta\theta_2 = 0.0758$, $\Delta\theta_3 = -0.2997$ and $\Delta V_3 = -0.0326$.

Iteration 2:

Improved estimates are $V^1 = [1.0\ 1.0\ 0.9674]$ and $\theta^1 = [0.0\ 0.0758\ -0.2997]$. These yield the following:

$$\widetilde{I^1} = Y\widetilde{V^1} = \begin{bmatrix} 2.8825 - 0.5295i \\ 6.8185 - 0.3632i \\ -9.3910 + 2.2273i \end{bmatrix}$$

$$P^1 + jQ^1 = \begin{bmatrix} 2.8825 + 0.5295i \\ 6.7715 + 0.8781i \\ -9.3161 + 0.6239i \end{bmatrix}$$

$$J = \begin{bmatrix} 25.3845 & -15.4370 & 4.5164 \\ -14.2661 & 25.3516 & -6.8363 \\ 7.3395 & -12.0187 & 27.4957 \end{bmatrix}$$

At end of this iteration, $\Delta P_2 = 0.2285$, $\Delta P_3 = -0.6839$ and $\Delta Q_3 = -1.6239$. The corrections required for the estimate are $\Delta\theta_2 = -0.0096$, $\Delta\theta_3 = -0.0540$ and $\Delta V_3 = -0.0801$.

Iteration 3:

New estimates are $V^2 = [1.0\ 1.0\ 0.8873]$ and $\theta^2 = [0.0\ 0.0662\ -0.3537]$.

$$\widetilde{I^2} = Y\widetilde{V^2} = \begin{bmatrix} 3.3584 - 1.6237i \\ 7.0794 - 1.8866i \\ -10.1021 + 4.7419i \end{bmatrix}$$

$$P^2 + jQ^2 = \begin{bmatrix} 3.3584 + 1.6237i \\ 6.9391 + 2.3507i \\ -9.8662 - 0.8423i \end{bmatrix}$$

$$J = \begin{bmatrix} 23.9119 & -13.9671 & 5.2202 \\ -12.7733 & 22.6948 & -8.5568 \\ 7.3060 & -12.1398 & 23.6784 \end{bmatrix}$$

At end of this iteration, $\Delta P_2 = 0.0609$, $\Delta P_3 = -0.1338$ and $\Delta Q_3 = -0.1577$. The corrections required in the estimate are $\Delta\theta_2 = -0.0014$, $\Delta\theta_3 = -0.0112$ and $\Delta V_3 = -0.0120$.

Iteration 4:
New estimates are $V^3 = [1.01.00.8753]$ and $\theta^3 = [0.0\ 0.0648\ -0.3649]$.

$$\widetilde{I^{*3}} = Y\widetilde{V^3} = \begin{bmatrix} 3.4521 - 1.7968i \\ 7.1507 - 2.1263i \\ -10.2612 + 5.1382i \end{bmatrix}$$

$$P^3 + jQ^3 = \begin{bmatrix} 3.4521 + 1.7968i \\ 6.9979 + 2.5850i \\ -9.9958 - 0.9965i \end{bmatrix}$$

$$J = \begin{bmatrix} 23.6776 & -13.7333 & 5.3743 \\ -12.5298 & 22.2635 & -8.8914 \\ 7.3307 & -12.2085 & 23.1571 \end{bmatrix}$$

At end of this iteration, $\Delta P_2 = 0.0021$, $\Delta P_3 = -0.0042$ and $\Delta Q_3 = -0.0035$. The corrections required in the estimate are $\Delta\theta_2 = -0.00003$, $\Delta\theta_3 = -0.0003$ and $\Delta V_3 = -0.0003$.

At the end of this iteration, the correction required in the estimate is very small. A couple of more iterations can produce even better results. Readers are advised to try further iterations to obtain a better estimate and also to gain an understanding that a large number of iterations are not often necessary. The reader may furthermore observe that a higher convergence rate is achieved during the first few iterations.

2.6 Power flow implementation

In the previous sections, we have learnt to calculate the power flow solution using simple steps for a three-bus system. However, for large networks, a power flow program is essential. In this section, two Matlab functions for calculating the admittance matrix and load flow are introduced. Both functions follow simple steps explained in the previous section and are not designed to guarantee the highest computational efficiency. They are tested for small networks and sufficient for building dynamic simulations discussed in this book.

There are many commercial and free power flow programs available. Any valid power flow solution will be sufficient to proceed following this book, and there is no need to change from a power flow solver the user

Table 2.1 Format of Bus matrix.

Bus No.	Voltage V	Angle δ	Generation P_G	Q_G	Load P_L	Q_L	G_{SHUNT}	B_{SHUNT}	Bus_id
1	1.02		5						2
2					5	1			3
3	1	0							1

Table 2.2 Format of Line matrix.

From bus	To bus	Resistance pu	Reactance pu	Line charging pu	Tap ratio
01	02	0.001	0.01	0.02	1

is already familiar with. If the user chooses to use another power flow program, extra care needs to be taken to name all outputs of the program according to the rest of the program and Simulink simulation.

As with any program, the power flow program used in this book expects input data to be presented in a specific format using bus matrix and line matrix. The network buses can be divided into generator (PV), load (PQ) and slack buses as previously discussed. They are identified using a parameter, bus_id = {1, 2, 3}. Load buses have the bus_id = 3 and generator buses have the bus_id = 2, while the slack bus has a bus_id = 1. Format of an example Bus matrix is given in Table 2.1. Compulsory elements are filled and others are optional or zero.

Similarly, the transmission lines connecting the bus bars also need to be defined in the program. The line data are presented using Line matrix as shown in Table 2.2. The lines are characterized by resistance, reactance, line charging and tap ratio.

The code for the matlab functions *form_Ymatrix*, which returns admittance matrix, and *power_flow*, which returns power flow solutions, are given below. The readers are advised to copy both the codes and save it as *form_Ymatrix.m* and *power_flow.m*, respectively, in a new folder and rename the folder Simulation. The program is tested using a four-machine test system example in the next section. The % symbol is used to provide nonexecuted comments in the Matlab code in Script 2.1 and 2.2.

2.7 Study case: four-machine system

The four-machine system in Fig. 2.5 is commonly studied in the power system community (Kundur, 1994). The system contains 4 generators G1–G4, 11 buses, 4 transformers, 8 transmission lines, 2 loads (indicated by arrows) and 2 shunt capacitances. Tables 2.3 and 2.4 list bus and line data of the system in their respective units.

```
function [Y] = form_Ymatrix(bs,ln)
% Form Y matrix from bus and line data

Y = zeros(size(bs,1),size(bs,1));  % Initialising Y bus to zero
sy = size(Y); % size of Y bus matrix

% This block of code adds line series admittance and shunt admittance to the diagonal
% elements of Y. First obtain index of diagonal elements corresponding to ln(:,1)
t = sub2ind(sy, ln(:,1), ln(:,1));
% Matrix Kt ensures correct admittance is added when more than one line connects two
% buses
[t1, t2, t3] = unique(t); Kt = zeros(size(t2,1),size(t3,1));
Kt(((1:size(Kt,2))-1)*size(Kt,1)+t3')=1;
% assigning values to diagonal elements
Y(t1) = Y(t1) + Kt*(1./(ln(:,3)+1i*ln(:,4))+1i*0.5*ln(:,5));

% This block does same as the previous block for the buses in ln(:,2)
t = sub2ind(sy, ln(:,2), ln(:,2));
[t1, t2, t3] = unique(t); Kt = zeros(size(t2,1),size(t3,1));
Kt(((1:size(Kt,2))-1)*size(Kt,1)+t3')=1;
Y(t1) = Y(t1) + Kt*(1./(ln(:,3)+1i*ln(:,4))+1i*0.5*ln(:,5));

% obtaining index of off-diagonal elements ln(:,1) to ln(:,2)
t = sub2ind(sy, ln(:,1), ln(:,2));
[t1, t2, t3] = unique(t); Kt = zeros(size(t2,1),size(t3,1));
Kt(((1:size(Kt,2))-1)*size(Kt,1)+t3')=1;
Y(t1) = Y(t1) - Kt*(1./(ln(:,3)+1i*ln(:,4)));

% obtaining index of off-diagonal elements ln(:,2) to ln(:,1)
t = sub2ind(sy, ln(:,2), ln(:,1));
[t1, t2, t3] = unique(t); Kt = zeros(size(t2,1),size(t3,1));
Kt(((1:size(Kt,2))-1)*size(Kt,1)+t3')=1;
Y(t1) = Y(t1) - Kt*(1./(ln(:,3)+1i*ln(:,4)));

% Adding shunt admittances at buses in the diagonal elements
Y = Y + diag(bs(:,8) + 1i*bs(:,9));

end
```

Script 2.1 Program to calculate Ymatrix.

2.8 Exercise

In Chapter 1, the benefits of handling power system calculations using a per unit system have been introduced. The readers are asked to convert the system parameters to the per unit system using the formulas discussed in Chapter 1 and compare it with the bus and line matrix provided in Script 2.3.

- Can you explain why the actual line data for the first four rows contain two entries for the inductance value, while the pu line data have only one entry?
- What is the source of the shunt capacitance in the line matrix?
- The capacitance connected at buses seven and nine are specified in MVar, not Farad. Why is this?

Now save script in Script 2.3 as *four_mac_data.m* in the *Simulation* folder. Run the m-file *four_mac_data*. The workspace on the Matlab main window should now show the bus and line matrices. Everything is in place to run the power flow for the four-machine test system shown in Fig. 2.5. Create an m-file named *four_mac_run.m* with the program in Script 2.4.

Run *four_mac_run.m*. The result, *bus_sol and line_flow* will be displayed in the command window. The slack bus for the simulation was set at Bus 3, where the real power and reactive power are the fourth and fifth element

```
function [bus_sln, flow] = power_flow(Y, bs, ln)

% The program solves load flow equations for a power system

bs(:,3) = bs(:,3)*pi/180; % converting angle from degrees to radians
V0 = bs(:,2); A0 = bs(:,3); nbs = size(bs,1);
bsl = find(bs(:,10)==1);   bpv = find(bs(:,10)==2);   ...
    bpq = find(bs(:,10)==3);

% Active and reactive power specified at PV and PQ buses

PQ = [bs(bpv,4)-bs(bpv,6);bs(bpq,4)-bs(bpq,6); bs(bpq,5)-bs(bpq,7)];

% Initial estimate of voltage angle (PV and PQ bus) and magnitude (PQ
bus)
vt0 = [bs(sort([bpv;bpq]),3);bs(bpq,2)];

itrn = 1;
while(true)
    % Updating voltage magnitude and angle
    T = [zeros(size(bpq,1),size(bpq,1)+size(bpv,1))
eye(size(bpq,1))];
    V0(bpq) = T*vt0;
    T = [eye(size(bpv,1)+size(bpq,1)) ...
        zeros(size(bpq,1)+size(bpv,1),size(bpq,1))];

    A0(sort([bpv; bpq])) = T*vt0;

% Calculate voltage, current and power based on the estimate

    v0 = V0.*exp(1i*A0);    i0 = Y*v0;   pq0 = v0.*conj(i0);

% Find difference in active and reactive power
    dpq = PQ-[real(pq0(sort([bpv;bpq])));imag(pq0(sort([bpq])))];

    K = diag(v0)*conj(Y)*diag(conj(v0));  % Calculating K matrix

    % Building Jacobian Matrix from K matrix
    Jp = [imag(K)-diag(imag(pq0)) ...
        (real(K)+diag(real(pq0)))./(ones(nbs,1)*V0')];

    Jq = [diag(real(pq0))-real(K) ...
        (imag(K)+diag(imag(pq0)))./(ones(nbs,1)*V0')];

    J = [Jp;Jq];
    J(sort([bsl; nbs+bsl;nbs+bpv]),:)=[];
    J(:,sort([bsl; nbs+bsl;nbs+bpv]))=[];

    dvt = J\dpq;    % Finding change in voltage magnitude and angle

    vt0 = vt0 + dvt;    % Updating voltage magnitude and angle

    itrn = itrn + 1;
    if max(abs(dvt))<1e-16 || itrn >50
        break
    end

end
```

Script 2.2 Program to calculate power flow.

```
% Building solved bus matrix
bus_sln = bs;
bus_sln(:,2:3) = [abs(v0) angle(v0)*180/pi];
bus_sln(sort([bpv;bsl]),4:5) = [real(pq0(sort([bpv;bsl]))) ...
    imag(pq0(sort([bpv;bsl])))];
bus_sln(sort([bpq]),6:7) = -[real(pq0(sort([bpq]))) ...
    imag(pq0(sort([bpq])))];

% calculating line flow
sln = size(ln,1);   flow = zeros(2*sln,4);
t = sub2ind(size(Y), ln(:,1), ln(:,2));
yft = Y(t);
flw = -v0(ln(:,1)).*conj(((v0(ln(:,1))-v0(ln(:,2))).*yft));
flow(1:sln, 1:4) = [ln(:,1) ln(:,2) real(flw) imag(flw)];

flw = -v0(ln(:,2)).*conj(((v0(ln(:,2))-v0(ln(:,1))).*yft));
flow(sln+1:2*sln, 1:4) = [ln(:,2) ln(:,1) real(flw) imag(flw)];
end
```

Script 2.2 Cont'd.

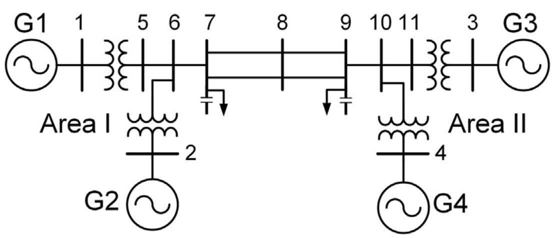

Figure 2.5 A four-machine, two-area test system (Kundur, 1994).

Table 2.3 Bus data of the four-machine, two-area system in actual values.

#	Bus V (kV)	Delta (degree)	Pgen (MW)	Qgen (MVar)	Pload (MW)	Qload (MVar)	Pshunt (MW)	Qshunt (MVar)	Bus type
1	20	0	700	0.0	0.00	0.0	0.0	0.0	PV
2	20	0	700	0.0	0.00	0.0	0.0	0.0	PV
3	20	0	0.0	0.0	0.00	0.0	0.0	0.0	Slack
4	20	0	700	0.0	0.00	0.0	0.0	0.0	PV
5	230	0	0.0	0.0	0.00	0.0	0.0	0.0	PQ
6	230	0	0.0	0.0	0.00	0.0	0.0	0.0	PQ
7	230	0	0.0	0.0	967	100	0.0	200	PQ
8	230	0	0.0	0.0	0.00	0.0	0.0	0.0	PQ
9	230	0	0.0	0.0	1767	100	0.0	350	PQ
10	230	0	0.0	0.0	0.00	0.0	0.0	0.0	PQ
11	230	0	0.0	0.0	0.00	0.0	0.0	0.0	PQ

Table 2.4 Line data of the four-machine, two-area system in actual values.

From bus	To bus	R (Ω)	L (mH)	C (μF)	Tap ratio	Tap phase (degree)
01	05	0.0000	23.4 HV (0.17719 LV)	0.000000	1.0	0.0
02	06	0.0000	23.4 HV (0.17719 LV)	0.000000	1.0	0.0
03	11	0.0000	23.4 HV (0.17719 LV)	0.000000	1.0	0.0
04	10	0.0000	23.4 HV (0.17719 LV)	0.000000	1.0	0.0
05	06	1.3225	35.1	0.219380	1.0	0.0
10	11	1.3225	35.1	0.219380	1.0	0.0
06	07	0.5290	14.0	0.087751	1.0	0.0
09	10	0.5290	14.0	0.087751	1.0	0.0
07	08	5.8190	154.4	0.965260	1.0	0.0
07	08	5.8190	154.4	0.965260	1.0	0.0
08	09	5.8190	154.4	0.965260	1.0	0.0
08	09	5.8190	154.4	0.965260	1.0	0.0

```
%four_mac_data.m % bus No., Voltage Mag., Voltage Angle, Pgen,
Qgen, Pload, Qload, Pshunt, % Qshunt, bus type
bus = [...
1 1.00    00.0   7.000    0.00 0.0000   0.000    0.00   0.00 2;
2 1.00    00.0   7.000    0.00 0.0000   0.000    0.00   0.00 2;
3 1.00    00.0   0.0000   0.00 0.0000   0.000    0.00   0.00 1;
4 1.00    00.0   7.0000   0.00 0.0000   0.000    0.00   0.00 2;
5 1.00    00.0   0.0000   0.00 0.0000   0.000    0.00   0.00 3;
6 1.00    00.0   0.0000   0.00 0.0000   0.000    0.00   0.00 3;
7 1.00    00.0   0.0000   0.00 9.6700   1.000    0.00   2.00 3;
8 1.00    00.0   0.0000   0.00 0.0000   0.000    0.00   0.00 3;
9 1.00    00.0   0.0000   0.00 17.670   1.000    0.00   3.50 3;
10 1.00   00.0   0.0000   0.00 0.0000   0.000    0.00   0.00 3;
11 1.00   00.0   0.0000   0.00 0.0000   0.000    0.00   0.00 3];

%from bus, To bus, Resistant, Inductance, Capacitance, tap-ratio,
tap-phase
line = [01 05 0.0000 0.0167 0.00000 1.0 0.0;
         02 06 0.0000 0.0167 0.00000 1.0 0.0;
         03 11 0.0000 0.0167 0.00000 1.0 0.0;
         04 10 0.0000 0.0167 0.00000 1.0 0.0;
         05 06 0.0025 0.0250 0.04375 1.0 0.0;
         10 11 0.0025 0.0250 0.04375 1.0 0.0;
         06 07 0.0010 0.0100 0.01750 1.0 0.0;
         09 10 0.0010 0.0100 0.01750 1.0 0.0;
         07 08 0.0110 0.1100 0.19250 1.0 0.0;
         07 08 0.0110 0.1100 0.19250 1.0 0.0;
         08 09 0.0110 0.1100 0.19250 1.0 0.0;
         08 09 0.0110 0.1100 0.19250 1.0 0.0];
```

Script 2.3 Script for bus and line matrices of the four-machine test system.

```
%four_mac_run.m
four_mac_data
[Y] = form_Ymatrix(bus,line);
[bus_sol, line_flow] = power_flow(Y,bus, line)
```

Script 2.4 Script for running the four-machine test system power flow.

in the row, respectively. If everything goes right, the slack bus active and reactive power generation will be 7.2343 pu and 1.6024 pu, respectively, for the data mentioned above. Alternatively the same information can be found in the workspace. The *line_flow* shows active and reactive power flows in the transmission lines.

2.9 Exercise

Note the power flow is different from bus A to bus B compared with bus B to bus A. For example, note the power flow between buses 1 to 5 and 5 to 1. Try to explain the reason. Then look at the power flow of buses 5 to 6 and 6 to 5 and explain.

Let us try some more exercises:

- Reduce active power demand (Pload) at Bus 7 to 867 MW. What is the new result? The slack bus should now produce 631.94 MW and 148.84 MVar. Why is it that with the load drawing 100 MW less power, the slack only produces 91.49 MW less?
- With no change in reactive power load, why is there a change in reactive power generation?
- Now make G1 slack bus, G3 generator bus and set G3 output equal 700 MW. What is the new result? The slack bus should now produce 618.65 MW and 132.77 MVar. Why is it that simply swapping the slack bus and nothing else gives a different result for the apparent power production?
- Try making further changes and see if you can explain the results. For further simulations, set the data back to the original values.

2.10 Including the network in the Simulink time domain simulation

The power flow solution is the starting point to almost any time domain simulation in power systems. With this first step successfully mastered, running a time domain simulation of the network is not very difficult. As mentioned in the introduction chapter, given that certain circumstances are satisfied, the network can be represented as an impedance matrix for time domain simulations. Chapter V discusses the modelling of loads in detail. In the meantime, we will make the simplifying assumption that the loads are of impedance type, and we will include all loads implicitly in the impedance matrix.

A few more lines need to be added at the end of *four_mac_run.m*, before we are ready to set up the Simulink model. Copy code in Script 2.5 and append it in the *four_mac_run.m*.

In the first instance, create a Simulink model, selecting *file* then *new* then *model* and save this as *Network_Timedomain*. In the new model, the library browser can be found under the view tab. The blocks you require can be dragged across to the new model from there. To build the model, follow the steps as indicated in Fig. 2.6, until you have the same system.

```
%Calculates the vector of apparent power S̃ injected by the
%generators for each system bus.

        PQ = bus_sol(:,4)+1i*bus_sol(:,5);

%Calculates the complex voltage value for all buses.

        vol = bus_sol(:,2).*exp(1i*bus_sol(:,3)*pi/180);

%Calculates the complex value of the current injected by the
%generators at each system bus

        Icalc = conj(PQ./vol);


%PL and QL are the real and reactive power drawn by loads at all
%system buses.

        PL = bus_sol(:,6);
        QL = bus_sol(:,7);

%V is the voltage magnitude at all system buses.

        V = bus_sol(:,2);

%The loads are assumed to be impedance type loads here and are
%included in the Ỹ matrix. For more information on modelling loads,
%see Chapter V. The Ỹ matrix is then inverted to find the impedance
%matrix.

        YPL = PL./V.^2;
        YQL = QL./V.^2;
        Y = Y + diag(YPL-j*YQL);
        Z = inv(Y);
```

Script 2.5 Program to calculate impedance matrix.

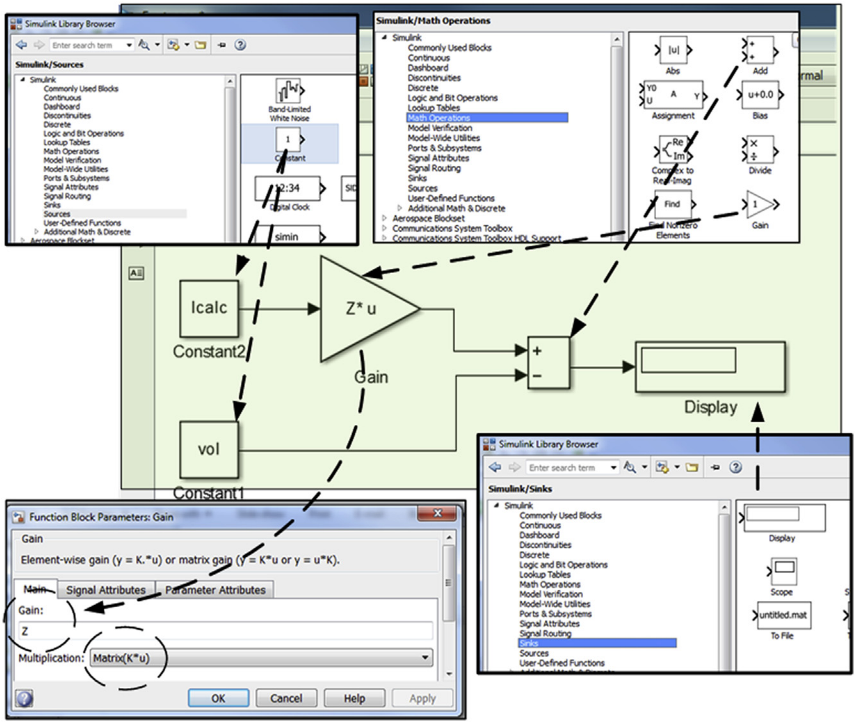

Figure 2.6 Creating the first Simulink model.

Once complete, click the run buttom, the black arrow on the top banner. If everything has been set up correctly, the output of the gain block should be vol; hence, vol–vol should be zero (or very, very small according to the precision of the calculation). The result from this first Simulink simulation can be seen in Fig. 2.7. We can see that for all bus voltages, the difference is 10^{-15} or smaller, which is zero for the purpose of the simulation. We will use this method for representing network in simulation models containing synchronous and asynchronous machines in the later chapters.

If this simulation throws up an error, check that the values from your power flow and initialization are still in the Matlab workspace. Open Block Parameters of the Gain block and ensure Multiplication option is set to Matrix (K*u). Similarly List of Signs of Add block is set to ±. Ensure that the Simulink model is in the same root directory as the power flow and initialization. If in doubt, rerun the initialization and the Simulink model. If the display does not show all numbers close to zero, go back and validate your power flow results and check the initialization.

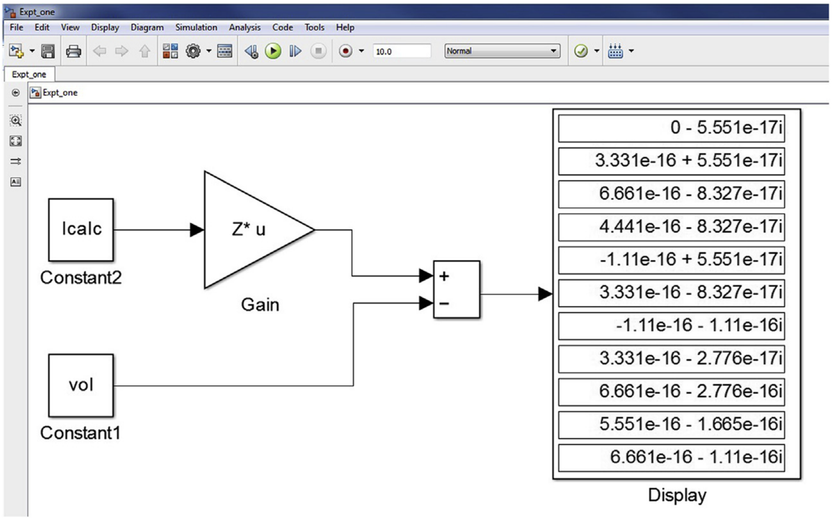

Figure 2.7 Result from the first Simulink simulation.

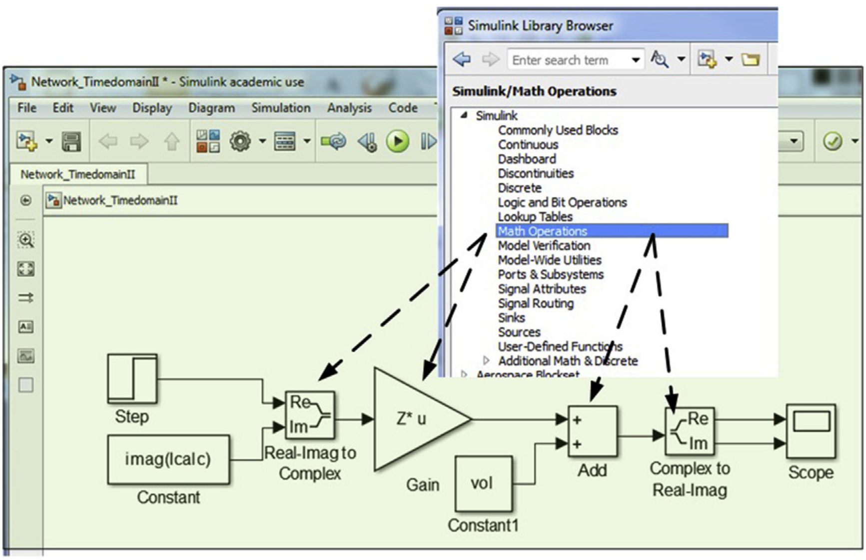

Figure 2.8 Network simulation with step change.

Is everything working so far? Time to introduce a step change. As before, follow the steps indicated in Fig. 2.8 to build a new model.

Under Sources in the library browser, there is a block called Step. Drag this into the model. Try setting the initial value to Icalc and click ok.

This should throw up an error message 'Invalid setting'. The reason for this is that certain Simulink blocks do not handle complex numbers. What to do now? The solution is to treat the real and imaginary parts separately. For this, four inbuilt functions are useful real(x), imag(x), abs(x) and angle(x). These provide the real part, imaginary part, magnitude and angle (in radians) of the complex number x. Let us say, we only want to introduce a step in the real part of Icalc, not in the imaginary part. The constant Icalc turns into imag(Icalc). Simple calculations like this can be typed directly into the Simulink model. The initial value for step is real(Icalc). Setting the final value of Icalc to real(Icalc)*2 leads to a 200% increase in the real part of Icalc at the time of the step. By default, this is set to 1 s. Now the real and imaginary parts need to be recombined to a complex number. You could either multiply the imaginary part by j and then add them or use another block from the library. In the math operations section, there is a block called 'real-imag to complex' to recombine the signals where no multiplication with the imaginary unit j is required. Another useful block from the library is the scope, found under sinks. Add this to your model. The scope, just like the step, does not take complex numbers as input. The real and imaginary parts or magnitude and angle can, however, be plotted separately. Use the Complex to Real-Imag block to split the signal. Double click on the scope and select the parameter symbol in the top ribbon (to the right of the printer symbol). In the general category, set the number of axis to two, and in the history selection, deselect the option 'limit data points to last:'. It is possible to stretch a box to make it larger. Simulink will show the name of the variable, e.g., in a constant block, as long as the box is large enough.

Run the model. Then double click the scope. This opens a window with the output from the scope, which will display plot as shown in Fig. 2.9.

2.11 Conclusions

This chapter has covered how to form the admittance matrix for a given network. The example used is a three-bus system with a PQ, PV and slack bus. The power flow computation has been discussed. For this purpose, the formation of the Jacobian matrix has been described, and the three-bus system was used as the example. The step-by-step iterative procedure of gaining increasingly accurate power flow solutions has been shown by using the numerical example of the three bus case. With the background behind the formation of the admittance and Jacobian matrix and the iterative power flow solution covered, a simple power flow program

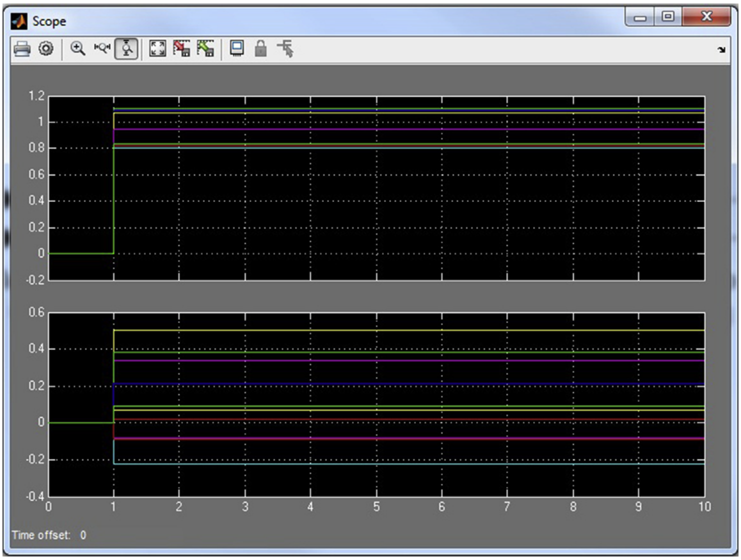

Figure 2.9 Scope showing step change in voltages.

has been introduced. In the example, the conversion of the four machine data from actual to pu data is covered, as the power engineer may be given data either in pu or in actual values. Hence, the conversion process is an important first step in the power flow calculation. Having attained a power flow solution of sufficiently high precision, the network can be modelled for time domain analysis in Simulink. Detailed guidance on this has been provided with snapshots of the simulation setup, as this is the first chapter using Simulink. Later chapters will assume a certain degree of familiarity of the user in building Simulink models. It was shown how to validate and troubleshoot the Simulink result and how to introduce a step change. You are now ready to go to the next chapter to learn how to integrate a detailed model of a conventional generator with the Simulink model of the network.

References

Arrillaga, J., Watson, N.R., 2001. Computer Modelling of Electrical Power Systems, Second Edition. John Wiley & Sons, Ltd.

Kundur, P., 1994. Power System Stability and Control. McGraw-Hill, Inc.