CHAPTER THREE

Synchronous machine modelling

3.1 Synchronous machine introduction

With ever-evolving history over the past 130 years, synchronous machine as technology (Neidhofer, 1992) has driven the development and progress of electrification in the planet through interconnected AC power network. The power networks in major countries each have thousands of synchronous generators producing power either at 50 or 60 Hz to keep the lights on. Both from the design and operational perspective, it is one of the major challenges to an engineer to model and analyze the machine. The dissipation of heat because of losses in the conductors and stresses on the insulation because of voltage have always challenged the engineer for going for large capacity individual machine with higher generation voltage. Accordingly, various currents and voltages are limited as defined by the generator capability curve. With continuous progress in cooling and insulation technologies, the capacity and constructional features of synchronous generator have continued to change but not the principle of synchronous generation. The current development trend is more on high-temperature superconductor technology for winding and air cooling for heat dissipation.

This chapter covers the simulation model for synchronous machines. First, the general operation of synchronous machines is described, detailing the differences between cylindrical rotor and salient pole machines. Next, a simple study system is introduced, the single machine infinite bus (SMIB) system. In the second section, the d-q and D-Q reference frames are introduced, which are essential for synchronous machine modelling. Dynamic equations for the machine modelling are provided. The model considers only the operation of the generator within its capability curve. Various limits such as rotor field heating current limit, stator heating current limit, armature core end iron heating limit, stability limit, etc., need to be modelled additionally. The initialization procedure to determine a steady-state operating point of the synchronous machine model is described, where the initialization is described both for system base and machine base. The Simulink implementation is described step by step, including the initial conditions. The Simulink section explicitly explains

Simulation of Power System with Renewables
ISBN: 978-0-12-811187-1
https://doi.org/10.1016/B978-0-12-811187-1.00003-2

how to model the generator in machine base and system base. The reader is instructed into the art of debugging Simulink models in a methodical manner, leading to a working model. By the end of the chapter, the reader will be ready to run a time domain simulation with a synchronous machine in machine and system base and understand how the results compare. Various dynamic synchronous machine models are introduced, to raise awareness among the reader about the different modelling assumptions made. The chapter concludes with step-by-step guidance of building a dynamic simulation model for a two-area test system consisting of four synchronous machines.

3.2 Synchronous machine operation

In this subsection, we discuss some of the fundamental concepts related to synchronous machines and how they work. This understanding will help the reader to be able to follow the later sections of this chapter.

Fig. 3.1 shows the schematic diagram of a three-phase synchronous machine. The machine consists of a stator with armature windings and a rotor with field windings; the rotor and stator are separated by a small air gap just enough to have clearance. The direction of rotor rotation is indicated in the figure; the electrical speed of rotation ω is directly linked (synchronized) with the network frequency, which is the main characteristic of a

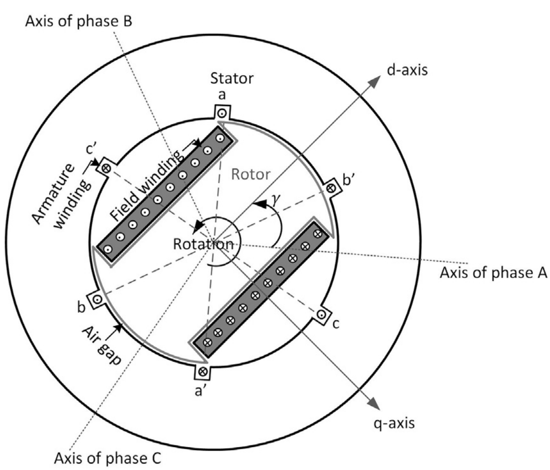

Figure 3.1 Schematic of a three-phase synchronous machine.

synchronous machine. In a machine with one pole pair, the mechanical rotational speed ω_m is the same as the electrical speed of rotation. In a machine with more pole pairs (n), the rotor needs to turn slower to achieve the same variation in the magnetic field, such that $\omega = \omega_m n$. The field current in the field winding of the rotor is a DC current supplied by a field exciter. The armature windings are laid out, such that the three phases are 120 electrical degrees apart, where 360 electrical degrees is the angular distance between identical magnetic conditions. This definition is important in machines with multiple pole pairs. For the general understanding and development of equations, we can use a single-pole pair machine, without loss of generality. As the rotor rotates at an electrical speed ω, it can be naturally understood that the induced voltages in the armature are time varying with ω, with a phase difference of 120° between each phase.

Let us use the well-known right-hand rule to work out the meaning of the different axes in Fig. 3.1. The stator has three armature windings for the three phases, the first winding is between a and a', another winding is between b and b' and the third winding is between c and c', as indicated in Fig. 3.1. Using the right-hand rule, can one determine the reason for the direction of phase A, B and C axes, shown in Fig. 3.1? In the same way, we can investigate how the orientation of the field winding on the rotor is related to the direction of the axis named d-axis in Fig. 3.1.

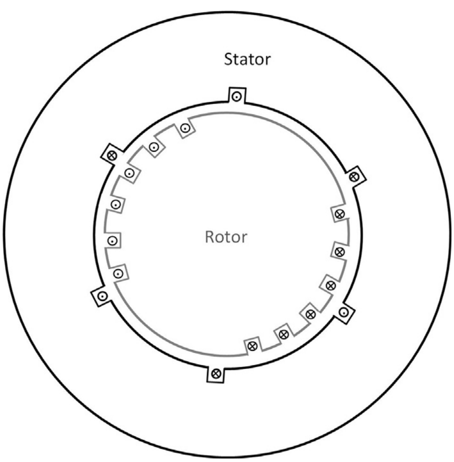

Figure 3.2 Cylindrical rotor synchronous machine.

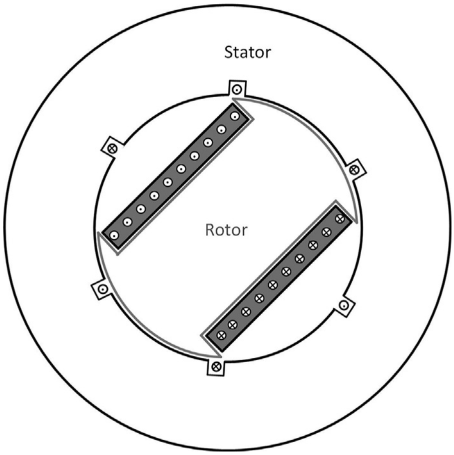

Figure 3.3 Salient pole synchronous machine.

There are two general types of rotors, cylindrical rotors and salient pole rotors. The difference between the two types can be clearly seen in Figs. 3.2 and 3.3.

It can be seen that for both the cylindrical and salient pole machine, there exist two axis of symmetry with respect to the rotor shape and air gap. These two axes of symmetry are the d-axis and q-axis, shown in Figure 3.1. As the rotor with the rotor circuits is rotating relative to the stator circuits, the time-varying flux path leads to self- and mutual inductances of the armature windings as well as mutual inductances between armature and rotor circuits, which are time varying, according to the rotor position (Clarke, 1957). Time-varying inductances will complicate the machine model, and resulting equations are computationally expensive to solve. The dq-transformation discussed in the next section is helpful to simplify the machine model.

3.3 Reference frame

In this section, we will see how the symmetry of the dq-axis can be used to analyze a machine, where the rotor circuits are rotating at a constant speed relative to the stator circuits. The dq-transformation avoids the need to work with time-varying inductances, which makes it a powerful tool for the analysis of synchronous machines. This section explains the mathematical transformation between the three-phase voltages of phase A, B and C and the dq0 representation. This section also covers the dq-reference frame of synchronous machine and the DQ-reference of the system, which is required for a network with several generators, which will operate in different dq-frames. The transition from one reference frame to the other is described as well.

Direct-quadrature-zero (dq0) transformation is frequently used in the analysis of three-phase circuits. As shown in Fig. 3.1, the direct component is aligned along the field axis (d-axis), which is perpendicular to the plane of the field windings. The quadrature component is aligned with the quadrature (q) axis, which lags the field axis by 90°. The d-axis leading the q-axis is IEEE convention. As the field axis is part of the rotor, it rotates at the speed ω in the direction indicated in the figure. The d-axis and q-axis rotate together with the rotor. In effect, the dq0 transformation changes the reference frame from a stationary frame to a rotating one, such that rotating phasors appear stationary from the perspective of this rotating reference frame. In any three-phase balanced system, the zero components are zero, such that we are left with a direct and quadrature component; hence, we will call the reference frame a dq-reference frame.

The readers should be aware at this point that there are multiple conventions in use. In simulations, care has to be taken to use the same convention consistently. This book uses the generator convention and power invariant park transform. The generator convention power injected into a network is positive, while power drawn from a network is negative. The orientation of the direct and quadrature axis has been extensively discussed in the literature. An IEEE Committee report (IEEE committee report, 1969) on phasor diagrams in 1969 concluded that it is preferred for the direct axis to lead the quadrature axis by 90°. 'Using this principle, the quadrature axis can be taken as the real axis. The direct axis then becomes the imaginary axis…'. In this book we follow this convention, such that $V_t = V_q + jV_d$. Fig. 3.4 shows the direct axis, quadrature axis, V_t, V_q and V_d.

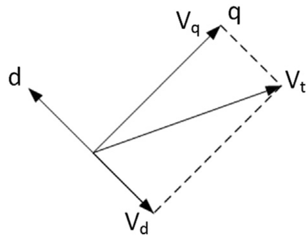

Figure 3.4 Phasor diagram of dq-frame.

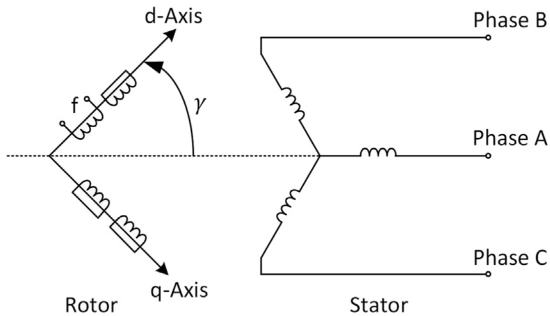

Figure 3.5 Showing position of dq-axis in relation with three-phase stator voltages.

The angle between the axis of phase A and the d-axis is denoted here as γ, as can be seen in Figs. 3.1 and 3.5. As the rotor and the dq-axis are turning at a speed ω, γ will be time varying at $\gamma = \gamma_0 + \omega t$. The field coil and one equivalent damper coil are on the d-axis, while there are two equivalent damper coils on the q-axis.

As the dq-reference frame is aligned with the field of the synchronous machine, it is easy to imagine how there may be several different dq-reference frames in a system, which has several synchronous machines. When simulating such a system, we need a common reference frame. This reference frame is the network reference frame, named DQ-reference frame. The DQ-reference frame in relation to the dq-reference frame is shown in Fig. 3.6.

When the study system contains an infinite bus, the slack bus will be the infinite bus by nature. If the infinite bus angle is zero degrees, the Q-axis of the DQ-frame is aligned with the infinite bus voltage; if the infinite bus has an angle of say 10°, the Q-axis would be lagging the infinite bus voltage by

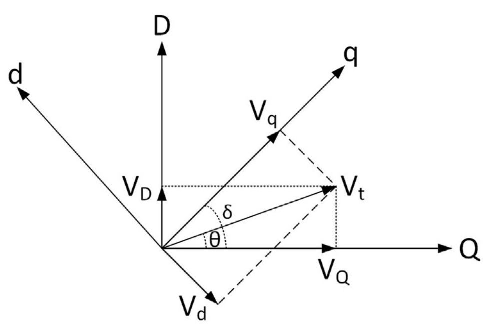

Figure 3.6 Phasor diagram DQ- and dq-frame.

10°. In a study system without an infinite bus, the angle of the slack bus and hence the network voltage reference can change during the simulation. As the slack bus angle is arbitrary, we can retrospectively change all system angles, such that the slack bus angle remains fixed along with the network voltage reference. This subtle point will become clear in Chapter 5, during discussion of the four-machine system.

θ is the voltage angle of the voltage V_t with reference to the zero degree network reference angle.

δ is the angle by which the q-axis leads the Q-axis, where the q-axis is aligned with the field flux—produced rotational EMF or speed voltage of the respective synchronous machine and the Q-axis is aligned with a network voltage angle of zero, where all network angles are measured relative to a common reference bus. All machine dq-reference frames can now be expressed relative to the DQ-reference frame.

The machine equations are in their respective dq-reference frames, which align with the field of the machine. The network equations are in the DQ-reference frame, which is a common reference frame for all machines as often imposed by the network. At the interfaces between the machines and the network, we need to convert from the dq-reference frames to the DQ-reference frame and vice versa. In Chapter 2 on power flow, we have seen that the network model takes current as an input and gives voltage as an output. The current can be, for example, from the synchronous machines into the network, and the network voltage is passed as an input to the synchronous machines, choosing the dq-reference frame, which is aligned with the respective synchronous machine. From Fig. 3.4, we can see that the conversion between dq-frame and DQ-frame can be achieved using Eqs. (3.1) and (3.2).

$$V_q + jV_d = (V_Q + jV_D)e^{-j\delta} \tag{3.1}$$

$$I_q + jI_d = (I_Q + jI_D)e^{-j\delta} \tag{3.2}$$

So far, we have talked about the dq-frame and DQ-frame; however, we have not discussed how we can convert the rotating three-phase quantities into the stationary two-phase dq-representation. Let us look once in detail at how this transformation works, later we can simply use the results. If V is the RMS line-to-line voltage, then $\sqrt{2}\ V$ is the peak value of the line-to-line voltage and V is the line-to-neutral peak voltage. In a three-phase system, the phases are equally spaced apart. With a full rotation at 2π, this means that the phases are $\frac{2\pi}{3}$ spaced apart. If ω is the rotational speed of the three phasors, we can write a vector of the three phasors. If the phasors are balanced and equally spaced, the vector is as follows:

$$V_{3ph} = \begin{bmatrix} \sqrt{\frac{2}{3}}\,V\sin(\omega t + \theta) \\ \sqrt{\frac{2}{3}}\,V\sin\left(\omega t - \frac{2\pi}{3} + \theta\right) \\ \sqrt{\frac{2}{3}}\,V\sin\left(\omega t + \frac{2\pi}{3} + \theta\right) \end{bmatrix} \tag{3.3}$$

The dq0 transformation is calculated by multiplying a matrix with the balanced three-phase vector, to find the dq0 vector, where the 0 element is zero.

$$V_{dq0} = \begin{bmatrix} V_d \\ V_q \\ 0 \end{bmatrix} \tag{3.4}$$

$$K * V_{3ph} = V_{dq0} \tag{3.5}$$

The inverse of this matrix is multiplied with V_{dq0} to transform the voltage back to the three-phase version.

$$K^{-1} * V_{dq0} = V_{3ph} \tag{3.6}$$

Such a matrix K has been proposed by Park and was later revised such that the transformation is power invariant.

$$K = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\gamma & \cos\left(\gamma - \frac{2\pi}{3}\right) & \cos\left(\gamma + \frac{2\pi}{3}\right) \\ \sin\gamma & \sin\left(\gamma - \frac{2\pi}{3}\right) & \sin\left(\gamma + \frac{2\pi}{3}\right) \\ \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \tag{3.7}$$

This matrix has a very useful property when we need to find the inverse of the matrix. It is an orthogonal matrix. Let us write a simple code to check this. We chose any arbitrary angle for γ, in this example code $\frac{\pi}{8}$. Now, we calculate the value of K and the inverse of K, which is M2 and the transpose of K, which is M3. If we run Script 3.1 code, we can see that M2 and M3 are

```
gamma=pi/8;
K =sqrt(2/3)*[cos(gamma) cos(gamma -(2*pi/3)) cos(gamma
+(2*pi/3)); sin(gamma) sin(gamma -(2*pi/3)) sin(gamma
+(2*pi/3)); 1/sqrt(2)   1/sqrt(2)   1/sqrt(2)];
M2=inv(K)
M3=K.'
```

Script 3.1 Inverse and transpose of orthogonal matrix.

identical; this is because K is an orthogonal matrix that fulfils the condition $A^T * A = I$, where I is the identity matrix. Calculate M3*K and confirm if it is the identity matrix I. This result is very useful because we can take the transpose of the matrix K, rather than calculating an inverse.

$$K^{-1} = \sqrt{\frac{2}{3}}\begin{bmatrix} \cos\gamma & \sin\gamma & \sqrt{\frac{1}{2}} \\ \cos\left(\gamma - \frac{2\pi}{3}\right) & \sin\left(\gamma - \frac{2\pi}{3}\right) & \sqrt{\frac{1}{2}} \\ \cos\left(\gamma + \frac{2\pi}{3}\right) & \sin\left(\gamma + \frac{2\pi}{3}\right) & \sqrt{\frac{1}{2}} \end{bmatrix} \tag{3.8}$$

We will now calculate V_{dq0} and V_{3ph} using (3.5) and (3.6). We only need to do this once to understand the transformation; later, we can rely on the result and avoid the detailed steps.

According to (3.5), V_{dq0} is

$$\sqrt{\frac{2}{3}}\begin{bmatrix} \cos\gamma & \cos\left(\gamma - \frac{2\pi}{3}\right) & \cos\left(\gamma + \frac{2\pi}{3}\right) \\ \sin\gamma & \sin\left(\gamma - \frac{2\pi}{3}\right) & \sin\left(\gamma + \frac{2\pi}{3}\right) \\ \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} * \begin{bmatrix} \sqrt{\frac{2}{3}}V\sin(\omega t + \theta) \\ \sqrt{\frac{2}{3}}V\sin\left(\omega t - \frac{2\pi}{3} + \theta\right) \\ \sqrt{\frac{2}{3}}V\sin\left(\omega t + \frac{2\pi}{3} + \theta\right) \end{bmatrix} \tag{3.9}$$

To calculate this, we quickly need to remind ourselves of a few useful trigonometric identities.

$$\sin\alpha\cos\beta = \frac{1}{2}\left(\sin(\alpha+\beta) + \sin(\alpha-\beta)\right) \tag{3.10}$$

$$\sin\alpha\sin\beta = \frac{1}{2}\left(\cos(\alpha-\beta) - \cos(\alpha+\beta)\right) \tag{3.11}$$

$$\frac{2\pi}{3} = \frac{-4\pi}{3}, \frac{-2\pi}{3} = \frac{4\pi}{3} \tag{3.12}$$

$$\sin(-\alpha) = -\sin(\alpha) \tag{3.13}$$

$$\sin(\alpha) + \sin\left(\alpha - \frac{2\pi}{3}\right) + \sin\left(\alpha + \frac{2\pi}{3}\right) = 0 \tag{3.14}$$

We also introduce a new reference angle δ, which is rotating at the rotational speed ω such that the angle γ can be written as γ = δ + ωt.

First, we calculate V_d from the first row of K and V_{3ph} as seen in (3.9).

$$\begin{aligned} V_d = \frac{1}{3}V &\left(\sin(\omega t + \theta + \gamma) + \sin(\omega t + \theta - \gamma) + \sin\left(\omega t + \theta + \gamma - \frac{4\pi}{3}\right)\right. \\ &\left. + \sin(\omega t + \theta - \gamma) + \sin\left(\omega t + \theta + \gamma + \frac{4\pi}{3}\right) + \sin(\omega t + \theta - \gamma)\right) \\ &= -V\sin(-\omega t - \theta + \gamma) = -V\sin(\delta - \theta) \end{aligned} \tag{3.15}$$

Next, we calculate V_q from the second row of K and V_{3ph} as seen in (3.9).

$$\begin{aligned} V_q = \frac{1}{3}V &\left(\cos(\gamma - \omega t - \theta) - \cos(\gamma + \omega t + \theta) + \cos(\gamma - \omega t - \theta)\right. \\ &- \cos\left(\gamma + \omega t + \theta - \frac{4\pi}{3}\right) + \cos(\gamma - \omega t - \theta) \\ &\left. - \cos\left(\gamma + \omega t + \theta + \frac{4\pi}{3}\right)\right) = V\cos(\gamma - \omega t - \theta) = V\cos(\delta - \theta) \end{aligned} \tag{3.16}$$

Finally, we calculate the zero element from the third row of K and V_{3ph} as seen in (3.9). Keeping the trigonometric identity in mind, we can see by inspection that the last element is zero. This is the complete dq0 transform. Now, let us see if we get back to the original result if we transform back to three phases. For this, we follow (3.6).

According to (3.6), V_{3ph} is

$$\sqrt{\frac{2}{3}}\begin{bmatrix} \cos\gamma & \sin\gamma & \sqrt{\frac{1}{2}} \\ \cos\left(\gamma-\frac{2\pi}{3}\right) & \sin\left(\gamma-\frac{2\pi}{3}\right) & \sqrt{\frac{1}{2}} \\ \cos\left(\gamma+\frac{2\pi}{3}\right) & \sin\left(\gamma+\frac{2\pi}{3}\right) & \sqrt{\frac{1}{2}} \end{bmatrix} * \begin{bmatrix} -V\sin(\delta-\theta) \\ V\cos(\delta-\theta) \\ 0 \end{bmatrix} \tag{3.17}$$

The equation for the first phase V_a is

$$\begin{aligned} V_a &= \sqrt{\frac{1}{6}}\,V(-\sin(\delta-\theta+\gamma)-\sin(\delta-\theta-\gamma)+\sin(\gamma+\delta-\theta) \\ &\quad +\sin(\gamma-\delta+\theta)) \\ &= \sqrt{\frac{1}{6}}\,V(-\sin(\delta-\theta-\gamma)+\sin(\gamma-\delta+\theta)) \\ &= \sqrt{\frac{2}{3}}\,V\sin(\gamma-\delta+\theta)=\sqrt{\frac{2}{3}}\,V\sin(\omega t+\theta) \end{aligned} \tag{3.18}$$

The equation for the second phase V_b is

$$\begin{aligned} V_b &= \sqrt{\frac{1}{6}}\,V\left(-\sin\left(\delta-\theta+\gamma-\frac{2\pi}{3}\right)-\sin\left(\delta-\theta-\gamma+\frac{2\pi}{3}\right)\right. \\ &\quad \left.+\sin\left(\gamma+\delta-\theta-\frac{2\pi}{3}\right)+\sin\left(\gamma-\delta+\theta-\frac{2\pi}{3}\right)\right) \\ &= \sqrt{\frac{2}{3}}\,V\sin\left(\omega t-\frac{2\pi}{3}+\theta\right) \end{aligned} \tag{3.19}$$

The equation for the third phase V_c is

$$\begin{aligned} V_c &= \sqrt{\frac{1}{6}}\,V\left(-\sin\left(\delta-\theta+\gamma+\frac{2\pi}{3}\right)-\sin\left(\delta-\theta-\gamma-\frac{2\pi}{3}\right)\right. \\ &\quad \left.+\sin\left(\gamma+\delta-\theta+\frac{2\pi}{3}\right)+\sin\left(\gamma+\frac{2\pi}{3}-\delta+\theta\right)\right) \\ &= \sqrt{\frac{2}{3}}\,V\sin\left(\omega t+\frac{2\pi}{3}+\theta\right) \end{aligned} \tag{3.20}$$

This shows that the elements V_a, V_b and V_c of V_{3ph} are still the same as before. The three rotating voltage phasors in a balanced three-phase system can hence be represented using two stationary phasors, which significantly simplify modelling. In the future, we can skip the conversion. Instead of using three time-varying phasors V_{3ph}, we work with V_t, which is $\mathrm{V}e^{-j(\delta-\theta)}$in dq-frame and $\mathrm{V}e^{j\theta}$ in the DQ-frame. We can see this using (3.15), (3.16) and (3.1).

3.4 Dynamic equations of a synchronous machine in d-q reference frame

This section provides all the differential algebraic equations required for a time domain simulation of a synchronous machine. The equations capture the subtransient dynamic behaviour of a synchronous generator, with four equivalent coils as shown in Fig. 3.5 and a single lumped mass representation for the torque angle loop (Pal and Chaudhuri, 2010).

The simplest mechanical model is a single lumped mass model, as shown in Fig. 3.7. This very simple model provides information about the rotating speed of the generator, considering the applied mechanical torque and the produced electrical torque. The single-mass representation is sufficient for the power system phenomena studied in this book. The equations governing the dynamics of the single-mass model are (3.21) and (3.22).

Eq. (3.21) captures the change in generator rotor angle, due to a mismatch between the synchronous speed of the network and the rotor angular speed. $\omega_r - \omega_s$ has to be multiplied by the base value ω_b, as δ in (3.21) is an angle, which is not converted to the pu system.

$$\frac{d\delta}{dt} = \omega_b(\omega_r - \omega_s) \tag{3.21}$$

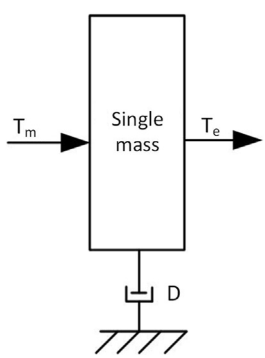

Figure 3.7 Single-mass model.

Eq. (3.22) is needed to describe the change in the rotor angular speed due to a mismatch of torques in the mechanical system. If the input mechanical torque is greater than the electrical torque, while accounting for damping, the rotor will speed up. If the electrical torque is greater than the mechanical torque, considering damping, the rotor will slow down. The rate at which the rotor speeds up or slows down is determined by the inertia of the generator.

$$\frac{d\omega_r}{dt} = \frac{1}{2H}\left(T_m - T_e - D(\omega_r - \omega_s)\right) \tag{3.22}$$

The mechanical torque is an input quantity for the synchronous machine simulation. The electrical torque can be calculated from the electrical equations of the machine, as shown in (3.23).

$$T_e = \frac{X_d'' - X_{ls}}{X_d' - X_{ls}} E_q' I_q + \frac{X_d' - X_d''}{X_d' - X_{ls}} \psi_{1d} I_q + \frac{X_d'' - X_{ls}}{X_d' - X_{ls}} E_q' I_d - \frac{X_d' - X_d''}{X_d' - X_{ls}} \psi_{2q} I_d - \left(X_d'' - X_d''\right) I_q I_d \tag{3.23}$$

The change in the transient EMF due to the field flux linkage (3.24) is indirectly proportional to the transient d-axis open-circuit time constant.

$$\frac{dE_q'}{dt} = \frac{1}{T_{d0}'}\left[-E_q' + E_{fd} + \left(X_d - X_d'\right)\left(I_d + \frac{X_d' - X_d''}{\left(X_d' - X_{ls}\right)^2}\left\{\psi_{1d} - E_q' - I_d\left(X_d' - X_{ls}\right)\right\}\right)\right] \tag{3.24}$$

Similarly, the transient EMF due to flux linkage in q-axis damper coil (3.25) is indirectly proportional to the transient q-axis open-circuit time constant.

$$\frac{dE_d'}{dt} = \frac{1}{T_{q0}'}\left[-E_d' + \left(X_q - X_q'\right)\left(-I_q + \frac{X_q' - X_q''}{\left(X_q' - X_{ls}\right)^2}\left\{-\psi_{2q} - E_d' + I_q\left(X_q' - X_{ls}\right)\right\}\right)\right] \tag{3.25}$$

The subtransient EMF due to flux linkage in the q-axis damper coil (3.26) is indirectly proportional to the subtransient q-axis open-circuit time constant.

$$\frac{d\,\psi_{2q}}{dt} = \frac{1}{T_{q0}^{''}} \left(-\psi_{2q} - E_d^{'} + I_q\left(X_q^{'} - X_{ls}\right)\right) \tag{3.26}$$

The subtransient EMF due to flux linkage in the d-axis damper coil (3.27) is indirectly proportional to the subtransient d-axis open-circuit time constant.

$$\frac{d\,\psi_{1d}}{dt} = \frac{1}{T_{d0}^{''}} \left(-\psi_{1d} + E_q^{'} + I_d\left(X_d^{'} - X_{ls}\right)\right) \tag{3.27}$$

Using the four state variables and the generator terminal voltage, which is an input to the synchronous machine model, the q- and d-axis stator current components can be calculated as seen in (3.28) and (3.29), respectively.

$$\begin{aligned} I_q = &\frac{R_s}{R_s^2 + -X_d''^2}\left(E_q'\frac{X_d'' - X_{ls}}{X_d' - X_{ls}} + \psi_{1d}\frac{X_d' - X_{ls}''}{X_d' - X_{ls}} - V_q\right) \\ &+ \frac{X_d''}{R_s^2 + -X_d''^2}\left(E_d'\frac{X_q'' - X_{ls}}{X_q' - X_{ls}} - \psi_{2q}\frac{X_q' - X_q''}{X_q' - X_{ls}} - V_d\right) \end{aligned} \tag{3.28}$$

$$\begin{aligned} I_d = &\frac{-X_d''}{R_s^2 + -X_d''^2}\left(E_q'\frac{X_d'' - X_{ls}}{X_d' - X_{ls}} + \psi_{1d}\frac{X_d' - X_{ls}''}{X_d' - X_{ls}} - V_q\right) \\ &+ \frac{R_s}{R_s^2 + -X_d''^2}\left(E_d'\frac{X_q'' - X_{ls}}{X_q' - X_{ls}} - \psi_{2q}\frac{X_q' - X_q''}{X_q' - X_{ls}} - V_d\right) \end{aligned} \tag{3.29}$$

Synchronous machines use governors to regulate the system frequency during changes in the generation—load balance, by adjusting the input torque to the generator. A simple governor can be represented in (3.30).

$$\frac{d\,T_m}{dt} = \frac{1}{T_g}\left(T_{m2} - T_m - \frac{\omega_r - \omega_s}{R_{gov}}\right) \tag{3.30}$$

Synchronous machines use excitation systems to regulate the generator bus voltage, by adjusting the generator field voltage. A simple static excitation system can be represented in (3.31).

$$\frac{d\,E_{fd}}{dt} = \frac{1}{T_a}\left(K_a E_t - K_a V_{ref} - E_{fd}\right) \tag{3.31}$$

List of variables:
δ Generator rotor angle
ω_s Synchronous speed
$\boldsymbol{H}$ Inertia constant
$\boldsymbol{D}$ Self-damping
Transient EMF due to field flux linkage
$\boldsymbol{E_{fd}}$ Field voltage
$\boldsymbol{R_s}$ Armature resistance
ψ_{1d} Subtransient EMF due to flux linkage in d-axis damper
$\boldsymbol{I_d}$ d-axis component of stator current
$\boldsymbol{V_d}$ d-axis component of generator terminal voltage
$\boldsymbol{X_d}, \boldsymbol{X'_d}, \boldsymbol{X''_d}$ Synchronous, transient and subtransient d-axis reactances
$\boldsymbol{T'_{do}}, \boldsymbol{T''_{do}}$ Transient and subtransient d-axis open-circuit time constants
$\boldsymbol{K_a}$ Static excitation gain
$\boldsymbol{E_t}$ Generator voltage magnitude
$\boldsymbol{T_g}$ Time constant of governor
$\boldsymbol{R_{gov}}$ Governor droop
ω_r Rotor angular speed
ω_b Base value of speed
$\boldsymbol{T_m}$ Mechanical torque
$\boldsymbol{T_e}$ Electrical torque
$\boldsymbol{E'_d}$ Transient EMF due to flux linkage in q-axis damper coil
$\boldsymbol{X_{ls}}$ Armature leakage reactance
ψ_{2q} Subtransient EMF due to flux linkage in q-axis damper
$\boldsymbol{I_q}$ q-axis component of stator current
$\boldsymbol{V_q}$ q-axis component of generator terminal voltage
$\boldsymbol{X_q}, \boldsymbol{X'_q}, \boldsymbol{X''_q}$ Synchronous, transient and subtransient q-axis reactances
$\boldsymbol{T'_{qo}}, \boldsymbol{T''_{qo}}$ Transient and subtransient q-axis open-circuit time constants
$\boldsymbol{T_a}$ Static excitation time constant
$\boldsymbol{V_{ref}}$ Excitation voltage reference
$\boldsymbol{T_{m2}}$ Generator load reference

3.5 Initialization of the dynamic model

In this section, we examine how to find the initial conditions for the dynamic equations discussed in the last section. For this purpose, we chose a very simple study system, which we will use for the remainder of the chapter. In this system, the synchronous machine is connected via a line to an infinite bus, as shown in Fig. 3.8. This network is commonly referred to

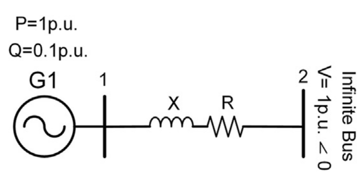

Figure 3.8 Single machine infinite bus (SMIB) system.

as SMIB system. As this will become apparent shortly, to find the initial conditions for the synchronous machine, the power flow solution, discussed in Chapter 2, is used first, to find the generator bus quantities. With the terminal quantities known, the synchronous machine initial conditions can be calculated.

An infinite bus is a bus, which has a fixed voltage (both magnitude and angle) and frequency. The infinite bus can be used to represent the connection to a strong grid, which will absorb the injected power at the infinite bus connection point, without a noticeable change in the voltage or frequency. While the voltage angle and magnitude at Bus 2 is fixed, the voltage magnitude and angle at Bus 1 varies according to the current injected at Bus 1. Please think about power system situations, where a section of a grid can be considered as a strong grid and can be replaced by an infinite bus. Can you think of an example?

The voltage at the infinite bus in this particular example is fixed as 1pu with a zero degree reference angle. Let us assume the generator has a power output of 7pu real power and 1.85pu reactive power. Normally, a typical 700 MW capacity machine will have 7.0 pu of power at 100 MW base. The power flow solution is required to find the voltage at Bus 1 and the power at Bus 2.

Start by creating a new m-file called *SMIB_data.m*. For the power flow, we require the bus and line data in the according format as described in Chapter 2. Before you read further, try and write the bus data yourself and then see if you got the same correct answer, as described in Script 3.2.

Bus 2 has a fixed voltage magnitude and angle, so it is clearly a slack bus, indicated by a 1 at the end of the line. The second and third elements for Bus 2 ensure that the bus voltage is 1pu with an angle of 0°. It was mentioned earlier that generators are often set as PV buses. In this case, this does not make sense as we know the real and reactive power and not the terminal voltage of the generator; hence, it is set to work as a PQ bus. This is indicated by a 3 at the end of the data for Bus 1.

```
% Bus No., Voltage, Angle, Pg, Qg,Pl, Ql, Gl, Bl, Bus type
bus = [...
1 1.00 0.00 7.00 1.85 0.00 0.00 0.00 0.00 3;
2 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1;
];
% From bus, To bus, R L C tap ratio
line = [01 02 0.0025 0.0417 0.0000 1.0 0.0];
```

Script 3.2 Single machine infinite bus (SMIB) data.

```
% Obtain load flow solution
[Y] = form_Ymatrix(bus,line);
% calculate pre-fault power flow solution
[bus_sol, line_flow] = power_flow(Y,bus, line);
```

Script 3.3 Power flow solver.

We can use the code from Chapter 2 in Script 3.3 to find the following power flow solution:

```
bus_sol =

1 1.0520 15.8480 7.0000 1.8500 0 0 0 0 3
2 1.0000 0       -6.8816 0.1254 0 0 0 0 1

line_flow =

1 2  7.0000 1.8500
2 1 -6.8816 0.1254
```

Next, we find the initial conditions for the synchronous machine. In the introduction chapter, pu conversion and base values have been introduced, and we have used this conversion also in Chapter 2. All pu calculations for the power flow have the same base value, the system base. The parameters for synchronous machines are frequently provided in a different pu system, where the base value is the machine base, which depends on the name plate rating of the machine. When machine parameters are provided in machine base, their parameters typically vary only within a certain range. This allows power system engineers to develop an intuition of machine parameters. For example, the system base may be 100MVA, while the machine base is 900MVA. For the simulation, the engineer has two options. One is to convert the whole system to be in the pu system with the same system base. The second option is to keep the machine parameters in their individual machine base and the network in the system base and convert the parameters at the interface. In this chapter, we will introduce both methods to show that in power systems, there can sometimes be several methods that are correct in their own right. Ultimately, the results will be equivalent if both methods are correct.

First, we write our main script SMIB_run, as shown in Script 3.4, which includes the bus, line, machine, excitation and governor data. The script also solves the load flow. When executing the line *Synch_parameter_gen_base*, the machine will be initialized with the machine base, and when the first line is *Synch_parameter_sys_base*, the machine will be initialized with the system base. *generic_Sync_Init* initializes the synchronous machines. Finally, a number of system quantities are calculated to facilitate the Simulink model. Vinf is the voltage vector of the infinite bus voltage

```
%SMIB_run
clear all
system_base_mva = 100.0;
s_f=60; %System frequency in Hz
s_wb=2*pi*s_f; %Base value radial frequency in rad/sec
s_ws=1; %p.u. value of synchronous speed

% Bus No., Voltage, Angle, Pg, Qg,Pl, Ql, Gl, Bl, Bus type
bus = [...
1 1.00 0.00 7.00 1.85 0.00 0.00 0.00 0.00 3;
2 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1;
];
% From bus, To bus, R L C tap ratio
line = [01 02 0.0025 0.0417 0.0000 1.0 0.0];

% MACHINE DATA STARTS

mac_con =[
1 900 0.2 0.0025 1.8 0.3 0.25 8 0.03 1.7 0.55 0.25 0.4 0.05
6.5 0];

% MACHINE DATA ENDS

% EXCITATION SYSTEM DATA
s_Ka=200;%Static excitation gain Padiyar p.328
s_Ta=0.02;%Static excitation time constant Padiyar p.328
% Governor Control SYSTEM DATA
s_Tg=0.2;%Kundur p.598
s_Rgov=0.05;%Kundur p.598

% Obtain load flow solution
[Y] = form_Ymatrix(bus,line);
% calculate pre-fault power flow solution
[bus_sol, line_flow] = power_flow(Y,bus, line);

% Synchronous machine initialisation

Smachs=[1];% buses with synchronous machines

Nbus=size(bus,1); % Number of buses
NSmachs=size(Smachs,1); % Number of synchronous machines

% program to initialise the synchronous machines

Synch_parameter_sys_base % or _gen_base, depending on the base
%system we decide to work in.

generic_Sync_Init

%Calculates the vector of apparent power S 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
%all system buses.
PL = bus_sol(:,6);
QL = bus_sol(:,7);
%V is the voltage magnitude at all system buses.
V = bus_sol(:,2);

Vinfvec=ones(Nbus,1);
Vinf=Vinfvec.*(bus_sol(2,2)*exp(1i*bus_sol(2,3)*pi/180));
 %bus voltage at infinite bus
Zline=line(3)+1i*line(4); %R+jX
%Line impedance
```

Script 3.4 SMIB_run

```
%Synch_parameter_gen_base.m
machine_numb=mac_con(:,1);
machine_base_mva=mac_con(:,2);
s_xls=mac_con(:,3); %Armature leakage reactance
s_Rs=mac_con(:,4); %Armature resistance
s_xd=mac_con(:,5); %d axis synchronous reactance
s_xdd=mac_con(:,6); %d axis transient reactance
s_xddd=mac_con(:,7); %d axis sub-transient reactance
s_Tdod=mac_con(:,8); %d axis open circuit transient time
%constant
s_Tdodd=mac_con(:,9); %d axis open circuit sub-transient time
%constant
s_xq=mac_con(:,10); %q axis synchronous reactance
s_xqd=mac_con(:,11); %q axis transient reactance
s_xqdd=mac_con(:,12); %q axis sub-transient reactance
s_Tqod=mac_con(:,13); %q axis open circuit transient time
%constant
s_Tqodd=mac_con(:,14); %q axis open circuit sub-transient time
%constant
s_H= mac_con(:,15); %Inertia constant
s_D=mac_con(:,16); %Self-damping
s_Rgov = s_Rgov;

ExpandSmachs= zeros(Nbus,NSmachs);
for counter=1:NSmachs
ExpandSmachs(Smachs(counter),counter)=1;
end

SelectSmachs= zeros(NSmachs,Nbus);
for counter=1:NSmachs
SelectSmachs(counter,Smachs(counter))=1;
end

s_Pg = bus_sol(Smachs,4) .*system_base_mva./machine_base_mva;
%generator real power in machine base
s_Qg = bus_sol(Smachs,5) .*system_base_mva./machine_base_mva;
%generator reactive power in machine base
```

Script 3.5 Synch_parameter_gen_base.m in machine base.

reference in complex form for all network buses, and Zline is the impedance of the single line.

Using *Synch_parameter_gen_base*, the machine will be initialized with the machine base as shown in Script 3.5. We can see that a base conversion is required for s_Pg and s_Qg. These values are taken from the power flow results, which are in system base. If we wish to keep the machine parameters in the machine base, we need to convert the generator power also from the system base to the machine base.

Using *Synch_parameter_sys_base*, the machine will be initialized with the system base as shown in Script 3.6. The difference between both codes is highlighted in bold letters.

```
%Synch_parameter_sys_base
machine_numb=mac_con(:,1);
machine_base_mva=mac_con(:,2);
s_xls=mac_con(:,3).*system_base_mva./machine_base_mva;
%Armature leakage reactance
s_Rs=mac_con(:,4).*system_base_mva./machine_base_mva;
%Armature resistance
s_xd=mac_con(:,5).*system_base_mva./machine_base_mva; %d axis
%synchronous reactance
s_xdd=mac_con(:,6).*system_base_mva./machine_base_mva; %d axis
%transient reactance
s_xddd=mac_con(:,7).*system_base_mva./machine_base_mva; %d
%axis sub-transient reactance
s_Tdod=mac_con(:,8); %d axis open circuit transient time
%constant
s_Tdodd=mac_con(:,9); %d axis open circuit sub-transient time
%constant
s_xq=mac_con(:,10).*system_base_mva./machine_base_mva; %q axis
%synchronous reactance
s_xqd=mac_con(:,11).*system_base_mva./machine_base_mva; %q
%axis transient reactance
s_xqdd=mac_con(:,12).*system_base_mva./machine_base_mva; %q
%axis sub-transient reactance
s_Tqod=mac_con(:,13); %q axis open circuit transient time
%constant
s_Tqodd=mac_con(:,14); %q axis open circuit sub-transient time
%constant
s_H= mac_con(:,15).*machine_base_mva./system_base_mva;
%Inertia constant
s_D=mac_con(:,16).*machine_base_mva./system_base_mva; %Self-
%damping
s_Rgov = s_Rgov.*system_base_mva./machine_base_mva;

ExpandSmachs= zeros(Nbus,NSmachs);
for counter=1:NSmachs
ExpandSmachs(Smachs(counter),counter)=1;
end

SelectSmachs= zeros(NSmachs,Nbus);
for counter=1:NSmachs
SelectSmachs(counter,Smachs(counter))=1;
end

s_Pg = bus_sol(Smachs,4);
%generator real power in machine base
s_Qg = bus_sol(Smachs,5);
%generator reactive power in machine base
```

Script 3.6 Synch_parameter_sys_base.m in system base.

Once all parameters are set for initialization of the synchronous machines in machine or system base, the machines can be initialized as shown in Script 3.7 (Singh and Pal, 2013).

```
%generic_Sync_Init.m
s_f=60; %System frequency in Hz
s_wb=2*pi*s_f; %Base value radial frequency in rad/sec
s_ws=1; %p.u. value of synchronous speed

Zg= s_Rs + 1i*s_xddd; %Zg for sub-transient model
Yg=1./Zg; %Yg for sub-transient model

s_volt = bus_sol(Smachs,2); %voltage magnitude generator bus
s_theta = bus_sol(Smachs,3)*pi/180; %voltage angle in rad/sec

s_Vg = s_volt.*(cos(s_theta) + 1i*sin(s_theta));
%generator terminal voltage in complex form in DQ frame
s_Ig = conj((s_Pg+1i*s_Qg)./s_Vg);
%current at generator terminal in complex form in DQ reference
s_Eq = s_Vg + (s_Rs+1i.*s_xq).*s_Ig;
%Effective internal voltage defined as shown
s_delta = angle(s_Eq);
%Generator rotor angle

s_id = -abs(s_Ig) .* (sin(s_delta - angle(s_Ig)));
s_iq = abs(s_Ig) .* cos(s_delta - angle(s_Ig));
s_vd = -abs(s_Vg) .* (sin(s_delta - angle(s_Vg)));
s_vq = abs(s_Vg) .* cos(s_delta - angle(s_Vg));
%converting quantities from DQ-frame to dq-frame(3.1)

s_Efd = abs(s_Eq) - (s_xd-s_xq).*s_id;
%Initial value for field voltage, to follow detailed
%derivation consult (Padiyar K. R., 2008)

s_Eqd = s_Efd + (s_xd - s_xdd).* s_id;
%Derived from (3.24) and (3.27)
s_Edd = -(s_xq-s_xqd) .* s_iq;
%Derived from (3.25) and (3.26)
s_Psi1d=s_Eqd+(s_xdd-s_xls).*s_id;
%Derived from(3.27)
s_Psi2q=-s_Edd+(s_xqd-s_xls).*s_iq;
%Derived from(3.26)

s_Te = s_Eqd.*s_iq.*(s_xddd-s_xls)./(s_xdd-s_xls)+
s_Psi1d.*s_iq.*(s_xdd-s_xddd)./(s_xdd-s_xls) +
s_Edd.*s_id.*(s_xqdd-s_xls)./(s_xqd-s_xls) -
s_Psi2q.*s_id.*(s_xqd-s_xqdd)./(s_xqd-s_xls)- s_iq.*s_id.*
(s_xqdd-s_xddd);
%This is (3.23)

s_Tm=s_Te;
%Derived from (3.22)
s_iQ = cos(s_delta).*s_iq - sin(s_delta).*s_id;
s_iD = sin(s_delta).*s_iq + cos(s_delta).*s_id;
%Conversion from dq-frame to DQ-frame (3.2)

s_Vref=s_volt+1./s_Ka.*s_Efd;
%Derived from (3.31)
```

Script 3.7 generic_Sync_Init.m.

If we execute *SMIB_run* with the command *Synch_parameter_gen_base*, all required variables appear in the workspace. For example, you should be able to see that s_Tm is 0.779, s_iQ is 0.765 and s_iD is 0.014.

If we run *SMIB_run* with the command *Synch_parameter_sys_base*, s_Tm is 7.013, s_iQ is 6.881 and s_iD is 0.125.

Can you find the relationship between s_Tm in system base and machine base and check if your results make sense?

Does the same relationship hold true for s_iQ and s_iD?

3.6 Simulink modelling

To run time domain simulations for this SMIB system, we need to create a Simulink model that represents the infinite bus and line and the synchronous machine. We start by creating a new Simulink file with two subsystems, as shown in Fig. 3.9. One subsystem represents the generator and the other subsystem the network. Any changes in current injection from the generator at Bus 1 affect the network voltage, and any changes in network voltage at Bus 1 affect the generator.

The network subsystem is a representation of the equation $V_1 = V_{inf} + I_1 * Z_{line}$ and is quickly built in Simulink, as shown in Fig. 3.10. Simply double click on the network subsystem to add the components.

The generator subsystem is more complex, so we will go step by step. The generator consists of the electrical and mechanical side. The mechanical side is represented by the torque angle loop, while the electrical side is represented in the subsystem named machine. The synchronous machine also

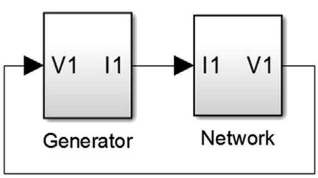

Figure 3.9 Single machine infinite bus system in Simulink.

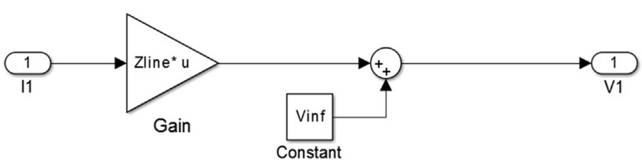

Figure 3.10 Infinite bus network in Simulink.

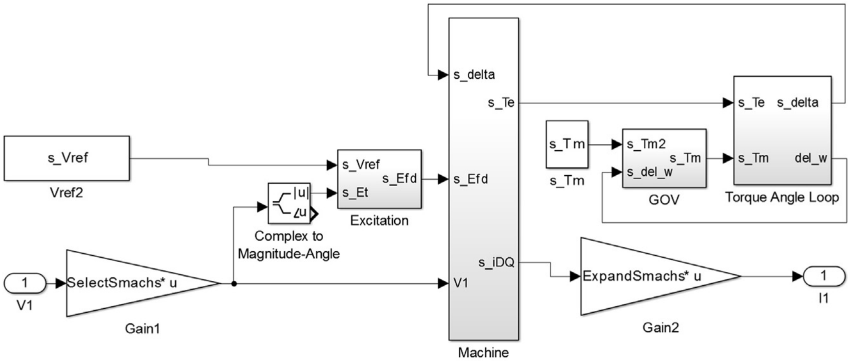

Figure 3.11 Generator representation in Simulink.

has a governor and excitation control. The gains at the voltage input and current output are required to select only the voltages of synchronous machine buses and to provide the synchronous machine currents for the synchronous machine buses. These subsystems can be seen in Fig. 3.11. Create these subsystems in your Simulink model, before we discuss what is inside of them.

The torque angle loop is built as shown in Fig. 3.12. It represents the generator mechanical dynamics given by Eqs. (3.21) and (3.22). This subsystem contains two integrator blocks, with the subtitle s_*wr* and s_*delta*. These need to be initialized. Double click on the block with subtitle s_*wr* and set the initial condition option to *s_ws*. A synchronous machine in steady state rotates at synchronous speed. For the second integrator with s_*delta*, set *s_delta* as the initial condition.

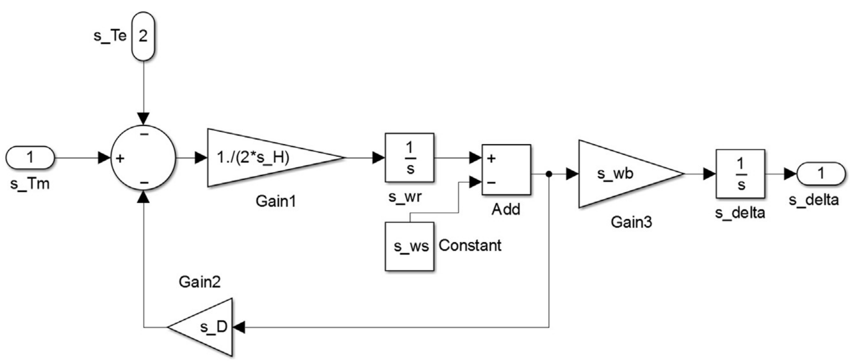

Figure 3.12 Torque angle loop.

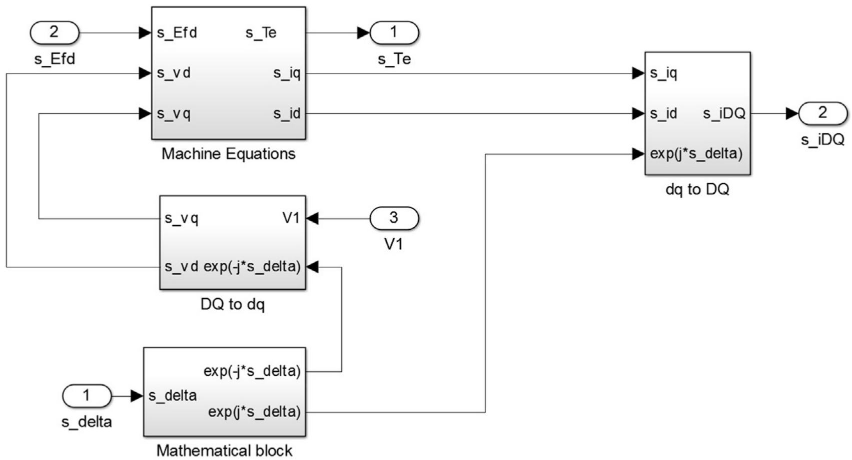

Figure 3.13 Electrical machine equations.

Fig. 3.13 shows the representation of the electrical side of the generator. We have talked about the dq-frame of generators and the DQ-frame of the network, why they are important and how to convert a quantity from one reference frame to the other. Now, we will see how to include this in our Simulink model. From the network block, the generator receives the voltage V_1 in DQ-frame, which is converted to dq-frame according to Eq. (3.1). The current that comes from the generator is in dq-frame and needs to be converted back to DQ-frame for the network according to Eq. (3.2). The machine Eqs. (3.23)–(3.29) describe what happens inside the synchronous machine.

The mathematical block creates the terms for the rotation of the phasor by $\pm\delta$. You can build it as shown in Fig. 3.14.

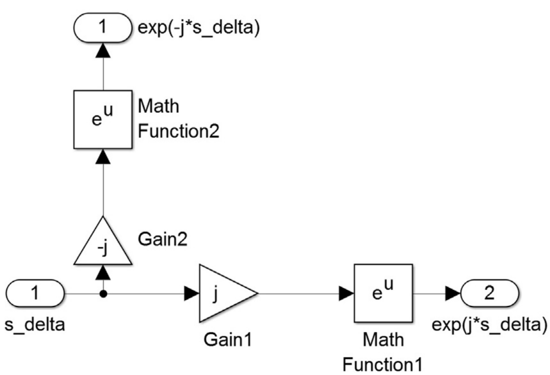

Figure 3.14 Mathematical block for exponentials.

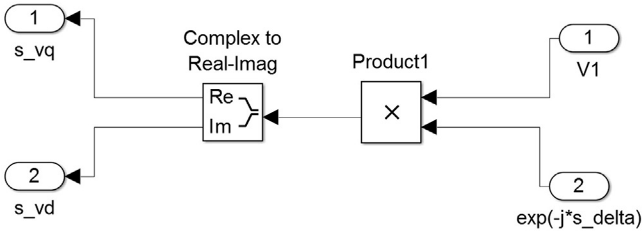

Figure 3.15 Converting from DQ- to dq-frame.

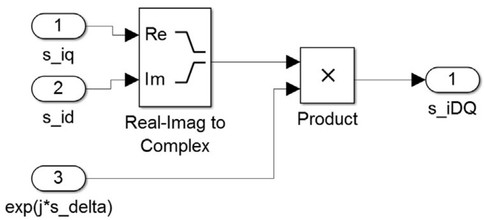

Figure 3.16 Conversion from dq- to DQ-frame.

Once the voltage is converted from DQ-frame to dq-frame, it is split into its real component (Vq) and imaginary component (Vd), as shown in Fig. 3.15.

The reverse process is used for the q- and d-current components of the generator, to convert the current to a DQ-frame for the network, as shown in Fig. 3.16.

As there are many machine equations, we use further subsystems to structure our Simulink model. This structure is shown in Fig. 3.17. Please build these subsystems before continuing to implement the individual equations.

Fig. 3.18 is a visual implementation of (3.24). Please confirm that the equation and the diagram are the same. Then implement the equation as shown here. Of course, the initial condition is s_Eqd here.

Fig. 3.19 is a visual implementation of (3.25). Implement the equation and compare the diagram with the equation. Make sure you set an appropriate initial condition here.

Can you see which equation Fig. 3.20 represents? Please ensure again to set an appropriate initial condition in the integrator block.

Please repeat the same procedure with Fig. 3.21.

Next, we need to calculate the electrical torque, as described in (3.23). The Simulink implementation of the equation can be seen in Fig. 3.22.

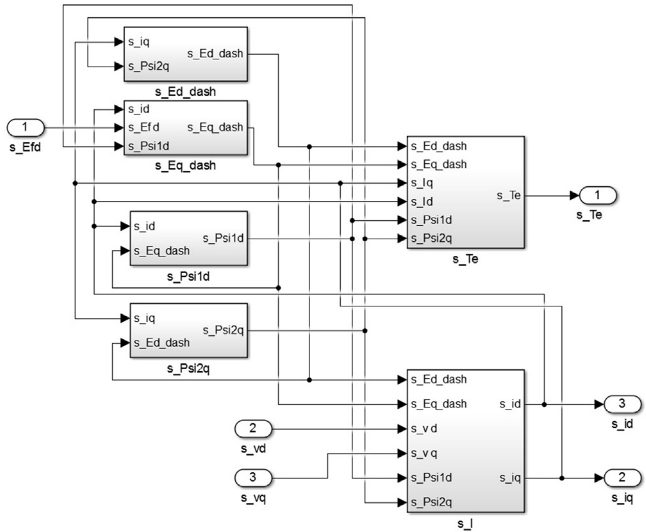

Figure 3.17 Machine equations.

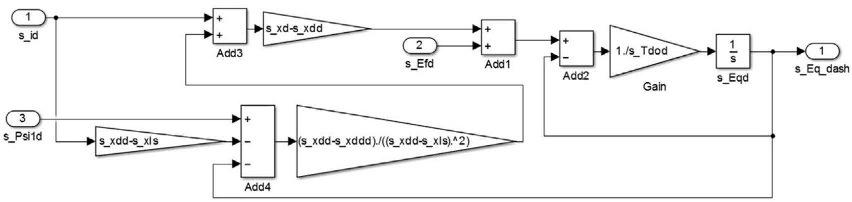

Figure 3.18 Transient EMF due to field flux linkage.

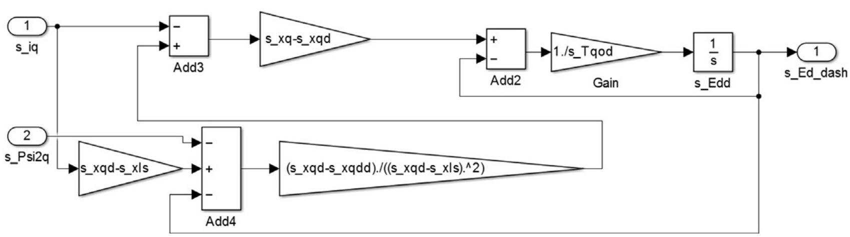

Figure 3.19 Transient EMF due to flux linkage in q-axis damper coil.

The final step is to implement Eqs. (3.28) and (3.29). As they take the same inputs, you will see in Fig. 3.23, that there is a lot of benefit implementing them as one subsystem.

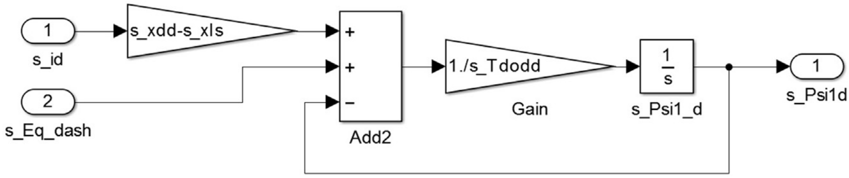

Figure 3.20 Subtransient EMF due to flux linage in d-axis damper.

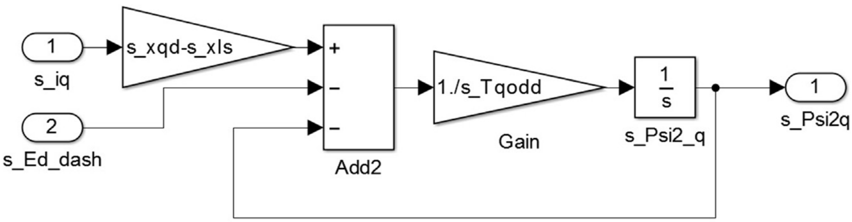

Figure 3.21 Subtransient EMF due to flux linkage in q-axis damper.

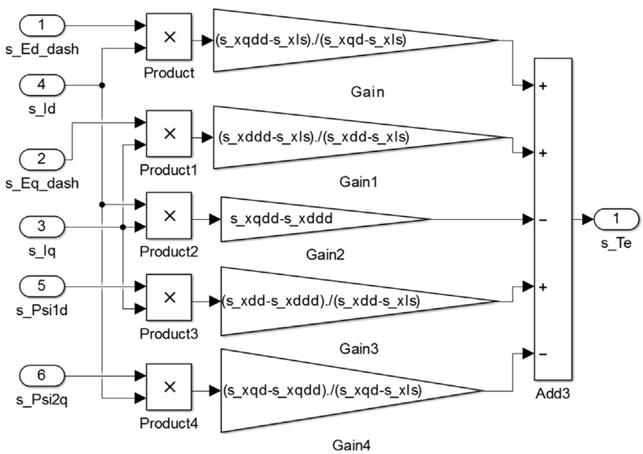

Figure 3.22 Electrical torque equation.

Can you see which equation Fig. 3.24 represents? Please ensure again to set *s_Efd./s_Ka* as initial condition in the integrator block.

Can you see which equation Fig. 3.25 represents? Please ensure again to set the correct initial condition, s_Tm.

Congratulations, you have finished building the synchronous machine model in system base. As we discussed for the initialization, you had a choice to model the machine in system base or machine base. Modelling the machine in machine base required a change of pu system at the network

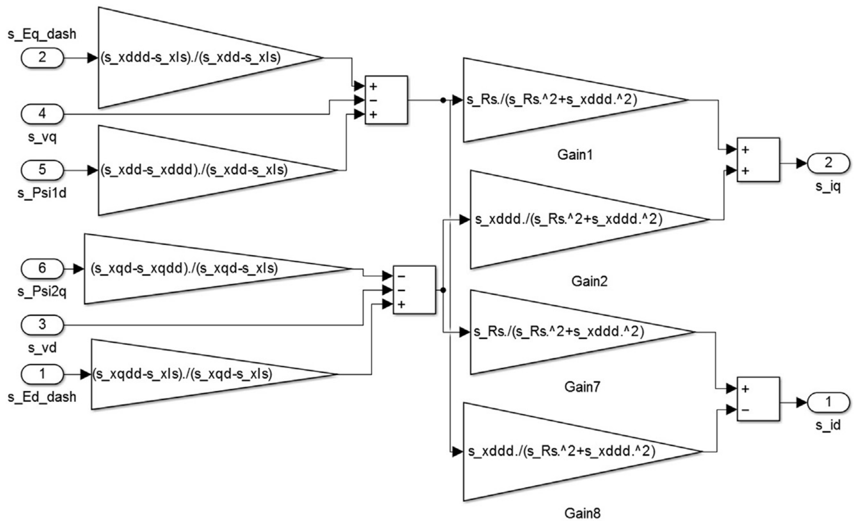

Figure 3.23 Equations for q- and d-axis stator current components.

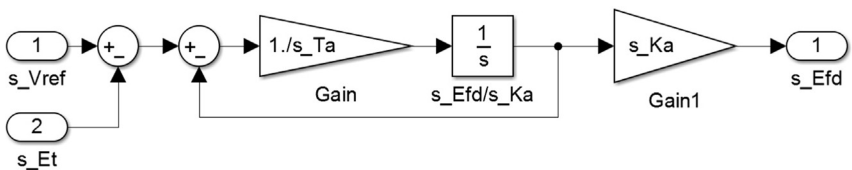

Figure 3.24 Equation for static excitation control.

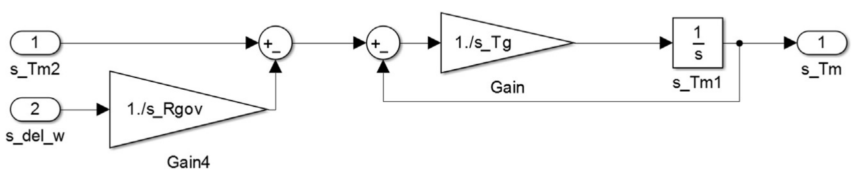

Figure 3.25 Equation for governor control.

interface. The only required change to the Simulink model is the Gain block seen in Fig. 3.26. This simulation will take the initialization code in machine base. As we can see from the display in Fig. 3.26, the electrical torque in machine base is 0.7792. Converting this to systems base, the torque is the same as in the previous simulation.

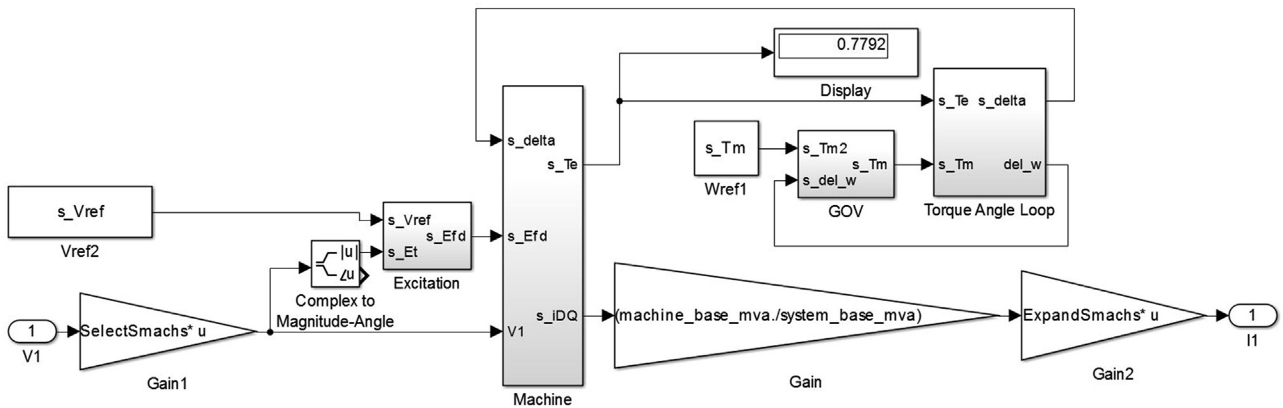

Figure 3.26 Alteration for simulation with generator in machine base.

3.7 Study case: single machine infinite bus test system time domain results

Now you have implemented the whole Simulink model, all that remains is to test if it works. Please make sure you have run the Matlab initialization and all required parameters are still in the workspace. Make sure the latest changes in your Simulink model are saved and then click run. You can connect a display as a sink on the signal s_Te and run again; if you did everything 100% correct, the value shown should be 7.013.

If you do not get this value, do not worry. You will be able to gain some skills in debugging. First, see if Simulink is giving you any useful warning and messages to point to a problem. If this does not make it clear what has gone wrong, split the problem into sections. You can temporarily disconnect all the submodules, testing them a module at a time. You can start from the top, as we created them here and work your way down to individual equations. What you have to do is simple; however, it can take a bit of time. Fig. 3.27 shows an example of this approach. All inputs are replaced with constant blocks, which have the correct initial value, according to your initialization program. You can then see for each output of the subsystem, if it produces the expected output value according to what you have calculated in the initialization program. If the result differs in a subsystem, this subsystem has a problem and warrants further investigation of the equations inside, following the same procedure of disconnecting the inputs and replacing them according with the constant values, found during initialization. Continue with this until you find the problem, fix it and then make sure everything is reconnected again.

With the working model, let us run a step-response simulation. From now on, let us only use the simulation in system base. Make sure it is initialized with the according m-file. Use a step change for the mechanical input torque and use the scope and multiplexer to see how the electrical torque changes with a change in mechanical torque. At 1 second, increase the governor load reference s_Tm2 by 1% and observe the result you get (Fig. 3.28).

Fig. 3.29 shows the step response you will see, where the step change is in the mechanical input torque and the oscillatory response is seen in the electrical torque. In steady state, the mechanical torque and electrical torque are the same, which can be seen in the pre- and postdisturbance behaviour. You can also observe the change in generator rotor angle during the step change; this is shown in Fig. 3.30. Use a second display for this and change

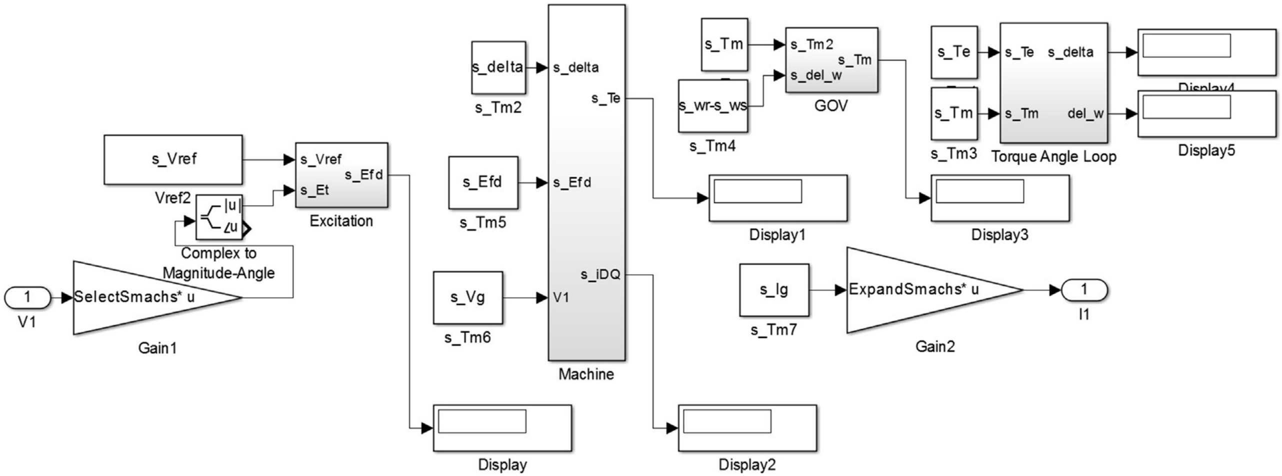

Figure 3.27 Procedure for debugging a Simulink model.

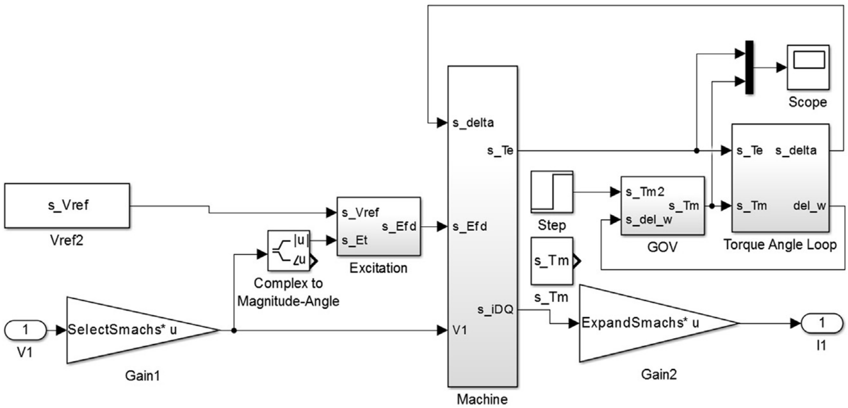

Figure 3.28 Step-response simulation.

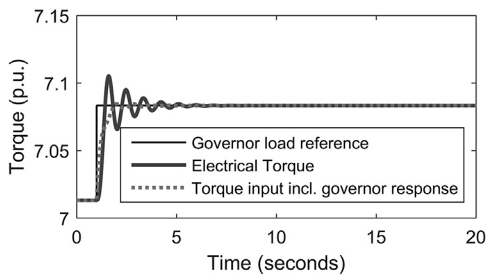

Figure 3.29 Synchronous machine step response to change in governor load reference.

the parameter setting for history, by unselecting the option limit data points to last 5000. You can then set the simulation run time to 60 s and run the simulation to see the result as shown here. Please convert the initial and final generator rotor angle into degrees.

3.8 Dynamic models of synchronous machines

The various dynamic models of synchronous machines one will encounter, working in power systems, vary in many respects. As mentioned earlier, different conventions for the dq-axis, the dq-transformation and the generator—load convention are in use.

Further differences can be found in the detail of modelling of the mechanical side. Models may range from a single-mass model to models that

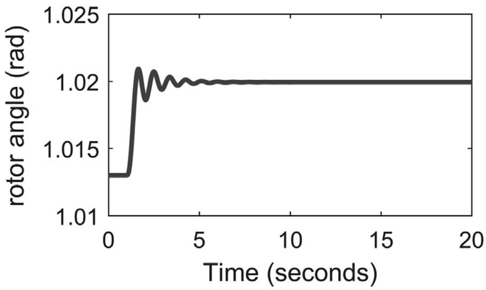

Figure 3.30 Change in generator rotor angle.

include several masses, which are coupled by springs and dampers. An example of a more detailed mechanical representation is shown in Fig. 3.31, where the mechanical side of the synchronous machine is represented by four sections, the high pressure, intermediate pressure, low pressure turbine and the generator. The diagram depicts the self-damping, mutual damping and spring constant associated with the system and the torques at various stages of the synchronous machine. The detailed mechanical model is only required for studies, where the focus is on possible interactions with the mechanical side, such as subsynchronous resonance phenomena. A detailed description of the treatment of such phenomena can be found in (Padiyar, 1999).

The representation of synchronous machines further varies according to the number of equivalent damper coils on the d- and q-axis. The number of damper coils used on each axis depends on whether the model is used to correctly capture the synchronous, transient or subtransient behaviour. Model 1.0 only has a field coil on the d-axis and no equivalent damper coils. Model 1.1 is the same as Model 1.0 with an additional equivalent damper coil on the q-axis; Model 2.1 has a further equivalent damper coil on the

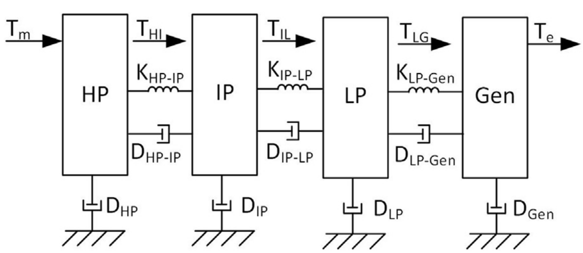

Figure 3.31 Four-mass system.

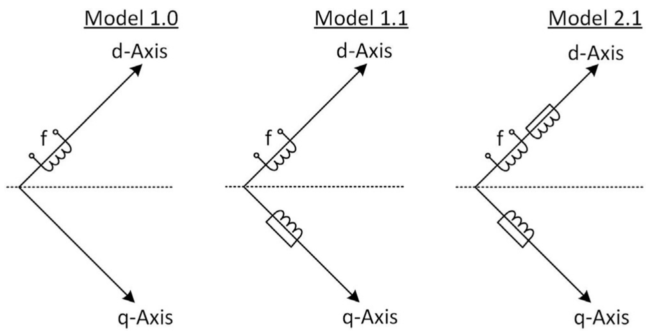

Figure 3.32 Representation of synchronous machine with varying number of equivalent damper coils.

d-axis. Model 2.2 is the one used in this book and is shown in Fig. 3.5. Padiyar, 2008 provides a detailed description of Model 1.0, 1.1, 2.1 and 2.2, which can be seen in Fig. 3.32.

We need to understand that models with a lower number of damper windings, such as Model 1.1, are only a special case of Model 2.2. For this, let us set the machine parameters as follows:

```
mac_con=[1 900 0.2 0.0025 1.8 0.3 0.25 8 0.03 1.7 0.3 0.25 0.4 0.05 6.5 0];
%Synchronous machine parameters
```

In a salient pole machine, the reactance of the quadrature axis differs from the reactance of the direct axis. Saliency can be seen in the steady-state, transient and subtransient model, where the difference is seen in the respective reactances. The modelling of saliency is described in detail by Padiyar, 2008. The model we are using does not cover saliency. By setting the parameters as shown above, we ensure that Model 1.1 has the same value as Model 2.2. Hence, there will be no saliency in either Model 1.1 or Model 2.2 in this example.

We can now run the Simulink model with the altered parameters and record the electrical torque, during a step change in mechanical torque.

Then we alter the subtransient reactances to the value of the transient reactances and rerun the simulation, with mac_con as shown below. As transient and subtransient reactances are the same, many terms of the dynamic equations drop out. This leads to Model 1.1, which has less damper windings with only synchronous and transient reactances. Please go through the generator equations and see how Model 1.1 is the special case of Model 2.2, where the transient and subtransient reactances are the same. Write out the equations of Model 1.1 by using the equations of Model 2.2 and setting X_d'' to equal X_d' and X_q'' to equal X_q'. We record the electrical torque again, during the same step change in mechanical torque as before.

```
mac_con=[1 900 0.2 0.0025 1.8 0.3 0.3 8 0.03 1.7 0.3 0.3 0.4 0.05 6.5 0];
%Synchronous machine parameters
```

Fig. 3.33 shows the result we get, if we plot the two saved electrical torque curved on top of one another. Both simulation results are identical during the steady state; however, Model 2.2 shows larger oscillations during the subtransient period, as the model includes the subtransient behaviour.

This result shows the difference between a transient model, such as Model 1.1, and a subtransient model, such as Model 2.2. We have also seen that Model 1.1 is only a special case of Model 2.2, rather than being a completely different model.

Synchronous machine models may also include magnetic saturation, experienced in actual generators. The nonlinearity introduced through saturation can be modelled by adjusting the mutual inductances for a saturation factor. A detailed description of this can be found in Kundur, 1994.

As can be seen from (3.28) and (3.29), the current components I_q and I_d are determined through a relationship of voltages V_q and V_d and the state variables of the synchronous machine. Eqs. (3.23–3.27) can be rewritten, in a more direct form, without the use of current terms I_q and I_d. The mixed current flux notation is used for ease of writing the equations. Further differences in synchronous machine models can be found, due to the timescale of the dynamics of interest. As explained in Chapter 1, the transients of a short circuit current can be split into a subtransient, transient and steady-state period. A model that aims to accurately capture the behaviour during

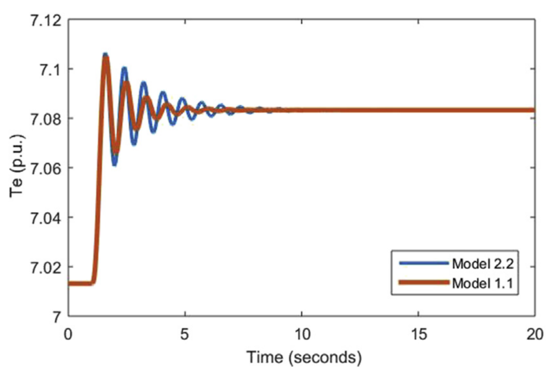

Figure 3.33 Comparison of the electrical torque Te during 1% step-up in mechanical torque using Model 2.2 and Model 1.1.

the subtransient period has to be more detailed than a model that captures the transient or steady-state period.

3.9 Simulation model of the two-area test system

The two-area test system shown in Figure 2.5 has four synchronous generators (G1, G2, G3 and G4) and two loads (at Bus-7 and Bus-9). The bus matrix and line matrix of the system are shown in Table 2.5 and Table 2.6, respectively. The synchronous machine parameters are specified using mac_con matrix given in Script 3.8. The script also lists the gain and

```
% COPY BUS DATA FROM TABLE 2.5
% COPY LINE DATA FROM TABLE 2.6

% ********************** MACHINE DATA STARTS **********************

% Machine data format
%       1. machine number,
%       2. bus number,
%       3. base mva,
%       4. leakage reactance x_l(pu),
%       5. resistance r_a(pu),
%       6. d-axis sychronous reactance x_d(pu),
%       7. d-axis transient reactance x'_d(pu),
%       8. d-axis subtransient reactance x"_d(pu),
%       9. d-axis open-circuit time constant T'_do(sec),
%      10. d-axis open-circuit subtransient time constant
%                T"_do(sec),
%      11. q-axis sychronous reactance x_q(pu),
%      12. q-axis transient reactance x'_q(pu),
%      13. q-axis subtransient reactance x"_q(pu),
%      14. q-axis open-circuit time constant T'_qo(sec),
%      15. q-axis open circuit subtransient time constant
%                T"_qo(sec),
%      16. inertia constant H(sec),
%      17. damping coefficient d_o(pu),
%      18. dampling coefficient d_1(pu),
%      19. bus number
%      20. saturation factor S(1.0)
%      21. saturation factor S(1.2)
% note: all the following machines use subtransient reactance model

mac_con =[
1 900 0.2 0.0025 1.8 0.3 0.25 8 0.03 1.7 0.55 0.25 0.4 0.05 6.5 0;
 2 900 0.2 0.0025 1.8 0.3 0.25 8 0.03 1.7 0.55 0.25 0.4 0.05 6.5 0;
 3 900 0.2 0.0025 1.8 0.3 0.25 8 0.03 1.7 0.55 0.25 0.4 0.05 6.5 0;
 4 900 0.2 0.0025 1.8 0.3 0.25 8 0.03 1.7 0.55 0.25 0.4 0.05 6.5 0];
% ********************** MACHINE DATA ENDS **********************

% ********************** EXCITATION SYSTEM DATA *****************
s_Ka=200;%Static excitation gain Padiyar p.328
s_Ta=0.02;%Static excitation time constant Padiyar p.328

% **** Governor Control SYSTEM DATA - not used *********
s_Tg=0.2;%Kundur p.598
s_Rgov=0.05;%Kundur p.598
```

Script 3.8 Two-area test system data.

```
% Save this program as find_impedance_matrix.m
% Copy script from Script 2.5 here

% Finding post-fault impedance matrix.
% Define post-fault admittance matrix
Yf = Y;
% Apply three phase fault at bus-8 by specifying very large
%admittance at Yf(8,8)
Yf(8,8) = 10000;
% Find post-fault impedance matrix.
Zf = inv(Yf);
```

Script 3.9 Program to find prefault and postfault impedance.

time constant of the static excitation system. Combine Table 2.5 and Table 2.6 as mentioned in Script 3.8, and save it as four_mac_data.m.

We will reuse programs and models developed so far to build a simulation model for the two-area test system. Please ensure the functions form_Ymatrix(bs,ln) and power_flow(Y, bs, ln) given in Section 2.6 are saved as form_Ymatrix.m and power_flow.m, respectively, in the current Matlab working folder. Also save Synch_parameter_sys_base.m and generic_Sync_Init.m in the current working folder.

In this chapter, we will simulate a three-phase ground fault at Bus-8 of the two-area test system. To simulate a three-phase ground fault, prefault and postfault impedance matrices are required. A three-phase ground fault at Bus-8 means a zero impedance from the bus to ground at the bus. Alternatively, we can add a large admittance at the diagonal entry corresponding to the bus in the admittance matrix and find the inverse of the new admittance matrix to get the postfault impedance matrix. Create program find_impedance_matrix.m using Script 3.9.

3.9.1 Simulink block representing multiple synchronous machines

In Section 3.5, we developed Simulink block for a synchronous machine. But the two-area test system has four synchronous machines in it. The synchronous machine Simulink block could be directly used to simulate any number of machines as long as the program is made to handle vector calculation. For example, the equation $A = B + CD$ can be represented in Matlab using `A = B + C*D` or `A = B + C.*D`. Both representations are correct. But the second one can be used for both scalar and vector calculation, as the '.' (dot) symbol before the * specifies dot product of two vectors. Hence, if second command is used, the size of vector A can be one or many. All figures in Section 3.5 show vector calculations. Readers encouraged to double check their model, as this is one of the common mistakes in early programming of multimachine system. For results presented in this section and also in Chapter 9, it is assumed that governors are not active in synchronous machines. Disconnect Governor model by connecting s_Tm constant block in Fig. 3.28 to s_Tm input of the torque angle loop block. By doing this, mechanical input torque to the generators will be constant during the course of the simulation. The change is made for ease of illustration of interarea oscillation mode, and the reader is encouraged to study the effect of governor on dynamics of the system.

Another important aspect is ensuring that correct number of inputs is passed to each block. Fig. 3.34 shows the Simulink program for the two-area system, where SelectSmachs and ExpandSmachs gains are shown outside the Generator block for clarity. The Generator block accepts a vector of generator bus voltages as input. Size of this vector must equal the number of generators present in the system. So correct bus voltages must be selected from a vector of all bus voltage outputs from the network block. There are different ways to do this. In Fig. 3.34, a 4x11 matrix, 'SelectSmachs', multiplies the bus voltage vector to obtain 4x1 generator bus voltages. Similarly, the network block requires a vector of bus current injections at all buses. In our example, the Generator block produces a current vector of size 4 corresponding to number of synchronous machines. The 'ExpandSmachs' block transforms this vector to a new vector of size 11 equal to the number of buses in the system. In the new vector, the output currents of synchronous generators are placed at the position corresponding to the synchronous machine buses and all other elements are set to zero. Script for both the

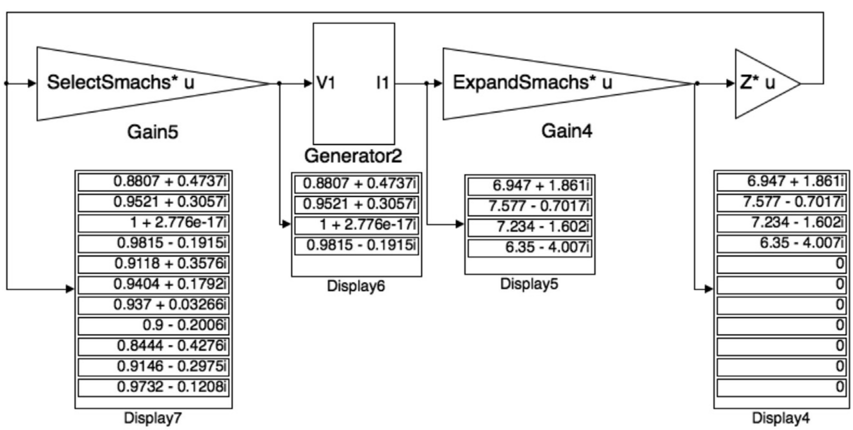

Figure 3.34 Two-area test system model.

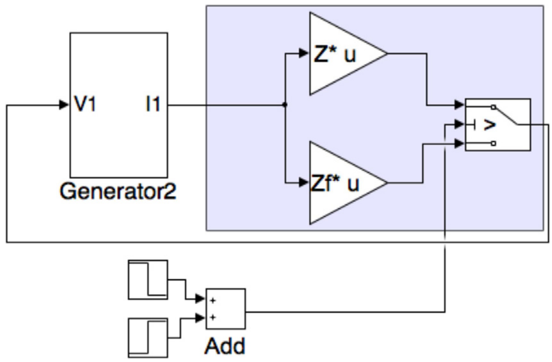

Figure 3.35 Dynamic simulation model of the two-area test system model to simulate three-phase ground fault.

matrices is given in Script 3.6. Now, build a Simulink model as shown in Fig. 3.35 using the synchronous machine block developed in Section 3.5.

Few more lines of code are required before we can run the model. Create two_area_synch.m using the code and instruction in Script 3.10. Build a Simulink model, two_area_synch_model.slx as shown in Fig. 3.35. The model simulates a three-phase ground fault at Bus-8 from

```
clear all

system_base_mva = 100.0;

% Copy the code in the Script 3.8

% Obtain load flow solution
[Y] = form_Ymatrix(bus,line);
% calculate pre-fault power flow solution
[bus_sol, line_flow] = power_flow(Y,bus, line);

% Synchronous machine initialisation

Smachs=[1;2;3;4];% buses with synchronous machines
Nbus=size(bus,1); % Number of buses
NSmachs=size(Smachs,1); % Number of synchronous machines


% program to initialise the synchronous machines
Synch_parameter_sys_base_fourmach
generic_Sync_Init

% Copy the code in the Script 3.9
```

Script 3.10 Program to initialize two-area test system model.

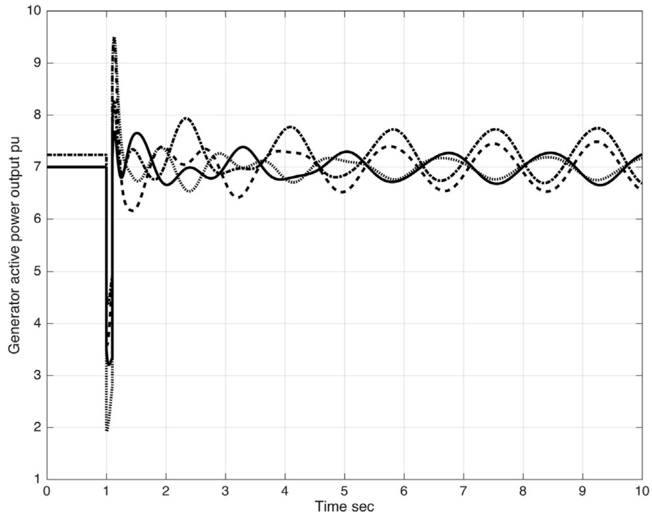

Figure 3.36 Synchronous machine active power output.

1 s to 1.1 s. This is achieved using the two gains having impedance matrices (*Z* and *Zf*) and a switch. The threshold parameter of the switch is set to 0.5. The switch selects the path of $Z*u$ (first input) when the second input is less than 0.5 (during normal operating condition) and changes to $Zf*\ u$ when second input is more than 0.5 (between time 1 and 1.1 s). Two step signals and an add block create the second input to the switch. One of the step signals changes from 1 to 0 at time 1.0 s and another step signal changes from 0 to 1 at time 1.1 s.

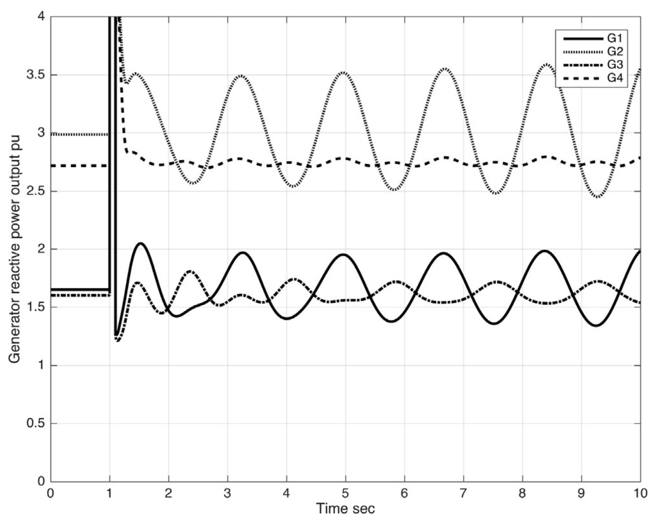

Figure 3.37 Synchronous machine reactive power output.

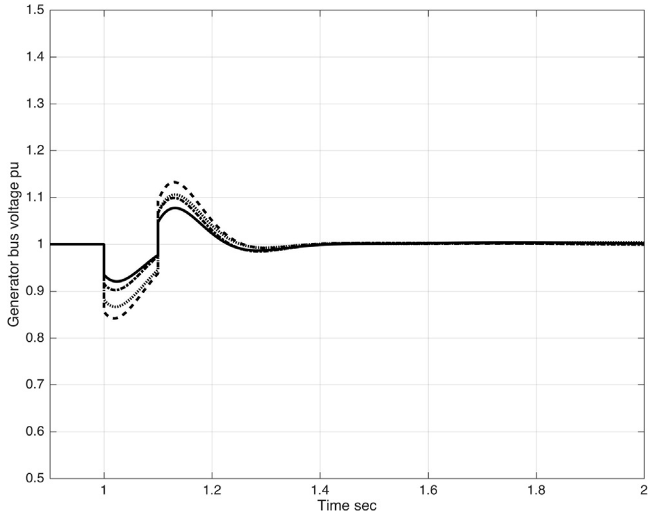

Figure 3.38 Synchronous machine bus voltages.

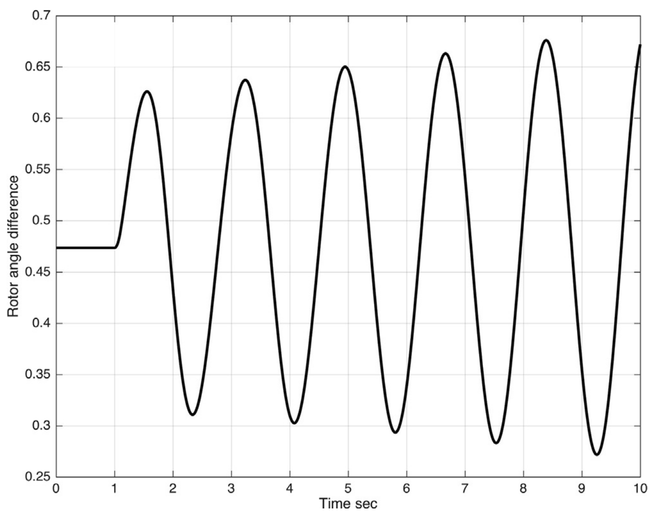

Figure 3.39 Rotor angle difference between the generator G1 and generator G3.

Now you are ready to test your model. Use sample results in Figs. 3.36–3.39 to validate your results. The plots show generator active power outputs, reactive power outputs, terminal voltages and angle difference between G1 and G3.

References

Clarke, E., 1957. Circuit Analysis of A-C Power Systems. John Wiley & Sons, Inc, New York.

IEEE committee report, 1969. Recommended phasor diagram for synchronous machines. IEEE Trans. Power Apparatus Syst. 1593–1610.

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

Padiyar, K., 1999. Analysis of Subsynchronous Resonance in Power Systems. Kluwer Academic Publishers, Norwell.

Padiyar, K.R., 2008. Power System Dynamics Stability and Control, second ed. BS Publications, Hyderabad.

Pal, B., Chaudhuri, B., 2010. Robust Control in Power Systems. Springer, New York.

Singh, A.K., Pal, B.C., 2013. IEEE PES Task Force on Benchmark Systems for Stability Controls, Report on the 68-Bus, 16-Machine, 5-Area System. Version 3.3.

Neidhofer, G., October 1992. The evolution of the synchronous machine. Engineering Science and Education Journal 239–248.