Chapter 1
Numerical simulation of the electrical power system model

In this chapter, we will explain numerical simulation methods for power system models described by nonlinear differential algebraic equation systems. We will also provide guidelines for building a structured numerical simulation environment.

The chapter is organized as follows. First, in Section 1, we will explain the difficulty of calculating the time response of power system models. Next, in section 2, we will describe the process of power flow calculation, which numerically explores the equilibrium state of power system models. In section 3, we will discuss how to determine the steady-state values of internal states of generators and constants of load models that are consistent with the steady-state values of bus voltages and power determined by power flow calculation.

In section 4, we will explain the calculation methods for time responses to changes in initial conditions, load fluctuations, and ground faults, and illustrate the calculation of time responses. Finally, in section 5, we will discuss synchronism phenomena of bus voltages in steady-state power flow states from an advanced perspective, and analyze them mathematically.

1 Calculation of time response of an electrical power system

1.1 Challenges in calculating time response

In Chapter **??**, we explained that the mathematical model of the entire power system, including generator and load models coupled with transmission line models, is described by a system of nonlineadr differential algebraic equations. Therefore, the time response of the power system model can be obtained by numerically integrating these equations with appropriate initial values and external inputs. However, when performing numerical simulations of the power system model, it is necessary to consider specific characteristics of the power system, such as:

- If external inputs are not properly set, the demand and supply will not be in equilibrium, and the steady-state value of the frequency deviation will not be zero, causing the rotor angle of the generator to continue to change.
- There are an infinite number of combinations of external input values that result in zero steady-state value for frequency deviation. Therefore, it is necessary to specify realistic external input values.
- The values of the voltage phase and current phase of the bus bar group must be determined so that they are consistent as dependent variables with respect to the state variables of the generator group.

Therefore, simply using the differential-algebraic equation solver that is standardly implemented in MATLAB is not sufficient to accurately execute numerical simulations of the power system. This is one of the factors that makes the calculation of the time response of the power system model difficult.

1.2 Calculation steps

The standard calculation procedure for the time response of a power system model described by nonlinear differential algebraic equations is divided into the following three steps:

(A) To specify the power system state in a steady state where supply and demand are balanced, calculate the values of the phase angle and voltage of all buses in the steady state using the admittance matrix determined from the transmission network.
(B) Calculate the steady-state values of the internal voltage and rotor angle of each generator, the external input values to the generator, and the impedance value of each load, so that they are consistent with the steady-state values of the bus current phase and voltage phase determined in Step A.
(C) Using the power system state where supply and demand are balanced calculated in Steps A and B as the initial value, calculate the time response under various disturbances of different magnitudes such as giving perturbations to the internal state of the generator, grounding the voltage of the bus, or changing the parameter values of the load.

From the viewpoint of system control engineering, Step A can be understood as ”determining one equilibrium point to perform numerical analysis from among an infinite number of equilibrium points.” As described in Section 3, when the steady-state values of the phase angle and voltage of all buses are given, there must exist steady-state values of the internal state of the generator, external input values to the generator, and the load parameter values that realize them. Therefore, calculating the steady-state values of the phase angle and voltage of all buses is equivalent to calculating the equilibrium point of the power system model represented by differential algebraic equations. In power system engineering, this process is called the **power flow calculation**.

It should be noted that calculating the steady-state values of the internal state of the generator, external input values, and the load parameter values in Step B is an indirect procedure of determining the mathematical model of the load from the results of the power flow calculation in Step A. For example, if you want to set the load connected to a certain bus as a constant impedance model, you need to reverse calculate the impedance value of the load using the current phase of the bus calculated by the power flow calculation divided by the voltage phase. If setting load model parameters as desired values, in Step C, the time response to load parameter fluctuations is calculated while changing the load parameters to desired values.

Finally, in Step C, the time response of the electrical power system model is calculated under various conditions depending on the purpose. For example, if the external input to each generator and load parameter calculated in Steps A and B are set as constants in the model, and the time response is calculated with the initial value appropriate for generators, the internal state of the generators asymptotically converges to the steady state calculated in Step B over time. However, to set valid initial values such that asymptotic convergence is established, as in Steps A and B, the equilibrium point that serves as the reference for the analysis must be calculated first. The calculated equilibrium point must be stable in an appropriate sense.

2 Numerical analysis of steady-state in power flow calculation

In this Section, we describe an overview of power flow calculation to numerically search the steady-state of the power system, as explained in Step A of Section 1.2 and the implementation method using MATLAB.

Given the admittance matrix $\boldsymbol{Y}$, the distribution of current and voltage phasors of the busbar group that satisfies the following relationship is referred to as the **power flow distribution** at time t.

$$\begin{bmatrix} \boldsymbol{I}_1(t) \\ \vdots \\ \boldsymbol{I}_N(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{Y}_{11} & \cdots & \boldsymbol{Y}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{N1} & \cdots & \boldsymbol{Y}_{NN} \end{bmatrix}}_{\boldsymbol{Y}} \begin{bmatrix} \boldsymbol{V}_1(t) \\ \vdots \\ \boldsymbol{V}_N(t) \end{bmatrix} \tag{1}$$

$$\left(|\boldsymbol{I}_1(t)|, \angle\boldsymbol{I}_1(t), |\boldsymbol{V}_1(t)|, \angle\boldsymbol{V}_1(t), \ldots |\boldsymbol{I}_N(t)|, \angle\boldsymbol{I}_N(t), |\boldsymbol{V}_N(t)|, \angle\boldsymbol{V}_N(t)\right) \tag{2}$$

Each current and voltage phasors changes with time, and from the laws of physics of current and voltage, these must satisfy Equation 1 at any arbitrary time t.

Power flow calculation is a computational process to find ”one of the steady-state power flow distributions”. A steady-sate power flow distribution refers to the

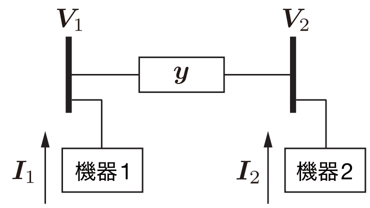


Fig. 1 Power system model consisting of two bus bars

condition that all the buses satisfy the following for given constant current phasors $\boldsymbol{I}_i^\star$ and voltage phasors $\boldsymbol{V}_i^\star$.

$$\boldsymbol{I}_i(t) = \boldsymbol{I}_i^\star, \qquad \boldsymbol{V}_i(t) = \boldsymbol{V}_i^\star, \qquad \forall t \geq 0$$

Using the definitions of active power and reactive power supplied to the bus bars in the following equation, the current phasors can be eliminated.

$$P_i(t) + \boldsymbol{j}Q_i(t) = \boldsymbol{V}_i(t)\overline{\boldsymbol{I}}_i(t) \tag{3}$$

Thus, the system of equations in Equation 1 is equilvalent to:

$$\begin{aligned} P_1(t) + \boldsymbol{j}Q_1(t) &= \sum_{j=1}^{N} \overline{\boldsymbol{Y}}_{1j} |\boldsymbol{V}_1(t)||\boldsymbol{V}_j(t)| e^{\boldsymbol{j}\left(\angle \boldsymbol{V}_1(t) - \angle \boldsymbol{V}_j(t)\right)} \\ &\vdots \\ P_N(t) + \boldsymbol{j}Q_N(t) &= \sum_{j=1}^{N} \overline{\boldsymbol{Y}}_{Nj} |\boldsymbol{V}_N(t)||\boldsymbol{V}_j(t)| e^{\boldsymbol{j}\left(\angle \boldsymbol{V}_N(t) - \angle \boldsymbol{V}_j(t)\right)} \end{aligned} \tag{4}$$

Therefore, depending on the context, the distribution of active power, reactive power, and voltage phasors satisfying Equation 4 is referred to as the power flow distribution at time t.

$$\left(P_1(t), Q_1(t), |\boldsymbol{V}_1(t)|, \angle \boldsymbol{V}_1(t), \ldots, P_N(t), Q_N(t), |\boldsymbol{V}_N(t)|, \angle \boldsymbol{V}_N(t)\right) \tag{5}$$

2.1 Outline of the power flow calculation

Let us explain the characteristics of the power flow calculation using a simple example consisting of two bus bars.

Example 1.1 Power flow calculation of a two bus bar power system model

Let us consider an electrical power system consisting of two bus bars, as shown in Figure 1. We assume that each bus is connected to either a load or a generator, but in power flow calculations, it is not necessary to specify the type of device connected.

The admittance of a transmission line that connects two bus bars is denoted by $\boldsymbol{y} \in \mathbb{C}$. When using the basic transmission line model, the voltage and current phasors at the buses are related by:

$$\begin{bmatrix} \boldsymbol{I}_1^\star \\ \boldsymbol{I}_2^\star \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} & -\boldsymbol{y} \\ -\boldsymbol{y} & \boldsymbol{y} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1^\star \\ \boldsymbol{V}_2^\star \end{bmatrix} \tag{6}$$

However, to indicate that these values are in a steady-state, they are markeed with a "$\star$". Here, by eliminating the current phasors using the relationship in Equation 3, we obtain a system of equations equivalent to Equation 6 in terms of steady-state active power, reactive power, and voltage phasors as follows:

$$\begin{aligned} P_1^\star + \boldsymbol{j}Q_1^\star &= \overline{\boldsymbol{y}} \left(|\boldsymbol{V}_1^\star|^2 - |\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| e^{\boldsymbol{j}(\angle \boldsymbol{V}_1^\star - \angle \boldsymbol{V}_2^\star)} \right) \\ P_2^\star + \boldsymbol{j}Q_2^\star &= \overline{\boldsymbol{y}} \left(|\boldsymbol{V}_2^\star|^2 - |\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| e^{\boldsymbol{j}(\angle \boldsymbol{V}_2^\star - \angle \boldsymbol{V}_1^\star)} \right) \end{aligned} \tag{7a}$$

The objective of the power flow calculation is to determine the following set of values that satisfy the above simultaneous equations.

$$(P_1^\star, Q_1^\star, |\boldsymbol{V}_1^\star|, \angle \boldsymbol{V}_1^\star, P_2^\star, Q_2^\star, |\boldsymbol{V}_2^\star|, \angle \boldsymbol{V}_2^\star)$$

We denote the conductance and susceptance of a transmission line as:

$$g := \mathrm{Re}[\boldsymbol{y}], \qquad b := \mathrm{Im}[\boldsymbol{y}]$$

Then, considering the real and imaginary parts of Equation 7a, we obtain a system of four equations:

$$\begin{aligned} P_1^\star &= g|\boldsymbol{V}_1^\star|^2 - g|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \cos \angle \boldsymbol{V}_{12}^\star - b|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \sin \angle \boldsymbol{V}_{12}^\star \\ P_2^\star &= g|\boldsymbol{V}_2^\star|^2 - g|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \cos \angle \boldsymbol{V}_{21}^\star - b|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \sin \angle \boldsymbol{V}_{21}^\star \\ Q_1^\star &= -b|\boldsymbol{V}_1^\star|^2 + b|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \cos \angle \boldsymbol{V}_{12}^\star - g|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \sin \angle \boldsymbol{V}_{12}^\star \\ Q_2 &= -b|\boldsymbol{V}_2^\star|^2 + b|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \cos \angle \boldsymbol{V}_{21}^\star - g|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \sin \angle \boldsymbol{V}_{21}^\star \end{aligned} \tag{7b}$$

where $\angle \boldsymbol{V}_{ij}^\star$ represents $\angle \boldsymbol{V}_i^\star - \angle \boldsymbol{V}_j^\star$.

The phase of the voltage phasor is only meaningful in terms of the difference value; thus, the number of variables that must be practically determined is seven. Therefore, there are three variables of freedom in Equation 7b. To determine these, one can choose appropriate values for the three variables $(|\boldsymbol{V}_1^\star|, |\boldsymbol{V}_2^\star|, \angle \boldsymbol{V}_{12}^\star)$. This allows us to calculate the remaining variables $(P_1^\star, P_2^\star, Q_1^\star, Q_2^\star)$

However, with this method, while the voltage phase of each bus can be set to any value, it is not possible to set the power supplied or consumed by each bus to any value. That is, it is not possible to specify which bus is a load bus that consumes power, or which bus is a generator bus that supplies power. Therefore, in order to perform numerical simulations with realistic settings, it is often necessary to properly determine the voltage phase values to achieve the specified active and reactive power values.

For example, let us consider finding the $(|\boldsymbol{V}_1^\star|, |\boldsymbol{V}_2^\star|, \angle\boldsymbol{V}_{12}^\star)$ that realize balanced active power supplied to the bus as:

$$P_1^\star = 1, \qquad P_2^\star = -1 \tag{8}$$

This corresponds to determining the voltage phase distribution when the device connected to bus 1 and bus 2 both supply and consume active power, respectively, with both having a value of 1 in a steady-state condition. By adding the equations in Equation 7b with respect to $P_1^\star$ and $P_2^\star$, we obtain:

$$\begin{aligned} 0 &= g\left\{|\boldsymbol{V}_1^\star|^2 + |\boldsymbol{V}_2^\star|^2 - 2|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \cos\angle\boldsymbol{V}_{12}^\star\right\} \\ &= g\left\{\left(|\boldsymbol{V}_1^\star| - |\boldsymbol{V}_2^\star|\right)^2 + 2|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star|\left(1 - \cos\angle\boldsymbol{V}_{12}^\star\right)\right\} \end{aligned}$$

Note that in realistic power flow conditions, the phase difference in voltage phase between the buses, $\angle\boldsymbol{V}_{12}^\star$, is within the range of $\pm\frac{\pi}{2}$. Moreover, it should be noted that if the phase angle difference of the generator or the voltage phase difference of the bus exceeds the range of $\pm\frac{\pi}{2}$, the slope of the sine function will be inverted, resulting in an unrealistic transmission characteristic of active power.

From this, if the conductance g of the transmission line, which is the real part of $\boldsymbol{y}$, is not equal to zero, then the voltage phase that satisfies this equation must necessarily satisfy:

$$|\boldsymbol{V}_1^\star| = |\boldsymbol{V}_2^\star|, \qquad \angle\boldsymbol{V}_{12}^\star = 0 \tag{9}$$

However, Equation 9 implies that both $P_1^\star$ and $P_2^\star$ are equal to 0. Therefore, it can be concluded that a steady-state current state satisfying Equation 8 does not exist as long as g is not equal to 0. This is because the conductance component (resistance component) of the transmission line causes power loss, and in the setting of Equation 8, it indicates that the power supply and demand are not balanced across the entire system. Therefore, it is important to note that for certain values of active and reactive power, there might not exist solutions to Equation 7.

Next, let us assume that the conductance g of the transmission line is zero for simplicity. Additionally, assume that the ground capacitance is sufficiently small and the susceptance b is negative. Then:

$$P_1^\star = -b|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \sin\angle\boldsymbol{V}_{12}^\star, \qquad P_2^\star = b|\boldsymbol{V}_1^\star||\boldsymbol{V}_2^\star| \sin\angle\boldsymbol{V}_{12}^\star$$

In this case, the voltage phasor distribution must be such that $P_1^\star = -P_2^\star$.

For example, in the case where the value of Equation 8 is specified, the absolute value of the voltage phasor can be specified as:

$$|\boldsymbol{V}_1^\star| = \sqrt{\tfrac{2}{|b|}}, \qquad |\boldsymbol{V}_2^\star| = \sqrt{\tfrac{2}{|b|}}$$

As a result, the phase difference can be determined as:

$$\angle V^\star_{12} = \frac{\pi}{6}$$

Since three or more variables have already been determined, the reactive power is automatically determined as:

$$Q^\star_1 = 2 - \sqrt{3}, \qquad Q^\star_2 = 2 - \sqrt{3}$$

As shown in Example 1.1, the power flow calculation is a procedure that determines a set of $4N$ constants, given the admittance matrix $\boldsymbol{Y}$ of the power network and $2N$ simultaneous equations in steady state:

$$\begin{aligned} P^\star_1 + \boldsymbol{j}Q^\star_1 &= \sum_{j=1}^{N} \overline{\boldsymbol{Y}}_{1j} |\boldsymbol{V}^\star_1||\boldsymbol{V}^\star_j| e^{\boldsymbol{j}(\angle \boldsymbol{V}^\star_1 - \angle \boldsymbol{V}^\star_j)} \\ &\vdots \\ P^\star_N + \boldsymbol{j}Q^\star_N &= \sum_{j=1}^{N} \overline{\boldsymbol{Y}}_{Nj} |\boldsymbol{V}^\star_N||\boldsymbol{V}^\star_j| e^{\boldsymbol{j}(\angle \boldsymbol{V}^\star_N - \angle \boldsymbol{V}^\star_j)} \end{aligned} \tag{10}$$

$$\left(P^\star_1, Q^\star_1, |\boldsymbol{V}^\star_1|, \angle \boldsymbol{V}^\star_1, \ldots, P^\star_N, Q^\star_N, |\boldsymbol{V}^\star_N|, \angle \boldsymbol{V}^\star_N, \right) \tag{11}$$

where $|\boldsymbol{V}^\star_i|$ and $\angle \boldsymbol{V}^\star_i$ are the magnitude and phase angle of the voltage at bus i in polar form, and $P^\star_i$ and $Q^\star_i$ are the active and reactive power injections at bus i, respectively. Note that the phase angle of the voltage is only meaningful in a relative sense, so there are effectively only $(4N - 1)$ variables to be determined.

As described in Section 1.2, the power flow calculation can be interpreted as a procedure for finding the equilibrium point of a power system model that is capable of balancing demand and supply throughout the system. The characteristics of individual devices, such as generators and loads, are not considered in this process, and only the steady-state values of inputs and outputs at each bus are determined.

More precisely, Step B in Section 1.2 is used to calculate the equilibrium point of the internal state of the system of differential-algebraic equations. The calculation of Step B is described in Section 3.

2.2 Numerical Search Method for Steady-State Power Flow

Generally, the standard models of the Institute of Electrical Engineers in Japan [**?**], the IEEE 39-bus system model [**?**], and the IEEE 68-bus system model [**?**], provide data sheets with standard values for the power supplied by each generator bus and the power consumed by each load bus, in addition to the impedance values of each transmission line. By specifying $2N$ variables based on these standard values, the remaining variables can be explored numerically.

The data sheets typically provide the values of active and reactive power consumed by each load bus, as well as the active power supplied by each generator bus and the magnitude of the voltage phase at that bus. Therefore, $2N$ steady-state values can be specified in advance using these values. However, as shown in Example 1.1, if all steady-state values of active power at every bus are specified in advance, the remaining variables cannot be set to any value that satisfies the system of equations in 4 due to the impact of transmission losses. For instance, if the values of active or reactive power at some load buses are specified at a different steady-state value from that on the data sheet, the steady-state values of active and reactive power that should be supplied to the generator bus change, and as a result, the steady-state values of power flowing through the transmission network and the voltage phase of the bus also change, leading to a change in the total transmission loss for the entire system. Therefore, if steady-state values of active power at every generator bus are specified in advance, the system of equations in 4 cannot generally be solved.

A typical solution to this problem is to introduce a special generator bus called the **slack bus** to resolve it. At the slack bus, instead of specifying the active power, the phase of the voltage phase is specified. At this time, only the relative value of the voltage phase at each bus has meaning, so the steady-state value of the phase of the slack bus can be set to 0 without losing generality. As a result, the active power at the slack bus is automatically determined to be consistent with the total transmission loss for the entire system. The above steps can be summarized as follows.

(a) Based on the data sheet, the value of $(|\boldsymbol{V}_{i_0}^\star|, \angle \boldsymbol{V}_{i_0}^\star)$ is specified for the slack bus, the value of $(P_i^\star, |\boldsymbol{V}_i^\star|)_{i\in\mathcal{I}_\mathrm{G}\setminus\{i_0\}}$ is specified for other generator buses, and the value of $(P_i^\star, Q_i^\star)_{i\in\mathcal{I}_\mathrm{L}}$ is specified for the load bus bar.
(b) Other variables are numerically searched to satisfy simultaneous equations of Equation 4.

Here, $\mathcal{I}_\mathrm{G}$ is the set of indices for generator buses, $\mathcal{I}_\mathrm{L}$ is the set of indices for load buses, and $i_0 \in \mathcal{I}_\mathrm{G}$ represents the index of the slack bus. Unconnected buses are treated as load buses with zero consumed active and reactive power.

Table 1 Data sheet and power flow calculation results (1)

	Bus 1	Bus 2	Bus 3
$P_i^\star$	0.5	−3	
$Q_i^\star$		0	
$\lvert\boldsymbol{V}_i^\star\rvert$	2		2
$\angle\boldsymbol{V}_i^\star$			0

(a) Data sheet

	Bus 1	Bus 2	Bus 3
$P_i^\star$			2.5006
$Q_i^\star$	0.0157		0.1388
$\lvert\boldsymbol{V}_i^\star\rvert$		1.9969	
$\angle\boldsymbol{V}_i^\star$	−0.0490	−0.0596	

(b) Power flow calculation results

Table 2 Data sheet and power flow calculation results (2)

	Bus 1	Bus 2	Bus 3
$P_i^\star$		-3	0.5
$Q_i^\star$		0	
$\lvert V_i^\star\rvert$	2		2
$\angle V_i$	0		

(a) Data sheet

	Bus 1	Bus 2	Bus 3
$P_i^\star$	2.5158		
$Q_i^\star$	-0.0347		0.1759
$\lvert V_i^\star\rvert$		1.9918	
$\angle V_i^\star$		-0.0538	-0.0419

(b) Power flow calculation results

Example 1.2 Power flow calculation based on the datasheet Let us consider the electrical power system model consisting of 3 bus bars discussed in Example **??**. For the admittance matrix of the power grid of Equation **??**, the admittance values of the two transmission lines are set to be:

$$\boldsymbol{y}_{12} = 1.3652 - \boldsymbol{j}11.6041, \qquad \boldsymbol{y}_{23} = -\boldsymbol{j}10.5107 \tag{12}$$

Please note that the conductance (real part of $\boldsymbol{y}_{23}$) of the transmission line that connects bus bar 2 and bus bar 3 is set to zero, so the active power transmission loss on this line is zero.

First, let us consider a case where bus bar 1 is the generator bus, bus bar 2 is the load bus, and bus bar 3 is the slack bus. Specifically, we assume that the values in Table 1(a) are specified for each bus bar. In this case, the set of variable satisfying the simultaneous equations in Equation 4 can be obtained as shown in Table 1(b). The transmission los in this case is:

$$P_1^\star + P_2^\star + P_3^\star = 6.2562 \times 10^{-4}, \qquad Q_1^\star + Q_2^\star + Q_3^\star = 1.5450 \times 10^{-1}$$

The ratio of power transmission loss to the consumed power of the load is about 0.02%, which is very small. The reason why the power transmission loss for the active power is small is that most of the active power consumed by the load on bus bar 2 (about 2.5 pu out of 3 pu) is supplied by the generator on bus bar 3, and only 0.5 pu is supplied by the generator on bus bar 1. In other words, most of the active power consumed by the load on bus bar 2 is supplied using the right transmission line, which has no power transmission loss.

Next, consider the case where bus bar 1 is the slack bus, bus bar 2 is the load bus, and bus bar 3 is the generator bus, and specify the variables for each bus bar as shown in Table 2(a). The result of the power flow calculation in this case is shown in Table 2(b). The power loss in this case is given by:

$$P_1^\star + P_2^\star + P_3^\star = 1.5826 \times 10^{-2}, \qquad Q_1^\star + Q_2^\star + Q_3^\star = 1.4120 \times 10^{-1}$$

The ratio of power loss to consumed power of the load in this case is about 0.52%. As compared to the previous example, it can be seen that the power loss due to active power transmission has increased. This is because the majority of the active power consumed by bus 2 is being supplied using the left transmission line where power losses occur. On the other hand, it can also be seen that the reactive power loss has decreased as compared to the previous example.

From Example 1.2, it can be seen that the magnitude of the overall transmission losses of the system varies depending on the power flow distribution that is obtained. Generally, when supplying power using transmission lines with a large conductance (or resistance) component, the transmission losses of active power increase. As the transmission losses increase, the generation cost, among other economic costs, also increases because a larger amount of generation is required to supply the same amount of consumed active power.

On the other hand, it is important to note that an economically efficient steady-state power flow distribution may not necessarily correspond to a high stability equilibrium point. Therefore, in practical applications, it is important to search for better equilibrium points by considering trade-offs between economic efficiency and stability. The process of searching for such better equilibrium points is called **optimal power flow calculation** in power system engineering. In this book, we discuss the relationship between the selection of equilibrium points and stability in Chapters **??** and **??**.

2.3 The relationship between the admittance matrix and transmission loss

The following derivation provides a mathematical expression for transmission losses in any power flow distribution for a general power network consisting of N buses. Assuming that the real and imaginary parts of the admittance matrix $\boldsymbol{Y}$, which are the conductance matrix G and susceptance matrix B, respectively, are symmetric, appropriate constants $\phi_{ij} = \phi_{ji}$ and $\psi_{ij} = \psi_{ji}$ can be used to represent G_{ij} and B_{ij}, respectively, as follows:

$$G_{ij} = \begin{cases} \sum_{j=1}^{N} \phi_{ij}, & i = j \\ -\phi_{ij}, & i \neq j \end{cases} \qquad B_{ij} = \begin{cases} -\sum_{j=1}^{N} \psi_{ij}, & i = j \\ \psi_{ij}, & i \neq j \end{cases} \tag{13}$$

Here, G_{ij} and B_{ij} denote the (i, j)th element of the conductance matrix G and the susceptance matrix B, respectively. Equation 13 is equivalent to defining:

$$\phi_{ii} := \sum_{j=1}^{N} G_{ij}, \qquad \phi_{ij} := -G_{ij}, \qquad \psi_{ii} := -\sum_{j=1}^{N} B_{ij}, \qquad \psi_{ij} := B_{ij}$$

Using this expression, the following fact can be shown.

Theorem 1.1 (Expression of transmission loss through bus bar voltage phasor) *For Equation 4, the transmission losses of active and reactive power for the entire system are defined as:*

$$L_P(t) := P_1(t) + \cdots P_N(t), \qquad L_Q(t) := Q_1(t) + \cdots Q_N(t) \tag{14}$$

These transmission losses are given by:

$$\begin{aligned} L_P(t) &= \sum_{i=1}^{N} \phi_{ii} |\boldsymbol{V}_i(t)|^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \phi_{ij} W\big(\boldsymbol{V}_i(t), \boldsymbol{V}_j(t)\big) \\ L_Q(t) &= \sum_{i=1}^{N} \psi_{ii} |\boldsymbol{V}_i(t)|^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \psi_{ij} W\big(\boldsymbol{V}_i(t), \boldsymbol{V}_j(t)\big) \end{aligned} \tag{15}$$

where the following is used:

$$W(\boldsymbol{V}_i, \boldsymbol{V}_j) := \big(|\boldsymbol{V}_i| - |\boldsymbol{V}_j|\big)^2 + 2|\boldsymbol{V}_i||\boldsymbol{V}_j|\big\{1 - \cos(\angle \boldsymbol{V}_i - \angle \boldsymbol{V}_j)\big\}$$

Proof For simplification, time t is omitted. Also, we denote $\angle \boldsymbol{V}_i - \angle \boldsymbol{V}_j$ as $\angle \boldsymbol{V}_{ij}$. From Equation 4, we have $\boldsymbol{Y}_{ij} = G_{ij} + \boldsymbol{j} B_{ij}$, and $\boldsymbol{Y}_{ij} = \boldsymbol{Y}_{ji}$, therefore:

$$\begin{aligned} \sum_{i=1}^{N} (P_i + \boldsymbol{j} Q_i) &= \sum_{i=1}^{N} \overline{\boldsymbol{Y}}_{ii} |\boldsymbol{V}_i|^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \overline{\boldsymbol{Y}}_{ij} |\boldsymbol{V}_i||\boldsymbol{V}_j| \left(e^{\boldsymbol{j} \angle \boldsymbol{V}_{ij}} + e^{\boldsymbol{j} \angle \boldsymbol{V}_{ji}} \right) \\ &= \sum_{i=1}^{N} \overline{\boldsymbol{Y}}_{ii} |\boldsymbol{V}_i|^2 + 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} \overline{\boldsymbol{Y}}_{ij} |\boldsymbol{V}_i||\boldsymbol{V}_j| \cos \angle \boldsymbol{V}_{ij} \end{aligned}$$

Thus, using the notation of G_{ij} and B_{ij} from Equation 13, we obtain:

$$\begin{aligned} L_P &= \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \phi_{ij} \right) |\boldsymbol{V}_i|^2 - 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} \phi_{ij} |\boldsymbol{V}_i||\boldsymbol{V}_j| \cos \angle \boldsymbol{V}_{ij} \\ L_Q &= \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \psi_{ij} \right) |\boldsymbol{V}_i|^2 - 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} \psi_{ij} |\boldsymbol{V}_i||\boldsymbol{V}_j| \cos \angle \boldsymbol{V}_{ij} \end{aligned}$$

where we focus on the first term of L_P. Because $\phi_{ij} = \phi_{ji}$, we have:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \phi_{ij} |\boldsymbol{V}_i|^2 = \sum_{i=1}^{N} \phi_{ii} |\boldsymbol{V}_i|^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \phi_{ij} |\boldsymbol{V}_i|^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \phi_{ij} |\boldsymbol{V}_j|^2$$

If we rewrite the first term of L_P using this, we can obtain:

$$
\begin{aligned}
L_P &= \sum_{i=1}^{N} \phi_{ii} |\boldsymbol{V}_i|^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \phi_{ij} \left(|\boldsymbol{V}_i|^2 + |\boldsymbol{V}_j|^2 - 2|\boldsymbol{V}_i||\boldsymbol{V}_j| \cos \angle \boldsymbol{V}_{ij} \right) \\
&= \sum_{i=1}^{N} \phi_{ii} |\boldsymbol{V}_i|^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \phi_{ij} \left\{ \left(|\boldsymbol{V}_i| - |\boldsymbol{V}_j| \right)^2 + 2|\boldsymbol{V}_i||\boldsymbol{V}_j| (1 - \cos \angle \boldsymbol{V}_{ij}) \right\}
\end{aligned}
$$

Therefore, L_P can be expressed in the form of Equation 15. With the same steps, L_Q can be expressed with the format of Equation 15.

Theorem 1.1 shows that the discussion of electrical power loss in the electrical power system of two bus bars shown in the Example 1.1 can be generalized to power systems consisting of any number of buses. Here, if we express the admittance matrix $\boldsymbol{Y}$ as in Section **??**:

$$
\boldsymbol{Y} = \boldsymbol{Y}_0 + \boldsymbol{j}\, \mathsf{diag}(b_i)_{i \in \{1,\dots,N\}}
$$

In this case, it is known that $\boldsymbol{Y}_0 \mathbb{1} = 0$, which means that the sum of each row of the real and imaginary parts of $\boldsymbol{Y}_0$ is 0. Thus, for all $i \in 1, \dots, N$:

$$
\phi_{ii} = 0, \qquad \psi_{ii} = -b_i
$$

Furthermore, since the conductance of each transmission line is nonnegative and the susceptance is negative, it follows that ϕ_{ij} and ψ_{ij} are nonnegative for all $i \neq j$. Therefore, as long as ϕ_{ij} is not 0, loss will always occur when transmitting active power between a bus bar i and a bus bar j. This implies that the transmission loss of active power $L_P(t)$ is positive at any time t for the entire power system. Similarly, if b_i is sufficiently small; in other words, if the capacitance to ground of a transmission line is sufficiently small, loss of reactive power $L_Q(t)$ is also positive. In other words, connecting devices with capacitive characteristics to the buses reduces the loss of reactive power.

3 The steady state of generators achieving desired power supply

In this Section, we explain how to retroactively calculate the steady-state values of the internal states and external inputs of generators, as well as the parameter values of loads, to achieve the power flow state described by the results of the power flow calculation explained in Step B of Section 1.2.

3.1 The steady state of generators that achieves a desired supply of electric power

Let us consider a generator model that inputs the voltage phasor in Section **??**. For simplicity of notation, subscripts i have been omitted:

$$\begin{aligned}\dot{\delta} &= \omega_0 \Delta\omega \\ M\Delta\dot{\omega} &= -D\Delta\omega - P + P_{\mathrm{mech}} \\ \tau\dot{E} &= -\tfrac{X}{X'}E + \left(\tfrac{X}{X'} - 1\right)|\boldsymbol{V}|\cos(\delta - \angle\boldsymbol{V}) + V_{\mathrm{field}}\end{aligned} \tag{16}$$

Here, if the active power and reactive power are the outputs:

$$P = \frac{|\boldsymbol{V}|E}{X'}\sin(\delta - \angle\boldsymbol{V}), \qquad Q = \frac{|\boldsymbol{V}|E}{X'}\cos(\delta - \angle\boldsymbol{V}) - \frac{|\boldsymbol{V}|^2}{X'} \tag{17}$$

The purpose here is to determine the steady-state values of the internal state and external input values of a generator that are consistent with the given constant values of active power, reactive power, and voltage phasor of the bus to which the generator is connected, as the result of power flow calculation. Specifically, let $(P^\star, Q^\star, |V^\star|, \angle\boldsymbol{V}^\star)$ denote the given set of active power, reactive power, and absolute value and phase of the voltage phasor. Then, the objective is to determine the steady-state values of the internal state of the generator $(\delta^\star, E^\star)$ and the external input values $(P^\star_{\mathrm{mech}}, V^\star_{\mathrm{field}})$ that satisfy the following set of equations:

$$\begin{aligned}p^\star &= \tfrac{|v^\star|e^\star}{X'}\sin(\delta^\star - \angle\boldsymbol{v}^\star), \\ q^\star &= \tfrac{|v^\star|e^\star}{X'}\cos(\delta^\star - \angle\boldsymbol{v}^\star) - \tfrac{|v^\star|^2}{X'}, \\ 0 &= -p^\star + p^\star_{\mathrm{mech}}, \\ 0 &= -\tfrac{X}{X'}e^\star + \left(\tfrac{X}{X'} - 1\right)|v^\star|\cos(\delta^\star - \angle\boldsymbol{v}^\star) + v^\star_{\mathrm{field}}\end{aligned} \tag{18}$$

The set of equations in 18 represents the equilibrium point of the generator model when the steady-state value of the angular frequency deviation $\Delta\omega$ in Equation 16 is zero. Also, the given $(P^\star, Q^\star, |V^\star|, \angle\boldsymbol{V}^\star)$ corresponds to the input and output values of the generator model for each bus.

The specific values of the steady-state internal states of the generator that satisfy Equation 18 are given by:

$$\begin{aligned}\delta^\star &= \angle\boldsymbol{v}^\star + \arctan\left(\frac{p^\star}{q^\star + \frac{|\boldsymbol{v}^\star|^2}{X'}}\right), \\ e^\star &= \frac{X'}{|\boldsymbol{v}^\star|}\sqrt{\left(q^\star + \frac{|\boldsymbol{v}^\star|^2}{X'}\right)^2 + (p^\star)^2}\end{aligned} \tag{19a}$$

The steady-state values of the mechanical and field inputs are:

$$p^\star_{\text{mech}} = p^\star,$$
$$v^\star_{\text{field}} = \frac{\frac{X}{|\boldsymbol{v}^\star|}\left\{\left(q^\star + \frac{|\boldsymbol{v}^\star|^2}{X'}\right)\left(q^\star + \frac{|\boldsymbol{v}^\star|^2}{X}\right) + (p^\star)^2\right\}}{\sqrt{\left(q^\star + \frac{|\boldsymbol{v}^\star|^2}{X'}\right)^2 + (p^\star)^2}} \tag{19b}$$

Please refer to Section 3.3 for the derivation process of these equations.

3.2 Load Parameters for Achieving Desired Power Consumption

In this section, we explain how to set the parameters of the load model described in Section **??** to achieve the desired power consumption. By eliminating the current phase using Equation 3, the fixed impedance model can be expressed as follows:

$$p + \boldsymbol{j}q = -\frac{|\boldsymbol{v}|^2}{\overline{\boldsymbol{z}}^\star_{\text{load}}} \tag{20}$$

Here, the subscript i of the bus is omitted for simplicity. This can be interpreted as the fixed impedance model of the load, where the current phase $\boldsymbol{V}$ is the input and the active power P and reactive power Q are the outputs. If the values of the active power and reactive power determined by power flow calculations are represented as $P^\star$ and $Q^\star$, respectively, and the absolute value of the voltage phase is represented as $|\boldsymbol{V}^\star|$, then the real part (resistance) and imaginary part (reactance) of the load impedance $\boldsymbol{z}^\star_{\text{load}}$ can be calculated as:

$$\mathrm{Re}[\boldsymbol{z}^\star_{\text{load}}] = -\frac{p^\star|\boldsymbol{v}^\star|^2}{(p^\star)^2 + (q^\star)^2}, \qquad \mathrm{Im}[\boldsymbol{z}^\star_{\text{load}}] = -\frac{q^\star|\boldsymbol{v}^\star|^2}{(p^\star)^2 + (q^\star)^2} \tag{21}$$

Similarly, for the fixed current model, since the equation can be expressed as:

$$p + \boldsymbol{j}q = \overline{\boldsymbol{i}}^\star_{\text{load}}|\boldsymbol{v}| \tag{22}$$

The real and imaginary parts of the load current parameter can be calculated as:

$$\mathrm{Re}[\boldsymbol{i}^\star_{\text{load}}] = \frac{p^\star}{|\boldsymbol{v}^\star|}, \qquad \mathrm{Im}[\boldsymbol{i}^\star_{\text{load}}] = -\frac{q^\star}{|\boldsymbol{v}^\star|}$$

For the fixed power model, since the equation can be expressed as:

$$p + \boldsymbol{j}q = p^\star_{\text{load}} + \boldsymbol{j}q^\star_{\text{load}} \tag{23}$$

Then, clearly, the parameters are:

$$p^\star_{\text{load}} = p^\star, \qquad q^\star_{\text{load}} = q^\star$$

By setting these parameter values in the load model, the steady-state flow state obtained by power flow calculation can be achieved.

3.3 mathematical relationship between generator internal state and input/output

In the following, we mathematically analyze the internal state of the generator, the active and reactive power provided to the bus bar, and voltage phasor of the bus bar. We use the generator model discussed in Section **??**. However, for ease of notation, we omit the subscript i and use the model as follows:

$$\begin{aligned}\dot{\delta} &= \omega_0 \Delta\omega \\ M\Delta\dot{\omega} &= -D\Delta\omega - P + P_{\mathrm{mech}} \\ \tau\dot{E} &= -\tfrac{X_{\mathrm{d}}}{X_{\mathrm{d}}'}E + \left(\tfrac{X_{\mathrm{d}}}{X_{\mathrm{d}}'} - 1\right)|\boldsymbol{V}|\cos(\delta - \angle\boldsymbol{V}) + V_{\mathrm{field}}\end{aligned} \tag{24}$$

When the output is given as active and reactive power, we have:

$$\begin{aligned}P &= \frac{|\boldsymbol{V}|E}{X_{\mathrm{d}}'}\sin(\delta - \angle\boldsymbol{V}) - \left(\frac{1}{X_{\mathrm{d}}'} - \frac{1}{X_{\mathrm{q}}}\right)|\boldsymbol{V}|^2\sin(\delta - \angle\boldsymbol{V})\cos(\delta - \angle\boldsymbol{V}), \\ Q &= \frac{|\boldsymbol{V}|E}{X_{\mathrm{d}}'}\cos(\delta - \angle\boldsymbol{V}) - |\boldsymbol{V}|^2\left(\frac{\cos^2(\delta - \angle\boldsymbol{V})}{X_{\mathrm{d}}'} + \frac{\sin^2(\delta - \angle\boldsymbol{V})}{X_{\mathrm{q}}}\right)\end{aligned} \tag{25}$$

Note that X_{d}' and X_{q} are equal to X', and if we replace X_{d} with X, this model matches the generator model discussed in Section 3.1. When the output is given as current phasors, we have:

$$\begin{aligned}|\boldsymbol{I}|\cos(\delta - \angle\boldsymbol{I}) &= \frac{|\boldsymbol{V}|}{X_{\mathrm{q}}}\sin(\delta - \angle\boldsymbol{V}), \\ |\boldsymbol{I}|\sin(\delta - \angle\boldsymbol{I}) &= \frac{E - |\boldsymbol{V}|\cos(\delta - \angle\boldsymbol{V})}{X_{\mathrm{d}}'}\end{aligned} \tag{26}$$

Note that equations 25 and 26 are equivalent outputs, meaning that for any given $(\delta, E, |\boldsymbol{V}|, \angle\boldsymbol{V})$, there exists a one-to-one relationship between (P, Q) and $(|\boldsymbol{I}|, \angle\boldsymbol{I})$.

The following facts are shown for this generator model.

Lemma 1.1 (Relationship between generator internal states and input/output)
Consider Equation 25 as a system of equations in $\delta - \angle\boldsymbol{V}$ and E. The solution is given by:

$$\delta - \angle\boldsymbol{V} = \arctan\left(\frac{P}{Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}}}\right), \tag{27a}$$

$$E = \frac{\frac{X_{\mathrm{d}}'}{|\boldsymbol{V}|}\left\{\left(Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}}\right)\left(Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{d}}'}\right) + P^2\right\}}{\sqrt{\left(Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}}\right)^2 + P^2}} \tag{27b}$$

where $|\boldsymbol{V}| \neq 0$*.*

Conversely, if we consider Equation 27 as a system of equations in P and Q, the solution is given by Equation 25.

Proof First, we derive Equation 27 from Equation 25. Multiplying P by $\cos(\delta - \angle \boldsymbol{V})$ and Q by $\sin(\delta - \angle \boldsymbol{V})$ and taking the difference, we obtain:

$$P\cos(\delta - \angle \boldsymbol{V}) - Q\sin(\delta - \angle \boldsymbol{V}) = \frac{|\boldsymbol{V}|^2}{X_\mathrm{q}}\sin(\delta - \angle \boldsymbol{V})$$

Dividing both sides by $\cos(\delta - \angle \boldsymbol{V})$, we obtain Equation 27a. Next, we show the relation of Euqtion 27b. Rewriting P and Q in terms of $\boldsymbol{I}$ using the Equation 3, Equation 25 is equivalently transformed to Equation 26. Furthermore, this is equivalent to:

$$|\boldsymbol{V}|e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})} = E - X'_\mathrm{d}|\boldsymbol{I}|\sin(\delta - \angle \boldsymbol{I}) + \boldsymbol{j}X_\mathrm{q}|\boldsymbol{I}|\cos(\delta - \angle \boldsymbol{I}) \tag{28}$$

The relationship in Equation 27a can be expressed in complex numbers as:

$$\frac{e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})} - e^{-\boldsymbol{j}(\delta - \angle \boldsymbol{V})}}{e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})} + e^{-\boldsymbol{j}(\delta - \angle \boldsymbol{V})}} = \underbrace{\frac{P}{Q + \frac{|\boldsymbol{V}|^2}{X_\mathrm{q}}}}_{\alpha}\boldsymbol{j}$$

Therefore, we obtain:

$$e^{-\boldsymbol{j}(\delta - \angle \boldsymbol{V})} = \frac{1 - \alpha\boldsymbol{j}}{1 + \alpha\boldsymbol{j}}e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})}$$

By considering the complex conjugate of Equation 3, it can be transformed into an equivalent form given by:

$$|\boldsymbol{I}|e^{\boldsymbol{j}(\delta - \angle \boldsymbol{I})} = \frac{P + \boldsymbol{j}Q}{|\boldsymbol{V}|}e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})} \tag{29}$$

Then, by taking the complex conjugate of the above equation:

$$|\boldsymbol{I}|e^{-\boldsymbol{j}(\delta - \angle \boldsymbol{I})} = \frac{P - \boldsymbol{j}Q}{|\boldsymbol{V}|}\cdot\frac{1 - \alpha\boldsymbol{j}}{1 + \alpha\boldsymbol{j}}e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})} \tag{30}$$

From Equations 29 and 30:

$$\begin{aligned} |\boldsymbol{I}|\sin(\delta - \angle \boldsymbol{I}) &= \frac{1}{|\boldsymbol{V}|}\cdot\frac{\alpha P + Q}{1 + \alpha\boldsymbol{j}}e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})}, \\ |\boldsymbol{I}|\cos(\delta - \angle \boldsymbol{I}) &= \frac{1}{|\boldsymbol{V}|}\cdot\frac{P - \alpha Q}{1 + \alpha\boldsymbol{j}}e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})} \end{aligned}$$

By substituting these into Equation 28 and rewriting $\boldsymbol{I}$ in terms of P and Q, we can show that when Equation 27a holds, Equation 25 is equivalent to:

$$E = \frac{X'_{\mathrm{d}}}{|\boldsymbol{V}|} \left\{ \left(Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}} \right) \left(Q + \frac{|\boldsymbol{V}|^2}{X'_{\mathrm{d}}} \right) + P^2 \right\} \frac{Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}} - \boldsymbol{j} P}{\left(Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}} \right)^2 + P^2} e^{\boldsymbol{j}(\delta - \angle \boldsymbol{V})} \tag{31}$$

Here, we have used the fact that from Equation 27a:

$$Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}} - \boldsymbol{j} P = \left| Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}} - \boldsymbol{j} P \right| e^{-\boldsymbol{j}(\delta - \angle \boldsymbol{V})}$$

Furthermore, since $|E| = E$ and

$$\left(Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}} \right)^2 + P^2 = \left| Q + \frac{|\boldsymbol{V}|^2}{X_{\mathrm{q}}} - \boldsymbol{j} P \right|^2$$

Equation 27b can be obtained.

We can follow the reverse steps to derive Equation 25 from Equation 27. Using Equation 27a, we can replace the E in Equation 27b with the expression in Equation 31. As mentioned earlier, when Equation 27a holds, Equation 31 is equivalent to Equation 25. □

Lemma 1.1 shows that there is a one-to-one relationship between the pairs $(\delta - \angle \boldsymbol{V}, E)$ and (P, Q). Specifically, it shows that the internal state of generators (δ, E) can be uniquely determined from the input/output $(|\boldsymbol{V}|, \angle \boldsymbol{V})$ or (P, Q). Please note that the relationship in Equation (27) holds at any time t, regardless of whether the system is in steady-state or transient. The following theorem gives a relationship that holds between the input, output, and internal state under a steady state of generators.

Theorem 1.2 (Relationship between generator internal states and input/output in steady state)

Consider the generator model given by Equations 24 and 25. Let $\boldsymbol{V}^\star$, $\Delta\omega^\star$, $\angle \boldsymbol{V}^\star$, $P^\star$, and $Q^\star$ be real constants. Let us assume the inputs of mechanical torque and field voltage are constants given by:

$$\begin{aligned} P_{\mathrm{mech}}(t) &= D\Delta\omega^\star + P^\star, \\ V_{\mathrm{field}}(t) &= \frac{\frac{X_{\mathrm{d}}}{|\boldsymbol{V}^\star|} \left\{ \left(Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X_{\mathrm{q}}} \right) \left(Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X_{\mathrm{d}}} \right) + (P^\star)^2 \right\}}{\sqrt{\left(Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X_{\mathrm{q}}} \right)^2 + (P^\star)^2}} \end{aligned} \tag{32a}$$

and the input due to the voltage phase angle of the bus are given by:

$$|\boldsymbol{V}(t)| = |\boldsymbol{V}^\star|, \qquad \angle \boldsymbol{V}(t) = \omega_0 \Delta\omega^\star t + \angle \boldsymbol{V}^\star \tag{32b}$$

where ω_0 is the nominal frequency. Then, in steady state, the rotor angle, angular frequency deviation, and internal voltage are given by

$$
\begin{aligned}
\delta(t) &= \angle \boldsymbol{V}(t) + \arctan\left(\frac{P^\star}{Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X_\mathrm{q}}}\right),\\
\Delta\omega(t) &= \Delta\omega^\star,\\
E(t) &= \frac{\frac{X'_\mathrm{d}}{|\boldsymbol{V}^\star|}\left\{\left(Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X_\mathrm{q}}\right)\left(Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X'_\mathrm{d}}\right) + (P^\star)^2\right\}}{\sqrt{\left(Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X_\mathrm{q}}\right)^2 + (P^\star)^2}}
\end{aligned}
\tag{33}
$$

Furthermore, the active and reactive power supplied to bus bars are constants and given by:

$$
P(t) = P^\star, \qquad Q(t) = Q^\star \tag{34}
$$

Proof First, we demonstrate that Equation 34 holds for the output under the assumption that Equation 33 is a solution to the differential equation in Equation 24 with the input given by Equation 32. As shown in Lemma 1.1, considering Equation 33 as equations for $P^\star$ and $Q^\star$, their solutions are given by

$$
\begin{aligned}
P^\star &= \frac{|\boldsymbol{V}^\star| E(t)}{X'_\mathrm{d}} \sin\left(\delta(t) - \angle \boldsymbol{V}(t)\right)\\
&\quad - \left(\frac{1}{X'_\mathrm{d}} - \frac{1}{X_\mathrm{q}}\right) |\boldsymbol{V}^\star|^2 \sin\left(\delta(t) - \angle \boldsymbol{V}(t)\right) \cos\left(\delta(t) - \angle \boldsymbol{V}(t)\right),\\
Q^\star &= \frac{|\boldsymbol{V}^\star| E(t)}{X'_\mathrm{d}} \cos\left(\delta(t) - \angle \boldsymbol{V}(t)\right)\\
&\quad - |\boldsymbol{V}^\star|^2 \left(\frac{\cos^2\left(\delta(t) - \angle \boldsymbol{V}(t)\right)}{X'_\mathrm{d}} + \frac{\sin^2\left(\delta(t) - \angle \boldsymbol{V}(t)\right)}{X_\mathrm{q}}\right)
\end{aligned}
$$

which implies Equation 34.

Next, we verify that Equation 33 is a solution to the differential equation in Equation 24 with the input given by equation 32. The differential equation for δ and $\Delta\omega$ in Equation 24 is equivalent to

$$
\frac{M}{\omega_0}\ddot{\delta}(t) + \frac{D}{\omega_0}\dot{\delta}(t) + P(t) - P_\mathrm{mech}(t) = 0
$$

If we substitute the expressions for $P_\mathrm{mech}(t)$ from Equation 32a, $P(t)$ from Equation 34, and $\delta(t)$ from Equation 33 into the relationship given by Equation 32b, we can see that this differential equation is satisfied. Similarly, by substituting the expressions for $V_\mathrm{field}(t)$ from Equation 32a, $\delta(t) - \angle \boldsymbol{V}(t)$, and $E(t)$ from Equation 33 into Equation 24, we can see that the differential equation for E is satisfied. Here, we use the fact that:

$$\cos\left(\arctan\left(\frac{P^\star}{Q^\star + \frac{|V^\star|^2}{X_\mathrm{q}}}\right)\right) = \frac{Q^\star + \frac{|V^\star|^2}{X_\mathrm{q}}}{\sqrt{\left(Q^\star + \frac{|V^\star|^2}{X_\mathrm{q}}\right)^2 + (P^\star)^2}}$$

From the above, we prove Theorem 1.2.

From Theorem 1.2, we can determine the values of mechanical input $P^\star_\mathrm{mech}$ and field input $V^\star_\mathrm{field}$ that are necessary to achieve the voltage phase, active power, and reactive power of the bus bars determined by power flow calculation, as well as the steady-state behavior of internal states (δ, E) at that time. Note that in Equation 32b, the phase of the voltage phasor is not a constant, but the steady-state value of the angular frequency deviation $\Delta\omega^\star$, which usually should be set to zero as a constant with practical significance, is determined only by the value of $\angle \boldsymbol{V}^\star$ obtained from the power flow calculation. Clearly, when $\Delta\omega^\star = 0$,

$$P_\mathrm{mech}(t) = P^\star, \qquad \delta(t) = \angle \boldsymbol{V}^\star + \arctan\left(\frac{P^\star}{Q^\star + \frac{|\boldsymbol{V}^\star|^2}{X_\mathrm{q}}}\right)$$

From the above discussion, we can see that even if we determine $(P^\star, Q^\star, |\boldsymbol{V}^\star|, \angle \boldsymbol{V}^\star)$ by power flow calculation without considering the dynamic characteristics of the generator, we can uniquely calculate $(P^\star_\mathrm{mech}, V^\star_\mathrm{field})$ that are consistent with them. This result leads to Equation (19).

Furthermore, the following theorem gives an equivalence relation between the inputs and internal states in steady-state of the generator.

Theorem 1.3 (Equivalence relation between input/output and internal state of generator)

For the generator model given by Equations 24 and 25, the necessary and sufficient conditions for Equations 35 to hold for all $t \geq 0$ *are that Equations 36 hold for all* $t \geq 0$, *where*

$$\frac{d^2\delta}{dt^2}(t) = 0, \quad \frac{dE}{dt}(t) = 0, \quad \frac{dP_\mathrm{mech}}{dt}(t) = 0, \quad \frac{dV_\mathrm{field}}{dt}(t) = 0 \tag{35}$$

and

$$\frac{dP}{dt}(t) = 0, \quad \frac{dQ}{dt}(t) = 0, \quad \frac{d|\boldsymbol{V}|}{dt}(t) = 0, \quad \frac{d^2\angle \boldsymbol{V}}{dt^2}(t) = 0 \tag{36}$$

Moreover, when equations 35 or 36 hold, the following holds for all $t \geq 0$:

$$\Delta\omega(t) = \frac{1}{\omega_0}\frac{d\angle \boldsymbol{V}}{dt}(t) \tag{37}$$

which is a constant.

Proof First, we show that if Equation 35 holds, then Equation 36 also holds. From Equation 24, since $\Delta\omega$, E, P_mech, and V_field are all constants, we can see that P and $|\boldsymbol{V}|\cos(\delta - \angle \boldsymbol{V})$ are also constants. Therefore, from the two equations in Equation 25, we can see that Q and $|\boldsymbol{V}|\sin(\delta - \angle \boldsymbol{V})$ are also constants. Additionally,

$$|\boldsymbol{V}|^2 \cos^2(\delta - \angle \boldsymbol{V}) + |\boldsymbol{V}|^2 \sin^2(\delta - \angle \boldsymbol{V}) = |\boldsymbol{V}|^2$$

and since the left-hand side is constant, we can see that $|\boldsymbol{V}|$ is also constant. Furthermore, in the first equation of Equation 27, the right-hand side is constant, so the derivatives of $\angle \boldsymbol{V}$ and δ are equal for any order. Therefore, we have

$$\frac{d^2 \angle \boldsymbol{V}}{dt^2} = \frac{d^2 \delta}{dt^2} = 0$$

Next, we show that if Equation 36 holds, then Equation 35 also holds. From Equation 27, we know that E is constant and that the second derivative of δ is 0, which means that $\Delta\omega$ is constant. Therefore, from Equation 24, since P and $|\boldsymbol{V}| \cos(\delta - \angle \boldsymbol{V})$ are constants, we can see that P_{mech} and V_{field} are also constants. Equation 37 is obvious from the fact that the derivatives of $\angle \boldsymbol{V}$ and δ are equal. □

As Theorem 1.3 shows, the fact that the external inputs $(P_{\mathrm{mech}}, V_{\mathrm{field}})$ to the generator are constant and the internal states (δ, E) are in a steady state is equivalent to the input/output $(P, Q, |\boldsymbol{V}|, \angle \boldsymbol{V})$ to the bus being in a steady state. Therefore, the procedures for power flow calculations that determine the active and reactive power and voltage phase angles for each bus are mathematically equivalent to searching for one of the equilibrium points of the power system model, assuming that all internal states and external inputs of the generators are in steady state. This is the steady-state of the entire power system.

4 Time Response Calculation of Power System Model

In this Section, we explain a method to calculate the time response of the electric power system model explained as Step C in Section 1.2. We will also numerically analyze the behavior of the electrical power system against several disturbances.

4.1 Initial Value Response

In this section, we explain the method for calculating the time response of the power system model, which was described as Step C in Section 1.2. We also numerically analyze the behavior of the power system model in response to several disturbances.

By following the procedures outlined in Sections 2 and 3, the equilibrium point of the power system model can be obtained as a steady-state where all generator frequency deviations are zero. Specifically, by using the bus variables determined in the power flow calculation, the initial values of the internal states and external inputs of each generator model, as shown in Theorem 1.2, can be set. In addition, the constants derived in Section 3.2 can be set for each load model. By doing so, the differential-algebraic equation system representing the power system model is in equilibrium at the given steady-state power flow condition.

In particular, if the obtained equilibrium point is stable in a meaningful sense, then even if some perturbation occurs in the internal states of the generators, the internal states of the power system model will asymptotically converge to the original steady-state values. We confirm this fact in the following example.

Table 3 Steady-state values of generators for flow calculation results of Table 1

i	$P^{\star}_{\mathrm{mech}i}$ [pu]	$V^{\star}_{\mathrm{filed}i}$ [pu]	$\delta^{\star}_{i}$ [rad]	$\Delta\omega^{\star}_{i}$ [pu]	$E^{\star}_{i}$ [pu]
1	0.5000	2.0442	0.0670	0	2.0210
3	2.5006	2.5062	0.3870	0	2.2097

Table 4 Steady-state values of generators for flow calculation results of Table 2

i	$P^{\star}_{\mathrm{mech}i}$ [pu]	$V^{\star}_{\mathrm{filed}i}$ [pu]	$\delta^{\star}_{i}$ [rad]	$\Delta\omega^{\star}_{i}$ [pu]	$E^{\star}_{i}$ [pu]
1	2.5158	2.7038	0.5356	0	2.3069
3	0.5000	2.1250	0.0390	0	2.0654

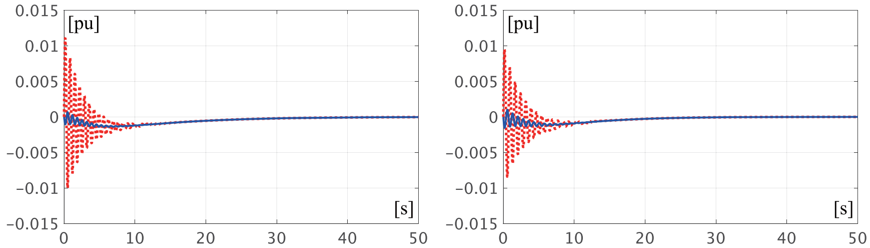
Example 1.3 Initial response of the electrical power system model Let us consider the electrical power system model consisting of three bus bars handled in the Example 1.2. Let us consider a situation where the generator model of Equation 16 is connected to bus bar 1 and bus bar 3, and a load model with constant impedance of Equation 20 is connected to bus bar 2. For the generator model, we use the constants for generator 1 and generator 2 presented in **??**.

For the two results of the power flow calculation in 1 and 2, we calculate the steady values of mechanical torque, field voltage, rotor argument, and interval voltage for generator 1 and generator 2 based on Equation (19). The calculated steady values are shown in 3 and 4. Similarly, if we back calculate the load impedance based on Equation 21, it can be obtained as in the first row of 5 and 6. Please note that, depending on the result of the power flow calculation, the internal state of the generators, the steady value of the external input, and the load impedance vary.

Let us perturb the steady value of the internal state of the generators as the initial value. Specifically:

$$\begin{bmatrix} \delta_1(0) \\ \delta_3(0) \end{bmatrix} = \begin{bmatrix} \delta_1^{\star} + \frac{\pi}{6} \\ \delta_3^{\star} \end{bmatrix}, \qquad \begin{bmatrix} E_1(0) \\ E_3(0) \end{bmatrix} = \begin{bmatrix} E_1^{\star} + 0.1 \\ E_3^{\star} \end{bmatrix} \tag{38}$$

The initial value of the frequency deviation is 0. Figure 2 shows the initial value response of the frequency deviation corresponding to the two power flow calculations at this time. The solid blue line represents the frequency deviation of generator 1,



(a) Steady-state power flow of Table 1 (b) Steady-state power flow of Table 2

Fig. 2 Time response of frequency deviation for initial value disturbance
(Solid blue line: $\Delta\omega_1$, Dashed red line:$\Delta\omega_3$)

while the broken red line is the frequency deviation of generator 3. With either one of the steady power flow distributions, oscillations with a similar frequency were generated. Since the system frequency was set to 60 [Hz], a frequency deviation of 0.015 [pu] is equal to 0.9 [Hz].

Table 5 Impedance value of load [pu]

Load	$\mathrm{Re}[z^{\star}_{\mathrm{load2}}]$	$\mathrm{i}[z^{\star}_{\mathrm{load2}}]$
Steady-state	1.3293	0
Increased laod	1.3426	0
Decreased load	1.3160	0

(a) Results of power flow calculation for Table1

Table 6 Impedance value of load [pu]

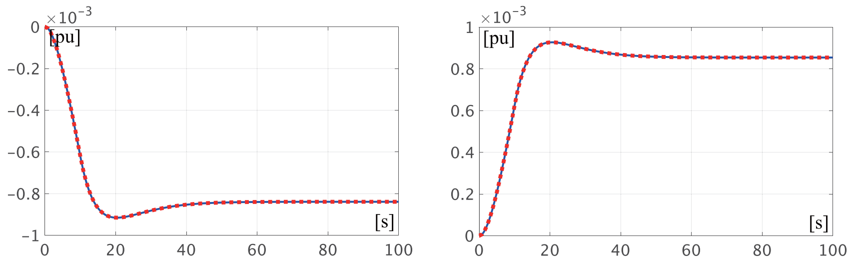
Load	$\mathrm{Re}[z^{\star}_{\mathrm{load2}}]$	$\mathrm{i}[z^{\star}_{\mathrm{load2}}]$
Steady-state	1.3224	0
Increased load	1.3356	0
Decreased load	1.3092	0

(a) Results of power flow calculation for Table 2

4.2 Response to Parameter Variations of the Load Model

For a power system model in steady-state, if any of the constants in the load model are changed, the values of the voltage and current phases at all buses generally change. At this point, the power supply and demand in the entire system generally no longer balance, so unless the value of the mechanical input is appropriately modified according to the given excitation input value, the angular frequency deviation of each generator will not converge to zero. Let's verify this in the following example.

Example 1.4 Time Response of Power System Model to Load Impedance Changes Let us calculate the time response of the angular frequency deviation when the load impedance value changes under the same setting as Example 1.3.



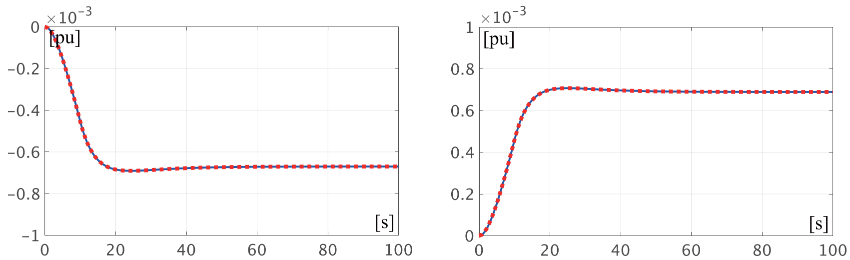
(a) If resistance is increased by 1%   (b) If the resistance is reduced by 1%

Fig. 3 Time response of angular frequency deviation to load change
(Steady-state state of Table 1, line type is the same as Figure 2.)

Specifically, we calculate the time response by increasing or decreasing the resistance of the load by 1% based on the results of the power flow calculation. The increased and decreased load impedance values are shown in the second and third rows of Tables 5 and 6, respectively.

The calculation results are shown in Figures 3 and 4. The solid blue line represents the angular frequency deviation of generator 1, and the dashed red line represents that of generator 3. In all cases, it can be seen that the angular frequency deviations of the two generators change in sync. Also, since increasing the resistance generally increases the consumption of active power, the frequency of the generators decreases. The opposite is true when the resistance is decreased. It should be noted that since the mechanical input of the generators is fixed at the value before the change in resistance, the balance of active power supply and demand is not maintained, and the steady-state value of the angular frequency deviation is nonzero. Furthermore, although the ratio of the change in load resistance is the same in both Figures 3 and 4, the resulting values of the angular frequency deviation are different. This suggests that the sensitivity (stability) to disturbances changes depending on how the equilibrium point of the power system model is chosen.

In Example 1.4, unlike the result in Example 1.3, the generator's frequency deviation has a non-zero steady-state value. In order to make this frequency deviation zero, it is necessary to adjust the mechanical input or excitation input of the generator group appropriately. On the other hand, to make the steady-state value of the frequency deviation zero, frequency control is generally performed by a control algorithm that adjusts the mechanical input of the generator group to balance the demand and supply of active power. However, it is practically difficult to accurately measure the changes in all loads. Therefore, feedback control that automatically searches for the value of the mechanical input that achieves the demand-supply balance through control operations based on an integrator is necessary. The details of this are described later in Sections **??** and **??**.



(a) If resistance is increased by 1%

(b) If the resistance is reduced by 1%

Fig. 4 Time response of angular frequency deviation to changes in load
(Steady-state of Table 2, line type is the same as Figure 2.)

4.3 Response to ground fault

4.3.1 What is ground fault?

When an electric circuit comes into contact with the ground due to contact with an object or lightning strike, a phenomenon called **ground fault** occurs, and a large current flows into the ground. Ground fault detection devices quickly operate in power systems to detect the occurrence of ground fault and block the ground fault current. If the ground fault current is removed and the blocking is released, the power system operation before the occurrence of ground fault can be restored.

The time required to detect and remove ground fault is generally about 70 [ms]. During this time, the ground fault current flowing into the ground can cause severe disturbances to the power system operation. Note that the disturbance due to ground fault is not modeled as an external input, but is modeled as a "switching to a different power system model during the time when the ground fault is still present."

4.3.2 Formulation of bus bar ground fault

Bus bar ground faults discussed in this book are modeled assuming that the voltage phasor of the bus is constrained to zero during the time the fault persists. Without losing generality, we explain it for the case where a fault occurs on bus bar 1 of a power system model consisting of N bus bars. Additionally, we assume that the fault occurs at time 0 and persists until time t_0. Under these conditions, the power flow state in Equation 2 satisfies the algebraic equation of the normal operating state in Equation 1 for $t < 0$ and $t \geq t_0$. However, for the time period when the fault persists, $t \in [0, t_0)$, the power flow state must satisfy the algebraic equation in Equation 39, which excludes the equation related to the current phasor of the faulted bus $\boldsymbol{I}_1(t)$, and the constraint condition on the voltage phasor of the faulted bus:

$$\begin{bmatrix} \boldsymbol{I}_2(t) \\ \vdots \\ \boldsymbol{I}_N(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{22} & \cdots & \boldsymbol{Y}_{2N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{N2} & \cdots & \boldsymbol{Y}_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_2(t) \\ \vdots \\ \boldsymbol{V}_N(t) \end{bmatrix} \tag{39a}$$

and the constraint condition on the voltage phase of the faulted bus:

$$|\boldsymbol{V}_1(t)| = 0 \tag{39b}$$

Note that for buses and time periods where a ground fault has not occurred, the voltage and current phasors follow the differential or algebraic equations that represent each equipment's model. This is the same as during normal operation. Therefore, in numerical simulations, it is necessary to switch between the following two models:

- During periods where a ground fault has not occurred, the normal power system model is used.
- During periods where a ground fault is ongoing, the power system model is used with the affected bus and its connected equipment and transmission lines removed.

The current phasor $\boldsymbol{I}1(t)$ flowing from device 1 to bus 1 during a ground fault is determined by the dynamic characteristics of the device. Specifically, if the device is a generator, then its internal state evolves over time according to following differential equations for $t \in [0, t_0)$, where $|\boldsymbol{V}1(t)|$ is set to 0.

$$\begin{aligned} \dot{\delta}_1 &= \omega_0 \Delta\omega_1 \\ M_1 \Delta\dot{\omega}_1 &= -D_1 \Delta\omega_1 + P_{\mathrm{mech}1} \\ \tau_1 \dot{E}_1 &= -\tfrac{X_1}{X_1'} E_1 + V_{\mathrm{field}1} \end{aligned} \tag{40}$$

In this case, current phasor $\boldsymbol{I}_1(t)$ is given with the following as an output of generators:

$$|\boldsymbol{I}_1(t)| = \frac{E_1(t)}{X_1'}, \qquad \angle \boldsymbol{I}_1(t) = \delta_1(t) - \frac{\pi}{2}$$

On the other hand, if device 1 is modeled as a constant impedance load, then $|\boldsymbol{I}1(t)|$ is also zero since $|\boldsymbol{V}1(t)|$ is zero as given by Eq. (**??**). For other load models, the situation is generally similar, except that for constant power loads, $|\boldsymbol{I}1(t)|$ becomes infinite, so numerical instability must be taken into account in simulations. Note that the ground fault current phasor flowing from bus 1 to ground can be expressed as

$$\boldsymbol{I}_1'(t) := \boldsymbol{I}_1(t) - \sum_{j=2}^{N} \boldsymbol{Y}_{1j} \boldsymbol{V}_j(t), \qquad t \in [0, t_0)$$

where N is the number of buses.

The value of the ground fault current phasor does not need to be calculated for the purpose of running numerical simulations of power system models. On the other hand, when a generator is connected to the ground fault bus, it is important to calculate the value of the generator's internal state at the time t_0 when the ground fault is cleared, so it is necessary to solve the differential equations in Equation 40. For the initial internal states of each generator at the time of the ground fault occurrence, arbitrary values can be set for any flow state. In this book, appropriate values obtained as a result of the power flow calculation for a steady-state flow condition are used. Note that at the time of occurrence and clearance of the ground fault, the internal states of each generator are continuous, but the voltage and current phasors of each bus change discontinuously.

The calculation of the time response for bus-ground faults can be summarized in the following steps:

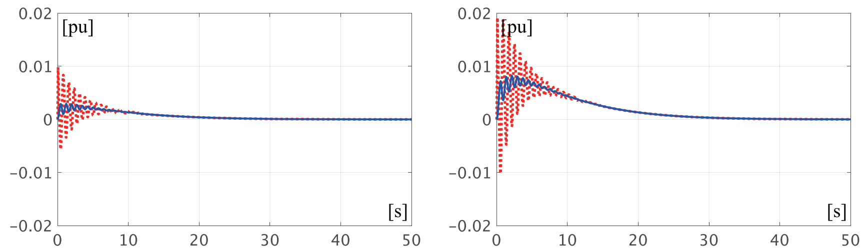
(a) Set the variable values at the steady-state condition obtained from the power flow calculation as the initial values of the internal states of each generator.
(b) Calculate the time evolution using the power system model that excludes the faulted bus and the devices and transmission lines connected to it, for the time interval $t \in [0, t_0)$, where the fault persists.
(c) If a generator is connected to the faulted bus, set the bus voltage phase to zero for $t \in [0, t_0)$, and calculate the time evolution of its internal state.
(d) Set the values of the internal states of each generator at time t_0, and calculate the time evolution of the system after the fault is cleared using the usual power system model with all devices connected.

Let us calculate the time response for bus-ground faults using these steps.

Example 1.5 Time response of power system model to bus bar ground fault

In the same setting as Examples 1.3 and 1.4, let us calculate the time response of the frequency deviation of the bus due to ground fault. Specifically, we set the two steady-state power flow states obtained by power flow calculation as initial values and calculate the time response when a ground fault occurs at bus 1. For comparison, we consider two cases where the time until the fault is cleared is 50 [ms] and 100 [ms], respectively.

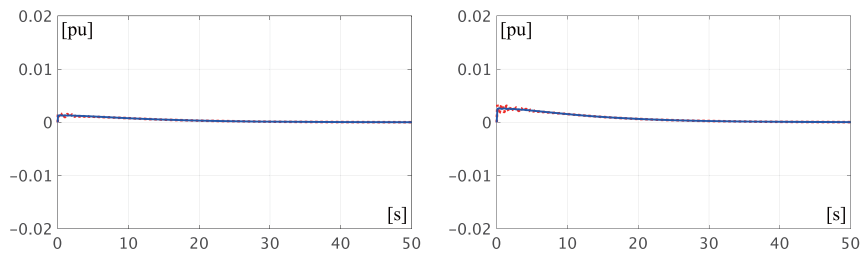
The calculation results are shown in Figures 5 and 6. The solid blue line represents the frequency deviation of generator 1, and the dashed red line represents the frequency deviation of generator 3. From Figure 5, it can be seen that the frequency oscillation of generator 3 is large in the steady-state power flow state shown in Table 1. In particular, a larger oscillation occurs when the duration of the ground fault is 100 [ms]. The reason why the oscillation of generator 3 is larger is that, as shown in Table **??**, the inertia constant of generator 1 is large at 100 [s], while that of generator 3 is small at 12 [s]. In other words, the oscillation of the generator with larger inertia causes the larger oscillation of the generator with smaller inertia.



(a) Ground fault at 50 [ms]

(b) Ground fault at 100 [ms]

Fig. 5 Time response of angular frequency deviation to ground fault
(Steady-state of 1, line type is the same as 2)



(a) Ground fault at 50 [ms]

(b) Ground fault at 100 [ms]

Fig. 6 Time response of angular frequency deviation to ground fault
(Steady-state of 2, line type is the same as 2)

On the other hand, it can be seen from Figure 6 that the frequency oscillation for the same ground fault is small in the steady-state power flow state shown in Table 2. This is because in the steady-state power flow state of Table 1, generator 3 with small inertia supplied most of the active power consumed by the load, while in the steady-state power flow state of Table 2, generator 1 with large inertia supplied most of the active power. In general, synchronous generators have the characteristic that their sensitivity to disturbances increases as the supplied power approaches its maximum value. In the steady-state power flow state of Table 2, the stability of generator 3 with small inertia is relatively high, so the sensitivity of the frequency deviation to the ground fault is low.

From Example 1.5, it can be seen that the stability of the power system model for ground faults varies depending on the steady-state power flow condition (equilibrium point). In particular, in Example 1.2, the steady-state power flow condition in Table 1 was superior from the perspective of transmission loss, while in Example 1.5, the steady-state power flow condition in Table 2 was found to be superior in terms of system stability with respect to ground faults. These examples suggest the importance of exploring desirable equilibrium points, taking into account trade-offs between factors such as economic efficiency and stability.

It should be noted that ground faults may occur at various locations and not only on specific buses or transmission lines, so it is necessary to improve the overall stability of the power system in a meaningful way. Control mechanisms for this purpose are explained in Sections **??** and **??**.

Furthermore, from the simulation results of Examples 1.3 to 1.5, it can be observed that when the internal state settles into a steady state without diverging, ”all generator frequency deviations have converged to the same value.” Similar results are also observed in Example **??**. This synchronization of frequency in the steady-state power flow condition is a universal phenomenon in power system models. Frequency control, which will be discussed in Sections **??** and **??**, is based on the assumption that the frequency will automatically synchronize in the steady-state power flow condition, and control algorithms are constructed accordingly.

5 Synchronization of bus bar voltage in a steady power flow distribution

In this section, we mathematically investigate the frequency synchronization phenomenon observed in Section 4 from the perspective of the graph structure of the power transmission network. Based on the results of Theorem 1.3, we introduce the following definition.

Definition 1.1 (Synchronization of the steady power flow distribution and bus bar voltage)

Consider the power system model in which the devices are coupled by the simultaneous equations of Equation 4. For all buses i, if the following hold for all $t \geq 0$, the power system is said to be in a **steady-state power flow**.

$$\frac{dP_i}{dt}(t) = 0, \quad \frac{dQ_i}{dt}(t) = 0, \quad \frac{d|\boldsymbol{V}_i|}{dt}(t) = 0, \quad \frac{d^2\angle\boldsymbol{V}_i}{dt^2}(t) = 0 \tag{41}$$

[1]

Moreover, if the power system is in a steady-state power flow and Equation 42 holds, the buses i and j are said to be **synchronized** in the steady-state power flow.

$$\frac{d\angle\boldsymbol{V}_i}{dt}(t) = \frac{d\angle\boldsymbol{V}_j}{dt}(t) \tag{42}$$

As stated in Theorem 1.3, the validity of Equation 41 for the generator bus is equivalent to the internal state of the generator and the external input being in steady

---

[1] Please note that the mathematical definition of this “steady power flow distribution” is unique to this book and is not typically used in electrical power system engineering.

state. Moreover, if any selected pair of buses (i, j) are in synchronized in the sense of Definition 1.1, it can be concluded that the angular frequency deviation of all generators converges to the same value. Note that the validity of Equation 41 for any bus applies to the current phase $\boldsymbol{I}_i$.

$$\frac{d|\boldsymbol{I}_i|}{dt}(t) = 0, \qquad \frac{d^2\angle\boldsymbol{I}_i}{dt^2}(t) = 0, \qquad \frac{d\angle\boldsymbol{I}_i}{dt}(t) = \frac{d\angle\boldsymbol{V}_i}{dt}(t)$$

This means that it can be easily verified by the equations:

$$|P_i(t) + \boldsymbol{j}Q_i(t)| = |\boldsymbol{V}_i(t)||\boldsymbol{I}_i(t)|, \qquad \angle(P_i(t) + \boldsymbol{j}Q_i(t)) = \angle\boldsymbol{V}_i(t) - \angle\boldsymbol{I}_i(t)$$

Let $\mathcal{N}_i$ denote the set of adjacent buses connected to bus i via transmission lines. In other words:

$$\mathcal{N}_i := \left\{ j : \boldsymbol{Y}_{ij} \neq 0, \quad j \neq i \right\}$$

Note that bus i is not included in $\mathcal{N}_i$. The number of adjacent buses connected to bus i is denoted by $|\mathcal{N}_i|$ and is called the **degree** of bus i.

The number of adjacent buses connected to bus i via transmission lines is denoted as $|\mathcal{N}_i|$, and this is referred to as the **degree** of bus i, which is equal to $|\mathcal{N}_i|$. In addition, the power system model assumes a steady-state condition, which can be expressed as:

$$\angle\boldsymbol{V}_i(t) = \Omega_i t + \phi_i$$

where, Ω_i and ϕ_i are constant.

Under this assumption, the power balance equation for bus i in Equation 4 is given by:

$$P_i + \boldsymbol{j}Q_i = \sum_{j=1}^{N} \overline{\boldsymbol{Y}}_{ij}|\boldsymbol{V}_i||\boldsymbol{V}_j|e^{\boldsymbol{j}\left\{(\Omega_i-\Omega_j)t+\phi_i-\phi_j\right\}}$$

Here, assuming a steady-state condition, the active and reactive power supplied to bus i, as well as the magnitude of the bus voltage phasor, are all constants. If we denote these quantities as $P_i^\star$, $Q_i^\star$, and $|\boldsymbol{V}_i^\star|$, respectively, then the power balance equation can be transformed into the following:

$$\underbrace{\frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} r_{ij}e^{\boldsymbol{j}(\Omega_{ij}t+\Phi_{ij})}}_{\boldsymbol{C}_i(t)} = z_i \tag{43}$$

where $\Omega_{ij} := \Omega_i - \Omega_j$ is the frequency difference between bus i and bus j. The constants r_{ij}, Φ_{ij}, and z_i are defined as:

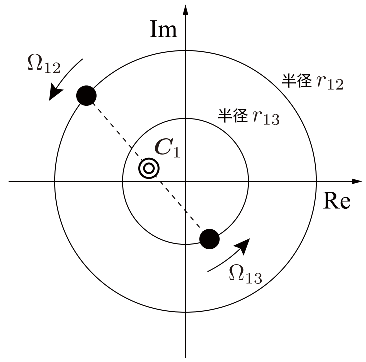


Fig. 7 When bus bar 1 is connected to bus bars 2 and 3

$$r_{ij} := |V_i^{\star}||V_j^{\star}||Y_{ij}|, \qquad \Phi_{ij} := \phi_i - \phi_j - \angle Y_{ij},$$
$$z_i := \frac{\{P_i^{\star} - \mathrm{Re}[Y_{ii}]|V_i^{\star}|^2 + j(Q_i^{\star} + \mathrm{i}[Y_{ii}]|V_i^{\star}|^2)\}}{|\mathcal{N}_i|}$$

In the following, we consider deriving the equation $\Omega_{ij} = 0$ for all $j \in \mathcal{N}_i$, which represents synchronization with adjacent buses, from the Equation 43. Here, the Equation 43 represents that the center of gravity $C_i(t)$ of $|\mathcal{N}_i|$ points that move at a constant speed along the circumference of a circle with radius r_{ij} and initial phase Φ_{ij} and angular velocity Ω_{ij}, is invariant at a point z_i on the complex plane.

Figure 7 illustrates the relationship when bus 1 is connected to buses 2 and 3. Based on this fact, the following result can be obtained.

Lemma 1.2 (Synchronization of busbars derived from power flow equations)
Consider $C_i(t)$ in Equation 43 for real constants r_{ij}, Ω_i, Ω_j, and Φ_{ij}, where $r_{ij} > 0$. If $|\mathcal{N}_i| = 1$, then $C_i(t)$ being a constant independent of t is equivalent to Equation **??**. *If $|\mathcal{N}_i| = 2$, then $C_i(t)$ being a constant independent of t is equivalent to either Equation* **??** *being true, or to:*

$$\Omega_{j_1} = \Omega_{j_2}, \qquad r_{ij_1} = r_{ij_2}, \qquad |\Phi_{ij_1} - \Phi_{ij_2}| = \pi \tag{44}$$

where $\mathcal{N}_i = \{j_1, j_2\}$. If $|\mathcal{N}_i| = 3$, then $C_i(t)$ being a constant independent of t is equivalent to Equation **??** *holding, or to either of the following:*

- *For $\mathcal{N}_i = j_1, j_2, j_3$, Equation 45 holds, where*

$$\Omega j_1 = \Omega j_2 = \Omega j_2, \qquad \sum j \in \mathcal{N}_i r_{ij} e^{j\Phi_{ij}} = 0, \tag{45}$$

- *$\Omega_i = \Omega_{j_3}$ holds for $j_3 \in \mathcal{N}_i$ and Equation 44 holds, where $\mathcal{N}_i \setminus j_3 = j_1, j_2$.*

Proof By applying Lemma 1.3 at the end of the chapter, it can be shown that the cases where $|\mathcal{N}_i| = 1$ and $|\mathcal{N}_i| = 2$ hold. Therefore, we consider the case where $|\mathcal{N}_i| = 3$ below.

For simplicity of notation, let $j \in \{1, 2, 3\}$, and denote r_{ij}, Φ_{ij}, Ω_i, and $\boldsymbol{C}_i$ as r_j, Φ_j, Ω_0, and $\boldsymbol{C}_0$, respectively. First, we consider the case where $\Omega_j \neq \Omega_0$ for all $j \in \{1, 2, 3\}$. From Lemma 1.3, it follows that when Equation **??** holds, Equation 43 is equivalent to

$$\Omega_1 = \Omega_2 = \Omega_3, \qquad \sum j = 1^3 r_j e^{\boldsymbol{j}\Phi_j} = 0$$

which implies Equation 45. However, if $\Omega_1 = \Omega_2 = \Omega_3$, then by the same argument as in the case $|\mathcal{N}_i| = 1$, it follows that $\boldsymbol{C}_i(t)$ is a constant independent of t, and Equation **??** is equivalent to Equation **??**, leading to a contradiction.

Next, we consider the case where Equation **??** does not hold, i.e., there exists $j \in \{1, 2, 3\}$ such that $\Omega_0 = \Omega_j$. Without loss of generality, we may assume that $\Omega_0 = \Omega_3$. In this case, we have

$$\boldsymbol{C}_0(t) = \frac{1}{3}\left\{ r_3 e^{\boldsymbol{j}\Phi_3} \sum_{j=1}^{2} r_j e^{\boldsymbol{j}\{(\Omega_0 - \Omega_j)t + \Phi_j\}} \right\}.$$

Thus, we can discuss the invariance with respect to t in the same way as the case $|\mathcal{N}_i| = 2$. Therefore, the fact that $\boldsymbol{C}_0(t)$ is a constant independent of t is equivalent to Equation **??** or:

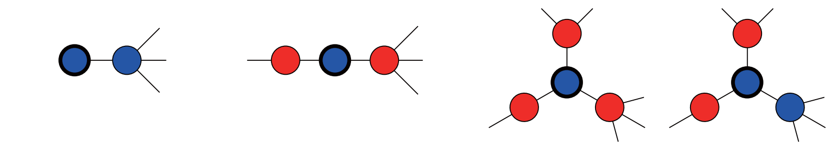
$$\Omega 1 = \Omega 2, \qquad r_1 = r_2, \qquad |\Phi_1 - \Phi_2| = \pi.$$

Therefore, the proposition is proved. □

Lemma 1.2 shows that when the degree of the target bus is 1, i.e., for a bus at the endpoint of a line like in 8(a), it synchronizes with its neighboring bus. When the degree of the target bus (the node indicated by a thick line) is 2, i.e., for a bus on a chain-like path like in 8(b), at least its two neighboring buses synchronize. Furthermore, when the degree of the target bus is 3, i.e., for a bus at a node connected by three power lines like in 8(c), either the three buses except the target bus synchronize, or at least one of the adjacent buses synchronizes with the target bus. This means that situations where only one of the adjacent buses does not synchronize or situations where no pair of buses synchronize do not occur.

Similar analyses can be performed for cases where the degree of the target bus is greater than or equal to 4. However, deriving equivalent synchronization conditions generally becomes complicated due to higher-order equations involving Ω_i and Ω_j, and multiple combinations of some adjacent buses synchronizing while others do not. Nevertheless, it is generally shown that if any $|\mathcal{N}_i| - 1$ adjacent buses out of the $|\mathcal{N}_i|$ buses adjacent to the target bus i synchronize with bus i, then the remaining one also synchronizes.

By combining the conditions for degrees 3 and below as shown in Lemma 1.2, it is possible to show the synchronization of all buses even when there are buses with degrees greater than or equal to 4 in the power grid. For example, it can be shown for a power grid with a tree structure, which is connected and has no cycles, that all buses synchronize.



(a) Bus bar degree 1 (b) Bus bar degree 2 (c) Bus bar degree 3

Fig. 8 Synchronization with adjacent busbars according to the degree of the busbar

It is possible to perform a similar analysis even when the degree of the bus of interest is four or greater. However, because the resulting conditions become higher-order equations in terms of Ω_i and Ω_j, and there are multiple combinations where only certain adjacent buses synchronize, writing down equivalent conditions regarding synchronization generally becomes complicated. However, if any of the $|\mathcal{N}_i| - 1$ adjacent buses to the bus of interest i synchronize, then the remaining one will generally synchronize as well.

By combining the conditions for degrees 3 or lower shown in Lemma 1.2, it is possible to demonstrate synchronization of all buses even when the network contains buses with degree 4 or greater. For example, it can be shown for a power network with a tree structure (i.e., connected with no cycles) that all buses will synchronize.

Theorem 1.4 (Synchronization of Generators in a Tree-Structured Power System) *Consider the power system model where the devices are connected by a network of transmission lines represented by a system of simultaneous equations as shown in Equation 4. When the graph of the power system network has a tree structure, all generators synchronize in steady-state operating conditions.*

Proof We focus on the bus at the endpoint indicated by the bold line in 9(a). Since the degree of the bus is 1, the adjacent bus is synchronized with the endpoint bus. Next, if the adjacent bus to the endpoint is on a chain path, its degree is 2, so at least the two adjacent buses are synchronized. By repeating this process, as shown in 9(a), the synchronization of all buses is demonstrated on the chain path connecting the endpoint and buses of degree 3 or higher.

Similarly, all buses with degree 3 or higher that exist on nodes from another endpoint are synchronized, so all buses on chain paths connecting the buses of the bold nodes in 9 are synchronized. By repeating this discussion, it is demonstrated that all buses in the power system tree are synchronized. □

There is no degree restriction on the buses in the power grid with a tree structure, as stated in Theorem 1.4. Thus, even if there are buses with degree 4 or higher in the power grid, it is sometimes possible to deduce the synchronization of all buses using only the graph structure information. Similarly, the following theorem can be proven:

Fig. 9 Synchronization of bus bars in a power grid with a tree structure

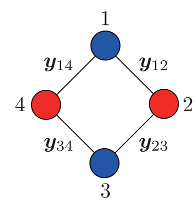


Fig. 10 Synchronization of four busbars in a ring-structured transmission network

Theorem 1.5 (Synchronization of Buses in a Ring-Structured Power Grid) *Consider a power system model in which a group of devices is connected by a system of power transmission lines described by the set of equations in 4. If the power grid has a circular structure and the total number of buses is odd, then all buses will be synchronized in the steady-state power flow.*

Proof If we focus on one bus, the synchronization of the two buses at each end of it is shown. By repeating this process, the synchronization of all buses is shown when the total number of buses is odd. □

Theorem 1.5 demonstrates that for a circular structured power grid with an odd number of transmission lines, all the main lines will synchronize regardless of the admittance values of the power grid. On the other hand, for an even number of transmission lines, it is not possible to conclude the synchronization of all the main lines for a circular structured power grid without additional information such as admittance values.

The following example demonstrates that even when the number of transmission lines is even, the synchronization of all main lines can be inferred with only some information about the admittance values.

Example 1.6 Synchronization of four bus bars in a power grid with a ring structure Let us consider synchronization of bus bars under a steady power flow distribution in a power grid with a ring structure shown in Figure 10. Since the number of bus bars is even, synchronization of bus bars cannot be shown using only the information of the graph structure as in Theorem 1.5. However, as it is shown with red and blue in Figure 10, synchronization of alternating bus bars is shown by Lemma 1.2. Therefore, for all bus bars to show synchronization, all that is necessary is to show that one of the conditions of Equation 44 is not satisfied for at least one bus bar.

Let $\boldsymbol{y}_{ij}$ denote the admittance of the transmission line connecting bus i and bus j. Then, the central condition in Equation 44 for each bus can be expressed as follows:

$$\begin{aligned} |\boldsymbol{V}_2^\star||\boldsymbol{y}_{12}| = |\boldsymbol{V}_4^\star||\boldsymbol{y}_{14}|, \qquad |\boldsymbol{V}_1^\star||\boldsymbol{y}_{12}| = |\boldsymbol{V}_3^\star||\boldsymbol{y}_{23}|, \\ |\boldsymbol{V}_2^\star||\boldsymbol{y}_{23}| = |\boldsymbol{V}_4^\star||\boldsymbol{y}_{34}|, \qquad |\boldsymbol{V}_3^\star||\boldsymbol{y}_{34}| = |\boldsymbol{V}_1^\star||\boldsymbol{y}_{14}| \end{aligned} \tag{46}$$

Expressing this in matrix form, we have:

$$\underbrace{\begin{bmatrix} 0 & |\boldsymbol{y}_{12}| & 0 & -|\boldsymbol{y}_{14}| \\ -|\boldsymbol{y}_{12}| & 0 & |\boldsymbol{y}_{23}| & 0 \\ 0 & -|\boldsymbol{y}_{23}| & 0 & |\boldsymbol{y}_{34}| \\ |\boldsymbol{y}_{14}| & 0 & -|\boldsymbol{y}_{34}| & 0 \end{bmatrix}}_{S} \begin{bmatrix} |\boldsymbol{V}_1^\star| \\ |\boldsymbol{V}_2^\star| \\ |\boldsymbol{V}_3^\star| \\ |\boldsymbol{V}_4^\star| \end{bmatrix} = 0$$

A necessary condition for the existence of a positive vector $(|\boldsymbol{V}_1^\star|, \ldots, |\boldsymbol{V}_4^\star|)$ that satisfies this equation is that the matrix S on the left-hand side is not invertible. However, note that even if S is singular, it does not necessarily mean that there exists a positive vector that is its nullspace. From the sparse structure of the column vectors, it can be seen that S is non-invertible if and only if:

$$|\boldsymbol{y}_{12}||\boldsymbol{y}_{34}| = |\boldsymbol{y}_{14}||\boldsymbol{y}_{23}| \tag{47}$$

Therefore, all buses are synchronized unless the admittance matrix satisfies this condition. It is also shown that if Equation 47 holds, then a necessary and sufficient condition for the existence of a positive vector that satisfies the central condition in Equation 44 is Equation 47.

Next, let us consider the right-hand side conditions in Equation 44 for buses 1 and 3:

$$|\phi_2 - \phi_4 + \angle\boldsymbol{y}_{12} - \angle\boldsymbol{y}_{14}| = \pi, \qquad |\phi_2 - \phi_4 + \angle\boldsymbol{y}_{23} - \angle\boldsymbol{y}_{34}| = \pi$$

Similarly, if we focus on bus bar 2 and bus bar 4, we obtain the following equations:

$$|\phi_1 - \phi_3 + \angle\boldsymbol{y}_{12} - \angle\boldsymbol{y}_{23}| = \pi, \qquad |\phi_1 - \phi_3 + \angle\boldsymbol{y}_{14} - \angle\boldsymbol{y}_{34}| = \pi$$

In general, when the ground capacitance is sufficiently small, the real part of the admittance, which is the conductance component, is non-negative, and the imaginary part, which is the susceptance component, is negative. In other words

$$\angle\boldsymbol{y}_{ij} \in \left[-\frac{\pi}{2}, 0\right]$$

Note that if we satisfy the above conditions, then the necessary and sufficient condition for the existence of $(\phi_1, \ldots, \phi_4)$ that satisfy the above conditions is:

$$\angle \boldsymbol{y}_{12} - \angle \boldsymbol{y}_{14} = \angle \boldsymbol{y}_{23} - \angle \boldsymbol{y}_{34} \tag{48}$$

Therefore, unless the admittance matrix satisfies this condition, all busbars will synchronize.

From the above discussion, it can be seen that the necessary and sufficient conditions for the existence of one or more sets of unsynchronized busbars in steady-state power flow are given by Equations 47 and 48. These two conditions suggest that only the opposite busbars will synchronize with each other when the power transmission network shown in Figure 10 has a specific symmetry with respect to the values of admittance.

From Example 1.6, it can be seen that the condition given by Equation 44 represents a specific symmetry with respect to the values of admittance in the power transmission network. In practical applications, it is a universal fact that all busbars will synchronize in steady-state power flow for sparse transmission networks with low degree of each busbar, unless there is a specific symmetry in the graph structure. In fact, to the best of the authors' knowledge, it has been numerically verified that all busbars synchronize in steady-state power flow for any power system model parameters set to realistic values.

6 Implementation of Power Flow Calculation

Let us consider building a numerical simulation environment for analyzing and controlling power systems. In order to correctly perform numerical simulations of large and complex power system models, it is useful to employ object-oriented thinking and programming techniques, which allow us to describe them as a group of modules separated by their functions. In this section, we introduce an implementation method of power flow calculation using Matlab, based on this approach. For basic syntax of Matlab, please refer to [**?**] or [**?**]. For object-oriented programming with Matlab, please refer to [**?**], and for general principles of object-oriented programming, please refer to [**?**] or [**?**].

6.1 Solving Algebraic Equations

In order to perform power flow calculation, it is necessary to solve the algebraic equations in Equation 10. In this section, we will show how to use MATLAB to explore the solutions of algebraic equations using a simple example.

Example 1.7 Searching for a solution for algebraic equations

Consider the following set of algebraic equations:

$$x^2 - y = 0\, x^2 + y^2 - 2 \qquad = 0 \tag{49a}$$

We want to find the values of (x, y) that satisfy these equations using numerical computations. Note that the solutions are $(x, y) = (-1, 1)$ and $(1, 1)$. MATLAB provides a convenient command for solving algebraic equations called `fsolve` from the `optimization toolbox`. To use this command, we need to implement a function $f(x, y)$ such that the algebraic equations in Equation 49 can be written in the form $f(x, y) = 0$. We implement the function $f(x, y)$ in Program 1.1.

```
function out = func_ex1(x_in)

x = x_in(1);
y = x_in(2);

out = zeros(2, 1);
out(1) = x^2 - y;
out(2) = x^2 + y^2 - 2;

end
```

Program 1.1 func_ex1.m

Next, using this function, we will write a program to explore the solution of the algebraic equation by executing `fsolve`. The program for this is shown in Program 1.2.

```
options = optimoptions('fsolve', 'Display', 'iter');
x0 = [0.1; 0.5];
x_sol = fsolve(@func_ex1, x0, options)
```

Program 1.2 main_ex1.m

Here, the first line's `options` sets the options to be given to `fsolve` to solve the algebraic equation, and in this example, the optimization process is displayed. Also, the second line `x0 = [0.1; 0.5];` represents the initial values for numerically searching for solutions. When this program is executed, the following results are obtained.

execution1.1

```
  Iteration  Func-count     f(x)     (Omitted)
      0          3          3.2677
      1          6        0.282554
(Omitted)
      4         15     3.60447e-14

The equation is solved
(Omitted)
  x_sol =
      1.0000
      1.0000
```

From this result, it can be seen that as the iteration progresses, the value of $f(x)$ decreases, and a solution is found where $f(x)$ is almost 0. The solution found through this search is $(1, 1)$, which is indeed a solution to the algebraic equation. However, it should be noted that numerical solution methods using `fsolve` do not always find all the solutions, and only one of the solutions may be obtained.

Next, let's show the case where the other solution is obtained. When the second line of Program 1.2 is changed to `x0 = [-0.1;0.5];` and executed, the following result is obtained.

execution1.2

```
 Iteration  Func-count     f(x)
     0          3          3.2677
     1          6        0.282554
(Omitted)
     4         15     3.60447e-14

The equation has been solved.
(Omitted)
x_sol =

   -1.0000
    1.0000
```

In this execution result, the solution $(-1, 1)$ is obtained. In numerical solution methods for nonlinear algebraic equations, it is important to note that different solutions can be obtained depending on the initial values. As a reference, let's check the execution result when the initial values set in the second line of 1.2 are set as `x0 = [-0.1; -0.1]`:

execution1.3

```
  Iteration  Func-count     f(x)    (Omitted)
     0          3          2.8125
     1          6         1.89068
(Omitted)
    29         62            1.75

No solution found.
(Omitted)
x_sol =

    0.0000
   -1.2247
```

As seen in this example, depending on the initial values given, even for equations with solutions, correct solutions may not be obtained. In practice, it is important to provide initial values close to the desired solution.

6.2 Simple implementation of power flow calculation

Next, we will introduce a simple implementation method for power flow calculation.

Example 1.8 Implementation method for power flow calculations

Consider performing power flow calculations on the power system model consisting of 3 buses in Example 1.2. Since power flow calculation is also a type of calculation that solves algebraic equations, it is necessary to implement the $f(x)$ part of $f(x) = 0$ as in Example 1.7.

If the admittance values of the two transmission lines are set to the values in Equation 12, the steady-state power flow state that is consistent with the data sheet in 2(a) is as follows.

$$\left(|\boldsymbol{I}_1^\star|, \angle\boldsymbol{I}_1^\star, |\boldsymbol{V}_1^\star|, \angle\boldsymbol{V}_1^\star, |\boldsymbol{I}_2^\star|, \angle\boldsymbol{I}_2^\star, |\boldsymbol{V}_2^\star|, \angle\boldsymbol{V}_2^\star, |\boldsymbol{I}_3^\star|, \angle\boldsymbol{I}_3^\star, |\boldsymbol{V}_3^\star|, \angle\boldsymbol{V}_3^\star\right)$$

Given the admittance matrix $\boldsymbol{Y}$, the current is uniquely determined by the voltage, so the power flow calculation is essentially a process of determining the set of voltages for the buses. Therefore, we can take the real and imaginary parts of the voltage phasors for all buses and arrange them as variables x in the algebraic equations. [2] In this case, the voltage phasors, current phasors, active and reactive power for each bus are all functions of x. Denoting them as $\hat{\boldsymbol{V}}_i(x)$, $\hat{\boldsymbol{I}}_i(x)$, $\hat{P}_i(x)$, and $\hat{Q}_i(x)$, respectively, the algebraic equations to be solved are given by:

$$\begin{aligned}
|\boldsymbol{V}_1^\star| - |\hat{\boldsymbol{V}}_1(x)| &= 0 \\
\angle\boldsymbol{V}_1^\star - \angle\hat{\boldsymbol{V}}_1(x) &= 0 \\
P_2^\star - \hat{P}_2(x) &= 0 \\
Q_2^\star - \hat{Q}_2(x) &= 0 \\
P_3^\star - \hat{P}_3(x) &= 0 \\
|\boldsymbol{V}_3^\star| - |\hat{\boldsymbol{V}}_3(x)| &= 0
\end{aligned}$$

If we implement the functions on the left side, it becomes a program 1.3.

Though this function has five output arguments, with `fsolve`, only the first output argument is used. The remaining arguments were added to confirm the result.

```
function [out, Vhat, Ihat, Phat, Qhat] = func_ex2(x)
% x: [Real(V1), Imag(V1),
%     Real(V2), Imag(V2),
%     Real(V3), Imag(V3)]';

y12 = 1.3652 - 11.6040j;
y23 = -10.5107j;
Y = [y12, -y12, 0;
    -y12, y12+y23, -y23;
    0, -y23, y23];

```

---

[2] It is also possible to take the magnitude and phase angle of the voltage phasors as x.

```
V1abs = 2;
V1angle = 0;

P2 = -3;
Q2 = 0;

P3 = 0.5;
V3abs = 2;

V1hat = x(1) + 1j*x(2);
V2hat = x(3) + 1j*x(4);
V3hat = x(5) + 1j*x(6);

Vhat = [V1hat; V2hat; V3hat];

Ihat = Y*Vhat;
PQhat = Vhat.*conj(Ihat);
Phat = real(PQhat);
Qhat = imag(PQhat);

out = [V1abs-abs(V1hat); V1angle-angle(V1hat);
    P2-Phat(2); Q2-Qhat(2);
    P3-Phat(3); V3abs-abs(V3hat)];
end
```

Program 1.3 func_ex2.m

As mentioned in Example 1.7, when numerically searching for solutions to algebraic equations, the choice of initial values is important. In power flow calculations, an initial value called the **flat start** is often used. In the flat start, the initial value of x is set by

$$|\hat{\boldsymbol{V}}_i(x)| = 1, \qquad \angle \boldsymbol{V}_i(x) = 0$$

for all buses. This corresponds to setting the real and imaginary parts of $\boldsymbol{V}_i$ to 1 and 0, respectively. The specific program for the flat start is as follows.

```
1  x0 = [1; 0; 1; 0; 1; 0];
2  options = optimoptions('fsolve', 'Display', 'iter');
3  x_sol = fsolve(@func_ex2, x0, options);
4
5  [~, V, I, P, Q] = func_ex2(x_sol);
6  Vabs = abs(V);
7  Vangle = angle(V);
8  display('Vabs:'), display(Vabs')
9  display('Vangle:'), display(Vangle')
10 display('P:'), display(P')
11 display('Q:'), display(Q')
```

Program 1.4 main_ex2.m

The initial value for the flat start is set in the first line of Program 1.4. Then, on the fifth line, the voltage phasors, current phasors, active power, and reactive power are calculated using the solution of the algebraic equation. Finally, on the sixth line and below, the absolute value, phase angle, active power, and reactive power of the voltage phasors are displayed. The result of running Program 1.4 is shown. Note that this result matches the values shown in Table 2(b).

execution1.4

```
 Iteration  Func-count     f(x)     (Omitted)
     0          7           11.25
     1         14         3.28327
(Omitted)
     5         42     4.19428e-28

The equation has been solved.
(Omitted)
Vabs:
    2.0000    1.9918    2.0000

Vangle:
    0.0000   -0.0538   -0.0419

P:
    2.5158   -3.0000    0.5000

Q:
   -0.0347    0.0000    0.1759
```

6.3 Implementation of Power Flow Calculation using Separated Modules

In the previous section, a simple implementation of power flow calculation was described. However, in this implementation, it is necessary to rewrite the function corresponding to Program 1.3 every time the admittance matrix is changed. Also, if the number or type of buses is changed, the entire program needs to be rewritten. In this section, to enable power flow calculation for large-scale power systems or implementation by multiple people, we consider dividing the program into a group of modules. This leads to a well-structured implementation.

First, let's consider modularizing the implementation of the admittance matrix in Program 1.3.

Example 1.9 Separation of Implementation of Admittance Matrix

Let's separate the part of Program 1.3 that specifies the admittance matrix. For this purpose, we can make the variable `Y` an input argument of the function. For example, a modification like Program 1.5 can be considered.

```
function [out, Vhat, Ihat, Phat, Qhat] = func_ex3(x, Y)
(Same as lines 12 through 34 of program 3-3)
end
```

Program 1.5 function_ex2.m

However, since `fsolve` requires a "function with only the parameters to be optimized as arguments", the modified Program 1.5 cannot be used directly. This problem can be solved by using a technique called **currying** of functions. Specifically, program 1.5 can be modified as follows.

```
x0 = [1; 0; 1; 0; 1; 0];
options = optimoptions('fsolve', 'Display', 'iter');

y12 = 1.3652 - 11.6040j;
y23 = -10.5107j;
Y = [y12, -y12, 0;
    -y12, y12+y23, -y23;
    0, -y23, y23];

func_curried = @(x) func_ex3(x, Y);

x_sol = fsolve(func_curried, x0, options);
[~, V, I, P, Q] = func_curried(x_sol);
(Same as line 6 and below in program 3-4)
```

Program 1.6 main_ex3.m

The "`@(x) expression of x`" used creates a function that returns the value of the expression with argument `x`. This type of function that is created without a name is called an **anonymous function**. When an anonymous function contains variables that are not included in its arguments, such as `Y` in this case, it uses a constant from the workspace. Therefore, it is possible to generate a new function by fixing some of the variables among multiple arguments and leaving only the remaining variables as arguments. This technique is called currying. Using currying, a function that takes any number of arguments can be used as a function with a specified number of arguments. This allows for the separation of functionality without using global variables.

Next, let us consider the part of the Program 1.3 that calculates the constraint conditions for each busbar. In the Program 1.3, the constraint conditions for each busbar are directly written, so it is necessary to rewrite the program every time the number of busbars is changed. Furthermore, there is an implicit case distinction depending on the type of busbar, and if the types of busbars are increased, it is necessary to modify the processing of the case distinction.

To write a transparent program without using case distinctions, it is useful to use the concept of **polymorphism** in object-oriented programming. In the following example, we explain the concept of polymorphism through an implementation example.

Example 1.10 Implementation of Power Flow Calculation using Polymorphism

In power flow calculation, different constraint conditions are set depending on the type of busbar, but all busbars have a common property of "having constraint conditions.” By implementing this common property as a ”method with a common name,” a simple program can be created. To understand this, let’s define classes corresponding to slack busbars, generator busbars, and load busbars in Program 1.7 as shown in Program 1.10.

```
classdef bus_slack

  properties
    Vabs
    Vangle
  end

  methods
    function obj = bus_slack(Vabs, Vangle)
      obj.Vabs = Vabs;
      obj.Vangle = Vangle;
    end

    function out = get_constraint(obj, Vr, Vi, P, Q)
      Vabs = norm([Vr; Vi]);
      Vangle = atan2(Vi, Vr);
      out = [Vabs-obj.Vabs; Vangle-obj.Vangle];
    end
  end
end
```

Program 1.7 bus_slack.m

```
classdef bus_generator

  properties
    P
    Vabs
  end

  methods
    function obj = bus_generator(P, Vabs)
      obj.P = P;
      obj.Vabs = Vabs;
    end

    function out = get_constraint(obj, Vr, Vi, P, Q)
      Vabs = norm([Vr; Vi]);
      out = [obj.P-P; Vabs-obj.Vabs];
    end
  end
end
```

Program 1.8 bus_generator.m

```
classdef bus_load

  properties
    P
```

```
    Q
  end

  methods
    function obj = bus_load(P, Q)
      obj.P = P;
      obj.Q = Q;
    end

    function out = get_constraint(obj, Vr, Vi, P, Q)
      out = [P-obj.P; Q-obj.Q];
    end
  end
end
```

Program 1.9 bus_load.m

The classes defined here, `bus_slack`, `bus_generator`, and `bus_load`, all have a method called `get_constraint` with a common input and output. In this case, a program that uses these classes can calculate the appropriate constraint conditions by calling `get_constraint` without being aware of what type of busbar it is. Using this idea, Program 1.3 can be rewritten as follows.

```
function [out, V, I, P, Q] = func_power_flow(x, Y, a_bus)

V = x(1:2:end) + 1j*x(2:2:end);
I = Y*V;
PQhat = V.*conj(I);
P = real(PQhat);
Q = imag(PQhat);

out_cell = cell(numel(a_bus), 1);

for i = 1:numel(a_bus)
  bus = a_bus{i};
  out_cell{i} = bus.get_constraint(...
    real(V(i)), imag(V(i)), P(i), Q(i));
end
out = vertcat(out_cell{:});

end
```

Program 1.10 func_power_flow.m

The program 1.10 assumes that the cell array, which is the set of mother lines, is input as `a_bus`. An example of how to use this function is shown in the program 1.11.

```
(Same as lines 1-8 in program 3-6)

a_bus = cell(3, 1);
a_bus{1} = bus_slack(2, 0);
a_bus{2} = bus_load(-3, 0);
a_bus{3} = bus_generator(0.5, 2);

```

```
8  func_curried = @(x) func_power_flow(x, Y, a_bus);
9
10 x_sol = fsolve(func_curried, x0, options);
11
12 (Same as Program 3-4, line 6 and following)
```

Program 1.11 main_ex4.m

In the program 1.11, the slack bus bar, load bus bar, and generator bus bar are defined respectively and assigned to the cell array `a_bus`. At this time, in the program1.10, `get_constraint` is executed for each bus bar assigned to `a_bus`. Since `get_constraint` is implemented for each bus line type, the appropriate processing is performed without having to describe the case. Such different processing for similar program calls is called polymorphism, and is one of the important concepts in object-oriented programming.

In an implementation using polymorphism like Program 1.10, it has the advantage of easy adaptability to changes in the number and configuration of buses. Specifically, by changing the number and definition of buses in lines 3-6 of Program 1.11, Program 1.10 can be used without modification.

Moreover, adding new types of buses is also easy. Program 1.10 assumes only that a method called `get_constraint` with appropriate inputs and outputs exists for the bus. In other words, if it has `get_constraint`, it is defined as a bus. This is called ”duck typing”, a concept that comes from ”if it walks like a duck and quacks like a duck, then it must be a duck.” This allows module designers to define a new bus class by implementing only `get_constraint` without having to pay attention to the internal processing of Program 1.10. Thus, this approach clarifies the scope of implementation for module designers.

The above has allowed us to prospectively implement a program for calculating the tidal currents given an admittance matrix. Furthermore, let us implement the computation of the admittance matrix by dividing it into a group of modules.

Example 1.11 Calculation of Admittance Matrix using Object-oriented Programming

The admittance matrix is generally defined by multiple power transmission lines. Therefore, we create a class that represents a power transmission line and consider using a collection of its instances. A power transmission line connects two busbars, so it is necessary to identify the numbers of these busbars. Additionally, regardless of the type of power transmission line model, the admittance matrix $\boldsymbol{Y}_{\text{branch}}$ satisfying

$$\begin{bmatrix} \boldsymbol{I}_{\text{from}} \\ \boldsymbol{I}_{\text{to}} \end{bmatrix} = \boldsymbol{Y}_{\text{branch}} \begin{bmatrix} \boldsymbol{V}_{\text{from}} \\ \boldsymbol{V}_{\text{to}} \end{bmatrix} \tag{50}$$

is generally determined. Here, $\boldsymbol{V}_{\text{from}}$ and $\boldsymbol{V}_{\text{to}}$ are the voltages of the busbars at both ends of the power transmission line, and $\boldsymbol{I}_{\text{from}}$ and $\boldsymbol{I}_{\text{to}}$ are the currents flowing from the busbars to the power transmission line at both ends. For example, in the simple power transmission line discussed in Example **??**, we have

$$\boldsymbol{Y}_{\text{branch}} = \begin{bmatrix} y & -y \\ -y & y \end{bmatrix}$$

In summary, it is sufficient to return the information of the two connected buses and the admittance matrix Y_{branch} to define a transmission line. An example implementation of a transmission line based on this idea is as follows.

```
classdef branch

  properties
    y
    from
    to
  end

  methods
    function obj = branch(from, to, y)
      obj.from = from;
      obj.to = to;
      obj.y = y;
    end

    function Y = get_admittance_matrix(obj)
      y = obj.y;
      Y = [y, -y;
        -y,  y];
    end
  end
end
```

Program 1.12 branch.m

The function to compute the admittance matrix using this transmission line class is implemented as follows

```
function Y = get_admittance_matrix(n_bus, a_branch)

  Y = zeros(n_bus, n_bus);

  for i = 1:numel(a_branch)
    br = a_branch{i};
    Y_branch = br.get_admittance_matrix();
    Y([br.from, br.to], [br.from, br.to]) =...
      Y([br.from, br.to], [br.from, br.to]) + Y_branch;
  end

end
```

Program 1.13 get_admittance_matrix.m

The input argument `n_bus` is the number of bus lines. Also, `a_branch` is a cell array containing the transmission lines. This program can be used as follows:

```
a_branch = cell(2, 1);
a_branch{1} = branch(1, 2, 1.3652-11.6040j);
```

```
a_branch{2} = branch(2, 3, -10.5107j);

Y = get_admittance_matrix(3, a_branch)
```

Program 1.14 main_admittance_matrix.m

In the program1.13, we only assume that the transmission line has the data `from` and `to` and that it has the method `get_admittance_matrix` and the method `get_admittance_matrix`. Therefore, when considering other transmission line models such as π-type models When considering other transmission line models, such as π-type models, the variables `to` and `from` and the method `get_admittance_matrix` are also assumed to be used. The program program 1.13 can be used without any modification by simply implementing a class with variables named `to`, `from` and a method named `get_admittance_matrix`.

From the above, the program to calculate tidal currents can be summarized as follows.

Example 1.12 Implementation result of power flow calculation

The program implemented in the above example can be organized as a function for performing power flow calculation, which is shown in Program 1.15.

```
function [V, I, P, Q] = calculate_power_flow(a_bus, a_branch)

  n_bus = numel(a_bus);
  Y = get_admittance_matrix(n_bus, a_branch);

  func_curried = @(x) func_power_flow(x, Y, a_bus);

  x0 = kron(ones(n_bus, 1), [1; 0]);
  options = optimoptions('fsolve', 'Display', 'iter');
  x_sol = fsolve(func_curried, x0, options);

  [~, V, I, P, Q] = func_curried(x_sol);

end
```

Program 1.15 calculate_power_flow.m

This program takes a set of buses and a set of transmission lines as inputs and performs power flow calculations. It can be used as shown in Program 1.16.

```
a_bus = cell(3, 1);
a_bus{1} = bus_slack(2, 0);
a_bus{2} = bus_load(-3, 0);
a_bus{3} = bus_generator(0.5, 2);

a_branch = cell(2, 1);
a_branch{1} = branch(1, 2, 1.3652-11.6040j);
a_branch{2} = branch(2, 3, -10.5107j);

[V, I, P, Q] = calculate_power_flow(a_bus, a_branch);

```

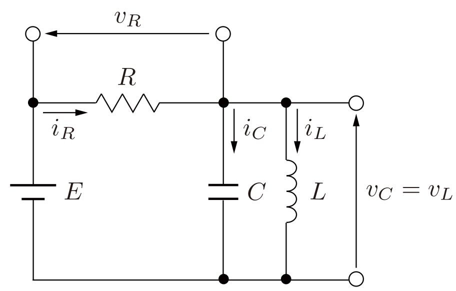


Fig. 11 Differential Algebraic Equations Example: LC Parallel Circuit

```
12 display('Vabs:'), display(abs(V)')
13 display('Vangle:'), display(angle(V)')
14 display('P:'), display(P')
15 display('Q:'), display(Q')
```

Program 1.16 main_power_flow.m

If it is necessary to perform power flow calculations for different power grids, other programs can be used without modification by changing only the definitions of buses and transmission lines in lines 1 to 8 of Program 1.16. Also, if a new type of bus or transmission line is required, a new class implementing the data and methods with predetermined names can be implemented. Therefore, compared to a simple implementation like Example 1.8, this program has high readability and extensibility.

---

7 Implementation method for time response calculation of power system models

In this section, we will explain how to numerically calculate the time response of a power system model using MATLAB . To calculate the time response of a power system model, it is necessary to solve a system of differential-algebraic equations that combine differential equations representing the dynamic characteristics of the equipment and algebraic equations representing the power flow. First, let's look at an example of a simple differential-algebraic equation system and its solution method.

Example 1.13 Numerical solution method for a simple differential-algebraic equation system

Let us perform a numerical simulation for the case where R, L, C, and E are all equal to 1 in the simple electrical circuit shown in Figure 11. The dynamic elements of this circuit are the coil L and the capacitor C, and their differential equations are given by:

$$L\dot{i}_L = v_L \tag{51a}$$
$$C\dot{v}_C = i_C \tag{51b}$$

where the initial values are set to $i_L(0) = 0$ and $v_C(0) = 0$. Also, the algebraic equations given by Ohm's law and Kirchhoff's law are:

$$v_R = Ri_R \tag{52a}$$
$$i_R = i_L + i_C \tag{52b}$$
$$v_L = v_C \tag{52c}$$
$$E = v_C + v_R \tag{52d}$$

Using the algebraic equations in (52), redundant variables such as v_R and i_R can be eliminated, resulting in an equivalent system of ordinary differential equations. This operation corresponds to Kron reduction, and the resulting system of ordinary differential equations is given by:

$$\dot{i}_L = \frac{1}{L}v_C \tag{53a}$$
$$\dot{v}_C = \frac{1}{RC}(E - v_C) - \frac{1}{C}i_L \tag{53b}$$

First, let's write a program to solve this system of ordinary differential equations. The `ode45` solver in MATLAB is commonly used for solving systems of ordinary differential equations. By implementing the function $f(t,x)$ for the ordinary differential equation $\dot{x} = f(t,x)$, the solver can compute the solution of the system of ordinary differential equations. Implementing the right-hand side of (53), the program becomes as shown in Program 1.17.

```
function dx = func_RLC_ode(x, R, C, L, E)

iL = x(1);
vC = x(2);

diL = vC/L;
dvC = (E-vC)/R/C - iL/C;

dx = [diL; dvC];

end
```

Program 1.17 func_RLC_ode.m

As a result, when `ode45` is executed and the ordinary differential equation system is solved, the Program 1.18 becomes:

```
R = 1;
```

```
L = 1;
C = 1;
E = 1;

func = @(t, x) func_RLC_ode(x, R, C, L, E);
x0 = [0; 0];
tspan = [0 30];

[t, x] = ode45(func, tspan, x0);

plot(t, x)
```

Program 1.18 main_RLC_ode.m

The output variable `x` in the program 1.18 is a matrix in which the time series of i_L and v_C are arranged vertically. Therefore, note that the time series of other variables such as i_C and i_R need to be additionally calculated using the algebraic equation of the Equation 52.

Next, consider directly finding the solution of the system of differential algebra equations without going through Kron reduction. In this case, since the physical algebraic equation can be written down as it is, there is an advantage that the description is easy even if the system becomes complicated. One of the commands that can solve the system of differential algebra with MATLAB is `ode15s`. The target is a system of differential algebraic equations described in the following format.

$$M\dot{x} = f(t, x) \tag{54}$$

Applying the expression (51) and the expression (52):

$$\underbrace{\begin{bmatrix} 0 & L & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} \dot{i}_R \\ \dot{i}_L \\ \dot{i}_C \\ \dot{v}_R \\ \dot{v}_L \\ \dot{v}_C \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} v_L \\ i_C \\ v_R - Ri_R \\ i_R - i_L - i_C \\ v_L - v_C \\ E - v_C - v_R \end{bmatrix}}_{f(t,x)}$$

Note that the order in which the elements on the 3rd to 6th lines on the right side are placed is arbitrary. Implementing this right-hand side results in a Program 1.19.

```
function dx = func_RLC_dae(x, R, C, L, E)

  iR = x(1);
  iL = x(2);
  iC = x(3);
  vR = x(4);
  vL = x(5);
  vC = x(6);

```

```
  diL = vL;
  dvC = iC;

  con1 = vC-vL;
  con2 = E-vC-vR;
  con3 = iR-(iC+iL);
  con4 = vR-iR*R;

  dx = [diL; dvC; con1; con2; con3; con4];
end
```

Program 1.19 func_RLC_dae.m

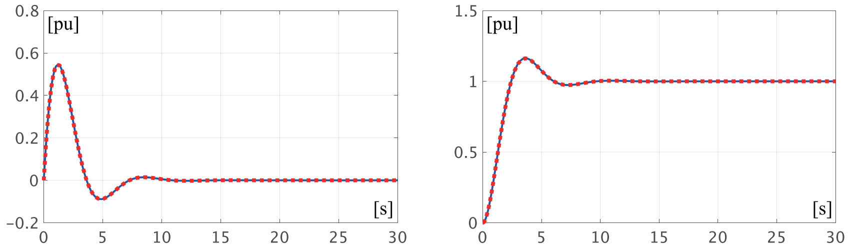
This function can be used to solve a system of differential algebraic equations such as in Program 1.20.

```
1  function dx = func_RLC_dae(x, R, C, L, E)
2
3    iR = x(1);
4    iL = x(2);
5    iC = x(3);
6    vR = x(4);
7    vL = x(5);
8    vC = x(6);
9
10   diL = vL;
11   dvC = iC;
12
13   con1 = vC-vL;
14   con2 = E-vC-vR;
15   con3 = iR-(iC+iL);
16   con4 = vR-iR*R;
17
18   dx = [diL; dvC; con1; con2; con3; con4];
19 end
```

Program 1.20 main_RLC_dae.m

In this program, M of the expression 54 is set in the option on the 13th line. Also, the initial value of the state is set in the 10th line, but only the state of the differential equation, that is, the 2nd and 6th elements, has meaning. The state of the algebraic equation does not need to be calculated and set by the user himself because the value that satisfies the equation is automatically searched by `ode15s`. In fact, the algebraic equation is not satisfied by the initial value with all the elements as 0, but the solution of the system of differential algebraic equations is calculated without any problem. The solution `x` obtained in the 15th line is a matrix in which the time series of $v_R, v_L, v_C, i_R, i_L, i_C$ are arranged, and all of them are different from the case of equivalent conversion to the ordinary differential equation system. The time series of the variables of is calculated at once.

Figure 12 shows the time response of i_L and v_C when the differential algebraic equation system is solved directly and when it is converted to an ordinary differential equation system. In each figure, two lines are displayed: one is the solution obtained



(a) Time response of i_L.

(b) Time response of v_C.

Fig. 12 Time response of LC parallel circuits (Blue solid line: ode45, red dashed line: ode15s)

by `ode45` (blue solid line) and the other is the solution obtained by `ode15s` (red dashed line). It is clear that the two solutions are equal.

7.1 Simple implementation of time response calculations for power system models

The following example describes how to implement a numerical simulation of a power system model.

Example 1.14 Simple implementation of power system simulation

Let us implement a program to numerically compute the time response of the power system model discussed in the Example 1.3. To describe this system, we need the differential equations in Equation **??** for bus bars 1 and 3, to which the generators are connected. Also, the algebraic equation of Equation **??** holds. A constant-impedance load model is connected to bus bar 2 and the algebraic equation of Equation **??** is satisfied. Furthermore, the algebraic system of equations of Kirchhoff's law, Equation 1, holds for the entire system at any given time.

Here, we consider describing the vertically arranged vector x of δ_1, $\Delta\omega_1$, E_1, δ_3, $\Delta\omega_3$, E_3, $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, $\boldsymbol{V}_3$, $\boldsymbol{I}_1$, $\boldsymbol{I}_2$, and $\boldsymbol{I}_3$ in the form of:

$$M\dot{x} = f(t, x)$$

An implementation example of the right-hand side function $f(t, x)$ is shown in Program 1.21.

```
function dx = func_simulation_3bus(x, Y, parameter)

  delta1 = x(1);
  omega1 = x(2);
  E1 = x(3);
```

```
delta3 = x(4);
omega3 = x(5);
E3 = x(6);
V1 = x(7)  + 1j*x(8);
V2 = x(9)  + 1j*x(10);
V3 = x(11) + 1j*x(12);
I1 = x(13) + 1j*x(14);
I2 = x(15) + 1j*x(16);
I3 = x(17) + 1j*x(18);

omega0 = parameter.omega0;

X1 = parameter.X1;
X1_prime = parameter.X1_prime;
M1 = parameter.M1;
D1 = parameter.D1;
tau1 = parameter.tau1;
Pmech1 = parameter.Pmech1;
Vfield1 = parameter.Vfield1;

z2 = parameter.z2;

X3 = parameter.X3;
X3_prime = parameter.X3_prime;
M3 = parameter.M3;
D3 = parameter.D3;
tau3 = parameter.tau3;
Pmech3 = parameter.Pmech3;
Vfield3 = parameter.Vfield3;

P1 = real(V1*conj(I1));
P3 = real(V3*conj(I3));

dx1 = [omega0 * omega1;
    (-D1*omega1-P1+Pmech1)/M1;
    (-X1/X1_prime*E1+...
    (X1/X1_prime-1)*abs(V1)*cos(delta1-angle(V1))+Vfield1)/tau1
  ];

dx3 = [omega0 * omega3;
    (-D3*omega3-P3+Pmech3)/M3;
    (-X3/X3_prime*E3+...
    (X3/X3_prime-1)*abs(V3)*cos(delta3-angle(V3))+Vfield3)/tau3
  ];

con1 = I1-(E1*exp(1j*delta1)-V1)/(1j*X1_prime);
con2 = V2+z2*I2;
con3 = I3-(E3*exp(1j*delta3)-V3)/(1j*X3_prime);

con_network = [I1; I2; I3] - Y*[V1; V2; V3];

dx = [dx1; dx3;real(con1); imag(con1); real(con2);
    imag(con2); real(con3); imag(con3);
    real(con_network); imag(con_network)];
```

```

end
```

Program 1.21 func_simulation_3bus.m

In this program, the voltage and current phasors included in `x` are expressed by arranging their real and imaginary parts. Additionally, the `parameter` structure is used to specify variables collectively. Solving the differential algebraic equation system using Program 1.21 results in Program 1.22.

```
a_branch = cell(2, 1);
a_branch{1} = branch(1, 2, 1.3652-11.6040j);
a_branch{2} = branch(2, 3, -10.5107j);
Y = get_admittance_matrix(3, a_branch);

parameter = struct();

parameter.M1 = 100;
parameter.D1 = 10;
parameter.tau1 = 5.14;
parameter.X1 = 1.569;
parameter.X1_prime = 0.936;
parameter.Pmech1 = 2.5158;
parameter.Vfield1 = 2.7038;

parameter.z2 = 1.3224;

parameter.M3 = 12;
parameter.D3 = 10;
parameter.tau3 = 8.97;
parameter.X3 = 1.220;
parameter.X3_prime = 0.667;
parameter.Pmech3 = 0.5000;
parameter.Vfield3 = 2.1250;

parameter.omega0 = 60*2*pi;

x0 = [0.5357 + pi/6; 0; 2.3069 + 0.1;...
    0.0390; 0; 2.0654; zeros(12, 1)];
M = blkdiag(eye(6), zeros(12, 12));

tspan = [0 50];

options = odeset('Mass', M);
func = @(t, x) func_simulation_3bus(x, Y, parameter);
[t, x] = ode15s(func, tspan, x0, options);

plot(t, x(:, [2, 5]))
```

Program 1.22 main_simulation_simple.m

In Program 1.22, the initial value response of the power system model is calculated, and the angular frequency deviations of the two generators are plotted.

7.2 Implementation method for time response calculation using a group of partitioned modules

In this section, we explain how to separate the program described in the previous section into functions and modify it into a highly extensible program.

Example 1.15 Modularization of generators and loads The program 1.21 consists of two steps: dividing the input `x` into the state variables, voltage and current of each device, and computing differential and algebraic equations. Let us consider how to write a program that executes these two steps in a clear view.

First, in order to properly partition the variable `x`, we need to know the number of states of each device. Also, for all devices, the time derivative of the state $\frac{dx}{dt}$ in the differential equation and $f(x)$ in the algebraic equation $f(x) = 0$ is computed. Using the concept of duck typing, if a device has functions to return the number of states, perform time differentiation, and compute algebraic equations, it can be defined as a device. If we implement a generator and a load as a device with these functions, then we have a Program 1.23 and a Program 1.24.

```
classdef generator < handle

  properties
    omega0
    X
    X_prime
    M
    D
    tau
    Pmech
    Vfield
  end

  methods
    function obj = generator(omega0, M, D, tau,...
        X, X_prime, Pmech, Vfield)

      obj.omega0 = omega0;
      obj.X = X;
      obj.X_prime = X_prime;
      obj.M = M;
      obj.D = D;
      obj.tau = tau;
      obj.Pmech = Pmech;
      obj.Vfield = Vfield;
    end

    function nx = get_nx(obj)
      nx = 3;
    end

    function [dx, con] = get_dx_constraint(obj, x, V, I)
      delta = x(1);
```

```
      omega = x(2);
      E = x(3);
      P = real(V*conj(I));

      Pmech = obj.Pmech;
      Vfield = obj.Vfield;

      X = obj.X;
      X_prime = obj.X_prime;
      D = obj.D;
      M = obj.M;
      tau = obj.tau;

      omega0 = obj.omega0;

      dE = (-X/X_prime*E+...
        (X/X_prime-1)*abs(V)*cos(delta-angle(V))...
        +Vfield)/tau;
      dx = [omega0 * omega;
        (-D*omega-P+Pmech)/M;
        dE];
      con = I-(E*exp(1j*delta)-V)/(1j*X_prime);
      con = [real(con); imag(con)];
    end
  end

end
```

Program 1.23 generator.m

```
classdef load_impedance < handle

  properties
    z
  end

  methods
    function obj = load_impedance(z)
      obj.z = z;
    end

    function nx = get_nx(obj)
      nx = 0;
    end

    function [dx, con] = get_dx_constraint(obj, x, V, I)
      dx = [];
      z = obj.z;
      con = V+z*I;
      con = [real(con); imag(con)];
    end
  end

end
```

Program 1.24 load_impedance.m

In these programs, the method `get_nx` returns the number of states and `get_dx_constraint` returns the time derivative of the state and the constraint conditions. With the device defined in this way, the Program1.21 can be rewritten as Program1.25.

```
1  function out = func_simulation(t, x, Y, a_component)
2
3    n_component = numel(a_component);
4    x_split = cell(n_component, 1);
5    V = zeros(n_component, 1);
6    I = zeros(n_component, 1);
7
8    idx = 0;
9    for k = 1:n_component
10     nx = a_component{k}.get_nx();
11     x_split{k} = x(idx+(1:nx));
12     idx = idx + nx;
13   end
14
15   for k = 1:n_component
16     V(k) = x(idx+1) + x(idx+2)*1j;
17     idx = idx + 2;
18   end
19
20   for k = 1:n_component
21     I(k) = x(idx+1) + x(idx+2)*1j;
22     idx = idx + 2;
23   end
24
25   dx = cell(n_component, 1);
26   con = cell(n_component, 1);
27
28   for k = 1:n_component
29     component = a_component{k};
30     xk = x_split{k};
31     Vk = V(k);
32     Ik = I(k);
33     [dx{k}, con{k}] = component.get_dx_constraint(xk, Vk, Ik);
34   end
35
36   con_network = I - Y*V;
37
38   out = vertcat(dx{:});
39   out = [out; vertcat(con{:})];
40   out = [out; real(con_network); imag(con_network)];
41
42 end
```

Program 1.25 func_simulation.m

In the Program1.25, it is assumed that the cell array of the device is input to `a_component`. In addition, the variable `x` is divided in lines 9 through 23. At this time, by obtaining the number of device states in line 10, the division can be performed appropriately. In lines 28 to 34, the time derivative of the states and the constraints are computed. In line 33 of the program, we call `get_dx_constraint`, which is implemented using polymorphism. In line 38 of the Program,`vertcat(dx{:})` vertically combines all the elements of the cell array `dx`. That is, it is equivalent to `[dx{1}; dx{2}; ... ; dx{end}]`.

Let us consider running a numerical simulation using Program 1.25. Specifically, the numerical simulation part of the Program1.22 can be summarized as follows:

```
1  function [t, x, V, I] = ...
2    simulate_power_system(a_component, Y, x0, tspan)
3  
4    n_component = numel(a_component);
5    a_nx = zeros(n_component, 1);
6    for k = 1:n_component
7      component = a_component{k};
8      a_nx(k) = component.get_nx();
9    end
10   nx = sum(a_nx);
11 
12   M = blkdiag(eye(nx), zeros(n_component*4));
13   options = odeset('Mass', M, 'RelTol', 1e-6);
14 
15   y0 = [x0(:); zeros(4*n_component, 1)];
16 
17   [t, y] = ode15s(@(t, x) func_simulation(t, x, Y, a_component)
       ,...
18     tspan, y0, options);
19 
20 
21   x = cell(n_component, 1);
22   V = zeros(numel(t), n_component);
23   I = zeros(numel(t), n_component);
24 
25   idx = 0;
26 
27   for k = 1:n_component
28     x{k} = y(:, idx+(1:a_nx(k)));
29     Vk = y(:, nx+2*(k-1)+(1:2));
30     V(:, k) = Vk(:, 1) + 1j*Vk(:, 2);
31     Ik = y(:, nx+2*n_component+2*(k-1)+(1:2));
32     I(:, k) = Ik(:, 1) + 1j*Ik(:, 2);
33     idx = idx + a_nx(k);
34   end
35 
36 end
```

Program 1.26 simulate_power_system.m

In the Program1.26, lines 5 to 9 define the matrix`M`— using the number of states obtained from each device. In line 17, the differential algebraic equations are solved using the program 1.25. In lines 27 to 34, the result is divided into the state variables of each device, the voltage phasors and current phasors of each bus bar, and returned.

Furthermore, rewriting the Program1.22 results in the Program1.27.

```
a_branch = cell(2, 1);
a_branch{1} = branch(1, 2, 1.3652-11.6040j);
a_branch{2} = branch(2, 3, -10.5107j);

Y = get_admittance_matrix(3, a_branch);

gen1 = generator(60*2*pi, 100, 10, 5.14,...
  1.569, 0.936, 2.5158, 2.7038);

load2 = load_impedance(1.3224);

gen3 = generator(60*2*pi, 12, 10, 8.97,...
  1.220, 0.667, 0.5000, 2.1250);

a_component = {gen1; load2; gen3};

x0 = [0.5357 + pi/6; 0; 2.3069 + 0.1; 0.0390; 0; 2.0654];
tspan = [0 50];

[t, x, V, I] = simulate_power_system(a_component, Y, x0, tspan);

plot(t, [x{1}(:, 2), x{3}(:, 2)])
```

Program 1.27 main_simulation_3bus.m

In the Program1.27, an array `a_component` of devices representing generators and loads is created and the function `simulate_power_system` defined in the Program 1.26 is executed.

When changing the structure of the power transmission network or the devices to be connected in numerical calculations of time response, it is sufficient to change `a_branch` and `a_component` in program 1.27, and it is not necessary to modify programs from 1.23 to 1.26. Moreover, if using other devices than generators and loads in programs 1.23 and 1.24, it is enough to implement a class with `get_nx` and `get_dx_constraint` and use it.

Thanks to this modularization, the resulting program is more readable and extensible compared to the implementation in Example 1.14.

In the execution results of this program, the state of each device is returned in the cell array `x`. Let's confirm that the elements of `x` form a matrix consistent with the number of state variables for each device. Here, the device connected to bus 2 is a load, and the number of states is 0, so `x{2}` is an empty matrix with 0 columns. The program plots the second element of the states of the generators connected to bus 1 and bus 3, i.e., the angular frequency deviation.

In Example 1.15, it is possible to implement time-response calculations of the power system model with clear visibility by modularizing the generator and load modules. However, in Program 1.27, it is necessary to calculate in advance the values of the external input of the generator and the impedance of the load that match the desired equilibrium point. Since these values depend on the dynamic characteristics of each equipment, there is room for improvement from the perspective of functional separation. To solve this problem, let's consider extending the program described in Example 1.15 and making it functionally more separated.

Example 1.16 Adding a Method to Calculate the Steady State of Generators and Loads

In numerical calculations of the time response of power system models, achieving a steady-state flow that realizes the desired power supply is important. At this time, Equation 19 and Equation 21 must be satisfied for each generator and load, respectively. Since these equations are related to the state equations of each device, it is appropriate to implement them in the classes of `generator` and `load_impedance`. From this perspective, by adding a method to calculate the steady state that achieves the desired power supply to Program 1.23 and Program 1.24, Programs 1.28 and 1.29 are obtained.

```
classdef generator < handle

(Same as lines 3-12 of program 3-23)

  methods

  (Same as lines 15-57 in program 3-23)

    function x_equilibrium = set_equilibrium(obj, V, I, P, Q)
      Vabs = abs(V);
      Vangle = angle(V);

      X = obj.X;
      X_prime = obj.X_prime;

      delta = Vangle + atan(P/(Q+Vabs^2/X_prime));
      E = X_prime/Vabs*sqrt((Q+Vabs^2/X_prime)^2+P^2);

      x_equilibrium = [delta; 0; E];

      obj.Pmech = P;
      obj.Vfield = X*E/X_prime ...
        - (X/X_prime-1)*Vabs*cos(delta-Vangle);
    end

  end

end
```

Program 1.28 generator.m

```
classdef load_impedance < handle

(Same as lines 3-5 in program 3-24)

  methods

```

```
  (Same as lines 8 through 21 in Program 3-24)

    function x_equilibrium = set_equilibrium(obj, V, I, P, Q)
      x_equilibrium = [];
      obj.z = -V/I;
    end

  end

end
```

Program 1.29 load_impedance.m

In these programs, `set_equilibrium` is a method that takes the voltage phase `V`, current phase `I`, active power `P`, and reactive power `Q` of the bus in the desired steady state as input and returns the steady state value of the state `x_equilibrium`. Theoretically, it is sufficient to provide only `V` and `P` without providing `I`, but `I` is also provided for convenience. Using the parameters obtained from the power flow calculation, a simulation similar to Example 1.15 can be described as follows.

```
a_bus = cell(3, 1);
a_bus{1} = bus_slack(2, 0);
a_bus{2} = bus_load(-3, 0);
a_bus{3} = bus_generator(0.5, 2);

a_branch = cell(2, 1);
a_branch{1} = branch(1, 2, 1.3652-11.6040j);
a_branch{2} = branch(2, 3, -10.5107j);

gen1 = generator(60*2*pi, 100, 10, 5.14, 1.569, 0.936, [], []);
load2 = load_impedance([]);
gen3 = generator(60*2*pi, 12, 10, 8.97, 1.220, 0.667, [], []);

a_component = {gen1; load2; gen3};

[V, I, P, Q] = calculate_power_flow(a_bus, a_branch);
Y = get_admittance_matrix(3, a_branch);

x_equilibrium = cell(numel(a_component), 1);
for k=1:numel(a_component)
   x_equilibrium{k} =...
     a_component{k}.set_equilibrium(V(k), I(k), P(k), Q(k));
end

x0 = vertcat(x_equilibrium{:});
x0(1) = x0(1) + pi/6;
x0(3) = x0(3) + 0.1;
tspan = [0 50];

[t, x, V, I] = simulate_power_system(a_component, Y, x0, tspan);

plot(t, [x{1}(:, 2), x{3}(:, 2)])
```

Program 1.30 main_simulation_3bus_equilibrium.m

In Program 1.30, the process of defining the power system model using the classes for the bus, transmission line, generator, and load, and performing power flow calculations to obtain the time response is made clear. Users can execute numerical simulations without paying attention to the internal dynamic characteristics by simply specifying the physical constants of each equipment. In this example, it is assumed that each equipment model has the method `set_equilibrium`. This corresponds to changing the definition of equipment in terms of duck typing.

In the above examples, numerical calculations for the initial value response of the power system model were discussed. Next, we will describe the implementation method for time response calculations for ground faults.

Example 1.17 Numerical calculation of time response to ground fault In order to calculate the response to a ground fault, the voltage at the bus bar where the fault occurred should be fixed at 0 for a certain time, as described in section 4.3. This can be implemented as follows by modifying some constraints in line 36 of the Program 1.25.

```
function out = func_simulation(x, Y, a_component, bus_fault)

  (Same as lines 3-34 of program 3-25)

  con_network = I - Y*V;
  con_network(bus_fault) = V(bus_fault);

  (Same as lines 38 through 40 in program 3-25)

end
```

Program 1.31 func_simulation.m

In this program, an input argument has been added, and it is assumed that the number of the bus bar where the ground fault occurs is assigned to `bus_fault`. The Program 1.26 is changed to correspond to this change, resulting in Program 1.32.

```
function [t, x, V, I] = simulate_power_system(a_component, ...
  Y, x0, tspan,bus_fault, tspan_fault)

  if nargin < 5
    bus_fault = [];
  end

  if nargin < 6
    tspan_fault = [0, 0];
  end

  (Same as lines 4-15 of program 3-26)

  if isempty(bus_fault)
    [t, y] = ode15s(...
      @(t, x) func_simulation(t, x, Y, a_component, []),...
```

```
17       tspan, y0, options);
18   else
19     [t1, y1] = ode15s(...
20       @(t, x) func_simulation(t, x, Y, a_component, bus_fault)
     ,...
21     tspan_fault, y0, options);
22 
23     [t2, y2] = ode15s(...
24       @(t, x) func_simulation(t, x, Y, a_component, []),...
25     [tspan_fault(2), tspan(2)], y1(end, :), options);
26 
27     t = [t1; t2];
28     y = [y1; y2];
29   end
30 
31   (Same as lines 21 through 34 of program 3-26)
32 
33 end
```

Program 1.32 simulate_power_system.m

In this program, the bus bar and time interval where the ground fault occurs are added to the input arguments. However, lines 4 through 10 set default values if these values are not entered, so the program can be used without specifying a ground fault as in the Program 1.27. Such a property that allows past programs to be used in newer versions is called **backward compatibility**.

In the Program 1.32, a ground fault is specified in lines 19 to 21, and the numerical simulation after the ground fault is cleared is performed in lines 23 to 25. Note that the initial value of the state in the numerical simulation after the ground fault is cleared is the final value of the time response calculation during the ground fault. If a program that numerically simulates the time response to a ground fault is written using this program, it will be Program1.33.

```
(Same as lines 1 through 23 of program 3-30)

x0 = vertcat(x_equilibrium{:});
tspan = [0 50];

fault_bus = 1;
fault_tspan = [0, 50e-3];

[t, x, V, I] = simulate_power_system(a_component, Y, x0, tspan
    ,...
  fault_bus, fault_tspan);

plot(t, [x{1}(:, 2), x{3}(:, 2)])
```

Program 1.33 main_simulation_3bus_fault.m

Execution of this program yields results equivalent to 6a.

Finally, the calculation of the time response to the input signal is described in the follwing example.

Example 1.18 Numerical computation of time response to input signals

In the examples we have dealt with so far, we consider performing numerical simulations that vary the mechanical input P_{mech} of the generator or the size of the load, as in Example 4.2. To do this, we need to modify the program to consider external inputs.

In order to consider external inputs, it is necessary that the number of inputs received by each device is explicitly stated in the device definitions. Therefore, we add the requirement that each device definition must have a method that returns the number of inputs.

Furthermore, we modify Program 1.28 and 1.29 to reflect external inputs in the calculation of state variable time derivatives and constraint conditions. These modifications result in Program 1.34 and Program 1.35.

```
classdef generator < handle

  (Same as lines 3-12 of program 3-23)

  methods

  (Same as lines 15-57 in program 3-23)

    function nu = get_nu(obj)
      nx = 1;
    end

    function [dx, con] = get_dx_constraint(obj, x, V, I, u)

  (Same as lines 33 through 36 of program 3-23)

      Pmech = obj.Pmech + u;

  (Same as lines 39-56 in program 3-23)

    end

  (Same as lines 9 through 24 of program 3-28)

  end
end
```

Program 1.34 generator.m

```
classdef load_impedance < handle

  (Same as lines 3-5 in program 3-24)

  methods

  (Same as lines 8 through 21 in Program 3-24)

    function nu = get_nu(obj)
      nu = 2;
    end
```

```

    function [dx, con] = get_dx_constraint(obj, x, V, I, u)
      dx = [];
      z = real(obj.z)*(1+u(1)) + 1j*imag(obj.z)*(1+u(2));
      con = V+z*I;
      con = [real(con); imag(con)];
    end

  (Same as lines 9 through 12 in program 3-29)

  end

end
```

Program 1.35 load_impedance.m

In these programs, the method `get_nu` returns the number of inputs that can be received, and `get_dx_constraint` has been modified to appropriately handle the input `u`.

In this case, the input to the generator represents the increment of P_{mech}, while the input to the load represents the change in the real and imaginary parts of the impedance.

By using these devices, it is possible to calculate input responses. Modifying Program 1.31 and 1.32 to accommodate input responses results in Program 1.36 and 1.37.

```
function out = func_simulation(t, x, Y, a_component,...
  bus_fault, U, bus_U)

  (Same as lines 3-26 of program 3-25)

  for k = 1:n_component

  (Same as lines 29-32 of program 3-25)

    if ismember(k, bus_U)
      uk = U{bus_U==k}(t);
    else
      uk = zeros(component.get_nu(), 1);
    end

    [dx{k}, con{k}] = component.get_dx_constraint(xk, Vk, Ik, uk);

  end

  (Same as lines 5-6 in program 3-31)

  (Same as lines 38 through 40 in program 3-25)

end
```

Program 1.36 func_simulation.m

```

function [t, x, V, I] = simulate_power_system(a_component, ...
  Y, x0, tspan,bus_fault, tspan_fault, U, bus_U)

  (Same as lines 4 through 10 in program 3-32)

  if nargin < 7
      U = {};
      bus_U = [];
  end

  (Same as lines 4-15 of program 3-26)

  if isempty(bus_fault)
    func = @(t, x) func_simulation(t, x, Y, a_component,...
      [], U, bus_U);
    [t, y] = ode15s(func, tspan, y0, options);
  else
    func = @(t, x) func_simulation(t, x, Y, a_component,...
      bus_fault, U, bus_U);
    [t1, y1] = ode15s(func, tspan_fault, y0, options);

    func = @(t, x) func_simulation(t, x, Y, a_component,...
      [], U, bus_U);
    [t2, y2] = ode15s(func, ...
      [tspan_fault(2), tspan(2)], y1(end, :), options);

    t = [t1; t2];
    y = [y1; y2];
  end
  (Same as lines 21 through 34 of program 3-26)

end
```

Program 1.37 simulate_power_system.m

In these programs, the input arguments `U` and `bus_U` have been added. Here, `bus_U` is assumed to be assigned the number of the bus that specifies the input signal. In addition, `U` is a cell array with the number of elements specified by the bus. Each element is a function that takes the time `t` as an argument and returns the input signal. Note that in line 11 of Program 1.36, the element corresponding to the `k`-th bus is selected from `U`.

An example program that uses the modified functions to perform simulations is shown in Program 1.38.

```
  (Same as lines 1 through 25 of program 3-30)

tspan = [0 50];

bus_U = [1; 2];
U = {@(t) 0; @(t) [0.05*t/50; 0.05*t/50]};

[t, x, V, I] = simulate_power_system(a_component, Y, x0,...
    tspan, [], [], U, bus_U);
```

```
10
11 plot(t, [x{1}(:, 2), x{3}(:, 2)])
```

Program 1.38 main_simulation_3bus_input.m

In Program 1.38, input signals are defined in lines 6 and 7. In this program, the load impedance is increased by 5seconds. The input signal for bus 1 always returns 0, so it has no meaning, but it was added to demonstrate how to specify inputs. In the case of bus 3, no input is specified, so the input is automatically set to 0.

With these changes, time response calculations can be performed for external inputs. Since these changes have been made with backward compatibility in mind, by appropriately modifying Program 1.38, initial value response calculations, ground fault calculations, and time response calculations for various power systems can be performed. Furthermore, the other programs in the program group can be used without any modifications. This is the advantage of implementing the programs as a group of modules.

Mathematical Appendix

Lemma 1.3 *For real constants r_i, ω_i, and phi_i.*

$$\boldsymbol{C}_n(t) := \sum_{i=1}^{n} r_i e^{\boldsymbol{j}(\omega_i t+\phi_i)}$$

However, $r_i > 0$ and $\phi_i \in [0, 2\pi)$. In this case, the necessary and sufficient condition for $\boldsymbol{C}_1$ to be a constant independent of t is $\omega_1 = 0$. Also, the necessary and sufficient condition for $\boldsymbol{C}_2$ to be a constant independent of t is $\omega_1 = \omega_2 = 0$. Besides that,

$$\omega_1 = \omega_2, \qquad r_1 = r_2, \qquad |\phi_2 - \phi_1| = \pi$$

Furthermore, when ω_1, ω_2, and ω_3 are all non-zero, the necessary and sufficient condition for $\boldsymbol{C}_3$ to be a constant independent of t is

$$\omega_1 = \omega_2 = \omega_3, \qquad \sum_{i=1}^{3} r_i e^{\boldsymbol{j}\phi_i} = 0$$

Proof First, by setting the derivative of $\boldsymbol{C}_1$ with respect to t to 0:

$$r_1\omega_1 e^{\boldsymbol{j}(\omega_1 t+\phi_1)} = 0$$

Therefore, the necessary and sufficient condition for $\boldsymbol{C}_1$ to be a constant independent of t is that $\omega_1 = 0$. Next, by setting the derivative of $\boldsymbol{C}_2$ with respect to t to 0:

$$r_1\omega_1 e^{\boldsymbol{j}(\omega_1 t+\phi_1)} + r_2\omega_2 e^{\boldsymbol{j}(\omega_2 t+\phi_2)} = 0 \tag{55}$$

Multiply both sides by $e^{-\boldsymbol{j}(\omega_1 t+\phi_1)}$ and further differentiate by t.

$$r_2\omega_2(\omega_2 - \omega_1)e^{\boldsymbol{j}\{((\omega_2-\omega_1)t+\phi_2-\phi_1)\}} = 0$$

This is equivalent to $\omega_2(\omega_2 - \omega_1) = 0$. Thus, $omega_1 = \omega_2$ is obtained. In particular, if $omega_1 = \omega_2 = 0$, then the equation55 is satisfied for any r_1, r_2, ϕ_1, ϕ_1. Also, when ω_1 and ω_2 are non-zero:

$$r_1 e^{\boldsymbol{j}\phi_1} + r_2 e^{\boldsymbol{j}\phi_2} = 0$$

This is equivalent to $r_1 = r_2$ and $|\phi_2 - \phi_1| = \pi$.

Finally, consider $\boldsymbol{C}_3$. As before, by setting the derivative of $\boldsymbol{C}_2$ with respect to t to 0

$$r_1\omega_1 e^{\boldsymbol{j}(\omega_1 t+\phi_1)} + r_2\omega_2 e^{\boldsymbol{j}(\omega_2 t+\phi_2)} + r_3\omega_3 e^{\boldsymbol{j}(\omega_3 t+\phi_3)} = 0 \tag{56}$$

Multiply both sides by $e^{-\boldsymbol{j}(\omega_1 t+\phi_1)}$ and further differentiate by t.

$$r_2\omega_2(\omega_2 - \omega_1)e^{\boldsymbol{j}\{(\omega_2-\omega_1)t+\phi_2-\phi_1)\}} + r_3\omega_3(\omega_3 - \omega_1)e^{\boldsymbol{j}\{(\omega_3-\omega_1)t+\phi_3-\phi_1)\}} = 0$$

Similarly, multiplying both sides by $e^{-\boldsymbol{j}\{(\omega_2-\omega_1)t+\phi_2-\phi_1\}}$ and further differentiating by t yields

$$r_3\omega_3(\omega_3 - \omega_1)(\omega_3 - \omega_2)e^{\boldsymbol{j}\{(\omega_3-\omega_2)t+\phi_3-\phi_2)\}} = 0$$

This is because $\omega_3 \neq 0$.

$$(\omega_3 - \omega_1)(\omega_3 - \omega_2) = 0 \tag{57a}$$

By a similar procedure, from Equation 56

$$(\omega_2 - \omega_1)(\omega_3 - \omega_2) = 0, \qquad (\omega_3 - \omega_1)(\omega_2 - \omega_1) = 0 \tag{57b}$$

Equation57 is equivalent to $\omega_1 = \omega_2 = \omega_3$. In this case, since $omega_1$,ω_2,ω_3 are non-zero, the Equation 56 becomes:

$$\sum_{i=1}^{n} r_i e^{\boldsymbol{j}\phi_i} = 0$$