

MATLAB-Based Programs for Power System Dynamic Analysis

ISMAEL ABDULRAHMAN^{ID}

Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN 38505, USA
e-mail: ikabdulrah42@students.tntech.edu

ABSTRACT This paper presents MATLAB-based programs developed for power system dynamic analysis. The programs can be used for educational purposes and research studies. With the program, time-domain simulation, system linearization, modal analysis, participation factor analysis and visualization, optimal placement of controller, feedback signal selection, frequency response analysis, and control design can be obtained. In addition to solving a power system problem, the package provides a symbolic and vectorized representation of the model in time domain and state space. The package uses the full advantages of MATLAB's powerful solvers for solving non-stiff and stiff problems. Both explicit and implicit techniques are used for solving the differential algebraic equations (DAEs). The synchronous machines are assumed to be equipped with exciter, turbine, and stabilizer. The loads can be modeled as voltage-dependent and independent loads. The test systems used in this paper are the IEEE 9-bus and 68-bus systems, and Texas's 2007-bus synthetic power system. Different types of disturbances are applied to the systems including generator-side and network-side disturbances. The results demonstrate the efficiency and educational values of the package for researchers and students.

INDEX TERMS Dynamic analysis of multi-machine power system, differential algebraic equations, MATLAB, Simulink.

R_s	Stator resistance in pu	K_A	Amplifier gain
X_d	d-axis reactance in pu	K_E	Separate or self-excited constant
X'_d	Transient d-axis reactance in pu	E'_q	q-axis transient internal voltages in pu
X''_d	Sub-transient d-axis reactance in pu	E'_d	d-axis transient internal voltages in pu
X_q	q-axis reactance in pu	E	Internal voltage in pu
X'_q	Transient q-axis reactance in pu	Ψ_{1d}	Damper winding 1d flux linkages in pu
X''_q	Sub-transient q-axis reactance in pu	Ψ_{2q}	Damper winding 2q flux linkages in pu
H	Shaft inertia constant in s	δ	Rotor angle in rad
w_s	Generator synchronous speed in rad per second	w	Angular speed of generator in rad per second
T'_{do}	d-axis time constant associated with E'_q in second	\bar{V}_i	Complex voltage phasor
T''_{do}	d-axis time constant associated with Ψ_{1d} in second	V	Magnitude of bus voltage in pu
T'_{qo}	q-axis time constant associated with E'_d in second	θ	Angle of bus voltage in rad
T''_{qo}	q-axis time constant associated with Ψ_{2q} in second	\bar{I}_{Gi}	Generator complex current phasor
T_A	Amplifier time constant in s	I_{Gi}	Generator current magnitude in pu
T_{CH}	Incremental steam chest time constant in s	γ_i	Generator current angle in rad
T_{SV}	Steam valve time constant in s	I_d	d-axis current in pu
		I_q	q-axis current in pu
		α_{ik}	Angle of admittance Y_{ik} in rad
		E_{fd}	Field voltage in pu
		V_R	Exciter input in pu
		R_F	Rate feedback in pu

T_M	Mechanical input torque in pu
P_{SV}	Steam valve position in pu
P_C	Control power input in pu
R_D	Speed regulation quantity in Hz/pu
V_{ref}	Reference voltage input in pu
S_E	Saturation function
T_{FW}	Frictional windage torques

I. INTRODUCTION

DYNAMIC behavior of an interconnected power system can be studied using time-domain simulation, modal analysis, and frequency response analysis. Mathematically, a multi-machine power system can be modeled in time domain using a set of differential algebraic equations (DAEs). These equations can be solved using either an implicit or explicit method. In the implicit method, the differential equations (DEs) are converted to algebraic equations (AEs) using a numerical approach such as Euler's or trapezoidal method [1, p.165]. The new set of equations are purely algebraic equations which can be solved simultaneously using a numerical technique. In the explicit method, the DEs are solved for the state variables independently using a numerical integration method, whereas the AEs are solved for the algebraic variables in a separate process. The other types of system analysis are modal analysis and frequency response analysis used in the small-signal study. For these techniques, the nonlinear equations representing the power system need to be linearized around an operating point, and the system can put in a state-space form.

Several commercial softwares have been developed for simulating power systems including PSS\E, PowerWorld Simulator, ATP EMTP, etc. These softwares are well-written, fast, computationally efficient, accurate and come with a wide range of electrical components and models. However, commercial software has three major drawbacks [2]–[3]: (1) it is “closed-source” program, that is, the user is not allowed to modify the code sources except for some special cases (2) it is expensive (3) it provides less educational code sources and requires training courses. On the other hand, MATLAB is a high-level language software, less expensive, available at almost all universities, and students and researchers are familiar with it. One of the recognized MATLAB-based and open-access toolbox developed for simulating and linearizing a multi-machine power system is Power System Toolbox (PST) [4]. It has been used by many researchers, and most recently became the base simulation program for the textbook [1]. Another related and widely-used package is Power System Analysis Toolbox (PSAT) [5] which provides a Simulink-based library to build a power system. MatDyn [3] and PowSysGUI [6] are another two software packages developed for power system dynamic analysis.

However, none of the aforementioned open-source programs considers the full advantages of MATLAB's powerful solvers. Instead, they use their own coded programs for solving the DAEs of the system. Therefore, it is not possible

to switch to some other powerful solvers available as built-in functions in MATLAB when the adopted solver fails to achieve the desired function evaluation. For instance, it is found that PSAT fails when the loads are modeled as constant power in a simple test system (9-bus system) for a specific type of disturbance, and it supports only two numerical methods to solve the problem. In fact, PSAT extracts the network data from a single line diagram built-in Simulink, which means we need a huge amount of time and effort to build and simulate such a system. Therefore, PSAT is not used for solving large-scale power systems [5]. While PST supports relatively faster time-domain simulation and linearization, it uses only one numerical solver and its linearization technique is based on a perturbation approximation by calculating the Jacobians numerically. According to the program's manual, “*there is some loss of accuracy, particularly in the zero eigenvalue which is characteristic of most inter-connected power systems*” [7]. Further, in PST, one cannot use the program to show the set of DEAs and its linearized form in a symbolic and analytical way, which is useful for education purposes. The other two open-source programs are relatively newer than PST and PSAT but they are used for simulating only special cases in power systems; for instance, loads must be modeled as constant impedance to eliminate the algebraic equations of the power flows. Further, MatDyn solves the problem only by partition method. Similar to PSAT, PowSysGUI uses Simulink to simulate the system in time-domain.

In addition, these open-source programs are not aimed to be used for some important tasks in power systems such as residue analysis, controller feedback signal selection, optimal location of controller, and tuning parameters needed for control design. On the other hand, MATLAB offers a set of powerful and numerically-stable solvers that can be used for solving non-stiff and stiff DAEs including *ode45*, *ode113*, *ode15s*, *ode23*, *ode23t*, *ode23tb*, and *ode23s*. Several adjustable setting options are available for each solver to overcome some numerical issues in solving stiff problems. MATLAB also provides some other efficient solvers for solving nonlinear algebraic equations including *fsolve* and *lsqnonlin* functions which can be used for solving the problem by the explicit method. Similarly, these functions come with several useful options that can be set for the solvers.

This paper presents a MATLAB-based program to solve a set of DAEs representing a power system by taking the full advantages of MATLAB solvers and settings. The program can be used for simulating small- and large-scale power system. Further, the program provides a generic analytical method for linearizing the power system. A symbolic form of DAEs and its exact linearized equations can also be generated in a state-space form. This is useful for academic studies and educational purposes. Moreover, several useful tasks in power systems can be implemented using the program including participation factor analysis and visualization, optimal locations of controller, controller signal selection, and system parameters tuning. Design of newly installed controllers can

also be obtained without changing the main DAEs structure. The developed program is simple, fast, built following a clear mathematical description in the textbook references and can be used to develop a customized graphical user interface.

The rest of this paper is presented as follows. In section 2, the mathematical model of the system is described. Next, the proposed program is presented in section 3. A time-domain simulation example is given in section 4. Small signal stability is studied in Section 5 including system linearization, modal analysis frequency response analysis, and control design. Section 6 presents the educational value of the program. Finally, the conclusion is introduced in Section VIII.

II. SYSTEM MODELING

The DAEs of a multi-machine power system with m machines and n buses can be written as follows [1]:

A. SYNCHRONOUS GENERATOR

$$T'_{doi} \frac{dE'_{qi}}{dt} = -E'_{qi} - (X_{di} - X'_{di}) \left[I_{di} - \frac{(X'_{di} - X''_{di})}{(X'_{di} - X_{ls})^2} \times (\Psi_{1di} + (X'_{di} - X_{ls}) I_{di} - E'_{qi}) \right] + E_{fd} \quad (1)$$

$$T''_{doi} \frac{d\Psi_{1di}}{dt} = -\Psi_{1di} + E'_{qi} - (X'_{di} - X_{ls}) I_{di} \quad (2)$$

$$T'_{qoi} \frac{dE'_{di}}{dt} = -E'_{di} + (X_{qi} - X'_{qi}) \left[I_{qi} - \frac{(X'_{qi} - X''_{qi})}{(X'_{qi} - X_{ls})^2} \times (\Psi_{2qi} + (X'_{qi} - X_{ls}) I_{qi} + E'_{di}) \right] \quad (3)$$

$$T''_{qoi} \frac{d\Psi_{2qi}}{dt} = -\Psi_{2qi} - E'_{di} - (X'_{qi} - X_{ls}) I_{qi} \quad (4)$$

$$\frac{d\delta_i}{dt} = w_i - w_s \quad (5)$$

$$\begin{aligned} \frac{2H_i}{w_s} \frac{dw_i}{dt} = & T_{Mi} - \frac{X''_{di} - X_{ls}}{(X'_{di} - X_{ls})} E'_{qi} I_{qi} - \frac{(X'_{di} - X''_{di})}{(X'_{di} - X_{ls})} \Psi_{1di} I_{qi} \\ & - \frac{(X''_{qi} - X_{ls})}{(X'_{qi} - X_{ls})} E'_{di} I_{di} + \frac{(X'_{qi} - X''_{qi})}{(X'_{qi} - X_{ls})} \Psi_{2qi} I_{di} \\ & - (X''_{qi} - X'_{di}) I_{di} I_{qi} - T_{FW} \end{aligned} \quad (6)$$

B. EXCITATION SYSTEM (IEEE TYPE I)

$$T_{Ei} \frac{dE_{fdi}}{dt} = -(K_{Ei} + S_{Ei} (E_{fdi})) E_{fdi} + V_{Ri} \quad (7)$$

$$T_{Fi} \frac{dR_{fi}}{dt} = -R_{fi} + \frac{K_{fi}}{T_{fi}} E_{fdi} \quad (8)$$

$$T_{Ai} \frac{dV_{Ri}}{dt} = -V_{Ri} + K_{Ai} R_{fi} - \frac{K_{Ai} K_{fi}}{T_{fi}} E_{fdi} + K_{Ai} (V_{refi} - V_i) \quad (9)$$

C. TURBINE SYSTEM

$$T_{Chi} \frac{dT_{Mi}}{dt} = -T_{Mi} + P_{SVi} \quad (10)$$

$$T_{SVi} \frac{dP_{SVi}}{dt} = -P_{SVi} + P_{Ci} - \frac{1}{R_{Di}} \left(\frac{w_i}{w_s} - 1 \right) \quad (11)$$

D. AEs (STATOR AND POWER FLOW)

$$\begin{aligned} R_{si} I_{di} - X''_{qi} I_{qi} - \frac{(X''_{qi} - X_{ls})}{(X'_{qi} - X_{ls})} E'_{di} + \frac{(X'_{qi} - X''_{qi})}{(X'_{qi} - X_{ls})} \Psi_{2qi} \\ + V_i \sin(\delta_i - \theta_i) = 0 \end{aligned} \quad (12)$$

$$\begin{aligned} 0 = R_{si} I_{qi} + X''_{di} I_{di} - \frac{X''_{di} - X_{ls}}{(X'_{di} - X_{ls})} E'_{qi} - \frac{(X'_{di} - X''_{di})}{(X'_{di} - X_{ls})} \Psi_{1di} \\ + V_i \cos(\delta_i - \theta_i) = 0 \end{aligned} \quad (13)$$

$$\begin{aligned} I_{di} V_i \sin(\delta_i - \theta_i) + I_{qi} V_i \cos(\delta_i - \theta_i) + P_{Li}(V_i) \\ - \sum_{k=1}^n V_i V_k Y_{ik} \cos(\theta_i - \theta_k - \alpha_{ik}) = 0 \end{aligned} \quad (14)$$

$$\begin{aligned} I_{di} V_i \cos(\delta_i - \theta_i) - I_{qi} V_i \sin(\delta_i - \theta_i) + Q_{Li}(V_i) \\ - \sum_{k=1}^n V_i V_k Y_{ik} \sin(\theta_i - \theta_k - \alpha_{ik}) = 0 \end{aligned} \quad (15)$$

$$P_{Li}(V_i) - \sum_{k=1}^n V_i V_k Y_{ik} \cos(\theta_i - \theta_k - \alpha_{ik}) = 0 \quad (16)$$

$$Q_{Li}(V_i) - \sum_{k=1}^n V_i V_k Y_{ik} \sin(\theta_i - \theta_k - \alpha_{ik}) = 0 \quad (17)$$

where $i = 1, \dots, m$ for all state variables, machine inputs, and references. The ik^{th} entries of the admittance matrix are used for the load flow analysis. For a given load power, there are $(n + m)$ complex AEs that are used to solve for V_i , θ_i ($i = 1, \dots, n$), I_{di} , I_{qi} ($i = 1, \dots, m$) in term of the states δ_i , E'_{di} , E'_{qi} , Ψ_{1di} , Ψ_{2qi} ($i = 1, \dots, m$). Also, the outputs from V_R and P_{SV} should have some maximum and minimum values.

E. PSS MODEL

Power system stabilizer (PSS) is a controller used in a power system to provide more damping torque to the machine excitation systems in phase with rotor speed deviation [1]. A conventional PSS consists of three main components (1) phase-lead compensator- $\left(\frac{T_1 s + 1}{T_2 s + 1} \right)^{nb}$ - where T_1 , and T_2 are the compensator time constants, and nb is the number of compensation blocks that can have a value of 1, 2, or more (2) washout high-pass filter- $\left(\frac{T_w s}{T_w s + 1} \right)$ - where T_w is the washout time constant (3) gain (k_{pss}) [1]. If T_1 is greater than T_2 , the compensator is called a phase-lead compensator, whereas it is called a phase-lag compensator if T_1 is smaller than T_2 . Usually, the ranges of T_1 and T_2 are 0.1 – 1s and 0.01 – 0.1s, respectively, but they could have some other values near these ranges [1]. The objective of the optional washout high-pass filter is to pass the input signal at high frequencies (the gain of washout transfer function becomes a unity) and wash any

small deviation in the input (the washout magnitude becomes zero). Tuning the controller parameters will be discussed in Section 5.6.

F. LOAD MODEL

In practice, loads are a combination of constant power, constant current, and constant impedance. The following equation is used to represent the loads [8]:

$$P_{Li} = P_{L0} \left(\frac{V_i}{V_0} \right)^\alpha \quad Q_{Li} = Q_{L0} \left(\frac{V_i}{V_0} \right)^\beta \quad (18)$$

where P_{Li} , Q_{Li} , and V_{Li} are steady-state values of active power, reactive power, and voltage magnitude obtained from load flow analysis prior to a disturbance, whereas α and β refer to the active and reactive load parameters, respectively. These parameters could have a value of 0, 1, or 2 when the loads are modeled as constant power (voltage-independent), constant current, and constant voltage (voltage-dependent), respectively. For actual data of load parameters obtained from field tests, these parameters might be slightly different as indicated in [8, Table 2].

G. REFERENCE ANGLE

For a dynamic system, there should be a reference for the rotational components. Without a reference, the angle variables show a continuous increase in their plots. A common reference is the center of inertia (COI) defined as below [1]:

$$\delta_{COI} = \frac{\sum_1^m M_i \delta_i}{\sum_1^m M_i}, \quad \omega_{COI} = \frac{\sum_1^m M_i \omega_i}{\sum_1^m M_i} \quad (19)$$

where $M_i = \frac{2 * H_i}{\omega_s}$. In (12)–(17), the angle variables always come as difference pairs; therefore, there is no need to add a reference for the angles since they cancel each other. A satisfactory transformation is to replace the constant ω_s in (5) by the rotational signal ω_{COI} . In this paper, ω_{COI} is used as a reference component for the rotational parts in the system.

H. INITIAL VALUES CALCULATION

The first step in solving a dynamic system is to compute the steady-state values of the algebraic variables and the initial conditions of the state variables and constant inputs. For the algebraic variables such as bus voltage phasors $\bar{V}_i = V_i e^{j\theta_i}$ and generator output powers ($P_{Gi} + jQ_{Gi}$), load flow analysis is carried out using the open-source package MATPOWER [9]. The package comes with several powerful solvers and options for solving a regular and optimal power flow. Also, the initial values of the state variables are required to be calculated from the DAEs by setting the derivative parts to zero. The procedure for computing the initial values can be found in [10].

III. THE PROPOSED DAEs STRUCTURE

The developed program solves the DAEs using implicit and explicit methods. In the implicit method, all the DAEs are put in one file and called by the solver. The result is the time

domain data for the states and algebraic variables. However, this method can be used for only one type of disturbance: a step-change in the inputs (voltage reference or the mechanical input). For the network disturbance, which is more common, such as load change, line outage or a three-phase fault, the built-in *ode* solver does not allow the user to change these parameters once the solver is initiated, and hence the simulation stops at the time of disturbance. To solve this limitation in the implicit approach, the DAEs are separated into two sets of equations: DEs and AEs. The AEs are put as a nested function inside the main function used for solving the DEs. The AEs are solved using one of the MATLAB's solvers (*fsolve*, *lqsnonlin*) whereas the DEs are solved using one of the *ode* solvers, specifically *ode15s* or *ode23t*. All variables and parameters inside the nested function are visible in the main function. Another way to solve this limitation is to separate the solution into three different parts; pre-fault, during-fault, and post-fault. This method will be used in large-scale power systems.

For a three-phase fault at a bus in the system, usually, a reduced form of the admittance matrix ($n - 1$ by $n - 1$) is constructed and used for the fault system. A new set of initial values for the algebraic variables at the time of disturbance in the network side are necessary for solving the algebraic equations during the fault. The state variables do not change incautiously with the network side disturbance, and therefore there is no need to calculate the new initial values for the state variables. However, the proposed method does not require such a reduction to the system or calculation of new initial values. Instead, the faulted system is solved by setting the voltage phasor of the faulted bus to zero and eliminating their corresponding algebraic equations. The original admittance matrix remains unchanged. To speed up the simulation, the initial values for the algebraic variables required by the solver should be updated at each time step. Otherwise, it takes a relatively long time to find a solution to the problem at each time step. This can be achieved by letting the set of algebraic variables global variables inside the main function.

The next two sections describe the vectorized model of the DAEs used in this paper and the symbolic representation adopted by the proposed program.

A. VECTORIZATION MODEL OF DAEs

It is preferred to form the DAEs in a vector form to speed up the simulation by eliminating the for-loops in the system. One challenge to implementing this vectorization is when there is a summation term inside the equation, specifically the power flow algebraic equations. To overcome this challenge, a matrix grid solution is used in this paper which significantly speeds the simulation time. MATLAB has a function for this task called "*ndgrid*". An example is given below to clarify the concept. Suppose that our system consists of nine buses and three generators as in the IEEE 9-bus test system which is widely used in power system dynamic analysis. The algebraic equation (14)–(17) contains summation terms that sum the power flows between the bus in the consideration

and each bus in the system. To vectorize the summation parts of (14)-(17), the following four matrices are generated using the function “*ndgrid*”:

$$a = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix} \quad (20)$$

$$b = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix} \quad (21)$$

$$c = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \end{bmatrix} \quad (22)$$

$$d = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix} \quad (23)$$

The summation terms in (14)–(17) can be replaced with their vectorized forms as follows:

$$\begin{aligned} & \sum_{k=1}^n V_i V_k Y_{ik} \cos(\theta_i - \theta_k - \alpha_{ik}) \\ &= \text{sum}(V(1:m)'. * V(1:n). \\ & \quad * \text{abs}(Y(1:m, 1:n). \\ & \quad * \cos((\theta(a) - \theta(b) - \arg(Y(1:m, 1:n))), 2) \end{aligned} \quad (24)$$

$$\begin{aligned} & \sum_{k=1}^n V_i V_k Y_{ik} \sin(\theta_i - \theta_k - \alpha_{ik}) \\ &= \text{sum}(V(1:m)'. * V(1:n). \\ & \quad * \text{abs}(Y(1:m, 1:n). \\ & \quad * \sin((\theta(a) - \theta(b) - \arg(Y(1:m, 1:n))), 2) \end{aligned} \quad (25)$$

$$\begin{aligned} & \sum_{k=1}^n V_i V_k Y_{ik} \cos(\theta_i - \theta_k - \alpha_{ik}) \\ &= \text{sum}(V(m+1:n)'. * V(1:n). \\ & \quad * \text{abs}(Y(m+1:n, 1:n). \\ & \quad * \cos((\theta(c) - \theta(d) - \arg(Y(m+1:n, 1:n))), 2) \end{aligned} \quad (26)$$

$$\begin{aligned} & \sum_{k=1}^n V_i V_k Y_{ik} \sin(\theta_i - \theta_k - \alpha_{ik}) \\ &= \text{sum}(V(m+1:n)'. * V(1:n). \\ & \quad * \text{abs}(Y(m+1:n, 1:n). \\ & \quad * \sin((\theta(c) - \theta(d) - \arg(Y(m+1:n, 1:n))), 2) \end{aligned} \quad (27)$$

where the MATLAB function “*sum*” with the value “2” refers to summation of elements in each row, the dot “.” refers to an element-by-element matrix multiplication, the comma “,” refers to a matrix transpose, and *abs* and *arg* refer to

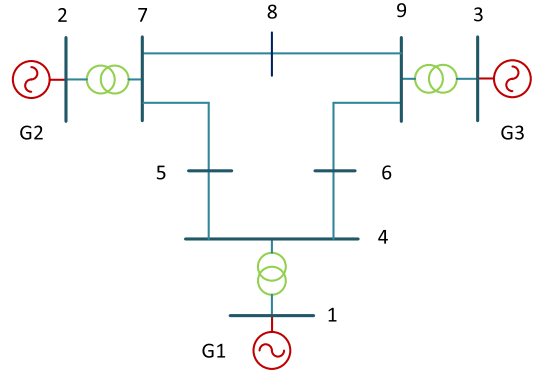


FIGURE 1. Test system.

the magnitudes and angles of bus voltages. The left-hand-side of the above equations are for a standard summation method achieved by for-loops, whereas the right-hand-side of the equations are for the vectorized method adopted in this paper. All other terms of (1)-(17) can be vectorized easily using an element-by-element matrix mathematic operation.

B. SYMBOLIC EXPRESSION OF DAEs AND THE LINEARIZED SYSTEM

The program provides also a symbolic expression of the DAEs and its exact linearized system. This can be achieved using some MATLAB functions from Symbolic Math Toolbox. All system parameters and state and algebraic variables can be written as symbols. Then, one can substitute some or all symbols with their numerical values. It can be used for educational purposes, or to check the set of equations before running the program. Another possible use of this symbolic expression is to set some constant parameters as variables and use MATLAB optimization functions to optimize those specific parameters based on a given pre-defined objective function. Several intelligent optimization techniques can be employed using the program including genetic algorithm and particle swarm optimizations. Unlike PST and Simulink whose linearization is based on a perturbation approximation, this method gives an exact analytical method of linearization.

IV. SIMULATION EXAMPLE

IEEE 9-bus network is used in this section as a test system. The power flow data, transmission lines, and the dynamic data of the generator and excitation system can be found in [1]. The machine sub-transient and turbine parameters for the IEEE 9-bus are estimated from [11]. The single line diagram of the test system is shown in Fig. 1.

A. TIME DOMAIN SIMULATION

1) IMPLICIT SOLUTION OF DAEs

A step-change in the mechanical input or the voltage reference is a type of a generator side disturbance. MATLAB’s ode solvers can efficiently solve a system with such a disturbance without a need of separating DAEs into DEs and AEs. Fig. 2 shows the system response against a step-change in the mechanical power of G1 with a 10% reduction at $t=1s$

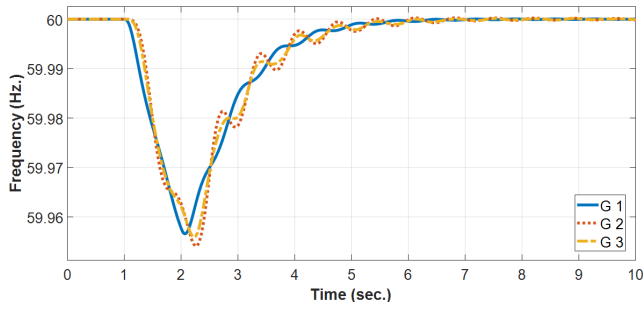


FIGURE 2. System response against generator-side disturbance.

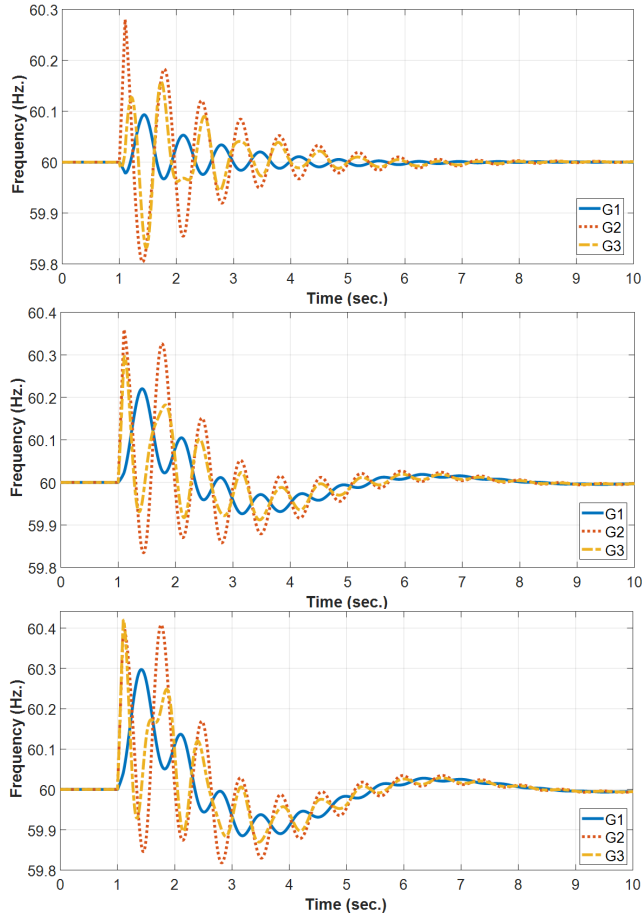


FIGURE 3. System response against network-side disturbance with constant power (top), constant current (middle) and constant impedance (bottom).

and lasts for 1s. The total running time for this example on a regular computer is 1.03s.

2) EXPLICIT SOLUTION OF DAEs

A three-phase fault at bus 5 (load bus) is applied to the system, assuming three scenarios of load mode: constant power, constant current, and constant impedance. The system is simulated with these scenarios and the results are shown in Fig. 3. Since the fault occurred at a load bus, and power is a function of voltage and current components, load power at this bus becomes zero (removed) during the fault. Notably,

two types of oscillations are observed as a response to the fault: individual and group oscillation. The group solicitations are noticed due to including voltage in the load model for the cases of constant current and constant impedance.

We can solve the system using a single-run simulation from $t = 0$ until the simulation time and give a conditional statement to switch between the operation modes. We can also separate the solution into three simulation parts (pre-fault, during-fault, and post-fault). The result is the combination of these three parts of the simulation. Both methods are provided in the program and give similar results. However, for the single-run simulation method, a smaller step size (around $1e-3$) is recommended to capture the dynamic behavior of the system around the switching time, which in turns, slows down the simulation.

V. ANALYSIS BASED ON SYSTEM LINEARIZATION

The nonlinear DAEs are linearized around an operating point $\{x_o, y_o, z_o\}$ and a set of linear DAEs is obtained [12]:

$$\begin{aligned} \dot{x} &= f(x, z, u) \\ 0 &= g(x, z, u) \\ y &= h(x, z, u) \end{aligned} \quad (28)$$

where f , g , and h denote the vectors of differential equations, algebraic equations, and output equations, and x, z, u , and y denote the vector of state variables, algebraic variables, inputs, and outputs, respectively.

Linearizing (28) around an operating point $\{x_o, y_o, z_o\}$ gives the following set of equations:

$$\Delta \dot{x} = \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial z} \Delta z + \frac{\partial f}{\partial u} \Delta u \quad (29)$$

$$0 = \frac{\partial g}{\partial x} \Delta x + \frac{\partial g}{\partial z} \Delta z + \frac{\partial g}{\partial u} \Delta u \quad (30)$$

$$\Delta y = \frac{\partial h}{\partial x} \Delta x + \frac{\partial h}{\partial z} \Delta z + \frac{\partial h}{\partial u} \Delta u \quad (31)$$

By eliminating the algebraic variable vector, the linearized DAEs can be put in the following state-space form:

$$\begin{aligned} \Delta \dot{x} &= A \Delta x + B \Delta u \\ \Delta y &= C \Delta x + D \Delta u \end{aligned} \quad (32)$$

where A (state matrix), B (input matrix), C (output matrix), and D (feed-forward matrix) denote the partial derivative matrices of the original system given in (28) and can be put in the form:

$$\begin{aligned} A &= \left[\frac{\partial f}{\partial x} - \frac{\partial f}{\partial z} \left(\frac{\partial g}{\partial z} \right)^{-1} \frac{\partial g}{\partial x} \right] \\ B &= \left[\frac{\partial f}{\partial u} - \frac{\partial f}{\partial z} \left(\frac{\partial g}{\partial z} \right)^{-1} \frac{\partial g}{\partial u} \right] \\ C &= \left[\frac{\partial h}{\partial x} - \frac{\partial h}{\partial z} \left(\frac{\partial g}{\partial z} \right)^{-1} \frac{\partial g}{\partial x} \right] \\ D &= \left[\frac{\partial h}{\partial u} - \frac{\partial h}{\partial z} \left(\frac{\partial g}{\partial z} \right)^{-1} \frac{\partial g}{\partial u} \right] \end{aligned} \quad (33)$$

TABLE 1. Numerical results of modal analysis (lightly-damped modes).

Eigenvalue	Damping Ratio (%)	Frequency (Hz.)
$-0.65117 \mp 9.0749i$	7.1571	1.4443
$-1.1563 \mp 14.905i$	7.7347	2.3721

With the Symbolic Math Toolbox installed, the system is linearized around an operating point using the above four matrices. The result is a set of symbolic expressions of the state-space form of the system.

A. EIGENVALUE, EIGENVECTOR, DAMPING RATIO, AND FREQUENCY

After linearizing the system and calculating the matrices A, B, C, and D, the function “*eig*” in MATLAB is used to compute the eigenvalues ($\lambda = \sigma \pm \omega$) of the system and the right-left eigenvectors from these matrices, where σ is the real part of the eigenvalue (λ) and ω is the imaginary part of λ . Then, the frequency (f in Hz.) and damping ratio (ζ in %) for each oscillatory mode is computed from the formulas ($f = \frac{\omega}{2\pi}$) and ($\zeta = -\frac{\sigma}{\sqrt{\sigma^2 + \omega^2}} * 100\%$), respectively.

The lightly-damped modes in the system are listed in Table 1. As can be observed, the system has two lightly-damped modes with damping ratios less than 10%.

B. MODE SHAPE

Mode shape is a way to measure the relative activity between the state variables and the modes. It provides important information about the contribution of state variables in the excitation of critical modes. Mathematically, mode shape is the polar plot of the right eigenvector matrix of the linearized system [12]. Fig. 4 shows the mode shapes of the lightly-damped modes. The magnitudes of the mode shape (lengths of the arrows) give the contribution amount of the state variables in the modes, whereas the phase angle displacement provides more information about the interaction of each state variable against the others for a specific mode. The mode shape of the critical mode ($-0.65117 \mp 9.0749i$) clearly indicates that machine-2 is the generator with the highest contribution in exciting this mode (the evidence is the length of the arrow colored in red).

C. PARTICIPATION FACTOR ANALYSIS

The product of right and left eigenvectors of the linearized system gives numerical information about the role of each state variable in the modes activity. Mathematically, the partition factor of each state to each mode is calculated using the following equation [12]:

$$p_{ki} = \frac{|v_{ki}| |\omega_{ki}|}{\sum_{k=1}^n |v_{ki}| |\omega_{ki}|} \tag{34}$$

$$P_{ki_normalized} = \frac{P_{ki}}{\max |p_{ki}|} \tag{35}$$

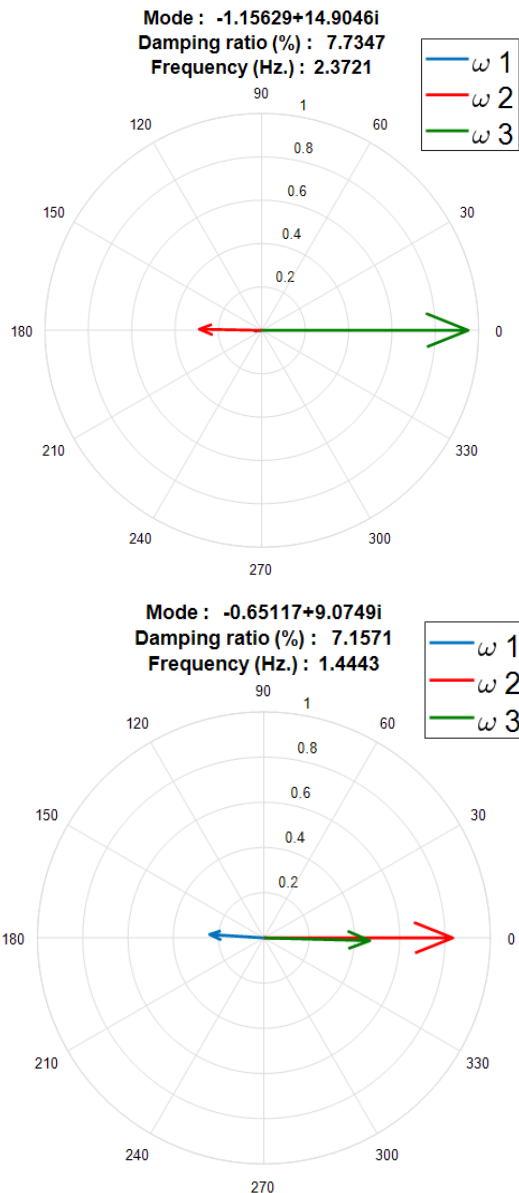


FIGURE 4. Mode shape plots for the lightly-damped modes.

where v_{ki} and ω_{ki} are the right and left eigenvectors of the system matrix, respectively, and p_{ki} is the participation factor of the k^{th} state variable into the i^{th} mode. The participation factors are normalized based on the highest value as illustrated in (35).

The program stores the numerical results of participation factor analysis and the corresponding names in an excel sheet. Since for this system, there are 33 states (each machine adds 11 state variables) and 33 eigenvalues, it is useful to visualize these numerical data in color. Fig. 5 shows the numerical results of this analysis visualized as a three-dimensional view using the MATLAB function “*pcolor*”. The state variables are shown on the x - axis, the modes with their specifications are shown on the y - axis, and the results of participation factor analysis are shown as a third dimension (color values).

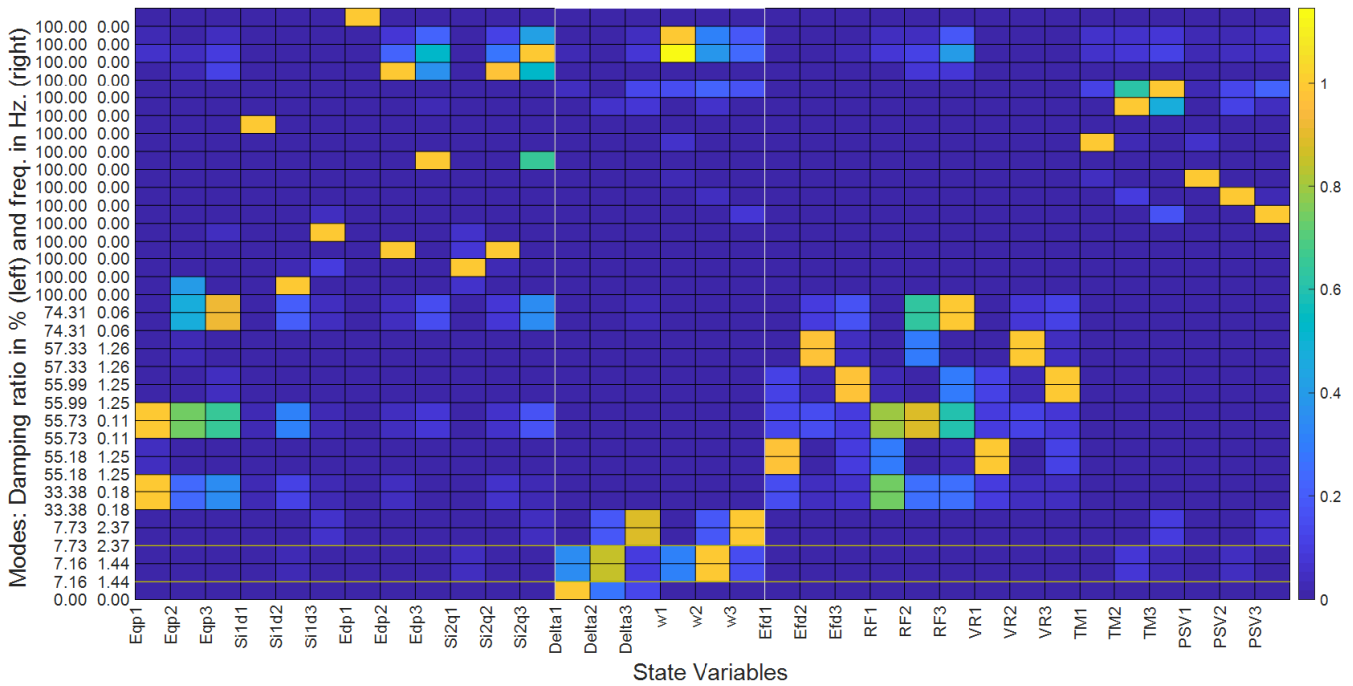


FIGURE 5. Participation factor map for the system.

We are more interested to see the damping ratios of the modes and their frequencies sorted in descending order. These two specifications are numerically displayed on the y-axis of the map. We can also highlight critical modes in the system (lightly-damped modes) and critical state variables (electromechanical states of rotor angle and speed). These are shown as horizontal and vertical lines on the map. Note that for a complex conjugate mode, the eigenvalues pair has the same damping ratio and frequency. Notably, the electromechanical state variables have the highest contribution to excitation of the lightly-damped modes. Also, Generator 2 exhibits relatively more impact on the critical mode $(-0.65117 \mp 9.0749i)$. It should be mentioned that since there is no infinite bus in the system, there is a zero-eigenvalue arisen because of redundancy in the state variables, specifically the phase angle [12].

D. MODAL CONTROLLABILITY, MODAL OBSERVABILITY, AND MODAL RESIDUE

Participation factor approach is a powerful technique for highlighting the mode activity in the state variables when the inputs and outputs are considered constants. However, the approach does not tell us how different controller’s inputs and outputs affect the relationship between the state variables and the modes. It cannot be used for solving problems of optimal placement of controller and feedback signal selection. Residue analysis, on the other hand, removes this drawback of the participation factor by including the inputs and outputs.

Mathematically, modal residue (R_i) is the product of modal observability (Cv_i) and modal controllability (w_iB)— that is, $R_i = Cv_iw_iB$, where R_i denotes the modal residue for

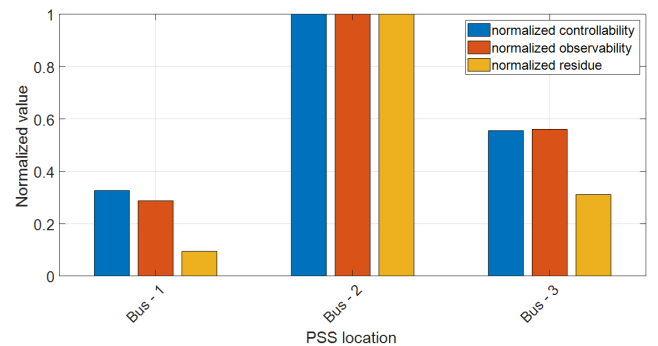


FIGURE 6. Optimal location of controller.

mode i , and v_i, w_i denote the right and left eigenvectors of the system, respectively. The optimal location of stabilizer can be obtained from the residue analysis. The bus with higher residue-magnitude is a good candidate for the controller to be installed at. The residue phase-angle tells us the amount of phase angle we need to compensate the phase lag [12]. Fig. 6 shows the normalized controllability, observability, and residue of the critical mode. The transfer function is calculated between the mechanical power reference (input) and the rotor speed (output) of machine 1. Clearly, to damp out the system oscillations caused by the critical mode, the optimal locations for the stabilizer to be installed are Bus 2, 3, then 1. The time-domain simulation, mode shape plot for the critical mode, and the participation-factor map share the same conclusion obtained from the residue analysis.

Residue analysis can also be used for the best input signal for the controller. However, care must be taken since different measurement signals do not have the same base scale

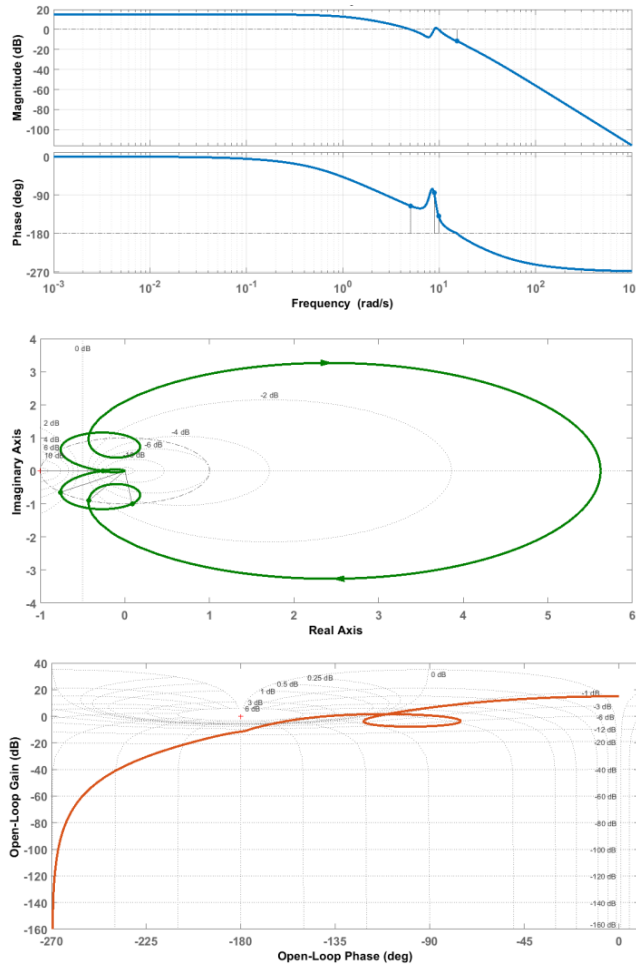


FIGURE 7. Frequency response analysis using bode (top), Nyquist (middle) and Nichols (bottom).

although most signal units are per unit. Reference [12] gives more details about this problem.

E. FREQUENCY RESPONSE ANALYSIS

So far, we have employed two analysis techniques for studying the dynamic behavior of power system: time-domain simulation and modal analysis. The program can be used for another analysis approach— that is, frequency response analysis with Bode, Nyquist, and Nichols diagrams. Stability of a system can be readily studied from these figures. For a closed-loop system to be stable, the open-loop gain and phase margins obtained from the Bode diagram must be positive and relatively large. Typical design values for the gain and phase margins are 6 dB and 30° – 60°, or 45° as average, respectively [13, p. 675]). Stability of the system can also be observed from the Nyquist and Nichols diagrams. The critical point for these methods is (-1, 0). The closed-loop system is stable if the plot does not encircle the critical point (Nyquist) or passes or approaches this point (Nichols). The results displayed in Fig. 7 show that the minimum gain and phase margins are 11.6 dB and 40.4° at frequency 15.2 and 9.83 rad/s, respectively. These frequencies are near the fre-

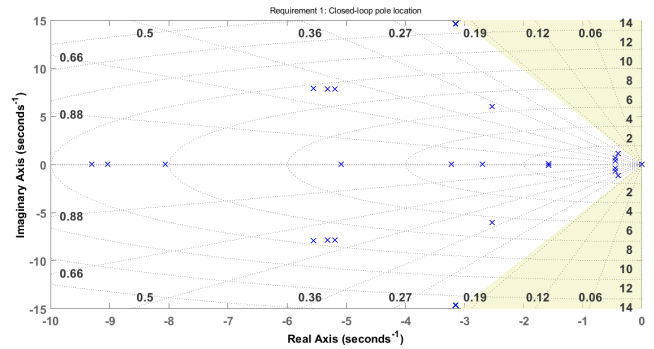


FIGURE 8. System tuning according to the pre-defined design target ($\zeta > 20\%$).

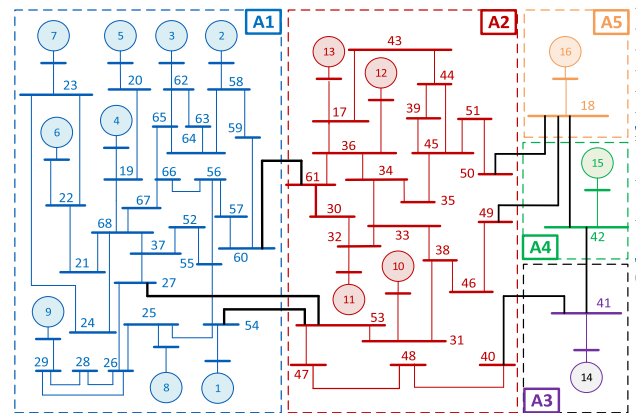


FIGURE 9. IEEE 68-bus dynamic test system [15].

quencies of lightly-damped modes where a small jump occurs in the gain and phase plots of Bode, Nichols, and Nyquist diagrams.

F. CONTROL DESIGN AND SYSTEM TUNING

Another use of the program is to design a feedback controller and tune its parameters according to some desired design goals without changing the DAEs structure and dimension. MATLAB has two powerful tools for this task namely “*sys-tune*” and “*feedback*”. Several design requirements can be assigned to the system tuner including reference tracking, minimum damping ratio, overshoot constraint, gain and phase margin, sensitivity, disturbance rejection, etc.

In this section, a PSS controller with two compensation blocks and a gain is designed and added to machine-2 (optimal location). The controller input is the rotor speed deviation of machine-2. The design goal requirement is the minimum damping ratio to be greater than 20%. The original DAEs model remains unchanged and is represented as an open-loop transfer function. Then, the function “*feedback*” is used to connect the DAEs to the controller with a reference input (ω_s) and a negative feedback signal (ω). The error signal is connected to the controller compensator which is in turn connected to the excitation system of machine-2. To tune the controller parameters, we need first to provide the minimum and maximum values for each tunable parameter. In this

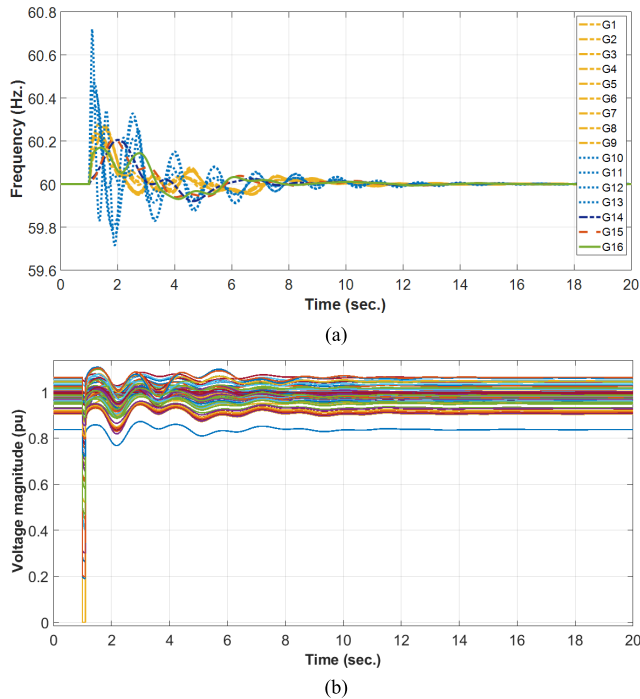


FIGURE 10. a. Frequency plots of generators in IEEE 68-bus system. b. Bus voltage magnitudes in IEEE 68-bus system.

design, the ranges of the time constants T_1, T_2, T_3, T_4 , and the gain are set to 0.1–1s, 0.01–0.1s, 0.1–1s, 0.01–0.1s, and 1–50, respectively. Then, the function “*systune*” is used to tune the parameters. The optimal tuned values for the parameters are found to be 0.2721s, 0.0990s, 0.2891s, 0.1000s, and 1.8776, respectively. Fig. 8 shows the positions of eigenvalues for the new closed-loop system with the goal requirements. Clearly, the system response is improved and the minimum damping ratio for the system with the controller is increased from 7.1% to 21%. The user can also use a different controller model, such as a PID controller. The function “*tunablePID*” can be used for this type of controller.

G. LARGE-SCALE POWER SYSTEMS

Large-scale systems such as IEEE 68-bus test system shown in Fig. 9 can also be simulated using the developed program. Fig. 10a shows the frequency plots of the generators following a three-phase fault at bus 17—the bus connected to the largest power generation. Generators within the same area are plotted in one color to see how the generators behave during a severe disturbance. Simulation time for this system with the detailed model of synchronous generators and using the implicit method is found to be 11.26s. The voltage magnitudes for the 68 buses in the system are also plotted and displayed in Fig. 10b. Note that algebraic variables such as voltage phasors change right at the disturbance time, whereas the state variables such as frequencies do not change suddenly at the time of disturbance.

To verify the efficiency of the proposed program to solve larger practical system, Texas’s 2007-bus synthetic system is employed and simulated. This system is used only to

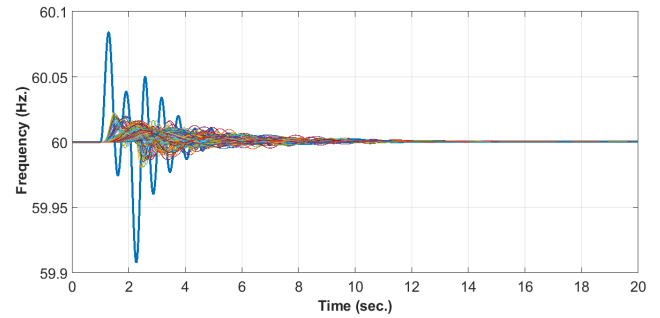


FIGURE 11. System response in Texas’s 2007-bus synthetic test.

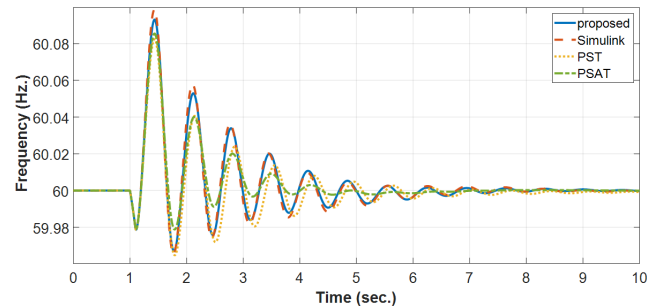


FIGURE 12. Comparison between the methods.

show the capability of the proposed program to simulate large systems. The static and dynamic data for this system are not real values but are generated by some studies in recent years [14]. Figure 11 shows the simulation result for this system after applying a disturbance near Generator 1. The required time for this 20s-period simulation is 158s based on a regular computer and 0.01s step size assuming detailed model of generators. This system is also added to the package.

VI. COMPARISON OF METHODS

The proposed program is simple comparing to the other open-source packages; it consists of three to five *m*-files at the most for the time-domain simulation. Among these files, one is dedicated for the constant parameters, another file is specialized to compute the initial values of all algebraic and state variables, and one or three separate files for the DAEs before, during, and after applying a disturbance.

For the time-domain comparison, a three-phase fault is applied at Bu-5 in the 9-bus test system for the period $t = 1 - 1.1 s$ and is cleared naturally at $t = 1.1s$. Figure 12 shows a comparison between the proposed method and the ones used by PST, PSAT and Simulink program proposed in [15]. Note that for the PSAT, the loads were converted to constant impedances; otherwise, the simulation stops at the time of disturbance without reaching a solution. Notably, the results obtained from these programs are close to each other, especially with the proposed program and Simulink.

The required time for the simulation using the proposed program is quite faster. For instance, in [6], it is mentioned that the required time to simulate the 68-bus system was 162.599s, and 65.37s for the PST, whereas it required only 11.26s for the proposed program to solve the system

including solving load flow analysis, initial conditions calculation, and plotting the results. System model, computer features and simulation time (20s) were all similar in this comparison.

Comparing to the Simulink programs in [15]–[16], the proposed program is much faster, completely vector-based with exact symbolic linearization, and can be used to create GUI programs, whereas the Simulink programs use explicit solution to solve the DAEs. However, when the loads are treated as constant impedances, the two programs show approximately similar required time for the simulation.

VII. EDUCATIONAL VALUE OF THE PROGRAM

The programs developed in this study are useful for students and researchers. Students can easily build and modify a multi-machine power system following a mathematical model of each component in the system. Connecting theoretical background to simulation practice helps students to gain academic experience in the field. The symbolic representation of the system modeled as a set of DAEs and its exact linearization provides analytic information about the problem. Students can compare between different numerical solutions such as implicit and explicit approaches using the program. Further, students can learn how to make their programs more efficient using vectorization method instead of for-loops. The control design is another advantage of this program. People in the field including students and researchers can design a controller without changing the DAEs structure.

VIII. CONCLUSION

This paper presented a MATLAB-based program for simulating multi-machine power systems considering implicit and explicit solutions for generator- and network-side disturbances. The program uses the MATLAB's built-in solvers for time-domain simulation benefiting from the vectorization method that significantly speeds up the simulation. Power system linearization can also be obtained from the program with a symbolic and exact representation. Modal analysis, mode shape, participation factor analysis and visualization, optimal location of controller, feedback signal selection, control design, and frequency response analysis are some other

tasks that can be achieved using the program. The program is aimed to be used as teaching tools or for research studies to simulate small and large-scale power system.

REFERENCES

- [1] P. Sauer, M. A. Pai, and J. H. Chow, *Power System Dynamics and Stability*, 2nd ed. Hoboken, NJ, USA: Wiley, 2017.
- [2] L. Vanfretti and F. Milano, "Experience with PSAT as free and open-source software for power system education and research," *Int. J. Elect. Eng. Educ.*, vol. 47, no. 1, pp. 47–62, 2008.
- [3] S. Cole and R. Belmans, "MatDyn, a new MATLAB-based toolbox for power system dynamic simulation," in *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1129–1136, Aug. 2011.
- [4] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE Trans. Power Syst.*, vol. 7, no. 4, pp. 1559–1564, Nov. 1992.
- [5] F. Milano, "An open source power system analysis toolbox," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1199–1206, Aug. 2005.
- [6] S. Ekinici, A. Demiroren, and H. Zeynelgil, "PowSysGUI: A new educational software package for power system stability studies using MATLAB/Simulink," *Int. J. Elect. Eng. Educ.*, vol. 54, no. 4, pp. 283–298, 2017.
- [7] J. Chow and G. Rogers, "Power System Toolbox Manual, Version 3.0," Tech. Rep., 2008. [Online]. Available: <https://www.ecse.rpi.edu/~chowj/PSTMan.pdf>
- [8] W. Xu, E. Vaahedi, Y. Mansour, and J. Tamby, "Voltage stability load parameter determination from field tests on BC Hydro's system," *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1290–1297, Aug. 1997.
- [9] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
- [10] I. Abdulrahman and G. Radman, "Wide-area-based adaptive neuro-fuzzy SVC controller for damping interarea oscillations," *Can. J. Elect. Comput. Eng.*, vol. 41, no. 3, pp. 133–144, Summer 2018.
- [11] P. Anderson and A. Fouad, *Power System Control and Stability*, 2nd ed. New Delhi, India: Wiley, 2003.
- [12] B. Chaudhuri and B. Pal, *Robust Control in Power Systems*. New York, NY, USA: Springer, 2005.
- [13] R. Dorf and R. Bishop, *Modern Control Systems*, 12th ed. New Delhi, India: Pearson, 2014.
- [14] *Illinois Center for a Smarter Electric Grid (ICSEG)*, Inf. Trust Inst., Univ. Illinois Urbana-Champaign, Urbana, IL, USA, Jun. 2016. [Online]. Available: <https://icseg.iti.illinois.edu/power-cases/>
- [15] I. Abdulrahman and G. Radman, "Simulink-based program for simulating multi-machine power systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Portland, OR, USA, Aug. 2018, pp. 1–5.
- [16] I. Abdulrahman and G. Radman, "Simulink-based programs for power system dynamic analysis," in *Electrical Engineering*, vol. 101, no. 2. Berlin, Germany: Springer, Jun. 2019, pp. 345–356.
- [17] *MATLAB-Based Programs for Power System Dynamic Analysis*. Accessed: Jun. 29, 2019. [Online]. Available: https://www.dropbox.com/s/jekjdjhpdc1u9198/MATLAB_All_Programs.zip?dl=0

...