MODELLING AND SIMULATION OF NONLINEAR INTERCONNECTED
LARGE-SCALE SYSTEMS

F. Crusca* and M. Aldeen**

*Department of Electrical and Computer Systems Engineering, Building 72, Monash University, Melbourne, Victoria 3800, Australia. E-mail: francesco.crusca@eng.monash.edu.au

**Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria, 3010, Australia. E-mail: aldeen@unimelb.edu.au

Abstract: A novel modular approach to the modelling of nonlinear interconnected complex systems is introduced in this paper. In the approach, a complex system is decomposed into self-contained easily identified modules to achieve easy linking of individual models. Our proposed approach to modelling differs from the traditional approaches in the following important aspects: (i) it is nonlinear, modular-based, systematic and graphical; (ii) it is suitable for fault diagnosis; (iii) depending on the level of complexity required, each module may be changed separately quite easily without having to change the other modules; and (iv) it draws from the powerful capabilities of the widely used packages of Simulink™ and Matlab™. These features provide for a generic package that can simulate a complex interconnected system of any description and configuration, as well as simulating any kind of fault anywhere in the system by merely changing the input vector. Examples of two power systems are used as case studies, where it is shown that complex systems can be constructed and simulated easily by appropriately combining
components together. The simulation studies show the dynamical behavior of the two systems with and without faults being applied to them.

**Keywords:** Simulation, modelling, complex systems, nonlinear dynamics, multi-machine power systems.

1. Introduction

Complex systems usually comprise a number of components interconnected together to form a unit which performs a specific task. In order to study the dynamic characteristics of such systems, a model needs to be developed. It is advantageous that the model can incorporate fault conditions in any part of the system, so that both normal and abnormal operating conditions may be simulated. Such a model is often described by a set of differential and algebraic equations. Detailed models may require a very large number of such equations, some of which may be nonlinear. The full set of these equations forms the basis for investigation into the system’s dynamics via detailed simulation. The main features that need to be considered are discussed next.

*Modularity:* As large-scale interconnected systems are physically comprised of a number of subsystems, it is natural that each of these subsystems be modeled separately and, through input-output signals, be interconnected to others. In this way a number of advantages can be realized, amongst which are flexibility, simplicity, facilitation of configuration, and ease of implementation. As a result, a highly modular based model of the interconnected system can be
obtained, one that will permit the user to analyse the behaviour of the system as a whole just as easily as analyzing any particular subsystem.

Linearisation: In some cases it is desirable to be able to automatically extract a local linearisation at some given equilibrium operating point. The simulator developed in this paper is capable of providing this feature. Such linearised models are often used as the first step in the state estimation and/or controller design and analysis phase. It needs to be noted, however, that such linearisations lead to the inconvenience of having a model that is only valid in a neighbourhood of the original equilibrium operating condition. Another inconvenience is that each time an operating point changes, due to faults or malfunctions, the linearisation process must be repeated. This is not normally an easy process, when large-scale complex systems are considered. However, using our approach, the facility to automatically synthesise local linearisations is readily incorporated.

Fault Modelling: There is an ever increasing need for introducing fault detection, isolation and control schemes in complex engineering systems ranging from nuclear reactors, to flight controllers, to industrial process control systems. In order to achieve this, a fresh modeling philosophy is required. In this aspect the following three requirements are paramount:

(a) each component of the system should be modeled with fault conditions effected internally such that both normal and fault conditions can be simulated uniformly and systematically;
(b) the ability to isolate and remove the faulted component from the system;
(c) the ability to differentiate between fault signals and normal signals.

Numerical Considerations: Through simulation, the dynamic behaviour and main characteristics of the system can be studied and analysed. Digital simulation of complex
dynamical systems involves using numerical methods to arrive at approximate solutions to differential equations. This numerical approximation problem is in itself as challenging as the actual problem of modelling of the original system. This is especially true for highly nonlinear systems which are ill-conditioned, and have dynamics over a wide range of time-scales [1]. This complicates the solution process considerably. In this specific case, numerical solution procedures applicable for “stiff” systems are often better suited than ordinary solvers for the problem. Therefore it is important that any modelling should be done in conjunction with simulation.

Debugging: Another important issue that needs to be addressed when modelling and simulating complex systems is considered is debugging. For complex nonlinear systems debugging can be very messy and time consuming. Therefore a unified approach to modelling, debugging and simulation should be taken when dealing with large complex nonlinear systems.

In this paper, we present a block-diagram graphical modular approach to the modelling, simulation and debugging of complex non-linear dynamic systems. The approach utilises the graphical capabilities of the Matlab™ and Simulink™[2] software packages. The approach is generic and can apply to any complex interconnected systems. However, for the sake of illustration, we take the specific case of modelling of power systems, and illustrate the main features of the approach through two case studies.

2. Modelling of Power Systems

Fig. (1) shows the main functional components of a power system. These are the synchronous generator, exciter, turbine-governor, power system stabiliser (PSS), transmission network and the
machine-network reference frame transformation\(^1\). In the following we provide a detailed analysis of each of the components shown in fig. 1.

Figure (1) main components of power system

2.1 Synchronous Generator Model

A detailed model of a synchronous generator which is often used in the dynamic stability studies comprises differential and algebraic equations for the stator, rotor, field, and three damper windings, one in the \(d\)-axis and two in the \(q\)-axis [3-5]. The common approach has been to treat

\(^1\) Although, a PSS is shown, it is not an essential modelling component and is therefore outside the scope of this paper. Its primary function is to improve the dynamic stability of the power system, though once designed it is quite easy to implement.
these equations as one whole set of equations. However our approach differs from the conventional approach in that we first decompose the whole set of equations into self-contained physically meaningful subsets, and then obtain a solution by interconnecting these subsets together, as shown in fig. (2).

In this figure, the generator dynamics are partitioned into electrical and mechanical components. The electrical component deals with the electromagnetic equations that govern the behaviour of the flux-linkages between the mutually coupled stator and rotor windings. The mechanical component deals with the rotational dynamics of the turbine generator rotor shaft. The figure reveals that there is a “one-way” interaction from the electromagnetic equations, by means of the
air-gap electrical torque, to the mechanical subsystem. The motivation for performing the proposed decomposition is threefold: (i) to provide a better insight, through modularization, into the internal dynamics of the individual components. This reveals the pattern of the interaction from the electrical to the mechanical subsystem. In this way we believe that we have achieved the correct balance between modelling complexity and ease of analysis; (ii) to facilitate debugging; and (iii) to facilitate identification of fault occurrences and their origins.

2.1.1 d-axis flux dynamic equation

The dynamics of the flux-voltage relationship along the d-axis may be modelled by the following set of nonlinear differential equations:

\[ \dot{\psi}_d = -\omega_b r_d i_d + v_d + (\omega_b + \delta)\psi_q \]  
\[ \dot{\psi}_F = -\omega_b r_F i_F + v_F \]  
\[ \dot{\psi}_D = -\omega_b r_D i_D + v_D \]  

where \( \psi_d, \psi_F, \psi_D \) are direct-axis and damper windings flux linkages, \( v_d, v_F, v_D \) are direct-axis and damper windings voltages, \( \omega_b \) is the base speed and \( \delta \) is the shaft speed.

If we define the states as: \( \psi_{qGQ} = [\psi_q, \psi_G, \psi_Q]^T \), and inputs as: \( i_{dFD} = [i_d, i_F, i_D]^T \), and \( v_{dFD} = [v_d, v_F, v_D]^T \), then (2.1)-(2.3) can be represented in state-space form as:

\[ \dot{\psi}_{dFD} = f_{dFD}(\psi_{qGQ}, i_{dFD}, v_{dFD}, \delta) \]  

Equation (2.4) is now programmed in Simulink™ as a block with the inputs and outputs shown in fig. (3).
Figure (3) *d-axis* flux linkage model

The block marked “psiDotFD” has been programmed as a Matlab™ S-function. S-functions have been used in this paper because they are conveniently programmed, through the use of a set of provided standard templates, in a number of programming languages. They allow systems of differential and algebraic equations of the form \( \dot{x} = f(x, u, t); y = g(x, u, t) \) to be readily programmed and numerically solved. Typically, S-functions are coded in Matlab™’s own scripting language. However, in cases where the execution speed needs to be optimized, they may be coded using higher level languages such as C.

The relationship between the flux linkage and the currents is a nonlinear periodic function of rotor angle, \( \theta \). However, following application of Park’s transformation [6], this relationship becomes linear (neglecting saturation). In the *d-axis* this is represented by

\[
\begin{align*}
\psi_d &= L_d i_d + L_{AD} i_F + L_{AD} i_D \\
\psi_F &= L_{AD} i_d + L_{F} i_F + L_{AD} i_D \\
\psi_D &= L_{AD} i_d + L_{AD} i_F + L_{D} i_D
\end{align*}
\] (2.5, 2.6, 2.7)

where \( i_d, i_F, i_D \) are the direct-axis and damper windings currents, \( L_d, L_F, L_D \) are the self inductance of the direct-axis and damper windings, and \( L_{AD} \) is the mutual inductance between the d-axis and damper windings.
2.1.2 \textit{q-axis flux dynamic equation}

The flux-voltage dynamics in the \textit{q-axis} are described by the following nonlinear matrix-vector equation:

\[ \dot{\psi}_q = -\omega_B r_i_q + v_q - (\omega_B + \delta)\psi_d \] (2.8)

\[ \dot{\psi}_g = -\omega_B r_i_g + v_g \] (2.9)

\[ \dot{\psi}_q = -\omega_B r_i_q + v_Q \] (2.10)

Equations (2.8)-(2.10) can be represented in state-space form as:

\[ \dot{\psi}_{qGQ} = f_{qGQ} (\psi_{dFQ}, i_{qGQ}, v_{qGQ}, \delta) \] (2.11)

Following the same approach for the \textit{d-axis}, this equation is programmed as shown in fig. (4).

![Figure (4) q-axis flux linkage model](image)

Similarly, the \textit{q-axis} current-flux relationship also becomes linear (neglecting saturation) and is described by

\[ \psi_q = L_q i_d + L_{dQ} i_F + L_{qQ} i_Q \] (2.12)
\[ \psi_g = L_{dQ} i_d + L_{dF} i_F + L_{dQ} i_Q \]  
(2.13) 
\[ \psi_Q = L_{dQ} i_d + L_{dQ} i_F + L_{dQ} i_Q \]  
(2.14)

The flux-voltage dynamic equations of fig.’s (3-4) may now be combined and specific components of \(d\)- and \(q\)-axis flux linkages and currents may be extracted and combined together (e.g., \(\psi_{qd} = [\psi_q \quad \psi_d]^T\), \(i_{qd} = [i_q \quad i_d]^T\), and \(v_{qd} = [v_q \quad v_d]^T\)) to produce a model for the overall synchronous generator flux dynamics, as shown in fig. (5a). The S-function components shown here are readily interconnected using Simulink\textsuperscript{TM}. Once the interconnections have been graphically programmed, it is then possible to focus on the input-output behaviour of the overall module. This is achieved by using a further feature of Simulink\textsuperscript{TM}, where the various components are grouped together into an overall component, and, after introduction of the stator-referred field voltage input, \(E_{FD}\), the flux-voltage dynamics may be represented as shown in fig. (5b).
2.1.3. Swing equation

The swing equation represents the mechanical motion of the rotor shaft and is written as

$$\ddot{\delta} + D \dot{\delta} = \frac{\omega_b}{2H} T_a$$

(2.15)
\[ \theta = \omega_d t + \delta + \pi / 2 \]  
(2.16)

where \( T_a = T_m - T_e \), \( \omega_d = 2\pi f \) (rad/s) and the electrical torque is obtained from \( T_e = (\Psi_d i_q - \Psi_q i_d) / 3 \). The parameter \( D \) represents the damping effect of the load-frequency relationship. A state-space representation for the swing equation is derived as

\[ x_{SE} = f_{SE}(x_{SE}, T_m, T_e) \]  
(2.17)

where the state vector is defined as \( x_{SE} = [\delta \quad \dot{\delta}]^T \). This set of equations can be programmed as shown in fig. (6), where the block “swingEqn” is an S-function.

Figure (6) swing equation dynamics

2.1.4. Reference frames

The modified orthogonal form of Park’s transformation [3,4] is used to relate the machine reference frame to that of the network as discussed below.

*machine-system interface:* In order to interface the machine to the network, the following transformation is used.
\[
\begin{bmatrix}
V_{dm} \\
V_{qm}
\end{bmatrix} = \begin{bmatrix}
\cos \delta & -\sin \delta \\
\sin \delta & \cos \delta
\end{bmatrix}
\begin{bmatrix}
V_{ds} \\
V_{qs}
\end{bmatrix}
\]  
(2.18)

This is programmed as shown in Fig (7a), where the block “VN2vqd” is an S-function.

Figure (7a) voltage transformation

**system-machine interface:** In order to complete the interface between the network and the machine, the following transformation from the system reference frame back to the machine frame is also required.

\[
\begin{bmatrix}
I_{ds} \\
I_{qs}
\end{bmatrix} = \begin{bmatrix}
\cos \delta & \sin \delta \\
-sin \delta & \cos \delta
\end{bmatrix}
\begin{bmatrix}
I_{dm} \\
I_{qm}
\end{bmatrix}
\]  
(2.19)

This is programmed as shown in fig. (7b), where the block “iqd2IN” is an S-function.

Figure (7b) current transformation

2.1.5. Synchronous machine complete model

If we combine fig.’s (5b), (6), (7a) and (7b), we arrive at a complete model of a synchronous machine as shown in fig. (8a). The inputs are the field voltage, \( E_{FD} \), mechanical torque, \( T_m \), and
the machine terminal voltage, \( V_N \), obtained from solving the network load flow problem (initialization problem). The outputs of the model are the load angle, \( \delta \), speed, \( \dot{\delta} \), terminal voltage magnitude, \( |V_N| \), accelerating torque, \( T_a \), and the machine current, \( I_N \). This model can be represented in a more compact form as shown in Fig (8b).

Figure (8a) synchronous machine model
Figure (8b) compact input-output representation of synchronous machine model

2.2. IEEE Type 1S Excitation Model

Figure (9) block-diagram representation of excitation system model

In fig. (9), a model of an IEEE Type 1S excitation system [7] is shown. This may be represented by the third order model shown below:
\begin{align}
\dot{V}_1 &= -\frac{1}{\tau_R} V_1 + \frac{K_R}{\tau_R} \left| V_N \right| \\
\dot{x}_w &= -\frac{1}{\tau_F} x_w + \frac{K_F}{\tau_F} \dot{E}_{FD} \\
\dot{V}_A &= -\frac{1}{\tau_A} V_A - \frac{K_A}{\tau_A} \left( V_{REF} + V_s - V_F - V_R \right) \\
E_{FD} &= \begin{cases} 
V_{R_{max}} & V_A > V_{R_{max}} \\
V_A & V_{R_{min}} \leq V_A \leq V_{R_{max}} \\
V_{R_{max}} & V_A < V_{R_{min}} 
\end{cases}
\end{align}

Equations (2.20)-(2.23) may be re-written in state-space form as:

\begin{align}
\dot{x}_{ex} &= f \left( x_{ex}, V_{REF}, V_N, V_s \right) \\
y &= E_{FD} = C_{ex} x_{ex}
\end{align}

This is programmed as shown in Fig (10a), using basic Simulink model components. Thereafter, the model is compacted as shown in fig. (10b).

Figure (10a) exciter model
Figure (10b) input-output representation of exciter model

2.3. Governor-Turbine Model

A simplified model of an IEEE turbine-governor mechanism [8] is shown in fig. (11). As shown in the figure, this comprises four basic components, namely, the speed governor, the speed relay, the servomotor and the steam turbine. For this system a third order model is derived as shown below.

Figure (11) IEEE steam turbine speed governing system model

\[
\dot{P}_s = -\frac{1}{T_{SR}} P_s + \frac{1}{T_{SR}} P_{SR} \quad (2.26)
\]

\[
\dot{P}_v = -\frac{1}{T_{SM}} P_v + \frac{1}{T_{SM}} P_s \quad (2.27)
\]

\[
\alpha = \frac{1}{T_{SM}} (P_c - P_v)
\]
\[
\dot{P}_m = -\frac{1}{T_{CH}} P_m + \frac{1}{T_{CH}} P_v
\]

\[
\beta = \begin{cases} 
\dot{P}_{v_{open}} & \alpha > \dot{P}_{v_{open}} \\
\alpha & \dot{P}_{v_{closed}} \leq \alpha \leq \dot{P}_{v_{open}} \\
\dot{P}_{v_{closed}} & \alpha < \dot{P}_{v_{closed}}
\end{cases}
\]

\[
\dot{\sigma} = \beta
\]

\[
P_v = \begin{cases} 
P_{v_{max}} & \sigma > P_{v_{max}} \\
\sigma & P_{v_{min}} \leq \sigma \leq P_{v_{max}} \\
P_{v_{min}} & \sigma < P_{v_{min}}
\end{cases}
\]

Equations (2.26)-(2.28) are now represented in state-space form as

\[
\dot{x}_G = f_G \left(x_G, P_r, \dot{\delta} \right) \tag{2.29}
\]

\[
y_G = g_G \left(x_G, P_r, \dot{\delta} \right) \tag{2.30}
\]

The governor model is programmed as shown in fig. (12a), and then compacted into a module as shown in fig. (12b).
2.4. Complete Generating Unit Model

A complete model of a power system generating unit is shown in fig. (13a). As shown, the inputs are the excitation system command voltage, $V_{REF}$, speed-governor command, $P_r$, the machine terminal voltage, $V_N$, obtained from solving the network load flow problem (initialization problem), and the speed deviation of the reference machine, $\delta_{REF}$. The outputs of the model are
the relative load angle, $\delta$, the speed, $\dot{\delta}$, terminal voltage magnitude, $|V_N|$, active power at terminals, $P_N$, accelerating torque, $T_a$, stator-referred field-voltage, $E_{FD}$, mechanical input power, $P_m$, gate valve power, $P_{GV}$, rate of change of gate valve power, $\dot{P}_{GV}$, and the machine terminal current, $I_N$. This can be represented in a more compact way as shown in fig. (13b).

Figure (13a) Simulink™ model diagram of generating unit
2.5. Multi-Machine Power System Model

A multi-machine power system can be easily constructed from the modules described in section 2, as shown in fig. (14). This can be accomplished by choosing any number of generating units and connecting them to a common transmission network of lines and loads. Once the network is interfaced with the multi-machine system an initialization (load-flow) of the network is carried out to determine the interface variables, which are the terminal voltages and currents of each generator within the multi-machine system. These are then transformed via the network-machine transformation into the machine’s own internal coordinate system.
The flexibility of the modular modelling approach is now evident in that any unit can be configured from the graphical user interface to contain all or a selection of the standard components described in section 2. This flexibility is very useful from a practical viewpoint as not all synchronous generators necessarily require exciters, or power system stabilisers, or even governors.

3. Case Studies

In this section we present two case studies. In Section 3.1, we illustrate how to model and simulate a single-machine infinite-bus power system under normal and faulted conditions. In
Section 3.2, we extend our approach to model and simulate a three-machine power system under both normal and faulted conditions.

3.1 Modelling and Simulation of a Single Machine Infinite Bus System

To illustrate the main features of the modular modelling approach, we consider a simple case of a single-machine connected to an infinite bus bar via a double-circuit transmission line and a local load as shown in fig. (15). For the sake of simplicity, we consider the scenario where line 1 is under line-to-ground fault and the other two lines are assumed to be fault-free. A detailed model for this scenario is next presented.
3.1.1 Network Model

A detailed analytical model of the transmission network will now be developed. To illustrate the ease with which faults can be introduced into the system, we consider the case where line 1 is subjected to a solid three-phase line to ground fault at a location $\eta_1$ along the length of the line, where $\eta_1$ is a free variable in the range $[0,1]$. It follows that choosing $\eta_1 = 0$ leads to the fault being placed at the machine terminals, while choosing $\eta_1 = 1$ will place the fault at the infinite bus bar. To allow modelling of the fault, we introduce the fault signal $f_i$. This is defined as
follows: \( f_i = 0 \) signifies no fault condition and \( f_i = 1 \) signifies fault condition at location \( \eta_i \).

Line 1 may therefore be modeled as follows:

\[
I_{N1} = \eta_i^{-1}Y_iV_N + (1-f_i)\left(1-\eta_i^{-1}\right)Y_iV_N - (1-f_i)Y_iV_b
\]

(3.1)

Introducing a change of variables, \( \eta_i \rightarrow \rho_i = f_i / \eta_i \), allows (3.1) to be rewritten more compactly as:

\[
I_{N1} = (1-f_1-\rho_1)Y_iV_N - (1-f_1)Y_iV_b
\]

(3.2)

Similarly, the voltage-current relationship for lines 2 and 3 are modeled as:

\[
I_{N2} = Y_2V_N
\]

(3.3)

\[
I_{N3} = Y_3V_N
\]

(3.4)

where \( Y_i = Z_i^{-1} \), \( i \in \{1,2,3\} \).

Thus the total terminal current is:

\[
I_N = \left\{ (1-f_1-\rho_1)Y_1 + Y_2 + Y_3 \right\}V_N - (1-f_1)Y_iV_b
\]

(3.5)

This may be rearranged and expressed as

\[
V_N = f(I_N, d)
\]

(3.6)

where \( d \equiv [f_1 \quad \rho_1]^T \) is a vector of fault parameters. Equation (3.6) is represented compactly as shown in fig. (16). Note that the fault signal and the fault location as expressed by \( d \) now enter into the network as a disturbance input signal. Similarly, any other component of the power system can be modelled so that the fault signal can be extracted in this way. This feature enables the estimation of the fault parameters through the use of appropriate estimation schemes [9] and therefore distinguishes our approach from that adopted in the application-specific toolbox [10], for example.
3.1.2. Simulation Studies

The problem of obtaining a solution to the case study outlined in section 3.1 is quite involved as is typically the case with power systems dynamics studies. The complexity stems from the fact that a wide-range of time-scales is involved. For this reason we have elected to use the solver “ode23tb” (stiff/TR-BDF2) in Matlab™ rather than the general purpose “ode45”.

In the following we present the simulation results carried out on the single-machine infinite-bus system shown in fig. (15), for the conditions outlined above. The data used for this study is given in [3]. The following case is simulated: a 5% step in excitation, $V_{REF}$, is introduced at time $t=0.5$ (s), followed by a fault applied at the midpoint along line 1 at time $t=5.5$ (s). The fault is cleared after 100 (ms), and thereafter the faulted line is dropped. In fig.’s (17) the responses of the main dynamic indicators, such as the load-angle, speed-deviation, and terminal voltage magnitude responses, are shown. It can be seen that the system returns to equilibrium after each of the disturbances. From fig. (17a), it can be seen that the load angle decreases following the application of the step command to the excitation system, and from fig. (17d) it can also be seen that the machine voltage magnitude follows the command signal, as expected. In fig.
(17e) the machine voltage magnitude, \(|V_n|\), is seen to fall from its initial value of 1.1 p.u. (per unit) to a low as 0.5 p.u. immediately upon application of the fault. During the time that the fault is applied the rotor begins to accelerate due to the accelerating torque. As a result, the speed and the load angle increase (as shown in fig.’s (17a) and (17b), respectively). Following clearing of the fault (by dropping the faulted line), the speed deviation returns to zero, and the load angle settles at a new equilibrium operating point. Due to the rapid clearing of the fault, the system stability is seen to be maintained.

Figure (17a) load-angle response
Figure (17b) speed deviation response

Figure (17c) voltage response
Figure (17d) blow-up of voltage response during first 5 seconds

Figure (17e) blow-up of voltage response prior to application and following clearing of the fault
3.2 Modelling and Simulation of a 3-Machine System

The modularity of the modelling approach is best illustrated through a more comprehensive system of three machines connected to a network of 9 buses and three loads, as shown in fig. (18). Full data for the system operating condition is available in [3] (page 38). In the same reference, the data for the machines is given in (page 98). In our study, all three machines are equipped with exciters and speed governors as described in Sections 2.2 an 2.3, respectively.

![Diagram of three-machine, nine bus system](image)

Figure (18): three-machine, nine bus system (from [3] (page 38))

3.2.1 Network Model
In this section we derive a detailed analytical model of the transmission network, incorporating fault conditions. To illustrate the ease with which faults can be introduced into the system, we consider the case where a solid three-phase line to ground fault is arbitrarily applied to a line between nodes \( j \) and \( k \). All of the other lines are assumed to be fault-free. A detailed model for this situation is developed next.

![Figure (19) detailed double-circuit model for faulted line \( j-k \)](image)

From fig. (19), we write:

\[
i_j = \left\{ \frac{1}{2} y_{jk} (V_j - V_k) + \frac{1}{2} y_{jk} (V_j - V_k) + y_{jj} V_j \right\} (1 - f) + \left\{ \frac{1}{2} y_{jk} (V_j - V_k) + y_{jj} V_j + \frac{1}{2} y_{jk} V_{jj} \right\} f \quad (3.7)
\]

\[
i_k = \left\{ \frac{1}{2} y_{jk} (V_k - V_j) + \frac{1}{2} y_{jk} (V_k - V_j) + y_{kk} V_k \right\} (1 - f) + \left\{ \frac{1}{2} y_{jk} (V_k - V_j) + y_{kk} V_k + \frac{1}{2(1-\eta)} y_{jk} V_k \right\} f \quad (3.8)
\]

or, more compactly, as:
\[ i_j = \left\{ y_{jk} (V_j - V_k) + y_{kj} V_j \right\} - \left\{ \frac{1}{2} y_{jk} (V_j - V_k) - \frac{1}{2 \eta} y_{jk} V_j \right\} f \]  \hspace{1cm} (3.9)

\[ i_k = \left\{ y_{jk} (V_k - V_j) + y_{kj} V_k \right\} - \left\{ \frac{1}{2} y_{jk} (V_k - V_j) - \frac{1}{2 (1 - \eta)} y_{jk} V_k \right\} f \]  \hspace{1cm} (3.10)

With reference to equations (3.9) and (3.10), the terms in \( i_j \) and \( i_k \) within the first set of braces are entries in the Y admittance matrix, and the terms in the second set of braces form the fault entries, which may be modelled as follows. The \( j^{th} \) entry is

\[ g(j) = \frac{y_{jk}}{2 \eta} V_j - \frac{1}{2} y_{jk} (V_j - V_k) \]  \hspace{1cm} (3.11)

And the \( k^{th} \) entry is

\[ g(k) = \frac{y_{jk}}{1 - \eta} V_k - y_{jk} (V_k - V_j) \]  \hspace{1cm} (3.12)

Consequently, the network equation can be written as

\[ I = YV + DVf \]  \hspace{1cm} (3.13)

where the node currents are denoted by \( I = [I_1 \ I_2 \ \ldots \ \ I_n]^T \), the node voltages by \( V = [V_1 \ V_2 \ \ldots \ V_n]^T \), and \( Y \) is the nodal admittance matrix. The matrix \( D \) in equation (3.13) is a matrix of the same dimensions as \( Y \), with the non-zero elements given by:

\[ D_{jj} = \frac{1 - \eta}{2 \eta} y_{jk}, \quad D_{jk} = D_{kj} = \frac{1}{2} y_{jk}, \quad D_{kk} = \frac{\eta}{2 (1 - \eta)} y_{jk} \]  \hspace{1cm} (3.14)

Note that equation (3.13) shows that the fault signal, \( f \), enters into the network equation as an external input signal. Similarly, any other component of the power system can be modelled so that the corresponding fault signals can be expressed in this way. Also, the equation reveals that when the system is operating under normal operating conditions, that is when \( f = 0 \), the network model is reduced to the standard form, i.e., to:
3.2.2. Simulation Studies

The three machine system example of fig. (18) is used in this simulation example. To carry out simulation studies, we first construct a nonlinear simulator from the basic modules of power systems described in section (2). In this case, three generating units, each with a synchronous machine, an exciter and a governor are interconnected to the network of 9 buses and three loads, through the SIMULINK graphical interface, as shown in fig. (20). As can be seen, the network has an external input representing the fault signal, and is so modeled as to allow the simulation of both faulty and normal conditions.

For the sake of illustration, we study the case where line 5-7 (modeled as a double-circuit line, as shown in fig. (19) ) is subjected to a solid three-phase line-to-ground fault. Using the power system nonlinear simulator shown in fig. (20), a 0.05 pu step increase in excitation, $V_{REF,1}$, is applied to unit #1 at time $t = 1.0$ (s). This is followed by a fault applied at the midpoint along line 5-7 at time $t = 6$ (s). The fault is cleared after 100 (ms), and thereafter the faulted line is dropped.
Fig. (21) shows the speed response of each of the three machines for the same condition, overlaid. The response in the first 6 seconds is quite predictable, as the machine speeds experience some oscillation after the application of the step change in machine #1 reference voltage, $V_{REF,1}$. When the fault is applied at $t = 6$ (s), the speed of all three machines rapidly increases due to the accelerating torque that results from the temporary reduction in active power at each of the machine terminals immediately following the application of the fault. However, as the fault is cleared, the speeds return to steady state, after a few cycles of oscillations. Due to the dropping of the line, the speeds do not return to zero, as the network configuration is now changed. Note that all the machines keep in synchronism throughout the response, as is expected of a stable system.
The load angle responses are shown in fig. (22). The top figure shows the absolute values of the angles, while the bottom figure shows the relative load angles of machines #2 and #3 with respect to that of machine #1 – these are respectively denoted as $\delta_{21} = \delta_2 - \delta_1$ and $\delta_{31} = \delta_3 - \delta_1$.

From the top figure we see that for the first second the load angles do not change. This is because the system is in equilibrium, due to the absence of external stimulus. When the reference voltage of machine one is increased by 5%, the load angles slip back to allow for the required increase in the machines air-gap fluxes. When the fault is applied at $t = 6$ s, the load angles initially slip forward quite rapidly to allow for the generation of the extra power required to cover the fault. However, when the fault is removed and the faulted line is cleared, the absolute value of the load
angles resume their slide down. This is in line with the speed responses, where the speeds settle to lower steady state levels after the removal of the faulted line.

The bottom figure shows the responses of the relative load angles of machines #2 and #3 with respect to machine #1. It is quite clear that these responses reflect those of the absolute load angles. More importantly, the figure shows that the machines remain in synchronism, with the second swing much smaller than the first, consistent with the fact that the system settles into a stable new operating condition after the clearance of the fault. The figure also shows that when the fault is cleared and the line is dropped, the relative load angles settle to new equilibrium levels, after a few cycles. These levels are determined by the nodal admittance matrix of the newly configured transmission system.
Figure (22) response of load angles. [TOP: absolute load angles, $\delta_1$, $\delta_2$, and $\delta_3$; BOTTOM: relative angles $\delta_{21} = \delta_2 - \delta_1$ and $\delta_{31} = \delta_3 - \delta_1$]

Fig. (23) shows the responses of the terminal voltages of the three machines for the same simulation conditions. The top figure shows the overall response to the stated simulation conditions. The figure also reveals that after the fault is removed and the faulted line is dropped, the voltages are eventually restored to their respective command values. The middle figure shows in more detail the response during the first 6 seconds. It clearly shows that following the application of the initial 0.05 pu step increase in $V_{REF,1}$, the terminal voltage magnitude of machine #1, $|v_{N,1}|$, tracks the change in the command level as expected. The terminal voltages of the other two machines, after an initial transient are restored to their initial values by the respective voltage regulation systems (exciters) on these machines. The bottom figure is a blow up of the top figure during the application and removal of the fault. From this figure it can be seen that when the fault is applied, all three terminal voltages experience an initial excursion from their nominal values, due to the changes in the power flows in the network.
Figure (23): terminal voltage response of machines #1, #2, and #3. [TOP: Overall response; MIDDLE: first 6 seconds; BOTTOM: blown up responses between \( t = 5.98 \) s and \( t = 6.14 \) s].

4. Conclusions

This paper presents a new approach to the modelling and simulation of nonlinear interconnected systems. A key feature of this approach is that the system is decomposed into self-contained modules with well defined sets of inputs and outputs, so that each module can be linked to the next easily, using an intuitive graphical interface, to configure any structure for the system under study. Another key feature of our approach is that it permits fault detection and identification studies of nonlinear interconnected dynamical systems. It is shown that the task of constructing highly complex systems from their basic components is greatly facilitated using our approach, so
that analysis may be carried out at the component as well as at the system level without any extra programming or reconfiguration.

The capabilities of the software package developed have been illustrated through two case studies: (i) a single-machine infinite bus system; and (ii) a three-machine, nine-bus power system. For each case, a fault is placed on one of the transmission lines. In both case studies, it is shown that a system can be easily assembled by combining any appropriate selection of component models of the machine, exciter, governor and network. The simulation results show the responses of the main dynamic indicators of the system such as load angle, speed and terminal voltage reflect the actual physical behaviour of the system under the postulated conditions.

5. References