2019

MATLAB GUI Based Educational Simulation Tool Box for Power Analysis

Tharuka Senevirathne
Minnesota State University, Mankato

Recommended Citation

Senevirathne, T. (2019). MATLAB GUI based education simulation tool box for power analysis [Master's thesis, Minnesota State University, Mankato]. Cornerstone: A Collection of Scholarly and Creative Works for Minnesota State University, Mankato. https://cornerstone.lib.mnsu.edu/etds/902/

MATLAB GUI BASED EDUCATIONAL SIMULATION TOOL BOX FOR POWER ANALYSIS

A Thesis
Presented to
The Academic Faculty

By

Tharuka Senevirathne

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Electrical Engineering in the
Department of Electrical and Computer Engineering and Technology

Minnesota State University, Mankato

May 2019

MATLAB GUI BASED EDUCATIONAL SIMULATION TOOL BOX FOR POWER ANALYSIS

Approved by:

. . . .

Dr. Vincent Winstead, Advisor
Department of Electrical and Computer Engineering and Technology
Minnesota State University, Mankato

. . . .

Dr. Muhammad Khaliq
Department of Electrical and Computer Engineering and Technology
Minnesota State University, Mankato

. . . .

Dr. Jianwu Zeng
Department of Electrical and Computer Engineering and Technology
Minnesota State University, Mankato

Date Approved: April , 2019

Acknowledgments

First, I would like to thank my adviser, Dr.Vincent Winstead. His guidance allowed me to freely explore my topic and deepen my understanding. Dr.Winstead always helped me whenever I ran into a trouble spot or had a question about my research or writing. I would also like to extend my gratitude to members of my committee, Dr.Muhammad Khaliq and Dr.Jianwu Zeng.

Finally, I must express my very profound gratitude to my parents and to my wife Shalinie for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

MATLAB GUI BASED EDUCATIONAL SIMULATION TOOL BOX FOR POWER ANALYSIS

By

Tharuka Senevirathne

Submitted to the Department of Electrical and Computer Engineering and Technology
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering in the
Minnesota State University, Mankato
May 2019

Abstract One of the most important and complex tasks in power engineering is the analysis of power systems under fault conditions. The detection and analysis of these faults are important to guarantee that the dependability and stability of the power system does not decline as a result of a critical event such as a fault. This thesis will conduct research on how a power system under fault conditions behaves and will examine the various scenarios of faults such as three-phase, single line-to-ground, line-to-line, and double line-to-ground faults. A simplified method based on symmetrical components is used to construct the mathematical models that calculate the fault currents and the fault voltages. A MATLAB/ Graphical User Interface based simulation tool has been developed to use as an educational tool and these models have been integrated into a MATLAB/ Graphical User Interface. This GUI has an option to enter input data and select the fault type and will provide the calculation of three-phase, single line-to-ground, line-to-line, and double line-to-ground faults on power systems. These calculated fault currents and voltages display on a GUI screen along with the time variations of signal. A so called ”Educational resources zone” was also added with various text from literature and videos on power system analysis so that the students can have a better understanding. In this way, the developed MATLAB GUI simulation tool can be used by students to check the accuracy of the hand calculations and to observe the

transient waveforms of fault current and voltages.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Faults occur in power systems when circuit insulation fails due to system over-voltages caused by lightning or other natural events like heavy winds or snow, or due to other mechanical causes. This will result in fault current and this current is determined by the generator voltages and by the system impedance between the machine voltages and the fault[2]. These fault currents can be several times higher than the normal operating currents. Because of this, when a fault occurs in a power system, it can cause severe damage to the system which will result in power failure, permanent damage to the equipment, personal injury or even death.

So as to avert such an event, fault analysis of power systems was introduced. Fault analysis is defined as the way towards assessing the system voltages and currents under different kinds of short circuit conditions. There are several types of short circuit conditions that can happen in a power system. a short circuit involving all phasors in contact with each other in a three phase system is known as a symmetrical three phase fault. A Line to Line fault occurs when only two phasors are in contact. A Single line to ground fault happens when a phase is in contact with a ground conductor or with a neutral conductor.

Double line to ground faults happen when two phasors are in contact with the ground or neutral conductor[6]. Short circuit analysis or the fault analysis of a power system yields the currents and voltages during and after the fault and these measurements can be used to discover the required safety features and the required over-current protection system[4]. Fault analysis calculations are simplified by using the symmetrical components method. This method provides the simplified way to analyze the unbalanced systems during faults. It involves transforming the three phasors into a new set of components called symmetrical components, which are defined as zero, positive and negative sequences[3].

As mentioned above, fault analysis is a main part of analysis of a power system. Some concepts and theoretical calculations can be really complex and students can find it difficult to understand. Using a visual tool to calculate and present the results makes the complex concepts and theories of fault analysis more understandable. There have been many power system analysis tools developed in the past using a Graphical User Interface (GUI) platform to help understand and to get more accurate calculations. Graphical User Interfaces (GUI) is one of the visual tools to provide the user with interactive visual communication with the tasks on hand and the solution process[6]. In this thesis, a MATLAB GUI based educational power system analysis tool has been developed to calculate the currents and voltages during the fault conditions. This tool provides the calculation of three phase fault, Single phase to ground, phase to phase and double phase to ground. Also it has the ability to present the graphical transient variations of fault current and fault voltage signals during the fault on the GUI screen. This tool also includes the "Educational Resources Zone" which includes series of videos and text documents.

1.1 Types of Fault

There are two types of faults which can happen on any power systems. One is symmetrical faults and the other is asymmetrical faults. Often times faults that occur are asymmetrical faults like single line to ground, line to line and double line to ground faults. Balanced three phase faults can be identified as a symmetrical faults. Also, faults can be classified as shunt faults and series faults. Series faults are those faults which occur in the impedance of the line and do not involve neutral or ground, Also it does not involve any interconnection between the phasors. Shunt faults are the most common type of fault and they involve unbalanced faults between phasors or between the ground and phasors[7]. This thesis only considers shunt faults. These shunt faults can be categorized into four types.

1. Single line-to-ground fault

2. Line-to-line fault

3. Double line-to-ground

4. Three-phase fault

In the analysis of power systems under fault conditions, it is necessary to make a distinction between the types of fault to ensure the best results possible in the analysis. In a three phase power system, the types of fault that can occur are classified by the combination of conductors or buses that are faulted together.

1.1.1 Single line-to-ground fault

This type of fault occurs on a transmission line when one phase of any transmission line comes in a contact with the ground or neutral conductor. There are many common reasons for this type of failure like ice, high-speed wind, falling tree or any other incident. 70% of all transmission lines faults fall under this category[4]. A Single line to ground fault with phase a in contact with ground is shown in figure 1-1:

1.1.2 Line-to-line fault

This type of fault occurs when the two phasors come in contact with each other. The Most common reasons are high winds and tree impacts. Out of all transmission line faults, 15% are considered line-to-line faults.[4]. A Line to line fault between phase b an phase c is shown below in Figure 1-2.

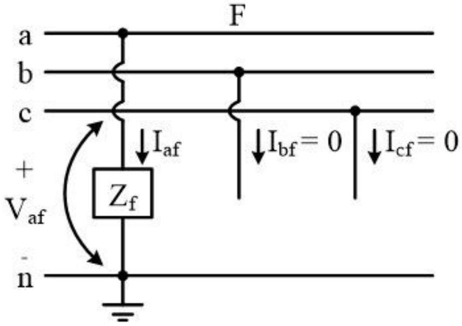


Figure 1-1: Single Line-to-ground fault.[4]

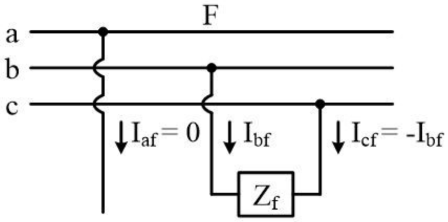


Figure 1-2: Line-to-Line fault.[4]

1.1.3 Double line-to-ground

Double line to ground faults can occur when two phasors comes in contact with ground via either tree impact or other incident. Unlike the Single line to ground fault, this fault involves two of the phasors instead of one. 10% of all transmission lines faults are under this type of fault[4].

1.1.4 Three-phase fault

A three phase bolted fault is the condition that all three phasors are in contact with each other. While this type of fault does not happen often it is the most severe type of fault with maximum fault current. Only 5% of all transmission lines faults are three phase faults[4]. A general representation of a three phase fault is shown in the figure 1-4:

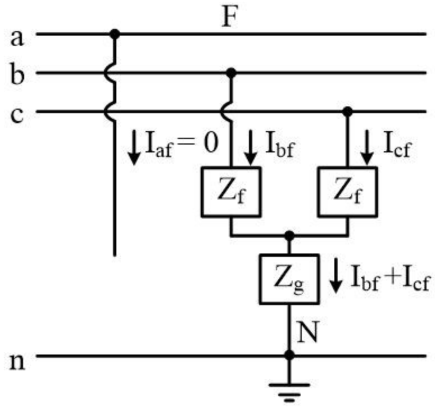


Figure 1-3: Double Line-to-ground fault.[4]

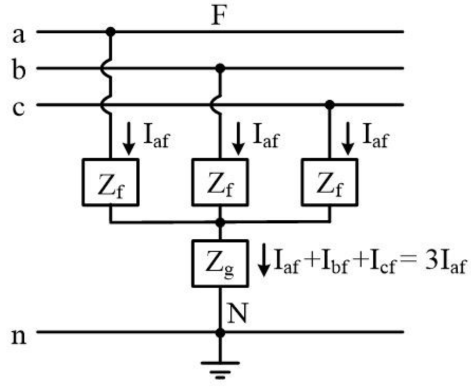


Figure 1-4: Three phase fault[4]

1.2 Method of Analysis

Most of the time short circuited power systems are not the balance three phase systems and it is either not possible or extremely difficult to solve the system using per phase analysis. A first step requires transformation in to their symmetrical components to find the fault currents and fault voltages.In 1918 C.L Fortescue introduced a method named symmetrical component method to do this and analyze any unbalanced power system[3]. Using this method, solving such unbalanced system was simplified using a balanced representation. This symmetrical component method is considered as the basis of all traditional fault analysis approaches to solve unbalanced power systems[3].

CHAPTER 2

Analysis of Faulted Power System

Under typical conditions, a power system operates under balanced conditions with all components carrying normal currents with the normal bus voltages. When a short circuit occurs in the system, normal operation can be disturbed. A short circuit is a failure in the system that obstructs the normal current flow. "A short circuit fault happens when the insulation of the system fails resulting in low impedance path either between phases or phase(s) to ground"[5]. This failure can produce very high currents to flow in the circuit. Because of this, when a fault occurs in a power system, it can cause severe damage to the system which will result in power failure, permanent damage to the equipment, personal injury or even death. So as to avert such an event, Fault analysis of power systems was introduced. Short circuit analysis or the fault analysis of a power system presents the currents and voltages during and after the fault. Using these currents and voltages, required over-current protection systems can be designed.

2.1 Symmetrical Components and Fortescue's Theory

The symmetrical component method is basically a modeling technique that permits systematic analysis and design of three-phase systems. The majority of faults in power systems are asymmetrical so obtaining a direct solution of such a circuit can be very difficult. The Fortescue method was introduced by C.L. Fortescue in 1918[2].This method of symmetrical components allows one to solve unbalanced systems using balanced techniques. The use of this technique simplifies the analysis procedure of unbalanced systems and in addition helps improve in rising the understanding of the system behavior throughout the fault conditions. Fortescue sequestered asymmetrical 3-phase voltages and currents into three sets of symmetrical components. Fortescue's theorem suggests an unbalanced system of "n" related phasors can be resolved into "n" systems of balanced phasors called Symmetrical Components of the original phasors. The "n" phasors of each set of components are equal in length, and the angles between adjacent phasors of the set are equal. Any unbalanced fault can be converted into three independent symmetrical components which differ in the phase sequence[6]. These components consist of a positive sequence, negative sequence and a zero sequence.

2.1.1 Positive Sequence Components

Positive sequence components consist of three phasors with equal magnitudes and 120° apart from each other. The phase sequence are in the same order of the original phasors. The same case applies for the positive current phasors. This sequence is also called the "abc" sequence and is usually denoted by the symbol "+" or "1" [4]. Representation of positive sequence components are shown in figure 2-1.

2.1.2 Negative Sequence Components

Negative sequence components consists= of three phasors with equal magnitudes and 120° apart from each other. The phase sequence are in the opposite order of the original phasors. The same case applies for the negative current phasors. This sequence also called the "acb"

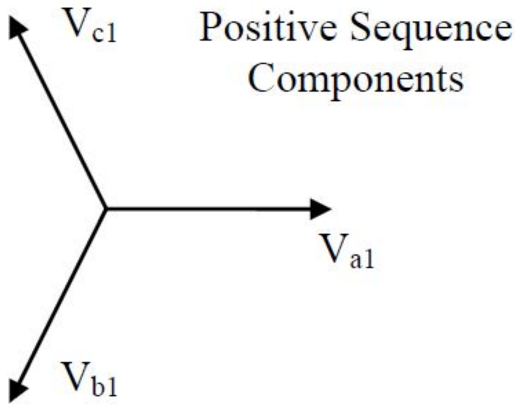


Figure 2-1: Positive sequence components.[4]

sequence and is usually denoted by the symbol "-" or "2"[4].Representation of negative sequence components are shown in figure 2-2.

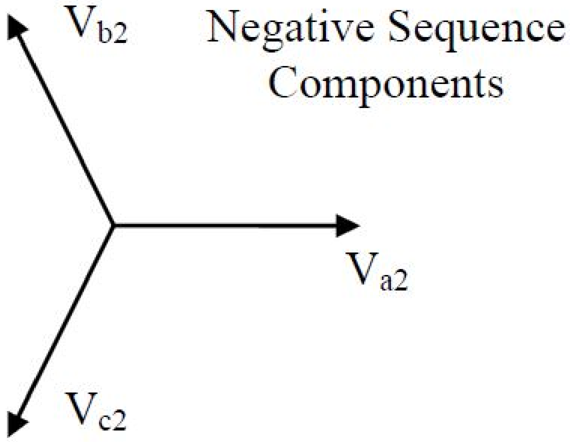


Figure 2-2: Negative sequence components.[4]

2.1.3 Zero Sequence Components

Zero sequence components consist of three phasors with equal magnitudes and zero phase displacement. The phasor components are in phase with each other. This sequence is known by the symbol "0". Under an asymmetrical fault condition, this sequence symbolizes the residual electricity in the system in terms of voltages and currents where a ground or a fourth wire exists. Zero sequence components results from ground currents entering to the power system through any grounding point within the electrical system.

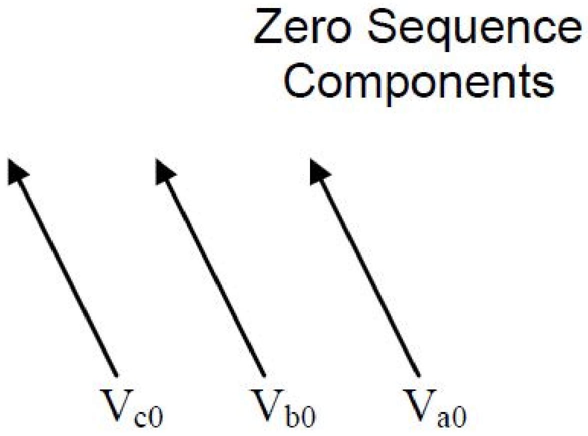


Figure 2-3: Zero sequence components.[4]

Each of the original unbalanced phasors are the sum of its symmetrical components. Here Va, Vb and Vc are the original phase voltages and the three phasors of voltages are shown by subscripts a, b and c . The subscripts 1, 2 and 0 refer to positive, negative and zero sequence components, respectively.

$$V_a = V_{a0} + V_{a1} + V_{a2} \tag{2.1}$$

$$V_b = V_{b0} + V_{b1} + V_{b2} \tag{2.2}$$

$$V_c = V_{c0} + V_{c1} + V_{c2} \tag{2.3}$$

An ”a” operator is developed to indicate the phase difference between these phasors. where ”a” operator is represented below a phasor with unity magnitude and a phase angle of 120°.[2]

$$a = 1\underline{/120^\circ} \tag{2.4}$$

So the following equations can obtained from 2.4:

$$a^2 = 1\underline{/-120^\circ} \tag{2.5}$$

$$a^3 = 1\underline{/0^\circ} \tag{2.6}$$

Using operator ”a”, three phase components have been expressed using single phase component. Every phase b and phase c component in equation 2.1 to 2.3 can be defined with phase a component and operator ”a”

$$V_{b1} = a^2 V_{a1} \tag{2.7}$$

$$V_{c1} = a V_{a1} \tag{2.8}$$

$$V_{b2} = a V_{a2} \tag{2.9}$$

$$V_{c2} = a^2 V_a \tag{2.10}$$

$$V_{b0} = V_{a0} \tag{2.11}$$

$$V_{c0} = V_{a0} \tag{2.12}$$

Substituting equations (2.7-2.12) into equations (2.1-2.3) respectively, we can get the phase voltages.

$$V_a = V_{a0} + V_{a1} + V_{a2} \tag{2.13}$$

$$V_b = V_{a0} + a^2 V_{a1} + a V_{a2} \tag{2.14}$$

$$V_c = V_{a0} + a V_{a1} + a^2 V_{a2} \tag{2.15}$$

The equations above can be written in compact matrix form as:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} \tag{2.16}$$

Defining A as a 3x3 transformation matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \tag{2.17}$$

Now the Equation 2.16 can be written as:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = A \begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} \tag{2.18}$$

This equation matrix can be inverted in order to obtain the positive, negative and zero sequences from the system phasors:

$$\begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} = A^{-1} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{2.19}$$

Where A^{-1} is the inverse matrix of A:

$$A^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \tag{2.20}$$

Fault voltages and currents can be obtained by using these equations. Also, these equations can express the line currents and the line-to-line voltages of any power system under fault conditions.

2.2 Sequence Networks of a Power System

The sequence impedance network is described as a balanced network that is equivalent to the balance power system under an imagined working condition so that only single sequence components of voltage and current are present within the system. Asymmetrical faults can be calculated using the symmetrical components at different points of a power system network. The positive sequence network determines the load flow in power system[8].

All power systems consist of positive, negative and zero sequence networks and three sequence currents flow through these networks. Four main different unbalanced faults can be represented by interconnecting these three sequence currents in different ways. Actual phase currents and phase voltages can be determined by calculating these three sequence current and voltages during the faults.

A short circuit in the power system means that the system is no longer balanced and in a unbalanced state of operation. This unsymmetrical position of the power system is replaced by the balanced, positive and negative symmetrical sequence set and zero sequence set. The post fault voltage and current are determined by the response of the system using each component set[8].

2.2.1 Sequence Networks of a loaded Synchronous Generator

A three-phase synchronous generator with synchronous per phase impedance Z_s with neutral grounded through an impedance Z_n is shown in Figure 2-4. The generator is supplying a balanced three phase load. The generator voltages E_a, E_b and E_c are balanced. So these voltages can be treated as positive sequence set and can be shown in the equation below[5]:

$$[E]_{abc} = \begin{bmatrix} 1 \\ a^2 \\ a \end{bmatrix} [E_a] \tag{2.21}$$

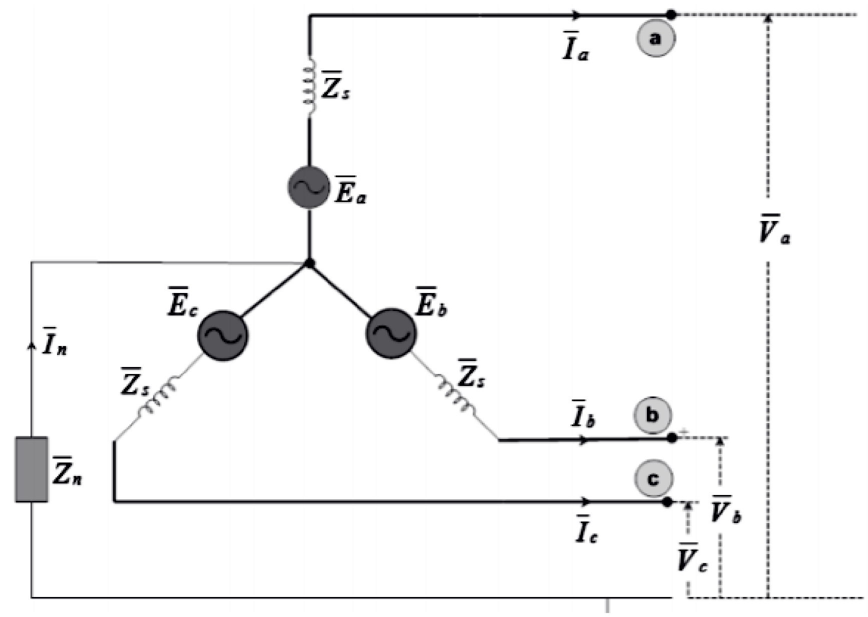


Figure 2-4: Three phase synchronous generator supplying a balanced load.[5]

Using equation 2.21, one can write the following equation,

$$[V]_{abc} = [E]_{abc} - [Z]_{abc}[I]_{abc} \tag{2.22}$$

where,

$[V]_{abc}$ are the terminal phase voltage

$[I]_{abc}$ are the terminal phase currents

$[Z]_{abc}$ represents the impedance matrix of the system

Phase components in equation (2.22) can be replaced by equivalent sequence quantities, using the transformation equation (2.19). And is shown in equation 2.23. The current equation can be written in the same way.

$$[V]_{012} = [E]_{012} - [Z]_{012}[I]_{012} \tag{2.23}$$

where, $[Z]_{012}$ is generator sequence impedance Matrix and is defined as:

$$[Z]_{012} = \begin{bmatrix} Z_s + 3Z_n & 0 & 0 \\ 0 & Z_s & 0 \\ 0 & 0 & Z_s \end{bmatrix} \tag{2.24}$$

E_{012} is the generated sequence voltage vector and is defined as $\begin{bmatrix} 0 \\ E_a \\ 0 \end{bmatrix}$ since generated voltages only contain the positive sequence component and are always balanced. Substituting these two values in equation (2.23) yields,

$$\begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} = \begin{bmatrix} 0 \\ E_a \\ 0 \end{bmatrix} - \begin{bmatrix} Z_0 & 0 & 0 \\ 0 & Z_1 & 0 \\ 0 & 0 & Z_2 \end{bmatrix} \begin{bmatrix} I_{a0} \\ I_{a1} \\ I_{a2} \end{bmatrix} \tag{2.25}$$

Expanding the above equation, separate equations can be written for each of the sequence components as[5]:

$$\boxed{\begin{aligned} V_{a0} &= -Z_0 I_{a0} \\ V_{a1} &= E_a - Z_1 I_{a1} \\ V_{a2} &= -Z_2 I_{a2} \end{aligned}} \tag{2.26}$$

The three sequence networks of a synchronous generator are shown in Figure 2-5. Looking at figure 2-5 and the equation 2.26, it is shown that the sequence components are independent from each other which means the current of a particular sequence produces a voltage drop in that sequence only[5].

2.2.2 Sequence networks of a transmission line

A transmission line does not have current or voltage sources present in its equivalent model and is considered a passive device. The line behavior does not change no matter the direction

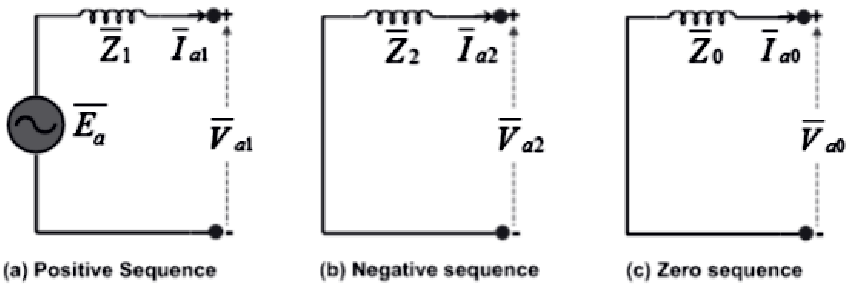


Figure 2-5: The sequence networks of a synchronous generator.[5]

of the current. So this is also known as a bilateral device because of these qualities, the positive sequence voltages produce the same voltage drops as negative-sequence voltages. This means that the phase sequence of the applied voltage makes no difference. Therefore, the impedance of a transmission line for its positive-sequence and negative-sequence are the same[4]. $Z_1 = Z_2$.

However, the zero sequence currents are in phase and current flows through the conductors and come back through ground wires. The zero sequence impedance Z_0 is different from the positive impedance of transmission line,Z_1, and the negative impedance of transmission line, Z_2, as a result of the inclusion of the ground return path. Usually Z_0 is three times larger than Z_1 or Z_2[5]. Sequence networks of the transmission lines are shown in Fig 2.6

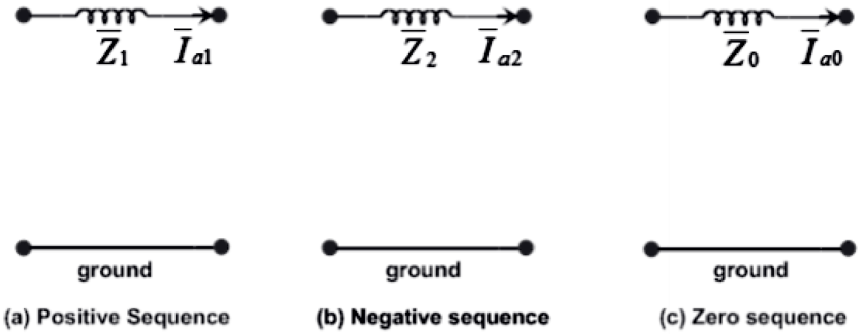


Figure 2-6: The sequence networks of a transmission line[5]

2.2.3 Sequence networks of a transformer

Transformer leakage impedance is equaled to its positive sequence series impedance. Since a transformer is also a static device like a transmission line, leakage impedance will not change with the direction of the phase sequence of applied voltage. So like the transmission line,the positive and negative sequence impedances are equal. As a result, the transformer's negative sequence is also equal to its leakage impedance [5], i.e

$$Z_1 = Z_2 = Z_{leakage} \tag{2.27}$$

The Zero sequence currents flow through the transformer if current paths continue in both the primary and secondary sides of the transformer. For such cases zero sequence impedance is also equal to positive and negative sequence impedance for the transformer $Z_1 = Z_2 = Z_0$.

Again since transformer is also a static device, the positive and negative sequence networks are identical to the positive and negative sequence networks of the transmission line as shown in figure 2-6. However the zero sequence network of transformer depends on the winding connections and whether the neutrals are grounded or not. One thing to keep in mind is that an open circuit will exist on either primary or secondary sides if there is no ground to return to for the currents. The different cases of three-phase transformer connections and their equivalent zero-sequence networks are shown below.

1. **Star-Star connections with both neutrals grounded**

   Since both the neutrals are grounded, paths for currents to flow exist in both primary and secondary sides. The sum of unbalanced phasors are equal to three times the zero sequence current[5]. Therefore, the zero sequence current flows in both primary and secondary sides and the zero sequence equivalent circuit is shown in the figure 2-7:

2. **Star-Star connections with only one neutral grounded**

   In the case of only one neutral grounded, the currents of the ungrounded winding should adds up to zero, and zero sequence currents cannot exist in that winding. Therefore,

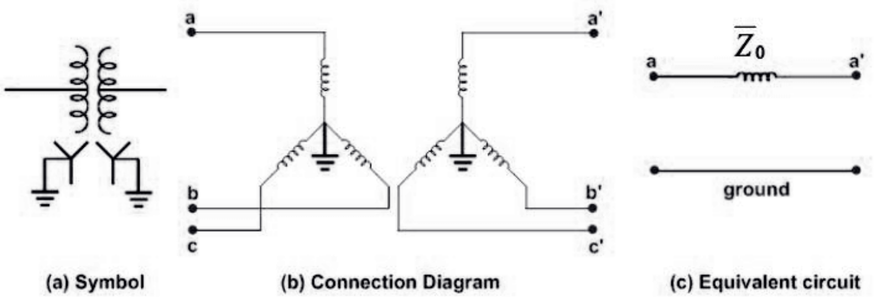


Figure 2-7: The zero-sequence equivalent circuit of a Star-Star transformer with both neutrals grounded.[5]

zero sequence currents cannot exists in grounded winding either. In this case, the transformer is shown as an open circuit between the primary and secondary sides[5]. An equivalent circuit is shown in the figure 2-8:

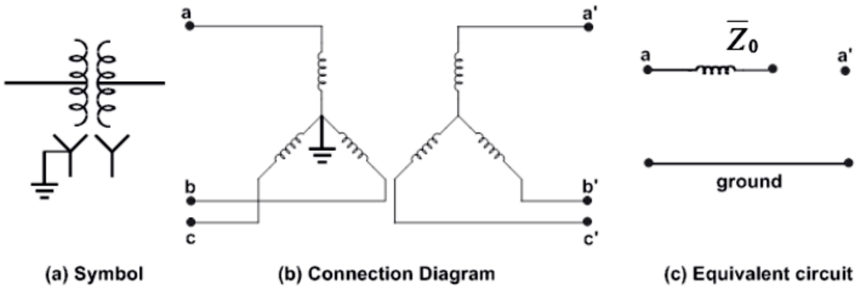


Figure 2-8: The zero-sequence equivalent circuit of a Star-Star transformer with only one neutral grounded.[5]

3. **Star-Star connections with no neutral grounded**

   Similar to case 2, The phase currents should add up to zero in both the windings and therefore the zero sequence currents can not exist in any winding. In this case, the transformer is shown as an open circuit between the primary and secondary sides[5]. An equivalent circuit is shown in figure 2-9:

4. **Star-delta connections with neutral grounded**

   In this case, a zero sequence current will circulate in the delta winding as a result of the grounded star winding. Still this zero sequence current can only exist in the closed

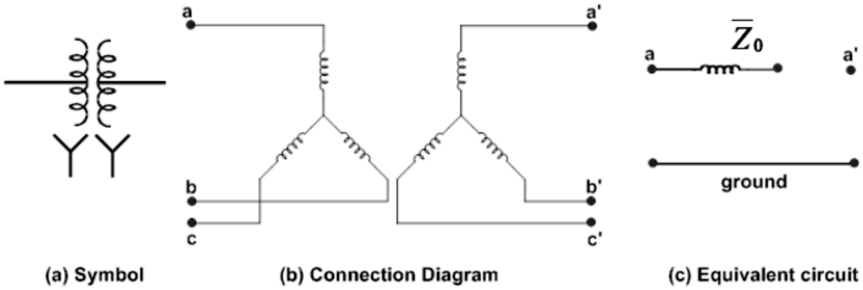


Figure 2-9: The zero-sequence equivalent circuit of a Star-Star transformer with both neutrals ungrounded.[5]

delta winding and not on the line side of the winding. Because of this,an open circuit exist between the star and the delta sides. Since the zero-sequence currents can exist on the line-side of the grounded star winding, the zero-sequence equivalent circuit can show as leakage impedance if the transformer is connected to ground on the star side of the transformer and an open circuit exists between the two windings[5]. The equivalent circuit for this connection is shown in Figure.2-10

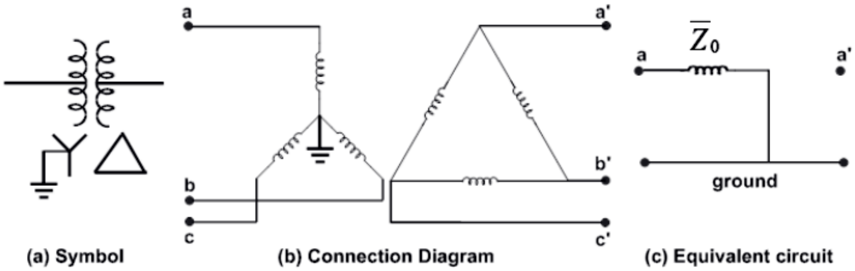


Figure 2-10: The zero-sequence equivalent circuit of a Star-Delta transformer with Star-side neutral grounded.[5]

5. **Star-delta connections with no neutral grounded**

   In this case also since no neutral is grounded, no zero sequence currents can flow in either star side or the delta side of the transformer[5]. Therefore, in this case the transformer is shown as a open circuit between two windings in the figure.2-11 below:

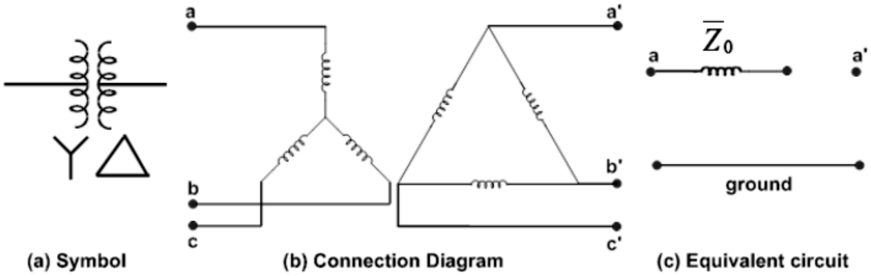


Figure 2-11: The zero-sequence equivalent circuit of a Star-Delta transformer with Star-side neutral ungrounded.[5]

6. **Delta-delta connections with no neutral grounded**

   Since no neutrals are grounded, zero sequence currents can only exit in closed delta windings and not on the line side of both sides. Thus, an open circuit exits between windings. However because of the currents circulating in the closed delta windings, the zero sequence impedance is shown with ground[5]. The equivalent circuit for this connection is shown in Figure 2-12.

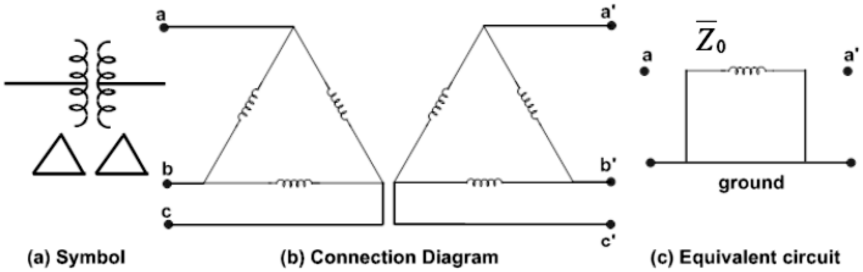


Figure 2-12: The zero-sequence equivalent circuit of a Delta-Delta transformer.[5]

In a special case where the neutral is grounded trough a impedance Z_n, the total Z_0 would be:

$$Z_{0total} = Z_0 + 3Z_n \tag{2.28}$$

This configuration is shown in the figure below:

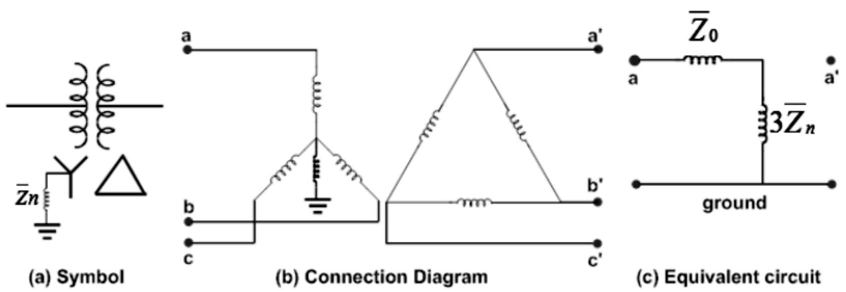


Figure 2-13: The zero-sequence equivalent circuit of a Star-Delta transformer with neutral grounded through impedance.[5]

2.3 Per unit Calculation

Per unit system is a method used in power system analysis to express the values of voltage, current, impedance or the admittance in per unit of the base quantity. Rated values are usually taken as the base quantity. For example,

Base voltage = rated voltage or

Base current = rated current of the machine.

The definition of the per unit system can be expressed as the ratio of actual value in any unit and the base value of the same unit per unit values are dimensionless[4].

$$Per\,Unit\,Value = \frac{Actual\ value\ in\ any\ unit}{Base\ value\ in\ the\ same\ unit} \tag{2.29}$$

The value of base power and the base voltage were chosen first to calculate the per unit values. Based on these values, other base values can be obtained automatically as shown below.

$$Per\ unit\ KV = \frac{Actual\ KV}{Base\ KV} \tag{2.30}$$

$$Base\ current\ I_B = \frac{KVA_B}{KV_B} \tag{2.31}$$

$$Per\ unit\ current\ I_{pu} = \frac{Actual\ value\ of\ current}{Base\ current} \tag{2.32}$$

By putting the base current value from equation 2.33 in equation 2.34 we get:

$$Per\ unit\ current\ I_{pu} = \frac{Actual\ value\ of\ current}{KVA_B/KV_B} \tag{2.33}$$

$$Per\ unit\ current\ I_{pu} = \frac{Actual\ value\ of\ current * KV_B}{KVA_B} \tag{2.34}$$

And

$$Base\ impedance\ Z_B = \frac{Base\ KV * 1000}{I_B} \tag{2.35}$$

$$Z_{pu} = \frac{Actual\ Impedance}{Base\ Impedance} \tag{2.36}$$

Now by putting the values of base impedance

$$Z_{pu} = \frac{Actual\ Impedance * KVA_B}{(KV_B)^2 * 1000} \tag{2.37}$$

CHAPTER 3

Fault Calculation In Three Phase Systems

3.1 Formation of bus admittance matrix (Y_{BUS})

In any power system, power is injected in to buses from power sources like generators. Loads take the power away from the system. There are buses with only power generators and some are with only loads. There may be some buses with generators and loads and some with static capacitors for power compensation. When there is a power deficiency in a bus, surplus power at some buses are transported through transmission lines to the bus with a power deficiency[4]. This concept was used in forming the Y_{BUS} matrix.

Any power system with 'n' number of buses can be represented form using the following equation:

$$\boxed{[I_{BUS}] = [Y_{Bus}][V_{BUS}]} \tag{3.1}$$

Where,

I_{BUS} is the injected bus currents

V_{BUS} is the bus voltages

Y_{BUS} is the (n x n) bus admittance matrix

This equation can be rewrite in matrix formation for 'n' number of buses as follows:

$$\boxed{\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \dots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix}} \tag{3.2}$$

and the elements of the Y bus admittance can be calculated as below:

For i=[1,2,3,...,n],

Y_{ii}= total sum of all the admittances connected to bus "i"

Y_{ij}= if the bus 'i' and bus 'j' are connected together, then Y_{ij} is equal to the negative of the total sum of all the admittances connected between bus "i" and bus 'j'

Y_{ij}= 0, when bus 'i' and bus 'j' are not physically connected together.

3.1.1 Unbalanced fault analysis using impedance matrix (Z_{BUS})

Unbalanced faults can be calculated using the Thevenin's equivalent networks at fault point of three sequence networks. As discussed in previous chapters, all three sequence networks are independent from each other and impedance matrices of these can be calculated separately as $[Z_{BUS}^{(0)}]$, $[Z_{BUS}^{(1)}]$ and $[Z_{BUS}^{(2)}]$. The Z_{BUS} matrix is suitable to use for fault analysis as its diagonal elements are the Thevenin's impedance of the network as seen from different buses[5]. For example, if the fault occurs at the k^{th} bus of the system, the Thevenin's equivalent impedances can be found from the impedance matrices as $Z_{kk}^{(0)}$, $Z_{kk}^{(1)}$ and $Z_{kk}^{(2)}$. Therefore, calculation of impedance matrix is a important step in the fault analysis and it can be obtained by taking the inverse of admittance matrix $[Y_{BUS}]$.

$$\boxed{[Z_{BUS}] = [Y_{BUS}]^{-1}} \tag{3.3}$$

Therefore, equation 3.1 can be rewrite as:

$$\boxed{[V_{BUS}] = [Z_{BUS}][I_{BUS}]} \tag{3.4}$$

3.2 Three-phase fault

Three phase faults occur when all three phasors are in contact with each other simultaneously. Since all three phasors are involved, the system remains balanced during these faults. Because of this, three phase faults are called symmetrical faults and the fault analysis are done on per phase basis[5]. These faults can be very severe but this event does not happen frequently. When these faults occur, it can cause serious damage to the electrical power system even though system remains balanced. General representation of the three phase fault is shown in figure 3-1 below.

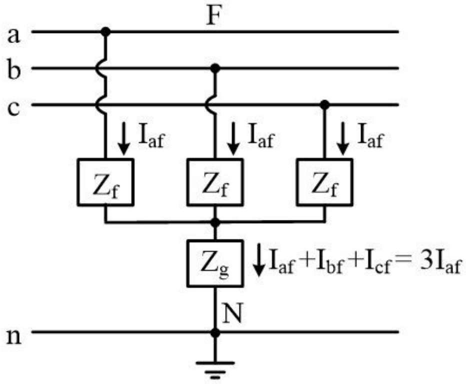


Figure 3-1: Three phase fault.[4]

Here Z_f is the fault impedance and it is applied equally to the all three phases. If Z_f=0, then the fault is known as the solid fault or the bolted fault. Fault currents reach to its maximum value with the bolted fault. Usually, the fault is created by connecting fault impedance at the fault location in the power system. Then, as discussed in the previous section, fault can be solved using the Thevenin's equivalent network as seen from the faulted bus point. This can be obtained using the sequence network diagram. Typical sequence network diagram of a balanced three phase fault is shown in the figure 3-2:

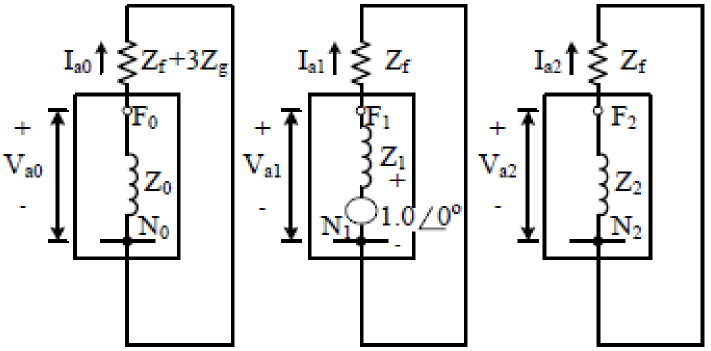


Figure 3-2: Sequence network diagram of a balanced three phase fault.[4]

Looking at the figure, it can be seen that only the positive sequence network has the internal voltage and because of that, only the positive sequence current and voltage needs to be calculated. Also, the pre-fault voltage for all the buses are taken as $1.0\underline{/0^\circ}$. Hence, the corresponding sequence components of the currents are

$$\boxed{\begin{aligned} &I_{a0} = 0 \\ &I_{a1} = \frac{1.0\underline{/0^\circ}}{Z_1 + Z_f} \\ &I_{a2} = 0 \end{aligned}} \tag{3.5}$$

and for the bolted faults, where Z_f=0,

$$\boxed{I_{a1} = \frac{1.0\underline{/0^\circ}}{Z_1}} \tag{3.6}$$

The phase currents can be obtained using the transformation matrix in equation 2.17:

$$\boxed{\begin{bmatrix} I_{af} \\ I_{bf} \\ I_{cf} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ I_{a1} \\ 0 \end{bmatrix}} \tag{3.7}$$

From figure 3.2, the equations for sequence components of the fault voltages can be obtained:

$$\boxed{\begin{aligned} &V_{a0} = 0 \\ &V_{a1} = Z_f I_{a1} \\ &V_{a2} = 0 \end{aligned}} \tag{3.8}$$

Same procedure as before, the phase voltages can be acquired using the transformation matrix in equation 2.17:

$$\boxed{\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ V_{a1} \\ 0 \end{bmatrix}} \tag{3.9}$$

For the bolted faults where Z_f= 0,

$$\boxed{\begin{aligned} &V_a = 0 \\ &V_b = 0 \\ &V_c = 0 \end{aligned}} \tag{3.10}$$

3.3 Single Line to ground fault

The single line to ground fault which is commonly known as SLG fault occurs when one of the conductors makes contact with either ground or the neutral. This is the most common type of fault and it happens more often than the other types of fault. General representation of the SLG fault is shown in the figure 3-3. Figure shows a fault at fault point 'F' in the power system through a Z_f fault impedance. Phase 'a' is usually assumed to be the faulted phase[4].

In this case of single line to ground fault, all the sequence networks are required for the calculation[5]. Figure 3-4 is showing the sequence network diagram for the SLG fault. Since the generator is not loaded, line to ground currents and the line to ground voltages can be

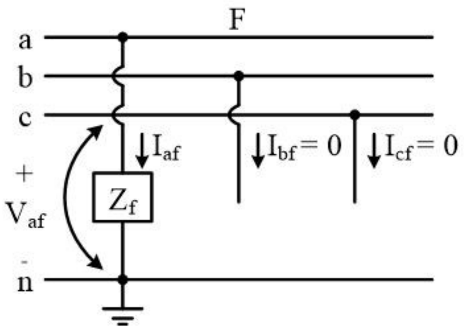


Figure 3-3: Single Line-to-ground fault.[4]

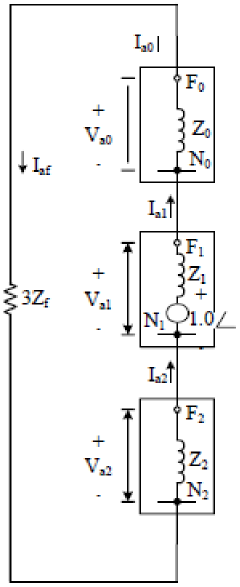


Figure 3-4: Sequence network diagram of a SLG fault.[4]

obtained at the fault point F as follows:

$$\boxed{\begin{array}{l} V_{af} = Z_f I_{af} \\ I_{bf} = 0 \\ I_{cf} = 0 \end{array}} \tag{3.11}$$

Using the inverse transformation matrix A^{-1} in equation 2.20, the symmetrical components of the fault currents can be obtained.

$$\begin{bmatrix} I_{a0} \\ I_{a1} \\ I_{a2} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} I_{af} \\ 0 \\ 0 \end{bmatrix} \tag{3.12}$$

Solving the above equation, the values of symmetrical component of fault current I_{af} can be acquired as:

$$\boxed{I_{a0} = I_{a1} = I_{a2} = \frac{1}{3} I_{af}} \tag{3.13}$$

Same as above, the voltage of phase 'a' can be expressed in regards to symmetrical components from the equation 2.19 as:

$$V_{af} = V_{a0} + V_{a1} + V_{a2} \tag{3.14}$$

Substituting the equation 3.14 with values from equations 2.26-2.28 and applying $I_{a0} = I_{a1} = I_{a2}$ from equation 3.13, the following can be written.

$$V_{af} = 1.0\underline{/0^\circ} - (Z_0 + Z_1 + Z_2) I_{a0} \tag{3.15}$$

Since $V_{af} = Z_{af} I_{af} = 3 Z_{af} I_{a0}$, and from equations 3.11 and 3.13, equation 3.15 can be re written as:

$$3 Z_f I_{a0} = 1.0\underline{/0^\circ} - (Z_0 + Z_1 + Z_2) I_{a0} \tag{3.16}$$

Hence, the symmetrical component I_{a0} of the fault current is

$$\boxed{I_{a0} = \frac{1.0\underline{/0^\circ}}{Z_0 + Z_1 + Z_2 + 3 Z_{af}}} \tag{3.17}$$

Therefore, the fault current can be obtained as

$$\boxed{I_{af} = \frac{3.0\angle 0^\circ}{Z_0 + Z_1 + Z_2 + 3Z_{af}}} \tag{3.18}$$

3.4 Line-to-line fault

Line to line fault is an asymmetrical fault and it occurs when two conductors makes contact with each other. This may occur on either overhead transmission lines or underground transmission lines. Figure 3-5 shows a three phase system with line to line fault representation between phase 'b' and phase 'c' and with the fault impedance of Z_f at the fault point F. For the simplicity of fault calculations, we assume the Phase 'b' and phase 'c' to be the faulted phases[4]. Figure 3-6 shows the sequence network of the line to line fault.

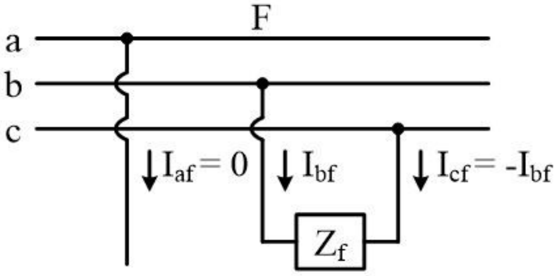


Figure 3-5: Line-to-Line fault.[4]

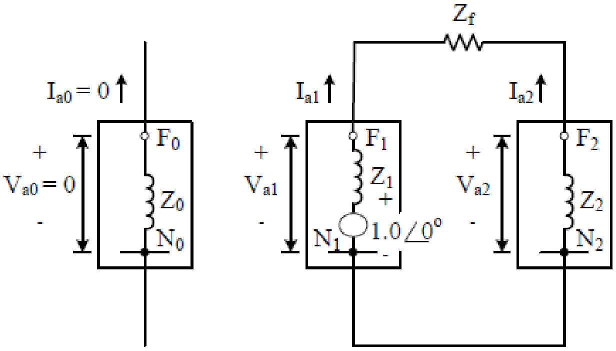


Figure 3-6: Sequence network diagram of a LL fault.[4]

Since the generator is not loaded, the terminal conditions at the fault point are

$$\begin{aligned} I_{af} &= 0 \\ I_{bf} + I_{cf} &= 0 \\ V_{bf} - V_{cf} &= Z_f I_{bf} \end{aligned} \tag{3.19}$$

Substituting the terminal conditions and using the transformation matrix A^{-1} in equation 2.20, symmetrical components of the currents can be obtained as:

$$\begin{bmatrix} I_{a0} \\ I_{a1} \\ I_{a2} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} I_{af} \\ I_{bf} \\ -I_{bf} \end{bmatrix} \tag{3.20}$$

Solving this equation , the symmetrical components of current I_{af} are,

$$\boxed{\begin{aligned} I_{a0} &= 0 \\ I_{a1} &= \frac{1}{3}(a - a^2) I_{bf} \\ I_{a2} &= -I_{a1} \end{aligned}} \tag{3.21}$$

Therefore, the sequence currents can be obtained as:

$$\boxed{\begin{aligned} I_{a0} &= 0 \\ I_{a1} &= -I_{a2} \\ I_{a2} &= \frac{1.0\angle 0^\circ}{Z_1 + Z_2 + Z_f} \end{aligned}} \tag{3.22}$$

Hence, the phase currents during the fault can be calculated using the equation 2.16 for currents.

$$\begin{bmatrix} I_{af} \\ I_{bf} \\ I_{cf} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ I_{a1} \\ -I_{a1} \end{bmatrix} \tag{3.23}$$

Solving the above equation, the expressions for I_{bf} and I_{cf} can be written as :

$$\boxed{\begin{aligned} & I_{af} = 0 \\ & I_{bf} = (a^2 - a)I_{a1} \\ & I_{cf} = -I_{bf} \end{aligned}} \tag{3.24}$$

3.5 Double line-to-ground

Double line to ground fault occurs when two conductors are short circuited and makes contact with the ground. Typical representation of double line to ground fault with fault impedance of Z_f at fault point F and line to ground impedance of Z_g is shown in the figure 3-7:

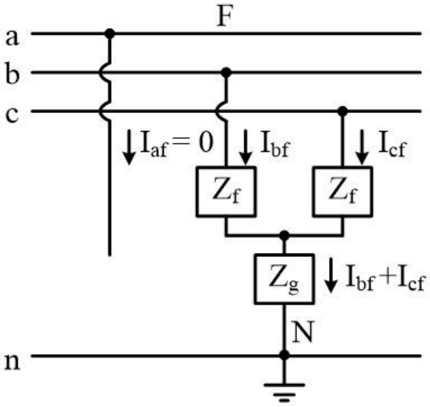


Figure 3-7: Double Line-to-ground fault.[4]

The sequence network of double line to ground fault is shown in figure 3-8: Since this is a unloaded three phase generator, the terminal conditions when the fault happens at the fault point can be obtained as:

$$V_{bf} = V_{cf} = Z_f I_f = Z_f (I_{bf} + I_{cf}) \tag{3.25}$$

$$I_{af} = I_{a0} + I_{a1} + I_{a2} = 0 \tag{3.26}$$

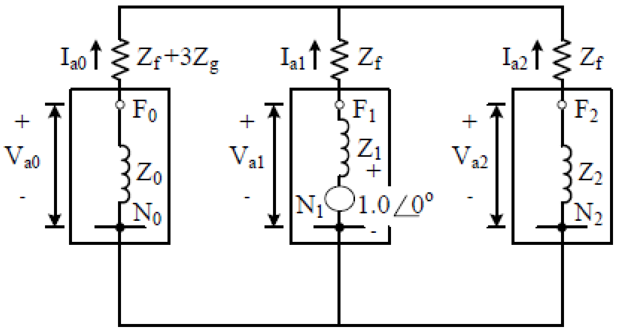


Figure 3-8: Sequence network diagram of a LLG fault.[4]

Using equation 2.16 phase currents V_{bf} and V_{cf} can be written as:

$$\begin{aligned} V_{bf} &= V_{a0} + a^2V_{a1} + aV_{a2} \\ V_{cf} &= V_{a0} + aV_{a1} + a^2V_{a2} \end{aligned} \tag{3.27}$$

Since $V_{bf} = V_{cf}$ from equation 3.25, following equation can be obtained,

$$V_{a1} = V_{a2} \tag{3.28}$$

And I_{bf} and I_{cf} in there sequence components form and simplifying the equation, the equation for V_{bf} can be written as:

$$V_{bf} = Z_f(2I_{a0} - (I_{a1} + I_{a2})) \tag{3.29}$$

and applying equation 3.26 to equation 3.29

$$V_{bf} = 3Z_fI_{a0} \tag{3.30}$$

Substituting V_{bf} from equation 3.30 in equation 3.27 and further simplifying

$$\begin{aligned} 3Z_fI_{a0} &= V_{a0} + (a^2 + a)V_{a1} \\ 3Z_fI_{a0} &= V_{a0} - V_{a1} \end{aligned} \tag{3.31}$$

The zero sequence component of the phase current can be obtained by substituting the values of V_{a0} and V_{a1} from equations 2.26-2.28

$$I_{a0} = -\frac{(1.0\underline{/0^\circ} - Z_1 I_{a1})}{(Z_0 + 3Z_f)} \tag{3.32}$$

Value of I_{a2} can be obtained in the same way as:

$$I_{a2} = -\frac{(1.0\underline{/0^\circ} - Z_1 I_{a1})}{Z_2} \tag{3.33}$$

Substituting these values in the equation 3.26, the value of the I_{a1} is found as

$$\boxed{I_{a1} = \frac{1.0\underline{/0^\circ}}{Z_1 + \dfrac{Z_2(Z_0 + 3Z_f)}{(Z_0 + Z_2 + 3Z_f)}}} \tag{3.34}$$

Furthermore, The negative and zero sequence currents can be calculated easily using current division

$$\boxed{\begin{aligned} I_{a2} &= -\frac{Z_0}{(Z_0 + Z_2)} * I_{a1} \\ I_{a0} &= -\frac{Z_2}{(Z_0 + Z_2)} * I_{a1} \end{aligned}} \tag{3.35}$$

And since V_{bf}=$Z_f I_f$ fault current I_f, conclusion can be made that

$$\boxed{I_f = 3I_{a0}} \tag{3.36}$$

Finally, these symmetrical components of currents can be transform in to phase currents using the transformation matrix in equation 2.16

3.6 Time variation of signal for transient fault currents

The transient fault currents for a power system can be analyzed by using many methods. Some of these traditional methods includes, two reaction theory, Clarke's components, etc[1].

However, using these traditional methods are extremely difficult and also these methods were not designed to analyze the power systems with a load. In this thesis, transient fault currents were analyzed using a simplified method based on the symmetrical components of synchronous machine in cases of three phase fault, single line to ground fault, line to line fault and double line to ground fault. This method has the ability to calculate the fault conditions even when there is a load present in the power system. However, in this thesis only the non loaded power systems are considered and load current was taken as a zero.
There are few main features in this method, one is that the positive phase sequence reactance is expressed in time varying numerical value. The other one is to express the negative sequence value in direct axis sub-transient value also to express the source voltage of positive sequence component as a complex value[1]. Due to the extensive information about the derivation of the method, and since it is not the main objective of this thesis, only the general introduction and main equations are discussed here. The fault current has three components. AC component, DC component and the load current component. The three phase fault current of phase a is derived from the two reaction theory and given by the following equation[1].

$$\boxed{\begin{aligned} I_a = &\left\{\left(\frac{1}{X''_d} - \frac{1}{X'_d}\right)e^{-\dfrac{t}{T''_d}} + \left(\frac{1}{X'_d} - \frac{1}{X_d}\right)e^{-\dfrac{t}{T'_d}} + \frac{1}{X_d}\right\}\sqrt{2}|E|cos(\omega t + \theta - \frac{\pi}{2}) \\ &- \frac{1}{X''_d}\sqrt{2}|E|cos(\theta - \frac{\pi}{2})e^{-\dfrac{t}{T_a}} + i_{a(l)} \end{aligned}} \tag{3.37}$$

Where;

$|E|$: Pre-fault RMS value of phase a voltage

X''_d,X'_d,X_d : Direct-axis subtransient, transient and synchronous reactances

T''_d, T'_d : Direct-axis fault subtransient and transient time constants

T_a : Armature time constant

ω: Rated angular velocity

θ: Phase angle (t=0)

$i_{a(l)}$: Load current before the fault occurrence

t: Time after the fault

The parenthesized value of this equation 3.37 is equal to the positive sequence reactance and it is denoted by X_1[1]:

$$X_{1(t)} = \left(\frac{1}{X_d''} - \frac{1}{X_d'}\right)e^{-\frac{t}{T_d''}} + \left(\frac{1}{X_d'} - \frac{1}{X_d}\right)e^{-\frac{t}{T_d'}} + \frac{1}{X_d} \tag{3.38}$$

And the complex voltage phase a voltage is given by,

$$E_a = \sqrt{2}|E|cos(\omega t + \theta - \frac{\pi}{2}) \tag{3.39}$$

Using the above mentioned equations,symmetrical component theories and equivalent circuits, The transient fault currents for each phase can be calculated using the equations shown below[1].

Three phase fault

$$\boxed{\begin{aligned} I_a &= \frac{\sqrt{2}|E|}{X_{1(t)}}sin(\omega t + \theta) - \frac{\sqrt{2}|E|}{X_d''}sin\theta e^{-\frac{t}{T_a}} + i_{a(l)} \\ I_b &= \frac{\sqrt{2}|E|}{X_{1(t)}}sin(\omega t + \theta - \frac{2\pi}{3}) - \frac{\sqrt{2}|E|}{X_d''}sin(\theta - \frac{2\pi}{3})e^{-\frac{t}{T_a}} + i_{b(l)} \\ I_c &= \frac{\sqrt{2}|E|}{X_{1(t)}}sin(\omega t + \theta + \frac{2\pi}{3}) - \frac{\sqrt{2}|E|}{X_d''}sin(\theta + \frac{2\pi}{3})e^{-\frac{t}{T_a}} + i_{c(l)} \end{aligned}} \tag{3.40}$$

Single line to ground fault

$$\boxed{\begin{aligned}
&I_a = \frac{3\sqrt{2}|E|}{X_{1(t)} + Z_2 + Z_0} sin(\omega t + \theta) - \frac{3\sqrt{2}|E|}{X_d'' + Z_2 + Z_0} sin\theta e^{-\dfrac{t}{T_a}} + i_{a(l)} \\
&I_b = i_{b(l)} \\
&I_c = i_{c(l)}
\end{aligned}} \tag{3.41}$$

Line to line fault

$$\boxed{\begin{aligned}
&I_a = i_{a(l)} \\
&I_b = -\frac{\sqrt{6}|E|}{X_{1(t)} + Z_2} cos(\omega t + \theta) + \frac{\sqrt{6}|E|}{2X_d''} cos(\theta) e^{-\dfrac{t}{T_a}} + i_{b(l)} \\
&I_c = \frac{\sqrt{6}|E|}{X_{1(t)} + Z_2} cos(\omega t + \theta) - \frac{\sqrt{6}|E|}{2X_d''} cos(\theta) e^{-\dfrac{t}{T_a}} + i_{b(l)}
\end{aligned}} \tag{3.42}$$

Double line to ground fault

$$\boxed{\begin{aligned}
&I_a = i_{a(l)} \\
&I_b = \frac{\sqrt{6}|E|}{Z_2 + Z_0} \left\{ \frac{Z_2}{X_{1(t)} + Z_{20}} sin(\omega t + \theta - \frac{5\pi}{6}) - \frac{Z_2}{X_d'' + Z_{20}} sin(\theta - \frac{5\pi}{6}) e^{-\dfrac{t}{T_a}} \right. \\
&\quad \left. + \frac{Z_0}{X_{1(t)} + Z_{20}} sin(\omega t + \theta - \frac{\pi}{2}) - \frac{Z_0}{X_d'' + Z_{20}} sin(\theta - \frac{\pi}{2}) e^{-\dfrac{t}{T_a}} \right\} + i_{b(l)} \\
&I_c = -\frac{\sqrt{6}|E|}{Z_2 + Z_0} \left\{ \frac{Z_2}{X_{1(t)} + Z_{20}} sin(\omega t + \theta + \frac{5\pi}{6}) - \frac{Z_2}{X_d'' + Z_{20}} sin(\theta + \frac{5\pi}{6}) e^{-\dfrac{t}{T_a}} \right. \\
&\quad \left. + \frac{Z_0}{X_{1(t)} + Z_{20}} sin(\omega t + \theta + \frac{\pi}{2}) - \frac{Z_0}{X_d'' + Z_{20}} sin(\theta + \frac{\pi}{2}) e^{-\dfrac{t}{T_a}} \right\} + i_{c(l)}
\end{aligned}} \tag{3.43}$$

Where, $Z_{20} = \dfrac{Z_2 Z_0}{Z_2 + Z_0}$

3.7 Calculation of bus voltages during unbalanced faults

The voltage of any bus during the unbalanced faults can be calculated using the following set of equations. The pre-fault voltage is usually taken as 1.0 $\underline{/0^\circ}$.

$$\boxed{\begin{aligned} &V_{a0} = -Z_{ik}^{(0)} I_k^{(0)} \\ &V_{a1} = 1.0\underline{/0^\circ} - Z_{ik}^{(1)} I_k^{(1)} \\ &V_{a2} = -Z_{ik}^{(2)} I_k^{(2)} \end{aligned}} \tag{3.44}$$

The phase voltages can be calculated using equation 2.16

$$\boxed{\begin{bmatrix} V_{af} \\ V_{bf} \\ V_{cf} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix}} \tag{3.45}$$

Finally, the time variation signal of the fault voltage for each phase can be obtained from

$$\boxed{\begin{aligned} &V_{af} = |V_{af}| cos(\omega t + V_{af}^{\theta}) \\ &V_{bf} = |V_{bf}| cos(\omega t + V_{bf}^{\theta}) \\ &V_{cf} = |V_{cf}| cos(\omega t + V_{cf}^{\theta}) \end{aligned}} \tag{3.46}$$

Where phase angle of phase a, phase b and phase c is denoted by V_{af}^{θ}, V_{bf}^{θ} and V_{cf}^{θ} respectively.

CHAPTER 4

MATLAB GUI and Simulation

The MATLAB GUI also known as the graphical user interface is a point-and-click software application. This GUI application makes the things easier for end users because it eradicates the need to learn a language or type commands which can be very frustrating. In this thesis, MATLAB GUI based simulation tool has been developed for students to calculate the fault currents and voltages of a power system. This tool provides the calculation of three phase fault, Single line to ground, line to line and double line to ground faults. Also, it has the ability to present the graphical transient variations of fault current and fault voltage signals during the fault on the GUI screen. A so called "Educational resources zone" was also added to this GUI, so the students can get a better understanding of the theories and calculations of power system analysis.

4.1 Structure of Developed Interface

The step by step chart of the developed MATLAB GUI tool is shown in the figure below.

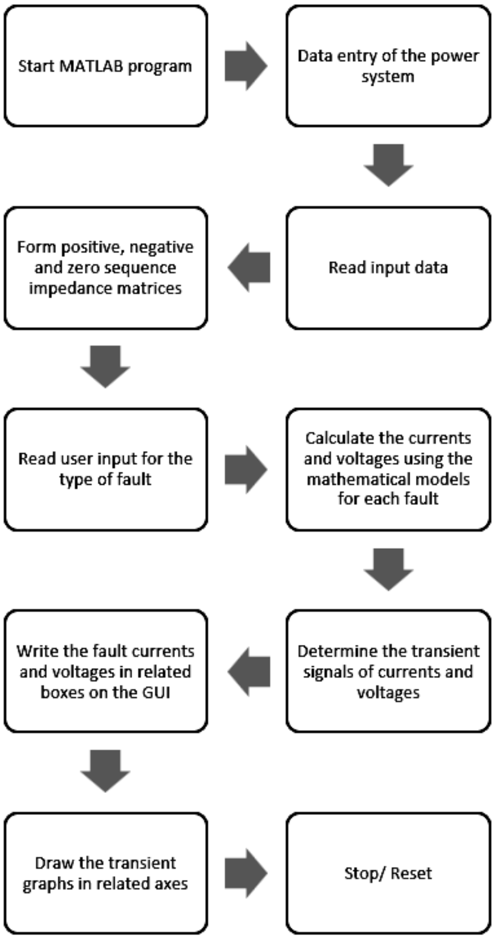

Start MATLAB program
Data entry of the power system
Read input data
Form positive, negative and zero sequence impedance matrices
Read user input for the type of fault
Calculate the currents and voltages using the mathematical models for each fault
Determine the transient signals of currents and voltages
Write the fault currents and voltages in related boxes on the GUI
Draw the transient graphs in related axes
Stop/ Reset

Figure 4-1: step-by-step chart of the developed MATLAB GUI interface

In the start of the MATLAB GUI program, the user will have a chance to enter data of the power system that needs to be analyzed, The program is designed to take user's input and adjust the questions that are provided to the user. The user will have to describe the power system in terms of number of buses, number of generators, number of lines and number of transformers in the system. Once the initial data has been obtained, the program will ask for more specific questions about the each generator, line and the transformer to use for the calculations. Data for sample 5-bus power system and sample 6-bus power system are already integrated in to the software by default. Once all the data has been entered, the program will read the input data and form the positive, negative and zero sequence impedance matrices to use in the calculations. According to the user defined fault condition, the calculations for the fault voltages and fault current will start using the already integrated mathematical models discussed in chapter 3. Also, the program will determine the transient signals of currents and voltages too. Finally, the program will write the calculated fault currents and voltages in the related boxes on the GUI and also it will draw the transient graphs on the related axes. Figure 4-2 shows the final appearance of the developed GUI interface.

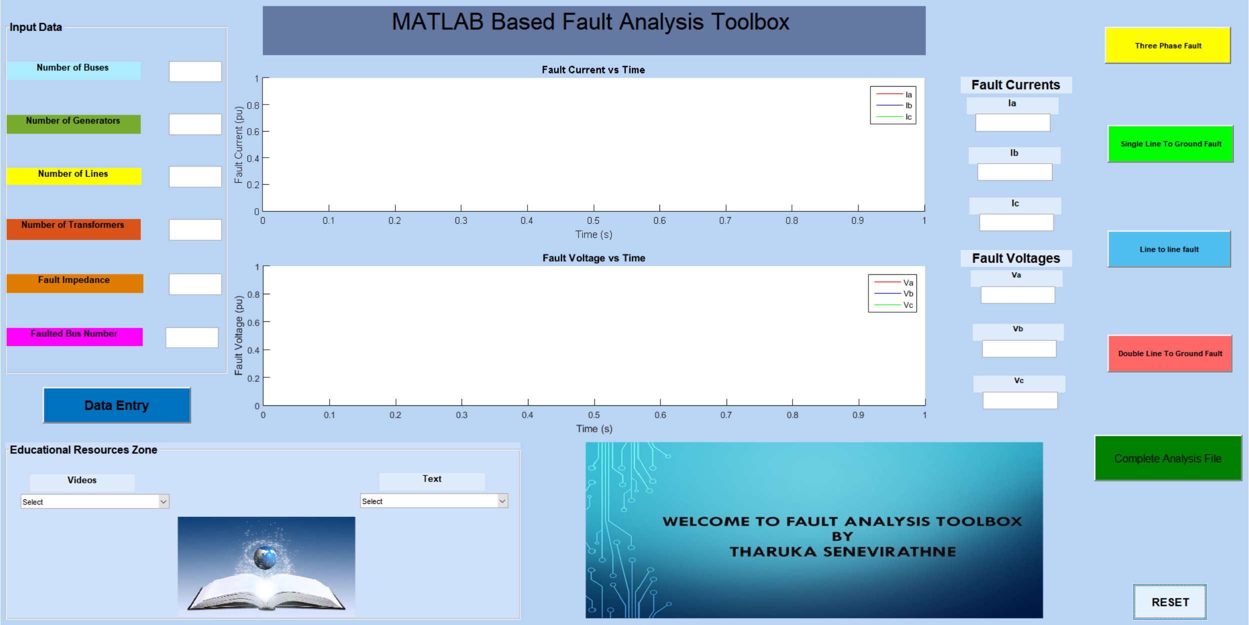


Figure 4-2: Final appearance of developed MATLAB GUI interface

4.1.1 Input Data

As it can be seen in the figure 4-2, the input data section has been created for entering the number of buses, number of generators, number of lines and the number of transformers of the power system. The user can also enter the information about fault in the same ”Input Data” section such as, fault impedance (Z_f) and the faulted bus number. Using the data entry button, the user can describe the power system in more details by answering specific question about generators, lines and transformers. These specific questions include the following,

for each generator,

1. What is the bus number connected to the generator?

2. What is the positive/negative sequence impedance of the generator?

3. What is the zero sequence impedance of the generator?

4. Grounding impedance of the generator and if ungrounded enter 1000

And for each line,

1. What is the starting bus number?

2. What is the ending bus number?

3. What is the positive/negative sequence impedance of the line?

4. What is the zero sequence impedance of the line?

Finally, for each transformer,

1. What is the bus number of high voltage side bus number of the transformer?

2. Please enter the connection code → 0-star grounded 1-star ungrounded 2-delta

3. What is the bus number of low voltage side bus of the transformer?

4. Please enter the connection code → 0-star grounded 1-star ungrounded 2-delta

5. What is the positive/negative sequence impedance of the transformer?

6. What is the zero sequence impedance of the transformer?

Using the answers to these specific questions, any power system can be represented. Figure 4-3 shows the Input data section with the data entry button. This program takes all the input values as string values. But input values have to be in numeric form in order to do the numeric calculations. Therefore after entering, all the values are converted to numeric values using 'str2double' command in MATLAB. The pushbutton option is used as a data entry button and all the entered data is saved in different .txt documents separately, to access during the calculations. For example, generator data is saved in 'Generator_Data.txt' and transformer data is saved in 'Transformer_Data.txt'

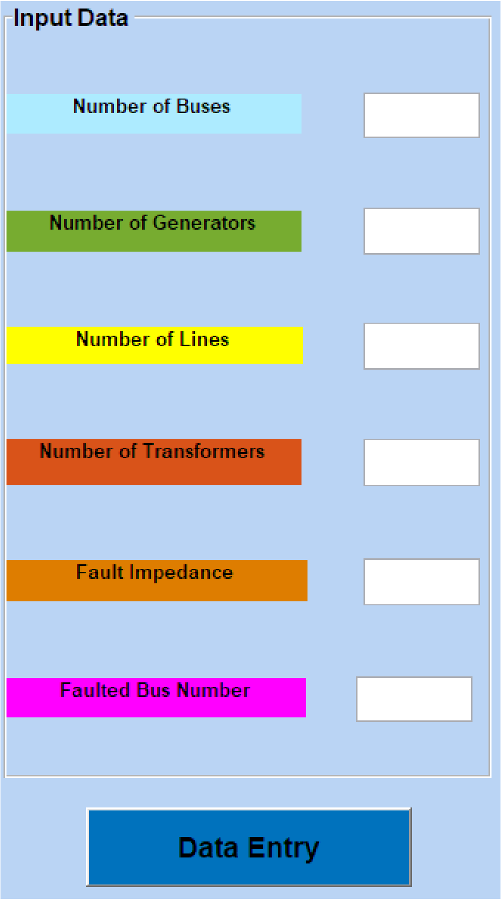


Figure 4-3: Input data section of developed MATLAB GUI interface

4.1.2 Fault conditions

Four pushbuttons have been created for the four types of fault conditions. Once the user has all the input data entered to the system these push buttons can be used to run the desired calculations for the three phase fault, single line to ground fault, line to line fault and the double line to ground fault. These pushbuttons are programmed to execute the integrated mathematical models on button press. Figure 4-4 shows the pushbuttons for the different fault conditions.

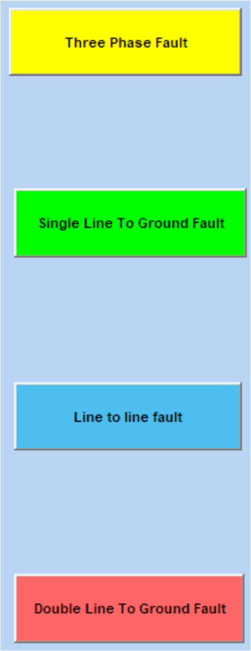


Figure 4-4: Fault conditions section of developed MATLAB GUI interface

4.1.3 Axes and dialog boxes

Axes and dialog boxes are used to draw the resulting transient graphs and to write the obtained values for fault currents and fault voltages. These achieved values have been converted back to character strings to display in the dialog boxes. All the calculations are done in per unit system and results are also shown in per-unit. Axes and dialog boxes for displaying results are shown in figure 4-5.

4.2 Educational Resources Zone

'Educational Resources Zone' is a area in the developed GUI, that has a series of videos and text documents to use as learning materials. These videos and text materials have been selected from the internet to give the students opportunity to learn about power system analysis as they do the calculations in the same platform. All the documents are added using the drop down menu option in GUI. Some of the topics in the Educational resources zone include are:

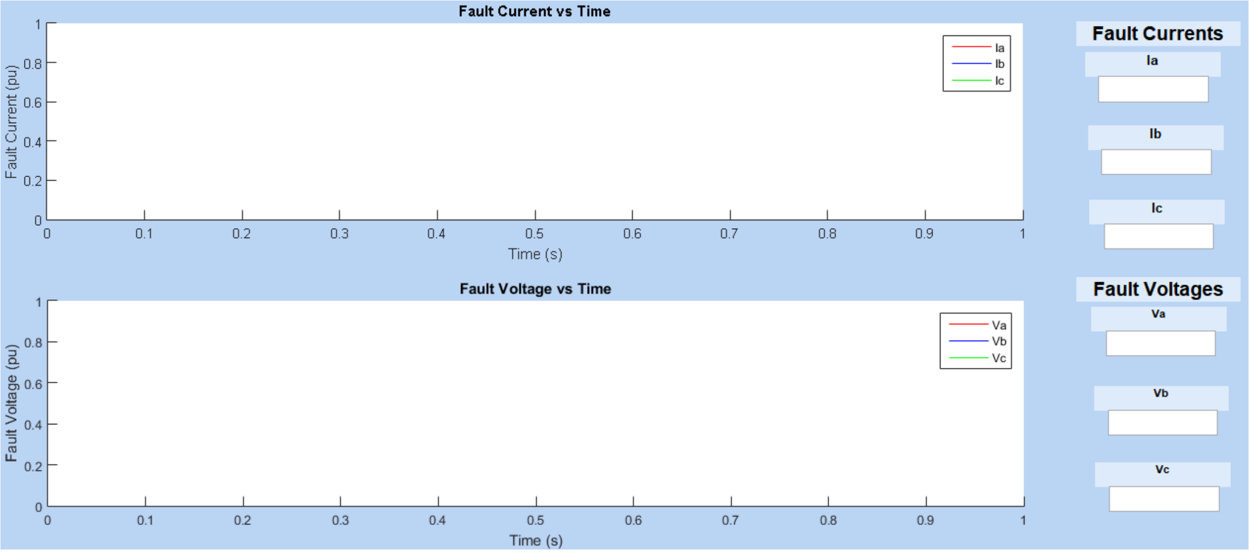


Figure 4-5: Axes and dialog boxes of developed MATLAB GUI interface

1. Z bus matrix formation

2. Symmetrical fault analysis

3. Sequence networks

4. Fault analysis using sequence network

It has the capability to add as many as materiel to the program. Figure 4-6 shows the Educational resources zone in the GUI application.

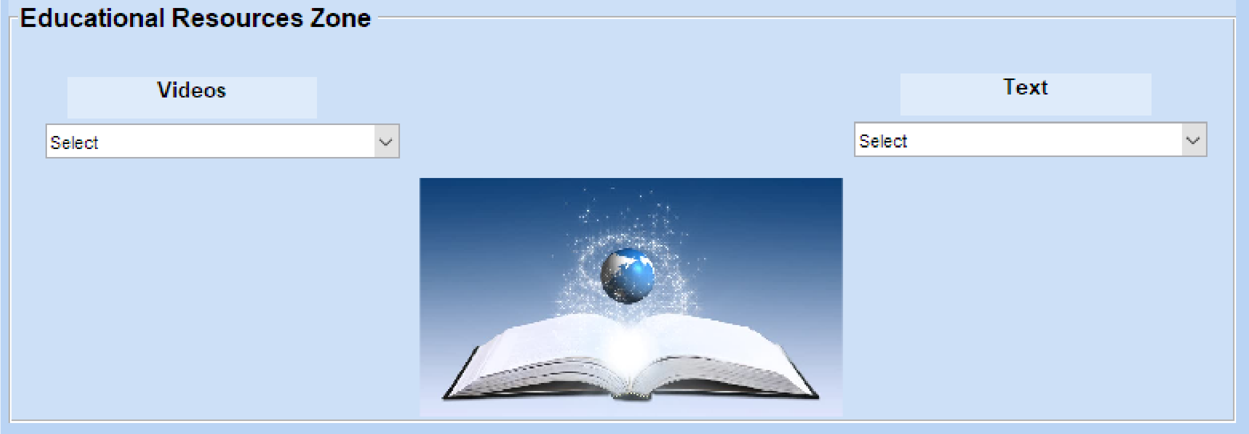


Figure 4-6: "Educational Resources Area" of developed MATLAB GUI interface

CHAPTER 5

Results and Analysis

In this section, a sample 5-bus power system is used to show the validity of the developed MATLAB program and the accuracy of the hand calculations. Positive sequence, negative sequence and zero sequence bus impedance matrices will be formed using the technique discussed in section 3.1.1 and it will be used to find the Thevenin's impedance of the network seen from different buses. The accuracy of the hand calculations will be determined by comparing the results with the results of developed MATLAB program. The single line diagram of the example 5-bus system with two generators , three transmission lines and two transformers is shown in the figure 5-1 below[5]. And values of the example 5 bus power system parameters are listed in table 5-1.

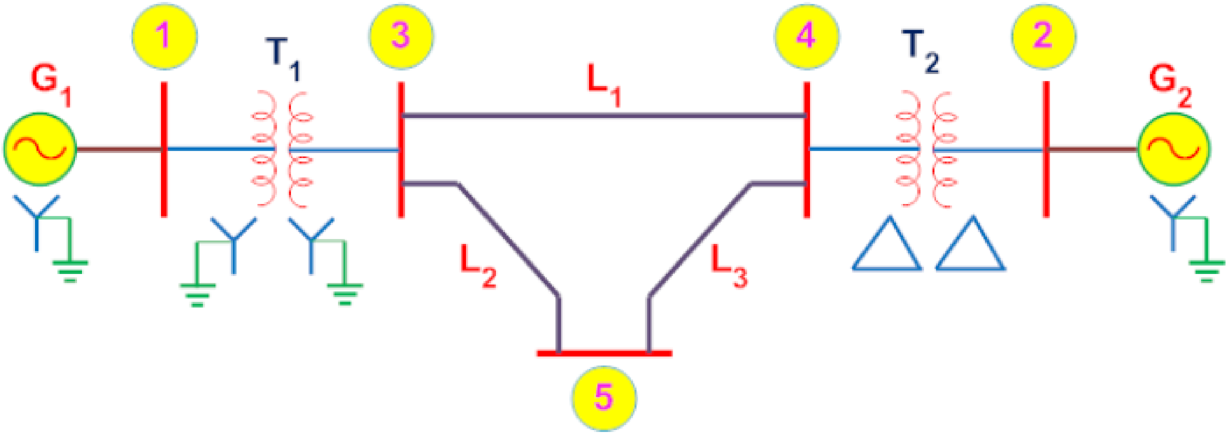


Figure 5-1: Single line diagram of example 5 bus power system.[5]

Table 5.1: System parameters of 5 bus power system.[5]

Component	Positive sequence impedance (pu)	Negative sequence impedance(pu)	Zero Sequence impedance(pu)
Generator 1 (G1)	0.2	0.2	0.05
Generator 2 (G2)	0.2	0.2	0.05
Transformer 1 (T1)	0.05	0.05	0.05
Transformer 2 (T2)	0.05	0.05	0.05
Line 1 (L1)	0.1	0.1	0.3
Line 2 (L2)	0.1	0.1	0.3
Line 3 (L3)	0.1	0.1	0.3

Pre-fault value for all the buses in the example 5 bus syetem is taken as $E_i(0)$= 1.0 $\underline{/0°}$ where i=1,2,3

5.1 Three-phase fault

5.1.1 Hand calculation results

Three phase fault is tested by creating a bolted fault at bus 5. Therefore, Z_f=0. For the three phase bolted fault only the positive impedance sequence network is required. Positive sequence bus impedance matrix was formed using the techniques discussed in the section

3.1.1. Only the positive sequence impedances are used here.

$$\left[Z_{bus}^{(1)}\right] = \begin{bmatrix} j0.1294 & j0.0706 & j0.1118 & j0.0882 & j0.10 \\ j0.0706 & j0.1294 & j0.0882 & j0.1118 & j0.10 \\ j0.1118 & j0.0882 & j0.1397 & j0.1103 & j0.1250 \\ j0.0882 & j0.1118 & j0.1103 & j0.1397 & j0.10 \\ j0.10 & j0.10 & j0.1250 & j0.1250 & j0.1750 \end{bmatrix} pu \tag{5.1}$$

Since this is a bolted fault, equations 3.5 and 3.6 can be use to calculate the symmetrical components of the phase currents.

$$I_5^{(1)} = \frac{1.0\underline{/0^\circ}}{Z_{55}^{(1)}} \tag{5.2}$$

where $Z_{55}^{(1)}$ is the Thevenin's equivalent for positive sequence impedance at faulted bus 5 and can be obtained from equation 5.1. Therefore,

$$\boxed{\begin{aligned} I_5^{(1)} &= \frac{1.0\underline{/0^\circ}}{j0.1750} \\ I_5^{(1)} &= -j5.7143pu \end{aligned}} \tag{5.3}$$

and from equation 3.5

$$\boxed{\begin{aligned} I_5^{(0)} &= 0 \\ I_5^{(2)} &= 0 \end{aligned}} \tag{5.4}$$

The phase of the fault currents can be calculated using the transformation matrix in equation 2.17.

$$\boxed{\left[I_5^{(abc)}\right] = \begin{bmatrix} I_{af} \\ I_{bf} \\ I_{cf} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ j5.7143 \\ 0 \end{bmatrix} = \begin{bmatrix} 5.7143\underline{/-90^\circ} \\ 5.7143\underline{/150^\circ} \\ 5.7143\underline{/30^\circ} \end{bmatrix}} \tag{5.5}$$

The fault voltages at each bus can be calculated as shown below.

Bus 1 voltage during the fault :

$$\begin{aligned} V_1^{(1)} &= E_1(0) - Z_{15}^{(1)} I_5^{(1)} \\ &= 1.0\underline{/0^\circ} - (j0.10) * (-j5.7143) \\ &= 0.42857pu \end{aligned} \tag{5.6}$$

and since this a balanced fault, phase voltages can be obtained as:

$$\boxed{\left[V_1^{(abc)}\right] = \begin{bmatrix} 0.42857\underline{/0^\circ} \\ 0.42857\underline{/-120^\circ} \\ 0.42857\underline{/120^\circ} \end{bmatrix}} \tag{5.7}$$

Bus 2 voltage during the fault :

$$\begin{aligned} V_2^{(1)} &= E_2(0) - Z_{25}^{(1)} I_5^{(1)} \\ &= 1.0\underline{/0^\circ} - (j0.10) * (-j5.7143) \\ &= 0.42857pu \end{aligned} \tag{5.8}$$

and since this a balanced fault, phase voltages can be obtained as:

$$\boxed{\left[V_2^{(abc)}\right] = \begin{bmatrix} 0.42857\underline{/0^\circ} \\ 0.42857\underline{/-120^\circ} \\ 0.42857\underline{/120^\circ} \end{bmatrix}} \tag{5.9}$$

Bus 3 voltage during the fault :

$$\begin{aligned} V_3^{(1)} &= E_3(0) - Z_{35}^{(1)} I_5^{(1)} \\ &= 1.0\underline{/0^\circ} - (j0.125) * (-j5.7143) \\ &= 0.28571pu \end{aligned} \tag{5.10}$$

and since this a balanced fault, phase voltages can be obtained as:

$$\boxed{\left[V_3^{(abc)}\right] = \begin{bmatrix} 0.28571\underline{/0^\circ} \\ 0.28571\underline{/-120^\circ} \\ 0.28571\underline{/120^\circ} \end{bmatrix}} \tag{5.11}$$

Bus 4 voltage during the fault :

$$\begin{aligned} V_4^{(1)} &= E_4(0) - Z_{45}^{(1)} I_5^{(1)} \\ &= 1.0\underline{/0^\circ} - (j0.125) * (-j5.7143) \\ &= 0.28571pu \end{aligned} \tag{5.12}$$

and since this a balanced fault, phase voltages can be obtained as:

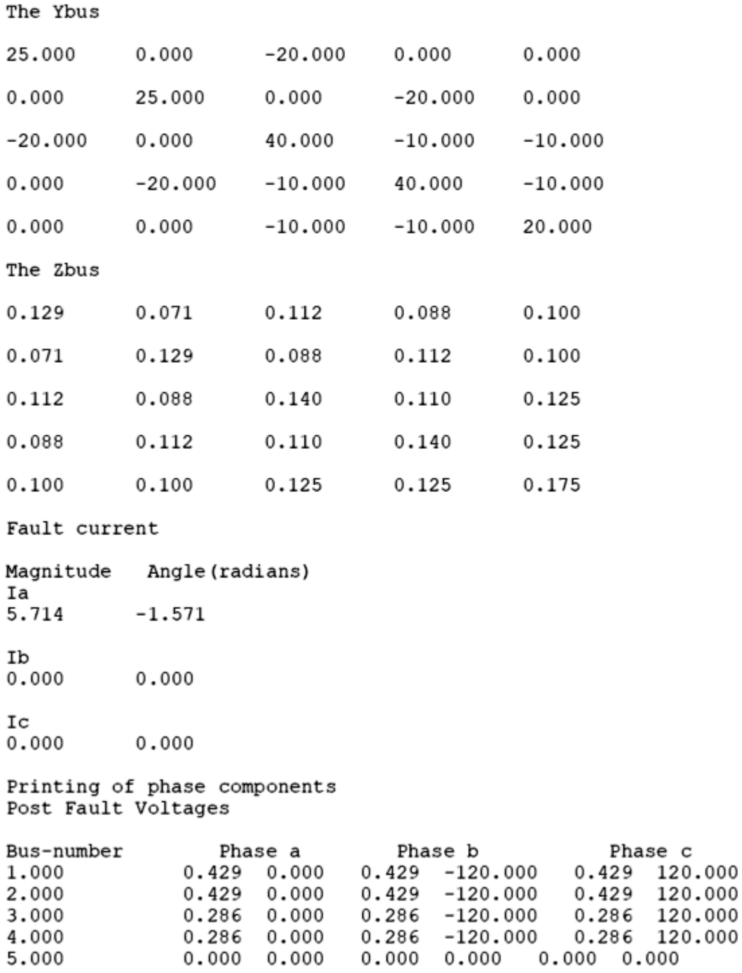
$$\boxed{\left[V_4^{(abc)}\right] = \begin{bmatrix} 0.28571\underline{/0^\circ} \\ 0.28571\underline{/-120^\circ} \\ 0.28571\underline{/120^\circ} \end{bmatrix}} \tag{5.13}$$

Since this is a bolted fault and the fault impedance Z_f=0, the bus voltages under faulted conditions at bus 5 are

$$\boxed{\left[V_5^{(abc)}\right] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}} \tag{5.14}$$

5.1.2 MATLAB toolbox results

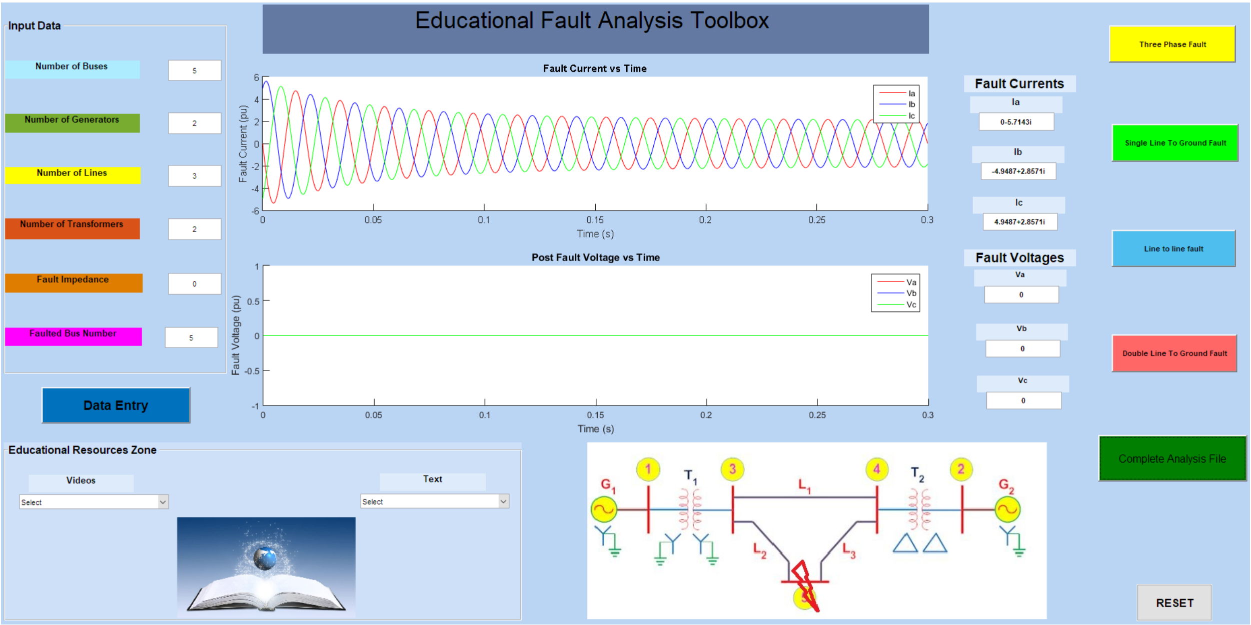
Fault analysis for the sample 5-bus system shown in the figure 5-1 was achieved using the developed MATLAB tool box. Same system parameters entered in Table 5.1 were used to set up the problem. Results were printed in to a .txt file and those results are shown in figure 5-2. A screen-shot of GUI interface with fault analysis results and transient graphs of



```
The Ybus

25.000     0.000      -20.000    0.000      0.000

0.000      25.000     0.000      -20.000    0.000

-20.000    0.000      40.000     -10.000    -10.000

0.000      -20.000    -10.000    40.000     -10.000

0.000      0.000      -10.000    -10.000    20.000

The Zbus

0.129      0.071      0.112      0.088      0.100

0.071      0.129      0.088      0.112      0.100

0.112      0.088      0.140      0.110      0.125

0.088      0.112      0.110      0.140      0.125

0.100      0.100      0.125      0.125      0.175

Fault current

Magnitude   Angle(radians)
Ia
5.714      -1.571

Ib
0.000      0.000

Ic
0.000      0.000

Printing of phase components
Post Fault Voltages

Bus-number        Phase a        Phase b          Phase c
1.000          0.429  0.000   0.429  -120.000   0.429  120.000
2.000          0.429  0.000   0.429  -120.000   0.429  120.000
3.000          0.286  0.000   0.286  -120.000   0.286  120.000
4.000          0.286  0.000   0.286  -120.000   0.286  120.000
5.000          0.000  0.000   0.000  0.000    0.000  0.000
```

Figure 5-2: Three Phase fault analysis from developed MATLAB tool box

fault current and fault voltage signals are shown in figure 5-3.



Educational Fault Analysis Toolbox

Input Data

Number of Buses: 5
Number of Generators: 2
Number of Lines: 3
Number of Transformers: 2
Fault Impedance: 0
Faulted Bus Number: 5

Fault Currents
Ia: 0-5.7143i
Ib: -4.9487+2.8571i
Ic: 4.9487+2.8571i

Fault Voltages
Va: 0
Vb: 0
Vc: 0

Educational Resources Zone
Videos: Select
Text: Select

Figure 5-3: Three Phase fault analysis - MATLAB tool box

In this figure, we can see that voltages dropped to zero in all three phases during the fault at the faulted bus 5. Therefore, the fault currents are at the maximum value.

5.2 Single Line-to-ground fault

In this section, Analysis of the single line to ground fault is done using the equations discussed in chapter 3 and also using the developed MATLAB tool box. The single line to ground fault is tested by creating a fault at bus 5 with fault impedance Z_f=0.1pu.

5.2.1 Hand calculation results

All three positive, negative and zero sequence networks are required for the analysis of the single line to ground fault since this is a unbalanced fault. The positive sequence network and positive impedance matrix are the same as used in three phase fault and negative impedance is also identical to the positive sequence network[5]. Therefore, $Z_{bus}^{(2)} = Z_{bus}^{(1)}$ and shown below

$$\left[Z_{bus}^{(1)}\right] = \left[Z_{bus}^{(2)}\right] = \begin{bmatrix} j0.1294 & j0.0706 & j0.1118 & j0.0882 & j0.10 \\ j0.0706 & j0.1294 & j0.0882 & j0.1118 & j0.10 \\ j0.1118 & j0.0882 & j0.1397 & j0.1103 & j0.1250 \\ j0.0882 & j0.1118 & j0.1103 & j0.1397 & j0.10 \\ j0.10 & j0.10 & j0.1250 & j0.1250 & j0.1750 \end{bmatrix} pu \tag{5.15}$$

The zero sequence equivalent network is different from the positive and negative sequence network since transformer connections and grounding and also the generator grounding comes in to consideration. Sequence networks of transformers with different connections were discussed in the section 2.2.3. Transformer T_1 has star-star connection with both neutrals grounded and because of that the zero sequence impedance of T_1 is connected between bus 1 and bus 2 as usual. But transformer T_2 has two delta connections and as discussed in section 2.2.3.6 , it is connected as a short circuited winding and zero sequence currents circulate in the delta connected winding. Using these techniques, the zero sequence bus impedance matrix $Z_{bus}^{(0)}$ is obtained and it is given as :

$$\left[Z_{bus}^{(0)}\right] = \begin{bmatrix} j0.05 & 0.0 & j0.05 & j0.05 & j0.05 \\ 0.0 & j0.05 & 0.0 & 0.0 & 0.0 \\ j0.05 & 0.0 & j0.10 & j0.10 & j0.10 \\ j0.05 & 0.0 & j0.10 & j0.30 & j0.20 \\ j0.05 & 0.0 & j0.10 & j0.20 & j0.30 \end{bmatrix} pu \tag{5.16}$$

Now since all the impedance matrices are obtained, the symmetrical components of fault current can be calculated using equation 3.17 derived in chapter 3.

$$\begin{aligned} I_5^{(1)} &= \frac{1.0\underline{/0^\circ}}{Z_{55}^{(0)} + Z_{55}^{(1)} + Z_{55}^{(2)} + 3Z_f} \\ &= \frac{1.0\underline{/0^\circ}}{j0.30 + j0.175 + j0.175 + (3 * j0.1)} \\ &= -j1.05263pu \end{aligned} \tag{5.17}$$

and from the equation 3.13

$$\boxed{I_5^{(0)} = I_5^{(1)} = I_5^{(2)} = -j1.05263pu} \tag{5.18}$$

The phase components of the fault current at bus 5 can be obtained using the transformation matrix A in equation 2.17

$$\boxed{\left[I_5^{(abc)}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} -j1.05263 \\ -j1.05263 \\ -j1.05263 \end{bmatrix} = \begin{bmatrix} 3.1579\underline{/-90^\circ} \\ 0 \\ 0 \end{bmatrix} pu} \tag{5.19}$$

Bus voltage are calculated using equations 3.44-3.45 in chapter 3

$$\begin{aligned} V_i^{(0)} &= -Z_{ik}^{(0)} I_k^{(0)} \\ V_i^{(1)} &= E_i^{(1)}(0) - Z_{ik}^{(1)} I_k^{(1)} \\ V_i^{(2)} &= -Z_{ik}^{(2)} I_k^{(2)} \end{aligned} \tag{5.20}$$

The above equations can be written in compact form as

$$\begin{bmatrix} V_i^{(0)} \\ V_i^{(1)} \\ V_i^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ V_i \\ 0 \end{bmatrix} - \begin{bmatrix} Z_{ik}^{(0)} & 0 & 0 \\ 0 & Z_{ik}^{(1)} & 0 \\ 0 & 0 & Z_{ik}^{(2)} \end{bmatrix} \begin{bmatrix} I_k^{(0)} \\ I_k^{(1)} \\ I_k^{(2)} \end{bmatrix} \tag{5.21}$$

Fault voltages at bus 1 during the fault are:

$$\begin{bmatrix} V_1^{(0)} \\ V_1^{(1)} \\ V_1^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.05 & 0 & 0 \\ 0 & j0.10 & 0 \\ 0 & 0 & j0.10 \end{bmatrix} \begin{bmatrix} -j1.05263 \\ -j1.05263 \\ -j1.05263 \end{bmatrix} \tag{5.22}$$

Therefore,

$$\begin{bmatrix} V_1^{(012)} \end{bmatrix} = \begin{bmatrix} -0.05263 \\ 0.894737 \\ -0.105263 \end{bmatrix} pu \tag{5.23}$$

Using the transformation matrix in equation 2.16 phase components of the voltage can be obtained as:

$$\boxed{\begin{bmatrix} V_1^{abc} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} -0.05263 \\ 0.894737 \\ -0.105263 \end{bmatrix} = \begin{bmatrix} 0.7368\underline{/0^\circ} \\ 0.9747\underline{/-117.3197^\circ} \\ 0.9747\underline{/117.3197^\circ} \end{bmatrix}} \tag{5.24}$$

Fault voltages at bus 2 during the fault are:

$$\begin{bmatrix} V_2^{(0)} \\ V_2^{(1)} \\ V_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.0 & 0 & 0 \\ 0 & j0.10 & 0 \\ 0 & 0 & j0.10 \end{bmatrix} \begin{bmatrix} -j1.05263 \\ -j1.05263 \\ -j1.05263 \end{bmatrix} \tag{5.25}$$

Therefore,

$$\begin{bmatrix} V_2^{(012)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.894737 \\ -0.105263 \end{bmatrix} pu \tag{5.26}$$

Using the transformation matrix in equation 2.16 phase components of the voltage can be obtained as:

$$\boxed{\left[V_2^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.894737 \\ -0.105263 \end{bmatrix} = \begin{bmatrix} 0.7895\underline{/0^\circ} \\ 0.9517\underline{/-114.5036^\circ} \\ 0.9517\underline{/114.5036^\circ} \end{bmatrix}} \tag{5.27}$$

Fault voltages at bus 3 during the fault are:

$$\begin{bmatrix} V_3^{(0)} \\ V_3^{(1)} \\ V_3^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.10 & 0 & 0 \\ 0 & j0.125 & 0 \\ 0 & 0 & j0.125 \end{bmatrix} \begin{bmatrix} -j1.05263 \\ -j1.05263 \\ -j1.05263 \end{bmatrix} \tag{5.28}$$

Therefore,

$$\left[V_3^{(012)}\right] = \begin{bmatrix} -0.105263 \\ 0.868421 \\ -0.131578 \end{bmatrix} pu \tag{5.29}$$

Using the transformation matrix in equation 2.16 phase components of the voltage can be obtained as:

$$\boxed{\left[V_3^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} -0.105263 \\ 0.868421 \\ -0.131578 \end{bmatrix} = \begin{bmatrix} 0.6316\underline{/0^\circ} \\ 0.9871\underline{/-118.677^\circ} \\ 0.9871\underline{/118.677^\circ} \end{bmatrix}} \tag{5.30}$$

Fault voltages at bus 4 during the fault are:

$$\begin{bmatrix} V_4^{(0)} \\ V_4^{(1)} \\ V_4^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.20 & 0 & 0 \\ 0 & j0.125 & 0 \\ 0 & 0 & j0.125 \end{bmatrix} \begin{bmatrix} -j1.05263 \\ -j1.05263 \\ -j1.05263 \end{bmatrix} \tag{5.31}$$

Therefore,

$$\left[V_4^{(012)}\right] = \begin{bmatrix} -0.210526 \\ 0.868421 \\ -0.131578 \end{bmatrix} pu \tag{5.32}$$

Using the transformation matrix in equation 2.16 phase components of the voltage can be obtained as:

$$\boxed{\left[V_4^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} -0.210526 \\ 0.868421 \\ -0.131578 \end{bmatrix} = \begin{bmatrix} 0.5263\underline{/0^\circ} \\ 1.0417\underline{/-123.76^\circ} \\ 1.0417\underline{/123.76^\circ} \end{bmatrix}} \tag{5.33}$$

Fault voltages at bus 5 during the fault are:

$$\begin{bmatrix} V_5^{(0)} \\ V_5^{(1)} \\ V_5^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.30 & 0 & 0 \\ 0 & j0.175 & 0 \\ 0 & 0 & j0.175 \end{bmatrix} \begin{bmatrix} -j1.05263 \\ -j1.05263 \\ -j1.05263 \end{bmatrix} \tag{5.34}$$

Therefore,

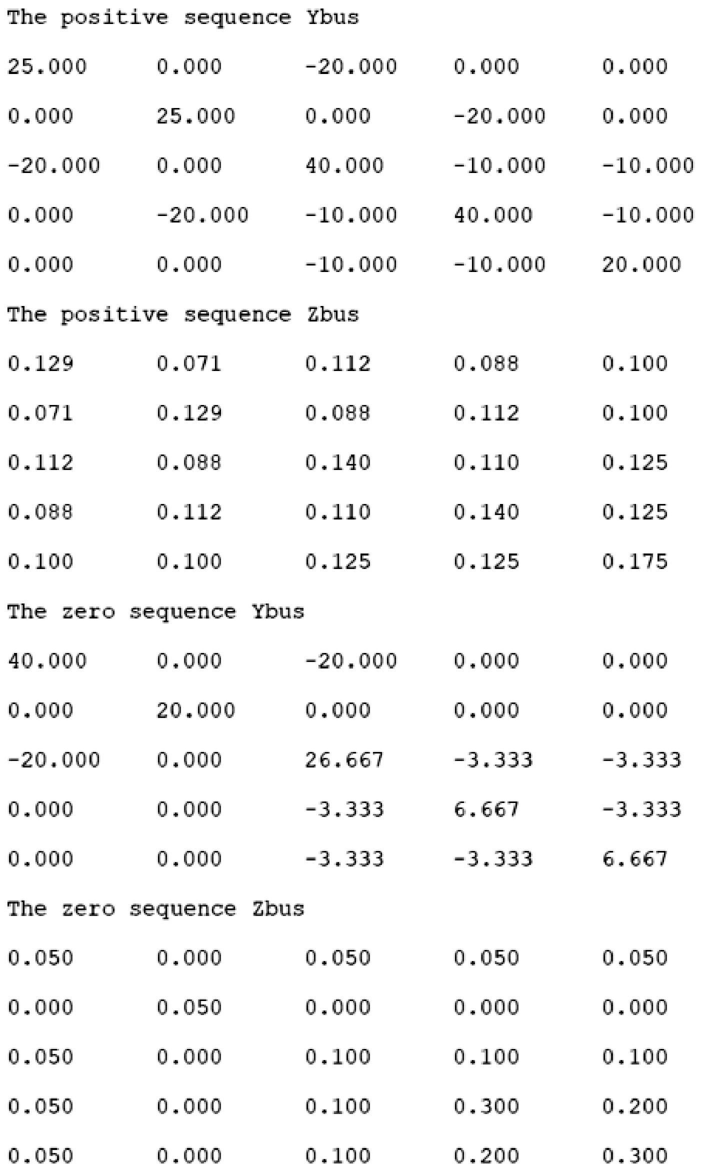
$$\left[V_5^{(012)}\right] = \begin{bmatrix} -0.315789 \\ 0.815789 \\ -0.184210 \end{bmatrix} pu \tag{5.35}$$

Using the transformation matrix in equation 2.16 phase components of the voltage can be obtained as:

$$\boxed{\left[V_5^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} -0.315789 \\ 0.815789 \\ -0.184210 \end{bmatrix} = \begin{bmatrix} 0.3158\underline{/0^\circ} \\ 1.0719\underline{/-126.10^\circ} \\ 1.0719\underline{/126.10^\circ} \end{bmatrix}} \tag{5.36}$$

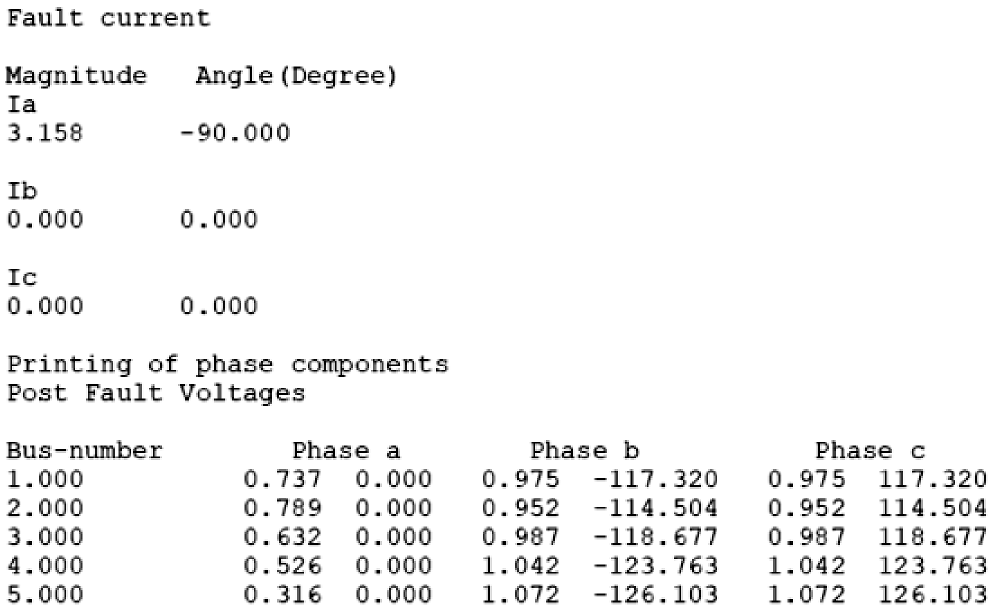
5.2.2 MATLAB toolbox results

Single line to ground fault analysis was also done using the developed MATLAB tool box and results are shown in the figures below. Figure 5-4 shows the calculated values of impedance matrices and figure 5-5 shows the fault currents and fault voltages at each bus and finally, figure 5-6 shows the GUI screen with transient graphs.



```
The positive sequence Ybus
25.000      0.000       -20.000     0.000       0.000
0.000       25.000      0.000       -20.000     0.000
-20.000     0.000       40.000      -10.000     -10.000
0.000       -20.000     -10.000     40.000      -10.000
0.000       0.000       -10.000     -10.000     20.000
The positive sequence Zbus
0.129       0.071       0.112       0.088       0.100
0.071       0.129       0.088       0.112       0.100
0.112       0.088       0.140       0.110       0.125
0.088       0.112       0.110       0.140       0.125
0.100       0.100       0.125       0.125       0.175
The zero sequence Ybus
40.000      0.000       -20.000     0.000       0.000
0.000       20.000      0.000       0.000       0.000
-20.000     0.000       26.667      -3.333      -3.333
0.000       0.000       -3.333      6.667       -3.333
0.000       0.000       -3.333      -3.333      6.667
The zero sequence Zbus
0.050       0.000       0.050       0.050       0.050
0.000       0.050       0.000       0.000       0.000
0.050       0.000       0.100       0.100       0.100
0.050       0.000       0.100       0.300       0.200
0.050       0.000       0.100       0.200       0.300
```

Figure 5-4: Calculated impedance matrices of single line to ground fault



```
Fault current

Magnitude   Angle(Degree)
Ia
3.158       -90.000

Ib
0.000       0.000

Ic
0.000       0.000

Printing of phase components
Post Fault Voltages

Bus-number        Phase a         Phase b           Phase c
1.000          0.737  0.000   0.975  -117.320   0.975  117.320
2.000          0.789  0.000   0.952  -114.504   0.952  114.504
3.000          0.632  0.000   0.987  -118.677   0.987  118.677
4.000          0.526  0.000   1.042  -123.763   1.042  123.763
5.000          0.316  0.000   1.072  -126.103   1.072  126.103
```

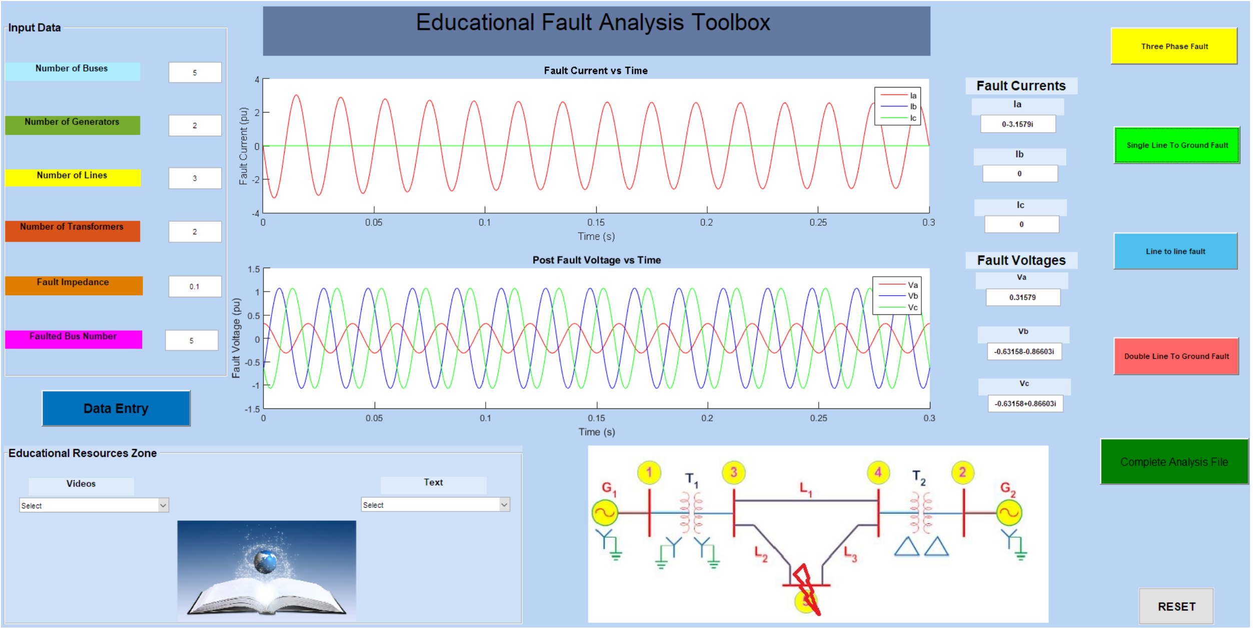
Figure 5-5: Fault currents and fault voltages of single line to ground fault

Analyzing the results, it shows that fault current of phase b and phase c are zero due to the fact that only the phase a is involved in the fault.

5.3 Line-to-line fault

Analyzing of line to line required only positive sequence network and negative sequence network. Therefore, the same $Z_{bus}^{(1)}$ and $Z_{bus}^{(2)}$ are calculated for single line to ground analysis will be used. The fault is created between phase b and phase c at bus 5 and the fault impedance is taken as $Z_f = j0.1$ pu.

$$\left[Z_{bus}^{(1)}\right] = \left[Z_{bus}^{(2)}\right] = \begin{bmatrix} j0.1294 & j0.0706 & j0.1118 & j0.0882 & j0.10 \\ j0.0706 & j0.1294 & j0.0882 & j0.1118 & j0.10 \\ j0.1118 & j0.0882 & j0.1397 & j0.1103 & j0.1250 \\ j0.0882 & j0.1118 & j0.1103 & j0.1397 & j0.10 \\ j0.10 & j0.10 & j0.1250 & j0.1250 & j0.1750 \end{bmatrix} pu \tag{5.37}$$



Input Data

Number of Buses: 5

Number of Generators: 2

Number of Lines: 3

Number of Transformers: 2

Fault Impedance: 0.1

Faulted Bus Number: 5

Fault Currents

Ia: 0-3.1579i

Ib: 0

Ic: 0

Fault Voltages

Va: 0.31579

Vb: -0.63158-0.86603i

Vc: -0.63158+0.86603i

Figure 5-6: Single line to ground fault analysis - MATLAB tool box

5.3.1 Hand calculation results

Since there is no ground path, zero sequence currents can not flow. Therefore, from equation 3.19

$$\begin{aligned} I_5^{(1)} &= \frac{1.0\underline{/0^\circ}}{Z_{55}^{(1)} + Z_{55}^{(2)} + Z_f} \\ &= \frac{1.0\underline{/0^\circ}}{j0.175 + j0.175 + j0.1} \\ &= -j2.222 \ pu \end{aligned} \tag{5.38}$$

Negative sequence current can be obtained from equation 3.21

$$\boxed{I_5^{(2)} = -I_5^{(1)} = j2.222 \ pu} \tag{5.39}$$

The phase components of the fault current at bus 5 can be obtained using the transformation matrix A in equation 2.17

$$\boxed{\left[I_5^{(abc)}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ -j2.222 \\ j2.222 \end{bmatrix} = \begin{bmatrix} 0 \\ -3.849 \\ 3.849 \end{bmatrix} pu} \tag{5.40}$$

Bus voltages during fault can be calculated using equation 5.20 as

For bus 1,

zero sequence voltage is zero

$$V_1^{(0)} = 0 \tag{5.41}$$

Positive and negative sequence voltage at bus 1 can be calculated as

$$\begin{aligned} V_1^{(1)} &= E_1^{(1)} - Z_{15}^{(1)} I_5^{(1)} \\ &= 1.0 - j0.10 * (-j2.222) \\ &= 0.7778\ pu \end{aligned} \tag{5.42}$$

$$\begin{aligned} V_1^{(2)} &= -Z_{15}^{(2)} I_5^{(2)} \\ &= -j0.10 * (j2.222) \\ &= 0.2222\ pu \end{aligned} \tag{5.43}$$

Using the transformation matrix in equation 2.16 phase components of the voltage can be obtained as:

$$\boxed{\left[V_1^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.7778 \\ 0.2222 \end{bmatrix} = \begin{bmatrix} 1\underline{/0^\circ} \\ 0.6938\underline{/-135.998^\circ} \\ 0.6938\underline{/135.998^\circ} \end{bmatrix}} \tag{5.44}$$

Given that bus 2 zero sequence voltage is zero,

$$V_2^{(0)} = 0 \tag{5.45}$$

positive and negative sequence voltages at bus 2 can be calculated as

$$\begin{aligned} V_2^{(1)} &= E_2^{(1)} - Z_{25}^{(1)} I_5^{(1)} \\ &= 1.0 - j0.10 * (-j2.222) \\ &= 0.7778\ pu \end{aligned} \tag{5.46}$$

$$\begin{aligned} V_2^{(2)} &= -Z_{25}^{(2)} I_5^{(2)} \\ &= -j0.10 * (j2.222) \\ &= 0.2222\ pu \end{aligned} \tag{5.47}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\left[V_1^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.7778 \\ 0.2222 \end{bmatrix} = \begin{bmatrix} 1\underline{/0^\circ} \\ 0.6938\underline{/-135.998^\circ} \\ 0.6938\underline{/135.998^\circ} \end{bmatrix}} \tag{5.48}$$

For bus 3,
zero sequence voltage is zero

$$V_3^{(0)} = 0 \tag{5.49}$$

positive and negative sequence voltage at bus 3 can be calculated as

$$\begin{aligned} V_3^{(1)} &= E_3^{(1)} - Z_{35}^{(1)} I_5^{(1)} \\ &= 1.0 - j0.125 * (-j2.222) \\ &= 0.7222\ pu \end{aligned} \tag{5.50}$$

$$\begin{aligned} V_3^{(2)} &= -Z_{35}^{(2)} I_5^{(2)} \\ &= -j0.125 * (j2.222) \\ &= 0.2778\ pu \end{aligned} \tag{5.51}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\left[V_1^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.7222 \\ 0.2778 \end{bmatrix} = \begin{bmatrix} 1\underline{/0^\circ} \\ 0.6309\underline{/-142.410^\circ} \\ 0.6309\underline{/142.410^\circ} \end{bmatrix}} \tag{5.52}$$

For bus 4,

zero sequence voltage is zero

$$V_4^{(0)} = 0 \tag{5.53}$$

positive and negative sequence voltage at bus 4 can be calculated as

$$\begin{aligned} V_4^{(1)} &= E_1^{(1)} - Z_{45}^{(1)} I_5^{(1)} \\ &= 1.0 - j0.125 * (-j2.222) \\ &= 0.7222\ pu \end{aligned} \tag{5.54}$$

$$\begin{aligned} V_4^{(2)} &= -Z_{45}^{(2)} I_5^{(2)} \\ &= -j0.125 * (j2.222) \\ &= 0.2778\ pu \end{aligned} \tag{5.55}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\left[V_4^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.7222 \\ 0.2778 \end{bmatrix} = \begin{bmatrix} 1\underline{/0^\circ} \\ 0.6309\underline{/-142.410^\circ} \\ 0.6309\underline{/142.410^\circ} \end{bmatrix}} \tag{5.56}$$

For bus 5,

zero sequence voltage is zero

$$V_5^{(0)} = 0 \tag{5.57}$$

positive and negative sequence voltage at bus 5 can be calculated as

$$\begin{aligned} V_5^{(1)} &= E_5^{(1)} - Z_{55}^{(1)} I_5^{(1)} \\ &= 1.0 - j0.175 * (-j2.222) \\ &= 0.6111\ pu \end{aligned} \tag{5.58}$$

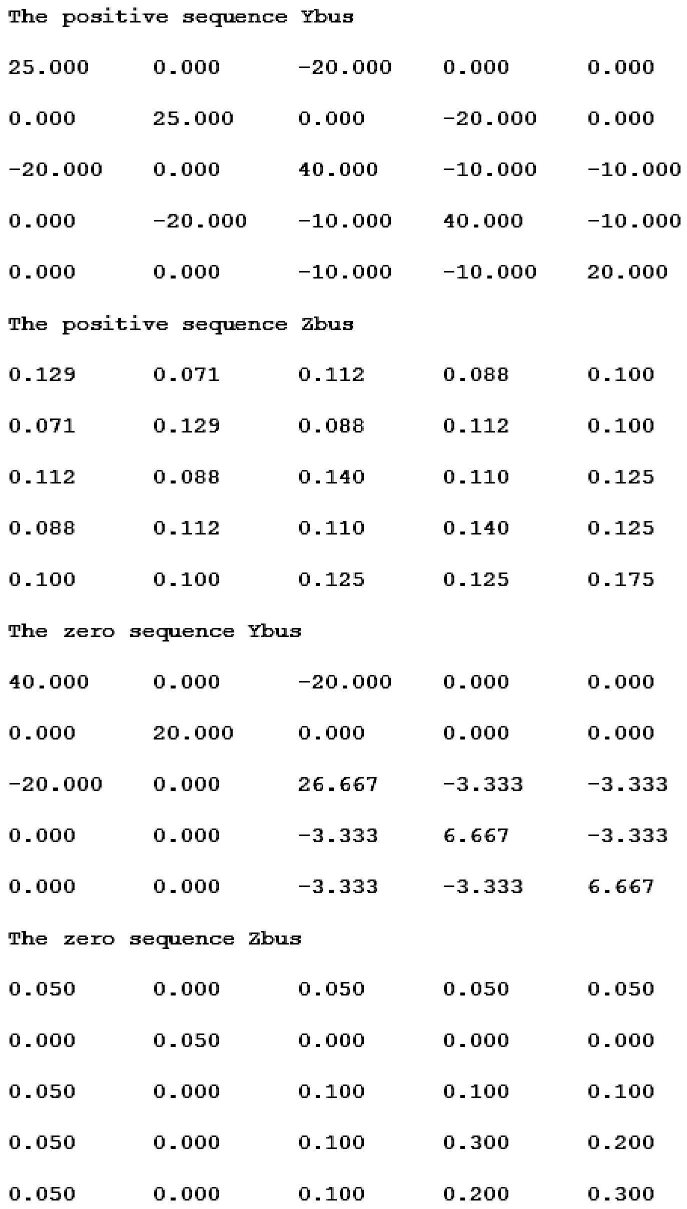
$$\begin{aligned} V_5^{(2)} &= -Z_{55}^{(2)} I_5^{(2)} \\ &= -j0.175 * (j2.222) \\ &= 0.3889\ pu \end{aligned} \tag{5.59}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\begin{bmatrix} V_5^{abc} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.6111 \\ 0.3889 \end{bmatrix} = \begin{bmatrix} 1\underline{/0^\circ} \\ 0.5357\underline{/-158.953^\circ} \\ 0.5357\underline{/158.953^\circ} \end{bmatrix}} \tag{5.60}$$

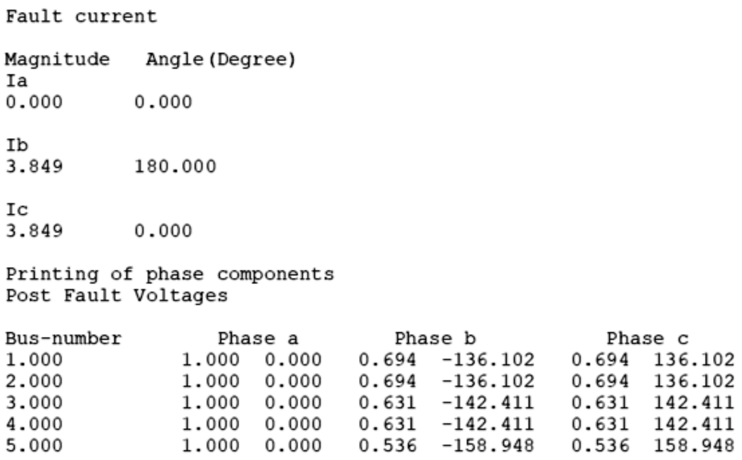
5.3.2 MATLAB toolbox results

Line to line fault analysis results from the developed MATLAB tool box are shown in the figures below. Figure 5-7 shows the calculated values of impedance matrices and figure 5-8 shows the fault currents and fault voltages at each bus and finally, figure 5-9 shows the GUI screen with transient graphs.



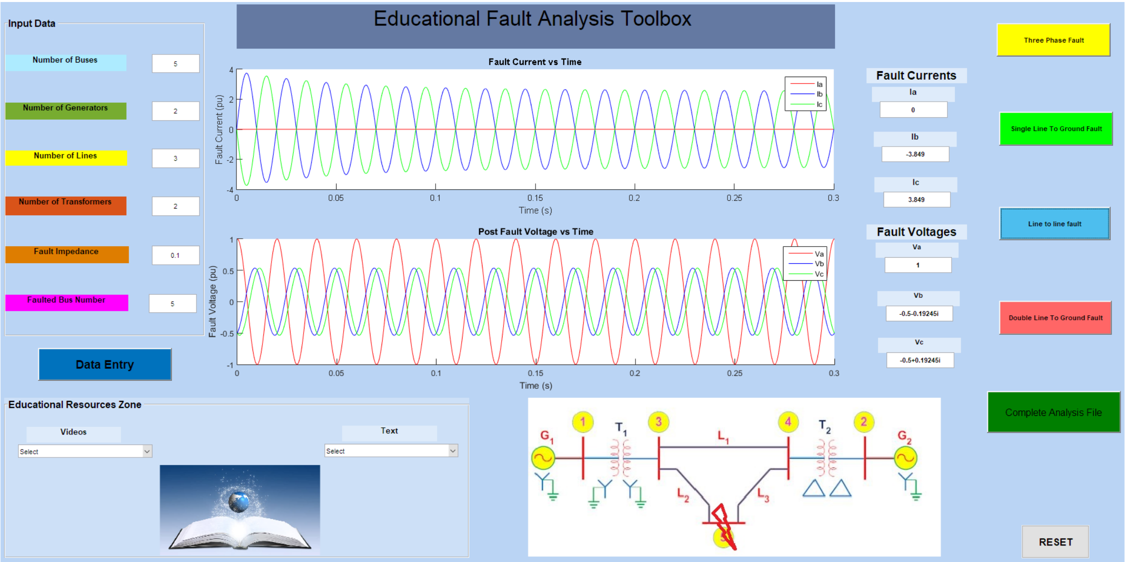
```
The positive sequence Ybus

25.000     0.000      -20.000    0.000      0.000

0.000      25.000     0.000      -20.000    0.000

-20.000    0.000      40.000     -10.000    -10.000

0.000      -20.000    -10.000    40.000     -10.000

0.000      0.000      -10.000    -10.000    20.000

The positive sequence Zbus

0.129      0.071      0.112      0.088      0.100

0.071      0.129      0.088      0.112      0.100

0.112      0.088      0.140      0.110      0.125

0.088      0.112      0.110      0.140      0.125

0.100      0.100      0.125      0.125      0.175

The zero sequence Ybus

40.000     0.000      -20.000    0.000      0.000

0.000      20.000     0.000      0.000      0.000

-20.000    0.000      26.667     -3.333     -3.333

0.000      0.000      -3.333     6.667      -3.333

0.000      0.000      -3.333     -3.333     6.667

The zero sequence Zbus

0.050      0.000      0.050      0.050      0.050

0.000      0.050      0.000      0.000      0.000

0.050      0.000      0.100      0.100      0.100

0.050      0.000      0.100      0.300      0.200

0.050      0.000      0.100      0.200      0.300
```

Figure 5-7: Calculated impedance matrices of line to line fault



```
Fault current

Magnitude   Angle(Degree)
Ia
0.000       0.000

Ib
3.849       180.000

Ic
3.849       0.000

Printing of phase components
Post Fault Voltages

Bus-number          Phase a        Phase b          Phase c
1.000          1.000  0.000   0.694  -136.102   0.694  136.102
2.000          1.000  0.000   0.694  -136.102   0.694  136.102
3.000          1.000  0.000   0.631  -142.411   0.631  142.411
4.000          1.000  0.000   0.631  -142.411   0.631  142.411
5.000          1.000  0.000   0.536  -158.948   0.536  158.948
```

Figure 5-8: Fault currents and fault voltages of line to line fault



Input Data

Number of Buses: 5

Number of Generators: 2

Number of Lines: 3

Number of Transformers: 2

Fault Impedance: 0.1

Faulted Bus Number: 5

Fault Currents

Ia: 0

Ib: -3.849

Ic: 3.849

Fault Voltages

Va: 1

Vb: -0.5-0.19245i

Vc: -0.5+0.19245i

Figure 5-9: Line to line fault analysis - MATLAB tool box

line to line analysis of the fault conditions shows that fault current at phase a is zero and voltage at phase a is unchanged. Also voltages at both phase b and phase c are equal. Reason for this is that the fault is between phase b and phase c and they are in contact with each other. Also, the fault does not involve phase a.

5.4 Double line-to-ground

5.4.1 Hand calculation results

All three positive, negative and zero sequence networks are needed to do the calculations for the double line to ground fault. Therefore, the same $Z^{(1)_{bus}}$, $Z^{(2)_{bus}}$ and $Z^{(2)_{bus}}$ calculated before will be used. The fault is created involving phase 'b' and phase 'c' with fault impedance of $Z_f = 0$.

$$\left[Z^{(1)}_{bus}\right] = \left[Z^{(2)}_{bus}\right] = \begin{bmatrix} j0.1294 & j0.0706 & j0.1118 & j0.0882 & j0.10 \\ j0.0706 & j0.1294 & j0.0882 & j0.1118 & j0.10 \\ j0.1118 & j0.0882 & j0.1397 & j0.1103 & j0.1250 \\ j0.0882 & j0.1118 & j0.1103 & j0.1397 & j0.10 \\ j0.10 & j0.10 & j0.1250 & j0.1250 & j0.1750 \end{bmatrix} pu \tag{5.61}$$

$$\left[Z^{(0)}_{bus}\right] = \begin{bmatrix} j0.05 & 0.0 & j0.05 & j0.05 & j0.05 \\ 0.0 & j0.05 & 0.0 & 0.0 & 0.0 \\ j0.05 & 0.0 & j0.10 & j0.10 & j0.10 \\ j0.05 & 0.0 & j0.10 & j0.30 & j0.20 \\ j0.05 & 0.0 & j0.10 & j0.20 & j0.30 \end{bmatrix} pu \tag{5.62}$$

Positive sequence current of double line to ground fault is calculated using equation 3.34

$$\boxed{\begin{aligned} I_5^{(1)} &= \frac{1.0}{j0.175 + \dfrac{j0.175(j0.30 + 3*0)}{(j0.30 + j0.175 + 3*0)}} \\ &= -j3.5023 \end{aligned}} \tag{5.63}$$

The negative and zero sequence currents are calculated using current division as in equation 3.35.

$$\boxed{\begin{aligned} I_5^{(2)} &= -\frac{j0.30}{(j0.30+j0.175)} * -j3.5023 \\ &= j2.212\ pu \end{aligned}} \tag{5.64}$$

$$\boxed{\begin{aligned} I_5^{(0)} &= -\frac{j0.175}{(j0.30+j0.175)} * -j3.5023 \\ &= j1.2903\ pu \end{aligned}} \tag{5.65}$$

Therefore, The phase components of the fault current at bus 5 can be obtained using the transformation matrix A in equation 2.17.

$$\boxed{\left[I_5^{(abc)}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} j1.2903 \\ -j3.503 \\ j2.212 \end{bmatrix} = \begin{bmatrix} 0 \\ 5.313\underline{/158.64^\circ} \\ 5.313\underline{/21.36^\circ} \end{bmatrix} pu} \tag{5.66}$$

Bus voltage are calculated using the equation 3.36 in chapter 3.

$$\begin{aligned} V_i^{(0)} &= -Z_{ik}^{(0)} I_k^{(0)} \\ V_i^{(1)} &= E_i^{(1)}(0) - Z_{ik}^{(1)} I_k^{(1)} \\ V_i^{(2)} &= -Z_{ik}^{(2)} I_k^{(2)} \end{aligned} \tag{5.67}$$

The above equations can be written in compact form as

$$\begin{bmatrix} V_i^{(0)} \\ V_i^{(1)} \\ V_i^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ V_i \\ 0 \end{bmatrix} - \begin{bmatrix} Z_{ik}^{(0)} & 0 & 0 \\ 0 & Z_{ik}^{(1)} & 0 \\ 0 & 0 & Z_{ik}^{(2)} \end{bmatrix} \begin{bmatrix} I_k^{(0)} \\ I_k^{(1)} \\ I_k^{(2)} \end{bmatrix} \tag{5.68}$$

The Fault voltages at bus 1 are:

$$\begin{bmatrix} V_1^{(0)} \\ V_1^{(1)} \\ V_1^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.05 & 0 & 0 \\ 0 & j0.10 & 0 \\ 0 & 0 & j0.10 \end{bmatrix} \begin{bmatrix} j1.290 \\ -j3.503 \\ j2.212 \end{bmatrix} \tag{5.69}$$

Therefore,

$$\begin{bmatrix} V_1^{(012)} \end{bmatrix} = \begin{bmatrix} 0.0645 \\ 0.6497 \\ 0.2212 \end{bmatrix} pu \tag{5.70}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\begin{bmatrix} V_1^{abc} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0.0645 \\ 0.6497 \\ 0.2212 \end{bmatrix} = \begin{bmatrix} 0.9355\underline{/0^\circ} \\ 0.5248\underline{/-134.99^\circ} \\ 0.5248\underline{/134.99^\circ} \end{bmatrix}} \tag{5.71}$$

Fault voltages at bus 2 are:

$$\begin{bmatrix} V_2^{(0)} \\ V_2^{(1)} \\ V_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & j0.10 & 0 \\ 0 & 0 & j0.10 \end{bmatrix} \begin{bmatrix} j1.290 \\ -j3.503 \\ j2.212 \end{bmatrix} \tag{5.72}$$

Therefore,

$$\begin{bmatrix} V_2^{(012)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.6497 \\ 0.2212 \end{bmatrix} pu \tag{5.73}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\begin{bmatrix} V_2^{abc} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0.6497 \\ 0.2212 \end{bmatrix} = \begin{bmatrix} 0.8710\underline{/0^\circ} \\ 0.5722\underline{/-139.56^\circ} \\ 0.5722\underline{/139.56^\circ} \end{bmatrix}} \tag{5.74}$$

The fault voltages at bus 3 are:

$$\begin{bmatrix} V_3^{(0)} \\ V_3^{(1)} \\ V_3^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.10 & 0 & 0 \\ 0 & j0.125 & 0 \\ 0 & 0 & j0.125 \end{bmatrix} \begin{bmatrix} j1.290 \\ -j3.503 \\ j2.212 \end{bmatrix} \tag{5.75}$$

Therefore,

$$\left[V_3^{(012)}\right] = \begin{bmatrix} 0.1290 \\ 0.5622 \\ 0.2765 \end{bmatrix} pu \tag{5.76}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\left[V_3^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0.1290 \\ 0.5622 \\ 0.2765 \end{bmatrix} = \begin{bmatrix} 0.9766\underline{/0^\circ} \\ 0.3815\underline{/-139.56^\circ} \\ 0.3815\underline{/139.56^\circ} \end{bmatrix}} \tag{5.77}$$

Fault voltages at bus 4 are:

$$\begin{bmatrix} V_4^{(0)} \\ V_4^{(1)} \\ V_4^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.20 & 0 & 0 \\ 0 & j0.125 & 0 \\ 0 & 0 & j0.125 \end{bmatrix} \begin{bmatrix} j1.290 \\ -j3.503 \\ j2.212 \end{bmatrix} \tag{5.78}$$

Therefore,

$$\left[V_4^{(012)}\right] = \begin{bmatrix} 0.2581 \\ 0.5622 \\ 0.2765 \end{bmatrix} pu \tag{5.79}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\left[V_4^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0.2581 \\ 0.5622 \\ 0.2765 \end{bmatrix} = \begin{bmatrix} 1.0968\underline{/0^\circ} \\ 0.2954\underline{/-123.10^\circ} \\ 0.2954\underline{/123.10^\circ} \end{bmatrix}} \tag{5.80}$$

Fault voltages at bus 5 are:

$$\begin{bmatrix} V_5^{(0)} \\ V_5^{(1)} \\ V_5^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix} - \begin{bmatrix} j0.30 & 0 & 0 \\ 0 & j0.175 & 0 \\ 0 & 0 & j0.175 \end{bmatrix} \begin{bmatrix} j1.290 \\ -j3.503 \\ j2.212 \end{bmatrix} \tag{5.81}$$

Therefore,

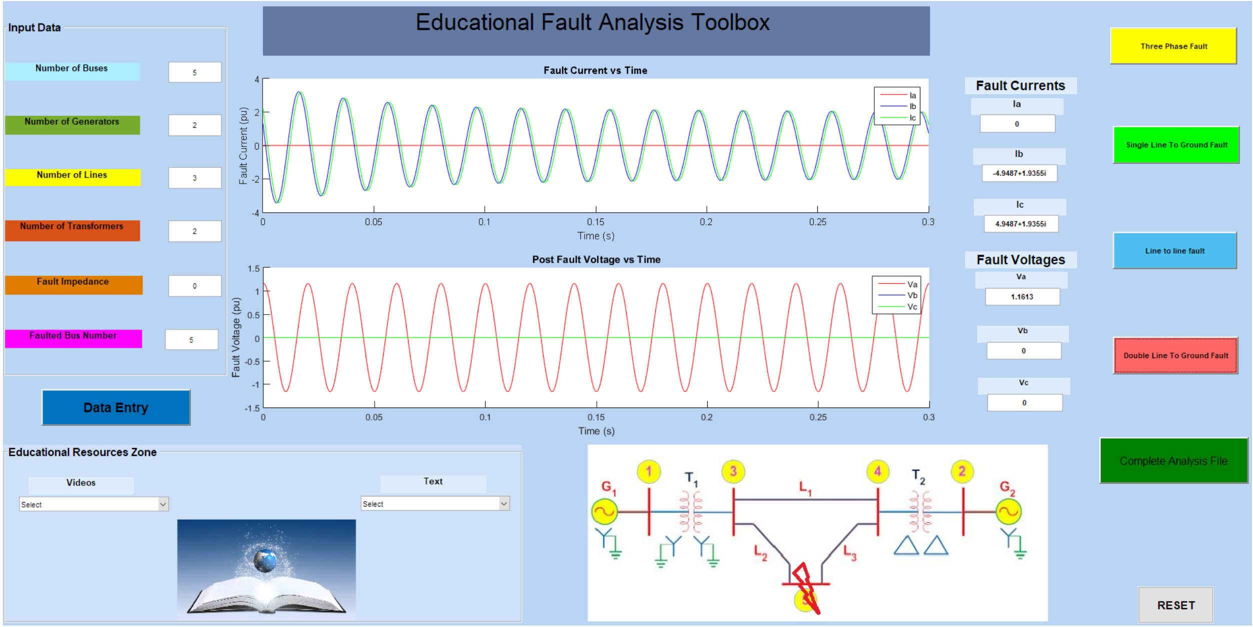
$$\left[V_5^{(012)}\right] = \begin{bmatrix} 0.3871 \\ 0.3871 \\ 0.3871 \end{bmatrix} pu \tag{5.82}$$

Using the transformation matrix in equation 2.16, phase components of the voltage can be obtained as:

$$\boxed{\left[V_5^{abc}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & 1 & a^2 \end{bmatrix} \begin{bmatrix} 0.3871 \\ 0.3871 \\ 0.3871 \end{bmatrix} = \begin{bmatrix} 1.1613\underline{/0^\circ} \\ 0 \\ 0 \end{bmatrix}} \tag{5.83}$$

5.4.2 MATLAB toolbox results

Double line to ground fault analysis was also done using the developed MATLAB tool box and results are shown in the figures below. Figure 5-10 shows the GUI screen with transient graphs and figure 5-11 shows the fault currents and fault voltages at each bus. Finally, figure 5-12 shows the calculated values of impedance matrices. Analysis shows that the phase voltages for phase b and phase c are zero. These phases are in contact with ground and fault current is only flowing through faulted phases only. Double line to ground fault analysis shows that the phase voltages for phase b and phase c are zero. These phases are in contact with ground and fault current is only flowing through faulted phases only.



Number of Buses 5
Number of Generators 2
Number of Lines 3
Number of Transformers 2
Fault Impedance 0
Faulted Bus Number 5

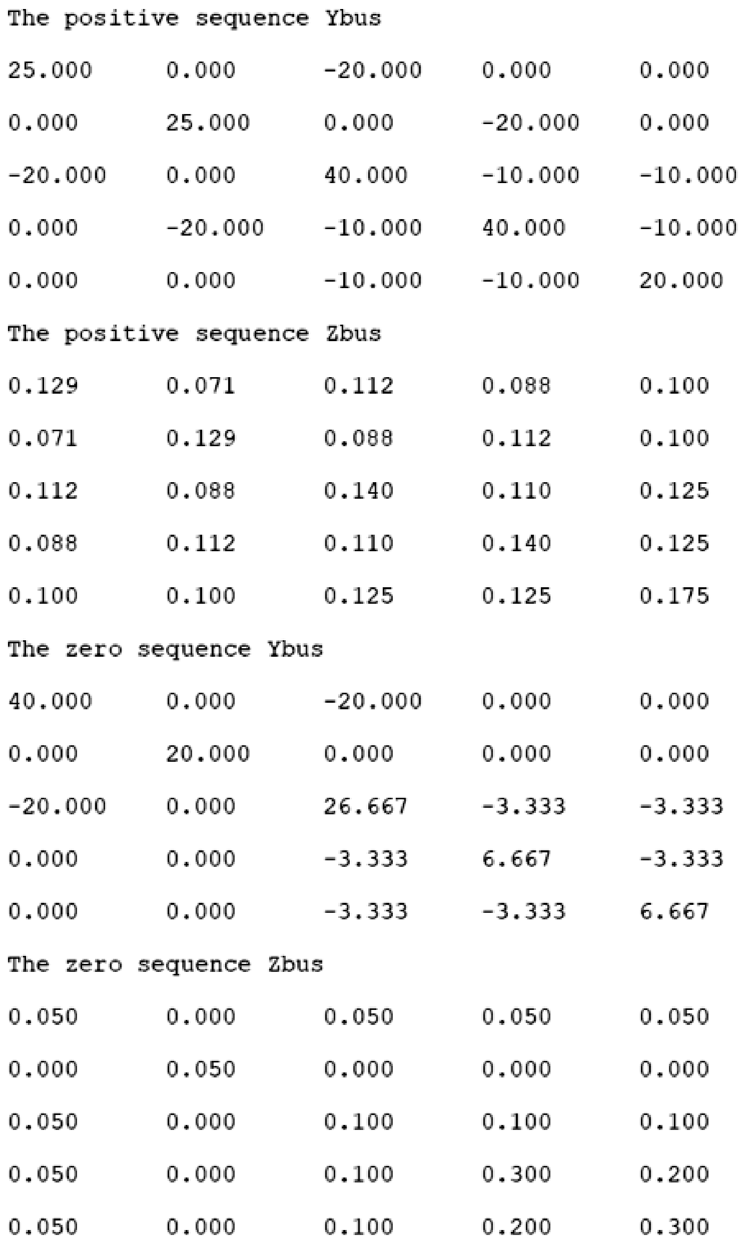
Fault Currents
Ia 0
Ib -4.9487+1.9355i
Ic 4.9487+1.9355i

Fault Voltages
Va 1.1613
Vb 0
Vc 0

Figure 5-10: Double line to ground fault analysis - MATLAB tool box

```
Fault current

Magnitude    Angle(Degree)
Ia
0.000       0.000

Ib
5.314       158.639

Ic
5.314       21.361

Printing of phase components
Post Fault Voltages

Bus-number          Phase a        Phase b          Phase c
1.000           0.935  0.000   0.525  -134.986   0.525  134.986
2.000           0.871  0.000   0.572  -139.560   0.572  139.560
3.000           0.968  0.000   0.381  -139.560   0.381  139.560
4.000           1.097  0.000   0.295  -123.098   0.295  123.098
5.000           1.161  0.000   0.000  -0.000   0.000  0.000
```

Figure 5-11: Fault currents and fault voltages of double line to ground fault



```
The positive sequence Ybus
25.000     0.000      -20.000    0.000      0.000
0.000      25.000     0.000      -20.000    0.000
-20.000    0.000      40.000     -10.000    -10.000
0.000      -20.000    -10.000    40.000     -10.000
0.000      0.000      -10.000    -10.000    20.000
The positive sequence Zbus
0.129      0.071      0.112      0.088      0.100
0.071      0.129      0.088      0.112      0.100
0.112      0.088      0.140      0.110      0.125
0.088      0.112      0.110      0.140      0.125
0.100      0.100      0.125      0.125      0.175
The zero sequence Ybus
40.000     0.000      -20.000    0.000      0.000
0.000      20.000     0.000      0.000      0.000
-20.000    0.000      26.667     -3.333     -3.333
0.000      0.000      -3.333     6.667      -3.333
0.000      0.000      -3.333     -3.333     6.667
The zero sequence Zbus
0.050      0.000      0.050      0.050      0.050
0.000      0.050      0.000      0.000      0.000
0.050      0.000      0.100      0.100      0.100
0.050      0.000      0.100      0.300      0.200
0.050      0.000      0.100      0.200      0.300
```

Figure 5-12: Calculated impedance matrices of double line to ground fault

CHAPTER 6

Conclusion

In this thesis, MATLAB based educational tool box has been developed to analyze and help students with the short circuit faults in power systems. The symmetrical component method was used to set up the mathematical models to calculate the three phase fault, single line to ground fault, line to line fault and the double line to ground fault in a power systems. Developed MATLAB tool box has a user friendly graphical user interface to help students with calculating fault conditions without having to learn a language or type commands. Data of two different power systems already entered in the software by default for students to learn and compare their results with the software results. And, the student has the ability to enter data for new power system to perform the analysis. After entering the system parameters to the input data section of the software and by choosing the fault type, all the calculations are performed. All the results are displayed in the graphical user interface with transients graphs for fault currents and post fault voltages. Also, all the calculated results are printed to .txt file and student can open it by pressing the "Complete Analysis file" button on the interface. A so called "Educational resources zone " also included in the software with series of educational videos and text documents on power system analysis to help student understand theories while they do the calculations. A sample system of non loaded 5 bus power system with two generators, two transformers and three transmission lines has been implemented to show the validity of the developed educational toolbox. The software has been operated using the values listed in table 5.1 and results were compared with the hand calculations to validate the accuracy of hand calculation results. The fault were created at the bus 5 of the system. All the hand calculated results for each type of faults agreed with software results and validity of the software has been verified.

Some observations have been made based on the analysis.In a three phase bolted fault where Z_f=0, fault currents are at the maximum value and the voltages dropped in all three phases to zero during the fault at the faulted bus. Also, analysis for single line to ground shows that the fault current of phase b and phase c are zero due to the fact that only the phase a is involved in the fault. In the line to line analysis of the fault conditions shows that fault current at phase a is zero and voltage at phase a is unchanged. Also voltages at both phase b and phase c are equal. Reason for this is that the fault is between phase b and phase c and they are in contact with each other. Also, the fault does not involve phase a. Double line to ground fault analysis shows that the phase voltages for phase b and phase c are zero. These phases are in contact with ground and fault current is only flowing through faulted phases only.

Looking at the transient graphs for fault current for each fault, one can observe that there are three sections in the graph. First period is the sub transient period and fault current is very large and it falls rapidly during this period. Second sections is the transient period and fault current falls at a slower rate than the sub transient period. Finally the current reaches the steady value and this section is known as the steady state period. Usually, the sub transient fault current can be as much as ten times the steady state current and transient current can be five times the steady state current.

APPENDIX A

MATLAB Code

A.1 MATLAB code for the main fault analysis tool box

```
function varargout = Fault_Analysis_GUI(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Gui_OpeningFcn, ...
                   'gui_OutputFcn',  @Gui_OutputFcn, ...
                   'gui_LayoutFcn',  []  , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end


function Gui_OpeningFcn(hObject, ~, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function popupmenu5_CreateFcn (varargin)
function popupmenu4_CreateFcn (varargin)
function axes6_CreateFcn (varargin)
function edit32_CreateFcn (varargin)
function edit19_CreateFcn (varargin)
function edit19_DeleteFcn (varargin)



% --- Outputs from this function are returned to the command line.
function varargout = Gui_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;
```

```

axes(handles.axes6)
matlabImage = imread('bc.jpg');
image(matlabImage)
axis off
axis image
axes(handles.axes8)
matlabImage = imread('education.jpg');
image(matlabImage)
axis off
axis image


%——Input data——%

function Input_nb_Callback(hObject, eventdata, handles)
nbuses= str2double(get(handles.Input_nb, 'String'))

if nbuses==5
axes(handles.axes6)
matlabImage = imread('5bus.JPG');
image(matlabImage)
axis off
axis image

elseif nbuses==6
axes(handles.axes6)
matlabImage = imread('6bus.JPG');
```

```
image(matlabImage)
axis off
axis image
else
axes(handles.axes6)
matlabImage = imread('bc.jpg');
image(matlabImage)
axis off
axis image
end


function Input_ng_Callback(hObject, eventdata, handles)
ngen= str2double(get(handles.Input_ng, 'String'))

function Input_nt_Callback(hObject, eventdata, handles)
nxfmrs = str2double(get(handles.Input_nt, 'String'))

function Input_nl_Callback(hObject, eventdata, handles)
nlines = str2double(get(handles.Input_nl, 'String'))

function edit18_Callback(hObject, eventdata, handles)
fault_imp= str2double(get(handles.edit18, 'String'))

function edit19_Callback(hObject, eventdata, handles)
fault_bus= str2double(get(handles.edit19, 'String'))

nbuses= str2double(get(handles.Input_nb, 'String'))
```

```

if nbuses==5
if fault_bus==1
      for k=1:5
      axes(handles.axes6)
matlabImage = imread('5bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('5bus1.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

end


end

if fault_bus==2
      for k=1:5
      axes(handles.axes6)
matlabImage = imread('5bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('5bus2.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

end


      end
if fault_bus==3
      for k=1:5
      axes(handles.axes6)
matlabImage = imread('5bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('5bus3.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

```

```
end


    end
 if fault_bus==4
     for k=1:5
     axes(handles.axes6)
matlabImage = imread('5bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('5bus4.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

end


end

 if fault_bus==5
     for k=1:5
axes(handles.axes6)
matlabImage = imread('5bus.JPG');
```

```
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('5bus5.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

    end
end
end

if nbuses==6
if fault_bus==1
    for k=1:5
    axes(handles.axes6)
matlabImage = imread('6bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('6bus1.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

end


end

if fault_bus==2
      for k=1:5
      axes(handles.axes6)
matlabImage = imread('6bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('6bus2.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

end


      end
if fault_bus==3
```

```
     for k=1:5
     axes(handles.axes6)
matlabImage = imread('6bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('6bus3.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

end


     end
 if fault_bus==4
     for k=1:5
     axes(handles.axes6)
matlabImage = imread('6bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('6bus4.JPG');
```

```
image(matlabImage)
axis off
axis image
pause(0.5)

end


     end
 if fault_bus==5
     for k=1:5
     axes(handles.axes6)
matlabImage = imread('6bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('6bus5.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

     end



```

```
end
      if fault_bus==6
      for k=1:5
      axes(handles.axes6)
matlabImage = imread('6bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('6bus6.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)

      end
      else

for k=1:5
axes(handles.axes6)
matlabImage = imread('6bus.JPG');
image(matlabImage)
axis off
axis image
pause(0.5)
axes(handles.axes6)
matlabImage = imread('6buse.JPG');
```

```
image(matlabImage)
axis off
axis image
pause(0.5)
end

    end
end


% --- Executes on button press in Single_line_to_ground_fault.
function slgfault_Callback(hObject, eventdata, handles)
nbuses= str2double(get(handles.Input_nb, 'String'));
nxfmrs = str2double(get(handles.Input_nt, 'String'));
nlines = str2double(get(handles.Input_nl, 'String'));
ngen= str2double(get(handles.Input_ng, 'String'));
fault_imp= str2double(get(handles.edit18, 'String'));;
fault_bus= str2double(get(handles.edit19, 'String'));
Yp=Poistive_ybus(nbuses,ngen,nlines,nxfmrs);
Yz=Zero_ybus(nbuses,ngen,nlines,nxfmrs);
Zp=inv(Yp);
Zn=Zp;
check=diag(Yz);
k=find(check==0);
if length(k)==0
    Zz=inv(Yz);
end
if length(k)~=0
```

```
    Zz=pinv(Yz);
end
% pre-fault voltage vectors
vector1=ones(nbuses);
V_1=vector1(:,1);
vector0=zeros(nbuses);
V_2=vector0(:,1);
V_0=V_2;
%calculation of fault current
I_1=V_1(fault_bus)/((Zp(fault_bus,fault_bus)+Zn(fault_bus,
    fault_bus)+Zz(fault_bus,fault_bus)+3*fault_imp)*1i);;
I_2=I_1;;
I_0=I_1;;
a=-0.5+(0.866025*1i);;
i012=[I_0;I_1;I_2];;
p=[1 1 1;1 a^2 a;1 a a^2];;
iabc=p*i012;
ia=round (iabc(1),5);
ib=round (iabc(2),5);
ic=round (iabc(3),5);
set(handles.I1_edit, 'String', num2str(ia));
set(handles.I2_edit, 'String', num2str(ib));
set(handles.I0_edit, 'String', num2str(ic));
%For time variation of fault current and voltage signal
cla((handles.Fault_current_axes))
axes(handles.Fault_current_axes);
xd2=Zp(fault_bus,fault_bus)*j;
xd1=0.35j;
```

```
xd=0.5j;
z2=0.175j;
z0=0.30j;
zf=fault_imp*j;
td2=0.035;
td1=0.5;
Ta=0.17;
w=2*pi*50;
q=0;
t = 0:0.0001:0.3;
xp=(((((1/xd2)-(1/xd1)).*exp(-t/td2))+(((1/xd1)-(1/xd)).*exp(-t/td1))+(1/xd)));
x1=(xp.^-1);
ia=3*(1./(x1+z2+z0+(3*zf)).*sin(2*pi*50*t+(q)));-(1./xd2).*sin(pi/2)*exp(-t/Ta);
ib=t*0;
ic=t*0;
hold on
plot(t,imag(ia),'r')
plot(t,imag(ib),'b')
plot(t,imag(ic),'g')
title('Fault Current vs Time')
xlabel('Time (s)')
ylabel('Fault Current (pu)')
legend('Ia','Ib','Ic');

%post fault voltage calculations
for i=1:nbuses
```

```

    V_1p(i)=V_1(i)-(Zp(i,fault_bus)*I_1*j);
    V_2p(i)=V_2(i)-Zn(i,fault_bus)*I_2*j;
    V_0p(i)=V_0(i)-Zz(i,fault_bus)*I_0*j;
end
V_1p_mag=abs(V_1p);
V_1p_phase=angle(V_1p);
V_2p_mag=abs(V_2p);
V_2p_phase=angle(V_2p);
V_0p_mag=abs(V_0p);
V_0p_phase=angle(V_0p);

V012=[V_0p(fault_bus);V_1p(fault_bus);V_2p(fault_bus)]
p=[1 1 1;1 a^2 a;1 a a^2];
Vabc=p*V012
V_1(i)
set(handles.V1_edit, 'String', num2str(Vabc(1)))
set(handles.V2_edit, 'String', num2str(Vabc(2)))
set(handles.V0_edit, 'String', num2str(Vabc(3)))

Va=abs(Vabc(1))*cos(w*t+angle(Vabc(1)));
Vb=abs(Vabc(2))*cos(w*t+angle(Vabc(2)));
Vc=abs(Vabc(3))*cos(w*t+angle(Vabc(3)));
cla((handles.Fault_voltages_axes))
axes(handles.Fault_voltages_axes);
hold on
plot(t,Va,'r')
plot(t,Vb,'b')
plot(t,Vc,'g')
title('Post Fault Voltage vs Time')
xlabel('Time (s)')
ylabel('Fault Voltage (pu)')
legend('Va','Vb','Vc');

%data entry into output file
fid=fopen('output.txt','w');
fprintf(fid,'The positive sequence Ybus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Yp(i,k));
    end
    fprintf(fid,'\n\n');
end
fprintf(fid,'The positive sequence Zbus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Zp(i,k));
    end
    fprintf(fid,'\n\n');
end

fprintf(fid,'The zero sequence Ybus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Yz(i,k));
    end
    fprintf(fid,'\n\n');
end
fprintf(fid,'The zero sequence Zbus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Zz(i,k));
    end
    fprintf(fid,'\n\n');
end

fprintf(fid,'Fault current\n\n');
fprintf(fid,'Magnitude   Angle(radians)\n');
fprintf(fid,'I1\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_1),angle(I_1));
fprintf(fid,'I2\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_2),angle(I_2));
fprintf(fid,'I0\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_0),angle(I_0));
fprintf(fid,'Printing of sequence components\n');
fprintf(fid,'Post Fault Voltages\n\n');
fprintf(fid,'Bus-number  Pos-seq-Magnitude  Pos-seq-Angle(radians)  Neg-seq-Magnitude  Neg-seq-Angle(radians)  Zero-seq-Magnitude  Zero-seq-Angle(radians)\n');
for i=1:length(V_1p)
    fprintf(fid,'%-10.3f  %-10.3f  %-10.3f  %-10.3f  %-10.3f  %-10.3f  %-10.3f\n',i,abs(V_1p(i)),V_1p_phase(i),abs(V_2p(i)),V_2p_phase(i),abs(V_0p(i)),angle(V_0p(i)));
end
```

```
fprintf(fid,'\n\n');
fclose('all');
clc



% --- Executes on button press in Line_to_Line_Fault.
function llfault_Callback(hObject, eventdata, handles)
nbuses= str2double(get(handles.Input_nb, 'String'));
nxfmrs = str2double(get(handles.Input_nt, 'String'));
nlines = str2double(get(handles.Input_nl, 'String'));
ngen= str2double(get(handles.Input_ng, 'String'));
fault_imp= str2double(get(handles.edit18, 'String'));
fault_bus= str2double(get(handles.edit19, 'String'));
Yp=Poistive_ybus(nbuses,ngen,nlines,nxfmrs);
Yz=Zero_ybus(nbuses,ngen,nlines,nxfmrs);
Zp=inv(Yp);
Zn=Zp;
check=diag(Yz);
k=find(check==0);
if length(k)==0
    Zz=inv(Yz);
end
if length(k)~=0
    Zz=pinv(Yz);
end
%preparation of pre-fault voltage vectors
vector1=ones(nbuses);
```

```
V_1=vector1(:,1);
vector0=zeros(nbuses);
V_2=vector0(:,1);
V_0=V_2;
%calculation of fault current
I_1=V_1(fault_bus)/((Zp(fault_bus,fault_bus)+Zn(fault_bus,
    fault_bus)+fault_imp)*j);
I_2=-I_1;
I_0=0;
a=-0.5+(0.866025*1i)
i012=[I_0;I_1;I_2]
p=[1 1 1;1 a^2 a;1 a a^2]
iabc=p*i012
ia=round (iabc(1),5)
ib=round (iabc(2),5)
ic=round (iabc(3),5)
set(handles.I1_edit, 'String', num2str(ia))
set(handles.I2_edit, 'String', num2str(ib))
set(handles.I0_edit, 'String', num2str(ic))
%For time variation of fault current and voltage signal
cla((handles.Fault_current_axes))
axes(handles.Fault_current_axes);
xd2=Zp(fault_bus,fault_bus)*j
xd1=0.35j;
xd=0.5j;
z2=0.175j;
z0=0.30j;
zf=fault_imp*j;
```

```
td2=0.035;
td1=0.5;
Ta=0.17;
w=2*pi*50;
q=pi/2;
t = 0:0.0001:0.3;
xp=(((((1/xd2)-(1/xd1)).*exp(-t/td2))+(((1/xd1)-(1/xd)).*exp(-t/
    td1))+(1/xd)));
x1=(xp.^-1);
IB= (1./(x1+z2+zf))*((a^2)-a);
IB(1)
IC=-IB;
IC(1)
ia=t*0;
ib=IB.*cos(w*t+(q));
ic=-IB.*cos(w*t+(q));
hold on
plot(t,real(ia),'r')
plot(t,real(ib),'b')
plot(t,real(ic),'g')
title('Fault Current vs Time')
xlabel('Time (s)')
ylabel('Fault Current (pu)')
legend('Ia','Ib','Ic');

%post fault voltage calculations
for i=1:nbuses
    V_1p(i)=V_1(i)-Zp(i,fault_bus)*I_1*j;
```

```
        V_2p(i)=V_2(i)-Zn(i,fault_bus)*I_2*j;
        V_0p(i)=V_0(i)-Zz(i,fault_bus)*I_0*j;
end
V_1p_mag=abs(V_1p);
V_1p_phase=angle(V_1p);
V_2p_mag=abs(V_2p);
V_2p_phase=angle(V_2p);
V_0p_mag=abs(V_0p);
V_0p_phase=angle(V_0p);

V012=[V_0p(fault_bus);V_1p(fault_bus);V_2p(fault_bus)];
p=[1 1 1;1 a^2 a;1 a a^2];
Vabc=p*V012;

set(handles.V1_edit, 'String', num2str(Vabc(1)))
set(handles.V2_edit, 'String', num2str(Vabc(2)))
set(handles.V0_edit, 'String', num2str(Vabc(3)))
Va=abs(Vabc(1))*cos(w*t+angle(Vabc(1)));
Vb=abs(Vabc(2))*cos(w*t+angle(Vabc(2)));
Vc=abs(Vabc(3))*cos(w*t+angle(Vabc(3)));
cla((handles.Fault_voltages_axes))
axes(handles.Fault_voltages_axes);
hold on
plot(t,Va,'r')
plot(t,Vb,'b')
plot(t,Vc,'g')
title('Post Fault Voltage vs Time')
xlabel('Time (s)')
```

```
ylabel('Fault Voltage (pu)')
legend('Va','Vb','Vc');

%data entry into output file
fid=fopen('output.txt','w');
fprintf(fid,'The positive sequence Ybus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Yp(i,k));
    end
    fprintf(fid,'\n\n');
end
fprintf(fid,'The positive sequence Zbus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Zp(i,k));
    end
    fprintf(fid,'\n\n');
end

fprintf(fid,'The zero sequence Ybus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Yz(i,k));
    end
    fprintf(fid,'\n\n');
end
fprintf(fid,'The zero sequence Zbus\n\n');
```

```
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Zz(i,k));
    end
    fprintf(fid,'\n\n');
end

fprintf(fid,'Fault current\n\n');
fprintf(fid,'Magnitude    Angle(radians)\n');
fprintf(fid,'I1\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_1),angle(I_1));
fprintf(fid,'I2\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_2),angle(I_2));
fprintf(fid,'I0\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_0),angle(I_0));
fprintf(fid,'Printing of sequence components\n');
fprintf(fid,'Post Fault Voltages\n\n');
fprintf(fid,'Bus-number  Pos-seq-Magnitude  Pos-seq-Angle(radians)  Neg-seq-Magnitude  Neg-seq-Angle(radians)  Zero-seq-Magnitude  Zero-seq-Angle(radians)\n');
for i=1:length(V_1p)
    fprintf(fid,'%-10.3f   %-10.3f   %-10.3f   %-10.3f   %-10.3f   %-10.3f   %-10.3f\n',i,abs(V_1p(i)),V_1p_phase(i),abs(V_2p(i)),V_2p_phase(i),abs(V_0p(i)),angle(V_0p(i)));
end

fprintf(fid,'\n\n');
fclose('all');
```

```
clc

% --- Executes on button press in Double_line_to_ground_fault.
function llgfault_Callback(hObject, eventdata, handles)
nbuses= str2double(get(handles.Input_nb, 'String'));
nxfmrs = str2double(get(handles.Input_nt, 'String'));
nlines = str2double(get(handles.Input_nl, 'String'));
ngen= str2double(get(handles.Input_ng, 'String'));
fault_imp= str2double(get(handles.edit18, 'String'));
fault_bus= str2double(get(handles.edit19, 'String'));
Yp=Poistive_ybus(nbuses,ngen,nlines ,nxfmrs);
Yz=Zero_ybus(nbuses,ngen,nlines ,nxfmrs);
Zp=inv(Yp);
Zn=Zp;
check=diag(Yz);
k=find(check==0);
if length(k)==0
    Zz=inv(Yz);
end
if length(k)~=0
    Zz=pinv(Yz);
end
%preparation of pre-fault voltage vectors
vector1=ones(nbuses);
V_1=vector1(:,1);
vector0=zeros(nbuses);
V_2=vector0(:,1);
V_0=V_2;
```

```
%calculation of fault current
Zth=((Zz(fault_bus,fault_bus)+3*fault_imp)*j*Zn(fault_bus,fault_bus)*j/((Zz(fault_bus,fault_bus)+3*fault_imp+Zn(fault_bus,fault_bus))*j))+Zp(fault_bus,fault_bus)*j;
I_1=V_1(fault_bus)/Zth
I_2=-I_1*(Zz(fault_bus,fault_bus)+3*fault_imp)*j/((Zz(fault_bus,fault_bus)+3*fault_imp+Zn(fault_bus,fault_bus))*j)
I_0=-I_1*(Zn(fault_bus,fault_bus))*j/((Zz(fault_bus,fault_bus)+3*fault_imp+Zn(fault_bus,fault_bus))*j)
a=-0.5+(0.866025*1i);
i012=[I_0;I_1;I_2];
p=[1 1 1;1 a^2 a;1 a a^2];
iabc=p*i012;
ia=round(iabc(1),5);
ib=iabc(2);
ic=iabc(3);
set(handles.I1_edit, 'String', num2str(ia));
set(handles.I2_edit, 'String', num2str(ib));
set(handles.I0_edit, 'String', num2str(ic));
cla((handles.Fault_current_axes));
axes(handles.Fault_current_axes);
xd2=Zp(fault_bus,fault_bus)*j;
xd1=0.35j;
xd=0.5j;
z2=0.175j;
z0=0.30j;
zf=fault_imp*j;
td2=0.035;
```

```
td1=0.5;
Ta=0.17;
w=2*pi*50;
q=pi/2;
t = 0:0.0001:0.3;
xp=(((((1/xd2)-(1/xd1)).*exp(-t/td2))+(((1/xd1)-(1/xd)).*exp(-t/td1))+(1/xd)));
x1=(xp.^-1);
x20=Zn(fault_bus,fault_bus)*Zz(fault_bus,fault_bus)/(Zn(fault_bus,fault_bus)+Zz(fault_bus,fault_bus));
ia=t*0;
ib= 1./(Zn(fault_bus,fault_bus)+Zz(fault_bus,fault_bus)).*((Zn(fault_bus,fault_bus)./(x1+x20).*sin((w.*t)+(q-5/6*pi)))+((Zz(fault_bus,fault_bus)./(x1+x20).*sin((w*t)+(q-pi./2)))));
ic=1./(Zn(fault_bus,fault_bus)+Zz(fault_bus,fault_bus)).*((Zz(fault_bus,fault_bus)./(x1+x20).*sin((w.*t)+(q-5/6*pi)))+((Zn(fault_bus,fault_bus)./(x1+x20).*sin((w*t)+(q-pi./2)))));
x20=Zn(fault_bus,fault_bus)*Zz(fault_bus,fault_bus)/(Zn(fault_bus,fault_bus)+Zz(fault_bus,fault_bus));
hold on
plot(t,imag(ia),'r')
plot(t,imag(ib),'b')
plot(t,imag(ic),'g')
title('Fault Current vs Time')
xlabel('Time (s)')
ylabel('Fault Current (pu)')
legend('Ia','Ib','Ic');
%post fault voltage calculations
```

```
for i=1:nbuses
    V_1p(i)=V_1(i)-Zp(i,fault_bus)*I_1*j;
    V_2p(i)=V_2(i)-Zn(i,fault_bus)*I_2*j;
    V_0p(i)=V_0(i)-Zz(i,fault_bus)*I_0*j;
end
V_1p_mag=abs(V_1p);
V_1p_phase=angle(V_1p);
V_2p_mag=abs(V_2p);
V_2p_phase=angle(V_2p);
V_0p_mag=abs(V_0p);
V_0p_phase=angle(V_0p);
V012=[V_0p(fault_bus);V_1p(fault_bus);V_2p(fault_bus)];
p=[1 1 1;1 a^2 a;1 a a^2];
Vabc=p*V012;
set(handles.V1_edit, 'String', num2str(round(Vabc(1),5)))
set(handles.V2_edit, 'String', num2str(round(Vabc(2),5)))
set(handles.V0_edit, 'String', num2str(round(Vabc(3),5)))
Va=abs(Vabc(1))*cos(w*t+angle(Vabc(1)));
Vb=abs(Vabc(2))*cos(w*t+angle(Vabc(2)));
Vc=abs(Vabc(3))*cos(w*t+angle(Vabc(3)));
cla((handles.Fault_voltages_axes))
axes(handles.Fault_voltages_axes);
hold on
plot(t,Va,'r')
plot(t,Vb,'b')
plot(t,Vc,'g')
title('Post Fault Voltage vs Time')
xlabel('Time (s)')
```

```
ylabel('Fault Voltage (pu)')
legend('Va','Vb','Vc');


%data entry into output file
fid=fopen('output.txt','w');
fprintf(fid,'The positive sequence Ybus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Yp(i,k));
    end
    fprintf(fid,'\n\n');
end
fprintf(fid,'The positive sequence Zbus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Zp(i,k));
    end
    fprintf(fid,'\n\n');
end

fprintf(fid,'The zero sequence Ybus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Yz(i,k));
    end
    fprintf(fid,'\n\n');
end
```

```
fprintf(fid,'The zero sequence Zbus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Zz(i,k));
    end
    fprintf(fid,'\n\n');
end

fprintf(fid,'Fault current\n\n');
fprintf(fid,'Magnitude   Angle(radians)\n');
fprintf(fid,'I1\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_1),angle(I_1));
fprintf(fid,'I2\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_2),angle(I_2));
fprintf(fid,'I0\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_0),angle(I_0));
fprintf(fid,'Printing of sequence components\n');
fprintf(fid,'Post Fault Voltages\n\n');
fprintf(fid,'Bus-number  Pos-seq-Magnitude  Pos-seq-Angle(radians)  Neg-seq-Magnitude  Neg-seq-Angle(radians)  Zero-seq-Magnitude  Zero-seq-Angle(radians)\n');
for i=1:length(V_1p)
    fprintf(fid,'%-10.3f   %-10.3f   %-10.3f   %-10.3f   %-10.3f   %-10.3f   %-10.3f\n',i,abs(V_1p(i)),V_1p_phase(i),abs(V_2p(i)),V_2p_phase(i),abs(V_0p(i)),angle(V_0p(i)));
end


```

```
fprintf(fid,'\n\n');
fclose('all');
done=1;
clc

% --- Executes on button press in threephfault1.
function threephfault1_Callback(hObject, eventdata, handles)
nbuses= str2double(get(handles.Input_nb, 'String'));
nxfmrs = str2double(get(handles.Input_nt, 'String'));
nlines = str2double(get(handles.Input_nl, 'String'));
ngen= str2double(get(handles.Input_ng, 'String'));
fault_imp= str2double(get(handles.edit18, 'String'));
fault_bus= str2double(get(handles.edit19, 'String'));
Yp=Poistive_ybus(nbuses,ngen,nlines,nxfmrs);
Zbus=inv(Yp);
vector1=ones(nbuses);
V_1=vector1(:,1);
vector1=ones(nbuses);
V_1=vector1(:,1);
%calculation of fault current
I_1=V_1(fault_bus)/((Zbus(fault_bus,fault_bus)+fault_imp)*j);
I_0=0;
I_2=0;
a=-0.5+(0.866025*1i);
i012=[I_0;I_1;I_2];
p=[1 1 1;1 a^2 a;1 a a^2]
iabc=p*i012;
ia=iabc(1);
```

```
ib=iabc(2);
ic=iabc(3);
set(handles.I1_edit, 'String', num2str(ia))
set(handles.I2_edit, 'String', num2str(ib))
set(handles.I0_edit, 'String', num2str(ic))

cla((handles.Fault_current_axes))
axes(handles.Fault_current_axes);
xd2=(Zbus(fault_bus,fault_bus))*j
xd1=0.35j;
xd=0.5j;
z2=0.175j;
z0=0.30j;
zf=fault_imp*j;
td2=0.035;
td1=0.5;
Ta=0.17;
w=2*pi*50;
q=0;
t = 0:0.0001:0.3;
xp=(((((1/xd2)-(1/xd1)).*exp(-t/td2))+(((1/xd1)-(1/xd)).*exp(-t/td1))+(1/xd)));
x1=(xp.^-1);
ia=((1./x1).*sin(((w*t)+(q))));
ib=((1./x1).*sin(((w*t)+(q-2.0944))));
ic=((1./x1).*sin(((w*t)+(q+2.094400006763))));
hold on
plot(t,imag(ia),'r')
plot(t,imag(ib),'b')
plot(t,imag(ic),'g')
title('Fault Current vs Time')
xlabel('Time (s)')
ylabel('Fault Current (pu)')
legend('Ia','Ib','Ic');

%post fault voltage calculations
for i=1:length(V_1)
    V_1p(i)=V_1(i)-Zbus(i,fault_bus)*I_1*j;
end
V_1p_mag=abs(V_1p);
V_1p_phase=angle(V_1p);
% vf_2 = V_1p_mag.*exp(i*V_1p_phase);
V012=[0;V_1p(i);0];
p=[1 1 1;1 a^2 a;1 a a^2];
Vabc=p*V012;
set(handles.V1_edit, 'String', num2str(Vabc(1)))
set(handles.V2_edit, 'String', num2str(Vabc(2)))
set(handles.V0_edit, 'String', num2str(Vabc(3)))
Va=abs(Vabc(1))*cos(w*t+angle(Vabc(1)));
Vb=abs(Vabc(2))*cos(w*t+angle(Vabc(2)));
Vc=abs(Vabc(3))*cos(w*t+angle(Vabc(3)));
cla((handles.Fault_voltages_axes))
axes(handles.Fault_voltages_axes);
hold on
plot(t,Va,'r')
plot(t,Vb,'b')
plot(t,Vc,'g')
title('Fault Voltage vs Time')
xlabel('Time (s)')
ylabel('Post Fault Voltage (pu)')
legend('Va','Vb','Vc');
%
%data entry into output file
fid=fopen('output.txt','w');
fprintf(fid,'The Ybus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Yp(i,k));
    end
    fprintf(fid,'\n\n');
end
fprintf(fid,'The Zbus\n\n');
for i=1:nbuses
    for k=1:nbuses
        fprintf(fid,'%-10.3f\t',Zbus(i,k));
    end
    fprintf(fid,'\n\n');
end
fprintf(fid,'Fault current\n\n');
fprintf(fid,'Magnitude   Angle(radians)\n');
fprintf(fid,'%-10.3f %-10.3f\n\n',abs(I_1),angle(I_1));
fprintf(fid,'Post Fault Voltages\n\n');
fprintf(fid,'Bus-number  Magnitude  Angle(radians)\n');
for i=1:length(V_1p)
```

```
    fprintf(fid,'%-10.3f   %-10.3f   %-10.3f\n',i,V_1p_mag(i),V_1p_phase(i));
end
fprintf(fid,'\n\n');
fclose('all');
clc



% --- Executes on button press in Complete analysis file.
function pushbutton9_Callback(hObject, eventdata, handles)

winopen('output.txt')


% --- Executes during object creation, after setting all properties.
function edit28_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in Reset.
function pushbutton10_Callback(hObject, eventdata, handles)

```

```

set(findobj(handles.figure1,'style','edit'),'string','')
axes(handles.axes6)
matlabImage = imread('bc.jpg');
image(matlabImage)
axis off
axis image
axes(handles.Fault_current_axes)
cla
axes(handles.Fault_voltages_axes)
cla
set(handles.popupmenu5, 'Value', 1);
set(handles.popupmenu4, 'Value', 1);
clc

% --- Executes on button press in Data entry.
function pushbutton13_Callback(hObject, eventdata, handles)
nbuses= str2double(get(handles.Input_nb, 'String'))
nxfmrs = str2double(get(handles.Input_nt, 'String'))
nlines = str2double(get(handles.Input_nl, 'String'))
ngen= str2double(get(handles.Input_ng, 'String'))
clc
commandwindow
% entry of generator data
for i=1:ngen
    disp('data entry for generator');
    disp(i);
    gen_bus(i)=input('enter the fault_bus number at which the
generator is connected');
    pos_gen(i)=input('enter the positive/negative sequence impedance of the generator');
    zero_gen(i)=input('enter the zero sequence impedance of the generator');
    gen_ground(i)=input('if the generator is grounded enter the grounding impedance and if ungrounded enter 1000');
end
gen=[gen_bus' pos_gen' zero_gen' gen_ground'];
gen=gen';
fid=fopen('Generator_Data.txt','w');
fprintf(fid,'%6.2f %6.2f %6.2f %6.2f\n',gen);
fclose(fid);
%entry of line data
for i=1:nlines
    disp('Please enter data for line');
    disp(i);
    start_bus(i)=input('What is the starting bus number');
    end_bus(i)=input('What is the ending bus number');
    pos_line(i)=input('What is the positive/negative sequence impedance of the line');
    zero_line(i)=input('What is the zero sequence impedance of the line');
    realtap_line(i)=input('What is the real part of off nominal tap of the line');
    imagtap_line(i)=input('What is the imaginary part of off nominal tap of the line');
end
```

```
linedata=[start_bus' end_bus' pos_line' zero_line' realtap_line' imagtap_line'];
linedata=linedata';
fid=fopen('Line_Data.txt','w');
fprintf(fid,'%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f\n',linedata);
fclose(fid);
%entry of transformer data
for i=1:nxfmrs
    disp('Please enter data for transformer');
    disp(i)
    hv_bus(i)=input('What is the bus number of high voltage side bus number of the transformer');
    hv_code(i)=input('Please enter the connection code --> 0-star grounded 1-star ungrounded 2-delta');
    lv_bus(i)=input('What is the bus number of low voltage side bus of the transformer');
    lv_code(i)=input('Please enter the connection code --> 0-star grounded 1-star ungrounded 2-delta');
    pos_trans(i)=input('What is  the positive/negative sequence impedance of the transformer');
    zero_trans(i)=input('What is  the zero sequence impedance of the transformer');
end
transdata=[hv_bus' hv_code' lv_bus' lv_code' pos_trans' zero_trans'];
transdata=transdata';
fid=fopen('Transformer_Data.txt','w');
fprintf(fid,'%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f\n',transdata);
```

```
fclose(fid);


% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
v = get(handles.popupmenu4,'Value');
if v == 1

elseif v == 2
    winopen('Power System Analysis-per unit reactance diagram.mp4');
elseif v == 3
    winopen('Unbalanced Fault Analysis.mp4');
elseif v == 4
    winopen('Sequence Networks_- Transformer,Generator, Transmission line.mp4');
elseif v == 5
    winopen('Power System Analysis  (fault analysis)-1.mp4');

end


% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
v = get(handles.popupmenu5,'Value');
if v == 1

elseif v == 2
```

```
    winopen('Zbus1.pdf');
elseif v == 3
    winopen('Zbus2.pdf');
elseif v == 4
    winopen('Zbus3.pdf');
elseif v == 5
    winopen('Symmetrical_faults.pdf');
elseif v == 6
    winopen('Sequence Networks.pdf');
elseif v == 7
    winopen('Fault analysis using sequence network.pdf');
elseif v == 8
    winopen('Unbalanced fault analysis.pdf');
elseif v == 9
    winopen('Example of fault calculation.pdf');
elseif v == 10
    winopen('Example of fault calculation 2.pdf');
end


% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes during object creation, after setting all properties.
function Input_nb_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_nb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function Input_ng_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to Input_ng (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function Input_nl_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_nl (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end


% --- Executes during object creation, after setting all properties.
function Input_nt_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Input_nt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function I1_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to I1_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function I2_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to I2_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
```

```
function I0_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to I0_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function V1_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to V1_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function V2_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to V2_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object creation, after setting all properties.
function V0_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to V0_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all
```

```
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object deletion, before destroying properties.
function V0_edit_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to V0_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes during object deletion, before destroying properties.
function edit18_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes when figure1 is resized.
function figure1_SizeChangedFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

A.2 Formation of positive sequence impedance matrix

```
function Yp=Ypositive(nbuses,ngen,nlines,nxfmrs)


```

```
Yp=zeros(nbuses);
%setting up default 5 bus and 6 bus power system data
if nbuses==5
fid=fopen('5Line_Data.txt','r');
read= textread('5Line_Data.txt');
fclose(fid);
else if nbuses==6
fid=fopen('6Line_Data.txt','r');
read= textread('6Line_Data.txt');
fclose(fid);
    else
        fid=fopen('Line_Data.txt','r');
read= textread('Line_Data.txt');
fclose(fid);
    end
end

%From Line data
x=read(:,1); %starting bus
y=read(:,2); %ending bus
zp=read(:,3); %Positive seq imp
for i=1:length(x)
    xx=(1/(zp(i)));
    xy=(-1/(zp(i)));
    yx=(-1/(zp(i)));
    yy=(1/(zp(i)));
    Yp(x(i),x(i))=Yp(x(i),x(i))+xx;
    Yp(y(i),y(i))=Yp(y(i),y(i))+yy;
    Yp(x(i),y(i))=Yp(x(i),y(i))+xy;
    Yp(y(i),x(i))=Yp(y(i),x(i))+yx;
end

%From Generator
if nbuses==5
fid=fopen('5Generator_Data.txt','r');
read=textread('5Generator_Data.txt');
fclose(fid);
else if nbuses==6
        fid=fopen('6Generator_Data.txt','r');
read=textread('6Generator_Data.txt');
fclose(fid);

    else
        fid=fopen('Generator_Data.txt','r');
read=textread('Generator_Data.txt');
fclose(fid);
    end
end

x=read(:,1); %generator bus
y=read(:,2); %Positive seq imp
for i=1:length(x)
    Yp(x(i),x(i))=Yp(x(i),x(i))+(1/y(i));
end

%From Transformer
```

```
if nbuses==5
fid=fopen('5Transformer_Data.txt','r');
read=textread('5Transformer_Data.txt');
fclose(fid);
else if nbuses==6
        fid=fopen('6Transformer_Data.txt','r');
read=textread('6Transformer_Data.txt');
fclose(fid);

    else
        fid=fopen('Transformer_Data.txt','r');
read=textread('Transformer_Data.txt');
fclose(fid);

    end
end


x=read(:,1); %HV bus
y=read(:,3); %LV bus
z=read(:,5); %Positive seq imp
for i=1:length(x)
    Yp(x(i),x(i))=Yp(x(i),x(i))+1/z(i);
    Yp(y(i),y(i))=Yp(y(i),y(i))+1/z(i);
    Yp(x(i),y(i))=Yp(x(i),y(i))+(-1)/z(i);
    Yp(y(i),x(i))=Yp(y(i),x(i))+(-1)/z(i);
end
zp=inv(Yp);
```

A.3 Formation of zero sequence impedance matrix

```
function Yz=Yzero(nbuses,ngen,nlines,nxfmrs)

Yz=zeros(nbuses);
%From Line
if nbuses==5
%From Line
fid=fopen('5Line_Data.txt','r');
read= textread('5Line_Data.txt');
fclose(fid);
else if nbuses==6
fid=fopen('6Line_Data.txt','r');
read= textread('6Line_Data.txt');
fclose(fid);
    else
        fid=fopen('Line_Data.txt','r');
read= textread('Line_Data.txt');
fclose(fid);
    end
end


x=read(:,1);
y=read(:,2);
zp=read(:,4);
for i=1:length(x)
    xx=(1/(zp(i)));
    xy=(-1/(zp(i)));
```

```
        yx=(-1/(zp(i)));
        yy=(1/(zp(i)));
        Yz(x(i),x(i))=Yz(x(i),x(i))+xx;
        Yz(y(i),y(i))=Yz(y(i),y(i))+yy;
        Yz(x(i),y(i))=Yz(x(i),y(i))+xy;
        Yz(y(i),x(i))=Yz(y(i),x(i))+yx;
end

%From Transformer
if nbuses==5
fid=fopen('5Transformer_Data.txt','r');
read=textread('5Transformer_Data.txt');
fclose(fid);
else if nbuses==6
                fid=fopen('6Transformer_Data.txt','r');
read=textread('6Transformer_Data.txt');
fclose(fid);

        else
                fid=fopen('Transformer_Data.txt','r');
read=textread('Transformer_Data.txt');
fclose(fid);

        end
end


x=read(:,1);
```

```
y=read(:,3);
hvx=read(:,2);
lvx=read(:,4);
z=read(:,6);
for i=1:length(x)
    if hvx(i)==0 & lvx(i)==0 %represents the case of grounded star at both ends
        Yz(x(i),x(i))=Yz(x(i),x(i))+1/z(i);
        Yz(y(i),y(i))=Yz(y(i),y(i))+1/z(i);
        Yz(x(i),y(i))=Yz(x(i),y(i))+(−1)*1/z(i);
        Yz(y(i),x(i))=Yz(y(i),x(i))+(−1)*1/z(i);
    end

    if hvx(i)==0 & lvx(i)==1 %represents the case of grounded star and ungrounded star
        Yz(x(i),x(i))=Yz(x(i),x(i))+0;
        Yz(y(i),y(i))=Yz(y(i),y(i))+0;
        Yz(x(i),y(i))=Yz(x(i),y(i))+0;
        Yz(y(i),x(i))=Yz(y(i),x(i))+0;
    end

    if hvx(i)==0 & lvx(i)==2 %represents the case of grounded star and delta
        Yz(x(i),x(i))=Yz(x(i),x(i))+1/z(i);
        Yz(y(i),y(i))=Yz(y(i),y(i))+0;
        Yz(x(i),y(i))=Yz(x(i),y(i))+0;
        Yz(y(i),x(i))=Yz(y(i),x(i))+0;
    end

    if hvx(i)==1 & lvx(i)==0 %represents the case of ungrounded star and grounded star
      Yz(x(i),x(i))=Yz(x(i),x(i))+0;
      Yz(y(i),y(i))=Yz(y(i),y(i))+0;
      Yz(x(i),y(i))=Yz(x(i),y(i))+0;
      Yz(y(i),x(i))=Yz(y(i),x(i))+0;
    end

    if hvx(i)==1 & lvx(i)==1 %represents the case of ungrounded star at both ends
      Yz(x(i),x(i))=Yz(x(i),x(i))+0;
      Yz(y(i),y(i))=Yz(y(i),y(i))+0;
      Yz(x(i),y(i))=Yz(x(i),y(i))+0;
      Yz(y(i),x(i))=Yz(y(i),x(i))+0;
    end

    if hvx(i)==1 & lvx(i)==2 %represents the case of ungrounded star and delta
      Yz(x(i),x(i))=Yz(x(i),x(i))+0;
      Yz(y(i),y(i))=Yz(y(i),y(i))+0;
      Yz(x(i),y(i))=Yz(x(i),y(i))+0;
      Yz(y(i),x(i))=Yz(y(i),x(i))+0;
    end

    if hvx(i)==2 & lvx(i)==0 %represents the case of delta and grounded star
      Yz(x(i),x(i))=Yz(x(i),x(i))+0;
```

```
        Yz(y(i),y(i))=Yz(y(i),y(i))+0;
        Yz(x(i),y(i))=Yz(x(i),y(i))+0;
        Yz(y(i),x(i))=Yz(y(i),x(i))+1/z(i);
    end

    if hvx(i)==2 & lvx(i)==1 %represents the case of delta and ungrounded star
        Yz(x(i),x(i))=Yz(x(i),x(i))+0;
        Yz(y(i),y(i))=Yz(y(i),y(i))+0;
        Yz(x(i),y(i))=Yz(x(i),y(i))+0;
        Yz(y(i),x(i))=Yz(y(i),x(i))+0;
    end

    if hvx(i)==2 & lvx(i)==2 %represents the case of delta at both ends
        Yz(x(i),x(i))=Yz(x(i),x(i))+0;
        Yz(y(i),y(i))=Yz(y(i),y(i))+0;
        Yz(x(i),y(i))=Yz(x(i),y(i))+0;
        Yz(y(i),x(i))=Yz(y(i),x(i))+0;
    end
end

%From Generator
if nbuses==5
fid=fopen('5Generator_Data.txt','r');
read=textread('5Generator_Data.txt');
fclose(fid);
else if nbuses==6
```

```
        fid=fopen('6Generator_Data.txt','r');
read=textread('6Generator_Data.txt');
fclose(fid);

    else
        fid=fopen('Generator_Data.txt','r');
read=textread('Generator_Data.txt');
fclose(fid);
    end
end


x=read(:,1);
z=read(:,3);
gnd=read(:,4);
for i=1:length(x)
    if gnd(i)==1000
        Yz(x(i),x(i))=Yz(x(i),x(i))+0;
    end
    if gnd(i)~=1000
        Yz(x(i),x(i))=Yz(x(i),x(i))+1/(z(i)+3*gnd(i));
    end
end
```

Bibliography

[1] T. Kai, N. Takeuchi, T. Funabashi, H. Sasaki. A Simplified Fault Currents Analysis Method considering Transient of Synchronous Machine, *IEEE Transactions on Energy-Conversion*. 12(3), 225-231, 1997.

[2] J Duncan Glover, et al.*Power System Analysis and Design*. Fifth, Cengage Learning, 2012.

[3] Jun Zhu. *Analysis Of Transmission System Faults the Phase Domain*, Texas A&M University. Master Thesis, 2004.

[4] Santamaria, Jorge. *"Analysis of Power Systems Under Fault Conditions"* , Core.Ac.Uk, 2011, doi:oai:scholarworks.calstate.edu:10211.9/1369.

[5] NPTEL: Computer Aided power System analysis,
`https://nptel.ac.in/courses/108107028/1#`

[6] Koç, Savas, and Zafer Aydoğmus. *"A MATLAB/GUI Based Fault Simulation Tool for Power System Education." Mathematical and Computational Applications*,vol. 14, no. 3, Dec. 2009, pp. 207–17, doi:10.3390/mca14030207. 12(3), 225-231, 1997.

[7] Ghimire, Sushma. *Analysis of Fault Location Methods on Transmission Lines.*2014, scholarworks.uno.edu/cgi/viewcontent.cgi article=2842&context=td.

[8] Circuit Globe: Power Systems Archives,
`circuitglobe.com/category/electrical-terms/power-systems#`