TRANSIENT STABILITY ANALYSIS OF MULTI-MACHINE POWER SYSTEM USING CRITICAL MACHINE ENERGY FUNCTIONS AND FIRST SWING STABILITY ANALYSIS OF SVC

A dissertation submitted in partial fulfilment of the requirement for the degree of

MASTER OF TECHNOLOGY

IN

POWER SYSTEM

by

BETSY SEBASTIAN

(Roll No: 2K11/PSY/04)

Under the guidance of

Dr. RACHANA GARG & Mr. S T NAGARAJAN

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
DELHI-110042

JUNE 2013

Delhi Technological University

Shahbad Daulatpur, Delhi-110042

Department of Electrical Engineering

CERTIFICATE

This is to certify that this dissertation entitled

TRANSIENT STABILITY ANALYSIS OF MULTI-MACHINE POWER SYSTEM USING CRITICAL MACHINE ENERGY FUNCTIONS AND FIRST SWING STABILITY ANALYSIS OF SVC

is an authentic report of the project done by

BETSY SEBASTIAN

in the partial fulfilment of the requirement for the award of the degree of Master of Technology in Power systems by the Delhi Technological University during the year 2012-13.

Project Guide:	**Head of the Department**
Dr. RACHANA GARG Associate Professor, Dept. of Electrical Engg DTU, Delhi	Prof. Madhusudan Singh Head of the Dept, Dept. of Electrical Engg DTU, Delhi

Co- Guide:

Mr. S. T. Nagrajan
Associate Professor
Dept. of Electrical Engg
DTU, Delhi

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude to all those who have been instrumental in the successful completion of this project.

Dr. Rachana Garg, Associate Professor, Dept. of Electrical Engineering, Delhi Technological University, my project guide, has guided me throughout this project. I sincerely thank her for her wholehearted support and guidance.

Dr. S.T.Nagarajan, Assistant Professor, Dept. of Electrical Engineering, Delhi Technological University, my project Co-guide has always provided me necessary suggestions and directions. I express my gratitude to him.

I would like to express my sincere thanks to Dr.Suman Bhowmick, Associate Professor, Dept. of Electrical Engineering, Delhi Technological University for his help and guidance. I would like to thank Prof. Madhusoodan Singh, HOD, Dept. of Electrical Engineering, DTU, for his help and cooperation.

I am grateful to my friend and research scholar Mr. Mohammed Iqbal for his suggestions and help. With great pleasure and respect, I express my utmost gratitude to my beloved parents Mr. Thomas Sebastian and Mrs. Liza A Chacko, for their constant encouragement and support. To all the named and many unnamed, my sincere thanks. Surely it is Almighty’s grace to get things done fruitfully.

ABSTRACT

Transient stability analysis is an indispensible part of utility system planning. The industry standard for transient stability performance varies from region to region, but it requires the ability of the system to withstand severe disturbances such as faults or switching of lines. Under this event, system linearization is not possible and the Differential Algebraic Equation model of the power system need to be solved. Even though several direct stability methods have been developed since the inception and identification of stability problem in multi-machine system, numerical method of solving the stability problem is still carried out exploiting the features of digital computation. The result of this method is still being used as the benchmark for determining the precision of direct methods of stability analysis. Keeping in hand with the convention, Numerical method of solving stability problem has been carried out in this project.

Stability analysis of multi-machine power system using energy functions of individual machine and group of machines has been the main subject matter of this research study. The simulations carried out reaffirm the concept of critical machines in stability analysis. These are the machines which contribute to instability, and their transient energy separate them from the rest of the system. Critical clearing time, an important parameter in design of circuit breakers and various other protection equipments has been determined using energy functions of individual machines, and critical group of machines. The values obtained using critical machine energy function are found to be more accurate.

Power systems worldwide are undergoing transformation into complex entities with the introduction of various fast controllers such as FACTS devices. Stability analysis incorporating FACTS devices is the need of the hour. Some preliminary effort in this regard has been carried out in this project. Initially the analysis have been carried out on SMIB system and later on expanded to multi-machine power system by utilising the advantages of MATLAB and DIGSILENT POWER FACTORY software.

TABLE OF CONTENTS

CONTENTS **PAGE NO**

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

SYMBOLS	QUANTITY
δ	Generator Rotor Angle
ω	Generator Rotor Speed
ω_B	Base Angular Frequency
H	Generator Inertia Constant
P_m	Mechanical Input Power to Generator
E'_q	Generator Quadrature Axis Voltage
E'_d	Generator Direct Axis Voltage
i_q	Generator Quadrature Axis Current
i_d	Generator Direct Axis Current
V_q	Generator Quadrature Axis Voltage
V_d	Generator Direct Axis Voltage
x_q	Generator Quadrature Axis Reactance
x_d	Generator Direct Axis Reactance
x'_q	Generator Quadrature Axis Transient Reactance
x'_d	Generator Direct Axis Transient Reactance
R_a	Armature Resistance
T'_{qo}	Quadrature Axis Open-Circuit Transient Time Constant of Generator
T'_{do}	Direct Axis Open-Circuit Transient Time Constant of Generator
E_{fd}	Generator Field Voltage
E_{fd}^{min}	Generator Field Voltage Lower Limit
E_{fd}^{max}	Generator Field Voltage Upper Limit
K_A	Automatic Voltage Regulator (AVR) Gain
T_A	Time constant of AVR

θ	Generator Rotor Angle in COI Reference
$\check{\gamma}$	Generator Rotor Speed in COI Reference
V_{ref}	Bus Reference Voltage Magnitude
V	Bus Voltage Magnitude
φ	Bus Voltage Angle
P_L	Real Power Load
Q_L	Reactive Power Load
P_G	Generator Real Power
Q_G	Generator Reactive Power
$Y_{ik}\angle\alpha_{ik}$	Element of Bus Admittance Matrix
a	Coefficient of Constant Impedance Real Load
b	Coefficient of Constant Impedance Reactive Load

CHAPTER 1

INTRODUCTION

As the population statistics paramount, with figures reaching 7 billion recently, the world is concerned over providing the masses with resources. Electricity was a luxury in the beginning of this century, but now it is considered as a basic necessity. Since its inception, the power industry has operated with clear demarcations between its generation, transmission, and distribution subsystems. But in last two decades the electric grid has undergone some drastic changes. Open market and power deregulation are the key characteristics of modern power systems. New interconnections, tie-lines, the use of new technologies, financial and regulatory constrains have increased its complexity. The streamlined power system encourages a vigorous competition among utilities. Industrialisation and consequently the rising standard of living have increased the demand for reliable, economic, and quality electric energy, which force the power systems to operate closer to their maximum load-carrying capacity posing a threat to its stability.

With deregulation of power supply utilities, the power networks are being viewed as highways for transmitting electric power from wherever it is available to places where required, depending on the pricing that varies with time of the day. Power system dynamics has an important demeanour on the satisfactory system operation. It is influenced by the dynamics of the system components such as generators, transmission lines, loads and other control equipment (HVDC and SVC controllers). The dynamic behaviour of power systems can be quite complex and a good understanding is essential for proper system planning and secure operation [1].

Power system stability has been the subject matter of detailed analysis and research since time immemorial. The methods for analysing power system stability problems have been influenced by the development of computational tools, stability theories, and power system control technologies [2]. In broad sense, developments in dynamic security monitoring and control of electric power systems are far behind static counterparts and still offer an excellent opportunity for extensive research and investigation.

1.1 INTRODUCTION TO THE PROBLEM

Transient stability analysis is an indispensible part of utility system planning and is routinely performed there. The industry standard for transient stability performance varies from region to region, but it requires the ability of the system to withstand severe disturbances. Transient stability analysis programs are being used by power system planning and operating engineers to predict the response of the system to various disturbances. In these simulations, the behaviour of a present or proposed power system is evaluated to determine its stability or its operating limits, or perhaps, to determine the need for additional facilities. Important conclusions and decisions are made based on the results of stability studies. Hence it is important to ensure that the results of stability studies are as timely and accurate as possible [3]. It has been in practice to study transient stability by numerical simulation, in utility planning. But the slow and cumbersome nature of this method precludes its use for online security analysis.

This necessitated the development of direct methods of transient stability analysis. Although direct stability methods have been in use since 1960’s, methods featuring energy functions have been the more preferred ones. These functions describe the system transient energy causing the generators to depart from the initial equilibrium state, and the power network’s ability to absorb this energy so that the synchronous machine may reach to a new post-disturbance equilibrium state [4].

In this project transient stability analysis using individual machine energy functions have been carried out, which was later modified to that of critical machine energy functions. The results obtained have been compared with that of time domain simulation and are found to be within acceptable accuracy limit. FACTS devices are becoming an essential part of modern power system, owing to their state of the art technology, capabilities and advantages. Hence, transient stability analysis considering their effect is also the need of the hour. In later part of project preliminary stability analysis considering a shunt FACTS device – SVC has also been carried out.

1.2 ORGANISATION OF THE DISSERTATION

A brief outline of the project is as follows. Chapter 2 deals with the basic concepts and terminologies of stability. Chapter 3 considers the mathematical modelling of power system components including synchronous generators along with its AVR, transmission lines, loads etc. The time domain simulation of the model obtained, was carried out using numerical techniques and the results have been presented along with some of the case study runs. A literature review of the various direct stability methods have been featured in Chapter 4. It also introduces the basic concepts of direct stability method using Transient energy function, and determination of critical clearing time using it. Chapter 5 briefs the direct stability method using critical machine energy function and determination of critical clearing time using it. The results obtained have been tabulated and also depicted graphically. Chapter 6 deals with the stability analysis of power system considering a shunt FACTS device – SVC. The results and conclusions have been discussed in Chapter 7. A comparison of all three stability analyses carried out in this project have been presented in this section. It also throws some light on the future prospective of the work.

CHAPTER 2

A REVIEW ON POWER SYSTEM STABILITY

The first electric power system was a dc system built by Edison in 1882. The subsequent power systems that were constructed in the late 19th century were all dc systems. However despite the initial popularity of dc systems by the turn of the 20th century ac systems started to outnumber them. The ac systems were thought to be superior as ac machines were cheaper than their dc counterparts and more importantly ac voltages are easily transformable from one level to other using transformers. The early stability problems of ac systems were experienced in 1920 when insufficient damping caused spontaneous oscillations or hunting. These problems were solved using generator damper winding and the use of turbine-type prime movers.

2.1 BASIC CONCEPTS AND DEFINITIONS

Power system stability is broadly defined as the property of a power system that enables it to remain in a state of operating equilibrium under normal operating conditions and to regain an acceptable state of equilibrium after being subjected to a disturbance [5]. A power system is said to be in steady state operating condition if all the measured (or calculated) physical quantities describing the operating condition of the system can be considered constant for the purpose of analysis [6]. While operating in steady state if a sudden change or a sequence of changes occurs in one or more parameters, or operating quantities of the system, the system has said to undergone a disturbance from its steady- state operating condition.

Power system stability is a single problem; but it is unfeasible to study it as it is. Instability of a power system can take different forms and can be influenced by a wide range of factors [7]. Classification of stability problems greatly ease its analysis, identification of essential factors contributing to instability, and formation of methods of improving stability. The classification is done based on the following considerations [5]:

i. The physical nature of resulting instability
ii. The size of disturbance considered
iii. The devices and time span that must be taken into consideration in order to determine stability
iv. The most appropriate method of calculation and prediction of stability

Following section gives a picture of the power system stability problem, identifying its classes and subclasses. As a practical necessity, the classification has been based on number of diverse considerations, making it difficult to select clearly distinct categories and to provide definitions that are rigorous and yet convenient for practical use. The stability is broadly classified as –

- Rotor angle stability
- Voltage stability

2.2 ROTOR ANGLE STABILITY

Rotor angle stability is the ability of interconnected synchronous machines of a power system to remain in synchronism. As primarily synchronous generators are used for generating power in grid, stability is generally implied by the ability of the synchronous generators to remain in ’synchronism’ or ’in step’. On the other hand, if the synchronous generators loose synchronism after a disturbance, then the system is called unstable. Instability can also be encountered without loss of synchronism [5].

The basic concept of ’synchronism’ can be explained in terms of mismatch between mechanical and electrical power. In the normal equilibrium condition, all the synchronous generators run at a constant speed and the difference between the rotor angles of any two generators is constant. Under any disturbance, the speed of the machines will deviate from the steady state values due to mismatch between mechanical and electrical powers (torque) and therefore, the difference of the rotor angles would also change. If these rotor angle deviations (between any pair of generators) attain steady state values (not necessarily the same as in the pre-disturbance condition) after some finite time, then the synchronous generators are said to be in ’synchronism’. On the other hand, if the rotor angle deviations keep on increasing indefinitely, then the machines are considered to have lost ’synchronism’. Under this ’out of step’ condition, the output power, voltage etc. of the generator continuously drift away from the corresponding pre-disturbance values until the protection system trips the machine. Any disturbance small or large can affect the synchronous operation.
The disturbances can be – small and large.

2.2.1 Steady-State or Small Signal Stability

It is the ability of the power system to maintain synchronism under small disturbances. If a power system is operating in a steady-state condition and it undergoes change which can be analysed by linearized versions of its dynamic and algebraic equations, a small disturbance is said to have occurred. For example, a change in the gain of AVR in the excitation system of large generating unit, small variations in load, generation etc. Power system small signal stability is affected by several factors. A few amongst them are -

- Initial operating conditions
- Strength of electrical connections among components in the power system
- Characteristics of various control devices

Under small disturbances change in the electrical torque of a synchronous generator can be resolved into two components, namely, **a) synchronizing torque (*Ts*)** - which is proportional to the change in the rotor angle and **b) damping torque (*Td*),** which is proportional to the change in the speed of the machine. For a stable case, both *Ts* and *Td* are positive Fig.2.1.a. Depending on the amounts of synchronizing and damping torques, small signal stability is again categorised as – i) steady increase in rotor angle due to lack of sufficient synchronising torque, (here *Td* is positive, *Ts* is negative), Fig.2.1.b ii)Rotor oscillations of increasing amplitude due to lack of sufficient damping torque (here *Td* is negative, *Ts* is positive)Fig.2.1.c .

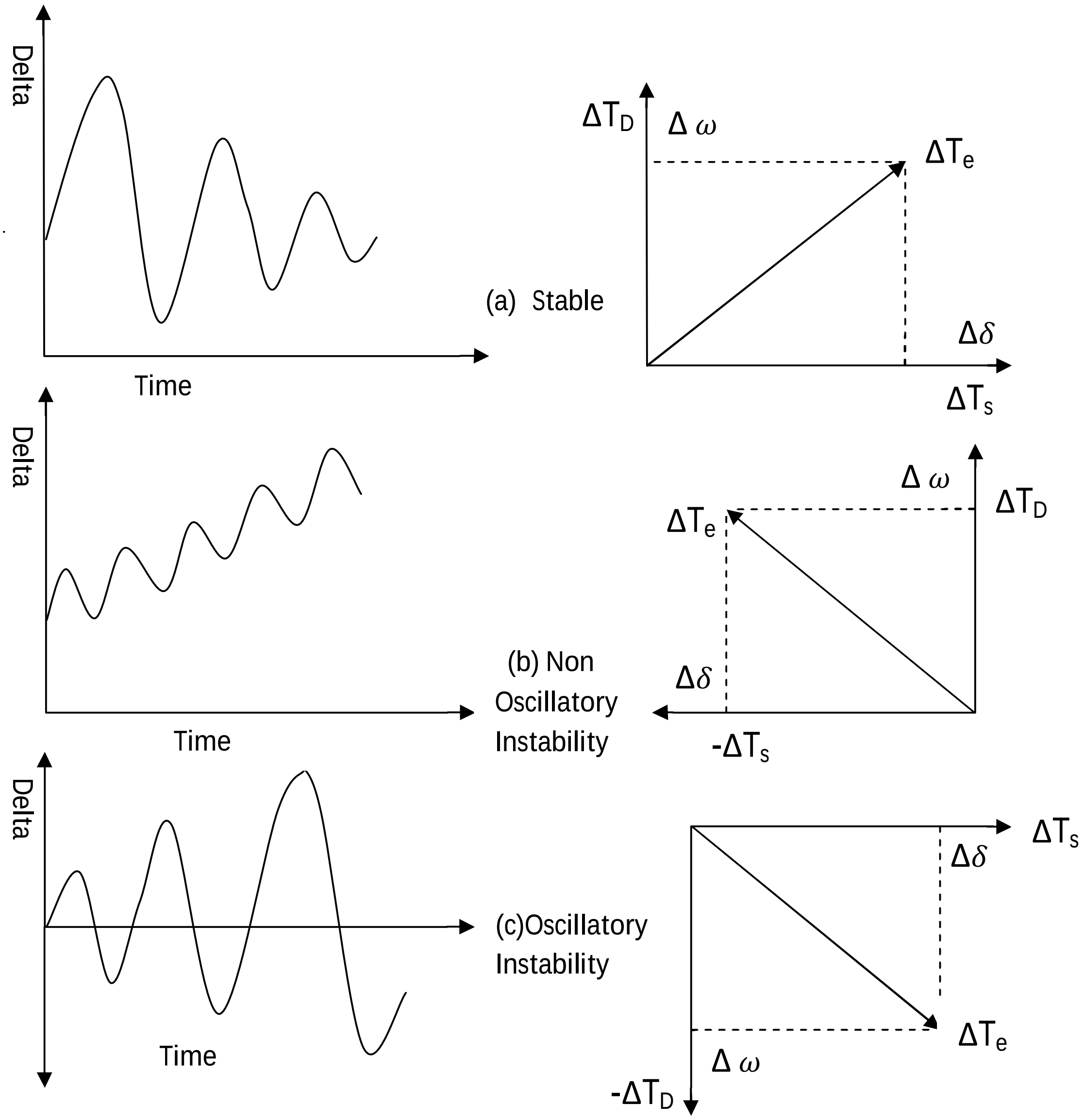


Fig 2.1 Different cases of small signal stability and the effects of synchronising and damping torques [5]

Small signal stability studies are usually less extensive in scope than transient stability studies. But, as small disturbances are inevitable in a power system operation, any power system that is unstable in terms of small signal stability cannot operate in practice [8].

2.2.2 Transient Stability

Transient stability is the ability of power system to maintain synchronism when subject to a severe transient disturbance. In this case, linearization of system equations is not possible. Examples of large disturbances are short circuits of different types, tripping or committing of large capacity generator, load or important transmission facility etc. Stability depends on both the initial operating state of the system and the severity of the disturbance. Usually, the system is altered so that the post-disturbance steady-state operation differs from that prior to the disturbance. Fig.2.2 shows various cases of stable and unstable behaviour of the generator. In case 1, under the influence of the fault, the generator rotor angle increases to a maximum, subsequently decreases and settles to a steady state value following oscillations with decreasing magnitude. In case 2, the rotor angle decreases after attaining a maximum value. However, subsequently, it undergoes oscillations with increasing amplitude. This type of instability is not caused by the lack of synchronizing torque; rather it occurs due to lack of sufficient damping torque in the post fault system condition. In case 3, the rotor angle monotonically keeps on increasing due to insufficient synchronizing torque till the protective relay trips it. This type of instability, in which the rotor angle never decreases, is termed as 'first swing instability'.

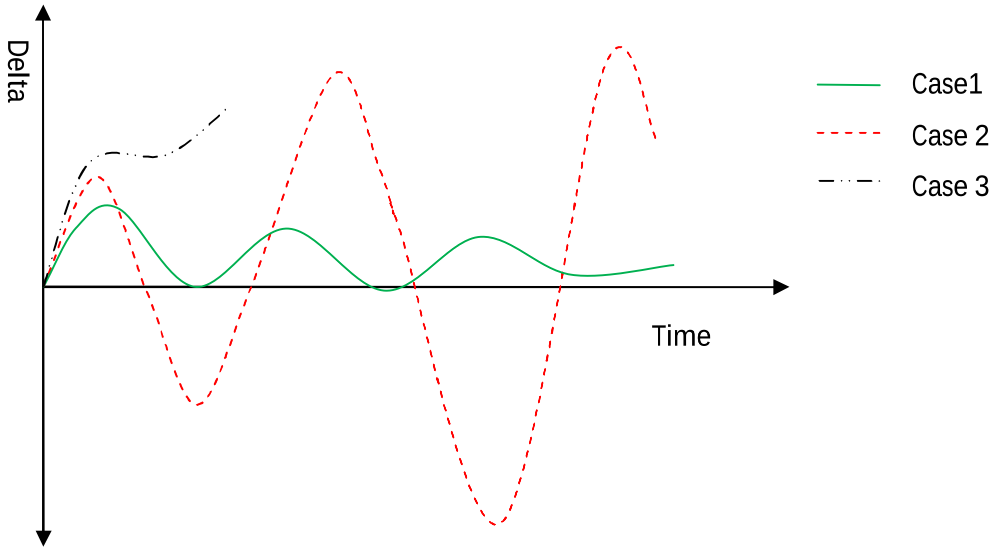


Fig 2.2 Rotor angle response to transient disturbances [5]

The term dynamic stability has also been widely used in literature as a class of rotor angle stability. However, it has been used by different authors to denote different aspects of the phenomenon. But owing to the confusion created by it, both CIGREE and IEEE have recommended that it not be used [5,9,10].

Transient stability analysis can be short-term (within 10s), mid-term (10s to several minutes) or long-term (several minutes to tens of minutes).

Short-term simulation considers rapidly changing electrical variables (including swinging among generators). Here the boiler and load frequency controller models are neglected [1]. Mid-term stability study focuses on synchronizing power oscillations between machines, including the effects of some of the slower phenomena, and possibly large voltage or frequency excursions [5]. Long-term stability analysis focuses on the slow phenomenon with long duration that occur after a large disturbance has happened, and the significant mismatch between active/reactive power generation and consumption. The phenomena of concern include: boiler dynamics, water gate and water-pipe dynamics of hydraulic turbines, control and protection of power plant and transmission system etc [8].

2.3 VOLTAGE STABILITY

Voltage stability is the ability of a power system to maintain steady acceptable voltages at all buses in the system under normal operating conditions and after being subjected to a disturbance [5]. Voltage instability occurs when following a disturbance change in system conditions causes a progressive and uncontrollable drop in voltage. The main factor causing instability is the inability of the power system to meet the demand for reactive power. The heart of problem is usually the voltage drop that occurs when active power and reactive power flow through inducting reactance associated with the transmission network [5]. For the purpose of analysis, voltage stability is classified into the following subclasses:

2.3.1 Large-Disturbance Voltage Stability-

It is a system's ability to control voltages following large disturbances such as system faults, loss of generation or circuit contingencies. This ability is determined by the system load characteristics and interaction of both continuous and discrete controls and protections [5].

2.3.2 Small Disturbance Voltage Stability-

It is a system's ability to control voltages following small perturbations such as incremental changes in system load. This form of stability is determined by the characteristics of load, continuous controls and discrete controls at a given instant of time [5].

2.4 NEED FOR STABILITY ANALYSIS

A major activity in utility system planning and operations is to test system transient stability relative to disturbances. Transient stability analysis programs are being used by power system planning and operation engineers to predict the response of the system to various disturbances. In these simulations, the behaviour of a present or proposed power system is evaluated to determine its stability or its operating limits, or perhaps, to determine the need for additional facilities. Important conclusions and decisions are made based on the results of stability studies. It is therefore important to ensure that the results of stability studies are as timely and accurate as possible [3]. Transient stability analysis also helps in initiating emergency controls like generator tripping.

CHAPTER 3

MODELLING AND TIME DOMAIN SIMULATION OF THE SYSTEM

Establishment of a mathematical model, describing the physical processes of a power system, is the foundation for the analysis and investigation of various power system problems. Correct and accurate computation for power system analysis requires a correct and accurate mathematical model of the power system. While developing model it is kept in mind that the mathematical model and associated solution method, complies with both computing accuracy and computing speed. This section considers the mathematical modelling of the various parts of a multi-machine power system consisting of transmission lines, loads and synchronous generators along with the excitation system. The power system under consideration has n buses and m generators.

In practical transient stability assessments, the system is judged as stable if it remains stable during the first few swings and the critical clearing time is determined on this basis. In the stability analysis carried out here damping is neglected because here only first swing transient stability is of importance. When synchronism is maintained in the first swing, it is assumed that damping will reduce the subsequent swings.

3.1 MATHEMATICAL MODELLING OF THE SYSTEM

3.1.1 Generator Representation

The synchronous machine is represented by model 1.1 (Field circuit with one equivalent damper on q axis). The dynamic model considered is a two axis machine model with saturation and network transients being neglected. The following assumptions were used in developing the model :

- The mmf in the airgap is distributed sinusoidally and harmonics are neglected.
- Saliency is restricted to the rotor. The effect of slots in the stator is neglected.
- Magnetic saturation and hysteresis are ignored.

The differential equations governing the i^{th} machine are given by (1) as,

$$\left.\begin{array}{c} \frac{d\delta_i}{dt} = \widetilde{\omega_i} \\[2ex] \frac{2 * H_i}{w_b}\frac{d\widetilde{\omega_i}}{dt} = P_{mi} - P_{ei} \\[2ex] T'_{doi} * \frac{dE'_{qi}}{dt} = -E'_{qi} + (x_{di} - x'_{di})i_{di} + E_{fdi} \\[2ex] T'_{qoi} * \frac{dE'_{di}}{dt} = -E'_{di} - (x_{qi} - x'_{qi})i_{qi} \end{array}\right\} \qquad (1)$$

Where the electrical power is given by (2) as ,

$$P_{ei} = E'_{qi} * i_{qi} + E'_{di} * i_{di} + (x'_{di} - x'_{qi})i_{qi} * i_{di} \qquad (2)$$

The stator algebraic equations can be written in different forms, so as to express the currents i_d and i_q in terms of network and state variables. In polar form they can be written as in (3),

$$\left.\begin{aligned} V_{di} &= E'_{di} - R_{ai} * i_{di} - x'_{qi} * i_{qi} \\ V_{qi} &= E'_{qi} - R_{ai} * i_{qi} + x'_{di} * i_{di} \end{aligned}\right\} \quad (3)$$

Where,

$$\left.\begin{aligned} V_{qi} &= V_i * sin(\varphi_i - \delta_i) \\ V_{di} &= V_i * \cos(\varphi_i - \delta_i) \end{aligned}\right\} \quad (4)$$

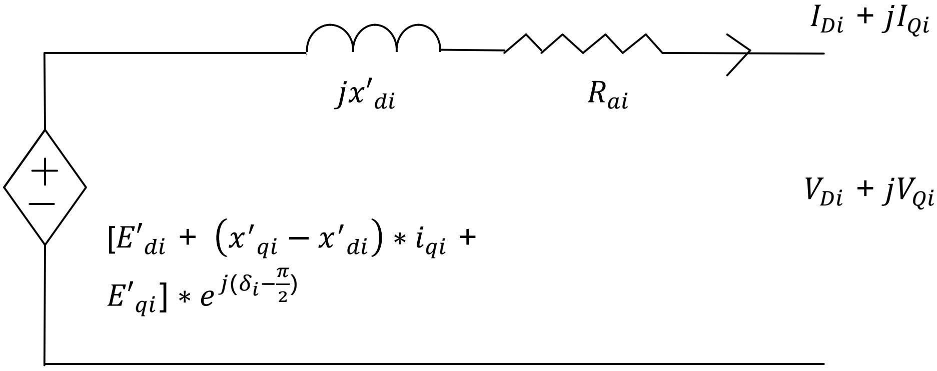

$I_{Di} + jI_{Qi}$
jx'_{di}
R_{ai}
$V_{Di} + jV_{Qi}$
$[E'_{di} + (x'_{qi} - x'_{di}) * i_{qi} + E'_{qi}] * e^{j(\delta_i - \frac{\pi}{2})}$

Fig 3.1 Synchronous machine two-axis model dynamic circuit [7]

3.1.2 Excitation System Representation

All the generators are assumed to be provided with static exciter. The excitation system considered is simple with one gain and one time constant. The behaviour of static exciter is defined by (5) :

$$T_{Ai} * \frac{dE_{xi}}{dt} = -E_{xi} + K_{Ai} * V_{refi} - K_{Ai} * V_i \quad (5)$$

The limits of Windup limiter is :

$$\left.\begin{aligned} &E_{fdi}^{min} \leq E_{xi} \leq E_{fdi}^{max}, \text{ then } E_{fdi} = E_{xi}, \\ &E_{xi} \geq E_{fdi}^{max} \quad \text{then } E_{fdi} = E_{fdi}^{max} \\ &E_{xi} \leq E_{fdi}^{min} \quad \text{then } E_{fdi} = E_{fdi}^{min} \end{aligned}\right\} \quad (6)$$

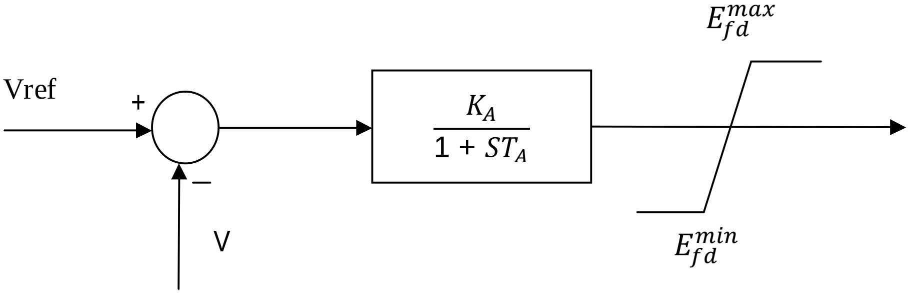


Fig 3.2 Excitation System

3.1.3 Network Representation

The network equations can be expressed either in power-balance or current-balance form. The network equations in the power-balance form for the generator bus *i= 1,2...,m* are:

$$\left.\begin{aligned} P_{Gi} - P_{Li}(V_i) - P_i &= 0 \\ Q_{Gi} - Q_{Li}(V_i) - Q_i &= 0 \end{aligned}\right\} \quad (7)$$

The network equations in the power-balance form for the non-generator bus *i=m+1,...,n* are:

$$\left.\begin{aligned} P_{Li}(V_i) + P_i &= 0 \\ Q_{Li}(V_i) + Q_i &= 0 \end{aligned}\right\} \quad (8)$$

Where,

$$\left.\begin{aligned} P_{Gi} &= V_i i_{qi} \cos(\varphi_i - \delta_i) + V_i i_{di} \sin(\varphi_i - \delta_i) \\ Q_{Gi} &= V_i i_{qi} \sin(\varphi_i - \delta_i) - V_i i_{di} \cos(\varphi_i - \delta_i) \end{aligned}\right\} \quad (9)$$

$$\left.\begin{aligned} P_i &= \sum_{k=1}^{n} V_i V_k Y_{ik} \cos(\varphi_k - \varphi_i + \alpha_{ik}) \\ Q_i &= -\textstyle\sum_{k=1}^{n} V_i V_k Y_{ik} \sin(\varphi_k - \varphi_i + \alpha_{ik}) \end{aligned}\right\} \quad (10)$$

3.1.4 Load Representation

The loads are modelled as constant impedances for simplicity. Hence the real power and reactive power loads at each bus are given by (11),

$$\left.\begin{aligned} P_{Li}(V_i) &= a_i * V_i^2 \\ Q_{Li}(V_i) &= b_i * V_i^2 \end{aligned}\right\} \quad (11)$$

where a_i and b_i are constants obtained from the steady state load flow solution. All the 8 equations together form DAE (Differential and Algebraic Equations).

3.2 DAE MODEL

3.2.1 Differential Equations

$$\left.\begin{aligned} &\frac{d\delta_i}{dt} = \widetilde{\omega_i} \\ &\frac{2 * H_i}{w_b}\frac{d\widetilde{\omega_i}}{dt} = P_{mi} - P_{ei} \\ &T'_{doi} * \frac{dE'_{qi}}{dt} = -E'_{qi} + (x_{di} - x'_{di})i_{di} + E_{fdi} \\ &T'_{qoi} * \frac{dE'_{di}}{dt} = -E'_{di} - (x_{qi} - x'_{qi})i_{qi} \\ &T_{Ai} * \frac{dE_{xi}}{dt} = -E_{xi} + K_{Ai} * V_{refi} - K_{Ai} * V_i \\ &\qquad\qquad\qquad\qquad i = 1,2.....m \end{aligned}\right\} \quad (12)$$

3.2.2 Algebraic Equation

a. Stator Algebraic Equations

$$\left.\begin{aligned} &E'_{di} - V_i * \cos(\varphi_i - \delta_i) - R_{ai} * i_{di} - x'_{qi} * i_{qi} = 0 \\ &E'_{qi} - V_i * sin(\varphi_i - \delta_i) - R_{ai} * i_{qi} + x'_{di} * i_{di} = 0 \\ &\qquad\qquad\qquad\qquad i = 1,2.....m \end{aligned}\right\} \quad (13)$$

b. Network Algebraic Equation

$$\left.\begin{aligned} &V_i i_{qi}\cos(\varphi_i-\delta_i)+\ V_i i_{di}\sin(\varphi_i-\delta_i\ )-a_i*V_i^2-\sum_{k=1}^{n}V_iV_kY_{ik}\cos(\varphi_k-\varphi_i+\alpha_{ik})=0\\ &V_i i_{qi}\sin(\varphi_i-\delta_i)-V_i i_{di}\cos(\varphi_i-\delta_i)-b_i*V_i^2+\sum_{k=1}^{n}V_iV_kY_{ik}\sin(\varphi_k-\varphi_i+\alpha_{ik})=0 \end{aligned}\right\} \quad (14)$$

i = 1,2..........m

$$\left.\begin{aligned} &a_i*V_i^2+\textstyle\sum_{k=1}^{n}V_iV_kY_{ik}\cos\left(\varphi_k-\varphi_i+\alpha_{ik}\right)=0\\ &b_i*V_i^2-\textstyle\sum_{k=1}^{n}V_iV_kY_{ik}\sin\left(\varphi_k-\varphi_i+\alpha_{ik}\right)=0 \end{aligned}\right\} \quad (15)$$

i=m+1,m+2.....n

The above equations (12), (13), (14) & (15) form a set of non-linear differential equations (DAE), thus we have:
differential equations for each machine i.e.

i. 5m differential equations
ii. 2 stator algebraic equation for each machine i.e. 2m algebraic equations
iii. 2 network algebraic equations at each bus i.e. 2n algebraic equations

Let 5m+2m+2n equations have the state variable matrix set X = $[\mathrm{X}_1^t,\mathrm{X}_2^t,\mathrm{X}_3^t\ldots.\mathrm{X}_m^t]^t$ where, $\mathrm{X_i}=[\ \delta_i,\widetilde{\omega_\iota},E'_{qi},E'_{di},E_{xi}]\ ^t$ and the algebraic variable matrix set Y = $[\mathrm{Y}_1^{\mathrm{t}},\mathrm{Y}_2^{\mathrm{t}}\ldots.\mathrm{Y}_{\mathrm{n}}^{\mathrm{t}}\ ]$ where, $Y_i\ =[i_{qi},i_{di},\varphi_i,V_i]^t$, for buses 1 to m and $Y_i=[\varphi_i,V_i]^t$ for buses m+1 to n.

The differential algebraic equations can be symbolically written in terms of the above variables as given by (17),

$$\left.\begin{aligned} \dot{x}\ &=f(x,y)\\ 0&=g(x,y) \end{aligned}\right\} \quad (17)$$

3.3 NUMERICAL SOLUTION FOR DAE

The DAE model comprising of differential and algebraic equations can be solved by either an alternate solution method or simultaneous solution method.

3.3.1 Alternate Solution Method

The differential equations are first solved by applying a numerical integration algorithm and X is computed and then algebraic equations are solved using this to obtain Y.

3.3.2 Simultaneous Solution Method

This method uses implicit methods. First the differential equations are transformed into algebraic equations according to an implicit integration formula. The transformed equations together with the original algebraic equations form a set of simultaneous algebraic equations. By solving this set one obtains the desired results [8].

Simultaneous implicit method is more stable than partitioned explicit method [7].As simultaneous implicit method is the one used most commonly, here also that method is used to solve DAE.

3.4 SIMULTANEOUS IMPLICIT METHOD

The initial conditions need to be found first and can be computed by solving (18) & (19) ,

$$0 = f(x_0, y_0) \tag{18}$$

$$0 = g(x_0, y_0) \tag{19}$$

In the SI methods the differential equations (17) are algebraized using an implicit method (Numerical method).

For a system of differential equations represented by (20) as,

$$\dot{x} = f(x, y) \tag{20}$$

numerical method involves integrating the above equation (20) from t_n to t_{n+1}, the value of x at the instant t_{n+1} can be written as,

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(x, t)dt \tag{21}$$

The numerical algorithm is to compute x_n in terms of previous value(s) and derivatives evaluated at previous discrete points. Some of the common solution methods include-

- Forward Euler (FE) Method
- Backward Euler (BE) Method
- Trapezoidal Integration (TI) Method
- Runga Kutta (RK) Method

In order to take the advantage of large time step sizes the numerical integration method used in this project is trapezoidal integration method. Using trapezoidal rule of integration x_{n+1} can be expressed as follows:

$$x_{n+1} = x_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \tag{22}$$

Then the resulting algebraic equations are solved along with the algebraic equations of DAE using Newton's method.

3.5 CALCULATION OF INITIAL CONDITIONS

System initial conditions are obtained by pre-fault load flow. The real and reactive power (P_g and Q_g), the voltage magnitude (V_0) and angle(φ_0) at the generator terminals are obtained from the Steady State Load Flow (SSLF) calculations. In steady state the derivatives of state variables are zero, i.e, $\dot{x} = 0$, in (17). The initial conditions obtained are given by (23)-(26):

$$\left.\begin{aligned} \bar{I}_{g0i} = I_{g0i} < \beta_{0i} &= \frac{P_{g0i} - Q_{g0i}}{V_{0i} < \varphi_{0i}} \\ E_{q0i} < \delta_{0i} = V_{0i} < \varphi_{0i} &+ \left(R_{ai} + jx_{qi}\right) * I_{g0i} < \beta_{0i} \end{aligned}\right\} \tag{23}$$

$$\left.\begin{aligned} i_{doi} &= -I_{g0i}\sin(\delta_{0i} - \beta_{0i}) \\ i_{qoi} &= I_{g0i}\cos(\delta_{0i} - \beta_{0i}) \\ V_{doi} &= -V_{0i}\sin(\delta_{0i} - \varphi_{0i}) \\ V_{qoi} &= V_{0i}\cos(\delta_{0i} - \varphi_{0i}) \end{aligned}\right\} \tag{24}$$

$$\left.\begin{aligned} E_{fdoi} &= E_{qoi} - \left(x_{di} - x_{qi}\right) * i_{doi} \\ E'_{qoi} &= E_{fdoi} + \left(x_{di} - x'_{di}\right) * i_{doi} \\ E'_{doi} &= \left(x_{qi} - x'_{qi}\right) * i_{qoi} \\ P_{eoi} &= E'_{qoi} * i_{qoi} + E'_{doi} * i_{doi} + \left(x'_{di} - x'_{qi}\right) * i_{qoi} * i_{doi} = P_{moi} \end{aligned}\right\} \quad (25)$$

$$V_{refi} = \frac{E_{fdoi}}{K_{Ai}} + V_i \quad (26)$$

i=1,2,.....m

After finding the initial conditions, numerical solutions need to be found out using trapezoidal method. Let the subscripts n and $n+1$ denote the time instants t_n and t_{n+1}. Then integrating the differential equations in (20) t_n from to t_{n+1} using the trapezoidal rule as (27) & (28),

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(x,t)dt \quad (27)$$

$$x_{n+1} = x_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \quad (28)$$

Where, $h = t_{n+1} - t_n$ is the integration time step. Now from (17)

$$0 = g(X_{n+1}, Y_{n+1}) \quad (29)$$

Rearranging the equations (28) and (29) we get,

$$\left.\begin{aligned} [x_{n+1} - \tfrac{h}{2}[f(x_n, y_n)] - [x_n + \tfrac{h}{2}[f(x_{n+1}, y_{n+1})]] &\triangleq F(x_{n+1}, y_{n+1}) = 0 \\ g(x_{n+1}, y_{n+1}) &\triangleq G(x_{n+1}, y_{n+1}) = 0 \end{aligned}\right\} \quad (30)$$

At each time step (30) are solved by Newtons method. The Newtons iterates are

$$\left.\begin{aligned} &\begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}^k \begin{bmatrix} \Delta X_{n+1} \\ \Delta Y_{n+1} \end{bmatrix}^k \begin{matrix} = \\ = \end{matrix} \begin{bmatrix} F \\ G \end{bmatrix}^k \\ &x_{n+1}^{(k+1)} = x_{n+1}^{(k)} + \Delta x^{(k)} \\ &y_{n+1}^{(k+1)} = y_{n+1}^{(k)} + \Delta y_{n+1}{}^{(k)} \end{aligned}\right\} \quad (31)$$

Where k is the iteration number for time step t_{n+1}.

The Jacobian matrices J_1 ,J_2 ,J_3 and J_4 can be derived analytically or calculated numerically. These are given by,

$$\left.\begin{aligned} J_1(i,j) &= \frac{\partial F_i(x,y)}{\partial x_j} \\ J_2(i,j) &= \frac{\partial F_i(x,y)}{\partial y_j} \\ J_3(i,j) &= \frac{\partial G_i(x,y)}{\partial x_j} \\ J_4(i,j) &= \frac{\partial G_i(x,y)}{\partial y_j} \end{aligned}\right\} \quad (32)$$

3.6 DISTURBANCE SIMULATION

We assume a network disturbance at time $t=0$. Then the parameters in algebraic equations change at $t=0^+$. The algebraic variables change instantaneously but the state variables do not. Thus at time $t=0^+$ only the network equations, only the algebraic equations need to be solved for y_{0+} as given by (33)

$$0 = g^f \ (x_0, y_{0+}) \tag{33}$$

where the superscript f indicates algebraic equations corresponding to the faulted state. With the value of y_{0+} so obtained the trapezoidal method is then applied. The initial guess for the vector $[x^t \ y^t]^t$ at time instant t_{n+1} is the converged value at the previous time instant, i.e,

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix}^0 = \begin{bmatrix} x_n \\ y_n \end{bmatrix} \tag{34}$$

3.7 SOLVING ONLY ALGEBRAIC EQUATIONS

Whenever there is a need to solve the algebraic equations (13), (14), (15) for the algebraic variables alone, these equations can be rewritten in terms of Kron's variables and the resulting set of linear equations can be solved non-iteratively. This is illustrated below as follows, The network algebraic equations in (14) and (15) can be represented in current balance form as in (35) & (36) ,

$$\left.\begin{aligned} &\textstyle\sum_{k=1}^{n}(G_{ik} * V_{Qk} - B_{ik} * V_{Dk}) - i_{Qi} + a_i * V_{Qi} + b_i * V_{Di} = 0 \\ &\textstyle\sum_{k=1}^{n}(B_{ik} * V_{Qk} - G_{ik} * V_{Dk}) - i_{Di} - b_i * V_{Qi} + a_i * V_{Di} = 0 \\ &\qquad\qquad i=1,2....m \end{aligned}\right\} \tag{35}$$

$$\left.\begin{aligned} &\textstyle\sum_{k=1}^{n}(G_{ik} * V_{Qk} - B_{ik} * V_{Dk}) + a_i * V_{Qi} + b_i * V_{Di} = 0 \\ &\textstyle\sum_{k=1}^{n}(B_{ik} * V_{Qk} - G_{ik} * V_{Dk}) - b_i * V_{Qi} + a_i * V_{Di} = 0 \qquad i=m+1,m+2....n \end{aligned}\right\} \tag{36}$$

$$\left.\begin{aligned} \begin{bmatrix} V_{qi} \\ V_{di} \end{bmatrix} &= \begin{bmatrix} \cos\partial_i & \sin\partial_i \\ -\sin\partial_i & \cos\partial_i \end{bmatrix} \begin{bmatrix} V_{Qi} \\ V_{Di} \end{bmatrix} \\ \begin{bmatrix} i_{qi} \\ i_{di} \end{bmatrix} &= \begin{bmatrix} \cos\partial_i & \sin\partial_i \\ -\sin\partial_i & \cos\partial_i \end{bmatrix} \begin{bmatrix} i_{Qi} \\ i_{Di} \end{bmatrix} \\ &\qquad i=1,2....m \end{aligned}\right\} \tag{37}$$

Voltage and admittance can be written as in (38) & (39)

$$V_{Qi} + jV_{Di} = V_i e^{j\varphi_i} \tag{38}$$

$$i=1,2.......n$$

$$G_{ik} + jB_{ik} = Y_{ik}e^{j\alpha_{ik}} \quad (39)$$

i=1,2.......n and k=1,2...n

Now from (3) and (37) it can be derived that

$$\begin{bmatrix} \mathrm{i}_{Qi} \\ \mathrm{i}_{Di} \end{bmatrix} = \frac{1}{\mathrm{Ra}_i^2 + \mathrm{x}'_{di} * \mathrm{x}'_{qi}} * \begin{bmatrix} \cos\partial_i & -\sin\partial_i \\ \sin\partial_i & \cos\partial_i \end{bmatrix} \begin{bmatrix} \mathrm{Ra}_i & \mathrm{x}'_{di} \\ -\mathrm{x}'_{qi} & \mathrm{Ra}_i \end{bmatrix} \begin{bmatrix} \mathrm{E}'_{qi} \\ \mathrm{E}'_{di} \end{bmatrix} - \frac{1}{\mathrm{Ra}_i^2 + \mathrm{x}'_{di} * \mathrm{x}'_{qi}}$$

$$* \begin{bmatrix} \cos\partial_i & -\sin\partial_i \\ \sin\partial_i & \cos\partial_i \end{bmatrix} \begin{bmatrix} \mathrm{Ra}_i & \mathrm{x}'_{di} \\ -\mathrm{x}'_{qi} & \mathrm{Ra}_i \end{bmatrix} \begin{bmatrix} \cos\partial_i & \sin\partial_i \\ -\sin\partial_i & \cos\partial_i \end{bmatrix} \begin{bmatrix} \mathrm{V}_{Qi} \\ \mathrm{V}_{Di} \end{bmatrix}$$

i=1,2....m (40)

(36), (37) and (40) form a set of *2m+2n* linear equations in *2m+2n* linear variables viz, V_{D1} to V_{Dn}, V_{Q1} to V_{Qn}, i_{D1} to i_{Dm} and i_{Q1} to i_{Qm}, which can be solved directly.

3.8 CENTRE OF INERTIA FRAME OF REFERENCE

The equations describing the behaviour of the synchronous machines are formulated with respect to a fictitious centre known as centre of inertia frame of reference. In this project all the equations and simulations have been done with respect to centre of inertia frame of reference. For example, even if all the machine rotor angles are accelerating, they may be in synchronism. Hence, it is necessary to use centre of inertia frame of reference. Defining the COI angle δ_r, given by (41) & (42) as,

$$\delta_r = \frac{\sum_{k=1}^{m} H_k * \delta_k}{H_T} \quad (41)$$

Where,

$$H_T = \sum_{k=1}^{m} H_k \quad (42)$$

The rotor angles θ_i' s with respect to COI is given as

$$\theta_i = \delta_i - \delta_r$$

And the equation for COI is given by (43)-(45) as,

$$\frac{2*H_i}{\omega_B} * \frac{d\breve{\gamma}_\iota}{dt} = P_{mi} - P_{ei} - \frac{H_i}{H_T} - P_{COI} \quad (43)$$

Where

$$\breve{\gamma}_\iota = \frac{d\theta_i}{dt} \quad (44)$$

$$P_{COI} = \sum_{i=1}^{m} P_{mi} - P_{ei} \quad (45)$$

All the above equations are defined for *i=1,2,....m.*

3.9 ADVANTAGES OF TIME DOMAIN SIMULATION

Time domain simulation (TDS) is one amongst the commonly used methods of stability analysis and is generally regarded as the best available analysis tool. The approach has several advantages:

- It is directly applicable to any level of detail of power system models.
- All the information of state variables during transient as well as steady-state is available.
- Simulation results can be directly interpreted by system operators.

The results of this simulation are usually used to determine the correctness of various direct methods. This method also suffers from certain limitations such as:

- It requires intensive time-consuming numerical integration, therefore it has not been suitable for on-line applications.
- It does not provide information regarding the degree of stability (when the system is stable) and the degree of instability (when the system is unstable).
- It does not provide information as to how to derive preventive control when the system is deemed unstable [3].

3.10 CASE STUDIES

In this section some cases of TDS is presented. The simulations were carried out in MATLAB and MATLAB Simulink. The Test Systems selected are:

1. An SMIB system (System data and parameters given in appendix)
2. 10 generator, 39-bus New England system (Single line diagram and data given in appendix).

The IEEE 39-bus (New England) power system is an equivalent power system of subsystems of the New England area and Canada. It consists of 39 buses including 10 generator buses connected to 10 generators, 12 transformers, 34 transmission lines, and 19 loads. The system is shown in Appendix.

3.10.1 SMIB System –

A three phase fault at sending end of one of the circuits of the transmission line was considered(fault occurs at t=1sec). The fault was cleared by tripping the faulted line connecting buses 1 and 2. The critical clearing time was found to be between 110 ms and 111 ms. The swing curve for SMIB system is shown below. Fig 3.2 shows the stable case and Fig 3.4 the unstable case. The simulations were carried out by modelling the equations in MATLAB SIMULINK.

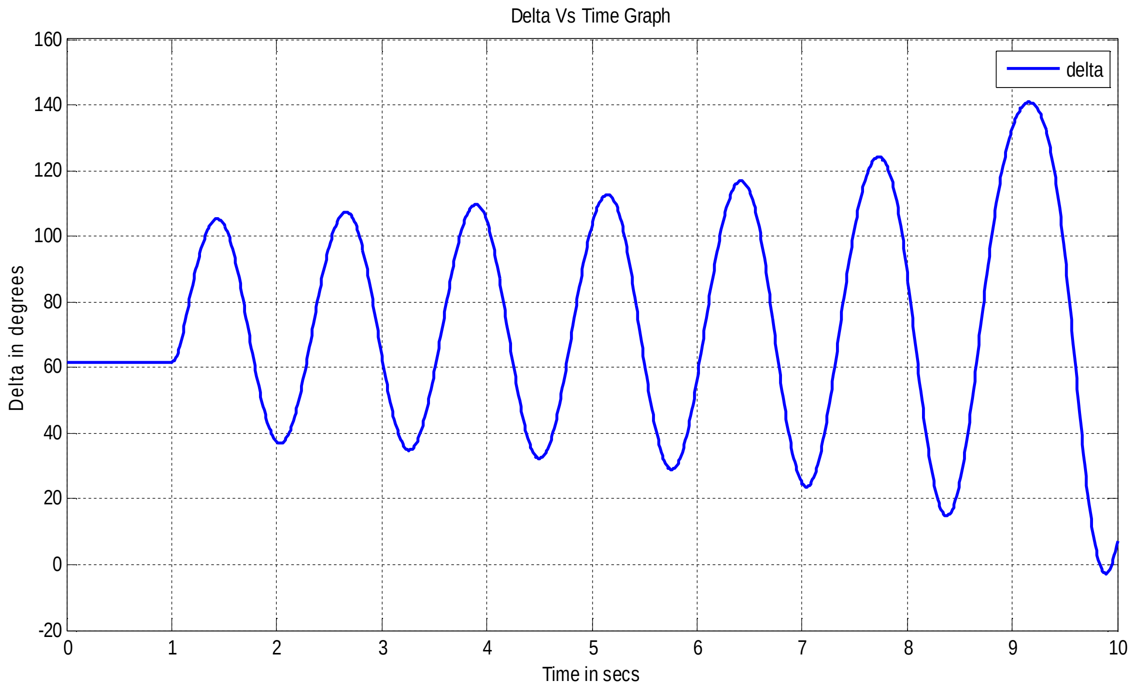


Fig 3.3. Swing curve for SMIB system for t_c =110ms

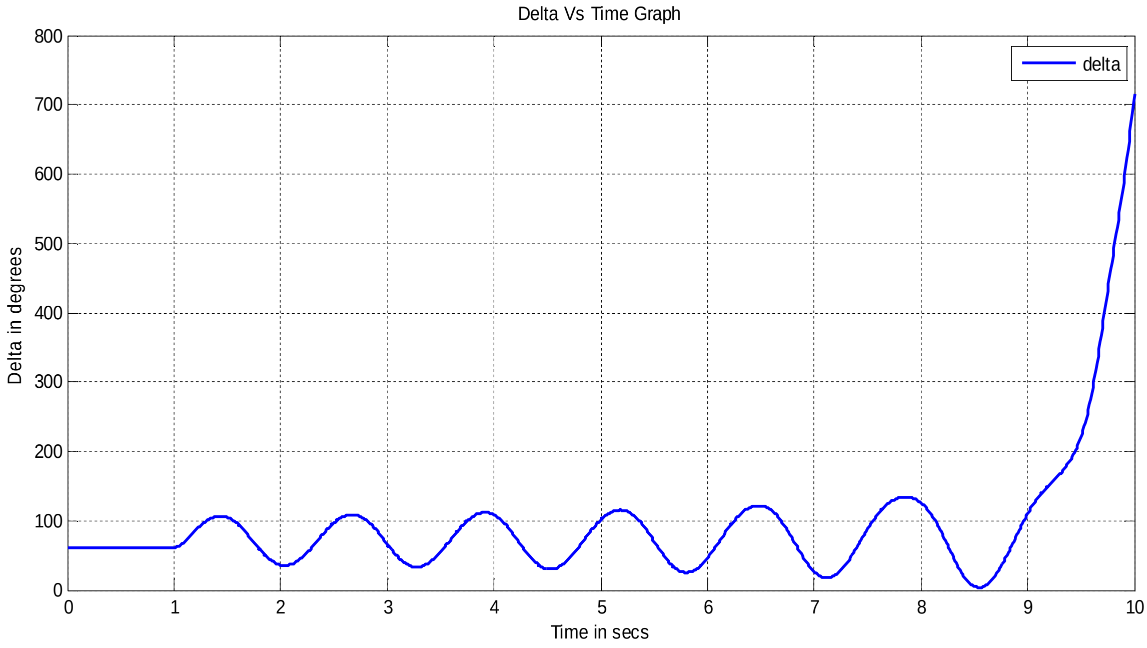


Fig 3.4 Swing curve for SMIB system for t_c =111ms

3.10.2 10 Generator 39 Bus System (IEEE 10 Bus System) –

The case of the above SMIB system was expanded to that of a multi-machine system. Here time domain simulations were carried out for the 10 machine 39 bus New England system. In each of the cases, a three phase fault on different bus locations were assumed. The fault was cleared by tripping some transmission line. A case of three phase fault on bus # 22 has been shown below. The fault was cleared was tripping line between bus # 22 and bus # 21. The critical clearing time was found to be between 0.20s and 0.21s. The swing curves for all the generators have been shown in Fig 3.5 for stable case and Fig 3.6 for unstable case. Similar tine domain simulations were carried out for 28 different cases and the results have been tabulated in Table 3.1. All the analysis were carried out by developing the necessary algorithm in MATLAB.

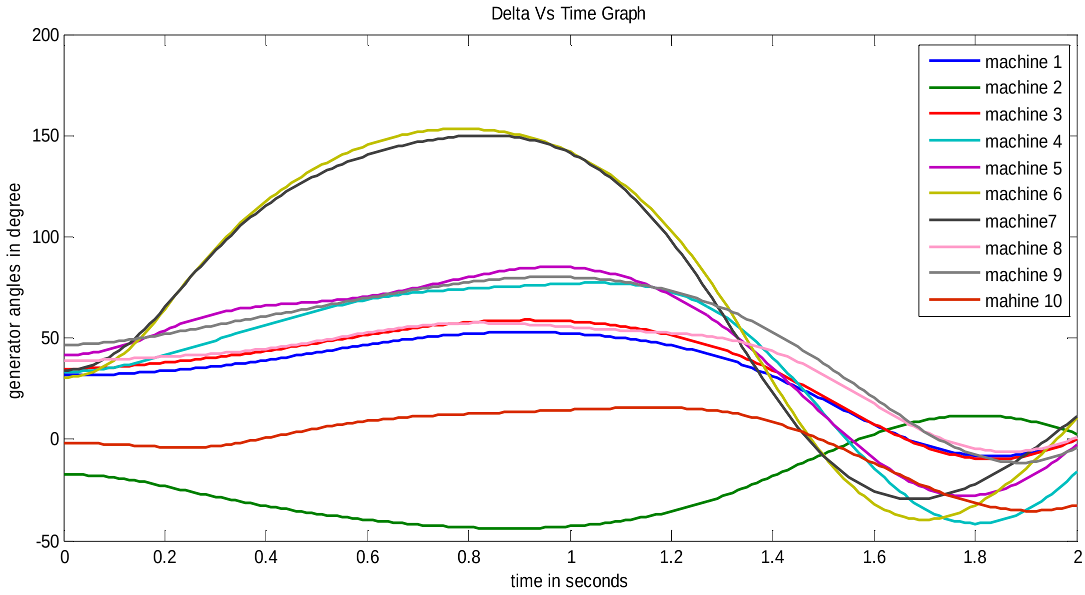


Fig 3.5 Swing curves for 10 generator 39 bus system with respect to COI frame of reference for fault on bus #22 for t_c = 0.20s.

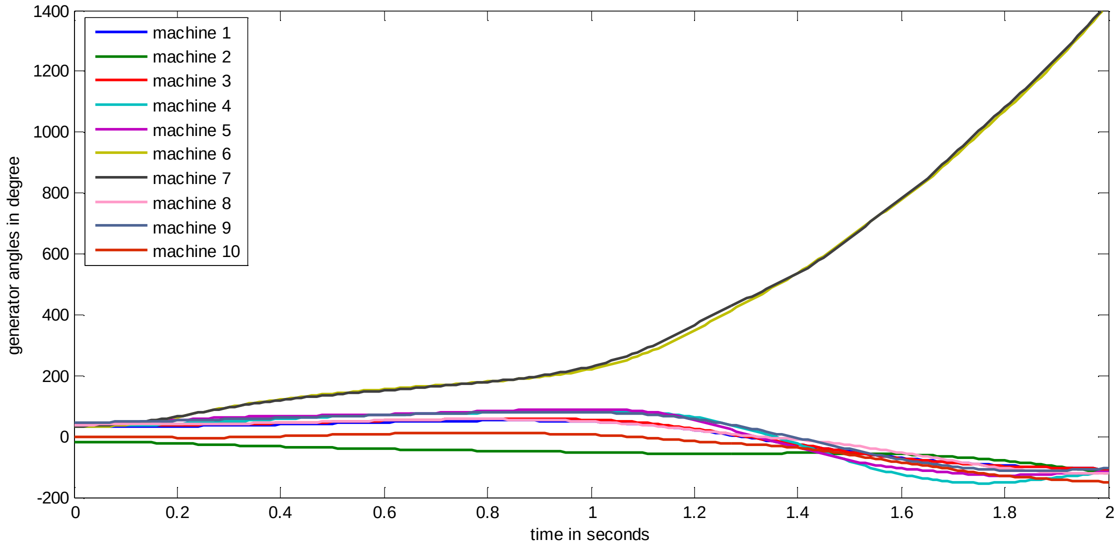


Fig 3.6 Swing curves for 10 generator 39 bus system with respect to COI frame of reference for fault on bus #22 for t_c = 0.21s.

TABLE 3.1 Critical clearing time for fault at various locations using time domain simulation

Sl No	Fault Bus #	Line cleared	Critical Clearing Time (s)
1.	14	14 – 34	0.34 – 0.35
2.	16	16 – 31	0.25 – 0.26
3.	22	22 – 21	0.20 – 0.21
4.	28	28 – 29	0.15 – 0.16
5.	20	20 – 31	0.26 – 0.27
6.	34	34 – 14	0.35 – 0.36
7.	31	31 – 16	0.30 – 0.31
8.	26	26 – 25	0.19 – 0.20
9.	29	29 – 28	0.12 – 0.13
10.	31	31 – 20	0.29 – 0.30
11.	21	21 – 22	0.25 – 0.26
12.	36	36 – 35	0.25 – 0.26
13.	26	26 – 29	0.17 – 0.18
14.	29	29 – 26	0.14 – 0.15
15.	22	22 – 23	0.27 – 0.28
16.	23	23 – 22	0.25 – 0.26
17.	18	18 – 15	0.34 – 0.35
18.	21	21 – 36	0.30 – 0.31
19.	36	36 – 21	0.25 – 0.26
20.	33	33 – 34	0.29 – 0.30
21.	34	34 – 33	0.33 – 0.34
22.	31	31 – 38	0.30 – 0.31
23.	12	12 – 25	0.34 – 0.35
24.	25	25 – 26	0.21 – 0.22
25.	25	25 – 12	0.19 – 0.20
26.	35	35 – 36	0.39 – 0.40
27.	12	12 – 11	0.29 – 0.30
28.	38	38 – 13	0.41 – 0.42

CHAPTER 4

TRANSIENT STABILITY ANALYSIS USING ENERGY FUNCTION

The literatures in the beginning of this century has appreciated the usefulness of conventional method (TDS) of stability analysis. In practical circumstances transient stability analysis envisions to periodically scan the system state variables and predict or calculate the transient behaviour of the system when subjected to specified disturbances. The digital calculations are continually updated by actual readings from power system. For online analysis, possibly 50 to 100 transient stability studies of the existing system are requires to be run periodically, possibly every 5 or 10 minutes during peak load periods [11, 12, 13]. This requirement is far beyond the ability of TDS. Hence, even though the results of TDS are accurate, the computation involved increases with system size and there are chances that it may become computationally prohibitive for large power networks. Current deregulated environment, competitive markets, steadily increasing loads scenario etc at times can result in cascaded outages which can even lead to system collapse if there are no fast and adaptive stability analysis tools. Therefore, there is an inevitable need for a fast, accurate and direct method of transient stability analysis.

4.1 LITERATURE REVIEW ON VARIOUS DIRECT METHODS OF STABILITY ANALYSIS

In 1892, A. M. Lyapunov, in his famous Ph.D dissertation [7, 14] proposed that the stability of the equilibrium point of a non-linear dynamic system can be ascertained without numerical integration. It was actually a generalisation of the concept of the energy associated with a system. Since 1948, when the results of Lyapunov appeared in English language along with the potential applications there has been massive literature surrounding this topic. Power system stability analysis using the application of energy function started with the early work of Magnusson[7, 15] and Aylett[7, 16] followed by a formal application of the more general Lyapunov's method by El-Abaid and Nagappan [7, 17]. an algorithm to determine critical clearing time was also proposed in [17]. Since then many Lyapunov's functions have been developed but the first integral of motion, which is sum of kinetic and potential energies appeared to have provided the best results. Thus transient Energy function(TEF) method evolved as a popular direct method of stability analysis of a multi-machine power system. Determination of CCT using TEF has been investigated and exhaustively reported in numerous articles and have been summarised in [18]-[20]. TEF method has been basically used with classical model of power system and CCT has been determined using numerical integration method. But TEF method had some modelling limitations and for some cases energy functions did not even existed [20].

PAPER PUBLISHED : R. Garg, B. Sebastian, "Transient Stability Analysis of a General Multi-Machine Power system using Energy Functions considering Detailed Generator Model", *International Journal for Applied Engineering Research,*Vol 8, No 7, March 2013.

The authors of [2, 4] proposed a direct stability analysis method using individual machine or group of machine energy functions. The energy function of individual machine is compared to a critical energy, which is found using Potential Energy Boundary Surface (PEBS) method. A theoretical foundation of PEBS has been provided in [21].

Another method for transient stability analysis of multi-machine system amplifying the advantages of TEF while circumventing its difficulties was developed by Xue in [20]-[23]. He decomposed a multi-machine system into two subsystems and the machine in each subsystem was replaced by an equivalent machine. The two machine equivalent was then further reduced to an SMIB model through extended equal area criterion [EEAC]. This procedure was continued for various faults leading to one or more equivalent SMIB model and CCT was determined in each model. The actual value of CCT was considered to be the one with smallest value. The suitability of EEAC was further examined and scrutinized in reference [20] & [23]. Lemmon in [24] used the quasi stable equilibrium point (QUEP) to calculate the transient stability of multi-machine system in real time. For a given fault the method considered several critical clusters based on the machine states(speed and angle) at fault clearing. The actual critical cluster is considered to be that having the least stability margin. The machines in actual critical cluster are aggregated into a single machine and the rest of machines in the system are similarly combined and replaced by another machine. Then two machine model is replaced b equivalent SMIB model. The method assess the transient stability of original system, for a given fault clearing time, but does not provides the CCT [20].

Unlike the above two methods Rastgoufard in [25] proposed a modified EAC to determine the transient stability of multi-machine system without representing the system by an equivalent SMIB model. The method uses individual machine energy function to determine accelerating and decelerating energies of the critical machine. By observing the values of these energies the transient stability for a given fault clearing time can be checked. Paper [26] deals with the stability analysis of AC/DC power system. It does not uses a routine extension of the method applied to AC systems, but uses a different approach based on post fault DC link dynamics.

In 1989 determination of the First swing stability of a power system i.e, the existence of the first peak or maximum value of the rotor angle in the post fault period by the standard EAC of the SMIB model was studied. The existence of a peak of the post fault rotor angle indicates that the machine speed is not monotonic (it changes sign) in the post fault period. A method of determining the first swing stability limit of a multi-machine system, using the above idea is described in reference [27]. The method is based on determination the existence of peaks of the post fault rotor angles by observing the roots of time derivative of rotor angles expressed by truncated Taylor series expansions. It actually used faulted angular deviation to identify the SDMs. Then only severely disturbed machines(SDM) (which are in general responsible for system instability) were checked for existence of peak angles. In reference [28] Donga and Potta developed a large step size based numerical integration method to determine the existence of peaks of the post fault rotor angles of severely disturbed machines. This paper used the machine states (angle and speed) at fault clearing to identify the SDMs. Unlike the above two papers, paper [29] used a combination of electrical output power and machine inertia constant can also be used to detect the most SDM.

Paper [20] also proposed a direct method of finding the existence of peak of the post fault rotor angle of the most severely disturbed machines by comparing to areas or integrals like the standard EAC, but without replacing the system by an equivalent SMIB model. The transient stability problem has been treated as a boundary value problem in [30]. The authors of [31] used critical trajectory method.

The researches based on energy functions and its variants had used classical model during the 19th century, but the same have been used with detailed machine model in the 20th century [32], SDMs[2]. With the advent of FACTS devices, transient stability analysis considering the effects of different FACTs devices is an area that is recently gaining momentum. Paper [33] documents the utilisation of transient energy function to assess first swing stability of power systems including SVCs. Paper [34] and [35] includes STATCOM and UPFC in transient stability analysis.

4.2 TRANSIENT ENERGY FUNCTION METHOD

The Direct method of stability analysis based on transient energy has received considerable attention since the early works of Magnusson and Aylett. The energy based methods are a special case of more general Lyapunov’s second methods [36]. The transient energy functions describe the system transient energy and power network’s ability to “absorb” this energy so that the synchronous machines reach new post-disturbance equilibrium state[4]. Energy function analysis assumes that a post fault system has a stable equilibrium point(SEP). Hence, the problem of transient stability evaluation is to determine whether the initial state (for the post fault system) lies in the region of stability around the SEP [1].

Let $A(x_s)$ be the region of stability around the SEP (x_s) and $x(t_c)$ be the state of system during fault clearing, shown in Fig 4.1. If $x(t_c) \epsilon A(x_s)$ the system is stable.

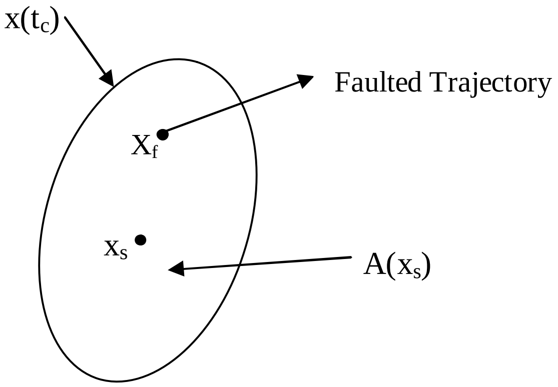


Fig 4.1 Definition of Critical Clearing Time [1]

There are two steps in this evaluation

i. Characterization of the boundary of the stability region, $ðA_s$.
ii. Computation of post-fault initial state $x(t_c)$ and comparing with $ðA_{s.}$

A clear understanding of the transient energy approach can be described by considering ball rolling analogy [36] on the inner surface of a bowl as shown in Fig 4.2. The area inside the bowl represents the region of stability, and the area outside is the region of instability. The bowl is irregular in shape so that different points on the rim have different heights.

Initially the ball is resting at the bottom of the bowl, and this state is referred as the stable equilibrium point (SEP). When some kinetic energy is injected into ball, causing it to move in particular direction, the ball will roll up the inside face of the bowl along a path determined by the direction of initial motion. The position at which the ball will stop is governed by the kinetic energy initially injected. If the ball converts all its kinetic energy into potential energy before reaching the rim, then it will roll back and eventually settle down at the stable equilibrium again. However, if the kinetic energy injected is high enough to cause the ball go over the rim, and then the ball will enter the region of instability and will not return the stable equilibrium point. The surface inside the bowl represents the potential energy surface, and the rim of the bowl represents the potential energy boundary surface (PEBS)[18].

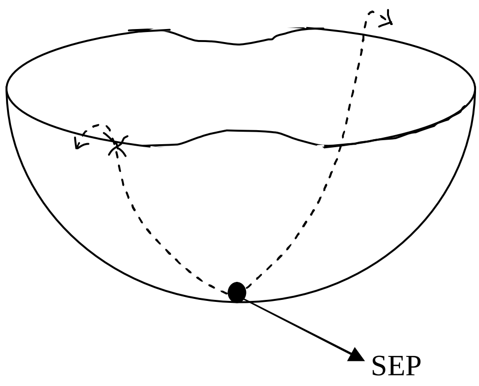


Fig 4.2 Rolling Ball Theory [36]

Energy function can be used to approximate the stability region. An expression for energy function has been derived below:

Rearranging equation (43), equation of COI, multiplying with $\frac{d\theta_i}{dt}$, and integrating from *0* to *t*, we obtain equation (46) as,

$$\int_0^t \left(\frac{2*H_i}{w_b} * \frac{d\gamma_i}{dt}\right) \frac{d\theta_i}{dt} dt + \int_0^t \left(-P_{mi} + P_{ei} + \frac{H_i}{H_T} * P_{COI}\right) \frac{d\theta_i}{dt} dt = W_i\ (t) \qquad (46)$$

It is to be noted that energy function *W* is always defined for post-disturbance system.

Splitting the terms of energy function, we can obtain expression for Kinetic energy W_{ki} and Potential energy W_{pi} for i^{th} machine as (47) & (48) :

$$W_{ki}(t) = \int_0^t \left(\frac{2*H_i}{w_b} * \frac{d\gamma_i}{dt}\right) \frac{d\theta_i}{dt} dt = \frac{2*H_i}{w_b} * \left(\frac{d\theta_i}{dt}\right)^2 \qquad (47)$$

And $$W_{pi}(t) = \int_0^t \left(-P_{mi} + P_{ei} + \frac{H_i}{H_T} * P_{COI}\right) \frac{d\theta_i}{dt} dt \qquad (48)$$

$$W_i(t) = W_{pi}(t) + W_{ki}(t) \tag{49}$$

The individual machine energy functions have been calculated using trapezoidal rule. Summation of Kinetic, Potential and Total energies of all the machines gives the Kinetic, Potential and Total energy for the entire system as in equation (50),

$$\left.\begin{aligned} W_p &= \sum_{i=1}^{n} W_{pi}(t) \\ W_p &= \sum_{i=1}^{n} W_{pi}(t) \\ W &= \sum_{i=1}^{n} W_i(t) \end{aligned}\right\} \tag{50}$$

After the formulation of appropriate energy function the next step is to estimate the stability region by calculating critical energy W_{cr} .

Critical clearing time for a single machine system is the point where potential Energy is maximum. For an SMIB system, the Kinetic Energy goes to zero along post fault trajectory at the instants when potential energy is maximum. Also for a system with no damping, the oscillations don't decay and exchange between the potential and kinetic energies is continually taking place [1]. For a multi-machine system the critical clearing time can be obtained by Potential Energy Boundary Surface (PEBS) method.

4.3 POTENTIAL ENERGY BOUNDARY SURFACE METHOD

The procedure for calculation of critical clearing time using PEBS method has been described in [1] and is as follows:

i. Run a sustained fault case and integrate the fault-on trajectory until W_p reaches a local maximum at $t=t_m$. This gives the value of critical Energy W_{cr}.

ii. From the fault-on trajectory, find the instant at which $W = W_{cr}$. This gives the critical clearing time t_{cr} as predicted by Energy function method. The graphical illustration of the above is shown in Fig 4.3.

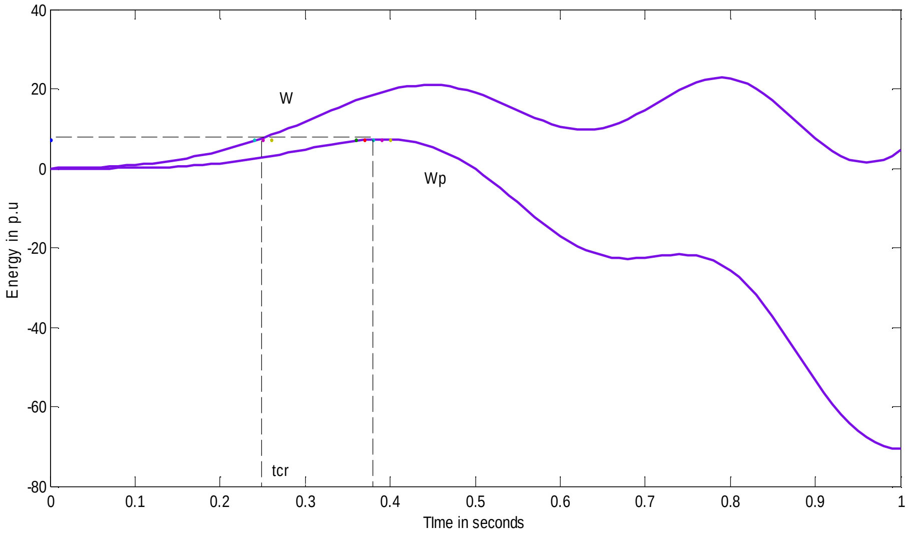


Fig 4.3 Determination of t_{cr} using PEBS

4.4 CASE STUDIES

4.4.1 SMIB system –

The disturbance considered is same as before and was simulated in Simulink utility of MATLAB. The critical clearing time obtained is 120 ms. (Here the fault occurs at t=1secs).

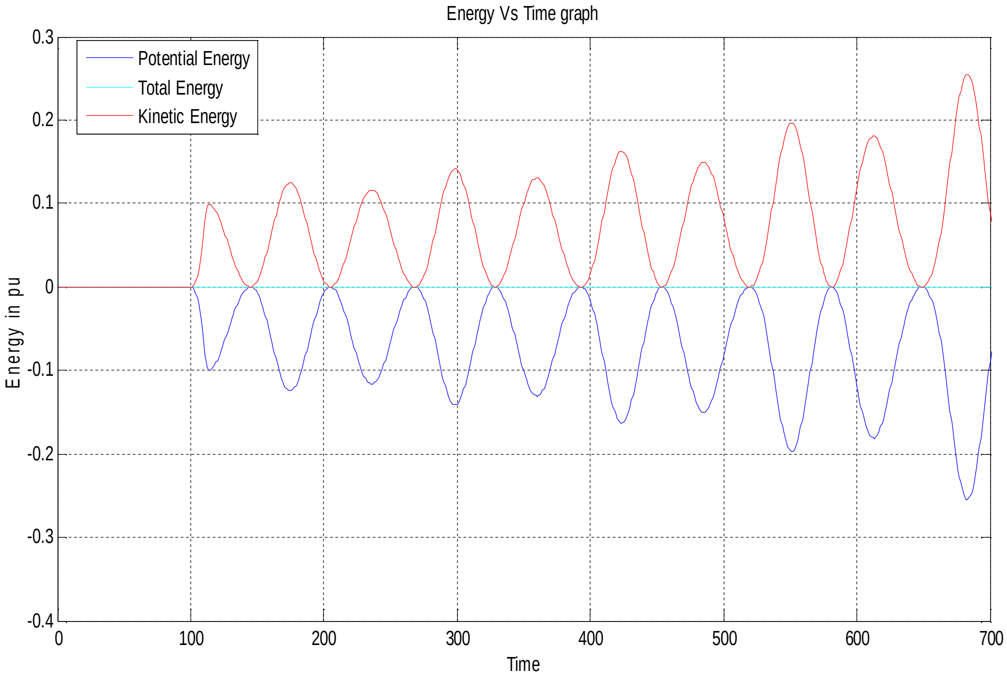


Fig.4.4 Plots of Energy Function of the SMIB System

4.4.2 10 Machine 39 Bus System (IEEE 39 Bus System)-

Sustained fault simulation was run for three phase fault on bus #22 and cleared b opening line between bus #22 and bus #21. The energy function plots have been shown in Fig 4.5. The critical clearing time was obtained between 0.23 – 0.24 sec which is found to be close to the values obtained by TDS. The 28 fault cases used for TDS have been simulated in a similar manner and CCT have been predicted by Energy function method. The results have been shown in Table 4.1. The predicted clearing time is found to be in approximation with the values obtained by TDS. Hence, Energy function method is a very accurate and reliable Direct method of stability analysis. MATLAB software was used to develop the algorithm for Direct stability method.

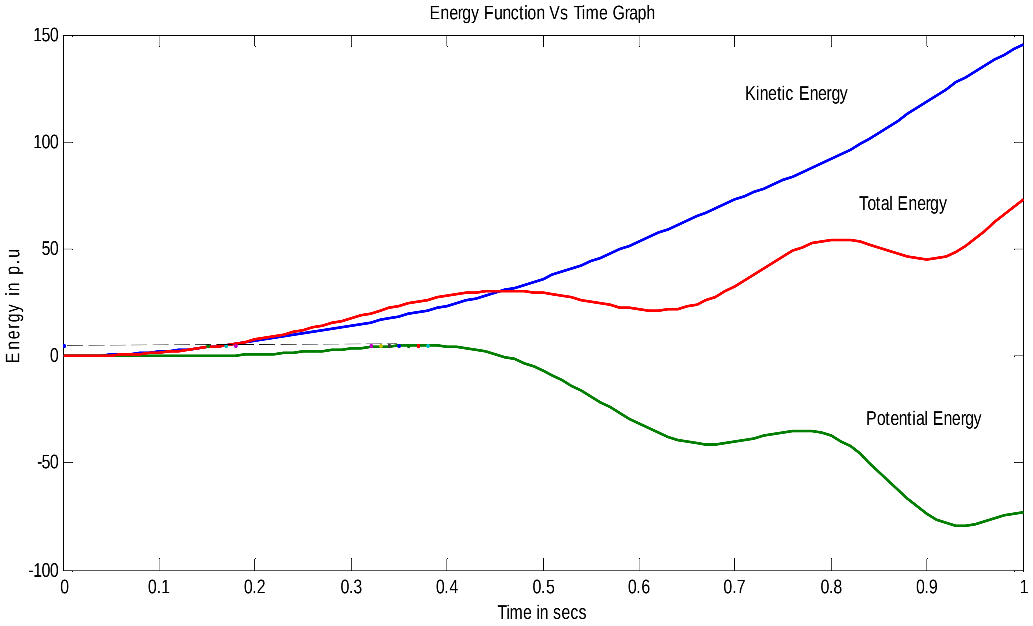


Fig 4.5 Plots of Energy Function of 10 Generator 39 Bus System

TABLE 4.1 Comparison of CCTs of Transient Simulation and Energy Function method

Sl No	Fault Bus #	Line cleared	CCT by TDS (secs)	CCT by TEF (secs)
1.	14	14 – 34	0.34 – 0.35	0.31-0.32
2.	16	16 – 31	0.25 – 0.26	0.22-0.23
3.	28	28 – 21	0.20 – 0.21	0.16-0.17
4.	20	20 – 31	0.15 – 0.16	0.14-0.15
5.	22	22 – 21	0.26 – 0.27	0.24-0.25
6.	34	34 – 14	0.35 – 0.36	0.32-0.33
7.	31	31 – 16	0.30 – 0.31	0.26-0.27
8.	26	26 – 25	0.19 – 0.20	0.18-0.19
9.	29	29 – 28	0.12 – 0.13	0.10-0.11
10.	31	31 – 20	0.29 – 0.30	0.26-0.27
11.	21	21 – 22	0.25 – 0.26	0.23-0.24
12.	36	36 – 35	0.25 – 0.26	0.23-0.24
13.	26	26 – 29	0.17 – 0.18	0.17-0.18
14.	29	29 – 26	0.14 – 0.15	0.12-0.13
15.	22	22 – 23	0.27 – 0.28	0.22-0.23
16.	23	23 – 22	0.25 – 0.26	0.22-0.23
17.	18	18 – 15	0.34 – 0.35	0.30-0.31
18.	21	21 – 36	0.30 – 0.31	0.27-0.28
19.	36	36 – 21	0.25 – 0.26	0.23-0.24
20.	33	33 – 34	0.29 – 0.30	0.26-0.27
21.	34	34 – 33	0.33 – 0.34	0.31-0.32
22.	31	31 – 38	0.30 – 0.31	0.26-0.27
23.	12	12 – 25	0.34 – 0.35	0.34-0.35
24.	25	25 – 26	0.21 – 0.22	0.19-0.20
25.	25	25 - 12	0.19 – 0.20	0.19-0.20
26.	35	35 – 36	0.39 – 0.40	0.40-0.41
27.	12	12 – 11	0.29 – 0.30	0.32-0.33
28.	38	38 – 13	0.41 – 0.42	0.42-0.43

CHAPTER 5

TRANSIENT STABILITY ANALYSIS USING CRITICAL MACHINE ENERGY FUNCTION

The transient stability analysis of multi-machine system using energy functions carried out in previous chapter showed that the results of critical clearing time obtained is in close approximation with the results of TDS. The success of Energy function techniques in first swing stability analysis lead the researchers to investigate further so as to obtain a better accountability of the sysem transient energy especially the energy causing separation of one or more of machines from the system, and where it resides [4]. The authors of [37,38] discovered that significant portion of system transient energy do not contribute to system separation; hence corrections need to be applied to transient stabiltiy assessment using TEF. By "pragmatic" means (engineeirng judgement) it was preposed by researchers that the problem was more elementary rather than making "corrections" in the expression for energy functions. Later on from the simulations carried out by researchers of [4] it became apparent that the system separation does not depend on the total system energy, but rather on the transient energy of individual machines or groups of machines tending to separate from from the rest. It was hence reasoned that a new approach was needed.

The studies in [4] had used classical model of generators, neglecting flux decay and exciter controls. The transient stability analysis carried out in this chapter using critical machine energy function has used the 1.1 model of generator eliminating the major shortcomings of classical model. Whenver a power system becomes unstable, it is usually split into two groups (one amongst which consists of a unique cutset) across which the angle becomes unbounded. The important consideration here is the dentification of the critical machines.Conventionally, critical machines are identified by examining the relative swing curves between machines [39]. But it is not a feasible option even in offline applications to search thoroughly the critical cluster amonsgt all possible combination of machines. Significant redunction on the number of machine combinations has been obtained b using the sorter trajectory approach [40].

With the introduction of structure preserving model, an index of critical and vulnerable cutset is defined for determining the critical cluster [41]. However, in this method, all the cutsets have to be tested with the characteristics of fault-dependence of transient stability ignored. In [42], the concepts of key cutset based on structure preserving model were prposed to analyse the critical cluster of power system. An alternative approach based on ranking of critical machines has been presented in [43] and realized in the EEAC method.

5.1 STABILITY ANALYSIS USING CRITICAL GROUP OF MACHINES

Let C represent the set of generators belonging to critical group determined by sustained fault simulation. The critical clearing time can be calculated by using the energies of only the critical machines, in the individual machine energy function expression obtained earlier in (47), (48) & (49) which are reproduced here :

$$W_{ki}(t) = \int_0^t \left(\frac{2 * H_i}{w_b} * \frac{d\gamma_i}{dt}\right)\frac{d\theta_i}{dt}dt = \frac{2 * H_i}{w_b} * \left(\frac{d\theta_i}{dt}\right)^2$$

$$W_{pi}(t) = \int_0^t \left(-P_{mi} + P_{ei} + \frac{H_i}{H_T} * P_{COI}\right)\frac{d\theta_i}{dt}dt$$

$$W_i(t) = W_{pi}(t) + W_{ki}(t)$$

i = 1,2,....m

5.2 DETERMINATION OF CRITICAL GROUP OF MACHINES

A method based on heuristics proposed in [32] has been adopted here to determine the critical group of machines. It is as follows:

i. Run sustained fault simulation and find time instant t_m at which total potential energy of the system reaches a local maximum as seen from Fig 5.1.

ii. Calculate rotor angles (with respect to COI reference) θ for all the machines at $t = t_m$ and sort them in descending order.

iii. Compute the difference between the consecutive rotor angles and find the maximum amongst them. The machine which lies above this maximum difference forms the critical group and rest of the machines form the on-critical group.

5.3 DETERMINATION OF CRITICAL CLEARING TIME

After deciding the critical group of machines, the critical clearing time is obtained by the procedure given below [32]:

i. Calculate the sum of potential energy functions and the sum of total energy functions for the generators belonging to critical group. It is obtained as (51):

$$\left.\begin{aligned} W_p{}^C(t) &= \sum_{i \epsilon C} W_{pi}(t) \\ W^C(t) &= \sum_{i \epsilon C} W_i(t) \end{aligned}\right\} \quad (51)$$

ii. A sustained fault case is run and fault on trajectory is integrated until $W_p{}^c$ reaches a local maximum at $t = t_{mc}$. This value is W^C_{pmax}.

iii. From the fault-on trajectory, find the instant at which $W^C = W^C_{pmax}$. This instant is the critical clearing time as obtained by the critical machine energy function.

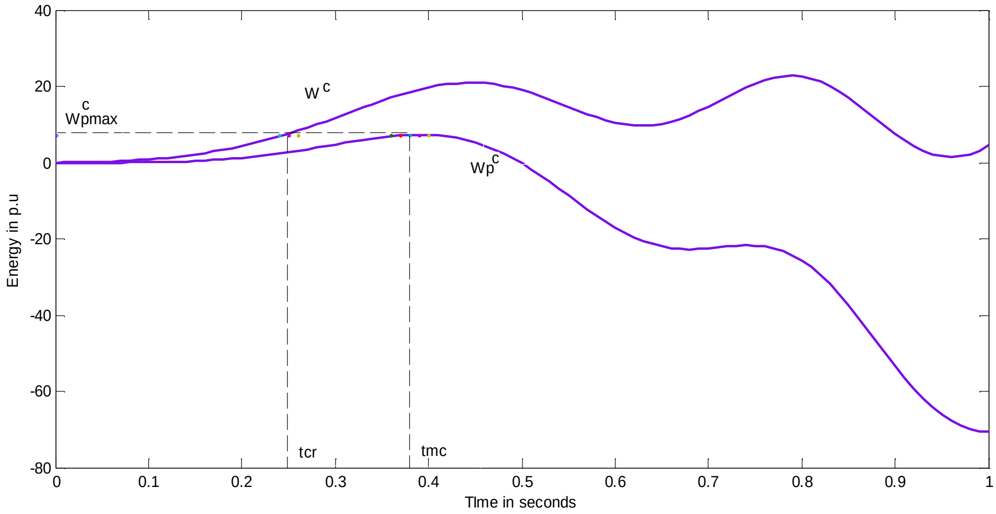


Fig 5.1 Determination of CCT by Critical group Energy Function

5.4 CASE STUDIES

5.4.1 SMIB System –

Since SMIB system has only one machine it itself is the critical machine and the procedure and results are same as in TEF method.

5.4.2 10 Generator 39 Bus System –

For the three phase fault on Bus # 22, cleared by tripping line 22 – 21, a sustained fault simulation was run and the swing curves have been obtained. The critical group of machines is found to comprise of machines 6 and 7, as shown in Fig 5.2. The energy function plot for the same has been shown in Fig 5.3 and the critical clearing time t_{cr} was obtained as 0.17s – 0.18s. The t_{cr} obtained is found to be more close to that of TDS than obtained from TEF method. MATLAB algorithm was developed for the above analysis.

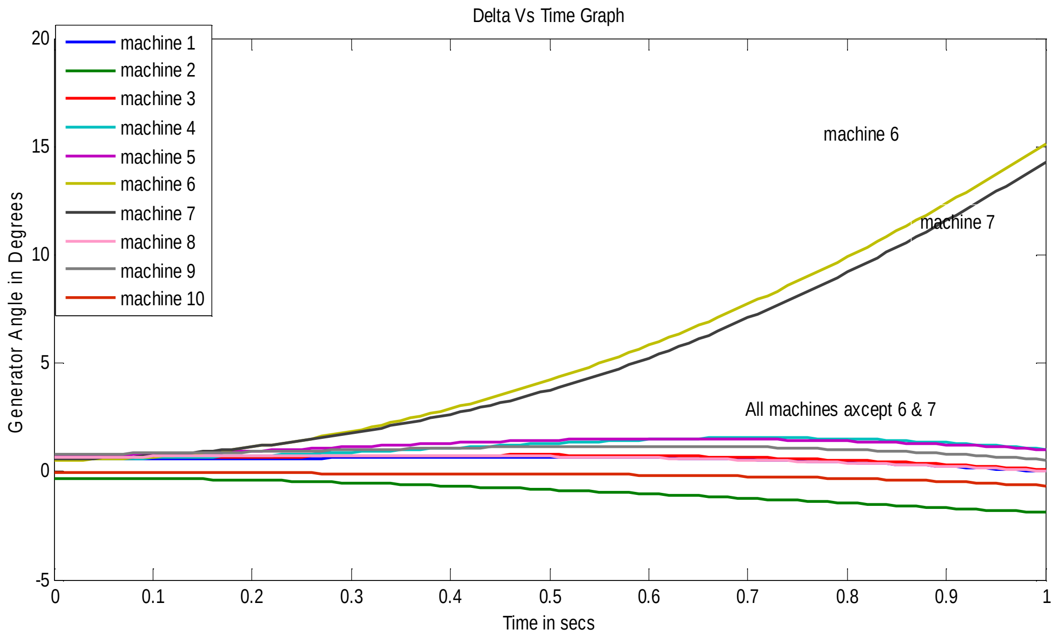


Fig 5.2 Swing curves for 10 generator 39 bus system indicating critical group of machines

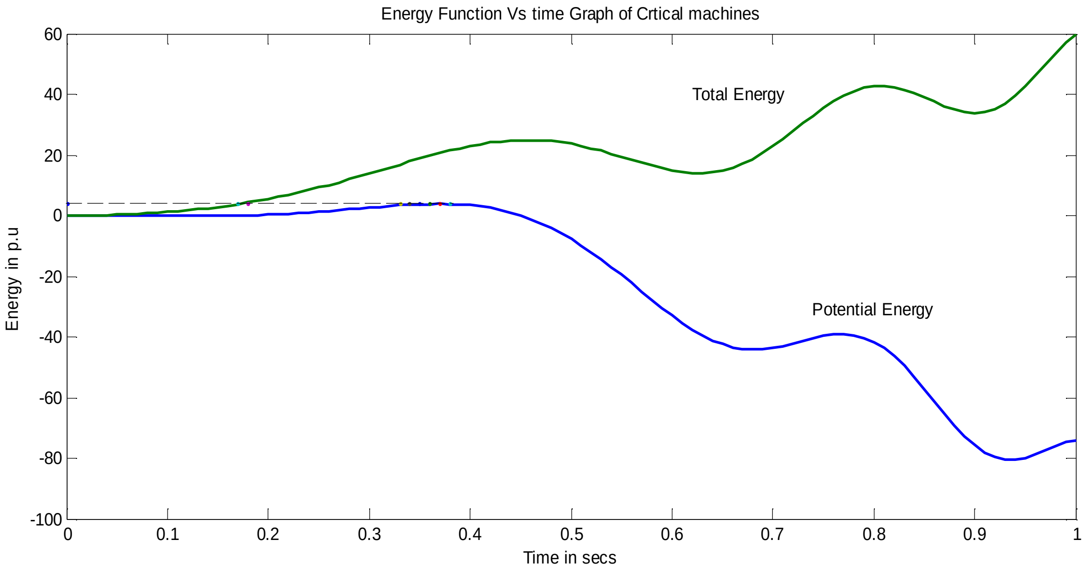


Fig 5.3 Plot of Energy functions for critical group of machines for 10 generator 39 bus system

All the 28 different fault cases considered earlier for TDS as well as TEF method has also been simulated here using critical machine energy function method and the critical clearing time has been obtained. In each case, the machines belonging to critical group has been identified by the procedure described earlier and hence the critical clearing time has been found out by using its energy functions. The results are presented in Table 5.1. It can be observed that the critical clearing time obtained by this method is more approximate to the values of time domain simulation compared to those obtained by transient energy function method. It can be observed from the simulation of critical group of machines that the instability is caused by the separation of one group of generator which accelerate from the rest of the generators which decelerate, but there may be certain exceptions to this [32].

TABLE 5.1 10 Generator 39 Bus System: CCTs obtained by Critical machine Energy Function

Sl No	Fault Bus#	Line Cleared	Critical Group Machines	CCT by Critical Group Energy Function (secs)
1	14	14 – 34	All except 2	0.32 – 0.33
2	16	16 – 31	1, 3	0.25 – 0.26
3	22	22 – 21	6,7	0.17 – 0.18
4	28	28 – 29	9	0.15 – 0.16
5	20	20 – 31	3	0.25 – 0.26
6	34	34 – 14	All except 2	0.34 – 0.35
7	31	31 – 16	1,3	0.28 – 0.29
8	26	26 – 25	9	0.18 – 0.19
9	29	29 – 28	9	0.11 – 0.12
10	31	31 – 20	1,3	0.29 – 0.30
11	21	21 – 22	6, 7	0.24 – 0.25
12	36	36 – 35	All except 2, 10	0.25 – 0.26
13	26	26 – 29	9	0.17 – 0.18
14	29	29 – 26	9	0.13 – 0.14
15	22	22 – 23	6, 7	0.25 – 0.26
16	23	23 – 22	7	0.23 – 0.24
17	18	18 – 15	All except 2	0.32 – 0.33
18	21	21 – 36	6,7	0.28 – 0.29
19	36	36 – 21	All except 2, 10	0.24 – 0.25
20	33	33 – 34	3	0.28 – 0.29
21	34	34 – 33	All except 2	0.32 – 0.33
22	31	31 – 38	All except 2	0.29 – 0.30
23	12	12 – 25	All except 2	0.35 – 0.36
24	25	25 – 26	All except 2, 10	0.17 – 0.18
25	25	25 – 12	All except 2, 10	0.14 – 0.15
26	35	35 – 36	All except 2, 10	0.43 – 0.44
27	12	12 – 11	All except 2, 10	0.35 – 0.36
28	38	38 – 13	All except 2, 10	0.45 – 0.46

CHAPTER 6

FIRST SWING STABILITY ANALYSIS OF A POWER SYSTEM INCLUDING SVC

It is always desirable to transmit as much power as possible through transmission lines, cables etc, unfailing the requirements of stability and security of supply. Major limiting factors in power transmission are thermal factors in cables, short transmission lines, generators etc, but in long lines and cables variations in voltage and maintenance of stability is the constrain. Both voltage profile and stability can be improved by using ‘reactive compensation’. In earlier days active and reactive compensation was carried out by regulating terminal voltages by generator excitation control, by transformer tap changers, by using fixed reactors, capacitors etc which by switched in and out by mechanical switchgears. Synchronous condensers and large generators were used in places where it was necessary to continuously vary reactive power. But since 1970, with the introduction and development of Power Electronics, power electronic equipments (building blocks of FACT devices) have become the leading players in the area of power compensation.

The FACTS is a concept based on power-electronic controllers, which enhance the value of transmission networks by increasing the use of their capacity [44]. They are increasingly being employed in power systems owing to their versatile advantages. The one important aspect of FACTS devices which is of interest in this project is its ability to enlarge the safe operating limits of a transmission system without risking stability.

6.1 STABILITY IMPROVEMENT BY FACTS DEVICES

The first swing limit of a single machine infinite bus system is usually determined by using equal area criterion(EAC). During the occurrence of fault, the output power of the machine reduces drastically while the input mechanical power remains almost constant causing the machine to accelerate. This is represented by the accelerating area in EAC. To maintain the first swing stability the machine must transfer the excess energy to the network once the fault is cleared. This excess energy transferring capability of the machine is represented by another area known as the decelerating area, which depends on the post fault network condition. By enlarging the decelerating area in the early part of post fault period, stability limits can be enhanced. This is possible only if the network conditions are controlled fast enough to enlarge the decelerating area dynamically. FACTS devices with the inherent feature of fast controls is capable of controlling the network conditions, so as to enhance the decelerating area and improve the first swing stability limit of a system [34] .

FACTS devices are based on their connection types are classified as – Series, Shunt and combined types. Series connected devices includes TCSC, SSSC etc, shunt controlled devices includes SVC, TCR, STATCOM etc and combined type includes – UPFC, TCPST (thyristor controlled phase shifting transformer) etc.

6.2 STATIC VAR COMPENSATOR

A Static Var Compensator is a first generation FACTS device. It is a shunt connected static generator and/or absorber of reactive power in which the output is varied to maintain or control the specific parameters of an electric power system [44]. Its basic elements are thyristor controlled reactor (TCR) and thyristor switched capacitor (TSC). The application of SVC was initially for load compensation of fast changing loads such as steel mills and arc furnaces. Here it was used for dynamic power factor improvement and to balance the current on source side whenever needed. Its application as transmission line compensators commenced in the late seventies. Here it is used mainly to [45]:

i. Increase power transfer in long lines
ii. Improve stability with fast acting voltage regulation
iii. Damp low frequency oscillations due to swing (rotor) modes
iv. Damp sub synchronous frequency oscillations due to torsional modes
v. Control dynamic overvoltages

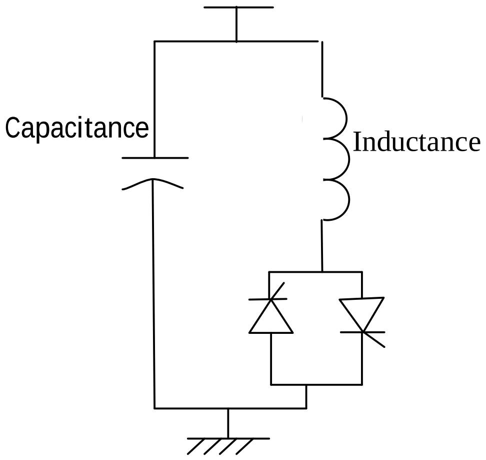


Fig 6.1 Static Var Compensator

For small disturbances, mostly a linear continuous control is sufficient to damp the oscillations [46] but for large disturbances, a bang-bang type control (BBC) becomes necessary to maintain stability and improve damping [33], [46]&[47]. In the transient stability analysis carried out here, discontinuous control (BBC) is used. In BBC, the mode of operation of the device is changed (from full capacitive to full inductive or vice versa) at some discrete points. The machine speed signal is used to change the mode of operation [46],[48] in BBC. The SVC control strategy used here is described by equation (52), [34]:

$$B_{SVC} = \begin{cases} B_{SVC}^{max} : if\ \omega \geq -\beta\ \omega_{max}\ during\ the\ first\ swing \\ k*\omega :\ B_{SVC}^{min} \leq k*\omega \leq B_{SVC}^{max}: in\ subsequent\ swings \end{cases} \tag{52}$$

Here ω_{max} is the maximum speed of the machine and it usually occurs at fault clearing and β is a small positive constant; k is a positive gain and its value depends on the SVC rating.

6.3 CASE STUDY

6.3.1 SMIB System –

A Generator connected to infinite bus through two transmission lines is considered. Corresponding data given in appendix. The system is simulated initially without SVC and its clearing time is obtained as 0.067s – 0.068s. Later on an SVC is connected at the bus at the centre of transmission line. Simulations are carried out for different values of susceptance of SVC and the corresponding critical clearing times were found out. The results are tabulated in Fig 6.1. it is found that on increasing the B_{SVC} injected by SVC critical clearing time is also improved. The limits of susceptance were taken as – Capacitive limit was taken as B_{SVC}^{max} and the inductive rating was taken as half of the capacitive rating ($B_{SVC}^{min} = -0.5B_{SVC}^{max}$). It was observed that SVC improved the CCT from 0.067s -0.068s to 0.151s - 0.152s. The above analysis was carried out by developing algorithm for the SVC control strategy in MATLAB.

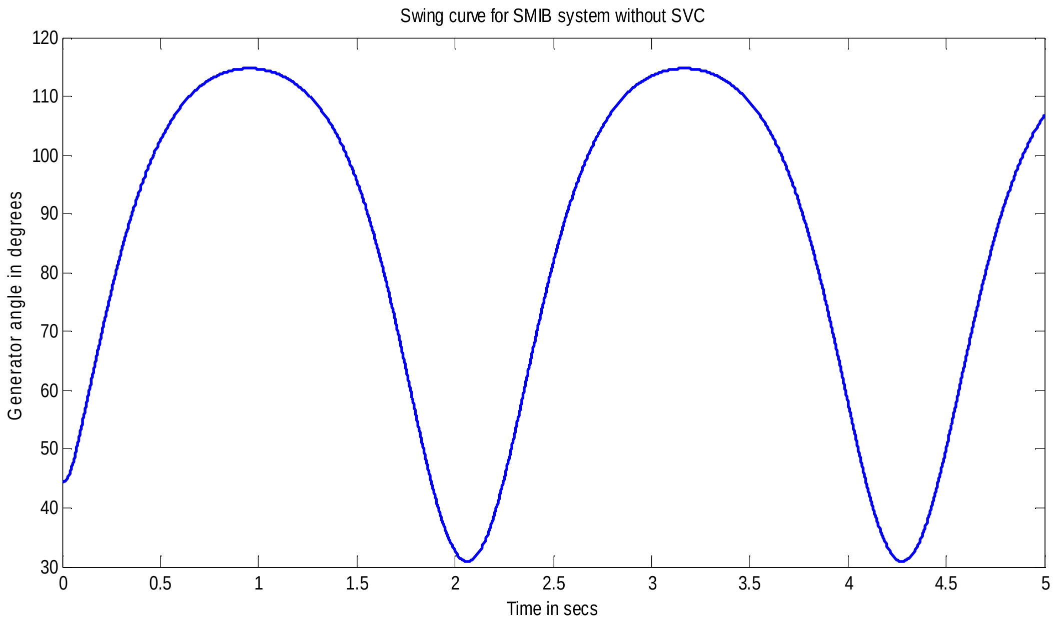


Fig 6.2 Swing curve for the SMIB system without SVC (t_{cr} = 67ms-68 ms)

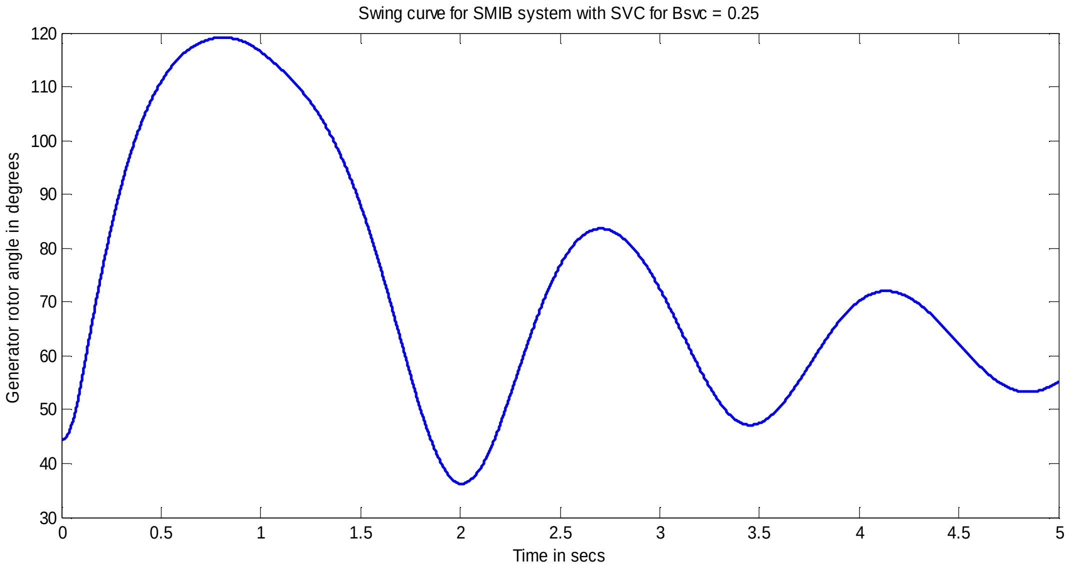


Fig 6.3 Swing curve for SMIB system with SVC (B_{SVC}^{max} = 0.25 p.u) and $t_{cr\,=}$ 0.096s – 0.097s

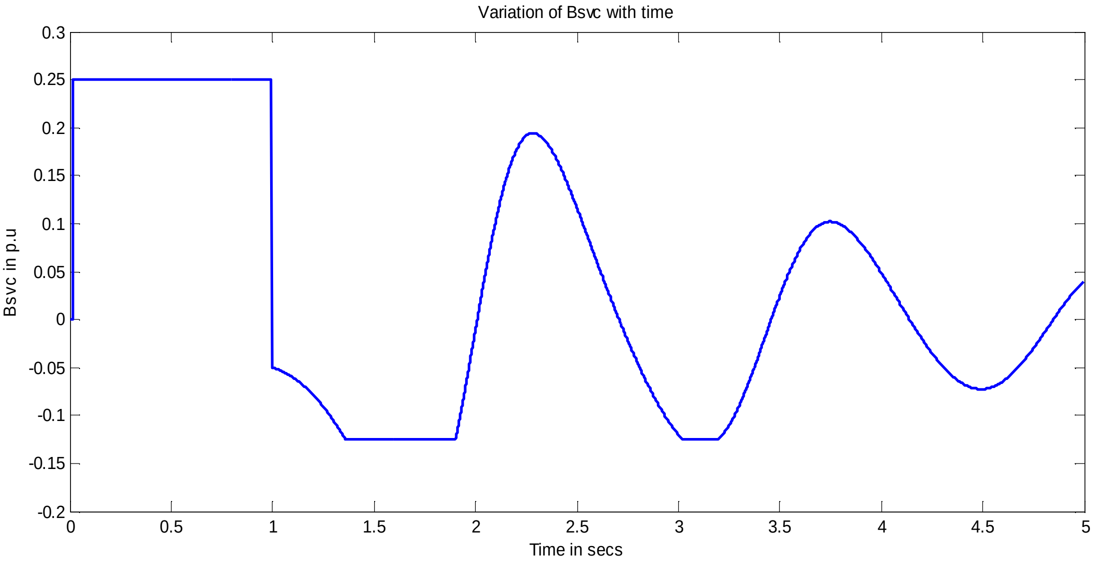


Fig 6.4 Variation of B_{SVC} with Time for B_{SVC}^{max} = 0.25 p.u ($t_{cr\,=}$ 0.096s – 0.097s)

TABLE 6.1 Clearing Time of SMIB system for various SVC ratings

Sl No	B_{SVC}^{max} in p.u	Critical Clearing Time (ms)
1	0	0.067 – 0.068
2	0.25	0.096 – 0.097
3	0.50	0.119 – 0.120
4	0.75	0.136 – 0.137
5	1	0.151 – 0.152

6.3.2 3 Generator 9 Bus System –

3 machine 9 bus system also known as WSCC (Western System Coordinating Council) is used to show the effect of SVC on transient stability analysis of multi-machine power system. The single line diagram and data are give in appendix. Here three phase fault was assumed at various locations , SVC was placed randomly at certain buses and system was simulated (fault was cleared during simulation). Simulations were carried out by assuming generator 1 on bus #1 as the reference machine. Simulations were carried out using DIGSILENT POWERFACTORY software and the corresponding graphs have been obtained. The graphs shown here are for a three phase fault at Bus No# 8, which is cleared by tripping a line connected to the faulted bus. The swing curve for 9 Bus system without any SVC is shown in Fig 6.5. Then SVC is placed at Bus No# 9 and swing curves for stable and unstable cases in shown in Fig 6.6 and Fig 6.7 respectively.

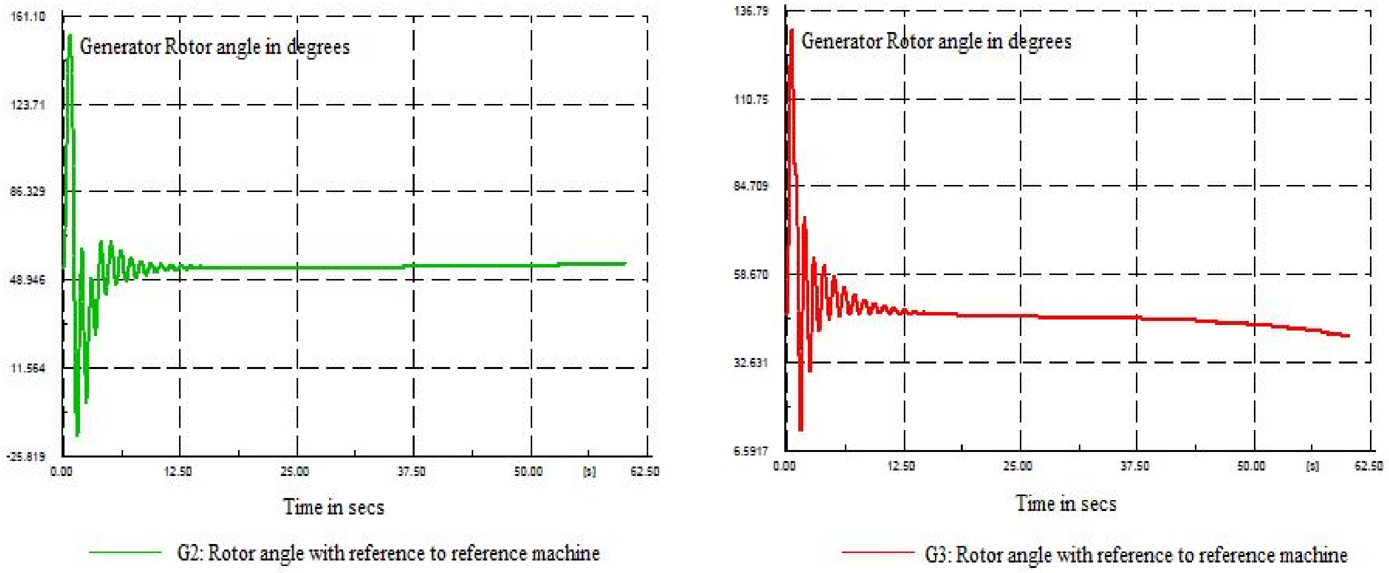


Fig 6.5 Swing curves for 3 generator 9 Bus system without SVC

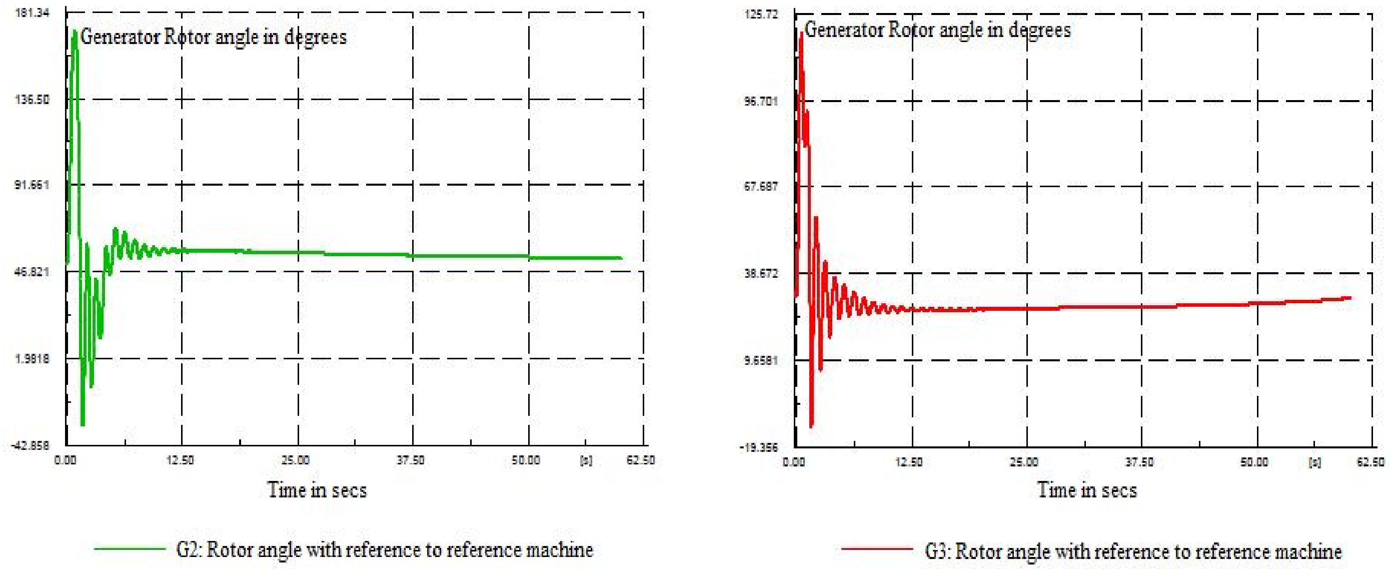


Fig 6.6 Swing Curves for 3 generator 9 Bus system, Fault on Bus 8, SVC at Bus 9, B_{SVC}^{max} = 0.25 p.u, t$_c$ = 0.36s

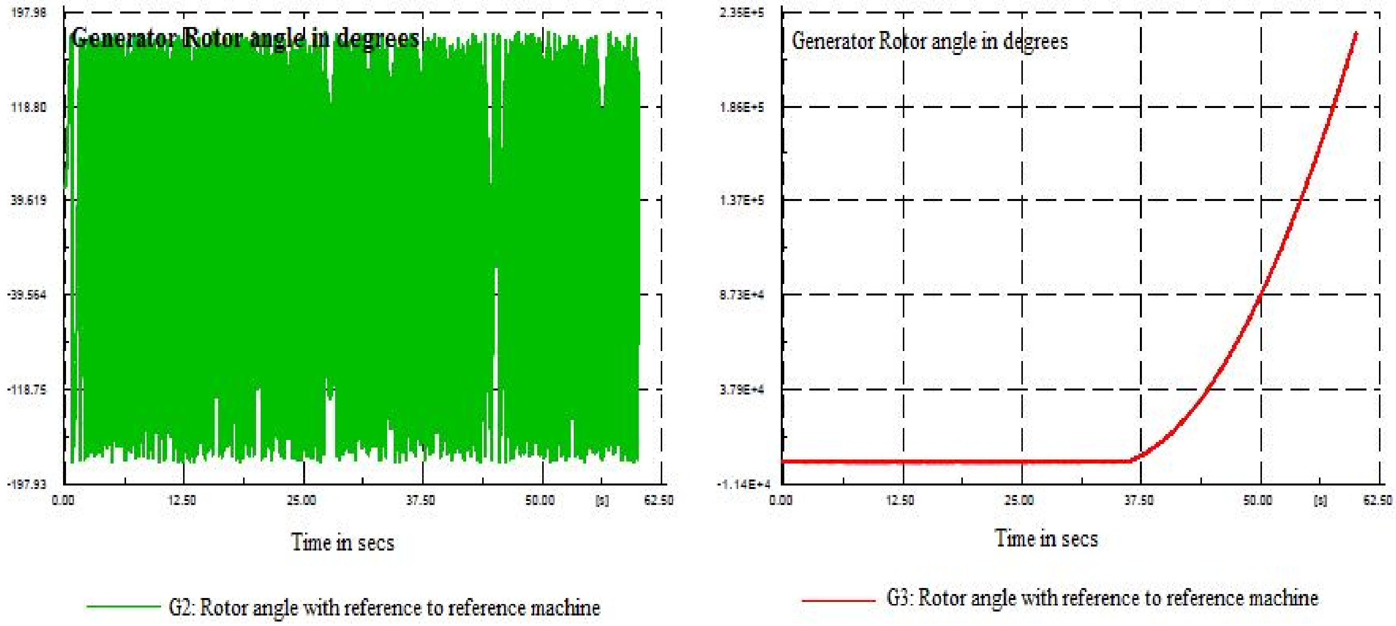


Fig 6.7 Swing Curves for 3 generator 9 Bus system for fault on Bus 8, SVC at Bus 9, B_{SVC}^{max} = 0.25 p.u, t$_c$ = 0.37s

It can be seen in general that an SVC placed in main power transfer path usually plays an important role in improving stability limit and hence critical clearing time. Simulation results for faults on other buses and for different SVC locations is tabulated in Table 6.2

TABLE 6.2 Critical Clearing Time of 3 generator 9 Bus system with various locations of SVC

Sl No	Fault Bus#	CCT without SVC (secs)	SVC location Bus#	CCT with SVC (secs)
1	5	0.23 – 0.24		
			7	0.27 – 0.28
			5	0.22 – 0.23
			4	0.24 – 0.25
			6	0.23 – 0.24
			9	0.27 – 0.28
			8	0.23 – 0.24
2	6	0.38 – 0.39		
			7	0.46 – 0.47
			5	0.38 – 0.39
			4	0.41 – 0.42
			6	0.39 – 0.40
			9	0.50 – 0.51
			8	0.40 – 0.41
3	7	0.23 – 0.24		
			7	0.25 – 0.26
			5	0.23 – 0.24
			4	0.24 – 0.25
			6	0.23 – 0.24
			9	0.27 – 0.28
			8	0.24 – 0.25
4	8	0.29 – 0.30		
			7	0.35 – 0.36
			5	0.29 – 0.30
			4	0.32 – 0.33
			6	0.30 – 0.31
			9	0.36 – 0.37
			8	0.31 – 0.32
5	9	0.22 – 0.23		
			7	0.25 – 0.26
			5	0.22 – 0.23
			4	0.24 – 0.25
			6	0.23 – 0.24
			9	0.27 – 0.28
			8	0.23 – 0.24
6	4	0.28 – 0.29		
			7	0.34 – 0.35
			5	0.28 – 0.29
			4	0.30 – 0.31
			6	0..29 – 0.30
			9	0.35 – 0.36
			8	0.29 – 0.30

CHAPTER 7
RESULTS AND CONCLUSION

This project carried out with the aim of exploring various facets of transient stability analysis of a multi-machine power system bestowed the following conclusions.

7.1 CONCLUSIONS

Initially stability analysis was carried out on an SMIB system using MATLAB SIMULINK and the concept was further extended to multi-machine power system (10 Generator New England System) by developing algorithm in MATLAB. Results of Time Domain simulation (which is the conventional, indirect method of stability analysis) serves as the point of reference. The salient feature of the simulation carried out is the use of detailed generator modelling which is a very much valid modelling in practical power systems. TDS is simple, it has the flexibility to include any degree of machine modelling but becomes prohibitive at times in case of large power system owing to its time consuming and tedious nature of calculations.

To overcome the disadvantages of TDS, subsequently Direct methods of stability analysis are used. Transient energy function method, which uses the energy functions of all the machines of the system for CCT calculation, was found to be faster than TDS, and the results obtained by this method are within acceptable accuracy limit. Another direct stability method, a modification of TEF, which uses the energy functions of certain critical machines that mostly affect the stability of the system is then expounded. Here the critical group of machines, which usually causes instability by accelerating from rest of machines were identified for different fault location and then an algorithm to determine critical clearing time based on the energy functions of critical group of machines was developed in MATLAB. The results of CCT obtained by this method is then compared with those of TDS and TEF methods. Critical machine energy function outweighs the TEF method in many aspects. It is faster than TEF and the results obtained were more closer to those of the benchmark results of TDS. This can be verified from Table 7.1. This affirms the use of Critical Group Energy Function method as a more reliable, faster method of stability analysis for online dynamic security assessment of realistic power systems. The results obtained can be applied to practical power systems with little modifications to include tap changing transformers and other power system equipments.

TABLE 7.1 Comparison of CCTs of Transient Simulation and Energy Function method

Sl No	Fault Bus#	Line cleared	CCT by TDS (secs)	CCT by TEF (secs)	CCT from Critical M/c Energy Function
1.	14	14 – 34	0.34 – 0.35	0.32 – 0.33	0.32 – 0.33
2.	16	16 – 31	0.25 – 0.26	0.22 - 0.23	0.25 – 0.26
3.	28	28 – 29	0.15 – 0.16	0.14 – 0.15	0.15 – 0.16
4.	20	20 – 31	0.26 – 0.27	0.24 – 0.25	0.25 – 0.26
5.	22	22 – 21	0.20 – 0.21	0.16 – 0.17	0.17 – 0.18
6.	34	34 – 14	0.35 – 0.36	0.32 – 0.33	0.34 – 0.35
7.	31	31 – 16	0.30 – 0.31	0.26 – 0.27	0.28 – 0.29
8.	21	21 – 22	0.25 – 0.26	0.23 – 0.24	0.24 – 0.25
9.	26	26 - 25	0.19 – 0.20	0.18 – 0.19	0.18 – 0.19
10.	31	31 – 20	0.29 – 0.30	0.26 - 0.27	0.29 – 0.30
11.	36	36 – 35	0.25 – 0.26	0.23 – 0.24	0.25 – 0.26
12.	29	29 – 28	0.12 – 0.13	0.10 – 0.11	0.11 – 0.12
13.	26	26 – 29	0.17 – 0.18	0.17 – 0.18	0.17 – 0.18
14.	29	29 – 26	0.14 – 0.15	0.12 – 0.13	0.13 – 0.14
15.	22	22 – 23	0.27 – 0.28	0.27 – 0.28	0.25 – 0.26
16.	23	23 – 22	0.25 – 0.26	0.22 – 0.23	0.23 – 0.24
17.	18	18 – 15	0.34 – 0.35	0.30 – 0.31	0.32 – 0.33
18.	21	21 – 36	0.30 – 0.31	0.27 – 0.28	0.28 – 0.29
19.	36	36 – 21	0.25 – 0.26	0.22 – 0.23	0.24 – 0.25
20.	33	33 – 34	0.29 – 0.30	0.26 – 0.27	0.28 – 0.29
21.	34	34 – 33	0.33 – 0.34	0.31 – 0.32	0.32 – 0.33
22.	31	31 – 38	0.30 – 0.31	0.26 – 0.27	0.29 – 0.30
23.	12	12 – 25	0.34 – 0.35	0.34 – 0.35	0.35 – 0.36
24.	25	25 – 26	0.21 – 0.22	0.19 – 0.20	0.17 – 0.18
25.	25	25 – 12	0.19 – 0.20	0.19 – 0.20	0.14 – 0.15
26.	35	35 – 36	0.39 – 0.40	0.40 – 0.41	0.43 – 0.44
27.	12	12 – 11	0.29 – 0.30	0.32 – 0.33	0. 35 – 0.36
28.	38	38 – 13	0.41 – 0.42	0.42 – 0.43	0.45 – 0.46

In current scenario where power systems are transforming into complex networks with faster control and bidirectional power flows, transient stability analysis considering FACTS devices is the need of the hour. Transient stability analysis was carried out on SMIB system after connecting an SVC. It was found that with different ratings of susceptance injected by the device, the critical clearing time also varies and it increases with an increase in the susceptance value. Later on SVC was included in multi-machine system (3 machine WSCC system) and transient stability analysis was carried out by varying the location of SVC, using DIGSILENT POWERFACTORY software. It was found that in the 3 machine 9 bus system, buses #7 and #9 are of strategic importance. Whenever an SVC is connected on any of these buses, the critical clearing time was observed to be highest, especially with SVC on bus #9. This can be attributed to the fact that the generator connected to bus # 9 is having lower inertia than that connected to bus #7. Hence, without SVC machine connected to bus #9 has a greater tendency of going out of step. It was also observed that with increase in SVC rating, the CCT is increasing. But the results obtained are of preliminary nature and needs further investigation.

7.2 FUTURE SCOPE

The transient stability analysis has been an indispensible part of power system dynamics since time immemorial and will continue to be the same. With introduction of newer control techniques, power electronic devices etc the system is operating more closer to its stability limits. Some of the improvements that can be carried out further in this area are as follows:

1. The results of direct stability methods presented in this project needs further investigation and testing on large realistic power systems. The study of three phase fault carried out here can further be tested for other types of faults and contingencies.
2. The load model assumed here was of constant impedance type. Stability studies by considering dynamic load models can also be performed.
3. The energy function analysis which was carried out for multi-machine power system can be extended to power systems with different types of HVDC and FACTS controllers like SVC, TCSC, UPFC etc.
4. Since Distributed generations are currently becoming an integral part of newer power systems, stability analysis including the effects of DG will provide a better insight into the stability aspects of those power systems.
5. For determination of the most optimum location and rating of SVC’s, optimisation techniques like GA, PSO etc can be used.

REFERENCES

[1] K. R. Padiyar, *Power System Dynamics Stability and Control*, BS Publications, Hyderabad, 2nd edition, 2008.

[2] H. H. A. Mahroon, I. Leevongwat, P. Rastgoufard, " A practical method for power systems transient stability and security analysis", *IEEE PES*, T & D, pp:1-6, 2012.

[3] D.C.Hsiao, C.C.Chia, G.Cauley, ' Direct stability analysis of electric power systems using energy functions: theory, applications, and perspective', *IEEE Proc*, Vol:83, Issue:11, pp 1497 – 1529, 1995.

[4] N. Michel, A. A. Foud, V. Vittal, " Power system transient stability using individual machine energy functions" , *IEEE Transactions on Circuits and Systems*, Vol.30, no.5, pp. 266- 276, May 1983.

[5] P. Kundur, *Power System Stability and control,* Tata McGraw Hill publications, New Delhi, 2007.

[6] J. J. Grainger, W.D. Stevenson Jr., *Power System Analysis* , Tata McGraw Hill publications, New Delhi, 2003.

[7] Peter W. Sauer, M. A. Pai, *Power System dynamics and stability*, Prentice Hall publications, 1997.

[8] X. F. Wang, Y. Song, M. Irving, *Modern Power System Analysis*, Springer Science Publications, 2008.

[9] CIGREE working group 32-03, " Tentative classification and terminologies relating to stability problems of power system", *Electra* No 56, 1978.

[10] IEEE Task Force, "Proposed Terms and Definitions for Power System Stability", *IEEE Trans* Vol. PAS-101, pp.1894-1898, July 1982.

[11] H. F. William, S.A. Louie, G. W. Wills, "Feasibility of Lyapunov function for stability analysis of electric power systems having upto 60 generators", *IEEE trans*, Vol PAS-91, pp- 1145-1153, 1972.

[12] G. A. Luders, "Transient stability of multi-machine power systems via direct method of Lyapunov", *IEEE trans,* Vol. PAS-90,no.1, pp.23-36, 1971.

[13] N. Kakimoto, M. Hayashi, " Transient stability analysis of multi-machine power system by Lyapunov's direct method", IEEE Trans on PAS, Vol. 96, pp.276-284, 1977.

[14] M. Lyapunov, the general problem of the stability of the motion, The mathematical society of Kharkov,Russia 1892. English translation, Taylor and Francis Ltd, London, 1992.

[15] P. C. Magnusson, "Transient Energy method of calculating stability", *AIEEE trans.,* 66, pp-747-755, 1947

[16] P. D. Aylett, " The Energy-Integral criterion of transient stability limits of power systems", Proc. Of the institution of Electrical Engineers (London), 105C, pp, 257-536, Sept 1958.

[17] H. El-Abaid and K. Nagappan, " Transient Stability Region of multi-machine power systems", *IEEE Trans. Power Appar. Syst, PAS-85, 2,* pp-169-178, Feb 1987.

[18] M. A. Pai, *Energy function analysis for power system stability,* Kluwer Academic Publishers, 1989.

[19] A. A. Foud, V. Vittal, *Power system transient stability analysis using the transient energy function method,* Prentice-Hall, 1992.

[20] M. H. Haque, " Equal- area criterion: An extension for multi-machine power systems", *IEEE Proc Trans Distrib,* Vol. 141, no 3, May 1994.

[21] H. D. Chiang, F. F. Wu, P. P. Varaiya, C. W. Tan, " Theory of the Potential Energy boundary Surface", IEEE Proc. On Decision and Control, Dec 1985.

[22] Y. Xue, Th.V. Cutsem, R. Pavella, " Extended Equal Area criterion Justifications, Generalizations, Applications", IEEE trans, PWRS-4, pp-44-52, 1989.

[23] Y. Xue, L. Wehenkel, R. Belhomme, R. Rousseaux, P. Pavella, M.Euxible, E. Heilbronn, J. F. Lesigne, " Extended Equal Area criteria revisited", IEEE trans, PWRS-7(3), pp-1012-1022, 1992.

[24] W. W. Lemmon, K. R. C. Mammandur, W. R. Barcello, " Transient stability prediction and control in real-time by QUEP", IEEE trans, PWRS-4(2), pp- 627-642, 1989

[25] P. Rastgoufard, A. Yazdankhah, R. A. Schluester, " Multi-machine equal area based power system transient stability measure", IEEE trans, PWRS-3(1), pp-188-196, 1989.

[26] M. A. Pai, K. R. Padiyar, C. Padhakrishna, " Transient stability analysis of multimahcine AC/DC power systems via Energy function method", IEEE trans, Vol, PAS-100, No-12, Dec 1981.

[27] M. H. Haque, A. H. M. A. Rahim, "Determination of first swing stability limit of multi-machine power systems through Taylor series expansions", IEEE Proc,C, pp-373-379, 1989

[28] Y. Dong, H. R. Pota, "Fast transient stability assessment using large step-size numerical integration", IEEE Proc C, 138(4), pp-377-383,1991

[29] V. Machias, J. L. Souflis, "Unstable machine identification in power system transient stability studies, ETZ Arch, pp-49-51, 1987

[30] A. Priyadi, N. Yorino, M. Eghbal, Y. Zoka, Y. Sasaki, H. Yasuda, and H. Kakui, "Transient stability assessment as boundary value problem," Electric Power Conference, 2008. EPEC 2008. IEEE Canada , vol., no., pp.1-6, 6-7 Oct. 2008.

[31] A. Priyadi, N. Yorino, D. K. Jha, Y. Zoka, Y. Sasaki, H. Yasuda, and H. Kakui, "Critical Trajectory Method for Transient Stability Analysis," Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on , vol., no., pp.291-295, 20-22 Dec. 2008.

[32] R. M. Iqbal , S. Krishna, "Online dynamic security assessment of power system using critical group energy functions", *IEEE Power India Conf*, pp.1-6, 2012.

[33] M. H. Haque, ."Application of energy function to assess the first-swing stability of a power system with a SVC," *Generation, Transmission and Distribution, IEE Proceedings-* , vol.152, no.6, pp. 806- 812, 4 Nov. 2005.

[34] M. H. Haque, " Improvement of first swing stability limit by utilizing full benefit of shunt FACTS devices" *IEEE trans, PAS,* Vol.19, No.4, Nov 2004.

[35] V. Azbe, U. Gabrijel, D. Povh, R. Mihalic, " The Energy Funciton of a general multi-machine system with Unified Power Flow Controller*", IEEE trans PAS*, Vol.20, No.3, August 2005.

[36] P. Vaidya, R. V. Patil, " Comparison of different types of transient stability assessments during congestion", *IEEE ICAESM,* pp.232-237, March 2012.

[37] A.A.Foud, S.E Stanton, “Transient Stability of a Multi-Machine Power System Part I: Investigation of System Trajectories”, *IEEE Transactions PAS*, Vol: PAS-100 , Issue: 7, pp-3408 – 3416, 1981.

[38] A. A. Foud, “ Transient stability margin as a tool for dynamic security assessment”, EPRI report EL-1755, Mar 1981.

[39] W. P. Yuan, K. W. Chan, Y. Zhang, “ Identification of Critical Cluster in Transient Stability Study Using Line Potential Energy Method”, *IEEE Power Engineering Society meeting*, 2006.

[40] K.W. Chan, Q. Zhou, T.S. Chung, “Transient Stability Margin Assessment for Large Power System Using Time Domain Simulation Based on Hybrid Extended Equal Area Criterion Method”, *APSCOM’2000*, pp. 405-409, 2000.

[41] A.R. Bergen, D.J. Hill, “A Structure Preserving Model for Power System Stability Analysis”, *IEEE Transaction on Power Apparatus and System* Vol. PAS-100, No.1, pp. 25-35, January, 1981.

[42] G. Mu, G.W. Cai, Z. Hu, Z.H. Wang, “Network-machine Combined Transient Stability Assessment Method-Key Cutsets Method”, Journal of Tsinghua University (Sci.&Tech.) Vol.7, pp.97-101, 1997.

[43] Y. Xue, M. Pavella, “Critical-cluster Identification in Transient Stability Studies” , *IEE Proc.* Vol. 140, No. 6, pp. 481-489, Nov. 1993.

[44] R. M. Mathur, R. K. Varma, *Thyristor based FACTS Controllers for Electrical Transmission System*, John Wiley & Sons Publicaions, 2002.

[45] K. R. Padiyar, *FACTS Controllers in Power Transmission and Distribution,* New Age International Publishers, New Delhi, 2007.

[46] E.Z. Zhou, “Application of static var compensators to increase power system damping”, IEEE Trans. Power Syst., 1993, 8, (2), pp. 655–661

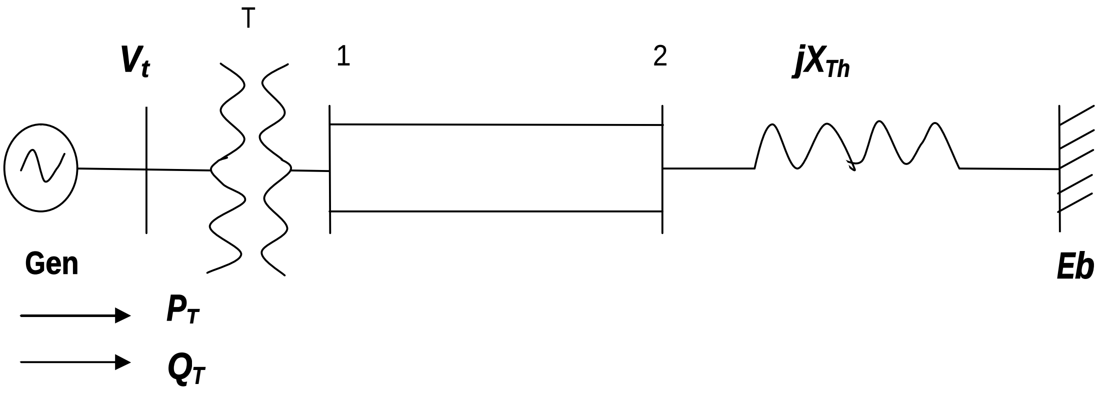
[47] E. Lerch, D. Povh, L. Xu, “Advanced SVC control for damping power system oscillations”, IEEE Trans. Power Syst., 1991, 6, (2),pp. 524–531.

[48] N. G. Hingorani, L. Gyugyi, *Understanding FACTS: Concepts and Technology of Flexible ac Transmission Systems*. New York, IEEE Press, 1999.

APPENDIX A

SINGLE MACHINE INFINITE BUS SYSTEM

A.1 SINGLE LINE DIAGRAM



A.2 SYSTEM DATA

Generator : R_{ai} = 0.00327, x_d = 1.7572, x_q = 1.5846, x'_d = 0.4245, x'_q= 1.04, T'_{do} = 6.66, T'_{qo}= 0.44, H = 3.542, f_b = 50 Hz.

Transformer: R_t = 0.0, x_t = 0.1364

Transmission Line: (per circuit) R_l = 0.08593, x_l = 0.8125, B_c = 0.1184

(These parameters are representative of a 400 kV, 400km long line with 50% shunt compensation)

Excitation System : Static Exciter with single time constant AVR is used

T_A = 0.025, K_A = 400, E_{fd}^{max} = 6.0, E_{fd}^{min} = -6.0.

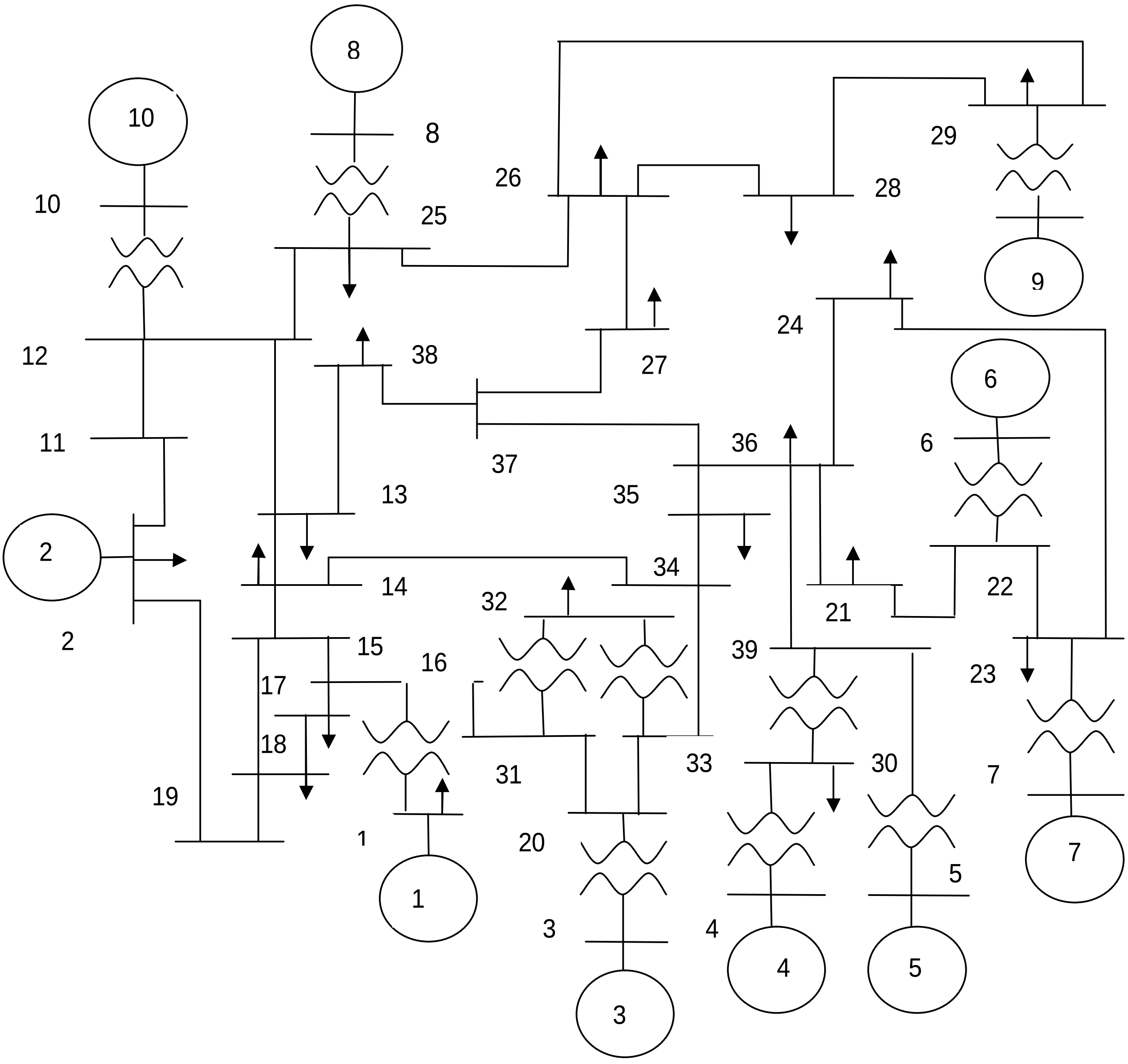
Operating Data : E_b =1.0, P_t = 0.6, Q_t = 0.02224, V_t = 1.05, θ = 21.65, X_{th} = 0.13636.

All Data are on a 1000 MVA base.

APPENDIX B

39 BUS 10 GENERATOR NEW ENGLAND SYSTEM

B.1 SINGLE LINE DIAGRAM



B.2 LINE DATA

Bus # From	Bus # To	R_l	X_l	B_c
37	27	0.0013	0.0173	0.3216
37	38	0.0007	0.0082	0.1319
36	24	0.0003	0.0059	0.068
36	21	0.0008	0.0135	0.2548
36	39	0.0016	0.0195	0.304
36	37	0.0007	0.0089	0.1342
35	36	0.0009	0.0094	0.171
34	35	0.0018	0.0217	0.366
33	34	0.0009	0.0101	0.1723
28	29	0.0014	0.0151	0.249
26	29	0.0057	0.0625	1.029
26	28	0.0043	0.0474	0.7802
26	27	0.0014	0.0147	0.2396
25	26	0.0032	0.0323	0.513
23	24	0.0022	0.035	0.361
22	23	0.0006	0.0096	0.1846
21	22	0.0008	0.0135	0.2548
20	33	0.0004	0.0043	0.0729
20	31	0.0004	0.0043	0.0729
19	2	0.001	0.025	1.2
18	19	0.0023	0.0363	0.3804
17	18	0.0004	0.0046	0.078
16	31	0.0007	0.0082	0.1389
16	17	0.0006	0.0092	0.113
15	18	0.0008	0.0112	0.1476
15	16	0.0002	0.0026	0.0434
14	34	0.0008	0.0129	0.1382
14	15	0.0008	0.0128	0.1342
13	38	0.0011	0.0133	0.2138
13	14	0.0013	0.0213	0.2214
12	25	0.007	0.0086	0.146
12	13	0.0013	0.0151	0.2572
11	12	0.0035	0.0411	0.6987
11	2	0.001	0.025	0.75

B.3 MACHINE DATA

Gen #	R_a	X_d	X_d’	X_q	X_q’	H	T_{do}’	T_{qo}’	T_c	D
1	0	0.295	0.0647	0.282	0.0647	30.3	6.56	1.5	0.01	0
2	0	0.02	0.006	0.019	0.006	500	6.0	0.7	0.01	0
3	0	0.2495	0.0531	0.237	0.0531	35.8	5.7	1.5	0.01	0
4	0	0.33	0.066	0.31	0.066	26	5.4	0.44	0.01	0
5	0	0.262	0.0436	0.258	0.0436	28.6	5.69	1.5	0.01	0
6	0	0.254	0.05	0.241	0.05	34.8	7.3	0.4	0.01	0
7	0	0.295	0.049	0.292	0.049	26.4	5.66	1.5	0.01	0
8	0	0.29	0.057	0.28	0.057	24.3	6.7	0.41	0.01	0
9	0	0.2106	0.057	0.205	0.057	34.5	4.79	1.96	0.01	0
10	0	0.2	0.004	0.190	0.004	42	5.7	0.5	0.01	0

B.4 TRANSFORMER DATA

Bus From #	Bus To #	R_t	X_t	Tap
39	30	0.0007	0.0138	1
39	5	0.0007	0.0142	1
32	33	0.0016	0.0435	1
32	31	0.0016	0.0435	1
30	4	0.0009	0.018	1
29	9	0.0008	0.0156	1
25	8	0.0006	0.0232	1
23	7	0.0005	0.0272	1
22	6	0	0.0143	1
20	3	0	0.02	1
16	1	0	0.025	1
12	10	0	0.0181	1

B.5 LOAD FLOW DATA

Bus #	V	φ	P_G	Q_G	P_L	Q_L
1	0.9820	0	5.5202	1.6184	0.0920	0.0460
2	1.0300	-10.9695	10.000	2.2624	11.0400	2.5000
3	0.9831	2.3411	6.5000	1.6606	0	0
4	1.0123	3.1662	5.0800	1.5510	0	0
5	0.9972	4.1895	6.3200	0.8381	0	0
6	1.0493	5.1982	6.5000	2.8105	0	0
7	1.0635	7.9916	5.6000	2.2967	0	0
8	1.0278	1.8417	5.4000	0.2757	0	0
9	1.0265	7.5440	8.3000	0.5970	0	0
10	1.0475	-4.0075	2.5000	1.8388	0	0
11	1.0350	-9.3204	0	0	0	0
12	1.0166	-6.4428	0	0	0	0
13	0.9855	-9.4407	0	0	3.2200	0.0240
14	0.9498	-10.3723	0	0	5.0000	1.8400
15	0.9506	-9.1191	0	0	0	0
16	0.9521	-8.3461	0	0	0	0
17	0.9441	-10.8026	0	0	2.3380	1.8400
18	0.9448	-11.3656	0	0	5.2200	1.7600
19	1.0071	-11.1839	0	0	0	0
20	0.9585	-5.5889	0	0	0	0
21	0.9849	-4.3401	0	0	2.7400	1.1500
22	1.0149	0.1908	0	0	0	0
23	1.0122	-0.0816	0	0	2.7450	0.8466
24	0.9735	-6.8016	0	0	3.0860	0.9220
25	1.0257	-4.9744	0	0	2.2400	0.4720
26	1.0124	-6.2086	0	0	1.3900	0.1700
27	0.9918	-8.3297	0	0	2.8100	0.7550
28	1.0164	-2.4678	0	0	2.0600	0.2760
29	1.0187	0.4580	0	0	2.8350	0.2690
30	0.9842	-2.0190	0	0	6.2800	1 .0300
31	0.9549	-6.5292	0	0	0	0
32	0.9349	-6.5128	0	0	0.0750	0.8800
33	0.9560	-6.3776	0	0	0	0

34	0.9546	-8.2165	0	0	0	0
35	0.9568	-8.5349	0	0	3.2000	1.5300
36	0.9737	-6.8917	0	0	3.2940	0.3230
37	0.9814	-8.1014	0	0	0	0
38	0.9815	-9.0864	0	0	1.5800	0.3000
39	0.9849	-1.0189	0	0	0	0

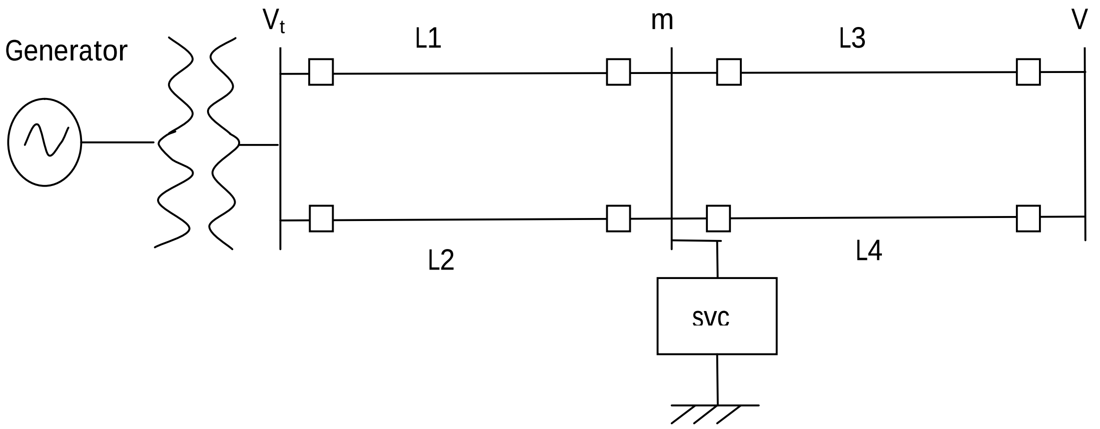
B.6 AVR Data

All machines are provided with single time constant AVR with K_A =25, T_A = 0.025, $E_{fd}^{min} = -10$, $E_{fd}^{max} = 10$

APPENDIX C

SMIB SYSTEM WITH SVC

C.1 SINGLE LINE DIAGRAM



C.2 DATA :

Generator : H = 5s, f = 60 Hz, X' = 0.3 *p.u,*

Transformer : X = 0.1 *p.u*

Transmission Line : X = 0.4 *p.u* of each line,

Prefault Power delivered by machine = 1 *p.u,*

Terminal Voltage : V_t = 1.0 *p.u,*

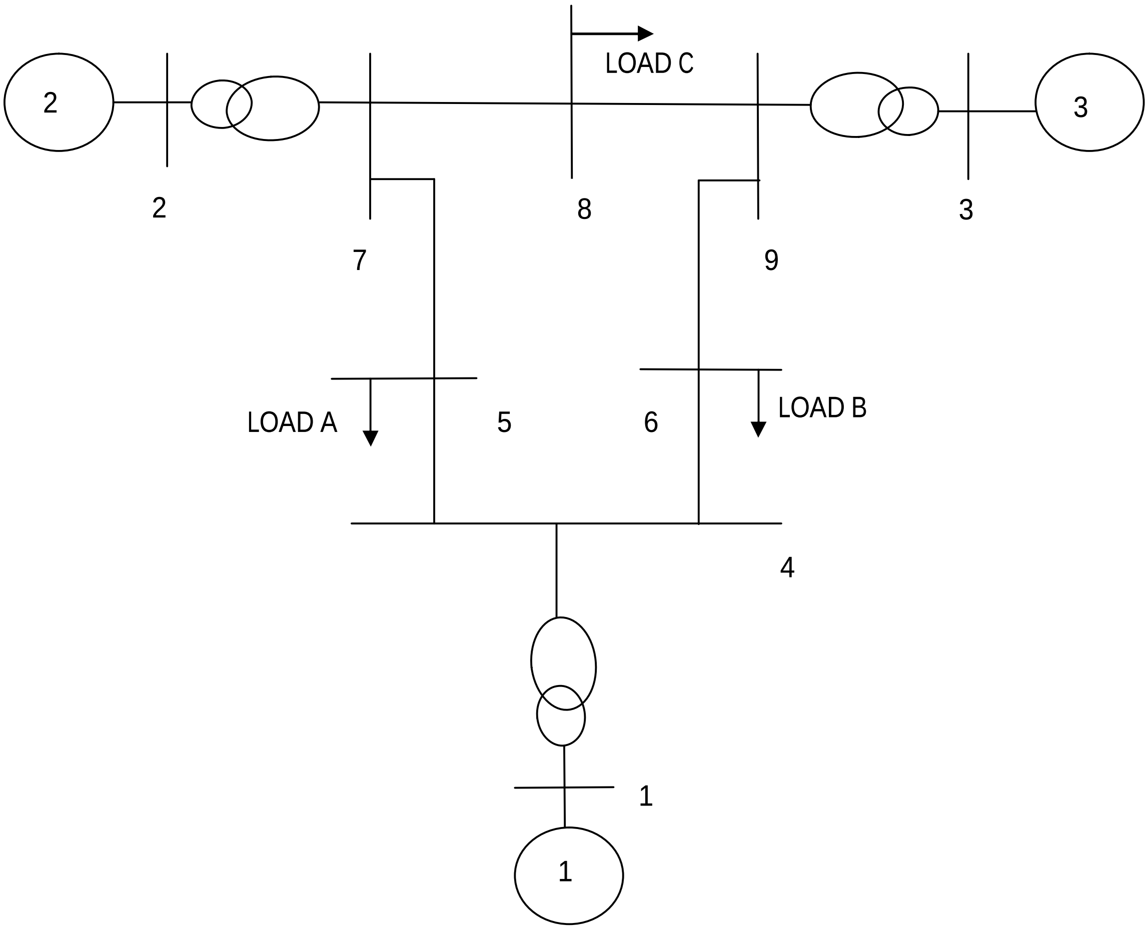
Infinite Bus Voltage : V = 0.95 *p.u,*

Prefault internal Voltage of machine for operating point : 0.864 + j0.842

APPENDIX D

9 BUS 3 MACHINE WSCC SYSTEM

D.1 SINGLE LINE DIAGRAM



D.2 MACHINE DATA

Generator	1	2	3
Rated MVA	247.5	192.0	128.0
kV	16.5	18.0	13.8
Power factor	1.0	0.85	0.85
Type	hydro	steam	steam
Speed (r/min)	180	3600	3600
X_d(p.u)	0.1460	0.8958	1.3125
X_d'(p.u)	0.0608	0.1198	0.1813
X_q(p.u)	0.0969	0.8645	1.2578
X_q'(p.u)	0.0969	0.1969	0.25
X_l(p.u)	0.0336	0.0521	0.0742
T_{do}'(secs)	8.96	6.00	5.89
T_{qo}'(secs)	0	0.535	0.600
M(p.u)	0.0507	0.0590	0.0125

D.3 LINE DATA

Bus From #	Bus To #	R_l	X_l	B_c
7	8	0.0085	0.0720	0.0745
8	9	0.0119	0.1008	0.1045
7	5	0.0320	0.1610	0.1530
9	6	0.0390	0.1700	0.1790
5	4	0.0100	0.0850	0.0880
4	6	0.0170	0.0920	0.0790
2	7	0.0000	0.0625	1.0000
9	3	0.0000	0.0586	1.0000

D.4 LOAD DATA

Parameter	LOAD A	LOAD B	LOAD C
P_l(MW)	125	90	100
Q_l(MVar)	50	30	35

APPENDIX E

E.1 MATLAB PROGRAM FOR TIME DOMAIN SIMULATION OF IEEE 39 BUS 10 GENERATOR SYSTEM

```
clc
n = input('\n enter the number of buses')
f=60;
load line_data.txt
load transformer_data.txt
load slackbus_data.txt
load PVbus_data.txt
load PQbus_data.txt
load machine_data.txt
[m,o]=size(line_data);
for i =1:1:m
        k = line_data(i,1);
        l = line_data(i,2);
        r = line_data(i,3);
        z(k,l)= complex(r,line_data(i,4));
        z(l,k)= z(k,l);
        s(k,l) = (complex(0,line_data(i,5)))/2;
end
[g,h] = size(transformer_data);
for i =1:1:g
     k = transformer_data(i,1);
        l = transformer_data(i,2);
        rt =transformer_data(i,3);
        z(l,k)= z(k,l);
end
y = [zeros(n,n)];
for i =1:1:n
     for j =1:1:n
         if z(i,j)==0
                 c(i,j)=0;
         else
             c(i,j)= (1/z(i,j));
          end
     end
 end

 for i=1:1:n
    for j=1:1:n

        y(i,i)= y(i,i)+c(i,j)+ s(i,j);
    end
 end
Ybus = y;
[mp,op]=size(PVbus_data);
[mq,oq] = size(PQbus_data);
```

```
mv = [slackbus_data(1,2);PVbus_data(1:mp,2);ones(1,mq)'];
th = [slackbus_data(1,3); zeros(1,mp+mq)'];
pg = [ 0;PVbus_data(1:mp,3);PQbus_data(1:mq,4)];
qg = [zeros(1,mp+1)';PQbus_data(1:mq,5)];
pd = [slackbus_data(1,4);PVbus_data(1:mp,4);PQbus_data(1:mq,2)];
qd = [slackbus_data(1,5);PVbus_data(1:mp,5);PQbus_data(1:mq,3)];
npv = mp+1;
P = n-1;
Q = n-npv;
r = P+Q;
itr=0;
acc=1;
realy = real(y);
imagy = imag (y);
[theta , yy] = cart2pol(realy,imagy);

while acc > 1e-12
  [pcal,qcal]=powers(mv,yy,theta,th,n,n);
  accp = p'-pcal;
  accq = q'-qcal;
  accpq =[ accp(2:n)  accq(npv+1:n)];
 for i=2:1:n
     for k = 2:1:n
        if i==k
           H(i,i)= ((-1)*qcal(i)- (mv(i)*mv(i)*imagy(i,i)));
        else
           H(i,k)=(mv(i)*mv(k)*yy(i,k)*sin(th(i)-theta(i,k)-th(k)));
        end
      end
 end

 for i=npv+1:1:n
   for k = 2:1:n
     if i==k
       M(i,i)= (pcal(i)- (mv(i)*mv(i)* realy(i,i)));
     else
       M(i,k) = ((-1)* (mv(i)*mv(k)*yy(i,k)*cos(th(k)+theta(i,k)-
th(i))));
     end
    end
 end

 for i = 2:1:n
     for k = npv+1:1:n
        if i==k
         N(i,i)= (pcal(i)+ (mv(i)*mv(i)*realy(i,i)));
        else
         N(i,k) = (mv(i)*mv(k)*yy(i,k)*cos(theta(i,k)+th(k)-th(i)));
         end
      end
 end
```

```
for i=npv+1:1:n
   for k = npv+1:1:n
     if i==k
       qcal(i);
       L(i,i)= (qcal(i)-(imagy(i,i)*mv(i)*mv(i)));
     else
       L(i,k)= (mv(i)*mv(k)*yy(i,k)*sin(th(i)-th(k)-theta(i,k)));
     end
   end
end
jac =[ H(2:n,2:n) N(2:n,npv+1:n; M(npv+1:n,2:n)  L(npv+1:n,npv+1:n)];
        pq = ((jac)\(accpq'));
        thnew = th + [ 0; pq(1:n-1)];
        mvnew = mv + mv.*[zeros(1,npv)';pq(n:r)];
        mv = mvnew;
        th = thnew;
        itr = itr+1;
end
[pcal,qcal]=powers(mv,yy,theta,th,npv,n);
pgen(1) = pcal(1)+ slackbus_data(1,4);
qgen(1) = qcal(1)+ slackbus_data(1,5);
for i = 2:1:npv
    k = i-1;
   qgen(i) = qcal(i) + PVbus_data(k,5);
end
th = (th*180)/pi;
Pg = [ pgen(1); pg(2:n)];
Qg = [ qgen(1:npv)';qg(npv+1:n)];
pd;
qd ;
fprintf ('\n  \tvoltage \tangle \t\tPg \t\tQg \t\t\tPd \t\tQd')
answer = [zeros(n,6)];
answer = [ mv th Pg  Qg pd qd]
the = (th*pi)/180;
%---------------------------------------------------------
%solving for initial values of state variables
[Vcart,Thcart] = pol2cart(the,mv);
Igo= zeros (npv,1);
Eqo= zeros (npv,1);
for i =1:1:npv
    Igo(i)= complex(Pg(i),-Qg(i))/ (complex(Vcart(i),-Thcart(i)));
    Eqo(i)= complex(Vcart(i),Thcart(i))
        +((complex(machine_data(i,2),machine_data(i,5)))*Igo(i));
    Ido(i)= (-abs(Igo(i))* sin(angle(Eqo(i))-angle(Igo(i))));
    Iqo(i)= abs(Igo(i))*cos(angle(Eqo(i))-angle(Igo(i)));
    Vdo(i)= (-mv(i)*sin(angle(Eqo(i))-the(i)));
    Vqo(i)= mv(i)*cos(angle(Eqo(i))-the(i));
    Efdo(i)= abs(Eqo(i))- ((machine_data(i,3)- machine_data(i,5))*
         Ido(i));
    Etqo(i)= Efdo(i)+((machine_data(i,3)- machine_data(i,4))*
         Ido(i));
    Etdo(i)= (-1)*((machine_data(i,5)- machine_data(i,6))* Iqo(i));
    Peo(i)= Etqo(i)*Iqo(i) + Etdo(i)*Ido(i)+((machine_data(i,4)-
         machine_data(i,6))*Ido(i)*Iqo(i));
    Pm(i) = Peo(i);
    Vref(i)= (Efdo(i)/(machine_data(i,12)))+ mv(i);
end

fb = input ('enter the index of faulted bus')
fprintf(' the line connected to %4i that is tripped',fb)
fbl = input('\n enter its index')
Id = Ido;
Iq = Iqo;
Etd = Etdo;
Etq = Etqo;
Eq = Eqo;
mv;
Ra = [machine_data(1:npv,2)];
Xtq = [machine_data(1:npv,6)];
Xtd = [machine_data(1:npv,4)];
delta = angle(Eq);
itr = 0;

G11=[zeros(npv,npv)];
G12=[zeros(npv,npv)];
G21=[zeros(npv,npv)];
G22=[zeros(npv,npv)];
G13=[zeros(npv,n)];
G14=[zeros(npv,n)];
G23=[zeros(npv,n)];
G24=[zeros(npv,n)];
G31=[zeros(n,npv)];
G32=[zeros(n,npv)];
G41=[zeros(n,npv)];
G42=[zeros(n,npv)];
G33=[zeros(n,n)];
G34=[zeros(n,n)];
G43=[zeros(n,n)];
G44=[zeros(n,n)];
maxerror = 1;
while(maxerror > 1e-6)
    [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
  for i = 1:1:n
       for j = 1:1:n
        if (i<=npv) && (j<=npv)
         if i==j
        G11(i,i)= (-Xtq(i));
        G12(i,i)= (-Ra(i));
        G21(i,i)= (-Ra(i));
        G22(i,i)= (Xtd(i));
```

```
         else
        G11(i,j)=0;
        G12(i,j)=0;
        G21(i,j)=0;
        G22(i,j)=0;
         end
        end
        if (i<=npv) && (j<=n)
            if (i==j)
        G13(i,i)= (- mv(i)*cos(the(i)-angle(Eq(i))));
        G14(i,i)= (-sin(the(i)-angle(Eq(i))));
        G23(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
        G24(i,i)= (-cos(the(i)-angle(Eq(i))));
            elseif (i~=j)
        G13(i,j)=0;
        G14(i,j)=0;
        G23(i,j)=0;
        G24(i,j)=0;
            end
        end
          if(i<=n) && (j<=npv)
            if(i==j)
        G31(i,i)= (mv(i)*cos(the(i)-angle(Eq(i))));
        G32(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
        G41(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
        G42(i,j)=(-mv(i)*cos(the(i)-angle(Eq(i))));
            else
        G31(i,j)=0;
        G32(i,j)=0;
        G41(i,j)=0;
        G42(i,j)=0;
            end
            end
        if (i<=n)&&(j<=n)
        if(i==j)
             if(i<=npv)
G33(i,i)= (-mv(i)*Iq(i)*sin(the(i)- angle(Eq(i))))+
    (mv(i)*Id(i)*cos(the(i)- angle(Eq(i))))+ qcal(i)+
    (mv(i)*mv(i)*imag(y(i,i)));
G34(i,i)= (Iq(i)*cos(the(i)-angle(Eq(i))))+ (Id(i)*sin(the(i)-
     angle(Eq(i))))- 2*a(i)*mv(i)- mv(i)* real(y(i,i))-int(i);
G43(i,i)= (mv(i)*Iq(i)*cos(the(i)- angle(Eq(i))))+
     (mv(i)*Id(i)*sin(the(i)- angle(Eq(i))))-pcal(i)+
     (mv(i)*mv(i)*real(y(i,i)));
G44(i,i)= (Iq(i)*sin(the(i)- angle(Eq(i))))- (Id(i)*cos(the(i)-
    angle(Eq(i))))- 2*b(i)*mv(i)+(mv(i)*imag(y(i,i))+intf(i));
            else
G33(i,i)= (-qcal(i)-(mv(i)*mv(i)*imag(y(i,i))));
G34(i,i)= (2*a(i)*mv(i))+(mv(i)*real(y(i,i)))+int(i);
G43(i,i)= pcal(i)-(mv(i)*mv(i)*real(y(i,i)));
G44(i,i)= (2*b(i)*mv(i))-(mv(i)*imag(y(i,i)))-intf(i);
```

```
             end
        else
            if(i<=npv)
G33(i,j)= (mv(i)*mv(j)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
G34(i,j)= (-mv(i)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G43(i,j)=(mv(i)*mv(j)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G44(i,j)=(mv(i)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
            elseif(i>npv)
G33(i,j)= (-mv(i)*mv(j)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
G34(i,j)=  (mv(i)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G43(i,j)=(-mv(i)*mv(j)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G44(i,j)=(-mv(i)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
                  End
          end
        end
    end
end
       jac1 = [G11 G12 G13 G14;
               G21 G22 G23 G24;
               G31 G32 G33 G34;
               G41 G42 G43 G44];
jacob = [jac1(1:2*npv+fb-1,1:2*npv+fb-1) jac1(1:2*npv+fb-
             1,2*npv+fb+1:2*npv+n+fb-1) jac1(1:2*npv+fb-
             1,2*npv+n+fb+1:2*npv+2*n);
         jac1(2*npv+fb+1:2*npv+n+fb-1,1:2*npv+fb-1)
         jac1(2*npv+fb+1:2*npv+n+fb-1,2*npv+fb+1:2*npv+n+fb-1)
         jac1(2*npv+fb+1:2*npv+n+fb-1,2*npv+n+fb+1:2*npv+2*n);
    jac1(2*npv+n+fb+1:2*npv+2*n,1:2*npv+fb-1)
jac1(2*npv+n+fb+1:2*npv+2*n,2*npv+fb+1:2*npv+n+fb-1)
jac1(2*npv+n+fb+1:2*npv+2*n,2*npv+n+fb+1:2*npv+2*n)];

[equation]=equcal(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qc
al,n,npv,fb);
    Iq = Iq + error(1:npv);
    Id = Id + error(npv+1:2*npv);
    the(1:fb-1) = the(1:fb-1) + error(2*npv+1:2*npv+fb-1);
    the(fb+1:n) = the(fb+1:n) + error(2*npv+fb:2*npv+n-1);
    mv(1:fb-1)=mv(1:fb-1)+ error(2*npv+n:2*npv+n+fb-2);
    mv(fb+1:n)=mv(fb+1:n)+ error(2*npv+n+fb-1:2*npv+2*n-2);
    the(fb)=0;
    mv(fb)=0;
    itr = itr+1;

[equation]=equcal(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qc
al,n,npv,fb);
maxerror = (max(abs(equation)));
end
mv
the
Iq
Id
```

```
w = zeros(npv,1);
tc = input('enter the clearing time in seconds')
st=input('enter the value of simulation time')
wb = 2*pi*f;
hh = input('enter the value of step of integration')
ntc = tc/hh;
nst = (st-tc)/hh;
Delta = zeros(npv,T+1);
THETA = zeros(npv,T+1);
gamma = zeros(npv,T+1);
differen = zeros(npv,T+1);
THER = zeros(1,T+1);
WO = zeros(1,T+1);
EFD = zeros(npv,T+1);
ETQ = zeros(npv,T+1);
ETD  = zeros(npv,T+1);
W = zeros(npv,T+1);
ID  = zeros(npv,T+1);
IQ = zeros(npv,T+1);
MV = zeros(n,T+1);
ANGLE = zeros(n,T+1);
PE = zeros(npv,T+1);
H =  [machine_data(1:npv,7)];
Td =  [machine_data(1:npv,8)];
Tq =  [machine_data(1:npv,9)];
Ta =  [machine_data(1:npv,13)];
Ka =  [machine_data(1:npv,12)];
Xq =  [machine_data(1:npv,5)];
Xd =  [machine_data(1:npv,3)];
Efd = Efdo;
Efx = Efdo;
itr=0;
nt = 1;
Delta(1:npv,1)= delta(1:npv);
W(1:npv,1)= w(1:npv);
EFD(1:npv,1)= Efdo(1:npv);
ETQ(1:npv,1)= Etqo(1:npv);
ETD(1:npv,1)= Etdo(1:npv);
IQ(1:npv,1)= Iq(1:npv);
ID(1:npv,1)= Id(1:npv);
MV(1:n,1)= mv;
ANGLE(1:n,1)= the;
PE(1:npv,1)= Peo;
%-------------------------------------------------------
%solution after integrating the system fom 0 to tc
while nt<=ntc
    itrr = 0;
    deltaa = delta;
    ww = w;
    Etqq = Etq;
    Etdd = Etd;
    Efdd = Efd;
    Efxx = Efx;
    Idd= Id;
    Iqq = Iq;
    mvv = mv;
    accc=1;
while accc>1e-6
  [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
  [jac2]=jacobbian(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,q
  cal,int,intf,n,npv,hh,Xd,Xq,wb,Td,Tq,Ta,Ka,H,y,yy,theta);
  jacobb =[ jac2(1:7*npv+fb-1,1:7*npv+fb-1)     jac2(1:7*npv+fb-1,
7*npv+fb+1:7*npv+n+fb-1)  jac2(1:7*npv+fb-1,
7*npv+n+fb+1:7*npv+2*n);
            jac2(7*npv+fb+1:7*npv+n+fb-1,1:7*npv+fb-1)
jac2(7*npv+fb+1:7*npv+n+fb-1, 7*npv+fb+1:7*npv+n+fb-1)
jac2(7*npv+fb+1:7*npv+n+fb-1, 7*npv+n+fb+1:7*npv+2*n);
            jac2( 7*npv+n+fb+1:7*npv+2*n,1:7*npv+fb-1)  jac2(
7*npv+n+fb+1:7*npv+2*n,7*npv+fb+1:7*npv+n+fb-1) jac2(
7*npv+n+fb+1:7*npv+2*n, 7*npv+n+fb+1:7*npv+2*n)];
[eeq11,eeq22,eeq33,eeq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
equationcal
(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,delta
a,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,Ta,Ka,
Vref,Pm,H,mvv);

equationn = [eeq11(1:npv) eeq22(1:npv) eeq33(1:npv) eeq44(1:npv)
eeq55(1:npv) eeq66(1:npv) eeq77(1:npv) eeq88(1:fb-1) eeq88(fb+1:n)
eeq99(1:fb-1) eeq99(fb+1:n)]' ;
 errorr = (-jacobb\equationn);
 Etq = Etq + (errorr(2*npv+1: 3*npv))';
 Etd = Etd + (errorr(3*npv+1:4*npv))';
 Efx = Efx + (errorr(4*npv+1:5*npv))';
 Iq = Iq + (errorr(5*npv+1:6*npv));
 Id = Id + (errorr(6*npv+1:7*npv));
 the(1:fb-1) = the(1:fb-1) + errorr(7*npv+1:7*npv+fb-1);
 the(fb+1:n) = the(fb+1:n) + errorr(7*npv+fb:7*npv+n-1);
 mv(1:fb-1)=mv(1:fb-1)+ errorr(7*npv+n:7*npv+n+fb-2);
 mv(fb+1:n)=mv(fb+1:n)+ errorr(7*npv+n+fb-1:7*npv+2*n-2);
     for i = 1:1:npv
       if (Efx(i)>= machine_data(i,15))&& (Efx(i)<=
          machine_data(i,14))
           Efd(i)= Efx(i);
       elseif Efx(i)< machine_data(i,15)
           Efd(i)= machine_data(i,15);
       elseif Efx(i)> machine_data(i,14)
          Efd(i)= machine_data(i,14);
     end
     end
[pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
```

```
[eeq11,eeq22,eeq33,eq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
equationcal
(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,delta
a,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,Ta,Ka,
Vref,Pm,H,mvv);
equationn = [eeq11(1:npv) eeq22(1:npv) eeq33(1:npv) eeq44(1:npv)
eeq55(1:npv) eeq66(1:npv) eeq77(1:npv) eeq88(1:fb-1) eeq88(fb+1:n)
eeq99(1:fb-1) eeq99(fb+1:n)]' ;
accc= max(abs(equationn));
itrr = itrr+1 ;
end

[Pe] =  elepow(Etq,Etd,Id,Iq,npv,machine_data);
iteration(nt) = itrr;
nt = nt+1;
Delta(1:npv,nt)= delta(1:npv);
W(1:npv,nt)= w(1:npv);
EFD(1:npv,nt)= Efd(1:npv);
ETQ(1:npv,nt)= Etq(1:npv);
ETD(1:npv,nt)= Etd(1:npv);
IQ(1:npv,nt)= Iq(1:npv);
ID(1:npv,nt)= Id(1:npv);
MV(1:n,nt)= mv(1:n);
ANGLE(1:n,nt)= the(1:n);
PE(1:npv,nt) = Pe(1:npv);
end
%----------------------------------------------------------
%new y bus after removing fault and hence solving algebraic
equations
y(fb,fb)= y(fb,fb)+y(fb,fbl);
y(fbl,fbl)= y(fbl,fbl)+y(fb,fbl);
y;
[theta , yy] = cart2pol(real(y),imag(y));
M2 = [eye(n,npv) zeros(n,npv);
      zeros(n,npv) eye(n,npv)];
EE = [Etq';Etd'];
for i =1:1:n
     for j =1:1:n
         G(i,j)= real(y(i,j));
         B(i,j)= imag(y(i,j));
         if i<=npv && j<=npv
         if i==j
         C(i,i)= cos(delta(i));
         D(i,i)= sin(delta(i));
         z1(i,i)= (Ra(i)/(Ra(i)*Ra(i)+ Xtd(i)*Xtq(i)));
         z2(i,i)= (Xtd(i)/(Ra(i)*Ra(i)+ Xtd(i)*Xtq(i)));
         z3(i,i)= (-Xtq(i)/(Ra(i)*Ra(i)+ Xtd(i)*Xtq(i)));
         elseif i~=j
         C(i,j)=0;
         D(i,j)=0;
         z1(i,j)=0;
         z2(i,j)=0;
         z3(i,j)=0;
         end
         end

     end
 end

 Z = [ z1 z2;
        z3 z1];
 D1 = [ C  -D;
        D   C];

 D2 =  [ C  D;
        -D   C];
 LL = [ G+diag(a) -B+diag(b);
           B-diag(b)   G+diag(a)];
 ANI = (eye(2*npv)+D1*Z*D2*M2'*(LL\M2))\D1*Z*EE;
 ANIQD = D2*ANI;
 ANIQ = ANI(1:npv);
 ANID = ANI(npv+1:2*npv);
 ANVQ = ANV(1:n)
 ANVD = ANV(n+1:2*n);
 Iq=ANIQ;
 Id=ANID;
 [the,mv] = cart2pol(ANVQ,ANVD);
%---------------------------------------------------------
%solution after integrating the system fom tc to t
nt1 = 1;
while nt1<=nst
itrr = 0;
    deltaa = delta;
    ww = w;
    Etqq = Etq;
    Etdd = Etd;
    Efdd = Efd;
    Efxx = Efx;
    Idd= Id;
    Iqq = Iq;
    mvv = mv;
accc=1;
while accc>1e-6
 [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
 [jac2]=jacobbian(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qc
   al,int,intf,n,npv,hh,Xd,Xq,wb,Td,Tq,Ta,Ka,H,y,yy,theta);
 [eeq11,eeq22,eeq33,eeq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
  equationcal
  (Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,del
  taa,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,Ta
  ,Ka,Vref,Pm,H,mvv);
equat = [eeq11 eeq22 eeq33 eeq44 eeq55 eeq66 eeq77 eeq88 eeq99]' ;
```

```
 err = (-jac2\equat);
 delta = delta + (err(1:npv));
 w = w + (err(npv+1:2*npv));
 Etq = Etq + (err(2*npv+1: 3*npv))';
 Etd = Etd + (err(3*npv+1:4*npv))';
 Efx = Efx + (err(4*npv+1:5*npv))';
 Iq = Iq + (err(5*npv+1:6*npv));
 Id = Id + (err(6*npv+1:7*npv));
 the = the + err(7*npv+1:7*npv+n);
 mv=mv+ err(7*npv+n+1:7*npv+2*n);

 for i = 1:1:npv
  if (Efx(i)>= machine_data(i,15))&& (Efx(i)<= machine_data(i,14))
          Efd(i)= Efx(i);
  elseif Efx(i)< machine_data(i,15)
          Efd(i)= machine_data(i,15);
  elseif Efx(i)> machine_data(i,14)
           Efd(i)= machine_data(i,14);
  End
 end
 [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
 [eeq11,eeq22,eeq33,eeq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
   equationcal
   (Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,del
   taa,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,Ta
   ,Ka,Vref,Pm,H,mvv);
 equat = [eeq11 eeq22 eeq33 eeq44 eeq55 eeq66 eeq77 eeq88 eeq99]' ;
 accc= max(abs(equat));
 itrr = itrr+1 ;
end
 [Pe] =  elepow(Etq,Etd,Id,Iq,npv,machine_data);
 iteration1(nt1)=itrr;
 Delta(1:npv,nt+nt1)=delta(1:npv);
 W(1:npv,nt+nt1)= w(1:npv);
 EFD(1:npv,nt+nt1)= Efd(1:npv);
 ETQ(1:npv,nt+nt1)= Etq(1:npv);
 ETD(1:npv,nt+nt1)= Etd(1:npv);
 IQ(1:npv,nt+nt1)= Iq(1:npv);
 ID(1:npv,nt+nt1)= Id(1:npv);
 MV(:,nt+nt1)= mv(1:n);
 ANGLE(:,nt+nt1)= the(1:n);
 PE(1:npv,nt+nt1)=Pe(1:npv);
 nt1 = nt1+1;
 end

 Ht=0;
 Pcoi = 0;
 Ht=sum(H);
 dif = zeros(npv,T+1);
 for i =1:1:T+1
     summ(i)=0;
```

```
    summ1(i)=0;
for j=1:1:npv
    summ(i)=summ(i)+H(j)*Delta(j,i);
    summ1(i)=summ1(i)+H(j)*W(j,i);
end
  THER(i)=summ(i)/Ht;
  WO(i) = summ1(i)/Ht;
end
for i=1:1:npv
 THETA(i,:)=Delta(i,:)-THER(1,:);
 gamma(i,:)=W(i,:)-WO(1,:);
 dif(i,:)=Pm(1,i)- PE(i,:);
end
Pcoi = sum(dif);
t=[0:hh:st];
THETA = THETA*180/pi;
figure(1);
plot(t,THETA)
xlabel('time in seconds')
ylabel('generator angles in degree')
figure (2);
plot (t,W)
xlabel('time in seconds')
ylabel('speed in p.u')
figure(3);
plot(t,ETQ)
xlabel('time in seconds')
ylabel('Etq in p.u')
figure(4);
plot(t,ETD)
xlabel('time in seconds')
ylabel('Etd in p.u')
figure(5);
plot(t,EFD)
xlabel('time in seconds')
ylabel('Efd in p.u')
figure(6);
plot(t,IQ)
xlabel('time in seconds')
ylabel('Iq in p.u')
figure(7);
plot(t,ID)
xlabel('time in seconds')
ylabel('Id in p.u')
figure(8);
plot(t,MV)
xlabel('time in seconds')
ylabel('magnitude of voltage in p.u')
figure(9);
plot(t,ANGLE)
xlabel('time in seconds')
```

```
ylabel('theta in p.u')
```

E.2 MATLAB PROGRAM FOR TRANSIENT STABILITY ANALYSIS OF 39 BUS 10 GENERATOR SYSTEM USING TEF METHOD

```
clc
n = input('\n enter the number of buses')
f=60;
load line_data.txt
load transformer_data.txt
load slackbus_data.txt
load PVbus_data.txt
load PQbus_data.txt
load machine_data.txt
[m,o]=size(line_data);
for i =1:1:m
        k = line_data(i,1);
        l = line_data(i,2);
        r = line_data(i,3);
        z(k,l)= complex(r,line_data(i,4));
        z(l,k)= z(k,l);
        s(k,l) = (complex(0,line_data(i,5)))/2;

end
[g,h] = size(transformer_data);
for i =1:1:g
     k = transformer_data(i,1);
        l = transformer_data(i,2);
       rt =transformer_data(i,3);
                z(l,k)= z(k,l);
end
y = [zeros(n,n)];
for i =1:1:n
     for j =1:1:n
         if z(i,j)==0
                 c(i,j)=0;
         else
             c(i,j)= (1/z(i,j));

         end
     end
end
for i=1:1:n
    for j=1:1:n

        y(i,i)= y(i,i)+c(i,j)+ s(i,j);
    end

 end
 Ybus = y;
 [mp,op]=size(PVbus_data);
 [mq,oq] = size(PQbus_data);
 mv = [slackbus_data(1,2);PVbus_data(1:mp,2);ones(1,mq)'];
 th = [slackbus_data(1,3); zeros(1,mp+mq)'];
 pg = [ 0;PVbus_data(1:mp,3);PQbus_data(1:mq,4)];
 qg = [zeros(1,mp+1)';PQbus_data(1:mq,5)];
 pd = [slackbus_data(1,4);PVbus_data(1:mp,4);PQbus_data(1:mq,2)];
 qd = [slackbus_data(1,5);PVbus_data(1:mp,5);PQbus_data(1:mq,3)];
 p = [0;(pg(2:n)-pd(2:n))];
 q = [0;(qg(2:n)-qd(2:n))];
 npv = mp+1;
 P = n-1;
 Q = n-npv;
 r = P+Q;
 itr=0;
 acc=1;
[theta , yy] = cart2pol(realy,imagy);
 while acc > 1e-12
   [pcal,qcal]=powers(mv,yy,theta,th,n,n);
   accpq =[ accp(2:n)  accq(npv+1:n)];
 for i=2:1:n
   for k = 2:1:n
     if i==k
       H(i,i)= ((-1)*qcal(i) - (mv(i)*mv(i)*imagy(i,i)));
     else
       H(i,k)= (mv(i)*mv(k)*yy(i,k)*sin(th(i)-theta(i,k)-th(k)));
     end
    end
 end
 for i=npv+1:1:n
    for k = 2:1:n
        if i==k
           M(i,i)= (pcal(i)- (mv(i)*mv(i)* realy(i,i)));
        else
           M(i,k) =((-1)* (mv(i)*mv(k)*yy(i,k)*cos(th(k)+theta(i,k)-
 th(i))));
         end
     end
 end
 for i = 2:1:n
   for k = npv+1:1:n
       if i==k
         N(i,i)= (pcal(i)+ (mv(i)*mv(i)*realy(i,i)));
       else
         N(i,k) = (mv(i)*mv(k)*yy(i,k)*cos(theta(i,k)+th(k)-th(i)));
       end
     end
 end
 for i=npv+1:1:n
    for k = npv+1:1:n
```

```
      if i==k
        qcal(i);
        L(i,i)= (qcal(i)-(imagy(i,i)*mv(i)*mv(i)));
       else
        L(i,k)= (mv(i)*mv(k)*yy(i,k)*sin(th(i)-th(k)-theta(i,k)));
       end
    end
end
jac = [ H(2:n,2:n) N(2:n,npv+1:n) ; M(npv+1:n,2:n)
     L(npv+1:n,npv+1:n)];
                acc = max(abs(pq));
        thnew = th + [ 0; pq(1:n-1)];
        mvnew = mv + mv.*[zeros(1,npv)';pq(n:r)];
        mv = mvnew;
        th = thnew;
        itr = itr+1;
end
[pcal,qcal]=powers(mv,yy,theta,th,npv,n);
pgen(1) = pcal(1)+ slackbus_data(1,4);
qgen(1) = qcal(1)+ slackbus_data(1,5);
for i = 2:1:npv
    k = i-1;
    qgen(i) = qcal(i) + PVbus_data(k,5);
end
th = (th*180)/pi;
Pg = [ pgen(1); pg(2:n)];
Qg = [ qgen(1:npv)';qg(npv+1:n)];
pd;
qd ;
fprintf ('\n  \tvoltage \tangle \t\tPg \t\tQg \t\t\tPd \t\tQd')
answer = [zeros(n,6)];
answer = [ mv th Pg  Qg pd qd]
the = (th*pi)/180
%----------------------------------------------------------
%solving for initial values of state variables
[Vcart,Thcart] = pol2cart(the,mv);
Igo= zeros (npv,1);
Eqo= zeros (npv,1);
Ido = zeros (npv,1);
Iqo = zeros (npv,1);
for i =1:1:npv
  Igo(i)= complex(Pg(i),-Qg(i))/ (complex(Vcart(i),-Thcart(i)));
  Eqo(i)= complex(Vcart(i),Thcart(i))
     +((complex(machine_data(i,2),machine_data(i,5)))*Igo(i));
  Ido(i)= (-abs(Igo(i))* sin(angle(Eqo(i))-angle(Igo(i))));
  Iqo(i)= abs(Igo(i))*cos(angle(Eqo(i))-angle(Igo(i)));
  Vdo(i)= (-mv(i)*sin(angle(Eqo(i))-the(i)));
  Vqo(i)= mv(i)*cos(angle(Eqo(i))-the(i));
  Efdo(i)= abs(Eqo(i))- ((machine_data(i,3)- machine_data(i,5))*
     Ido(i));
  Etqo(i)= Efdo(i)+((machine_data(i,3)- machine_data(i,4))*
     Ido(i));
  Etdo(i)= (-1)*((machine_data(i,5)- machine_data(i,6))* Iqo(i));
  Peo(i)= Etqo(i)*Iqo(i) + Etdo(i)*Ido(i)+((machine_data(i,4)-
     machine_data(i,6))*Ido(i)*Iqo(i));
  Pm(i) = Peo(i);
  Vref(i)= (Efdo(i)/(machine_data(i,12)))+ mv(i);
end
fb = input ('enter the index of faulted bus')
fprintf(' the line connected to %4i that is tripped',fb)
fbl = input('\n enter its index')
Id = Ido;
Iq = Iqo;
Etd = Etdo;
Etq = Etqo;
Eq = Eqo;
mv;
Ra = [machine_data(1:npv,2)];
Xtq = [machine_data(1:npv,6)];
Xtd = [machine_data(1:npv,4)];
delta = angle(Eq);
itr = 0;

G11=[zeros(npv,npv)];
G12=[zeros(npv,npv)];
G21=[zeros(npv,npv)];
G22=[zeros(npv,npv)];
G13=[zeros(npv,n)];
G14=[zeros(npv,n)];
G23=[zeros(npv,n)];
G24=[zeros(npv,n)];
G31=[zeros(n,npv)];
G32=[zeros(n,npv)];
G41=[zeros(n,npv)];
G42=[zeros(n,npv)];
G33=[zeros(n,n)];
G34=[zeros(n,n)];
G43=[zeros(n,n)];
G44=[zeros(n,n)];
maxerror = 1;
while(maxerror > 1e-6)
    [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
  for i = 1:1:n
       for j = 1:1:n
        if (i<=npv) && (j<=npv)
         if i==j
        G11(i,i)= (-Xtq(i));
        G12(i,i)= (-Ra(i));
        G21(i,i)= (-Ra(i));
        G22(i,i)= (Xtd(i));
         else
         G11(i,j)=0;
         G12(i,j)=0;
         G21(i,j)=0;
         G22(i,j)=0;
          end
         end
         if (i<=npv) && (j<=n)
             if (i==j)
         G13(i,i)= (- mv(i)*cos(the(i)-angle(Eq(i))));
         G14(i,i)= (-sin(the(i)-angle(Eq(i))));
         G23(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
         G24(i,i)= (-cos(the(i)-angle(Eq(i))));
             elseif (i~=j)
         G13(i,j)=0;
         G14(i,j)=0;
         G23(i,j)=0;
         G24(i,j)=0;
             end
         end

             if(i<=n) && (j<=npv)
             if(i==j)
         G31(i,i)= (mv(i)*cos(the(i)-angle(Eq(i))));
         G32(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
         G41(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
         G42(i,j)=(-mv(i)*cos(the(i)-angle(Eq(i))));
             else
         G31(i,j)=0;
         G32(i,j)=0;
         G41(i,j)=0;
         G42(i,j)=0;
             end
             end
         if (i<=n)&&(j<=n)
         if(i==j)
              if(i<=npv)
G33(i,i)= (-mv(i)*Iq(i)*sin(the(i)- angle(Eq(i))))+
     (mv(i)*Id(i)*cos(the(i)- angle(Eq(i))))+ qcal(i)+
     (mv(i)*mv(i)*imag(y(i,i)));
G34(i,i)= (Iq(i)*cos(the(i)-angle(Eq(i))))+ (Id(i)*sin(the(i)-
     angle(Eq(i))))- 2*a(i)*mv(i)- mv(i)* real(y(i,i))-int(i);
G43(i,i)= (mv(i)*Iq(i)*cos(the(i)- angle(Eq(i))))+
     (mv(i)*Id(i)*sin(the(i)- angle(Eq(i))))-pcal(i)+
     (mv(i)*mv(i)*real(y(i,i)));
G44(i,i)= (Iq(i)*sin(the(i)- angle(Eq(i))))- (Id(i)*cos(the(i)-
     angle(Eq(i))))- 2*b(i)*mv(i)+(mv(i)*imag(y(i,i))+intf(i));
                 Else
G33(i,i)= (-qcal(i)-(mv(i)*mv(i)*imag(y(i,i))));
G34(i,i)= (2*a(i)*mv(i))+(mv(i)*real(y(i,i)))+int(i);
G43(i,i)= pcal(i)-(mv(i)*mv(i)*real(y(i,i)));
G44(i,i)= (2*b(i)*mv(i))-(mv(i)*imag(y(i,i)))-intf(i);
```

```
             end
         else
                   if(i<=npv)
G33(i,j)= (mv(i)*mv(j)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
G34(i,j)= (-mv(i)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G43(i,j)=(mv(i)*mv(j)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G44(i,j)=(mv(i)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
                  elseif(i>npv)
G33(i,j)= (-mv(i)*mv(j)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
G34(i,j)=  (mv(i)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G43(i,j)=(-mv(i)*mv(j)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G44(i,j)=(-mv(i)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
                   End
           end
         end
     end
  end
        jac1 = [G11 G12 G13 G14;
                G21 G22 G23 G24;
                G31 G32 G33 G34;
                G41 G42 G43 G44];
  jacob = [jac1(1:2*npv+fb-1,1:2*npv+fb-1) jac1(1:2*npv+fb-1,2*npv+fb+1:2*npv+n+fb-1) jac1(1:2*npv+fb-1,2*npv+n+fb+1:2*npv+2*n);
           jac1(2*npv+fb+1:2*npv+n+fb-1,1:2*npv+fb-1)
  jac1(2*npv+fb+1:2*npv+n+fb-1,2*npv+fb+1:2*npv+n+fb-1)
  jac1(2*npv+fb+1:2*npv+n+fb-1,2*npv+n+fb+1:2*npv+2*n);
           jac1(2*npv+n+fb+1:2*npv+2*n,1:2*npv+fb-1)
  jac1(2*npv+n+fb+1:2*npv+2*n,2*npv+fb+1:2*npv+n+fb-1)
  jac1(2*npv+n+fb+1:2*npv+2*n,2*npv+n+fb+1:2*npv+2*n)];

  [equation]=equcal(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb);
      Iq = Iq + error(1:npv);
      Id = Id + error(npv+1:2*npv);
      the(1:fb-1) = the(1:fb-1) + error(2*npv+1:2*npv+fb-1);
      the(fb+1:n) = the(fb+1:n) + error(2*npv+fb:2*npv+n-1);
      mv(1:fb-1)=mv(1:fb-1)+ error(2*npv+n:2*npv+n+fb-2);
      mv(fb+1:n)=mv(fb+1:n)+ error(2*npv+n+fb-1:2*npv+2*n-2);
      the(fb)=0;
      mv(fb)=0;
      itr = itr+1;
  [equation]=equcal(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb);
      maxerror = (max(abs(equation)));
  end
  mv
  the
  Iq
  Id
```

```
w = zeros(npv,1);
st=input('enter the value of simulation time')
wb = 2*pi*f;
hh = input('enter the value of step of integration')
SDelta = zeros(npv,T+1);
WO = zeros(1,T+1);
SEFD = zeros(npv,T+1);
SEQT = zeros(npv,T+1);
SEDT  = zeros(npv,T+1);
SW = zeros(npv,T+1);
SID  = zeros(npv,T+1);
SIQ = zeros(npv,T+1);
SMV = zeros(n,T+1);
SANGLE = zeros(n,T+1);
SPE = zeros(npv,T+1);
gamma = zeros(npv,T+1);
dif = zeros(npv,T+1);

nt = 1;
itr=0;
iteration = zeros(T);
H =  [machine_data(1:npv,7)];
Td =  [machine_data(1:npv,8)];
Tq =  [machine_data(1:npv,9)];
Ta =  [machine_data(1:npv,13)];
Ka =  [machine_data(1:npv,12)];
Xq =  [machine_data(1:npv,5)];
Xd =  [machine_data(1:npv,3)];
SDelta(1:npv,1)= delta(1:npv);
SW(1:npv,1)= w(1:npv);
SEFD(1:npv,1)= Efdo(1:npv);
SEQT(1:npv,1)= Etqo(1:npv);
SEDT(1:npv,1)= Etdo(1:npv);
SIQ(1:npv,1)= Iq(1:npv);
SID(1:npv,1)= Id(1:npv);
SMV(1:n,1)= mv;
SANGLE(1:n,1)= the;
SPE(1:npv,1)= Peo;
%-------------------------------------------------------------
while nt <= T
    itrr = 0;
    deltaa = delta;
    ww = w;
    Etqq = Etq;
    Etdd = Etd;
    Efdd = Efd;
    Efxx = Efx;
    Idd= Id;
    Iqq = Iq;
    mvv = mv;
accc=1;
```

```
while accc>1e-12
  [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
  [jac2]=jacobbian(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,q
    cal,int,intf,n,npv,hh,Xd,Xq,wb,Td,Tq,Ta,Ka,H,Ybus,yy,theta);

jacobb =[ jac2(1:7*npv+fb-1,1:7*npv+fb-1)     jac2(1:7*npv+fb-1,
7*npv+fb+1:7*npv+n+fb-1)  jac2(1:7*npv+fb-1,
7*npv+n+fb+1:7*npv+2*n);
          jac2(7*npv+fb+1:7*npv+n+fb-1,1:7*npv+fb-1)
jac2(7*npv+fb+1:7*npv+n+fb-1, 7*npv+fb+1:7*npv+n+fb-1)
jac2(7*npv+fb+1:7*npv+n+fb-1, 7*npv+n+fb+1:7*npv+2*n);
         jac2( 7*npv+n+fb+1:7*npv+2*n,1:7*npv+fb-1)  jac2(
7*npv+n+fb+1:7*npv+2*n,7*npv+fb+1:7*npv+n+fb-1) jac2(
7*npv+n+fb+1:7*npv+2*n, 7*npv+n+fb+1:7*npv+2*n)];

[eeq11,eeq22,eeq33,eeq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
equationcal
(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,delta
a,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,Ta,Ka,
Vref,Pm,H,mvv);

equationn = [eeq11(1:npv) eeq22(1:npv) eeq33(1:npv) eeq44(1:npv)
eeq55(1:npv) eeq66(1:npv) eeq77(1:npv) eeq88(1:fb-1) eeq88(fb+1:n)
eeq99(1:fb-1) eeq99(fb+1:n)]' ;
delta = delta + (errorr(1:npv));
 w = w + (errorr(npv+1:2*npv));
 Etq = Etq + (errorr(2*npv+1: 3*npv))';
 Etd = Etd + (errorr(3*npv+1:4*npv))';
 Efx = Efx + (errorr(4*npv+1:5*npv))';
 Iq = Iq + (errorr(5*npv+1:6*npv));
 Id = Id + (errorr(6*npv+1:7*npv));
 the(1:fb-1) = the(1:fb-1) + errorr(7*npv+1:7*npv+fb-1);
 the(fb+1:n) = the(fb+1:n) + errorr(7*npv+fb:7*npv+n-1);
 mv(1:fb-1)=mv(1:fb-1)+ errorr(7*npv+n:7*npv+n+fb-2);
 mv(fb+1:n)=mv(fb+1:n)+ errorr(7*npv+n+fb-1:7*npv+2*n-2);
 for i = 1:1:npv
   if (Efx(i)>= machine_data(i,15))&& (Efx(i)<= machine_data(i,14))
         Efd(i)= Efx(i);
     elseif Efx(i)< machine_data(i,15)
         Efd(i)= machine_data(i,15);
     elseif Efx(i)> machine_data(i,14)
          Efd(i)= machine_data(i,14);
     end
     end
[pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
[eeq11,eeq22,eeq33,eq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
   equationcal
   (Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,de
   ltaa,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,
   Ta,Ka,Vref,Pm,H,mvv);
```

```
equationn = [eeq11(1:npv) eeq22(1:npv) eeq33(1:npv) eeq44(1:npv)
   eeq55(1:npv) eeq66(1:npv) eeq77(1:npv) eeq88(1:fb-1)
   eeq88(fb+1:n) eeq99(1:fb-1) eeq99(fb+1:n)]' ;
accc= max(abs(equationn));
itrr = itrr+1 ;
end

[Pe] =  elepow(Etq,Etd,Id,Iq,npv,machine_data);
iteration(nt) = itrr;
nt = nt+1;
SDelta(1:npv,nt)= delta(1:npv);
SW(1:npv,nt)= w(1:npv);
SEFD(1:npv,nt)= Efd(1:npv);
SEQT(1:npv,nt)= Etq(1:npv);
SEDT(1:npv,nt)= Etd(1:npv);
SIQ(1:npv,nt)= Iq(1:npv);
SID(1:npv,nt)= Id(1:npv);
SMV(1:n,nt)= mv(1:n);
SANGLE(1:n,nt)= the(1:n);
SPE(1:npv,nt) = Pe(1:npv);
end
Ht=0;
Ht=sum(H);
for i =1:1:T+1
summ(i)=0;
summ1(i)=0;
for j=1:1:npv
summ1(i)=summ1(i)+H(j)*SW(j,i);
end
 WO(i) = summ1(i)/Ht;
end
for i=1:1:npv
 gamma(i,:)=SW(i,:)-WO(1,:);
end
%------------------------------------------------------------
y(fb,fb)= y(fb,fb)+y(fb,fbl);
y(fbl,fbl)= y(fbl,fbl)+y(fb,fbl);
y;
[theta , yy] = cart2pol(real(y),imag(y));

IDt  = zeros(npv,T+1);
IQt = zeros(npv,T+1);
differen = zeros(npv,T+1);
PEE = zeros(npv,T+1);
G= real(y);
B= imag(y);
for k = 1:1:T+1
    M2 = [eye(n,npv) zeros(n,npv);
     zeros(n,npv) eye(n,npv)];
    EE = [SEQT(:,k);SEDT(:,k)];
    Jkh=SDelta(:,k);
    D1=[diag(cos(jkh)) -diag(sin(jkh));
        diag(sin(jkh)) diag(cos(jkh))];
    D2=[diag(cos(SDelta(:,k))) diag(sin(SDelta(:,k)));
        -diag(sin(SDelta(:,k))) diag(cos(SDelta(:,k)))];
    LL = [ G+diag(a) -B+diag(b);
           B-diag(b)   G+diag(a)];

    xx1=Ra./(Ra.*Ra+Xtd.*Xtq);
    xx2=Xtd./(Ra.*Ra+Xtd.*Xtq);
    xx3=-Xtq./(Ra.*Ra+Xtd.*Xtq);
    xx4=Ra./(Ra.*Ra+Xtd.*Xtq);
    XX=[diag(xx1) diag(xx2);
    diag(xx3) diag(xx4)];
    ANI =inv(eye(2*npv)+D1*XX*D2*M2'*inv(LL)*M2)*D1*XX*EE;
    ANIQD = D2*ANI;
    IQt(:,k) = IqT;
    IDt(:,k) = IdT;
end
[PEE] = elepower(SEQT,SEDT,IQt,IDt,npv,machine_data,T);
for i=1:1:npv
differen(i,:)=Pm(1,i)- PEE(i,:);
end
PCOI = sum(differen);
[functv, PEval]=trap (hh,npv,T,Pm,PEE,H,Ht,PCOI,gamma);
KE = zeros(npv,T+1);
for i=1:1:npv
for j = 1:1:T+1
    KE(i,j)= (H(i)*power(gamma(i,j),2))/wb;
end
end
t=[0:hh:st];
KET = sum(KE);
PET = sum(PEval);
total = KET + PET;
[Wcr,tr] = max(PET);
matrixwcr = [Wcr*ones(1,tr)];
trr=[0:hh:tr*hh];
figure(10);
plot(t,KET,t,PET,t,total)
hold on
plot(trr,Wcr)
hold off
diftcr = total-Wcr;
[minVal,minInd]= min(abs(diftcr));
tcr = (minInd - 1)*hh;
```

E.3 MATLAB PROGRAM FOR TRANSIENT STABILITY NALYSIS OF 39 BUS 10 GENERATOR SYSTEM USING CRITICAL MACHINE ENERGY FUNCTION

```
clc
n = input('\n enter the number of buses')
f=60;
load line_data.txt
load transformer_data.txt
load slackbus_data.txt
load PVbus_data.txt
load PQbus_data.txt
load machine_data.txt
[m,o]=size(line_data);
for i =1:1:m
        k = line_data(i,1);
        l = line_data(i,2);
        r = line_data(i,3);
        z(k,l)= complex(r,line_data(i,4));
        z(l,k)= z(k,l);
        s(k,l) = (complex(0,line_data(i,5)))/2;
end
[g,h] = size(transformer_data);
for i =1:1:g
     k = transformer_data(i,1);
        l = transformer_data(i,2);
       rt =transformer_data(i,3);
        z(l,k)= z(k,l);
end
y = [zeros(n,n)];
for i =1:1:n
     for j =1:1:n
         if z(i,j)==0
                 c(i,j)=0;
         else
             c(i,j)= (1/z(i,j));
             end
     end
end
for i=1:1:n
    for j=1:1:n

        y(i,i)= y(i,i)+c(i,j)+ s(i,j);
    end

 end
Ybus = y;
[mp,op]=size(PVbus_data);
[mq,oq] = size(PQbus_data);
mv = [slackbus_data(1,2);PVbus_data(1:mp,2);ones(1,mq)'];
```

```
 th = [slackbus_data(1,3); zeros(1,mp+mq)'];
 pg = [ 0;PVbus_data(1:mp,3);PQbus_data(1:mq,4)];
 qg = [zeros(1,mp+1)';PQbus_data(1:mq,5)];
 pd = [slackbus_data(1,4);PVbus_data(1:mp,4);PQbus_data(1:mq,2)];
 qd = [slackbus_data(1,5);PVbus_data(1:mp,5);PQbus_data(1:mq,3)];
 npv = mp+1;
 P = n-1;
 Q = n-npv;
 r = P+Q;
 itr=0;
 acc=1;
[theta , yy] = cart2pol(realy,imagy);
 while acc > 1e-12
   [pcal,qcal]=powers(mv,yy,theta,th,n,n);

   accpq =[ accp(2:n)  accq(npv+1:n)];
 for i=2:1:n
   for k = 2:1:n
     if i==k
       H(i,i)= ((-1)*qcal(i) - (mv(i)*mv(i)*imagy(i,i)));
     else
       H(i,k)= (mv(i)*mv(k)*yy(i,k)*sin(th(i)-theta(i,k)-th(k)));
     end
    end
 end
 for i=npv+1:1:n
    for k = 2:1:n
        if i==k
           M(i,i)= (pcal(i)- (mv(i)*mv(i)* realy(i,i)));
        else
           M(i,k) =((-1)* (mv(i)*mv(k)*yy(i,k)*cos(th(k)+theta(i,k)-
 th(i))));
         end
     end
 end
 for i = 2:1:n
   for k = npv+1:1:n
       if i==k
         N(i,i)= (pcal(i)+ (mv(i)*mv(i)*realy(i,i)));
       else
         N(i,k) = (mv(i)*mv(k)*yy(i,k)*cos(theta(i,k)+th(k)-th(i)));
       end
     end
 end
 for i=npv+1:1:n
    for k = npv+1:1:n
       if i==k
         qcal(i);
         L(i,i)= (qcal(i)-(imagy(i,i)*mv(i)*mv(i)));
        else
         L(i,k)= (mv(i)*mv(k)*yy(i,k)*sin(th(i)-th(k)-theta(i,k)));
        end
     end
 end
 jac = [ H(2:n,2:n) N(2:n,npv+1:n) ; M(npv+1:n,2:n)
      L(npv+1:n,npv+1:n)];
         thnew = th + [ 0; pq(1:n-1)];
         mvnew = mv + mv.*[zeros(1,npv)';pq(n:r)];
         mv = mvnew;
         th = thnew;
         itr = itr+1;
 end
 [pcal,qcal]=powers(mv,yy,theta,th,npv,n);
 pgen(1) = pcal(1)+ slackbus_data(1,4);
 qgen(1) = qcal(1)+ slackbus_data(1,5);
 for i = 2:1:npv
     k = i-1;
     qgen(i) = qcal(i) + PVbus_data(k,5);
 end
 th = (th*180)/pi;
 Pg = [ pgen(1); pg(2:n)];
 Qg = [ qgen(1:npv)';qg(npv+1:n)];
 pd;
 qd ;
 fprintf ('\n  \tvoltage \tangle \t\tPg \t\tQg \t\t\tPd \t\tQd')
 answer = [zeros(n,6)];
 answer = [ mv th Pg  Qg pd qd]
 the = (th*pi)/180
 %---------------------------------------------------------
 %solving for initial values of state variables
 [Vcart,Thcart] = pol2cart(the,mv);
 Igo= zeros (npv,1);
 Eqo= zeros (npv,1);
 Ido = zeros (npv,1);
 Iqo = zeros (npv,1);
 for i =1:1:npv
   Igo(i)= complex(Pg(i),-Qg(i))/ (complex(Vcart(i),-Thcart(i)));
   Eqo(i)= complex(Vcart(i),Thcart(i))
      +((complex(machine_data(i,2),machine_data(i,5)))*Igo(i));
   Ido(i)= (-abs(Igo(i))* sin(angle(Eqo(i))-angle(Igo(i))));
   Iqo(i)= abs(Igo(i))*cos(angle(Eqo(i))-angle(Igo(i)));
   Vdo(i)= (-mv(i)*sin(angle(Eqo(i))-the(i)));
   Vqo(i)= mv(i)*cos(angle(Eqo(i))-the(i));
   Efdo(i)= abs(Eqo(i))- ((machine_data(i,3)- machine_data(i,5))*
      Ido(i));
   Etqo(i)= Efdo(i)+((machine_data(i,3)- machine_data(i,4))*
      Ido(i));
   Etdo(i)= (-1)*((machine_data(i,5)- machine_data(i,6))* Iqo(i));
   Peo(i)= Etqo(i)*Iqo(i) + Etdo(i)*Ido(i)+((machine_data(i,4)-
      machine_data(i,6))*Ido(i)*Iqo(i));
   Pm(i) = Peo(i);
   Vref(i)= (Efdo(i)/(machine_data(i,12)))+ mv(i);
```

```
end
fb = input ('enter the index of faulted bus')
fprintf(' the line connected to %4i that is tripped',fb)
fbl = input('\n enter its index')
Iq = Iqo;
Etd = Etdo;
Etq = Etqo;
Eq = Eqo;
mv;
Ra = [machine_data(1:npv,2)];
Xtq = [machine_data(1:npv,6)];
Xtd = [machine_data(1:npv,4)];
delta = angle(Eq);
itr = 0;

G11=[zeros(npv,npv)];
G12=[zeros(npv,npv)];
G21=[zeros(npv,npv)];
G22=[zeros(npv,npv)];
G13=[zeros(npv,n)];
G14=[zeros(npv,n)];
G23=[zeros(npv,n)];
G24=[zeros(npv,n)];
G31=[zeros(n,npv)];
G32=[zeros(n,npv)];
G41=[zeros(n,npv)];
G42=[zeros(n,npv)];
G33=[zeros(n,n)];
G34=[zeros(n,n)];
G43=[zeros(n,n)];
G44=[zeros(n,n)];
maxerror = 1;
while(maxerror > 1e-6)
    [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
  for i = 1:1:n
       for j = 1:1:n
        if (i<=npv) && (j<=npv)
         if i==j
        G11(i,i)= (-Xtq(i));
        G12(i,i)= (-Ra(i));
        G21(i,i)= (-Ra(i));
        G22(i,i)= (Xtd(i));
         else
        G11(i,j)=0;
        G12(i,j)=0;
        G21(i,j)=0;
        G22(i,j)=0;
         end
        end
        if (i<=npv) && (j<=n)
            if (i==j)
```

```
          G13(i,i)= (- mv(i)*cos(the(i)-angle(Eq(i))));
          G14(i,i)= (-sin(the(i)-angle(Eq(i))));
          G23(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
          G24(i,i)= (-cos(the(i)-angle(Eq(i))));
              elseif (i~=j)
          G13(i,j)=0;
          G14(i,j)=0;
          G23(i,j)=0;
          G24(i,j)=0;
              end
          end

              if(i<=n) && (j<=npv)
              if(i==j)
          G31(i,i)= (mv(i)*cos(the(i)-angle(Eq(i))));
          G32(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
          G41(i,i)= (mv(i)*sin(the(i)- angle(Eq(i))));
          G42(i,j)=(-mv(i)*cos(the(i)-angle(Eq(i))));
              else
          G31(i,j)=0;
          G32(i,j)=0;
          G41(i,j)=0;
          G42(i,j)=0;
              end
              end
          if (i<=n)&&(j<=n)
          if(i==j)
               if(i<=npv)
G33(i,i)= (-mv(i)*Iq(i)*sin(the(i)- angle(Eq(i))))+
     (mv(i)*Id(i)*cos(the(i)- angle(Eq(i))))+ qcal(i)+
     (mv(i)*mv(i)*imag(y(i,i)));
G34(i,i)= (Iq(i)*cos(the(i)-angle(Eq(i))))+ (Id(i)*sin(the(i)-
     angle(Eq(i))))- 2*a(i)*mv(i)- mv(i)* real(y(i,i))-int(i);
G43(i,i)= (mv(i)*Iq(i)*cos(the(i)- angle(Eq(i))))+
     (mv(i)*Id(i)*sin(the(i)- angle(Eq(i))))-pcal(i)+
     (mv(i)*mv(i)*real(y(i,i)));
G44(i,i)= (Iq(i)*sin(the(i)- angle(Eq(i))))- (Id(i)*cos(the(i)-
     angle(Eq(i))))- 2*b(i)*mv(i)+(mv(i)*imag(y(i,i))+intf(i));
                  Else
G33(i,i)= (-qcal(i)-(mv(i)*mv(i)*imag(y(i,i))));
G34(i,i)= (2*a(i)*mv(i))+(mv(i)*real(y(i,i)))+int(i);
G43(i,i)= pcal(i)-(mv(i)*mv(i)*real(y(i,i)));
G44(i,i)= (2*b(i)*mv(i))-(mv(i)*imag(y(i,i)))-intf(i);
               end
          else
                    if(i<=npv)
G33(i,j)= (mv(i)*mv(j)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
G34(i,j)= (-mv(i)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G43(i,j)=(mv(i)*mv(j)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G44(i,j)=(mv(i)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
                   elseif(i>npv)
```

```
G33(i,j)= (-mv(i)*mv(j)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
G34(i,j)=  (mv(i)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G43(i,j)=(-mv(i)*mv(j)*yy(i,j)*cos(the(j)-the(i)+theta(i,j)));
G44(i,j)=(-mv(i)*yy(i,j)*sin(the(j)-the(i)+theta(i,j)));
                    End
            end
          end
      end
  end
         jac1 = [G11 G12 G13 G14;
                 G21 G22 G23 G24;
                 G31 G32 G33 G34;
                 G41 G42 G43 G44];
  jacob = [jac1(1:2*npv+fb-1,1:2*npv+fb-1) jac1(1:2*npv+fb-
  1,2*npv+fb+1:2*npv+n+fb-1) jac1(1:2*npv+fb-
  1,2*npv+n+fb+1:2*npv+2*n);
           jac1(2*npv+fb+1:2*npv+n+fb-1,1:2*npv+fb-1)
  jac1(2*npv+fb+1:2*npv+n+fb-1,2*npv+fb+1:2*npv+n+fb-1)
  jac1(2*npv+fb+1:2*npv+n+fb-1,2*npv+n+fb+1:2*npv+2*n);
           jac1(2*npv+n+fb+1:2*npv+2*n,1:2*npv+fb-1)
  jac1(2*npv+n+fb+1:2*npv+2*n,2*npv+fb+1:2*npv+n+fb-1)
  jac1(2*npv+n+fb+1:2*npv+2*n,2*npv+n+fb+1:2*npv+2*n)];

  [equation]=equcal(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qc
  al,n,npv,fb);

      Id = Id + error(npv+1:2*npv);
      the(1:fb-1) = the(1:fb-1) + error(2*npv+1:2*npv+fb-1);
      the(fb+1:n) = the(fb+1:n) + error(2*npv+fb:2*npv+n-1);
      mv(1:fb-1)=mv(1:fb-1)+ error(2*npv+n:2*npv+n+fb-2);
      mv(fb+1:n)=mv(fb+1:n)+ error(2*npv+n+fb-1:2*npv+2*n-2);
      itr = itr+1;
  [equation]=equcal(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qc
  al,n,npv,fb);
      maxerror = (max(abs(equation)));
  end

  w = zeros(npv,1);
  st=input('enter the value of simulation time')
  wb = 2*pi*f;
  hh = input('enter the value of step of integration')
  T = st/hh;
  SDelta = zeros(npv,T+1);
  WO = zeros(1,T+1);
  SEFD = zeros(npv,T+1);
  SEQT = zeros(npv,T+1);
  SEDT  = zeros(npv,T+1);
  SW = zeros(npv,T+1);
  SID  = zeros(npv,T+1);
  SIQ = zeros(npv,T+1);
  SMV = zeros(n,T+1);
```

```
SANGLE = zeros(n,T+1);
SPE = zeros(npv,T+1);
gamma = zeros(npv,T+1);
dif = zeros(npv,T+1);

Efd = Efdo;
Etq = Etqo;
Etd = Etdo;
Pe = Peo;
Efx = Efdo;
nt = 1;
itr=0;
iteration = zeros(T);
H =  [machine_data(1:npv,7)];
Td =  [machine_data(1:npv,8)];
Tq =  [machine_data(1:npv,9)];
Ta =  [machine_data(1:npv,13)];
Ka =  [machine_data(1:npv,12)];
Xq =  [machine_data(1:npv,5)];
Xd =  [machine_data(1:npv,3)];
SDelta(1:npv,1)= delta(1:npv);
SW(1:npv,1)= w(1:npv);
SEFD(1:npv,1)= Efdo(1:npv);
SEQT(1:npv,1)= Etqo(1:npv);
SEDT(1:npv,1)= Etdo(1:npv);
SIQ(1:npv,1)= Iq(1:npv);
SID(1:npv,1)= Id(1:npv);
SMV(1:n,1)= mv;
SANGLE(1:n,1)= the;
SPE(1:npv,1)= Peo;
%-------------------------------------------------------------
while nt <= T
    itrr = 0;
    deltaa = delta;
    ww = w;
    Etqq = Etq;
    Etdd = Etd;
    Efdd = Efd;
    Efxx = Efx;
    Idd= Id;
    Iqq = Iq;
    mvv = mv;
accc=1;
while accc>1e-12
  [pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
  [jac2]=jacobbian(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,q
    cal,int,intf,n,npv,hh,Xd,Xq,wb,Td,Tq,Ta,Ka,H,Ybus,yy,theta);

jacobb =[ jac2(1:7*npv+fb-1,1:7*npv+fb-1)     jac2(1:7*npv+fb-1,
7*npv+fb+1:7*npv+n+fb-1)  jac2(1:7*npv+fb-1,
7*npv+n+fb+1:7*npv+2*n);
```

```
          jac2(7*npv+fb+1:7*npv+n+fb-1,1:7*npv+fb-1)
jac2(7*npv+fb+1:7*npv+n+fb-1, 7*npv+fb+1:7*npv+n+fb-1)
jac2(7*npv+fb+1:7*npv+n+fb-1, 7*npv+n+fb+1:7*npv+2*n);
         jac2( 7*npv+n+fb+1:7*npv+2*n,1:7*npv+fb-1)  jac2(
7*npv+n+fb+1:7*npv+2*n,7*npv+fb+1:7*npv+n+fb-1) jac2(
7*npv+n+fb+1:7*npv+2*n, 7*npv+n+fb+1:7*npv+2*n)];

[eeq11,eeq22,eeq33,eeq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
equationcal
(Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,delta
a,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,Ta,Ka,
Vref,Pm,H,mvv);

equationn = [eeq11(1:npv) eeq22(1:npv) eeq33(1:npv) eeq44(1:npv)
eeq55(1:npv) eeq66(1:npv) eeq77(1:npv) eeq88(1:fb-1) eeq88(fb+1:n)
eeq99(1:fb-1) eeq99(fb+1:n)]' ;
delta = delta + (errorr(1:npv));
 w = w + (errorr(npv+1:2*npv));
 Etq = Etq + (errorr(2*npv+1: 3*npv))';
 Etd = Etd + (errorr(3*npv+1:4*npv))';
 Efx = Efx + (errorr(4*npv+1:5*npv))';
 Iq = Iq + (errorr(5*npv+1:6*npv));
 Id = Id + (errorr(6*npv+1:7*npv));
 the(1:fb-1) = the(1:fb-1) + errorr(7*npv+1:7*npv+fb-1);
 the(fb+1:n) = the(fb+1:n) + errorr(7*npv+fb:7*npv+n-1);
 mv(1:fb-1)=mv(1:fb-1)+ errorr(7*npv+n:7*npv+n+fb-2);
 mv(fb+1:n)=mv(fb+1:n)+ errorr(7*npv+n+fb-1:7*npv+2*n-2);
 for i = 1:1:npv
   if (Efx(i)>= machine_data(i,15))&& (Efx(i)<= machine_data(i,14))
         Efd(i)= Efx(i);
     elseif Efx(i)< machine_data(i,15)
         Efd(i)= machine_data(i,15);
     elseif Efx(i)> machine_data(i,14)
          Efd(i)= machine_data(i,14);
     end
     end
[pcal, qcal,int,intf]=powerint(mv,yy,theta,the,n,n);
[eeq11,eeq22,eeq33,eq44,eeq55,eeq66,eeq77,eeq88,eeq99] =
   equationcal
   (Etd,Etq,mv,the,delta,Ra,Id,Iq,Xtd,Xtq,a,b,pcal,qcal,n,npv,fb,de
   ltaa,w,ww,hh,Etqq,Etdd,Xd,Xq,Iqq,Idd,wb,Efx,Efxx,Efd,Efdd,Td,Tq,
   Ta,Ka,Vref,Pm,H,mvv);
equationn = [eeq11(1:npv) eeq22(1:npv) eeq33(1:npv) eeq44(1:npv)
   eeq55(1:npv) eeq66(1:npv) eeq77(1:npv) eeq88(1:fb-1)
   eeq88(fb+1:n) eeq99(1:fb-1) eeq99(fb+1:n)]' ;
accc= max(abs(equationn));
itrr = itrr+1 ;
end

[Pe] =  elepow(Etq,Etd,Id,Iq,npv,machine_data);
iteration(nt) = itrr;
```

```
nt = nt+1;
SDelta(1:npv,nt)= delta(1:npv);
SW(1:npv,nt)= w(1:npv);
SEFD(1:npv,nt)= Efd(1:npv);
SEQT(1:npv,nt)= Etq(1:npv);
SEDT(1:npv,nt)= Etd(1:npv);
SIQ(1:npv,nt)= Iq(1:npv);
SID(1:npv,nt)= Id(1:npv);
SMV(1:n,nt)= mv(1:n);
SANGLE(1:n,nt)= the(1:n);
SPE(1:npv,nt) = Pe(1:npv);
end
Ht=0;
Ht=sum(H);
for i =1:1:T+1
summ(i)=0;
summ1(i)=0;
for j=1:1:npv
summ1(i)=summ1(i)+H(j)*SW(j,i);
end
 WO(i) = summ1(i)/Ht;
end
for i=1:1:npv
 gamma(i,:)=SW(i,:)-WO(1,:);
end
%---------------------------------------------------------
y(fb,fb)= y(fb,fb)+y(fb,fbl);
y(fbl,fbl)= y(fbl,fbl)+y(fb,fbl);
[theta , yy] = cart2pol(real(y),imag(y));

IDt  = zeros(npv,T+1);
IQt = zeros(npv,T+1);
differen = zeros(npv,T+1);
PEE = zeros(npv,T+1);
G= real(y);
B= imag(y);
for k = 1:1:T+1
    M2 = [eye(n,npv) zeros(n,npv);
     zeros(n,npv) eye(n,npv)];
    EE = [SEQT(:,k);SEDT(:,k)];
    Jkh=SDelta(:,k);
    D1=[diag(cos(jkh)) -diag(sin(jkh));
        diag(sin(jkh)) diag(cos(jkh))];
    D2=[diag(cos(SDelta(:,k))) diag(sin(SDelta(:,k)));
        -diag(sin(SDelta(:,k))) diag(cos(SDelta(:,k)))];
    LL = [ G+diag(a) -B+diag(b);
           B-diag(b)   G+diag(a)];

    xx1=Ra./(Ra.*Ra+Xtd.*Xtq);
    xx2=Xtd./(Ra.*Ra+Xtd.*Xtq);
    xx3=-Xtq./(Ra.*Ra+Xtd.*Xtq);
    xx4=Ra./(Ra.*Ra+Xtd.*Xtq);
    XX=[diag(xx1) diag(xx2);
    diag(xx3) diag(xx4)];
    ANI =inv(eye(2*npv)+D1*XX*D2*M2'*inv(LL)*M2)*D1*XX*EE;
    ANIQD = D2*ANI;
    IqT=ANIQD(1:npv);
    IdT=ANIQD(npv+1:2*npv);
    IQt(:,k) = IqT;
    IDt(:,k) = IdT;
end
[PEE] = elepower(SEQT,SEDT,IQt,IDt,npv,machine_data,T);
for i=1:1:npv
differen(i,:)=Pm(1,i)- PEE(i,:);
end
PCOI = sum(differen);
[functv, PEval]=trap (hh,npv,T,Pm,PEE,H,Ht,PCOI,gamma);
KE = zeros(npv,T+1);
for i=1:1:npv
for j = 1:1:T+1
    KE(i,j)= (H(i)*power(gamma(i,j),2))/wb;
end
end
t=[0:hh:st];
KET = sum(KE);
PET = sum(PEval);
[pks,locs] = findpeaks(PET);
 [SCRA,indSCRA] = sort(CRA,'descend');
[criticalangle,indcrit] = max(abs(diff(SCRA)));
critgrp=zeros(1,indcrit);
i=1;
while i<=indcrit
critgrp(i) = indSCRA(i);
i=i+1;
end
CritPE=PEval(critgrp,:);
CritKE=KE(critgrp,:);
[inn,inm]=size(critgrp);
if inm==1
    CPET=CritPE;
    CKET=CritKE;
else
    CPET=sum(CritPE);
    CKET=sum(CritKE);
end
Ctotal=CPET+CKET;
[Cpks,Clocs] = findpeaks(CPET);
CWcr=Cpks(1);
Ctr=Clocs(1);
Cmatrixwcr = [CWcr*ones(1,Ctr)];
Ctrr=[0:hh:Ctr*hh];
figure(10);
```

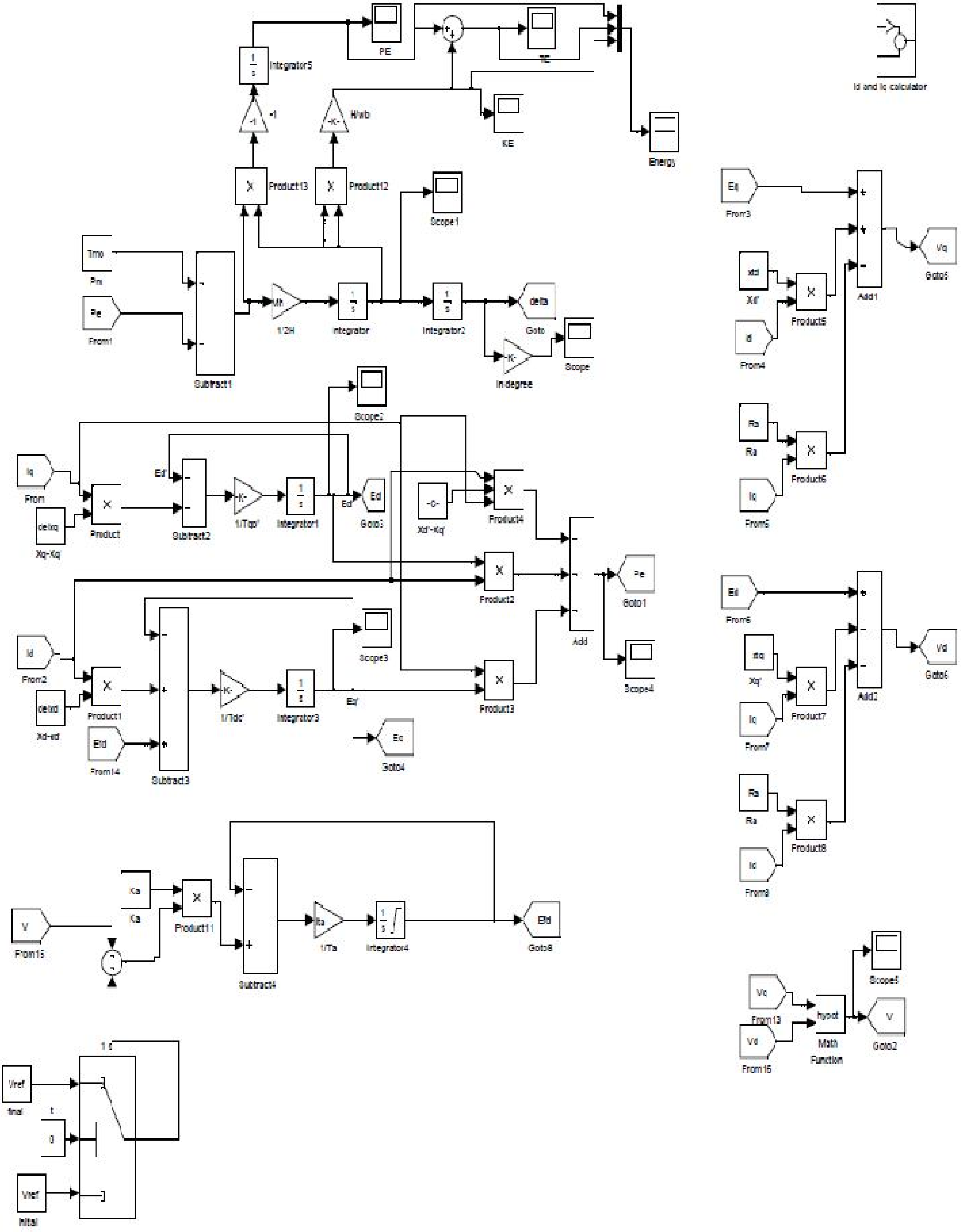
```
plot(t,CPET,t,Ctotal)
hold on
plot(Ctrr,CWcr)
hold off
for i=1:1:npv
    CTHETA(i,:)=SDelta(i,:)-COI(1,:);
end
figure(2);
plot(t,CTHETA);
```

E.4 MATLAB CODING AND SIMULINK MODEL FOR SINGLE MACHINE INFINITE BUS SYSTEM

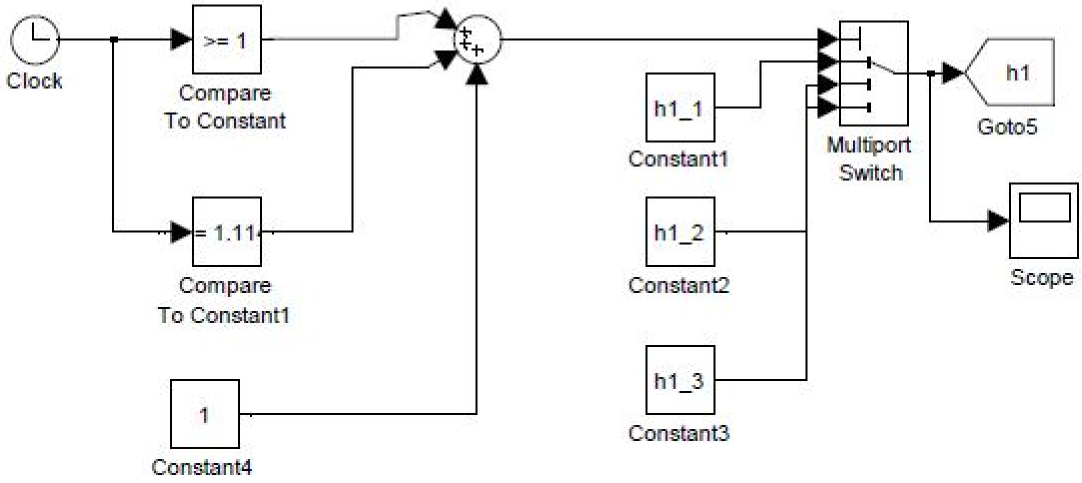
```
clc;
Vt=1.05;
Eb=1.0;
deltao=1.0734;
Etqo=.9699;
Etdo=-.2316;
Ra=.00327;
xd=1.7572;
xq=1.5845;
xtd=.4245;
xtq=1.04;
Tdo=6.66;
Tqo=.44;
H=3.542;
f=50;
wo=2*pi*f;
Efdo=1.4794;
Efdmax = 6;
Efdmin = -6;
delxq = xq-xtq;
delxd = xd-xtd;
delxtqd = xtd-xtq;
Vref=1.054;
Ka= 400;
Ta= 0.025;
Tmo= 0.6011;
Teo=Tmo;
%----------------------------------------------------------
zt=.1364j;
y1=.1184j;
zb=.13636j;
y2=.1184j;
y3=0;
z2=0;
z1=(.08593+.8125j)/2;
Z2=1/(y2+1/zb)+z2;
Z1=1/(y3+1/Z2)+z1;
```

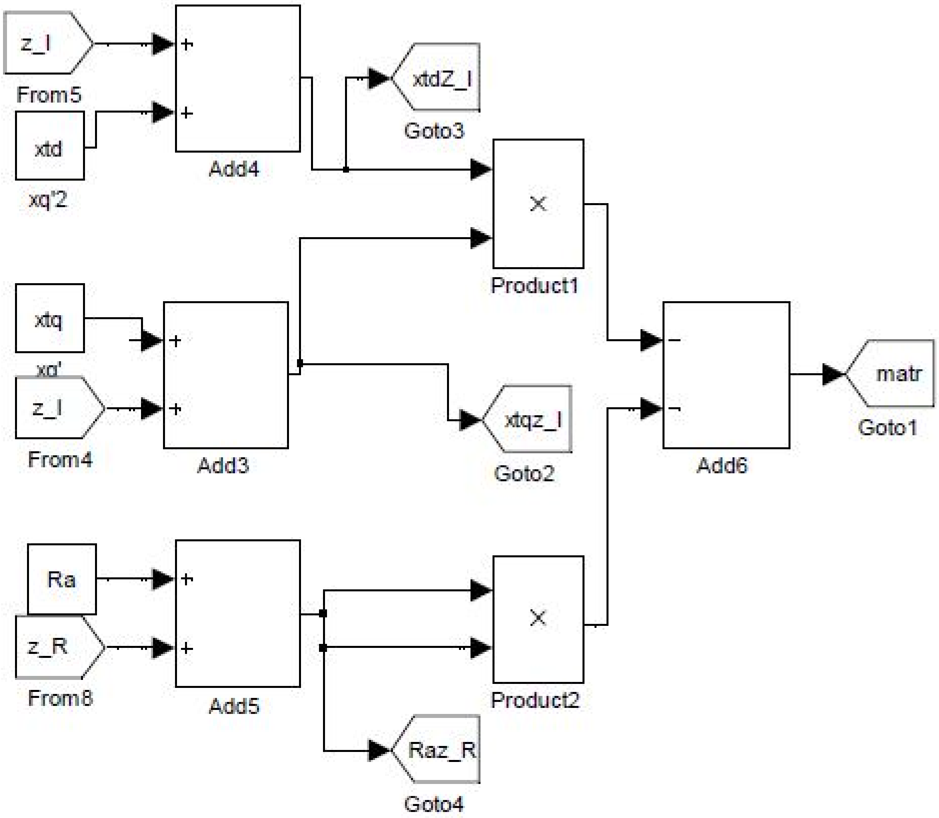
```
Z=1/(y1+1/Z1)+zt;
H3=(1/y2)/(1/y2+zb);
H2=1/(1+y3*z2);
H1=(1/y1)/(1/y1+Z1);
h=H1*H2*H3;
z_R_1=real(Z);
z_I_1=imag(Z);
h1_1=real(h);
h2_1=imag(h);
%--------------------------------------
zt=.1364j;
y1=.1184j/2;
zb=.13636j;
y2=.1184j/2;
y3=0;
z2=0;
z1=(.08593+.8125j);
Z=1/(y1+1/Z1)+zt;
H3=(1/y2)/(1/y2+zb);
H2=1/(1+y3*z2);
H1=(1/y1)/(1/y1+Z1);
h=H1*H2*H3;
z_R_3=real(Z);
z_I_3=imag(Z);
Mh = (0.5*wo/H);
ita = 1/Ta;
itdo = 1/Tdo;
itqo = 1/Tqo;
sim 'smib'
```

MATLAB SIMULINK MODEL OF SMIB SYSTEM

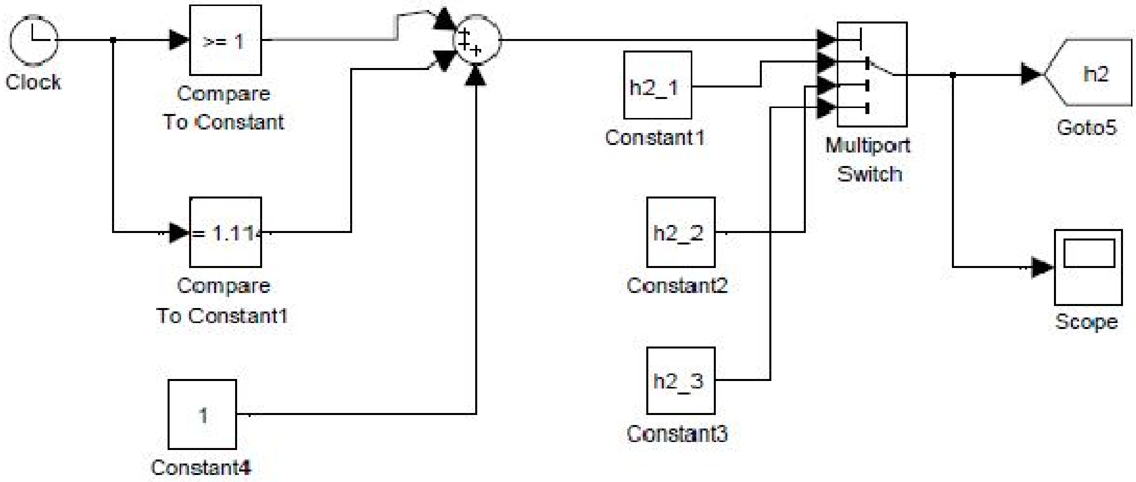


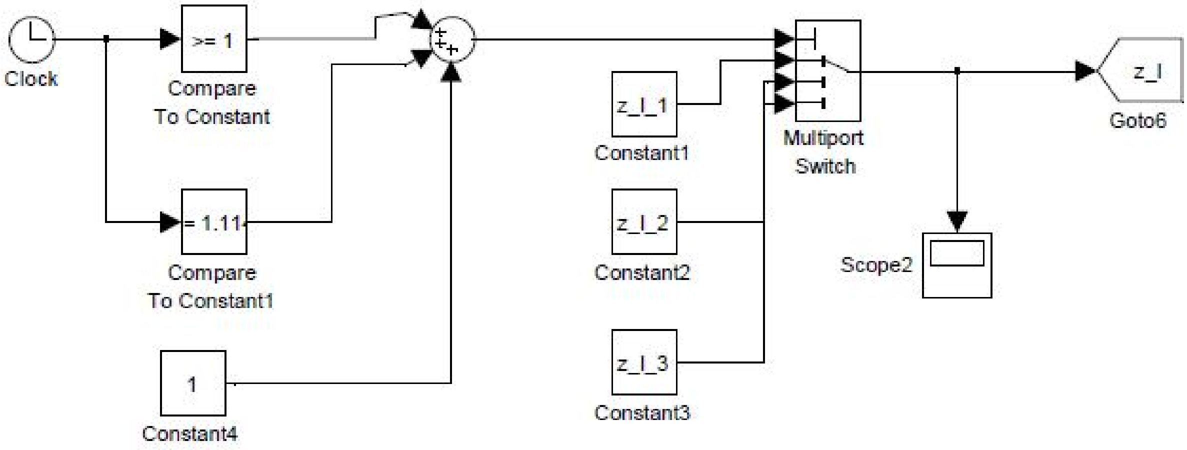
SUBSYSTEMS OF SMIB SIMULINK MODEL

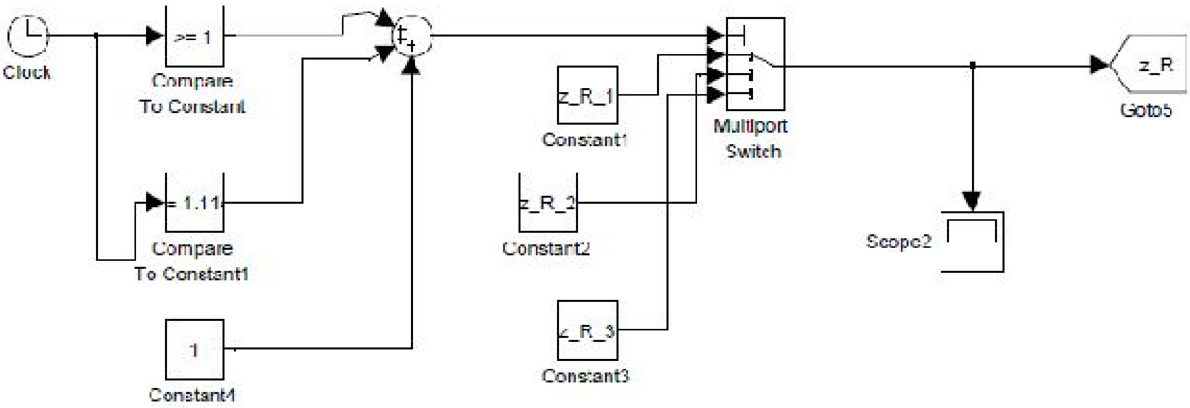




SUBSYSTEMS OF SMIB SIMULINK MODEL







E.5 MATLAB CODING FOR TRANSIENT STABILITY ANALYSIS OF SMIB SYSTEM WITH SVC

```
clc;
H=5;
f=60;
xd=0.3;
M = H/(180*f);
tc = input('enter the fault clearing time')
t = input('enter simulation time')
Bsvcmax = input('enter the maximum value of Bsvc')
Bsvcmin = -0.5*Bsvcmax;
hh=0.001;
xt=0.1;
xl=0.4;
E = 1.2065;
deltao = 44.26;
X10 = xd+xt+0.5*xl;
X20 = 0.5*xl;
X120 = X10+X20;
Peo = Eb*E*sind(deltao)/X120;
i=1;
deldelta=zeros(1,m);
Pa=zeros(1,m);
Pe=zeros(1,m);
delta = zeros(1,m+1);
BSVC = zeros(1,m);
w = zeros(1,m);
wo=0;
BSVC(1)=0;
swg=0;
while i<n+1
    if i==1
        delta(1)=deltao;
        w(1)=wo;
        Pe(1)=0;
        Pa(1) = 0.5*(Pm-Pe(1));
        delPam(i)= ((hh*hh*Pa(i))/M);
        delta(i+1)=delta(i)+deldelta(i);
        Deltar = delta*pi/180;
        w(i+1)= Deltar(i+1)-Deltar(i)/hh ;

    else
        Pe(i)=0;
        Pa(i)=Pm-Pe(i);
        delPam(i)= ((hh*hh*Pa(i))/M);
        deldelta(i)=deldelta(i-1)+delPam(i);
        w(i+1)= (Deltar(i+1)-Deltar(i))/hh ;
    end
            i=i+1;
            if w(i)<=0.5 && swg==0
                Bsvc=0;
                BSVC(i)=Bsvc;
                swg=0;
            end
            if w(i)>0.5 && swg==0
                swg=1;
            end
            if w(i)<=-0.5 && swg==1
                swg=2;
            end
            if swg==1
                Bsvc=Bsvcmax;
                BSVC(i)=Bsvc;
            end
            if Bsvc >= Bsvcmax
                Bsvc=Bsvcmax;
                BSVC(i)=Bsvc;
            elseif Bsvc<=Bsvcmin
                Bsvc=Bsvcmin;
                BSVC(i)=Bsvc;
            else

                BSVC(i)=Bsvc;
            end

end
Pe_tc0 =0;
Pa_tc0=Pm-Pe_tc0;
X1 = xd+xt+0.5*xl;
X2 = xl;
X12 = X1+X2-Bsvc*X1*X2;
Pe_tc=(E*Eb*sind(delta(n+1)))/X12;
Pe(i)=Pe_tc;
Pa_tc = Pm-Pe_tc;
Pa_tcavg=0.5*(Pa_tc0+Pa_tc);
Pa(i)=Pa_tcavg;
deldelta_tc0=deldelta(n);
deldelta_tcavg = deldelta(n)+delPam(i);
delta_tcavg=delta(n+1)+deldelta_tcavg;
i=n+2;
while i<m+1
    if i==n+2;
         delta(i)=delta_tcavg;
         Deltar = delta*pi/180;
         w(i)=(Deltar(i)-Deltar(i-1))/hh;
         deldelta(i-1)=deldelta_tcavg;
          if w(i)<0.5 && swg==0
                Bsvc=0;
                BSVC(i)=Bsvc;
                swg=0;
```

```
        end
        if w(i)>=0.5 && swg==0
            swg=1;
        end
        if w(i)<=-0.5 && swg==1
            swg=2;
        end
        if swg==1
            Bsvc=Bsvcmax;
            BSVC(i)=Bsvc;
        end

         if Bsvc >= Bsvcmax
            Bsvc=Bsvcmax;
            BSVC(i)=Bsvc;
        elseif Bsvc<=Bsvcmin
            Bsvc=Bsvcmin;
            BSVC(i)=Bsvc;
end

   X1 = xd+xt+0.5*xl;
   X2= xl;
   X12=X1+X2-Bsvc*X1*X2;
   Pe(i)=(E*Eb*sind(delta(i)))/X12;
   Pa(i)=Pm-Pe(i);
   delPam(i)=((hh*hh*Pa(i))/M);
   deldelta(i)=deldelta(i-1)+delPam(i);
   delta(i+1)=delta(i)+deldelta(i);
   Deltar = delta*pi/180;
   w(i)=(Deltar(i+1)-Deltar(i))/hh;
        if w(i)<0.5 && swg==0
            Bsvc=0;
            BSVC(i)=Bsvc;
            swg=0;
        end
        if w(i)>=0.5 && swg==0
            swg=1;
        end
        if w(i)<=-0.5 && swg==1
            swg=2;
        end
        if swg==1
            Bsvc=Bsvcmax;
            BSVC(i)=Bsvc;
        end

        if Bsvc >= Bsvcmax
            Bsvc=Bsvcmax;
            BSVC(i)=Bsvc;
        elseif Bsvc<=Bsvcmin
            Bsvc=Bsvcmin;
```

```
                BSVC(i)=Bsvc;
            else

                BSVC(i)=Bsvc;
             end
end
tt=0:hh:t;
th=0:hh:t-hh;
figure(1);
plot(tt,delta)
figure(2);
plot(th,BSVC)
```