2014

Power System Differential Model with Application to Grid Dynamic Simulation

Boyu Wang
Louisiana State University and Agricultural and Mechanical College

POWER SYSTEM DIFFERENTIAL MODEL WITH APPLICATION TO GRID DYNAMIC SIMULATION

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

Department of Electrical and Computer Engineering

by
Boyu Wang
B.S., Harbin Institute of Technology at Weihai, 2012
August 2014

ACKNOWLEDGEMENTS

First of all, I would like to thank to my major professor Dr. Mehraeen to give me the opportunity working with me. Thanks to his timely support and guide me on the way of research. He encourages and inspires me to explore more. Without his knowledge, I cannot finish this work.

I also would like to thank to Dr. Vaidyanathan and Mr. McAnelly to be my committee member. Thank you for your valuable time.

Moreover, I am grateful to all my colleagues and friends. Specially, thanks to Shanna who is helping me understand the code and previous work.

Last but not the least, I would like to thank to my parents and my girlfriend Wan, who is always by my side, care about my lives and support me as much as you can.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Nonlinearity of power system is always one of the difficulties when dealing with dynamic simulation of power systems. Solving differential-algebraic equations representing power systems are difficult without losing nonlinearity, especially for large power systems. This thesis shows an alternative method to solve nonlinear dynamical power system by producing a purely differential representation of the power systems. This new representation converts the algebraic equations to differential equations in order to have an absolute differential system. By using Runge-Kutta algorithm to solve this differential system, the results of the power system simulations are compared to trapezoidal integration algorithm commonly used to solve the differential-algebraic equations.

In this thesis, IEEE 14-bus system and IEEE 118-bus system are tested with both classical generator model generator model and two-axis generator model in MATLAB. The proposed algorithm shows significantly faster convergence comparted to trapezoidal integration method in larger power systems. It is a great improvement to shorten the simulation time in while keeping the same accuracy in large power systems.

Keywords: Runge-Kutta algorithm, differential-algebraic equation, trapezoidal, classical generator model, two-axis model.

CHAPTER 1. INTRODUCTION

This chapter introduces the conventional differential-algebraic (DA) model that describes the dynamic behavior of the power system. The transient stability analysis of the power system requires to solving the DA equations. One of the obstacles to solve such nonlinear equations is that there are no known methods available to analytically find their time domain solution, and thus, numerical methods are greatly utilized to solve the differential-algebraic equations.

This chapter has three sections. The first section briefly introduces DA model using the generator in a general mathematical form. The second section shows the differential equations describing two-axis generator model in the power system. The last section talks about the algebraic equations when dealing with multi-machine power system. Multi-machine power system means many synchronous generators of different buses are interconnected together by transformers and transmission lines [1].

1.1 Power System Differential-Algebraic Model

The differential-algebraic equations (DAE) model the power system as follows:

$$\begin{cases} \dot{y} = f(x, y, u) \\ 0 = g(x, y) \end{cases} \tag{1}$$

In (1), f represents all dynamic characteristics of the power system which is mainly generator dynamics. g explains the power balance equations for both generator buses as well as non-generator buses. The system state variables, such as generator angle δ, generator speed ω and other generator dynamical parameters are shown by y. Vector x denotes all system algebraic variables, including bus voltages V and bus angles θ. Input vector u represents inputs to the power system such as generator input torque and reference voltage V_{ref}.

1.2 Differential Equations

For transient stability analysis it is necessary to introduce, two-axis model which is more accurate than the classical generator model [2] for extended transient analysis. Grainger and Stevenson (1994) gave more details about this model, which is considered in the following [3].

Unlike round-rotor machine, the salient-pole machine has a narrower air gap along the direct axis than the quadrature axis for each pole. The two-axis generator model is shown in Figure 1.

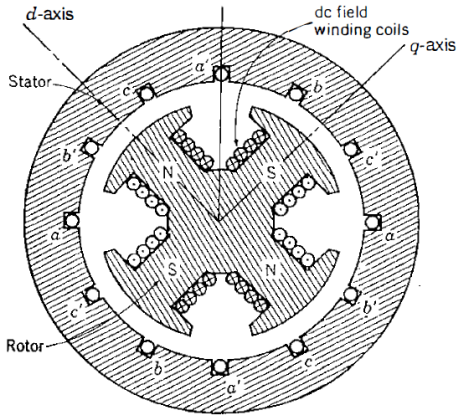


Figure 1 Two-axis generator model

By neglecting d-axis and q-axis open circuit sub-transient time constants T_{do}' and T_{qo}', a two-axis generator model generator differential equations are formed as follows as [3]:

$$\dot{\delta}(i) = \omega(i) - \omega_s \tag{2}$$

where δ denotes the generator angle; ω denotes the generator speed; ω_s is the nominal speed of the generator, i represents the generator number. Next, the speed can be represented as

$$\dot{\omega}(i) = \frac{1}{M} \times \{T_M(i) + \frac{V_i}{x_d'(i)} \times \left[E_d'(i)\cos\big(\theta(i) - \delta(i)\big) + E_q'(i) \times \sin\big(\theta(i) - \delta(i)\big)\right]\} \tag{3}$$

where M denotes the inertia constant of generator; T_M is mechanical input torque; V denotes the bus voltage; x'_d is the direct-axis transient reactance; E'_d and E'_q are the transient voltage of d-axis and q-axis, respectively; and θ is the angle of the bus. Variables E'_d and E'_q are defined as

$$\dot{E}'_q(i) = \frac{1}{T_{d0}(i)} \times \left[(-1) \times \frac{x_d(\mathrm{i})}{x'_d(i)} \times E'_q(\mathrm{i}) + \frac{x_d(\mathrm{i}) - x'_d(i)}{x'_d(i)} \times \mathrm{V(I)} \times \cos\big(\theta(i) - \delta(i)\big) + E_{fd}(\mathrm{i})\right] \tag{4}$$

and

$$\dot{E}'_d(i) = \frac{1}{T_{q0}(i)} \times [(-1) \times \frac{x_q(\mathrm{i})}{x'_d(i)} \times E'_d(\mathrm{i}) - \frac{x_q(\mathrm{i}) - x'_d(i)}{x'_d(i)} \times \mathrm{V(i)} \times \sin(\theta(i) - \delta(i))] \tag{5}$$

where T_{d0} is the open circuit time constant of d-axis; x_d is the d-axis reactance; E_{fd} is the steady state internal voltage of the armature; T_{qo} is the open circuit time constant of q-axis; and x_q is the q-axis reactance. Also,

$$\dot{E_{fd}}(i) = \frac{1}{T_E(i)} \times [(-1) * K_E(i) \times E_{fd}(i) + V_R(i)] \tag{6}$$

where T_E is the electrical torque; V_R denotes the exciter input; and K_E is an exciter gain. The exciter voltage dynamics is shown as

$$\dot{V}_R(i) = \frac{1}{T_A(i)} [K_A(i) \times R_F(i) - V_R(i) - K_A(i) \times \frac{K_F(i)}{T_F(i)} \times E_{fd}(i) + K_A(i) \times (V_{ref}(i) - V(i))] \tag{7}$$

where K_A is the amplifier gain; R_F is the rate feedback; T_A denotes the amplifier time constant; V_{ref} is the reference voltage; K_F is the feedback gain; T_F is the feedback time constant. The following dynamics represent the turbine dynamical behavior as

$$\dot{R}_F(i) = \frac{1}{T_F(i)} \times [\frac{K_F(i)}{T_F(i)} \times E_{fd}(i) - R_F(i)] \tag{8}$$

$$\dot{T}_M(i) = \frac{1}{T_{RH}(i)} \times [(-1) \times T_M(i) + (1 - K_{HP}(i) \times \frac{T_{RH}(i)}{T_{CH}(i)}) \times P_{CH}(i)$$

$$+K_{HP}(i) \times \frac{T_{RH}(i)}{T_{CH}(i)} \times P_{SV}(i)] \quad (9)$$

where P_{SV} is the steam value position; T_{RH} is the output torque of the steam; K_{HP} denotes the high pressure turbine gain; T_{CH} is the steam chest output torque; P_{CH} is the steam chest pressure.

$$\dot{P}_{CH}(i) = \frac{1}{T_{CH}(i)} \times (P_{SV}(i) - P_{CH}(i)) \quad (10)$$

$$\dot{P}_{SV}(i) = \frac{1}{T_{SV}(i)} \times (P_C(i) - P_{SV}(i) - \frac{\omega(i)}{\omega_s \times R(i)}) \quad (11)$$

where T_{SV} denotes the steam value torque; P_C denotes the output of a load reference motor; R is the speed regulation.

For generator classical model, there are five parameters for generator, δ, ω, T_M, P_{CH}, P_{SV}. The last three parameters are turbine parameters, which stay the same with two-axis generator model. Generator angle δ is also the same function as (2). However, due to the lack of the transient status, the classical generator model keep the internal voltage constant, which leads to a different function of ω as

$$\omega = \frac{1}{M}\left[T_M(i) - E_g(i) \times \frac{V(i) \times \sin(\delta(i) - \theta(i))}{x_d'}\right] \quad (12)$$

where E_g denotes the internal voltage.

1.3 Algebraic Equations

There are two sets of algebraic equations, stator equation and network power balance equations.

The stator power balance equations can be derived by the following algebraic equations as

$$P_{Li} + \sum_{j=1}^{ng} B_{ij} V_i [E'_{qj} \sin(\theta_i - \delta_j) + E'_{dj} \cos(\theta_i - \delta_j)] + \sum_{j=ng+1}^{Nb+ng} B_{ij} V_i V_j \sin(\theta_i - \theta_j) = 0 \quad (13)$$

$$-Q_{Li} + \sum_{j=1}^{ng} B_{ij} V_i [E'_{qj} \cos(\theta_i - \delta_j) - E'_{dj} \sin(\theta_i - \delta_j)] + \sum_{j=ng+1}^{Nb+ng} B_{ij} V_i V_j \cos(\theta_i - \theta_j) = 0 \quad (14)$$

where P_{Li} and Q_{Li} denote the active power and reactive power of the load on bus i; and B_{ij} represents the susceptance between bus i and bus j. The non-generator bus power balance algebraic equations are shown as

$$P_{Li} + \sum_{j=ng+1}^{Nb+ng} B_{ij} V_i V_j \sin(\theta_i - \theta_j) = 0 \quad (15)$$

$$-Q_{Li} + \sum_{j=ng+1}^{Nb+ng} B_{ij} V_i V_j \cos(\theta_i - \theta_j) = 0 \quad (16)$$

where Nb denotes the number of buses, ng denotes the number of generators.

The stator variables relationship can be derived from circuit model of Figure 2 in [3],

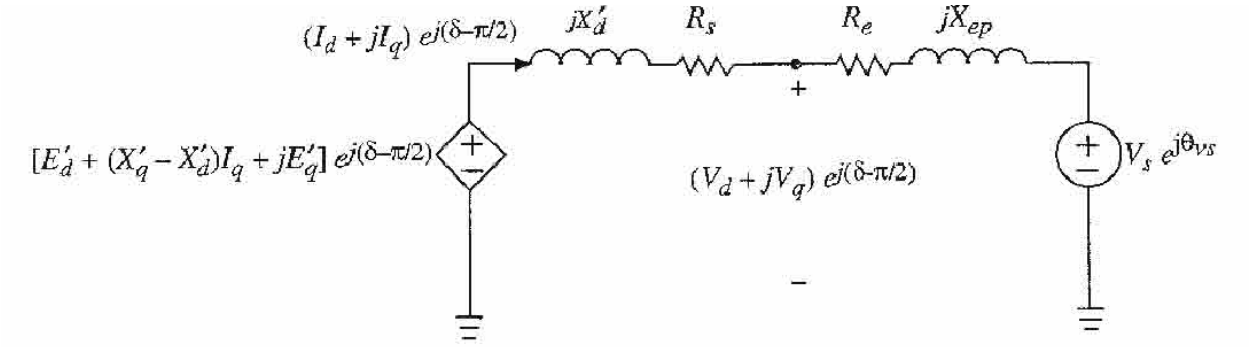


Figure 2 Stator algebraic relationship diagram

The network power balance diagram of Bus i is depicted in Figure 3.

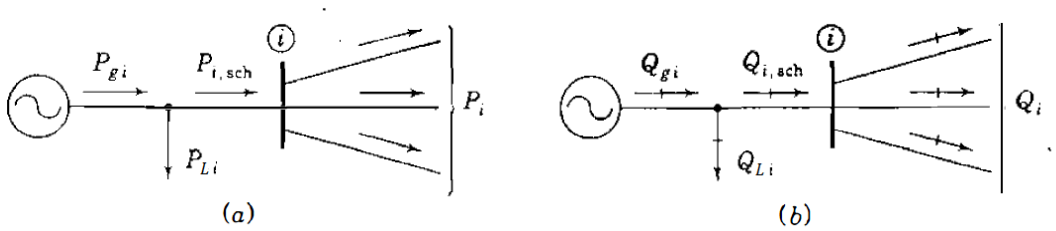


Figure 3 Network power balance diagram of Bus i

In Figure 3, P_{gi} and Q_{gi} denote the real power generation and reactive power generation. $P_{i,sch}$, $Q_{i,sch}$ denote the scheduled power entering the network. P_{Li} and Q_{Li} denote the demand power of the load on Bus i. P_i and Q_i are the calculated value that flow to the network. By calculating the power difference ΔP and ΔQ as (17) and (18), then adjusting them to as close as to zero, power balance equation are obtained.

$$\Delta \mathrm{P}_i = P_{i,sch} - P_i \tag{17}$$

$$\Delta \mathrm{Q}_i = Q_{i,sch} - Q_i \tag{18}$$

The differential and algebraic equations need to be solved together to obtain the power system states. However, duo to nonlinearity involved in the power system model, this is not easy. Previous work used linearization to solve for power system states.

Based on (1) both differential equations and algebraic equations are linearized as:

$$\Delta \dot{x} = \frac{\partial f}{\partial x} \times \Delta x + \frac{\partial f}{\partial y} \times \Delta y + \frac{\partial f}{\partial u} \times \Delta u \tag{19}$$

$$0 = \frac{\partial g}{\partial x} \times \Delta x + \frac{\partial g}{\partial y} \times \Delta y \tag{20}$$

By solving (20),

$$\Delta \mathrm{y} = (\frac{\partial g}{\partial y})^{-1} \times \frac{\partial g}{\partial x} \times \Delta x \tag{21}$$

is obtained.

Substitute (21) to (19), a new differential equation that contains algebraic properties shows as

$$\Delta\dot{x} = \left[\frac{\partial f}{\partial x} - \frac{\partial f}{\partial y} \times \left(\frac{\partial g}{\partial y}\right)^{-1} \times \frac{\partial g}{\partial x}\right] \times \Delta x + \frac{\partial f}{\partial u} \times \Delta u \tag{22}$$

There are some limitations of linearization method. One of the most critical disadvantages is the process of linearization cause the losing of nonlinearity. Some natural effects of nonlinear system are neglected and the system behaves as a linear system. Another limitation is the linearization is based on certain operating point, the system can only be estimated or predicted around that point. It cannot detects system's global behavior [4].

CHAPTER 2. CONVENTIONAL POWER SYSTEM TRANSIENT ANALYSIS ALGORITHMS

This chapter introduces the dynamic power system simulation. The conventional tools for power system dynamical analysis use trapezoidal integration algorithm to solve differential-algebraic equations. More detailed algorithms that consider dynamic characteristics of generator stators and transmission lines include Runge-Kutta algorithm. These algorithms will be explained here.

2.1 Trapezoidal Integration Algorithm

The trapezoidal integration method converts the differential equation into an algebraic equation and solve it using numerical methods. For a general differential equation $\frac{dx}{dt} = f(x,t)$ one has

If $t = t_0, x = x_0,$

If $t = t_1 = t_0 + \Delta t$, then x can be estimated integral form:

$$x_1 = x_0 + \int_{t_0}^{t_1} f(x,\tau)d\tau \tag{23}$$

In trapezoidal integration algorithm, integral area is approximately equal to trapezoidal area, as was shown in Figure 4 below:

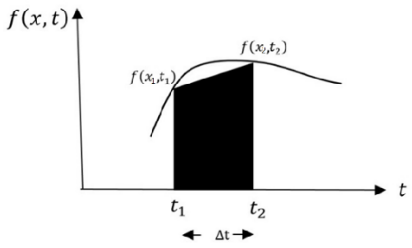


Figure 4 Trapezoidal integration algorithm integral area

For (22),

$$x_1 = x_0 + \frac{\Delta t}{2}[f(x_0, t_0) + f(x_1, t_1)]$$

or

$$x_1 - x_0 - \frac{\Delta t}{2}[f(x_0, t_0) + f(x_1, t_1)] = 0 \quad (24)$$

When t = t_{n+1} , the general relationship for x is:

$$x_{n+1} - x_n + \frac{\Delta t}{2}[f(x_n, t_n) + f(x_{n+1}, t_{n+1})] = 0 \quad (25)$$

2.2 Model Development

In this thesis, two-axis generator model system and classical generator model are utilized to simulate a synchronous generator. Although these two systems have distinct differential expressions, the process of simulating them is the same. The process is based on Newton-Raphson method. By building the Jacobian matrix and reducing the error, the trapezoidal integration algorithm can obtain a precise result. Jacobian matrix is a matrix that is formed all by first order partial derivatives in a particular pattern.

There are three status of whole system: pre-fault condition, during the fault condition, and after the fault condition. The simulation follows the steps below:

1) Input the Bus, line data, generator data, faulted time, fault cleared time and simulation time.
2) Initialize all the parameters, including variables and invariables.
3) Calculate the differences ΔF_1 related of all algebraic equations regarding to all power balance equations.
4) Calculate difference ΔF_2 related to differential equations, which is explained later.

5) Form the Jacobian matrix, using the derivatives of all equation pertaining to steps 4 and 5 and calculate the variable changes.
6) Update the variables by using the calculated variable changes and define the error
7) If the error is smaller than 10^{-5}, then go to step 3. Otherwise, go to step 9.
8) Renew all the variables initial value, time interval and save simulation results.
9) If current simulation time is smaller than the maximum simulation time, go to step 3. Otherwise output the results.

2.2.1 Newton-Raphson Algorithm

Newton-Raphson is a widely adopted iterative algorism for nonlinear equation sets. It is also the widely used computerized algorism for power system load flow, especially for larger system [6]. Assume the nonlinear equation set is:

$$\begin{cases} \Delta F_1(x, \mathrm{y}) = 0 \\ \Delta F_2(x, \mathrm{y}) = 0 \end{cases} \tag{26}$$

Its approximate solutions are $x_1^{(0)}$, $x_2^{(0)}$, ..., $x_n^{(0)}$, $y_1^{(0)}$, $y_2^{(0)}$, ..., $y_m^{(0)}$, n denotes the number of dynamics, m denotes the number of algebraic variables, assume that the difference between approximate solutions and exact solutions are Δx_1, Δx_2, …, Δx_n, Δy_1, Δy_2, …, Δy_m then,

$$\begin{cases} \Delta F_1\left(x_1^{(0)} + \Delta x_1, x_2^{(0)} + \Delta x_2, \dots, x_n^{(0)} + \Delta x_n, y_1^{(0)} + \Delta y_1, y_2^{(0)} + \Delta y_2, \dots, y_m^{(0)} + \Delta y_m\right) = 0 \\ \Delta F_2\left(x_1^{(0)} + \Delta x_1, x_2^{(0)} + \Delta x_2, \dots, x_n^{(0)} + \Delta x_n, y_1^{(0)} + \Delta y_1, y_2^{(0)} + \Delta y_2, \dots, y_m^{(0)} + \Delta y_m\right) = 0 \end{cases} \tag{27}$$

Expanding each equation with Taylor series, take the first equation as example:

$$\Delta F_1\left(x_1^{(0)} + \Delta x_1, x_2^{(0)} + \Delta x_2, \dots, x_{\mathrm{n}}^{(0)} + \Delta x_{\mathrm{n}}, y_1^{(0)} + \Delta y_1, y_2^{(0)} + \Delta y_2, \dots, y_m^{(0)} + \Delta y_m\right)$$

$$= \Delta F_1\left(x_1^{(0)}, x_2^{(0)} \dots, x_n^{(0)}, y_1^{(0)}, y_2^{(0)}, \dots, y_m^{(0)}\right) + \left.\frac{\partial \Delta F_1}{\partial x_1}\right|_0 \Delta x_1 +$$

$$\left.\frac{\partial \Delta F_1}{\partial x_2}\right|_0 \Delta x_2 + \cdots + \left.\frac{\partial \Delta F_1}{\partial x_n}\right|_0 \Delta x_n \left.\frac{\partial \Delta F_1}{\partial y_1}\right|_0 \Delta y_1 + \left.\frac{\partial \Delta F_1}{\partial y_2}\right|_0 \Delta y_2 + \cdots + \left.\frac{\partial \Delta F_1}{\partial y_m}\right|_0 \Delta y_m + \phi_1 = 0$$

or shorten as

$$\Delta F\left(x^{(0)} + \Delta x, y^{(0)} + \Delta y\right) = \Delta F\left(x^{(0)}, y^{(0)}\right) + \sum_{i=1}^{n} \left.\frac{\partial \Delta F}{\partial x_{\mathrm{i}}}\right|_0 \Delta x_{\mathrm{i}} + \sum_{j=1}^{m} \left.\frac{\partial \Delta F}{\partial y_{\mathrm{j}}}\right|_0 \Delta y_j + \phi_1 = 0 \tag{28}$$

In the equation, $\left.\frac{\partial \Delta F_1}{\partial x_1}\right|_0, \left.\frac{\partial \Delta F_1}{\partial x_2}\right|_0, \ldots, \left.\frac{\partial \Delta F_1}{\partial x_n}\right|_0, \left.\frac{\partial \Delta F_1}{\partial y_1}\right|_0, \left.\frac{\partial \Delta F_1}{\partial y_2}\right|_0, \ldots, \left.\frac{\partial \Delta F_1}{\partial y_m}\right|_0$ are estimated by substituting $x_1^{(0)}, x_2^{(0)} \ldots, x_n^{(0)}, y_1^{(0)}, y_2^{(0)} \ldots, y_m^{(0)}$ into (27). ϕ_1 is a function of product of high-order partial derivative of ΔF_1 and high order containsΔx_1,Δx_2, …,Δx_n, Δy_1,Δy_2, …,Δy_m. When the approximate solution is close to exact solution, the high order of Δx and Δy can be ignored, such that ϕ_1 can be ignored as well. Therefore,

$$\Delta F_1\left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}, y_1^{(0)}, y_2^{(0)} \ldots, y_m^{(0)}\right) + \left.\frac{\partial \Delta F_1}{\partial x_1}\right|_0 \Delta x_1 +$$

$$\left.\frac{\partial \Delta F_1}{\partial x_2}\right|_0 \Delta x_2 + \cdots + \left.\frac{\partial \Delta F_1}{\partial x_n}\right|_0 \Delta x_n \left.\frac{\partial \Delta F_1}{\partial y_1}\right|_0 \Delta y_1 + \left.\frac{\partial \Delta F_1}{\partial y_2}\right|_0 \Delta y_2 + \cdots + \left.\frac{\partial \Delta F_1}{\partial y_m}\right|_0 \Delta y_m = 0$$

$$\Delta F_2\left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}, y_1^{(0)}, y_2^{(0)} \ldots, y_n^{(0)}\right) + \left.\frac{\partial \Delta F_1}{\partial x_1}\right|_0 \Delta x_1 +$$

$$\left.\frac{\partial \Delta F_1}{\partial x_2}\right|_0 \Delta x_2 + \cdots + \left.\frac{\partial \Delta F_1}{\partial x_n}\right|_0 \Delta x_n \left.\frac{\partial \Delta F_1}{\partial y_1}\right|_0 \Delta y_1 + \left.\frac{\partial \Delta F_1}{\partial y_2}\right|_0 \Delta y_2 + \cdots + \left.\frac{\partial \Delta F_1}{\partial y_m}\right|_0 \Delta y_m = 0 \tag{29}$$

These are linear system equations. They can be modified to matrix equations:

$$\begin{bmatrix} \Delta F_1\left(x^{(0)}, y^{(0)}\right) \\ \Delta F_2\left(x^{(0)}, y^{(0)}\right) \end{bmatrix} = \begin{bmatrix} \frac{\partial \Delta F_1}{\partial x} & \frac{\partial \Delta F_1}{\partial y} \\ \frac{\partial \Delta F_2}{\partial x} & \frac{\partial \Delta F_2}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta \mathrm{x} \\ \Delta \mathrm{y} \end{bmatrix} \tag{30}$$

Or simplified as:

$$\Delta \mathrm{F} = \mathrm{J}\Delta \mathrm{X} \tag{31}$$

Where J is the Jacobian matrix of functiong_i; ΔX is column vector ofΔx_i and ofΔy_i; ΔF is the difference column vector, which denotes the difference of all differential-algebraic equations. ΔF is the tracking difference that the system tries to reach at zero. ΔF_1 represents the difference calculated by using trapezoidal integration algorithm as

$$\Delta \mathrm{F}_1 = x_{n+1} - x_n - \frac{\Delta t}{2} \times [f(x_n, y_n, u) + f(x_{n+1}, y_{n+1}, u)] = 0 \tag{32}$$

For two-axis generator model, there are 10 parameters that need to be concerned, which are δ, ω, E_q', E_d', E_{fd}, V_R, R_F, T_M, P_{ch}, P_{sv}. All these variables denote asx_i. . $\Delta \mathrm{F}_2$ presents the power difference as (13) and (14), as well as $\Delta \mathrm{F}_1$ shows difference for dynamic variables, repeat to every generator bus. In addition, $\Delta \mathrm{F}_2$ utilizes (15) and (16) for calculating power balance of every node.

Substituting $x_i^{(0)}$and $y_i^{(0)}$ into the function, each elements of $\Delta \mathrm{F}$ and J can be estimated. $x_i^{(0)}$ and $y_i^{(0)}$can be calculated by any means to solve linear matrix equations. After the first iteration, the new value of x_i is$x_i^{(1)} = x_i^{(0)} + \Delta x_i^{(0)}$, as well as y_iSubstitute $x_i^{(1)}$ and $y_i^{(1)}$ into the equation, the new value of each element of $\Delta \mathrm{F}$ and J can be estimated, so that calculate $x_i^{(1)}$ and $y_i^{(1)}$ as

$$x_i^{(2)} = x_i^{(1)} + \Delta x_i^{(1)} \tag{33}$$

$$y_i^{(2)} = y_i^{(1)} + \Delta y_i^{(1)} \tag{34}$$

The final solution of (34) is generated through iterations by iterations.

The calculation steps of Newton-Raphson are shown below:

1) Set the initial value for each bus voltage $V^{(0)}$、$\theta^{(0)}$ and each generator dynamics denoted as$y_i^{(0)}$.
2) Estimate power difference vector$\Delta P^{(0)}$, $\Delta Q^{(0)}$ and using trapezoidal integration algorithm to estimate$\Delta F_2^{(0)}$.

3) Estimate each element of Jacobian matrix.

4) Solve the equation and ΔX

5) Modify each variable:

$$\begin{cases} x^{(1)} = x^{(0)} + \Delta x^{(0)} \\ y^{(1)} = y^{(0)} + \Delta y^{(0)} \end{cases} \tag{35}$$

6) Estimate $\Delta F^{(1)}$ with $x^{(1)}$ and$y^{(1)}$.

7) Examine convergence. The condition of convergence is:

$$\left|\Delta F^{(K)}\right| < \varepsilon \tag{36}$$

where $\left|\Delta F^{(K)}\right|$ is the absolute value of maximum element of vector $\Delta F^{(K)}$. It can show the power difference of final results. In this case$\varepsilon = 10^{-5}$.

8) If$\left|\Delta F^{(K)}\right| < \varepsilon$, output simulation results, otherwise repeat from step 3 and iterate until satisfy condition of convergence.

The error detection loop shows as Figure 5.

2.3 Runge-Kutta Method and Procedure

Power system is a nonlinear system, which includes differential equations and algebraic equations as discussed above. However, it is always challenging to solve them both at the same time. In this method, a purely differential equations are formed and simulated by using Runge-Kutta method. Runge-Kutta algorithm is widely used in simulation program and based on the known derivatives of equations and initial value of all variables. Here are the basic rules of this method.

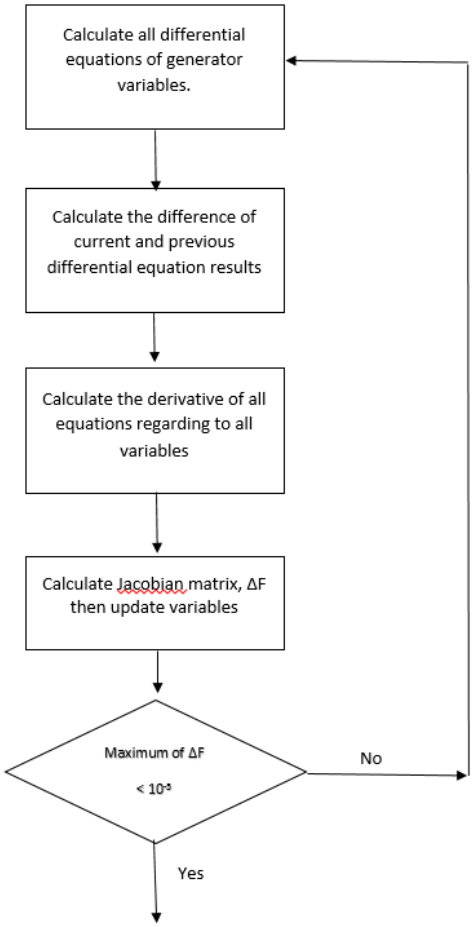


Figure 5 Error detection loop

Runge-Kutta method is very similar to Taylor series solution. Different order of Runge-Kutta method is depended on different items reserved from Taylor series expansion. The more effective items are reserved, the higher order Runge-Kutta method can get, and the more accurate the results are.

In this paper, forth order Runge-Kutta method is used. For step (n+1), the general function for x is:

$$x_{n+1} = x_n + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] \tag{37}$$

where,

$$k_1 = f(x_n, t_n)\Delta t \tag{38}$$

$$k_2 = f\left(x_n + \frac{k_1}{2}, t_n + \frac{\Delta t}{2}\right)\Delta t \tag{39}$$

$$k_3 = f(x_n + \frac{k_2}{2}, t_n + \frac{\Delta t}{2})\Delta t \tag{40}$$

$$k_4 = f(x_n + k_3, t_n + \Delta t)\Delta t \tag{41}$$

The physical explanations for the previous formulas are shown below:

k_1= (slope of beginning time step) Δt

k_2= (first approximation of slop of middle time step) Δt

k_3= (second approximation of slop of middle time step) Δt

k_4= (slope of last time step) Δt

$$\Delta x = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{42}$$

Consequently, Δx is the increment value of x. It depends on the weighted average of slope of beginning, middle and last point of time step.

In this algorithm test, dynamic stability is simulated and analyzed. Dynamic stability concerns the influence of a small fault that lasts a long time with control devices [3]. In this case, the system is tested in relatively long time in fault situation and without any control applications. For example, [5] provides a renewable energy source to control the whole system.

The procedure of this algorithm is as Figure 6:

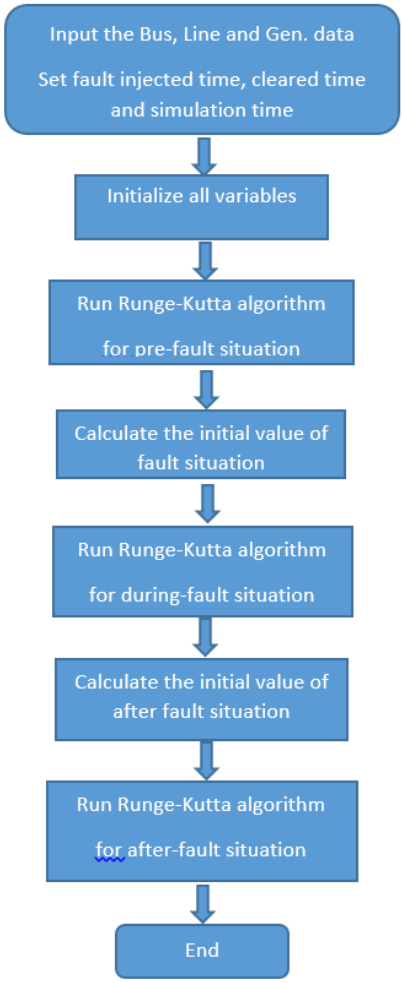


Figure 6 Runge-Kutta algorithm procedure

CHAPTER 3. POWER SYSTEM DYNAMICAL MODEL DEVELOPMENT

This chapter explains the development of power system dynamic model. It is divided into three parts. In the first section, all equations are developed. Then we focuses on the system modeling and simulation background introduction. In this section, all basic settings are provided and explained. Finally, simulation results and analysis is shown, which is the comparison of two method. In this chapter the benefits of Runge-Kutta method is shown, that is, without losing the accuracy, but saving lots of time.

3.1 Center of Inertia Coordinate Frame

Center of Inertia coordinate is a replacement of synchronization rotational coordinate axes, which is a time varying coordinate axes. It contributes to show the dynamic behavior of each generator. Essentially, it can isolate the vibration or energy of transient status from stability analysis.

Power system center of inertia follows the equations below:

$$M_i = \frac{2 \times H_i}{\omega_s} \tag{46}$$

$$\delta_{coi} = \frac{1}{M_t} \sum_{i=1}^{n} M_i \times \delta_i \tag{47}$$

$$\omega_{coi} = \frac{1}{M_t} \sum_{i=1}^{n} M_i \times (\omega_i - \omega_s) \tag{48}$$

where $M_t = \sum_{i=1}^{n} M_i$, M_i is the rotor inertia time constant. H_i is the inertia constant.

Under the center of inertia coordinate axes frame, generator angle, generator speed and bus angle can represent as:

$$\delta = \delta_i - \delta_{coi} \tag{49}$$

$$\omega = \omega_i - \omega_{coi} \tag{50}$$

$$\varphi = \theta_i - \delta_{coi} \tag{51}$$

where $\delta_i, \omega_i, \theta_i$ are the absolute value of generator angle, generator speed and bus angle. δ, ω, φ are the generator angle, generator speed and bus angle that regards to the center of inertia, which are used thought out all the Runge-Kutta algorithm.

3.2 Algebraic Equations Differentiation

One of the major challenges of power system is solving and simulating the differential-algebraic equations. In this method, all algebraic equations need to be converted to differential

equations in order to have a pure differential equations. Take two-axis generator model as an example:

$$\frac{\partial \Delta P}{\partial x} = \frac{\partial \Delta P}{\partial V} \times \dot{V} + \frac{\partial \Delta P}{\partial \varphi} \times \dot{\varphi} + \frac{\partial \Delta P}{\partial \delta} \times \dot{\delta} + \frac{\partial \Delta P}{\partial E'_q} \times \dot{E}'_q + \frac{\partial \Delta P}{\partial E'_d} \times \dot{E}'_d = 0 \tag{52}$$

$$\frac{\partial \Delta Q}{\partial x} = \frac{\partial \Delta Q}{\partial V} \times \dot{V} + \frac{\partial \Delta Q}{\partial \varphi} \times \dot{\varphi} + \frac{\partial \Delta Q}{\partial \delta} \times \dot{\delta} + \frac{\partial \Delta Q}{\partial E'_q} \times \dot{E}'_q + \frac{\partial \Delta Q}{\partial E'_d} \times \dot{E}'_d = 0 \tag{53}$$

By solving (52) an (53) for $\dot{V}$ and$\dot{\varphi}$, the algebraic equations can be converted to a set of dynamic equations as follow：

$$\begin{bmatrix} \dot{V} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} A & C \\ B & D \end{bmatrix}^{-1} \times \begin{bmatrix} E & G & K \\ F & H & L \end{bmatrix} \times \begin{bmatrix} \dot{\delta} \\ \dot{E}'_q \\ \dot{E}'_d \end{bmatrix} \tag{54}$$

where

A= $\frac{\partial \Delta P}{\partial V}$, C=$\frac{\partial \Delta P}{\partial \varphi}$, E= $\frac{\partial \Delta P}{\partial \delta}$, G= $\frac{\partial \Delta P}{\partial E'_q}$, K=$\frac{\partial \Delta P}{\partial E'_d}$, B= $\frac{\partial \Delta Q}{\partial V}$, D= $\frac{\partial \Delta Q}{\partial \varphi}$, F= $\frac{\partial \Delta Q}{\partial \delta}$, H= $\frac{\partial \Delta Q}{\partial E'_{qj}}$, L= $\frac{\partial \Delta Q}{\partial E'_{dj}}$.

Specifically,

$$a_{ii} = \sum_{j=1}^{n} B_{i+n,j}\left[E'_{qj} \sin(\varphi_{i+n} - \delta_j) + E'_{dj} \cos(\varphi_{i+n} - \delta_j)\right] + \sum_{j=n+1}^{N+n} B_{i+n,j} V_j \sin(\varphi_{i+n} - \varphi_j) + dP \qquad (i = 1,2,\dots,N.\, i \neq j) \tag{55}$$

$$\text{and } a_{ij} = V_{i+n} B_{i+n,j} \sin(\varphi_{i+n} - \varphi_j) \qquad (j = n+1, n+2, \dots, N+n) \tag{56}$$

denote the elements of A. Next, C is represented as

$$c_{ii} = V_{i+n} \sum_{j=1}^{n} B_{i+n,j}\left[E'_{qj} \cos(\varphi_{i+n} - \delta_j) - E'_{dj} \sin(\varphi_{i+n} - \delta_j)\right] + V_{i+n} \sum_{j=n+1}^{N+n} B_{i+n,j} V_j \cos(\varphi_{i+n} - \varphi_j) - V_{i+n}^2 B_{i+n,i+n} \qquad (i = 1,2,\dots,N.\, i \neq j) \tag{57}$$

and $c_{ij} = -V_{i+n}B_{i+n,j}\,V_j\cos(\varphi_{i+n} - \varphi_j) \qquad (i = 1,2,\dots,N.\ j = n+1, n+2, \dots, N+n)$

(58)

Then,

$e_{ij} = V_{i+n}B_{i+n,j}[E'_{dj}\sin(\varphi_{i+n} - \delta_j) - E'_{qj}\cos(\varphi_{i+n} - \delta_j)]$

$(i = 1,2,\dots,N.j = n+1, n+2, \dots, N+n)$ (59)

denote the elements of E. Matrix G has elements as

$$g_{ij} = V_{i+n}B_{i+n,j}\sin(\varphi_{i+n} - \delta_j) \qquad (i = 1,2,\dots,N.\ j = 1,2,\dots,n) \tag{60}$$

Also,

$$k_{ij} = V_{i+n}B_{i+n,j}\cos(\varphi_{i+n} - \delta_j) \qquad (i = 1,2,\dots,N.\ j = 1,2,\dots,n) \tag{61}$$

denote the elements of K. Another Matrix B is introduced as

$b_{ii} = \sum_{j=1}^{n} B_{i+n,j}\left[E'_{qj}\cos(\varphi_{i+n} - \delta_j) + E'_{dj}\sin(\varphi_{i+n} - \delta_j)\right] + \sum_{j=n+1}^{N+n} B_{i+n,j}\,V_j\cos(\varphi_{i+n} -$

$\varphi_j) - dQ \qquad (i = 1,2,\dots,N.\ i \neq j)$ (62)

and $b_{ij} = V_{i+n}B_{i+n,j}\cos(\varphi_{i+n} - \varphi_j) \qquad (j = n+1, n+2, \dots, N+n)$ (63)

Where after,

$$d_{ii} = -V_{i+n}\sum_{j=1}^{n} B_{i+n,j}\left[E'_{qj}\sin(\varphi_{i+n} - \delta_j) - E'_{dj}\cos(\varphi_{i+n} - \delta_j)\right] +$$

$$V_{i+n}\sum_{j=n+1}^{N+n} B_{i+n,j}\,V_j\sin(\varphi_{i+n} - \varphi_j) \qquad (i = 1,2,\dots,N.\ i \neq j) \tag{64}$$

and $d_{ij} = V_{i+n}B_{i+n,j}\,V_j\sin(\varphi_{i+n} - \varphi_j) \qquad (i = 1,2,\dots,N.\ j = n+1, n+2, \dots, N+n)$

(65)

denote the elements of D. Along with Matrix D,

$f_{ij} = V_{i+n}B_{i+n,j}[E'_{qj}\sin(\varphi_{i+n} - \delta_j) - E'_{dj}\cos(\varphi_{i+n} - \delta_j)]$

$(i = 1,2,\dots,N.j = n+1, n+2, \dots, N+n)$ (66)

denote the elements of F. Besides,

$$h_{ij} = V_{i+n}B_{i+n,j}\cos\left(\varphi_{i+n} - \delta_j\right) \qquad (i = 1,2,\dots,N.\ j = 1,2,\dots,n) \tag{67}$$

denote the elements of H. The last matrix L is explicated as

$$l_{ij} = -V_{i+n}B_{i+n,j}\sin\left(\varphi_{i+n} - \delta_j\right) \qquad (i = 1,2,\dots,N.\ j = 1,2,\dots,n) \tag{68}$$

3.3 Variables Change in Fault and After Fault Condition

In classical generator model, the generator's internal voltages stay unchanged. However, with introducing the transient parameters to the generators, the internal voltage is no longer constant. In spite of this, E'_d and E'_q still stay constant at the moment of fault. From Figure 2, the internal voltage has one element I_q that relates to the changed variables bus voltage V and bus angle θ, which are immediately changed when a fault is injected or cleared as

$$I_q = \frac{V\times\sin(\delta-\theta)}{x_q} \tag{69}$$

In Runge-Kutta algorithm, the previous status of I_q is used for the faulted or after-fault condition. Similar to trapezoidal integration algorithm, the simulation is separated into three parts, pre-fault, fault, and after-fault periods. The initial values need to be calculated for every status, especially for fault condition and after-fault condition. For example, when there is a fault, the line impedance of the system changes. Based on this new impedance and pre-fault internal voltage and generator angle, bus voltages and bus angles for faulted condition are initialized. The same procedure is done for after-fault condition. The small-disturbance rotor angle stability analysis cannot be utilized in long-term simulation [7] because the error is accumulated in fault condition and for generator angles due to linearization error. They could keep increasing or decreasing until the system collapses if the fault stays so long.

CHAPTER 4 SIMULATION RESULTS AND ANALYSIS

4.1 System Modelling and Simulation Background

All two-axis generator model systems run for 10 seconds (classical generator model systems run for 5 seconds) in order to get the most efficiency and clear simulation results. First few seconds is no fault condition, which means there is no fault in the system. From then on, a ground fault happens on Bus 8, the value is 1j or 5j. The fault is cleared after a short assumed time (normally, the fault is about 0.1 second or 0.2 second, for presenting result reason, in these simulations, fault is cleared after a longer time) and the system is back to no fault condition for the

rest simulation time. There are four major parameters that are compared, including bus voltage, bus angle, generator angle and generator speed. In addition, the efficiency of these two method is compared.

4.1.1 Load-Impedance Conversion

As discussed above, there is no resistance in the transmission lines. In Runge-Kutta method, the impedance of the whole system keep constant. However, in trapezoidal integration algorithm, the system is keeping the active power and reactive power constant. For comparison reasons, the trapezoidal method uses modified bus data, which converts all the load to impedance in pre-fault condition and keeps it constant.

Table 1 is the load chart of IEEE 14 bus system:

Table 1 Load Chart of 14 Bus System

Bus No.	Active Generation P_g (p.u.)	Reactive Generation Q_g (p.u.)	Active Load P_l (p.u.)	Reactive Load Q_l (p.u.)	Conductance G (p.u.)	Susceptance B (p.u.)
1	0	0	0	0	0	0
2	0.183	0.297	0	0	0	0
3	-0.942	0.044	0	0	0	0
4	-0.112	0.047	0	0	0	0
5	0	0.174	0	0	0	0
6	0	0	0.478	-0.039	0	0
7	0	0	0	0	0	0
8	0	0	0.076	0.016	0	0
9	0	0	0.595	0.166	0	0.19

10	0	0	0.39	0.058	0	0
11	0	0	0.035	0.018	0	0
12	0	0	0.061	0.016	0	0
13	0	0	0.135	0.058	0	0
14	0	0	0.349	0.05	0	0

The procedure to convert all the load to the conductance G and susceptance B is to use Newton-Raphson load flow method to calculate the bus voltages and angles. Then as the function:

$$G = \frac{P_l}{V^2} \tag{70}$$

$$B = \frac{Q_l}{V^2} \tag{71}$$

After the conversion process, the trapezoidal integration algorithm uses the new modified bus data as Table 2,

Table 2 Modified Bus Data for IEEE 14 Bus System

Bus No.	Active Generation P_g (p.u.)	Reactive Generation Q_g (p.u.)	Active Load P_l (p.u.)	Reactive Load Q_l (p.u.)	Conductance G (p.u.)	Susceptance B (p.u.)
1	0	0	0	0	0	0
2	0.183	0.297	0	0	0	0
3	-0.942	0.044	0	0	0	0
4	-0.112	0.047	0	0	0	0
5	0	0.174	0	0	0	0
6	0	0	0	0	0.4464	0.0364
7	0	0	0	0	0	0
8	0	0	0	0	0.0706	-0.0149
9	0	0	0	0	0.5425	0.0386
10	0	0	0	0	0.3562	-0.0530

11	0	0	0	0	0.0314	-0.0162
12	0	0	0	0	0.0541	-0.0142
13	0	0	0	0	0.1205	-0.0518
14	0	0	0	0	0.3205	-0.0459

In this way, these two algorithm are simulated under the same circumstance, which is constant impedance condition.

4.1.2 Generator Replacement

In order to reduce the complexity of the system and to simplify the computational process, all the generators are moved to the top buses. The main procedure is as Figure 7:

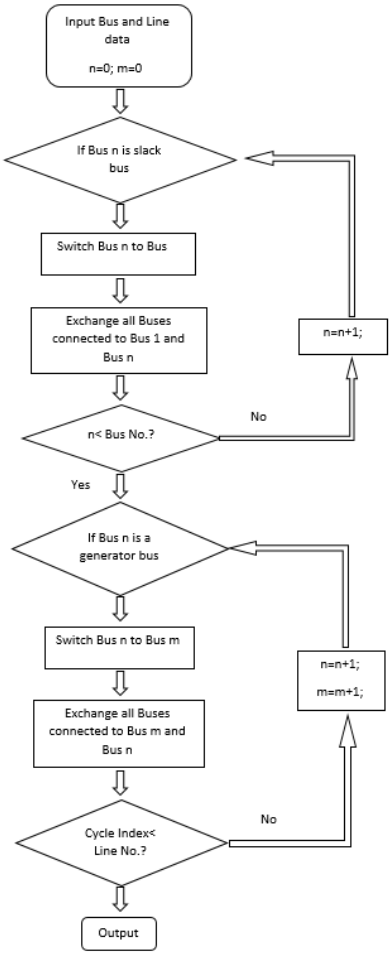


Figure 7 Generator replacement procedure flow

4.2 Simulation Results and Analysis

In this section, two strategies are compared with simulation results and analysis the difference of those two in data prospective in fault condition. When a small disturbance inject to the system, the generator angle and speed will derive from the steady state value and cause a vibration [5]. This can be divided to two sections. In the first section, IEEE 14 bus system is simulated and compared. The generator in the system is classical generator model, which is the simplest model. Second section focuses on two-axis generator model. Three system is simulated and analyzed, which including IEEE 14 bus system and IEEE 118 bus system.

4.2.1 Classical Generator Model

From the beginning to 2 seconds, all parameters keep constant for both cases because it is steady state when there is no fault. From 2 to 3 second, which is the fault situation, the system is in transient stability situation. The system adjusts all the parameters and causes the vibration. After 3 second is fault clear situation. Because of the change of generator parameters, such as generator angle and speed, the system is keeping vibrating in the rest of the simulation. The diagram of 14 Bus system shows as Figure 8. Figure 9 shows the δ of all generators.

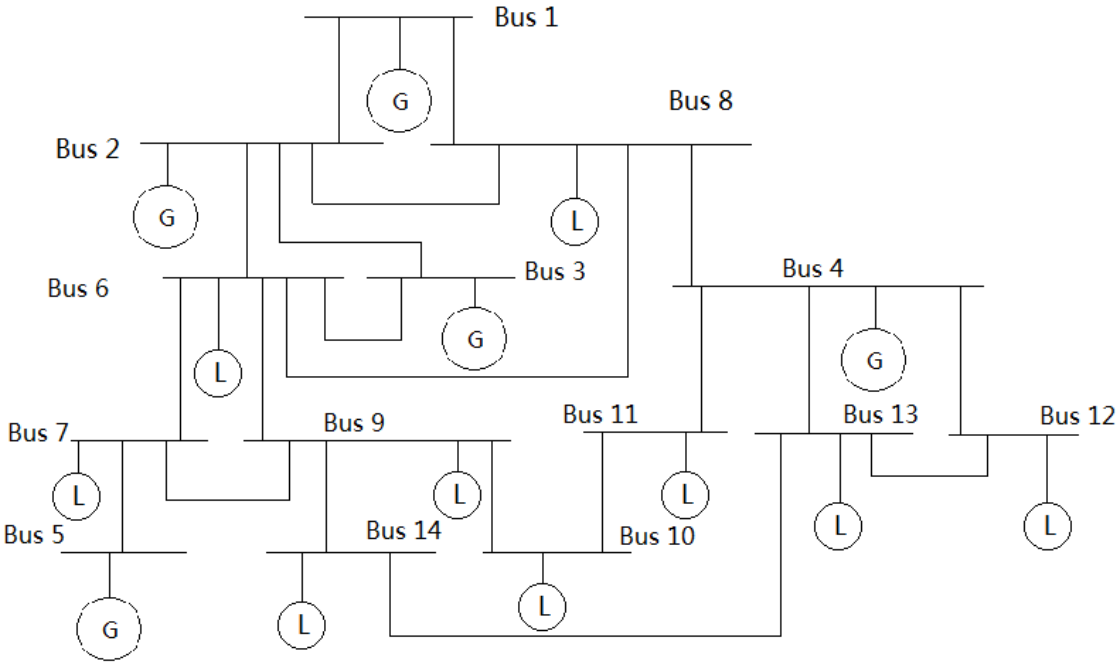
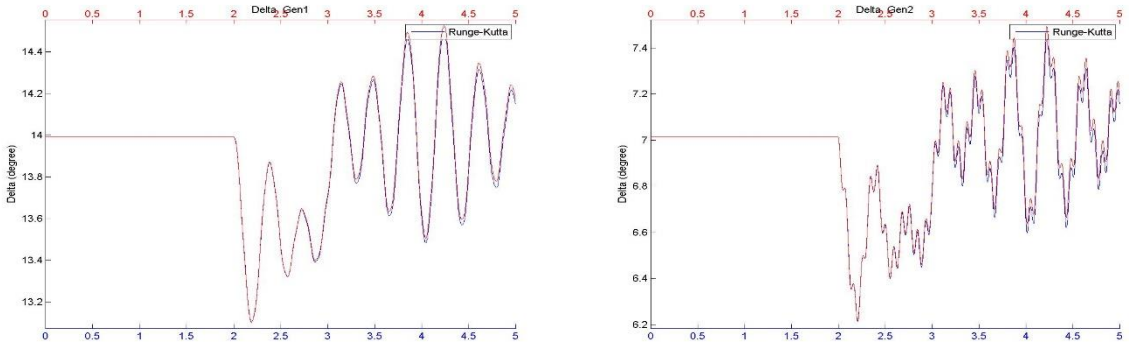
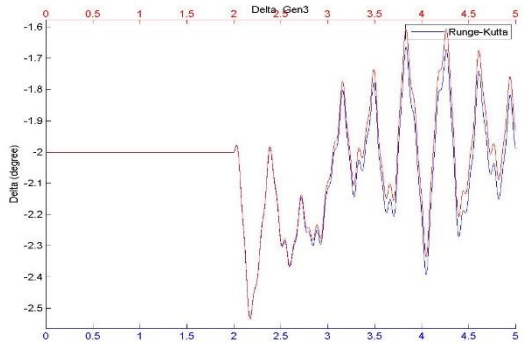


Figure 8 14 Bus diagram

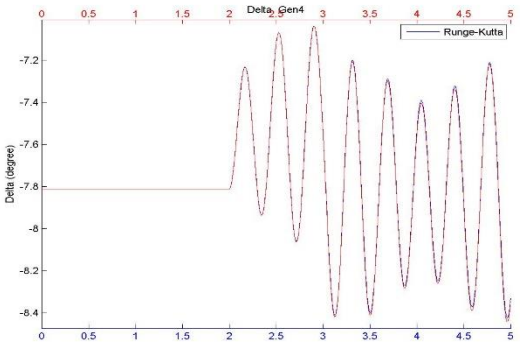


(a)

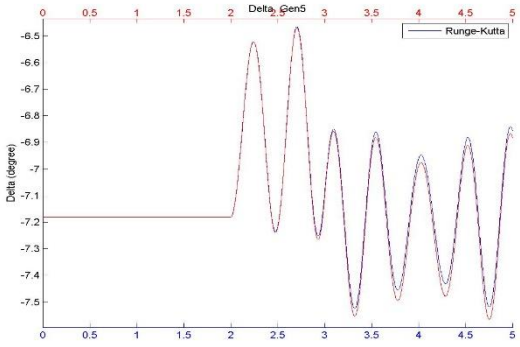
(b)



(c)



(d)



(e)

Figure 9 Comparison of δ for two methods of classical generator model power system. (a) Gen 1_ δ (b) Gen 2_ δ (c) Gen 3_ δ (d) Gen 4_ δ (e) Gen 5_ δ

As figure shows, they all perfectly match with trapezoidal method. What needs to be mentioned here is the δ shown above is the δ respect to the center of inertia δ. In practice, δ will keep increasing to infinity. As to δ respect to COI δ, it is easier and clearer to compare. In this figure, the generator angle becomes steadier after the fault cleared, but never back to normal condition because of the change of initial point. Figure 10 shows the five generator speed.

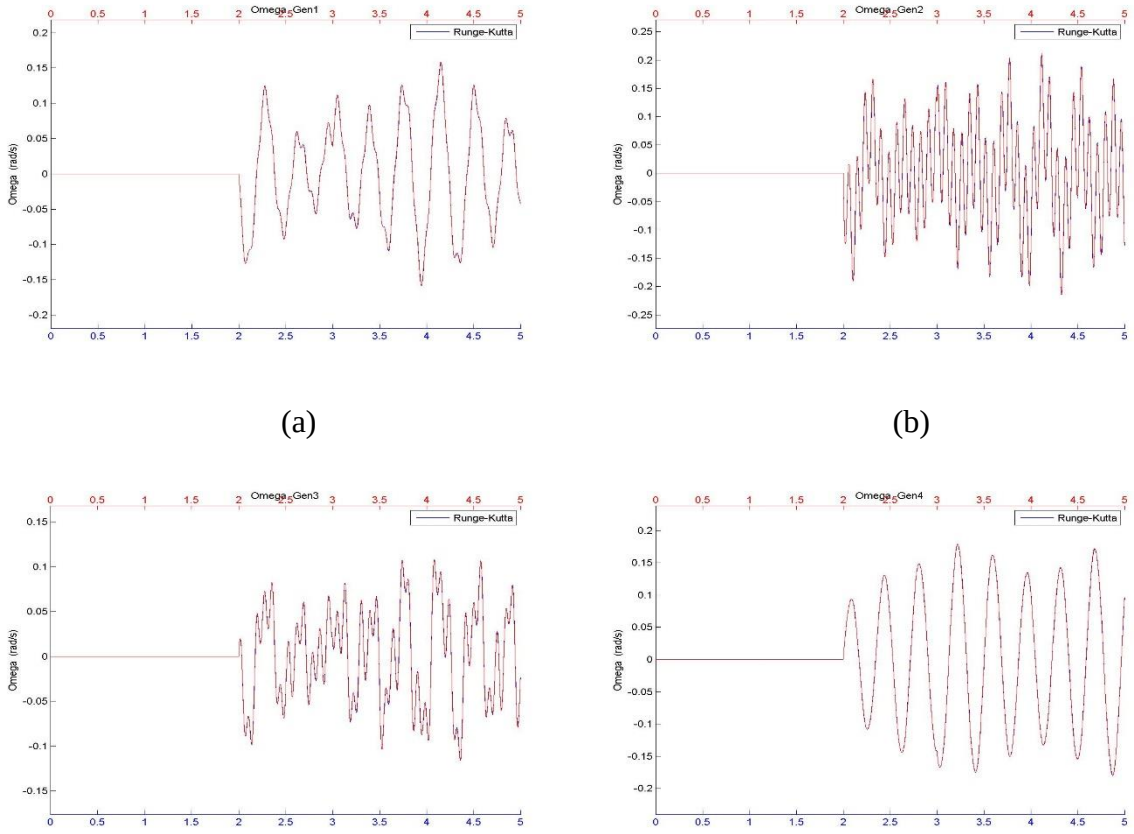
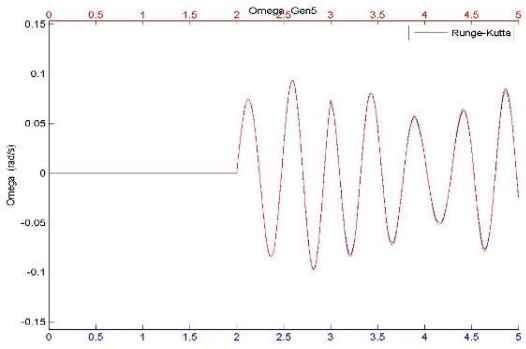


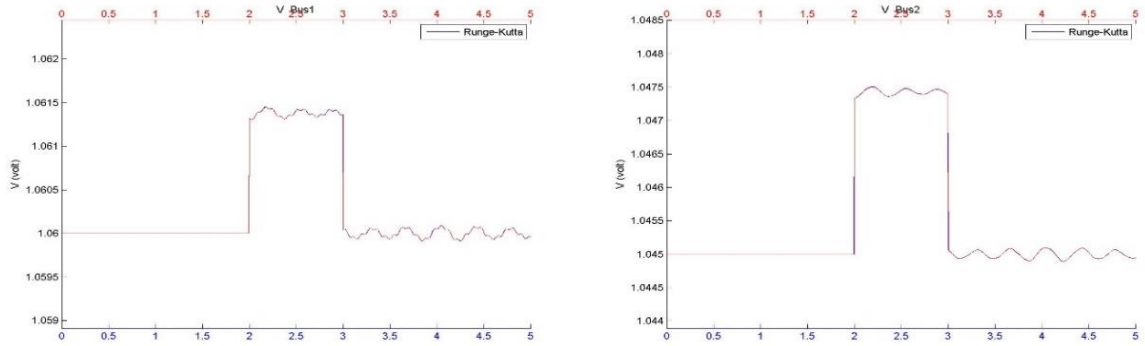
Figure 10 ω of all generators simulated for classical generator model system. (a) Gen 1_ω (b) Gen 2_ω (c) Gen 3_ω (d) Gen 4_ω (e) Gen 5_ω



(e)

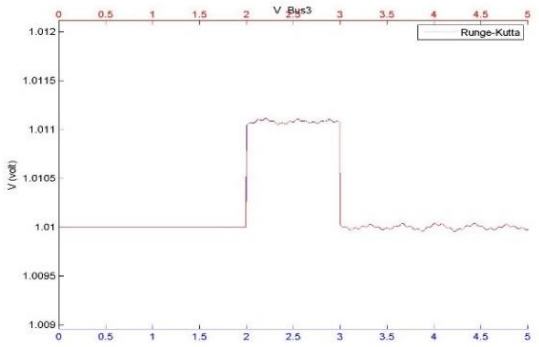
Figure continued

As shown above, they are the ω of all generators. It is the same as the δ, all ω are respect to COI ω for a better comparison. These two methods again show a great match in ω prospective. And combine the δ, these two methods have perfectly matches on generator side in whole simulation period. Figure 11 and Figure 12 reveal the comparison of two algebraic parameters.

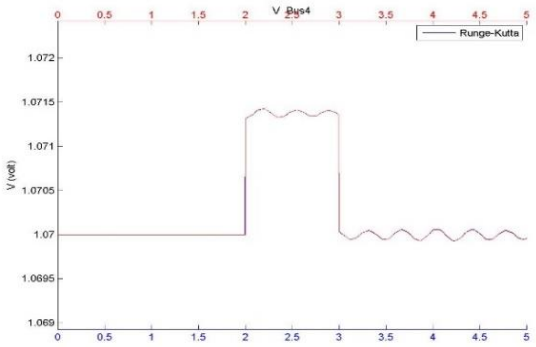


(a) (b)

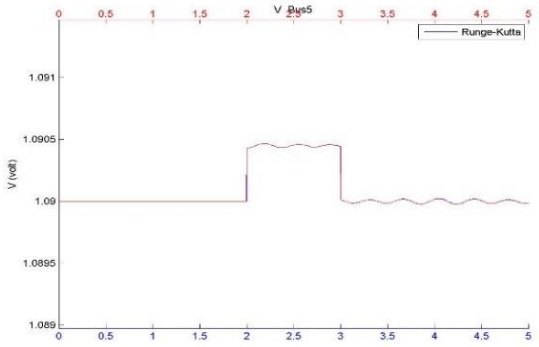
Figure 11 All 14 bus voltages of classical generator model system. (a) Bus 1_V (b) Bus 2_V (c) Bus 3_V (d) Bus 4_V (e) Bus 5_V (f) Bus 6_V (g) Bus 7_V (h) Bus 8_V (i) Bus 9_V (j) Bus 10_V (k) Bus 11_V (l) Bus 12_V (m) Bus 13_V (n) Bus 14_V



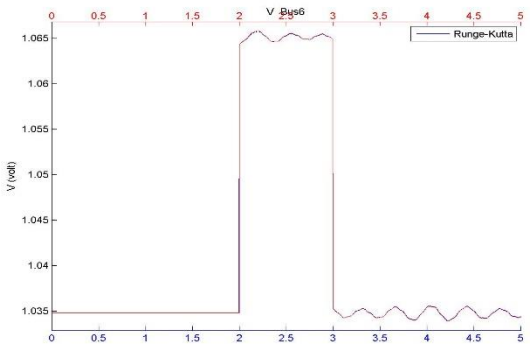
(c)



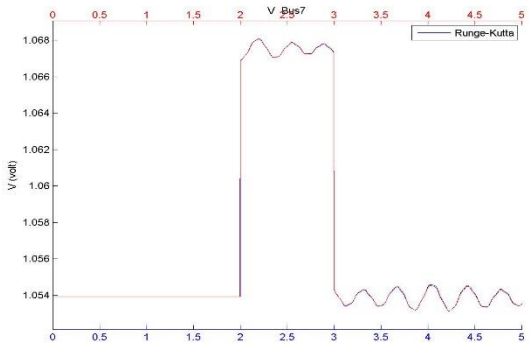
(d)



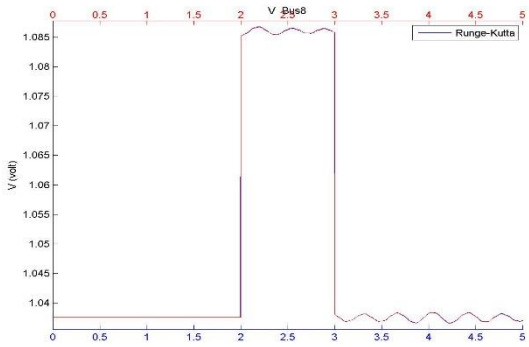
(e)



(f)

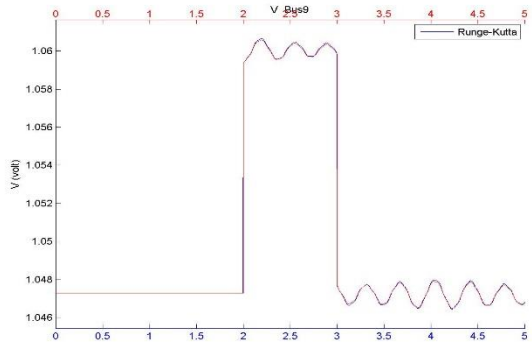


(g)

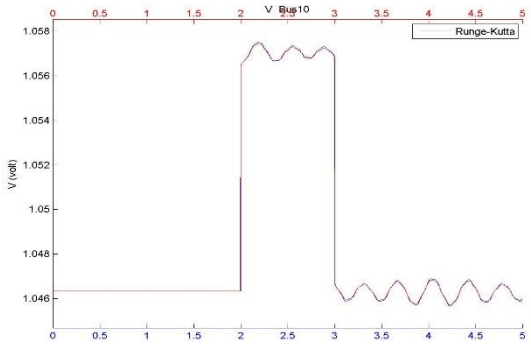


(h)

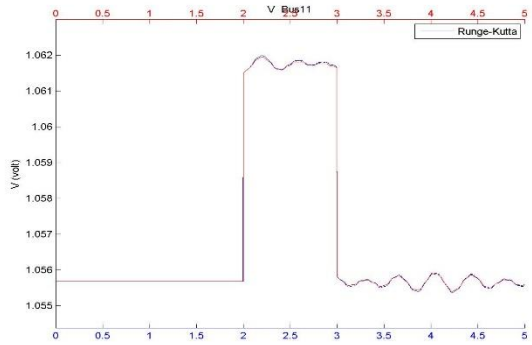
Figure continued



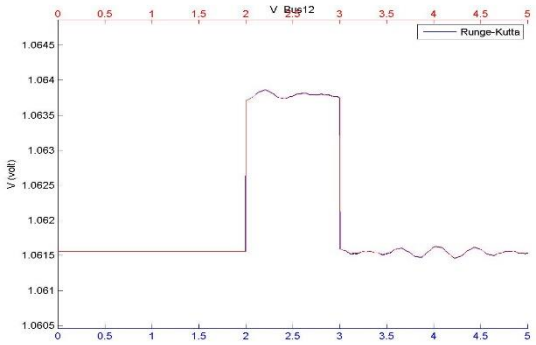
(i)



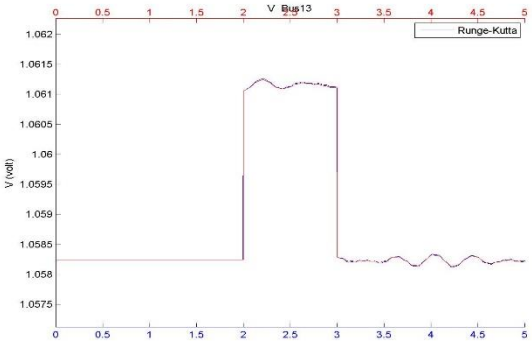
(j)



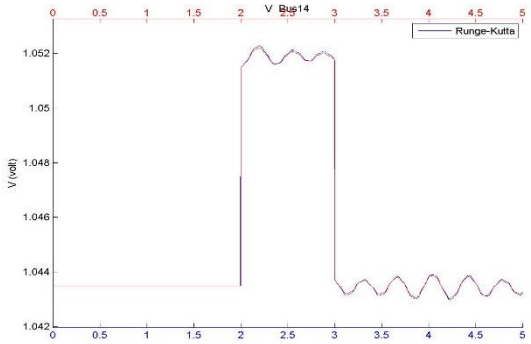
(k)



(l)



(m)

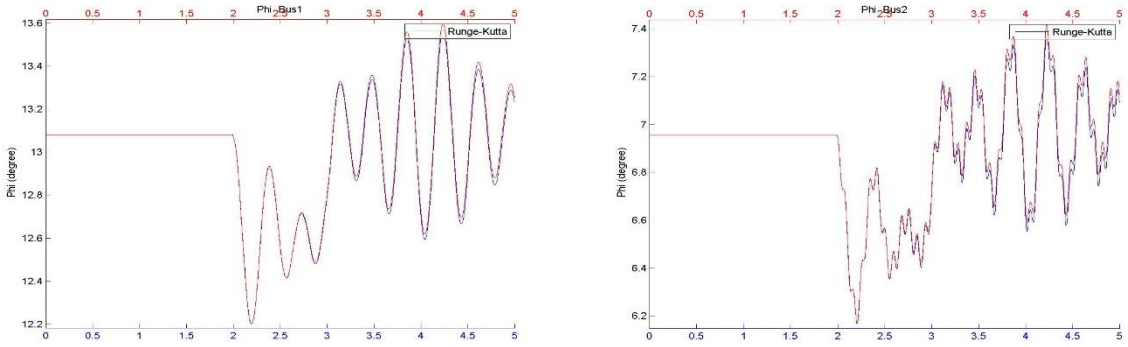


(n)

Figure continued

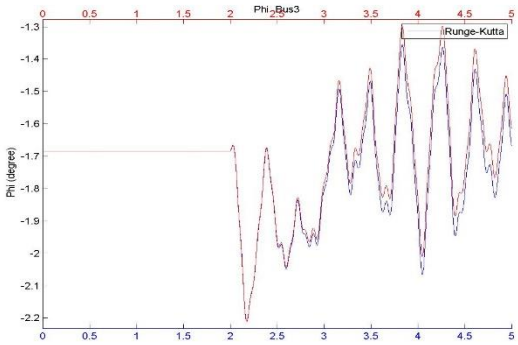
From this figure, a clearly significant voltage jump happens at 2 second because of the fault, for both methods, to the same after fault value. From bus 1 to bus 5 are all generator buses, which have relatively smaller change when suddenly a fault or the fault suddenly is cleared. That

is because the generator can balance those differences. However, for other buses that have only load on them, the magnitude of the bus voltage has around 0.01 -0.03 p.u. change. That is around 1% voltage jump or drop. Normally, the maximum voltage change the industry can endure is 5%. So even this small fault can make great difference. In Figure 8, bus8 connects to bus 2, bus 4, and bus 6. Because of the generator bus has the capability to absorb some energy and mostly reduce the voltage change. However, load bus, bus 6 has nothing to do with the fault. That is why bus 8 and bus 6 are most effected by the fault and increase the voltage 0.03 p.u., which is pretty huge difference on industry side and may cause great damage.

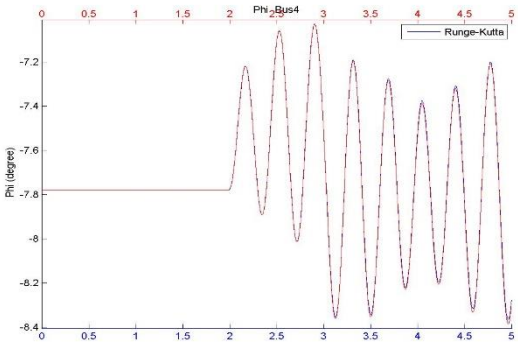


(a) (b)

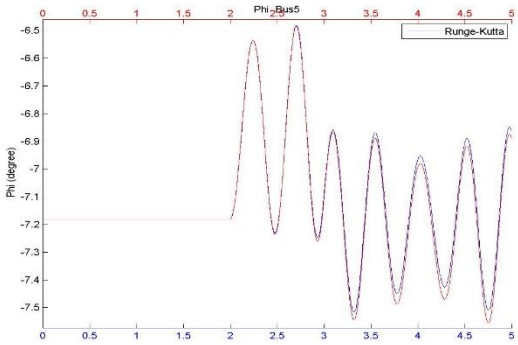
Figure 12 All 14 bus angles of classical generator model system. (a) Bus 1_ φ (b) Bus 2_ φ (c) Bus 3_ φ (d) Bus 4_ φ (e) Bus 5_ φ (f) Bus 6_ φ (g) Bus 7_ φ (h) Bus 8_ φ (i) Bus 9_ φ (j) Bus 10_ φ (k) Bus 11_ φ (l) Bus 12_ φ (m) Bus 13_ φ (n) Bus 14_ φ



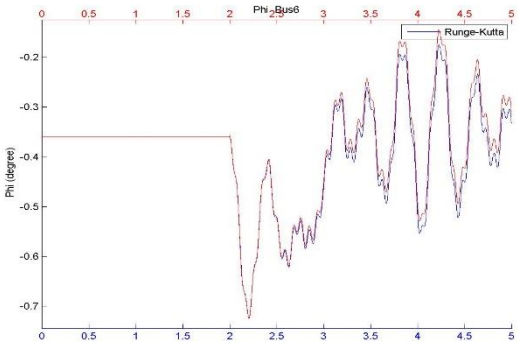
(c)



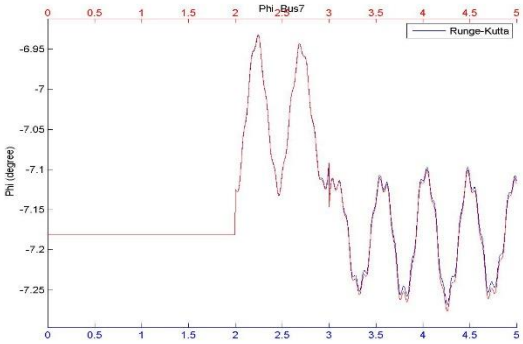
(d)



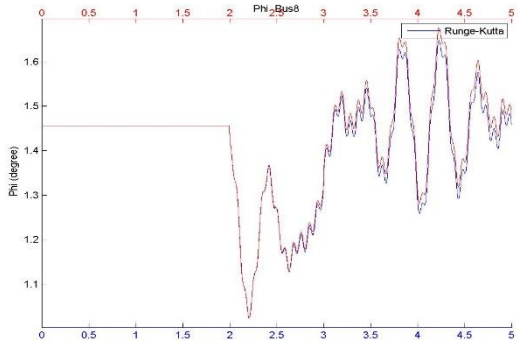
(e)



(f)

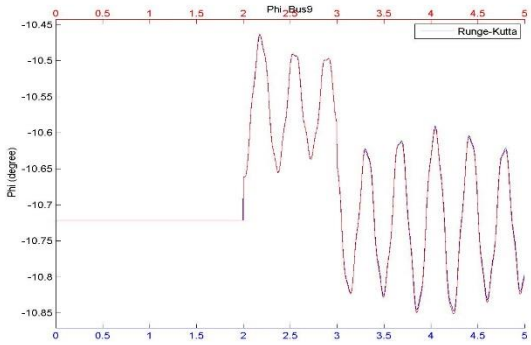


(g)

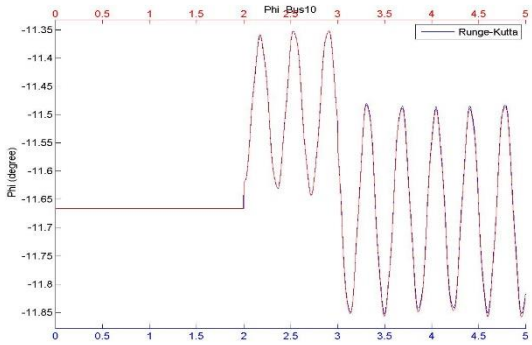


(h)

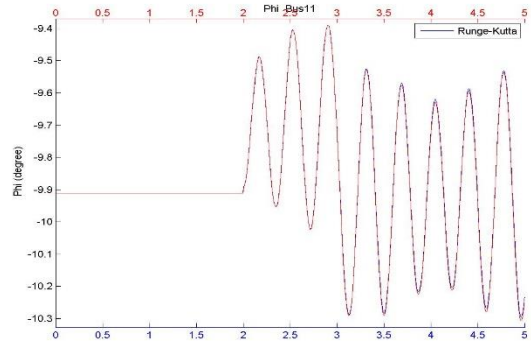
Figure continued



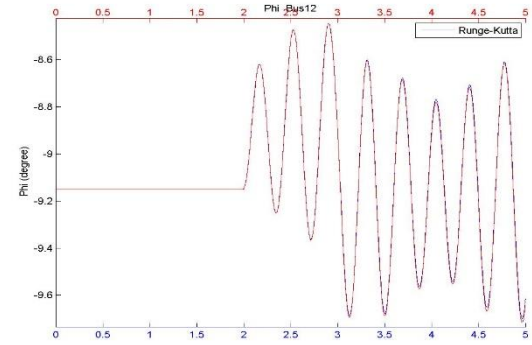
(i)



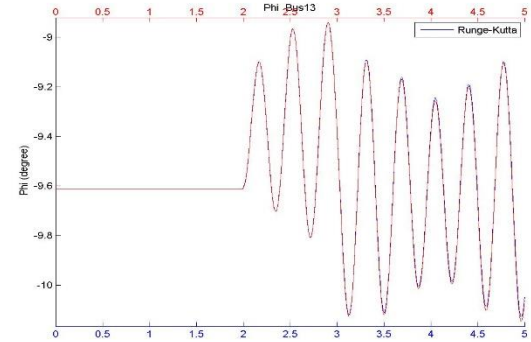
(j)



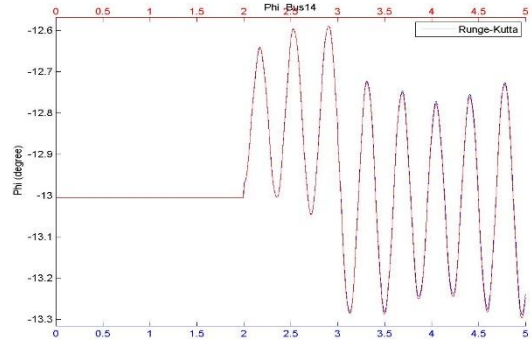
(k)



(l)



(m)



(n)

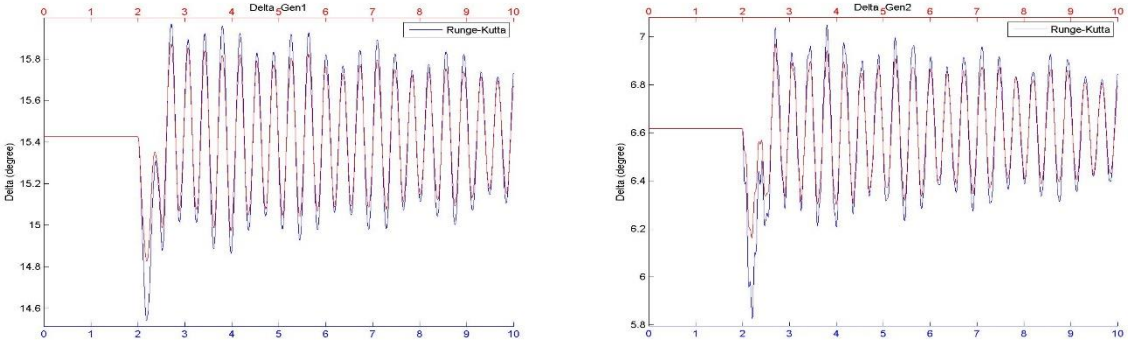
Figure continued

The same procedure for the bus angle, φ is calculated respect to COI δ for a better comparison results. Combine this figure and last voltage figure, these two methods have the same results and accuracy for 14 bus classical generator model power system on the network side.

However, Runge-Kutta method takes 5.4 seconds to run the main program, it is 18 times shorter than trapezoidal method, which takes 97 seconds. It is more efficiency to run the Runge-Kutta method without losing accuracy to have the same great performance.

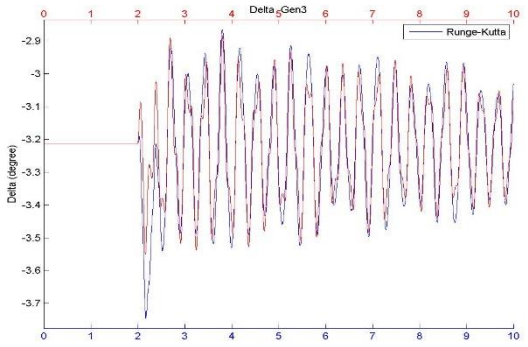
4.2.2 Two-axis Generator Model

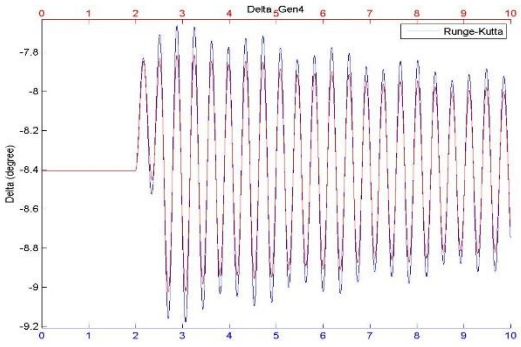
In this section, two-axis generator model is simulated. The fault happens at 2 second, and lasts for 0.5 second. The whole simulation time is 10 seconds. The fault is 1j for smaller system and for 118 bus system, both small fault and larger fault is simulated. First of this section, a smaller 14 bus system is tested. Figure 13 shows the comparison of δ for two methods of 14 bus two axis power system.



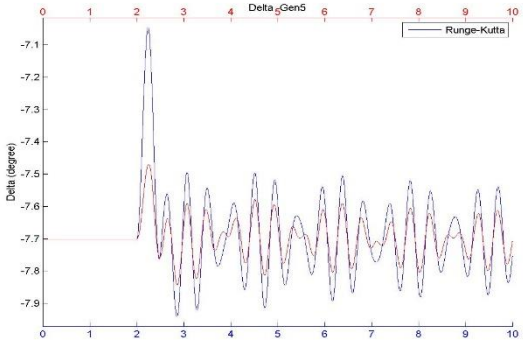
(a) (b)

Figure 13 Comparison of δ for two methods of 14 bus two axis power system. (a) Gen 1_ δ (b) Gen 2_ δ (c) Gen 3_ δ (d) Gen 4_ δ (e) Gen 5_ δ





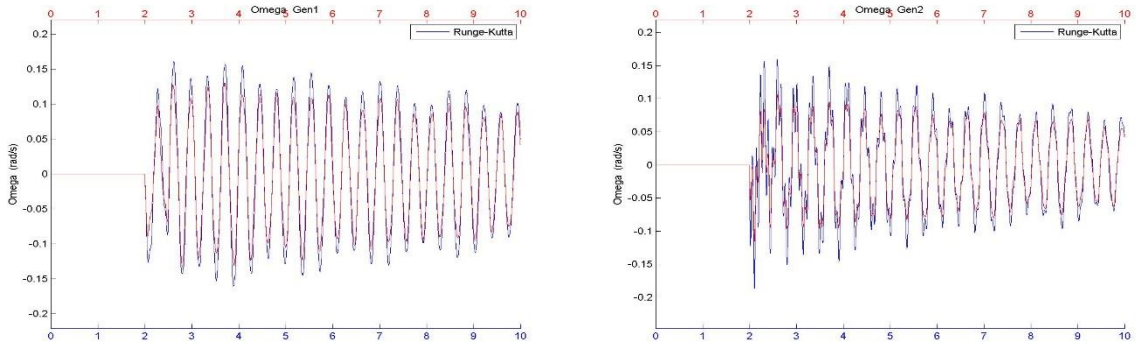
(c) (d)



(e)

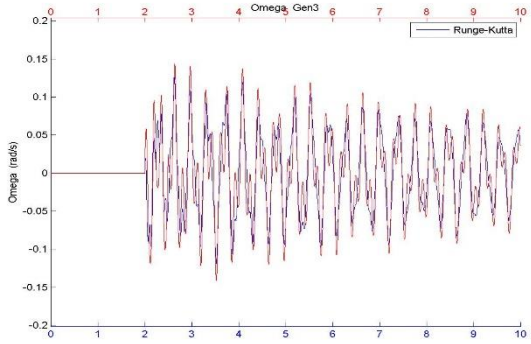
Figure continued

It is clear that compare to classical generator model, two-axis generator model is more complex and these two methods have some difference in δ comparison. In the sub-Figure (c), when a fault is injects, δ of generator 3 increases suddenly for both methods. However, they jump to different after fault value. In addition, all δ do not match in the same magnitude, especially for generator 5. Runge-Kutta method has a larger vibration than trapezoidal method. The following figure (Figure 14) displays the comparison of generator speed of two methods. Figure 15 shows the bus voltage of 14 bus two-axis generator model system.

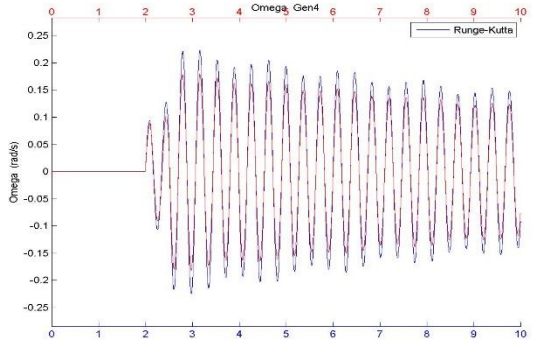


(a)

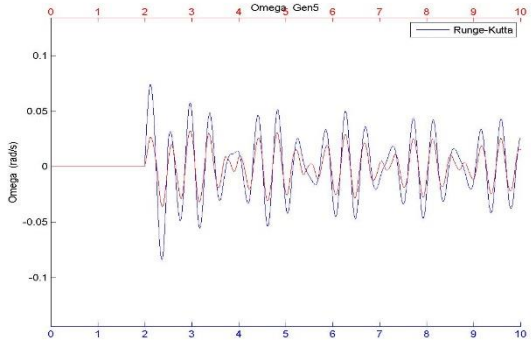
(b)



(c)

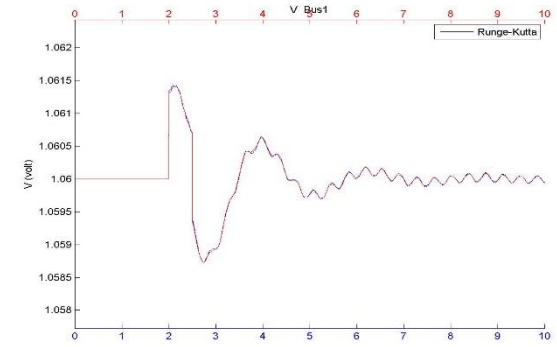


(d)

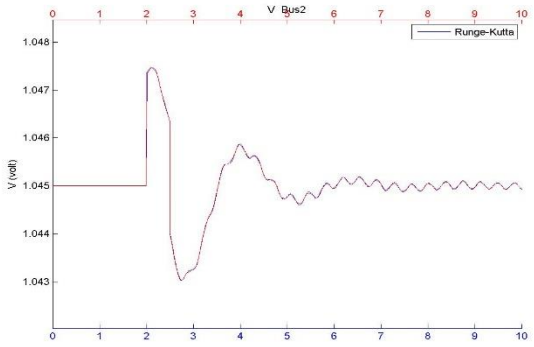


(e)

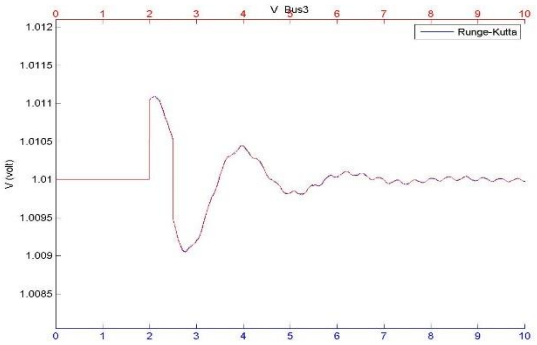
Figure 14 ω of all generators simulated for 14 bus two-axis generator model system. (a) Gen 1_ω (b) Gen 2_ω (c) Gen 3_ω (d) Gen 4_ω (e) Gen 5_ω



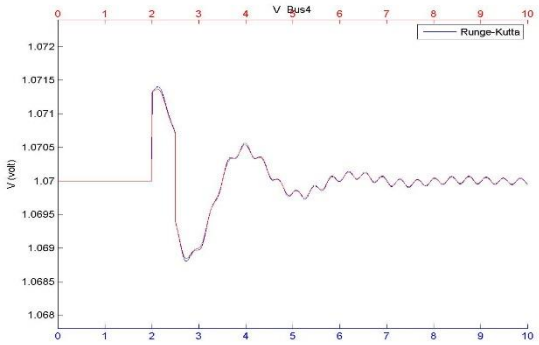
(a)



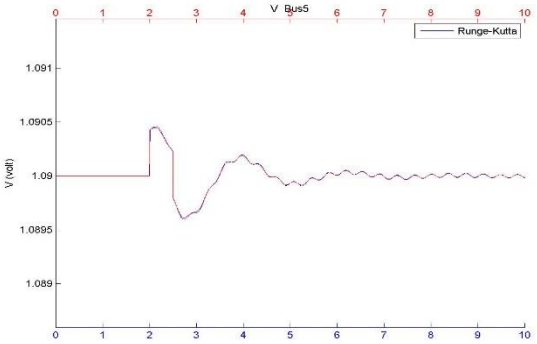
(b)



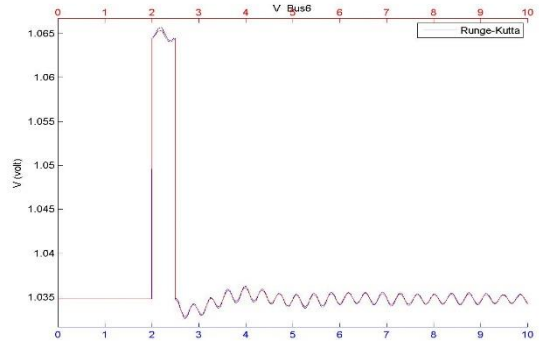
(c)



(d)

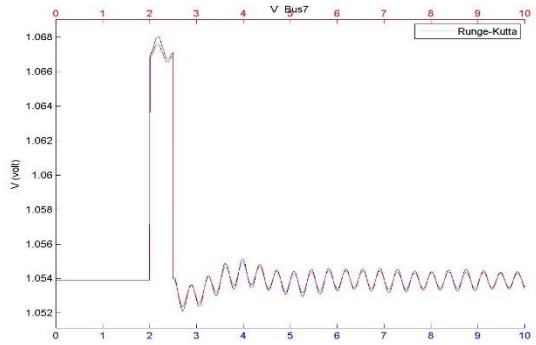


(e)

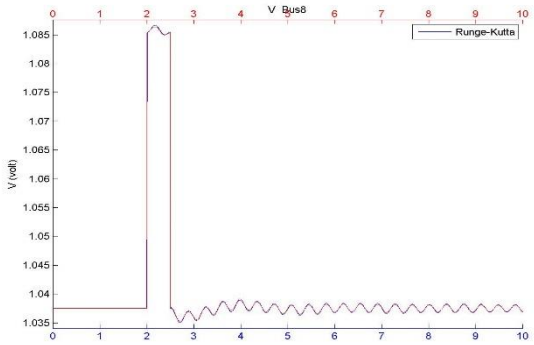


(f)

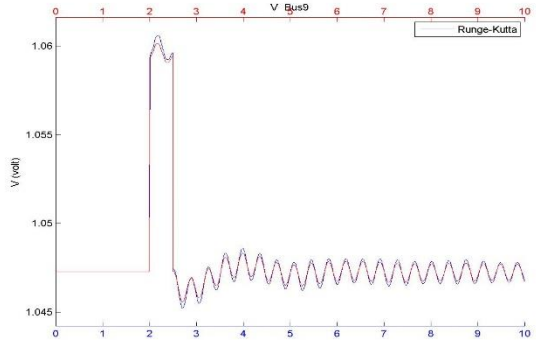
Figure 15 All 14 Bus voltage of 14 bus two-axis generator model system. (a) Bus 1_V (b) Bus 2_V (c) Bus 3_V (d) Bus 4_V (e) Bus 5_V (f) Bus 6_V (g) Bus 7_V (h) Bus 8_V (i) Bus 9_V (j) Bus 10_V (k) Bus 11_V (l) Bus 12_V (m) Bus 13_V (n) Bus 14_V



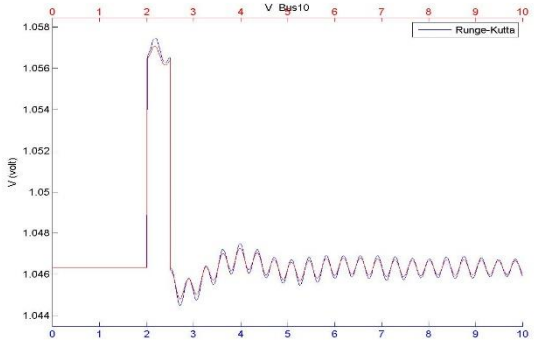
(g)



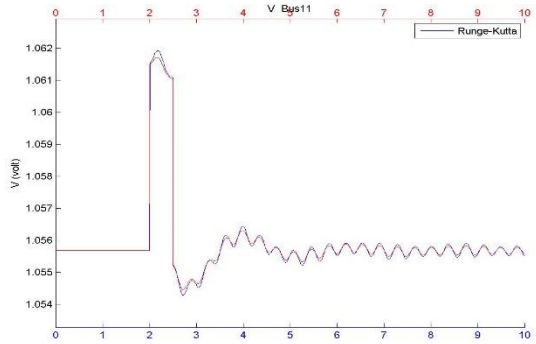
(h)



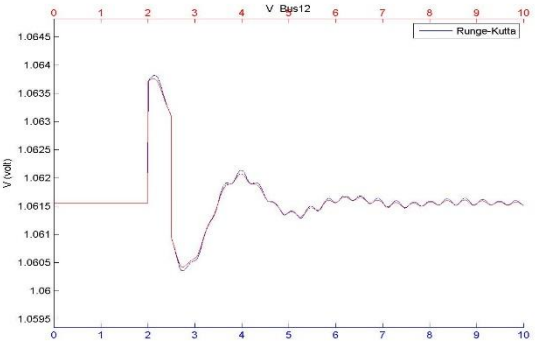
(i)



(j)

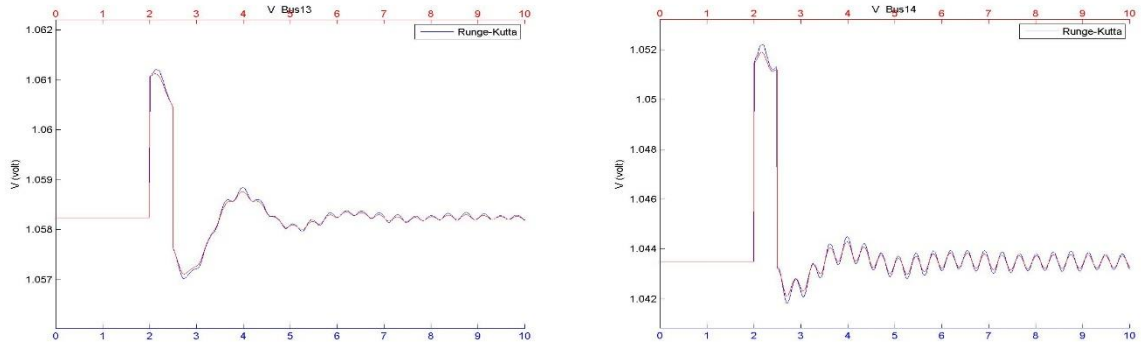


(k)



(l)

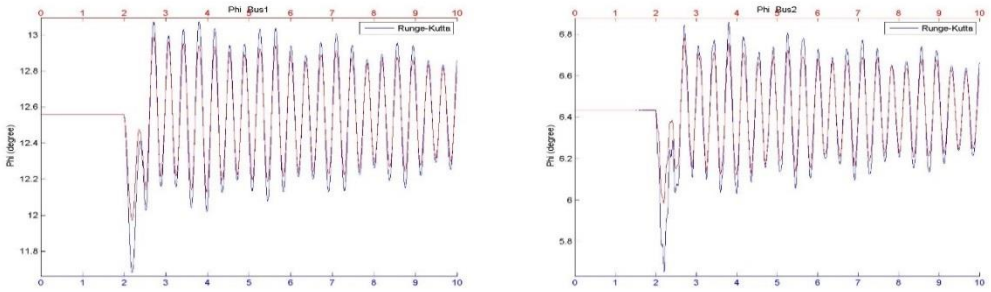
Figure continued



(m) (n)

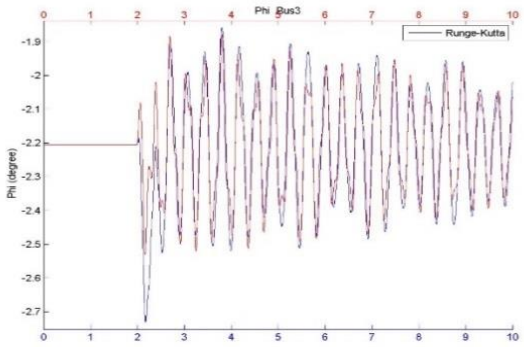
Figure continued

The comparison is promising. However, there is still 10^{-4} difference for the voltage change of these two method when facing a fault. Then, Figure 16 shows as following.

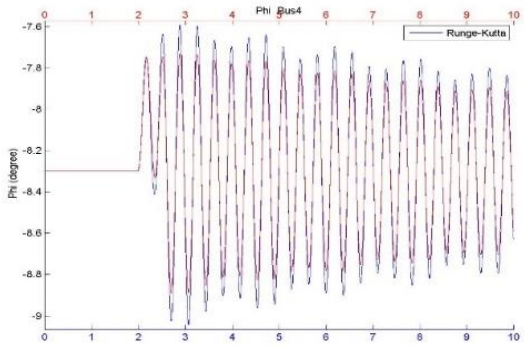


(a) (b)

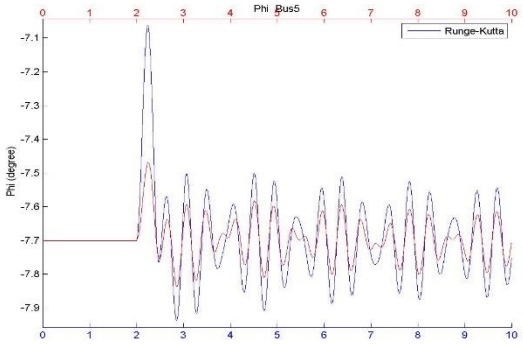
Figure 16 All 14 Bus angles of 14 bus two-axis generator model system. (a) Bus 1_ φ (b) Bus 2_ φ (c) Bus 3_ φ (d) Bus 4_ φ (e) Bus 5_ φ (f) Bus 6_ φ (g) Bus 7_ φ (h) Bus 8_ φ (i) Bus 9_ φ (j) Bus 10_ φ (k) Bus 11_ φ (l) Bus 12_ φ (m) Bus 13_ φ (n) Bus 14_ φ



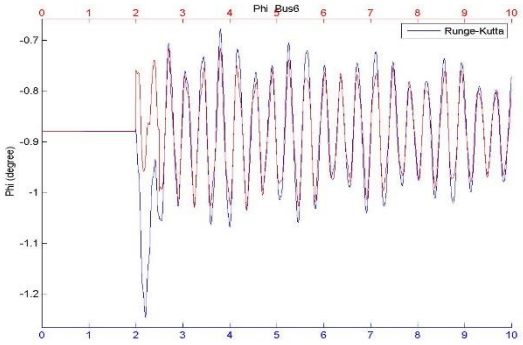
(c)



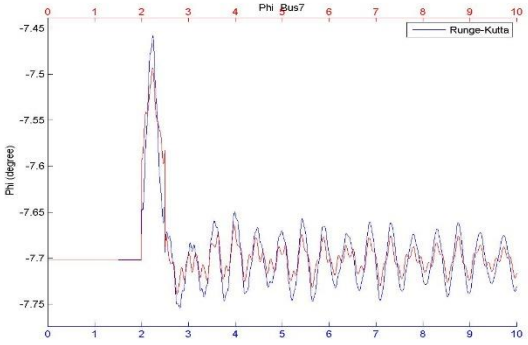
(d)



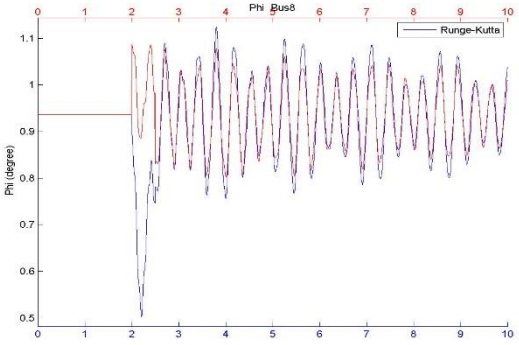
(e)



(f)

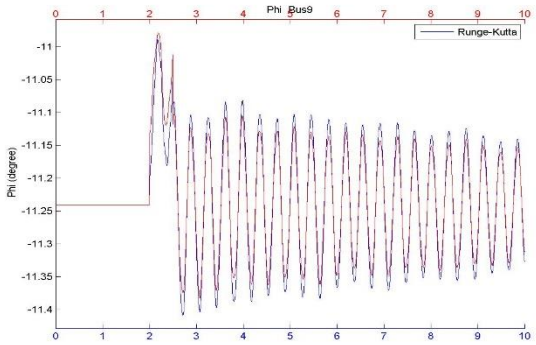


(g)

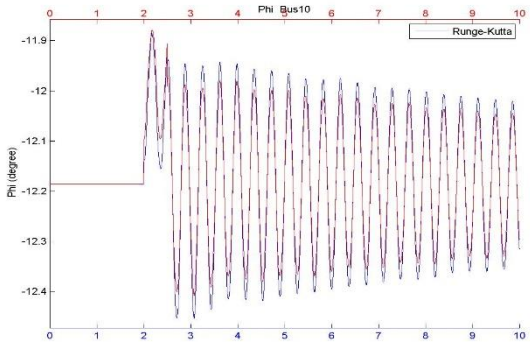


(h)

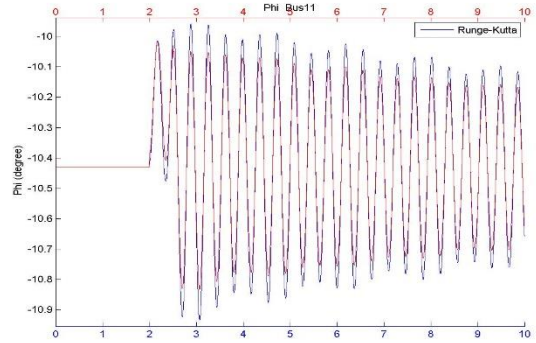
Figure continued



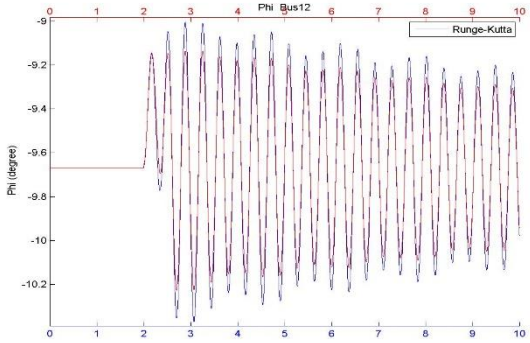
(i)



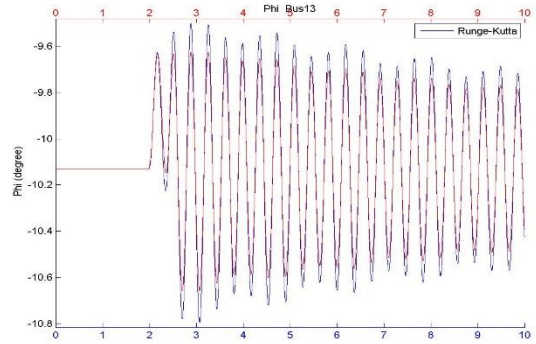
(j)



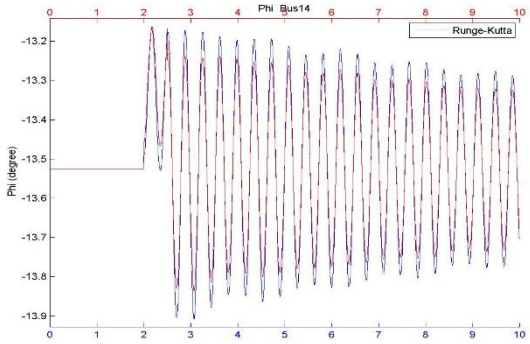
(k)



(l)



(m)



(n)

Figure continued

After the comparison, the results turn out that almost all φ are unmatched. Especially the faulted bus and load bus connected to it, which are bus 8 and bus 6.

After repeatedly testing, the reason may be the stator algebraic equations. When substitute the stator equations to power balance algebraic equations, previous voltage and bus angle are used. However, when deal with faulted situation, the pre-fault stator variables cannot be used because of the immediately changed variables. The simulation above is using the previous stator variables, which cause the different initial after fault value for all variables, even some changes are very small.

The second part of this section is to test a larger system, modified 118 bus system with 20 generators on first 20 buses. Figure 17 gives a diagram of 118 bus system. In case a small fault will not have great effects to the system, 5j fault is injected to have a remarkably change. Several important results are shown, full results in Appendix A. Figure 18 shows an enlarged figure of 118 bus system based on the faulted bus. Figure 19 and Figure 20 give an opinion of the comparison of selected generator angles. Figure 21 shows the generator angle of two selected generator bus. Figure 22 and Figure 23 explain the comparison of bus voltages and bus angles that are selected as an example.

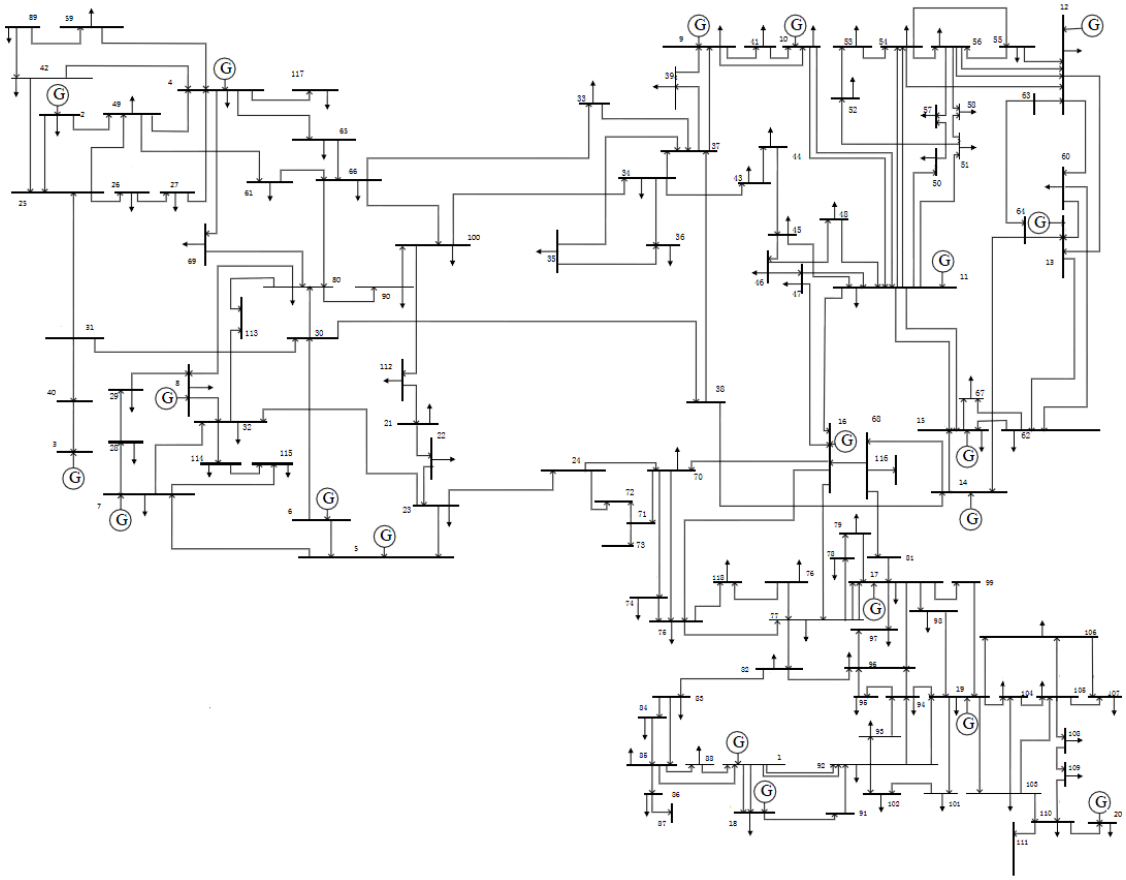


Figure 17 Modified 118 bus system diagram

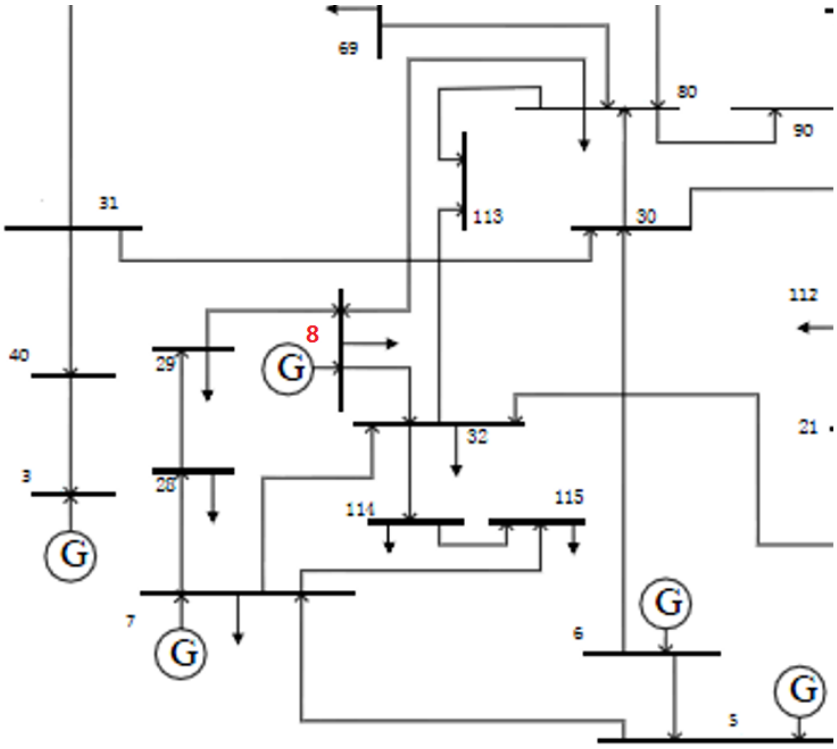
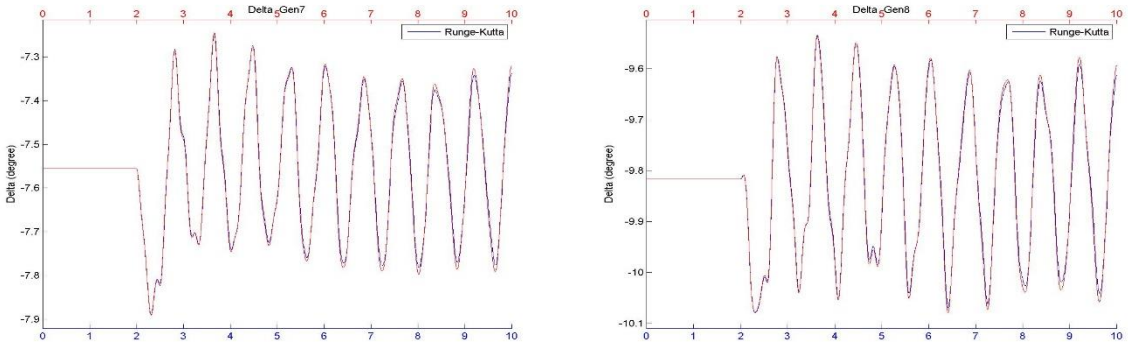
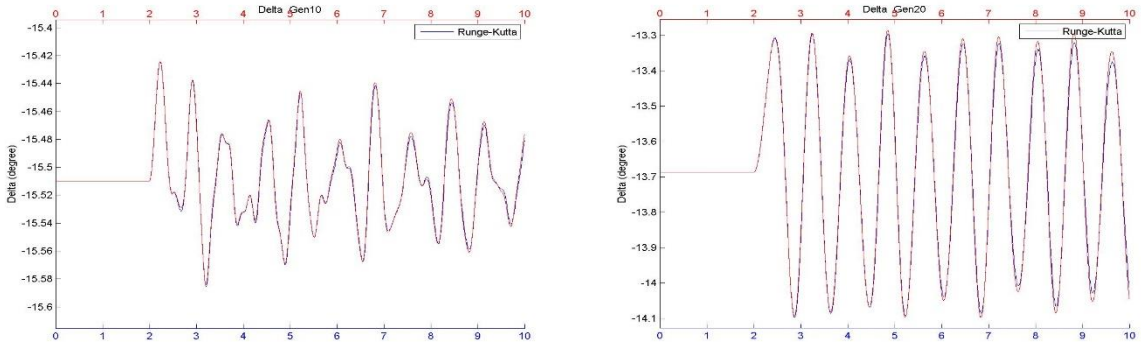


Figure 18 Relative buses connect to faulted bus



(a) (b)

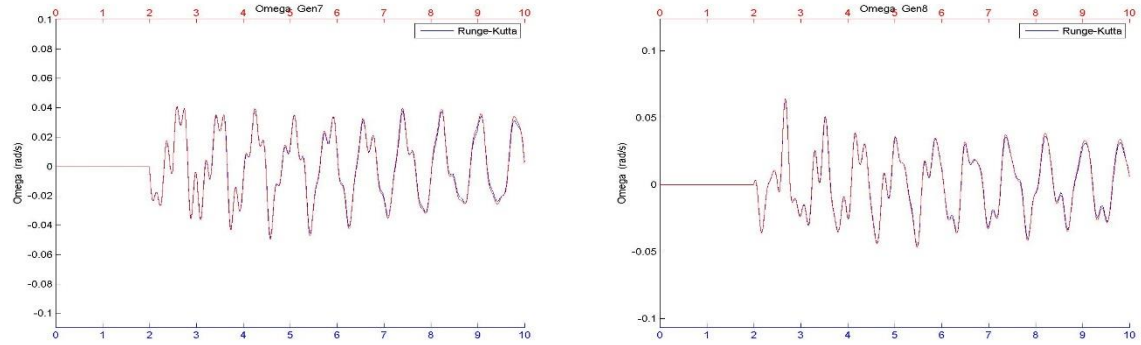
Figure 19 Selected δ of 118 bus two-axis generator model system. (a) Gen 7_ δ (b) Gen 8_ δ



(a) (b)

Figure 20 Selected δ of 118 bus two-axis generator model system. (a) Gen 10_ δ (b) Gen 20_ δ

As shown in figures above, it is clear that these two methods results match with each other in this larger system, even for the after fault initial value is the same for every δ. In addition, Figure 20 (a) and (b) show the same changes when the status changes.



(a) (b)

Figure 21 Selected ω of 118 bus two-axis generator model system. (a) Gen 7_ ω (b) Gen 8_ ω

Figure 21 shows the largest shock range ω, which are the faulted bus and the closest generator bus. All the other ω matches and smooth for whole simulation time.

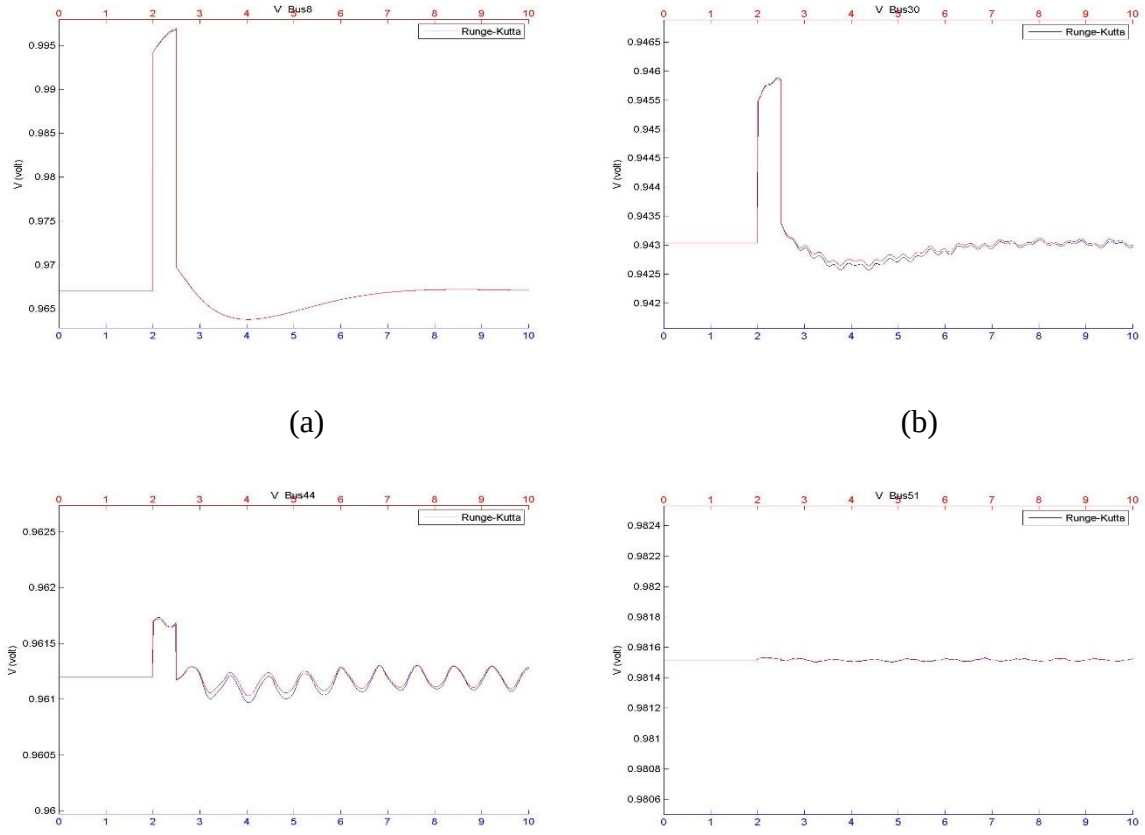


Figure 22 Selected bus voltage of 118 bus two-axis generator model system. (a) Bus 8_V (b) Bus 30_V (c) Bus 44_V (d) Bus 51_V

The above Figure 22 shows faulted bus and buses that have different distance from the fault bus. The 30 directly connects to bus 8, bus 44 is far away from the faulted bus, and bus 51 is even farther. As it shows, on faulted bus, the voltage jumps almost 0.03 p.u., which gives more than 2% voltage change. On the connected bus, the voltage jump reduces to 0.003 p.u., which because the generator on bus 8 tries to keep the voltage constant and not to influence other buses. For not being connected bus or even farther buses, the influence of the fault is too small to notice, that is shown in (d).

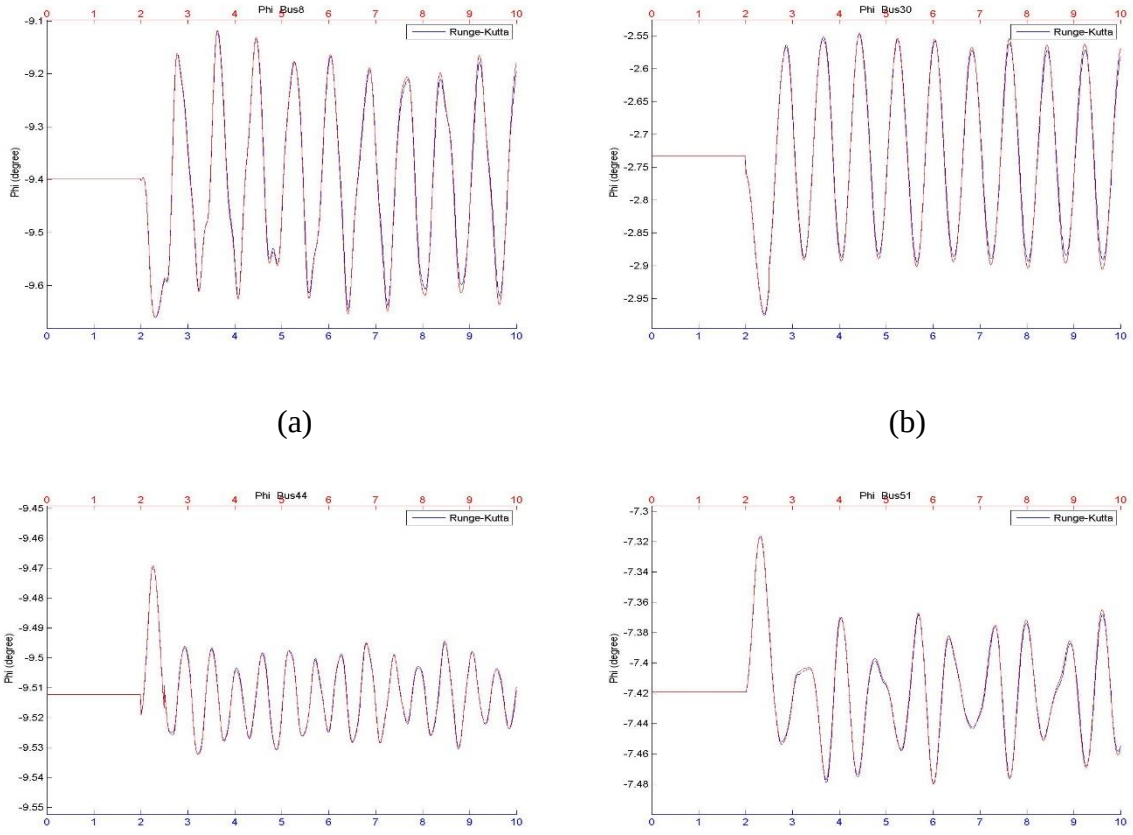


Figure 23 Selected φ of 118 bus two-axis generator model system. (a) Bus 8_ φ (b) Bus 30_ φ (c) Bus 44_ φ (d) Bus 51_ φ

Even though some bus voltages have almost no change, but φ changes significantly for every buses. The φ change is related to the inner power balance of the whole system and can be hardly predicted.

In this larger system simulation, the results of two methods comparison are great in both frequency and magnitude. Besides, the Runge-Kutta method takes 400 seconds, compares to trapezoidal method's 14300 seconds, it is 35.8 times shorter.

CHAPTER 5. CONCLUSIONS AND FUTRUE WORK

5.1 Conclusions

In this thesis, a new methodology is used to solve differential-algebraic equations. To reduce the complexity of solving it, all algebraic equations are converted to differential equations. This pure differential equations are solved by Runge-Kutta algorithm and compared to previous trapezoidal integration algorithm. For classical generator model generator systems, these two methods have exactly the same results and Runge-Kutta algorithm's simulation time is about 20 times shorter than the other. When switching to the two-axis generator model generator system, Runge-Kutta algorithm's result on smaller system is inaccurate, because the fault can greatly change generator variables, which cannot be assumed or predicted before fault. However, for larger system like 118 bus system, even a larger fault cannot have this much influence on generator side. The results of these two algorithms match again. This is because the larger system have more generators that capable to reduce the influence of the fault in order to avoid the fault attack to the rest of the system. When the fault is huge or more solid, the same results are shown as the smaller system.

5.2 Future Work

The next step of this study is to solving the stator algebraic equation when facing a fault. By bypassing the stator algebraic equation, the system can be simulated correctly when a fault is injected for both smaller system and larger one.

REFERENCES

[1] Grainger, J. J., & Stevenson, W. D. *Power system analysis* (Vol. 621). New York: McGraw-Hill., 1994.

[2] Mehraeen, S., Jagannathan, S., & Crow, M. L. Novel dynamic representation and control of power systems with FACTS Devices. *Power Systems, IEEE Transactions on, 25*(3), 1542-1554, 2010.

[3] Sauer, P. W., & Pai, M. A. *Power system dynamics and stability* (Vol. 4). Upper Saddle River, NJ: Prentice Hall, 1998.

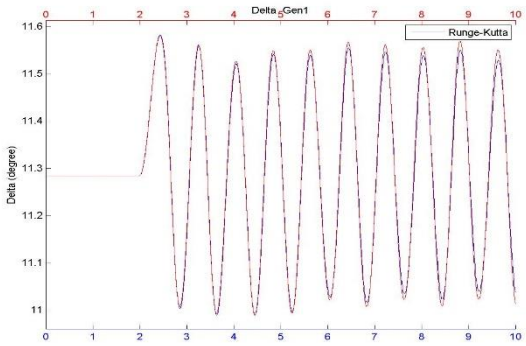
[4] Dixon, W. E., Behal, A., Dawson, D. M., & Nagarkatti, S. P. (2002). *Nonlinear Control of Engineering Systems*. Birkhauser Boston.

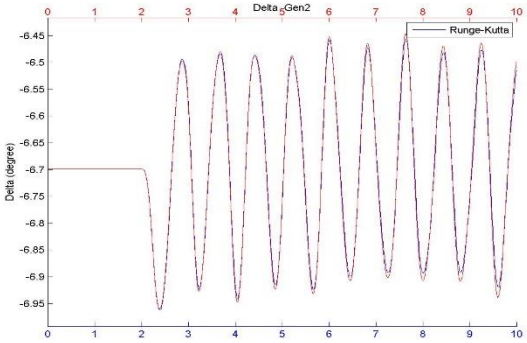
[5] Kazemlou, S., & Mehraeen, S. Stability of multi-generator power system with penetration of renewable energy sources. In *Power and Energy Society General Meeting, 2012 IEEE* (pp. 1-6). IEEE, Jul, 2012.

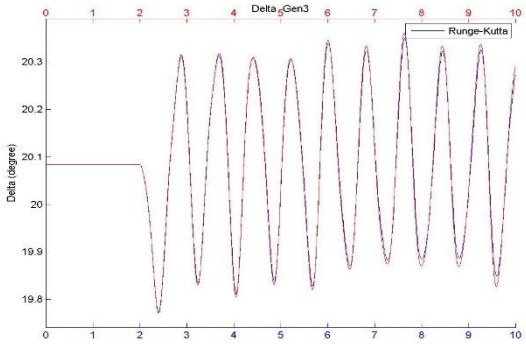
[6] Saadat, H. Power system analysis. WCB/McGraw-Hill, 1990.

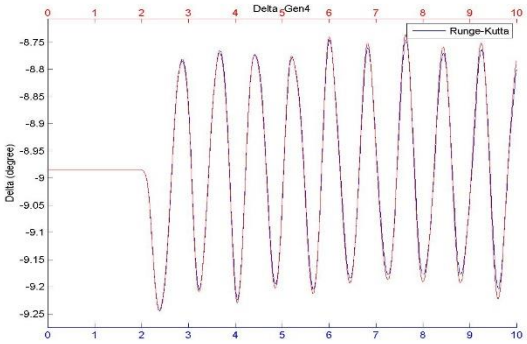
[7] Kundur, P., Paserba, J., Ajjarapu, V., Andersson, G., Bose, A., Canizares, C., Hatziargyriou, N., Hill, D., Stankovic, A., Taylor, C., Cutsem, V. T., and Vittal, V. "Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1387–1401, Aug. 2004.

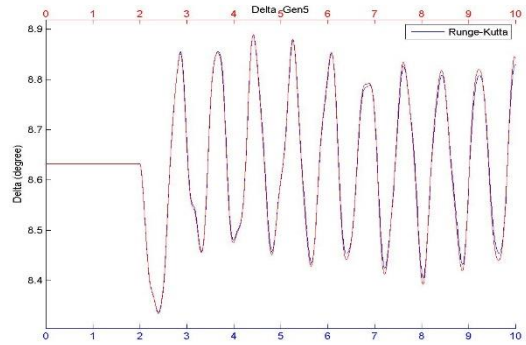
APPENDIX A: ALL SIMULATION RESULTS FOR 118 BUS SYSTEM

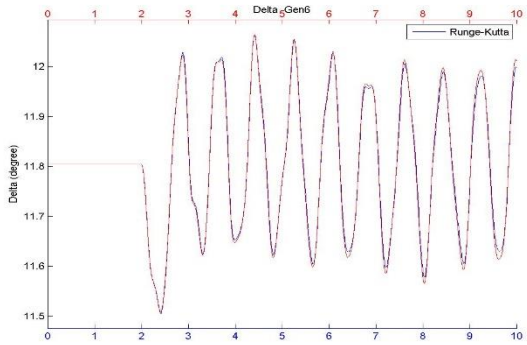


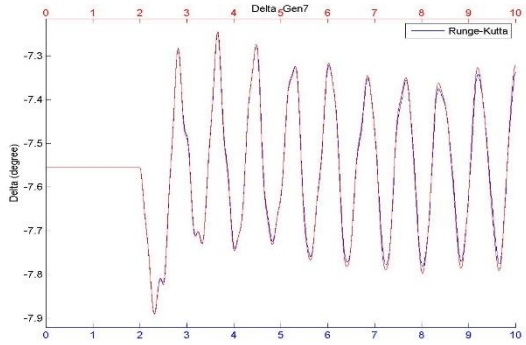


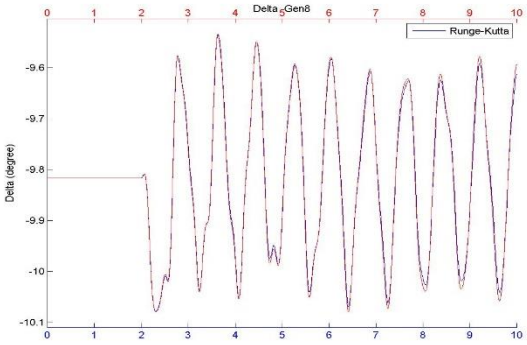


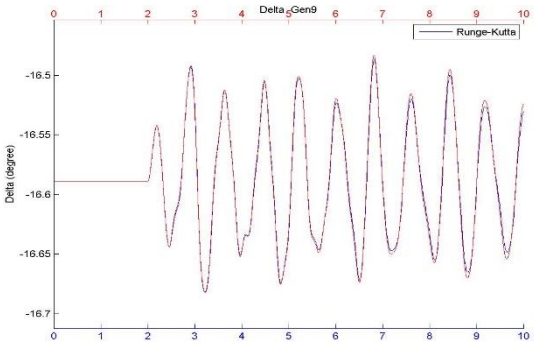


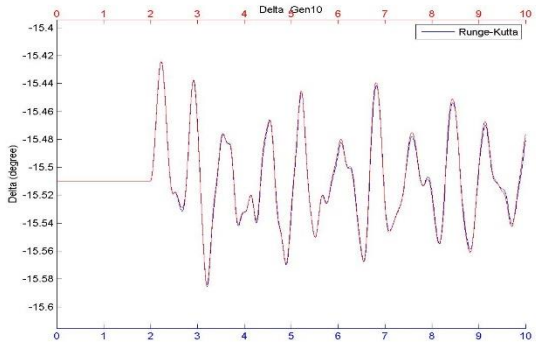


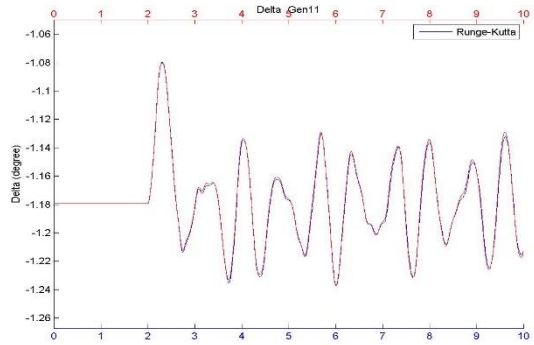


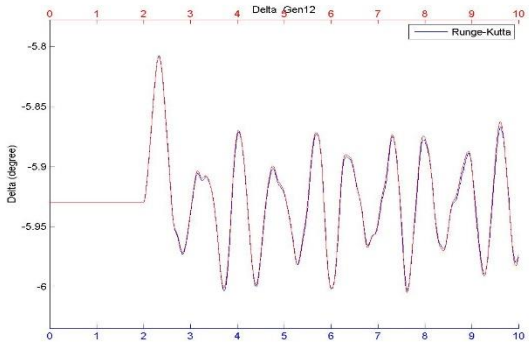


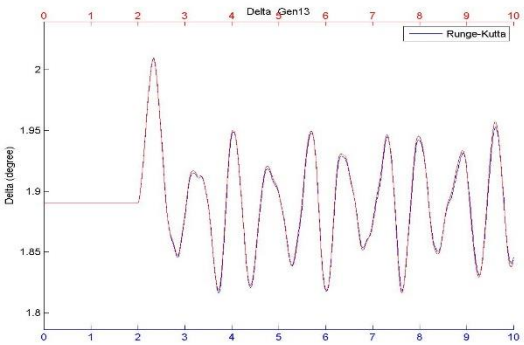


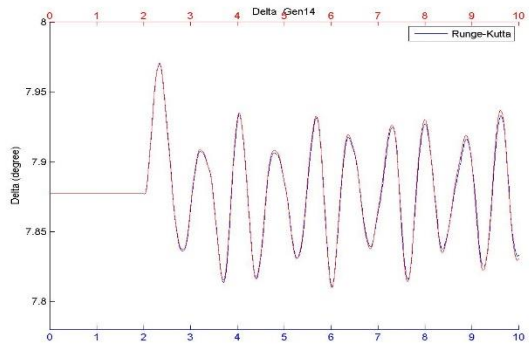

Delta_Gen9
Runge-Kutta
Delta (degree)

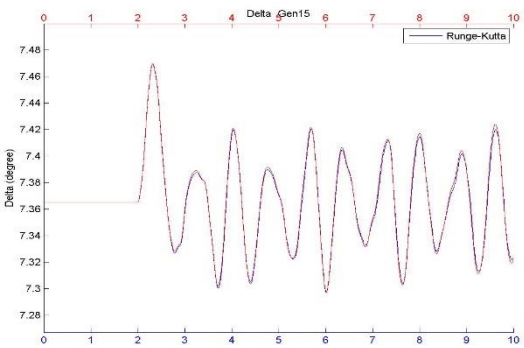

Delta_Gen10
Runge-Kutta
Delta (degree)

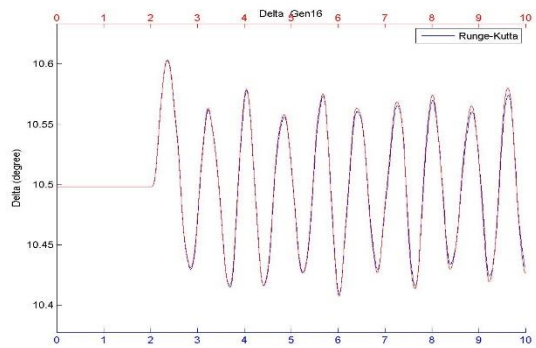

Delta_Gen11
Runge-Kutta
Delta (degree)

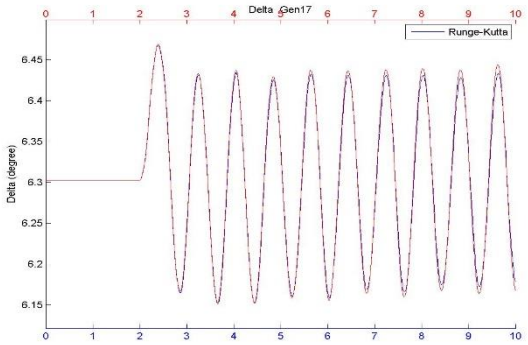

Delta_Gen12
Runge-Kutta
Delta (degree)

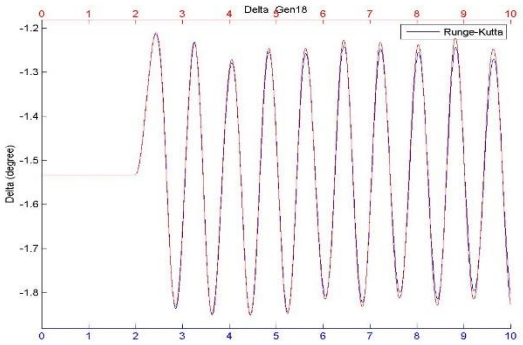

Delta_Gen13
Runge-Kutta
Delta (degree)

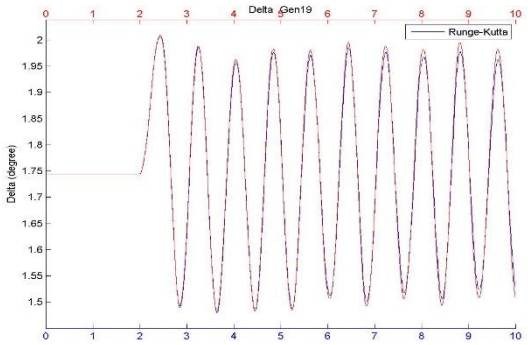

Delta_Gen14
Runge-Kutta
Delta (degree)

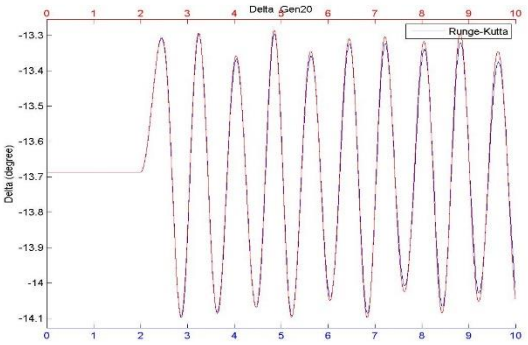

Delta_Gen15
Runge-Kutta
Delta (degree)

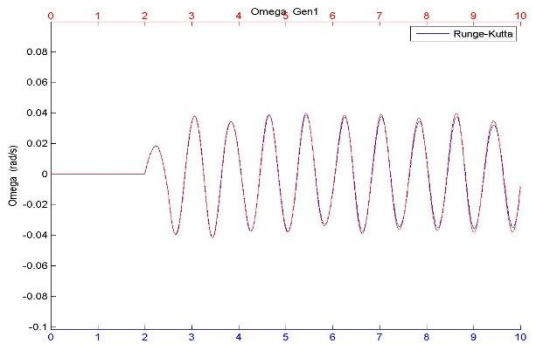

Delta_Gen16
Runge-Kutta
Delta (degree)

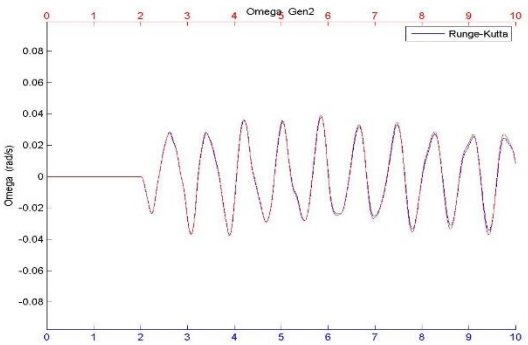

Delta_Gen17
Runge-Kutta
Delta (degree)

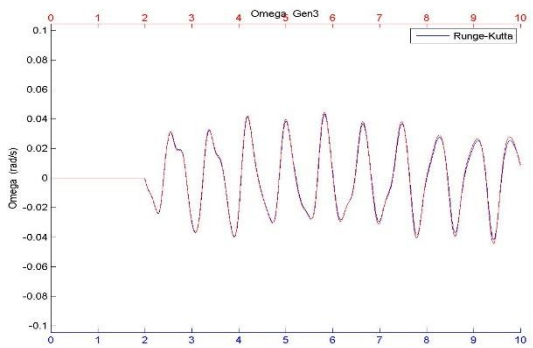

Delta_Gen18
Runge-Kutta
Delta (degree)

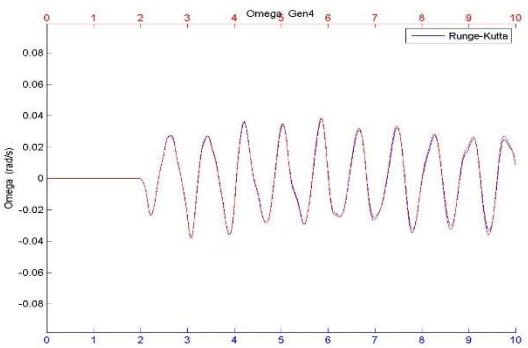

Delta_Gen19
Runge-Kutta
Delta (degree)

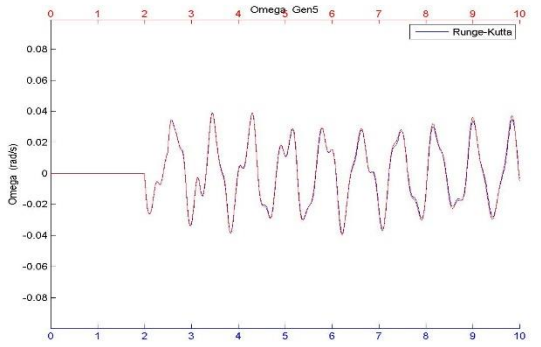

Delta_Gen20
Runge-Kutta
Delta (degree)

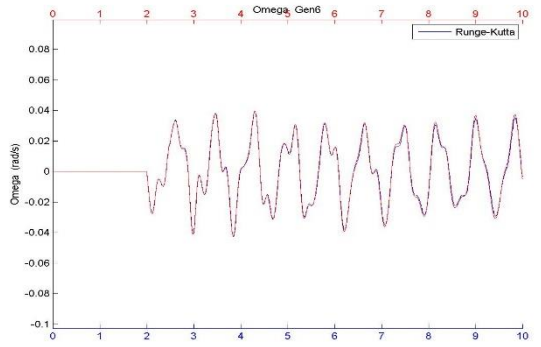

Omega_Gen1
Runge-Kutta
Omega (rad/s)

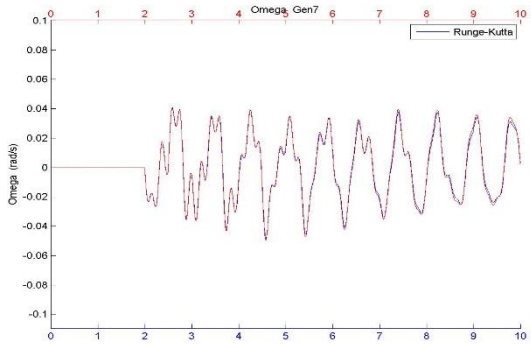

Omega_Gen2
Runge-Kutta
Omega (rad/s)

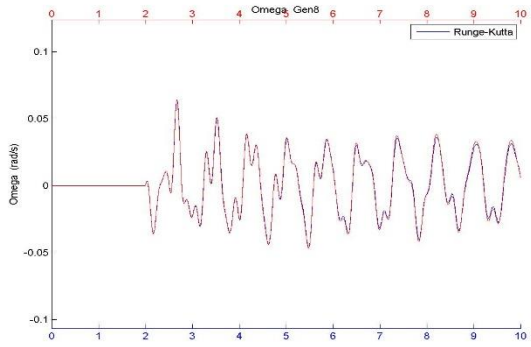

Omega_Gen3
Runge-Kutta
Omega (rad/s)

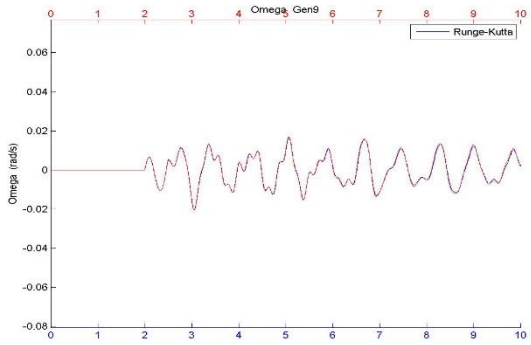

Omega_Gen4
Runge-Kutta
Omega (rad/s)

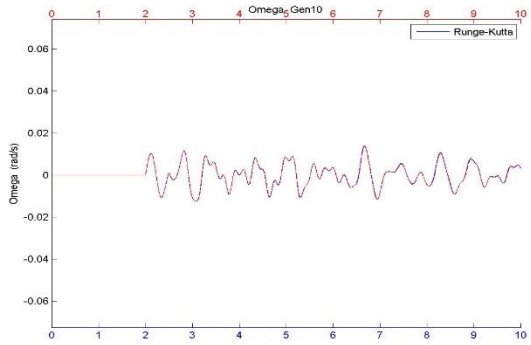

Omega_Gen5
Runge-Kutta
Omega (rad/s)

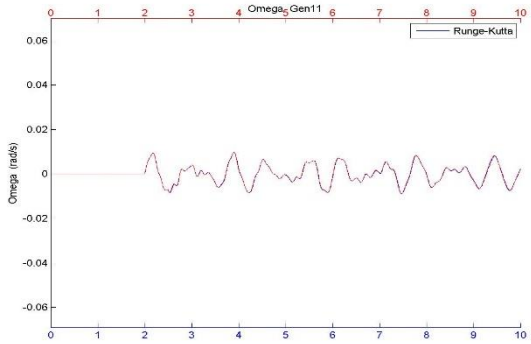

Omega_Gen6
Runge-Kutta
Omega (rad/s)

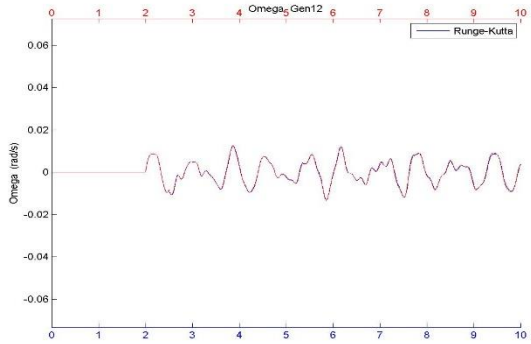

Omega_Gen7
Runge-Kutta
Omega (rad/s)

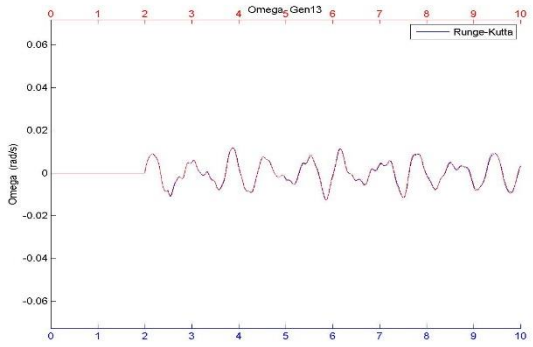

Omega_Gen8
Runge-Kutta
Omega (rad/s)

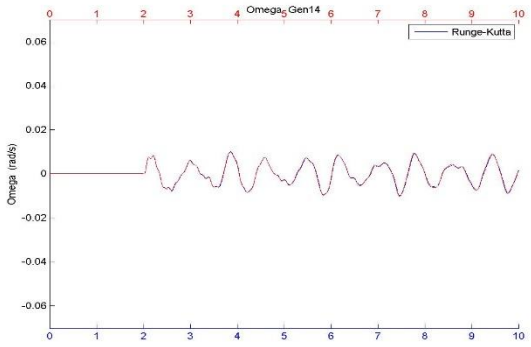

Omega_Gen9
Runge-Kutta
Omega (rad/s)

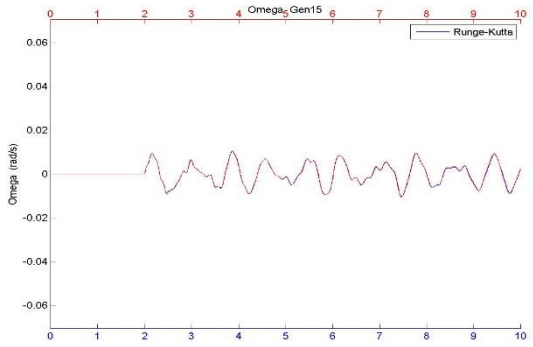

Omega_Gen10
Runge-Kutta
Omega (rad/s)

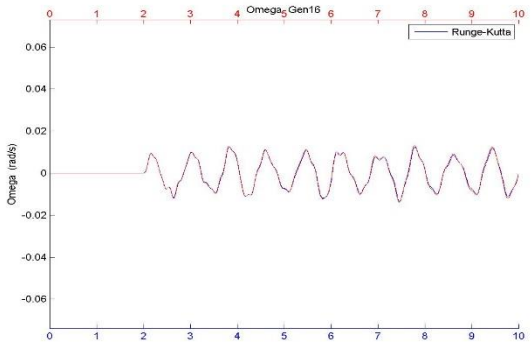

Omega_Gen11
Runge-Kutta
Omega (rad/s)

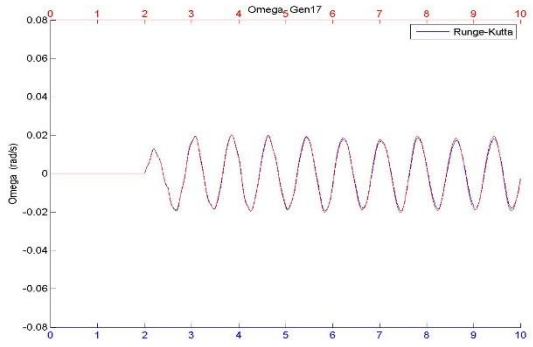

Omega_Gen12
Runge-Kutta
Omega (rad/s)

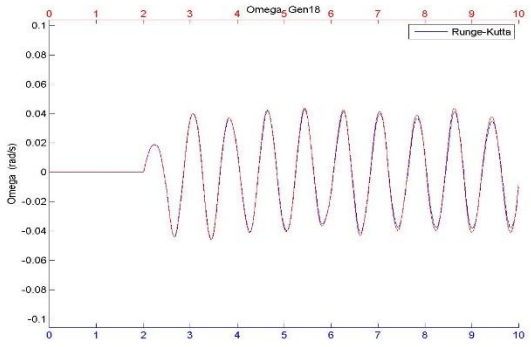

Omega_5 Gen13
Runge-Kutta
Omega (rad/s)

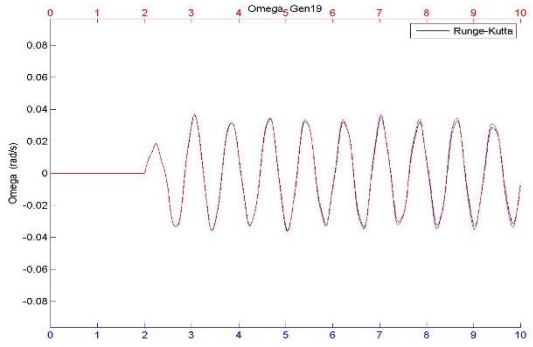

Omega_5 Gen14
Runge-Kutta
Omega (rad/s)

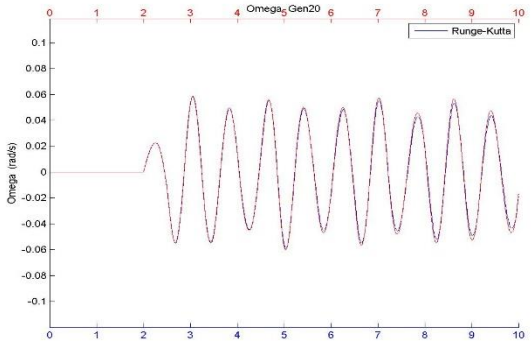

Omega_5 Gen15
Runge-Kutta
Omega (rad/s)

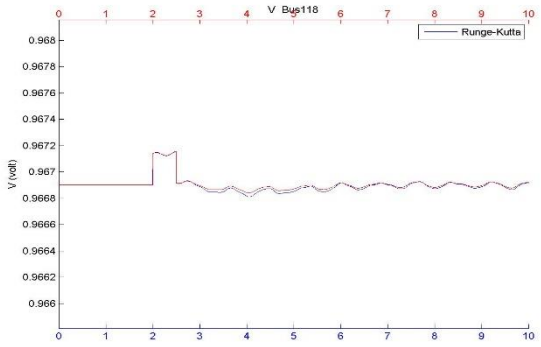

Omega_5 Gen16
Runge-Kutta
Omega (rad/s)

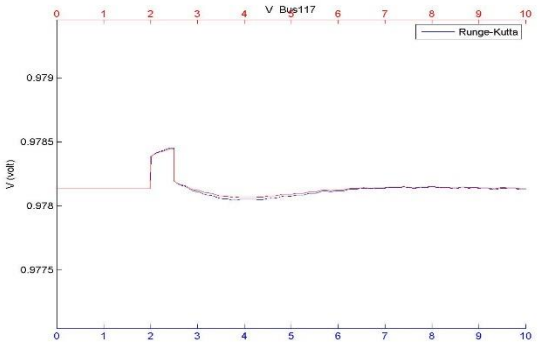

Omega_5 Gen17
Runge-Kutta
Omega (rad/s)

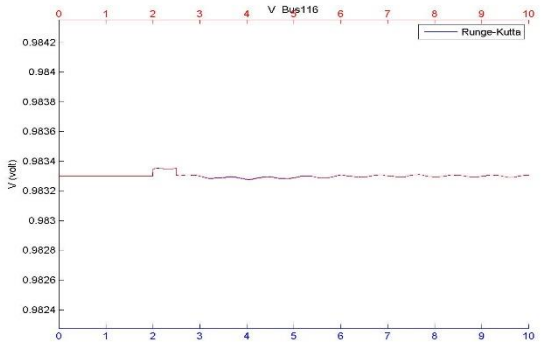

Omega_5 Gen18
Runge-Kutta
Omega (rad/s)

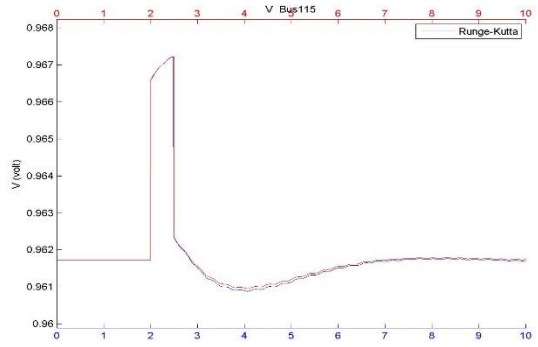

Omega_5 Gen19
Runge-Kutta
Omega (rad/s)

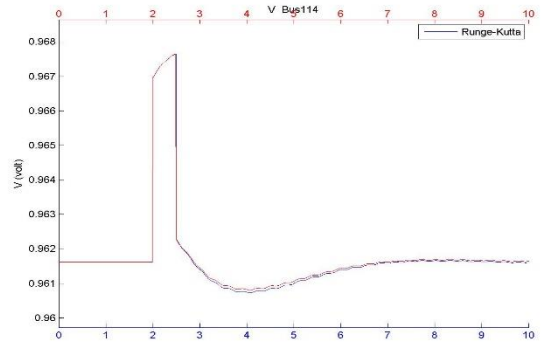

Omega_5 Gen20
Runge-Kutta
Omega (rad/s)

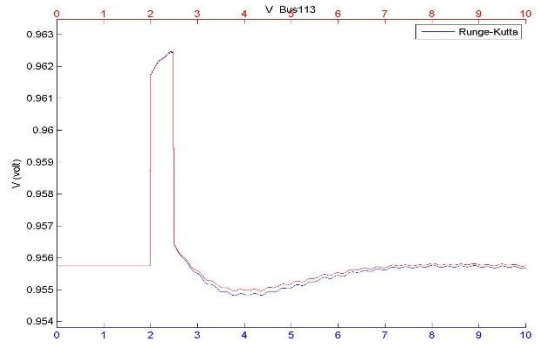

V Bus118
Runge-Kutta
V (volt)

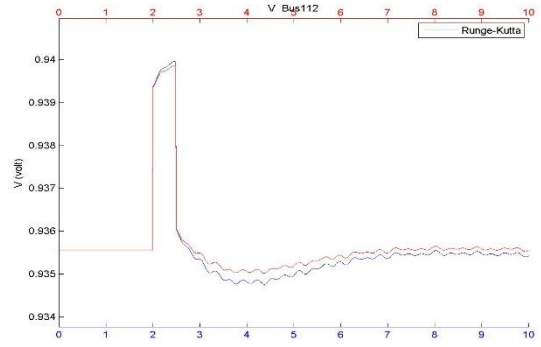

V Bus117
Runge-Kutta
V (volt)

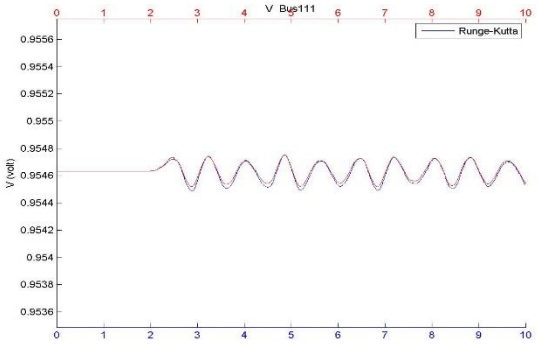

V Bus116
Runge-Kutta
V (volt)

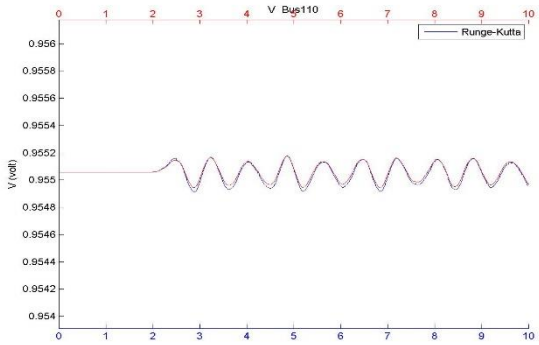

V Bus115
Runge-Kutta
V (volt)

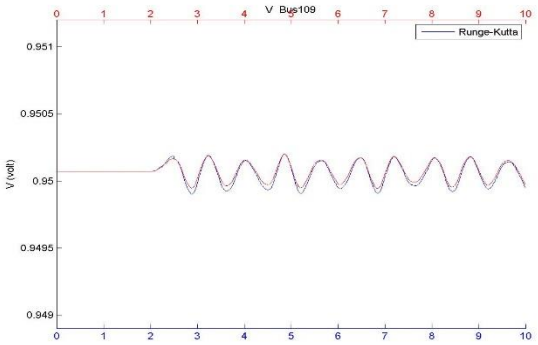

V Bus114
Runge-Kutta
V (volt)

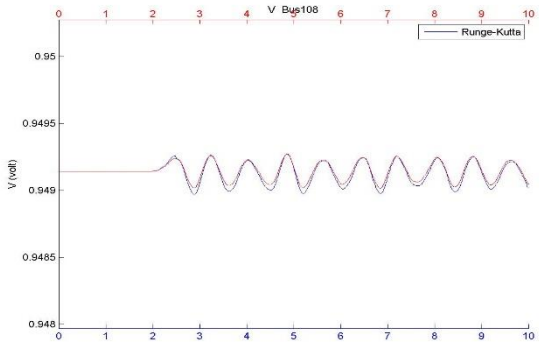

V Bus113
Runge-Kutta
V (volt)

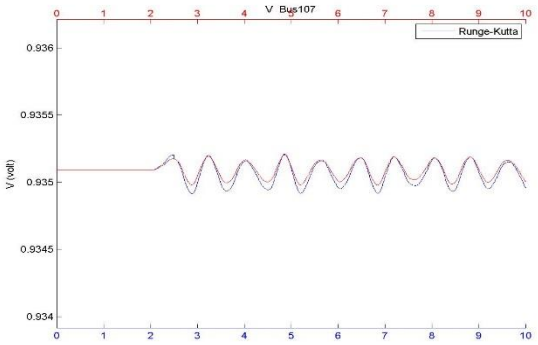

V Bus112
Runge-Kutta
V (volt)

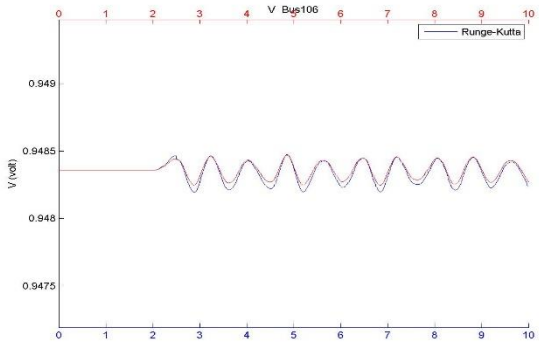

V Bus111
Runge-Kutta
V (volt)

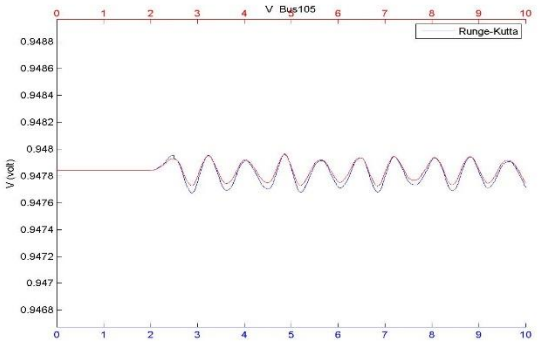

V Bus110
Runge-Kutta
V (volt)

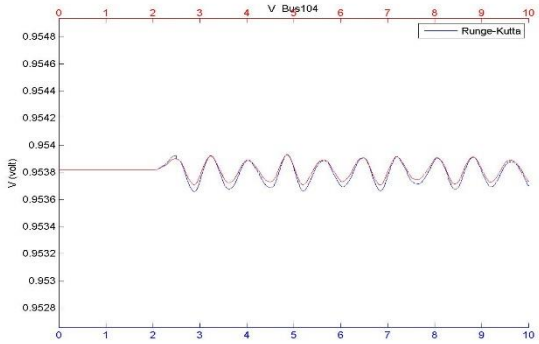

V Bus109
Runge-Kutta
V (volt)

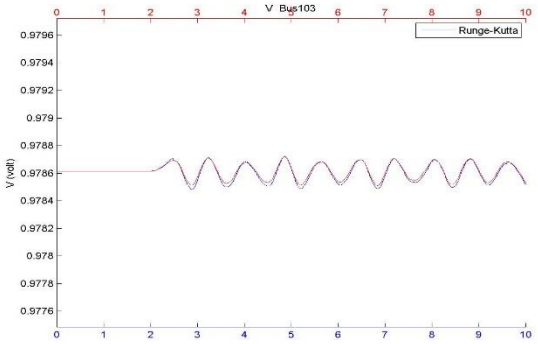

V Bus108
Runge-Kutta
V (volt)

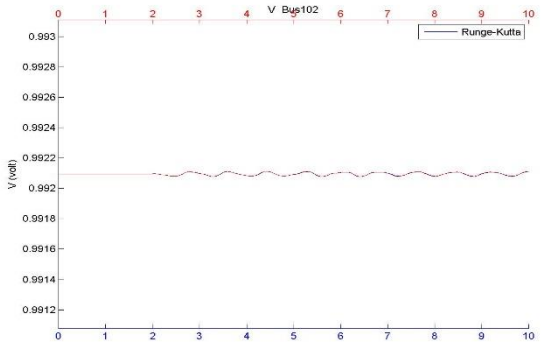

V Bus107
Runge-Kutta
V (volt)

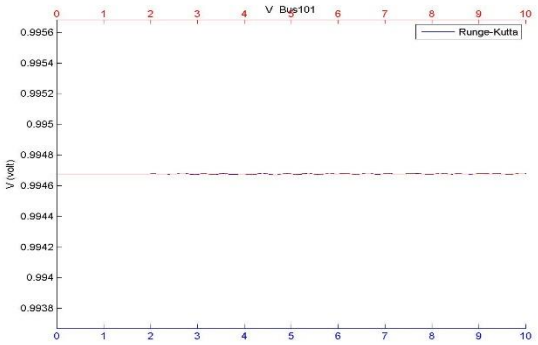

V Bus106
Runge-Kutta
V (volt)

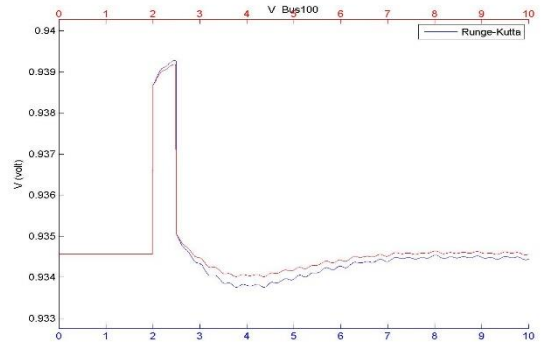

V Bus105
Runge-Kutta
V (volt)

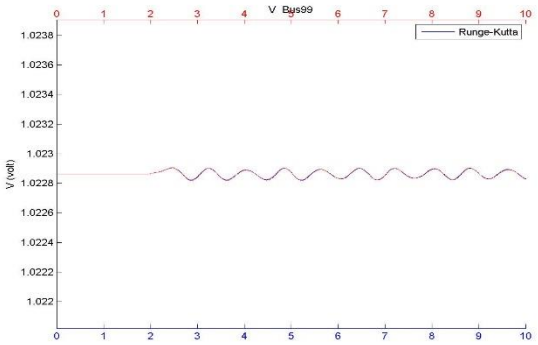

V Bus104
Runge-Kutta
V (volt)

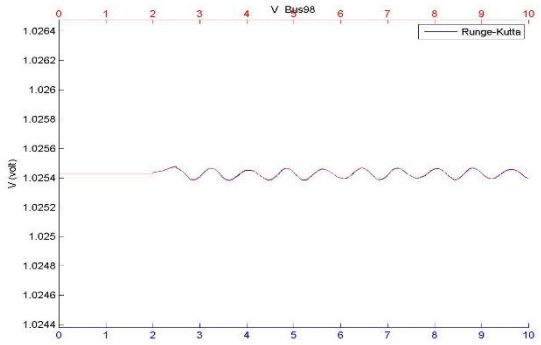

V Bus103
Runge-Kutta
V (volt)

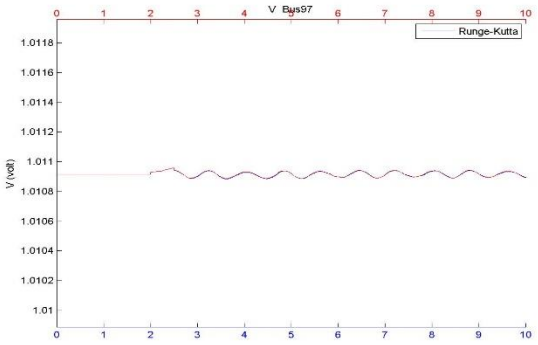

V Bus102
Runge-Kutta
V (volt)

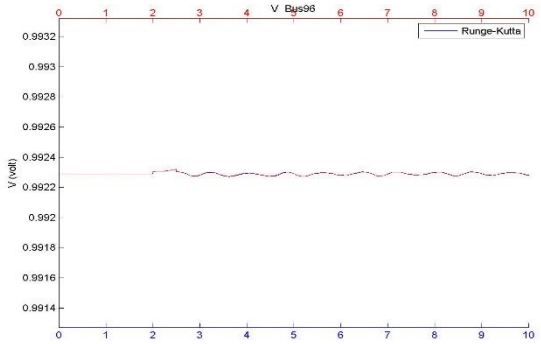

V Bus101
Runge-Kutta
V (volt)

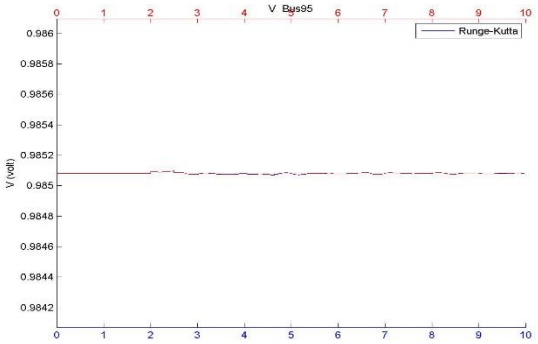

V Bus100
Runge-Kutta
V (volt)

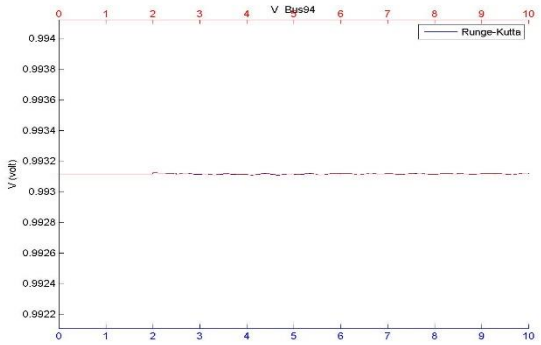

V Bus99
Runge-Kutta
V (volt)

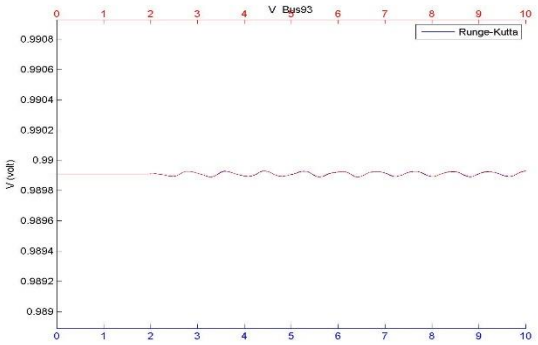

V Bus98
Runge-Kutta
V (volt)

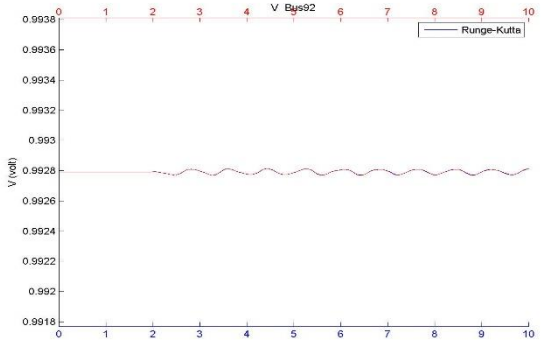

V Bus97
Runge-Kutta
V (volt)

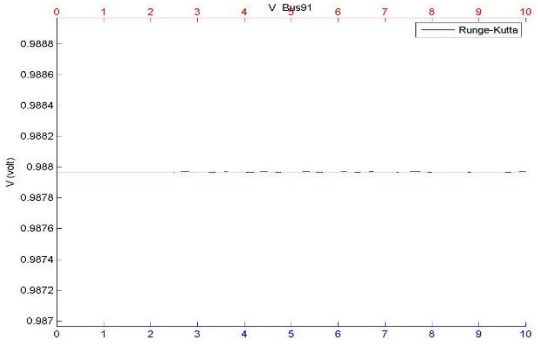

V Bus96
Runge-Kutta
V (volt)

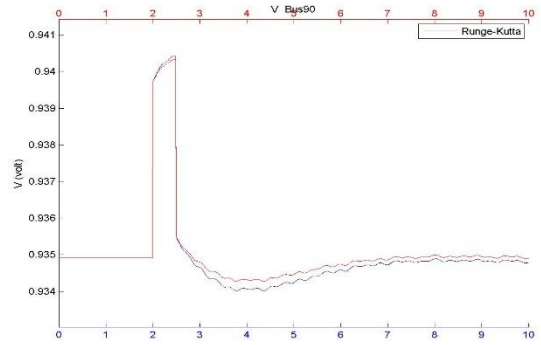

V Bus95
Runge-Kutta
V (volt)

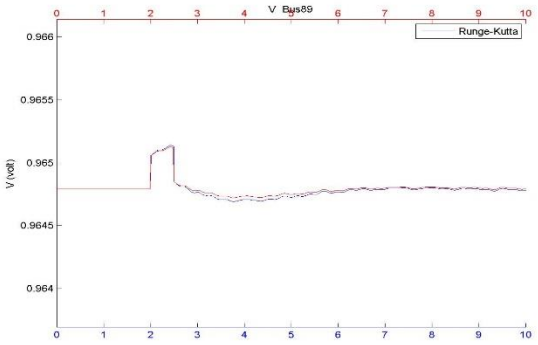

V Bus94
Runge-Kutta
V (volt)

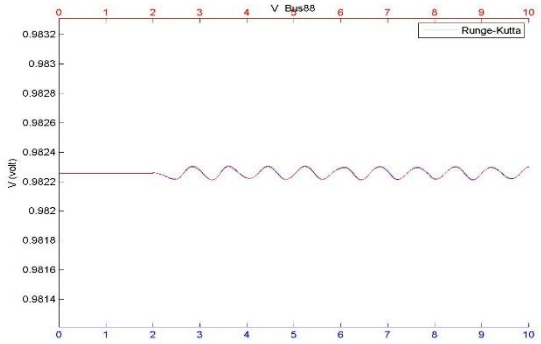

V Bus93
Runge-Kutta
V (volt)

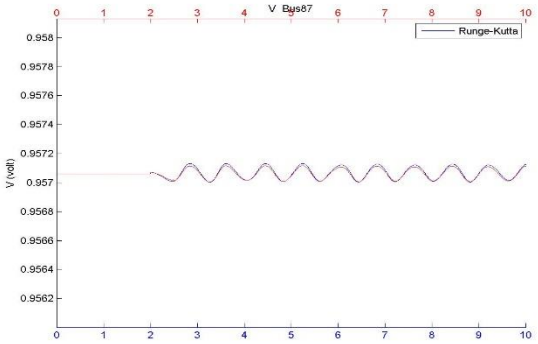

V Bus92
Runge-Kutta
V (volt)

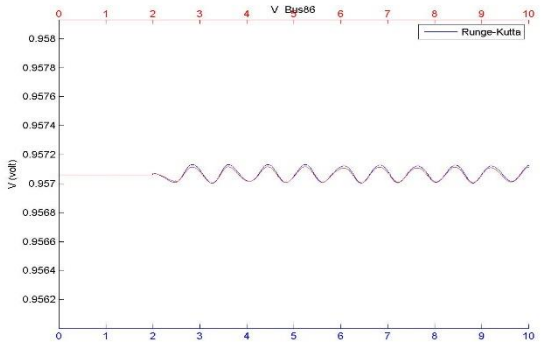

V Bus91
Runge-Kutta
V (volt)

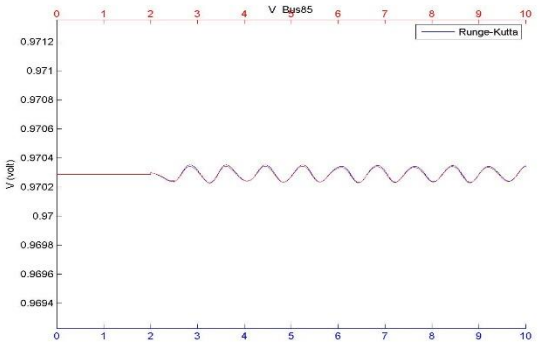

V Bus90
Runge-Kutta
V (volt)

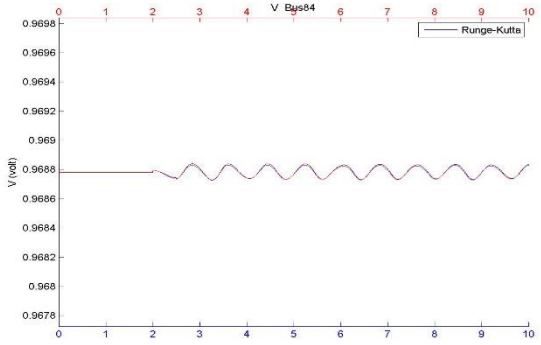

V Bus89
Runge-Kutta
V (volt)

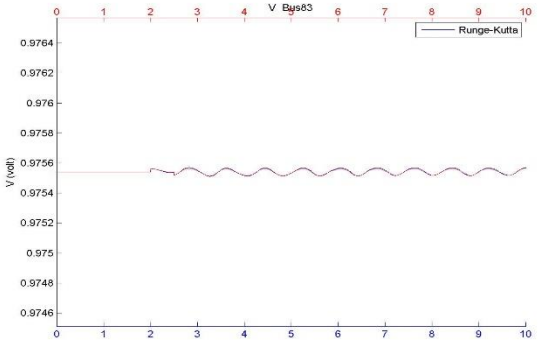

V Bus88
Runge-Kutta
V (volt)

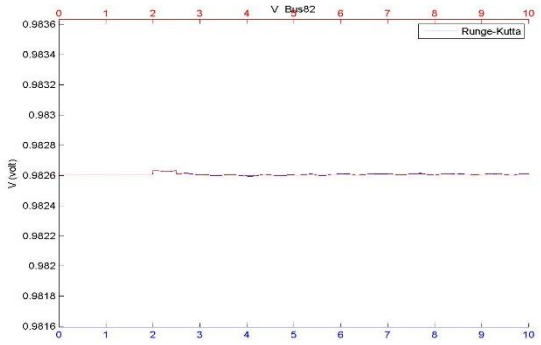

V Bus87
Runge-Kutta
V (volt)

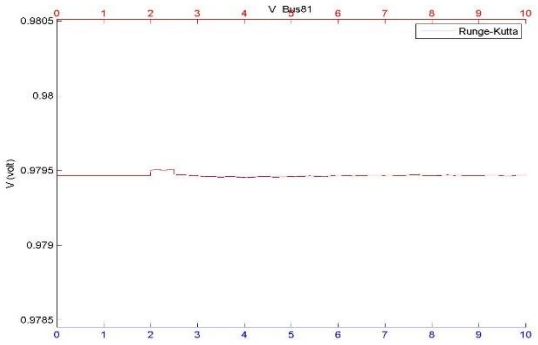

V Bus86
Runge-Kutta
V (volt)

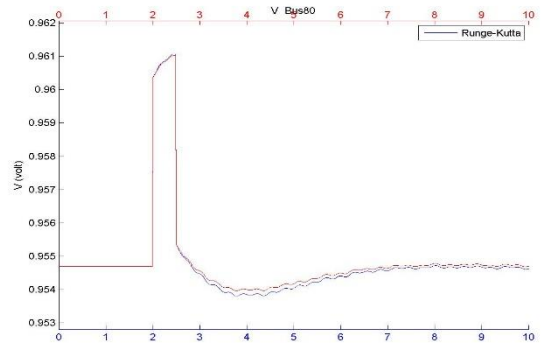

V Bus85
Runge-Kutta
V (volt)

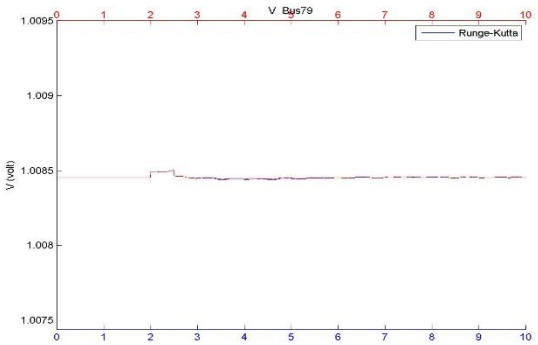

V Bus84
Runge-Kutta
V (volt)

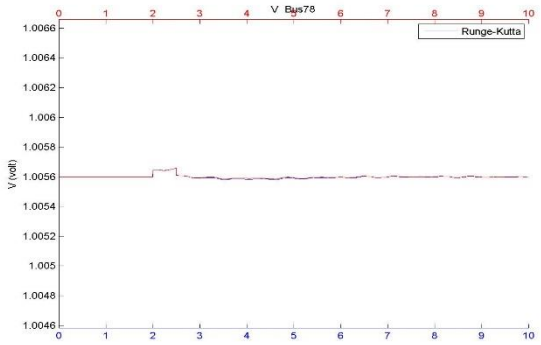

V Bus83
Runge-Kutta
V (volt)

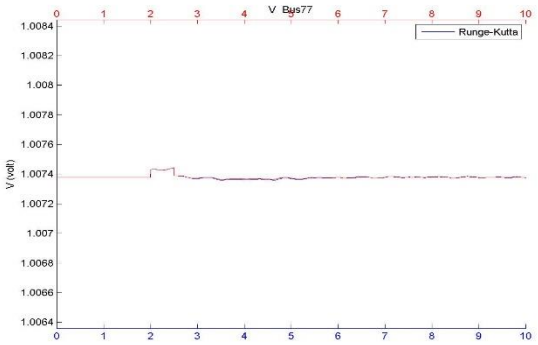

V Bus82
Runge-Kutta
V (volt)

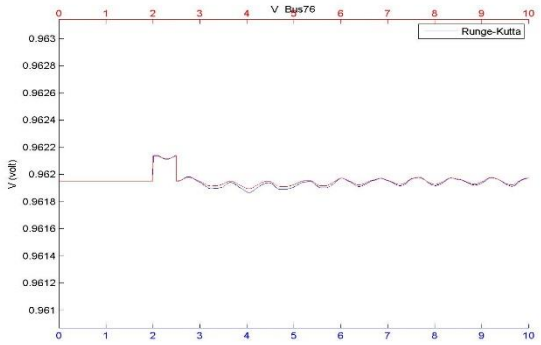

V Bus81
Runge-Kutta
V (volt)

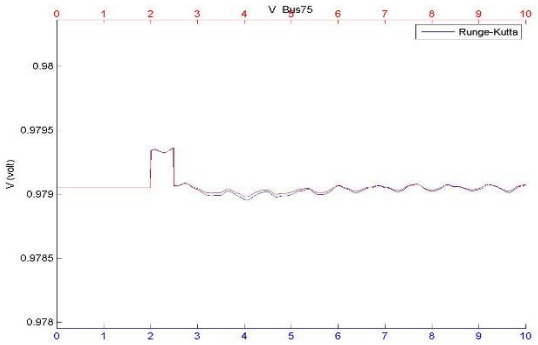

V Bus80
Runge-Kutta
V (volt)

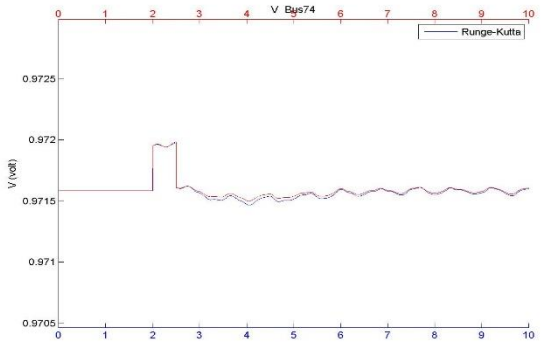

V Bus79
Runge-Kutta
V (volt)

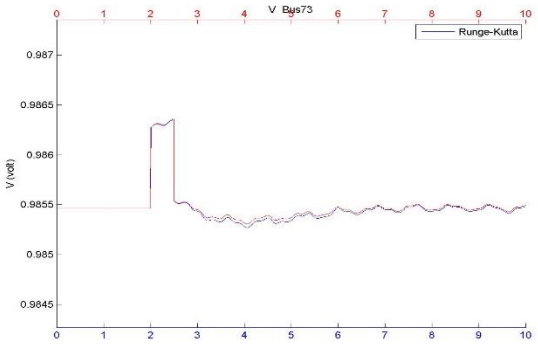

V Bus78
Runge-Kutta
V (volt)

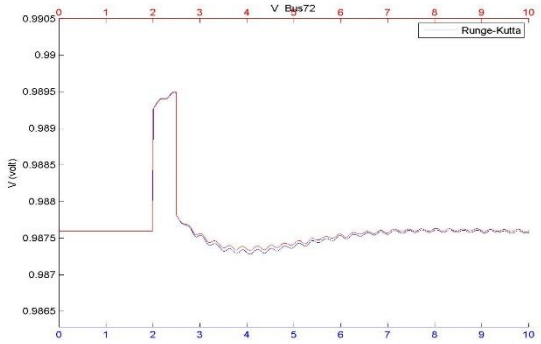

V Bus77
Runge-Kutta
V (volt)

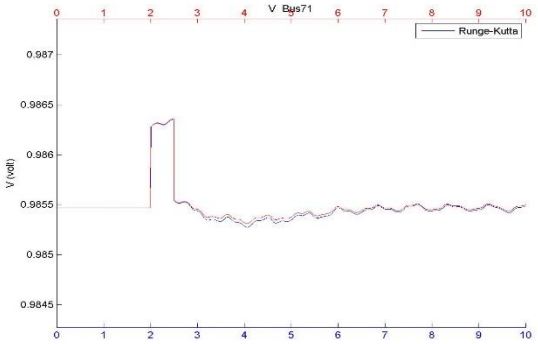

V Bus76
Runge-Kutta
V (volt)

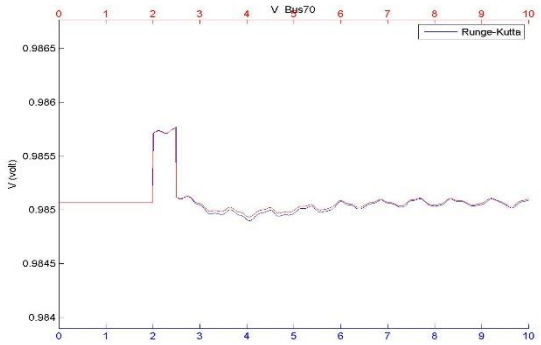

V Bus75
Runge-Kutta
V (volt)

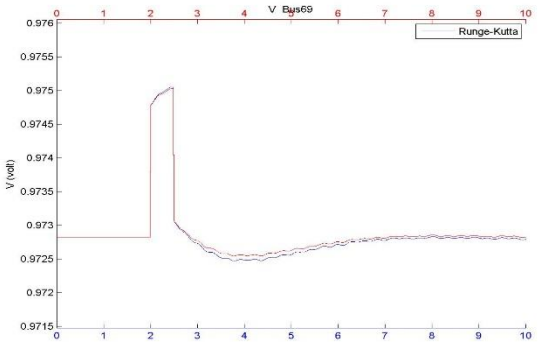

V Bus74
Runge-Kutta
V (volt)

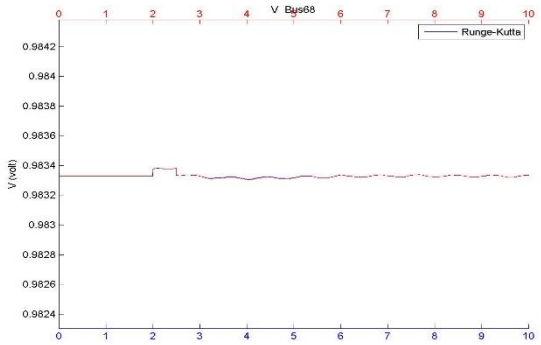

V Bus73
Runge-Kutta
V (volt)

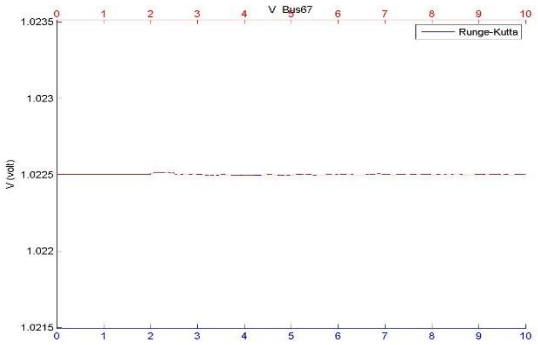

V Bus72
Runge-Kutta
V (volt)

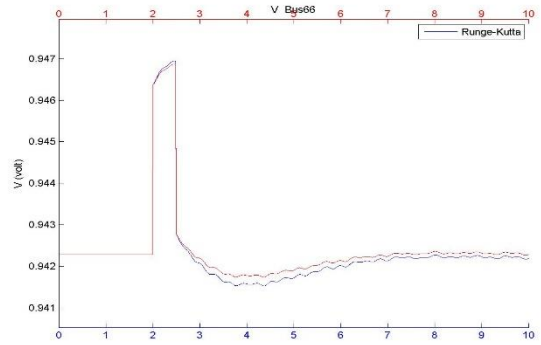

V Bus71
Runge-Kutta
V (volt)

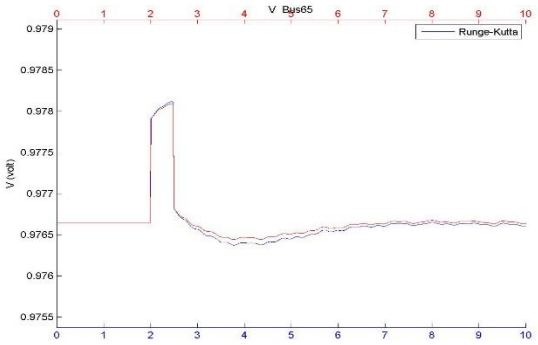

V Bus70
Runge-Kutta
V (volt)

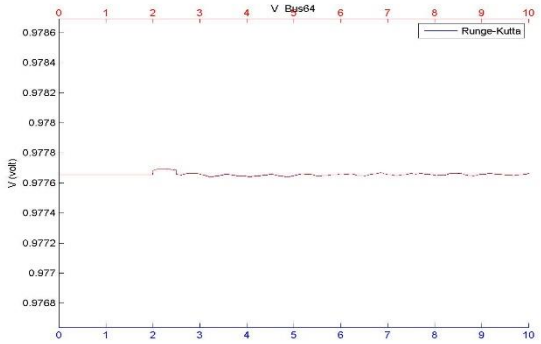

V Bus69
Runge-Kutta
V (volt)

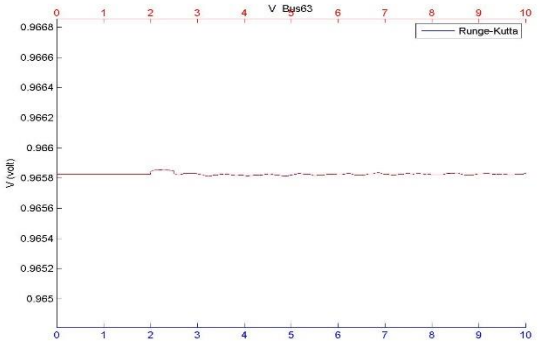

V Bus68
Runge-Kutta
V (volt)

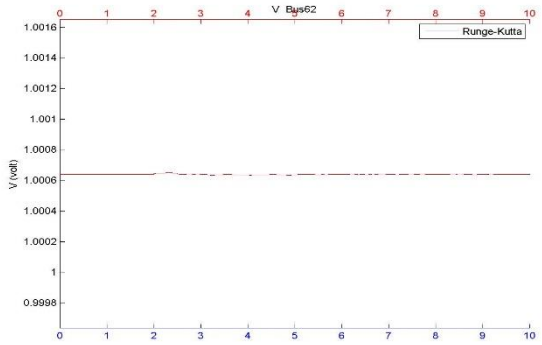

V Bus67
Runge-Kutta
V (volt)

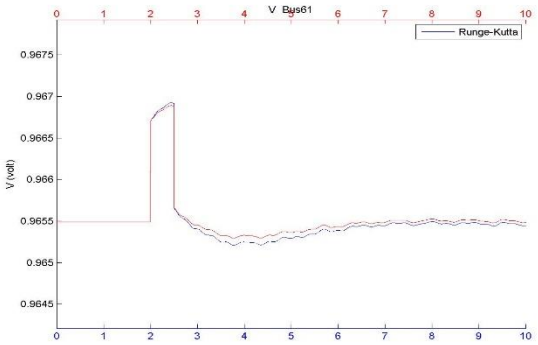

V Bus66
Runge-Kutta
V (volt)

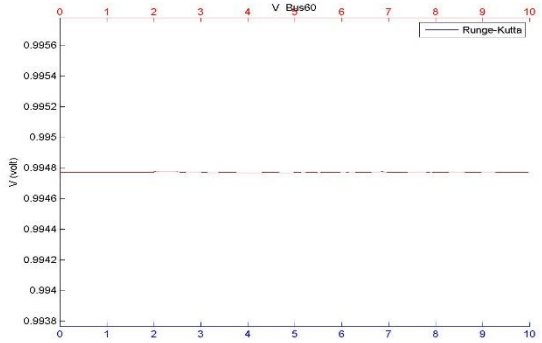

V Bus65
Runge-Kutta
V (volt)

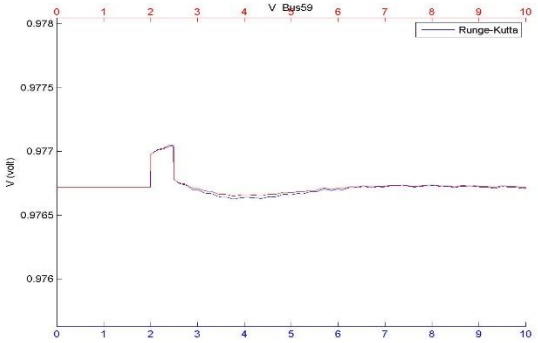

V Bus64
Runge-Kutta
V (volt)

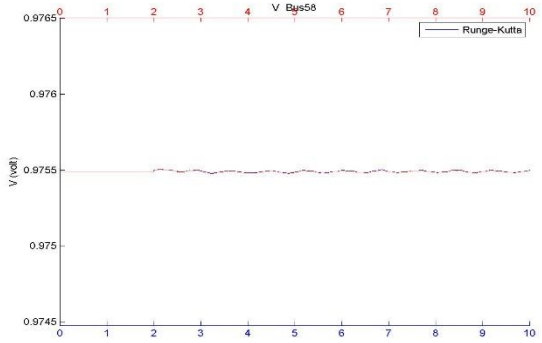

V Bus63
Runge-Kutta
V (volt)

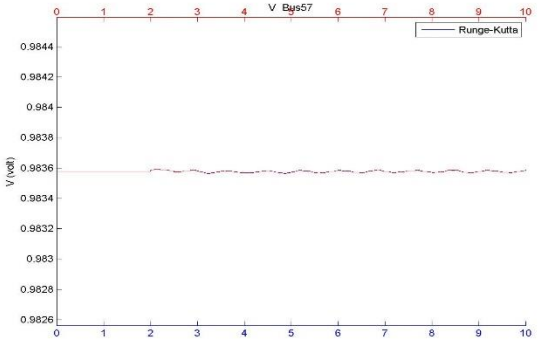

V Bus62
Runge-Kutta
V (volt)

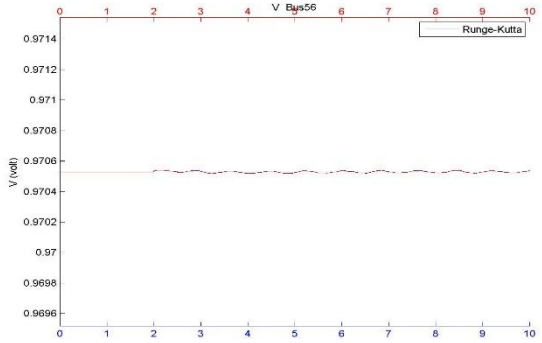

V Bus61
Runge-Kutta
V (volt)

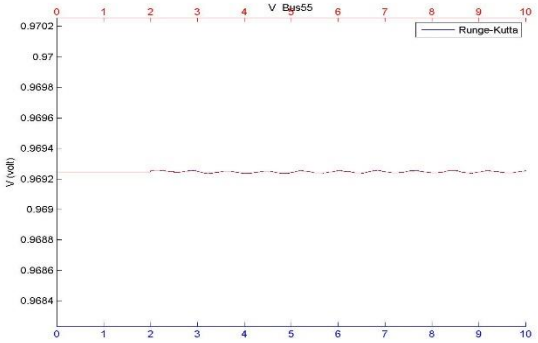

V Bus60
Runge-Kutta
V (volt)

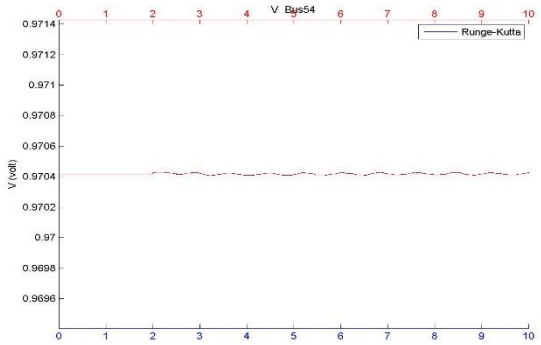

V Bus59
Runge-Kutta
V (volt)

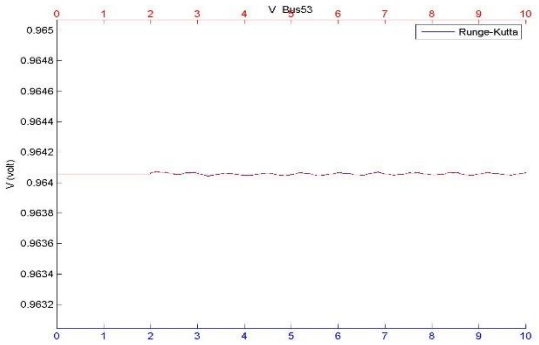

V Bus58
Runge-Kutta
V (volt)

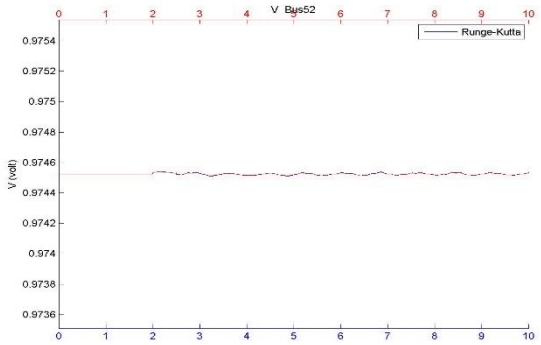

V Bus57
Runge-Kutta
V (volt)

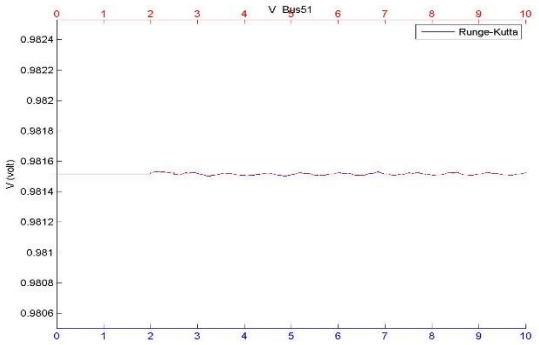

V Bus56
Runge-Kutta
V (volt)

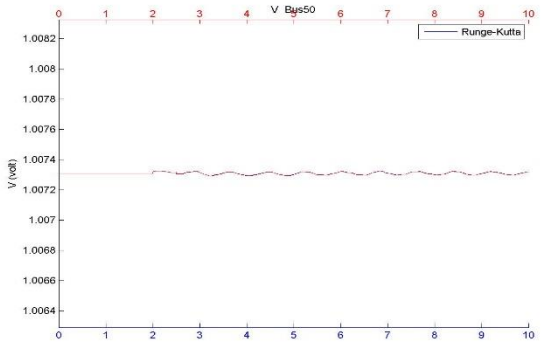

V Bus55
Runge-Kutta
V (volt)

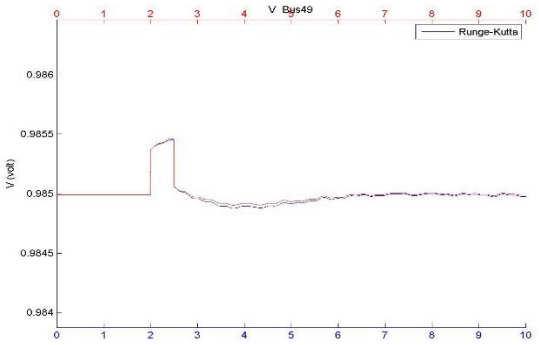

V Bus54
Runge-Kutta
V (volt)

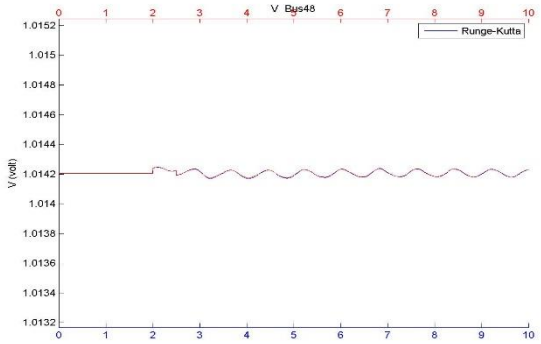

V Bus53
Runge-Kutta
V (volt)

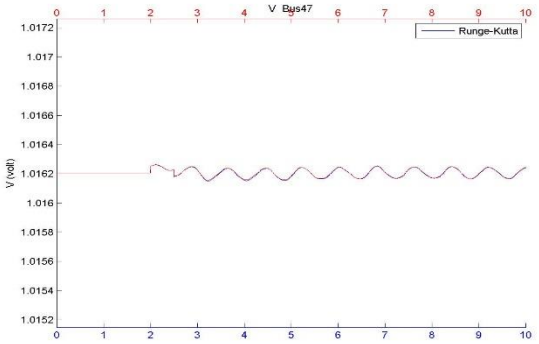

V Bus52
Runge-Kutta
V (volt)

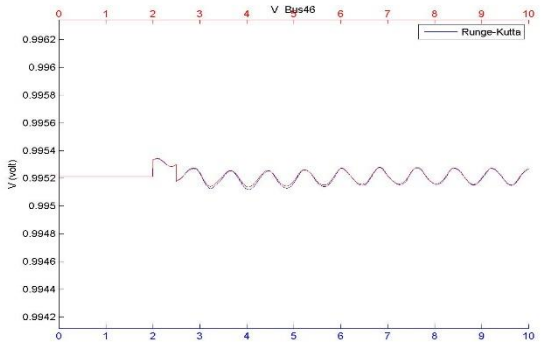

V Bus51
Runge-Kutta
V (volt)

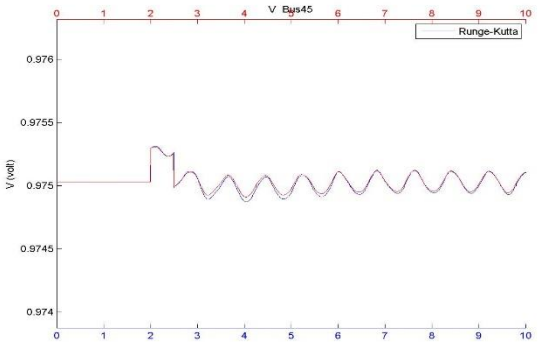

V Bus50
Runge-Kutta
V (volt)

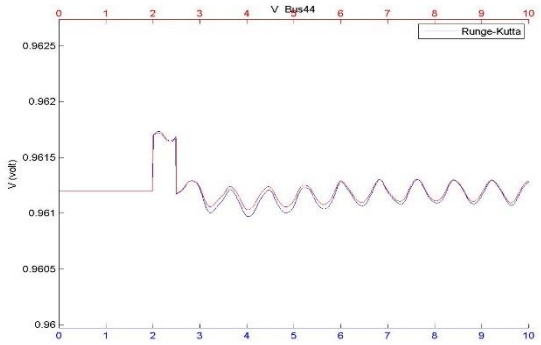

V Bus49
Runge-Kutta
V (volt)

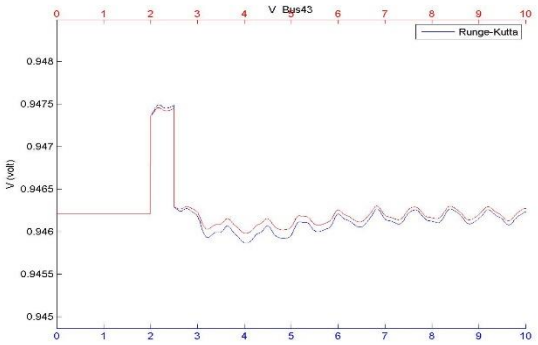

V Bus48
Runge-Kutta
V (volt)

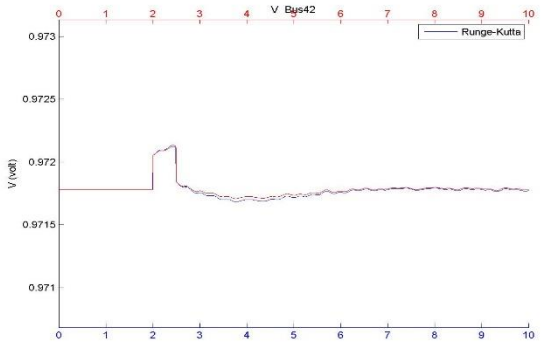

V Bus47
Runge-Kutta
V (volt)

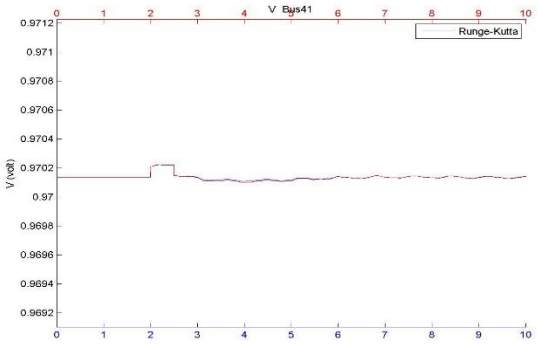

V Bus46
Runge-Kutta
V (volt)

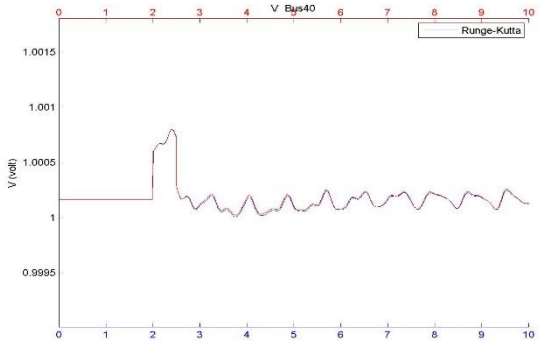

V Bus45
Runge-Kutta
V (volt)

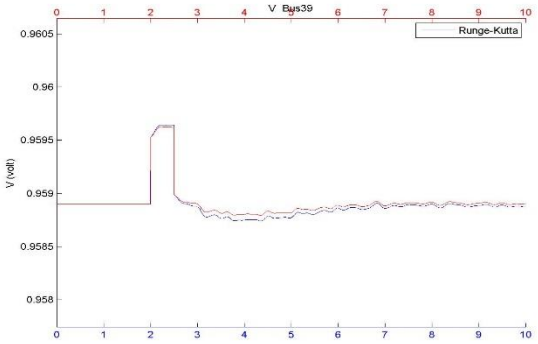

V Bus44
Runge-Kutta
V (volt)

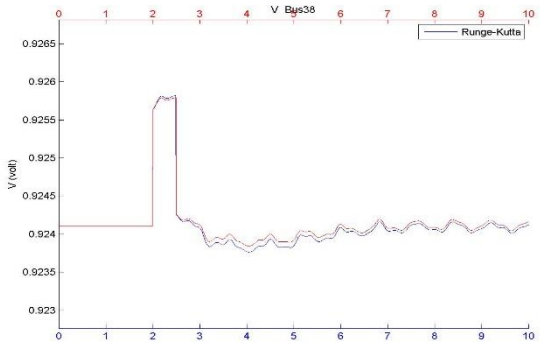

V Bus43
Runge-Kutta
V (volt)

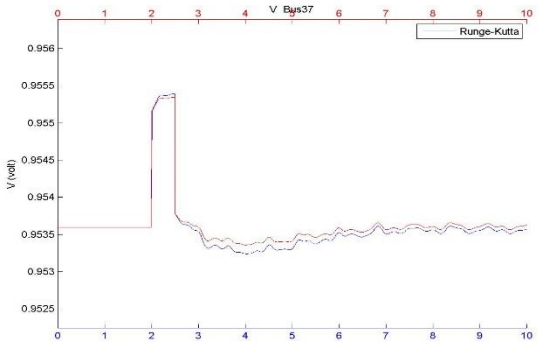

V Bus42
Runge-Kutta
V (volt)

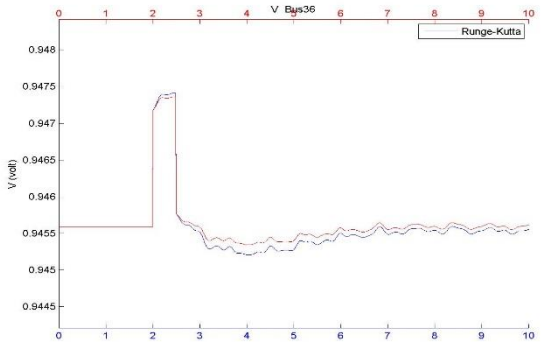

V Bus41
Runge-Kutta
V (volt)

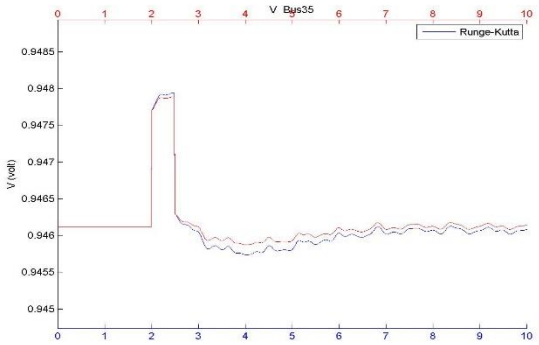

V Bus40
Runge-Kutta
V (volt)

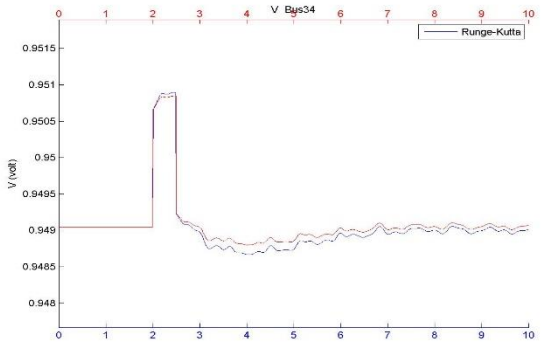

V Bus39
Runge-Kutta
V (volt)

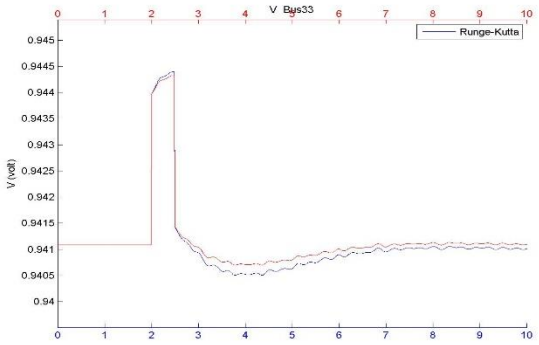

V Bus38
Runge-Kutta
V (volt)

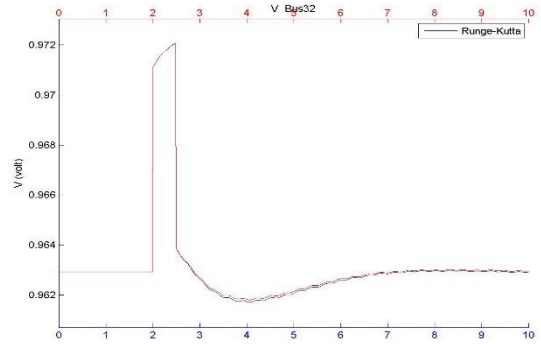

V Bus37
Runge-Kutta
V (volt)

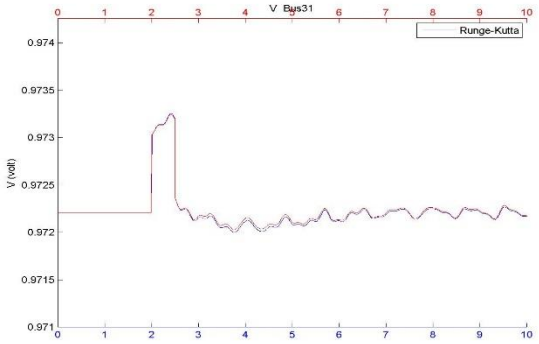

V Bus36
Runge-Kutta
V (volt)

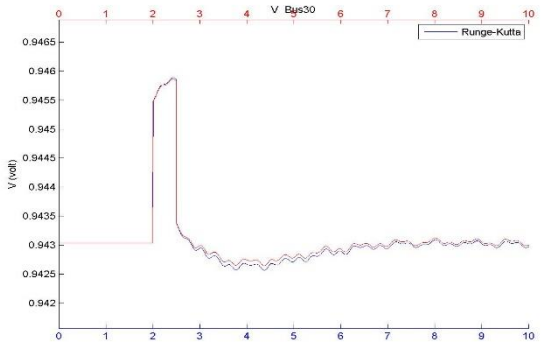

V Bus35
Runge-Kutta
V (volt)

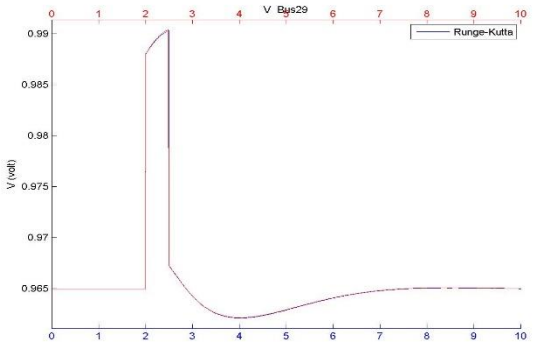

V Bus34
Runge-Kutta
V (volt)

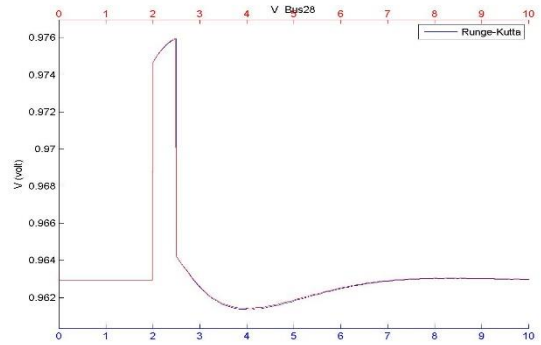

V Bus33
Runge-Kutta
V (volt)

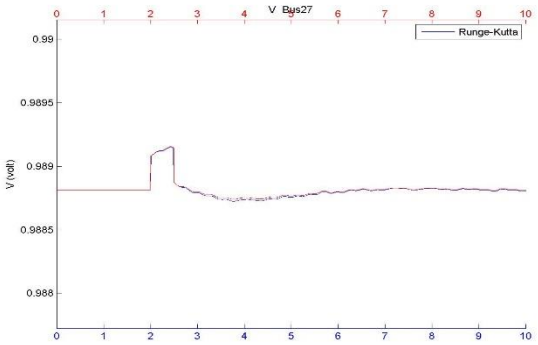

V Bus32
Runge-Kutta
V (volt)

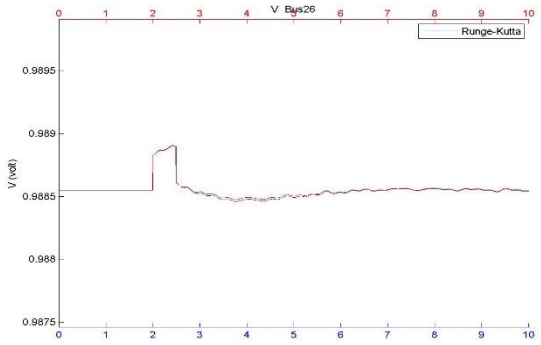

V Bus31
Runge-Kutta
V (volt)

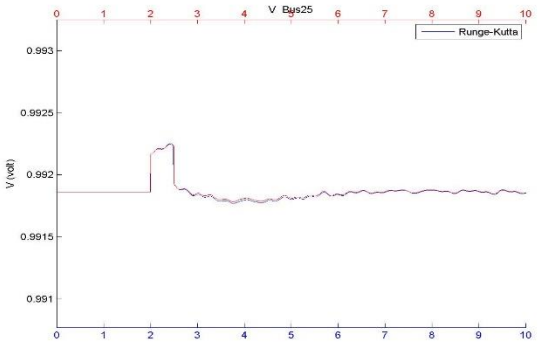

V_Bus30
Runge-Kutta
V (volt)

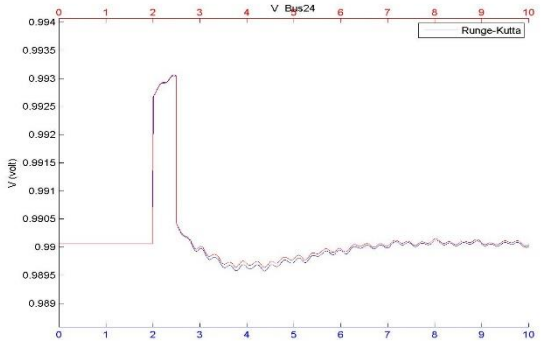

V_Bus29
Runge-Kutta
V (volt)

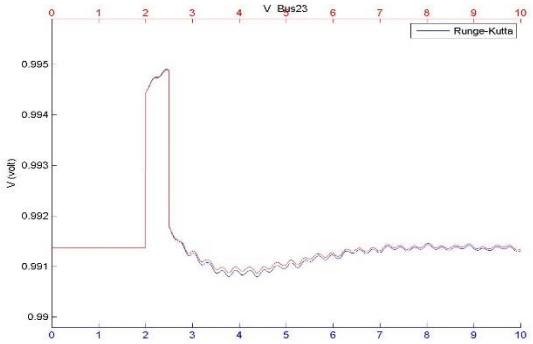

V_Bus28
Runge-Kutta
V (volt)

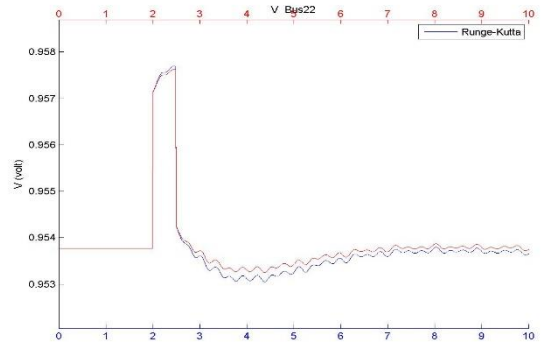

V_Bus27
Runge-Kutta
V (volt)

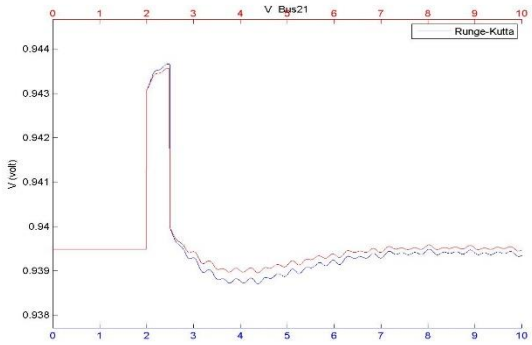

V_Bus26
Runge-Kutta
V (volt)

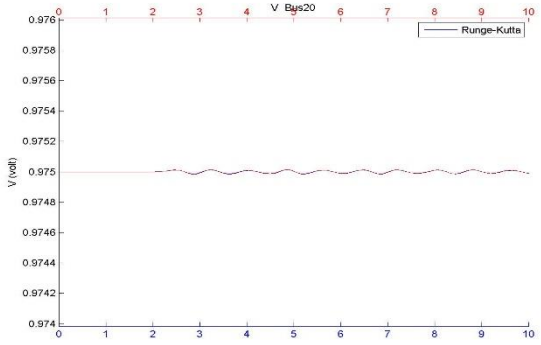

V_Bus25
Runge-Kutta
V (volt)

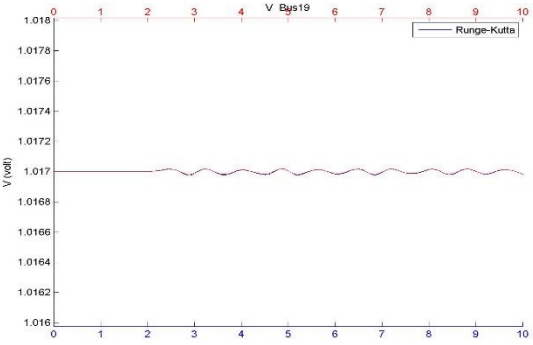

V_Bus24
Runge-Kutta
V (volt)

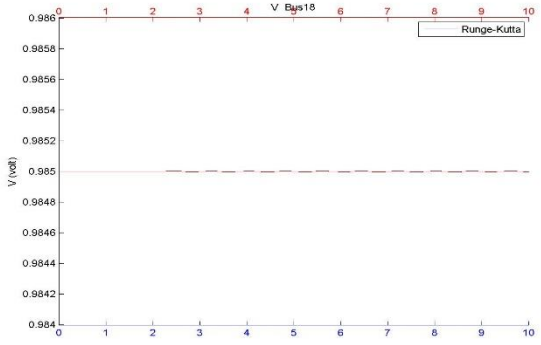

V_Bus23
Runge-Kutta
V (volt)

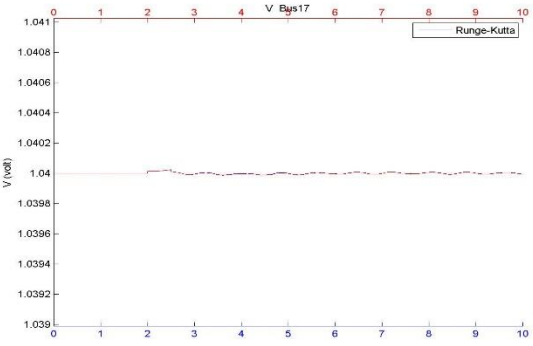

V_Bus22
Runge-Kutta
V (volt)

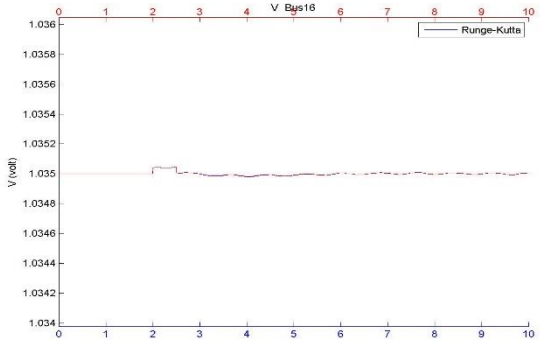

V_Bus21
Runge-Kutta
V (volt)

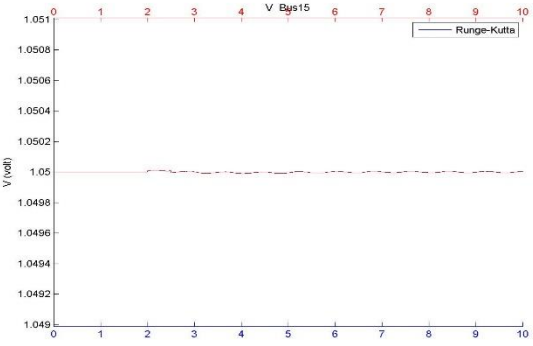

V_Bus20
Runge-Kutta
V (volt)

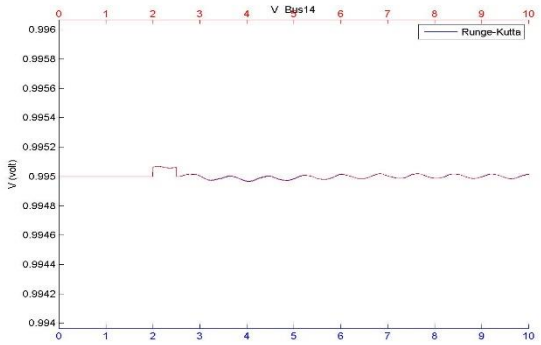

V_Bus19
Runge-Kutta
V (volt)

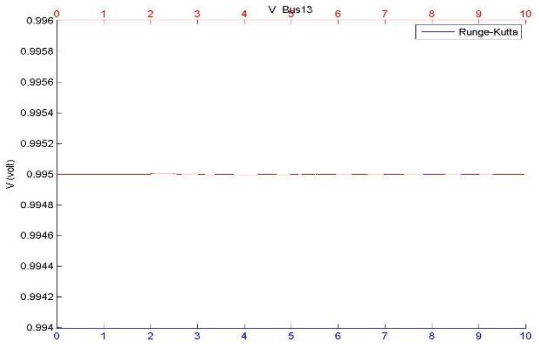

V_Bus18
Runge-Kutta
V (volt)

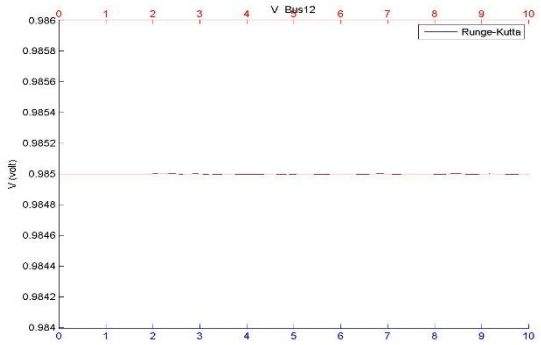

V_Bus17
Runge-Kutta
V (volt)

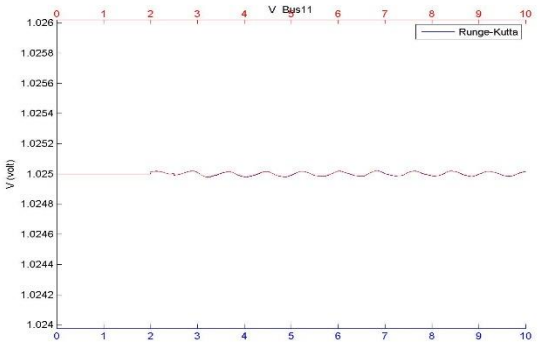

V_Bus16
Runge-Kutta
V (volt)

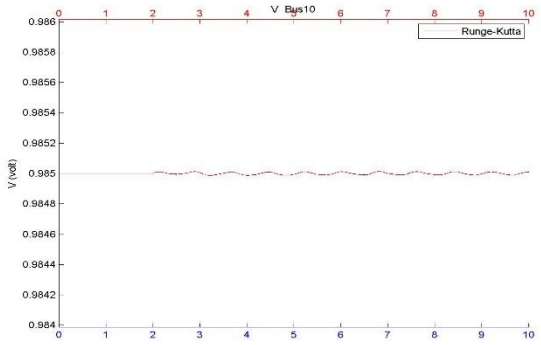

V_Bus15
Runge-Kutta
V (volt)

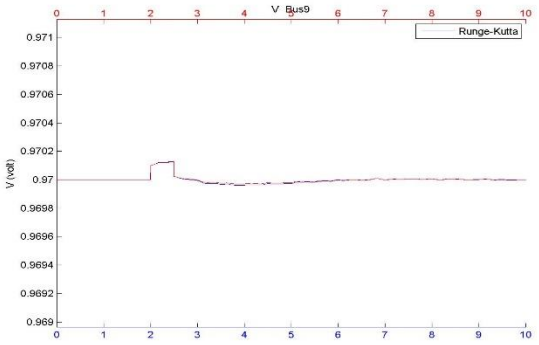

V Bus14
Runge-Kutta
V (volt)

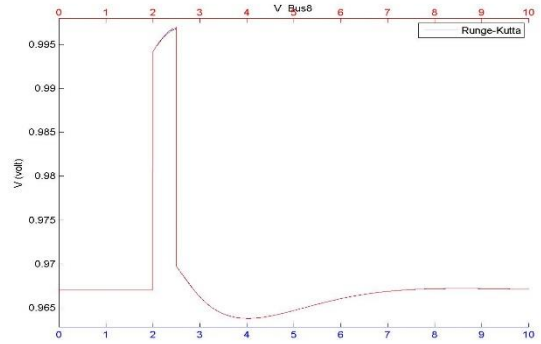

V Bus13
Runge-Kutta
V (volt)

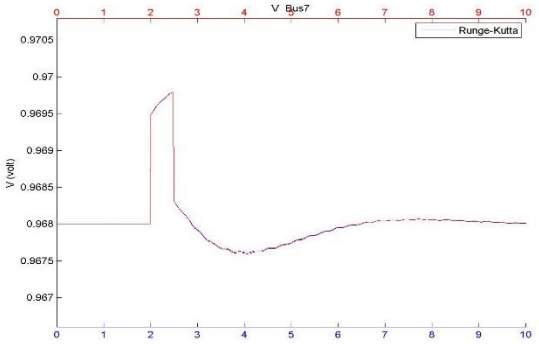

V Bus12
Runge-Kutta
V (volt)

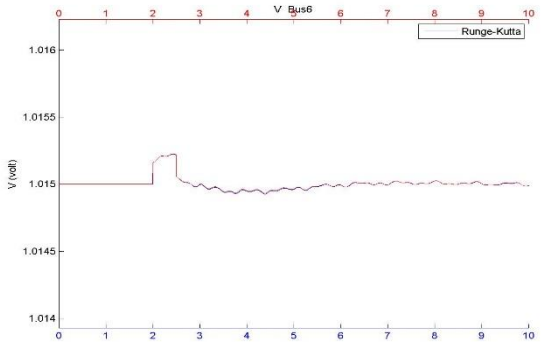

V Bus11
Runge-Kutta
V (volt)

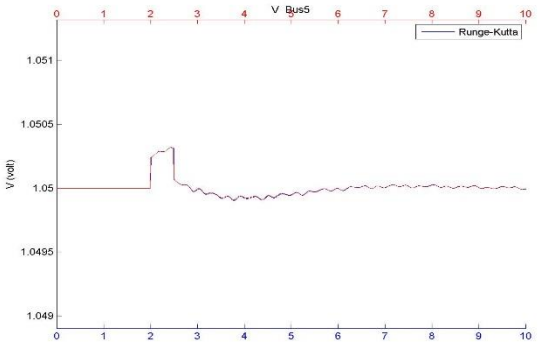

V Bus10
Runge-Kutta
V (volt)

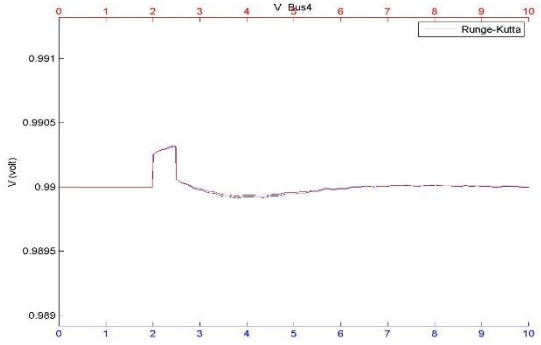

V Bus9
Runge-Kutta
V (volt)

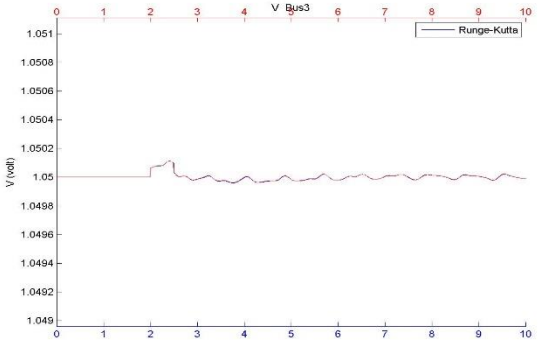

V Bus8
Runge-Kutta
V (volt)

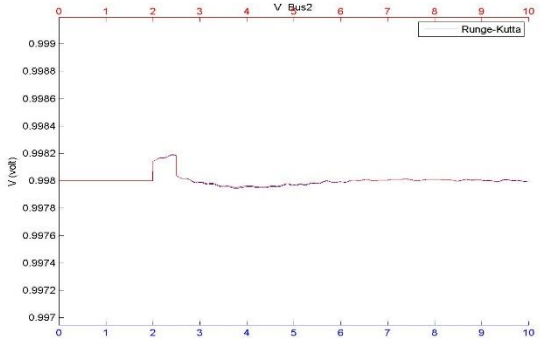

V Bus7
Runge-Kutta
V (volt)

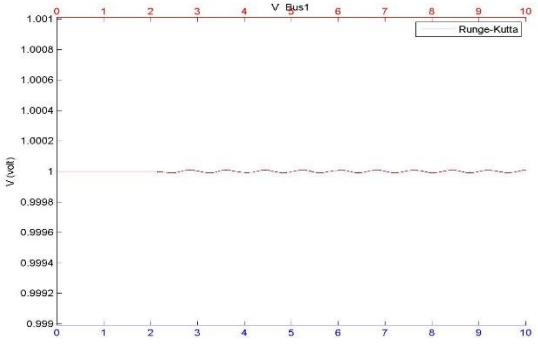

V Bus6
Runge-Kutta
V (volt)

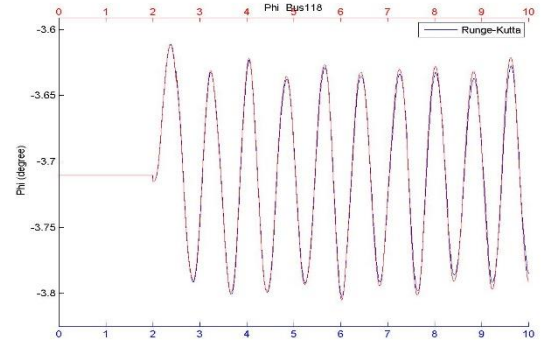

V Bus5
Runge-Kutta
V (volt)

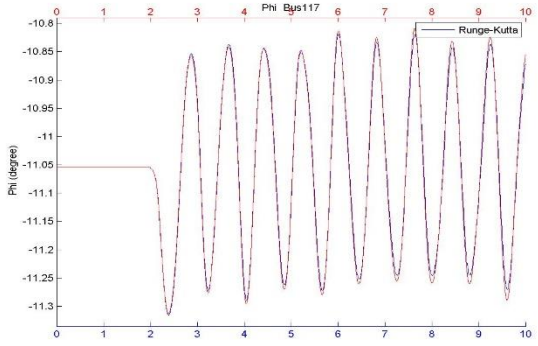

V Bus4
Runge-Kutta
V (volt)

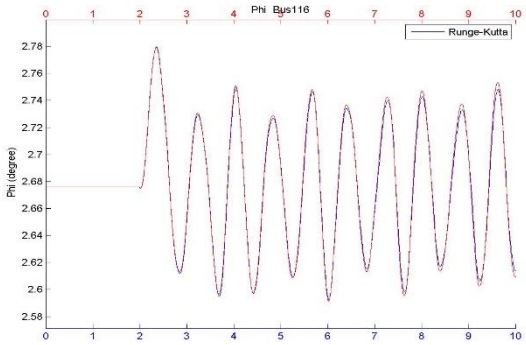

V Bus3
Runge-Kutta
V (volt)

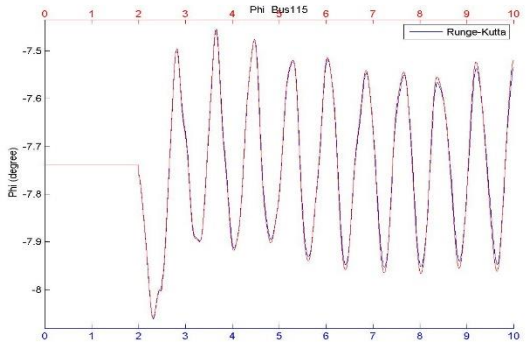

V Bus2
Runge-Kutta
V (volt)

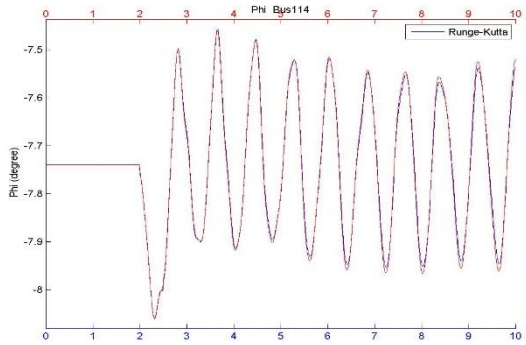

V Bus1
Runge-Kutta
V (volt)

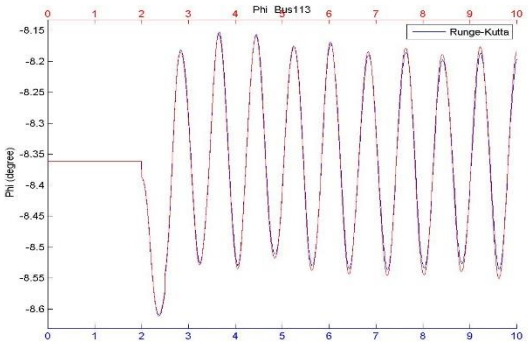

Phi Bus118
Runge-Kutta
Phi (degree)

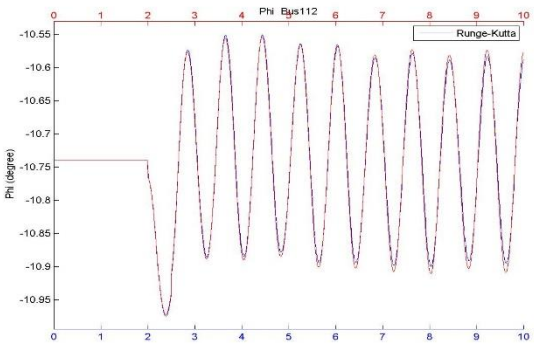

Phi Bus117
Runge-Kutta
Phi (degree)

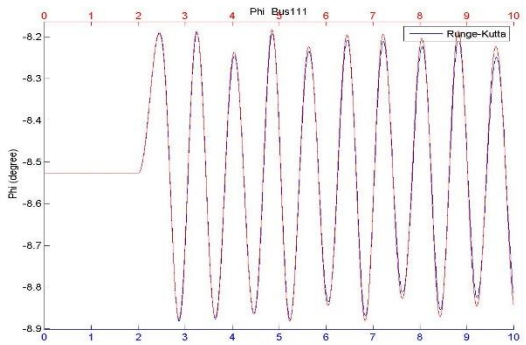

Phi Bus116
Runge-Kutta
Phi (degree)

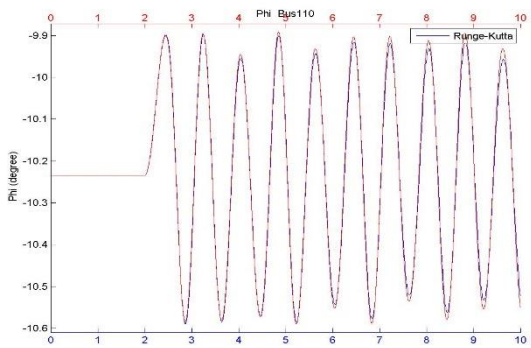

Phi Bus115
Runge-Kutta
Phi (degree)

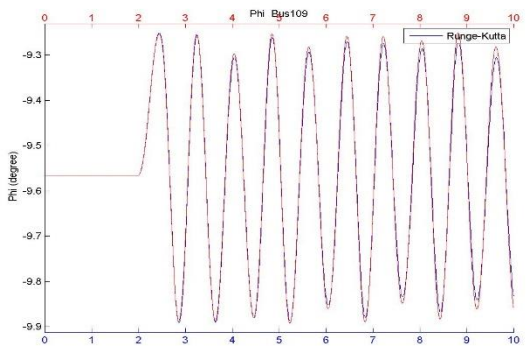

Phi Bus114
Runge-Kutta
Phi (degree)

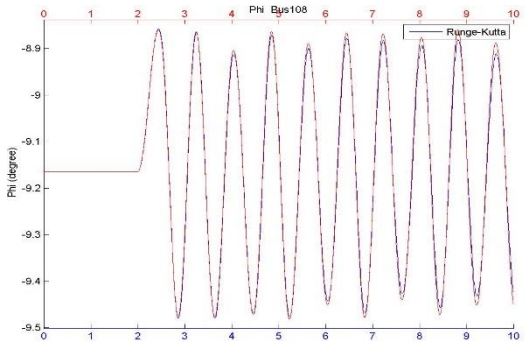

Phi Bus113
Runge-Kutta
Phi (degree)

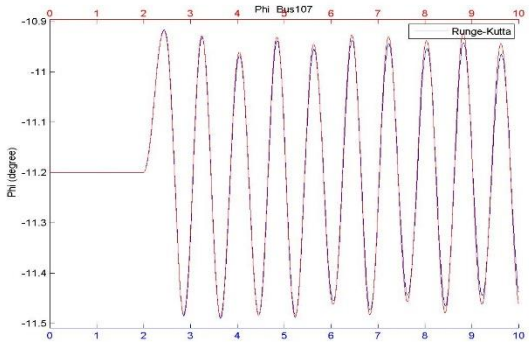

Phi Bus112
Runge-Kutta
Phi (degree)

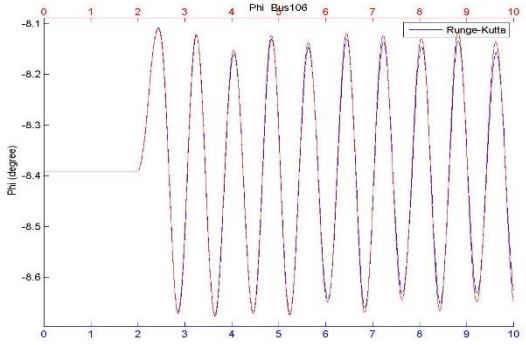

Phi Bus111
Runge-Kutta
Phi (degree)

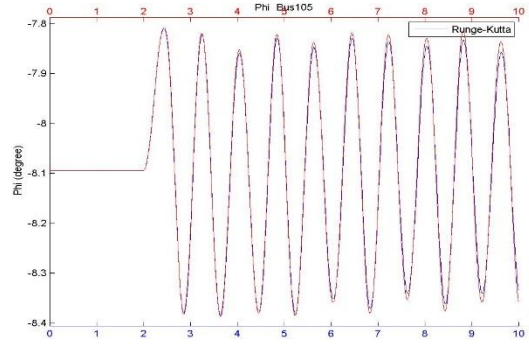

Phi Bus110
Runge-Kutta
Phi (degree)

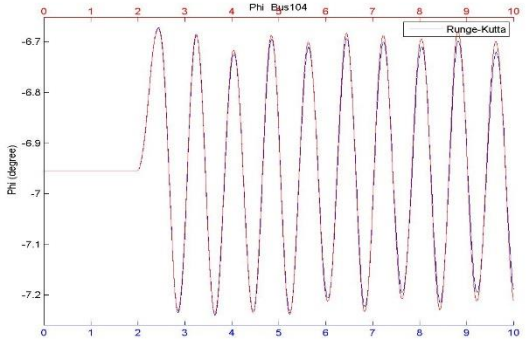

Phi Bus109
Runge-Kutta
Phi (degree)

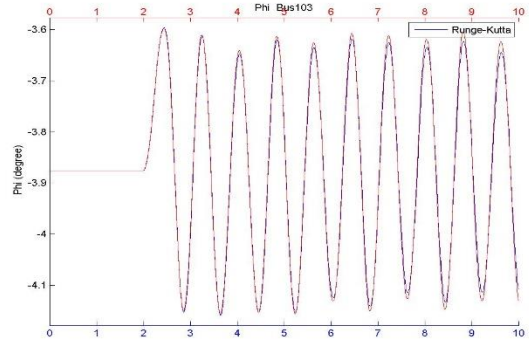

Phi Bus108
Runge-Kutta
Phi (degree)

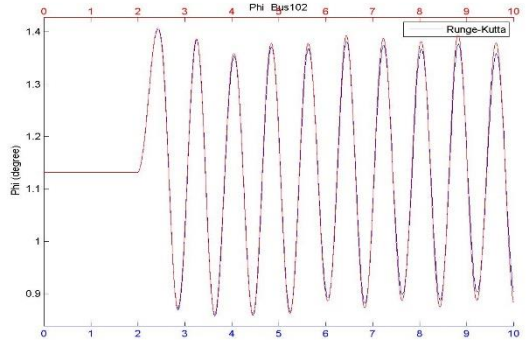

Phi Bus107
Runge-Kutta
Phi (degree)

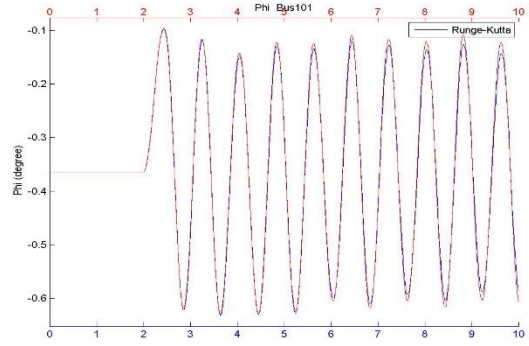

Phi Bus106
Runge-Kutta
Phi (degree)

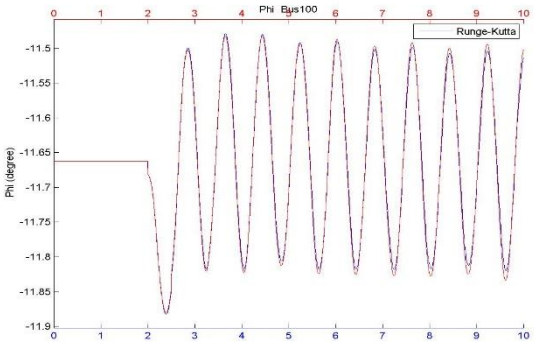

Phi Bus105
Runge-Kutta
Phi (degree)

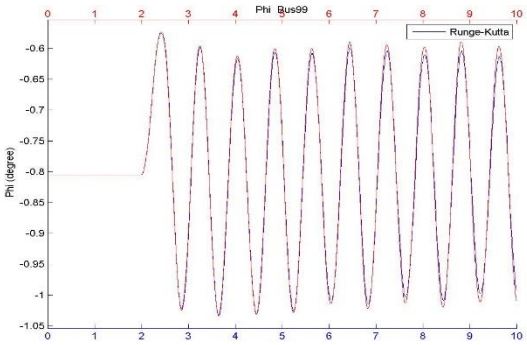

Phi Bus104
Runge-Kutta
Phi (degree)

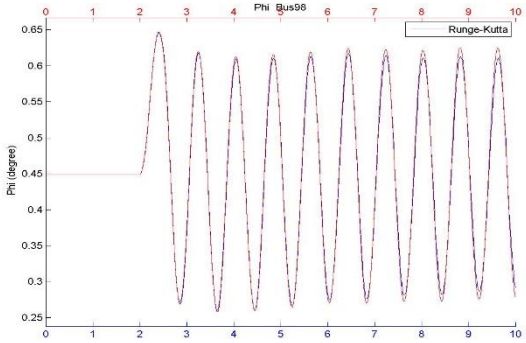

Phi Bus103
Runge-Kutta
Phi (degree)

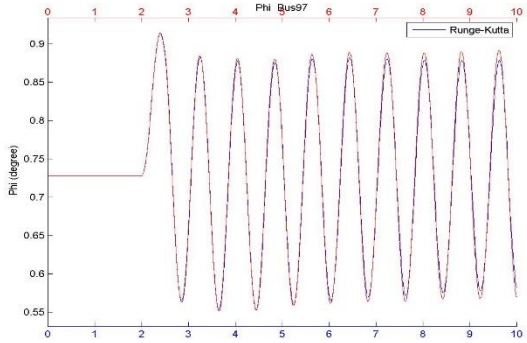

Phi Bus102
Runge-Kutta
Phi (degree)

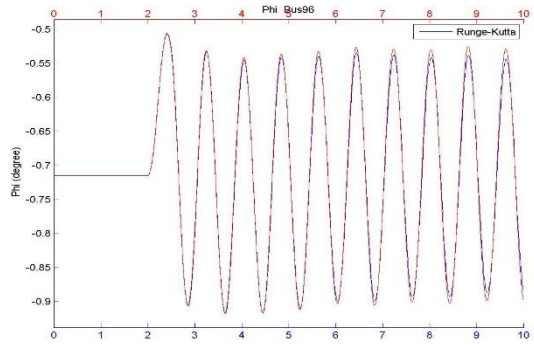

Phi Bus101
Runge-Kutta
Phi (degree)

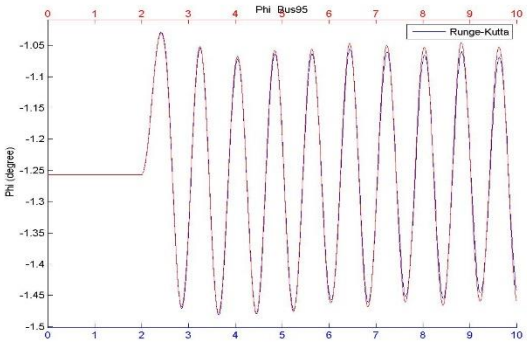

Phi Bus100
Runge-Kutta
Phi (degree)

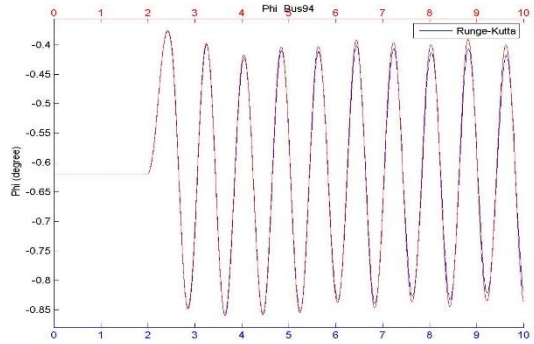

Phi Bus99
Runge-Kutta
Phi (degree)

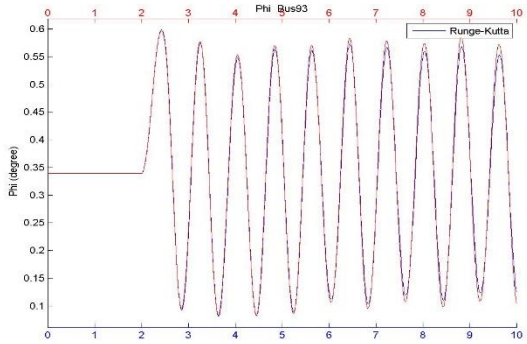

Phi Bus98
Runge-Kutta
Phi (degree)

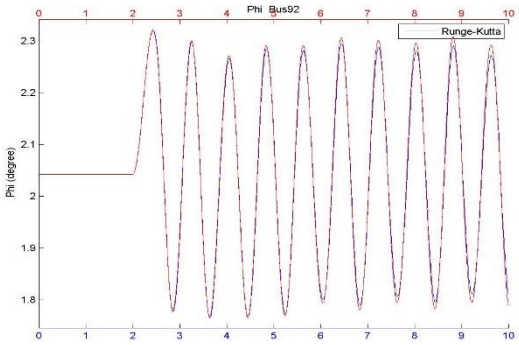

Phi Bus97
Runge-Kutta
Phi (degree)

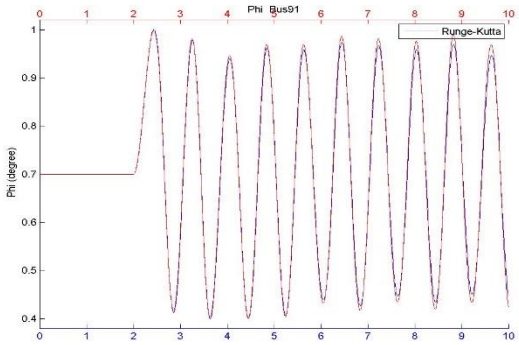

Phi Bus96
Runge-Kutta
Phi (degree)

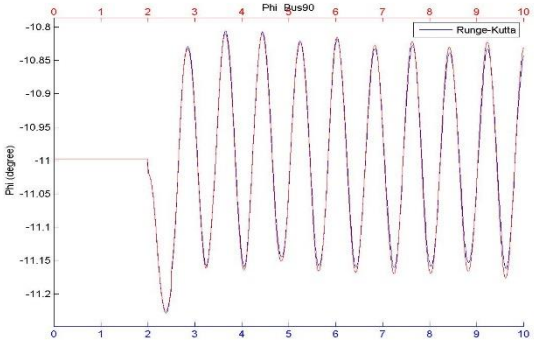

Phi Bus95
Runge-Kutta
Phi (degree)

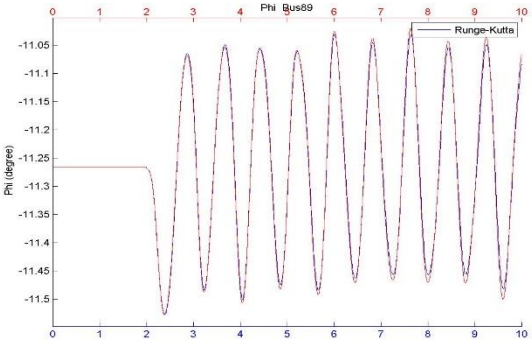

Phi Bus94
Runge-Kutta
Phi (degree)

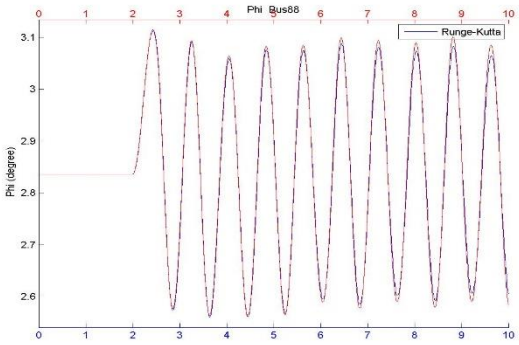

Phi Bus93
Runge-Kutta
Phi (degree)

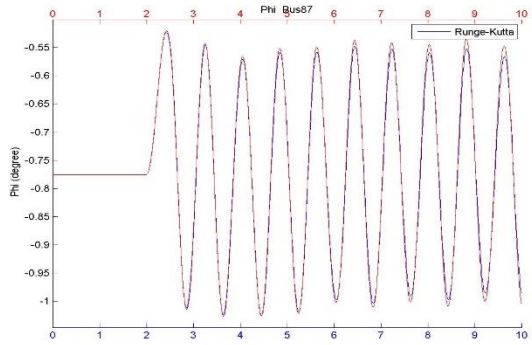

Phi Bus92
Runge-Kutta
Phi (degree)

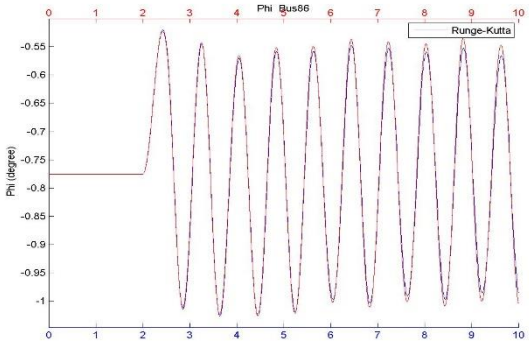

Phi Bus91
Runge-Kutta
Phi (degree)

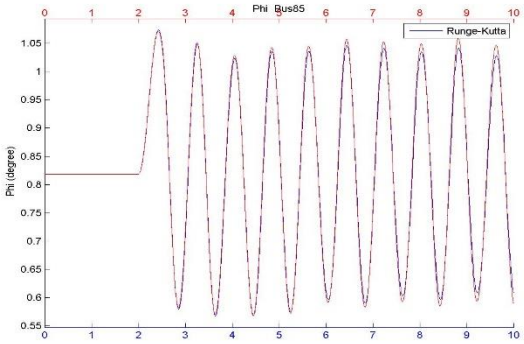

Phi Bus90
Runge-Kutta
Phi (degree)

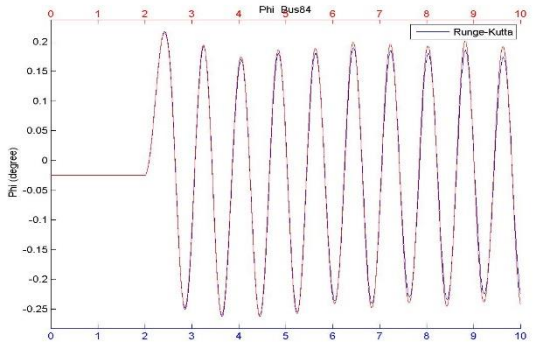

Phi Bus89
Runge-Kutta
Phi (degree)

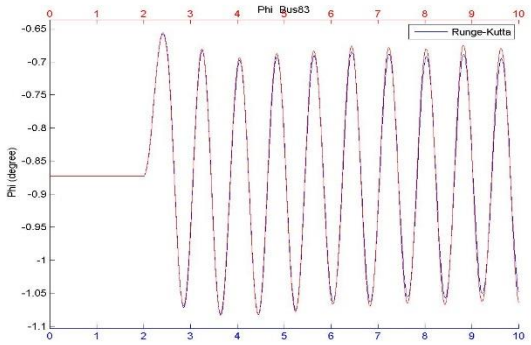

Phi Bus88
Runge-Kutta
Phi (degree)

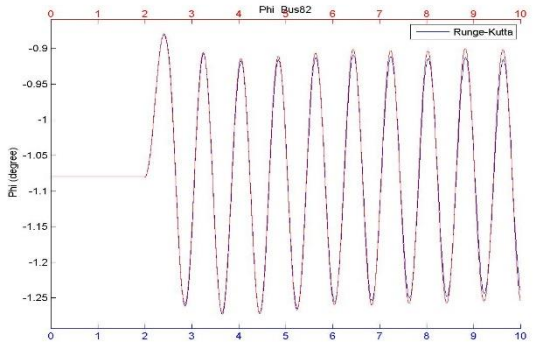

Phi Bus87
Runge-Kutta
Phi (degree)

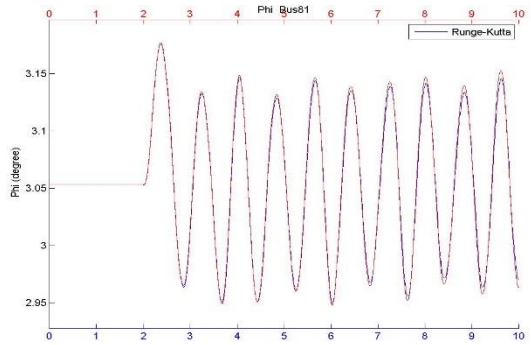

Phi Bus86
Runge-Kutta
Phi (degree)

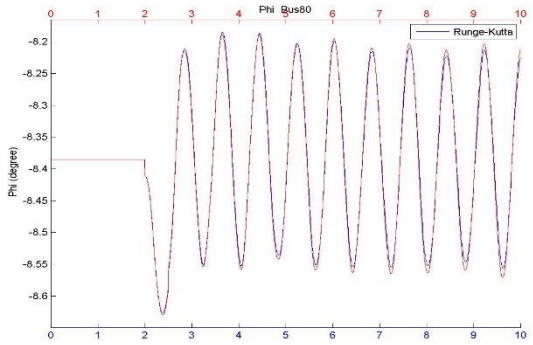

Phi Bus85
Runge-Kutta
Phi (degree)

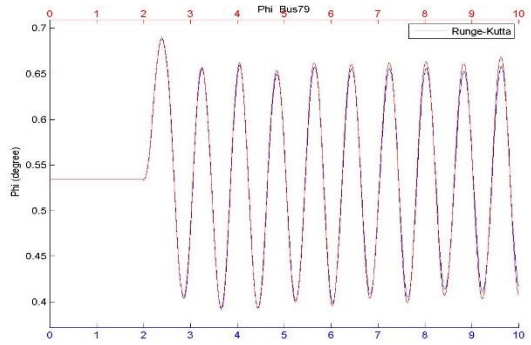

Phi Bus84
Runge-Kutta
Phi (degree)

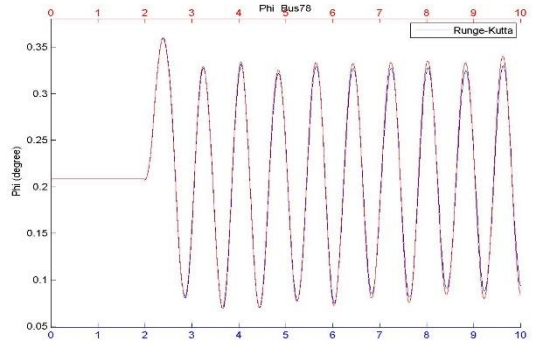

Phi Bus83
Runge-Kutta
Phi (degree)

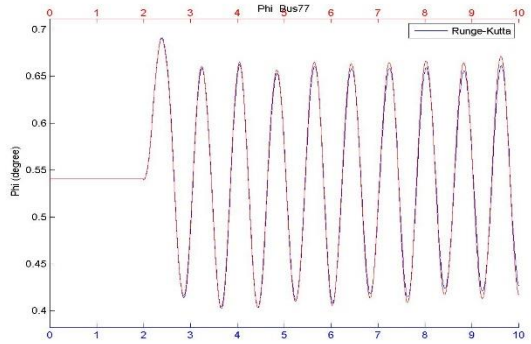

Phi Bus82
Runge-Kutta
Phi (degree)

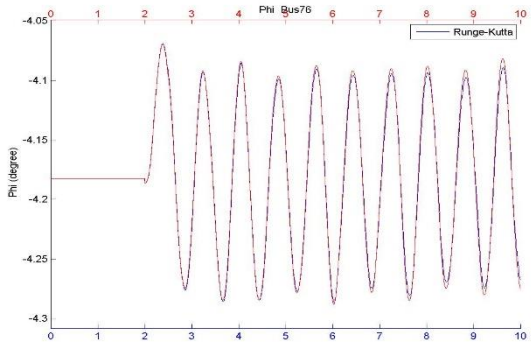

Phi Bus81
Runge-Kutta
Phi (degree)

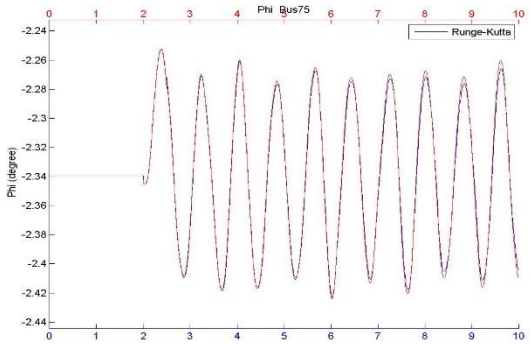

Phi Bus80
Runge-Kutta
Phi (degree)

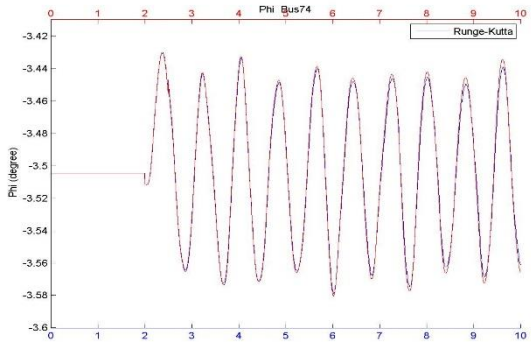

Phi Bus79
Runge-Kutta
Phi (degree)

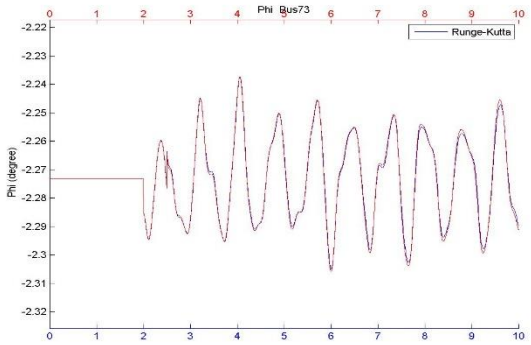

Phi Bus78
Runge-Kutta
Phi (degree)

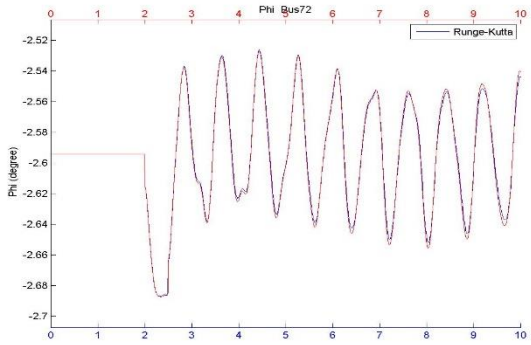

Phi Bus77
Runge-Kutta
Phi (degree)

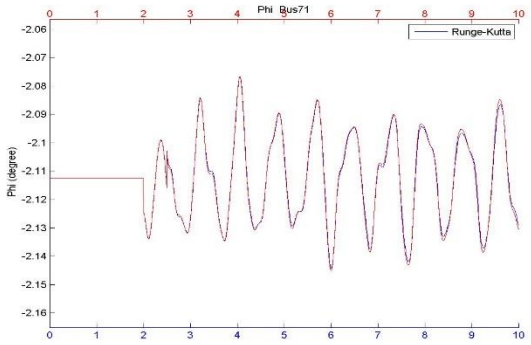

Phi Bus76
Runge-Kutta
Phi (degree)

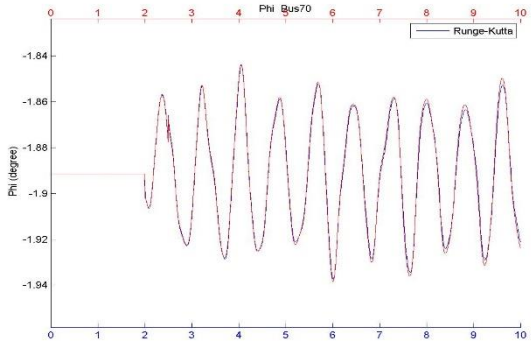

Phi Bus75
Runge-Kutta
Phi (degree)

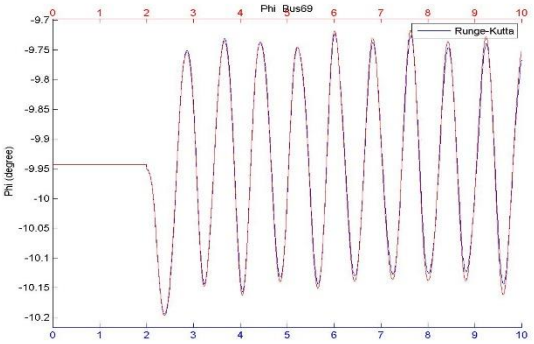

Phi Bus74
Runge-Kutta
Phi (degree)

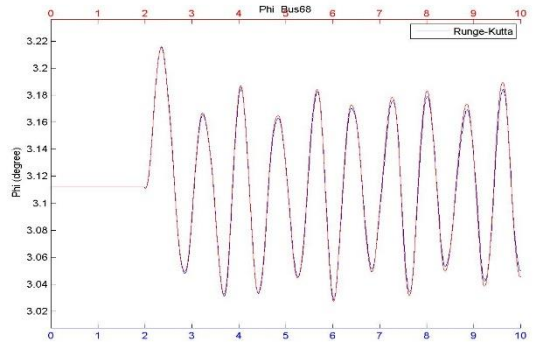

Phi Bus73
Runge-Kutta
Phi (degree)

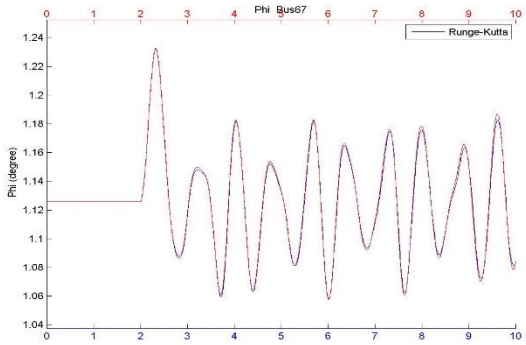

Phi Bus72
Runge-Kutta
Phi (degree)

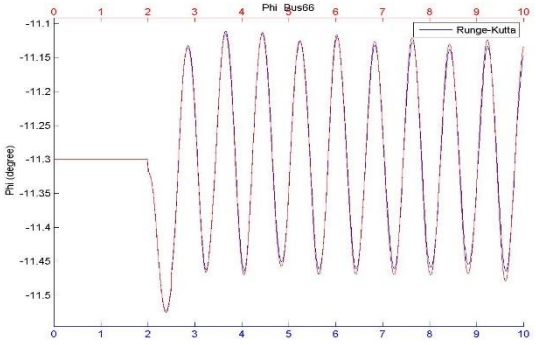

Phi Bus71
Runge-Kutta
Phi (degree)

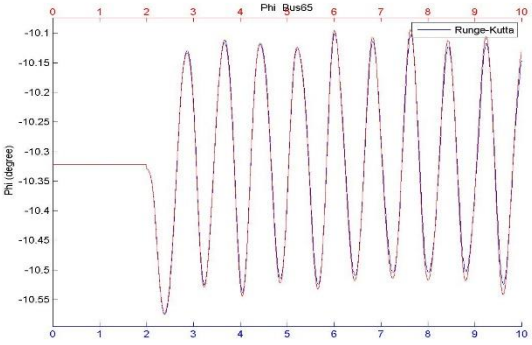

Phi Bus70
Runge-Kutta
Phi (degree)

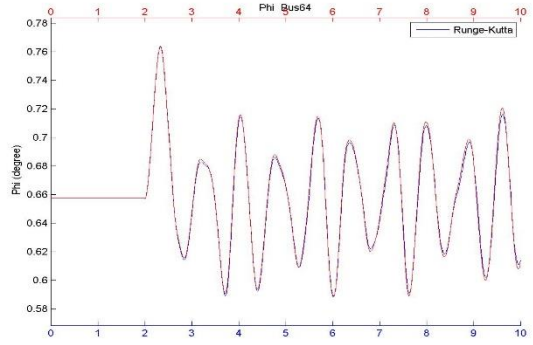

Phi Bus69
Runge-Kutta
Phi (degree)

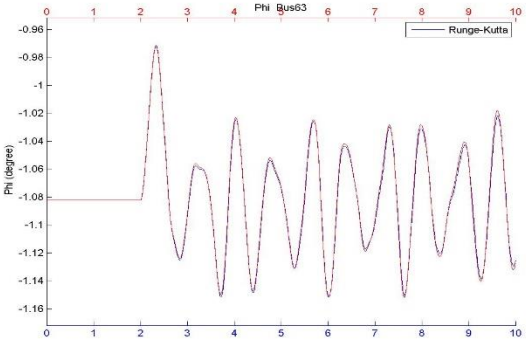

Phi Bus68
Runge-Kutta
Phi (degree)

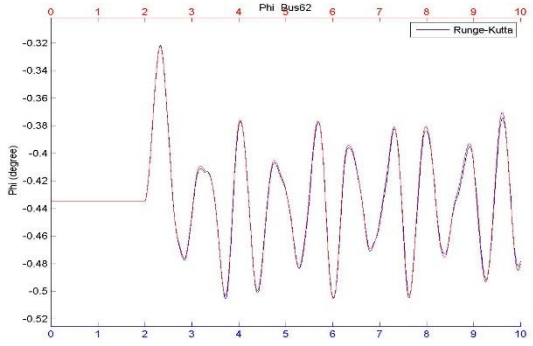

Phi Bus67
Runge-Kutta
Phi (degree)

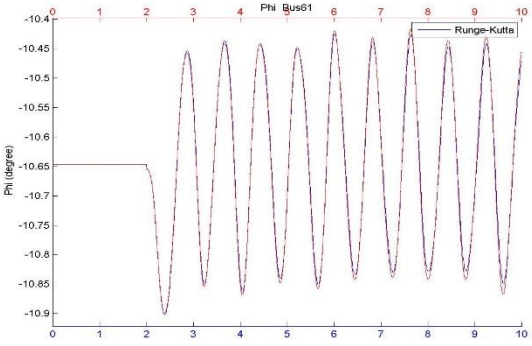

Phi Bus66
Runge-Kutta
Phi (degree)

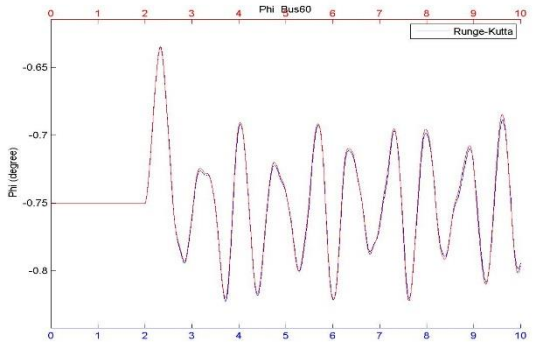

Phi Bus65
Runge-Kutta
Phi (degree)

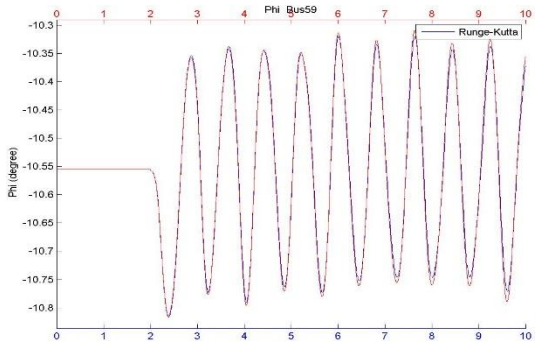

Phi Bus64
Runge-Kutta
Phi (degree)

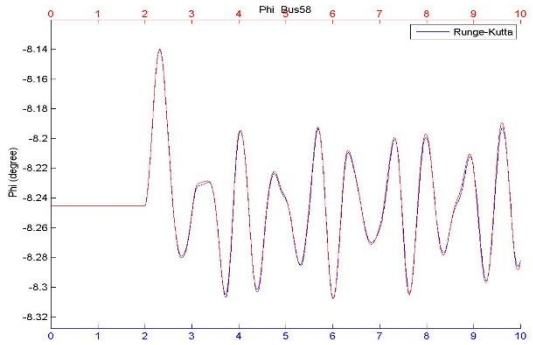

Phi Bus63
Runge-Kutta
Phi (degree)

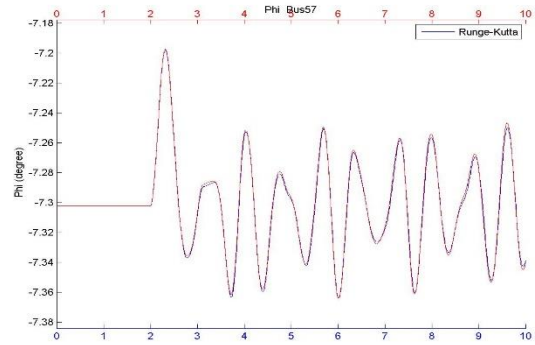

Phi Bus62
Runge-Kutta
Phi (degree)

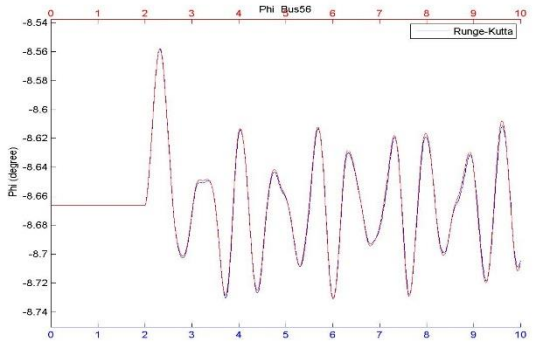

Phi Bus61
Runge-Kutta
Phi (degree)

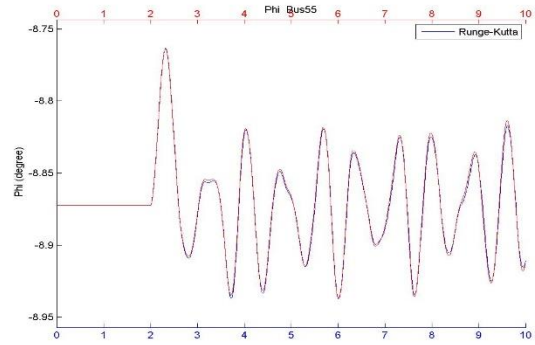

Phi Bus60
Runge-Kutta
Phi (degree)

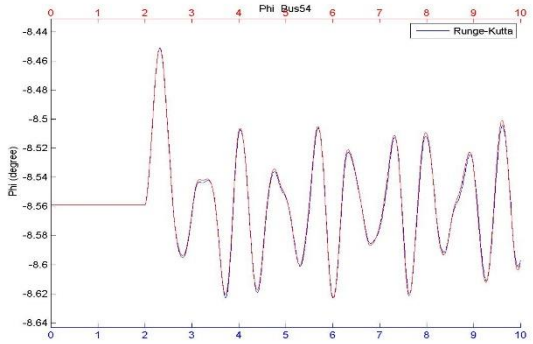

Phi Bus59
Runge-Kutta
Phi (degree)

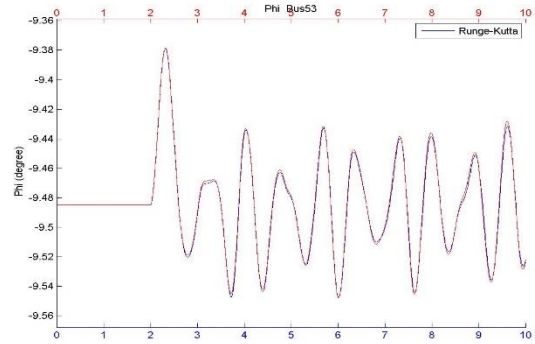

Phi Bus58
Runge-Kutta
Phi (degree)

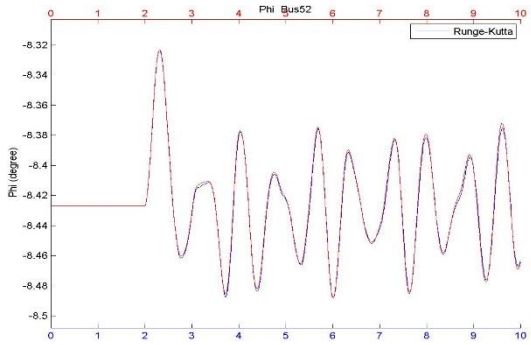

Phi Bus57
Runge-Kutta
Phi (degree)

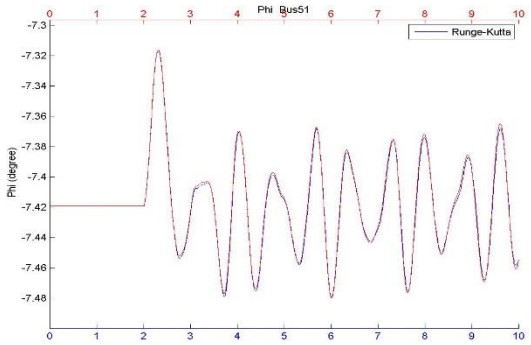

Phi Bus56
Runge-Kutta
Phi (degree)

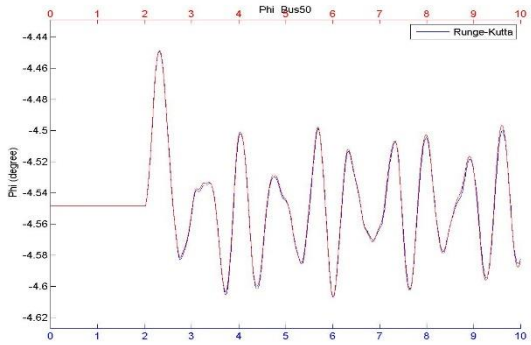

Phi Bus55
Runge-Kutta
Phi (degree)

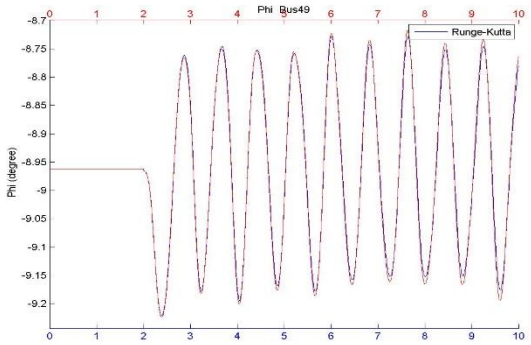

Phi Bus54
Runge-Kutta
Phi (degree)

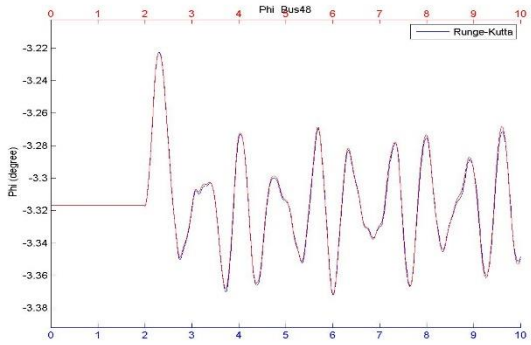

Phi Bus53
Runge-Kutta
Phi (degree)

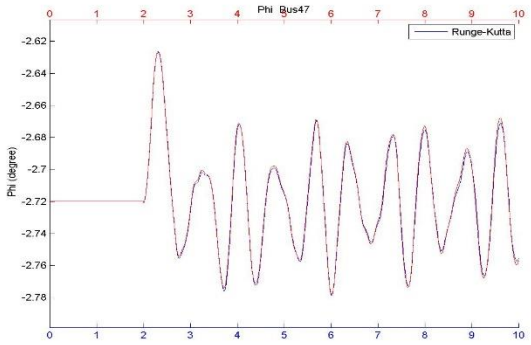

Phi Bus52
Runge-Kutta
Phi (degree)

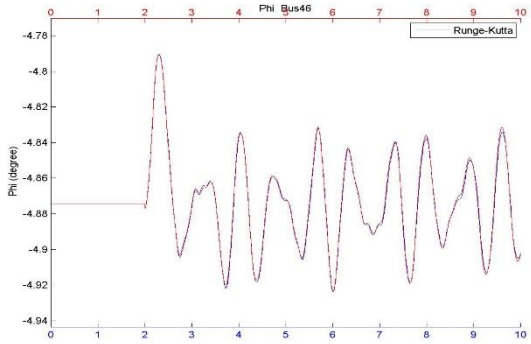

Phi Bus51
Runge-Kutta
Phi (degree)

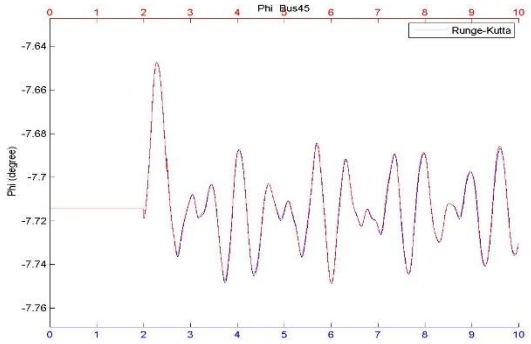

Phi Bus50
Runge-Kutta
Phi (degree)

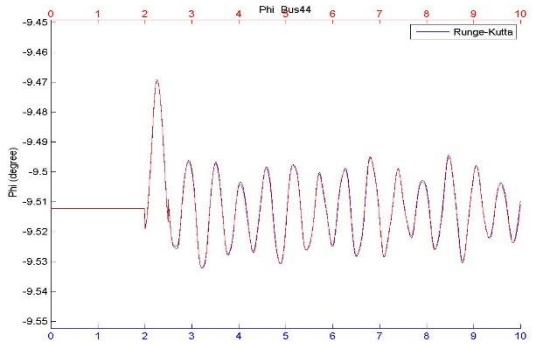

Phi Bus49
Runge-Kutta
Phi (degree)

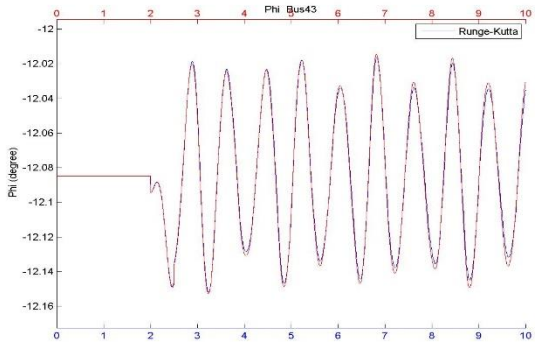

Phi Bus48
Runge-Kutta
Phi (degree)

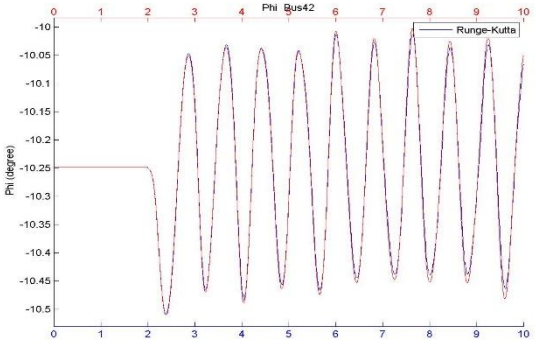

Phi Bus47
Runge-Kutta
Phi (degree)

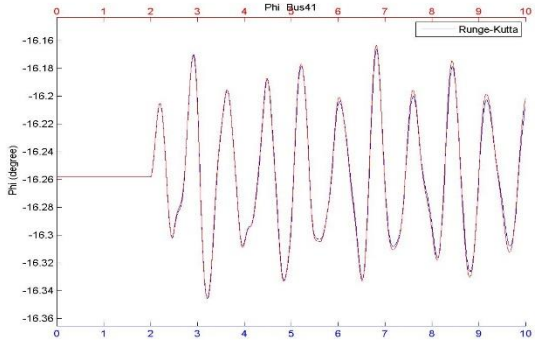

Phi Bus46
Runge-Kutta
Phi (degree)

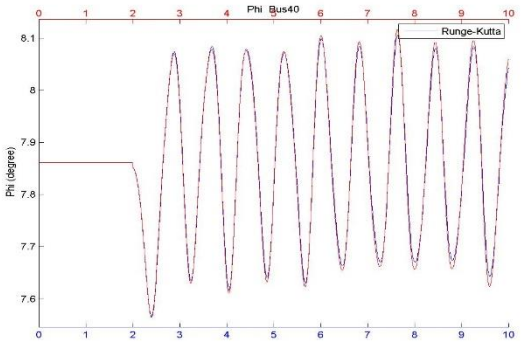

Phi Bus45
Runge-Kutta
Phi (degree)

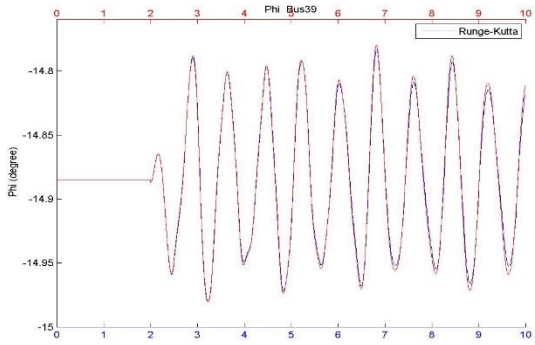

Phi Bus44
Runge-Kutta
Phi (degree)

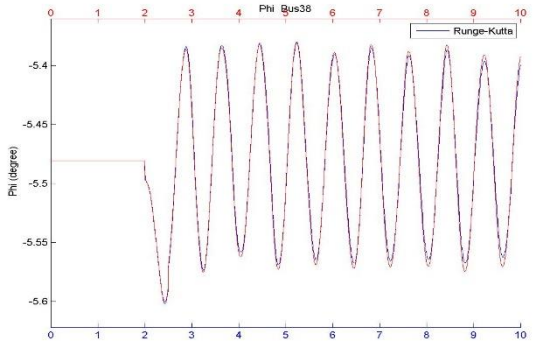

Phi Bus43
Runge-Kutta
Phi (degree)

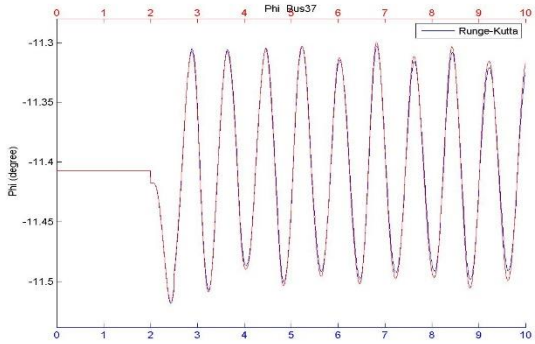

Phi Bus42
Runge-Kutta
Phi (degree)

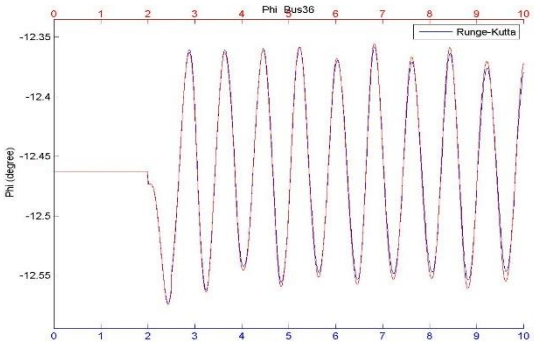

Phi Bus41
Runge-Kutta
Phi (degree)

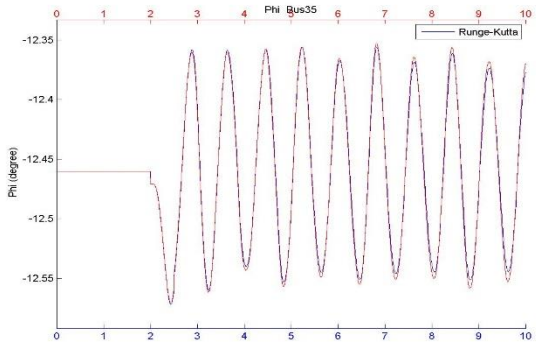

Phi Bus40
Runge-Kutta
Phi (degree)

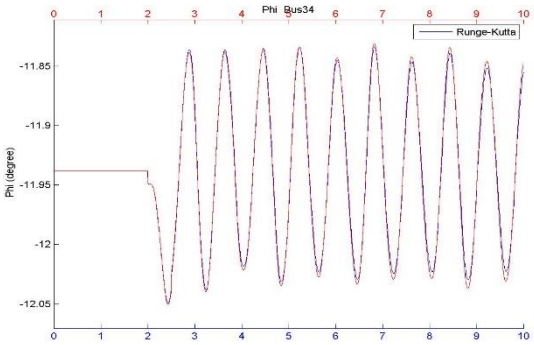

Phi Bus39
Runge-Kutta
Phi (degree)

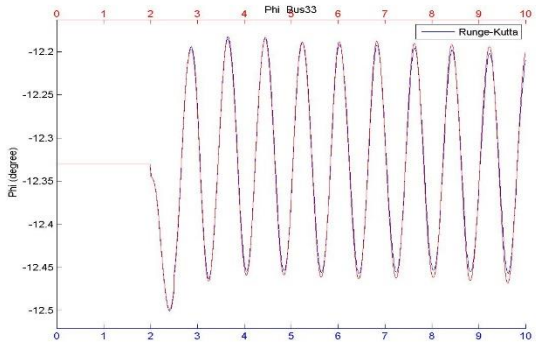

Phi Bus38
Runge-Kutta
Phi (degree)

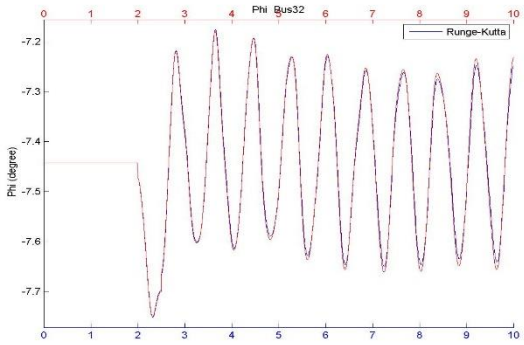

Phi Bus37
Runge-Kutta
Phi (degree)

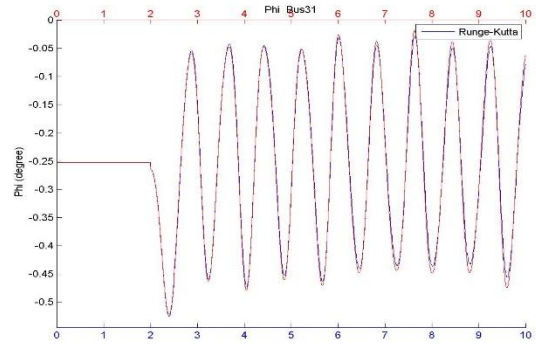

Phi Bus36
Runge-Kutta
Phi (degree)

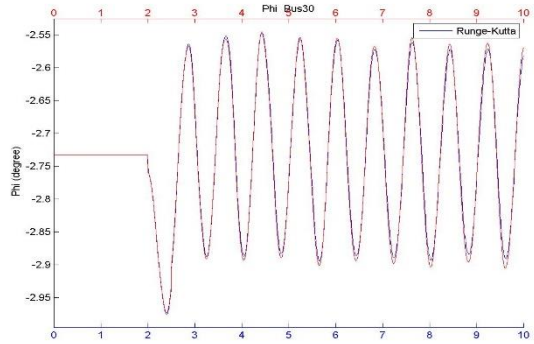

Phi Bus35
Runge-Kutta
Phi (degree)

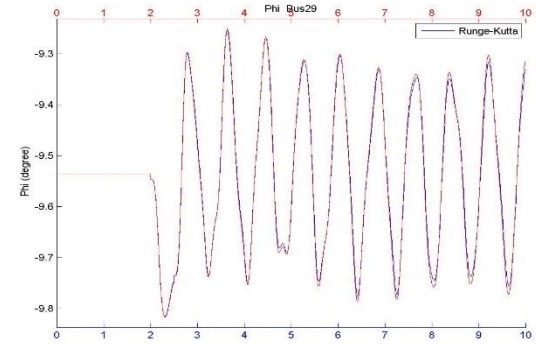

Phi Bus34
Runge-Kutta
Phi (degree)

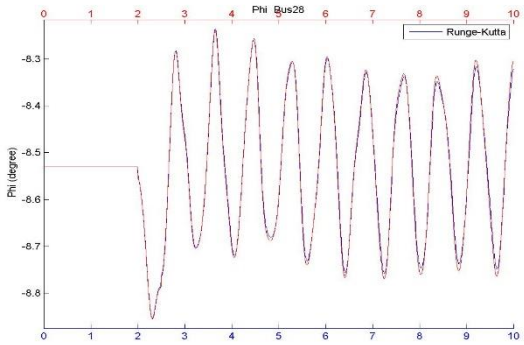

Phi Bus33
Runge-Kutta
Phi (degree)

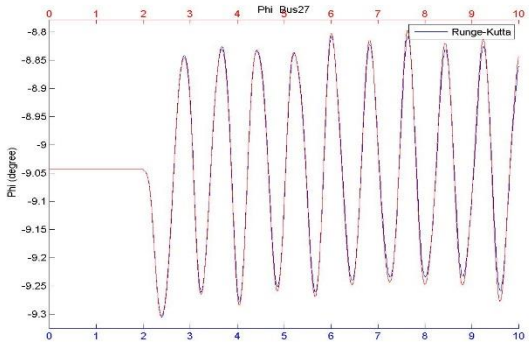

Phi Bus32
Runge-Kutta
Phi (degree)

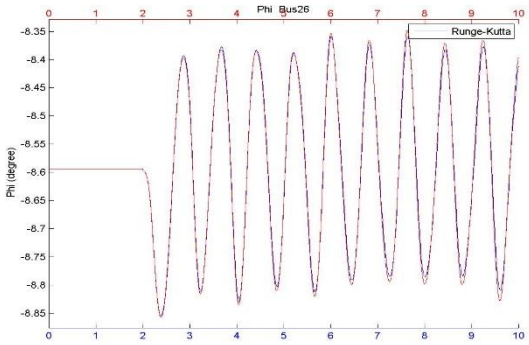

Phi Bus31
Runge-Kutta
Phi (degree)

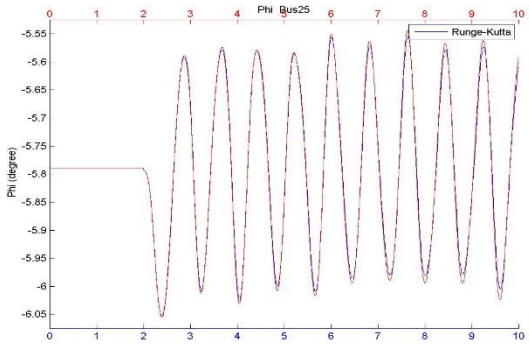

Phi Bus30
Runge-Kutta
Phi (degree)

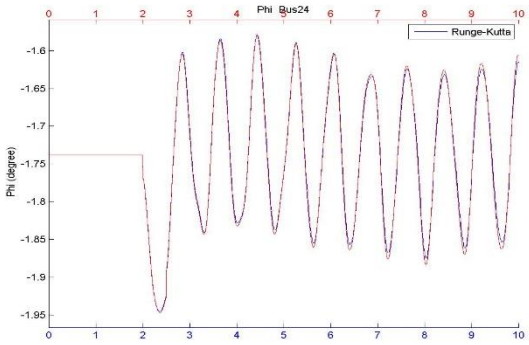

Phi Bus29
Runge-Kutta
Phi (degree)

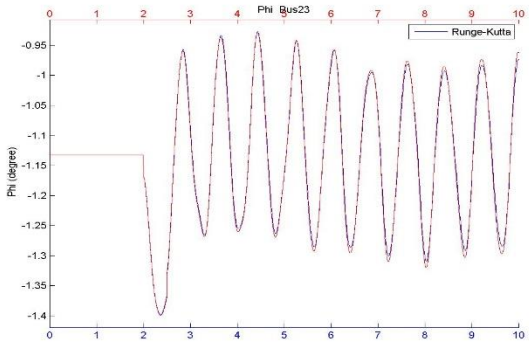

Phi Bus28
Runge-Kutta
Phi (degree)

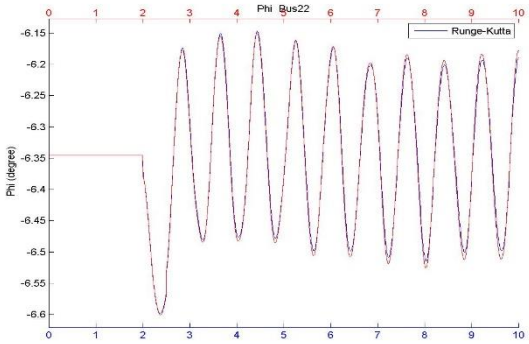

Phi Bus27
Runge-Kutta
Phi (degree)

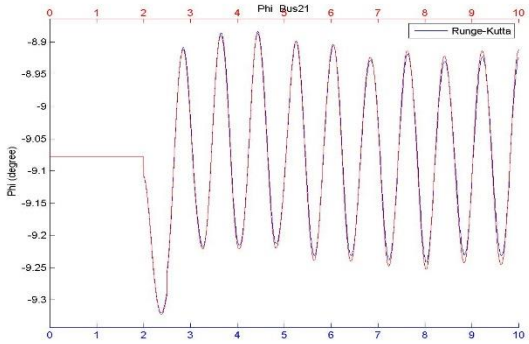

Phi Bus26
Runge-Kutta
Phi (degree)

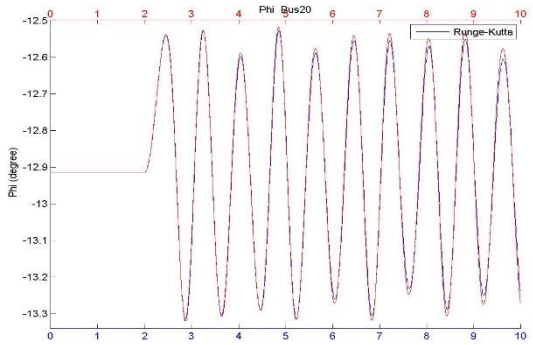

Phi Bus25
Runge-Kutta
Phi (degree)

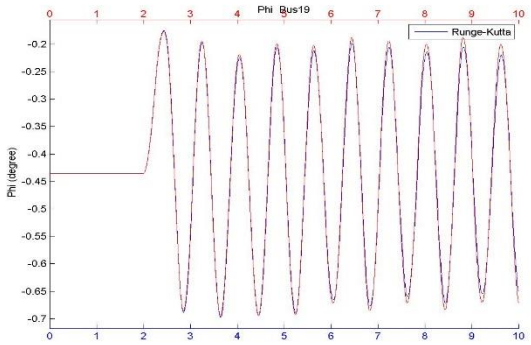

Phi Bus24
Runge-Kutta
Phi (degree)

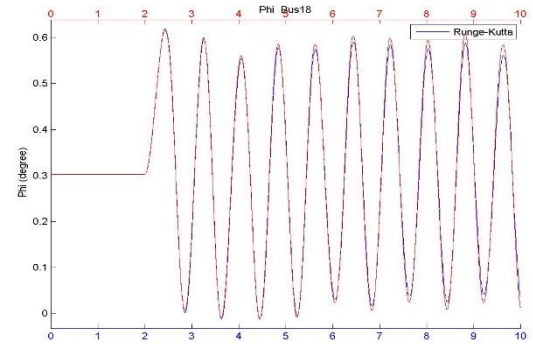

Phi Bus23
Runge-Kutta
Phi (degree)

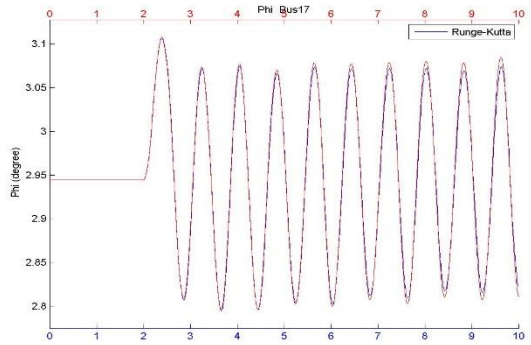

Phi Bus22
Runge-Kutta
Phi (degree)

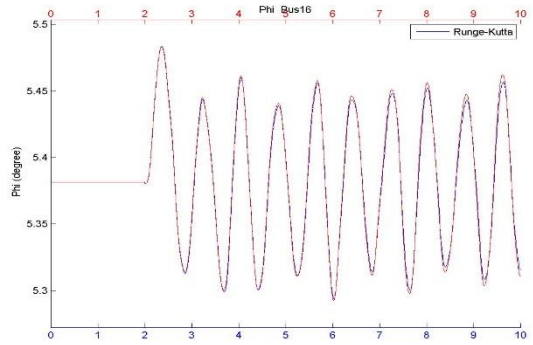

Phi Bus21
Runge-Kutta
Phi (degree)

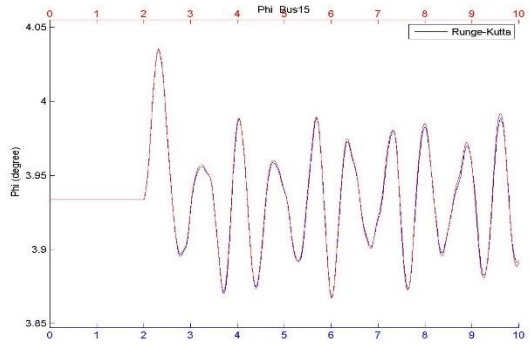

Phi Bus20
Runge-Kutta
Phi (degree)

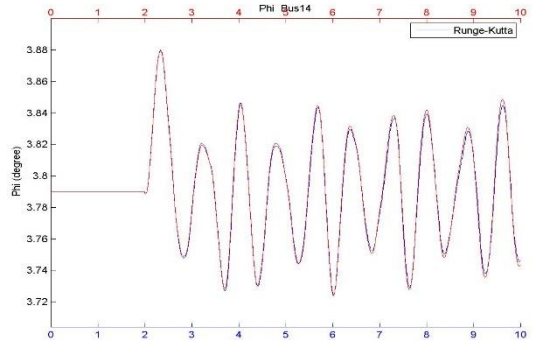

Phi Bus19
Runge-Kutta
Phi (degree)

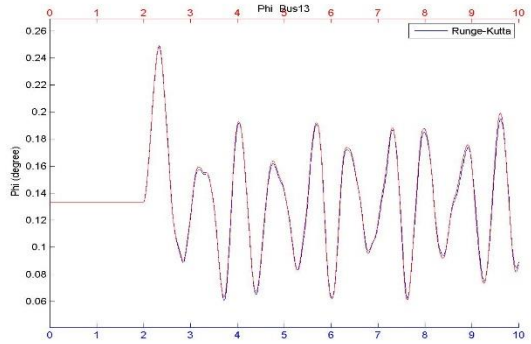

Phi Bus18
Runge-Kutta
Phi (degree)

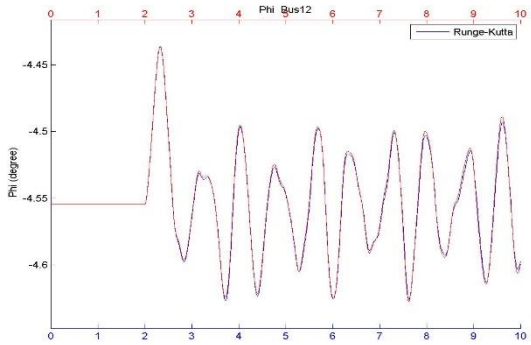

Phi Bus17
Runge-Kutta
Phi (degree)

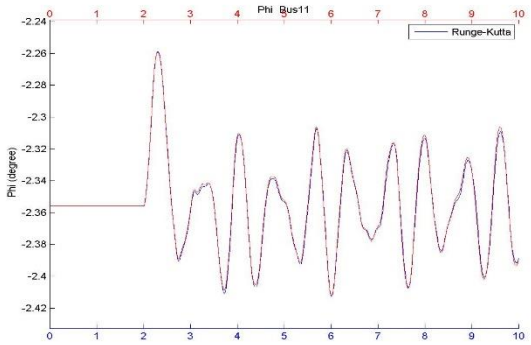

Phi Bus16
Runge-Kutta
Phi (degree)

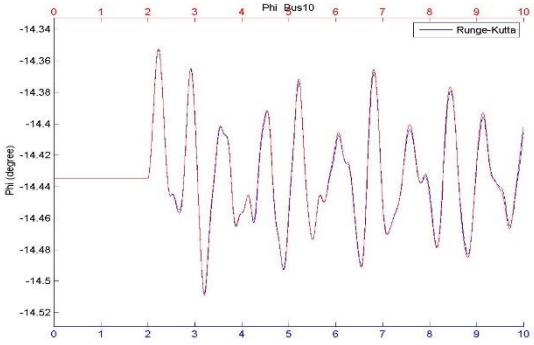

Phi Bus15
Runge-Kutta
Phi (degree)

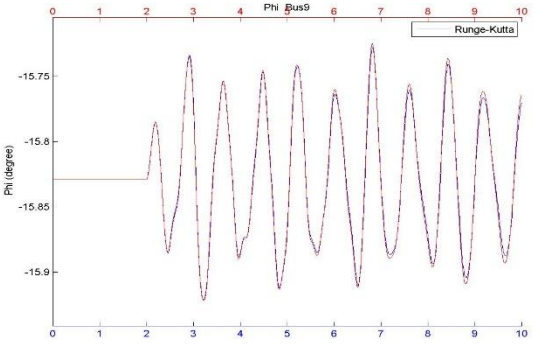

Phi Bus14
Runge-Kutta
Phi (degree)

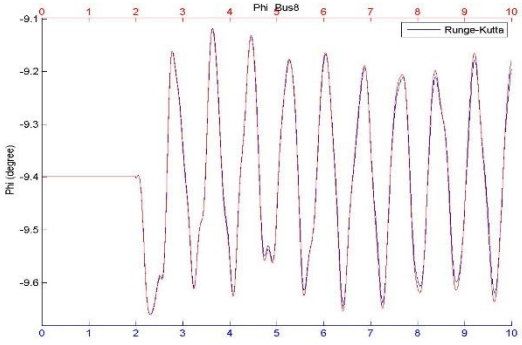

Phi Bus13
Runge-Kutta
Phi (degree)

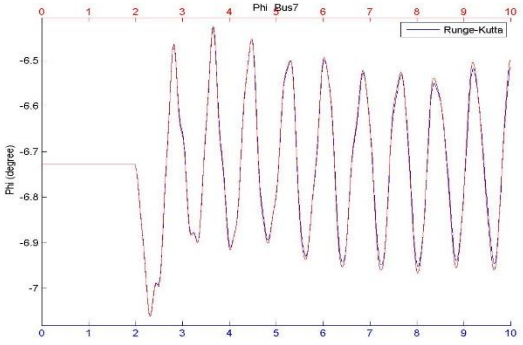

Phi Bus12
Runge-Kutta
Phi (degree)

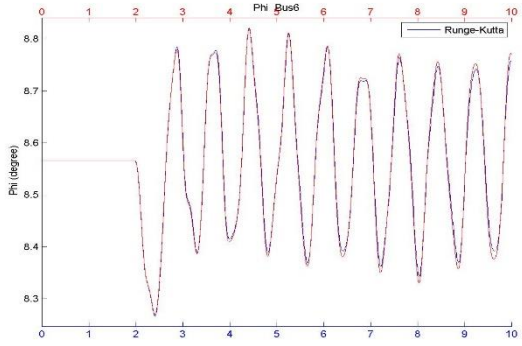

Phi Bus11
Runge-Kutta
Phi (degree)

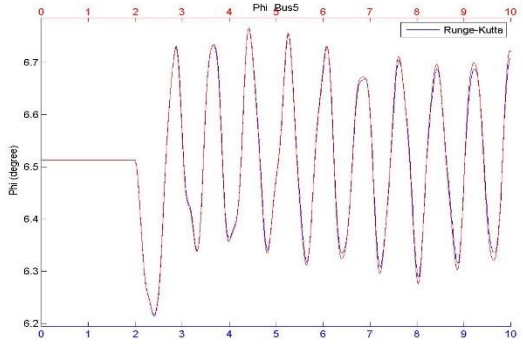

Phi Bus10
Runge-Kutta
Phi (degree)

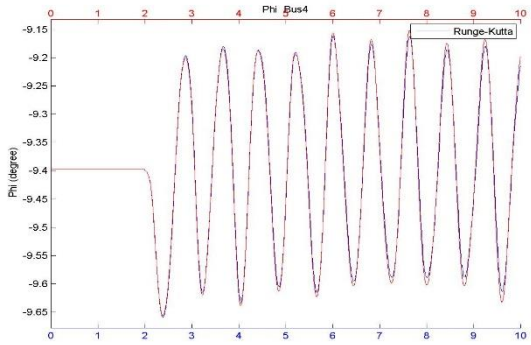

Phi Bus9
Runge-Kutta
Phi (degree)

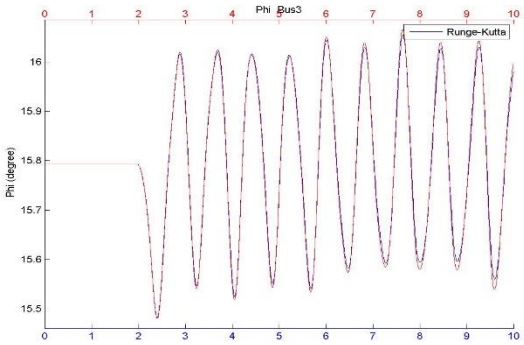

Phi Bus8
Runge-Kutta
Phi (degree)

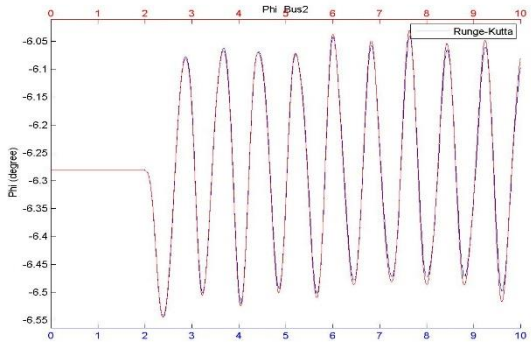

Phi Bus7
Runge-Kutta
Phi (degree)

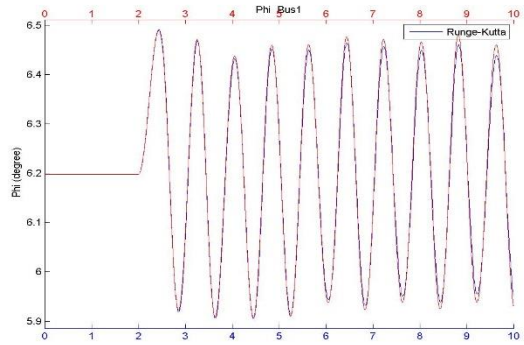

Phi Bus6
Runge-Kutta
Phi (degree)


Phi Bus5
Runge-Kutta
Phi (degree)


Phi_Bus4
Runge-Kutta
Phi (degree)


Phi_Bus3
Runge-Kutta
Phi (degree)


Phi_Bus2
Runge-Kutta
Phi (degree)


Phi_Bus1
Runge-Kutta
Phi (degree)

VITA

Boyu Wang was born in Hebei, China, in March 30 of 1990. After graduating high school from High School Affiliated to Renmin University, he attended Harbin Institute of Technology at Weihai, in 2008. He completed a Bachelor of Science degree in Electrical Engineering in 2012. In the fall of 2012, Boyu became a master student of Louisiana State University and he is a master candidate in the Department of Electrical Engineering and Computer Science at Louisiana State University.