Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

Erik-Anant Stedjan Narayan

Dynamically Simulating Power Systems Using a Self-Adaptive Time Step Method

Master's thesis in Energy and Environmental Engineering
Supervisor: Olav Bjarte Fosso
June 2020

NTNU
Norwegian University of Science and Technology

Erik-Anant Stedjan Narayan

Dynamically Simulating Power Systems Using a Self-Adaptive Time Step Method

Master's thesis in Energy and Environmental Engineering
Supervisor: Olav Bjarte Fosso
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

In this master's thesis, a second-order predictor-corrector method employing a self-adaptive time step (Gear's method) is implemented and investigated for dynamic power system simulations. The growing use of renewable energy sources and power electronic converters requires simulation tools capable of handling the increased system complexity. Gear's method has previously been shown to perform well for differential-algebraic equation (DAE) systems with ranging time constants, making it a promising candidate for dynamically simulating the future power system. Despite this, few published implementations exist, and strategies to improve the method have yet to be fully explored. This thesis presents a modeling of four separate power system cases with varying characteristics in terms of stability, size and complexity, saturation effects, topology changes and discrete system events, and degree of power electronic integration. Following this, Python implementations of Gear's method for each of the cases are developed, including comparisons with a commercially available DAE solver. The method was shown to perform well for the three systems dominated by synchronous generators. In addition, of the examined strategies, fixing the step length for at least 15 consecutive steps was shown to result in the largest improvement in terms of performance and solution accuracy. However, the method failed to successfully simulate the investigated multi-terminal direct current (MTDC) system. Even with the best performing strategy combination, Gear's method was unable to accurately capture the MTDC system voltages following a line outage. To conclude, Gear's method performed well, both in terms of computational resource use and solution accuracy, for systems dominated by synchronous generators or with limited power electronic integration. If the poor results obtained for the MTDC system were a result of some underlying modeling nature of power converters, or simply a feature of the particular formulation used in this thesis, could not be concluded.

Sammendrag

I denne masteroppgaven er en andreordens integrasjonsmetode basert på prediksjon og korreksjon med en selvregulerende steglengde (Gears metode) implementert og undersøkt for dynamiske kraftsystemsimuleringer. Den voksende bruken av fornybare energikilder og kraftomformere krever simuleringsmetoder som er i stand til å håndtere den økte systemkompleksiteten. Det har tidligere blitt vist at Gears metode egner seg godt til differensial-algebraiske ligningssystemer med sprikende tidskonstanter, noe som gjør den til en lovende kandidat til å dynamisk simulere framtidas kraftsystem. Til tross for dette finnes det få publiserte implementeringer og det gjenstår ennå å grundig utforske forbedringsstrategier. Denne oppgaven inneholder modellering av fire forskjellige kraftsystemcaser med varierende karakteristikker relatert til stabilitet, størrelse og kompleksitet, metningseffekter, topologiendringer og diskrete hendelser, og grad av kraftelektronisk innhold. Basert på dette blir Python-implementeringer av Gears metode for hver av de fire casene utviklet, inklusive sammenligninger med en kommersiell differensial-algebraisk ligningssystemløser. Metoden viste gode resultater for de tre synkrongeneratordominerte systemene. Av de undersøkte strategiene ga det å låse steglengden for minst 15 påfølgende steg den største økningen i ytelse og løsningsnøyaktighet. Derimot var metoden ikke i stand til å vellykket simulere det undersøkte multiterminale likestrøms-systemet (MTDC). Selv med den sterkest presterende strategien klarte Gears metode ikke å produsere akseptable simuleringsresultater av MTDC-spenningene ved linjeutfall. For å konkludere, Gears metode ga gode resultater, både med hensyn til ressursbruk og nøyaktighet, for systemer dominert av synkrongeneratorer eller med begrensede innslag av kraftelektronikk. Det var ikke mulig å konkludere om de svake resultatene oppnådd for MTDC-systemet var på grunn av et underliggende aspekt av kraftomformermodellering, eller på grunn av begrensninger i den spesifikke formuleringen brukt i oppgaven.

To Shankar, Gerd, Mina-Anette, and Maya-Elise

Preface

The following master's thesis concludes my final semester at the Department of Electric Power Engineering at the Norwegian University of Science and Technology. The workload corresponds to 30 ECTS credits.

The submission of this thesis also marks the crowning of five years of personal and academic growth; five years filled with challenges to overcome, lessons to learn, and moments to enjoy. The work presented in this thesis draws on a wide range of the knowledge and skills I have developed during my years in Trondheim and Aachen, and required me to push even further into my personal ever-increasing field of discovery.

It also bears mentioning that the outbreak of COVID-19 shaped the final months of my thesis work. Luckily, a kitchen table makes as good a desk as anything else and sunny days are still sunny days.

Nevertheless, I would like to take this opportunity to extend my sincere gratitude to my supervisor Olav Bjarte Fosso for his invaluable guidance and support, while always letting me remain the captain of my own ship. My co-supervisor Jalal Khodaparast Ghadikolaei is also deserving of my gratitude for providing a valuable discussion partner related to the method implementation and systems modeling.

Lastly, I would like to thank my friends, family and girlfriend for their unwavering love and support, and for never letting me forget what is truly important in life.

Trondheim, June 10, 2020

Contents

List of Tables

List of Figures

Acronyms

3MIB three machines to infinite bus.

AC alternating current.

DAE differential-algebraic equation.

DC direct current.

dq direct-quadrature, local version.

DQ direct-quadrature, global version.

MSE mean squared error.

MTDC multi-terminal direct current.

PI proportional-integral.

PSS power system stabilizer.

pu per unit.

SMIB single machine to infinite bus.

STATCOM static compensator.

VSC voltage source converter.

Chapter 1

Introduction

The following is the first chapter of this master's thesis, and serves as the thesis introduction. The chapter contains the background and motivation of the thesis, briefly highlighting the current need for the research carried out. Following this, the objectives and scope, including the planned limitations, of the thesis are stated, before the method used is briefly described. The chapter concludes with an outline of the thesis as a whole.

1.1 Background and Motivation

Today's ever-increasing use of renewable energy sources presents a substantial challenge to the field of power system analysis. As more converter-interfaced sources, such as solar and wind, are connected to the power system, the complexity of the differential-algebraic equation (DAE) system used in dynamic modeling increases. Increasing the share of power electronics in a system still largely dominated by synchronous generators widens the span of time constants present in the system, reducing the suitability of simulation methods with fixed time steps.

If renewable sources with their accompanying power electronics are to make a successful integration into the power system, simulation tools capable of handling the increased complexity are needed. Dynamic computer simulations form the foundation of both the development and implementation of new technologies, as well as daily operations and maintenance. Power systems constitute part of society's critical infrastructure, requiring extensive simulation work before real-world action can be taken.

Under the limitations imposed by the coming changes, conventional numerical integration methods employing fixed time step schemes are unable to balance the need for sufficient resolution with computational resource use. Either appropriate resolution is achieved, but at the cost of computational infeasibility, or the method is appropriately resource intensive, but possibly unable to capture the system's faster dynamics. Choosing an appropriate time step therefore becomes the key issue.

One possible solution would be to implement a numerical integration method with a variable time step, able to adapt to the dynamics displayed by the system. Gear's method is one such method [7]. It is based on a predictor-corrector scheme, of the

desired order, where the step length is adjusted appropriately based on a monitoring of the local truncation error.

However, other than some notable exceptions [6, 23], relatively little use of Gear's method for dynamic power systems simulations exists in the published literature. In addition, the previous implementations were done decades ago and are lacking in the descriptions required for reproducibility. The work done in [4] improved on this, confirming that Gear's method is suitable for simulating simple power system cases, and forming a stepping stone for further investigations into more complex and varied power system examples.

Therefore, a thorough description of Gear's method and implementation for more expanded and modern power systems are needed. Part of this includes a step-by-step explanation of the method together with the adaptations and strategies needed for dynamic power system simulation, as well as a comparison with commercially available simulation tools. In addition, the results and conclusions drawn by previous implementations should be further investigated. By doing so, a greater understanding of Gear's method as a potential dynamic simulation tool can be gained.

1.2 Objectives and Scope

The main objective of this master's thesis was to improve and expand upon the second-order version of Gear's method implemented in the author's specialization project [16] for dynamic power system simulations.

This overarching objective could be further divided into smaller objectives, more specifically to: (1) give an exhaustive description of the method in general, and this implementation in particular; (2) further gain experience and insight into the method as a dynamic power system simulation tool, over a range of power system cases; (3) verify previous implementations' conclusions and further evaluate proposed strategies; and (4) compare the simulation results obtained using Gear's method to other established simulation tools.

The key contributions of this thesis were to: (1) extend the list of power system cases and categories of cases implemented using Gear's method; (2) improve system initialization by combining load flow and other iterative methods; and (3) implement and evaluate proposed strategies for improving the method's computational performance and accuracy.

Still, the scope of this thesis contained some planned limitations. The objective was not to investigate the use of Gear's method for all possible power system cases. By investigating a few select uses, displaying a typical variety of power system cases, the aim was to be able to better understand the strong and weak points of the method, through the lens offered by these cases. The chosen cases would therefore inherently influence the results and possible conclusions.

1.3 Method

Firstly, a brief literature review of numerical integration methods for dynamic power system simulations was carried out. The literature review also contains the necessary

theory on power system modeling required to be able to follow along with the thesis. Secondly, Gear's method was described and implemented in Python [18], allowing for the simulation of four different power system cases of varying complexity. Building on this, the obtained simulation results were verified using MATLAB [14] and the effects of the step length strategies presented in [6] were evaluated.

1.4 Outline

The remainder of the master's thesis is structured as follows: Chapter 2 forms the theoretical foundation of the thesis, and builds the necessary understanding of numerical integration methods, power system modeling, and Gear's method.

Chapter 3 describes the four test systems implemented, as well as the actual modules, logic, and procedures used in the Python program. This includes a complete description of the differential and algebraic equations used in the system modeling, as well as a description of the validation method and performance metrics used to determine the effects of the applied improvement strategies.

Chapter 4 contains the results obtained from applying Gear's method to the four simulation systems. For each system a base case simulation was performed before the effects of applying different strategies to the method were explored. In addition, the results obtained from both the base case and strategy case simulations were compared to externally validated simulation results.

In Chapter 5 the results from Chapter 4 are discussed, and the effects of the applied strategies on the method's resource use and accuracy of the obtained results are highlighted.

Chapter 6 closes the master's thesis and presents its conclusions. The results and discussions are summarized and seen in the light of the objectives stated in Section 1.2. Lastly, a possible path for future work is proposed, including implementing Gear's method for other systems dominated by power converters, developing more sophisticated and automatic improvement strategies, and investigating the numerical stability of the method.

Appendix A contains the complete simulation results for all four cases, including parameter combinations and performance metrics scores.

1.5 Relationship with the Specialization Project

This master's thesis is a continuation of the work done in the author's specialization project [16]. Therefore, the structures are similar, and so is some of the content in the early chapters. However, the specialization project only served as a basis, the material has been reworked and expanded upon where appropriate, and the thesis can be considered as a complete standalone document.

Of the four cases investigated in this thesis, only the first one can be seen as an expanded, redone version of a case from the specialization project. The remaining three cases are all new and more complex than those presented in earlier work, but still based on the relevant lessons learned from the specialization project.

Chapter 2

Theoretical Foundation

The following chapter builds the theoretical foundation on which to form the remaining thesis. This includes a review of numerical integration methods, focusing on implicit methods and stiff equation systems, the general structure of dynamic power system models, steady-state calculations and modeling of both alternating- and direct current (AC and DC) systems, as well as coordinate transformations. Lastly, Gear's method is described. Chapter 2 can be seen as an expanded version of the theory chapter in the specialization project [16].

2.1 Numerical Integration Methods

In general, a set of non-linear differential equations do not have an analytical solution [13]. Instead, a numerical solution consisting of a series of function values $(y_1, y_2, \ldots, y_n, \ldots)$ satisfying the equation $\dot{y} = f(y, t)$ at time instants $(t_1, t_2, \ldots, t_n, \ldots)$ is used. Knowing all the previously calculated values $(\ldots, y_{n-2}, y_{n-1}, y_n)$, the following value y_{n+1} can be determined using a numerical integration method.

By their very nature, numerical solutions can only be seen as approximate solutions to differential equations, differing from their accurate values by a local error. This local error is made up by round-off and method errors. Round-off errors are introduced by the nature of floating-point arithmetic used by computers, while the type, order, and step length of the integration method determine the method error. Each new step taken by the integration method introduces new errors. The local error of any given step is therefore the sum of the local error introduced by itself and the propagated local errors from all the previous steps. If the local error does not increase from step to step, the integration method is called numerically stable. If, however, the local error grows unboundedly as the integration proceeds, the method is called numerically unstable. The cumulative effect of the errors may then cause the value of y_n to be meaningfully different from the accurate solution.

Numerical integration methods can generally be sorted into explicit and implicit methods. Only the latter category will be explored in this section, as these methods are particularly well-suited for dynamic power system simulations.

2.1.1 Implicit Integration Methods

For the differential equation $\dot{y} = f(y,t)$, the value y_{n+1} can be determined by integrating the function $y(t)$ along the time path from t_n to t_{n+1}. By approximating $f(y,t)$ by a power series $w(t)$ in the given time interval, the solution y_{n+1} can be determined by

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(y,t)dt \approx y_n + \int_{t_n}^{t_{n+1}} w(t)dt. \tag{2.1}$$

Alternatively, $y(t)$, in stead of $f(y,t)$, can be approximated by a power series. This gives $y(t) \approx w(t)$ and the coefficients in the approximating polynomial $w(t)$ are functions of consecutive values of $(\ldots, y_{n-2}, y_{n-1}, \ldots)$. Differentiating with respect to time gives $\dot{y}(t) \approx \dot{w}(t)$ or $\dot{w}(t) \approx f(y,t)$, resulting in the following implicit integration formula:

$$y_{n+1} = \sum_{j=0}^{r} a_j y_{n-j} + b_0 h f_{n+1}, \tag{2.2}$$

where $f_i = f(y_i, t_i) = f(y(t_i))$ is the value of the function at a given point in time t_i and h is the integration step length. a_j and b_0 are constant coefficients of the power series. r is the number of points used in the approximation and is referred to as the order of the method.

It is worth noting that in order to calculate y_{n+1} using the formulation in (2.2), the approximation polynomial $w(t)$ is calculated using the known values $(\ldots, f_{n-2}, f_{n-1}, f_n)$ and the unknown value of f_{n+1}. This is what makes the method implicit. Equation (2.2) is one of the so-called implicit Gear formulas.

Furthermore, (2.2) also shows the disadvantage of implicit methods, namely that y_{n+1} appears on both sides of the equation. This means that if the function $f(y,t)$ is non-linear, as it often is in power system modeling, the unknown y_{n+1} must be found iteratively.

One such iterative method is the Newton-Raphson method. For any equation $F(y) = 0$ Newton-Raphson's formula is

$$y^{(l+1)} = y^{(l)} - \left[\frac{\partial F}{\partial y}\right]_{(l)}^{-1} F\left(y^{(l)}\right), \tag{2.3}$$

where the bracketed indexes indicate the iteration number. For a given time step n, an initial guess $y^{(0)}$ is provided before (2.3) is used to calculate the next iteration step $y^{(l+1)}$. This process is repeated until the difference between two consecutive iteration calculations is acceptably small.

2.1.2 Stiffness of Differential Equation Systems

The time scale of the dynamics of the modeled system plays a key role in selecting the appropriate numerical integration method. The solution of any set of linear differential equations consists of a linear combination of exponential functions, each

describing the different system modes. These modes are again defined by the eigenvalues of the system, which are linked to the time scale of the different dynamics in the model.

When the eigenvalues are spread out over a large range in the complex plane, the solution will consist of a combination of fast-changing dynamics, corresponding to eigenvalues far away from the origin, and slow-changing dynamics, corresponding to eigenvalues close to zero. If this is the case, the differential equation system is referred to as a stiff system. For non-linear differential equations, the Jacobi matrix, i.e. the matrix of the system's first-order derivatives, and its eigenvalues are not constant. Therefore, a non-linear system can only be called stiff if its linear approximation is stiff.

Moreover, the main advantage of the Gear formulas, compared to similar formulas, is that they have a relatively large area of numerical stability, making them more suitable for use with stiff systems [13].

2.2 Structure of Power System Models

Figure 2.1 illustrates the general structure of power system models used for dynamic simulations. The overall power system representation includes models of synchronous generators and the associated excitation systems and prime movers, interconnecting transmission network and static loads, induction and synchronous motor loads, and other devices, such as power electronic converters and static compensators.

In addition to non-linear system equations, large discontinuities due to faults and network switching, and small discontinuities due to limits on system variables, appear in the system model [10].

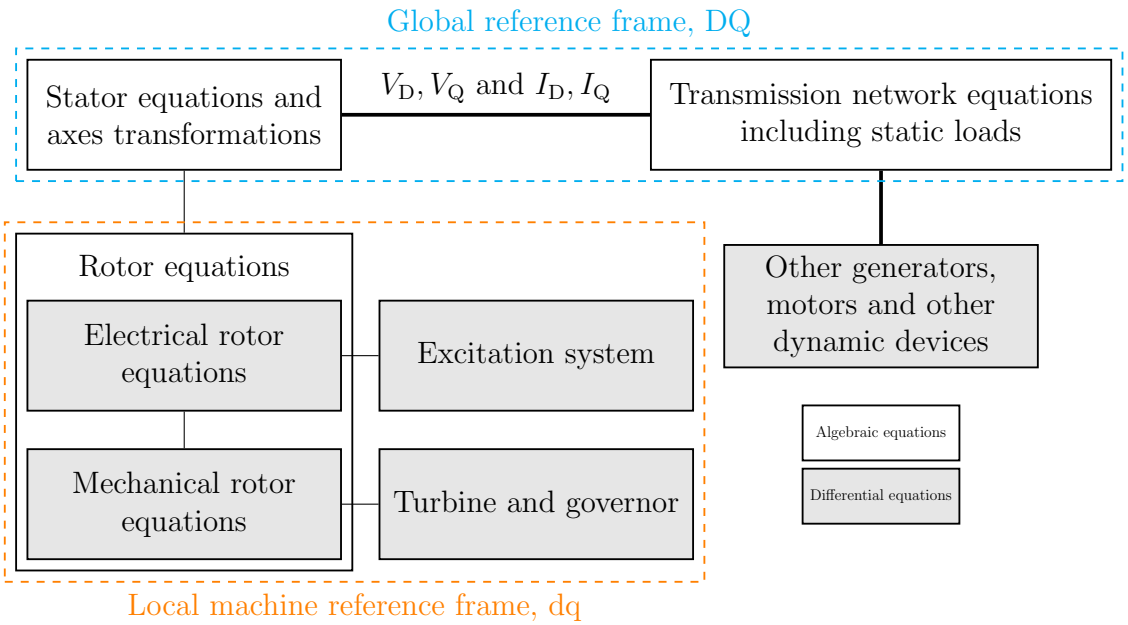


Figure 2.1: Structure of the power system model used for dynamic simulations, adapted from [10].

The model equations are arranged into a set of ordinary differential equations of the form

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, t), \tag{2.4a}$$

and sparse algebraic equations of the form

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{y}, t), \tag{2.4b}$$

where $\boldsymbol{y}$ is the vector of state variables (state vector) of dimensions $(K \times 1)$ and dot-notation is used to indicate differentiation with respect to time. Throughout this thesis bold face is used to indicate vectors or matrices. Dynamic modeling of power systems is therefore a differential-algebraic initial-value problem, meaning that the system solution is determined based on the state values at time $t = 0\,\mathrm{s}$.

The state vector can be further arranged into the form

$$\boldsymbol{y} = \begin{bmatrix} y_0 & y_1 & \dots & y_k \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{y}_{\mathrm{d}} & \boldsymbol{y}_{\mathrm{a}} \end{bmatrix}^{\mathrm{T}}, \tag{2.5}$$

where $\boldsymbol{y_d}$ and $\boldsymbol{y_a}$ are the differential and algebraic variables, respectively. T is used to denote the transpose of a vector or matrix. $\boldsymbol{y_d}$ has the dimensions $(N \times 1)$ and $\boldsymbol{y_a}$ has $(M \times 1)$, giving $K = M + N$. Some state variables appear in both the differential and in the algebraic equations, constituting the interface variables [22]. Typical examples are the stator internal voltages and current components of generators.

2.3 Steady-State Modeling of Power Systems

When performing dynamic power system simulations, the system has to be initialized. This is done by determining the system's steady-state load flow solution, meaning that the voltages and injected powers of all the buses in the network are determined. From this, each state variable can be initialized, giving an equilibrium.

Initially, the relationships between the network bus voltages and injected currents in a network consisting of N buses can be defined using the bus admittance matrix as follows [13]:

$$\begin{bmatrix} \underline{I}_0 \\ \vdots \\ \underline{I}_i \\ \vdots \\ \underline{I}_{(N-1)} \end{bmatrix} = \begin{bmatrix} \underline{Y}_{00} & \cdots & \underline{Y}_{0i} & \cdots & \underline{Y}_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \underline{Y}_{i0} & \cdots & \underline{Y}_{ii} & \cdots & \underline{Y}_{i(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \underline{Y}_{(N-1)0} & \cdots & \underline{Y}_{(N-1)i} & \cdots & \underline{Y}_{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} \underline{V}_0 \\ \vdots \\ \underline{V}_i \\ \vdots \\ \underline{V}_{(N-1)} \end{bmatrix} \quad \text{or} \quad \underline{\boldsymbol{I}} = \underline{\boldsymbol{Y}}\underline{\boldsymbol{V}}. \tag{2.6}$$

Note that throughout this thesis underlined letters will be used to denote phasors or complex quantities. The subscripts i, j represent bus numbers, making $\underline{V}_i$ the phasor voltage at bus i, i.e. $\underline{V}_i = V_i \angle \delta_i$. $\underline{I}_i$ is the algebraic net sum of the injected currents into bus i, with the positive current direction being defined as flowing into the bus. $\underline{Y}_{ij} \forall i \neq j$ is the mutual admittance between buses i and j, and equals the negative sum of the branch series admittance linking the two buses. $\underline{Y}_{ii}$ is the self-admittance of bus i and equals the sum of all the admittances connected directly to bus i, including any shunt admittances.

For a given bus i, the current injected can based on (2.6) be expressed in a rectangular coordinate system (D, Q) as

$$\underline{I}_i = \sum_{j=0}^{N-1} \underline{Y}_{ij}\underline{V}_j = \sum_{j=0}^{N-1} (G_{ij} + \mathrm{j}B_{ij})(V_{\mathrm{D}_j} + \mathrm{j}V_{\mathrm{Q}_j}), \tag{2.7}$$

where the complex admittances and bus voltages can be generally written with the complex operator j as $\underline{Y}_{ij} = G_{ij} + \mathrm{j}B_{ij}$ and $\underline{V}_j = V_{\mathrm{D}_j} + \mathrm{j}V_{\mathrm{Q}_j}$, respectively. Separating (2.7) into real and imaginary parts gives

$$I_{\mathrm{D}_i} = \sum_{j=0}^{N-1} (G_{ij}V_{\mathrm{D}_j} - B_{ij}V_{\mathrm{Q}_j}) \quad \text{and} \tag{2.8a}$$

$$I_{\mathrm{Q}_i} = \sum_{j=0}^{N-1} (B_{ij}V_{\mathrm{D}_j} + G_{ij}V_{\mathrm{Q}_j}). \tag{2.8b}$$

With this notation the complex network equation (2.6) can be transformed into the real-number domain to give

$$\begin{bmatrix} \boldsymbol{I}_0 \\ \vdots \\ \boldsymbol{I}_i \\ \vdots \\ \boldsymbol{I}_{(N-1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{00} & \cdots & \boldsymbol{Y}_{0i} & \cdots & \boldsymbol{Y}_{1(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{i0} & \cdots & \boldsymbol{Y}_{ii} & \cdots & \boldsymbol{Y}_{i(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{(N-1)0} & \cdots & \boldsymbol{Y}_{(N-1)i} & \cdots & \boldsymbol{Y}_{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_0 \\ \vdots \\ \boldsymbol{V}_i \\ \vdots \\ \boldsymbol{V}_{(N-1)} \end{bmatrix} \quad \text{or} \quad \boldsymbol{I} = \boldsymbol{Y}\boldsymbol{V}, \tag{2.9a}$$

where all the elements are now real submatrices of the form

$$\boldsymbol{I}_i = \begin{bmatrix} I_{\mathrm{D}_i} \\ I_{\mathrm{Q}_i} \end{bmatrix}, \quad \boldsymbol{V}_i = \begin{bmatrix} V_{\mathrm{D}_i} \\ V_{\mathrm{Q}_i} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{Y}_{ij} = \begin{bmatrix} G_{ij} & -B_{ij} \\ B_{ij} & G_{ij} \end{bmatrix}. \tag{2.9b}$$

Based on (2.7) the apparent power injected into any bus i can be expressed using polar notation as

$$\begin{aligned} \underline{S}_i = P_i + \mathrm{j}Q_i = \underline{V}_i\underline{I}_i^* &= \mathrm{e}^{\mathrm{j}\delta_i}\left[Y_{ii}V_i\mathrm{e}^{-\mathrm{j}(\delta_i+\theta_{ii})} + \sum_{j=0,j\neq i}^{N-1} V_jY_{ij}\mathrm{e}^{-\mathrm{j}(\delta_i+\theta_{ij})}\right] \\ &= V_i^2Y_{ii}\mathrm{e}^{-\mathrm{j}\theta_{ii}} + V_i\sum_{j=0,j\neq i}^{N-1} V_jY_{ij}\mathrm{e}^{\mathrm{j}(\delta_i-\delta_j-\theta_{ij})}, \end{aligned} \tag{2.10}$$

where star notation is used to denote complex conjugation and admittances are written as $\underline{Y}_{ij} = Y_{ij}\angle\theta_{ij}$. Separating the real and imaginary parts gives the active and reactive bus power injections P_i and Q_i, respectively, as

$$P_i = V_i^2Y_{ii}\cos\theta_{ii} + \sum_{j=0,j\neq i}^{N-1} V_iV_jY_{ij}\cos(\delta_i - \delta_j - \theta_{ij}) \quad \text{and} \tag{2.11a}$$

$$Q_i = -V_i^2 Y_{ii} \sin\theta_{ii} + \sum_{j=0, j\neq i}^{N-1} V_i V_j Y_{ij} \sin\left(\delta_i - \delta_j - \theta_{ij}\right). \tag{2.11b}$$

It is often useful to combine the use of polar and rectangular coordinates, expressing voltages in the former and admittances in the latter. This makes it possible to express (2.11) as

$$P_i = V_i^2 G_{ii} + \sum_{j=0, j\neq i}^{N-1} V_i V_j [B_{ij} \sin\left(\delta_i - \delta_j\right) + G_{ij} \cos\left(\delta_i - \delta_j\right)] \quad \text{and} \tag{2.12a}$$

$$Q_i = -V_i^2 B_{ii} + \sum_{j=0, j\neq i}^{N-1} V_i V_j [G_{ij} \sin\left(\delta_i - \delta_j\right) - B_{ij} \cos\left(\delta_i - \delta_j\right)]. \tag{2.12b}$$

2.3.1 Newton-Raphson Load Flow Solution Method

When solving the steady-state load flow problem using the Newton-Raphson method, the general form in (2.3) is used, expanded into matrix form. The goal is to determine the magnitude and angle of every bus voltage in the network, satisfying the stated active and reactive power injections (or withdrawals in the case of net loads).

Initially, one bus is defined to be the slack bus. The slack bus, often defined as bus number $(N-1)$ in an N-bus system (with zero-indexing), serves as a reference, resulting in a defined voltage of $\underline{V}_{N-1} = V_{N-1}\angle\delta_{N-1} \equiv 1.0\angle 0.0^\circ$ per unit (pu). The slack bus also serves to balance the system, eliminating any potential power imbalances caused by the other buses.

The remaining bus voltage magnitudes and angles are arranged into the state vector $\boldsymbol{x}$ as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\delta} & \boldsymbol{V} \end{bmatrix}^{\mathrm{T}}, \tag{2.13}$$

where $\boldsymbol{\delta}$ is the vector of the non-slack bus voltage angles and $\boldsymbol{V}$ is the vector containing the non-slack bus voltage magnitudes.

The stated active and reactive power injections are collected into the column vector $\boldsymbol{b}$ as

$$\boldsymbol{b} = \begin{bmatrix} P_0^{\mathrm{sp}} & \cdots & P_i^{\mathrm{sp}} & \cdots & P_{N-2}^{\mathrm{sp}} & Q_0^{\mathrm{sp}} & \cdots & Q_i^{\mathrm{sp}} \cdots & Q_{N-2}^{\mathrm{sp}} \end{bmatrix}^{\mathrm{T}}, \tag{2.14}$$

sp indicating set-point values. Note that the power injections of the slack bus $(N-1)$ are not stated.

The Newton-Raphson method requires an initial state vector guess. It is common to utilize a flat start, i.e. setting all voltage magnitudes equal to 1.0 pu and angles equal to 0.0°. The equations to be solved are the injected power equations (2.12) for each of the non-slack buses.

The Jacobian matrix $\boldsymbol{J}(\boldsymbol{\delta}, \boldsymbol{V})$ can be expressed as

$$\boldsymbol{J}(\boldsymbol{\delta}, \boldsymbol{V}) = \begin{bmatrix} \dfrac{\partial \boldsymbol{P}}{\partial \boldsymbol{\delta}} & \dfrac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}} \\ \dfrac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\delta}} & \dfrac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}} \end{bmatrix}, \tag{2.15}$$

where

$$\boldsymbol{P} = \begin{bmatrix} P_0 & \cdots & P_i & \cdots & P_{N-2} \end{bmatrix}^{\mathrm{T}} \quad \text{and} \tag{2.16a}$$

$$\boldsymbol{Q} = \begin{bmatrix} Q_0 & \cdots & Q_i & \cdots & Q_{N-2} \end{bmatrix}^{\mathrm{T}} \tag{2.16b}$$

as defined by (2.12), and

$$\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\delta}} = \begin{bmatrix} \frac{\partial P_0}{\partial \delta_0} & \cdots & \frac{\partial P_0}{\partial \delta_i} & \cdots & \frac{\partial P_0}{\partial \delta_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial P_i}{\partial \delta_0} & \cdots & \frac{\partial P_i}{\partial \delta_i} & \cdots & \frac{\partial P_i}{\partial \delta_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial P_{N-2}}{\partial \delta_0} & \cdots & \frac{\partial P_{N-1}}{\partial \delta_i} & \cdots & \frac{\partial P_{N-2}}{\partial \delta_{N-2}} \end{bmatrix}, \tag{2.17a}$$

$$\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}} = \begin{bmatrix} \frac{\partial P_0}{\partial V_0} & \cdots & \frac{\partial P_0}{\partial V_i} & \cdots & \frac{\partial P_0}{\partial V_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial P_i}{\partial V_0} & \cdots & \frac{\partial P_i}{\partial V_i} & \cdots & \frac{\partial P_i}{\partial V_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial P_{N-2}}{\partial V_0} & \cdots & \frac{\partial P_{N-2}}{\partial V_i} & \cdots & \frac{\partial P_{N-2}}{\partial V_{N-2}} \end{bmatrix} \quad \text{and} \tag{2.17b}$$

$$\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\delta}} = \begin{bmatrix} \frac{\partial Q_0}{\partial \delta_0} & \cdots & \frac{\partial Q_0}{\partial \delta_i} & \cdots & \frac{\partial Q_0}{\partial \delta_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial Q_i}{\partial \delta_0} & \cdots & \frac{\partial Q_i}{\partial \delta_i} & \cdots & \frac{\partial Q_i}{\partial \delta_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial Q_{N-2}}{\partial \delta_0} & \cdots & \frac{\partial Q_{N-2}}{\partial \delta_i} & \cdots & \frac{\partial Q_{N-2}}{\partial \delta_{N-2}} \end{bmatrix}, \tag{2.17c}$$

$$\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}} = \begin{bmatrix} \frac{\partial Q_0}{\partial V_0} & \cdots & \frac{\partial Q_0}{\partial V_i} & \cdots & \frac{\partial Q_0}{\partial V_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial Q_i}{\partial V_0} & \cdots & \frac{\partial Q_i}{\partial V_i} & \cdots & \frac{\partial Q_i}{\partial V_{N-2}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial Q_{N-2}}{\partial V_0} & \cdots & \frac{\partial Q_{N-2}}{\partial V_i} & \cdots & \frac{\partial Q_{N-2}}{\partial V_{N-2}} \end{bmatrix}. \tag{2.17d}$$

It is convenient to define the helper variables [3]

$$T_{ij} \equiv B_{ij} \sin(\delta_i - \delta_j) + G_{ij} \cos(\delta_i - \delta_j) \quad \text{and} \tag{2.18a}$$

$$U_{ij} \equiv G_{ij} \sin(\delta_i - \delta_j) - B_{ij} \cos(\delta_i - \delta_j), \tag{2.18b}$$

and by substituting (2.18) into (2.12) and differentiating, the Jacobi elements in (2.17) can be expressed as

$$\frac{\partial P_i}{\partial \delta_i} = -V_i \sum_{j=0, j\neq i}^{N-1} V_j U_{ij}, \tag{2.19a}$$

$$\frac{\partial P_i}{\partial \delta_j} = V_i V_j U_{ij}, \tag{2.19b}$$

$$\frac{\partial P_i}{\partial V_i} = 2V_i G_{ii} + \sum_{j=0, j\neq i}^{N-1} V_j T_{ij}, \tag{2.19c}$$

$$\frac{\partial P_i}{\partial V_j} = V_i T_{ij} \quad \text{and} \tag{2.19d}$$

$$\frac{\partial Q_i}{\partial \delta_i} = V_i \sum_{j=0, j\neq i}^{N-1} V_j T_{ij}, \tag{2.19e}$$

$$\frac{\partial Q_i}{\partial \delta_j} = -V_i V_j T_{ij}, \tag{2.19f}$$

$$\frac{\partial Q_i}{\partial V_i} = -2V_i B_{ii} + \sum_{j=0, j\neq i}^{N-1} V_j T_{ij}, \tag{2.19g}$$

$$\frac{\partial Q_i}{\partial V_j} = V_i U_{ij}. \tag{2.19h}$$

Finally, the elements can be arranged into the form of (2.3) to give

$$\boldsymbol{x}^{(l+1)} = \boldsymbol{x}^{(l)} - \boldsymbol{J}_{(l)}^{-1} \left(\boldsymbol{b} - \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{Q} \end{bmatrix}^{(l)} \right), \tag{2.20}$$

which is iterated upon until a satisfactorily accurate solution is found.

2.3.2 DC Load Flow Solution Method

A DC network can also be solved using the Newton-Raphson method, but the problem formulation varies slightly from that of AC networks.

In a network consisting of N number of nodes, still with zero-indexing, node number $(N-1)$ is defined as the slack node. This slack node is defined by its DC voltage, while the other $(N-1)$ nodes are defined by their injected active power [2]. The

term *node* is used as opposed to *bus* in order to distinguish the connection points of DC lines from those of AC lines.

The steady-state DC equations are

$$\boldsymbol{I}_{\mathrm{dc}} = \boldsymbol{Y}_{\mathrm{dc}}\boldsymbol{V}_{\mathrm{dc}} \quad \text{and} \tag{2.21a}$$

$$P_{\mathrm{dc}_i} = 2V_{\mathrm{dc}_i}I_{\mathrm{dc}_i} \quad \forall i < N, \tag{2.21b}$$

where $\boldsymbol{I}_{\mathrm{dc}}$ is the column vector of node current injections, $\boldsymbol{Y}_{\mathrm{dc}}$ is the node admittance matrix of the network, constructed using only line resistances, and $\boldsymbol{V}_{\mathrm{dc}}$ is the column vector of node voltages.

By solving each of the N equations in (2.21b) for I_{dc_i} and stacking them into a column vector, they can be combined with (2.21a) to form the following system of non-linear equations:

$$0 = \boldsymbol{Y}_{\mathrm{dc}}\boldsymbol{V}_{\mathrm{dc}} - \begin{bmatrix} \frac{P_{\mathrm{dc}_i}}{2V_{\mathrm{dc}_i}} \end{bmatrix}. \tag{2.22}$$

As node $(N-1)$ is classified as the slack node, the vector of unknowns $\boldsymbol{x}$ becomes

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} V_{\mathrm{dc}_0} \\ \vdots \\ V_{\mathrm{dc}_{(N-2)}} \\ P_{\mathrm{dc}_{(N-1)}} \end{bmatrix}. \tag{2.23}$$

The DC node admittance matrix is partitioned as follows:

$$\boldsymbol{Y}_{\mathrm{dc}} = \begin{bmatrix} \boldsymbol{Y}_{00} & \boldsymbol{y}_{01} \\ \boldsymbol{y}_{10} & y_{11} \end{bmatrix}, \tag{2.24}$$

where $\boldsymbol{Y}_{00}$ is an $(N-1)\times(N-1)$ matrix, $\boldsymbol{y}_{01}$ is a column vector with $(N-1)$ elements, $\boldsymbol{y}_{10}$ is a row vector with $(N-1)$ elements, and y_{11} is a scalar.

This formulation leads to the following equation system, which can be solved using the Newton-Raphson method:

$$0 = \boldsymbol{Y}_{00}\boldsymbol{x}_0 + \boldsymbol{y}_{01}\cdot V_{\mathrm{dc}_{\mathrm{ref}}} - \begin{bmatrix} \frac{P_{\mathrm{dc}_i}}{2x_{0_i}} \end{bmatrix} \quad \text{and} \tag{2.25a}$$

$$0 = \boldsymbol{y}_{10}\boldsymbol{x}_0 + y_{11}\cdot V_{\mathrm{dc}_{\mathrm{ref}}} - \frac{x_1}{2V_{\mathrm{dc}_{\mathrm{ref}}}}, \tag{2.25b}$$

where $V_{\mathrm{dc}_{\mathrm{ref}}}$ is the reference value for the DC voltage. From (2.25), all the unknown node voltages can be determined. By appending the slack node voltage to the end of $\boldsymbol{x}_0$, (2.21a) can be used to determine all the net injected node currents. The branch currents $\boldsymbol{I}_{\mathrm{cc}}$ are determined using the node connection matrix $\boldsymbol{Y}_{\mathrm{cc}}$ and the following equation:

$$\boldsymbol{I}_{\mathrm{cc}} = \boldsymbol{Y}_{\mathrm{cc}}\boldsymbol{V}_{\mathrm{dc}}. \tag{2.26}$$

$\boldsymbol{Y}_{\mathrm{cc}}$ has N columns and number of rows equal to the number of network branches. Each row represents one branch current, and contains a 1 in the column corresponding to the positively defined node of the branch and a (-1) at the negatively defined node. The remaining elements are zero. Typically, the positive branch current direction is defined from a lower node index to a higher node index.

2.4 Space Phasors and Two-Dimensional Reference Frames

A key issue when designing three-phase voltage source converter (VSC) systems is to enable the tracking of sinusoidal voltage or current commands [25]. Conventional proportional-integral (PI) controllers only enable the tracking of constant references, requiring a compensator of higher order and bandwidth if a sinusoidal signal is to be tracked.

The space phasor concept and the accompanying $\alpha\beta$- and dq-frame concepts provide a solution to the stated challenge, and therefore become central in the modeling of VSC systems.

2.4.1 Space Phasor Representation of a Balanced Three-Phase Function

A generic three-phase (a, b, c) balanced sinusoidal function with time-varying amplitude and frequency can be expressed as

$$f_{\mathrm{a}}(t) = \widehat{f}(t)\cos\left(\theta(t)\right), \tag{2.27a}$$

$$f_{\mathrm{b}}(t) = \widehat{f}(t)\cos\left(\theta(t) - \frac{2\pi}{3}\right) \quad \text{and} \tag{2.27b}$$

$$f_{\mathrm{c}}(t) = \widehat{f}(t)\cos\left(\theta(t) - \frac{4\pi}{3}\right), \tag{2.27c}$$

where

$$\theta(t) = \theta_0 + \int_0^{\mathrm{T}} \omega(\tau)d\tau, \tag{2.28}$$

$\widehat{f}(t)$ is the amplitude, and $\omega(t)$ is the angular frequency, all varying with respect to time. θ_0 is the initial phase angle of the function. The function could represent a three-phase signal or three time-varying parameters, e.g. inductances. For the sinusoidal function (2.27), the space phasor $\overrightarrow{f}$ is defined as

$$\overrightarrow{f}(t) \equiv \frac{2}{3}\left[\mathrm{e}^{\mathrm{j}0} f_{\mathrm{a}}(t) + \mathrm{e}^{\mathrm{j}\frac{2\pi}{3}} f_{\mathrm{b}}(t) + \mathrm{e}^{\mathrm{j}\frac{4\pi}{3}} f_{\mathrm{c}}(t)\right]. \tag{2.29}$$

By substituting for (2.27) in (2.29) and using the identities $\cos\theta = \frac{1}{2}\left(\mathrm{e}^{\mathrm{j}\theta} + \mathrm{e}^{-\mathrm{j}\theta}\right)$ and $\mathrm{e}^{\mathrm{j}0} + \mathrm{e}^{\mathrm{j}\frac{2\pi}{3}} + \mathrm{e}^{\mathrm{j}\frac{4\pi}{3}} = 0$, it is possible to write the space phasor as

$$\overrightarrow{f}(t) = \widehat{f}(t)\mathrm{e}^{\mathrm{j}\theta(t)}. \tag{2.30}$$

2.4.2 $\alpha\beta$- and dq-Reference Frames

The space phasor is a complex-valued function of time, conveniently expressed in the polar coordinate system. This kind of representation is suitable when the dynamics of amplitude and phase of the system are of interest. However, for the purposes of VSC control it is preferred to map space phasor equations to a Cartesian coordinate system. Such a system, commonly referred to as the $\alpha\beta$-frame, enables the manipulation of real-valued functions of time.

Still assuming a balanced three-phase system, i.e. that $f_\mathrm{a} + f_\mathrm{b} + f_\mathrm{c} = 0$ in (2.27), the space phasor in (2.29) can be decomposed into real and imaginary components as

$$\overrightarrow{f}(t) = f_\alpha(t) + \mathrm{j} f_\beta(t). \tag{2.31}$$

This abc/$\alpha\beta$-frame transformation can then be expressed as

$$\begin{bmatrix} f_\alpha(t) \\ f_\beta(t) \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} f_\mathrm{a}(t) \\ f_\mathrm{b}(t) \\ f_\mathrm{c}(t) \end{bmatrix}. \tag{2.32}$$

The $\alpha\beta$-frame offers a simplified framework in which to analyze balanced three-phase systems by reducing the system from three to two components. However, reference, feedback and feed-forward signals are generally still sinusoidal functions of time. The natural next step would then be to introduce a reference frame in which sinusoidal signals become constants under steady-state conditions. The dq-frame (direct-quadrature) is such a reference frame. Moreover, in the conventional abc-frame, a salient-pole synchronous machine representation contains time-varying self- and mutual inductances. In a proper dq-frame, these time-varying parameters appear as constants.

For the space phasor $\overrightarrow{f}(t) = f_\alpha(t) + \mathrm{j} f_\beta(t)$, the $\alpha\beta$/dq-transformation is defined as

$$f_\mathrm{d}(t) + \mathrm{j} f_\mathrm{q}(t) \equiv [f_\alpha(t) + \mathrm{j} f_\beta(t)]\, \mathrm{e}^{\mathrm{j}\varphi(t)}, \tag{2.33}$$

which is equivalent to a phase shift of $\overrightarrow{f}(t)$ by the angle $(-\varphi(t))$. From this it can be determined that the d- and q-components are orthogonal and lie along the same axes as the α- and β-components, only that the d- and q-axes are rotating. Usually, the rotational speed of the dq-frame is selected to be equal to that of the space phasor $\overrightarrow{f}$. In power system analysis this means that $\dot{\varphi} = \omega_\mathrm{s}$, where ω_s is the synchronous frequency of the system.

2.4.3 Global and Local dq-Frames

When modeling power systems, different parts of the model may use different dq-frames. Typically, network equations are formulated in a global, common reference frame, while machine equations are in a local reference frame, specific to the rotor. This presents the need for a method of converting between the different frames. Figure 2.2 shows the relative position of the machine-specific dq-frame and the global DQ-frame. The q-axis of a given generator is shifted with respect to the network's real axis D by the rotor angle δ [13].

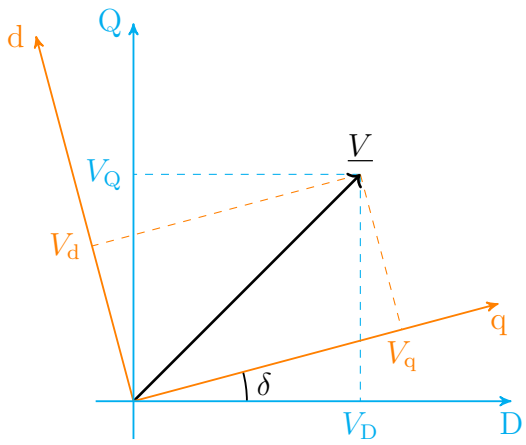


Figure 2.2: Relative position of the generator's rectangular (d, q) coordinates with respect to the network's complex (D, Q) coordinates, adapted from [13].

The relationship between the two coordinate systems is

$$\begin{bmatrix} V_\mathrm{D} \\ V_\mathrm{Q} \end{bmatrix} = \begin{bmatrix} -\sin\delta & \cos\delta \\ \cos\delta & \sin\delta \end{bmatrix} \begin{bmatrix} V_\mathrm{d} \\ V_\mathrm{q} \end{bmatrix} \quad \text{or} \quad \boldsymbol{V}_\mathrm{DQ} = \boldsymbol{T}\boldsymbol{V}_\mathrm{dq}. \tag{2.34a}$$

The transformation matrix $\boldsymbol{T}$ is unitary, i.e. $\boldsymbol{T}^{-1} = \boldsymbol{T}$. This means that, similar to the transformation from the local to the global reference frame, the transformation from the global to the local reference frame is given by

$$\begin{bmatrix} V_\mathrm{d} \\ V_\mathrm{q} \end{bmatrix} = \begin{bmatrix} -\sin\delta & \cos\delta \\ \cos\delta & \sin\delta \end{bmatrix} \begin{bmatrix} V_\mathrm{D} \\ V_\mathrm{Q} \end{bmatrix} \quad \text{or} \quad \boldsymbol{V}_\mathrm{dq} = \boldsymbol{T}\boldsymbol{V}_\mathrm{DQ}. \tag{2.34b}$$

2.5 Gear's Method

Gear's method is a numerical integration method for solving differential-algebraic equation (DAE) systems. The version used in this thesis is based on a second-order predictor-corrector scheme, *second-order* meaning that both the first and second derivatives of the state variables are used in the approximation. By monitoring the local truncation error, the step length is automatically adjusted to ensure the desired accuracy. This makes the method particularly suited for stiff equation systems, as the step length can be increased to reduce the computational intensity during steady-state or time spans with slow changing dynamics. When fast dynamics are dominant, the step length is automatically reduced so as to capture the fast response [6].

2.5.1 Method Overview

Figure 2.3 shows the steps and logic involved in Gear's method. Initially, the method requires a user-defined initial step length h_0 and initial conditions. An initial guess for the initial conditions is provided employing the Newton-Raphson load flow solution method described in Sections 2.3.1 or 2.3.2. This guess is then further iterated upon to ensure that the simulated system is in steady-state, i.e. that $\dot{\boldsymbol{y}} = \boldsymbol{0}$ in (2.4a).

Following this, initial predictions for the state variables $\boldsymbol{y}_{n+1}^{\mathrm{p}}$, along with the first and second derivatives $\dot{\boldsymbol{y}}_{n+1}^{\mathrm{p}}$ and $\ddot{\boldsymbol{y}}_{n+1}^{\mathrm{p}}$, for the time step $(n+1)$ are made.

The method then proceeds to the corrector part. Again using Newton-Raphson iterations, a deviation $\Delta\boldsymbol{y}_{n+1}$ between the predicted and accurate values is calculated. Based on this deviation, the local truncation error ε_{t} is calculated. The next steps are then determined depending on this value. If the truncation error is larger than the maximum permitted value, or well within the defined upper limit, the step length is adjusted. The first case is indicative of poor tracking, leading to a decreased step length. On the contrary, if the truncation error indicates that the step length is shorter than what is required to achieve the desired accuracy, it is increased. If the truncation error meets neither of these conditions, the algorithm jumps straight to the updating step.

However, if the case was that the truncation error was above the permitted limit, a new deviation $\Delta\boldsymbol{y}_{n+1}$ is calculated based on the reduced step length.

Once an appropriate deviation has been calculated, the prediction is updated to give the corrected values for the state variables $\boldsymbol{y}_{n+1}$ and its derivatives $\dot{\boldsymbol{y}}_{n+1}$ and $\ddot{\boldsymbol{y}}_{n+1}$ for the time step $(n+1)$.

The next step is updating the time variable t_n and iterating the step counter n. The process returns to the prediction step, looping through the described steps until a termination condition is met, typically that the time variable reaches the end of the set simulation duration.

Finally, the system eigenvalues are approximated numerically and the algorithm concludes, presenting the simulated state variables and derivatives for further manipulation, e.g. plotting.

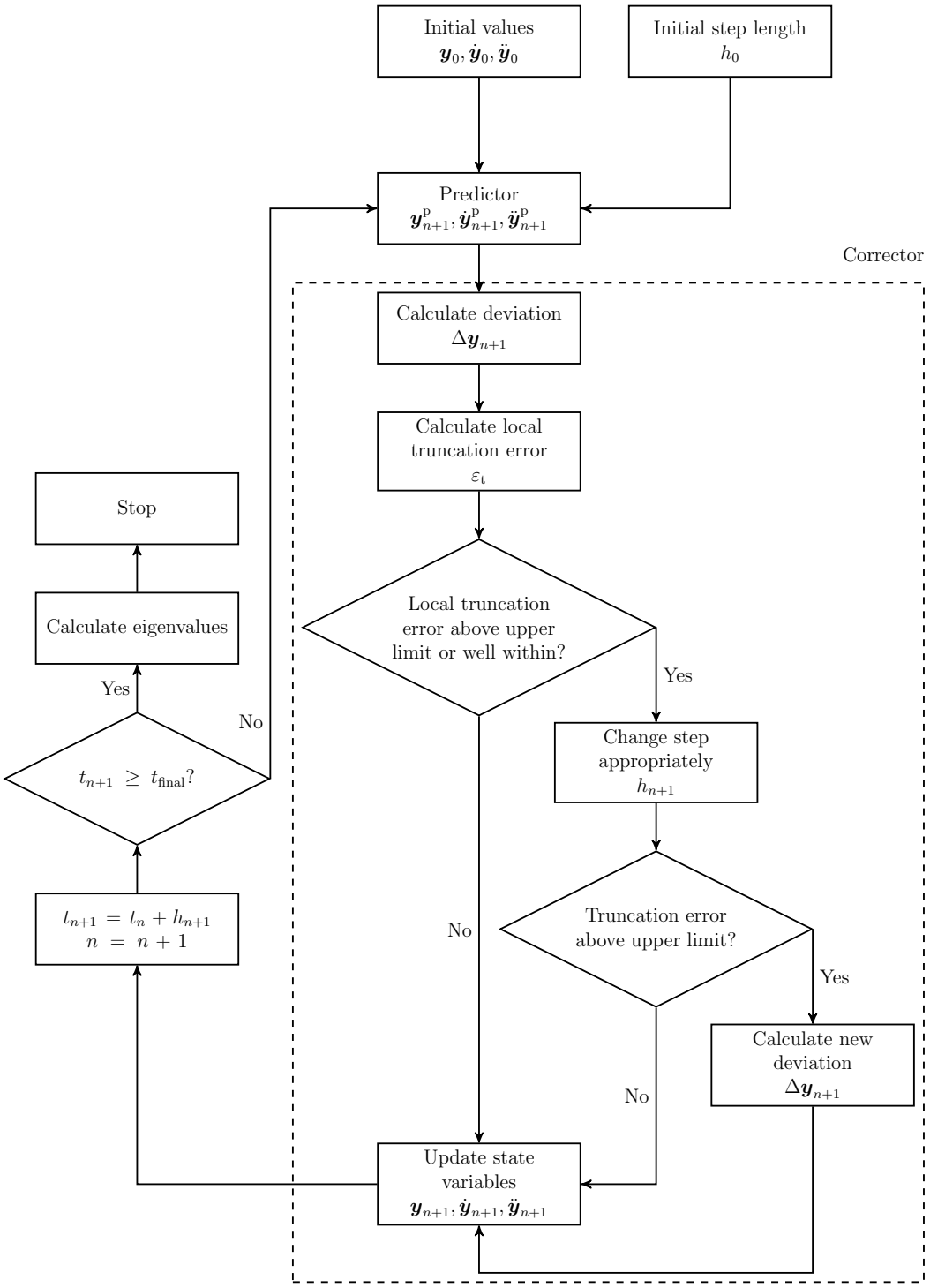


Figure 2.3: Flowchart for Gear's method, adapted from [4].

2.5.2 Method Description

The system equations are first arranged into the form in (2.4), with the state equation partitioned into differential variables $\boldsymbol{y}_{\mathrm{d}}$ and algebraic variables $\boldsymbol{y}_{\mathrm{a}}$ as in (2.5). This arrangement allows Gear's method to solve the K equations making up the DAE system, both the differential and algebraic.

Initial Conditions

It is vital that dynamic simulations start off with the appropriate initial conditions. As previously mentioned, the initial conditions for dynamic power system simulations of AC systems are normally found through a solved load flow case as described in Section 2.3.1. From this, the voltages and apparent powers at each bus are know, enabling the calculation of the injected currents. However, the voltage and current components are in the system's global reference frame, while the local versions are required in order to initialize each machine in the network. For this the rotor angle δ is required.

For round-rotor machines, the phasor $\underline{V}_{\mathrm{DQ}} + (R + \mathrm{j}X_{\mathrm{q}})\underline{I}_{\mathrm{DQ}}$ lies on the q-axis in steady-state [13, 22]. $\underline{V}_{\mathrm{DQ}}$ is the generator terminal voltage in the DQ-frame, $\underline{I}_{\mathrm{DQ}}$ is the delivered generator current, and R and X_{q} are the stator resistance and q-axis synchronous reactance of the generator, respectively. From this, the rotor angle between the real axis of the network frame and the machine q-axis can be determined. This enables the transformation of the terminal voltage and current into the machine-specific dq-frame, and the remaining states of the generator and other connected devices can be initialized.

However, due to propagated round-off errors, the system may not in fact be properly initialized to steady-state after this procedure. In order to solve this issue, the differential and algebraic equations in (2.4) are linearized, establishing the following equation system:

$$\begin{bmatrix} \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}_{\mathrm{d}}} & \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}_{\mathrm{a}}} \\ \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}_{\mathrm{d}}} & \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}_{\mathrm{a}}} \end{bmatrix}_{(l)} \begin{bmatrix} \Delta \boldsymbol{y}_{\mathrm{d}} \\ \Delta \boldsymbol{y}_{\mathrm{a}} \end{bmatrix}^{(l)} = \begin{bmatrix} -\Delta \boldsymbol{f} \\ -\Delta \boldsymbol{g} \end{bmatrix}^{(l)} \quad \text{or} \quad \boldsymbol{J}_{(l)} \Delta \boldsymbol{y}^{(l)} = \boldsymbol{D}^{(l)}, \tag{2.35}$$

where $\Delta \boldsymbol{y}^{(l)}$ is the linearized version of $\boldsymbol{y}^{(l)}$ for iteration step (l), not to be confused with the deviation $\Delta \boldsymbol{y}_{n+1}$.

In order to ensure that the system is initialized to steady-state, all the derivatives are set to zero, so the right hand side of (2.35), $\boldsymbol{D}^{(l)}$, becomes the negative of the derivatives of the differential equations and the negative of the deviations of the algebraic equations. More specifically,

$$\boldsymbol{D}^{(l)} = \begin{bmatrix} -\boldsymbol{f}(\Delta \boldsymbol{y}^{(l)}) \\ -\boldsymbol{g}(\Delta \boldsymbol{y}^{(l)}) \end{bmatrix}. \tag{2.36}$$

The initial conditions are then found iteratively by first calculating the mismatch vector $\boldsymbol{D}^{(l)}$ for the current $\Delta \boldsymbol{y}^{(l)}$. The result of the initialization based on the solved load flow is used as the initial guess for $\Delta \boldsymbol{y}^{(0)}$. After the mismatch has been calculated, a check is performed to determine if the largest element in the mismatch vector is smaller than the set Newton-Raphson tolerance. If the mismatch is sufficiently small at iteration step (l), the initial conditions are set as

$$\boldsymbol{y}_0 = \Delta \boldsymbol{y}^{(l)}. \tag{2.37}$$

Otherwise, the equation

$$\Delta \boldsymbol{y}^{(l+1)} = \Delta \boldsymbol{y}^{(l)} + \boldsymbol{J}_{(l)}^{-1} \boldsymbol{D}^{(l)}, \tag{2.38}$$

similar to (2.3), is solved and the process repeats until the mismatch is negligible.

Such an iterative initialization will also solve the potential issues caused by state saturation [22].

DC systems are initialized in a similar way. A load flow solution is found as described in Section 2.3.2, providing a starting point for the iterative process.

Prediction

Once the system is properly initialized, the first prediction step can be performed. Using the first and second derivatives of the state variables, as well as the step length h_{n+1} at time step t_n, the prediction for time step t_{n+1} is based on a second-order Taylor approximation of the state variables $\boldsymbol{y}$. The predicted values $\boldsymbol{y}^{\mathrm{p}}_{n+1}$, $\dot{\boldsymbol{y}}^{\mathrm{p}}_{n+1}$, and $\ddot{\boldsymbol{y}}^{\mathrm{p}}_{n+1}$ can be calculated as

$$\boldsymbol{y}^{\mathrm{p}}_{n+1} = \boldsymbol{y}_n + h_{n+1}\dot{\boldsymbol{y}}_n + \frac{h^2_{n+1}}{2!}\ddot{\boldsymbol{y}}_n, \tag{2.39a}$$

$$\dot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} = \dot{\boldsymbol{y}}_n + h_{n+1}\ddot{\boldsymbol{y}}_n \quad \text{and} \tag{2.39b}$$

$$\ddot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} = \ddot{\boldsymbol{y}}_n. \tag{2.39c}$$

Correction

According to [1], the state variables and their derivatives are related to the predicted values in the following way:

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}^{\mathrm{p}}_{n+1} + \left(\boldsymbol{y}_{n+1} - \boldsymbol{y}^{\mathrm{p}}_{n+1}\right), \tag{2.40a}$$

$$h_{n+1}\dot{\boldsymbol{y}}_{n+1} = h_{n+1}\dot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} + l_{1_{n+1}}\left(\boldsymbol{y}_{n+1} - \boldsymbol{y}^{\mathrm{p}}_{n+1}\right) \quad \text{and} \tag{2.40b}$$

$$\frac{h^2_{n+1}}{2!}\ddot{\boldsymbol{y}}_{n+1} = \frac{h^2_{n+1}}{2!}\ddot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} + l_{2_{n+1}}\left(\boldsymbol{y}_{n+1} - \boldsymbol{y}^{\mathrm{p}}_{n+1}\right). \tag{2.40c}$$

$l_{1_{n+1}}$ and $l_{2_{n+1}}$ are known constants dependent on step length and method order. For a second-order method [6] derives them as

$$l_{1_{n+1}} = \frac{2h_{n+1} + h_n}{h_{n+1} + h_n} \quad \text{and} \tag{2.41a}$$

$$l_{2_{n+1}} = \frac{h_{n+1}}{h_{n+1} + h_n}. \tag{2.41b}$$

Continuing, in order to calculate $\boldsymbol{y}_{n+1}$, an implicit function $\boldsymbol{U}(\boldsymbol{y}_{n+1})$ is defined, taking the difference between the approximated values of the first derivatives and their accurate values, i.e.

$$\boldsymbol{U}(\boldsymbol{y}_{n+1}) = \dot{\boldsymbol{y}}_{n+1} - \boldsymbol{f}(\boldsymbol{y}_{n+1}, t_{n+1}) = \boldsymbol{0}. \tag{2.42}$$

This way, the zeros of $\boldsymbol{U}$ give the state variables at step $(n+1)$. Multiplying (2.42) by the step length h_{n+1} and substituting for $\left(h_{n+1}\dot{\boldsymbol{y}}_{n+1}\right)$ using (2.40b) gives

$$\boldsymbol{U}(\boldsymbol{y}_{n+1}) = h_{n+1}\dot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} + l_{1_{n+1}}\left(\boldsymbol{y}_{n+1} - \boldsymbol{y}^{\mathrm{p}}_{n+1}\right) - h_{n+1}\boldsymbol{f}(\boldsymbol{y}_{n+1}, t_{n+1}) = \boldsymbol{0}. \tag{2.43}$$

However, the desired value is the mismatch between the predicted values and the accurate ones. By letting

$$\Delta \boldsymbol{y}_{n+1} = \boldsymbol{y}_{n+1} - \boldsymbol{y}^{\mathrm{p}}_{n+1}, \tag{2.44}$$

$\boldsymbol{U}$ can be redefined as

$$\boldsymbol{U}(\Delta \boldsymbol{y}_{n+1}) = h_{n+1}\dot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} + l_{1_{n+1}}\Delta \boldsymbol{y}_{n+1} - h_{n+1}\boldsymbol{f}(\Delta \boldsymbol{y}_{n+1} + \boldsymbol{y}^{\mathrm{p}}_{n+1}, t_{n+1}) = \boldsymbol{0}. \tag{2.45}$$

Similar to the initial values, the zeroes of $\boldsymbol{U}$ are found using Newton-Raphson iterations. Equation (2.45) is linearized, higher order terms are dropped, and is combined with the linearized version of the algebraic equations, $\Delta \boldsymbol{g}$, to give

$$\begin{bmatrix} \boldsymbol{L}_{1_{n+1}} - h_{n+1}\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}_{\mathrm{d}}} & -h_{n+1}\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}_{\mathrm{a}}} \\ \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}_{\mathrm{d}}} & \dfrac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}_{\mathrm{a}}} \end{bmatrix}_{(l)} \begin{bmatrix} \Delta \boldsymbol{y}_{\mathrm{d}} \\ \Delta \boldsymbol{y}_{\mathrm{a}} \end{bmatrix}^{(l)} = \begin{bmatrix} -\Delta \boldsymbol{U} \\ -\Delta \boldsymbol{g} \end{bmatrix}^{(l)} \quad \text{or} \quad \boldsymbol{A}_{(l)}\Delta \boldsymbol{y}^{(l)} = \boldsymbol{\Delta}^{(l)}, \tag{2.46}$$

where $\boldsymbol{L}_{1_{n+1}}$ is a diagonal matrix with all coefficients equal to $l_{1_{n+1}}$. As (2.46) is similar to (2.35), the same iterative solution method is employed. Equations (2.36) and (2.38) become respectively

$$\boldsymbol{\Delta}^{(l)} = \begin{bmatrix} -\boldsymbol{U}(\Delta \boldsymbol{y}^{(l)}) \\ -\boldsymbol{g}(\Delta \boldsymbol{y}^{(l)}) \end{bmatrix} \tag{2.47}$$

and

$$\Delta \boldsymbol{y}^{(l+1)} = \Delta \boldsymbol{y}^{(l)} + \boldsymbol{A}^{-1}_{(l)}\boldsymbol{\Delta}^{(l)}. \tag{2.48}$$

As an initial guess,

$$\Delta \boldsymbol{y}^{(0)} = \boldsymbol{y}^{\mathrm{p}}_{n+1} \tag{2.49}$$

is used.

Once the Newton-Raphson iterations have converged at iteration step (l) and $\Delta \boldsymbol{y}_{n+1} = \Delta \boldsymbol{y}^{(l)}$ has been determined, assuming an acceptable truncation error, (2.40) can be updated, using the redefinition in (2.44), according to:

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}^{\mathrm{p}}_{n+1} + \Delta \boldsymbol{y}_{n+1}, \tag{2.50a}$$

$$h_{n+1}\dot{\boldsymbol{y}}_{n+1} = h_{n+1}\dot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} + l_{1_{n+1}}\Delta \boldsymbol{y}_{n+1} \quad \text{and} \tag{2.50b}$$

$$\frac{h^2_{n+1}}{2!}\ddot{\boldsymbol{y}}_{n+1} = \frac{h^2_{n+1}}{2!}\ddot{\boldsymbol{y}}^{\mathrm{p}}_{n+1} + l_{2_{n+1}}\Delta \boldsymbol{y}_{n+1}. \tag{2.50c}$$

Local Truncation Error

Having determined the deviation $\Delta \boldsymbol{y}_{n+1}$, but before updating the state variables, the local truncation error ε_{t} is calculated. Reference [23] gives

$$\varepsilon_{\mathrm{t}_{n+1}} = K_{q_{n+1}} q! l_{q_{n+1}} \left\| \Delta \boldsymbol{y}_{n+1} \right\|_{\infty}, \tag{2.51}$$

where $\left\| \Delta \boldsymbol{y}_{n+1} \right\|_{\infty}$ is the infinity norm of $\Delta \boldsymbol{y}_{n+1}$, defined as

$$\left\| \Delta \boldsymbol{y}_{n+1} \right\|_{\infty} \equiv \max_{\{i=0,1,\ldots,k\}} \left| \Delta y_{i_{n+1}} \right|. \tag{2.52}$$

q is the order of the method, here $q = 2$, and $K_{q_{n+1}}$ is a constant. According to [6] $K_{2_{n+1}}$ is given by

$$K_{2_{n+1}} = \frac{1}{6} \cdot \frac{(h_{n+1} + h_n)^2}{h_{n+1}(2h_{n+1} + h_n)}. \tag{2.53}$$

$l_{q_{n+1}} = l_{2_{n+1}}$ is as defined in (2.41b).

Step Length Adjustment

According to [6] a step length change will give a truncation error ε of

$$\varepsilon = K_{q_{n+1}} q! l_{q_{n+1}} \left\| \Delta \boldsymbol{y}_{n+1} \right\|_\infty \left(\frac{h'_{n+1}}{h_{n+1}} \right)^q, \tag{2.54}$$

where h'_{n+1} is the new step length for step $(n+1)$ and h_{n+1} is the old one. Letting $\widehat{\varepsilon}$ be the maximum permitted truncation error and dividing (2.54) by (2.51) gives the adjusted step length as

$$h'_{n+1} = h_{n+1} \left(\frac{\widehat{\varepsilon}}{\varepsilon_{\mathrm{t}_{n+1}}} \right)^{\frac{1}{q}}, \tag{2.55}$$

or for a second-order method:

$$h'_{n+1} = h_{n+1} \sqrt{\frac{\widehat{\varepsilon}}{\varepsilon_{\mathrm{t}_{n+1}}}}. \tag{2.56}$$

Depending on if the current truncation error is larger or smaller than the maximum permitted truncation error, (2.56) will adjust the step length accordingly.

Eigenvalue Calculations

The final step before the algorithm concludes is to numerically approximate the eigenvalues of the simulated system. In order to do this, a system matrix without the algebraic variables must be established. Still, the effects the algebraic variables have on the interactions between the differential variables must be maintained. This is done by forming an $(N \times N)$ matrix, $\boldsymbol{G}(\boldsymbol{y}, t)$, from the elements in (2.35) as

$$\boldsymbol{G}(\boldsymbol{y}, t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}_\mathrm{d}} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}_\mathrm{a}} \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}_\mathrm{a}} \right)^{-1} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}_\mathrm{d}} \tag{2.57}$$

and evaluating it at the time step corresponding to $t = t_{\text{final}}$.

2.5.3 Step Length Strategies

Reference [6] suggests that merely following the step length adjustment strategy presented in Section 2.5.2, without any further safeguarding, may lead to the integration failing. This observation is supported by [4, 16]. It is therefor suggested to implement specific strategies for increasing and decreasing the step length.

If the local truncation error is found to be well within the permitted upper limit, or exceeds it, the step length is adjusted. However, as the step length is included in the correction matrix $\boldsymbol{A}_{(l)}$, changing the step length for each step would require a complete rebuild. It is therefore imperative to limit the number of times the step length is changed.

Increasing Step Length

There are several challenges associated with excessive step length increases. If the step length increases too much in one step, the likelihood of the next step being rejected due to poor state variable tracking increases. As the resource use of a failed step equals that of a successful one, it is desirable to limit the number of failed integration steps.

Furthermore, unstable simulation cases may be simulated as stable. If a disturbance brings about well-damped, fast oscillations together with slow, but unstable ones, the effect may be masked by a too large step length. The step length may have increased before any unstable modes become prevalent, giving the impression of a stable time-response. Having tight restrictions on the permitted truncation error will possibly handle these cases, but the number of corrector iterations required will increase.

Based on these concerns, the following strategies are proposed:

1. Applying a user-supplied scaling factor γ to the step length adjustment formula in (2.56), resulting in
$$h'_{n+1} = \gamma \cdot h_{n+1} \sqrt{\frac{\widehat{\varepsilon}}{\varepsilon_{t_{n+1}}}}. \tag{2.58}$$
Reference [6] suggests keeping the scaling factor in the range of 0.6 to 0.9. γ can effectively be viewed as a working tolerance compared to the maximum local truncation error.

2. Preventing the step length from increasing more than twofold between successive adjustments, i.e.
$$h'_{n+1} = \min\left\{h'_{n+1}, 2h_{n+1}\right\}. \tag{2.59}$$

3. Keeping the step length fixed for a user-specified number of steps. References [4, 6] suggest keeping the step length constant for 15 steps before it is allowed to be adjusted.

Decreasing Step Length

Non-linearities and discontinuities, such as state saturation and discrete events, in the simulated system may force the method to adopt a smaller step length than what is in fact necessary. One strategy to prevent this, as suggested by [6], is:

4. Preventing the step length from decreasing by more than a factor of two from one adjustment to the next:
$$h'_{n+1} = \max\left\{h'_{n+1}, \frac{h_{n+1}}{2}\right\}. \tag{2.60}$$

It is worth noting that truly discrete events are not possible when employing numerical integration methods. This is due to the fact that the system states before and after the discontinuity have to be separated by at least one step length.

Chapter 3

Implementation and Test Systems

The following chapter describes the implementation of Gear's method for four different power system cases. The first case is a simple system consisting of one generation unit connected through a transmission line to an infinite bus. The generation unit consists of a synchronous generator, exciter, hydro turbine and governor, and a power system stabilizer (PSS).

The second case is a more complex system, consisting of three generation units, connected to an infinite bus via a six-bus system, including dynamic loads. The third case incorporates more power electronic devices, consisting of a four-bus system with two generation units, a dynamic load, and a static compensator (STATCOM). The final system is made up of three voltage source converters (VSCs), each connected to an infinite bus, interconnected through a DC network.

Initially, an overview of the structure of the Python implementation is given. Following this, the method of verification using MATLAB is described, before the modeling of each of the simulated cases is presented.

As in [16], each strategy was tested separately and in combination so as to better understand their effects on the method. This is in contrast to previous implementations such as [4, 6], where the strategies have been successfully employed, but where only comparisons between applying all the strategies and applying none have been made.

3.1 Python Implementation

The Python implementation of Gear's method employs an object-oriented approach relying on modularization. In addition to the base Python platform, extensive use of additional packages and libraries was made. Numpy [20] was used for its scientific and numerical computation functions and classes, SciPy [21] was used for its linear algebra capabilities, and SymPy [24] was used for symbolic mathematics. In addition, Matplotlib [8] was used for visualization and plotting.

Each of the four cases had slight variations in how they were implemented. The overall structure, however, was shared and is shown in Figure 3.1.

The `main()` function is responsible for defining the necessary system parameters,

as well as calling the other required functions to run the simulation. This includes calling either the `newtonRaphsonLF()` function or the `DCsteadyState()` function for solving the initial load flow of the system. In addition, `main()` calls the constructors of the network objects, such as generators, VSCs, and the network itself. Finally, `main()` is responsible for constructing the `GearsMethod` object and subsequently calling its `simulate()` and `plot()` methods.

`newtonRaphsonLF()` takes the system admittance matrix, the stated injected bus powers, an initial guess for the state variables, and a maximum permitted error, and employs the method described in Section 2.3.1 to provide a steady-state solution to the system. It returns the bus voltage magnitudes and angles, net active and reactive bus power injections, and a boolean variable to indicate if the method converged. `DCsteadyState()` is similar and takes in the number of system nodes, the node admittance and node connection matrices, the injected active power of the non-slack nodes, the node voltage references, and an initial guess of the state variables. The function returns node voltages, injected power at the slack node, and injected currents and branch currents, using the method described in Section 2.3.2. Throughout this thesis a maximum permitted error of 1×10^{-4} was used for the steady-state solvers.

The network objects are all defined based on the same principles. Their constructors take in the necessary parameters, e.g. time constants, controller gains and saturation limits. They then create symbolic variables representing their own state variables. Each class contains an `updateSystemEquations()` method used to update, or create when the constructor is called, symbolic representations of the differential or algebraic equations of the object. If any interface variables are needed, local versions are created so as to not interfere with their original definition. An example of this is how a `Network` object requires the rotor angle of the connected synchronous generators. The system as a whole will only have access to the rotor angle variable defined by the `Generator` object, not the one defined by the `Network` object.

The `GearsMethod` class forms the core of the Python implementation. Its constructor takes in the solved load flow case, the objects required to construct the system, and the parameters required to run the simulation. This includes information about when a disturbance is applied and the strategies employed. The constructor then defines the necessary variables, such as the state vector. In addition, it calls the class' own `updateSystemEquations()` method. Here the symbolic variables and equations defined by the system objects are transformed into lambda functions which are used to calculate numerical values, i.e $\boldsymbol{f}(\boldsymbol{y}, t)$ and $\boldsymbol{g}(\boldsymbol{y}, t)$ in (2.4). This includes the derivatives of the Jacobian and correction matrices.

By first defining the system equations as symbolic equations, the SymPy library can be used to handle the required differentiation of the system equations, as doing so by hand would be highly inconvenient for systems of the size implemented here.

The `simulate()` method calls the two initialization methods, one using the result from the load flow and the other using iterations, and loops through the methods for handling disturbances, prediction, correction, saturation, and time variable updating. The loop is broken once the time variable equals or exceeds a final, predefined value. In addition to calling the method for calculating the eigenvalues, `simulate()` calls the method for verification and prints data about the simulation.

`initializeLF()` takes the result from the load flow case, calculates the voltage and current components in the required reference system, and provides an initial initialization of the system. `initializeDC()` functions in a similar manner, only adapted for DC systems. Following this, `initializeNR()` iterates on the equation system in (2.35) to set the final initial values of the system. The lambda functions `f()` and `g()` are used to return the results of the differential and algebraic equations for the current $\Delta \boldsymbol{y}^{(l)}$. Similarly, `J()` returns the Jacobian matrix $\boldsymbol{J}_{(l)}$ using the lambda functions `dfdyd()`, `dfdya()`, `dgdyd()` and `dgdya()`. Each of them return a submatrix of the Jacobian based on the current state vector.

The `disturbance()` method makes the appropriate changes to the system at the specified disturbance instance, e.g. changing system constants. If applicable, `post-Disturbance()` acts in a similar way to put the system in the proper state after a disturbance has been corrected.

Next, the `predictor()` method calculates the predicted values according to (2.39) and updates the prediction vectors for the state variables and derivatives.

Continuing, `corrector()` makes an initial guess for the deviation $\Delta \boldsymbol{y}_{n+1}$ and calls `newtonRaphson()` to set the appropriate values. It then calls `truncationError()` to calculate ε_{t}, and based on this it makes a decision to either update the state vector directly, or perform a step length adjustment. `changeStep()` is called if the calculated truncation error is above the upper limit, or if it is below half of the upper limit, as this is considered being well within the constraints. Based on the recommendations in [6] the upper limit for the truncation error was set relatively tightly at 1×10^{-5}. If the truncation error is above the limit, `corrector()` repeats the call to `newtonRaphson()`, but with the adjusted step length. Finally, the state variables and derivatives are updated according to (2.50).

`newtonRaphson()` takes the initial guess $\Delta \boldsymbol{y}^{(0)}$ passed from the `corrector()` method and enters a loop. In the loop, `U()` and the lambda function `g()` return the deviation vector $\boldsymbol{\Delta}^{(l)}$ according to (2.47). If the largest element in the deviation vector is above the limit set for convergence, $\Delta \boldsymbol{y}^{(l+1)}$ is calculated according to (2.48) and the steps are looped through again. The loop breaks when the deviation vector is sufficiently small and $\Delta \boldsymbol{y}_{n+1}$ is returned to the `corrector()` method. The convergence limit was set at 1×10^{-5}.

The `updateTime()` method simply appends the current simulation time to the end of the time vector, iterates the step variable n, and adds the current step length to the end of the step length vector. `eigenvalues()` approximates the system eigenvalues and eigenvectors based on the lambda functions `dfdyd()`, `dfdya()`, `dgdya()`, and `dgdyd()`.

After the simulation has concluded, `simulate()` calls the `verification()` method. Initially, `verification()` reads the result of the verification simulation from a text file. Following this, it interpolates between the verification points to ensure that the simulation and verification results are defined for the same time values, so as to make them comparable. The final function of `verification()` is to calculate the

mean squared error (MSE) of each state variable according to

$$MSE(\boldsymbol{y}, \widetilde{\boldsymbol{y}}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \widetilde{y}_i)^2, \tag{3.1}$$

where n is the time step index as before, y_i is the simulated value and $\widetilde{y}_i$ is the verified value for time step i [19]. The MSE will be unit-less as pu-values are used in the calculation. `verification()` then prints the state variable with the highest MSE along with the corresponding MSE value.

Lastly, `plot()` is called. The method uses the Matplotlib library to create a plot of the state variables of interest with respect to the time vector. The same is done for the step length vector, in addition to calling the `movingAverage()` method which returns the moving average of the step length vector. If saturation has been enforced during the simulation, a plot showing the affected time steps is displayed. `plot()` also plots the verified state variables, if required, and the system eigenvalues in the complex plane.

3.1.1 Verification Method

MATLAB was used to perform the verification calculations. The DAE system was defined and converted into function handles to be solved by MATLAB's ordinary differential equation solver `ode15i`, a variable-order method for solving fully implicit differential equations [15]. The results of the Python methods `initializeLF()` or `initializeDC()` were used as initial guesses for the MATLAB initializer. After solving the system, the results were written to a text file and made accessible to the Python program.

In addition to visual verification, the MSE for each state variable was calculated using (3.1).

3.1.2 Performance Metrics

When comparing the effects of the different strategies, the number of step length adjustments and mean step length were used in addition to the maximum MSE as performance metrics. As noted in Section 2.5.3, avoiding an excessive step adjustment frequency reduces the computational intensity of the method. It was therefor a goal to keep the number of step changes required to a minimum.

Furthermore, operating with a large mean step length indicates that few calculations are required to simulate a given time span. Achieving a large mean step length was therefore also desirable.

In order to determine the parameter combination resulting in the best performance across the three criteria, the Euclidean distances between the achieved and the ideal scores were used [5]. The Euclidean distances were calculated as

$$d(\boldsymbol{p}, \bar{\boldsymbol{p}}) = \left[\sum_{m=1}^{3}\left(\frac{p_m}{\bar{p}_m} - 1\right)^2\right]^{\frac{1}{2}}, \tag{3.2}$$

where for a given parameter combination, $\boldsymbol{p}$ is the vector of achieved performance scores, and $\bar{\boldsymbol{p}}$ is the vector of the ideal values. m indicates each of the three criteria. The scores have each been normalized so as to ensure equal weighting, and a smaller distance indicated an overall better score than a large distance. As with the MSE, the Euclidean distance will be unit-less.

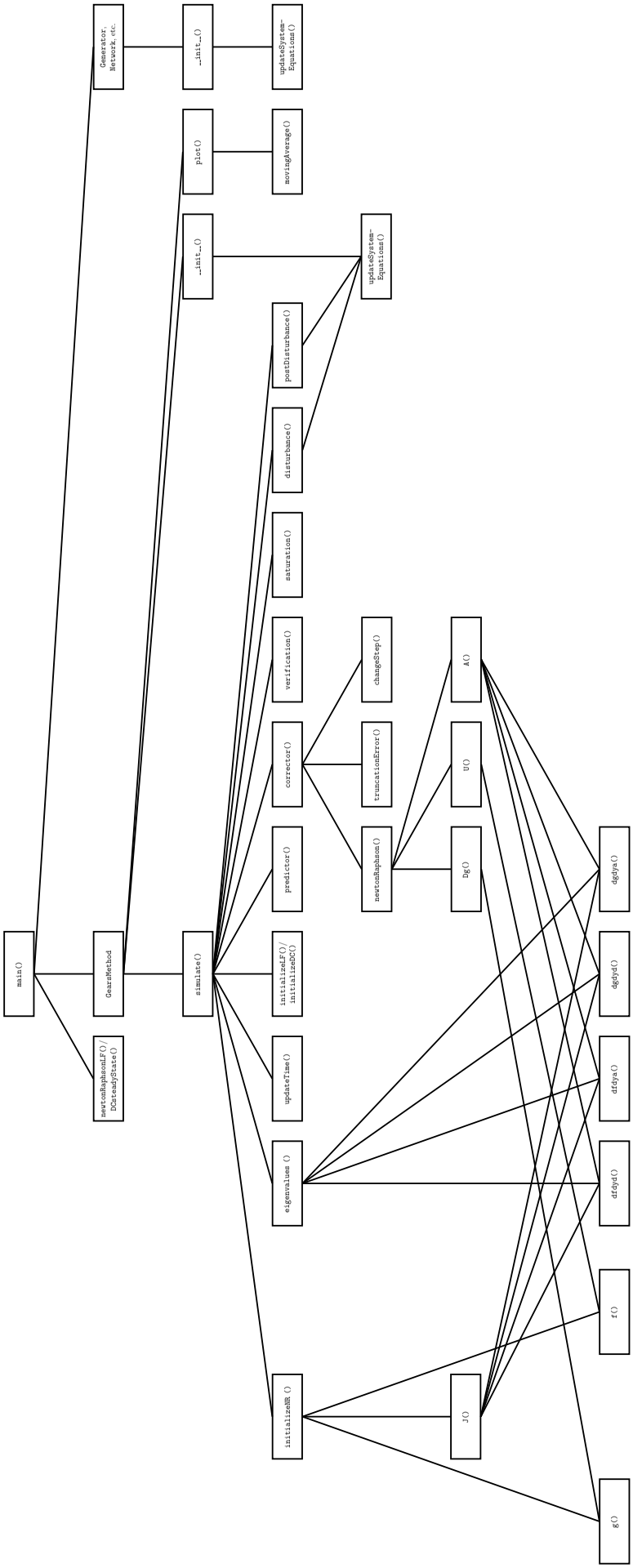


Figure 3.1: Structure of Python implementation of Gear's method.

3.2 Description of Single Machine to Infinite Bus System

The first simulation case is based on the system in Figure 3.2, and forms a basic building block for several of the following cases. It consist of a synchronous generator, an exciter, a hydro turbine and its governor, and a power system stabilizer. Together these components form a single generating unit, which again is connected to an infinite bus via a transmission line (SMIB).

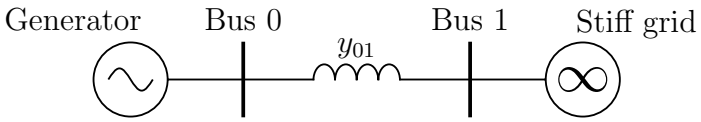


Figure 3.2: Single-line diagram of SMIB system.

The system as a whole is defined by 21 state variables, divided between 17 differential and four algebraic. The differential and algebraic state vectors are

$$\boldsymbol{y}_\mathrm{d} = \begin{bmatrix} \Delta\omega_\mathrm{s} & \delta & E'_\mathrm{q} & E'_\mathrm{d} & E''_\mathrm{q} & E''_\mathrm{d} & E_\mathrm{f} & V_\mathrm{r} & V_\mathrm{b} & V_\mathrm{f} & V_\mathrm{t} & V_\mathrm{w} & V_\mathrm{pss} & V_\mathrm{g} & V_\mathrm{p} & V_\mathrm{s} & P_\mathrm{m} \end{bmatrix}^\mathrm{T} \tag{3.3a}$$

and

$$\boldsymbol{y}_\mathrm{a} = \begin{bmatrix} V_{\mathrm{D}_0} & V_{\mathrm{Q}_0} & V_{\mathrm{D}_1} & V_{\mathrm{Q}_1} \end{bmatrix}^\mathrm{T}, \tag{3.3b}$$

respectively. $\Delta\omega_\mathrm{s}$, δ, E'_q, E'_d, E''_q, and E''_d relate to the synchronous generator model. $\Delta\omega_\mathrm{s}$ represents the rotor angle deviation from synchronous frequency, i.e. $\Delta\omega_\mathrm{s} = \omega - \omega_\mathrm{s}$, ω_s being the nominal system angular frequency. Throughout this thesis, 50 Hz is used as system frequency, resulting in $\omega_\mathrm{s} = 100\pi$ rad/s. The remaining variables represent the rotor angle, and the q- and d-axis components of the transient and subtransient internal emfs. E_f is the excitation emf, V_r is the regulator output voltage, V_b is the regulator input voltage, V_f is the feedback stabilization signal, and V_t is the output voltage from the transducer measuring the terminal voltage. V_w and V_pss define the PSS and represent the filtered frequency deviation and stabilization signals. The final four state variables, V_g, V_p, V_s, and P_m, are related to the turbine and its governor. They represent the gate signal from the governor, the rate of change of the gate signal, the transient droop variable and the mechanical output power of the turbine, respectively.

The algebraic variables are the D- and Q-components of the two bus voltages, referred to the global frame.

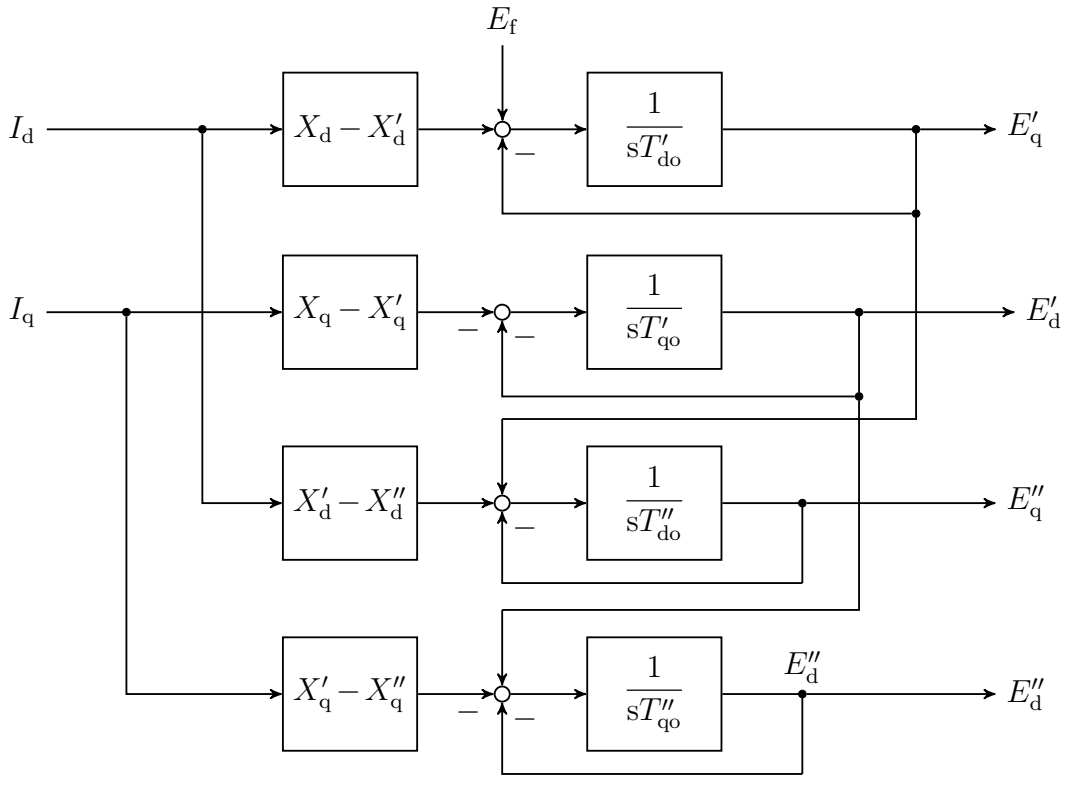
The simulated disturbance is an increase of the voltage reference V_ref by +1 %.

3.2.1 Synchronous Generator in Single Machine to Infinite Bus System

The synchronous generator is modeled using a sixth-order model [13]. The differential equations are

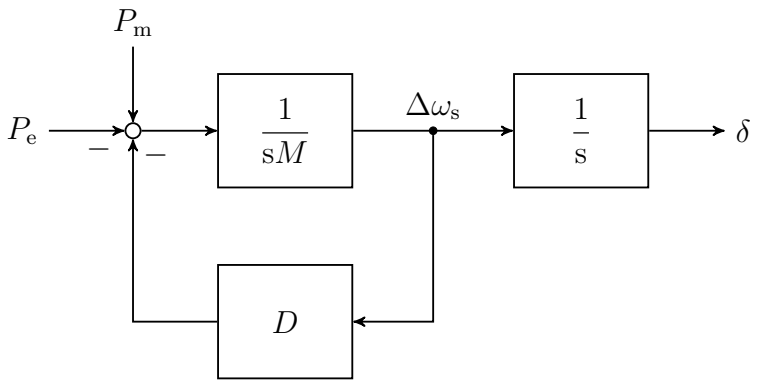
$$\begin{bmatrix} f_0(\boldsymbol{y},t) \\ f_1(\boldsymbol{y},t) \\ f_2(\boldsymbol{y},t) \\ f_3(\boldsymbol{y},t) \\ f_4(\boldsymbol{y},t) \\ f_5(\boldsymbol{y},t) \end{bmatrix} = \begin{bmatrix} \dot{\Delta\omega}_\mathrm{s} \\ \dot{\delta} \\ \dot{E}'_\mathrm{q} \\ \dot{E}'_\mathrm{d} \\ \dot{E}''_\mathrm{q} \\ \dot{E}''_\mathrm{d} \end{bmatrix} = \begin{bmatrix} \frac{1}{M}\left(P_\mathrm{m} - P_\mathrm{e} - D\Delta\omega_\mathrm{s}\right) \\ \Delta\omega_\mathrm{s} \\ \frac{1}{T'_\mathrm{do}}\left[E_\mathrm{f} - E'_\mathrm{q} + I_\mathrm{d}\left(X_\mathrm{d} - X'_\mathrm{d}\right)\right] \\ \frac{1}{T'_\mathrm{qo}}\left[-E'_\mathrm{d} - I_\mathrm{q}\left(X_\mathrm{q} - X'_\mathrm{q}\right)\right] \\ \frac{1}{T''_\mathrm{do}}\left[E'_\mathrm{q} - E''_\mathrm{q} + I_\mathrm{d}\left(X'_\mathrm{d} - X''_\mathrm{d}\right)\right] \\ \frac{1}{T''_\mathrm{qo}}\left[E'_\mathrm{d} - E''_\mathrm{d} - I_\mathrm{q}\left(X'_\mathrm{q} - X''_\mathrm{q}\right)\right] \end{bmatrix}, \tag{3.4}$$

where M is the machine coefficient of inertia and D is the damping coefficient related to windage and friction losses. T'_do, T'_qo, T''_do, and T''_qo are the direct- and quadrature-axis transient and subtransient short-circuit time constants. X_d and X_q, X'_d and X'_q, and X''_d and X''_q are the direct- and quadrature-axis synchronous, transient and subtransient reactances of the generator. The subsystem only consists of differential equations, and is illustrated in block diagram form in Figure 3.3. s is the Laplace operator.



(a) Electrical part.

Figure 3.3: Block diagram representation of synchronous generator in SMIB case.



(b) Mechanical part.

Figure 3.3: Block diagram representation of synchronous generator in SMIB case *(continued)*.

The description in (3.4) contains the current components I_d and I_q. However, the network description is voltage based, so the currents have to be eliminated.

By neglecting the armature resistance, the relationship between the dq-components of the machine's terminal voltage $\underline{V}_\mathrm{dq}$ and current $\underline{I}_\mathrm{dq}$ can be expressed as

$$\begin{bmatrix} V_\mathrm{d} \\ V_\mathrm{q} \end{bmatrix} = \begin{bmatrix} E''_\mathrm{d} \\ E''_\mathrm{q} \end{bmatrix} - \begin{bmatrix} 0 & X''_\mathrm{q} \\ -X''_\mathrm{d} & 0 \end{bmatrix} \begin{bmatrix} I_\mathrm{d} \\ I_\mathrm{q} \end{bmatrix} \quad \text{or} \quad \boldsymbol{V}_\mathrm{dq} = \boldsymbol{E}''_\mathrm{dq} - \boldsymbol{Z}_\mathrm{dq}\boldsymbol{I}_\mathrm{dq}. \tag{3.5}$$

However, in (3.5) the voltage components appear in the local reference frame, while the global ones are used in the system description. By applying the transformation in (2.34a) to the voltage components and solving for the currents, (3.5) becomes

$$\begin{bmatrix} I_\mathrm{d} \\ I_\mathrm{q} \end{bmatrix} = \begin{bmatrix} \frac{1}{X''_\mathrm{d}}(-E''_\mathrm{q} + V_\mathrm{D}\cos\delta + V_\mathrm{Q}\sin\delta) \\ \frac{1}{X''_\mathrm{q}}(E''_\mathrm{d} + V_\mathrm{D}\sin\delta - V_\mathrm{Q}\cos\delta) \end{bmatrix} \quad \text{or} \quad \boldsymbol{I}_\mathrm{dq} = \boldsymbol{Z}_\mathrm{dq}^{-1}(\boldsymbol{E}''_\mathrm{dq} - \boldsymbol{T}\boldsymbol{V}_\mathrm{DQ}), \tag{3.6}$$

where $\boldsymbol{V}_\mathrm{dq} = \boldsymbol{T}\boldsymbol{V}_\mathrm{DQ}$.

Still by neglecting the armature resistance, the active power delivered by the generator is given by

$$P_\mathrm{e} = V_\mathrm{d}I_\mathrm{d} + V_\mathrm{q}I_\mathrm{q}. \tag{3.7}$$

The description of the synchronous generator is completed by substituting for the current and active power in (3.4) using (3.6) and (3.7), with the voltages transformed into the global frame. The machine parameters are given in per unit values in Table 3.1.

Table 3.1: Parameter values for synchronous generator in SMIB.

Parameter	Value [pu]
M	6.20
D	0.00
T'_{do}	6.00
T'_{qo}	0.54
T''_{do}	0.05
T''_{qo}	0.05
X_{d}	0.89
X_{q}	0.86
X'_{d}	0.30
X'_{q}	0.30
X''_{d}	0.10
X''_{q}	0.10

3.2.2 Excitation System in Single Machine to Infinite Bus System

The excitation system is based on a type DC1A model [9], more specifically the following differential equation system:

$$\begin{bmatrix} f_6(\boldsymbol{y},t) \\ f_7(\boldsymbol{y},t) \\ f_8(\boldsymbol{y},t) \\ f_9(\boldsymbol{y},t) \\ f_{10}(\boldsymbol{y},t) \end{bmatrix} = \begin{bmatrix} \dot{E}_{\mathrm{f}} \\ \dot{V}_{\mathrm{r}} \\ \dot{V}_{\mathrm{b}} \\ \dot{V}_{\mathrm{f}} \\ \dot{V}_{\mathrm{t}} \end{bmatrix} = \begin{bmatrix} \frac{1}{T_{\mathrm{e}}}[V_{\mathrm{r}} - (S_{\mathrm{e}} + K_{\mathrm{e}})E_{\mathrm{f}}] \\ \frac{1}{T_{\mathrm{a}}}(K_{\mathrm{a}}V_{\mathrm{b}} - V_{\mathrm{r}}) \\ \frac{1}{T_{\mathrm{b}}}(V_{\mathrm{ref}} + V_{\mathrm{pss}} - V_{\mathrm{b}} - V_{\mathrm{f}} - V_{\mathrm{t}}) \\ \frac{1}{T_{\mathrm{f}}}\left[\frac{K_{\mathrm{f}}}{T_{\mathrm{e}}}(V_{\mathrm{r}} - (S_{\mathrm{e}} + K_{\mathrm{e}})E_{\mathrm{f}}) - V_{\mathrm{f}}\right] \\ \frac{1}{T_{\mathrm{r}}}\left(\sqrt{V_{\mathrm{D}}^2 + V_{\mathrm{Q}}^2} - V_{\mathrm{t}}\right) \end{bmatrix}. \tag{3.8}$$

T_{e}, T_{a}, T_{b}, T_{f}, and T_{r} are different time constants and K_{e}, K_{a}, and K_{f} are static gains. V_{ref} is the voltage magnitude reference value and V_{D} and V_{Q} are the DQ-components of the terminal voltage. The voltage components appear directly in the equation describing the transducer voltage as no load compensation voltage is employed, i.e. $V_{\mathrm{c}} = \sqrt{V_{\mathrm{D}}^2 + V_{\mathrm{Q}}^2}$. Following the modeling done in [6], the effect of the saturation function S_{e} is neglected.

Figure 3.4 gives the exciter equations in block diagram form, and Table 3.2 gives the exciter parameters. $V_{\mathrm{r}}^{\max}$ and $V_{\mathrm{r}}^{\min}$ are respectively the maximum and minimum regulator output limits.

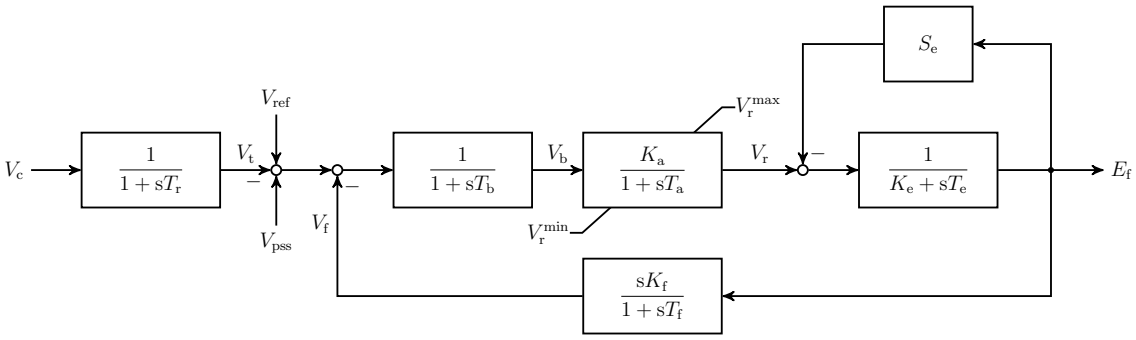


Figure 3.4: Block diagram representation of exciter in SMIB, adapted from [9].

Table 3.2: Parameter values for exciter in SMIB.

Parameter	Value [pu]
T_e	1.330
T_a	0.050
T_b	0.100
T_f	0.670
T_r	0.001
K_e	1.000
K_a	300.000
K_f	0.100
V_{ref}	1.000
V_r^{max}	20.000
V_r^{min}	-20.000
S_e	0.000

3.2.3 Power System Stabilizer in Single Machine to Infinite Bus System

The power system stabilizer is modeled by the differential equations

$$\begin{bmatrix} f_{11}(\boldsymbol{y},t) \\ f_{12}(\boldsymbol{y},t) \end{bmatrix} = \begin{bmatrix} \dot{V}_w \\ \dot{V}_{pss} \end{bmatrix} = \begin{bmatrix} \dfrac{K_{pss}}{M}(P_m - P_e) - \dfrac{1}{T_{pss}}V_w \\ \dfrac{1}{T_2}\left[\dfrac{K_{pss}T_1}{M}(P_m - P_e) + \left(1 - \dfrac{T_1}{T_{pss}}\right)V_w - V_{pss}\right] \end{bmatrix}, \quad (3.9)$$

where T_1 and T_2 are time constants of the phase compensator, and T_{pss} is the time constant of the washout filter [10]. K_{pss} is the static gain of the PSS and M is as before the coefficient of inertia of the machine. The delivered active power P_e is substituted for using (3.6) and (3.7). Figure 3.5 shows the block diagram and Table 3.3 gives the PSS parameters.

Note that Figure 3.5 has $\Delta\omega_s$ as its input, while $\Delta\omega_s$ has been substituted for using (3.4) in (3.9).

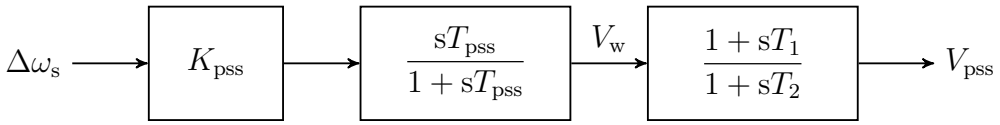


Figure 3.5: Block diagram representation of PSS in SMIB, adapted from [10].

Table 3.3: Parameter values for PSS in SMIB.

Parameter	Value [pu]
T_1	0.10
T_2	0.01
T_{pss}	3.00
K_{pss}	100.00

3.2.4 Hydro Turbine and Governor in Single Machine to Infinite Bus System

The fourth subsystem consists of a hydro turbine and accompanying governor, and is modeled by

$$\begin{bmatrix} f_{13}(\boldsymbol{y},t) \\ f_{14}(\boldsymbol{y},t) \\ f_{15}(\boldsymbol{y},t) \\ f_{16}(\boldsymbol{y},t) \end{bmatrix} = \begin{bmatrix} \dot{V}_g \\ \dot{V}_p \\ \dot{V}_s \\ \dot{P}_m \end{bmatrix} = \begin{bmatrix} V_p \\ \frac{1}{T_p}\left[-V_p + \frac{1}{T_g}\left(RP_{ref} - \frac{\Delta\omega_s}{\omega_s} - RV_g - V_s\right)\right] \\ \frac{1}{T_d}\left(D_d T_d V_p - V_s\right) \\ \frac{2}{T_w}\left(V_g - T_w V_p - P_m\right) \end{bmatrix}, \quad (3.10)$$

with T_g, T_p, T_d and T_w as time constants. P_{ref} is the active power reference of the governor and D_d and R are the transient and static droop coefficients [17]. Figure 3.6 illustrates the turbine and governor in block diagram form and Table 3.4 contains the parameters.

V_g^{max}, V_g^{min}, V_p^{max} and V_p^{min} are the maximum and minimum values of V_g and V_p, representing the limits on the gate position and the rate of change of the gate position.

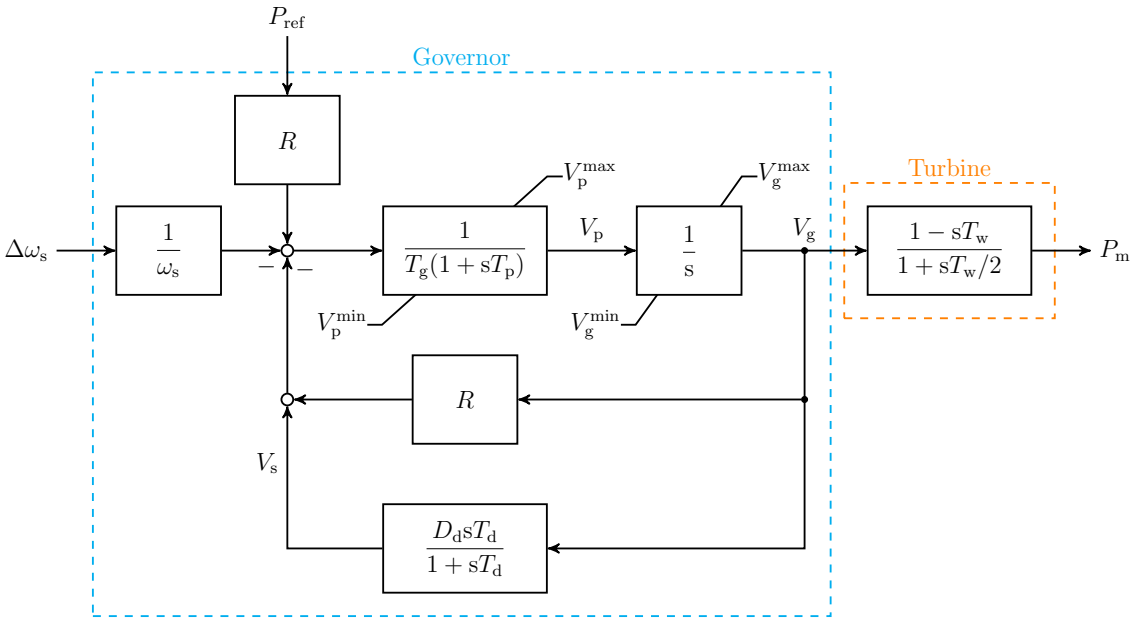


Figure 3.6: Block diagram representation of hydro turbine and governor in SMIB, adapted from [17].

Table 3.4: Parameter values for hydro turbine and governor in SMIB.

Parameter	Value [pu]
T_{g}	0.20
T_{p}	0.04
T_{d}	5.00
T_{w}	0.10
D_{d}	0.30
R	0.05
P_{ref}	1.00
$V_{\text{g}}^{\max}$	1.00
$V_{\text{g}}^{\min}$	0.00
$V_{\text{p}}^{\max}$	0.10
$V_{\text{p}}^{\min}$	-0.10

3.2.5 Transmission Network in Single Machine to Infinite Bus System

The system's network equations make up the algebraic part of the description. Based on (2.9) the two-bus system can be formulated as

$$\begin{bmatrix} I_{\text{D}_0} \\ I_{\text{Q}_0} \\ I_{\text{D}_1} \\ I_{\text{Q}_1} \end{bmatrix} = \begin{bmatrix} G_{00} & -B_{00} & G_{01} & -B_{01} \\ B_{00} & G_{00} & B_{01} & G_{01} \\ G_{10} & -B_{10} & G_{11} & -B_{11} \\ B_{10} & G_{10} & B_{11} & G_{11} \end{bmatrix} \begin{bmatrix} V_{\text{D}_0} \\ V_{\text{Q}_0} \\ V_{\text{D}_1} \\ V_{\text{Q}_1} \end{bmatrix}. \tag{3.11}$$

From (3.11) the equations for the two buses can be extracted as, keeping in mind that bus 1 is an infinite bus,

$$\begin{bmatrix} g_0(\boldsymbol{y},t) \\ g_1(\boldsymbol{y},t) \\ g_2(\boldsymbol{y},t) \\ g_3(\boldsymbol{y},t) \end{bmatrix} = \mathbf{0} = \begin{bmatrix} -I_{\mathrm{D}_0} + G_{00}V_{\mathrm{D}_0} - B_{00}V_{\mathrm{Q}_0} + G_{01}V_{\mathrm{D}_1} - B_{01}V_{\mathrm{Q}_1} \\ -I_{\mathrm{Q}_0} + B_{00}V_{\mathrm{D}_0} + G_{00}V_{\mathrm{Q}_0} + B_{01}V_{\mathrm{D}_1} + G_{01}V_{\mathrm{Q}_1} \\ V_{\mathrm{D}_1} - 1.0 \\ V_{\mathrm{Q}_1} \end{bmatrix}. \tag{3.12}$$

With this formulation $g_2(\boldsymbol{y},t)$ and $g_3(\boldsymbol{y},t)$ force $\underline{V}_1 = 1.0\angle 0.0°$.

From Figure 3.2 it can be seen that the injected current components I_{D_0} and I_{Q_0} at bus 0 equal the delivered current from the generation unit. The generator currents therefore have to be expressed in the global DQ-frame as functions of the state variables in (3.3).

Starting from (3.6), applying the transformation in (2.34a) to the current components, and solving for their global versions yields

$$\boldsymbol{I}_{\mathrm{DQ}} = \boldsymbol{T}^{-1}\boldsymbol{Z}_{\mathrm{dq}}^{-1}\boldsymbol{E}''_{\mathrm{dq}} - \boldsymbol{T}^{-1}\boldsymbol{Z}_{\mathrm{dq}}^{-1}\boldsymbol{T}\boldsymbol{V}_{\mathrm{DQ}}. \tag{3.13}$$

For ease of notation, two new transformations are defined as

$$\boldsymbol{Y}_{\mathrm{E}} \equiv \boldsymbol{T}^{-1}\boldsymbol{Z}_{\mathrm{dq}}^{-1} \quad \text{and} \tag{3.14a}$$

$$\boldsymbol{Y}_{\mathrm{V}} \equiv \boldsymbol{T}^{-1}\boldsymbol{Z}_{\mathrm{dq}}^{-1}\boldsymbol{T} = \boldsymbol{Y}_{\mathrm{E}}\boldsymbol{T}, \tag{3.14b}$$

making it possible to write (3.13) more compactly as

$$\boldsymbol{I}_{\mathrm{DQ}} = \boldsymbol{Y}_{\mathrm{E}}\boldsymbol{E}''_{\mathrm{dq}} - \boldsymbol{Y}_{\mathrm{V}}\boldsymbol{V}_{\mathrm{DQ}}. \tag{3.15}$$

Finally, the injected generator current components can be written as functions of state variables as

$$\begin{aligned} I_{\mathrm{D}}(\boldsymbol{y},t) = \frac{1}{X''_{\mathrm{d}}X''_{\mathrm{q}}} \big[& E''_{\mathrm{d}}X''_{\mathrm{d}}\cos\delta + E''_{\mathrm{q}}X''_{\mathrm{q}}\sin\delta \\ & - (V_{\mathrm{D}}/2)(X''_{\mathrm{q}} - X''_{\mathrm{d}})\sin 2\delta - V_{\mathrm{Q}}(X''_{\mathrm{q}}\sin^2\delta + X''_{\mathrm{d}}\cos^2\delta)\big] \end{aligned} \tag{3.16a}$$

and

$$\begin{aligned} I_{\mathrm{Q}}(\boldsymbol{y},t) = \frac{1}{X''_{\mathrm{d}}X''_{\mathrm{q}}} \big[& E''_{\mathrm{d}}X''_{\mathrm{d}}\sin\delta - E''_{\mathrm{q}}X''_{\mathrm{q}}\cos\delta \\ & + V_{\mathrm{D}}(X''_{\mathrm{d}}\sin^2\delta + X''_{\mathrm{q}}\cos^2\delta) + (V_{\mathrm{Q}}/2)(X''_{\mathrm{q}} - X''_{\mathrm{d}})\sin 2\delta\big]. \end{aligned} \tag{3.16b}$$

Note that all the variables are index-less as they all refer to the same bus and correspondingly connected generator. Substituting (3.16) into (3.12) completes the equation description of the SMIB case.

Table 3.5 gives the branch admittance for the line between buses 0 and 1. From this the conductances G_{ij} and susceptances B_{ij} in (3.12) can be found according to the description in Section 2.3.

Table 3.5: Branch admittance for network in SMIB case.

Branch	Admittance [pu]
0-1	$0.397 - \mathrm{j}19.912$

3.2.6 Initialization of Single Machine to Infinite Bus System Based on Load Flow Solution

The steady-state load flow case can be solved as described in Section 2.3.1, with the stated net power injection being given in Table 3.6.

Table 3.6: Stated net bus power injection for SMIB case.

Bus number	Stated net active power injection [pu]	Stated net reactive power injection [pu]
0	1.0	0.1

From the load flow solution, the remaining system is initialized. First, the bus voltages are transformed from polar to rectangular DQ-components using

$$V_{\mathrm{D}_i} = V_i \cos\delta_i \quad \text{and} \quad V_{\mathrm{Q}_i} = V_i \sin\delta_i, \tag{3.17}$$

where V_i as before is the voltage magnitude and δ_i is the voltage angle at bus i. These voltage components can then be used directly to initialize the algebraic variables.

These global voltage components, along with the injected apparent power, determine the injected generator currents as given by (2.10). From this, their local versions are determined as described in Section 2.5.2, and δ is initialized as a necessary step. In addition, in steady-state, the synchronous generator rotates at synchronous frequency, meaning that there is no angular frequency deviation and $\Delta\omega_\mathrm{s} = 0.0$. By letting $\boldsymbol{f}(\boldsymbol{y}, t) = \mathbf{0}$ and using the relationships given in (3.5), the following equations can be developed to initialize each of the remaining 15 differential states:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_4 \\ y_5 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \Delta\omega_\mathrm{s} \\ \delta \\ E''_\mathrm{q} \\ E''_\mathrm{d} \\ E'_\mathrm{q} \\ E'_\mathrm{d} \end{bmatrix} = \begin{bmatrix} 0.0 \\ \delta \\ V_\mathrm{q} - X''_\mathrm{d} I_\mathrm{d} \\ V_\mathrm{d} + X''_\mathrm{q} I_\mathrm{q} \\ E''_\mathrm{q} - I_\mathrm{d}(X'_\mathrm{d} - X''_\mathrm{d}) \\ -I_\mathrm{q}(X_\mathrm{q} - X'_\mathrm{q}) \end{bmatrix}, \tag{3.18a}$$

$$\begin{bmatrix} y_{14} \\ y_{15} \\ y_{13} \\ y_{16} \end{bmatrix} = \begin{bmatrix} V_\mathrm{p} \\ V_\mathrm{s} \\ V_\mathrm{g} \\ P_\mathrm{m} \end{bmatrix} = \begin{bmatrix} 0.0 \\ D_\mathrm{d} T_\mathrm{d} V_\mathrm{p} \\ \frac{1}{R}\left(-V_\mathrm{p} T_\mathrm{g} - \frac{\Delta\omega_\mathrm{s}}{\omega_s} - V_\mathrm{s} + R P_\mathrm{ref}\right) \\ V_\mathrm{g} - T_\mathrm{w} V_\mathrm{p} \end{bmatrix}, \tag{3.18b}$$

$$\begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} V_\mathrm{w} \\ V_\mathrm{pss} \end{bmatrix} = \begin{bmatrix} \frac{K_\mathrm{pss} T_\mathrm{pss}}{M}(P_\mathrm{m} - V_\mathrm{d} I_\mathrm{d} + V_\mathrm{q} I_\mathrm{q}) \\ \frac{K_\mathrm{pss} T_1}{M}(P_\mathrm{m} - V_\mathrm{d} I_\mathrm{d} + V_\mathrm{q} I_\mathrm{q}) + \left(1 - \frac{T_1}{T_\mathrm{pss}}\right) V_\mathrm{w} \end{bmatrix}, \text{ and} \tag{3.18c}$$

$$\begin{bmatrix} y_6 \\ y_7 \\ y_8 \\ y_{10} \\ y_9 \end{bmatrix} = \begin{bmatrix} E_\mathrm{f} \\ V_\mathrm{r} \\ V_\mathrm{b} \\ V_\mathrm{t} \\ V_\mathrm{f} \end{bmatrix} = \begin{bmatrix} E'_\mathrm{q} - I_\mathrm{d}(X_\mathrm{d} - X'_\mathrm{d}) \\ (S_\mathrm{e} + K_\mathrm{e})E_\mathrm{f} \\ \frac{V_\mathrm{r}}{K_\mathrm{a}} \\ \sqrt{V_\mathrm{D}^2 + V_\mathrm{Q}^2} \\ V_\mathrm{ref} + V_\mathrm{pss} - V_\mathrm{b} - V_\mathrm{t} \end{bmatrix}. \tag{3.18d}$$

This forms an initial steady-state solution guess, which is further iterated upon as described in Section 2.5.2.

3.3 Description of Three Machines to Infinite Bus System

The second system is an expanded version of the SMIB case. It consist of three generators connected via a six-bus system to an infinite bus (3MIB), including two dynamic loads, and is based on the benchmark system presented in [12]. Figure 3.7 shows the topology of the system.

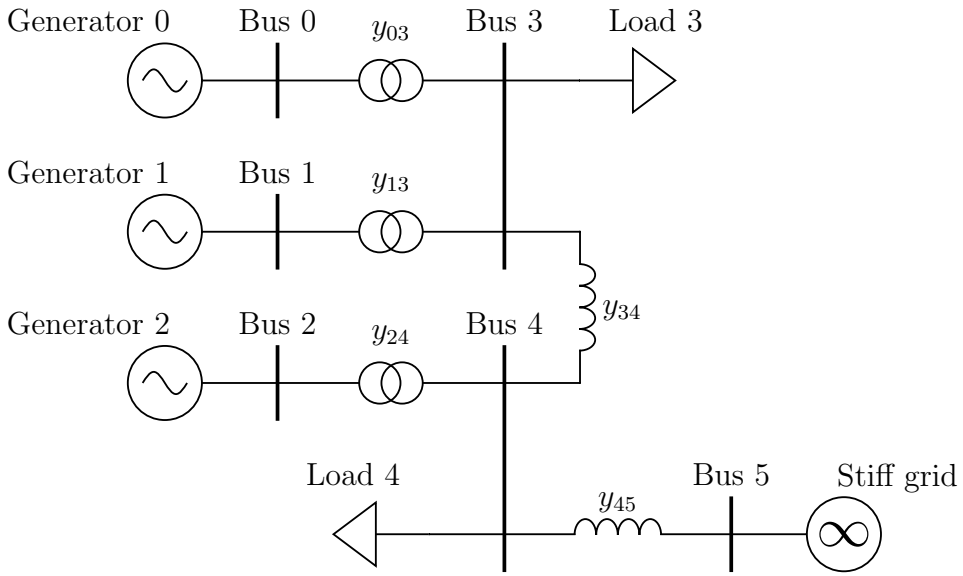


Figure 3.7: Single-line diagram of 3MIB system, adapted from [12].

The system is defined by 57 states, of which 45 are differential and 12 are algebraic. The three generators are defined similarly as in the SMIB case in Section 3.2, except that no PSS is included. The differential state vector is therefore

$$\boldsymbol{y}_\mathrm{d} = \begin{bmatrix} \boldsymbol{y}_{\mathrm{d}_0} \\ \boldsymbol{y}_{\mathrm{d}_1} \\ \boldsymbol{y}_{\mathrm{d}_2} \end{bmatrix}, \tag{3.19a}$$

where for $i = 0, 1, 2$:

$$\boldsymbol{y}_{\mathrm{d}_i} = \begin{bmatrix} \Delta\omega_{\mathrm{s}_i} & \delta_i & E'_{\mathrm{q}_i} & E'_{\mathrm{d}_i} & E''_{\mathrm{q}_i} & E''_{\mathrm{d}_i} & E_{\mathrm{f}_i} & V_{\mathrm{r}_i} & V_{\mathrm{b}_i} & V_{\mathrm{f}_i} & V_{\mathrm{t}_i} & V_{\mathrm{g}_i} & V_{\mathrm{p}_i} & V_{\mathrm{s}_i} & P_{\mathrm{m}_i} \end{bmatrix}^{\mathrm{T}}. \tag{3.19b}$$

The algebraic state variables are

$$\boldsymbol{y}_{\mathrm{a}} = \begin{bmatrix} V_{\mathrm{D}_0} & V_{\mathrm{Q}_0} & V_{\mathrm{D}_1} & V_{\mathrm{Q}_1} & V_{\mathrm{D}_2} & V_{\mathrm{Q}_2} & V_{\mathrm{D}_3} & V_{\mathrm{Q}_3} & V_{\mathrm{D}_4} & V_{\mathrm{Q}_4} & V_{\mathrm{D}_5} & V_{\mathrm{Q}_5} \end{bmatrix}^{\mathrm{T}}, \tag{3.19c}$$

defining the DQ-components of each of the six bus voltages.

The simulated disturbance is a change in the voltage reference of generators 0, 1, and 2 of $+3\,\%$, $-1\,\%$, and $-2\,\%$, respectively.

The synchronous generator and hydro turbine and governor subsystems are defined as in the SMIB case, with the state equations given by (3.4) and (3.10), respectively. By not including a PSS, the state equation for the regulator input voltage of the exciter is slightly modified from (3.8) and given as

$$\dot{V}_{\mathrm{b}_i} = \frac{1}{T_{\mathrm{b}_i}}(V_{\mathrm{ref}_i} - V_{\mathrm{b}_i} - V_{\mathrm{f}_i} - V_{\mathrm{t}_i}) \quad \text{for} \quad i = 0, 1, 2. \tag{3.20}$$

The differential descriptions of the three generation units are uncoupled, with the algebraic equations providing the coupling via the transmission network.

Table 3.7 gives the parameter values of the three generation units, separated into synchronous generator, exciter, and turbine and governor parameters.

Table 3.7: Parameter values for generation units in 3MIB case.

(a) Synchronous generator parameters.

	Value [pu]		
Parameter	Generator 0	Generator 1	Generator 2
M	6.20	6.20	6.20
D	0.00	0.00	0.00
T'_{do}	6.00	6.00	6.00
T'_{qo}	0.54	0.54	0.54
T''_{do}	0.05	0.05	0.05
T''_{qo}	0.05	0.05	0.05
X_{d}	0.89	0.89	0.89
X_{q}	0.86	0.86	0.86
X'_{d}	0.30	0.30	0.30
X'_{q}	0.30	0.30	0.30
X''_{d}	0.10	0.10	0.10
X''_{q}	0.10	0.10	0.10

Table 3.7: Parameter values for generation units in 3MIB case *(continued)*.

(b) Exciter parameters.

	Value [pu]		
Parameter	Generator 0	Generator 1	Generator 2
T_{e}	1.330	1.330	1.330
T_{a}	0.050	0.050	0.050
T_{b}	0.100	0.100	0.100
T_{f}	0.670	0.670	0.670
T_{r}	0.001	0.001	0.001
K_{e}	1.000	1.000	1.000
K_{a}	300.000	300.000	300.000
K_{f}	0.100	0.100	0.100
V_{ref}	1.040	1.040	1.020
$V_{\mathrm{r}}^{\mathrm{max}}$	20.000	20.000	20.000
$V_{\mathrm{r}}^{\mathrm{min}}$	-20.000	-20.000	-20.000
S_{e}	0.000	0.000	0.000

(c) Hydro turbine and governor parameters.

	Value [pu]		
Parameter	Generator 0	Generator 1	Generator 2
T_{g}	0.20	0.20	0.20
T_{p}	0.04	0.04	0.04
T_{d}	5.00	5.00	5.00
T_{w}	0.10	0.10	0.10
D_{d}	0.30	0.30	0.30
R	0.05	0.05	0.05
P_{ref}	14.04	14.04	8.00
$V_{\mathrm{g}}^{\mathrm{max}}$	15.00	15.00	15.00
$V_{\mathrm{g}}^{\mathrm{min}}$	0.00	0.00	0.00
$V_{\mathrm{p}}^{\mathrm{max}}$	0.10	0.10	0.10
$V_{\mathrm{p}}^{\mathrm{min}}$	-0.10	-0.10	-0.10

3.3.1 Transmission Network and Dynamic Loads in Three Machines to Infinite Bus System

Based on (2.9) and the one-line diagram in Figure 3.7, the six-bus system can be expressed as

$$\begin{bmatrix} I_{\mathrm{D}_0} \\ I_{\mathrm{Q}_0} \\ I_{\mathrm{D}_1} \\ I_{\mathrm{Q}_1} \\ I_{\mathrm{D}_2} \\ I_{\mathrm{Q}_2} \\ I_{\mathrm{D}_3} \\ I_{\mathrm{Q}_3} \\ I_{\mathrm{D}_4} \\ I_{\mathrm{Q}_4} \\ I_{\mathrm{D}_5} \\ I_{\mathrm{Q}_5} \end{bmatrix} = \begin{bmatrix} G_{00} & -B_{00} & 0 & 0 & 0 & 0 & G_{03} & -B_{03} & 0 & 0 & 0 & 0 \\ B_{00} & G_{00} & 0 & 0 & 0 & 0 & B_{03} & G_{03} & 0 & 0 & 0 & 0 \\ 0 & 0 & G_{11} & -B_{11} & 0 & 0 & G_{13} & -B_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{11} & G_{11} & 0 & 0 & B_{13} & G_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{22} & -B_{22} & 0 & 0 & G_{24} & -B_{24} & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{22} & G_{22} & 0 & 0 & B_{24} & G_{24} & 0 & 0 \\ G_{30} & -B_{30} & G_{31} & -B_{31} & 0 & 0 & G_{33} & -B_{33} & G_{34} & -B_{34} & 0 & 0 \\ B_{30} & G_{30} & B_{31} & G_{31} & 0 & 0 & B_{33} & G_{33} & B_{34} & G_{34} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{42} & -B_{42} & G_{43} & -B_{43} & G_{44} & -B_{44} & G_{45} & -B_{45} \\ 0 & 0 & 0 & 0 & B_{42} & G_{42} & B_{43} & G_{43} & B_{44} & G_{44} & B_{45} & G_{45} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_{54} & -B_{54} & G_{55} & -B_{55} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{54} & G_{54} & B_{55} & G_{55} \end{bmatrix} \begin{bmatrix} V_{\mathrm{D}_0} \\ V_{\mathrm{Q}_0} \\ V_{\mathrm{D}_1} \\ V_{\mathrm{Q}_1} \\ V_{\mathrm{D}_2} \\ V_{\mathrm{Q}_2} \\ V_{\mathrm{D}_3} \\ V_{\mathrm{Q}_3} \\ V_{\mathrm{D}_4} \\ V_{\mathrm{Q}_4} \\ V_{\mathrm{D}_5} \\ V_{\mathrm{Q}_5} \end{bmatrix}. \tag{3.21}$$

Using (3.21) and neglecting the resistive component of the branch impedances, the algebraic equations of the system can be extracted as

$$\begin{bmatrix} g_0(\boldsymbol{y},t) \\ g_1(\boldsymbol{y},t) \\ g_2(\boldsymbol{y},t) \\ g_3(\boldsymbol{y},t) \\ g_4(\boldsymbol{y},t) \\ g_5(\boldsymbol{y},t) \\ g_6(\boldsymbol{y},t) \\ g_7(\boldsymbol{y},t) \\ g_8(\boldsymbol{y},t) \\ g_9(\boldsymbol{y},t) \\ g_{10}(\boldsymbol{y},t) \\ g_{11}(\boldsymbol{y},t) \end{bmatrix} = \mathbf{0} = \begin{bmatrix} -I_{\mathrm{D}_0} - B_{00}V_{\mathrm{Q}_0} - B_{03}V_{\mathrm{Q}_3} \\ -I_{\mathrm{Q}_0} + B_{00}V_{\mathrm{D}_0} + B_{03}V_{\mathrm{D}_3} \\ -I_{\mathrm{D}_1} - B_{11}V_{\mathrm{Q}_1} - B_{13}V_{\mathrm{Q}_3} \\ -I_{\mathrm{Q}_1} + B_{11}V_{\mathrm{D}_1} + B_{13}V_{\mathrm{D}_3} \\ -I_{\mathrm{D}_2} - B_{22}V_{\mathrm{Q}_2} - B_{24}V_{\mathrm{Q}_4} \\ -I_{\mathrm{Q}_2} + B_{22}V_{\mathrm{D}_2} + B_{24}V_{\mathrm{D}_4} \\ -I_{\mathrm{D}_3} - B_{30}V_{\mathrm{Q}_0} - B_{31}V_{\mathrm{Q}_1} - B_{33}V_{\mathrm{Q}_3} - B_{34}V_{\mathrm{Q}_4} \\ -I_{\mathrm{Q}_3} + B_{30}V_{\mathrm{D}_0} + B_{31}V_{\mathrm{D}_1} + B_{33}V_{\mathrm{D}_3} + B_{34}V_{\mathrm{D}_4} \\ -I_{\mathrm{D}_4} - B_{42}V_{\mathrm{Q}_2} - B_{43}V_{\mathrm{Q}_3} - B_{44}V_{\mathrm{Q}_4} - B_{45}V_{\mathrm{Q}_5} \\ -I_{\mathrm{Q}_4} + B_{42}V_{\mathrm{D}_2} + B_{43}V_{\mathrm{D}_3} + B_{44}V_{\mathrm{D}_4} + B_{45}V_{\mathrm{D}_5} \\ V_{\mathrm{D}_5} - 1.0 \\ V_{\mathrm{Q}_5} \end{bmatrix}. \tag{3.22}$$

Buses 0-2 are generator buses and the injected currents I_{D} and I_{Q} are calculated as in the SMIB case. Buses 3 and 4, however, are load buses. The connected loads are dynamically modeled using polynomial functions [13] as

$$P_{\mathrm{L}_i} = P_{0_i}\left[p_{1_i}\left(\frac{V_i}{V_{0_i}}\right)^2 + p_{2_i}\left(\frac{V_i}{V_{0_i}}\right) + p_{3_i}\right] \quad \text{and} \tag{3.23a}$$

$$Q_{\mathrm{L}_i} = Q_{0_i}\left[q_{1_i}\left(\frac{V_i}{V_{0_i}}\right)^2 + q_{2_i}\left(\frac{V_i}{V_{0_i}}\right) + q_{3_i}\right] \quad \text{for} \quad i = 3, 4. \tag{3.23b}$$

P_{L} and Q_{L} are active and reactive power drawn by the load, and P_0, Q_0, and V_0 are the steady-state values of active power, reactive power, and voltage magnitude at the connected bus, respectively. The p- and q-coefficients determine the portion of the load which is modeled as constant impedance (1), constant current (2), and constant power (3). The $p-$ and q-coefficients must each sum to 1.0.

The load currents can be found by starting with (2.10), solving for the current, and splitting into real and imaginary components to give

$$I_{\mathrm{D}_i} = \frac{V_{\mathrm{D}_i}P_{\mathrm{L}_i} + V_{\mathrm{Q}_i}Q_{\mathrm{L}_i}}{V_{\mathrm{D}_i}^2 + V_{\mathrm{Q}_i}^2} \quad \text{and} \tag{3.24a}$$

$$I_{\mathrm{Q}_i} = \frac{V_{\mathrm{Q}_i}P_{\mathrm{L}_i} + V_{\mathrm{D}_i}Q_{\mathrm{L}_i}}{V_{\mathrm{D}_i}^2 + V_{\mathrm{Q}_i}^2} \quad \text{for} \quad i = 3, 4. \tag{3.24b}$$

By inserting (3.23) into (3.24), the current drawn by the load can be expressed purely as a function of state variables, more specifically as

$$I_{\mathrm{D}_i}(\boldsymbol{y}, t) = \frac{1}{V_{\mathrm{D}_i}^2 + V_{\mathrm{Q}_i}^2}\left(V_{\mathrm{D}_i}P_{0_i}\left[p_{1_i}\left(\frac{V_i}{V_{0_i}}\right)^2 + p_{2_i}\left(\frac{V_i}{V_{0_i}}\right) + p_{3_i}\right] + V_{\mathrm{Q}_i}Q_{0_i}\left[q_{1_i}\left(\frac{V_i}{V_{0_i}}\right)^2 + q_{2_i}\left(\frac{V_i}{V_{0_i}}\right) + q_{3_i}\right]\right) \quad \text{and} \tag{3.25a}$$

$$I_{\mathrm{Q}_i}(\boldsymbol{y}, t) = \frac{1}{V_{\mathrm{D}_i}^2 + V_{\mathrm{Q}_i}^2}\left(V_{\mathrm{Q}_i}P_{0_i}\left[p_{1_i}\left(\frac{V_i}{V_{0_i}}\right)^2 + p_{2_i}\left(\frac{V_i}{V_{0_i}}\right) + p_{3_i}\right] - V_{\mathrm{D}_i}Q_{0_i}\left[q_{1_i}\left(\frac{V_i}{V_{0_i}}\right)^2 + q_{2_i}\left(\frac{V_i}{V_{0_i}}\right) + q_{3_i}\right]\right) \quad \text{for} \quad i = 3, 4. \tag{3.25b}$$

Note that the current in (3.25) is defined as flowing out of the bus.

Table 3.8 gives the branch admittances for the 3MIB network. The admittances of branches 0-3, 1-3, and 2-4 are the sums of their respective transformer and transmission line admittances. Branch 3-4 represents a short transmission line and branch 4-5 represents a long one.

Table 3.8: Branch admittances for network in 3MIB case.

Branch	Admittance [pu]
0-3	−j156.006
1-3	−j156.006
2-4	−j89.286
3-4	−j27.778
4-5	−j8.333

The load coefficients (p_1, p_2, p_3) and (q_1, q_2, q_3) are given in Table 3.9.

Table 3.9: Polynomial load coefficients for network in 3MIB case.

	Active power			Reactive power		
Bus number	p_1	p_2	p_3	q_1	q_2	q_2
3	0.0	0.0	1.0	0.0	0.0	1.0
4	0.0	0.0	1.0	0.0	0.0	1.0

3.3.2 Initialization of Three Machines to Infinite Bus System Based on Load Flow Solution

The steady-state load flow case is solved as described in Section 2.3.1. The stated net power injections are given in Table 3.10.

Table 3.10: Stated net bus power injections for 3MIB case.

Bus number	Stated net active power injection [pu]	Stated net reactive power injection [pu]
0	14.040	4.344
1	14.040	4.344
2	8.000	4.662
3	-14.000	-1.000
4	-20.000	-1.000

The algebraic variables are as before initialized directly using the load flow solution. Each of the generation units can then be initialized, as in the SMIB case, according to (3.18). The exception is that the initialization of the PSS in (3.18c) is not included in this case. As a consequence, the exciter states are initialized slightly differently, namely according to

$$\begin{bmatrix} E_{\mathrm{f}_i} \\ V_{\mathrm{r}_i} \\ V_{\mathrm{b}_i} \\ V_{\mathrm{t}_i} \\ V_{\mathrm{f}_i} \end{bmatrix} = \begin{bmatrix} E'_{\mathrm{q}_i} - I_{\mathrm{d}_i}(X_{\mathrm{d}_i} - X'_{\mathrm{d}_i}) \\ (S_{\mathrm{e}_i} + K_{\mathrm{e}_i})E_{\mathrm{f}_i} \\ \dfrac{V_{\mathrm{r}_i}}{K_{\mathrm{a}_i}} \\ \sqrt{V_{\mathrm{D}_i}^2 + V_{\mathrm{Q}_i}^2} \\ V_{\mathrm{ref}_i} - V_{\mathrm{b}_i} - V_{\mathrm{t}_i} \end{bmatrix} \quad \text{for} \quad i = 0, 1, 2. \tag{3.26}$$

The remaining differential states can be initialized according to (3.18).

3.4 Description of System with Static Compensator

The two first cases, SMIB and 3MIB, are both conventional power system cases where the dynamics are dominated by synchronous generators. The third case, however, introduces more power electronics into the system. The system consists of two synchronous generators connected to a load bus. The network consists of four buses connecting the generators and load to an infinite bus. In addition, a STATCOM, a component based on a VSC, is connected to the load bus. Figure 3.8 shows the single-line diagram of the system.

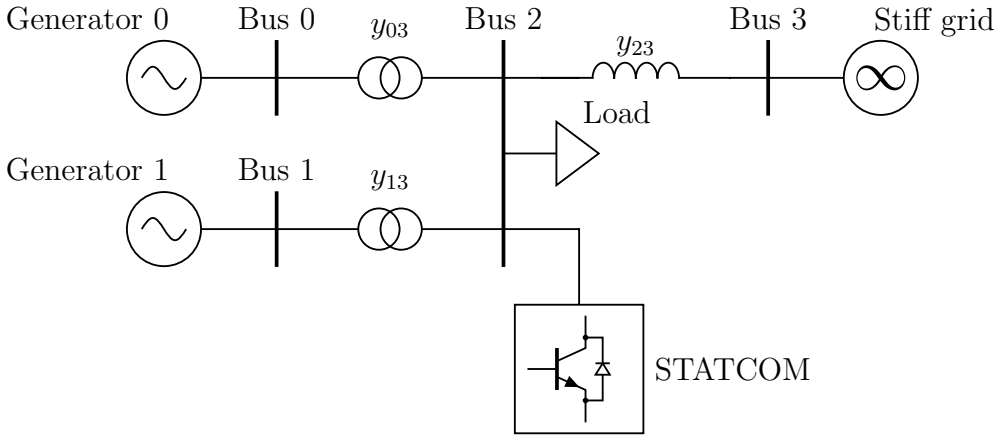


Figure 3.8: Single-line diagram of STATCOM system.

The system is defined by 40 states, 32 differential and eight algebraic. The two generators are defined as in the 3MIB case in Section 3.3, giving the differential state vector as

$$\boldsymbol{y}_{\mathrm{d}} = \begin{bmatrix} \boldsymbol{y}_{\mathrm{d}_0} \\ \boldsymbol{y}_{\mathrm{d}_1} \\ \boldsymbol{y}_{\mathrm{d}_2} \end{bmatrix}, \tag{3.27a}$$

where for $i = 0, 1$:

$$\boldsymbol{y}_{\mathrm{d}_i} = \begin{bmatrix} \Delta\omega_{\mathrm{s}_i} & \delta_i & E'_{\mathrm{q}_i} & E'_{\mathrm{d}_i} & E''_{\mathrm{q}_i} & E''_{\mathrm{d}_i} & E_{\mathrm{f}_i} & V_{\mathrm{r}_i} & V_{\mathrm{b}_i} & V_{\mathrm{f}_i} & V_{\mathrm{t}_i} & V_{\mathrm{g}_i} & V_{\mathrm{p}_i} & V_{\mathrm{s}_i} & P_{\mathrm{m}_i} \end{bmatrix}^{\mathrm{T}}, \tag{3.27b}$$

and

$$\boldsymbol{y}_{\mathrm{d}_2} = \begin{bmatrix} I_{\mathrm{svc}} & E_{\mathrm{svc}} \end{bmatrix}^{\mathrm{T}}. \tag{3.27c}$$

The algebraic state variables are

$$\boldsymbol{y}_{\mathrm{a}} = \begin{bmatrix} V_{\mathrm{D}_0} & V_{\mathrm{Q}_0} & V_{\mathrm{D}_1} & V_{\mathrm{Q}_1} & V_{\mathrm{D}_2} & V_{\mathrm{Q}_2} & V_{\mathrm{D}_3} & V_{\mathrm{Q}_3} \end{bmatrix}^{\mathrm{T}}, \tag{3.27d}$$

defining the DQ-components of each of the four bus voltages. I_{svc} is the current drawn by the STATCOM, directed out of the connected bus, and E_{svc} is the input to the STATCOM regulator.

The simulated disturbance is an islanding of the system, i.e. a disconnection of the two parallel lines making up the equivalent line 2-3 in Figure 3.8. The system is reconnected after 0.01 s, albeit through only one of the parallel lines.

Table 3.11 gives the parameter values for the two generating units in the network.

Table 3.11: Parameter values for generation units in STATCOM case.

(a) Synchronous generator parameters.

	Value [pu]	
Parameter	Generator 0	Generator 1
M	6.20	6.20
D	42.00	42.00
T'_{do}	6.00	6.00
T'_{qo}	0.54	0.54
T''_{do}	0.05	0.05
T''_{qo}	0.05	0.05
X_{d}	0.89	0.89
X_{q}	0.86	0.86
X'_{d}	0.30	0.30
X'_{q}	0.30	0.30
X''_{d}	0.10	0.10
X''_{q}	0.10	0.10

(b) Exciter parameters.

	Value [pu]	
Parameter	Generator 0	Generator 1
T_{e}	1.330	1.330
T_{a}	0.050	0.050
T_{b}	0.100	0.100
T_{f}	0.670	0.670
T_{r}	0.001	0.001
K_{e}	1.000	1.000
K_{a}	300.000	300.000
K_{f}	0.100	0.100
V_{ref}	1.040	1.040
$V_{\mathrm{r}}^{\mathrm{max}}$	20.000	20.000
$V_{\mathrm{r}}^{\mathrm{min}}$	-20.000	-20.000
S_{e}	0.000	0.000

Table 3.11: Parameter values for generation units in STATCOM case *(continued)*.

(c) Hydro turbine and governor parameters.

	Value [pu]	
Parameter	Generator 0	Generator 1
T_{g}	0.20	0.20
T_{p}	0.04	0.04
T_{d}	5.00	5.00
T_{w}	0.10	0.10
D_{d}	0.30	0.30
R	0.05	0.05
P_{ref}	5.000	8.00
$V_{\mathrm{g}}^{\mathrm{max}}$	15.00	15.00
$V_{\mathrm{g}}^{\mathrm{min}}$	0.00	0.00
$V_{\mathrm{p}}^{\mathrm{max}}$	0.10	0.10
$V_{\mathrm{p}}^{\mathrm{min}}$	-0.10	-0.10

3.4.1 Static Compensator

Figure 3.9 shows a simplified dynamic model of the STATCOM [13], modeled by two differential equations as

$$\begin{bmatrix} f_{30}(\boldsymbol{y},t) \\ f_{31}(\boldsymbol{y},t) \end{bmatrix} = \begin{bmatrix} \dot{I}_{\mathrm{svc}} \\ \dot{E}_{\mathrm{svc}} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{T_{\mathrm{svc}}}(E_{\mathrm{svc}} - I_{\mathrm{svc}}) \\ K_{\mathrm{svc}}\left(V_{\mathrm{ref}} - \sqrt{V_{\mathrm{D}}^2 + V_{\mathrm{Q}}^2} - \rho_{\mathrm{svc}} I_{\mathrm{svc}}\right) \end{bmatrix}. \tag{3.28}$$

K_{svc} and ρ_{svc} are static gains and T_{svc} is the regulator time constant. The regulator input is limited by $E_{\mathrm{svc}}^{\mathrm{max}}$ and $E_{\mathrm{svc}}^{\mathrm{min}}$. The maximum value corresponds to the largest inductive current the STATCOM can deliver, and the minimum value corresponds to the largest capacitive current. The STATCOM parameter values are given in Table 3.12.

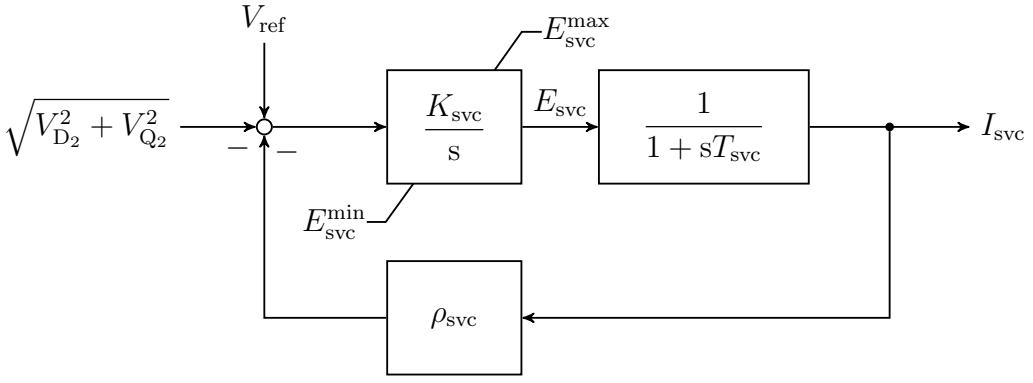


Figure 3.9: Block diagram representation of STATCOM, adapted from [13].

Table 3.12: Parameter values for static compensator in STATCOM case.

Parameter	Value [pu]
T_{svc}	0.001
K_{svc}	1.000
ρ_{svc}	5.000
V_{ref}	1.000
$E_{\text{svc}}^{\max}$	10.000
$E_{\text{svc}}^{\min}$	-10.000

3.4.2 Transmission Network in System with Static Compensator

Using the form in (2.9) and Figure 3.8, the four-bus system can be expressed as

$$\begin{bmatrix} I_{\text{D}_0} \\ I_{\text{Q}_0} \\ I_{\text{D}_1} \\ I_{\text{Q}_1} \\ I_{\text{D}_2} \\ I_{\text{Q}_2} \\ I_{\text{D}_3} \\ I_{\text{Q}_3} \end{bmatrix} = \begin{bmatrix} G_{00} & -B_{00} & 0 & 0 & G_{02} & -B_{02} & 0 & 0 \\ B_{00} & G_{00} & 0 & 0 & B_{02} & G_{02} & 0 & 0 \\ 0 & 0 & G_{11} & -B_{11} & G_{12} & -B_{12} & 0 & 0 \\ 0 & 0 & B_{11} & G_{11} & B_{12} & G_{12} & 0 & 0 \\ G_{20} & -B_{20} & G_{21} & -B_{21} & G_{22} & -B_{22} & G_{23} & -B_{23} \\ B_{20} & G_{20} & B_{21} & G_{21} & B_{22} & G_{22} & B_{23} & G_{23} \\ 0 & 0 & 0 & 0 & G_{32} & -B_{32} & G_{33} & -B_{33} \\ 0 & 0 & 0 & 0 & B_{32} & G_{32} & B_{33} & G_{33} \end{bmatrix} \begin{bmatrix} V_{\text{D}_0} \\ V_{\text{Q}_0} \\ V_{\text{D}_1} \\ V_{\text{Q}_1} \\ V_{\text{D}_2} \\ V_{\text{Q}_2} \\ V_{\text{D}_3} \\ V_{\text{Q}_3} \end{bmatrix}. \tag{3.29}$$

By neglecting line resistances, the algebraic equations can be extracted from (3.29) to give:

$$\begin{bmatrix} g_0(\boldsymbol{y},t) \\ g_1(\boldsymbol{y},t) \\ g_2(\boldsymbol{y},t) \\ g_3(\boldsymbol{y},t) \\ g_4(\boldsymbol{y},t) \\ g_5(\boldsymbol{y},t) \\ g_6(\boldsymbol{y},t) \\ g_7(\boldsymbol{y},t) \end{bmatrix} = \mathbf{0} = \begin{bmatrix} -I_{\text{D}_0} - B_{00}V_{\text{Q}_0} - B_{02}V_{\text{Q}_2} \\ -I_{\text{Q}_0} + B_{00}V_{\text{D}_0} + B_{02}V_{\text{D}_2} \\ -I_{\text{D}_1} - B_{11}V_{\text{Q}_1} - B_{12}V_{\text{Q}_2} \\ -I_{\text{Q}_1} + B_{11}V_{\text{D}_1} + B_{12}V_{\text{D}_2} \\ -I_{\text{D}_2} - B_{20}V_{\text{Q}_0} - B_{21}V_{\text{Q}_1} - B_{22}V_{\text{Q}_2} - B_{23}V_{\text{Q}_3} \\ -I_{\text{Q}_2} + B_{20}V_{\text{D}_0} + B_{21}V_{\text{D}_1} + B_{22}V_{\text{D}_2} + B_{23}V_{\text{D}_3} \\ V_{\text{D}_3} - 1.0 \\ V_{\text{Q}_3} \end{bmatrix}. \tag{3.30}$$

The injected currents at buses 0 and 1 are generator currents, and are calculated according to (3.16). The injected current at bus 2 is the sum of the load current, given by (3.25), and the STATCOM current I_{svc}. The STATCOM current is purely reactive, meaning that $\underline{I}_{\text{svc}} = \text{j}I_{\text{Q}_{\text{svc}}}$ [13].

The branch admittances for the STATCOM network in normal operation are given in Table 3.13, and the load coefficients are given in Table 3.14.

Table 3.13: Normal operation branch admittances for network in STATCOM case.

Branch	Admittance [pu]
0-2	−j156.006
1-2	−j156.006
2-3	−j89.2857

Table 3.14: Polynomial load coefficients for network in STATCOM case.

	Active power			Reactive power		
Bus number	p_1	p_2	p_3	q_1	q_2	q_2
2	0.2	0.8	0.0	0.0	0.0	1.0

Branches 0-2 and 1-2 represent transformers and branch 2-3 represents two parallel transmission lines connecting the system to the infinite bus.

During the islanding, the voltage at bus 3 and the admittance of line 2-3 change to reflect the broken connection between the network and the infinite bus. After 0.01 s the system is reconnected and the infinite bus voltage is restored and the admittance of branch 2-3 takes on its post-islanding value. Table 3.15 gives their values during and after the event.

Table 3.15: Islanding and post-islanding values for the STATCOM case.

(a) Islanding values.

Parameter	Value [pu]
V_{D_3}	0.000
jB_{23}	j0.000

(b) Post-islanding values.

Parameter	Value [pu]
V_{D_3}	1.000
jB_{23}	−j44.643

3.4.3 Initialization of System with Static Compensator Based on Load Flow Solution

The steady-state load flow case can be solved as described in Section 2.3.1. The stated net power injections are given in Table 3.16.

Table 3.16: Stated net bus power injections for STATCOM case.

Bus number	Stated net active power injection [pu]	Stated net reactive power injection [pu]
0	5.0	1.0
1	8.0	0.0
2	-15.0	-2.0

The load flow solution is used directly to initialize the algebraic variables. The two generating units can be initialized as described for the 3MIB case in Section 3.3.2. The STATCOM states are initialized as

$$\begin{bmatrix} y_{30} \\ y_{31} \end{bmatrix} = \begin{bmatrix} I_{\text{svc}} \\ E_{\text{svc}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho_{\text{svc}}} \left(V_{\text{ref}} - \sqrt{V_{\text{D}_2}^2 + V_{\text{Q}_2}^2} \right) \\ I_{\text{svc}} \end{bmatrix}. \tag{3.31}$$

3.5 Description of Multi-Terminal DC System

The fourth and final case varies considerably from the three previous ones. It consists of a multi-terminal direct current (MTDC) network, interconnecting three VSCs, of which one is modeled as the slack converter. Each of the VSCs are connected to an infinite grid as shown in Figure 3.10. R_{dc} and L_{dc} are the resistances and inductances of the DC lines, and T0-T2 are transformers.

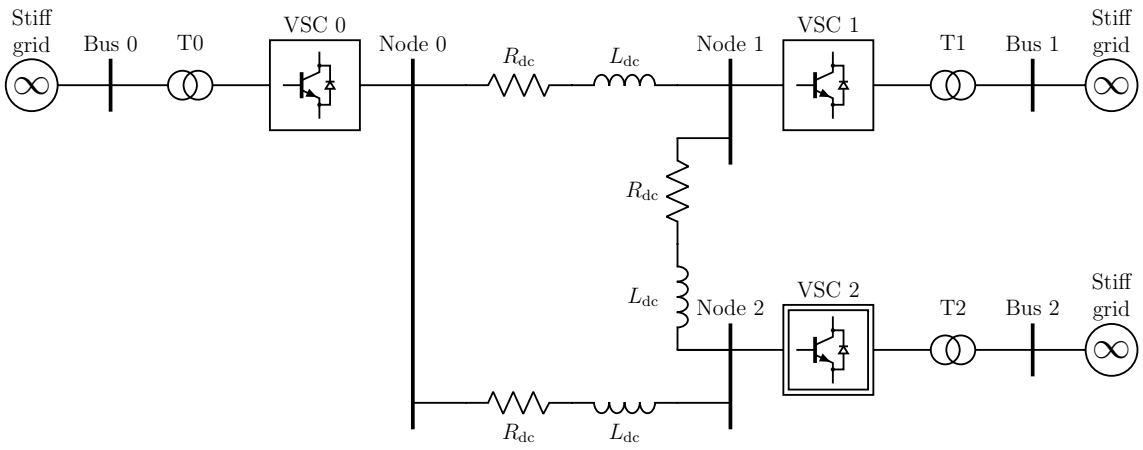


Figure 3.10: Single-line diagram of MTDC system.

The MTDC system is defined by 37 states. 31 states are differential and six are algebraic. The differential state vector is given as

$$\boldsymbol{y}_{\text{d}} = \begin{bmatrix} \boldsymbol{y}_{\text{d}_0} \\ \boldsymbol{y}_{\text{d}_1} \\ \boldsymbol{y}_{\text{d}_2} \\ \boldsymbol{y}_{\text{d}_3} \end{bmatrix}, \tag{3.32a}$$

where for $i = 0, 1$:

$$\boldsymbol{y}_{\mathrm{d}_i} = \begin{bmatrix} I_{\mathrm{d}_i} & I_{\mathrm{q}_i} & V_{\mathrm{dt}_i} & V_{\mathrm{qt}_i} & M_{\mathrm{d}_i} & M_{\mathrm{q}_i} & N_{\mathrm{d}_i} & N_{\mathrm{q}_i} \end{bmatrix}^{\mathrm{T}}, \tag{3.32b}$$

representing the non-slack converters. The slack converter is defined by the states

$$\boldsymbol{y}_{\mathrm{d}_2} = \begin{bmatrix} I_{\mathrm{d}_2} & I_{\mathrm{q}_2} & V_{\mathrm{dt}_2} & V_{\mathrm{qt}_2} & M_{\mathrm{d}_2} & M_{\mathrm{q}_2} & N_{\mathrm{d}_2} \end{bmatrix}^{\mathrm{T}}, \tag{3.32c}$$

and the DC grid is defined by

$$\boldsymbol{y}_{\mathrm{d}_3} = \begin{bmatrix} V_{\mathrm{dc}_0} & V_{\mathrm{dc}_1} & V_{\mathrm{dc}_2} & I_{01} & I_{02} & I_{12} & I_{\mathrm{dc}_2} & M_{\mathrm{dc}} \end{bmatrix}^{\mathrm{T}}. \tag{3.32d}$$

The algebraic states are the dq-components of the infinite buses:

$$\boldsymbol{y}_{\mathrm{a}} = \begin{bmatrix} V_{\mathrm{d}_0} & V_{\mathrm{q}_0} & V_{\mathrm{d}_1} & V_{\mathrm{q}_1} & V_{\mathrm{d}_2} & V_{\mathrm{q}_2} \end{bmatrix}^{\mathrm{T}}. \tag{3.32e}$$

For all three converters, including the slack converter, I_{d} and I_{q} are the dq-components of the AC side current. V_{dt} and V_{qt} are the dq-components of the AC voltage at each of the three converters. M_{d}, M_{q}, N_{d}, and N_{q} are states introduced due to the integrators in the current, voltage, and active power controllers of the converters. V_{dc} is the voltage on the DC sides of the converters, and I_{ij} are the currents in the DC branches. M_{dc} is an integration state originating from the DC voltage controller of the network.

Note that only one dq-reference frame is used, as apposed to several local and one global frame. This is because the system does not contain synchronous generators, but rather three infinite buses. If they are defined to have the same reference frame, the DC system follows, and only a single reference frame is required.

The simulated fault is the removal of the line between nodes 1 and 2 in Figure 3.10, forcing the current I_{12} to zero for the remainder of the simulation.

3.5.1 Voltage Source Converter

The following modeling of VSCs and their control systems is based on the modeling done by [2]. The equivalent phasor circuit of the voltage source converter is shown in Figure 3.11, where the converter is represented by the terminal voltage $\underline{V}_{\mathrm{t}}$ and the AC system is represented by $\underline{V}$. The two equivalent voltage sources are connected through an impedance $\underline{Z}_{\mathrm{c}} = R_{\mathrm{c}} + \mathrm{j}X_{\mathrm{c}}$, which includes the effect of the transformer. $\underline{I}$ is the current flowing between the two equivalent sources.

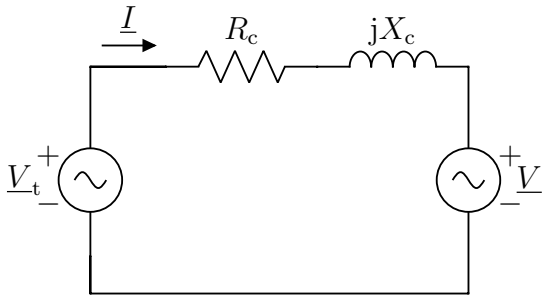


Figure 3.11: Equivalent circuit of VSC, adapted from [2].

Applying Kirchhoff's voltage law on the circuit in Figure 3.11 gives the equation

$$L_{\mathrm{c}}\underline{\dot{I}} = \underline{V}_{\mathrm{t}} - \underline{V} - R_{\mathrm{c}}\underline{I}. \tag{3.33}$$

Considering that (3.33) models a three-phase balanced system, it can be expressed as a space phasor as defined in (2.29). From this, (3.33) can first be written in $\alpha\beta$-coordinates, before finally being transformed into dq-components as given by the transformation in (2.33):

$$\begin{bmatrix} \dot{I}_{\mathrm{d}_i} \\ \dot{I}_{\mathrm{q}_i} \end{bmatrix} = \begin{bmatrix} -\dfrac{R_{\mathrm{c}_i}}{L_{\mathrm{c}_i}} I_{\mathrm{d}_i} + \omega_{\mathrm{s}} I_{\mathrm{q}_i} (V_{\mathrm{dt}_i} - V_{\mathrm{d}_i}) \\ -\dfrac{R_{\mathrm{c}_i}}{L_{\mathrm{c}_i}} I_{\mathrm{q}_i} - \omega_{\mathrm{s}} I_{\mathrm{d}_i} (V_{\mathrm{qt}_i} - V_{\mathrm{q}_i}) \end{bmatrix} \quad \text{for} \quad i = 0, 1, 2, \tag{3.34}$$

where the phasors $\underline{V}_{\mathrm{t}}$, $\underline{V}$, and $\underline{I}$ have been split into their dq-components and the index i has been included to indicate that the modeling applies to all three converters in the system. The angle $\omega_{\mathrm{s}}t$ required for the transformation is provided by a phase-locked-loop, assumed here to align the system voltage with the q-axis, i.e. $\underline{V}_{\mathrm{i}} = \mathrm{j}V_{\mathrm{q}_i}$.

The VSCs here modeled have a cascaded control structure, consisting of an inner current controller and an outer controller. The outer controller controls either reactive power and voltage, as with the slack converter, or active power, as with the remaining converters.

The dynamics of the current controller, voltage controller, active power controller, and power balance of the non-slack converters can be modeled as

$$\begin{bmatrix} \dot{V}_{\mathrm{dt}_i} \\ \dot{V}_{\mathrm{qt}_i} \\ \dot{M}_{\mathrm{d}_i} \\ \dot{M}_{\mathrm{q}_i} \\ \dot{N}_{\mathrm{d}_i} \\ \dot{N}_{\mathrm{q}_i} \end{bmatrix} = \begin{bmatrix} -\dfrac{K_{\mathrm{p1}_i}}{T_{\sigma_i}} I_{\mathrm{d}_i} + \omega_{\mathrm{s}} \dfrac{L_{\mathrm{c}_i}}{T_{\sigma_i}} I_{\mathrm{q}_i} - \dfrac{1}{T_{\sigma_i}} V_{\mathrm{dt}_i} + \dfrac{1}{T_{\sigma_i}} M_{\mathrm{d}_i} + \dfrac{K_{\mathrm{p1}_i}}{T_{\sigma_i}} I_{\mathrm{d,ref}_i} + \dfrac{1}{T_{\sigma_i}} V_{\mathrm{d}_i} \\ -\dfrac{K_{\mathrm{p1}_i}}{T_{\sigma_i}} I_{\mathrm{q}_i} - \omega_{\mathrm{s}} \dfrac{L_{\mathrm{c}_i}}{T_{\sigma_i}} I_{\mathrm{d}_i} - \dfrac{1}{T_{\sigma_i}} V_{\mathrm{qt}_i} + \dfrac{1}{T_{\sigma_i}} M_{\mathrm{q}_i} + \dfrac{K_{\mathrm{p1}_i}}{T_{\sigma_i}} I_{\mathrm{q,ref}_i} + \dfrac{1}{T_{\sigma_i}} V_{\mathrm{q}_i} \\ K_{\mathrm{j1}_i} (I_{\mathrm{d,ref}_i} - I_{\mathrm{d}_i}) \\ K_{\mathrm{j1}_i} (I_{\mathrm{q,ref}_i} - I_{\mathrm{q}_i}) \\ K_{\mathrm{j2}_i} (V_{\mathrm{ref}_i} - V_{\mathrm{q}_i}) \\ K_{\mathrm{j3}_i} (P_{\mathrm{ref}_i} - V_{\mathrm{q}_i} I_{\mathrm{q}_i}) \end{bmatrix}, \tag{3.35a}$$

where

$$I_{\mathrm{d,ref}_i} = N_{\mathrm{d}_i} + K_{\mathrm{p2}_i} (V_{\mathrm{ref}_i} - V_{\mathrm{q}_i}) \quad \text{and} \tag{3.35b}$$

$$I_{\mathrm{q,ref}_i} = \frac{P_{\mathrm{ref}_i}}{V_{\mathrm{q0}}} + N_{\mathrm{q}_i} + K_{\mathrm{p3}_i} (P_{\mathrm{ref}_i} - V_{\mathrm{q}_i} I_{\mathrm{q}_i}), \tag{3.35c}$$

for $i = 0, 1$. The slack converter is defined by (3.34) and

$$\begin{bmatrix} \dot{V}_{\mathrm{dt}_2} \\ \dot{V}_{\mathrm{qt}_2} \\ \dot{M}_{\mathrm{d}_2} \\ \dot{M}_{\mathrm{q}_2} \\ \dot{N}_{\mathrm{d}_2} \end{bmatrix} = \begin{bmatrix} -\frac{K_{\mathrm{p1}_2}}{T_{\sigma_2}} I_{\mathrm{d}_2} + \omega_{\mathrm{s}} \frac{L_{\mathrm{c}_2}}{T_{\sigma_2}} I_{\mathrm{q}_2} - \frac{1}{T_{\sigma_2}} V_{\mathrm{dt}_2} + \frac{1}{T_{\sigma_2}} M_{\mathrm{d}_2} + \frac{K_{\mathrm{p1}_2}}{T_{\sigma_2}} I_{\mathrm{d,ref}_2} + \frac{1}{T_{\sigma_2}} V_{\mathrm{d}_2} \\ -\frac{K_{\mathrm{p1}_2}}{T_{\sigma_2}} I_{\mathrm{q}_2} - \omega_{\mathrm{s}} \frac{L_{\mathrm{c}_2}}{T_{\sigma_2}} I_{\mathrm{d}_2} - \frac{1}{T_{\sigma_2}} V_{\mathrm{qt}_2} + \frac{1}{T_{\sigma_2}} M_{\mathrm{q}_2} + \frac{K_{\mathrm{p1}_2}}{T_{\sigma_2}} I_{\mathrm{q,ref}_2} + \frac{1}{T_{\sigma_2}} V_{\mathrm{q}_2} \\ K_{\mathrm{j1}_2}(I_{\mathrm{d,ref}_2} - I_{\mathrm{d}_2}) \\ K_{\mathrm{j1}_2}(I_{\mathrm{q,ref}_2} - I_{\mathrm{q}_2}) \\ K_{\mathrm{j2}_2}(V_{\mathrm{ref}_2} - V_{\mathrm{q}_2}) \end{bmatrix}, \tag{3.36a}$$

with

$$I_{\mathrm{d,ref}_2} = N_{\mathrm{d}_2} + K_{\mathrm{p2}_2}(V_{\mathrm{ref}_2} - V_{\mathrm{q}_2}) \quad \text{and} \tag{3.36b}$$

$$I_{\mathrm{q,ref}_2} = \frac{2I_{\mathrm{dc}_2}V_{\mathrm{dc}_2} - I_{\mathrm{d}_2}V_{\mathrm{dt}_2}}{V_{\mathrm{qt}_2}}. \tag{3.36c}$$

The K_{p}- and K_{j}-parameters are controller proportional and integral gains, T_σ is the time constant to model the delay between the voltage reference and actual terminal voltage of the converters, and P_{ref} and V_{ref} are active power and AC voltage reference values.

The parameter values for the three converters, including the slack converter, are given in Table 3.17.

Table 3.17: Parameter values for VSCs in MTDC case. VSC 2 is slack converter.

	Value [pu]		
Parameter	VSC 0	VSC 1	VSC 2
R_{c}	0.005	0.005	0.005
L_{c}	0.200	0.200	0.200
T_σ	0.100	0.100	0.100
K_{p1}	13.000	13.000	13.000
K_{p2}	10.000	10.000	10.000
K_{p3}	20.000	20.000	20.000
K_{j1}	5.000	5.000	5.000
K_{j2}	100.000	100.000	100.000
K_{j3}	40.000	40.000	40.000
V_{ref}	1.000	1.000	1.000
P_{ref}	-1.500	2.000	-

3.5.2 DC Circuit

Figure 3.12 shows the circuit of a node in an MTDC network where the positive current direction is defined as flowing from a lower node index number to a higher one. C_{dc} is a capacitance and R_{dc} and L_{dc} are still the resistance and inductance of the DC line. All lines are modeled with the same parameters.

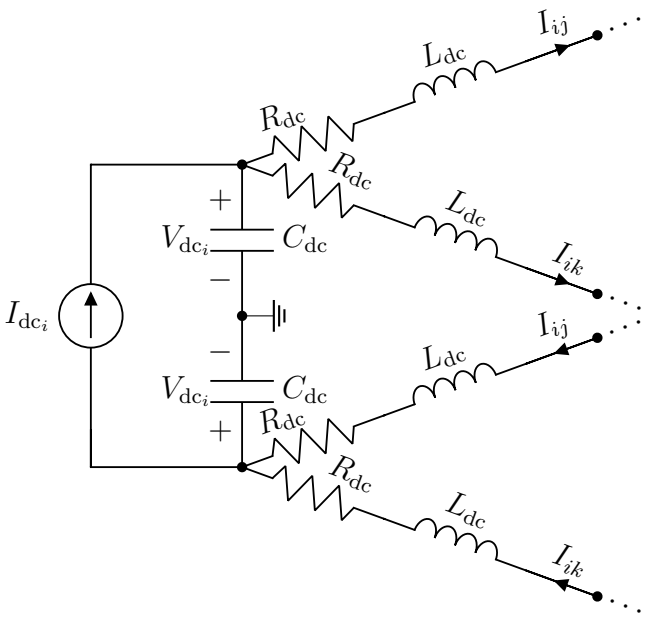


Figure 3.12: Equivalent circuit of MTDC-node i connected to nodes j and k, adapted from [2].

By using that the injected current at a non-slack node is calculated as

$$I_{\mathrm{dc}_i} = \frac{V_{\mathrm{dt}_i} I_{\mathrm{d}_i} + V_{\mathrm{qt}_i} I_{\mathrm{q}_i}}{2V_{\mathrm{dc}_i}} \quad \text{for} \quad i = 0, 1, \tag{3.37}$$

that the dynamics of the controlled slack converter current are modeled as

$$\dot{I}_{\mathrm{dc}_2} = \frac{1}{T_{\mathrm{dc}}}[-I_{\mathrm{dc}_2} + K_{\mathrm{pdc}}(V_{\mathrm{dc}_{\mathrm{ref}}} - V_{\mathrm{dc}_2}) + M_{\mathrm{dc}}] \quad \text{and} \tag{3.38a}$$

$$\dot{M}_{\mathrm{dc}} = K_{\mathrm{jdc}}(V_{\mathrm{dc}_{\mathrm{ref}}} - V_{\mathrm{dc}_2}), \tag{3.38b}$$

and Kirchhoff's current law, the DC network can be defined by the following differential equations:

$$\begin{bmatrix} f_{23}(\boldsymbol{y},t) \\ f_{24}(\boldsymbol{y},t) \\ f_{25}(\boldsymbol{y},t) \\ f_{26}(\boldsymbol{y},t) \\ f_{27}(\boldsymbol{y},t) \\ f_{28}(\boldsymbol{y},t) \\ f_{29}(\boldsymbol{y},t) \\ f_{30}(\boldsymbol{y},t) \end{bmatrix} = \begin{bmatrix} \dot{V}_{\mathrm{dc}_0} \\ \dot{V}_{\mathrm{dc}_1} \\ \dot{V}_{\mathrm{dc}_2} \\ \dot{I}_{01} \\ \dot{I}_{02} \\ \dot{I}_{12} \\ \dot{I}_{\mathrm{dc}_2} \\ \dot{M}_{\mathrm{dc}} \end{bmatrix} = \begin{bmatrix} \frac{1}{C_{\mathrm{dc}}}\left(\frac{V_{\mathrm{dt}_0}I_{\mathrm{d}_0}+V_{\mathrm{qt}_0}I_{\mathrm{q}_0}}{2V_{\mathrm{dc}_0}} - I_{01} - I_{02}\right) \\ \frac{1}{C_{\mathrm{dc}}}\left(\frac{V_{\mathrm{dt}_1}I_{\mathrm{d}_1}+V_{\mathrm{qt}_1}I_{\mathrm{q}_1}}{2V_{\mathrm{dc}_1}} - I_{01} - I_{12}\right) \\ \frac{1}{C_{\mathrm{dc}}}(I_{\mathrm{dc}_2}+I_{02}+I_{12}) \\ \frac{1}{L_{\mathrm{dc}}}(V_{\mathrm{dc}_0}-V_{\mathrm{dc}_1}-R_{\mathrm{dc}}I_{01}) \\ \frac{1}{L_{\mathrm{dc}}}(V_{\mathrm{dc}_0}-V_{\mathrm{dc}_2}-R_{\mathrm{dc}}I_{02}) \\ \frac{1}{L_{\mathrm{dc}}}(V_{\mathrm{dc}_1}-V_{\mathrm{dc}_2}-R_{\mathrm{dc}}I_{12}) \\ \frac{1}{T_{\mathrm{dc}}}[-I_{\mathrm{dc}_2}+K_{\mathrm{pdc}}(V_{\mathrm{dc}_{\mathrm{ref}}}-V_{\mathrm{dc}_2})+M_{\mathrm{dc}}] \\ K_{\mathrm{jdc}}(V_{\mathrm{dc}_{\mathrm{ref}}}-V_{\mathrm{dc}_2}) \end{bmatrix}. \tag{3.39}$$

K_{pdc} and K_{jdc} are static proportional and integral gains of the DC voltage controller, respectively, and $V_{\mathrm{dc}_{\mathrm{ref}}}$ is the DC voltage reference.

Table 3.18 gives the parameters of the DC network.

Table 3.18: Parameter values for DC network in MTDC case.

Parameter	Value [pu]
R_{dc}	0.003
L_{dc}	0.002
C_{dc}	4.000
T_{dc}	0.010
K_{pdc}	20.000
K_{jdc}	40.000
$V_{\mathrm{dc}_{\mathrm{ref}}}$	1.000

3.5.3 Transmission Network in Multi-Terminal DC System

As all the converters in the model are connected to an infinite bus, the only purpose of the algebraic equations is to ensure that the AC bus voltages are kept at unity magnitude and aligned with the q-axis. This leads to the formulation of the algebraic equations as

$$\begin{bmatrix} g_0(\boldsymbol{y},t) \\ g_1(\boldsymbol{y},t) \\ g_2(\boldsymbol{y},t) \\ g_3(\boldsymbol{y},t) \\ g_4(\boldsymbol{y},t) \\ g_5(\boldsymbol{y},t) \end{bmatrix} = \mathbf{0} = \begin{bmatrix} V_{\mathrm{d}_0} \\ V_{\mathrm{q}_0}-1.0 \\ V_{\mathrm{d}_1} \\ V_{\mathrm{q}_1}-1.0 \\ V_{\mathrm{d}_2} \\ V_{\mathrm{q}_2}-1.0 \end{bmatrix}. \tag{3.40}$$

3.5.4 Initialization Based on DC Load Flow Solution

The steady-state DC load flow solution of the system can be found using the method described in Section 2.3.2, where the admittance matrix $\boldsymbol{Y}_{\mathrm{dc}}$ is

$$\boldsymbol{Y}_{\mathrm{dc}} = \frac{1}{R_{\mathrm{dc}}} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}. \tag{3.41}$$

From this, the steady-state equations from (2.25) can be extracted as

$$0 = \frac{V_{\mathrm{dc}_0} - V_{\mathrm{dc}_1}}{R_{\mathrm{dc}}} + \frac{V_{\mathrm{dc}_0} - V_{\mathrm{dc}_{\mathrm{ref}}}}{R_{\mathrm{dc}}} - \frac{P_{\mathrm{dc}_0}}{2V_{\mathrm{dc}_0}}, \tag{3.42a}$$

$$0 = \frac{V_{\mathrm{dc}_1} - V_{\mathrm{dc}_0}}{R_{\mathrm{dc}}} + \frac{V_{\mathrm{dc}_1} - V_{\mathrm{dc}_{\mathrm{ref}}}}{R_{\mathrm{dc}}} - \frac{P_{\mathrm{dc}_1}}{2V_{\mathrm{dc}_1}} \quad \text{and} \tag{3.42b}$$

$$0 = -\frac{V_{\mathrm{dc}_0} + V_{\mathrm{dc}_1}}{R_{\mathrm{dc}}} + \frac{2V_{\mathrm{dc}_{\mathrm{ref}}}}{R_{\mathrm{dc}}} - \frac{P_{\mathrm{dc}_2}}{2V_{\mathrm{dc}_{\mathrm{ref}}}}. \tag{3.42c}$$

Each of the algebraic equations are initialized so that the equations in (3.40) are satisfied.

Following the DC load flow solution, the dq-components of the three converter currents can be determined using (2.10). The branch currents, DC voltages, and injected DC current at the slack node also follow directly from the load flow solution. The remaining non-slack converter states are initialized as

$$\begin{bmatrix} N_{\mathrm{d}_i} \\ N_{\mathrm{q}_i} \\ V_{\mathrm{dt}_i} \\ V_{\mathrm{qt}_i} \\ M_{\mathrm{d}_i} \\ M_{\mathrm{q}_i} \end{bmatrix} = \begin{bmatrix} I_{\mathrm{d}_i} - K_{\mathrm{p2}_i}(V_{\mathrm{ref}_i} - V_{\mathrm{q}_i}) \\ I_{\mathrm{q}_i} - \dfrac{P_{\mathrm{ref}_i}}{V_{\mathrm{q}_i}} - K_{\mathrm{p3}_i}(P_{\mathrm{ref}_i} - V_{\mathrm{q}_i} I_{\mathrm{q}_i}) \\ V_{\mathrm{d}_i} + (R_{\mathrm{c}} I_{\mathrm{d}_i} - \omega_{\mathrm{s}} L_{\mathrm{c}} I_{\mathrm{q}_i}) \\ V_{\mathrm{q}_i} + (R_{\mathrm{c}} I_{\mathrm{q}_i} + \omega_{\mathrm{s}} L_{\mathrm{c}} I_{\mathrm{d}_i}) \\ K_{\mathrm{p1}_i} I_{\mathrm{d}_i} - \omega_{\mathrm{s}} L_{\mathrm{c}} I_{\mathrm{l}_i} + V_{\mathrm{dt}_i} - K_{\mathrm{p1}_i}[N_{\mathrm{d}_i} + K_{\mathrm{p2}_i}(V_{\mathrm{ref}_i} - V_{\mathrm{q}_i})] - V_{\mathrm{d}_i} \\ K_{\mathrm{p1}_i} I_{\mathrm{q}_i} + \omega_{\mathrm{s}} L_{\mathrm{c}} I_i + V_{\mathrm{qt}_i} - K_{\mathrm{p1}_i}\left[\dfrac{P_{\mathrm{ref}_i}}{V_{\mathrm{q}_i}} + N_{\mathrm{q}_i} + K_{\mathrm{p3}_i}(P_{\mathrm{ref}_i} - V_{\mathrm{q}_i} I_{\mathrm{q}_i})\right] - V_{\mathrm{q}_i} \end{bmatrix} \tag{3.43}$$

for $i = 0, 1$. The remaining slack converter states are initialized similarly as

$$\begin{bmatrix} N_{\mathrm{d}_2} \\ V_{\mathrm{dt}_2} \\ V_{\mathrm{qt}_2} \\ M_{\mathrm{d}_2} \\ M_{\mathrm{q}_2} \end{bmatrix} = \begin{bmatrix} I_{\mathrm{d}_2} - K_{\mathrm{p2}_2}(V_{\mathrm{ref}_2} - V_{\mathrm{q}_2}) \\ V_{\mathrm{d}_2} + (R_{\mathrm{c}} I_{\mathrm{d}_2} - \omega_{\mathrm{s}} L_{\mathrm{c}} I_{\mathrm{q}_2}) \\ V_{\mathrm{q}_2} + (R_{\mathrm{c}} I_{\mathrm{q}_2} + \omega_{\mathrm{s}} L_{\mathrm{c}} I_{\mathrm{d}_2}) \\ K_{\mathrm{p1}_2} I_{\mathrm{d}_2} - \omega_{\mathrm{s}} L_{\mathrm{c}} I_{\mathrm{l}_2} + V_{\mathrm{dt}_2} - K_{\mathrm{p1}_2}[N_{\mathrm{d}_2} + K_{\mathrm{p2}_2}(V_{\mathrm{ref}_2} - V_{\mathrm{q}_2})] - V_{\mathrm{d}_2} \\ K_{\mathrm{p1}_2} I_{\mathrm{q}_2} + \omega_{\mathrm{s}} L_{\mathrm{c}} I_2 + V_{\mathrm{qt}_2} - K_{\mathrm{p1}_2}\left(\dfrac{2I_{\mathrm{dc}_2} V_{\mathrm{dc}_2} - I_{\mathrm{d}_2} V_{\mathrm{dt}_2}}{V_{\mathrm{qt}_2}}\right) - V_{\mathrm{q}_2} \end{bmatrix}. \tag{3.44}$$

The final state, the integration state M_{dc} from the DC grid model, is initialized as

$$M_{\mathrm{dc}} = I_{\mathrm{dc}_2} - K_{\mathrm{pdc}}(V_{\mathrm{dc}_{\mathrm{ref}}} - V_{\mathrm{dc}_2}). \tag{3.45}$$

Chapter 4

Implementation Results

This chapter contains the results obtained from implementing Gear's method for the four test systems. Each system was first implemented without any strategies applied, forming the base case simulation. The base case was then verified, before the effects of applying the different strategies from Section 2.5.3 were investigated, along with an additional verification.

4.1 Simulation of Single Machine to Infinite Bus

Table 4.1 shows the load flow solution which formed the basis for the initial guess for the system's steady-state solution. Following this, the system was initialized as described in Section 3.2.6.

Table 4.1: Steady-state load flow solution of SMIB case. Bus 1 is slack bus.

	Voltage		Net power injection	
Bus number	Magnitude [pu]	Angle [rad]	Active [pu]	Reactive [pu]
0	1.0047	0.0499	0.9998	0.1000
1	1.0000	0.0000	-0.9988	-0.0498

The calculated eigenvalues of the system are given in Table 4.2 and plotted in the complex plane in Figure 4.1. The system has a total of seventeen eigenvalues, of which eight are oscillating and none are unstable.

Table 4.2: Eigenvalues of SMIB system.

Eigenvalue	Real component	Imaginary component
1	-1000.0000	-
2	-100.0000	-
3	-46.3547	-
4	-43.3014	-
5	-30.9983	-
6	-19.9969	-
7, 8	-12.5250	±2.6087
9	-5.1545	-
10, 11	-0.5382	±15.1785
12	-0.3333	-
13, 14	-0.2822	±0.4470
15	-0.1527	-
16, 17	-0.0367	±0.4587

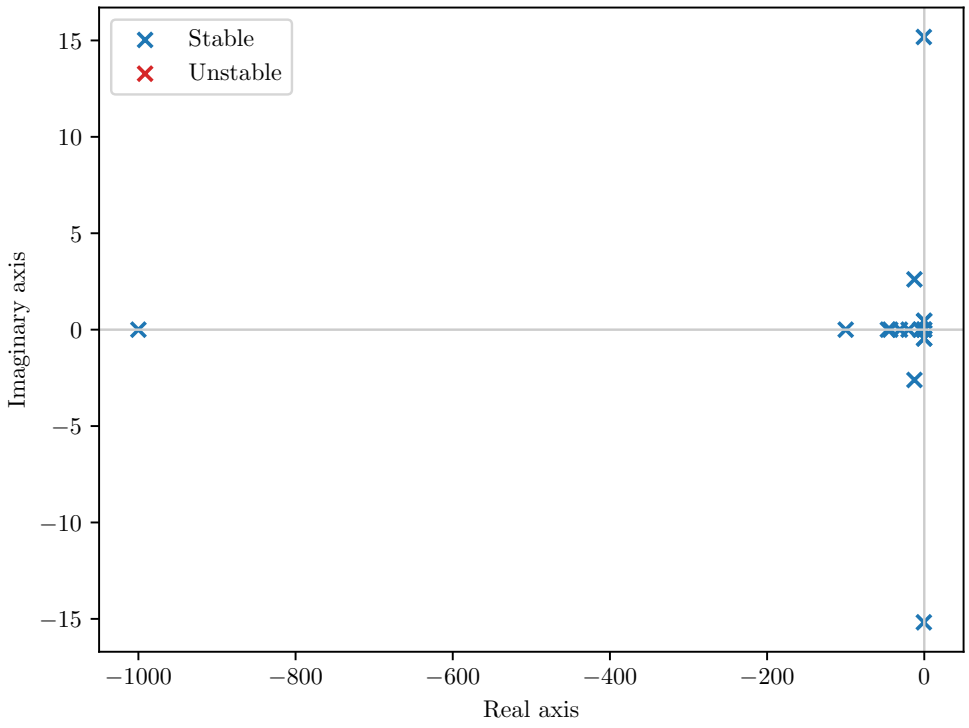


Figure 4.1: Eigenvalue plot for SMIB system.

4.1.1 Results of Base Case Implementation of Single Machine to Infinite Bus

Throughout this thesis, every simulation case, regardless of system or applied strategy, employed the same upper and lower step length limits and initial step length,

as recommended by [6]. The step length limits were set somewhat arbitrarily as $h^{\max} = 1 \times 10^{-2}\,\mathrm{s}$ and $h^{\min} = 1 \times 10^{-5}\,\mathrm{s}$. The effect of changing these limits were not explored, apart from some brief, non-systematic testing.

The base case simulation was performed with no strategies applied, apart from applying the upper and lower step length limits. The base case simulation's performance is shown in Table 4.3.

Table 4.3: Performance metrics for base case simulation of SMIB system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]
15.01	2367	599	1.00×10^{-2}	2.66×10^{-4}	6.34×10^{-3}

Figure 4.2 shows the simulated excitation emf E_f and corresponding step length. E_f was the state variable with the largest rate of change and is therefore highlighted.

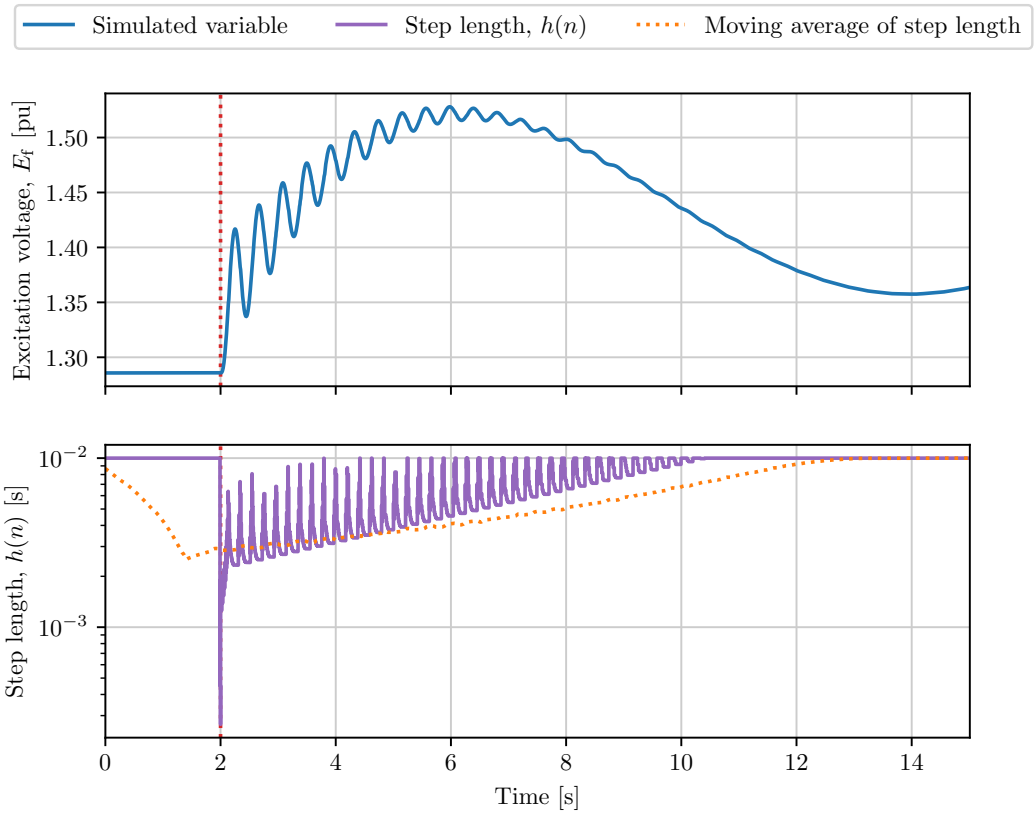


Figure 4.2: Base case simulation of SMIB system for $t \in [0, 15]$ s, with voltage reference change at $t = 2\,\mathrm{s}$.

Figure 4.2 shows how the step length promptly increased form its initial value to the upper limit during the steady-state portion of the simulation, before sharply falling at the time of the discrete event. Following this, the step length increased quickly a few times, before oscillating with sharp, uneven peaks. As the simulation progressed, the step length peaks periodically reached the upper step length limits, while the

valleys increased gradually. Just after 10 s, the step length settled at the upper limit as the fastest modes were damped out, and remained there for the remainder of the simulation.

In addition, from Table 4.3 it is worth noting that the simulation continued for 0.01 s longer that what was set as the termination time. At the end of the simulation the step length had settled at its upper limit. This, in combination with the fact that the penultimate calculation was done at 14.9994 s, meant that the termination condition was not fulfilled before 15.0094 s.

Note that the step length plot in Figure 4.2 is semi-logarithmic.

4.1.2 Verification of Single Machine to Infinite Bus Simulation Results

A verification calculation was performed as described in Section 3.1.1. Figure 4.3 shows the verified solution overlain the base case implementation of the SMIB system using Gear's method, as well as the step length.

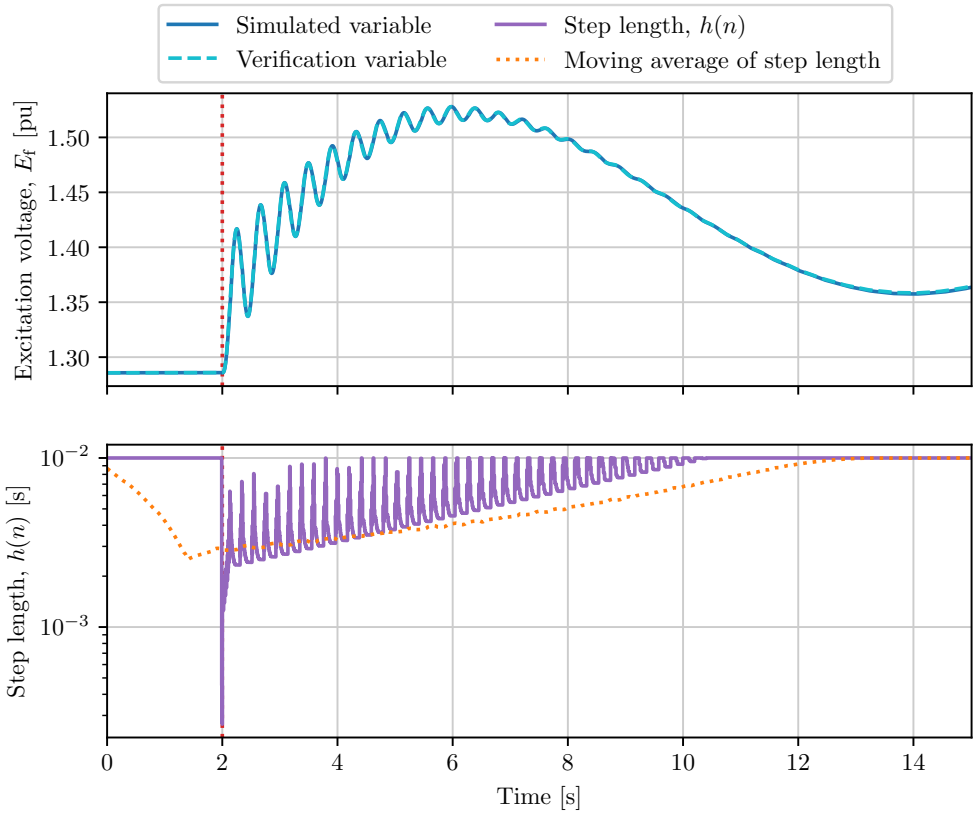


Figure 4.3: Verification of base case simulation of SMIB system for $t \in [0, 15]$ s, with voltage reference change at $t = 2$ s.

From Figure 4.3 it is evident that the base case solution strongly followed the verified solution. The largest MSE was calculated for the filtered frequency deviation signal V_w of the PSS, with a value of 2.43×10^{-4}.

4.1.3 Results of Implementation of Single Machine to Infinite Bus with Strategies

Following the verification of the base case implementation, simulations were carried out to investigate the effects of the proposed step length strategies from Section 2.5.3. Scaling factors γ of 1.0, 0.9, 0.8, 0.7, 0.6, and 0.5 were tested, giving six possible parameter values. This, combined with boolean true and false values for the step length increase and decrease limits, and fixing the step length for either at least 1 or 15 steps, gave a total of 48 parameter combinations. The goal was to determine each strategy's effect on the simulation. Therefore, the average number of step length changes, the average mean step length, and the average maximum MSE for each parameter value were calculated. In other words, one parameter value was held constant and the averages were determined of all the combinations satisfying that specific condition. Table A.1 in Appendix A contains the results of all the simulated combinations, on which Table 4.4 is based.

Table 4.4: Performance metrics for simulation strategies for SMIB case.

(a) Step scaling factor.

Step scaling factor	Average number of step changes	Average mean step length [s]	Average maximum MSE
1.0	339.38	6.48×10^{-3}	4.28×10^{-4}
0.9	241.63	6.36×10^{-3}	5.12×10^{-4}
0.8	215.63	6.15×10^{-3}	4.87×10^{-4}
0.7	331.88	5.99×10^{-3}	7.63×10^{-4}
0.6	407.00	5.90×10^{-3}	5.68×10^{-4}
0.5	758.86	5.61×10^{-3}	6.17×10^{-4}

(b) Step increase limit.

Step increase limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	423.75	5.84×10^{-3}	7.30×10^{-4}
False	315.95	6.39×10^{-3}	3.72×10^{-4}

(c) Step length fixed.

Step fixed for [steps]	Average number of step changes	Average mean step length [s]	Average maximum MSE
1	681.48	6.04×10^{-3}	5.70×10^{-4}
15	103.92	6.15×10^{-3}	5.57×10^{-4}

Table 4.4: Performance metrics for simulation strategies for SMIB case *(continued)*.

(d) Step decrease limit.

Step decrease limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	392.21	6.09×10^{-3}	2.51×10^{-4}
False	352.00	6.10×10^{-3}	9.19×10^{-4}

The results in Table 4.4 show that as the step scaling factor decreased, the average number of step length changes at first dropped, reached a minimum, before it increased to a higher value. The average mean step length was strictly decreasing along with reduced scaling factor, while the average maximum MSE generally rose with decreasing scaling factor.

A scaling factor of 0.8 gave the lowest average number of step changes, while a scaling factor of 1.0 resulted in the highest average mean step length. A scaling factor of 1.0 also resulted in the smallest average maximum MSE, while scaling by 0.7 gave rise to the highest MSE.

Fixing the step length for at least 15 steps greatly reduced the number of step length changes performed. It also resulted in a slightly higher average mean step length, and a lower average maximum MSE.

Both the limiting strategies resulted in poorer performance in terms of the average number of step length changes and the average mean step length, compared to not imposing such limits. Imposing a step decrease limit did, however, reduce the average maximum MSE, in contrast to imposing a step decrease limit.

No single case resulted in the best score on all three criteria. Still, by using the smallest average number of step length changes, largest average mean step length and smallest average maximum MSE, across all combinations, as ideal scores, (3.2) could be used to determine the overall best case. With $d = 1.719 \times 10^{-1}$, the combination of not applying a scaling factor, fixing the step length for at least 15 steps, not imposing a step increase limit, but imposing a step decrease limit, was used as the overall best case.

Table 4.5 and Figure 4.4 relate to the highlighted well performing combination, showing the performance results and simulation results, including the verified solution.

Table 4.5: Performance metrics for overall best simulation case of SMIB system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Max. MSE
15.01	2204	77	1.00×10^{-2}	1.00×10^{-3}	6.81×10^{-3}	1.29×10^{-4}

Figure 4.4 shows how, in the overall best performing case, the large step length reduction at the discrete event was eliminated. The general step length trend was

similar to that in the base case, as shown by the moving average curve. The fluctuations were not as sharp and remained in a narrower band below the upper limit. Still, from the voltage reference was changed, the step length required approximately the same amount of time to settle at the upper limit, as compared to the base case.

The verification curve still closely aligned with the result from the overall best simulation. Table 4.5 also indicates that a smaller maximum MSE was achieved, compared to the base case, and no discernible error was recorded with respect to the capture of the disturbance instance.

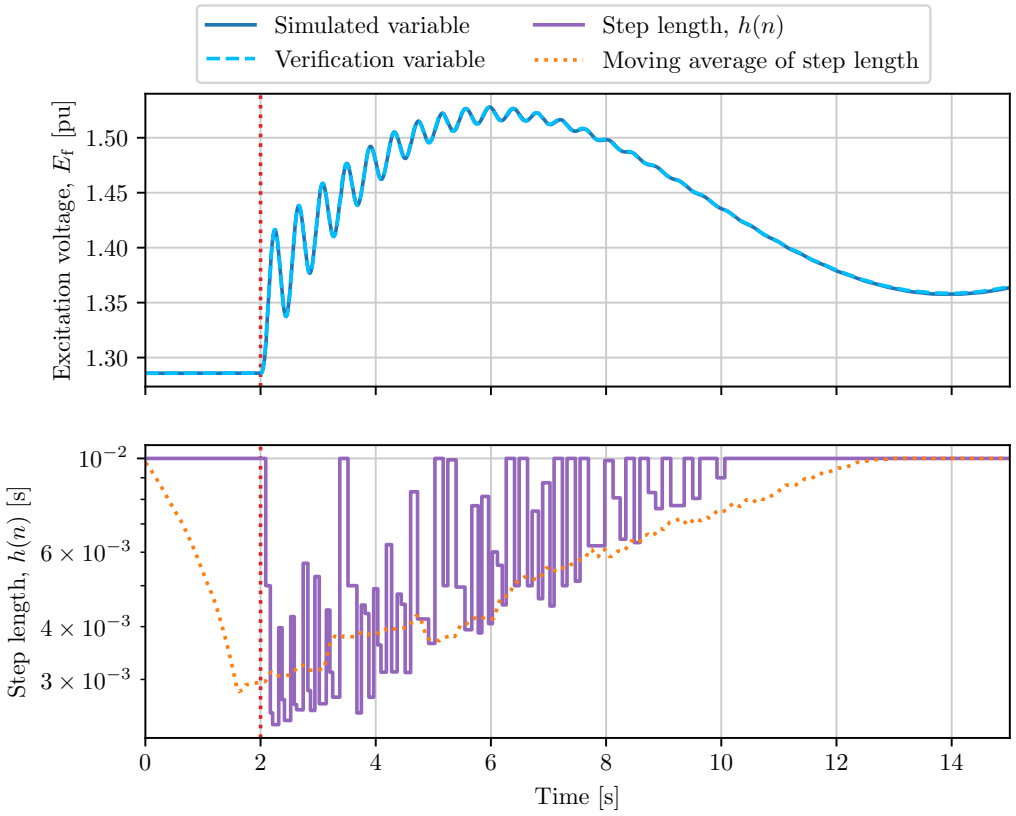


Figure 4.4: Overall best case simulation of SMIB system for $t \in [0, 15]$ s, with voltage reference change at $t = 2$ s and verification.

Outlier Simulation Cases

During the testing of the different parameter values, three outlier cases were produced. Table 4.6 shows an excerpt from the full result table in Appendix A, showing the relevant parameter combinations. The three cases resulted in large step length fluctuations over the entire permitted range and simulation results considerably different from the other combinations and the verification. Figure 4.5 shows the simulation result from the first outlier case, along with the verified result. The three outliers all had similar responses.

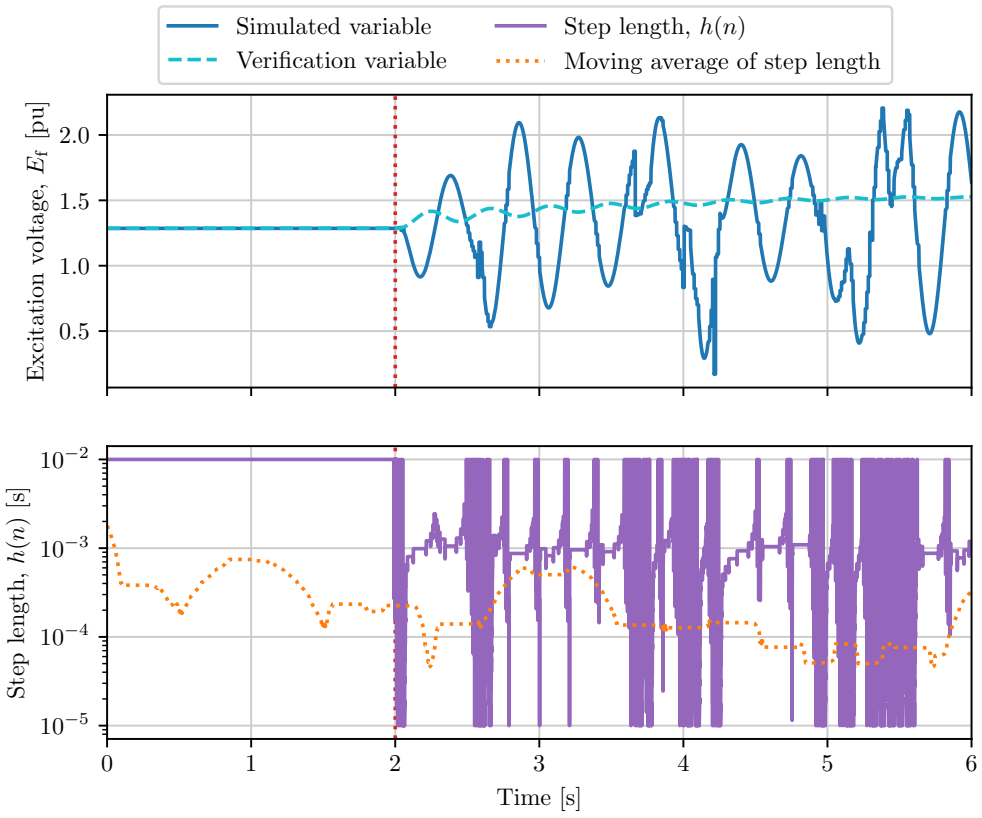


Figure 4.5: Outlier simulation case of SMIB system for $t \in [0, 6]$ s, with voltage reference change at $t = 2\,\text{s}$ and verification.

The three cases were considered as failed integrations, and were therefore not included when calculating the average performance metrics scores. In addition, as is evident from Figure 4.5, the three cases failed before completing a simulation of 15 s, preventing a comparison to other parameter combinations.

Table 4.6: Parameter combinations of outlier cases for the SMIB system.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit
0.8	False	1	False
0.7	False	1	False
0.5	False	1	False

4.2 Simulation of Three Machines to Infinite Bus

The load flow results of the 3MIB system are presented in Table 4.7, and are comparable to the load flow solution found in [12].

Table 4.7: Steady-state load flow solution of 3MIB case. Buses 0-2 are generator buses, buses 3 and 4 are load buses, and bus 5 is slack bus.

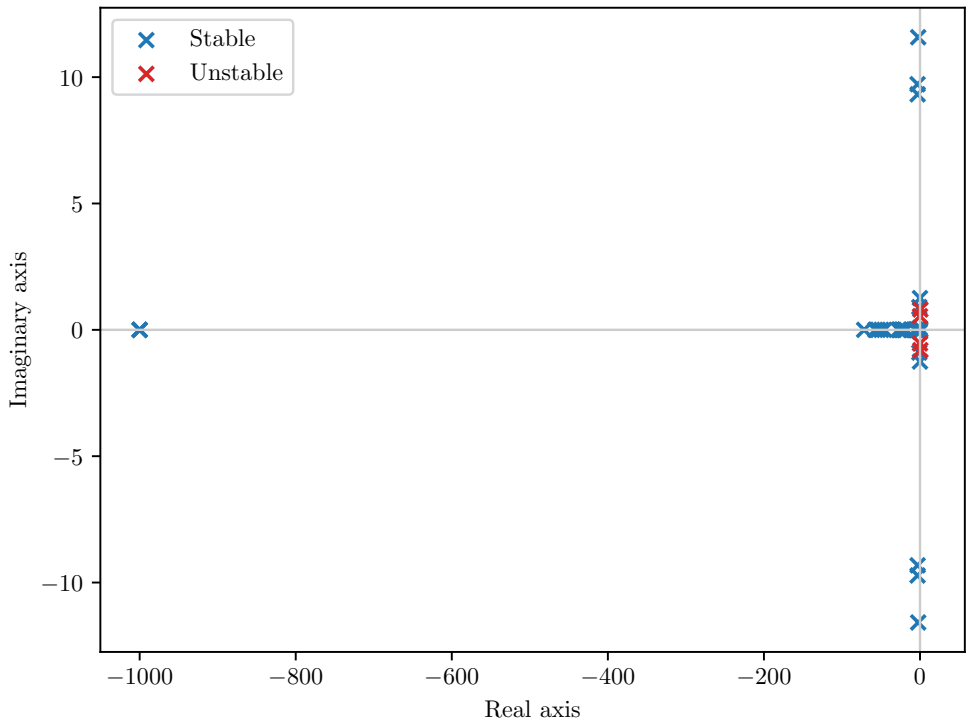
	Voltage		Net power injection	
Bus number	Magnitude [pu]	Angle [rad]	Active [pu]	Reactive [pu]
0	1.0402	0.8826	14.0400	4.3440
1	1.0402	0.8826	14.0400	4.3440
2	1.0201	0.3500	8.0000	4.6620
3	1.0171	0.7975	-13.9986	-1.0000
4	0.9729	0.2596	-20.0000	-1.0000
5	1.0000	0.0000	-2.0814	0.4974

Table 4.8 contains the numerically approximated eigenvalues of the simulated system. The system has 45 eigenvalues. Of these, sixteen form complex conjugate pairs, of which two pairs are unstable. Note that the real components of eigenvalues 1-3 appear to be equal, but are in fact distinct, given that a higher precision is used.

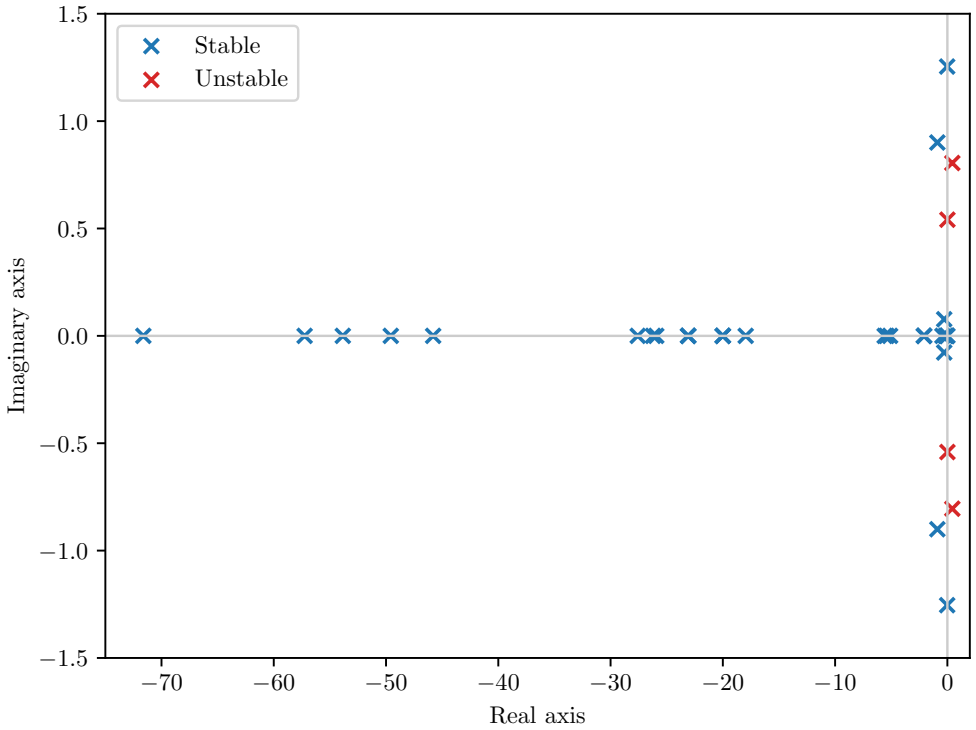
Figure 4.6 shows the eigenvalues plotted in the complex plane, indicating the two conjugate pairs constituting the unstable eigenvalues. Figure 4.6b also contains a zoomed in version of the complex plane to better highlight the clustering of poles.

Table 4.8: Eigenvalues of 3MIB system.

Eigenvalue	Real component	Imaginary component
1	-1000.0000	-
2	-1000.0000	-
3	-1000.0000	-
4	-71.6172	-
5	-57.2504	-
6	-53.8498	-
7	-49.5755	-
8	-45.8040	-
9	-27.5796	-
10	-26.2185	-
11	-25.9308	-
12	-23.0883	-
13	-23.0883	-
14	-23.0883	-
15	-20.0034	-
16	-20.0034	-
17	-20.0034	-
18	-17.9646	-
19	-5.5845	-
20	-5.4152	-
21	-5.1020	-
22, 23	-3.1094	±9.7206
24, 25	-3.0381	±9.3192
26, 27	-2.3303	±11.5796
28	-2.0815	-
29	-2.0812	-
30	-2.0809	-
31, 32	-0.8918	±0.9005
33	-0.4396	-
34, 35	-0.2825	±0.0774
36	-0.1566	-
37	-0.0260	-
38	-0.0260	-
39	-0.0260	-
40, 41	-0.0186	±1.2542
42, 43	0.0034	±0.5410
44, 45	0.4404	±0.8049



(a) All eigenvalues.



(b) Zoomed view.

Figure 4.6: Eigenvalue plot for 3MIB system.

4.2.1 Results of Base Case Implementation of Three Machines to Infinite Bus

By applying the same upper and lower step length limits as in the SMIB case in Section 4.1.1, but no other strategies, the base case simulation was performed. Table 4.9 shows the obtained performance metrics scores.

Table 4.9: Performance metrics for base case simulation of 3MIB system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]
25.00	3433	804	1.00×10^{-2}	1.00×10^{-5}	7.28×10^{-3}

Figure 4.7 shows the simulated angular frequency deviation $\Delta\omega_s$ for each of the three synchronous generators in the system, along with the step length. Figure 4.7 also includes a plot showing if saturation was enforced for at least one state variable at a given time step. This is in contrast to the SMIB system where the saturation limits were never met.

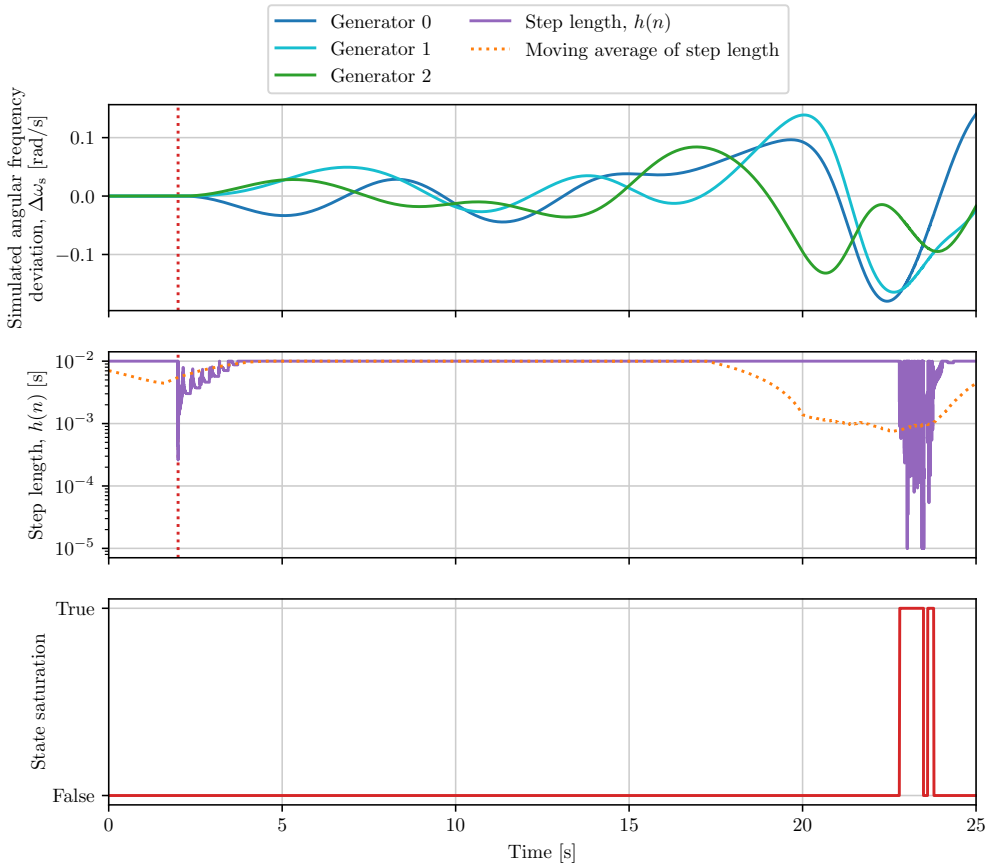


Figure 4.7: Base case simulation of 3MIB system for $t \in [0, 25]$ s, with voltage reference change at $t = 2$ s.

From Figure 4.7 it is evident that the step length initially progressed similar to in the SMIB base case in Figure 4.2. At the point of disturbance, the step length dropped from the upper limit, before entering into a oscillatory, yet increasing behavior. The shape of the peaks were similar to those in the SMIB base case, with sharp rises and falls similar to exponential decays. At around 4 s, the step length again settled at its upper limit.

However, once saturation was enforced, the step length again fell, eventually reaching the lower limit. From there it oscillated rapidly before once again falling as saturation ended. The step length gradually recovered, making a last considerable dip during the second saturation period, before it once again reached the upper limit for the remainder of the simulation.

4.2.2 Verification of Three Machines to Infinite Bus Simulation Results

Figure 4.8 shows the verified solution along with the solution obtained using Gear's method.

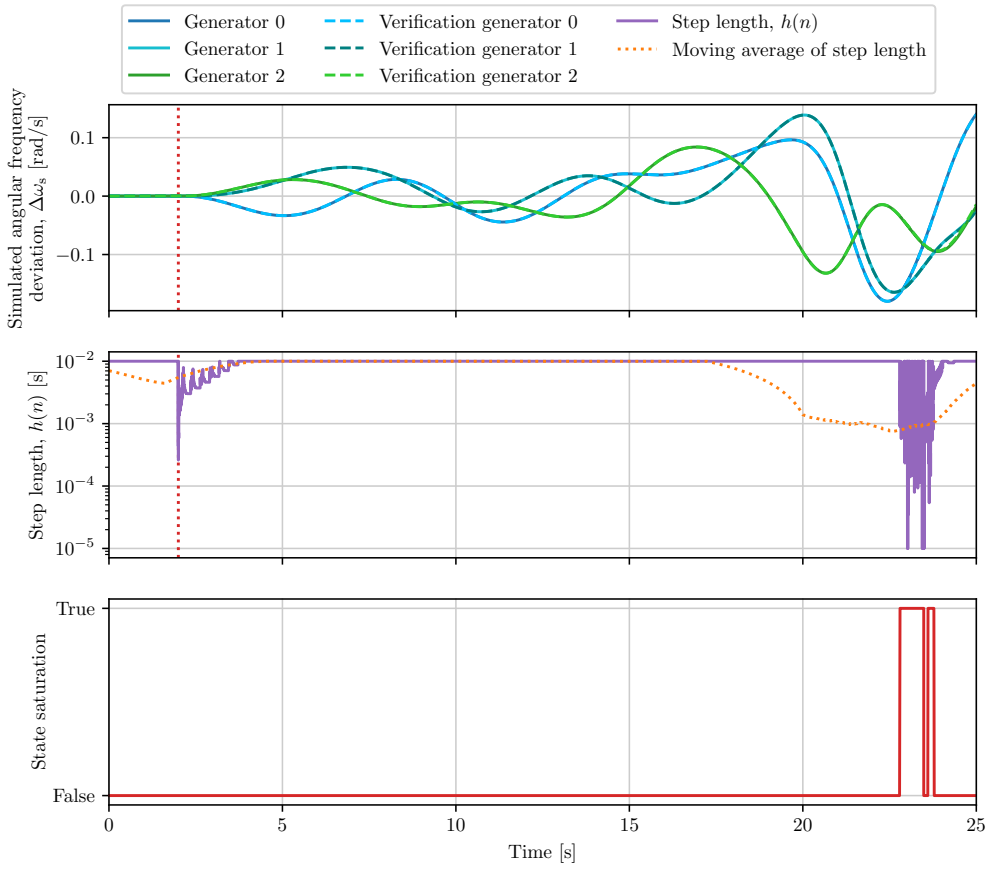


Figure 4.8: Verification of base case simulation of 3MIB system for $t \in [0, 25]$ s, with voltage reference change at $t = 2$ s.

As with the SMIB case, visual inspection indicated that there was a large overlap between the two solutions. However, the largest MSE was found for the regulator

output V_{r_0} of generating unit 0, with a value of 2.44×10^{-1}, considerably larger than for the SMIB case.

Note that the verification calculation in MATLAB did not include the effects of saturation.

4.2.3 Results of Implementation of Three Machines to Infinite Bus with Strategies

Again, following the verification of the base case results, the 48 different parameter combinations were applied to the method. The full results are given in Table A.2 in Appendix A, and the averaged values are given in Table 4.10.

Table 4.10: Performance metrics for simulation strategies for 3MIB case.

(a) Step scaling factor.

Step scaling factor	Average number of step changes	Average mean step length [s]	Average maximum MSE
1.0	446.88	8.20×10^{-3}	3.37×10^{-2}
0.9	506.75	7.94×10^{-3}	4.50×10^{-3}
0.8	487.38	7.67×10^{-3}	4.83×10^{2}
0.7	463.38	7.84×10^{-3}	8.34×10^{2}
0.6	455.43	7.93×10^{-3}	3.94×10^{-3}
0.5	578.86	7.80×10^{-3}	4.10×10^{-3}

(b) Step increase limit.

Step increase limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	617.88	7.56×10^{-3}	5.05×10^{-3}
False	347.55	8.27×10^{-3}	4.79×10^{2}

(c) Step length fixed.

Step fixed for [steps]	Average number of step changes	Average mean step length [s]	Average maximum MSE
1	983.95	6.78×10^{-3}	4.79×10^{2}
15	37.25	8.93×10^{-3}	1.59×10^{-3}

(d) Step decrease limit.

Step decrease limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	450.92	8.10×10^{-3}	3.80×10^{-3}
False	529.68	7.68×10^{-3}	4.79×10^{2}

From Table 4.10 it can bee seen that along with decreasing scaling factor, the average number of step length changes initially rose, then fell to a minimum before gradually increasing again. The average mean step length followed an inverted path, initially falling, rising to a plateau, before once again falling. The maximum MSE peaked sharply for the middle scaling factors.

Not applying a scaling factor resulted in the fewest number of average step length changes and once again gave the highest average mean step length. Not scaling did, however, result in a large average maximum MSE, the value being an order of magnitude larger than for the smaller scaling factors. Still, scaling factors of 0.8 and 0.7 gave considerably larger average maximum MSEs, compared to the other scaling factors. The smallest average maximum MSE was obtained with a scaling factor of 0.6.

Not applying a step increase limit resulted in a considerably fewer number of step length changes, as well as a higher average mean step length. It did, however, result in a markedly larger average maximum MSE, when compared to combinations where an increase limit was imposed.

Continuing, applying a step decrease limit resulted in improved performance across all three criteria, compared to not imposing such a limit. Preventing the step length from changing more than for every fifteenth step once again resulted in a considerably lower average number of step length changes. Moreover, the average mean step length was larger, and the average maximum MSE lower than when the step length was allowed to change every step.

Once again, no single parameter combination achieved the top score on all three criteria. However, by calculating the Euclidean distances according to (3.2) and defining the ideal points similarly as in Section 4.1.3, the overall best combination was found to be not applying a scaling factor, limiting the permitted frequency of step length changes, and not applying any increase or decrease limits. This combination gave $d = 1.202 \times 10^{-1}$.

Table 4.11 shows the performance metrics for the parameter combination with the lowest d-value, and Figure 4.9 shows the simulated results along with the verified solution.

Table 4.11: Performance metrics for overall best simulation case of 3MIB system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Max. MSE
25.00	2661	26	1.00×10^{-2}	1.00×10^{-3}	9.39×10^{-3}	1.38×10^{-3}

Figure 4.9 shows how, in the overall best performing case, the large step length reductions at the reference change and saturation were reduced. The step length fluctuations were not as rapid and remained closer to the upper limit, as indicated by the mean step length being longer than in the base case. As in the SMIB case, the step length still required a similar amount of time to settle at the upper limit after the disturbances in both cases.

Moreover, the maximum MSE was smaller in the best case, compared to the base case, as indicated by Table 4.11. Visual inspection of Figure 4.9 also supports a strong agreement between the solution from Gear's method and the verified MATLAB solution.

Interestingly, whereas the base case resulted in two distinct saturation periods, the best case simulation only resulted in a single one. The single saturation period spanned both saturation periods in the base case simulation.

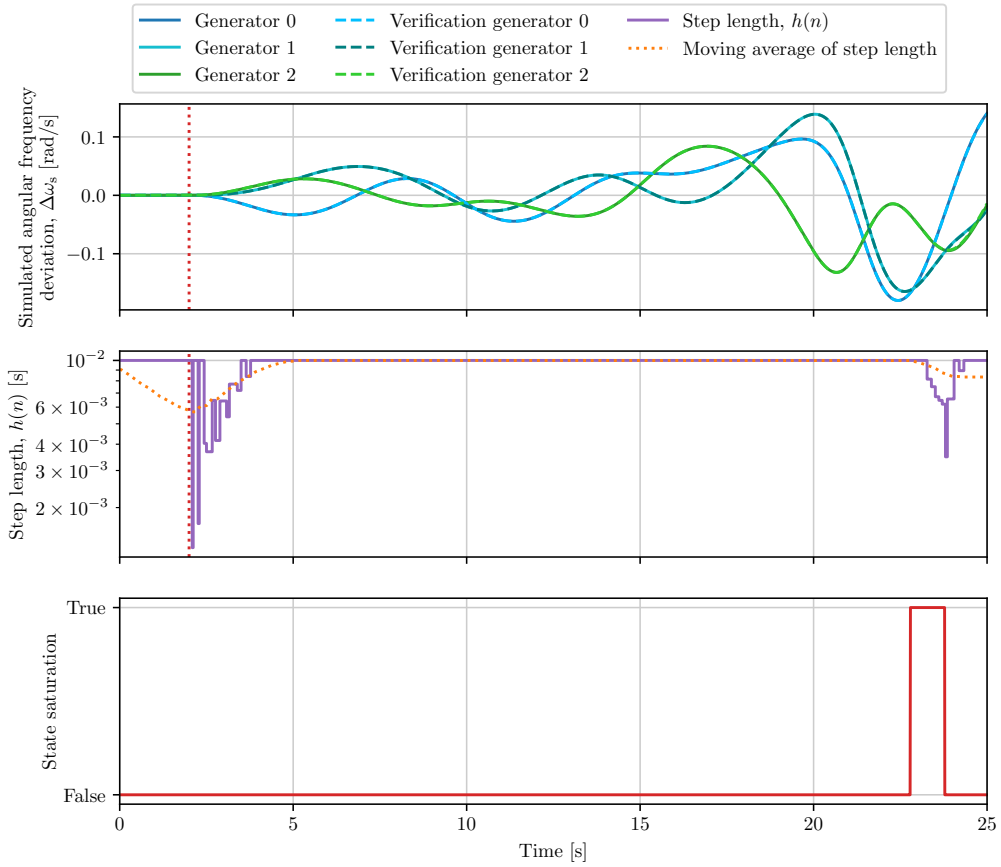


Figure 4.9: Overall best case simulation of 3MIB system for $t \in [0, 25]$ s, with voltage reference change at $t = 2\,\text{s}$ and verification.

Outlier Simulation Cases

Also for the 3MIB system, outlier cases were produced during the parameter testing. Two cases were considered as outliers and not included in the averaging calculations. Table 4.12 shows the relevant parameter combinations, and Figure 4.10 shows the simulation result of the first outlier case together with the verified solution.

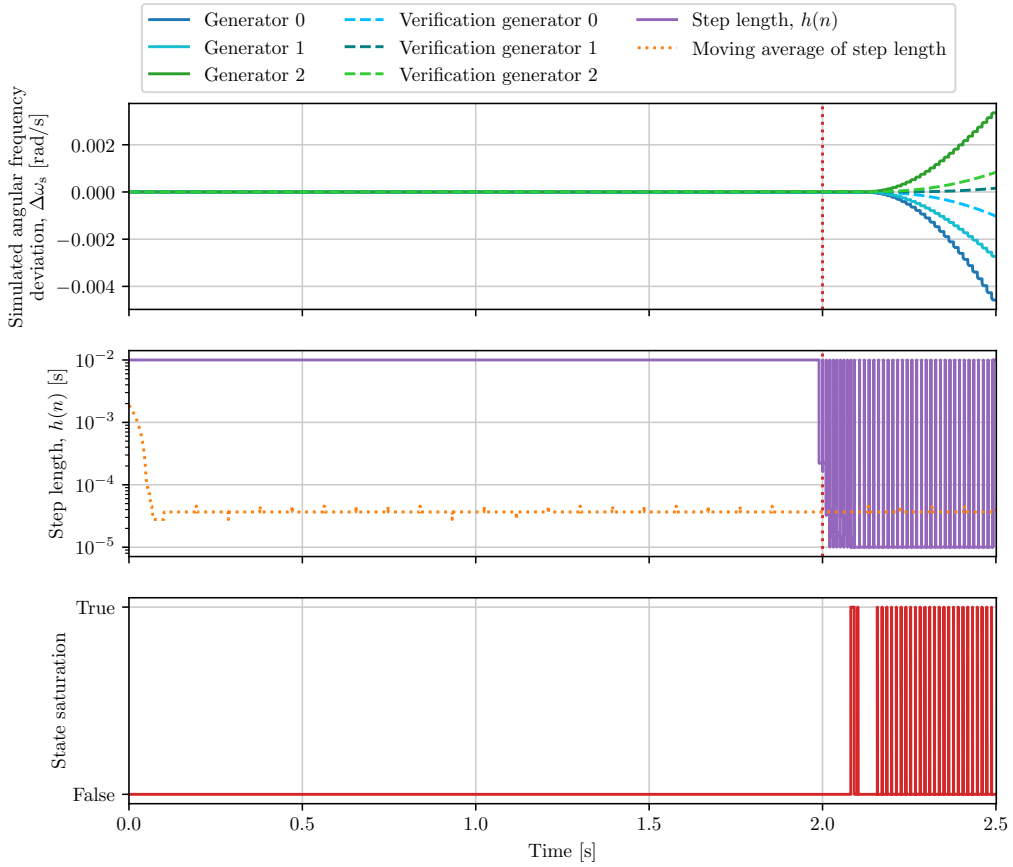


Figure 4.10: Outlier simulation case of 3MIB system for $t \in [0, 2.5]$ s, with voltage reference change at $t = 2$ s and verification.

The two outliers were produced by the same combinations that produced two of the outlier cases for the SMIB system, and the effects were similar. The two outlier cases failed shortly after the reference change at $t = 2$ s, due to the Newton-Raphson iterations not converging in the corrector step. The simulated variables also deviated considerably from the verified solution and contained a clear ripple. In addition, saturation was enforced earlier, and the step length oscillated rapidly between the upper and lower limits.

Table 4.12: Parameter combinations of outlier cases for the 3MIB system.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit
0.6	False	1	False
0.5	False	1	False

4.3 Simulation of System with Static Compensator

Table 4.13 gives the load flow solution of the STATCOM system.

Table 4.13: Steady-state load flow solution of STATCOM case. Buses 0 and 1 are generator buses, bus 2 is load and STATCOM bus, and bus 3 is slack bus.

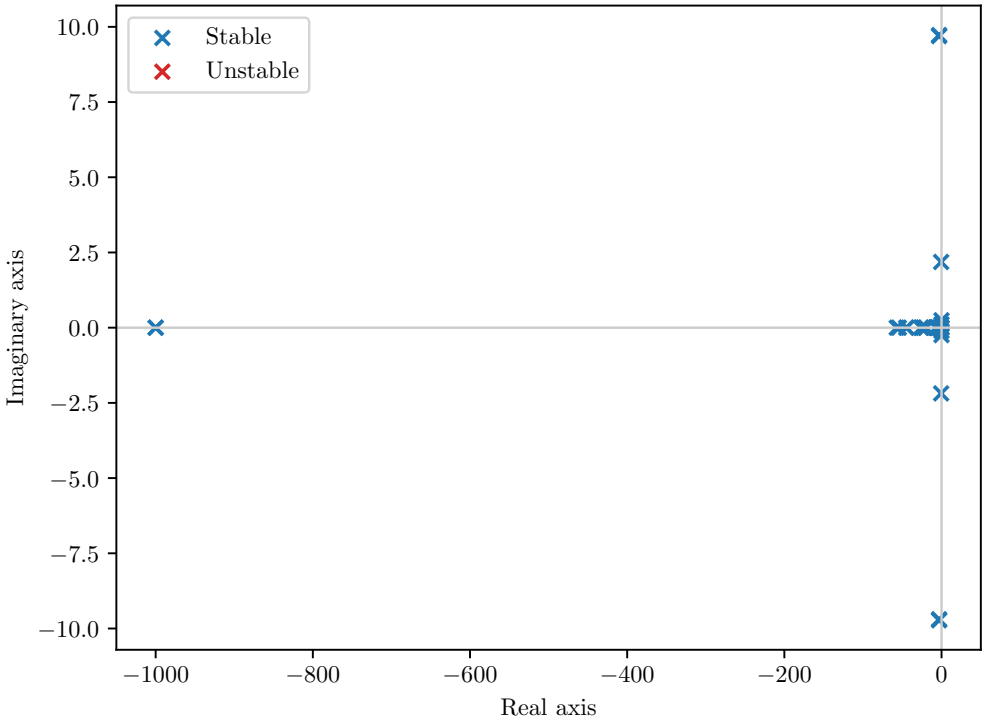
	Voltage		Net power injection	
Bus number	Magnitude [pu]	Angle [rad]	Active [pu]	Reactive [pu]
0	0.9875	0.0103	5.0000	1.0000
1	0.9801	0.0305	8.0000	0.0000
2	0.9815	-0.0228	-14.9993	-2.0000
3	1.0000	0.0000	1.9993	1.6741

The eigenvalues of the STATCOM system are presented in Table 4.14 and Figure 4.11. As in Table 4.8 for the 3MIB case, the two first eigenvalues are also here indistinguishable with the given decimal precision.

Table 4.14: Eigenvalues of STATCOM system.

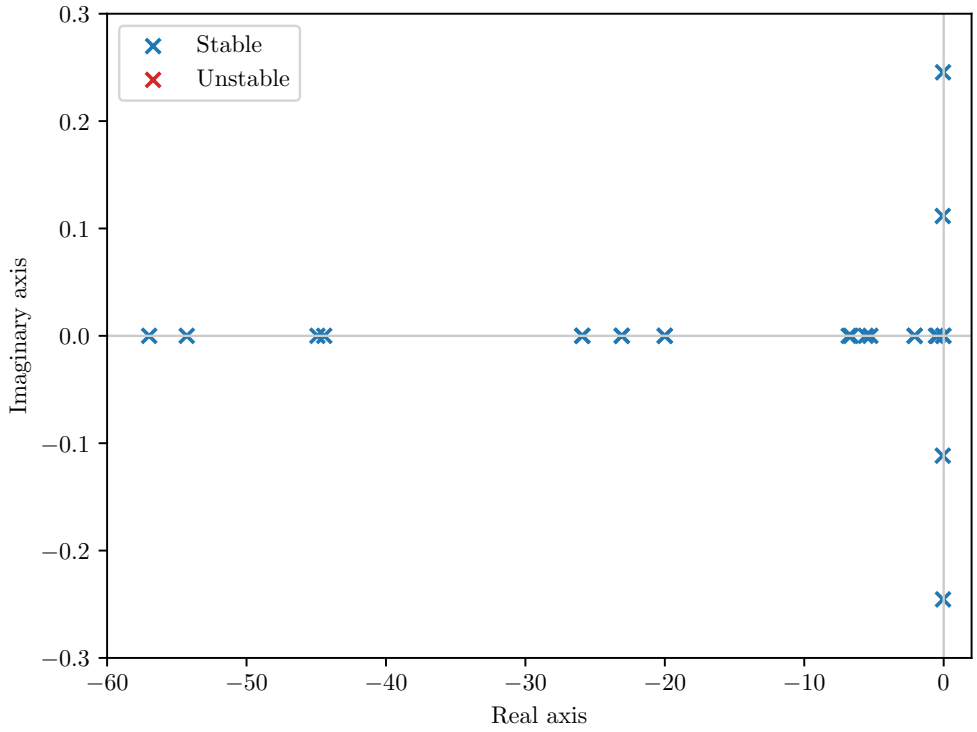
Eigenvalue	Real component	Imaginary component
1	-1000.0000	-
2	-1000.0000	-
3	-56.9844	-
4	-54.2877	-
5	-44.9112	-
6	-44.4322	-
7	-25.9339	-
8	-25.9053	-
9	-23.0872	-
10	-23.0871	-
11	-20.0052	-
12	-20.0052	-
13	-6.8136	-
14	-6.6587	-
15	-5.5875	-
16	-5.2582	-
17, 18	-3.1130	±9.7256
19, 20	-3.0958	±9.6933
21	-2.0837	-
22	-2.0836	-
23	-0.5527	-
24, 25	-0.4997	±2.1844
26	-0.4358	-
27, 28	-0.0671	±0.1116
29, 30	-0.0454	±0.2455
31	-0.0260	-
32	-0.0260	-

The STATCOM system has 32 eigenvalues with five oscillating pairs and no unstable eigenvalues. Figure 4.11b also highlights the main pole cluster by presenting a zoomed view.



(a) All eigenvalues.

Figure 4.11: Eigenvalue plot for STATCOM system.



(b) Zoomed view.

Figure 4.11: Eigenvalue plot for STATCOM system *(continued)*.

4.3.1 Results of Base Case Implementation of System with Static Compensator

Table 4.15 presents the performance metrics scores of the base case implementation of the STATCOM system. As in the previous cases, no strategies were applied, except the upper and lower step length limits presented for the SMIB case in Section 4.1.1.

Table 4.15: Performance metrics for base case simulation of STATCOM system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]
10.00	1433	169	1.00×10^{-2}	1.00×10^{-5}	6.98×10^{-3}

In addition, Figure 4.12 shows the simulated angular frequency deviation $\Delta\omega_s$ for the two synchronous generators in the system. The step length is also included in Figure 4.12, along with the network switching instance. The time interval presented in Figure 4.12 was chosen so as to highlight the region with the most step length adjustments. Note that none of the state variables reached their saturation limits during the simulation, and a saturation curve is therefore not included in Figure 4.12.

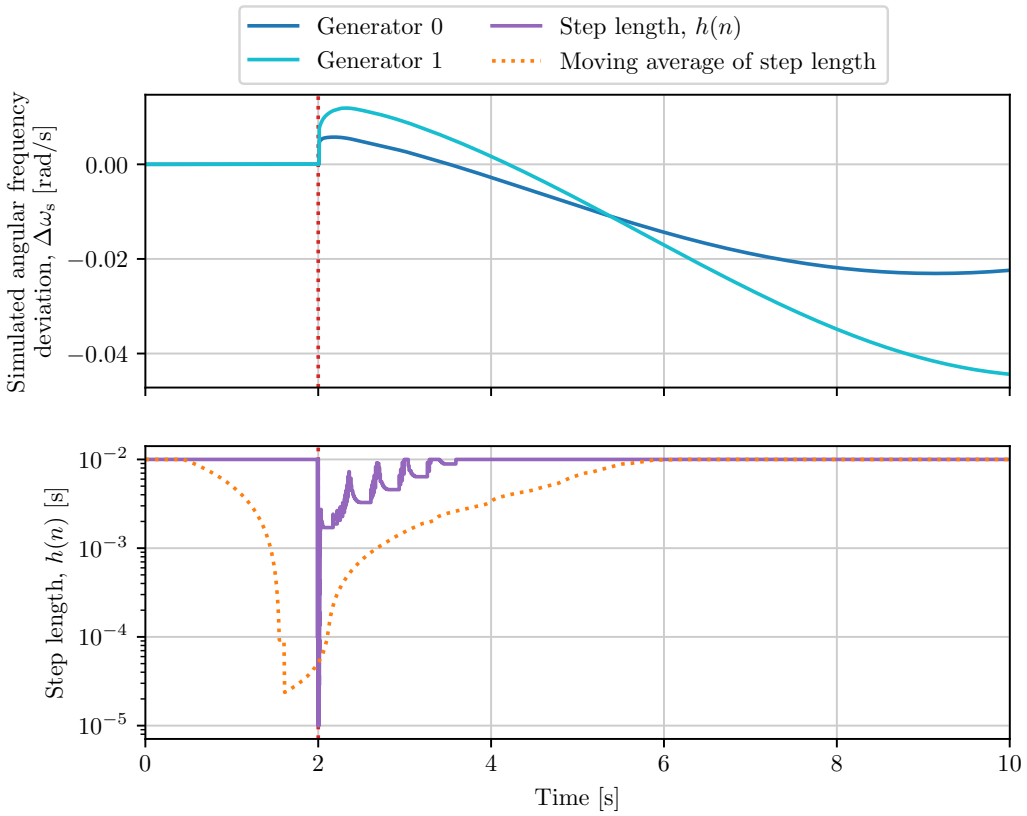


Figure 4.12: Base case simulation of STATCOM system for $t \in [0, 10]$ s, with islanding at $t = 2$ s and reconnection 0.01 s later.

From Figure 4.12 it can be seen that the step length initially settled at the upper limit, similar to in the two past cases. By close inspection, Figure 4.12 further reveals that the step length initially dropped just before the disconnection occurred, before falling even further to the lower limit at the disconnection instant. The step length then slightly rose, before once again falling to the floor value as the system was reconnected. Following this, the step length quickly recovered, albeit reaching the upper limit through a path consisting of uneven, steadily increasing oscillations. The general shape of the oscillations were similar to those observed for the SMIB and 3MIB cases, with pointed peaks and flattened valleys.

4.3.2 Verification of System with Static Compensator Simulation Results

The verified solution for the STATCOM system, along with the solution from the base case implementation, are shown in Figure 4.13. The maximum MSE was calculated for the transducer output V_{t_1} at bus 1 as 1.99. Not only is this an appreciably higher value than for the two machine to infinite bus systems, but Figure 4.13 also reveals a notable difference between the two solution methods' results.

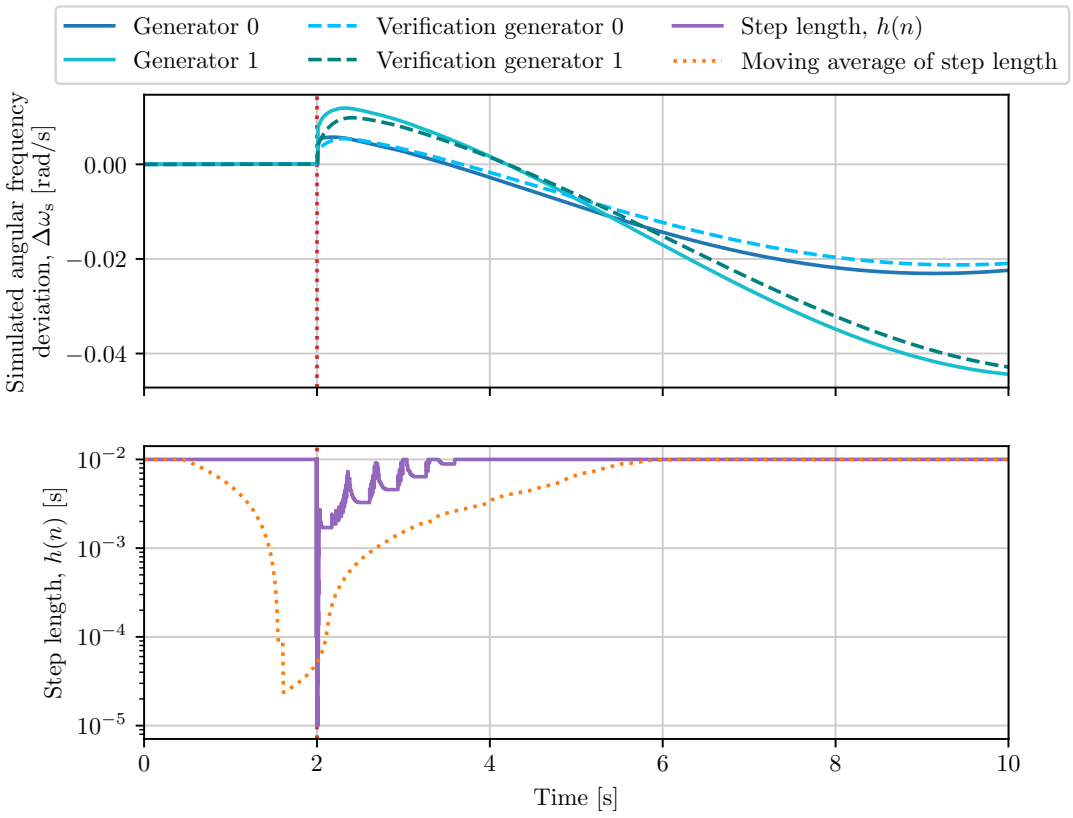


Figure 4.13: Verification of base case simulation of STATCOM system for $t \in [0, 10]$ s, with islanding at $t = 2$ s and reconnection 0.01 s later.

4.3.3 Results of Implementation of System with Static Compensator with Strategies

Continuing from the verification, Table 4.16 gives the averaged values from the application of the different strategy parameter combinations. Table A.3 in Appendix A gives the full results.

Table 4.16: Performance metrics for simulation strategies for STATCOM case.

(a) Step scaling factor.

Step scaling factor	Average number of step changes	Average mean step length [s]	Average maximum MSE
1.0	73.00	7.98×10^{-3}	5.77×10^{-1}
0.9	80.17	7.85×10^{-3}	5.73×10^{-1}
0.8	45.50	7.67×10^{-3}	6.11×10^{-1}
0.7	105.00	7.32×10^{-3}	6.44×10^{2}
0.6	59.00	7.47×10^{-3}	3.25×10^{-1}
0.5	86.80	7.31×10^{-3}	2.93×10^{-1}

Table 4.16: Performance metrics for simulation strategies for STATCOM case *(continued)*.

(b) Step increase limit.

Step increase limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	82.28	7.35×10^{-3}	4.51×10^{-1}
False	66.88	7.91×10^{-3}	2.42×10^{2}

(c) Step length fixed.

Step fixed for [steps]	Average number of step changes	Average mean step length [s]	Average maximum MSE
1	189.60	7.13×10^{-3}	3.87×10^{2}
15	27.29	7.81×10^{-3}	3.65×10^{-1}

(d) Step decrease limit.

Step decrease limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	25.42	8.14×10^{-3}	1.76×10^{-1}
False	102.09	7.33×10^{-3}	1.76×10^{2}

Table 4.16 shows how the average number of step changes rose and fell multiple times with decreasing scaling factor. The average mean step length, however, fell relatively steadily with decreasing scaling factor. For the average maximum MSE, a scaling factor of 0.7 resulted in a value three orders of magnitude larger than for the other values.

Scaling by 0.8 gave rise to the lowest average number of step changes, while not scaling gave the largest average mean step length. Applying a scaling factor of 0.5 resulted in the lowest average maximum MSE.

By applying a step increase limit, both performance metrics scores were worse compared to not applying an increase limit. It did, however, result in a considerably smaller average maximum MSE.

Furthermore, limiting the permitted frequency of step length adjustments gave a favorable score on all three performance metrics. The same was true for applying a step decrease limit.

As in the previous cases, no single parameter combination resulted in the highest score on all three performance metrics. Using (3.2) to calculate the Euclidean distances, with the ideal points defined as in the SMIB case in Section 4.1.3, the overall best case was determined to be the combination of not scaling, limiting the permitted step length adjustment frequency, applying a step length decrease limit, but no step length increase limit. By doing so, a d-score of 3.704×10^{-1} was achieved. Table 4.17 and Figure 4.14 relate to the overall best performing case, showing the

performance metrics scores and simulation results, including the verified solution, respectively.

Table 4.17: Performance metrics for overall best simulation case of STATCOM system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Max. MSE
10.00	1121	16	1.00×10^{-2}	1.00×10^{-3}	8.92×10^{-3}	6.41×10^{-2}

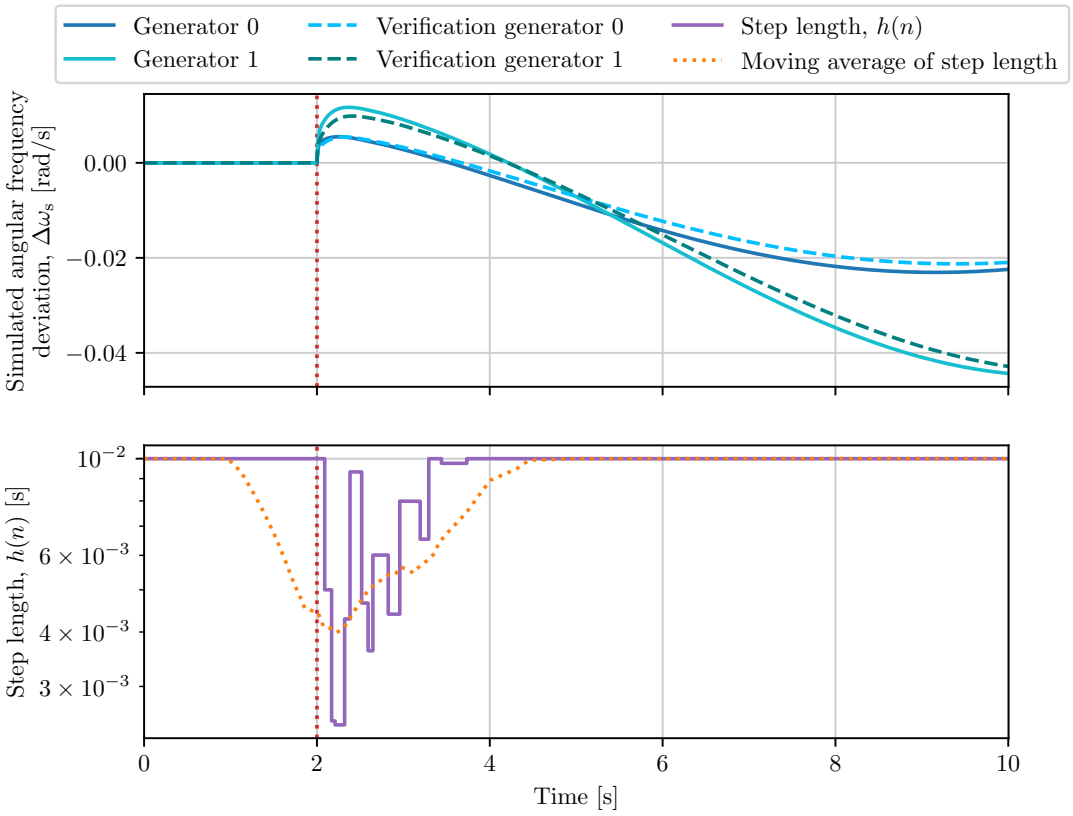


Figure 4.14: Overall best case simulation of STATCOM system for $t \in [0, 10]$ s, with islanding at $t = 2$ s, reconnection 0.01 s later, and verification.

As in the previous cases, the overall best strategy parameter combination reduced the step length fluctuations, both by increasing the shortest step length and reducing the number of fluctuations. Still, a similar amount of time was required for the step length to settle back at the ceiling value after the disturbance. The maximum MSE was also reduced when compared to the base implementation. Nevertheless, Figure 4.14 still shows a discernible difference between the solution obtained using Gear's method and the verified solution.

Outlier Simulation Cases

The testing of improvement strategies produced even more outlier cases for the STATCOM case, compared to the machine to infinite bus systems. A total of fourteen combinations resulted in outlier cases. In all the outlier cases the corrector was unable to converge for the time period between when the system was disconnected and when it was reconnected. Table 4.18 shows the parameter combinations of the failed cases.

Table 4.18: Parameter combinations of outlier cases for the STATCOM system.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit
1.0	True	1	True
1.0	False	1	True
0.9	True	1	True
0.9	False	1	True
0.8	True	1	True
0.8	False	1	True
0.7	True	1	True
0.7	False	1	True
0.6	True	1	True
0.6	False	1	True
0.6	False	1	False
0.5	True	1	True
0.5	False	1	True
0.5	False	1	False

4.4 Simulation of Multi-Terminal DC System

The branch currents $\boldsymbol{I}_{\text{cc}}$ were determined using the node connection matrix

$$\boldsymbol{Y}_{\text{cc}} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}. \tag{4.1}$$

The injected powers at the non-slack nodes, as well as the DC voltage reference are given in Table 4.19.

Table 4.19: Inputs to DC load flow solver for MTDC case.

Parameter	Value [pu]
P_{dc_0}	-1.5
P_{dc_1}	2.0
$V_{\text{dc}_{\text{ref}}}$	1.0

Table 4.20 gives the DC load flow solution divided into non-slack converter DC voltages, injected active power at slack converter, and injected as well as branch currents. From this, the system was initialized as described in Section 3.5.4.

Table 4.20: DC load flow solution for the MTDC case.

(a) Non-slack converter DC voltages.

Node number	DC voltage [pu]
0	0.9995
1	1.0012

(b) Injected active power at slack node.

Node number	Injected active power [pu]
2	-0.4968

(c) Injected node currents.

Node number	Injected DC current [pu]
0	-0.7504
1	0.9988
2	-0.2484

(d) Branch currents.

Branch	Branch current [pu]
0-1	-0.0017
0-2	-0.0005
1-2	0.0012

The eigenvalues of the system are presented in Table 4.21. Once again some entries in Table 4.21 appear equal due to the used decimal precision. The system has 27 distinct eigenvalues, where only three are not part of complex conjugate pairs. In addition, six eigenvalues form unstable conjugate pairs.

Table 4.21: Eigenvalues of MTDC system.

Eigenvalue	Real component	Imaginary component
1	-94.9138	-
2, 3	-19.7881	±1.2248
4, 5	-17.9874	±9.4102
6, 7	-2.1156	±11.4338
8, 9	-1.2398	±19.3782
10, 11	-0.8535	±1.5885
12	-0.2196	-
13, 14	-0.1323	±0.4758
15, 16	-0.1323	±0.4758
17, 18	-0.1323	±0.4758
19, 20	-0.1323	±0.4758
21	-0.0007	-
22, 23	8.2045	±336.4494
24, 25	8.2045	±336.4494
26, 27	9.4378	±316.8102

Figure 4.15 shows the eigenvalues plotted in the complex plane, with the unstable eigenvalues marked with red.

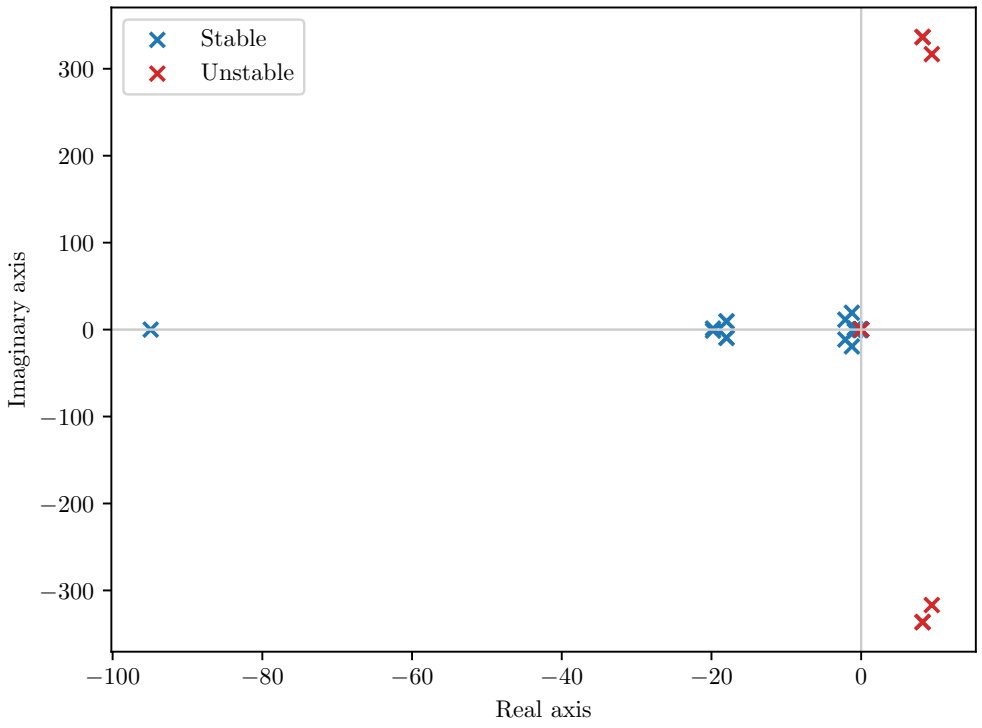


Figure 4.15: Eigenvalue plot for MTDC system.

4.4.1 Results of Base Case Implementation of Multi-Terminal DC System

The performance metrics scores of the base case implementation of the MTDC system are presented in Table 4.22.

Table 4.22: Performance metrics for base case simulation of MTDC system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]
1.40	296	76	1.00×10^{-2}	3.15×10^{-4}	4.73×10^{-3}

Figure 4.16 shows the simulated DC voltage V_{dc} at each of the three VSCs in the system, including the slack converter, along with the step length. From the figure it can be seen how the step length fell from its upper value at the disturbance instance, and was close to reaching the lower limit. From this, it made several sharp, brisk increases, before settling into a semi-periodic, increasing pattern.

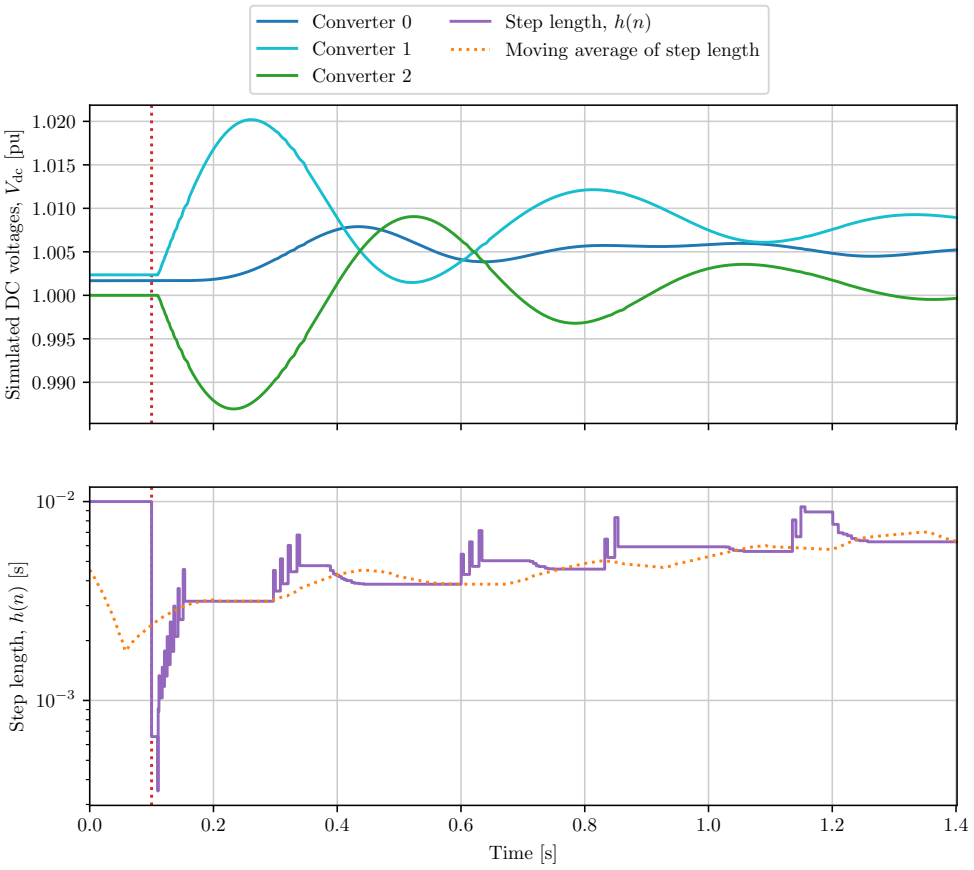


Figure 4.16: Base case simulation of MTDC system for $t \in [0, 1.4]$ s with DC line removal at $t = 0.1$ s.

At first the step length plot of the MTDC system might appear notably different from those of the previous three cases. However, after considering that the time

span of the MTDC case is considerably shorter than for the previous cases, the step length graphs becomes more similar. Seen from a similar scale as the other cases, the step length here too gradually increased via sharp points and flattened valleys. The step length did, however, not reach the ceiling value during the simulated time.

4.4.2 Verification of Multi-Terminal DC System Simulation Results

Figure 4.17 shows the verified MATLAB solution overlain the solution obtained using the base case implementation of Gear's method. The figure reveals large discrepancies between the two solutions, which is supported by the maximum MSE being 9.49×10^3 for the q-component of the terminal voltage, V_{qt_2}, at node 2.

Whereas the solution given by Gear's method was stable, the verified solution shows that the system in fact was unstable. For the non-slack converters, rather than the DC voltages converging to new stable operating points, high-frequent and exponentially increasing oscillations started appearing at around 1 s. The solution obtained by MATLAB is supported by the existence of poles in the right half-plane, as given by Table 4.21 and Figure 4.15.

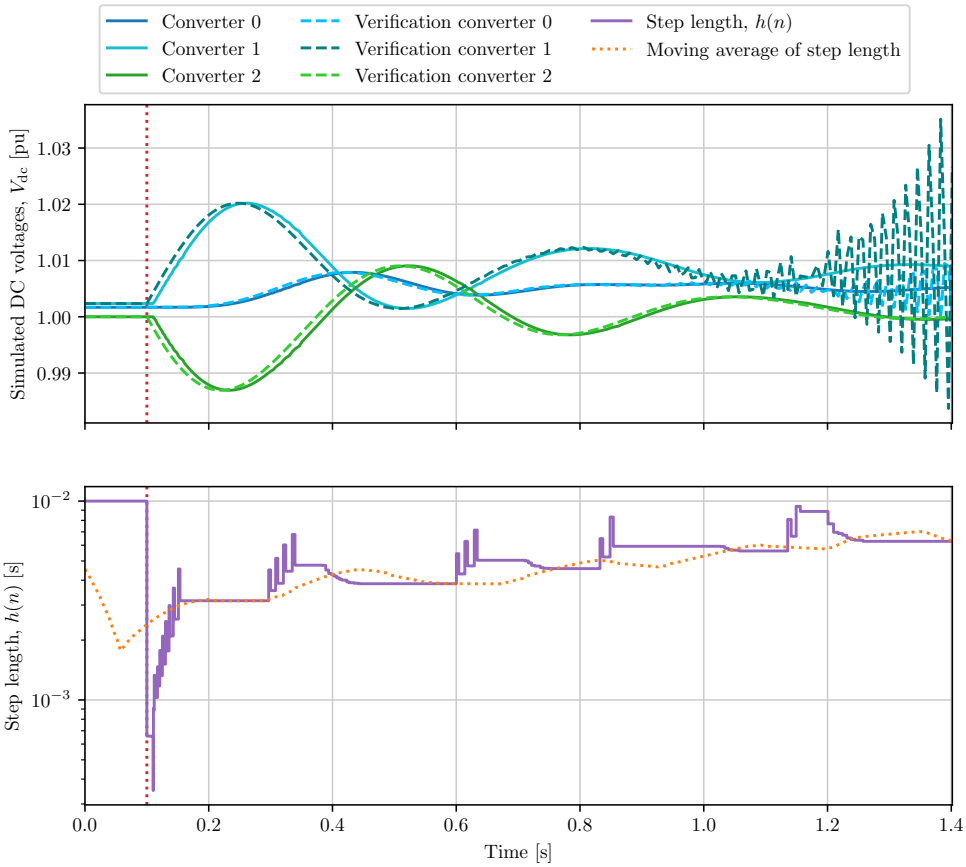


Figure 4.17: Verification of base case simulation of MTDC system for $t \in [0, 1.4]$ s with DC line removal at $t = 0.1$ s.

4.4.3 Results of Implementation of Multi-Terminal DC System with Strategies

Even though the base case implementation was unable to produce a result similar to the verified solution, it was still worth exploring the effects of applying the step length strategies. In all the previous cases, the overall best parameter combination resulted in a smaller maximum MSE than the base case solution, and it was worth exploring if applying the correct strategies would improve the accuracy of the Gear solution.

Table 4.23 gives the averaged scores on the performance metrics from applying the different step length strategies. The complete results are given in Table A.4 in Appendix A.

Table 4.23: Performance metrics for simulation strategies for MTDC case.

(a) Step scaling factor.

Step scaling factor	Average number of step changes	Average mean step length [s]	Average maximum MSE
1.0	46.13	5.02×10^{-3}	1.01×10^{4}
0.9	46.13	4.86×10^{-3}	9.73×10^{3}
0.8	26.63	4.67×10^{-3}	1.01×10^{4}
0.7	42.00	4.45×10^{-3}	9.25×10^{3}
0.6	42.00	4.40×10^{-3}	9.25×10^{3}
0.5	64.63	4.06×10^{-3}	9.43×10^{3}

(b) Step increase limit.

Step increase limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	45.08	4.52×10^{-3}	9.49×10^{3}
False	44.08	4.63×10^{-3}	9.76×10^{3}

(c) Step length fixed.

Step fixed for [steps]	Average number of step changes	Average mean step length [s]	Average maximum MSE
1	72.42	4.53×10^{-3}	9.23×10^{3}
15	16.75	4.61×10^{-3}	1.00×10^{4}

(d) Step decrease limit.

Step decrease limit	Average number of step changes	Average mean step length [s]	Average maximum MSE
True	42.50	4.66×10^{-3}	9.80×10^{3}
False	46.67	4.49×10^{-3}	9.46×10^{3}

From Table 4.23 it is evident that the lowest average number of step changes was obtained for a scaling factor of 0.8. The average mean step length, however, gradually decreased from its maximum value at 1.0 as the scaling factor was reduced. The lowest average MSE was obtained for scaling factors of 0.7 and 0.6.

Furthermore, applying a step increase limit gave a larger average number of step length changes and a lower average mean step length, albeit with a smaller average maximum MSE, compared to not applying such a limit. Applying a step decrease limit, however, gave an inverse result, i.e. resulting in both fewer average number of step changes and a higher average mean step length, but at the cost of a higher average maximum MSE.

Similar results were obtained by fixing the step length for at least fifteen steps. By doing so, a lower average number of step length changes were performed, a higher average mean step length was achieved, albeit with a higher average maximum MSE.

By employing (3.2) and with the ideal points defined as in the SMIB case in Section 4.1.3, the overall best parameter combination was determined. With $d = 1.195 \times 10^{-1}$, the overall best combination was found to be not scaling, limiting the permitted step change frequency and step increases, but not decreases. This combination also resulted in the lowest maximum MSE of all the 48 combinations. Table 4.24 shows the highlighted combination's performance metrics scores, and Figure 4.18 shows the simulation results along with the verified solution.

Table 4.24: Performance metrics for overall best simulation case of MTDC system.

Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Max. MSE
1.40	279	13	8.67×10^{-3}	1.00×10^{-3}	5.02×10^{-3}	8.37×10^{3}

Even with the strategy resulting in the lowest obtained maximum MSE, Gear's method was still unable to capture the unstable oscillations of the DC voltages. Fewer step length adjustments were made, and the step length remained in a narrower band below the upper limit, but the maximum MSE was not reduced in any meaningful way.

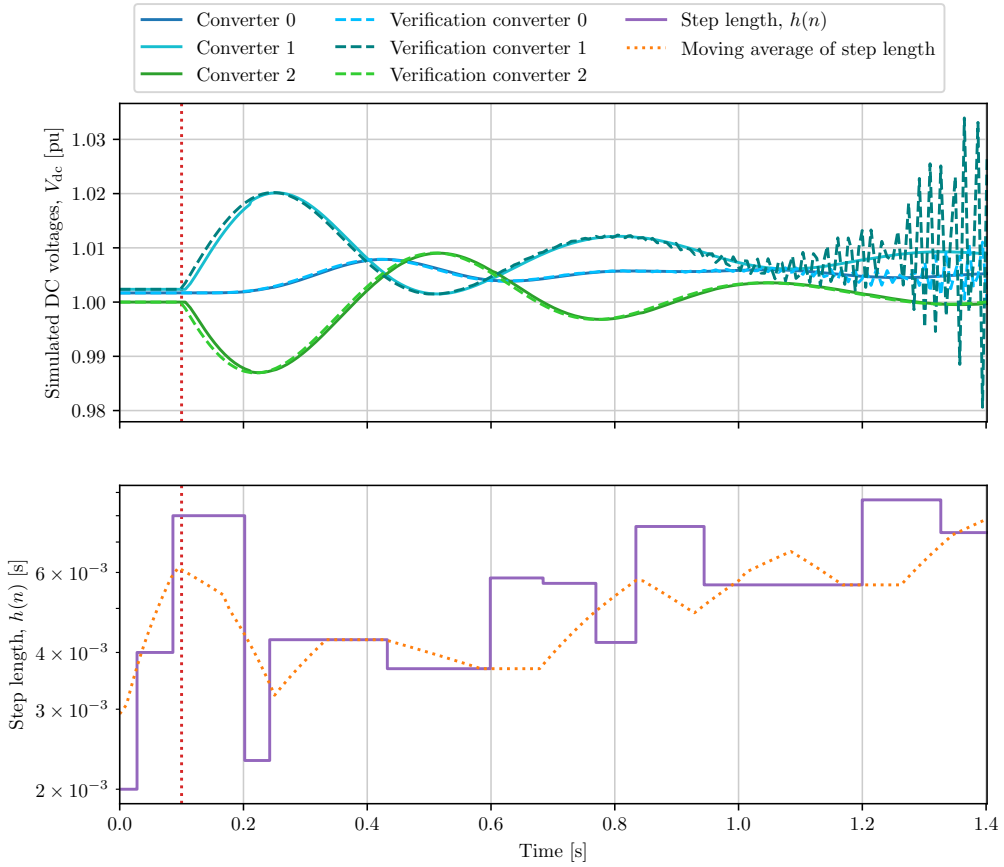


Figure 4.18: Overall best case simulation of MTDC system for $t \in [0, 1.4]$ s with DC line removal at $t = 0.1$ s and verification.

Chapter 5

Discussion

The following chapter contains the discussion of the results obtained in Chapter 4, on which the conclusions drawn in Chapter 6 are based.

Initially, a discussion of the method, performance metrics, and validation method used is carried out. Following this, the results from each of the four simulation cases are discussed and possible explanations for the findings are offered. Based on this, the more general findings are highlighted and compared with previously published results.

5.1 Method, Performance Metrics, and Validation

The metrics used to assess the performances of the simulation methods naturally influence the evaluation and possible conclusions drawn from the results. As a consequence, the performance metrics must be justified and linked to the evaluation and distinctions one wishes to make. By doing so, it can be ensured that the chosen performance metrics do in fact shed light on the performance of the investigated strategies.

Based on [4, 6], the number of step length adjustments and mean step length were chosen as performance metrics. In addition, the mean squared error was introduced as a measure of the accuracy of the obtained results. Even though these particular metrics can be justified as done in Section 3.1.2, it is still worth noting on their effects on the strategy evaluations.

When one of the performance metrics is the number of step length changes, it is to be expected that a strategy of restricting the permitted frequency of step length changes achieves a positive score, as it directly targets the metric. Yet, due to the nature of Gear's method, limiting the number of step length adjustments required, and in doing so reducing the number of failed iterations and correction matrix updates, does impact the performance of the method.

Moreover, the method of distinguishing between the effects of the different strategies deserves a second look. In an attempt to isolate the different strategies' effects on the method, mainly the average effects of each parameter was investigated. While

this formed a basis for determining the effect of each strategy, as was one of the stated aims of the thesis, it masked the possible inter-play between the different strategies. In order to mitigate this, the overall best performing case as determined by minimizing the Euclidean distance between the achieved scores and the ideal ones was also highlighted. In doing so, both the effects of the strategies alone and together could be better understood.

Still, as the ideal points were set based on the results from the investigated combinations, the best case was not measured against an external, objective goal, but rather one set by the cumulative results. By making the criteria self-referencing, it became harder to draw conclusions regarding the strategies on a more general basis, as this tied their performances to the particular case investigated.

Finally, the validation method should be discussed. One of the main aims of this thesis was to determine the suitability of Gear's method as a dynamic power system simulation solver. By directly using the DAE solver in MATLAB, rather than prebuild modules representing different power system components, the modeling done could not be evaluated. Any possible modeling errors, including discrete event handling, from the Python implementation would be directly imported into the MATLAB formulation.

However, by using the same system descriptions, the results from the two numerical solution methods could be directly compared, supporting one of the thesis goals. Such a direct formulation avoided the possible issue of not knowing if the two obtained results were comparable, which is often the case when black box system components are used.

5.2 Evaluation of Single Machine to Infinite Bus Simulation Results

Starting off, its is worth discussing why the discrete event, in this case a reference change, caused the step length to drop sharply. The predictor step of the algorithm makes its prediction based on the current step length and the state derivatives. It therefore has no information about the changes to the system caused by the event. The corrector on the other hand, calculates the deviation between the predicted solution and the accurate one. The reference change therefore resulted in a large deviation, and subsequently, a large truncation error. A large truncation error in turn led to a sharp drop in the step length.

Continuing, the results obtained in Section 4.1 confirmed the results from [6, 16], showing that Gear's method was able simulate higher-order models of stable single machine to infinite bus systems. In addition, strategies could be applied to improve the simulation results, both in terms of computational performance and accuracy. As indicated by Table 4.5, the overall best performing case reduced the number of step length adjustments by 87 %, increased the mean step length by 7 %, and reduced the maximum MSE by 47 %.

When looking at each of the strategies separately for the SMIB case, applying a scaling factor of 0.8 reduced the average number of step length changes, but at the

cost of both reducing the average mean step length and increasing the average maximum MSE. Fixing the step length for at least 15 steps improved the performance across all three metrics when compared to letting the step length change every step. Doing so reduced both the average number of step length changes and the average maximum MSE, as well as slightly raising the average mean step length. Both limiting strategies resulted in poorer computational performance. Imposing a step decrease limit improved the accuracy of the solution, while imposing a step increase limit did the opposite

The overall best simulation results, given the chosen performance metrics, were found by not applying a scaling factor, preventing the step length from changing more than every fifteenth step, and imposing a step decrease limit, but not a step increase limit. This is in contrast to what was found in [16], where both limiting strategies were found to have adverse effects. The solution accuracy was not investigated in [16], offering a possible answer to why different strategies were deemed most successful. In addition, the system descriptions themselves varied somewhat.

Regardless of the strategies applied, the difference between the Gear solution and the verified MATLAB solution could, for most use cases, be said to be negligible for this test system. Disregarding outlier cases, all parameter combinations resulted in a maximum MSE of at most 2.51×10^{-3}. Based on this, it would be possible to argue that a parameter combination more heavily favoring the computational metrics should have been chosen as the overall best performing case, as the maximum MSE still would have been considered acceptable.

Moreover, three outlier cases were observed. All three cases had the same parameter combinations as the base case, except that a scaling factor of either 0.8, 0.6, or 0.5 was applied. It is not obvious why these particular combinations resulted in outlier results. By closely inspecting Figure 4.5, it is evident that just after the reference change, the step length displayed several pulses of only a single step length, fluctuating over a large magnitude range. At each of these pulses, the simulated function made relatively large discrete steps, and it is possible that this is what initially led the result to deviate from the verified solution. As the simulation went on, the errors simply compounded.

5.3 Evaluation of Three Machines to Infinite Bus Simulation Results

The second test system increased the complexity of the simulated system, making it possible to gain further insight into the feasibility of Gear's method as a dynamic simulation tool. When comparing the overall best performing case to the base case implementation, the number of step length changes, mean step length and maximum MSE were improved by 97 %, 29 %, and 99 %, respectively. In addition to increasing the size of the simulated system, the effects of simulating an unstable system, as well as state saturation was explored.

The step length drop associated with the state saturation can be explained similarly as with discrete events. Saturation is enforced after the correction step, capping the affected state variables at their limits. The derivatives, however, are not altered.

The predictor therefore expects the states to continue on the paths pointed out by the derivatives. The deviation, as determined by the corrector, therefore becomes large, leading to a large truncation error and a step length fall.

For the 3MIB case, applying a scaling factor showed little improvement in terms of performance, but applying scaling factors of 0.9, 0.6, or 0.5 all improved the accuracy of the obtained results. Again, fixing the step length for at least 15 steps improved performance across all three criteria. Imposing a step increase limit resulted in poorer results across all three criteria. Imposing a decrease strategy, however, gave favorable results on all but the number of step length changes criteria.

Overall, the best performing case was determined to be the combination of not scaling, fixing the step length, and not imposing any increase or decrease limits.

Moreover, the solution obtained by Gear's method coincided well with the verified solution, both for the base and best case implementations.

The outlier cases should also be examined. A possible explanation lies in how saturation was enforced just after the reference change. This could explain the stepping nature of the solution, as the angular frequency deviations were limited by the false saturation of other states. This possibility is further supported by the steps coinciding with the saturation pulses and rapid step length fluctuations.

The combinations producing the outlier cases were equal to two of the combinations producing outliers in the SMIB case, i.e. base case configurations except with scaling factors of 0.6 and 0.5. Moreover, by inspecting the similar combinations with scaling factors of 0.8 and 0.7, it became clear that these produced maximum MSEs substantially larger than any of the other included combinations. It would have been possible to exclude these combinations as well, even though the associated number of step changes and mean step lengths were not irregular. Nevertheless, the overall best parameter combination and discussion of strategy effects would have remained unchanged.

5.4 Evaluation of System with Static Compensator Simulation Results

The third case presented Gear's method with the challenge of simulating a system with a considerable power electronic component in the form of a static compensator. Compared to the two previous systems, the accuracy of the obtained result was overall worse for the STATCOM case. The deviations could, however, depending on the application requirements, be seen as acceptable as the major features of the simulated results were comparable to the MATLAB solution.

By applying the appropriate strategies, the number of step length changes could be reduced by 91 %, the mean step length could be increased by 28 %, and the maximum MSE could be reduced by 97 %. When comparing each strategy, applying a step scaling factor once again reduced the number of step length changes, but at the cost of a lower mean step length and a higher maximum MSE. Imposing an increase limit gave on average reduced performance, but improved accuracy. Fixing the step length and imposing a decrease limit, on the other hand, both gave improved average

results on all three criteria.

Overall, the best performing case was found to be not scaling, imposing limits on the permitted frequency of step length changes and step decrease, but no increase limit.

For the STATCOM case, a considerable number of parameter combinations resulted in outlier cases. Regardless of scaling factor or if a step increase limit was imposed, the combination of not fixing the step length and imposing a decrease limit produced outliers. In addition, for scaling factors 0.6 and 0.5, not imposing a decrease limit resulted in outliers as well. The combination of scaling by 0.7 and not imposing any other limits, could possibly also have been treated as an outlier as it scored the worst out of the remaining combinations, and substantially so on both the number of step length adjustments and the maximum MSE.

One possible explanation for why these particular parameter combinations failed is offered by the fact that, in contrast to the successful simulations, these attempted to perform multiple calculations between the two discrete events. It is possible that after finding an acceptable solution before the second event, the system change resulted in a deviation too large for the Newton-Raphson iterations to converge. The stability of the Newton-Raphson method and how this affects Gear's method was not explored further, but is worth investigating in future work with the method.

5.5 Evaluation of Multi-Terminal DC System Simulation Results

The final simulation case deviated drastically from the previous, in that rather than being a system dominated by the dynamics of synchronous generators, it was a purely power electronic system. The MTDC system also stood out as the system where Gear's method produced the poorest results.

Gear's method was able to produce stable simulation results, and the method was improved both with regards to performance and accuracy by applying strategies. Still, even the parameter combination resulting in the lowest maximum MSE obtained unusable results. The method was unable to capture the fastest transients of the unstable system, instead presenting a stable solution. Based on this, any further discussion of the improvement strategies would be of little use.

It was not immediately clear why Gear's method was unable to properly simulate the MTDC system when it was able to simulate systems with larger ratios between the largest and smallest time constants. Changing the upper and lower step length limits, or forcing an artificially short step length showed no effect. Tightening the acceptable truncation error and permitted Newton-Raphson error limit, as suggested by [6], were also unsuccessful. Neither the first nor second derivatives showed any odd behavior either.

A final possible explanation for the poor tracking might be that the correction matrix $\boldsymbol{A}_{(l)}$ was close to being singular around when the unstable oscillations should have appeared. With an ill-conditioned system matrix, a small deviation in the input $\boldsymbol{\Delta}^{(l)}$ in (2.46) would lead to a large error in the state deviation $\Delta \boldsymbol{y}^{(l)}$ [11].

This is further supported by the fact that the method only returned 27 distinct eigenvalues when the system was originally defined by 31 differential equations. In contrast to the three other systems, Gear's method was unable to properly capture the complete dynamic range of the system.

Beyond what has already been stated, it was not explored further why Gear's method was unable to properly simulate the MTDC system. If similar results can be produced for other VSC-heavy systems, or if it is the result of this formulation in particular, is very much worth exploring further.

5.6 Evaluation of Gear's Method and Proposed Strategies

Based on the preceding discussion and results obtained, a general discussion of Gear's method and the proposed strategies can be carried out.

The results from all four cases indicated that fixing the step length was the single strategy with the largest impact on the method, both in terms of computational performance and accuracy. By restricting the permitted step length adjustment frequency, the negative effects of excessive drops or increases in step length were apparently mitigated. This indicated that simply fixing the step length for a given number of steps not only reduced the number of recalculations of the correction matrix required, it also handled the issues the limiting strategies were targeted at. Moreover, only in the MTDC case did fixing the step length not also lead to a meaningfully improved solution accuracy.

In all four cases, the overall best performing parameter combinations included not scaling the step length change. All four cases showed different effects on performance and accuracy from scaling, making it difficult to see any clear trends. However, the majority of the outlier cases were produced for smaller scaling factors, indicating that if scaling is employed, it should be kept around 1.0.

Of the four best performing combinations, three did not impose an increase limit, while two did not impose a decrease limit. This suggests that it is more important to prevent rapid step length decreases than rapid increases. Still, the advantage of applying any such limits has not been conclusively shown by the four cases. Reference [4] found similar results for their single-regulator system, albeit more strongly in favor of limiting step length decreases.

For all investigated cases, a positive correlation between improved method performance and solution accuracy was observed. Every best performing combination resulted in a lower maximum MSE compared to the base case.

However, the two systems with considerable power electronics, and in particular the MTDC system, showed poorer tracking accuracy than the systems dominated by synchronous generators. The solver does not inherently distinguish between synchronous generators and VSCs, it only has information about states and state equations. However, the results obtained displayed Gear's method giving poorer solution accuracy with an increased power electronics share. If this is due to some underlying modeling of power converters or simply the formulation used here, cannot

be clearly answered, and requires further investigation.

On a more general note, the conclusions possible to draw from the method employed here will to a large extent be system dependent. The observations made illustrate possible strengths and weaknesses, but on different systems, modifications may very well be required. Nevertheless, the work done here furthers the insights into a general solution strategy which can be adapted to suit specific use cases.

Chapter 6

Conclusions and Outlook

This final chapter contains the conclusions drawn from the master's thesis in relation to the objectives set in Section 1.2, and highlights and suggests possible avenues for future work.

6.1 Conclusions

Through four different power system cases this thesis has shown how a second-order version of Gear's method can be used for dynamic power system simulations. The cases explored cover a wide range of possible power system applications, displaying varied characteristics in terms of stability, size and complexity, saturation effects, topology changes and discrete system events, and degree of power electronic integration.

In addition to giving a thorough description of the method, different strategies for improving the computational performance and solution accuracy of the method were implemented and evaluated. Previous implementations' findings were confirmed, in that the method was able to adjust the step length to match the dynamics of the simulated system. In addition, it was shown how discrete events, such as reference changes, faults, and state saturation, caused the step length to drop sharply, hurting the computational performance of the method.

The key contribution of this thesis was to extend the list of systems and system types implemented for Gear's method, including increasing the size and introducing more power electronic components as compared to what already exists in published literature. Also, system initialization was improved, basing it on both load flow solutions and iterations, a technique not previously used for Gear's method.

In addition, the proposed improvement strategies were evaluated both separately and in combination, making it possible to determine each strategy's effect and, to an extent, in combination. The results obtained, with and without strategies applied, were furthermore compared to a commercially available differential-algebraic equation system solver, making it possible to not only explore the performance of the method, but the accuracy as well.

In general, the variable step length nature of the method lent itself well to the

simulation of power system models consisting of stiff differential equation systems and accompanying algebraic constraints, albeit with some stark limitations. For the three systems dominated by synchronous generators, Gear's method was able to produce results comparable to the verified solution, while it for the fully DC system was not. If this was due to some underlying modeling nature of power converters, or simply a feature of this particular formulation, could not be concluded.

For the successful simulation cases, fixing the step length for at least 15 steps was shown to give the largest improvement in terms of both performance and accuracy, of the strategies tested. By fixing the step length, the issue of excessive step length decreases was avoided, while also bettering the solution accuracy. For each case, not scaling step length changes, or at least keeping the scaling factor close to 1.0, produced the overall best results, as determined by the metrics used. Limiting the allowable step length changes by preventing a more than twofold change between consecutive steps could not conclusively be shown to have positive effects on the method.

On a final note, even though the conclusions drawn from the work done in this thesis can largely be seen as system dependent, they help further the understanding of a possible general solution strategy using Gear's method, which has to be adapted for specific cases.

6.2 Outlook and Further Work

Based on the work done in this master's thesis, the following aspects are worth exploring further: Implementing Gear's method for other VSC-dominated systems. The share of power converters in the modern power system is increasing, and Gear's method's ability to simulate such systems is of central importance when judging its viability.

Likewise, increasing the size of the simulated system would be interesting to see how the method handles systems of even more generation units and loads than what has been explored in this thesis.

Furthermore, continuing the development of strategies for adapting the step length would be valuable. This could include more sophisticated techniques, such as automatic parameter tuning and strategy selection. In addition, methods for primarily increasing the accuracy of the method, not just the performance, would also be of interest.

Finally, further exploring the numerical stability of the method, including the use of Newton-Raphson's iterative method, would be useful. Other root-finding methods could be explored and their effects on Gear's method determined.

Bibliography

[1] Byrne, G. D. and Hindmarsh, A. C. (1975) A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations, *ACM Trans. Math. Softw.*, 1(1), pp. 71–96. doi: 10.1145/355626.355636

[2] Cole, S. and Beerten, J. and Belmans, R. (2010) Generalized Dynamic VSC MTDC Model for Power System Stability Studies, *IEEE Trans. Power Syst.*, 25(3), pp. 1655–1662. doi: 10.1109/TPWRS.2010.2040846

[3] Debs, A. S. (1988) *Modern Power Systems Control and Operation*. 1st edn. Boston, MA: Kluwer Academic Publishers.

[4] Eidsvik, H. (2019) *Dynamic Simulation of Power Systems Based on a Second Order Predictor-Corrector Scheme*. Master's thesis. Norwegian University of Science and Technology, Trondheim. Available at: http://hdl.handle.net/11250/2625904 (accessed: August 26, 2019).

[5] Eiselt, H. A. and Sandblom, C.-L. (2007) *Linear Programming and its Applications*. 1st edn. Berlin: Springer-Verlag.

[6] Fosso, O. B. (2016) *Implementation of an Implicit Method for Numerical Integration of Differential Equations with Self-Adaptive Time-Steps (Gear's Method)*. Available at: https://www.doi.org/10.13140/RG.2.2.33826.43205 (accessed: August 27, 2019).

[7] Gear, C. W. (1971) The Automatic Integration of Ordinary Differential Equations, *Commun. ACM*, 14(3), pp. 176–179. doi: 10.1145/362566.362571

[8] Hunter, J. and Dale, D. and Firing, E. and Droettboom, M. and the Matplotlib Development Team (2020) *Matplotlib Documentation, v3.1.3*. Available at: https://matplotlib.org/3.1.3/contents.html (accessed: May 20, 2020).

[9] IEEE (2006) *421.5-2005 - IEEE Recommended Practice for Excitation System Models for Power System Stability Studies*. Available at: https://www.doi.org/10.1109/IEEESTD.2006.99499 (accessed: May 20, 2020).

[10] Kundur, P. (1994) *Power System Stability and Control*. 1st edn. New York, NY: McGraw-Hill.

[11] Lay, D. C. and Lay, S. R. and McDonald, J. (2016) *Linear Algebra and Its Applications*. 5th edn. global. Harlow: Pearson Education Ltd.

[12] Lima, L. (2014) *IEEE PES Task Force on Benchmark Systems for Stability Controls - Report on the 3-Generator System*. Available at: `http://www.sel.eesc.usp.br/ieee/index.htm` (accessed: January 15, 2020).

[13] Machowski, J. and Bialek, J. W. and Bumby, J. R. (2012) *Power System Dynamics: Stability and Control*. 2nd edn. with corr. Chichester: Wiley.

[14] MathWorks (2017) *MATLAB, v.9.3.0.713579 (R2017b)*. Available at: `https://www.mathworks.com/help/releases/R2017b/matlab/index.html` (accessed: May 20, 2020).

[15] MathWorks (2020) *Solve Differential Algebraic Equations (DAEs)*. Available at: `https://mathworks.com/help/symbolic/solve-differential-algebraic-equations.html` (accessed: May 20, 2020).

[16] Narayan, E.-A. S. (2019) *Dynamic Power System Simulation Using a Variable Time Step Method*. Specialization project. Norwegian University of Science and Technology, Trondheim. Available at: `https://www.doi.org/10.13140/RG.2.2.23865.80482/2` (accessed: January 15, 2020).

[17] Pourbeik, P. et al. (2013) *Technical Report PES-TR1 - Dynamic Models for Turbine-Governors in Power System Studies*. Available at: `https://site.ieee.org/fw-pes/files/2013/01/PES_TR1.pdf` (accessed: November 25, 2019).

[18] Python Software Foundation (2020) *The Python Language Reference, v3.7*. Available at: `https://docs.python.org/3.7/` (accessed: May 20, 2020).

[19] Sammut, C. and Webb, G. I. (ed.) (2017) *Encyclopedia of Machine Learning and Data Mining: Mean Squared Error*. 2nd edn. Boston, MA: Springer US.

[20] The SciPy Community (2020) *NumPy Manual, v1.18*. Available at: `https://numpy.org/doc/1.18/` (accessed: May 20, 2020).

[21] The SciPy Community (2020) *SciPy Reference Guide, v1.4.1*. Available at: `https://docs.scipy.org/doc/scipy-1.4.1/reference/` (accessed: May 20, 2020).

[22] Stott, B. (1979) Power System Dynamic Response Calculations, *Proc. IEEE*, 67(2), pp. 219–241. doi: `10.1109/PROC.1979.11233`

[23] Stubbe, M. and Bihain, A. and Deuse, J. and Baader, J. C. (1989) STAG - A New Unified Software Program for the Study of the Dynamic Behaviour of Electrical Power Systems, *IEEE Transactions on Power Systems*, 4(1), pp. 129–138. doi: `10.1109/59.32470`

[24] SymPy Development Team (2019) *SymPy Documentation, v1.5.1*. Available at: `https://docs.sympy.org/latest/index.html` (accessed: May 20, 2020).

[25] Yazdani, A. and Iravani, R. (2010) *Voltage-Sourced Converters in Power Systems: Modeling, Control, and Applications*. 1st edn. Hoboken, NJ: Wiley-IEEE Press.

Appendix A

Complete Simulation Results

This appendix contains the complete simulation results for all four simulated power system cases.

Table A.1: Complete results from simulations of SMIB system.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
1.0	True	1	True	2.0	15.01	2360	596	1.00×10^{-2}	9.35×10^{-4}	6.36×10^{-3}	3.48×10^{-4}	V_{r_0}	7.48
1.0	True	1	False	2.0	15.00	2369	599	1.00×10^{-2}	2.66×10^{-4}	6.33×10^{-3}	1.07×10^{-3}	V_{r_0}	1.13×10^{1}
1.0	True	15	True	2.0	15.00	2339	78	1.00×10^{-2}	1.00×10^{-3}	6.41×10^{-3}	3.65×10^{-4}	V_{r_0}	2.29
1.0	True	15	False	2.0	15.01	2353	88	1.00×10^{-2}	6.97×10^{-4}	6.38×10^{-3}	9.24×10^{-4}	V_{r_0}	7.33
1.0	False	1	True	2.0	15.01	2347	599	1.00×10^{-2}	9.35×10^{-4}	6.39×10^{-3}	1.24×10^{-4}	V_{w_0}	7.21
1.0	False	1	False	2.0	15.01	2367	599	1.00×10^{-2}	2.66×10^{-4}	6.34×10^{-3}	2.43×10^{-4}	V_{r_0}	7.30
1.0	False	15	True	2.0	15.01	2204	77	1.00×10^{-2}	1.00×10^{-3}	6.81×10^{-3}	1.29×10^{-4}	V_{r_0}	1.72×10^{-1}
1.0	False	15	False	2.0	15.01	2190	79	1.00×10^{-2}	1.00×10^{-3}	6.85×10^{-3}	2.17×10^{-4}	V_{r_0}	9.59×10^{-1}
0.9	True	1	True	2.0	15.00	2439	399	1.00×10^{-2}	8.41×10^{-4}	6.15×10^{-3}	3.18×10^{-4}	V_{r_0}	4.84
0.9	True	1	False	2.0	15.01	2451	407	1.00×10^{-2}	2.38×10^{-4}	6.12×10^{-3}	1.04×10^{-3}	V_{r_0}	9.54
0.9	True	15	True	2.0	15.01	2304	88	1.00×10^{-2}	1.00×10^{-3}	6.51×10^{-3}	3.97×10^{-4}	V_{r_0}	2.59
0.9	True	15	False	2.0	15.01	2440	91	1.00×10^{-2}	6.27×10^{-4}	6.15×10^{-3}	9.88×10^{-4}	V_{r_0}	7.91
0.9	False	1	True	2.0	15.01	2432	390	1.00×10^{-2}	8.41×10^{-4}	6.17×10^{-3}	1.20×10^{-4}	V_{w_0}	4.34
0.9	False	1	False	2.0	15.00	2424	397	1.00×10^{-2}	2.38×10^{-4}	6.19×10^{-3}	6.97×10^{-4}	V_{r_0}	6.90
0.9	False	15	True	2.0	15.00	2243	73	1.00×10^{-2}	1.00×10^{-3}	6.69×10^{-3}	1.57×10^{-4}	V_{r_0}	4.16×10^{-1}
0.9	False	15	False	2.0	15.00	2169	88	1.00×10^{-2}	1.00×10^{-3}	6.92×10^{-3}	3.76×10^{-4}	V_{r_0}	2.40
0.8	True	1	True	2.0	15.01	2499	313	1.00×10^{-2}	7.48×10^{-4}	6.00×10^{-3}	3.08×10^{-4}	V_{r_0}	3.74
0.8	True	1	False	2.0	15.01	2521	334	1.00×10^{-2}	2.10×10^{-4}	5.95×10^{-3}	1.03×10^{-3}	V_{r_0}	9.02
0.8	True	15	True	2.0	15.00	2441	97	1.00×10^{-2}	1.00×10^{-3}	6.15×10^{-3}	4.58×10^{-4}	V_{r_0}	3.15
0.8	True	15	False	2.0	15.01	2546	101	1.00×10^{-2}	5.58×10^{-4}	5.89×10^{-3}	1.07×10^{-3}	V_{r_0}	8.65
0.8	False	1	True	2.0	15.00	2497	316	1.00×10^{-2}	7.48×10^{-4}	6.01×10^{-3}	1.16×10^{-4}	V_{w_0}	3.33
0.8	False	1	False	2.0	-	-	-	-	-	-	-	-	-
0.8	False	15	True	2.0	15.01	2350	89	1.00×10^{-2}	1.00×10^{-3}	6.39×10^{-3}	1.11×10^{-4}	V_{w_0}	2.32×10^{-1}
0.8	False	15	False	2.0	15.00	2244	91	1.00×10^{-2}	9.19×10^{-4}	6.68×10^{-3}	3.13×10^{-4}	V_{r_0}	1.84
0.7	True	1	True	2.0	15.00	2552	559	1.00×10^{-2}	6.54×10^{-4}	5.88×10^{-3}	2.94×10^{-4}	V_{r_0}	6.86
0.7	True	1	False	2.0	15.00	2566	584	1.00×10^{-2}	1.83×10^{-4}	5.85×10^{-3}	1.02×10^{-3}	V_{r_0}	1.08×10^{1}
0.7	True	15	True	2.0	15.00	2603	106	1.00×10^{-2}	1.00×10^{-3}	5.76×10^{-3}	2.94×10^{-4}	V_{r_0}	1.72
0.7	True	15	False	2.0	15.00	2700	116	1.00×10^{-2}	4.88×10^{-4}	5.56×10^{-3}	1.19×10^{-3}	V_{r_0}	9.74
0.7	False	1	True	2.0	15.01	2528	532	1.00×10^{-2}	6.54×10^{-4}	5.94×10^{-3}	1.15×10^{-4}	V_{w_0}	6.29
0.7	False	1	False	2.0	15.00	2486	560	1.00×10^{-2}	1.00×10^{-5}	6.03×10^{-3}	2.51×10^{-3}	V_{r_0}	2.26×10^{1}
0.7	False	15	True	2.0	15.01	2478	107	1.00×10^{-2}	1.00×10^{-3}	6.05×10^{-3}	1.31×10^{-4}	V_{r_0}	5.15×10^{-1}
0.7	False	15	False	2.0	15.01	2187	91	1.00×10^{-2}	8.04×10^{-4}	6.86×10^{-3}	5.52×10^{-4}	V_{r_0}	3.98

Continued on next page.

Table A.1: Complete results from simulations of SMIB system *(continued)*.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
0.6	True	1	True	2.0	15.01	2576	785	1.00×10^{-2}	5.69×10^{-4}	5.83×10^{-3}	2.87×10^{-4}	V_{r_0}	9.88
0.6	True	1	False	2.0	15.01	2608	897	1.00×10^{-2}	1.55×10^{-4}	5.75×10^{-3}	9.73×10^{-4}	V_{r_0}	1.37×10^{1}
0.6	True	15	True	2.0	15.01	2674	120	1.00×10^{-2}	1.00×10^{-3}	5.61×10^{-3}	4.38×10^{-4}	V_{r_0}	3.02
0.6	True	15	False	2.0	15.00	2918	140	1.00×10^{-2}	4.18×10^{-4}	5.14×10^{-3}	1.36×10^{-3}	V_{r_0}	1.13×10^{1}
0.6	False	1	True	2.0	15.00	2481	716	1.00×10^{-2}	5.69×10^{-4}	6.05×10^{-3}	1.17×10^{-4}	V_{w_0}	8.81
0.6	False	1	False	2.0	-	-	-	-	-	-	-	-	-
0.6	False	15	True	2.0	15.01	2427	92	1.00×10^{-2}	1.00×10^{-3}	6.18×10^{-3}	1.92×10^{-4}	V_{r_0}	7.82×10^{-1}
0.6	False	15	False	2.0	15.01	2222	99	1.00×10^{-2}	6.89×10^{-4}	6.75×10^{-3}	6.11×10^{-4}	V_{r_0}	4.52
0.5	True	1	True	2.0	15.00	2748	1596	1.00×10^{-2}	4.52×10^{-4}	5.46×10^{-3}	2.23×10^{-4}	V_{r_0}	2.09×10^{1}
0.5	True	1	False	2.0	15.00	2805	1735	1.00×10^{-2}	1.98×10^{-4}	5.35×10^{-3}	8.80×10^{-4}	V_{r_0}	2.38×10^{1}
0.5	True	15	True	2.0	15.00	3034	166	1.00×10^{-2}	1.00×10^{-3}	4.94×10^{-3}	6.67×10^{-4}	V_{r_0}	5.18
0.5	True	15	False	2.0	15.00	3246	175	1.00×10^{-2}	3.49×10^{-4}	4.62×10^{-3}	1.57×10^{-3}	V_{r_0}	1.32×10^{1}
0.5	False	1	True	2.0	15.01	2276	1398	1.00×10^{-2}	4.52×10^{-4}	6.59×10^{-3}	1.30×10^{-4}	V_{w_0}	1.82×10^{1}
0.5	False	1	False	2.0	-	-	-	-	-	-	-	-	-
0.5	False	15	True	2.0	15.01	2567	121	1.00×10^{-2}	1.00×10^{-3}	5.85×10^{-3}	1.85×10^{-4}	V_{r_0}	9.49×10^{-1}
0.5	False	15	False	2.0	15.00	2332	121	1.00×10^{-2}	5.74×10^{-4}	6.43×10^{-3}	6.61×10^{-4}	V_{r_0}	5.00

Table A.2: Complete results from simulations of 3MIB system.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
1.0	True	1	True	2.0	25.00	3752	1056	1.00×10^{-2}	4.34×10^{-4}	6.66×10^{-3}	7.82×10^{-3}	V_{r_0}	4.33×10^{1}
1.0	True	1	False	2.0	25.01	3755	1037	1.00×10^{-2}	2.61×10^{-4}	6.66×10^{-3}	7.95×10^{-3}	V_{r_0}	4.25×10^{1}
1.0	True	15	True	2.0	25.00	2725	29	1.00×10^{-2}	1.00×10^{-3}	9.17×10^{-3}	1.27×10^{-3}	V_{r_0}	2.10×10^{-1}
1.0	True	15	False	2.0	25.00	2767	36	1.00×10^{-2}	6.80×10^{-4}	9.03×10^{-3}	1.43×10^{-3}	V_{r_0}	5.17×10^{-1}
1.0	False	1	True	2.0	25.00	3077	558	1.00×10^{-2}	4.32×10^{-4}	8.12×10^{-3}	4.20×10^{-3}	V_{r_0}	2.24×10^{1}
1.0	False	1	False	2.0	25.00	3433	804	1.00×10^{-2}	1.00×10^{-5}	7.28×10^{-3}	2.44×10^{-1}	V_{r_0}	1.94×10^{2}
1.0	False	15	True	2.0	25.01	2685	29	1.00×10^{-2}	1.00×10^{-3}	9.31×10^{-3}	1.30×10^{-3}	V_{r_0}	2.10×10^{-1}
1.0	False	15	False	2.0	25.00	2661	26	1.00×10^{-2}	1.00×10^{-3}	9.39×10^{-3}	1.38×10^{-3}	V_{r_0}	1.20×10^{-1}
0.9	True	1	True	2.0	25.00	3905	1140	1.00×10^{-2}	3.75×10^{-4}	6.40×10^{-3}	8.40×10^{-3}	V_{r_0}	4.68×10^{1}
0.9	True	1	False	2.0	25.01	3919	1155	1.00×10^{-2}	2.33×10^{-4}	6.38×10^{-3}	8.56×10^{-3}	V_{r_0}	4.75×10^{1}
0.9	True	15	True	2.0	25.01	2753	28	1.00×10^{-2}	1.00×10^{-3}	9.08×10^{-3}	1.38×10^{-3}	V_{r_0}	1.91×10^{-1}
0.9	True	15	False	2.0	25.01	2789	30	1.00×10^{-2}	6.12×10^{-4}	8.96×10^{-3}	1.59×10^{-3}	V_{r_0}	3.58×10^{-1}
0.9	False	1	True	2.0	25.00	3170	605	1.00×10^{-2}	3.90×10^{-4}	7.89×10^{-3}	4.62×10^{-3}	V_{r_0}	2.44×10^{1}
0.9	False	1	False	2.0	25.01	3943	1046	1.00×10^{-2}	1.00×10^{-5}	6.34×10^{-3}	8.53×10^{-3}	V_{r_0}	4.30×10^{1}
0.9	False	15	True	2.0	25.01	2731	24	1.00×10^{-2}	1.00×10^{-3}	9.15×10^{-3}	1.44×10^{-3}	V_{r_0}	1.36×10^{-1}
0.9	False	15	False	2.0	25.00	2685	26	1.00×10^{-2}	1.00×10^{-3}	9.31×10^{-3}	1.46×10^{-3}	V_{r_0}	1.71×10^{-1}
0.8	True	1	True	2.0	25.01	4094	1191	1.00×10^{-2}	3.28×10^{-4}	6.11×10^{-3}	9.03×10^{-3}	V_{r_0}	4.90×10^{1}
0.8	True	1	False	2.0	25.00	4027	1137	1.00×10^{-2}	2.06×10^{-4}	6.21×10^{-3}	8.78×10^{-3}	V_{r_0}	4.68×10^{1}
0.8	True	15	True	2.0	25.00	2777	31	1.00×10^{-2}	1.00×10^{-3}	9.00×10^{-3}	1.40×10^{-3}	V_{r_0}	3.12×10^{-1}
0.8	True	15	False	2.0	25.01	2832	34	1.00×10^{-2}	5.44×10^{-4}	8.83×10^{-3}	1.66×10^{-3}	V_{r_0}	5.21×10^{-1}
0.8	False	1	True	2.0	25.01	3110	454	1.00×10^{-2}	3.67×10^{-4}	8.04×10^{-3}	4.15×10^{-3}	V_{r_0}	1.81×10^{1}
0.8	False	1	False	2.0	25.00	5222	997	1.00×10^{-2}	1.00×10^{-5}	4.79×10^{-3}	3.86×10^{3}	V_{t_2}	3.04×10^{6}
0.8	False	15	True	2.0	25.00	2737	28	1.00×10^{-2}	1.00×10^{-3}	9.13×10^{-3}	1.43×10^{-3}	V_{r_0}	2.11×10^{-1}
0.8	False	15	False	2.0	25.00	2706	27	1.00×10^{-2}	1.00×10^{-3}	9.24×10^{-3}	1.57×10^{-3}	V_{r_0}	2.68×10^{-1}
0.7	True	1	True	2.0	25.00	3805	1004	1.00×10^{-2}	2.90×10^{-4}	6.57×10^{-3}	7.79×10^{-3}	V_{r_0}	4.12×10^{1}
0.7	True	1	False	2.0	25.01	3928	1116	1.00×10^{-2}	1.79×10^{-4}	6.37×10^{-3}	8.50×10^{-3}	V_{r_0}	4.59×10^{1}
0.7	True	15	True	2.0	25.01	2818	36	1.00×10^{-2}	1.00×10^{-3}	8.87×10^{-3}	1.49×10^{-3}	V_{r_0}	5.32×10^{-1}
0.7	True	15	False	2.0	25.00	2895	44	1.00×10^{-2}	4.76×10^{-4}	8.63×10^{-3}	1.84×10^{-3}	V_{r_0}	9.50×10^{-1}
0.7	False	1	True	2.0	25.00	3102	531	1.00×10^{-2}	3.38×10^{-4}	8.06×10^{-3}	3.94×10^{-3}	V_{r_0}	2.12×10^{1}
0.7	False	1	False	2.0	25.01	4029	910	1.00×10^{-2}	1.00×10^{-5}	6.21×10^{-3}	6.67×10^{3}	V_{t_2}	5.25×10^{6}
0.7	False	15	True	2.0	25.01	2791	35	1.00×10^{-2}	1.00×10^{-3}	8.96×10^{-3}	1.61×10^{-3}	V_{r_0}	5.33×10^{-1}
0.7	False	15	False	2.0	25.01	2755	31	1.00×10^{-2}	9.03×10^{-4}	9.07×10^{-3}	1.68×10^{-3}	V_{r_0}	4.36×10^{-1}

Continued on next page.

Table A.2: Complete results from simulations of 3MIB system *(continued)*.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
0.6	True	1	True	2.0	25.01	3925	1152	1.00×10^{-2}	2.45×10^{-4}	6.37×10^{-3}	8.23×10^{-3}	V_{r_0}	4.73×10^{1}
0.6	True	1	False	2.0	25.01	4099	1329	1.00×10^{-2}	1.37×10^{-4}	6.10×10^{-3}	8.94×10^{-3}	V_{r_0}	5.47×10^{1}
0.6	True	15	True	2.0	25.01	2853	39	1.00×10^{-2}	1.00×10^{-3}	8.76×10^{-3}	1.61×10^{-3}	V_{r_0}	6.83×10^{-1}
0.6	True	15	False	2.0	25.01	2952	51	1.00×10^{-2}	4.08×10^{-4}	8.47×10^{-3}	1.94×10^{-3}	V_{r_0}	1.25
0.6	False	1	True	2.0	25.00	3063	539	1.00×10^{-2}	2.94×10^{-4}	8.16×10^{-3}	3.73×10^{-3}	V_{r_0}	2.15×10^{1}
0.6	False	1	False	2.0	-	-	-	-	-	-	-	-	-
0.6	False	15	True	2.0	25.00	2803	37	1.00×10^{-2}	1.00×10^{-3}	8.92×10^{-3}	1.49×10^{-3}	V_{r_0}	5.71×10^{-1}
0.6	False	15	False	2.0	25.00	2858	41	1.00×10^{-2}	7.74×10^{-4}	8.75×10^{-3}	1.61×10^{-3}	V_{r_0}	7.60×10^{-1}
0.5	True	1	True	2.0	25.01	3940	1367	1.00×10^{-2}	3.30×10^{-4}	6.35×10^{-3}	7.87×10^{-3}	V_{r_0}	5.62×10^{1}
0.5	True	1	False	2.0	25.01	4334	1662	1.00×10^{-2}	1.24×10^{-4}	5.77×10^{-3}	9.66×10^{-3}	V_{r_0}	6.86×10^{1}
0.5	True	15	True	2.0	25.00	2953	59	1.00×10^{-2}	1.00×10^{-3}	8.46×10^{-3}	1.81×10^{-3}	V_{r_0}	1.52
0.5	True	15	False	2.0	25.01	3017	66	1.00×10^{-2}	3.40×10^{-4}	8.29×10^{-3}	2.18×10^{-3}	V_{r_0}	1.89
0.5	False	1	True	2.0	25.01	3023	791	1.00×10^{-2}	2.51×10^{-4}	8.27×10^{-3}	3.67×10^{-3}	V_{r_0}	3.20×10^{1}
0.5	False	1	False	2.0	-	-	-	-	-	-	-	-	-
0.5	False	15	True	2.0	25.00	2939	59	1.00×10^{-2}	1.00×10^{-3}	8.50×10^{-3}	1.63×10^{-3}	V_{r_0}	1.49
0.5	False	15	False	2.0	25.00	2792	48	1.00×10^{-2}	6.45×10^{-4}	8.95×10^{-3}	1.91×10^{-3}	V_{r_0}	1.12

Table A.3: Complete results from simulations of STATCOM system.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
1.0	True	1	True	2.0	-	-	-	-	-	-	-	-	-
1.0	True	1	False	2.0	10.01	1288	187	1.00×10^{-2}	1.00×10^{-5}	7.77×10^{-3}	4.99×10^{-2}	V_{r_0}	1.34×10^{1}
1.0	True	15	True	2.0	10.00	1205	22	1.00×10^{-2}	1.00×10^{-3}	8.30×10^{-3}	2.77×10^{-1}	V_{r_0}	4.63
1.0	True	15	False	2.0	10.00	1380	30	1.00×10^{-2}	4.18×10^{-5}	7.25×10^{-3}	9.91×10^{-1}	V_{r_0}	1.90×10^{1}
1.0	False	1	True	2.0	-	-	-	-	-	-	-	-	-
1.0	False	1	False	2.0	10.00	1433	169	1.00×10^{-2}	1.00×10^{-5}	6.98×10^{-3}	1.99	V_{t_1}	4.08×10^{1}
1.0	False	15	True	2.0	10.00	1121	16	1.00×10^{-2}	1.00×10^{-3}	8.92×10^{-3}	6.41×10^{-2}	V_{r_0}	3.70×10^{-1}
1.0	False	15	False	2.0	10.00	1152	14	1.00×10^{-2}	9.65×10^{-4}	8.68×10^{-3}	9.10×10^{-2}	V_{r_0}	8.35×10^{-1}
0.9	True	1	True	2.0	-	-	-	-	-	-	-	-	-
0.9	True	1	False	2.0	10.00	1307	162	1.00×10^{-2}	1.00×10^{-5}	7.65×10^{-3}	5.10×10^{-2}	V_{r_0}	1.15×10^{1}
0.9	True	15	True	2.0	10.01	1240	21	1.00×10^{-2}	1.00×10^{-3}	8.07×10^{-3}	2.80×10^{-1}	V_{r_0}	4.68
0.9	True	15	False	2.0	10.01	1428	30	1.00×10^{-2}	3.81×10^{-5}	7.01×10^{-3}	9.72×10^{-1}	V_{r_0}	1.86×10^{1}
0.9	False	1	True	2.0	-	-	-	-	-	-	-	-	-
0.9	False	1	False	2.0	10.01	1440	238	1.00×10^{-2}	1.00×10^{-5}	6.95×10^{-3}	1.98	V_{t_1}	4.25×10^{1}
0.9	False	15	True	2.0	10.01	1120	17	1.00×10^{-2}	1.00×10^{-3}	8.93×10^{-3}	6.38×10^{-2}	V_{r_0}	4.19×10^{-1}
0.9	False	15	False	2.0	10.01	1182	13	1.00×10^{-2}	8.69×10^{-4}	8.46×10^{-3}	9.21×10^{-2}	V_{r_0}	8.55×10^{-1}
0.8	True	1	True	2.0	-	-	-	-	-	-	-	-	-
0.8	True	1	False	2.0	10.01	1324	103	1.00×10^{-2}	1.00×10^{-5}	7.56×10^{-3}	4.97×10^{-2}	V_{r_0}	6.92
0.8	True	15	True	2.0	10.01	1259	24	1.00×10^{-2}	1.00×10^{-3}	7.94×10^{-3}	2.87×10^{-1}	V_{r_0}	4.85
0.8	True	15	False	2.0	10.01	1452	32	1.00×10^{-2}	3.44×10^{-5}	6.89×10^{-3}	1.02	V_{r_0}	1.96×10^{1}
0.8	False	1	True	2.0	-	-	-	-	-	-	-	-	-
0.8	False	1	False	2.0	10.01	1475	78	1.00×10^{-2}	1.00×10^{-5}	6.78×10^{-3}	2.15	V_{t_1}	4.26×10^{1}
0.8	False	15	True	2.0	10.00	1159	19	1.00×10^{-2}	1.00×10^{-3}	8.62×10^{-3}	6.27×10^{-2}	V_{r_0}	5.32×10^{-1}
0.8	False	15	False	2.0	10.01	1213	17	1.00×10^{-2}	7.72×10^{-4}	8.24×10^{-3}	9.39×10^{-2}	V_{r_0}	9.44×10^{-1}
0.7	True	1	True	2.0	-	-	-	-	-	-	-	-	-
0.7	True	1	False	2.0	10.01	1347	178	1.00×10^{-2}	1.00×10^{-5}	7.43×10^{-3}	5.13×10^{-2}	V_{r_0}	1.27×10^{1}
0.7	True	15	True	2.0	10.01	1278	30	1.00×10^{-2}	1.00×10^{-3}	7.83×10^{-3}	2.89×10^{-1}	V_{r_0}	4.99
0.7	True	15	False	2.0	10.01	1482	40	1.00×10^{-2}	4.11×10^{-5}	6.75×10^{-3}	1.02	V_{r_0}	1.96×10^{1}
0.7	False	1	True	2.0	-	-	-	-	-	-	-	-	-
0.7	False	1	False	2.0	10.00	1831	335	1.00×10^{-2}	1.00×10^{-5}	5.46×10^{-3}	3.86×10^{3}	V_{t_0}	7.77×10^{4}
0.7	False	15	True	2.0	10.01	1199	24	1.00×10^{-2}	1.00×10^{-3}	8.34×10^{-3}	6.34×10^{-2}	V_{r_0}	8.92×10^{-1}
0.7	False	15	False	2.0	10.00	1231	23	1.00×10^{-2}	6.76×10^{-4}	8.12×10^{-3}	9.62×10^{-2}	V_{r_0}	1.21

Continued on next page.

Table A.3: Complete results from simulations of STATCOM system *(continued)*.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
0.6	True	1	True	2.0	-	-	-	-	-	-	-	-	-
0.6	True	1	False	2.0	10.01	1349	158	1.00×10^{-2}	1.00×10^{-5}	7.41×10^{-3}	5.08×10^{-2}	V_{r_0}	1.12×10^{1}
0.6	True	15	True	2.0	10.01	1327	36	1.00×10^{-2}	1.00×10^{-3}	7.54×10^{-3}	2.97×10^{-1}	V_{r_0}	5.28
0.6	True	15	False	2.0	10.00	1583	51	1.00×10^{-2}	3.53×10^{-5}	6.32×10^{-3}	1.12	V_{r_0}	2.17×10^{1}
0.6	False	1	True	2.0	-	-	-	-	-	-	-	-	-
0.6	False	1	False	2.0	-	-	-	-	-	-	-	-	-
0.6	False	15	True	2.0	10.00	1247	28	1.00×10^{-2}	1.00×10^{-3}	8.02×10^{-3}	6.30×10^{-2}	V_{r_0}	1.19
0.6	False	15	False	2.0	10.00	1244	22	1.00×10^{-2}	5.79×10^{-4}	8.04×10^{-3}	9.54×10^{-2}	V_{r_0}	1.16
0.5	True	1	True	2.0	-	-	-	-	-	-	-	-	-
0.5	True	1	False	2.0	10.01	1369	288	1.00×10^{-2}	1.00×10^{-5}	7.31×10^{-3}	5.08×10^{-2}	V_{r_0}	2.12×10^{1}
0.5	True	15	True	2.0	10.01	1416	39	1.00×10^{-2}	1.00×10^{-3}	7.06×10^{-3}	2.96×10^{-1}	V_{r_0}	5.35
0.5	True	15	False	2.0	10.01	1615	50	1.00×10^{-2}	1.94×10^{-5}	6.19×10^{-3}	9.62×10^{-1}	V_{r_0}	1.86×10^{1}
0.5	False	1	True	2.0	-	-	-	-	-	-	-	-	-
0.5	False	1	False	2.0	-	-	-	-	-	-	-	-	-
0.5	False	15	True	2.0	10.01	1233	29	1.00×10^{-2}	1.00×10^{-3}	8.11×10^{-3}	6.34×10^{-2}	V_{r_0}	1.26
0.5	False	15	False	2.0	10.00	1266	28	1.00×10^{-2}	4.83×10^{-4}	7.90×10^{-3}	9.43×10^{-2}	V_{r_0}	1.47

Table A.4: Complete results from simulations of MTDC system.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
1.0	True	1	True	0.1	1.40	290	80	1.00×10^{-2}	7.73×10^{-4}	4.82×10^{-3}	9.49×10^{3}	V_{qt_2}	5.67
1.0	True	1	False	0.1	1.40	297	83	1.00×10^{-2}	3.51×10^{-4}	4.71×10^{-3}	9.09×10^{3}	V_{qt_2}	5.92
1.0	True	15	True	0.1	1.40	269	15	1.00×10^{-2}	1.00×10^{-3}	5.20×10^{-3}	1.13×10^{4}	V_{qt_2}	4.33×10^{-1}
1.0	True	15	False	0.1	1.40	279	13	8.67×10^{-3}	1.00×10^{-3}	5.02×10^{-3}	8.37×10^{3}	V_{qt_2}	1.19×10^{-1}
1.0	False	1	True	0.1	1.40	288	77	1.00×10^{-2}	7.73×10^{-4}	4.86×10^{-3}	9.39×10^{3}	V_{qt_2}	5.42
1.0	False	1	False	0.1	1.40	296	76	1.00×10^{-2}	3.51×10^{-4}	4.73×10^{-3}	9.49×10^{3}	V_{qt_2}	5.34
1.0	False	15	True	0.1	1.40	255	12	1.00×10^{-2}	1.00×10^{-3}	5.49×10^{-3}	1.28×10^{4}	V_{qt_2}	5.29×10^{-1}
1.0	False	15	False	0.1	1.40	264	13	1.00×10^{-2}	1.00×10^{-3}	5.31×10^{-3}	1.05×10^{4}	V_{qt_2}	2.70×10^{-1}
0.9	True	1	True	0.1	1.40	297	94	1.00×10^{-2}	6.95×10^{-4}	4.71×10^{-3}	8.53×10^{3}	V_{dt_2}	6.83
0.9	True	1	False	0.1	1.40	308	68	1.00×10^{-2}	3.12×10^{-4}	4.54×10^{-3}	9.10×10^{3}	V_{qt_2}	4.67
0.9	True	15	True	0.1	1.40	280	14	1.00×10^{-2}	1.00×10^{-3}	4.99×10^{-3}	1.00×10^{4}	V_{qt_2}	2.72×10^{-1}
0.9	True	15	False	0.1	1.41	281	15	9.80×10^{-3}	1.00×10^{-3}	4.99×10^{-3}	1.15×10^{4}	V_{qt_2}	4.59×10^{-1}
0.9	False	1	True	0.1	1.41	296	91	1.00×10^{-2}	6.95×10^{-4}	4.74×10^{-3}	1.02×10^{4}	V_{qt_2}	6.59
0.9	False	1	False	0.1	1.40	306	63	1.00×10^{-2}	3.12×10^{-4}	4.57×10^{-3}	9.18×10^{3}	V_{qt_2}	4.25
0.9	False	15	True	0.1	1.40	265	12	1.00×10^{-2}	1.00×10^{-3}	5.28×10^{-3}	9.78×10^{3}	V_{dt_2}	1.73×10^{-1}
0.9	False	15	False	0.1	1.41	279	12	1.00×10^{-2}	1.00×10^{-3}	5.04×10^{-3}	9.53×10^{3}	V_{qt_2}	1.61×10^{-1}
0.8	True	1	True	0.1	1.41	304	40	1.00×10^{-2}	6.25×10^{-4}	4.62×10^{-3}	1.01×10^{4}	V_{qt_2}	2.35
0.8	True	1	False	0.1	1.40	313	40	1.00×10^{-2}	2.74×10^{-4}	4.48×10^{-3}	9.44×10^{3}	V_{qt_2}	2.34
0.8	True	15	True	0.1	1.40	297	13	8.00×10^{-3}	1.00×10^{-3}	4.72×10^{-3}	9.91×10^{3}	V_{qt_2}	2.46×10^{-1}
0.8	True	15	False	0.1	1.40	303	16	9.89×10^{-3}	1.00×10^{-3}	4.62×10^{-3}	9.72×10^{3}	V_{qt_2}	4.03×10^{-1}
0.8	False	1	True	0.1	1.40	302	37	1.00×10^{-2}	6.25×10^{-4}	4.65×10^{-3}	1.02×10^{4}	V_{qt_2}	2.10
0.8	False	1	False	0.1	1.40	315	35	1.00×10^{-2}	2.74×10^{-4}	4.45×10^{-3}	8.98×10^{3}	V_{qt_2}	1.93
0.8	False	15	True	0.1	1.40	282	16	1.00×10^{-2}	1.00×10^{-3}	4.96×10^{-3}	1.19×10^{4}	V_{qt_2}	5.46×10^{-1}
0.8	False	15	False	0.1	1.40	288	16	1.00×10^{-2}	9.61×10^{-4}	4.86×10^{-3}	1.02×10^{4}	V_{dt_2}	4.15×10^{-1}
0.7	True	1	True	0.1	1.40	313	63	1.00×10^{-2}	6.25×10^{-4}	4.47×10^{-3}	8.92×10^{3}	V_{qt_2}	4.25
0.7	True	1	False	0.1	1.40	326	75	1.00×10^{-2}	2.36×10^{-4}	4.29×10^{-3}	8.78×10^{3}	V_{qt_2}	5.25
0.7	True	15	True	0.1	1.40	314	18	8.00×10^{-3}	1.00×10^{-3}	4.46×10^{-3}	9.10×10^{3}	V_{qt_2}	5.41×10^{-1}
0.7	True	15	False	0.1	1.40	319	18	8.46×10^{-3}	1.00×10^{-3}	4.40×10^{-3}	1.11×10^{4}	V_{qt_2}	6.29×10^{-1}
0.7	False	1	True	0.1	1.40	311	60	1.00×10^{-2}	6.25×10^{-4}	4.50×10^{-3}	8.83×10^{3}	V_{qt_2}	4.00
0.7	False	1	False	0.1	1.40	322	69	1.00×10^{-2}	2.36×10^{-4}	4.35×10^{-3}	9.32×10^{3}	V_{qt_2}	4.76
0.7	False	15	True	0.1	1.40	300	17	1.00×10^{-2}	1.00×10^{-3}	4.66×10^{-3}	8.82×10^{3}	V_{dt_2}	4.46×10^{-1}
0.7	False	15	False	0.1	1.40	315	16	1.00×10^{-2}	8.41×10^{-4}	4.44×10^{-3}	9.09×10^{3}	V_{qt_2}	3.94×10^{-1}

Continued on next page.

Table A.4: Complete results from simulations of MTDC system *(continued)*.

Step scaling factor	Step increase limit	Step fixed for [steps]	Step decrease limit	Disturbance [s]	Simulated time [s]	Steps	Step changes	Max. step length [s]	Min. step length [s]	Mean step length [s]	Maximum MSE		Distance from ideal
											Value	Variable	
0.6	True	1	True	0.1	1.40	305	47	1.00×10^{-2}	6.25×10^{-4}	4.59×10^{-3}	8.54×10^{3}	V_{qt_2}	2.92
0.6	True	1	False	0.1	1.40	327	91	1.00×10^{-2}	2.00×10^{-4}	4.28×10^{-3}	8.79×10^{3}	V_{qt_2}	6.59
0.6	True	15	True	0.1	1.40	326	16	8.00×10^{-3}	1.00×10^{-3}	4.30×10^{-3}	9.43×10^{3}	V_{qt_2}	4.17×10^{-1}
0.6	True	15	False	0.1	1.40	354	21	8.00×10^{-3}	1.00×10^{-3}	3.95×10^{-3}	9.14×10^{3}	V_{qt_2}	8.06×10^{-1}
0.6	False	1	True	0.1	1.40	307	52	1.00×10^{-2}	6.25×10^{-4}	4.56×10^{-3}	9.06×10^{3}	V_{qt_2}	3.34
0.6	False	1	False	0.1	1.40	318	75	1.00×10^{-2}	2.00×10^{-4}	4.40×10^{-3}	8.64×10^{3}	V_{dt_2}	5.25
0.6	False	15	True	0.1	1.40	310	15	1.00×10^{-2}	1.00×10^{-3}	4.53×10^{-3}	9.83×10^{3}	V_{qt_2}	3.51×10^{-1}
0.6	False	15	False	0.1	1.40	306	19	1.00×10^{-2}	7.20×10^{-4}	4.57×10^{-3}	1.06×10^{4}	V_{dt_2}	6.63×10^{-1}
0.5	True	1	True	0.1	1.41	314	90	1.00×10^{-2}	6.25×10^{-4}	4.47×10^{-3}	9.59×10^{3}	V_{qt_2}	6.50
0.5	True	1	False	0.1	1.41	329	107	1.00×10^{-2}	1.64×10^{-4}	4.27×10^{-3}	9.39×10^{3}	V_{qt_2}	7.92
0.5	True	15	True	0.1	1.40	356	20	8.00×10^{-3}	1.00×10^{-3}	3.94×10^{-3}	9.28×10^{3}	V_{qt_2}	7.32×10^{-1}
0.5	True	15	False	0.1	1.40	390	25	8.01×10^{-3}	1.00×10^{-3}	3.60×10^{-3}	9.16×10^{3}	V_{qt_2}	1.14
0.5	False	1	True	0.1	1.41	313	97	1.00×10^{-2}	6.25×10^{-4}	4.49×10^{-3}	9.43×10^{3}	V_{qt_2}	7.09
0.5	False	1	False	0.1	1.40	330	128	1.00×10^{-2}	1.64×10^{-4}	4.25×10^{-3}	9.03×10^{3}	V_{qt_2}	9.67
0.5	False	15	True	0.1	1.40	362	24	1.00×10^{-2}	1.00×10^{-3}	3.87×10^{-3}	1.07×10^{4}	V_{qt_2}	1.08
0.5	False	15	False	0.1	1.40	393	26	1.00×10^{-2}	6.00×10^{-4}	3.56×10^{-3}	8.84×10^{3}	V_{qt_2}	1.22

Erik-Anant Stedjan Narayan

Dynamically Simulating Power Systems Using a Self-Adaptive Time Step Method

NTNU
Norwegian University of
Science and Technology