Extending the Frequency Bandwidth of Transient Stability Simulation Using Dynamic Phasors

by

Matthew Alexander Kulasza

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

Abstract

Transient stability is an important measure in assessing a system's overall stability. It characterizes the ability of a power system to maintain stability following major disturbances. Modern power systems whose dynamics are increasingly complex and shaped by fast power electronic devices are testing the assumptions and limitations of existing transient stability simulation methods. This work presents a new method developed to improve transient stability simulation by extending its frequency bandwidth through dynamic phasors. This method constructs a continuous set of differential algebraic equations modeling a power system's dynamics using modified nodal analysis. These equations are numerically solved using a general purpose differential algebraic equations solver based on a variable step and variable order algorithm. This selection allows the method developed in this work to conduct accurate simulations while maintaining the advantages of demodulation in phasor-based analysis. Models are included for linear ac networks as well as two important components for power system simulation. The first is a new synchronous machine model, which uses a direct reference frame transformation to represent a synchronous machine as an inductance with nonlinear coupling to its rotor. The second is an HVdc transmission system model, which includes a converter component that is capable of accurately simulating low impedance faults near converter terminals. This accuracy is achieved through an ac current that is directly coupled to ac voltage transients and event monitoring to apply state changes when adverse operating conditions are detected. Simulation results are presented from a five hundred bus test system, which includes two HVdc transmission line models. The results show that the method is accurate for severe disturbances and is three hundred times faster than a detailed EMT simulation. This system was modified to also include a multimass turbine model and tuned to produce unstable subsynchronous oscillations. The results show that this method accurately simulates negatively damped modes and is still over thirty times faster than EMT simulations. Finally, results of a two thousand bus system are also presented, which are in good agreement with results from a conventional transient stability simulation program.

To my family, friends, and most of all, Kathryn

"This duck tells me nothing!"

– Daniel O'Mailley, The Rook

Acknowledgements

First I would like to thank my advisor, Dr. Udaya Annakkage, and co-advisor, Dr. Chandana Karawita, for their unwavering belief in me and this work even when I went down winding and unconventional paths. I would also like to thank my committee members, Dr. Ian Jeffrey, Dr. Shaahin Filizadeh, and Dr. Parimala Thulasiraman, for their support and advice during the course of this research. Finally, I would like to thank the external examiner of my presentation, Dr. Juri Jatskevich, for his thorough review of my thesis, positive comments, and insightful questions.

I would like to thank my ever patient wife Kathryn for all of her encouragement and support. Your determination and strength inspire me. Thank you for being there for it all and making sure I think twice about every decision.

I would like to thank my parents and in-laws for always believing in and supporting me no matter what direction I choose, and my sister for listening and helping me maintain perspective.

Last but not least, I would like to thank all of my family and friends who supported me through this journey.

Contents

List of Figures

Chapter 1

Introduction

Transient stability is defined as the ability of a power system's generators to maintain synchronism following significant disturbances, such as a short circuit fault or the loss of a transmission line [1]. Transient stability simulators have become essential tools for the purposes of power system planning and operation. Power systems are necessarily exposed to risks associated with environmental factors such as adverse weather conditions. Transient stability studies measure the capacity of a system to remain intact following the types of disturbances that occur as a result of exposure to the elements. Therefore, accurate simulations that can be used to assess transient stability are important in ensuring that power systems are and continue to be operated in a reliable manner.

In the past, transient stability was primarily affected by a system's electromechanical characteristics. Consequently, transient stability simulation tools were specifically designed to simulate electromechanical oscillations. However, as more power electronic components and renewable resources are integrated into modern grids, the factors influencing transient stability are becoming more complex and adapting simulation techniques is critical to the analysis of transient stability.

1.1 Thesis Statement

The goal of this thesis is to demonstrate that the frequency bandwidth of transient stability simulation is improved by the use of dynamic phasor-based network and component models.

A new method is developed in this thesis to construct equations that are numerically solved using an implicit variable step integrator. Improvement in the context of this thesis is evaluated by comparing results obtained using the new method against electromagnetic transient (EMT) simulation results. EMT simulators are considered the benchmark in power system simulation because they use small time steps to simulate models with a high degree of detail.

1.2 History of Transient Stability Simulation

Prior to digital computers, transient stability for early power systems was analyzed using the swing equation and equal area stability criterion [1], [2]. The swing equation is a second order differential equation describing the motion of a generator's rotor while the equal area criterion uses a generator's power-angle curve to establish stability limits and critical clearing times. The electrical network was represented by admittances and steady state phasor-based calculations were used for the swing equation in this type of analysis. This approach was sufficient for early systems because they were simple and could be represented by one or two machine models. However, interconnection between many small systems to cope with increasing energy demands led to complex power systems with dynamics that could not be determined analytically [1].

The first form of transient stability simulation known as the step-by-step or point-by-point method was developed in response to growing electrical networks [1]–[3]. A network analyzer or calculating board and manual numerical integration would be used together in a leap frog fashion. A network analyzer was a analog computer that could be programmed to solve power flow problems. They included components that mimic the behaviour of generators and transmission lines where power system models were programmed by connecting different components together [4]. At each time step, a network analyzer would provide a steady state operating point including the power outputs of all generators in a network. The swing equation of each generator would be numerically integrated using manual calculations and their power outputs, which were computed by the network analyzer. Rotor angles calculated by the swing equations would be used to update generator voltages, which were then programmed into the network analyzer. This process would be repeated until a system's stability could be assessed.

The combination of a steady state solution with numerical integration in an alternating fashion formed the basis for the first transient stability programs written for computers [4], [5]. In these programs, the differential equations associated with machines and other dynamic devices are represented as a set of first order ordinary differential equations. These equations are generally solved using an explicit numerical integration method such as the modified Euler approach [4]. A set of constraint equations was also required to represent the electrical network. In early programs, a power flow solution could provide the necessary behaviour much like a network analyzer in the step-by-step approach. Power flow equations are based on nodal analysis and the bus admittance matrix so a more general approach is to represent the network as a set of nodal equations [1]. The network constraints are represented as a large set of sparse and predominantly linear algebraic equations. They are constructed using standard nodal analysis techniques and represented by the bus admittance matrix. Dynamic devices such as generators must be represented as current sources in a nodal analysis framework.

A steady state representation of the electrical network is a simplification since the actual electrical network has components with dynamic characteristics as well. This simplification is known as the quasistationary assumption since the network is represented by a steady state model but the voltage and current phasors throughout the network change during the simulations [6]. One advantage of the quasistationary assumption is that relatively large time steps can be used for transient stability simulation since network transients tend to exhibit dynamics with much higher frequencies and faster time constants than electromechanical tranients. Another advantage of this assumption is that it reduces the number of variables required for simulations since dynamic network components must be modeled with their own state variables. Consequently, the quasistationary assumption greatly reduces the computational burden of transient stability simulation, which permits the analysis of large system models for long periods of time [1].

1.3 Problem Definition

The problem that this thesis seeks to address is the limited capacity for transient stability simulators designed and implmeneted using the quasistationary assumption to capture subsynchronous dynamics that are becoming more prevalent in modern power systems. Accurate results were obtained using the quasistationary assumption for early power systems because the factors affecting rotor angle stability were generally unaffected by electrical network dynamics [6]. However, coupling between electrical network and rotor dynamics began in the middle part of the twentieth century with the introduction of series compensation of transmission lines. Studies of series compensated systems demonstrated that the combination of generators and series capacitors could lead to conditions favourable for the excitation of unstable subsynchronous oscillations [7], [8]. The quasistationary assumption cannot be used to study these types of situations since it arises directly from the interaction between synchronous machines and dynamic network components [9]. Power electronic-based components constitute an important area of modern power systems that can also introduce interactions between networks and rotor dynamics. Practical experience with high voltage dc (HVdc) transmission systems in the later part of the twentieth century revealed that the HVdc system controls could interact with generators near the HVdc system's terminals, exciting unstable torsional oscillations [10], [11].

It became evident by the 1990s that power system stability was no longer constrained to rotor angle stability in part due to the interactions between generators and new grid components [12]. Researchers demonstrated that simulations based on the quasistationary assumption must be limited to dynamics with frequencies less than 3 Hz to ensure that the results obtained are accurate and reliable [13]. This result provided justification for the use of the quasistationary assumption in early power systems, where the frequencies of the dynamics of interest did not exceed this limit [6]. However, they also demonstrated that the quasistationary assumption is not suitable for simulation and analysis of dynamics in the range of 10 Hz to 60 Hz, such as torsional modes that arise from the interactions previously mentioned [10], [11], [14]. As a result of these findings, researchers began investigating methods for extending and replacing conventional stability analysis techniques.

Research into modern transient stability simulation has generally fallen into two categories. The first involves developing techniques to make EMT-type simulations of large power system models feasible. EMT-type simulation methods were originally designed to study electrical network transients with fast time constants and frequencies greater than synchronous speed [15]. EMT-type simulators require detailed element and network models and small time steps to accurately simulate EMT phenomena. These simulators were typically used to simulate small subsystems for short periods of time as a consequence of their computational burden. However, it is important to include a large portion of a power system model to properly assess the system's transient stability. Furthermore, transient stability analysis requires detailed synchronous machine models, which can pose some difficulties in EMT-type simulators due to initialization and interfacing requirements [15].

Research has examined the use of parallel and cluster-based computing [16] and hybrid simulation techniques combining EMT-type and conventional transient stability simulators [17] to address the aforementioned issues with transient stability analysis using EMT-type simulators. While this represents a promising area of research and development, EMT-type simulators include a level of modeling detail that is not necessary for general transient stability simulation and analysis. The second category of research into modern transient stability simulation is dynamic phasors, which is an alternative to EMT-type simulators. Research in this category aims to adjust conventional transient stability modeling to improve the phasor-based network model instead of partially or completely abandoning it in favour of EMT-type methods.

The idea of including network dynamics in phasor-based analysis was first examined by researchers performing small signal stability analysis of systems containing specialized equipment. Conventional small signal stability analysis methods are based on the same quasistationary assumption as transient stability analysis [1]. Researchers demonstrated that network dynamics are necessary to accurately analyze a system's stability when considering series compensated transmission lines [9] and power electronics components [12], [18]. These approaches were based on axis transformations and can be regarded as the first informal definitions of dynamic phasors.

As previously discussed, researchers identified a need for including network dynamics in transient stability simulation with limitations on the accuracy of the quasistationary assumption [13]. One of the first approaches to including network dynamics into large signal phasor-based simulation used an axis transformation and series approximations to construct models that include the effects of frequency dependence and variation [19]. Another early approach applied the idea of a Fourier series with time-dependent coefficients and filtering to obtain phasor-based network equations that include the dynamic behaviour of components like inductors and capacitors [20]. These approaches were important first steps towards including network dynamics in large signal phasor-based simulation as they demonstrated a significant improvement in accuracy over the quasistationary assumption. However, both approaches were limited in their usefulness for large system simulation as they were dependent on heuristics rather than formal algorithms.

A formal mathematical framework for dynamic, or time-varying, phasors was first defined for three-phase balanced systems based on a dq-axis transformation [6]. The dynamic phasor concept was then extended to general single-phase and unbalanced systems, though in this situation the original time domain signals were required to have a low pass characteristic [21]. An alternative general framework was later proposed based on analytic signals and the Hilbert transform [22]. A great deal of work has been conducted using the Hilbert transform-based dynamic phasor framework, most notably in frequency adaptative simulation [23]. Another dynamic phasor framework emerged from a seemingly unrelated area of research known as generalized state space averaging [24]. This framework is not limited to the dc and fundamental components of a system model since it is based on a Fourier series with time dependent coefficients. Research using this framework has seen a wide range of applications, including HVdc and flexible ac transmission system (FACTS) [25], [26], SSR analysis [27], and hybrid simulation applications [17], [28].

1.4 Thesis Motivation

The primary motivation of this research is to demonstrate that transient stability simulations with dynamic phasor-based models can accurately capture dynamics that are above the bandwidth of conventional quasistationary models. The application of dynamic phasors to network modeling in transient stability simulation has not received much attention despite the significant amount of research into dynamic phasors. The majority of research performed in dynamic phasors has focused on modeling and simulation of small subsystems and test cases. In the context of transient stability, research has been focused on adding detail to HVdc system [29], [30] and FACTS device [31] models using dynamic phasors within conventional transient stability simulation programs. However, previous research has demonstrated that a significant improvement in the accuracy of small signal stability analysis can be obtained using a dynamic phasor-based network model, even when only part of the network is modeled using dynamic phasors [12], [32]–[34].

The secondary motivation is that new methods for constructing DAEs describing a system's dynamics are required for dynamic phasor-based transient stability simulation. It is not trivial to include dynamic phasor-based network models in existing transient stability simulation programs. Conventional transient stability simulation methods are intrinsically tied to the quasistationary assumption. The dynamic phasor-based network components include a differential component so they are not compatible with the admittance matrix formulation. Furthermore, dynamic phasor-based synchronous machine models cannot be represented as a Norton equivalent because their stator equations must include both the stator flux dynamics and rotor speed deviations.

Another important motivation is that new numerical solution techniques are also required for dynamic phasor-based equations. Explicit integration methods are favoured to perform transient stability simulation to avoid a costly nonlinear solution step in the simulation procedures. Explicit integration methods are sufficient for quasistationary-based equations because high frequency, small time constant electrical network dynamics are eliminated from the system's equations. However, power system equations are quite stiff when both electromagnetic and electromechanical transients are considered in the same model due to the wide range of time constants [35]. Explicit integration methods tend to suffer from numerical instability problems unless the integration step is much smaller than the smallest time constants and largest frequencies. This property is a significant problem for stiff systems of DAEs since long simulation times are required for slow dynamics but small time steps are required for fast dynamics. As a result, implicit integration methods must be examined and tested for dynamic phasor-based transient stability simulation. Furthermore, power system dynamics are generally first dominated by fast electromagnetic transients followed by slow electromechanical transients [1]. This property implies that variable step size and error control are also important properties for any numerical integration methods considered in this research.

1.5 Thesis Objectives

This thesis is comprised of two main objectives. The first is to investigate a new method for constructing dynamic phasor-based model equations suitable for transient stability simulation. Existing circuit simulation techniques can be used to develop a new approach since it is possible to show that an instantaneous circuit is structurally similar to its dynamic phasor equivalent [36]. EMT-type programs use a method called numerical integrator subsitution (NIS), where each continuous element in a system model is substituted with a discrete equivalent [15]. This process involves first applying a numerical integration technique such as the trapezoidal method locally to each individual component. The resultant difference equations are then transformed to obtain a discrete Norton equivalent. A set of equations for the entire power system model is then constructed using the network's discrete equivalent representation along with KCL. This method has seen widespread use owing to its simplicity and flexibility. NIS has been explored for dynamic phasor-based simulation in previous research [37]–[39].

The NIS approach also suffers from problems that are particularly pronounced when considering nonlinear stiff systems of equations. Components with nonlinear voltage-current relationships, such as synchronous machines, cannot be expressed as a discrete Norton equivalent without introducing numerical delays and interface compensation [15]. Furthermore, the simulation time step becomes embedded in the network's equations when differential equations are locally integrated. As a result, it is difficult and computationally inefficient to vary the simulation time step. Therefore, the substitution method is not well-suited to system simulation when variable step size and error control properties are desired.

An alternative to the NIS approach is to instead construct a set of DAEs of the form

$$\mathbf{F}\left(t, \mathbf{y}, \dot{\mathbf{y}}\right) = \mathbf{0}, \tag{1.1}$$

which describes an entire model's dynamics as a continuous system of equations. These equations can then be numerically integrated using any DAE solution routine, including those with error control properties [35]. A dynamic phasor-based simulation method based on the sparse tableau approach that is well-suited to parallel processing has been previously investigated [40].

A novel dynamic phasor-based simulation approach was developed in previous reseach, which was used to simulate the IEEE 39 bus test system for both balanced and unbalanced disturbances [41], [42]. A limitation of the aforementioned methods is that the number of variables and equations rapidly grows as system size increases due to the amount of redundancy in their mathematical models.

There are a number of algorithms for systematically constructing a set of continuous differential equations describing the dynamic behaviour of circuits. The state space approach was one of the earliest methods researched for this purpose. State space models seek a minimal set of dynamic variables such that $\mathbf{y}$ constitutes a linearly independent set of variables [43]. There are many benefits of this approach from the perspective of both analysis and simulation. However, the minimal requirement in practice can prove to be too restrictive. Finding a minimal set of variables and constructing its equations for large systems is difficult and computationally intensive [44]. This situation is further complicated in the presence of components such as ideal transformers and controlled sources, where the task of finding a set of state variables becomes a combinatorial problem [45]–[47].

Research in circuit simulation turned its attention to approaches known as semistate methods due to the difficulties with the state space approach [44], [48]. These methods relax the minimal constraint and work with a set of variables and equations that may be linearly dependent. One such method is the modified nodal analysis (MNA) approach to circuit analysis, which is used by the circuit simulation package SPICE [49]. Therefore, MNA was selected as the analysis method for constructing dynamic-phasor based simulation equations.

The second objective of this thesis is to investigate a new method for numerically solving the dynamic phasor-based equations. Backward differentiation formula (BDF) methods are a common choice for numerically solving DAEs [35]. The BDF family of methods is designed to be particularly well-suited to solving stiff systems of equations and has been used in a number of previous research applications involving transient stability simulation [50], [51]. BDF methods have also been used in EMT simulation with predictors specifically tuned for sinusoidally varying quantities [52]. Therefore, BDF methods were selected to solve the transient stability MNA equations. Specifically, the Implicit Differential-Algebraic (`IDA`) solver software package was used in this work [53].

The third objective of this thesis is to evaluate the performance of the method developed in this work. Multimachine cases are developed containing a combination of HVdc transmission systems and ac equipment to create models of realistic modern power systems. Simulations of these cases are conducted to obtain a thorough understanding of the performance of the dynamic phasor-based method. These simulation cases are compared against a commerical EMT simulation program evaluating the dynamic phasor-based method's performance in terms of both accuracy and speed.

1.6 Thesis Organization

Background information on the core concepts underlying the simulation method presented in this thesis is discussed in Chapter 2. The mathematical framework used to define dynamic phasors is presented in Section 2.1. An introduction to MNA is presented in Section 2.2 with particular focus on using MNA to develop continuous DAEs of the form (1.1). Finally, an brief overview of the application of BDF methods for numerically integrating DAEs is presented in Section 2.3. This discussion includes an outline of the BDF methods as well as practical considerations such as variable initialization and event processing.

The dynamic phasor-based simulation method is presented in Chapter 3. First the linear branch model is presented in Section 3.1, whose variations of represent the majority of linear ac network components. This component is a valuable example of how stamps are constructed for the simulator developed in this work. Details on the program implementation are presented in Section 3.2, including a high level flow chart illustrating the logic used to develop the simulation program. This chapter also includes details on issues discovered while developing the simulator. Practical considerations of mixing real and complex variables as well as classifying network variables is presented in Section 3.3. Limitations on the BDF method imposed by concerns relating to numerical stability are discussed in Section 3.4. Finally, a generalization of numerical deficiencies that are common in power system models is discussed in Section 3.5.

The synchronous machine model for the simulation program is presented in Chapter 4. Background information is discussed in Section 4.1, which includes the rotor part of the synchronous machine model. Section 4.2 presents how the stator is modeled so that it is compatible with a dynamic phasor network model. A general approach to control modeling is presented in Section 4.3, which supports rapid modeling and prototyping of controllers. An extension to the synchronous machine's rotor model for multimass turbines is discussed in Section 4.4. Finally, results of two single machine infinite bus cases validating the synchronous machine, control, and multimass models is presented in Section 4.5.

The HVdc transmission system model for the simulation program is presented in Chapter 5. Steady state operation and modeling details for HVdc converters is presented in Section 5.1. The dynamic phasor model for converters is developed in Section 5.2. Transmission line modeling for the dc system is presented in Section 5.3. The harmonic filters and HVdc controls are discussed in Sections 5.4 and 5.5, respectively. Modifications to the converter and dc transmission system models required to model converter shut down and commutation failure are discussed in Section 5.6. Finally, validation results of the CIGRE HVdc benchmark model are presented in Section 5.7.

Simulation results of large power system models is presented in Chapter 6. Results of the IEEE 68 bus test system are presented in Section 6.1, which validates the simulation method against a commonly used test case. Results of a 500 bus test system are also included in the chapter. Section 6.2 presents the simulation results for an ac system consisting only of synchronous generators. Section 6.3 presents simulation results of the 500 bus ac system with two transmission lines replaced with HVdc transmission systems. Finally, Section 6.4 presents simulation results of the system further modified to include a multimass turbine model near one of the rectifier terminals. The final set of simulation results presented in this chapter are of a 2000 bus ac network, which are included in Section 6.5.

Conclusions and contributions of this thesis are presented and discussed in Chapter 7. Future work and projects to further develop the simulation method are also discussed in this chapter.

Chapter 2

Background

There are three core components to the simulation method and program developed in this work. The background and theory of the dynamic phasor framework that is used to create power system models is discussed in Section 2.1. The dynamic phasor models of a power system are assembled into system of continuous DAEs using MNA, which is described in Section 2.2. Finally, the DAEs produced by MNA must be numerically integrated by an implicit variable step solver, which is discussed in Section 2.3.

2.1 Dynamic Phasors

Steady state analysis of sinusoidal ac circuits is greatly simplified using the concept of a phasor. A linear circuit with sinusoidal sources at a single frequency produces a sinusoidal response at the same frequency in steady state. A sinusoidal signal can be completely described by its magnitude and phase angle encoded in a complex number [54]. The complex number corresponding to a sinusoidal signal is called its phasor. The feature of phasors that simplies circuit analysis is its differentiation property. Time differentiation of sinusoidal signals is equal to multiplication in phasor arithmetic, which greatly reduces the complexity of circuit analysis with capacitors and inductors. This property is exact in steady state analysis but is also a good approximation in low frequency dynamic analysis of ac networks, such as electromechnical transients in power systems. However, it cannot be used to analyze dynamics with characteristics that exceed the bandwidth limitations of the quasistationary assumption discussed in Section 1.3.

The concept of a dynamic phasor was introduced to improve phasor-based analysis methods. Dynamic phasors are complex numbers representing demodulated sinusoidal quantities like steady state phasors. However, dynamic phasors are defined with a phasor calculus framework that does not reduce time differentiation to multiplication [6]. Instead, dynamic phasors have time derivatives, which allows them to capture the dynamic characteristics of components such as capacitors and inductors. This property makes it possible for dynamic phasors to model power systems for studies that require network dynamics, such as a subsynchronous resonance.

Several dynamic phasor frameworks have been developed that differ in their mathematical definition to address a wide variety of system models. The first formalized approach was developed for balanced three-phase systems and is based on the Blondel transformation [6]. A general framework suitable for single-phase and unbalanced systems was later proposed based on analytic signals and the Hilbert transform [22]. A great deal of work has been conducted using the Hilbert transform-based dynamic phasor framework, most notably in frequency adaptative simulation [23]. Another dynamic phasor framework emerged from a seemingly unrelated area of research known as generalized state space averaging [24]. This framework is not limited to the dc and fundamental components of a system model since it is based on a Fourier series with time dependent coefficients. Research using this framework has seen a wide range of applications, including HVdc and flexible ac transmission system (FACTS) [25], [26], SSR analysis [27], and hybrid simulation [17], [28].

The method proposed in this work was developed with the goal of simulating network level transients in large power system models. Therefore, it was considered sufficient to use the balanced three phase dynamic phasor framework. Let f_0 be the fundamental or base frequency of a power system model and $\omega_0 = 2\pi f_0$ be its base angular frequency. The foundation of this framework is the linear time-varying orthonormal transform

$$\mathbf{B}\left(t\right) = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\left(\omega_0 t\right) & -\sin\left(\omega_0 t\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\omega_0 t - \frac{2\pi}{3}\right) & -\sin\left(\omega_0 t - \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \\ \cos\left(\omega_0 t + \frac{2\pi}{3}\right) & -\sin\left(\omega_0 t + \frac{2\pi}{3}\right) & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{2.1}$$

known as the Blondel transformation. The inverse of (2.1) is used to transform three-phase balanced signals to direct and quadrature axis components [6]. Let $X(t)$ and $\delta(t)$ be two real-valued functions of time and let $\mathbf{x}(t)$ be a real-valued vector in $\mathbb{R}^3$ defined as

$$\mathbf{x}(t) = \begin{bmatrix} X(t)\cos\left(\omega_0 t + \theta(t)\right) \\ X(t)\cos\left(\omega_0 t + \theta(t) - \frac{2\pi}{3}\right) \\ X(t)\cos\left(\omega_0 t + \theta(t) + \frac{2\pi}{3}\right) \end{bmatrix}, \tag{2.2}$$

which is a vector representation of a three-phase balanced system. Applying the transpose of the Blondel transform matrix to $\mathbf{x}(t)$ results in

$$\mathbf{B}^{\mathrm{T}}(t)\,\mathbf{x}(t) = \sqrt{3}\begin{bmatrix} X(t)\cos\left(\theta(t)\right) \\ X(t)\sin\left(\theta(t)\right) \\ 0 \end{bmatrix} \tag{2.3}$$

where the first element is the direct axis component of $\mathbf{x}(t)$ and the second is the quadrature axis component. The result of the transformation (2.3) illustrates how the Blondel transform can be used to develop a phasor operator [6]. Let $\mathcal{P}$ be an operator defined as

$$\mathcal{P}\left(\mathbf{x}(t)\right) = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & j & 0 \end{bmatrix}\mathbf{B}^{\mathrm{T}}(t)\,\mathbf{x}(t), \tag{2.4}$$

which maps three-phase balanced quantities to their corresponding dynamic phasor. Substituting the product (2.3) into (2.4)

$$\mathcal{P}\left(\mathbf{x}(t)\right) = X(t)\left(\cos\left(\theta(t)\right) + j\sin\left(\theta(t)\right)\right), \tag{2.5}$$

which simplifies to

$$\mathcal{P}\left(\mathbf{x}(t)\right) = X(t)\,e^{j\theta(t)}. \tag{2.6}$$

using Euler's identity. The right hand side of (2.6) is equal to the dynamic phasor of $\mathbf{x}(t)$, which is denoted by $\overline{X}(t)$. Substituting the dynamic phasor notation into (2.6)

$$\overline{X}(t) = \mathcal{P}\left(\mathbf{x}(t)\right), \tag{2.7}$$

which shows that $\mathcal{P}$ maps balanced three-phase sinusoidal signals defined as shown in (2.2) to their corresponding dynamic phasors. The first component of $\mathbf{x}(t)$, denoted $x_{\mathrm{a}}(t)$, can be recovered using the inverse transformation

$$x_{\mathrm{a}}(t) = \operatorname{Re}\left(\overline{X}(t)\, e^{j\omega_0 t}\right). \tag{2.8}$$

The second, $x_{\mathrm{b}}(t)$, and third, $x_{\mathrm{c}}(t)$, components can be recovered by shifting $x_{\mathrm{a}}(t)$ by the appropriate phase angles. The scaling property

$$\mathcal{P}(\alpha\mathbf{x}(t)) = \alpha\overline{X}(t) \tag{2.9}$$

can be derived from the definition of $\mathcal{P}$ (2.4).

The purpose behind defining a mathematically robust operator such as $\mathcal{P}$ is that it is possible to define phasor-based calculus properties [6]. Consider the time derivative of $\mathbf{x}(t)$, which is found by differentiating (2.2) with respect to time

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{X}(t)\cos(\omega_0 t + \theta(t)) - X(t)\sin(\omega_0 t + \theta(t))\left(\omega_0 + \dot{\theta}(t)\right) \\ \dot{X}(t)\cos\left(\omega_0 t + \theta(t) - \dfrac{2\pi}{3}\right) - X(t)\sin\left(\omega_0 t + \theta(t) - \dfrac{2\pi}{3}\right)\left(\omega_0 + \dot{\theta}(t)\right) \\ \dot{X}(t)\cos\left(\omega_0 t + \theta(t) + \dfrac{2\pi}{3}\right) - X(t)\sin\left(\omega_0 t + \theta(t) + \dfrac{2\pi}{3}\right)\left(\omega_0 + \dot{\theta}(t)\right) \end{bmatrix}. \tag{2.10}$$

Applying the transpose of the Blondel transformation to $\dot{\mathbf{x}}(t)$

$$\mathbf{B}^{\mathrm{T}}(t)\,\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{X}(t)\cos(\theta(t)) - X(t)\left(\omega_0 + \dot{\theta}(t)\right)\sin(\theta(t)) \\ \dot{X}(t)\sin(\theta(t)) + X(t)\left(\omega_0 + \dot{\theta}(t)\right)\cos(\theta(t)) \\ 0 \end{bmatrix}. \tag{2.11}$$

Applying $\mathcal{P}$ to $\dot{\mathbf{x}}(t)$, substituting (2.11) and rearranging the terms

$$\mathcal{P}(\dot{\mathbf{x}}(t)) = \left(\dot{X}(t) + jX(t)\left(\omega_0 + \dot{\theta}(t)\right)\right)(\cos(\theta(t)) + j\sin(\theta(t))). \tag{2.12}$$

Simplifying using the Euler identity and once again rearranging the terms

$$\mathcal{P}\left(\dot{\mathbf{x}}\left(t\right)\right) = \left(\dot{X}\left(t\right) + jX\left(t\right)\dot{\theta}\left(t\right)\right)e^{j\theta(t)} + j\omega_0 X\left(t\right)e^{j\theta(t)}. \tag{2.13}$$

The second term is $\overline{X}\left(t\right)$ with a coefficient of $j\omega_0$, which is the differentiation property for steady state phasors. The first term is equal to the time derivative of $\overline{X}\left(t\right)$, which can be found by differentiating the definition of a dynamic phasor with respect to time and applying the chain rule

$$\dot{\overline{X}}\left(t\right) = \dot{X}\left(t\right)e^{j\theta(t)} + jX\left(t\right)e^{j\theta(t)}\dot{\theta}\left(t\right). \tag{2.14}$$

Substituting $\overline{X}\left(t\right)$ and (2.14) into (2.13)

$$\mathcal{P}\left(\dot{\mathbf{x}}\left(t\right)\right) = \dot{\overline{X}}\left(t\right) + j\omega_0\overline{X}\left(t\right), \tag{2.15}$$

which defines the dynamic phasor differentiation property [6]. This property is the most important characteristic of dynamic phasors. Conventional transient stability simulation based on the quasistationary assumption are still based on the concept of time-varying phasors. The magnitudes and angles of voltages and currents still change over time during a transient stability simulation. However, differentiation in their models is handled using the steady state property where $\dot{\overline{X}}\left(t\right)$ is assumed to be equal to zero at all times [1]. The property shown in (2.15) illustrates that dynamic phasor-based models include the dynamics associated with network derivatives, which is the characteristic that sets them apart from standard steady state phasors.

Differential equations for per unit quantities such as those for synchronous machines can be simplified using a time base and running simulations in per unit time [1]. Let t_{base} be the time base and define its value using the base angular frequency according to

$$t_{\text{base}} = \frac{1}{\omega_0}. \tag{2.16}$$

It is possible to show that the differentiation property (2.15) simplifies to

$$\mathcal{P}\left(\dot{\mathbf{x}}\left(t\right)\right) = \dot{\overline{X}}\left(t\right) + j\overline{X}\left(t\right) \tag{2.17}$$

using the time base defined by (2.16). Furthermore, for brevity in notation the time dependency is assumed in all dynamic phasors and dropped from this point forward such that $\overline{X}$ is equivalent to $\overline{X}\left(t\right)$.

The dynamic phasor equivalents for basic circuit models can be constructed using the dynamic phasor operator and its properties. Linear resistors involve a simple scaling relationship between voltage and current. The per-unit dynamic phasor-based equivalent of a resistor with resistance R is

$$\overline{V} = R\overline{I}. \tag{2.18}$$

Linear capacitors and inductors both involve a scaling and differentation relationship between voltage and current. The dynamic phasor-based equivalent a capacitor with capacitance C is

$$\overline{I} = C\left(\dot{\overline{V}} + j\overline{V}\right). \tag{2.19}$$

Similarly, the dynamic phasor-based equivalent of an inductor with inductance L is

$$\overline{V} = L\left(\dot{\overline{I}} + j\overline{I}\right). \tag{2.20}$$

The phasor-based calculus framework developed for dynamic phasors illustrates that they have a mathematically rigorous foundation. Furthermore, the scaling and differentiation properties are used to develop dynamic phasor-based models of common linear components. These models can be substituted in-place when developing a dynamic phasor-based model for an ac network [36].

2.2 Modified Nodal Analysis

Standard nodal analysis is a popular method for analyzing and simulating circuits owing to the fact that its systematic approach to constructing circuit equations lends itself well to the development of simulation programs. Furthermore, nodal analysis is applicable to a wide variety of networks unlike mesh analysis, which is limited to planar circuits [54]. For a connected network with n nodes, one node is selected as the reference node and the voltage at the remaining $n-1$ nodes relative to the reference are selected as variables in $\mathbf{y}$. KCL is then used at each node, except for the reference node, to obtain $n-1$ equations in terms of the network's branch currents. Provided that each branch current may be expressed in terms of the network's node voltages, it is possible to rewrite the $n-1$ KCL equations in terms of the $n-1$ node voltage variables.

EMT-type simulators use NIS in conjunction with nodal analysis to construct a discrete set of difference equations that model the behaviour of continuous circuits [15]. Nodal analysis can also be used to construct continuous DAEs. Consider the circuit illustrated in Figure 2.1, which has two nodes not including the reference node. It follows that there are two variables in this system, $\overline{V}_1$ and $\overline{V}_2$, and two nodal equations

$$(G_\mathrm{s} + G)\,\overline{V}_1 - G\overline{V}_2 - \overline{I}_\mathrm{s}\,(t) = 0 \text{ and} \tag{2.21}$$

$$C\dot{\overline{V}}_2 + (G + jC)\,\overline{V}_2 - G\overline{V}_1 = 0. \tag{2.22}$$

The node associated with $\overline{V}_1$ does not have any capacitors, which implies that $\overline{V}_1$ and its associated nodal equation (2.21) are algebraic. $\overline{V}_2$ and its associated nodal equation (2.22) are differential as a result of the capacitor connected to ground. Therefore, the DAE system associated with the circuit in Figure 2.1 has two variables and equations, one differential and one algebraic.

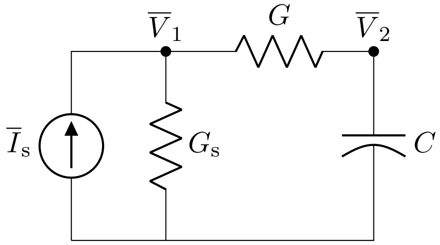

Figure 2.1: Circuit with DAEs that can be constructed using standard nodal analysis

Standard nodal analysis relies on a key assumption that breaks down for many realistic circuit models. Standard nodal analysis requires that each branch current be expressible in terms of the network's node voltages. Linear time-invariant inductors are an example of a simple component that violates this assumption if the equations are limited to DAEs of the form (1.1). Consider the example circuit with the capacitor replaced with an inductor as illustrated in Figure 2.2. The two node voltages are always selected as variables in nodal analysis and the nodal equations are

$$\left(G_{\mathrm{s}}+G\right)\overline{V}_1-G\overline{V}_2-\overline{I}_{\mathrm{s}}\left(t\right)=0 \text{ and} \tag{2.23}$$

$$G\overline{V}_2-G\overline{V}_1+\overline{I}_{\mathrm{L}}=0. \tag{2.24}$$

The nodal equation associated with $\overline{V}_1$ (2.23) is unchanged so its properties remain the same as the first circuit. The nodal equation associated with $\overline{V}_2$ (2.24) no longer includes a differential component for $\overline{V}_2$, so it is now an algebraic equation. (2.24) also includes a term for the inductor's current, $\overline{I}_{\mathrm{L}}$, which is the component that gives these nodal equations their differential characteristics. For a set of pure nodal equations, $\overline{I}_{\mathrm{L}}$ must be eliminated from (2.24) and replaced by an expression in terms of node voltages. However, it is not possible to express $\overline{I}_{\mathrm{L}}$ in terms of node voltages without integration. Therefore, the nodal equations for the circuit shown in Figure 2.2 cannot be expressed as a set of DAEs.

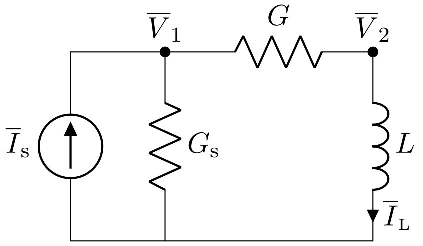

Figure 2.2: Circuit with DAEs that cannot be constructed using standard nodal analysis

MNA introduces a simple modification to the standard nodal analysis method to accommodate elements whose currents are not expressible in terms of node voltages [55]. MNA accommodates elements such as inductors by extending the set of variables to include the current through these branches. An additional constraint equation must also be introduced for each auxiliary current variable to ensure that there are as many equations as there are unknowns. For the example circuit shown in Figure 2.2, the inductor current $\overline{I}_{\mathrm{L}}$ is not eliminated from (2.24). Instead, $\overline{I}_{\mathrm{L}}$ is added as additional variable whose associated constraint equation is simply the inductor's constitutive relationship

$$L\dot{\overline{I}}_{\mathrm{L}} + jL\overline{I}_{\mathrm{L}} - \overline{V}_2 = 0. \tag{2.25}$$

The original two nodal equations (2.23) and (2.24) together with (2.25) constitutes a complete set of DAEs. The two nodal equations and variables are algebraic while the inductor's constraint equation is differential.

The MNA approach is not limited to current-controlled elements either. A variant called the charge-oriented MNA approach was developed for electronic circuit simulation that offers numerical advantages for highly capacitive circuits [56]. In this approach, capacitor charge and inductor flux are also included as system variables and their relationships to capacitor voltage and inductor current are included as constraint equations. This feature is particularly important in power system modeling because a natural way to model synchronous machines is to define the fluxes in the machine windings as variables.

One of the most important features of the MNA approach that makes it attractive for the simulation of general circuit models is that it is trivial to automate through the use of stamps [57]. Stamps are an abstract definition of a component type's effect on a general set of DAEs. Symbolic indexes and variables are used so that the same stamp applies to all instances of a given component type. Actual component instances are distinguished by an assembly routine, which constructs a concrete set of equations for a circuit model by assigning numerical values to stamp correspond to each of the circuit's component instances.

Stamps are frequently visualized using vectors and matrices but they can also be specified simply in terms of constraint equations and symbolic variables [57]. In general, components can specify any number of variables and the constraint equations can involve any combination of variables in the system's variable set. The only constraint on components that are connected to the electrical network is that they must define a current leaving each node to which they are connected. Consider a linear capacitor with capacitance C connected between two nodes i and j whose voltages are $\overline{V}_i$ and $\overline{V}_j$, respectively. A linear capacitor's current can be specified in terms of its voltage so it does not need to define additional variables. The current flowing away from node i, $\overline{I}_i$, is defined as

$$\overline{I}_i = C\left(\dot{\overline{V}}_i - \dot{\overline{V}}_j\right) + jC\left(\overline{V}_i - \overline{V}_j\right). \tag{2.26}$$

Similarly, the current flowing away from node j, $\overline{I}_j$, is defined as

$$\overline{I}_j = C\left(\dot{\overline{V}}_j - \dot{\overline{V}}_i\right) + jC\left(\overline{V}_j - \overline{V}_i\right). \tag{2.27}$$

The stamp defined by (2.26) and (2.27) completely captures any linear capacitor's effect on a system's DAEs. A linear capacitor instance would replace its current with either (2.26) or (2.27) in the KCL equations associated with terminals. How they are added to a system's equations depends on a simulator's specific implementation details.

2.3 Numerical Integration

Numerical integration is at the centre of simulation as it is what is used to estimate the behaviour of continuous DAEs. There are a wide variety of methods for performing numerical integration, all with properties that make them better or worse for solving a particular subset of problems. Selecting the correct integration method depends greatly on the type of problem being solved.

2.3.1 Backward Difference Formula Methods

The BDF methods are a family of multistep numerical integration methods specifically designed for stiff systems of differential equations [35]. They exhibit the stiff decay property, which allows BDF methods to pass through regions of high dynamic activity using large time steps without losing accuracy for low frequency features. Furthermore, numerical stability is maintained even with lower bounds applied to the time step. Explicit methods are generally unstable when the time step is too large and implicit methods without stiff decay are prone to numerical oscillations [35]. These properties are an advantage for power system transient stability analysis where a simulation that captures torsional oscillations but ignores high frequency network transients is sufficiently accurate.

Let t_i be the time of the i-th integration step and let $\mathbf{y}_i$ and $\dot{\mathbf{y}}_i$ be the variables and derivatives of a system of DAEs computed in the i-th integration step, respectively. Substituting these quantities into (1.1)

$$\mathbf{F}\left(t_i, \mathbf{y}_i, \dot{\mathbf{y}}_i\right) = \mathbf{0}, \tag{2.28}$$

which defines a set of n equations with $2n$ unknowns in $\mathbf{y}_i$ and $\dot{\mathbf{y}}_i$. Let h be the integration time step and let $\alpha_{k,j}$ and β_k be coefficients associated with the BDF method of order k. These coefficients are generally scaled such that $\alpha_{k,0}$ is equal to one [35]. The BDF method of order k approximates the derivative of $\mathbf{y}$ at step i according to

$$\dot{\mathbf{y}}_i = \frac{1}{h\beta_k} \sum_{j=0}^{k} \alpha_{k,j} \mathbf{y}_{n-j}. \tag{2.29}$$

Substituting (2.29) into (2.28)

$$\mathbf{F}\left(t_i, \mathbf{y}_i, \frac{1}{h\beta_k} \sum_{j=0}^{k} \alpha_{k,j} \mathbf{y}_{i-j}\right) = \mathbf{0}, \tag{2.30}$$

which is a set of n equations with n unknowns in $\mathbf{y}_i$ assuming that the solution is known at all previous steps. Therefore, the integration is advanced at each step by solving (2.30) for $\mathbf{y}_i$. It is important to note that (2.30) is only used to illustrate how the numerical integration process is advanced. A system's DAEs are defined in terms of $\dot{\mathbf{y}}_i$ in a general purpose solver since the solver automatically calculates the estimate (2.29) and passes that estimate to $\mathbf{F}$.

The system of equations defined in (2.30) is generally nonlinear so it requires a nonlinear solution method to find $\mathbf{y}_i$. The Newton-Raphson method is generally the most popular approach for this step given its stability and convergence properties [35]. This method requires the system's Jacobian matrix, $\mathbf{J}$, which is found by taking the total derivative of (2.30) with respect to $\mathbf{y}_i$

$$\mathbf{J}\left(t_i, \mathbf{y}_i, \dot{\mathbf{y}}_i\right) = \frac{\partial \mathbf{F}}{\partial \mathbf{y}_i} + \frac{1}{h\beta_{k,0}} \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{y}}_i}. \tag{2.31}$$

The first term in (2.31) defines the rate of change of $\mathbf{F}$ with respect to the variables, $\mathbf{y}$. The second term defines the rate of change of $\mathbf{F}$ with respect to the derivatives, $\dot{\mathbf{y}}$. Both of these matrices can be defined analytically using the system's continuous DAEs.

The Newton-Raphson method updates its estimation of a system's solution using a correction term $\Delta\mathbf{y}_i^m$ at step m, which is found by solving

$$\mathbf{J}\left(t_i, \mathbf{y}_i^m, \dot{\mathbf{y}}_i^m\right) \Delta\mathbf{y}_i^m = -\mathbf{F}\left(t_i, \mathbf{y}_i^m, \dot{\mathbf{y}}_i^m\right). \tag{2.32}$$

The derivative terms are approximated using $\mathbf{y}_i^m$ and the known terms from previous steps of the numerical integration routine. The solution is updated according to

$$\mathbf{y}_i^{m+1} = \mathbf{y}_i^m + \Delta\mathbf{y}_i^m. \tag{2.33}$$

Iterations are terminated when the norm of $\Delta\mathbf{y}_i^m$ falls below a user defined tolerance. A factor can also be included that is based on the convergence rate of the iterations. This factor accelerates the decision to exit when the convergence rate is high, and tends to perform extra iterations when the convergence rate is slow [35].

The most expensive step in the iteration process is the solution of a system of linear equations, which occurs once per iteration of the Newton-Raphson process [35]. There are two types of solvers that can be used at this stage for sparse systems. The first type is sparse direct solvers, which typically use a sparse *LU* factorization followed by forward and backward solutions. A pure implementation of the Newton-Raphson process would involve evaluating the Jacobian and computing its sparse factorization each time a linear solution is required. However, *LU* factorizations are generally expensive to compute so this approach would become extremely slow for large systems. A modified procedure is used at this stage instead to significantly increase the speed of BDF solvers. This procedure only evaluates the Jacobian and computes its factorization when iterations fail or the solver's step size has changed by a sufficient enough margin [35]. The other type of linear system solver that can be used at this stage is a Jacobian-free iterative solver [58]. Iterative linear system solvers avoid sparse matrices altogether and instead use matrix vector products to find an approximate solution to a system of equations. However, they require good preconditioners to obtain acceptable convergence rates.

The system of equations defined by (2.30) together with the modified Newton-Raphson iteration process define a fixed step and fixed order BDF solver. However, advanced implementations of the BDF method use variable step and take advantage of the fact that there are multiple orders in the same family of methods. The advantage of variable step is that the solver can use small time steps when required and otherwise pass through regions of low dynamic activity quickly using large time steps. Higher order methods are able to maintain a high degree of accuracy while using larger time steps when compared with their lower order counterparts [35]. Combining these two degrees of freedom leads to a sophisticated solver that is both accurate and fast.

2.3.2 Initial Conditions

Locating initial conditions is an important part of practical numerical solvers. Generally speaking, a DAE simulator needs to solve two types of initial condition problems. The first are initial conditions that are consistent with steady state assumptions at the beginning of a simulation. More generally, these initial conditions assume that the system's derivatives, $\dot{\mathbf{y}}$, are known and the solution seeks to find the system's variables, $\mathbf{y}$. The steady state initial condition problem is a generalization of the first initial condition problem since the solution seeks $\mathbf{y}$ assuming that $\dot{\mathbf{y}}$ is zero. The second type of initial condition problem is consistent with initial conditions that are sought after an event occurs. The assumption at this point is that the values of all differential variables in $\mathbf{y}$ must remain constant through the event. Otherwise, their derivatives, which appear in $\mathbf{F}$ take on values that are not numeric. Therefore, the second type of initial condition problem seeks the values of algebraic variables and the derivatives of differential variables given the values of differential variables.

An ideal initial conditions solver should take advantage of the routines that compute a system's residual and Jacobian when solving both types of problems. The first problem type is already formulated in a manner that uses the original Jacobian and residual. This problem involves simply solving (2.28) with a known value of the derivative vector. An estimate of the derivatives is not required since they are known in this type of problem. It follows that the Jacobian reduces to

$$\mathbf{J}\left(t, \mathbf{y}, \dot{\mathbf{y}}\right) = \frac{\partial \mathbf{F}}{\partial \mathbf{y}} \tag{2.34}$$

since $\dot{\mathbf{y}}$ is known. An important assumption in this problem type is that the system's Jacobian is nonsingular when the derivative components are ignored. This assumption generally holds but there are some models that break this assumption, such as pure integrators.

The second initialization problem type is more difficult to express in terms of the existing residual and Jacobian since the original Newton-Raphson process is formulated to solve for $\mathbf{y}_i$. It is possible to show that an estimate of the Jacobian required to solve the second initial conditions problem can be derived from the original Jacobian (2.31) through a scaling operation. This estimate provides a sufficient approximation to the Jacobian required to solve the second problem type for convergence of Newton-Raphson iterations, especially if the time step used to construct (2.31) is small. Furthermore, an approximate Jacobian is sufficient particularly if the Newton-Raphson iterations are improved through the line search algorithm, which is an optimized gradient search algorithm. However, the second problem type requires some additional information that the first does not require. The system must supply information about each variable classifying them as either different or algebraic. This information is obvious in many cases but it may be difficult obtain for variables whose equations consist of contributions from several different sources.

2.3.3 Event Location

Many real world systems are not completely continuous over time and may experience discontinuities or events. Events can be large and induce dramatic change in the system's variables, such as faults or components tripping. Events can also be a relatively small point of discontinuous behaviour, such as a limited component reaching its limits. A potential solution to these issues is to simply integrate through these events and apply the changes at the next possible step. However, this approach leads to errors that degrade the accuracy of the entire solution [35]. However, the solution is not affected if discontinuities occur at known points where the integration can be paused, the effects of an event result, and finally integration continued with valid initial conditions. Therefore, it is important to locate these events, adjust the system's equations, and use an initial condition solver to update the solution.

Events can be divided into two categories. The first type are events where the time point when they occur is known before the solution begins. These events can be handled by simply stopping the simulation at the required time, applying the required actions and computing new initial conditions, and then carrying on with the new DAEs. The second type are events that are triggered by a condition imposed on a combination of variables and inputs. These events are more difficult to handle as they occur at an unknown time and generally occur between time steps.

Events that fall in the second category can be located using a set of functions, $\mathbf{g}$, whose zero crossings represent state changes [59]. The root finding procedure attempts to locate a time t_{c} on the interval $[t_{\mathrm{a}}, t_{\mathrm{b}}]$ with corresponding values $\mathbf{y}_{\mathrm{c}}$ and $\dot{\mathbf{y}}_{\mathrm{c}}$ such that there is a g_i in $\mathbf{g}$ that satisfies

$$g_i\left(t_{\mathrm{c}}, \mathbf{y}_{\mathrm{c}}, \dot{\mathbf{y}}_{\mathrm{c}}\right) = 0. \tag{2.35}$$

If there are multiple functions satisfying (2.35) on the interval then the algorithm attempts to find the point that is closest to t_{a}. The value of each function in $\mathbf{g}$ is initialized at the beginning of the simulation and after each time step. A sign change of any function g_i implies that root exists within the time step. However, a simple sign check is prone to missing or erroneously reporting roots so additional steps must be added to obtain a robust root finding algorithm [59]. This algorithm must also compute the direction of zero crossings as many functions are only defined for a single direction.

Chapter 3

Method

The objective of this work is define an approach to constructing a general continuous set of DAEs of the form given by (1.1) for power system transient simulation using dynamic phasors and MNA. Minimizing the time to compute $\mathbf{F}$ is important for simulations using error control as $\mathbf{F}$ needs to be computed several times per step. The dynamic phasor-based simulation method proposed in this work uses MNA to construct a set of DAEs that has the partitioned form

$$\mathbf{F}\left(t_i, \mathbf{y}_i, \dot{\mathbf{y}}_i\right) = \mathbf{u}_i + \mathbf{T}\dot{\mathbf{y}}_i + \mathbf{A}\mathbf{y}_i + \mathbf{f}\left(t_i, \mathbf{y}_i, \dot{\mathbf{y}}_i\right) = \mathbf{0}. \tag{3.1}$$

The system's Jacobian is found by substituting (3.1) into (2.31), which is

$$\mathbf{J}\left(t_i, \mathbf{y}_i, \dot{\mathbf{y}}_i\right) = \mathbf{A} + \frac{\partial \mathbf{f}}{\partial \mathbf{y}_i} + \frac{1}{h\beta_k}\left(\mathbf{T} + \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{y}}_i}\right). \tag{3.2}$$

The vector $\mathbf{u}$ represents the constant inputs to a system's equations, such as the voltage at an infinite bus or the field voltage for a generator without an exciter. The sparse matrices $\mathbf{A}$ and $\mathbf{T}$ represent the constant coefficients for $\mathbf{y}$ and $\dot{\mathbf{y}}_i$, respectively. For example, the coefficients associated with lines and transformers are included in a system's equations through $\mathbf{A}$ and $\mathbf{T}$. Finally, the vector $\mathbf{f} \in \mathbb{R}^n$ represents the nonlinear and time-dependent components of a system's equations. For example, the electromagnetic torque in a generator's speed equation is included in a system's equations using $\mathbf{f}$ because it is a nonlinear combination of the generator's stator current and air-gap flux.

Partitioning a system's equations localizes the linear contributions into two sparse matrices. Routines for strictly linear components, such as branches and loads, are not required to compute **F**. This property is important as experimental results demonstrated that the number of routines that must be accessed when **F** is computed has a significant impact on simulation time. Furthermore, the two sparse matrix products in (3.1) can be performed in parallel to improve performance. The compressed sparse row (CSR) format was selected for **A** and **T** as it offers a small computational advantage over the compressed sparse column (CSC) format [60]. Furthermore, CSR-based sparse matrix-vector products are embarrassingly parallel, which can be used to improve simulation performance for large systems.

Section 3.1 will discuss the various combinations and associated stamps for the linear branch component, which is the building block and most common component encountered in power system models. A high-level overview of the program implemented to conduct the dynamic phasor-based simulations including a flowchart of the program is included in Section 3.2. The three succeeding sections include the motivation behind certain settings and modifications for the simulation program. Section 3.3 outlines the requirements for mixing complex and real variables and how differential and algebraic variables were identified for the numerical integration routines. Section 3.4 demonstrates why BDF methods must be limited to a maximum of second order limits when solving power system equations. Finally, Section 3.5 discusses numerical modifications required to ensure that the solver does not fail to converge when certain types of common equation deficiencies are present in a power system model.

3.1 Linear Branch

The majority of branches in power system models may be modeled using a combination of resistors, capacitors, inductors, and ideal transformers. A general version of the power system branch model is illustrated in Figure 3.1. It can be used to model the series component in either a π-section transmission line model or a transformer model. The advantage of this representation is that it eliminates a node voltage in its most common use, which is a series combination of a resistor and an inductor.

The stamp for this model depends on which of the resistance, capacitance, and inductance

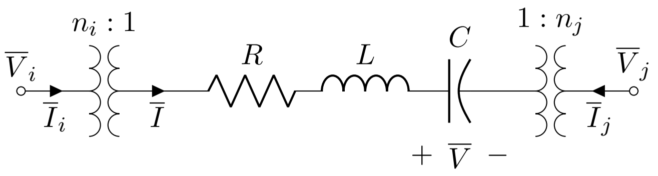

Figure 3.1: General linear branch

are nonzero. The tap ratios are always included in the model's equations and assumed to be nonzero. They are eliminated from a model by setting their value to one. The general stamp assumes that all three components in the linear branch are nonzero. There are two auxiliary variables in this case, one representing the branch current, $\overline{I}$, and one representing the capacitor voltage, $\overline{V}$. Their associated constraint equations are

$$L\dot{\overline{I}} + (R + jL)\,\overline{I} + \overline{V} - \frac{\overline{V}_i}{n_i} + \frac{\overline{V}_j}{n_j} = 0 \quad \text{and} \tag{3.3}$$

$$C\dot{\overline{V}} + jC\overline{V} - \overline{I} = 0, \tag{3.4}$$

which implies that both auxiliary variables are differential. The i and j port currents are related to the main branch current through their respective tap ratios and are given by

$$\overline{I}_i = \frac{\overline{I}}{n_i^*} \quad \text{and} \tag{3.5}$$

$$\overline{I}_j = -\frac{\overline{I}}{n_j^*}. \tag{3.6}$$

The most common variation is a series combination of an inductor and a resistor. The capacitor voltage variable, $\overline{V}$, and its associated constraint equation (3.4) are simply removed in this case. A special case of this variation is an inductor, which is the same model with R set to zero.

The inductor current variable, $\overline{I}$, must be eliminated in the variation consisting of a series combination of a capacitor and a resistor. The new constraint equation is found by rearranging (3.4) to

$$\overline{I} = C\dot{\overline{V}} + jC\overline{V} \tag{3.7}$$

and substituting into (3.3),

$$RC\dot{\overline{V}} + (1 + jRC)\,\overline{V} - \frac{\overline{V}_i}{n_i} + \frac{\overline{V}_j}{n_j} = 0. \tag{3.8}$$

The branch currents are also found by subsituting (3.7) into (3.5) and (3.6),

$$\overline{I}_i = \frac{C\dot{\overline{V}} + jC\overline{V}}{n_i^*} \quad \text{and} \tag{3.9}$$

$$\overline{I}_j = -\frac{C\dot{\overline{V}} + jC\overline{V}}{n_j^*}. \tag{3.10}$$

The final two variations are a single resistor and a single capacitor. No auxiliary variables or constraint equations are required in both cases. The port currents for the case of a single capacitor are

$$\overline{I}_i = C\left(\frac{\dot{\overline{V}}_i + j\overline{V}_i}{|n_i|^2} - \frac{\dot{\overline{V}}_j + j\overline{V}_j}{n_i^* n_j}\right) \quad \text{and} \tag{3.11}$$

$$\overline{I}_j = C\left(\frac{\dot{\overline{V}}_j + j\overline{V}_j}{|n_j|^2} - \frac{\dot{\overline{V}}_i + j\overline{V}_i}{n_i n_j^*}\right). \tag{3.12}$$

Similary, the port currents for the case of a single resistor are

$$\overline{I}_i = \frac{1}{R}\left(\frac{\overline{V}_i}{|n_i|^2} - \frac{\overline{V}_j}{n_i^* n_j}\right) \quad \text{and} \tag{3.13}$$

$$\overline{I}_j = \frac{1}{R}\left(\frac{\overline{V}_j}{|n_j|^2} - \frac{\overline{V}_i}{n_i n_j^*}\right). \tag{3.14}$$

All component parameters illustrated in Figure 3.1 were assumed to be complex to maximize the component's generality. Phase-shifting transformers use complex tap ratios, n_i and n_j [1]. The resistance, R, can be complex for a quasi-stationary branch model, which is useful for hybrid network models. Finally, the inductance, L, and capacitance, C, are generally real numbers, but the equations are complex so real values for L and C are simply a special case of the general model.

3.2 Implementation

A flowchart illustrating a high level overview of the dynamic phasor simulation program's general structure is provided in Figure 3.2. The flowchart shows that the program can be split into two main phases, which are initialization and simulation. The initialization phase is responsible for reading model data and ensuring its integrity, initializing variables and equations, and preparing the data for the simulation phase. The simulation section of the program is responsible for coordinating activities with the numerical solver, including computing the equations and Jacobians, and managing events. Sections 3.2.1 and 3.2.2 discuss the details of the initialization and simulation phases, respectively.

3.2.1 Initialization

The initialization stage begins by reading into memory the system's model data. This step includes a data integrity check that ensures there are no deficiencies such as a branch that is connected to an undefined bus. An initial operating point as calculated by an external power flow application must be provided with the system's data. The power flow solution provides an estimate to the initial state of the dynamic equations, which is the general approach to initializing transient stability simulations [1]. The power flow solution is computed using realistic operational information such as machine terminal voltages and power outputs, which components can use to calculate internal unknowns. Quantities such as controller references, which may not correlate with any real world values, would be required at initialization time without power flow data.

The next step in the initialization process is to determine an initialization order. Not all components in the system can be initialized in an arbitrary order, some depend on calculations from others. For example, a synchronous machine must be initialized before its exciter and governor as they require the initial field voltage and mechanical power values, respectively. If a model is restricted to certain types of components, initialization orders can be predetermined and hard-coded into a program. However, this does not leave room for flexibility for new types of components or generic systems, like the generic control block approach discussed in Section 4.3. A more general approach is to use dependency graphs and perform a topological

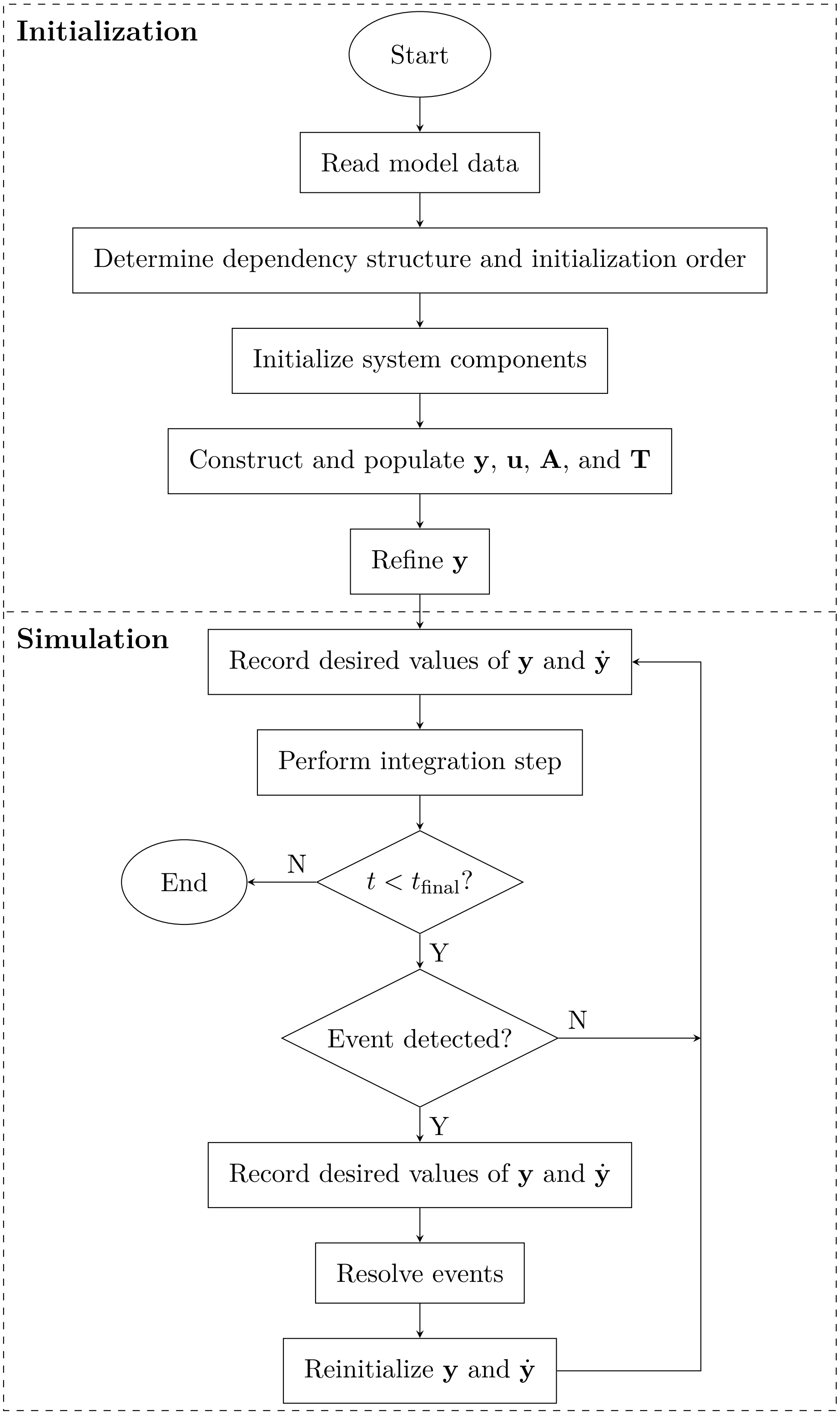
Initialization
Start
Read model data
Determine dependency structure and initialization order
Initialize system components
Construct and populate $\mathbf{y}$, $\mathbf{u}$, $\mathbf{A}$, and $\mathbf{T}$
Refine $\mathbf{y}$
Simulation
Record desired values of $\mathbf{y}$ and $\dot{\mathbf{y}}$
Perform integration step
End
N
$t < t_{\text{final}}$?
Y
Event detected?
N
Y
Record desired values of $\mathbf{y}$ and $\dot{\mathbf{y}}$
Resolve events
Reinitialize $\mathbf{y}$ and $\dot{\mathbf{y}}$

Figure 3.2: Flow chart illustrate the structure of the dynamic phasor transient stability simulation program

sort to establish an initialization order.

In general, a dependency graph is a directed graph whose vertices represent the elements of a system and the branches represent the dependency relationships where dependencies point towards their dependents [61]. Vertices are components while the branches are the values linking two components in the dynamic phasor simulation program. The dependency graph approach requires that components simply have knowledge of the components they are dependent on to be able to initialize themselves. A routine then assembles the dependency information into a graph. As an example, consider a generator that includes an exciter, governor, and stabilizer connected to a bus with a step-up transformer. The dependency graph of such an arrangement is shown in Figure 3.3. Bus components do not have any dependents as their voltages are known from the power flow input data.

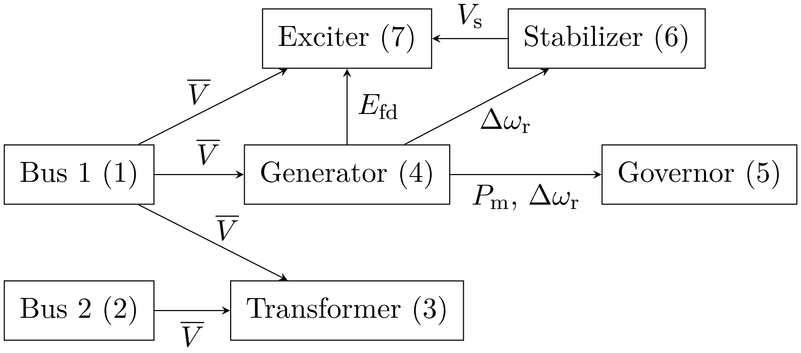

Figure 3.3: Dependency graph of a two bus system with a generator and step-up transformer including a valid initialization order

A topological ordering algorithm constructs an ordered list from the vertices of a directed acyclic graph. The ordered list is constructed such that if there is an edge pointing from vertex u to vertex v in the graph then u appears before v in the list [61]. For a dependency graph, the topological sort of a dependency graph produces a valid ordering where dependencies are always evaluated after their dependents. An example of a valid initialization order is included in Figure 3.3 where the number in parenthesis after a component name is it's position in the initialization order. A direct acyclic graph is guaranteed to have a topological ordering but it is not necessarily unique [61]. For example, a valid initialization order is still obtained if the positions of the transformer and generator are swapped.

An important aspect in the definition of topological sorting algorithms is that they operate on directed acyclic graphs. This implies another constraint on the dependency graph that cycles cannot exist to obtain a valid initialization order. A cycle would imply a circular dependency, which cannot be resolved in a linear ordering. Circular dependencies generally must be resolved by solving a set of equations, which was not implemented in the general system initialization routines. It was found that these situations did not occur frequently in models tested in this work. However, circular dependencies such as the one identified in the hydraulic governor model discussed in Section 4.3, must be resolved by collapsing components into a single model with only one initialization routine.

Each component's initialization routines are run using the order established by the topological sort of the initialization dependency graph. This step involves identifying a component's operating region, computing initial conditions for its variables and outputs, and validating the initial conditions. The initialization routines must choose the correct state based on any dependent components and verify that the operating mode is valid when components have multiple modes of operation. For example, a limiter component with a known output must verify that its output is within the modeled limits. The initial conditions must be adjusted to ensure the model can be properly initialized if any routine flags an error.

Models are initialized assuming that all of the derivatives are zero in transient stability simulations [1]. The model is in steady state when this condition is satisfied because of the demodulation effect of phasors. This property differs as compared to sinusoidal EMT models whose derivatives are generally not zero even though the system is in periodic steady state. The assumption that all derivatives are zero greatly simplifies the calculation of initial conditions. Many components such as branches and loads can use their differential equations to compute initial conditions with derivatives set to zero. However, additional data from power flow calculations is necessary for some components. For example, a synchronous machine initializes its stator current using its power output as calculated by the power flow routines.

The system's equations can be constructed once the initial conditions and operating points have been calculated and verified. The system's variables are first assigned an index, which is simply based on the associated component's position in a list. Ordering algorithms were not considered at this stage since the Implicit Differential-Algebraic (`IDA`) solver software package used to solve DAEs in this work includes linear solvers that use sophisticated ordering algorithms to maximize their performance [53]. The vectors **y** and **u** are populated by equation construction routines. The sparse matrices **A** and **T** are constructed and stored in compressed column format using an external data structure [62]. Furthermore, any preparations that can be done ahead of time to improve the performance of computing **f** and **J** are conducted at this stage. Finally, any static components such as loads and branches that do not contribute to **f** are discarded to reduce the size of the component list to strictly those that include dynamic calculations.

The final step in the initialization process is to refine the initial conditions calculated by the initialization routines. The values in **y** are refined using the system's differential equations to improve the initial value of **F**. This step is necessary because as previously mentioned some variables are initialized based on additional data and equations that are not their differential equations with derivatives set to zero. Furthermore, the power flow data computed by an external program might have a relatively large error tolerance compared to the numerical integration tolerances used by `IDA`. These two issues combined lead to a relatively large numerical error that accumulates in the rows of **F** that are associated with bus KCL equations. While the error is small by practical standards, the numerical solver reacts to them and tries to bring them within the more strict error tolerances of the numerical integration routines. `IDA` includes a routine called `IDACalcIC` that computes initial conditions according to the methods discussed in Section 2.3.2 [53]. The initial conditions are refined by calling this routine specifying the version that computes all values of **y** assuming $\dot{\mathbf{y}}$ is known.

3.2.2 Simulation

After the initialization procedures have been completed the system is prepared for the main simulation loop. `IDA` is a flexible general purpose solver so there are a number of parameters that must be set before simulation can begin. Some of these parameters are a result of more in depth investigation which will be discussed in subsequent sections. However, the type of sparse linear solver used in this program is based on observations from simulation trials. `IDA` is packaged with several direct and iterative sparse linear solvers used by the Newton process required in each step [53]. Trials were conducted using two direct sparse solvers packaged with `IDA`, Supernodal LU with multithreading (`SuperLU-MT`) and Clark Kent LU (`KLU`) [63], [64]. `KLU` is specifically designed to take advantage of properties that arise in circuit simulation [64], [65].

Tests indicate that KLU offered better performance over SuperLU-MT for the systems simulated in this work, which includes models with up to 15000 variables. Additional tests were conducted with some of the sparse iterative solvers and a incomplete LU preconditioner computed by Supernodal LU (SuperLU) [63]. Convergence was found to be extremely slow near disturbances using sparse methods. Therefore, KLU was selected as the default linear solver.

The first step of every loop is to record and store all of the variables associated with quantities of interest in the simulation. This includes bus voltages and machine states, but can also include combinations of variables. After this, the program passes all of the necessary state information to the solver routine IDASolve. This routine can return after each step of the solution or it can be used to emulate a fixed plot step using interpolation [53]. There is also a feature where IDA can be setup so that it does not pass beyond a known fixed time. This setting is useful for events with a known time such as faults. The dynamic phasor program was setup to use a combination of the fixed stopping point and the return each step setting so that the time step could be observed.

A step of the numerical integration method is initiated by calling IDASolve. This function will conduct either a single step or multiple steps depending on the desired output and return with a code indicating the status of the solution [53]. The flowchart in Figure 3.2 assumes a successful exit from this function, but it can return a variety of error codes if the step fails. In this case, the program simply quits after displaying the error code. If the solver indicates a successful step was taken, then the time variable is compared against the desired end time of the simulation. The simulation exits if the solver has stopped at a point in time beyond the end of the simulation. If the end time condition is not satisified, then the return code must be checked for an event flag. If an event did not occur, then the simulation loop ends and it returns to the start of the loop. On the other hand, if an event occurred then the appropriate actions must be taken to reset following the event.

IDA includes routines for detecting and locating events that is based on the framework discussed in Section 2.3.3 [53]. One of the successful exit flags of IDASolve indicates that an event occurred in the last step and that the solver stopped at the location of the event. The first step following an event is to record the desired variables at the point the event occurs. This step must be conducted prior to the event processing functions to capture the state immediately prior to the event. Once the event is resolved and states are reinitialized, the system will be in a state that represents the system immediately after the event. Therefore, recording before and after allows the simulation routine to capture jump discontinuities such as events.

The next step after recording the system's state is to resolve the events triggered in the last step. The solver also provides information as to which functions triggered the event. This information is used to identify which components triggered the event and to apply the appropriate actions. These actions might include changing a component's internal flags that might change how a specific equation calculates its contributions to $\mathbf{f}$. A component can also adjust values in $\mathbf{y}$ and $\dot{\mathbf{y}}$ that bring the solution close to the post-event solution. However, the reinitialization phase is primarily responsible for computing initial conditions for the new DAEs.

The final step in the event processing branch of the simulation loop is to reinitialize the DAE solution. The `IDACalcIC` routine is once again used for this purpose, but it is called with a flag indicating that it should solve the second problem type discussed in Section 2.3.2 [53]. This problem type assumes that the values of differential variables are known and that the solution seeks the derivatives of differential variables and the values of algebraic variables. At event boundaries, this problem type matches the available information since the DAE solution should not make instantaneous changes to differential variables to ensure all values in $\dot{\mathbf{y}}$ maintain physically consistent values. If the solver can find a consistent set of initial conditions, then the simulation returns to the top of the simulation loop to proceed from the event location with the adjusted system model and its corresponding solution.

3.3 Variables

Dynamic phasor and MNA-based circuit models produce a set of equations with variables that are both real and complex as well as differential and algebraic. These variables and equations can be combined into a single set of real DAEs but there are a few special considerations to ensure that the solver is setup properly. Section 3.3.1 will discuss the issues associated with combining real and complex variables into a single model particularly as it relates to error estimation. Section 3.3.2 will discuss how differential and algebraic variables are identified in a system's model, which is necessary for a number of features in IDA.

3.3.1 Real and Complex

A system's variables and equations are defined over the set of real numbers. This definition is necessary because IDA and many other general purpose numerical solvers are written to solve equations defined in terms of real numbers. Rotor and control equations are defined in terms of real-valued quantities so their variables and equations are directly included in $\mathbf{y}$ and $\mathbf{F}$. Complex variables are associated with dynamic phasors and a system's ac network equations. Complex variables and equations are included in $\mathbf{y}$ and $\mathbf{F}$ by splitting them into two real variables and equations using the same approach as other transient stability simulators [1]. However, all stamps are defined in terms of complex quantities as the procedure to split them can be handled automatically by an assembly program.

This split complex representation is necessary to simulate ac network equations using a real-valued solver like IDA. However, the weight of complex variables in local truncation error calculations must be adjusted to take this approach into account. Let $\mathbf{F}$ represent a generic n-dimensional system of equations and let e_i be the local truncation error for the ith variable at some step. Let ϵ_{r} and ϵ_{a} be the relative and absolute error tolerances, respectively. IDA computes the weight of the ith variable, w_i, according to

$$w_i = \frac{1}{\epsilon_{\text{r}} \left| y_i \right| + \epsilon_{\text{a}}} \tag{3.15}$$

by default if a user supplied function is not provided [53]. The weighted root mean square is computed using

$$\left\| \mathbf{e} \right\|_{\text{WRMS}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(w_i e_i \right)^2}, \tag{3.16}$$

which is used to determine convergence and future step sizes. Issues can arise with the split approach when a complex variable's angle is near either $k\pi$ or ${}^{(2k-1)\pi}/_{2}$ for any integer k. Without loss of generality, assume that there is a complex variable with nonzero magnitude and whose angle is close to zero. w_i for the imaginary part of this variable is dominated by ϵ_{a}, which is typically a small value such as 10^{-10}. The significant weight placed on this variable's imaginary part is not accurate considering the variable as a whole is nonzero. It was found that this inaccuracy can result in unnecessarily small time steps and convergence failures.

A solution is to use (3.15) only for real variables. The weight calculation is adjusted for complex variables to allow the error to account for their underlying connection. Let indices j and k be associated with a single complex variable, represented as two real variables. The weight for these variables is calculated according to

$$w_j = w_k = \frac{1}{\epsilon_{\mathrm{r}}\sqrt{y_j^2 + y_k^2} + \epsilon_{\mathrm{a}}}. \tag{3.17}$$

3.3.2 Differential and Algebraic

The equations produced by the MNA process with a dynamic phasor-based power system model produce both differential and algebraic variables. A system's complex network variables includes all bus voltages, which are predominantly differential due to the charging susceptance associated with transmission lines. Most practical models also include algebraic bus voltages, which are the result of buses without a capacitive connection to ground [56]. Inductive currents related to branches, loads, and stator windings are also differential network variables. Most bus voltages and branch currents are independent because transient stability models generally consist of capacitive buses separated by inductive currents.

Generators contribute the most significant number of real differential auxiliary variables through their rotor and control equations. Rotor variables include the flux in field and damper windings along with the rotor's angle and speed. Control variables depend on the type of exciter and governor used to model a generator. The MNA approach offers a great deal of flexibility with regards to variable selection because any number of variables can be included to simplify equations. However, the convenience of additional variables must be balanced with the hidden constraints and numerical difficulties that can be introduced by algebraic equations [35]. Additional variables in common models like generators can also have a negative impact on simulation performance because they increase the size of a set of DAEs. Therefore, differential state equations were used to model generators and their controls.

It is important to provide `IDA` with information on a variable's type for two reasons. First is that variable type is used for computation of initial conditions during the reinitialization phase as discussed in Section 3.2.2. Second is that IDA includes a setting that allows users to ignore algebraic variables in truncation error calculations [53]. The justification for this setting is that the response of a set of DAEs is typically dominated by the trajectory of its differential variables [35].

Both of these features assume that the user has knowledge regarding which variables are differential and which are algebraic. This information is available upon construction of (3.1) for auxiliary variables such as inductor currents and control variables. However, it is difficult to determine the membership of node voltages since their equations are influenced by many components. The membership of node voltages can be estimated using the diagonal of

$$\mathbf{J}_{\mathrm{d}} = \mathbf{T} + \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{y}}}. \tag{3.18}$$

It is assumed that a node voltage is a differential variable if its associated diagonal in $\mathbf{J}_{\mathrm{d}}$ is nonzero. This approach works well for the majority of system models, especially when repeated after any network configuration changes. However, it is possible that this approach fails to classify a node voltage properly if a component model includes a variable capacitance that can be zero momentarily depending on network conditions and control variables. This issue can be avoided with properly designed models that include thresholds to ensure variable capacitances are nonzero.

3.4 Integration Order Limit

IDA uses a variable order approach that selects a BDF method with order between one and five depending on the solution characteristics in the current and previous steps [53]. Figure 3.4 illustrates the regions of instability for the first two BDF methods and Figure 3.5 illustrates the regions for the third, fourth, and fifth methods [35]. The regions are depicted on a complex plane $z = h\lambda$ where h is the integration step size and λ is an eigenvalue of a system of DAEs. Numerical instability occurs for all values of z that fall within the boundaries of the regions.

Figure 3.4 illustrates that the regions of instability for the first two BDF methods are

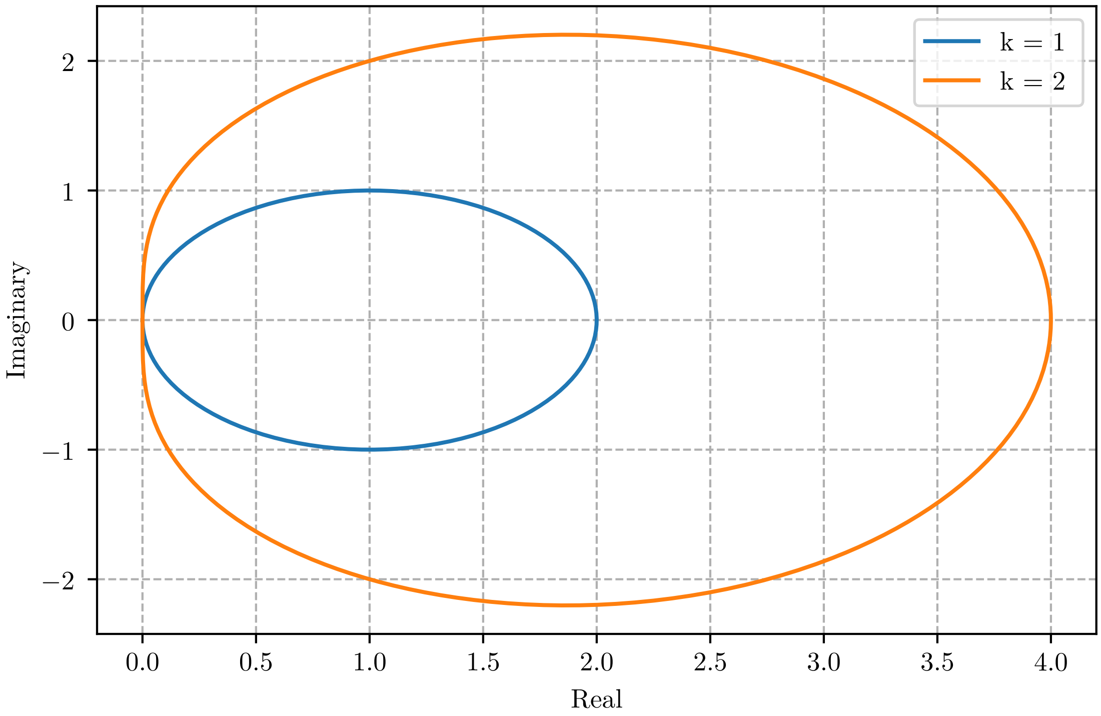

Figure 3.4: Regions of instability for the first two BDF methods

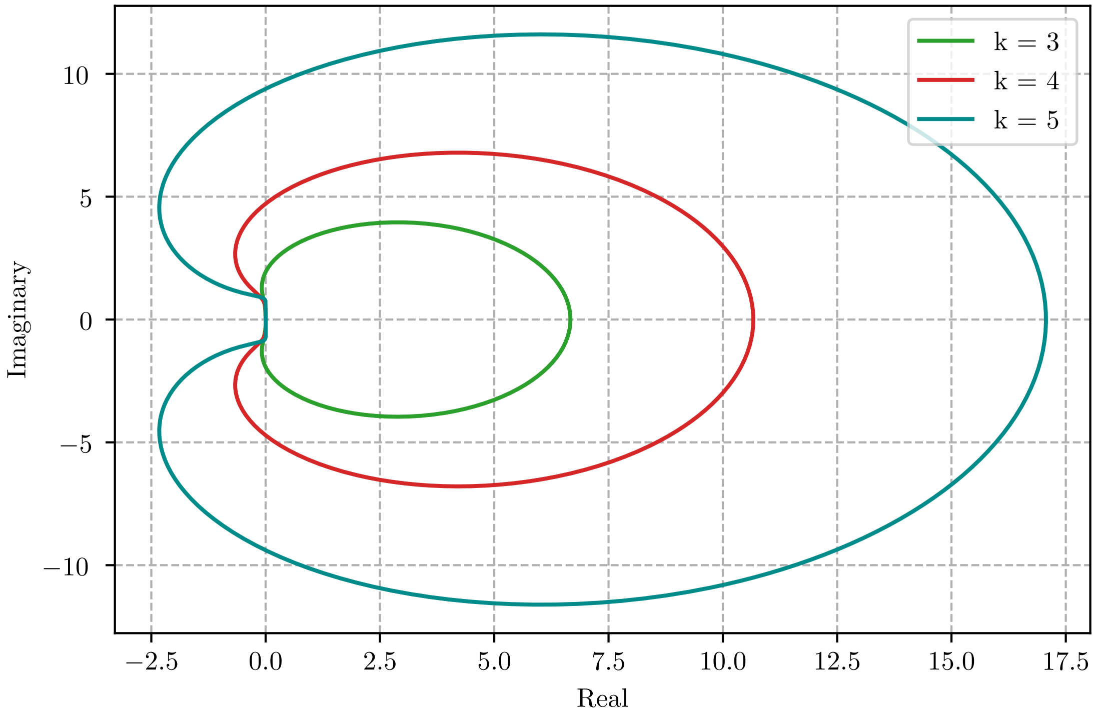

Figure 3.5: Regions of instability for the third, fourth, and fifth BDF methods

entirely contained on the positive half of the complex plane. It follows that these methods only experience numerical instability if the original DAE system includes unstable eigenvalues regardless of time step. Figure 3.5 illustrates that the region of instability for BDF methods of order three or greater includes parts of the negative half of the complex plane. These regions imply that the numerical solution for a DAE with only stable eigenvalues could be unstable depending on the integration step size. Furthermore, the regions also illustrate that the possibility of numerical instability depends on the damping of the eigenvalues.

Figure 3.6 illustrates the negative portion of the complex plane for the third BDF method. Lines are included that illustrate two values of eigenvalue damping. The line corresponding to 6.8 % damping shows that any eigenvalue with damping less than this value could fall within the unstable region depending on the integration step size. For example, the line corresponding to 5 % damping is within this region, which is a reasonable value for electromechanical modes in a power system model. Furthermore, the third BDF method is the least restrictive on eigenvalue damping as shown in Figure 3.5. It can be shown that eigenvalues integrated with the fourth and fifth BDF methods are restricted to a minimum damping of 28.6 % and 61.8 %, respectively.

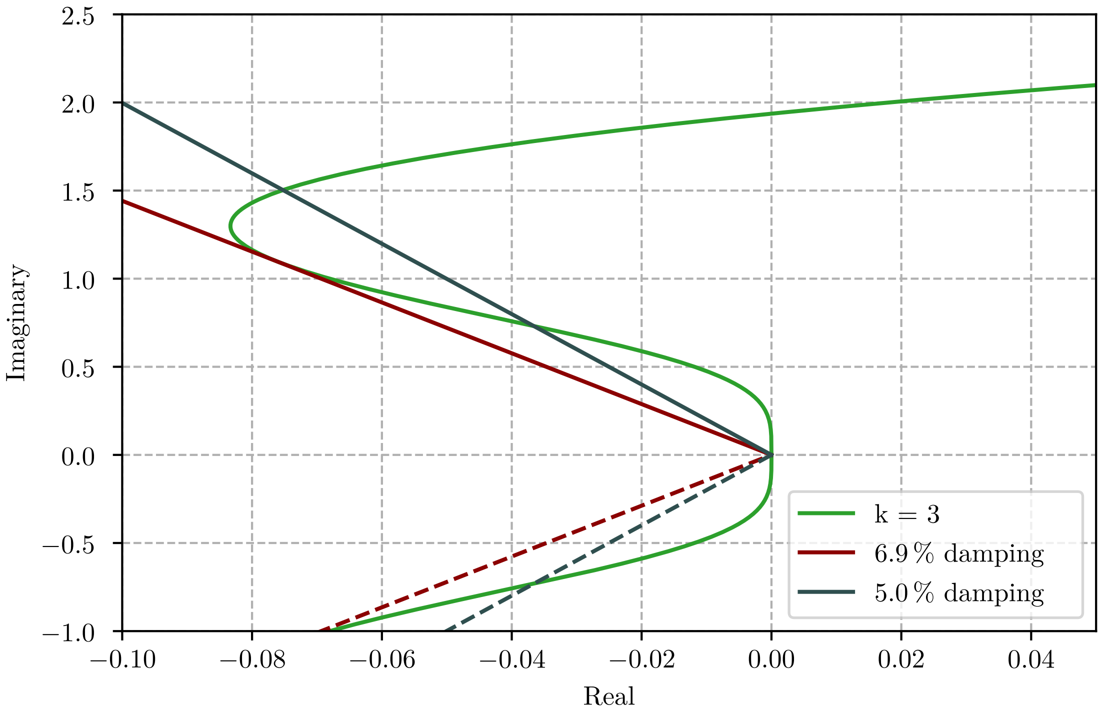

Figure 3.6: Region of instability for the third BDF method with two values of eigenvalue damping

There is a fixed mapping of a system's eigenvalues onto the z-plane if a fixed order and time step is used. However, the effect of the regions of instability is more complicated when a variable step and order is used. It was observed that an upper bound could be imposed on the integration time step by eigenvalues that are trapped by the region of instability. The simulation time step is small following disturbances so many eigenvalues are clustered neared the origin of the z-plane at this point. Eigenvalues move along their damping lines as the time step increases towards a region of instability. Numerical instability begins as eigenvalues enter these regions, which causes the local truncation error estimate to rise. The solver decreases its time step to decrease the local truncation error. This cycle repeats at various stages depending on the frequency and damping of the eigenvalues involved. The net effect of these cycles is a stable simulation, but the solver is forced to use a small time step for much longer than necessary to maintain numerical stability. Trials indicated that simulations required significantly more steps and CPU time when methods of order three or greater were enabled. Therefore, `IDA` was limited to a maximum order of two, which is consistent with other power system simulation packages that use the BDF method [50].

3.5 Floating Subnetworks

Consider a power system where all branches and loads are modeled using combinations of the component shown in Figure 3.1. Let G be a graph induced by the model's capacitors and let G_i be the ith connected component of G. G_i is said to be floating if there are no connections between the vertices of G_i and the ground vertex. A floating subnetwork is defined as the subset of buses, branches, and components induced by the vertices of a floating G_i. A grounded subnetwork is similarly defined, except induced by the vertices of a G_i that is not floating.

Floating subnetworks are a generalization of a wide variety of structural deficiencies that can result in poorly conditioned MNA equations. For example, generator models are frequently connected to the bulk electric system through a step-up transformer. Series connected inductive currents are the simplest example of an all-inductor cut-set, which results in a set of ill-conditioned MNA equations [66]. A generator terminal bus would be a single floating vertex in the graph G or a floating subnetwork of size one.

Assume that a circuit model has been created for a general power system and that G has been constructed for this circuit. Let $\mathbf{C}_i$ and $\mathbf{Y}_i$ be the capacitance and admittance matrices associated with G_i, respectively. Let $\mathbf{L}$ and $\mathbf{Z}$ be the block diagonal inductance and impedance matrices associated with inductive currents, respectively. Let $\mathbf{G}_{ij}$ be the conductance matrix associated with resistive branches connecting G_i to G_j. Finally, let $\mathbf{S}_i$ be the connection matrix relating G_i to the set of inductive currents, which includes the effects of transformer winding ratios. It is possible to show that the model's Jacobian can be written as

$$\mathbf{J} = \frac{1}{h\beta_k}\begin{bmatrix} \mathbf{C}_1 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{C}_n & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{L} \end{bmatrix} + \begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{G}_{1n} & \mathbf{S}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{G}_{n1} & \cdots & \mathbf{Y}_n & \mathbf{S}_n \\ \mathbf{S}_1^{\mathrm{T}} & \cdots & \mathbf{S}_n^{\mathrm{T}} & \mathbf{Z} \end{bmatrix}. \tag{3.19}$$

There must exist a tree induced by the capacitors in each G_i by definition. It can be shown that $\mathbf{C}_i$ is nonsingular if and only if there exists a path to ground through capacitors at each node in a circuit [66]. It follows that $\mathbf{C}_i$ is nonsingular for grounded subnetworks and singular for floating subnetworks. $\mathbf{J}$ is generally nonsingular for nonzero values of h because of matrices $\mathbf{Y}_i$, $\mathbf{G}_{ij}$, and $\mathbf{S}_i$. However, the first term of (3.19) dominates as h decreases. This term is a block diagonal matrix so the inverse of (3.19) can be approximated by

$$\mathbf{J}^{-1} \approx h\beta_k \begin{bmatrix} \mathbf{C}_1^{-1} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{C}_n^{-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{L}^{-1} \end{bmatrix} \tag{3.20}$$

for small time steps. The form illustrated by (3.20) shows that the approximate Jacobian is nonsingular if and only if there are no floating subnetworks present. The actual Jacobian is unlikely to be singular at small time steps if a floating subnetwork is present because of the second term in (3.19). However, a Jacobian will be ill-conditioned at small time steps when floating subnetworks are present, which can lead to convergence errors in the Newton iterations.

Floating subnetworks with algebraic voltages are an issue when reinitializing variables following disturbances. `IDA` does not ignore algebraic variables when computing initial conditions since it searches for consistent $\mathbf{y}$ and $\dot{\mathbf{y}}$ that satisfy (1.1) [67]. Floating subnetworks containing capacitors pose a more significant problem as their associated $\mathbf{C}_i$ is singular [68]. Floating capacitors tend to produce convergence failures during simulation as their nodes are marked as differential by the procedure discussed in Section 3.3.2.

Floating subnetworks can be eliminated by adding a small stray capacitance to ground at each node. An approximately equivalent action is to add a small value to each diagonal element t_{ii} of $\mathbf{T}$. It was found that even a small perturbation equal to $10^{-6} \times t_{ii} + 10^{-10}$ solves convergence failures associated with floating subnetworks.

Actual power system models include many synchronous machines, which are complex nonlinear devices. However, it will be shown in Section 4.2 that these devices interact with the network's equations through an inductive current. It follows that synchronous machines would not alter the subgraphs G_i induced by a network's capacitors. Furthermore, the effects of synchronous machines would be included in (3.19) in a similar manner as a linear inductor to ground so they do not affect any of the $\mathbf{C}_i$ matrices. Therefore, the issues relating to floating subnetworks are not affected by synchronous machines or any other similar inductive devices.

Chapter 4

Synchronous Machines

Synchronous machines are responsible for the majority of power generated throughout history [1]. Despite their continued displacement by wind, solar, and battery sources, they continue to represent a large share of power generated in modern power systems. Furthermore, their spinning masses provide a significant amount of inertia, which helps power systems maintain stability through disturbances. Therefore, synchronous machines will continue to represent an essential part of power systems and accurately modeling their behaviour will be essential to ensuring safe and reliable operation of power systems.

Synchronous machines are complex nonlinear components whose electrical and mechanical behaviour alone is modeled using eight differential equations as discussed in Section 4.1. Section 4.2 introduces the specifics of converting a generic synchronous machine model into one suitable for the dynamic phasor MNA simulator. Synchronous machines offer a great deal of control in ac power systems. Both their active and reactive power outputs can be controlled to maintain acceptable operating conditions. As a result, it is essential to model synchronous machine controls, which are discussed in Section 4.3. One of the major advantages of dynamic phasors is their ability to accurately model subsynchronous resonance [69]. The multimass turbine model is essential to capturing subsynchronous resonance, which is developed in Section 4.4. Finally, simulation results of two simple single machine cases are provided in Section 4.5 to validate the models developed in this chapter.

4.1 Background

Figure 4.1 illustrates a simplified physical model of the synchronous machine, displaying all of the information relevant to the model's equations. The stator's magnetic axis is aligned with the magnetic axis of the a-winding. The position of the rotor is measured relative to the magnetic axis of the stator's a-phase winding. The rotor's instantaneous angle is measured as the angular difference between the stator's a-phase winding and the quadrature or q-axis of the rotor. The direct or d-axis, which is aligned with the dc-fed magnetic field of the rotor, is defined as lagging the q-axis by 90 degrees.

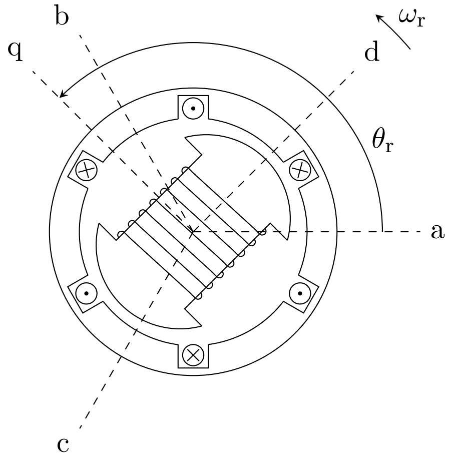

Figure 4.1: Simplified physical model of a synchronous machine

Figure 4.2 illustrates the equivalent circuit model for a synchronous machine's stator [1]. The voltage applied at the terminal, v_a, develops a current, i_a, in the stator circuit. The stator current induces a magnetic flux, ψ_a, in the stator's winding, which in turn induces a voltage behind the stator resistance, R_s. The stator flux includes both the mutual flux that also links the rotor's circuit and the flux lost to leakage, which is modeled by the leakage inductance, L_l. All quantities are defined in a similar way for other phases.

Figure 4.2 illustrates the stator's three-phase differential equations are

$$v_a = \dot{\psi}_a + R_s i_a, \tag{4.1}$$

$$v_b = \dot{\psi}_b + R_s i_b, \text{ and} \tag{4.2}$$

$$v_c = \dot{\psi}_c + R_s i_c. \tag{4.3}$$

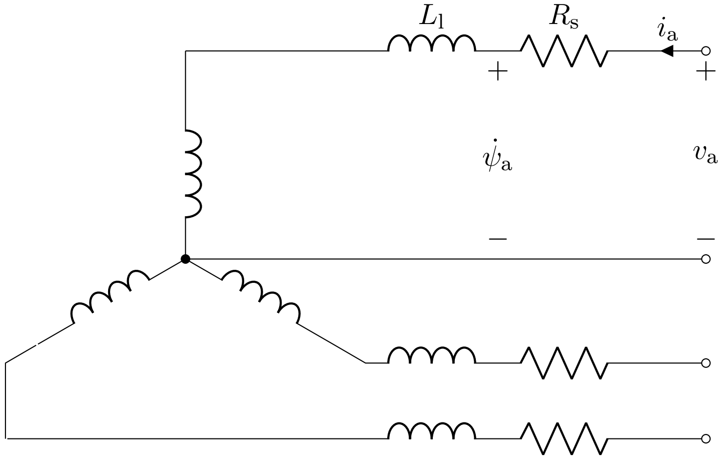

Figure 4.2: Three-phase electrical circuit model of a synchronous machine's stator

The mutual inductance linking the stator and rotor windings is time dependent since the rotor's windings physically move through the stator's fields [1]. The Park transformation is used to transform quantities from the stator's stationary reference frame to the rotor's rotational reference frame. This transformation is similar to the one used to define dynamic phasors in Section 2.1. Let x_d and x_q be direct and quadrature axis components corresponding to three-phase quantities x_a, x_b, and x_c. The three-phase quantities are referred to the rotational frame through the transformation

$$x_\mathrm{d} = x_\mathrm{a}\cos(\theta_\mathrm{r}) + x_\mathrm{b}\cos(\theta_\mathrm{r} - {}^{2\pi}\!/_{3}) + x_\mathrm{c}\cos(\theta_\mathrm{r} + {}^{2\pi}\!/_{3}) \text{ and} \tag{4.4}$$

$$x_\mathrm{q} = -\left(x_\mathrm{a}\sin(\theta_\mathrm{r}) + x_\mathrm{b}\sin(\theta_\mathrm{r} - {}^{2\pi}\!/_{3}) + x_\mathrm{c}\sin(\theta_\mathrm{r} + {}^{2\pi}\!/_{3})\right) \tag{4.5}$$

This transformation also includes a zero-sequence component so that it can be inverted. However, the zero-sequence component is always equal to zero in three-phase balanced systems so it was not included in the transformation's definition. The rotational components are referred to the stationary frame through the transformation

$$x_\mathrm{a} = x_\mathrm{d}\cos(\theta_\mathrm{r}) - x_\mathrm{q}\sin(\theta_\mathrm{r}), \tag{4.6}$$

$$x_\mathrm{b} = x_\mathrm{d}\cos(\theta_\mathrm{r} - {}^{2\pi}\!/_{3}) - x_\mathrm{q}\sin(\theta_r - {}^{2\pi}\!/_{3}), \text{ and} \tag{4.7}$$

$$x_\mathrm{c} = x_\mathrm{d}\cos(\theta_\mathrm{r} + {}^{2\pi}\!/_{3}) - x_\mathrm{q}\sin(\theta_\mathrm{r} + {}^{2\pi}\!/_{3}). \tag{4.8}$$

The derivatives of (4.6), (4.7), and (4.8) are

$$\dot{x}_{\mathrm{a}} = \dot{x}_{\mathrm{d}} \cos\left(\theta_{\mathrm{r}}\right) - \dot{x}_{\mathrm{q}} \sin\left(\theta_{\mathrm{r}}\right) - \dot{\theta}_{\mathrm{r}} \left(x_{\mathrm{d}} \sin\left(\theta_{\mathrm{r}}\right) + x_{\mathrm{q}} \cos\left(\theta_{\mathrm{r}}\right)\right), \tag{4.9}$$

$$\dot{x}_{\mathrm{b}} = \dot{x}_{\mathrm{d}} \cos\left(\theta_{\mathrm{r}} - {}^{2\pi}/_{3}\right) - \dot{x}_{\mathrm{q}} \sin\left(\theta_{\mathrm{r}} - {}^{2\pi}/_{3}\right) - \dot{\theta}_{\mathrm{r}} \left(x_{\mathrm{d}} \sin\left(\theta_{\mathrm{r}} - {}^{2\pi}/_{3}\right) + x_{\mathrm{q}} \cos\left(\theta_{\mathrm{r}} - {}^{2\pi}/_{3}\right)\right), \text{ and} \tag{4.10}$$

$$\dot{x}_{\mathrm{c}} = \dot{x}_{\mathrm{d}} \cos\left(\theta_{\mathrm{r}} + {}^{2\pi}/_{3}\right) - \dot{x}_{\mathrm{q}} \sin\left(\theta_{\mathrm{r}} + {}^{2\pi}/_{3}\right) - \dot{\theta}_{\mathrm{r}} \left(x_{\mathrm{d}} \sin\left(\theta_{\mathrm{r}} + {}^{2\pi}/_{3}\right) + x_{\mathrm{q}} \cos\left(\theta_{\mathrm{r}} + {}^{2\pi}/_{3}\right)\right), \tag{4.11}$$

respectively. The stator's differential equations in the rotating reference frame

$$v_{\mathrm{d}} = \dot{\psi}_{\mathrm{d}} - \dot{\theta}_{\mathrm{r}} \psi_{\mathrm{q}} + R_{\mathrm{s}} i_{\mathrm{d}} \text{ and} \tag{4.12}$$

$$v_{\mathrm{q}} = \dot{\psi}_{\mathrm{q}} + \dot{\theta}_{\mathrm{r}} \psi_{\mathrm{d}} + R_{\mathrm{s}} i_{\mathrm{q}} \tag{4.13}$$

are derived by applying the Park transformation to (4.1), (4.2), and (4.3).

Figures 4.3 and 4.4 illustrates the rotor's equivalent direct and quadrature axis circuit models, respectively [1]. A voltage is induced on the inputs to these circuits by the mutual flux ψ_{ad} and ψ_{aq}, which is the flux linking both the stator and rotor circuits. The mutual inductance linking both the stator and rotor is modeled by L_{ad} and L_{aq}.

The direct axis circuit includes two branches, one modeling the field winding and another to model the effects of damper windings. These two branches have a similar structure except that the field branch includes a dc source, e_{fd}, to model the exciter. The value of e_{fd} is determined by an exciter model, which will be discussed in Section 4.3. The current i_{fd} developed in the rotor circuit induces a magnetic flux ψ_{fd} in the field winding, which in turn develops a voltage behind the field winding resistance R_{fd}. The portion of ψ_{fd} that does not link the stator winding is modeled using the field winding's leakage inductance, L_{fd}. The damper winding quantities are defined in a similar manner except that they do not include a voltage source. The current and flux in the damper winding are induced by variations in the rotor's speed, which implies that $i_{1\mathrm{d}}$ is zero in steady state [1].

As illustrated in Figure 4.1, the field winding is aligned with the direct axis so the quadrature axis does not include a field winding component. The quadrature axis circuit includes two circuits each modeling damper windings. This arrangement is used to model round rotor machines while a salient pole machine is modeled using only one damper circuit, with quantities usually denoted by the 2q subscript [1].

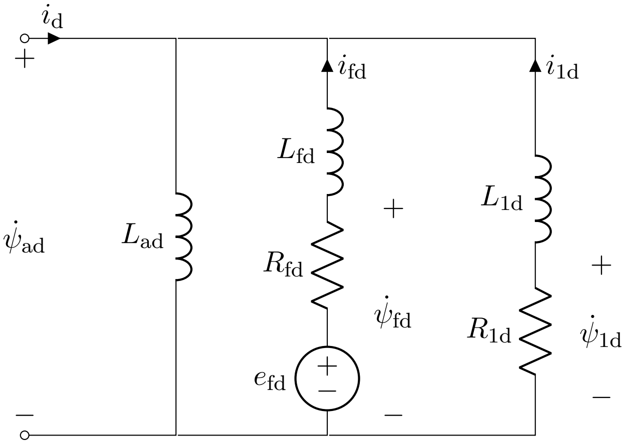

Figure 4.3: Equivalent rotor d-axis circuit model

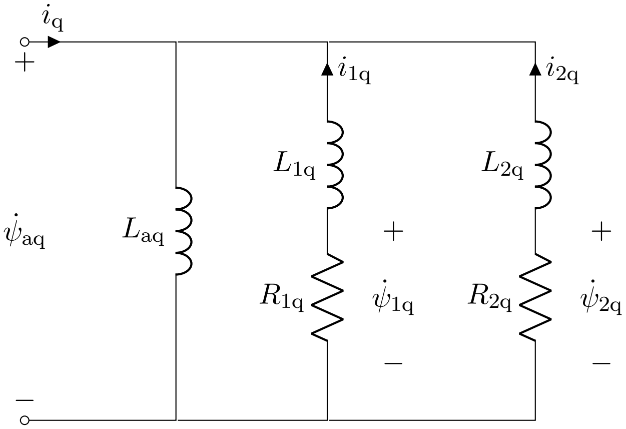

Figure 4.4: Equivalent rotor q-axis circuit model

The direct axis equivalent circuit has a differential equation associated with the voltage balance for each winding, which are

$$\dot{\psi}_{\mathrm{fd}} + R_{\mathrm{fd}} i_{\mathrm{fd}} - e_{\mathrm{fd}} = 0 \text{ and} \tag{4.14}$$

$$\dot{\psi}_{\mathrm{1d}} + R_{\mathrm{1d}} i_{\mathrm{1d}} = 0. \tag{4.15}$$

Similarly, the quadrature axis differential equations are

$$\dot{\psi}_{\mathrm{1q}} + R_{\mathrm{1q}} i_{\mathrm{1q}} = 0 \text{ and} \tag{4.16}$$

$$\dot{\psi}_{\mathrm{2q}} + R_{\mathrm{2q}} i_{\mathrm{2q}} = 0. \tag{4.17}$$

These equations are sufficient for modeling the electrical dynamics of a machine's rotor in transient stability studies. They also demonstrate that a round rotor model selects the four flux variables ψ_{fd}, ψ_{1d}, ψ_{1q}, and ψ_{2q} as differential variables. However, an alternative set of equations replaces damper flux variables ψ_{1d} and ψ_{1q} with their corresponding mutual flux variables ψ_{ad} and ψ_{aq} [70]. This approach was selected because mutual flux variables are better suited to the stator's equations as will be illustrated in Section 4.2. The relationship between magnetizing current and mutual flux for the direct and quadrature axes is

$$\frac{\psi_{\mathrm{ad}}}{L_{\mathrm{ad}}} = i_{\mathrm{d}} + i_{\mathrm{fd}} + i_{\mathrm{1d}} \text{ and} \tag{4.18}$$

$$\frac{\psi_{\mathrm{aq}}}{L_{\mathrm{aq}}} = i_{\mathrm{q}} + i_{\mathrm{1q}} + i_{\mathrm{2q}}, \tag{4.19}$$

respectively. The rotor currents must be eliminated from these expressions as they are not differential variables. This is accomplished using the relationship between winding flux variables and the mutual flux implied by the rotor's equivalent circuits. (4.18) and (4.19) in terms of magnetic flux is

$$\frac{\psi_{\mathrm{ad}}}{L_{\mathrm{ad}}} = i_{\mathrm{d}} + \frac{\psi_{\mathrm{fd}} - \psi_{\mathrm{ad}}}{L_{\mathrm{fd}}} + \frac{\psi_{\mathrm{1d}} - \psi_{\mathrm{ad}}}{L_{\mathrm{1d}}} \text{ and} \tag{4.20}$$

$$\frac{\psi_{\mathrm{aq}}}{L_{\mathrm{aq}}} = i_{\mathrm{q}} + \frac{\psi_{\mathrm{1q}} - \psi_{\mathrm{ad}}}{L_{\mathrm{1q}}} + \frac{\psi_{\mathrm{2q}} - \psi_{\mathrm{ad}}}{L_{\mathrm{2q}}}, \tag{4.21}$$

Differential equations for the direct and quadrature mutual flux

$$\left(\frac{1}{L_{\mathrm{ad}}} + \frac{1}{L_{\mathrm{fd}}} + \frac{1}{L_{\mathrm{1d}}}\right) \dot{\psi}_{\mathrm{ad}} = \dot{i}_{\mathrm{d}} + \frac{\dot{\psi}_{\mathrm{fd}}}{L_{\mathrm{fd}}} + \frac{\dot{\psi}_{\mathrm{1d}}}{L_{\mathrm{1d}}} \text{ and} \tag{4.22}$$

$$\left(\frac{1}{L_{\mathrm{aq}}} + \frac{1}{L_{\mathrm{1q}}} + \frac{1}{L_{\mathrm{2q}}}\right) \dot{\psi}_{\mathrm{aq}} = \dot{i}_{\mathrm{q}} + \frac{\dot{\psi}_{\mathrm{1q}}}{L_{\mathrm{1q}}} + \frac{\dot{\psi}_{\mathrm{2q}}}{L_{\mathrm{2q}}}. \tag{4.23}$$

are constructed by differentiating (4.20) and (4.21) with respect to time. Differential equations that are suitable for the rotor model

$$\left(\frac{1}{L_{\mathrm{ad}}}+\frac{1}{L_{\mathrm{fd}}}+\frac{1}{L_{\mathrm{1d}}}\right)\dot{\psi}_{\mathrm{ad}}-\dot{i}_{\mathrm{d}}-\frac{\dot{\psi}_{\mathrm{fd}}}{L_{\mathrm{fd}}}+\frac{R_{\mathrm{1d}}}{L_{\mathrm{1d}}}\left(\left(\frac{1}{L_{\mathrm{ad}}}+\frac{1}{L_{\mathrm{fd}}}\right)\psi_{\mathrm{ad}}-i_{\mathrm{d}}-\frac{\psi_{\mathrm{fd}}}{L_{\mathrm{fd}}}\right)=0 \text{ and} \tag{4.24}$$

$$\left(\frac{1}{L_{\mathrm{aq}}}+\frac{1}{L_{\mathrm{1q}}}+\frac{1}{L_{\mathrm{2q}}}\right)\dot{\psi}_{\mathrm{aq}}-\dot{i}_{\mathrm{q}}-\frac{\dot{\psi}_{\mathrm{2q}}}{L_{\mathrm{2q}}}+\frac{R_{\mathrm{1q}}}{L_{\mathrm{1q}}}\left(\left(\frac{1}{L_{\mathrm{aq}}}+\frac{1}{L_{\mathrm{2q}}}\right)\psi_{\mathrm{aq}}-i_{\mathrm{q}}-\frac{\psi_{\mathrm{2q}}}{L_{\mathrm{2q}}}\right)=0. \tag{4.25}$$

are obtained by eliminating $\dot{\psi}_{\mathrm{1d}}$ and $\dot{\psi}_{\mathrm{1d}}$ from (4.22) and (4.23) using the equivalent circuits and (4.18) and (4.19). This rotor circuit model neglects saturation because it is difficult to obtain a good match when saturation is included. The two major simulation programs PSSE and PSCAD/EMTDC use different approaches to saturation, so it introduces additional areas for disagreement between simulation programs that is related to design and modeling decisions rather than an inherent inconsistency in the model. However, it is trivial to include saturation in (4.24) and (4.25) [70]. The rotor is assumed to be linear and saturation-free in (4.18) and (4.19) through the use of linear inductances L_{ad} and L_{aq}. Instead of linear inductances, the relationship between the total magnetization current and mutual flux can be replaced by a nonlinear function representing the machine's saturation curve. The only restriction on the saturation curve is that it must be differentiable as implied by the actions leading to (4.22) and (4.23).

The final piece of the synchronous machine model is a set of equations modeling the physical motion of the rotor. The rotor's motion can be modeled using the rotational equations of motion since the rotor is a mass fixed in space and rotating about a single axis [1]. Let H be the machine's inertia constant and let P_{m} be the mechanical power applied to the shaft by either a load in the case of a motor or a prime mover in the case of a generator. The mechanical power is positive for loads and negative for prime movers since the motor convention was used in this synchronous machine model. The value of P_{m} is determined by any governor control systems attached to the generator, which will be discussed in Section 4.3. Let T_{e} be the electromagnetic torque, which represents the torque that is developed on the rotor by the stator. It can be expressed in terms of the stator current and mutual flux,

$$T_{\mathrm{e}}=\psi_{\mathrm{ad}}i_{\mathrm{q}}-\psi_{\mathrm{aq}}i_{\mathrm{d}}. \tag{4.26}$$

The swing equation is a second order differential equation, which can be expressed in terms of two first order differential equations

$$\dot{\theta}_{\mathrm{r}} - \omega_{\mathrm{r}} = 0 \text{ and} \tag{4.27}$$

$$2H\dot{\omega}_{\mathrm{r}} - T_{\mathrm{e}} + \frac{P_{\mathrm{m}}}{\omega_{\mathrm{r}}} = 0. \tag{4.28}$$

However, ω_{r} is equal to the system's base frequency ω_{r} in steady state. Therefore, (4.27) implies that θ_{r} is not stationary in steady state. The rotor angle deviation, δ_{r}, is defined as difference between the rotor angle and the system's reference angle,

$$\delta_{\mathrm{r}} = \theta_{\mathrm{r}} - \omega_0 t. \tag{4.29}$$

The rotor speed deviation, $\Delta\omega_{\mathrm{r}}$, is found by differentiating (4.29) and substituting in (4.27),

$$\Delta\omega_{\mathrm{r}} = \omega_{\mathrm{r}} - \omega_0. \tag{4.30}$$

Let D be the rotor's speed damping coefficient, which produces a torque proportional to the rotor's speed devation [1]. Substituting (4.29) and (4.30) into the mechanical differential equations (4.27) and (4.28),

$$\dot{\delta}_{\mathrm{r}} - \Delta\omega_{\mathrm{r}} = 0 \text{ and} \tag{4.31}$$

$$2H\dot{\Delta}\omega_{\mathrm{r}} + D\Delta\omega_{\mathrm{r}} + T_{\mathrm{e}} - \frac{P_{\mathrm{m}}}{1 + \Delta\omega_{\mathrm{r}}} = 0, \tag{4.32}$$

which are suitable for a transient stability simulation model.

Some of the above equations are in a final form suitable for the synchronous machine model, and others much be adapted for a multi machine dynamic phasor context. The following section will formalize the dynamic phasor model constructed for this work.

4.2 Dynamic Phasor Model

The derivative components of the stator flux, $\dot{\psi}_{\mathrm{d}}$ and $\dot{\psi}_{\mathrm{q}}$, are neglected in conventional transient stability simulation programs [1]. (4.12) and (4.13) can be transformed into a Norton equivalent model using this assumption, which is compatible with the admittance matrix approach to constructing network equations. This approach cannot be used in a dynamic phasor context since $\dot{\psi}_{\mathrm{d}}$ and $\dot{\psi}_{\mathrm{q}}$ cannot be neglected. An early dynamic phasor-based synchronous machine model incorporates stator dynamics by adding a derivative component to a machine's equivalent subtransient voltage source [71]. Another model for shifted frequency analysis uses the voltage behind reactance approach, which is also based on the subtransient source concept [72], [73]. An alternative general synchronous machine stator model is presented in this work that is not dependent on equivalent sources since a machine's nonlinear equations can be included in (3.1). Machines appear as nonlinear inductances from the perspective of electrical networks, which is an important observation when diagnosing and analyzing numerical failures in a system's solution as discussed in Section 3.5. The remainder of the machine model, which includes electrical and mechanical portions of the rotor and any associated controls, is based on well-known models used in both EMT and conventional transient stability programs [1].

The stator equations are defined in terms of the network's frame of reference for the model presented in this work. Stator differential equations (4.12) and (4.13) are defined in terms of the rotor's frame of reference, which can be rotated to the network's frame of reference. The network's frame of reference is defined to be one that rotates at the base frequency and has no initial phase shift. This definition implies that a cosine wave oscillating at the base frequency with no phase shift has a phasor that is aligned with the real axis of the network's frame of reference. Network phasors such as bus voltages and line current are all implicitly defined in the network's frame of reference.

Figure 4.5 illustrates the relationship between the rotor and network frames of reference in terms of the rotor angle and speed deviations. The rotor's frame of reference rotates at the rotor's angular velocity, ω_{r}. With the network's frame of reference fixed, the rotor's frame of reference appears to rotate at the rotor speed devation, $\Delta\omega_{\mathrm{r}}$, from the network's perspective. A phasor quantity $\overline{X}$ is included in this illustration. This phasor can be resolved into its real and

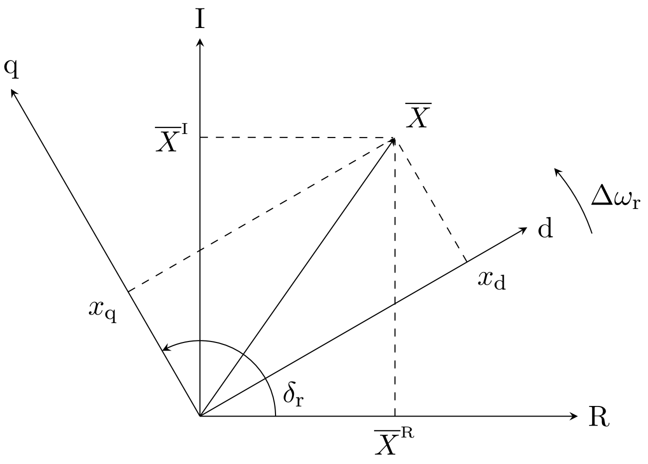
I
q
$\overline{X}$
$\overline{X}^{\mathrm{I}}$
$\Delta\omega_{\mathrm{r}}$
d
x_{d}
x_{q}
δ_{r}
R
$\overline{X}^{\mathrm{R}}$

Figure 4.5: Relationship between rotor and network frames of reference

imaginary components in the network's frame of reference. It can also be resolved into direct, x_{d}, and quadrature, x_{q}, components in the rotor's frame of reference.

Figure 4.5 illustrates that vectors defined in the network's frame of reference may be transferred to the rotor's frame of reference through a clockwise $\delta_{\mathrm{r}} - \pi/2$ rotation. In terms of complex arithmetic this is equivalent to

$$x_{\mathrm{d}} + jx_{\mathrm{q}} = je^{-j\delta_{\mathrm{r}}}\overline{X}. \tag{4.33}$$

The derivative of (4.33)

$$\dot{x}_{\mathrm{d}} + j\dot{x}_{\mathrm{q}} = je^{-j\delta_{\mathrm{r}}}\left(\dot{\overline{X}} - j\Delta\omega_{\mathrm{r}}\overline{X}\right) \tag{4.34}$$

is necessary for the rotor's mutual flux equations (4.24) and (4.25). Inverting the transformation (4.33) simply involves a rotation in the opposite direction,

$$\overline{X} = -je^{j\delta_{\mathrm{r}}}\left(x_{\mathrm{d}} + jx_{\mathrm{q}}\right). \tag{4.35}$$

The derivative of (4.35)

$$\dot{\overline{X}} = -je^{j\delta_{\mathrm{r}}}\left(\dot{x}_{\mathrm{d}} + j\dot{x}_{\mathrm{q}} + j\Delta\omega_{\mathrm{r}}\left(x_d + jx_q\right)\right) \tag{4.36}$$

is necessary for the stator's differential equations in the network's frame of reference. The transformations (4.33) and (4.34) are used to compute components for the rotor's equations that are defined using the network's variables. Similarly, transformations (4.35) and (4.36) are used to compute components for the stator's equations that are defined using the rotor's variables.

The stator's differential equations are transformed to the network's frame of reference by first combining (4.12) and (4.13) into a single expression

$$\dot{\psi}_{\mathrm{d}} + j\dot{\psi}_{\mathrm{q}} + j\omega_{\mathrm{r}}\left(\psi_{\mathrm{d}} + j\psi_{\mathrm{q}}\right) + R_{\mathrm{s}}\left(i_{\mathrm{d}} + ji_{\mathrm{q}}\right) - \left(v_{\mathrm{d}} + jv_{\mathrm{q}}\right) = 0. \tag{4.37}$$

Applying the rotation (4.35) to (4.37) and simplifying using the identity (4.36) gives the stator's differential equation in terms of network quantities,

$$\dot{\overline{\psi}} + j\overline{\psi} + R_{\mathrm{s}}\overline{I} - \overline{V} = 0. \tag{4.38}$$

The first two terms involving the stator's flux linkage are equal to an inductive voltage drop in dynamic phasors. It follows that (4.40) models an inductive element behind the armature resistance. Therefore, the stator's dynamic phasor equivalent circuit is the single-phase phasor equivalent of the three-phase model shown in Figure 4.2. Furthermore, synchronous machines are a shunt inductive branch, which means that they pose numerical conditioning issues as discussed in Section 3.5.

The stator differential equation (4.38) includes two variables, the stator's current and magnetic flux, one of which must be eliminated since the stator's constraint equation can only introduce a single additional variable. The stator's current and magnetic flux are related through the leakage inductance,

$$\overline{\psi} = \overline{\psi}_{\mathrm{a}} + L_{\mathrm{l}}\overline{I}. \tag{4.39}$$

The mutual flux in the network's frame of reference, $\overline{\psi}_{\mathrm{a}}$, is a nonlinear combination of rotor variables ψ_{ad}, ψ_{aq}, and δ_{r}. Consequently, the stator's differential equation will be nonlinear regardless of the choice of the stator's differential variable. Therefore, the best choice of stator variable is the current since it is required for the terminal's KCL equation. Selecting the stator current means that the machine's contribution to its terminal's KCL equation is a single differential variable, which can be trivially added to the KCL equation using matrix **A**. On the other hand, if the stator flux is selected, both the stator differential equation and the terminal's KCL equation require a nonlinear equation that must be evaluated each time step. While there are options to mitigate the impact of function calls, whenever possible they should be minimized to maximize simulation performance. The final form of the stator's differential equation is derived by substituting (4.39) into (4.38),

$$L_\mathrm{l}\dot{\overline{I}} + \left(R_\mathrm{s} + jL_\mathrm{l}\right)\overline{I} + \dot{\overline{\psi}}_\mathrm{a} + j\overline{\psi}_\mathrm{a} - \overline{V} = 0. \tag{4.40}$$

A machine uses **A** to introduce its contribution to its terminal's KCL equation with the stator current dynamic phasor selected as the stator's differential variable. The coefficient of $\overline{I}$ in **A** must also include a scaling factor, which reflects the difference between the machine's base MVA and the system's base MVA. The machine's base power is typically different than the system's base power and all of the machine's parameters and variables are defined on the machine's base.

The field voltage and mechanical power are both inputs to the machine. If an exciter (governor) is not included in the machine's model, then the field voltage (mechanical power) is constant. However, these devices can have a significant effect on stability and must be modelled for accurate studies [1]. The following section will discuss generator control modelling.

4.3 Controls

Generator control modelling is critical for properly evaluating the transient performance of a power system [1]. There are standard control models that attempt to provide relatively accurate representations of any manufacturer's design using generic parameters to tune their response. For example, the ac type four exciter model is a simple generic model for representing ac rotating excitation systems [74]. However, generic models do not provide sufficient accuracy in some cases and custom models must be developed that accurately represent their corresponding control device. Consequently, implementing control models is a common task that is the responsibility of many different users of a simulation program.

It is possible to express the control block diagrams of these components in terms of unique differential-algebraic equations for each model. However, developing dynamic equations directly can be a time consuming task. It can be particularly onerous if a model contains numerous state changing components, such as non-windup limits, as each combination of states must be considered to determine the overall effect on a model's equations. Therefore, a general approach was implemented in the DP simulation program to quickly model controllers and share code across many different models.

Generator and other control systems are generally specified using block diagrams. Many professional simulation programs with graphical user interfaces provide tools to implement controllers directly using those block diagrams. Control blocks were also implemented in the DP simulation program. Each individual block is a model with the same responsibilities as components such as the synchronous generator. They must define and initialize their variables and also construct their equations. However, control blocks do not interact with the network's equations so they do not specify currents for KCL equations. Instead, each control block must accept inputs and label those as its dependencies, and define outputs for use by other blocks.

As an example, consider the single pole component with non-windup limits shown in Figure 4.6 [1]. This component is found in many standard exciter and governor models [74], [75]. The output of this component follows the input according to the block's dynamics when it falls between $L_{\min}$ and $L_{\max}$. When the block's output exceeds its limits, the output is clamped by the limits and it ceases to respond to the input except to recover from the associated limit. This behaviour allows the block to respond to new inputs as soon as the input tries to force the block back in between the output limits.

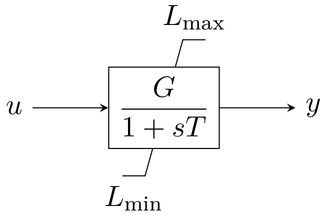

Figure 4.6: Single pole block with non-windup limits

The single pole illustrated in Figure 4.6 has three states, which are normal operation, at its minimum limit, and at its maximum limit. The pole has a single differential variable, y, whose associated constraint equation is

$$T\dot{y} + y - Gu = 0 \tag{4.41}$$

when operating in its normal state. The limits for this block are expressed in terms of two inequalities, which are used by `IDA`'s root finding routines to locate state transitions. The first detects transition from the normal state to the maximum limit state,

$$y - L_{\max} < 0. \tag{4.42}$$

The inequality is satisfied when the left hand side is negative, or $y < L_{\max}$. Therefore, the root equation associated with (4.42) must be setup to trigger on positive zero crossings. Similarly, the second inequality detects transitions from the normal state to the minimum limit state,

$$y - L_{\min} > 0. \tag{4.43}$$

The root equation associated with (4.43) must be setup to trigger on negative zero crossings since the inequality is satisfied when the left hand side is positive.

The single pole component must adapt its constraint equation when it transitions to either of the limited states. The desired behaviour is to lock the output to the limit that was triggered and prevent the block from further responding in the same direction as the limit. A simple way to achieve this effect is to prevent y from changing by fixing its derivative to zero,

$$T\dot{y} = 0. \tag{4.44}$$

When the event is resolved, the value of y is set to the triggered limit and (4.44) ensures it does not change as the simulation proceeds. An alternative choice is to use the constraint equation to fix y to the triggered limit. However, (4.44) includes a slight numerical advantage in the presence of small time steps since it maintains a diagonal element in $\mathbf{T}$.

A single root equation is required for the single pole block to transition from either limited state to the normal state. This equation must detect when the input attempts to force the block's output back into the normal operating range. In other words, the root equation must detect when the input implies negative values of $\dot{y}$ in (4.41) while the block is in the maximum limit state and positive values while the block is in the minimum limit state. Numerical hysteresis must be added to the limits to ensure that the block is firmly moving into the normal operating range and prevent the block from rapidly swapping between states. Let L'_{max} be the maximum limit perturbed by two small tolerances, ϵ_r and ϵ_a, according to

$$L'_{\text{max}} = L_{\text{max}} - \left(\epsilon_{\text{r}} \left|L_{\text{max}}\right| + \epsilon_{\text{a}}\right). \tag{4.45}$$

Possible values for ϵ_{r} and ϵ_{a} are 1×10^{-4} and 1×10^{-10}, respectively. (4.45) implies that L_{max} is slightly larger than L'_{max}, which is the desired outcome to ensure that the input is pushing the block in the correct direction. The inequality used to detect a transition from the maximum limited state to the normal state is

$$Gu - L'_{\text{max}} > 0. \tag{4.46}$$

The left-hand side is equal to $T\dot{y}$ as defined by (4.41) when $y = L'_{\text{max}}$. The root equation associated with (4.46) must be setup to trigger on negative zero crossings. Similarly, let L'_{min} be the minimum limit perturbed by ϵ_r and ϵ_a according to

$$L'_{\text{min}} = L_{\text{min}} + \left(\epsilon_r \left|L_{\text{min}}\right| + \epsilon_a\right). \tag{4.47}$$

Assembly of blocks into a controller is accomplished by a composite model, which defines the interface and data required for a specific control model such as the ac type four exciter. All of the control blocks required by the composite model are constructed as the first step. Then the composite model connects blocks together by calling the corresponding methods on its constituent blocks. Connections between blocks must be made for standard run-time operation of the model, which is identical to the controller's block diagram. Another set of connections between blocks must be made to allow the block models to identify an initialization order. The flow of information in the initialization graph is generally different from the block diagram. Furthermore, the control blocks are initialized by the simulator's initializer, which implies that the control system's initialization connections must be acyclic.

Figure 4.7 illustrates the type four standard ac exciter model [74]. The input voltage, $\left|\overline{V}\right|$, is measured using the bus's voltage dynamic phasor and the output of the controller is fed to its associated synchronous machine. For normal run-time operation, the blocks are connected exactly as shown in Figure 4.7. At initialization, the bus voltage is known from the power flow data and the field voltage is computed by the synchronous machine. Consequently, the exciter's initialization routines are dependent on the synchronous machine and its associated terminal bus. The transducer initializes its output using the bus's voltage. The output limiter ensures the machine's field voltage is between its maximum and minimum values and flags a value if it is not. If it is, then the output amplifier initializes its input with the field voltage, which the lead-lag block uses to initialize its input. The input limiter performs the same check as the output limiter. If the voltage is valid, the transducer output and the lead-lag input are used to initialize the reference voltage, which is a constant. Therefore, the type four ac exciter is an example of a simple model that does not require any additional support outside of some common control block models.

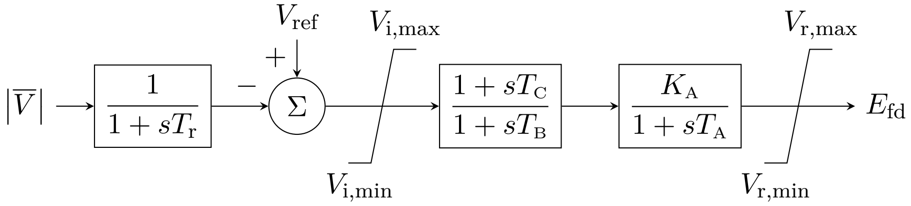

Figure 4.7: Type four ac exciter model control block diagram

There are some cases where the generic block system fails to produce a model that can be initialized. Figure 4.8 illustrates the hydraulic governor and turbine model [75]. The turbine side of the model represents a control system that cannot be initialized using the generic block system. The turbine subsystem must initialize the governor's output since the reference component is part of the governor. This requirement implies that there is a cycle in the turbine's initialization graph without the special information that $\Delta\omega_\mathrm{r}$ is zero in steady state. The turbine can be modeled using a single custom block in this situation to resolve the circular dependency.

A collapsed model of the hydraulic turbine requires only one differential variable, which is the output of its integrator, q. The input to the turbine is the governor's output, y_g, and the output is the mechanical power, P_m. The differential constraint equation associated with q is

$$T_\mathrm{w}\dot{q} + \left(\frac{q}{y_\mathrm{g}}\right)^2 - 1 = 0 \tag{4.48}$$

and its output is computed by

$$P_{\mathrm{m}} = A_{\mathrm{t}} \left(\frac{q}{y_{\mathrm{g}}} \right)^2 (q - q_{\mathrm{nl}}) - D_{\mathrm{t}} \Delta\omega_{\mathrm{r}} y_{\mathrm{g}}. \tag{4.49}$$

The output equation is implicitly used by the synchronous machine in (4.32) and not explicilty modeled as a constraint equation. These two equations are the only components required to model the hydraulic turbine's block. It does not require any state changes or event processing.

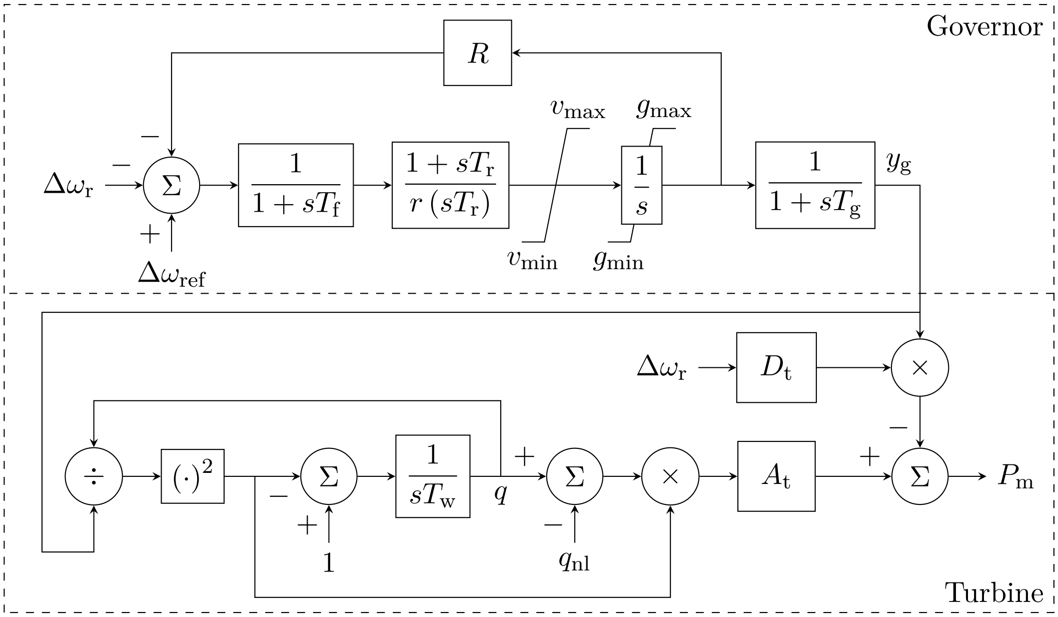

Figure 4.8: Hydraulic governor and turbine model control block diagram

Custom control blocks might be preferred over generic blocks even if a valid model can be produced using the generic approach. In particular, custom blocks are preferred if there is a large number of nonlinear algebraic blocks with few state changing components. The turbine model illustrated in Figure 4.8 is a good example of this situation. There is only one differential block while there are nine algebraic blocks, about half of which model nonlinear relationships. Blocks modeling nonlinear algebraic relationships greatly increase the number of function accesses per step, especially when they are included in a common model such as the hydraulic turbine. The custom block component on the other hand requires only one variable and one implicit nonlinear calculation for P_{m}. It is a relatively simple block that greatly reduces the number of function accesses per evaluation of the system's equations. Therefore, the hydraulic turbine's structure implies that the performance cost relative to the model's complexity does not justify a generic block model.

It is tempting to include algebraic variables in control block models to reduce and localize function accesses. For example, the output of the hydraulic turbine, P_m, could be chosen as an algebraic variable and the output equation (4.49) as its associated constraint equation. This selection would provide an advantage in the case where the turbine's output is used in a linear fashion as it could simply add an entry to $\mathbf{A}$. However, care must be taken when using this approach as it could introduce a significant number of additional variables. Furthermore, auxiliary variables can introduce numerical issues in a model, as it introduces algebraic variables and possibly hidden constraints [35]. Therefore, it was considered best practice to avoid algebraic variables and write functions to provide the effect of variables such as P_m.

4.4 Multimass Turbine

The final component included in the synchronous machine subset is the multimass component, which has been used in commercial EMT simulators and small signal stability analysis [15], [69]. Standard synchronous machine models assume that the entire shaft acts as a single rotating mass and is infinitely stiff. The torsional model relaxes this assumption and introduces several masses connected in a rotating spring-mass model as illustrated in Figure 4.9 [76]. This component only rotates and does not move in space like the synchronous machine model so its dynamics are completely described by rotational differential equations. The first mass is selected to be the generator mass, which simply means that the electromagnetic torque, T_e, is applied to this mass and no mechanical power is applied to it. The mechanical power is applied in a distributed fashion to the remaining masses.

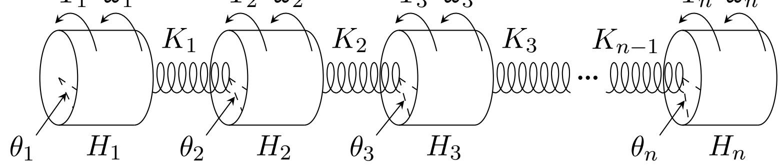

Figure 4.9: Multimass turbine model

Each mass has two equations (4.50) and (4.51) like the rotational equations used to model simple mechanical dynamics of machines. This models a swing equation or second order differential for each mass describing the rotational equations of motion. The main difference in this case is that each mass exerts a torque on its neighbouring masses according to their shared spring constant, K, that resists deviation from the rest state. Let T_i be the torque exerted on mass i, which is equal to T_e for the generator mass. For all masses except the generator mass, let α_i be the proportion of mechanical torque, T_m, exerted on mass i such that all proportions sum to one. It follows that the torque on mass i is equal to $-\alpha_i T_m$.

The differential equations for mass i in the multimass system are

$$\frac{d\delta_i}{dt} - \Delta\omega_i = 0 \text{ and} \tag{4.50}$$

$$2H_i \frac{d\Delta\omega_i}{dt} + D_i\Delta\omega_i + K_{i,i-1}\left(\delta_i - \delta_{i-1}\right) + K_{i,i+1}\left(\delta_i - \delta_{i+1}\right) - T_i = 0 \tag{4.51}$$

It should be noted that all indexes are in the range 1 to n. Any quantities associated with indexes 0 and $n+1$ must be ignored since the first and last masses in the multimass system have only a single neighbour.

The stator and electrical rotor equations are still used with the multimass model. The two mechanical equations, (4.31) and (4.32), are eliminated and replaced by the $2n$ equations of motion related to the multimass model. The rotor angle, δ_r, and speed deviation, $\Delta\omega_\mathrm{r}$, for the generator's electrical equations are equal to the angle and speed of the mass designated as the generator mass.

4.5 Single Machine Infinite Bus Test System

The following section will present simulation results of two simple single machine infinite bus test systems. These comparisons are provided as a benchmark to illustrate the accuracy of the dynamic phasor models in small single machine test cases. Section 4.5.1 contains a generator modeled with a standard round rotor and includes field and speed controllers. Section 4.5.2 contains a generator modeled with a multimass turbine and series capacitor to induce subsynchronous resonance.

4.5.1 Standard Turbine Model

Figure 4.10 illustrates the first test system, which is a single machine infinite bus system [1]. The machine on Bus 1 is connected through a step up transformer and transmission line to a large system represented by Bus 3, which is an infinite bus. The generator is modeled using a round rotor and includes an exciter, governor, and stabilizer.

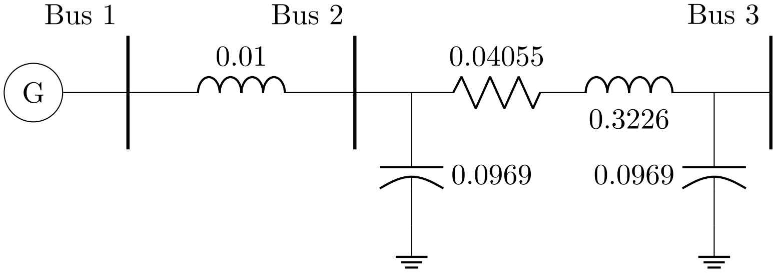

Figure 4.10: Single machine infinite bus test system

A 10 s simulation was carried out where a three-phase short circuit fault was at applied at Bus 2 for 0.1 s after 1 s of steady state operation. Simulations were carried out using the dynamic phasor (DP) program as well as commerical EMT and conventional transient stability (TS) simulators. Figure 4.11 illustrates the rotor speed deviation for the entire simulation. The results illustrate that the three programs are in good agreement for the long duration electromechanical oscillations of the rotor, which is the expected result. Figure 4.12 illustrates the rotor speed deviation for the first 0.3 s following the fault. The results illustrate that the exact behaviour is captured by the DP and EMT simulations, while the approximate behaviour is captured by the TS program. The oscillations shown in these results are equal to 50 Hz, which is the base frequency of the model. These oscillations are a result of the stator's differential transients [1]. Therefore, the expected behaviour is that the DP and EMT programs capture these transients and that the TS program neglects them but captures the approximate envelope of the rotor speed's dynamics during this period.

Figure 4.13 illustrates the generator's electromagnetic torque for the entire simulation. The electromagnetic torque has much more pronounced electromagnetic oscillations near the disturbance, which quickly decay leaving only the electromechanical oscillations. The three programs demonstrate excellent agreement for the electromechanical dynamics. Figure 4.14 illustrates the

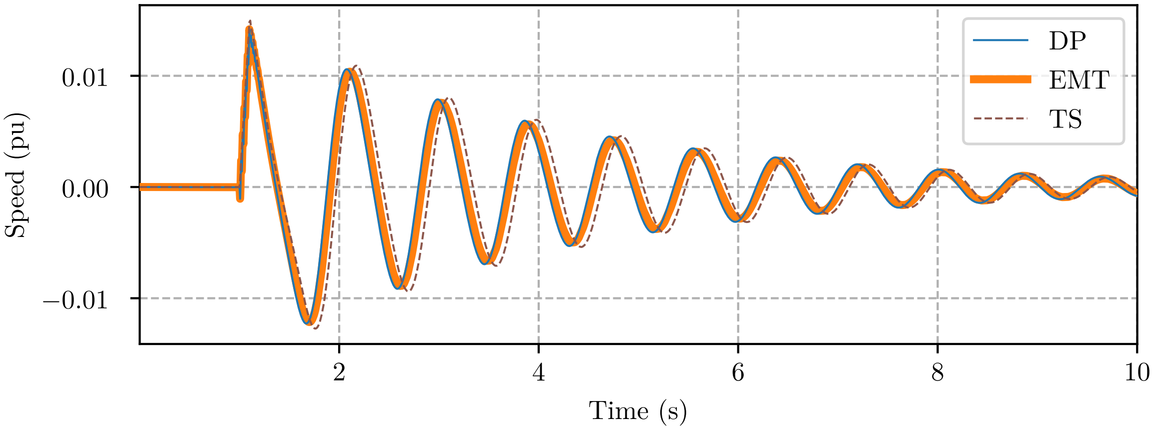

Figure 4.11: Rotor speed devation of the generator in the single machine infinite bus test system – entire simulation

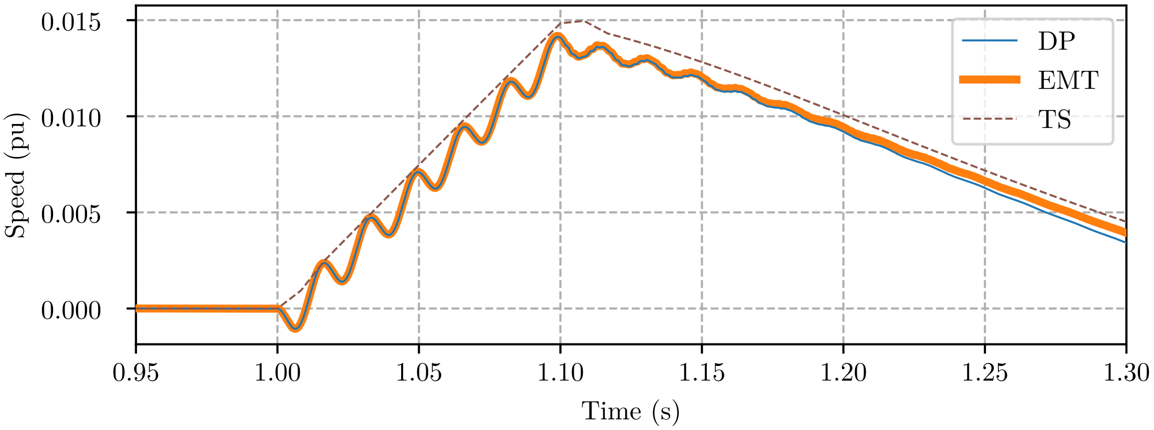

Figure 4.12: Rotor speed deviation of the generator in the single machine infinite bus test system – near the disturbance

electromagnetic torque dynamics within 0.3 s of the disturbance. The electromagnetic torque exhibits the same 50 Hz stator oscillations during the fault that were observed in the rotor speed deviation. Following the disturbance, there are oscillations in the range of 500 Hz, which are a result of the network's capacitors and inductors. These oscillations do not appear in the speed oscillations because the large mass of the rotor damps out the network transients. The DP and EMT results show excellent agreement for the network and stator transients while all three programs are in good agreement for the electromechanical transients.

Figure 4.15 illustrates the generator's terminal voltage near the point in the simulation where the fault is cleared. The DP and TS results are reconstructed from their phasor components using (2.8). The results illustrate that the voltage only consists of the base frequency

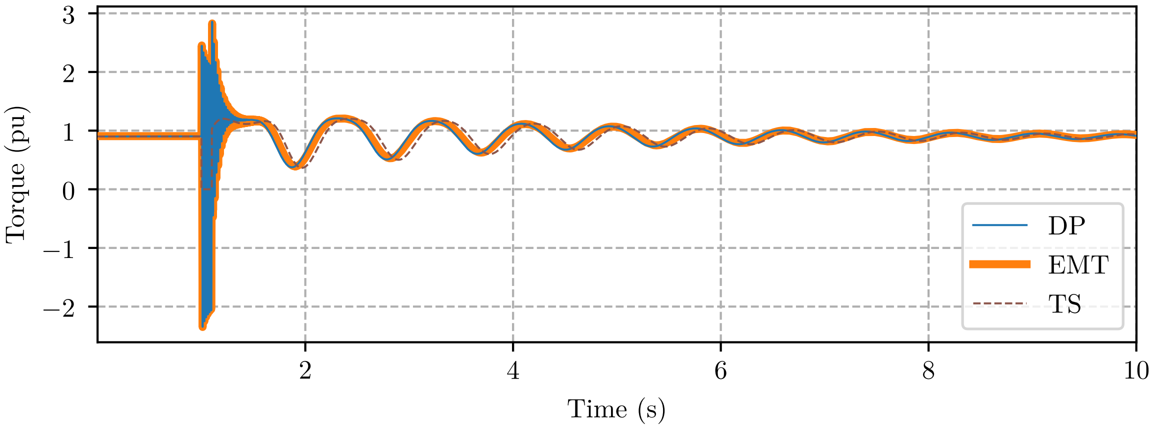

Figure 4.13: Electromagnetic torque of the generator in the single machine infinite bus test system – entire simulation

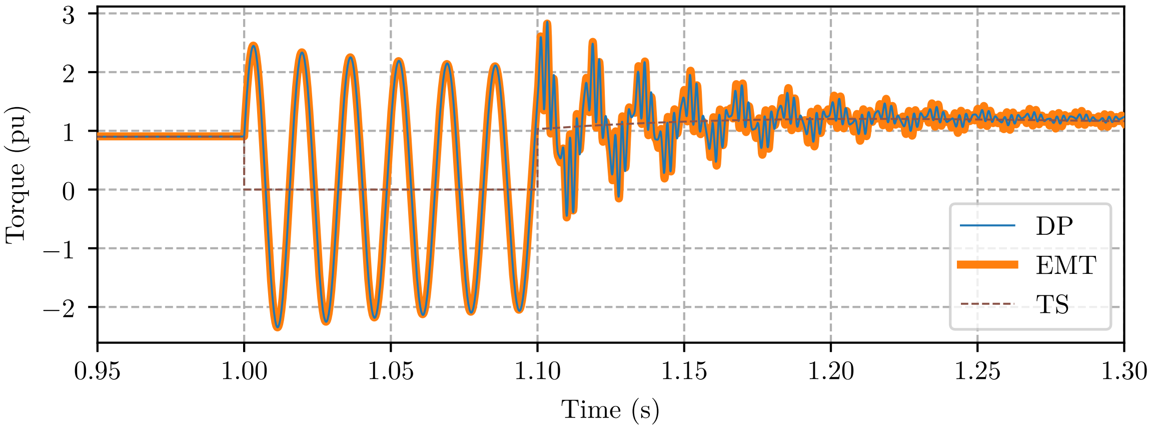

Figure 4.14: Electromagnetic torque of the generator in the single machine infinite bus test system – near the disturbance

component during the fault. However, high frequency network transients illustrated in the electromagnetic torque are extremely pronounced in the bus voltages once the fault is cleared. The three programs are in good agreement for the base frequency oscillations during the disturbance. Furthermore, the DP and EMT results are in excellent agreement for the high frequency transients following the fault.

The outputs of the three control elements modeled with the generator were also captured for comparison. Figure 4.16 illustrates the generator's field voltage, which is also its exciter's output. The three programs illustrate good agreement in both the frequency and dynamics of the field voltage transients. The EMT and DP programs match almost exactly for the duration of the simulations while the TS results include a small vertical offset. Figure 4.17 illustrates

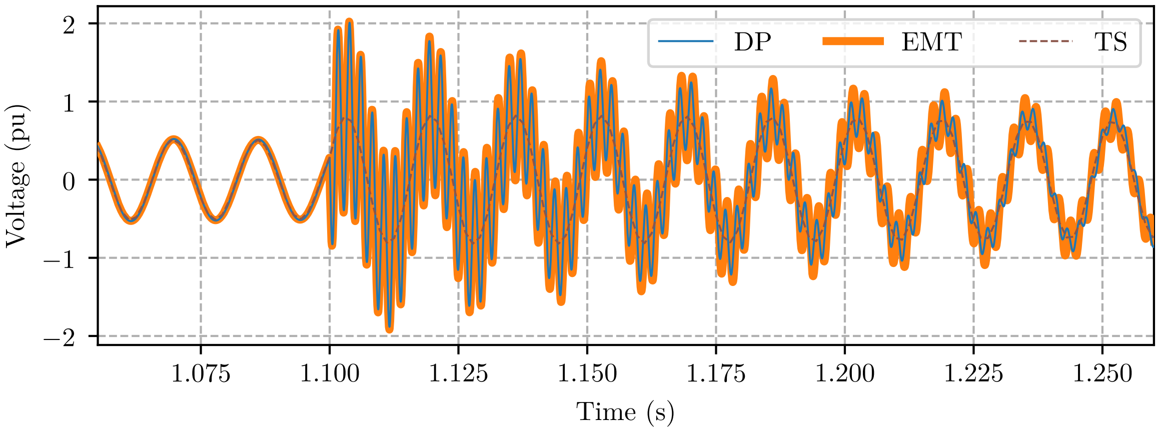

Figure 4.15: Terminal voltage of the generator in the single machine infinite bus test system

the output of the power system stabilizer, which demonstrates excellent agreement between the three programs for this control device. Finally, Figure 4.18 illustrates the generator's mechanical power, which is also its governor's output. The three programs show good agreement in the frequency and amplitude of oscillation in the mechanical power. There is a small vertical offset in the EMT program's results. This offset is the result of a small mismatch in the power losses computed by EMT when compared to the power flow program that was used to compute the initial conditions for the DP and TS simulations. This type of error was frequently observed in simulations and was generally in the same range as the one shown in Figure 4.18, which represents a mismatch of 0.5 %.

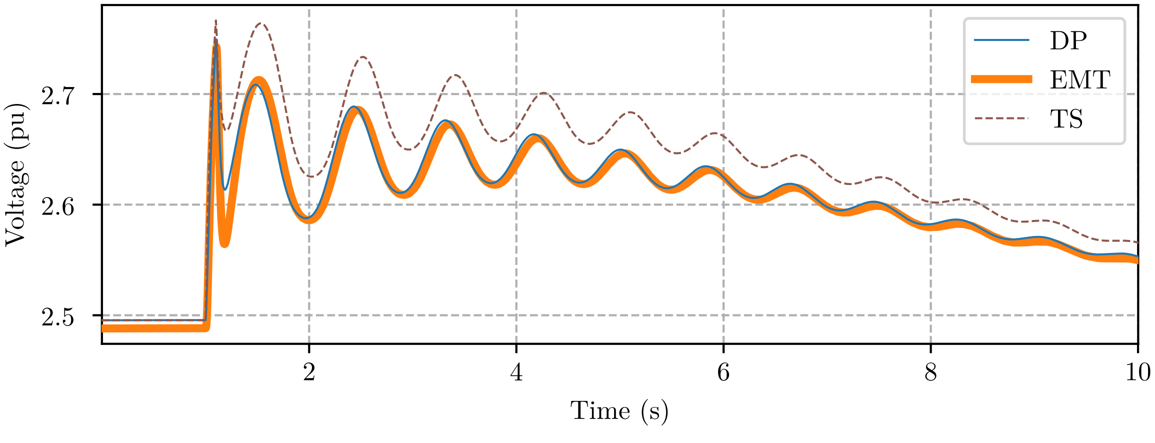

Figure 4.16: Field voltage of the generator in the single machine infinite bus test system

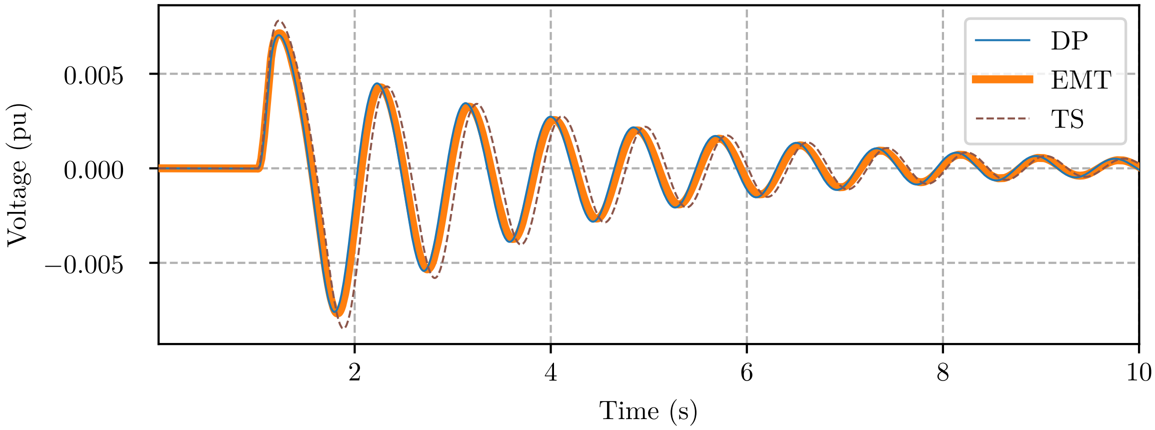

Figure 4.17: Stabilizer output of the generator in the single machine infinite bus test system

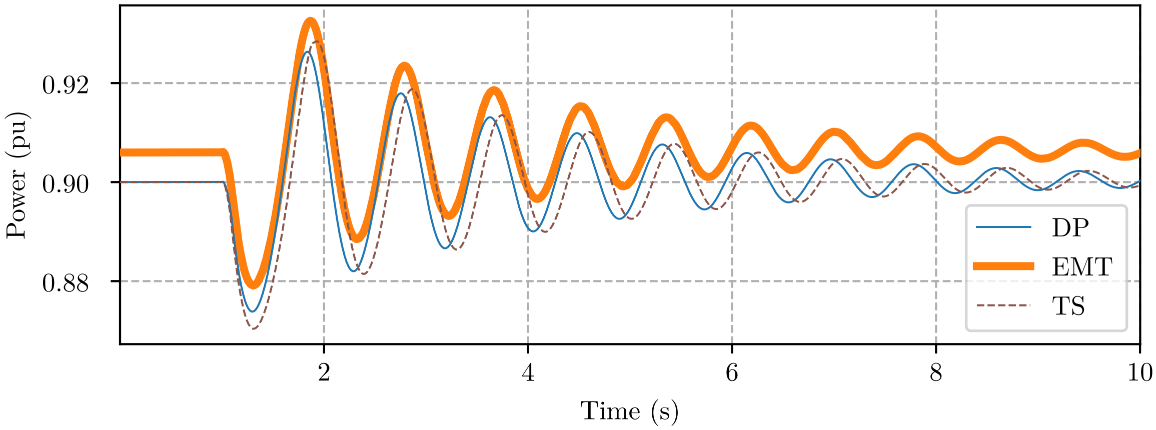

Figure 4.18: Mechanical power of the generator in the single machine infinite bus test system

4.5.2 Multimass Turbine Model

Figure 4.19 illustrates the single machine infinite bus test system used to test the multimass turbine model [76]. The generator in this model does not include any control models and uses a fixed field voltage and mechanical power. A capacitor was inserted in series with the generator to create an operating point with subsynchronous resonance. Bus 4 is modeled as an infinite bus in this system representing the remainder of a large power system.

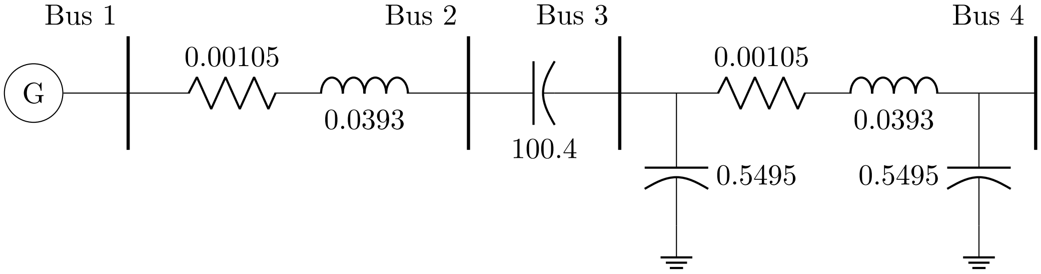

Figure 4.19: Single machine infinite bus test system including a multimass turbine model and series capacitor

A 20 s simulation was carried out with a 0.1 s three-phase short circuit fault applied on Bus 3 at 1 s. These simulations were only carried out using the DP and EMT programs since subsynchronous resonance cannot be modeled using the TS program. Furthermore, the DP program was run with relative error tolerances of 1×10^{-4}, 1×10^{-5}, and 1×10^{-6}. Figure 4.20 illustrates the generator's rotor speed deviation for the entire simulation. The results illustrate that all simulations produce unstable oscillations. However, the amplitude grows at a slower rate in the 1×10^{-4} DP simulation when compared to the other results.

Figure 4.21 illustrates the first second of the simulation following the disturbance. There are multiple modes of oscillation present at this point. However, the highest frequency and most apparent is around 30 Hz, which is in the range of torsional oscillations. The simulation results show excellent agreement between the EMT program and the DP program for all three values of the relative error tolerance.

Figure 4.22 illustrates the last 0.25 s of the simulation. At this point, a single oscillatory mode with frequency 32 Hz is present and represents the resonant frequency of the system. This mode is negatively damped so it grows in amplitude over time whereas all the other oscillation modes of the turbine are positively damped so they have decayed. These results show some

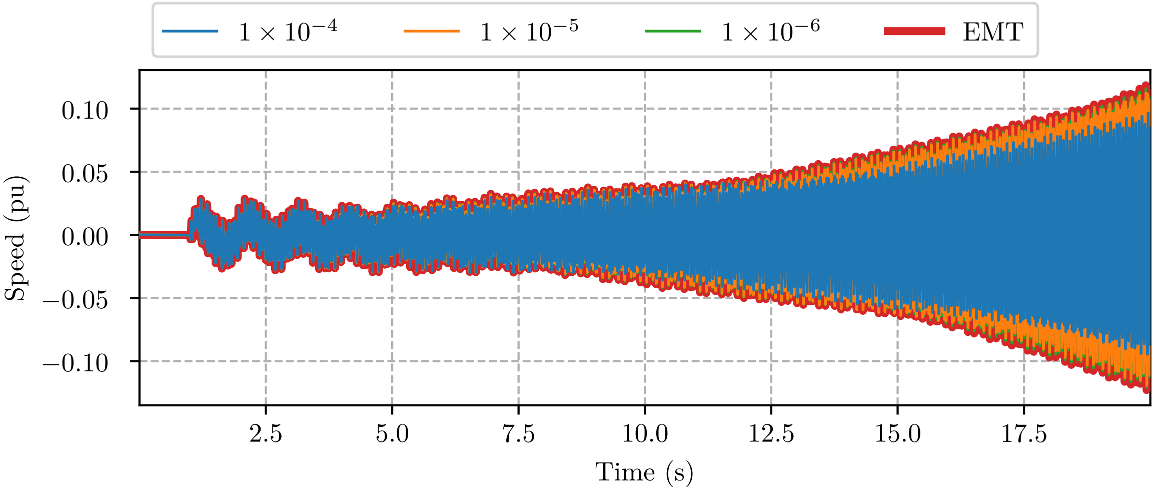

Figure 4.20: Rotor speed deviation of the generator in the multimass turbine test system – entire simulation

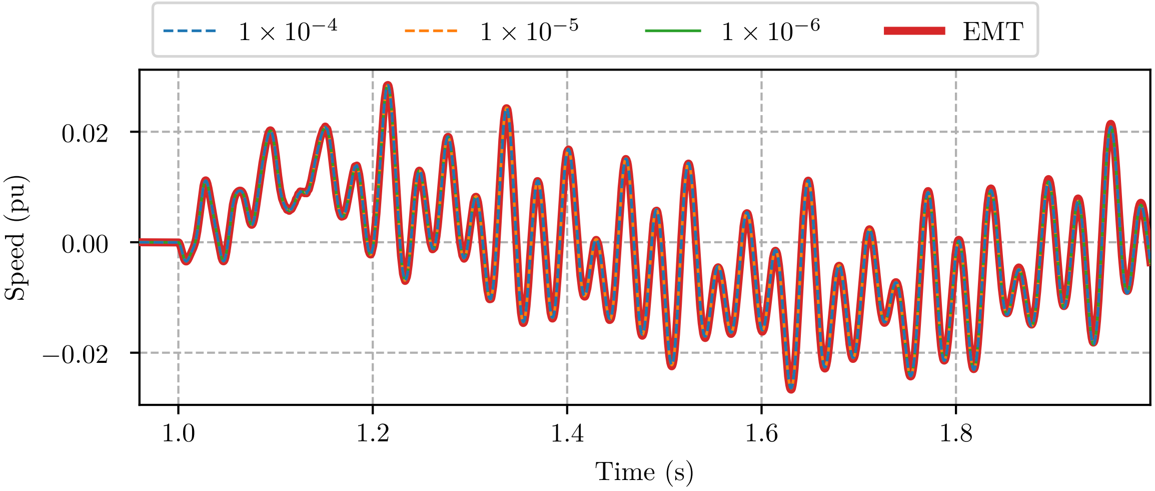

Figure 4.21: Rotor speed deviation of the generator in the multimass turbine test system – near the disturbance

amplitude and phase error in the 1×10^{-4} DP results. However, the 1×10^{-5} and 1×10^{-6} DP results are in excellent agreement with the EMT results.

The reason for the sensitivity in error tolerance in this case is related to the relationship between local truncation error and global error. The primary assumption behind variable step-size solvers is that local truncation error behaves similarly to global error. In other words, the solution's global error will be maintained at an acceptable level if the integrator controls and maintains a bound on the local truncation error. However, there are exceptions to this assumption particularly when eigenvalues with positive real parts are involved [35]. It is possible

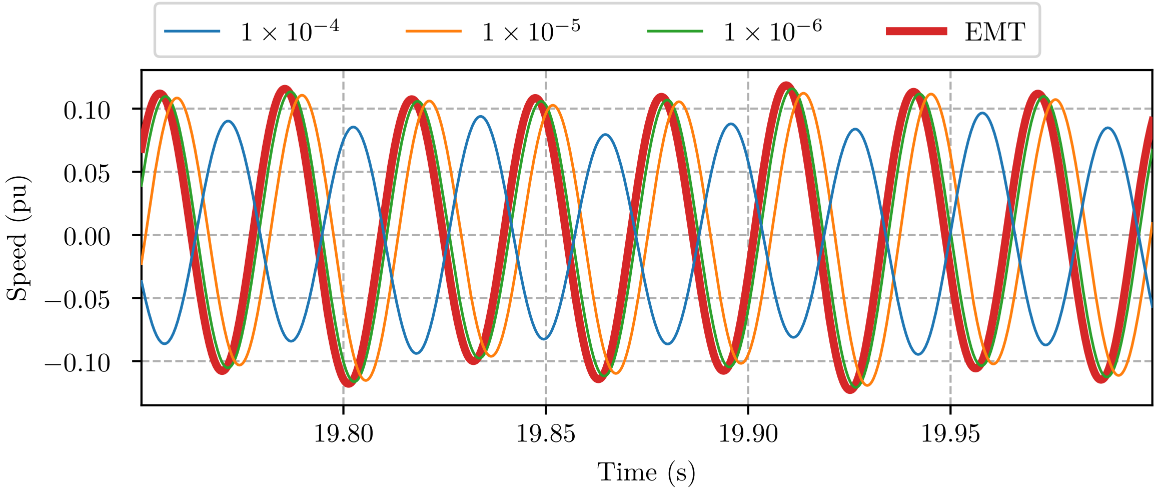

Figure 4.22: Rotor speed deviation of the generator in the multimass turbine test system – after 20 s

for the global error to grow at a much more significant rate than the local truncation error in these cases. This effect leads to a potentially poor representation of a system's dynamics even if the local truncation error is maintained to an acceptable tolerance.

The 1×10^{-4} DP simulation accurately captures the frequency of the negatively damped mode despite the relative error between the EMT solution and the 1×10^{-4} DP results. It follows that the 1×10^{-4} DP results can accurately detect the presence of subsynchronous resonance. Therefore, the 1×10^{-4} DP results still have practical value even if the unstable mode's damping is reproduced with some error.

Chapter 5

HVdc Transmission Systems

HVdc transmission represents an important innovation in the history of power systems. Power systems could grow beyond a few city blocks because of voltage transformation in ac networks [1]. However, dc transmission systems are better suited for long distance power transmission. They also offer a number of other benefits including providing an asynchronous link between different ac systems and better power transmission control. Furthermore, HVdc technology is rapidly improving and advancing making it a more attractive technology for many applications.

It is important for any power system simulation method to be able to accurately represent HVdc systems given their prevalence and growing numbers world wide. This work was limited in scope to the line commutated converter (LCC). This converter type is provided as an example of the dynamic phasor simulation method's capability to capture dynamics associated with power electronic systems. The foundations for modeling power electronic converters is provided, which can be used to create components for other converter systems.

Figure 5.1 illustrates the CIGRE benchmark HVdc transmission system, which is used as a common open source model used for HVdc research [77], [78]. It is a monopolar arrangement with twelve-pulse converters located at both the rectifier and inverter. The ac systems are modeled using equivalent sources and both are 50 Hz systems. This model forms the foundation for the discussions in the following chapters, providing context for the different parts of the HVdc transmission system model. However, the HVdc model developed in this work is not exclusive to the CIGRE benchmark model and can be extended to other transmission systems through minor modifications.

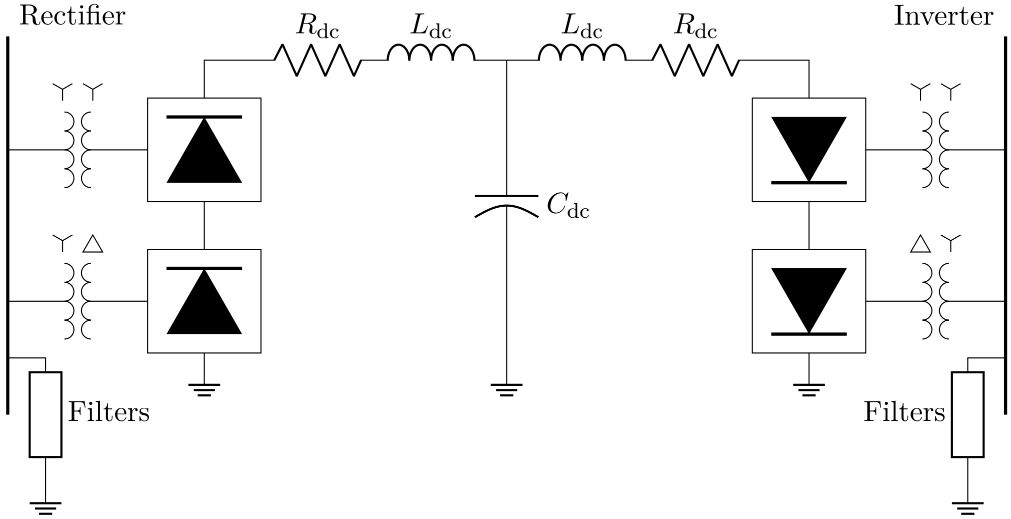

Figure 5.1: CIGRE benchmark HVdc model

Section 5.1 includes a brief introduction into steady state operation of LCCs. This discussion includes waveforms and equations for the two dependent quantities in the LCC model, which are the dc voltage and ac current. Section 5.2 adapts the equations presented in the background section to develop a model suitable for dynamic phasor simulation. Models for the dc transmission line and ac harmonic filters are provided in Sections 5.3 and 5.4, respectively. The control systems associated with the HVdc system are introduced in Section 5.5, which are modeled using the generic system developed for synchronous machine controls. Adaptations to the converter model to accommodate adverse operating conditions, which are common during large scale disturances, are discussed in Section 5.6. Finally, simulation results of a fault at the rectifier bus are included in Section 5.7 to illustrate the performance of the dynamic phasor HVdc model.

5.1 Background

Figure 5.2 illustrates the three-phase six-pulse converter bridge topology, which is the fundamental building block of the LCC based HVdc system [79]. The converter bridge is composed of six thyristors, which are fired in a sequence as indicated by their subscript. The manner in which they are operated produces dc voltage and current waveforms that are composed of six identical sections per cycle. These converters are sometimes referred to as six pulse converters and the dc side contains only sixth harmonics relative to the system's fundamental frequency. The inductance on the source side is included in the analysis of converters because it is fundamental to its operation. This inductance often models a transformer that is connected between the converter's terminals and a system.

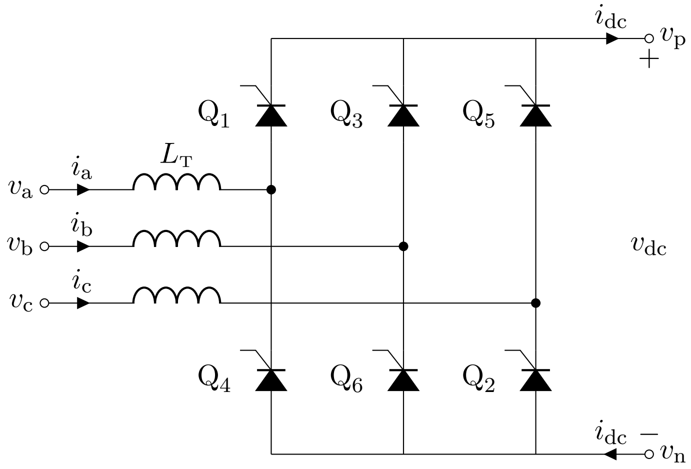

Figure 5.2: Three-phase six-pulse converter bridge

Analysis of the Graetz bridge includes two primary assumptions [79]. First is that the dc side current is constant, which is reasonable given the size of dc side inductances used for filtering and the dc cable's inductance. The second assumption is that the ac voltage is an ideal sinusoid. This assumption is also reasonable given that ac filters accompany practical HVdc installations to remove any harmonics produced by the HVdc converters. Consider the top three thyristors, Q_1, Q_3, and Q_5. The constant current assumption implies that at least one of the thyristors must be on at all times to conduct the dc side current. This circuit conducts the dc current using the phase associated with the last thyristor that was fired. The current can change phases instantaneously without the ac source inductance provided the phase voltage of the on-going phase is greater than the voltage of the off-going phase. The presence of the source side inductance makes the commutation process more complicated as the current cannot be instantaneously transferred from the off-going phase to the on-going phase. Instead there is a finite duration in which the current transitions from the one phase to the other. This period is referred to as the commutation angle or the overlap angle. It should be noted that all of the above discussion applies to the bottom three thyristors, Q_2, Q_4, and Q_6. However, the voltage of the off-going thyristor must be greater than the voltage of the on-going thyristor in the negative group.

Figures 5.3 and 5.4 illustrate the operation of the six-pulse bridge in terms of its voltages and currents, respectively. The commutation angle is denoted by μ. The delay between when commutation can occur between two phases and when commutation is initiated is referred to as the firing angle, α. The voltage waveforms illustrate that the positive and negative terminal waveforms follow the voltage of the phase that is presently conducting current during normal conduction. It will be shown that the voltage at the dc terminal is equal to the average of the on and off-going phases during the commutation period. The current waveforms illustrate that one of the phases in the top or bottom circuit carries the entire dc side current during normal conduction. The total current shared by the on and off-going phases is equal to the dc side current and it smoothly transitions from the off to the on-going phase during the commutation period.

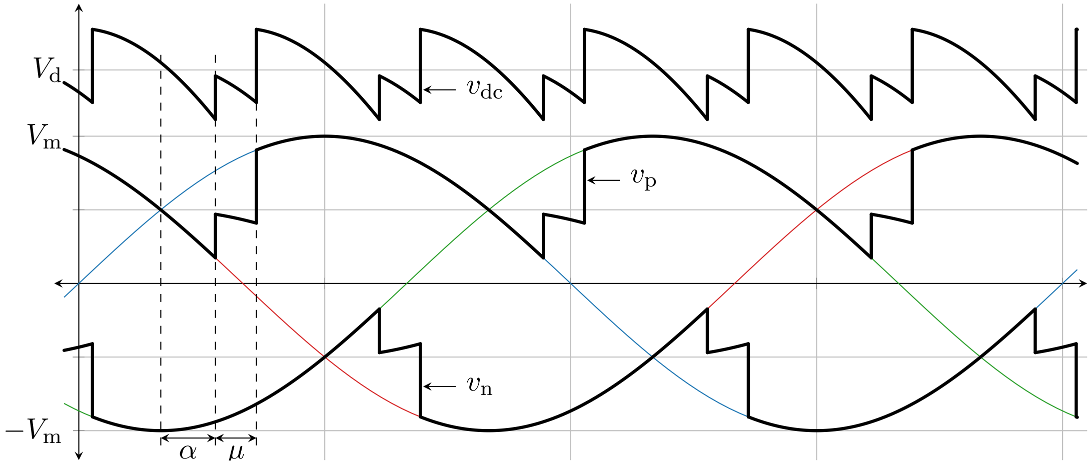

Figure 5.3: Terminal voltage on the dc side of a six-pulse bridge

Two phases are conducting current in either the positive or negative circuit during commutation. It is assumed that the commutation angle is less than sixty degrees such that commutation is either occurring in either the positive or the negative circuits but never both at the same time [79]. Figure 5.5 illustrates the commutation period where Q_1 or the a-phase takes over conduction of I_dc from Q_5 or the c-phase. Q_4 or the b-phase is in the normal conduction period for the negative circuit and is carrying $-I_\mathrm{dc}$.

Assume that subscripts x and y correspond to quantities associated with the on-going and

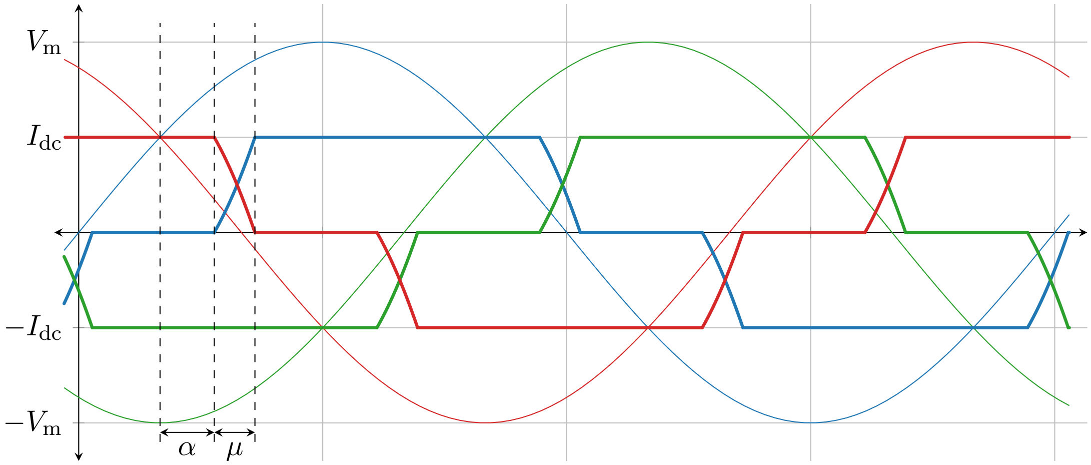

Figure 5.4: Phase current on the ac side of a six-pulse bridge

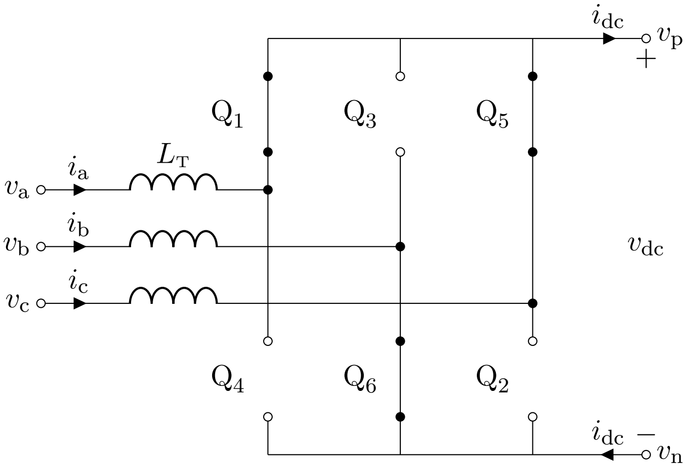

Figure 5.5: Commutation period where Q_1 takes over conduction of I_{dc} from Q_5

off-going phases, respectively. Furthermore, assume that subscript z corresponds to quantities of the phase that is not involved in commutation. For example, in Figure 5.5 v_x would be v_{a}, v_y would be v_{c}, and v_z would be v_{b}. Figure 5.5 shows that the on and off-going branches are connected in parallel during commutation. The voltage loop equation is

$$v_x - v_y - L_{\mathrm{T}} \left(\dot{i}_x - \dot{i}_y \right) = 0. \tag{5.1}$$

Let η be defined as a multiplier that is equal to 1 when commutation is occurring in positive thyristor group and −1 when commutation is occurring in the negative group. The current balance at the dc terminal is

$$i_x + i_y - \eta i_{\mathrm{dc}} = 0 \tag{5.2}$$

Differentiating (5.2) implies that

$$\dot{i}_x = -\dot{i}_y \tag{5.3}$$

since i_{dc} is assumed to be constant. An equation for the on-going current's derivative can be derived by substituting (5.3) into (5.1)

$$\dot{i}_x = \frac{v_x - v_y}{2L_{\mathrm{T}}}. \tag{5.4}$$

Let t_0 be the time at which the on-going thyristor is fired and commutation begins. The on-going current is found by integrating (5.4) starting at t_0

$$i_x = \frac{1}{2L_{\mathrm{T}}} \int\limits_{t_0}^{t} \left(v_x - v_y\right) \mathrm{d}\tau \tag{5.5}$$

given that i_x is zero when commutation begins. Figure 5.5 illustrates that v_{dc} is equal to

$$v_{\mathrm{dc}} = \eta \left(v_x - L_{\mathrm{T}} \dot{i}_x - v_z\right). \tag{5.6}$$

Substituting (5.4) for the derivative of i_x in (5.6)

$$v_{\mathrm{dc}} = \eta \left(\frac{v_x + v_y}{2} - v_z\right). \tag{5.7}$$

Commutation begins at an angle of α following the zero crossing of $v_x - v_y$. The positive zero crossing of this voltage is used for the positive thyristor group and the negative zero crossing is used for the negative thyristor group. Assuming that the ac voltage is balanced and sinusoidal, let V_{m} and θ be the magnitude and angle of the ac system phase voltage, respectively. Let ζ be the phase shift of v_x relative to the a-phase voltage. The on-going phase voltage can be written as

$$v_x\left(t\right) = V_{\mathrm{m}} \cos\left(\omega_0 t + \theta + \zeta\right). \tag{5.8}$$

Similary, the off-going phase voltage can be written as

$$v_y\left(t\right) = V_{\mathrm{m}} \cos\left(\omega_0 t + \theta + \zeta + {}^{2\pi}/_{3}\right) \tag{5.9}$$

since the off-going thyristor voltage leads the on-going thyristor by ${}^{2\pi}/_{3}$ radians. The voltage of the phase not involved in commutation can be written as

$$v_z\left(t\right) = V_{\mathrm{m}} \cos\left(\omega_0 t + \theta + \zeta - {}^{2\pi}/_{3}\right) \tag{5.10}$$

by process of elimination. The line voltage is found by subtracting (5.9) from (5.8)

$$v_x - v_y = \sqrt{3} V_{\mathrm{m}} \cos\left(\omega_0 t + \theta + \zeta - \frac{\pi}{6}\right). \tag{5.11}$$

The zero crossings of (5.11) occur when the condition

$$\omega_0 t + \theta + \zeta - \frac{\pi}{6} = \frac{\left(2k-1\right)\pi}{2} \tag{5.12}$$

is satisfied for any integer k. Conduction can begin at positive zero crossings for the positive thyristor group, which corresponds to even values of k. Conduction can begin at negative zero crossings for the negative thyristor group, which corresponds to odd values of k. As previously mentioned, commutation commences at time t_0 or at an angle of α after the first moment commutation can begin. Therefore, t_0 can be found by solving (5.12) and shifting the result by an angle of α

$$\omega_0 t_0 = \alpha - \theta - \zeta + k\pi - \frac{\pi}{3}. \tag{5.13}$$

(5.13) is expressed in terms of the actual a-phase angle, θ. However, in real converter systems the system's phase angle is estimated using a phase-locked loop [79]. Let ϕ be the output of the converter's phase-locked loop, which replaces θ in (5.13)

$$\omega_0 t_0 = \alpha - \phi - \zeta + k\pi - \frac{\pi}{3}. \tag{5.14}$$

The multiplier η used to distinguish between commutation in the positive and negative thyristor groups can be adapted to use k

$$\eta = (-1)^k . \tag{5.15}$$

Substituting (5.15) into the dc current balance equation (5.2)

$$i_x + i_y - (-1)^k i_{\mathrm{dc}} = 0. \tag{5.16}$$

An expression for the on-going current during commutation is found by solving the integral (5.5) substituting in (5.11) and (5.14)

$$i_x(t) = \frac{\sqrt{3}V_{\mathrm{m}}}{2X_{\mathrm{T}}} \left(\cos(\alpha + \theta - \phi + k\pi) - \cos\left(\omega_0 t + \theta + \zeta + \frac{\pi}{3} \right) \right) . \tag{5.17}$$

The off-going current can be found by substituting (5.17) into (5.16) and solving for i_y

$$i_y(t) = (-1)^k i_{\mathrm{dc}} - \frac{\sqrt{3}V_{\mathrm{m}}}{2X_{\mathrm{T}}} \left(\cos(\alpha + \theta - \phi + k\pi) - \cos\left(\omega_0 t + \theta + \zeta + \frac{\pi}{3} \right) \right) . \tag{5.18}$$

Commutation ends after the commutation angle, μ, has elapsed following the onset of commutation. At this point, the off-going current must be equal to zero and the on-going current must be equal to $(-1)^k i_{\mathrm{dc}}$. Substituting these boundary conditions into (5.17) and solving for μ when $k = 0$ gives an expression for the commutation angle

$$\mu = \cos^{-1}\left(\cos(\alpha + \theta - \phi) - \frac{2X_{\mathrm{T}} i_{\mathrm{dc}}}{\sqrt{3}V_{\mathrm{m}}} \right) - (\alpha + \theta - \phi) . \tag{5.19}$$

It is possible to show that the same expression is obtained for any choice of k. However, the choice of k must be resolved before solving for μ as the effect of rotating by a multiple of π is lost once the inverse cosine function is applied.

5.2 Dynamic Phasor Model

Developing a dynamic phasor model suitable for transient stability simulation involves computing the average value of all dc quantities and the first harmonic of all network quantities [69], [80]. The HVdc system model is a two terminal device with a current expression necessary at the rectifier and inverter from the perspective of MNA equations. The dc quantities are treated as internal variables even though they are the result of the dc transmission line's circuit model. These quantities are dc and modeled using real variables and equations while the MNA portion of the program was developed for ac quantities and complex variables. However, this limitation is part of the program's current implementation and it would be possible to extend the MNA program to accept both ac and dc nodes.

The model developed in this work uses the terminal bus ac voltage dynamic phasor, $\overline{V}_{\mathrm{ac}}$, and the average value of the dc side current, I_{dc}, as inputs. The terminal bus's KCL equation constrains its ac voltage and the dc transmission line constrains the dc current. Furthermore, it is assumed that the instantaneous dc side current i_{dc} used in the equations developed in the previous section is equal to its average value I_{dc}. The converter model calculates and outputs the average dc side voltage, V_{dc}, and the ac side current dynamic phasor, $\overline{I}_{\mathrm{ac}}$.

The average value of the dc side voltage is derived using the fact that the average value of the dc voltage at the positive dc node is equal to the negative value at the negative terminal. Furthermore, the voltage at the positive node consists of three equal segments per period of the fundamental frequency. Consequently, the average value of the dc voltage is found by evaluating

$$V_{\mathrm{dc}} = \frac{6\,(-1)^k}{T_0}\left(\int\limits_{t_0}^{t_0+\mu/\omega_0}\left(\frac{v_x+v_y}{2}-v_z\right)\mathrm{d}\tau + \int\limits_{t_0+\mu/\omega_0}^{t_0+T_0/6}\left(v_x-v_z\right)\mathrm{d}\tau\right). \tag{5.20}$$

Solving (5.20) using (5.8), (5.9), (5.9), and (5.14)

$$V_{\mathrm{dc}} = \frac{3\sqrt{3}V_{\mathrm{m}}}{2\pi}\left(\cos\left(\alpha+\theta-\phi\right)+\cos\left(\alpha+\mu+\theta-\phi\right)\right). \tag{5.21}$$

(5.21) shows that V_{dc} is independent of k and ζ. This result is expected since the average value of the dc side voltage is not dependent on the specific conduction and commutation interval used to evaluate (5.20). Substituting (5.19) into (5.21)

$$V_{\mathrm{dc}} = \frac{3\sqrt{3}V_{\mathrm{m}}}{\pi}\cos\left(\alpha+\theta-\phi\right)-\frac{3X_{\mathrm{T}}}{\pi}I_{\mathrm{dc}}. \tag{5.22}$$

Realistic HVdc transmission systems are comprised of multiple converter bridges. This arrangement offers a number of benefits including increased transmission capacity and harmonic cancellation [79]. The converter bridges are connected in series and firing signals are shifted to achieve cancellation of lower order harmonics. This phase shift does not affect the average value of dc quantities so the dc side voltages add in this arrangement. Consequently, if there are N bridges connected in series then the dc voltage is equal to

$$V_{\mathrm{dc}} = \frac{3\sqrt{3}NV_{\mathrm{m}}}{\pi}\cos\left(\alpha+\theta-\phi\right) - \frac{3NX_{\mathrm{T}}}{\pi}I_{\mathrm{dc}}. \tag{5.23}$$

An expression of the first harmonic of the ac current is required for KCL equations at the HVdc system's terminals. The first harmonic is a good approximation of a converter's ac current for realistic systems because the combination of ac filters and multiple converters eliminates most harmonic content [80]. The a-phase current must be used to derive a dynamic phasor as the phase of the ac current must be aligned with the ac voltage dynamic phasor. The on-going current is equal to (5.17) with $k = 0$ and $\zeta = 0$ when Q_1 takes over conduction from Q_5

$$i_{\mathrm{a},1}\left(t\right) = \frac{\sqrt{3}V_{\mathrm{m}}}{2X_{\mathrm{T}}}\left(\cos\left(\alpha+\theta-\phi\right) - \cos\left(\omega_0 t+\theta+\frac{\pi}{3}\right)\right). \tag{5.24}$$

The a-phase circuit then carries the full dc current in the positive direction until the gate signal for Q_3 is fired at time $t_0 + {}^{T_0}\!/_3$. The a-phase off-going current is equal to (5.18) with $\zeta = -\frac{2\pi}{3}$ and $k = 0$ when Q_3 takes over conduction from Q_1

$$i_{\mathrm{a},2}\left(t\right) = I_{\mathrm{dc}} - \frac{\sqrt{3}V_{\mathrm{m}}}{2X_{\mathrm{T}}}\left(\cos\left(\alpha+\theta-\phi\right) - \cos\left(\omega_0 t+\theta-\frac{\pi}{3}\right)\right). \tag{5.25}$$

The a-phase current does not conduct any current once commutation is complete until the gate signal for Q_4 is fired at time $t_0 + {}^{T_0}\!/_2$. The a-phase on-going current is equal to (5.17) with $\zeta = 0$ and $k = 1$ when Q_4 takes over conduction from Q_2

$$i_{\mathrm{a},3}\left(t\right) = -\frac{\sqrt{3}V_{\mathrm{m}}}{2X_{\mathrm{T}}}\left(\cos\left(\alpha+\theta-\phi\right) - \cos\left(\omega_0 t+\theta-\frac{2\pi}{3}\right)\right). \tag{5.26}$$

The a-phase circuit then carries the full dc current in the negative direction until the gate signal for Q_6 is fired at time $t_0 + {}^{5T_0}\!/_6$. The a-phase off-going current is equal to (5.18) with $\zeta = -\frac{2\pi}{3}$

and $k = 1$ when Q_6 takes over conduction from Q_4

$$i_{\mathrm{a},4}\left(t\right) = \frac{\sqrt{3}V_{\mathrm{m}}}{2X_{\mathrm{T}}}\left(\cos\left(\alpha+\theta-\phi\right) - \cos\left(\omega_0 t + \theta - \frac{4\pi}{3}\right)\right) - I_{\mathrm{dc}}. \tag{5.27}$$

The a-phase branch does not conduct any current after this interval until Q_1 takes over conduction from Q_5 at time $t_0 + T_0$ in the next cycle.

The definitions for the segments of i_{a} illustrate that there is a great deal of symmetry. It is possible to use this symmetry to simplify the process of deriving a dynamic phasor for the ac current. First of all, the negative half of the ac current waveforms shown in Figure 5.4 are equal to the positive halves reflected over the time axis and shifted to the right by a half cycle. In terms of the definitions (5.24) to (5.27)

$$i_{\mathrm{a},3}\left(t\right) = i_{\mathrm{a},1}\left(t - \frac{T_0}{2}\right) \text{ and} \tag{5.28}$$

$$i_{\mathrm{a},4}\left(t\right) = i_{\mathrm{a},2}\left(t - \frac{T_0}{2}\right). \tag{5.29}$$

Furthermore, the a-phase carries i_{dc} in the interval $t_0 + {}^{\mu}\!/\!{}_{\omega_0}$ to $t_0 + {}^{T_0}\!/\!{}_3$. The a-phase carries $-i_{\mathrm{dc}}$ in the interval $t_0 + {}^{T_0}\!/\!{}_2 + {}^{\mu}\!/\!{}_{\omega_0}$ to $t_0 + {}^{5T_0}\!/\!{}_6$, which is equal to the positive dc current interval shifted by ${}^{T_0}\!/\!{}_2$. It can be shown using symmetry that if c_1 is the fundamental complex Fourier coefficient for the positive section of the ac current then

$$\overline{I}_{\mathrm{ac}} = 4c_1. \tag{5.30}$$

The Fourier coefficient c_1 can be found by evaluating

$$c_1 = \frac{1}{T_0}\left(\int\limits_{t_0}^{t_0+{}^{\mu}/{}_{\omega_0}} i_{\mathrm{a},1}\left(\tau\right)e^{-j\omega_0\tau}\mathrm{d}\tau + \int\limits_{t_0+{}^{\mu}/{}_{\omega_0}}^{t_0+{}^{T_0}/{}_3} I_{\mathrm{dc}}e^{-j\omega_0\tau}\mathrm{d}\tau + \int\limits_{t_0+{}^{T_0}/{}_3}^{t_0+{}^{T_0}/{}_3+{}^{\mu}/{}_{\omega_0}} i_{a,2}\left(\tau\right)e^{-j\omega_0\tau}\mathrm{d}\tau\right). \tag{5.31}$$

Symmetry can also be observed in (5.24) and (5.25). These definitions illustrate that $i_{\mathrm{a},2}$ can be written in terms of $i_{\mathrm{a},1}$ and I_{dc} using the relationship

$$i_{\mathrm{a},2}\left(t\right) = I_{\mathrm{dc}} - i_{\mathrm{a},1}\left(t - \frac{T_0}{3}\right). \tag{5.32}$$

Substituting (5.32) into (5.31) and simplifying

$$c_1 = \frac{1}{T_0} \left(\int\limits_{t_0+\mu/\omega_0}^{t_0+T_0/3+\mu/\omega_0} I_{\mathrm{dc}} e^{-j\omega_0\tau} \mathrm{d}\tau + \sqrt{3} e^{j\frac{\pi}{6}} \int\limits_{t_0}^{t_0+\mu/\omega_0} i_{\mathrm{a},1}\left(\tau\right) e^{-j\omega_0\tau} \mathrm{d}\tau \right). \tag{5.33}$$

This equation illustrates that there are two parts to the ac current dynamic phasor. The first integral

$$c_{1,\mathrm{dc}} = \frac{1}{T_0} \int\limits_{t_0+\mu/\omega_0}^{t_0+T_0/3+\mu/\omega_0} I_{\mathrm{dc}} e^{-j\omega_0\tau} \mathrm{d}\tau \tag{5.34}$$

is related to the dc current. Evaluating (5.34) using the fact that k is even and $\zeta = 0$ in t_0 for Q_1

$$c_{1,\mathrm{dc}} = \frac{\sqrt{3}}{2\pi} I_{\mathrm{dc}} e^{j(\phi-\alpha-\mu)}. \tag{5.35}$$

This component is the dominant portion of the ac current dynamic phasor. The second integral in (5.33) after substituting the expression in (5.24) for $i_{\mathrm{a},1}$ is

$$c_{1,\mathrm{ac}} = \frac{3V_{\mathrm{m}} e^{j\frac{\pi}{6}}}{2X_{\mathrm{T}}T_0} \int\limits_{t_0}^{t_0+\mu/\omega_0} \left(\cos\left(\alpha+\theta-\phi\right) - \cos\left(\omega_0\tau+\theta+\frac{\pi}{3}\right)\right) e^{-j\omega_0\tau} \mathrm{d}\tau. \tag{5.36}$$

This component is related to the ac bus voltage and arises due to the commutation period [69], [80]. Evaluating (5.36)

$$c_{1,\mathrm{ac}} = \frac{3V_{\mathrm{m}} e^{j\theta}}{8\pi X_{\mathrm{T}}} \left(\left(1 + e^{j2(\phi-\alpha-\theta)}\right)\left(1 - e^{-j\mu}\right) + \frac{e^{j(\phi-\alpha-\theta)\left(e^{-j2\mu}-1\right)}}{2} - j\mu \right). \tag{5.37}$$

This completely is generally not included in dynamic phasor HVdc models. However, its contribution to the model's performance was found to be important for accurately modeling low impedance faults near the converter terminals. Combining (5.35) and (5.37) together and substituting into (5.30)

$$\begin{aligned} \overline{I} = \frac{2\sqrt{3}N}{\pi} I_{\mathrm{dc}} e^{j(\phi-\alpha-\mu)} + \frac{3NV_{\mathrm{m}} e^{j\theta}}{2\pi X_{\mathrm{T}}} \Bigg(&\left(1 + e^{j2(\phi-\alpha-\theta)}\right)\left(1 - e^{-j\mu}\right) \\ &+ \frac{e^{j(\phi-\alpha-\theta)\left(e^{-j2\mu}-1\right)}}{2} - j\mu \Bigg). \end{aligned} \tag{5.38}$$

A factor of N was also included in (5.38) to model stacked converter bridges as discussed in the dc voltage equation.

The final model must be compatible with the per unit system developed to model the ac system. The ac voltage input to the dc voltage expression (5.23) must be in terms of the terminal's per unit dynamic phasor. Let $V_{\text{base,s}}$ be the base line-to-line voltage of the converter transformer's secondary winding. In terms of the ac system's quantities, the dc voltage expression (5.23) is

$$V_{\text{dc}} = \frac{3\sqrt{2}NV_{\text{base,s}}}{\pi}\left|\overline{V}_{\text{ac}}\right|\cos\left(\alpha+\theta-\phi\right) - \frac{3NX_{\text{T}}}{\pi}I_{\text{dc}}. \tag{5.39}$$

The commutation angle expression (5.19) must also be adapted to use the ac system's quantities

$$\mu = \cos^{-1}\left(\cos\left(\alpha+\theta-\phi\right) - \frac{\sqrt{2}X_{\text{T}}I_{\text{dc}}}{V_{\text{base,s}}\left|\overline{V}_{\text{ac}}\right|}\right) - \left(\alpha+\theta-\phi\right). \tag{5.40}$$

The ac current must be specified in per unit to be compatible with the system model. (5.38) is the peak of the ac current's dynamic phasor so it must be scaled by the base peak current on the primary side of the converter's transformer. Let $V_{\text{base,p}}$ be the base line-to-line voltage of the converter transformer's primary winding and let S_{base} be the system's base power. The primary side base peak current, $I_{\text{base,p}}$, is defined in terms of these base two quantities as

$$I_{\text{base,p}} = \sqrt{\frac{2}{3}}\frac{S_{\text{base}}}{V_{\text{base,p}}}. \tag{5.41}$$

Dividing (5.38) by (5.41) and substituting in the ac voltage dynamic phasor

$$\begin{aligned}\overline{I}_{\text{ac}} = \frac{\sqrt{6}NV_{\text{base,p}}}{\pi S_{\text{base}}}I_{\text{dc}}e^{j(\phi-\alpha-\mu)} + \frac{3NV_{\text{base,p}}V_{\text{base,s}}}{2\pi X_{\text{T}}S_{\text{base}}}\overline{V}_{\text{ac}}&\left(\left(1+e^{j2(\phi-\alpha-\theta)}\right)\left(1-e^{-j\mu}\right)\right.\\&\left.+\frac{e^{j(\phi-\alpha-\theta)}\left(e^{-j2\mu}-1\right)}{2} - j\mu\right).\end{aligned} \tag{5.42}$$

5.3 Transmission Line

The converter model previously discussed takes the ac voltage and dc current as inputs and computes the ac current and dc voltage as outputs. Therefore, any dc transmission system model can be connected to the converter model provided it outputs a dc current value and accepts a dc voltage value as input. It was deemed sufficient to use the transmission line model illustrated in Figure 5.1 for this work. However, the dc transmission system model can be extended to use a traveling wave Bergeron components or even a multi-terminal dc system.

Figure 5.6 illustrates the two terminal equivalent circuit for the dc transmission line included in the CIGRE benchmark case [81]. The capacitor is unchanged by the presence of dynamic phasor-based converter models. However, the resistance and inductances in the series branches must be adapted to account for the effects of the converters and converter transformers. The effective dc resistance, R_{eff}, is equal to the total resistance as seen by the dc current. It takes into account the dc line and the effects of commutation on the dc voltage developed by the converter, which is

$$R_{\text{eff}} = R_{\text{dc}} + \frac{3NX_{\text{T}}}{\pi}. \tag{5.43}$$

Similarly, the effective dc inductance, L_{eff}, is equal to the total inductance as seen by the dc current [80], [81]. It takes into account the dc line inductance and the average value of the converter transformer inductance seen by the dc current. As illustrated in Section 5.1, the dc side of a converter is connected in series with two of the transformer inductance branches during normal conduction, one for the positive and one for the negative circuits. The dc side is connected in series with one inductance branch during commutation for the phase that is not involved in commutation. The two phases that are involved in commutation are connected in parallel as shown in Figure 5.5. As a result, the dc side is connected in series with half of a transformer inductance for the circuit involved in commutation. It can be shown that by taking the average over a sixth of the base period and including the dc line inductance, the effective dc inductance is

$$L_{\text{eff}} = L_{\text{dc}} + \left(2 - \frac{3\mu}{2\pi}\right) NL_{\text{T}}. \tag{5.44}$$

The dc voltage developed on the equivalent dc line model by the converters must be adjusted since R_{eff} takes into account the effects of commutation. Eliminating commutation from (5.39)

$$V_{\text{dc}} = \frac{3\sqrt{2}NV_{\text{base,s}}}{\pi} \left|\overline{V}_{\text{ac}}\right| \cos\left(\alpha + \theta - \phi\right). \tag{5.45}$$

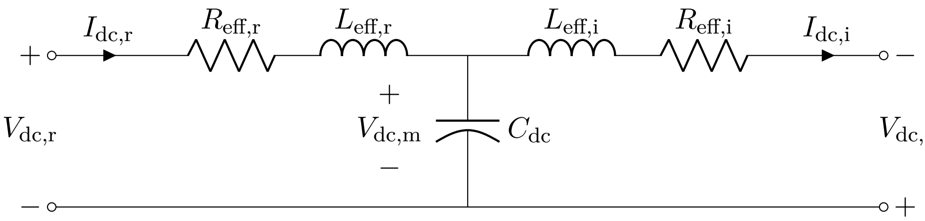

Figure 5.6: Equivalent dc transmission line model for dynamic phasor models

The dc transmission line model has three differential variables and equations. The first variable is the voltage across the dc line capacitor, $V_{\mathrm{dc,m}}$, and its constraint equation is

$$\omega_0 C_{\mathrm{dc}} \dot{V}_{\mathrm{dc,m}} - I_{\mathrm{dc,r}} + I_{\mathrm{dc,i}} = 0. \tag{5.46}$$

The factor of ω_0 scales the differential term by the time base since it must be defined in terms of per unit time to match the remainder of the system's equations. The second and third variables are related to the rectifier and inverter dc currents,

$$\omega_0 L_{\mathrm{eff,r}} \dot{I}_{\mathrm{dc,r}} + R_{\mathrm{eff,r}} I_{\mathrm{dc,r}} - V_{\mathrm{dc,r}} + V_{\mathrm{dc,m}} = 0 \text{ and} \tag{5.47}$$

$$\omega_0 L_{\mathrm{eff,i}} \dot{I}_{\mathrm{dc,i}} + R_{\mathrm{eff,i}} I_{\mathrm{dc,i}} - V_{\mathrm{dc,i}} - V_{\mathrm{dc,m}} = 0. \tag{5.48}$$

5.4 Harmonic Filters

Harmonic filters are included on the ac side of HVdc transmission systems to remove ac current harmonics generated by the converters [78]. Figure 5.7 illustrates the circuit model of the harmonic filters. There are two filters, a low and high frequency filter, along with a capacitor providing reactive power compensation [82]. The filters are simply combinations of the linear branch variations discussed in Section 3.1 and could be modeled by connecting the necessary components. However, in some cases it is easier or advantageous to define internal ac components rather than modify the network model. Therefore, the harmonics filters were modeled internally as part of the HVdc transmission system component to demonstrate this approach.

As mentioned in Section 5.2, $\overline{V}_{\mathrm{ac}}$ is the per unit voltage dynamic phasor of the converter's terminal bus. The high frequency filter shown in Figure 5.7 has a capacitor and an inductor.

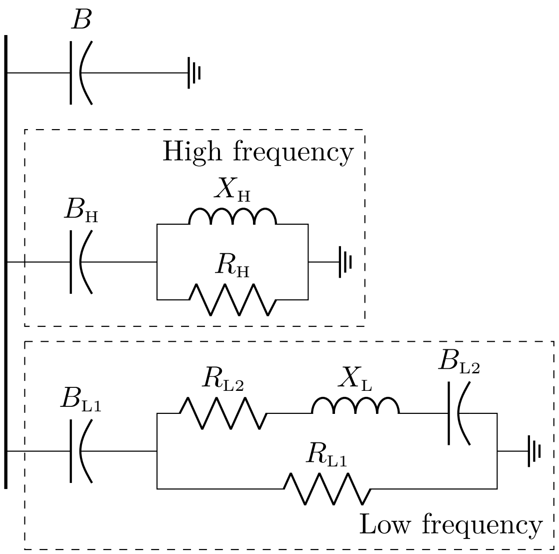

Figure 5.7: Converter ac filters and reactive power support

There is a differential variable and equation associated with each so the high frequency filter contributes two equations

$$X_{\text{H}}\dot{\overline{I}}_{\text{H}} + jX_{\text{H}}\overline{I}_{\text{H}} + \overline{V}_{\text{H}} - \overline{V} = 0 \text{ and} \tag{5.49}$$

$$B_{\text{H}}\dot{\overline{V}}_{\text{H}} + \left(\frac{1}{R_{\text{H}}} + jB_{\text{H}}\right)\overline{V}_{\text{H}} - \overline{I}_{\text{H}} - \frac{\overline{V}_{\text{ac}}}{R_{\text{H}}} = 0. \tag{5.50}$$

The low frequency filter shown in Figure 5.7 has two capacitors and an inductor. Each has a differential variable and equation associated with it so the low frequency filter contributes three equations

$$X_{\text{L}}\dot{\overline{I}}_{\text{L}} + (R_{\text{L2}} + jX_{\text{L}})\,\overline{I}_{\text{L}} + \overline{V}_{\text{L1}} + \overline{V}_{\text{L2}} - \overline{V}_{\text{ac}} = 0, \tag{5.51}$$

$$B_{\text{L2}}\dot{\overline{V}}_{\text{L2}} + jB_{\text{L2}}\overline{V}_{\text{L2}} - \overline{I}_{\text{L}} = 0, \text{ and} \tag{5.52}$$

$$B_{\text{L1}}\dot{\overline{V}}_{\text{L1}} + \left(\frac{1}{R_{\text{L1}}} + jB_{\text{L1}}\right)\overline{V}_{\text{L1}} - \overline{I}_{\text{L}} - \frac{\overline{V}_{\text{ac}}}{R_{\text{L1}}} = 0. \tag{5.53}$$

The total current leaving the converter's ac terminal, $\overline{I}_{\text{F}}$, is equal to the sum of the two filter currents along with the current flowing through the shunt capacitor

$$\overline{I}_{\text{F}} = B\left(\dot{\overline{V}}_{\text{ac}} + j\overline{V}_{\text{ac}}\right) + B_{\text{H}}\left(\dot{\overline{V}}_{\text{H}} + j\overline{V}_{\text{H}}\right) + B_{\text{L1}}\left(\dot{\overline{V}}_{\text{L1}} + j\overline{V}_{\text{L1}}\right) \tag{5.54}$$

(5.54) illustrates that the total ac component current is a linear combination of the filter state variables and converter bus voltage. Therefore, the converter filters are included as coefficients in **A** and **T** in the converter bus's KCL equation.

5.5 Controls

Control system models are also necessary for accurately describing the behaviour of a dc transmission system. The control block system used in generator control modeling described in Section 4.3 was also used to model the dc system's controls. Therefore, dynamic equations are constructed from control blocks specified using control system diagrams.

Figure 5.8 illustrates the rectifier controls, which controls the dc current at the rectifier [78]. The rectifier controls are a simple feedback control system where the rectifier's dc current is the measured value and the current order I_{ord} is computed by the inverter controls. The inverter controls compute the reference value because the rectifier uses a voltage dependent current limit (VDCL) system. The VDCL system reduces the current order when the inverter's dc voltage is low. This action helps protect the inverter against commutation failure and other negative consequences associated with low voltage conditions [1].

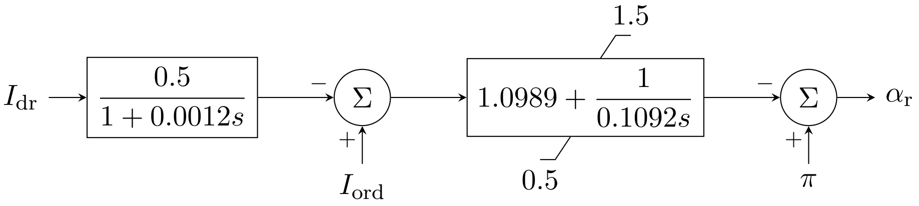

Figure 5.8: Rectifier control system

Figure 5.9 illustrates the inverter control system [78]. The top branch is a dc voltage controller that tries to maintain the inverter's dc voltage at the reference value. The middle branch is the VDCL controller, which computes the current order used by the rectifier controls. The piecewise linear function VDCL(V) used to compute the low voltage limit is illustrated in Figure 5.10. The bottom control loop is a constant current controller, which does not actively control the inverter's dc current during normal operation. This control loop attempts to maintain the inverter's current at a margin of 0.1 pu below the rectifier's current order. The inverter's constant current control system is used to maintain stable operation and allow an HVdc transmission system to find an operating point when the rectifier's voltage is low [1].

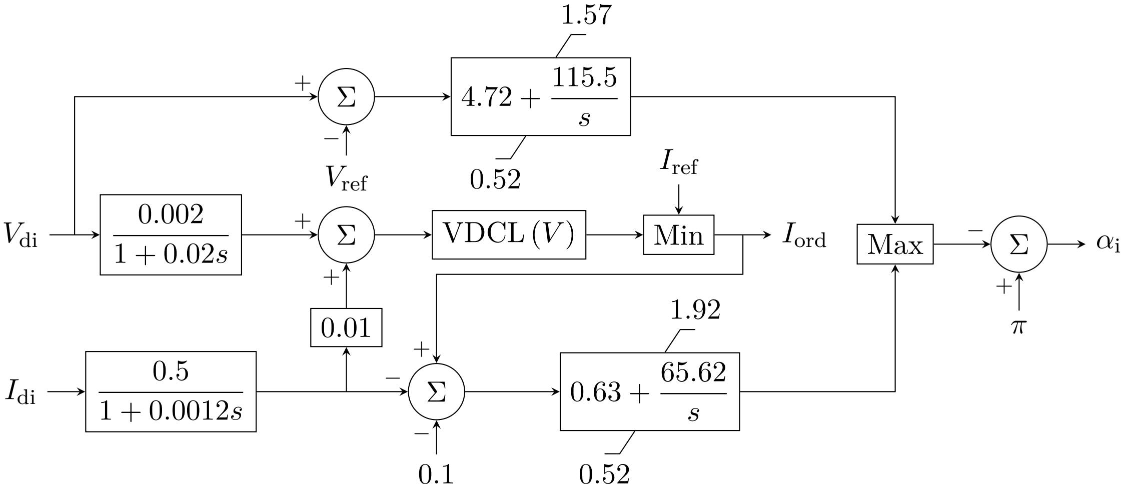

Figure 5.9: Inverter control system

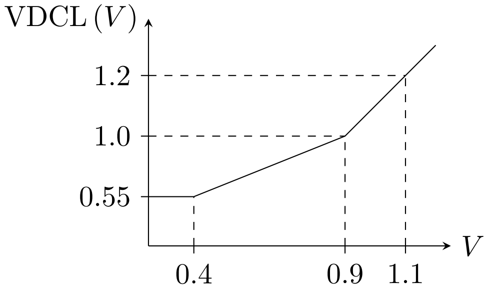

Figure 5.10: Inverter VDCL curve

The VDCL and current control branch match the original CIGRE benchmark system controls. However, a dc voltage controller was used instead of the original extinction angle control system. A dc voltage controller is a common control approach in HVdc systems [1]. Furthermore, it was found that the approximated value for a converter's extinction angle was not well matched with the measured value as produced by the EMT program. A mismatch in the extinction angle is not surprising since the EMT measurement is based on simulated thyristors firing while the dynamic phasor program is a calculation based on the firing and commutation angles computed by the converter model's equations. Therefore, it was determined that a dc voltage controller is a better choice for illustrating the capabilities of the dynamic phasor model.

As illustrated in Section 5.1, a converter's firing angle is measured relative to the zero crossing of its line voltage. This implies that the system angle must be measured to issue firing signals at the correct moments in time. The angle of a balanced set of three-phase sinusoids can be measured using a phase-locked loop (PLL) such as the one shown in Figure 5.11 [83]. The output of a PLL is a sawtooth waveform that linearly increases between 0 and 2π radians over a single cycle of the input waveforms in steady state. The output resets back to zero at 2π radians, which coincides with the positive zero crossing of the a-phase waveform. Therefore, the output is an angle that is synchronized or locked to the instantaneous angle of the a-phase signal. Firing pulses for a thyristor are generated when the output of the converter's PLL is greater than the desired firing angle.

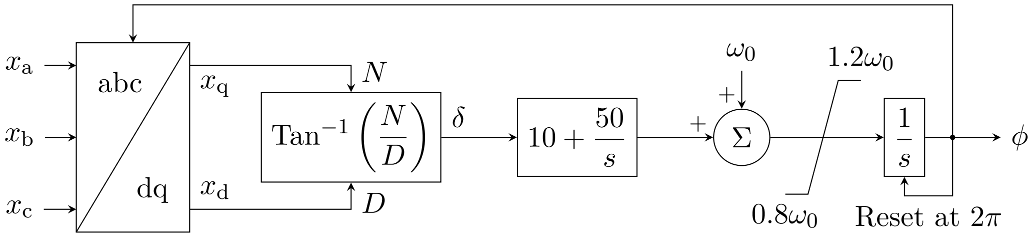

Figure 5.11: Three-phase phase locked loop control system

The PLL illustrated in Figure 5.11 cannot be used in a dynamic phasor-based simulation as it is based on a three-phase input. The angle of a converter's ac voltage does not require measurement in dynamic phasors as it is equal to the argument of the associated dynamic phasor's complex value. However, directly accessing the system angle ignores the dynamics associated with a converter's PLL. It is important include PLL dynamics in converter models as they can have a significant effect on an HVdc system's dynamics [83].

Figure 5.12 illustrates the dynamic phasor equivalent PLL model. The reference frame transformation and inverse tangent blocks in the PLL shown in Figure 5.11 are used to obtain an estimate of the phase angle error, δ. The phase angle error is the difference between angle of x_a, θ, and the estimated angle ϕ. As a result, the transformation and inverse tangent blocks can be replaced by a simple difference between the dynamic phasor's argument and the PLL's output. Furthermore, the dynamic phasor LCC model does not use firing pulses but rather an angle that is constant in steady state. It follows that the integrator input should be zero instead of the system's base frequency once the PLL has synchronized with the input. Therefore, the additional offset of ω_0 is removed and the output limits of the PI controller are shifted by $-\omega_0$.

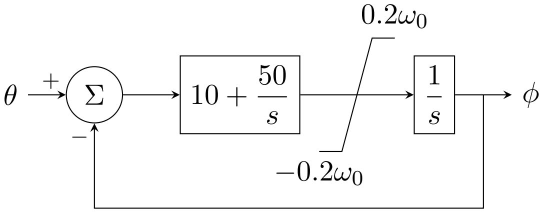

Figure 5.12: Dynamic phasor phase locked loop control system

5.6 Adverse Operating Conditions

The converter model developed in Section 5.2 assumes that the ac and dc systems are operating in steady state and balanced conditions. A model developed with these assumptions also operates well during significant disturbances. In these situations, a thyristor-based HVdc system model remains balanced and the thyristors continue to fire in the correct order with an acceptable commutation pattern. However, it is possible for the converter to enter into states which are both unbalanced and significantly dependent on the converter's switching characteristics. These states cannot be represented exactly using the dynamic phasor converter model. Dynamic phasor models attempt to capture approximate behaviour of unbalanced situations even with significant modifications [84]. Therefore, the following adjustments to the converter model strive to represent approximate converter behaviour during abnormal operation.

Adverse operating conditions are related to the commutation angle expression (5.19). This expression includes an inverse cosine term whose domain is $[-1, 1]$. It is possible to show that the domain of the inverse cosine function implies two bounds on its argument

$$\cos\left(\alpha + \theta - \phi\right) - \frac{2X_{\mathrm{T}}I_{\mathrm{dc}}}{\sqrt{3}V_{\mathrm{m}}} \leq 1 \text{ and} \tag{5.55}$$

$$\cos\left(\alpha + \theta - \phi\right) - \frac{2X_{\mathrm{T}}I_{\mathrm{dc}}}{\sqrt{3}V_{\mathrm{m}}} \geq -1. \tag{5.56}$$

The simulation may fail or it may continually decrease the time step until it produces an infinite loop due to the error control algorithms if these conditions are not satisfied. Therefore, a converter model with the capacity to model transient stability disturbances must guarantee that conditions (5.55) and (5.56) are met at all times.

The first condition (5.55) is related to converter shut down, which is discussed in Section 5.6.1. The second condition (5.56) is related to commutation failure, which is discussed in Section 5.6.2.

5.6.1 Converter Shut Down

A six-pulse converter bridge can only carry current in a single direction. However, it is possible for the dc system to exert forces on the dc current that attempt to reverse it during disturbances. A thyristor-based model naturally prevents the dc current from reversing since the thyristor models do not permit current to flow in the reverse direction. The dynamic phasor model developed in Section 5.2 does not inherently prevent the dc current from reversing so the condition must be inhibited manually.

The converter model must be equipped with an event that monitors the condition

$$I_{\mathrm{dc}} \geq 0 \tag{5.57}$$

and is triggered on its negative zero crossing. The converter must transition from normal condition to a shut down state when the event associated with (5.57) is triggered. In this state, the converter is no longer conducting current and the dc line equation associated with the converter's dc current must be adjusted to hold the dc current at zero. As in the case of the single pole control block, this model restricts movement of the dc current variable using its derivative

$$\dot{I}_{\mathrm{dc}} = 0. \tag{5.58}$$

It is possible to show using the commutation angle expression (5.19) that μ is zero when I_{dc} is zero. It follows from the ac current expression (5.42) that $\overline{I}_{\mathrm{ac}}$ is zero when both I_{dc} and μ are equal to zero. No additional equations need to be adjusted since I_{dc} equal to zero is naturally propagated to the dc line's capacitor equation.

Current begins to flow again when the conditions in the ac and dc system allow for conduction to occur in the correct direction. This behaviour cannot be modeled in a dynamic phasor converter model since there are no thyristors or ac voltages and currents. However, it is possible to monitor the conditions in the dc line and transition back to the normal conduction state when the dc line is trying to pull the current in the forward direction. In the case of the dc line model discussed in Section 5.3, this is equivalent to monitoring the series inductor voltage and trigger the appropriate state change when it becomes positive. This event can be modeled by the inequality

$$V_{\mathrm{dc,r}} - V_{\mathrm{dc,m}} \leq 0 \tag{5.59}$$

on the rectifier side where $V_{\mathrm{dc,r}}$ is computed using (5.45). (5.59) is triggered on its positive zero crossing.

The first condition (5.55) can be rearranged to produce a constraint on the dc current

$$I_{\mathrm{dc}} \geq \frac{\sqrt{3}V_{\mathrm{m}}}{2X_{\mathrm{T}}}\left(\cos\left(\alpha+\theta-\phi\right)-1\right). \tag{5.60}$$

The right hand side this condition is never greater than zero. This observation implies that the converter current reversal condition (5.57) imposes a stricter condition on the dc current. Therefore, (5.55) is guaranteed to be satisfied by converter models that include the current reversal logic described in this section.

5.6.2 Commutation Failure

A converter's extinction angle represents the interval between when a thyristor has ceased conduction and the next time its terminals must block a positive voltage [79]. Extinction angle, γ, is related to firing and commutation angles through the expression

$$\alpha + \mu + \gamma = \pi \tag{5.61}$$

during normal operating conditions. (5.61) illustrates that extinction angle decreases as firing angle increases and is smallest for converters operating as inverters. The extinction angle must not drop below some nonzero limit for practical thyristors, which is the interval required to allow them to fully de-ionize [1]. It is possible for a thyristor to begin conduction again without its firing signal present if its minimum angle condition is not met. This situation is known as commutation failure since the dc current does not successfully transition from the off-going phase to the on-going phase.

It is possible to show that modeling commutation failure ensures (5.56) is satisfied at all times. Let β be an angle defined such that

$$\pi - \beta = \cos^{-1}\left(\cos\left(\alpha + \theta - \phi\right) - \frac{2X_{\mathrm{T}}I_{\mathrm{dc}}}{\sqrt{3}V_{\mathrm{m}}}\right). \tag{5.62}$$

Substituting (5.62) into the commutation angle expression (5.19)

$$\alpha + \mu + \theta - \phi + \beta = \pi. \tag{5.63}$$

The system angle θ is equal to the measured angle ϕ in steady state, which reduces (5.63) to

$$\alpha + \mu + \beta = \pi. \tag{5.64}$$

Comparing (5.61) and (5.64) shows that β is equal to γ in steady state. Let the minimum extinction angle of a six-pulse bridge be γ_{min}. Commutation failure is approximated in the converter model using the condition

$$\beta \geq \gamma_{\mathrm{min}}. \tag{5.65}$$

The converter is assumed to be in the commutation failure state when (5.65) is not satisfied. Substituting (5.62) into (5.65) and rearranging terms

$$\cos^{-1}\left(\cos\left(\alpha + \theta - \phi\right) - \frac{2X_{\mathrm{T}}I_{\mathrm{dc}}}{\sqrt{3}V_{\mathrm{m}}}\right) \leq \pi - \gamma_{\mathrm{min}}. \tag{5.66}$$

It is possible to take the cosine of both sides of (5.66) and form a new commutation failure condition

$$\cos\left(\alpha + \theta - \phi\right) - \frac{2X_{\mathrm{T}}I_{\mathrm{dc}}}{\sqrt{3}V_{\mathrm{m}}} \geq -\cos\left(\gamma_{\mathrm{min}}\right) \tag{5.67}$$

because both sides of (5.66) are constrained to the range $[0, \pi]$. It follows that the converter can detect that commutation failure has occurred and change its state when an event monitoring (5.67) is triggered. If the thyristors are assumed to be ideal, then γ_{min} is equal to zero and

(5.67) simplifies to

$$\cos(\alpha + \theta - \phi) - \frac{2X_{\mathrm{T}}I_{\mathrm{dc}}}{\sqrt{3}V_{\mathrm{m}}} \geq -1. \tag{5.68}$$

(5.68) is generally used in dynamic phasor models to detect commutation failure [81]. Furthermore, the ideal case demonstrates that condition (5.56) is always satisfied by a converter that models commutation failure using (5.67) since

$$\cos(\gamma_{\mathrm{min}}) \leq 1. \tag{5.69}$$

Figure 5.13 illustrates the six-pulse converter during commutation failure [81]. In this case, Q_1 fails to take over conduction from Q_5 in the positive circuit. The next thyristor in the firing sequence is Q_2, which takes over conduction from Q_6. A short appears across the dc terminals because Q_2 and Q_5 are part of the same phase. This sequence of events will occur regardless of the phases involved since the off-going phase in one commutation period is always the on-going phase in the next commutation period.

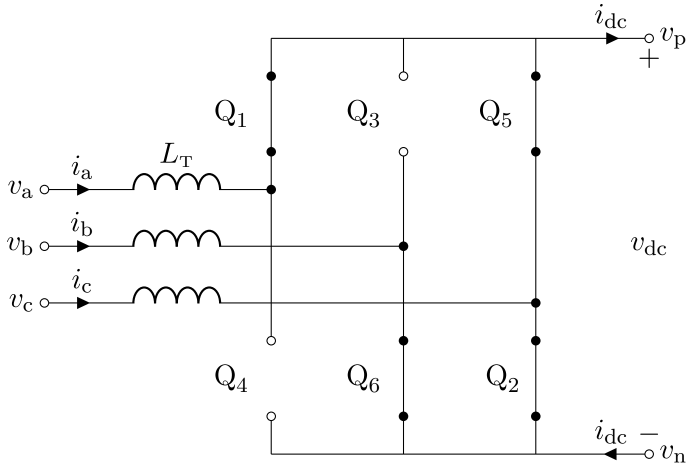

Figure 5.13: Commutation failure of Q_5 into Q_1 and subsequent commutation of Q_6 into Q_2

Figure 5.13 illustrates that commutation failure is by its nature unbalanced. Therefore, commutation failure cannot be adequately represented by a three phase balanced dynamic phasor model [81]. However, it is possible to capture the critical aspects of commutation failure and represent them in a dynamic phasor model even without modeling unbalanced ac systems.

The dc side of a converter quickly becomes a short circuit during commutation failure. This scenario does not require any changes to the dc line's differential equations. The only change required is that the dc voltage of the converter is set to zero [81]. The converter's topology as seen by the ac system during commutation failure is unbalanced and irregular so its behaviour must be approximated. A potential solution is to assume that $\overline{I}_{\mathrm{ac}}$ is equal to zero during commutation failure [81]. However, this approach did not lead to satisfactory results in the MNA program. A converter model using this approximation would transition back and forth between the normal and commutation failure states before finally settling on its new state. This behaviour was likely the result of the discontinuity in the ac system when $\overline{I}_{\mathrm{ac}}$ to suddenly set to zero.

The approach used in this converter model was to calculate $\overline{I}_{\mathrm{ac}}$ according to the normal conduction expression (5.42) and adjust how μ is calculated. The commutation angle cannot be accurately defined during commutation failure so it is assumed that μ is calculated according to

$$\mu = \pi - \alpha - \gamma_{\mathrm{min}}. \tag{5.70}$$

This expression enforces continuity at the event boundary allowing the model to make a smooth transition from one state to another.

The same condition (5.67) is used to transition from the commutation failure state back to normal conduction. However, the event that triggers the transition must monitor the condition in the opposite direction. Furthermore, numerical hysteresis must be introduced into the condition using the approach discussed in Section 4.3 to ensure that the simulation does not oscillate at the boundary between states.

5.7 Simulation Results

The CIGRE benchmark case illustrated in Figure 5.1 was modeled using the dynamic phasor and EMT simulators. A 0.1 s fault with a resistance of 0.004 pu was simulated on the rectifier bus to test the performance of the benchmark case. Figure 5.14 illustrates the dc current at the rectifier for two seconds following the disturbance. The results illustrate that this disturbance is severe enough to cause the dc current to decay completely and force the rectifier to shut down. The two methods are generally in good agreement and the dynamic phasor model accurately captures the behaviour of the rectifier current. There is a small mismatch during the fault and in the recovery phase of the current. The current simulated by the dynamic phasor program drops to a slightly lower value as compared to the EMT results. Consequently, the dynamic phasor converter model experiences several shut down and recovery cycles whereas the EMT program mostly remains in normal conduction mode. The dynamic phasor results illustrate a faster recovery phase following the disturbance. A possible source of this discrepancy is the impedance used to model the rectifier's ac system. The dynamic phasor converter model takes into account the effect of the converter transformer. However, the ac system is represented only by the converter's ac bus voltage. This method takes into account the dynamics of the ac bus voltage but it does not include the effect of any system inductance on the converter's ac current. Therefore, there might be a mismatch in the effective inductance as seen by the HVdc system, which would have an impact on the dc current dynamics.

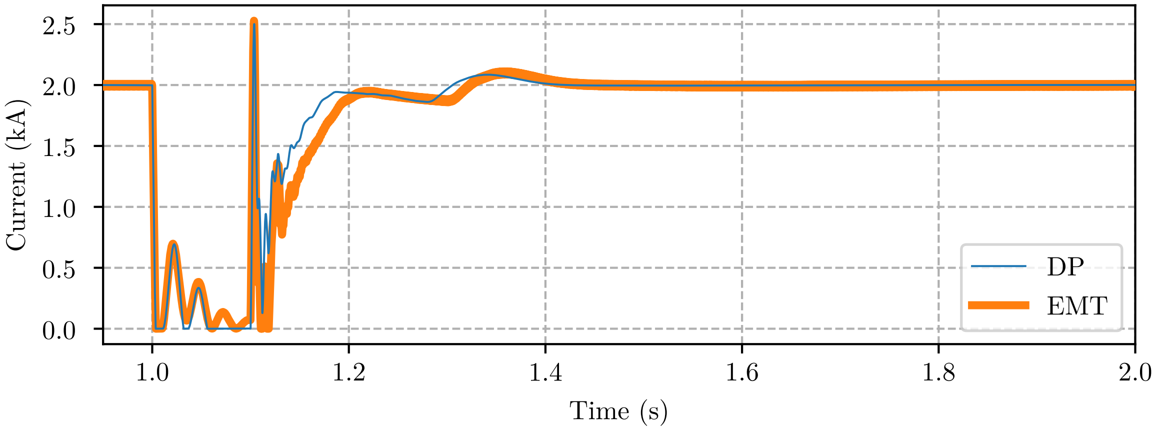

Figure 5.14: Rectifier dc current for a fault on the rectifier ac bus

Figure 5.15 illustrates the dc current at the inverter for two seconds following the disturbance. The results show similar behaviour as the rectifier dc current but less severe due to the transmission line. The inverter also transitions into the shut down state, which it recovers from once the disturbance is removed. The recovery profile of the inverter is also slightly faster in the dynamic phasor simulation as compared to the EMT simulation. However, despite this difference, the dynamic phasor model accurately captures the overall behaviour of the inverter's dc current.

Figure 5.16 illustrates the voltage across the dc line's capacitor for two seconds following

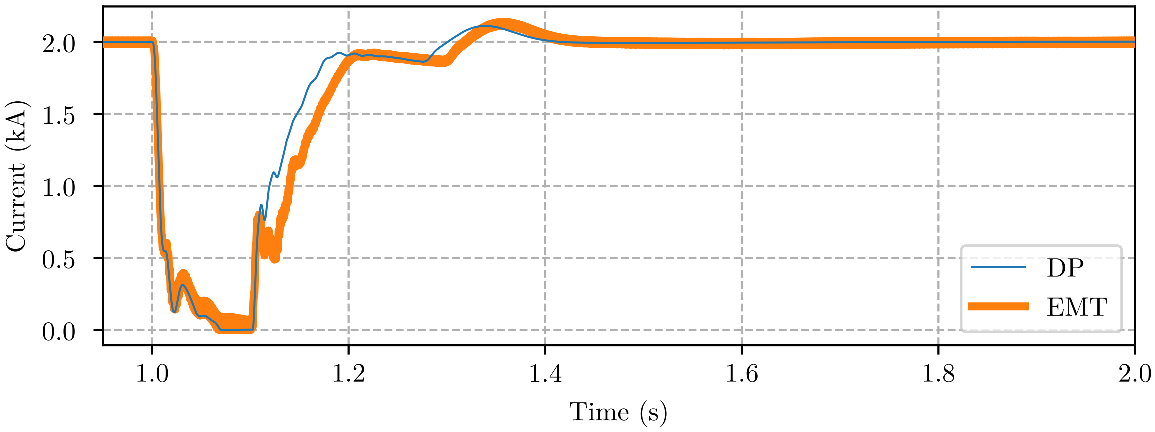

Figure 5.15: Inverter dc current for a fault on the rectifier ac bus

the disturbance. The results illustrate excellent agreement between the two programs for the portion during the disturbance. The differences in the recovery are more pronounced in the capacitor voltage, where the EMT results indicate a larger decrease than the dynamic phasor results after the disturbance is removed.

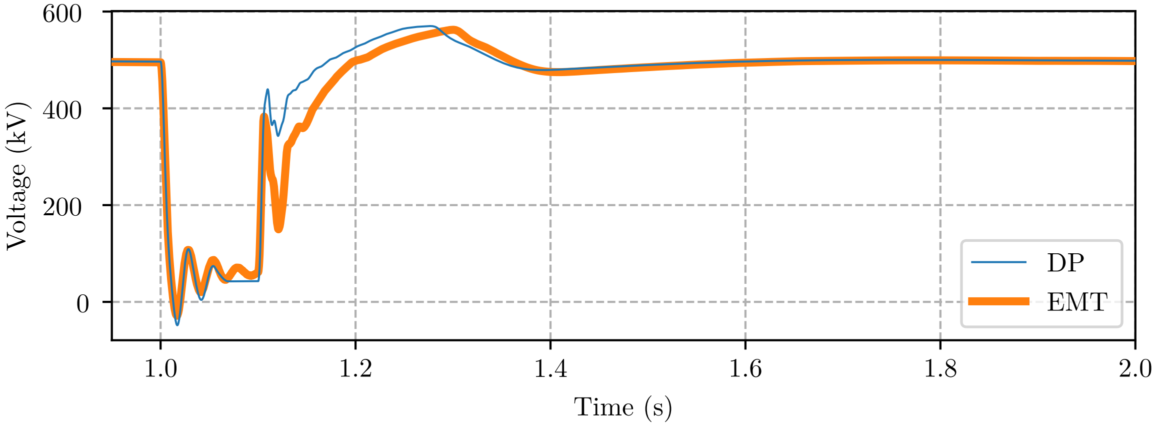

Figure 5.16: Voltage across the dc line's capacitor for a fault on the rectifier ac bus

Figure 5.17 illustrates the rectifier ac bus voltage near the moment where the disturbance is removed. The results illustrate good agreement between the DP and EMT programs. The EMT peaks are slightly larger following the disturbance but the overall behaviour of the ac voltage is well captured by the DP results. Figure 5.18 illustrates the rectifier's ac current in the same region as the ac voltage. The dynamic phasor program appears to recover at a slightly faster rate as compared to the EMT results, which matches with the dc current and voltage observations. However, the two programs are in good agreement after a few cycles.

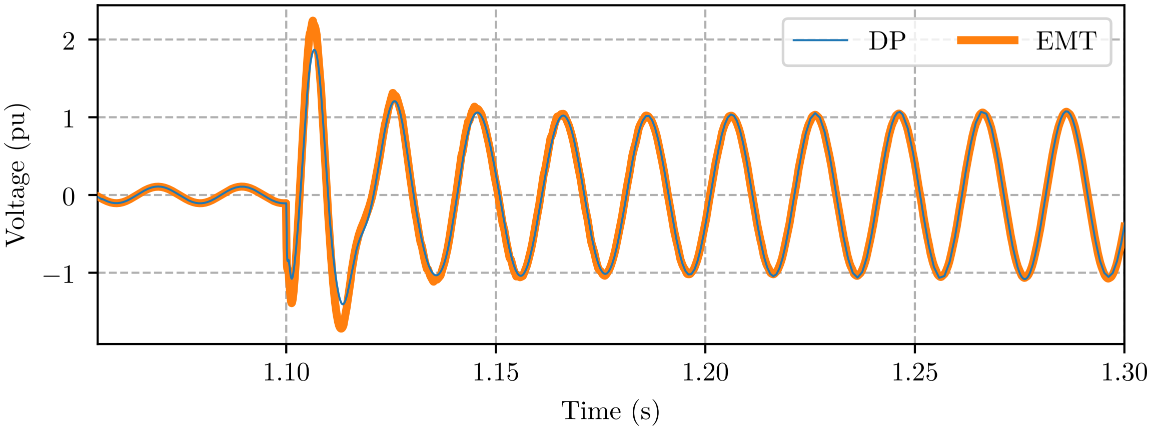

Figure 5.17: Rectifier ac voltage for a fault on the rectifier ac bus

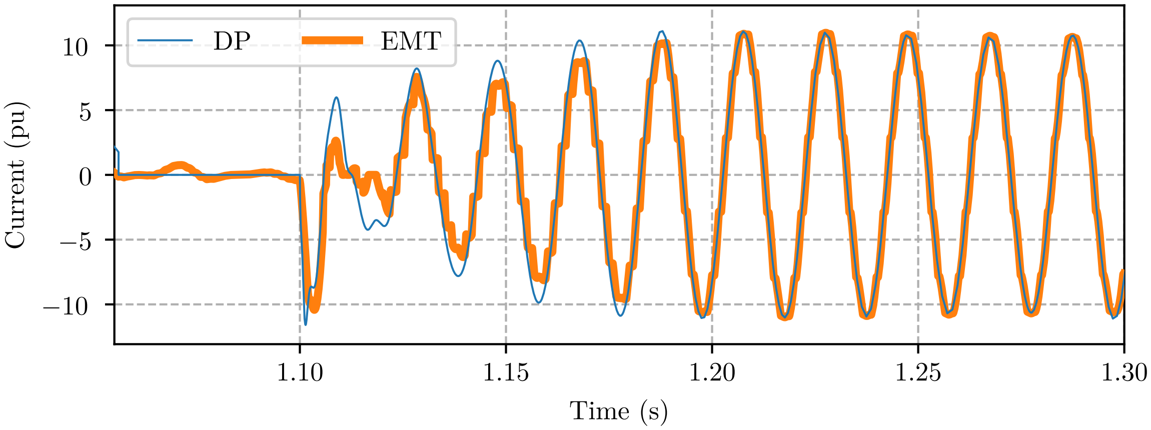

Figure 5.18: Rectifier ac current for a fault on the rectifier ac bus

Chapter 6

Simulation Results

This chapter will demonstrate several sets of simulation results comparing the dynamic phasor (DP) program developed in this work to commercial electromagentic transient (EMT) and transient stability (TS) simulators. The first set of simulation results are from the IEEE 68 bus test system, which is discussed in Section 6.1. The purpose of this section is to illustrate results obtained from a small multimachine system as a benchmark for comparison of the three simulation programs. They will demonstrate that the DP program maintains a high degree of accuracy for multimachine systems using similar measurements to the single machine infinite bus case. Several versions of a 500 bus test system are presented in Section 6.2. The first is the original network modeled using only ac equipment, which is used as a benchmark for this case. This network was augmented to include two HVdc transmission systems, which replace two ac transmission lines that were found to carry a significant amount of power. This network will demonstrate the capability of the method in simulating ac systems including HVdc transmission lines. The third version of this network includes a multimass turbine modeled on a generator near one of the rectifiers. The corresponding rectifier was tuned to induce subsynchronous oscillations to demonstrate the capacity of the DP method to model unstable torsional oscillations. Finally, simulation results from a 2000 bus test system is included in Section 6.5 to demonstrate the scalability of the DP method.

Table 6.1 provides a summary of the CPU timing results for the simulation of a short circuit fault in the dynamic phasor and EMT programs. These results were collected during the simulation experiments and are further discussed in the following sections. CPU time in the dynamic phasor program can be sensitive to fault location as it uses a variable step algorithm. Therefore, faults were carried out at every bus in the IEEE68, ACTIVSg500, and ACTIVSg2000 cases. The dynamic phasor results in Table 6.1 summarize this data with the best, worst, and average CPU times. EMT simulations were carried out for the IEEE68 and ACTIVSg500 cases, so values for the best, worst, and average speed-up are also provided for these cases. A single run of the ACTIVSg500-HVdc and ACTIVSg500-SSTI was carried out in this work, so the dynamic phasor CPU time and speed up for these cases is included in the best column. The results show that the dynamic phasor program is generally at least 100 times faster than the EMT program. The ACTIVSg2000 could not be compared against EMT simulation due to resource constraints. However, the results show that the slowest dynamic phasor simulation of the ACTIVSg2000 case is faster than the EMT simulation of the IEEE68 bus network, which is 25 times smaller than the ACTIVSg2000 case.

Table 6.1: Summary of CPU timing results and speed up

Case	**CPU Time (s)**				**Speed Up**		
	EMT	**Best**	**Worst**	**Average**	**Best**	**Worst**	**Average**
IEEE68	130	0.14	1.14	0.46	949	114	283
ACTIVSg500	570	1.3	10	3.4	431	57	167
ACTIVSg500-HVdc	1422	4.6	–	–	309	–	–
ACTIVSg500-SSTI	2088	20.9	–	–	100	–	–
ACTIVSg2000	–	6.4	117.2	23.6	–	–	–

6.1 IEEE68

The first case considered is the IEEE68 network, which consists of 68 buses and sixteen generators. Fiften generators are modeled using round rotor generator models and type four ac exciters and one is modeled as a voltage source to provide a frequency reference for the system. The MNA equations for this system includes 535 variables where 92.1 % are differential. The Jacobian for this model has 2159 nonzeros, which means it is only 0.75 % filled.

The location of a disturbance can have a significant impact on the DP program's performance since modes excited and events triggered by the disturbance vary throughout a network model. A three-phase short circuit fault was applied at each bus in the network and CPU timing information was gathered during each simulation. The fault is applied at 1 s for a duration of 0.1 s and the simulation is run for 20 s for each bus. This approach was used to examine how much variation there is in the DP program's performance. It also provides a more complete picture of how the DP method's performance compares to commercial simulators.

Figure 6.1 illustrates the distribution of CPU times recorded for bus faults in the IEEE68 network. The CPU times are presented as both a distribution and a cumulative line. The distribution is normalized such that the height of the tallest bin is equal to one and the cumulative line is also normalized such that its final value is one. The results demonstrate that the CPU times for disturbances in this network fall in the range of 0.14 s to 1.14 s indicating that nearly an order of magnitude separates the slowest and fastest simulations. However, the CPU times are well clustered between 0.1 s and 0.65 s with 75 % of all simulations requiring less than 0.55 s of CPU time.

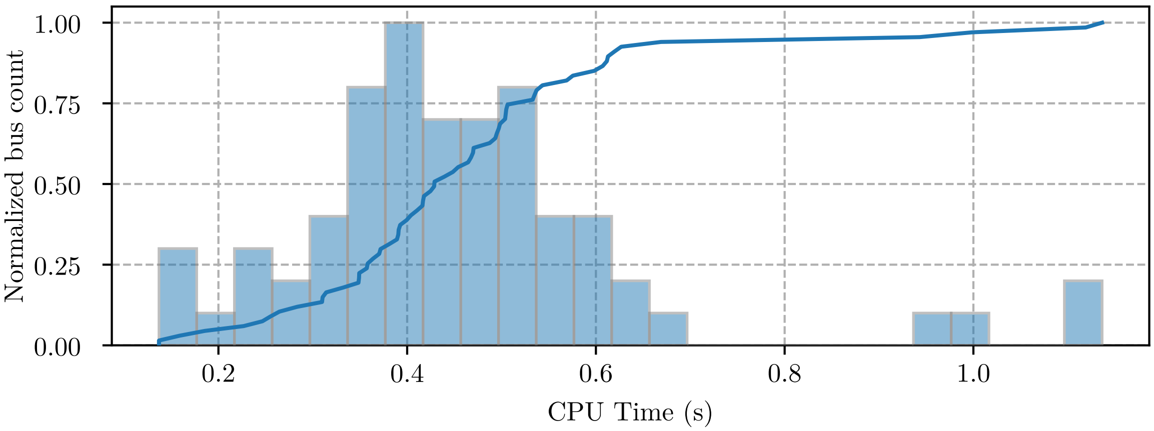

Figure 6.1: Distribution of CPU time required by bus fault simulations in the IEEE68 network

A fault at bus 63 was selected to compare the DP, TS, and EMT simulation methods. Bus 63 is a non-generator bus and with a differential voltage variable. Figure 6.2 illustrates a comparison of the instantaneous voltage at bus 63 where the DP and TS results have been reconstructed according to (2.8). The results show that the DP and EMT programs both capture the post-fault high frequency network oscillations while the TS program models the average value of the bus voltage. Figure 6.3 illustrates the first cycle following the disturbance, which appears to show that the DP and EMT oscillations are not well matched. Analysis of the DP simulation results shows that it selected an integration time step near 1 microsecond for this portion of the simulation so the EMT simulation was repeated with a time step of 10 µs.

Figure 6.4 illustrates the same time frame and EMT results obtained using a time step of 10 µs, which demonstrates excellent agreement between the DP and EMT simulators. However, it is important to note that this additional simulation was conducted to illustrate that the DP program maintains a high level of accuracy when modeling network transients in multimachine systems. These oscillations have a frequency of 3.2 kHz and in practice are not a concern for subsynchronous studies. There are many factors such as traveling wave transmission line models that impact these transients in more realistic power system models.

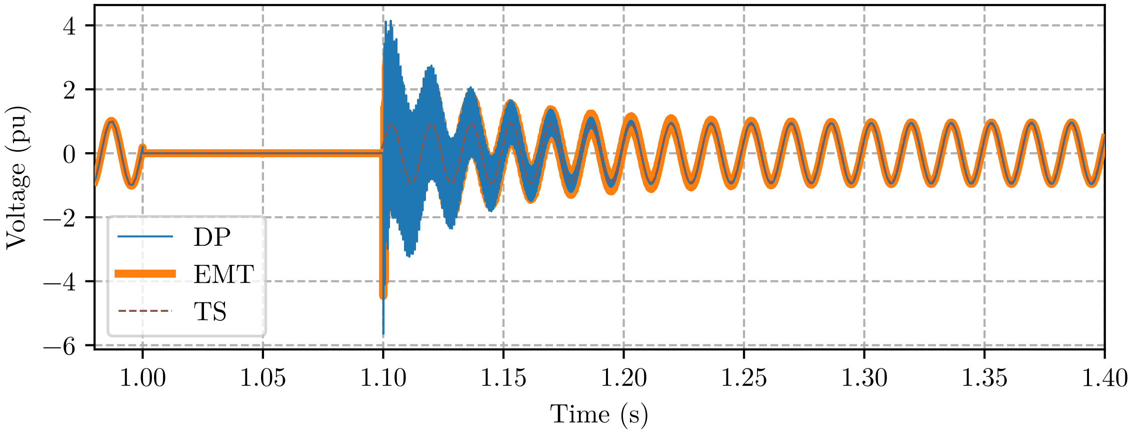

Figure 6.2: Bus 63 voltage response to a fault on bus 63 in the IEEE68 network – first 0.4 s following the disturbance

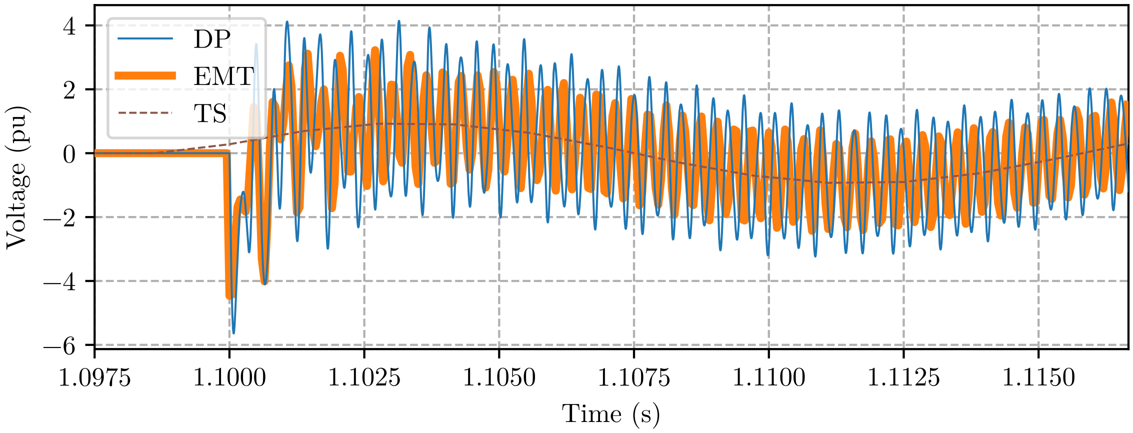

Figure 6.3: Bus 63 voltage response to a fault on bus 63 in the IEEE68 network – first cycle following the disturbance

The generator at bus 2 was selected for comparison between the three programs as it is closest to the disturbance. Figure 6.5 illustrates the response of the stator current produced

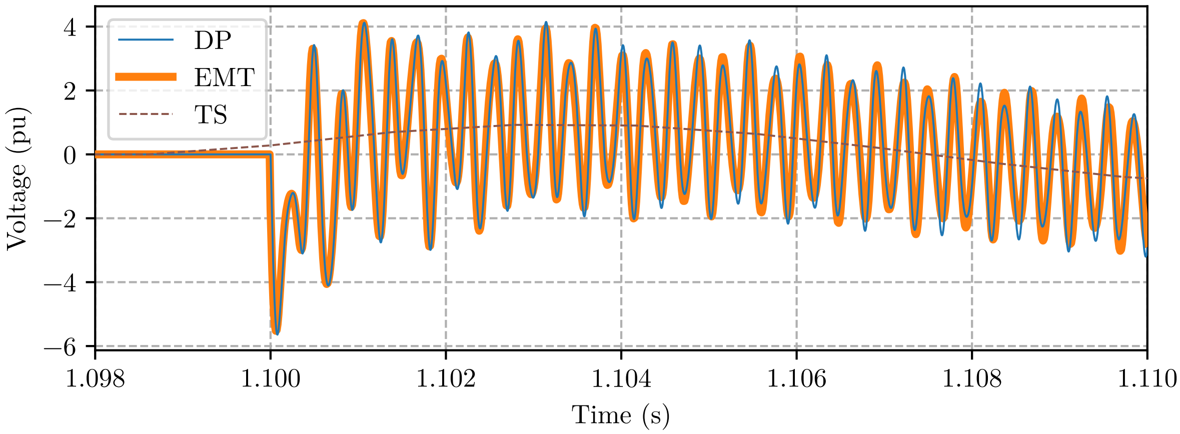

Figure 6.4: Bus 63 voltage response to a fault on bus 63 in the IEEE68 network – first cycle following the disturbance and an EMT time step of 10 µs

by generator 2 for the first 2 s following the disturbance. These results illustrate that all three programs are in good agreement for the system's electromechanical characteristics, which appear as oscillations near 1 Hz in the envelope of the stator current. Figure 6.6 illustrates the stator current for the first 0.25 s following the disturbance, which illustrates the amplitude variations and dc offset nearest the fault. The DP and EMT programs are in excellent agreement for these dynamics, whereas the TS program only captures the low frequency amplitude fluctuations and completely neglects the dc offset.

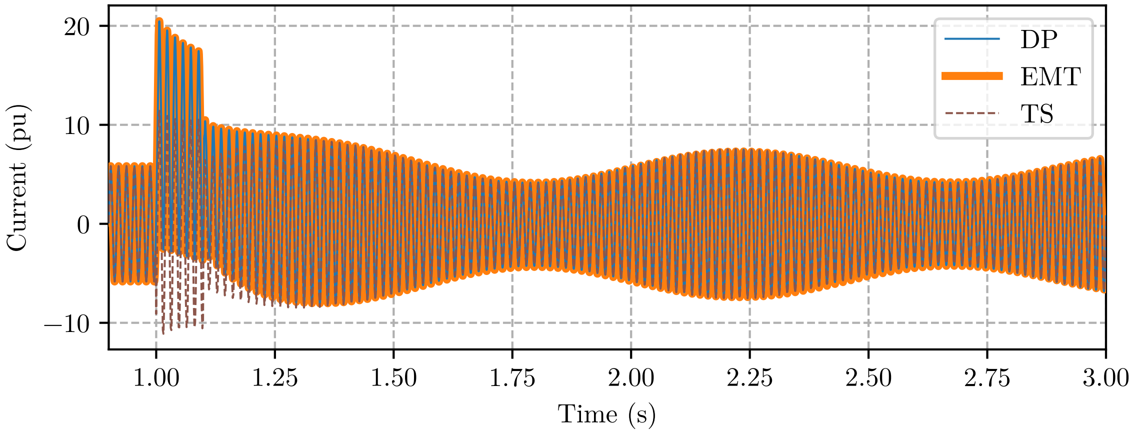

Figure 6.5: Generator 2 stator current response to a fault on bus 63 in the IEEE68 network – first 2 s following the disturbance

Figure 6.7 illustrates the response of the air-gap torque in generator 2 for the entire simulation. These results show that the three simulation programs are in excellent agreement for

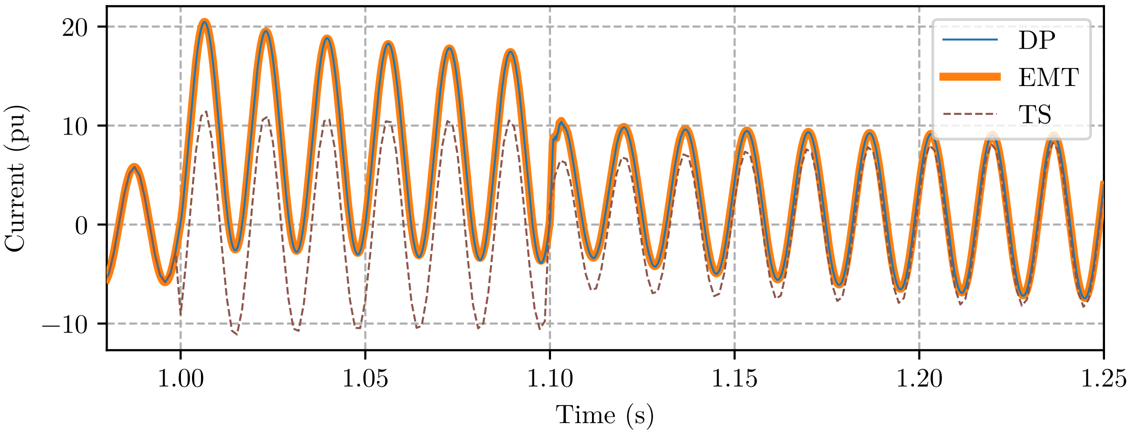

Figure 6.6: Generator 2 stator current response to a fault on bus 63 in the IEEE68 network – first 0.25 s following the disturbance

the electromechanical transients near 1 Hz. Figure 6.8 illustrates the air-gap torque for the first 0.25 s following the disturbance. This plot illustrates that the DP and EMT programs are in excellent agreement for the 60 Hz stator transients that are ignored by the TS program. However, if the average value of the air-gap torque is considered then the three programs are in good agreement.

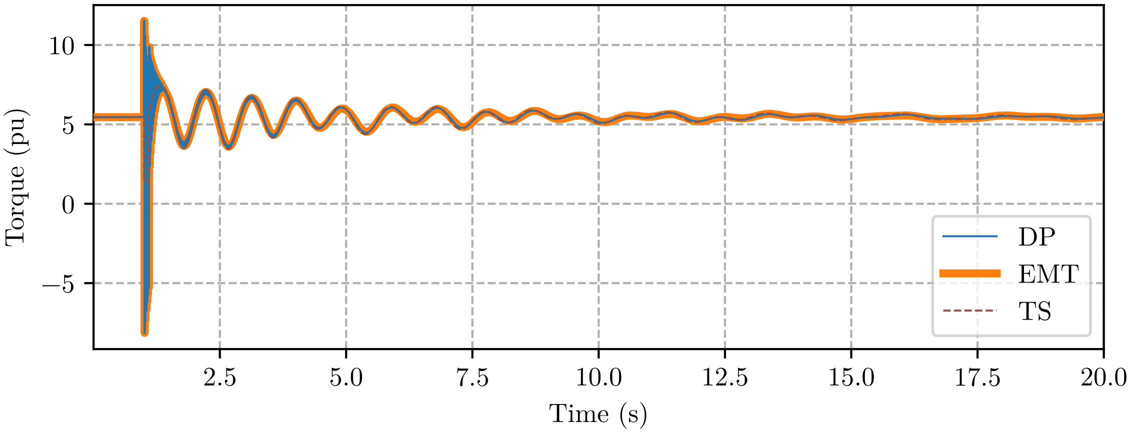

Figure 6.7: Generator 2 air-gap torque response to a fault on bus 63 in the IEEE68 network – entire simulation

Figure 6.9 illustrates the response of the rotor speed of generator 2 for the entire simulation. The frequency of the rotor speed oscillations simulated by the three programs is in good agreement, but there are some differences in the damping of these oscillations. These differences become more pronounced as the simulation continues. Well matched frequencies and slight

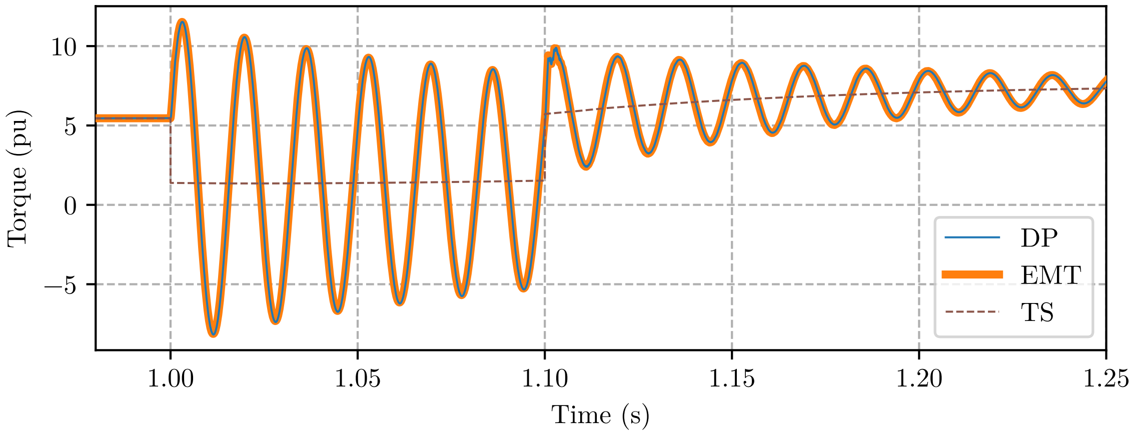

Figure 6.8: Generator 2 air-gap torque response to a fault on bus 63 in the IEEE68 network – first 0.25 s following the disturbance

damping mismatches in rotor speed oscillations was frequently observed when comparing the three simulation programs. Further investigation revealed that the damping of rotor speed oscillations is sensitive to stator resistance. It follows that rotor speed oscillations would be sensitive to numerical interfacing resistances like the ones used in EMT-type simulators [15]. However, the mismatches were generally found to be small like the ones observed in Figure 6.9. Figure 6.10 illustrates the rotor speed dynamics for the first 0.6 s following the disturbance. The results illustrate excellent agreement between the DP and EMT simulators including the 60 Hz oscillations that arise from stator transients present in the air-gap torque.

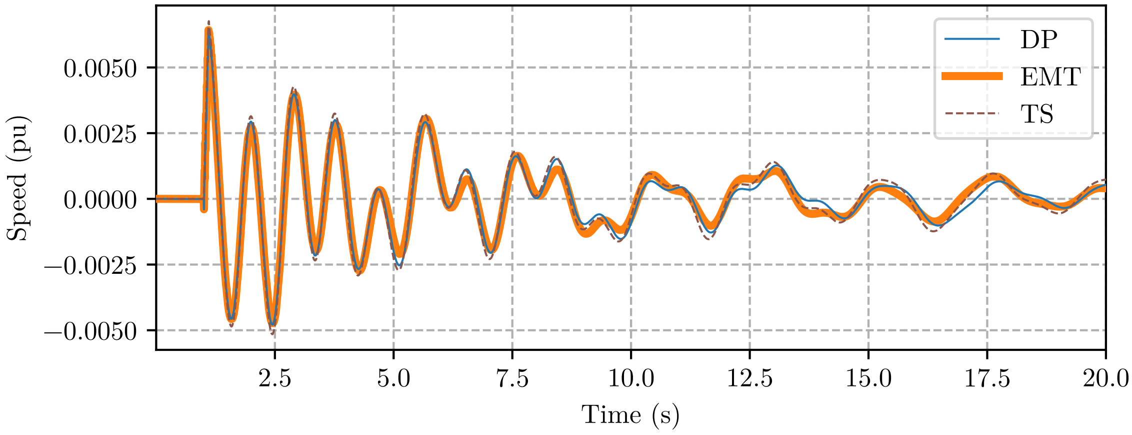

Figure 6.9: Generator 2 rotor speed response to a fault on bus 63 in the IEEE68 network – entire simulation

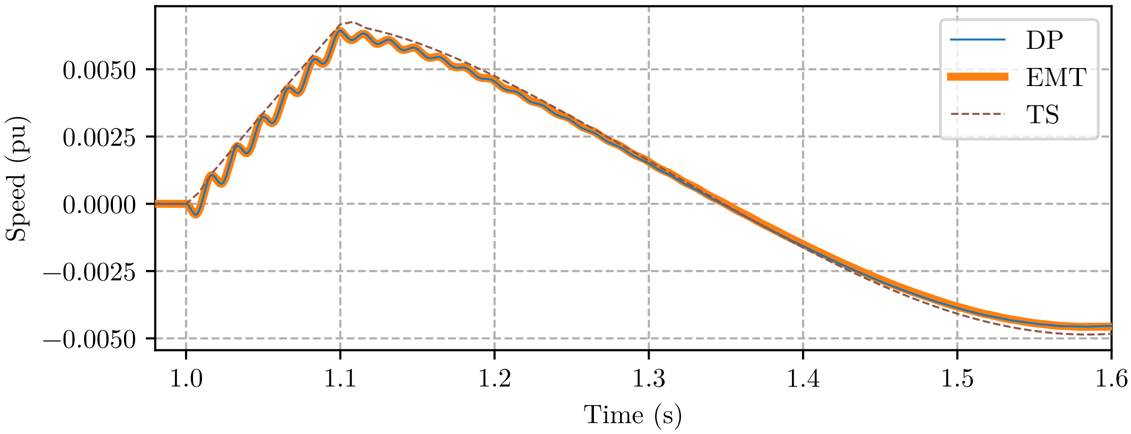

Figure 6.10: Generator 2 rotor speed response to a fault on bus63 in the IEEE68 network – first 0.6 s following the disturbance

6.2 ACTIVSg500

The second system that was used to validate the DP method is a five hundred bus system known as the ACTIVSg500 network [85], [86]. This network model is a synthetic approximation of the South Carolina power grid and is freely available for research [87]. Two modifications were made to the available data for this work. The first modification is that all generators sharing a common high voltage bus were combined into a single aggregate generator. The second modification is that the dynamic data was adjusted to use a fixed set of generator, exciter, turbine, and power system stabilizer data. A random model from each category was assigned to all but one generator in the network. The generator located at the swing bus was modeled using a voltage source to provide the network with a frequency reference.

The modified ACTIVSg500 power grid model used in this section contains 441 buses and 31 generators. The size of the MNA system of equations for this model is 2996. In this model, 2592 or 86.5 % of the variables are differential. The MNA equations for this network include 2996 variables, of which 2592 or 86.5 % are differential. The Jacobian associated with the MNA equations for this system has 11766 nonzero elements, which represents only 0.13 % of the total elements.

Figure 6.11 illustrates the distribution of CPU times recorded for bus faults in the ACTIVSg500 network. The results demonstrate that the CPU times for disturbances in this network fall in the range of 1.3 s to 10.1 s. As in the IEEE68 bus case, nearly an order of magnitude separates the slowest and fastest simulations. However, the CPU times are well clustered between 1 s and 4 s with 75 % of all simulations requiring less than 4 s of CPU time.

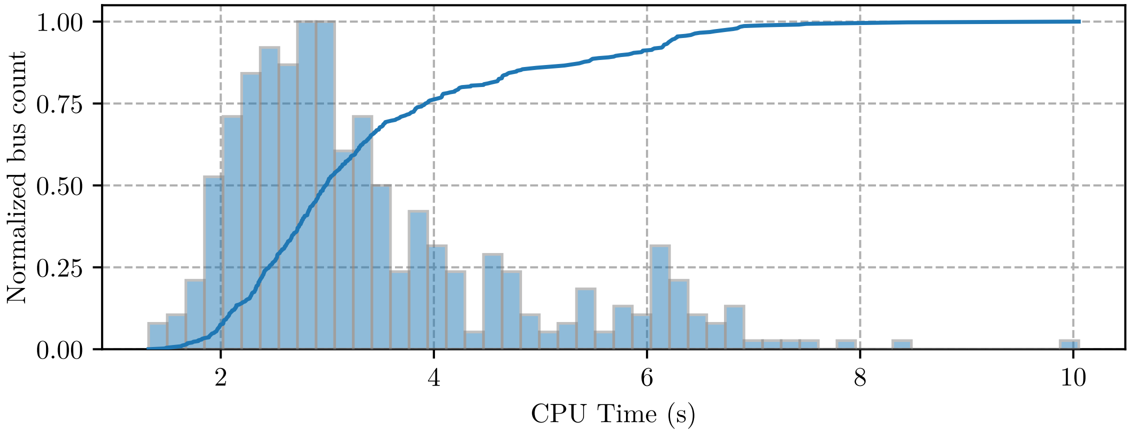

Figure 6.11: Distribution of CPU time required by bus fault simulations in the ACTIVSg500 network

A fault at bus 297 was selected to compare simulation results between the DP program and commercial EMT and TS simulators. Bus 297 is a load bus with a differential voltage variable and the DP program requires 6.7 s of CPU time to simulate this disturbance. The EMT simulator was run on the same hardware and it was found that it requires 570 s of CPU time to carry out a 20 s simulation of this case. Therefore, the DP program is 85 times faster than the EMT simulator for this disturbance. Furthermore, the CPU time required for EMT bus fault simulations should not vary considerably as it uses a fixed step integrator. Speed-up factors can be reasonably estimated using the same CPU time for all bus faults. It follows that the DP program is between 57 and 431 times faster than the EMT simulator for this case and 75 % of cases are 143 times faster than EMT simulation.

Figure 6.12 illustrates the response of bus 297's voltage for 3 cycles following the disturbance. These results illustrate good agreement between the DP and EMT programs for network transients. In particular, the DP program demonstrates the capability of accurately predicting post-fault over voltages. Study of these dynamics is generally reserved for EMT programs as conventional TS program neglect them. The results illustrate good agreement between the three programs considering that the TS results capture the average value, as expected.

The generators at buses 82 and 442 were selected for comparison as they are the closest generators to the location of the disturbance. Figures 6.13 and 6.14 illustrate the stator cur-

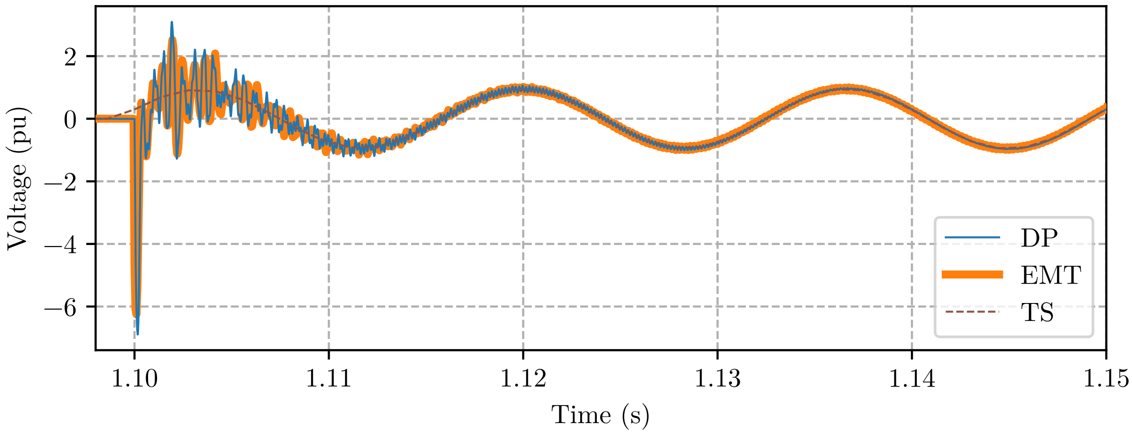

Figure 6.12: Bus 297 voltage response to a fault on bus 297 in the ACTIVSg500 network

rent responses for the first second following the fault. The stator current responses illustrate good agreement between the EMT and DP program. The DP program accurately captures both the amplitude fluctuations and dc offset that inductive currents exhibit following sudden disturbances [1]. The TS results illustrate the effect of the quasistationary assumption where the amplitude changes are accurately modeled but the dc offset is neglected. The dc offset is a consequence of the differential nature of inductors so it is not captured by an algebraic stator model.

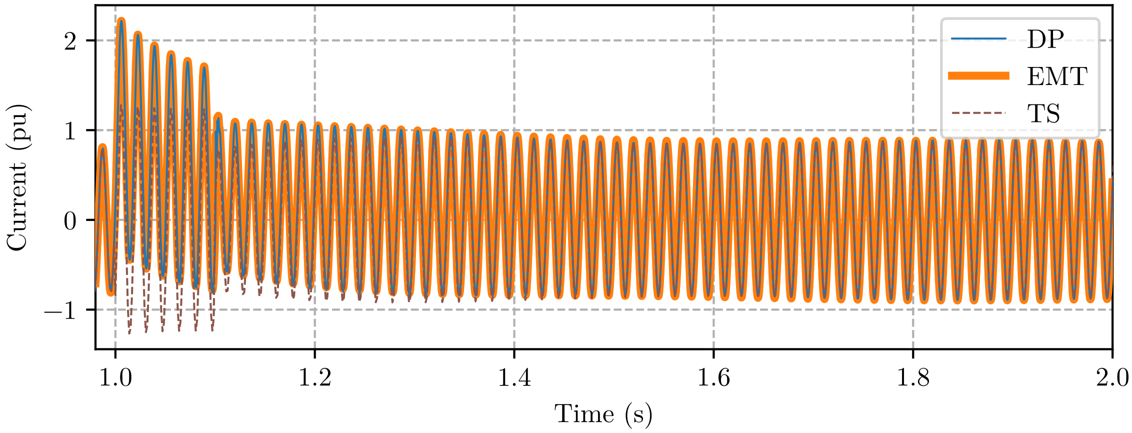

Figure 6.13: Generator 82 stator current response to a fault on bus 297 in the ACTIVSg500 network

Figures 6.15 and 6.16 illustrate the air-gap torque responses of generators for the first 2 s following the fault. These results illustrate good agreement between the three programs for electromechanical oscillations. The DP results also accurately captures the fundamental frequency oscillations that are the result of stator dynamics. These oscillations are neglected by the TS program whose results model the average value of the air-gap torque.

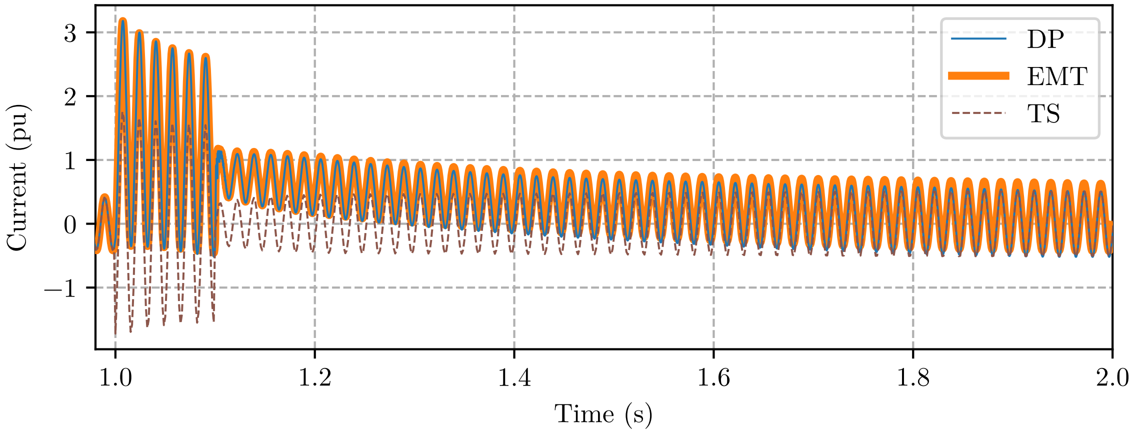

Figure 6.14: Generator 442 stator current response to a fault at bus 297 in the ACTIVSg500 network

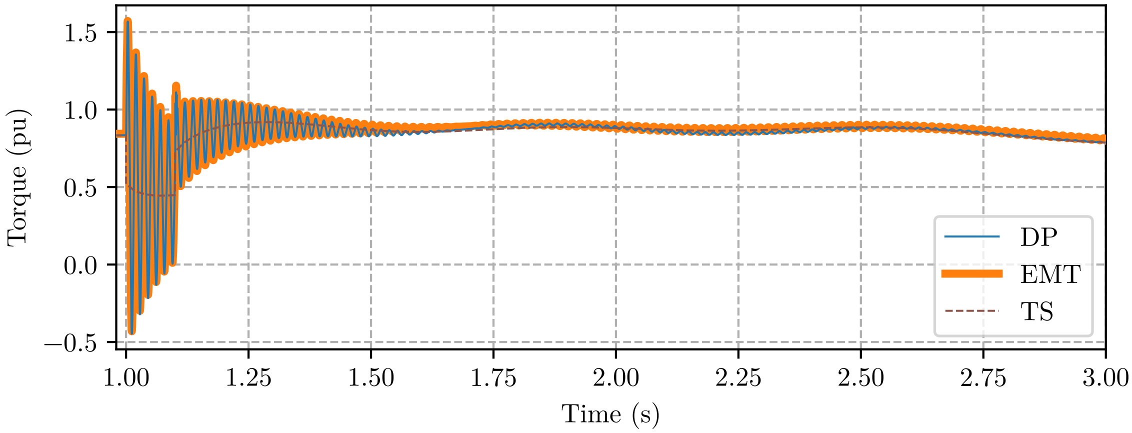

Figure 6.15: Generator 82 air-gap torque response to a fault at bus 297 in the ACTIVSg500 network

Figures 6.17 and 6.18 illustrate the generator speed responses to the fault for the entire simulation. The electromechanical oscillations are dominated by a 0.4 Hertz mode in this case. There are some minor differences in the damping of this mode, which is frequently observed in rotor speed responses. However, the differences are minor for both generators and the rotor speed responses are almost indistinguishable, indicating good agreement between the three programs. Furthermore, the DP and EMT results are also in good agreement for the fundamental frequency oscillations near the fault induced by the stator dynamics, which are neglected by the TS program.

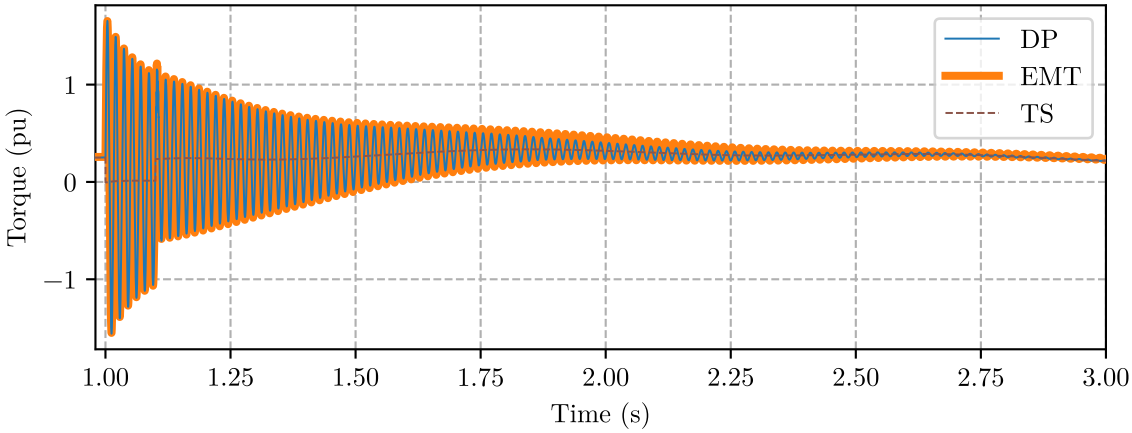

Figure 6.16: Generator 442 air-gap torque response to a fault at bus 297 in the ACTIVSg500 network

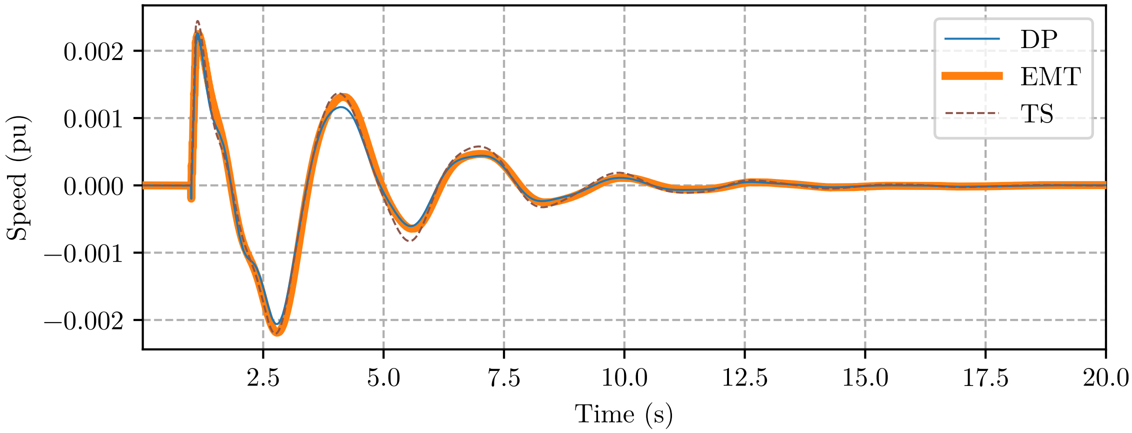

Figure 6.17: Generator 82 rotor speed response to a fault at bus 297 in the ACTIVSg500 network

Figure 6.19 illustrates the elapsed CPU time and step count, both normalized to their final value, for the simulated disturbance. Step refers to either a numerical integration step or a reinitialization step in this context. This plot demonstrates that in this case, there is a sudden almost step-like change following the fault in both the elapsed CPU time and step count. The region near the disturbance is highlighted and additional details are provided, which is the 0.2 s following the fault. The results show that the simulation consumes approximately 40 % of its total CPU time and 25 % of the total steps taken simulating the fault. Another 25 % of CPU time and 40 % of steps is consumed simulating the post-disturbance system. The simulation

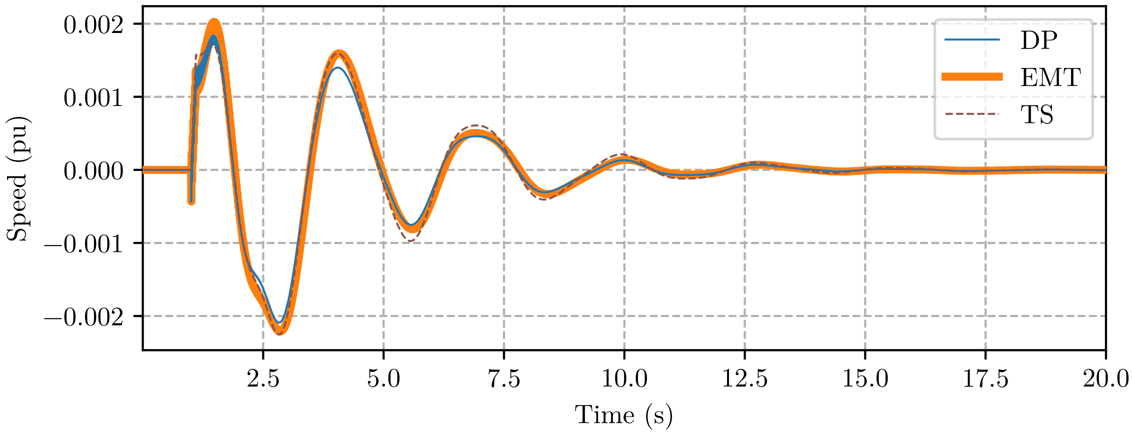

Figure 6.18: Generator 442 rotor speed response to a fault at bus 297 in the ACTIVSg500 network

uses 65 % of both CPU time and steps just to simulate the fault and 0.1 s of the post-fault dynamics. The main plot indicates that approximately 90 % of the CPU time and steps is used in the first 4 s following the fault. Therefore, 15 s or 75 % of this simulation requires only 10 % of the computational resources.

The most expensive steps can also be estimated by the results shown in Figure 6.19. Relatively speaking, expensive steps occur when the elapsed CPU time rises at a greater rate than the step count. Conversely, the least expensive steps occur when the step count rises at a greater rate than the elapsed CPU time. The results in Figure 6.19 indicate that the steps taken during the fault are more expensive than the steps taken immediately following removal of the fault. The period between 1.2 s and 5 s illustrates that that there are some fluctuations in the relative rates of change but for the most part the steps have a fairly consistent cost. The most expensive portion of the simulation is the reinitialization and first step associated with the beginning of the disturbance. The combined cost of these two steps accounts for 1.55 % of the total CPU time over two steps which is only 0.0061 % of the total step count.

Figure 6.20 illustrates the residual and Jacobian evaluation counts, both normalized to their final value, for the simulated disturbance. The normalized step count is also included for reference. The region of time between 1 s and 1.2 s is once again highlighted for additional details regarding progression of these operations through the fault. These results illustrate that approximately 70 % of all residual and Jacobian evaluations are performed within 0.2 s of the fault. The residual line rises faster than the step count line during the fault suggesting that more

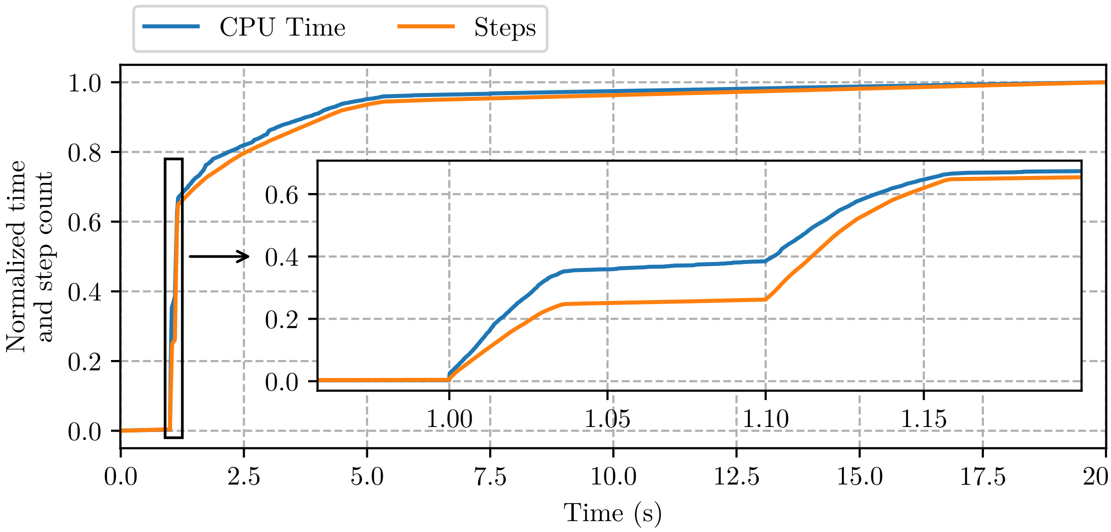

Figure 6.19: Normalized elapsed CPU time and step count for simulation of a fault on bus 297 in the ACTIVSg500 network

residuals per step are required in during this period of time when compared with the rest of the simulation. The two follow roughly the same trajectory for the remainder of the simulation with the step count rising to eventually catch up with the residual line. This behaviour suggests that a consistent number of residual evaluations is required per step after the fault occurs. The Jacobian evaluations shown in Figure 6.20 demonstrate that the Jacobian is frequently reconstructed and refactored during the fault. The results show that the period of time during the fault accounts for 50 % of all Jacobians used in the simulation. The largest increase once again occurs at the fault with 2.6 % and 0.92 % of residual and Jacobian evaluations conducted in 0.0061 % of steps.

Figure 6.21 illustrates the reinitializations over time, normalized to the total number of reinitializations, for the simulated disturbance. Each bar represents 0.5 s of simulation time and the solid line is the cumulative count. The first bar starts at 1 s as events do not occur before the disturbance and all bars are normalized to the height of the first bar. These result show that nearly 60 % of reinitializations occur in the first second following the disturbance and that almost all reinitializations have occurred after 5 s. This behaviour is expected in this case as all reinitializations except for the two associated with applying and removing the fault are a consequence of limiter devices in the generator controls. Controller variables will be the most active during and immediately following the fault, which will trigger limiters for generators near the disturbance.

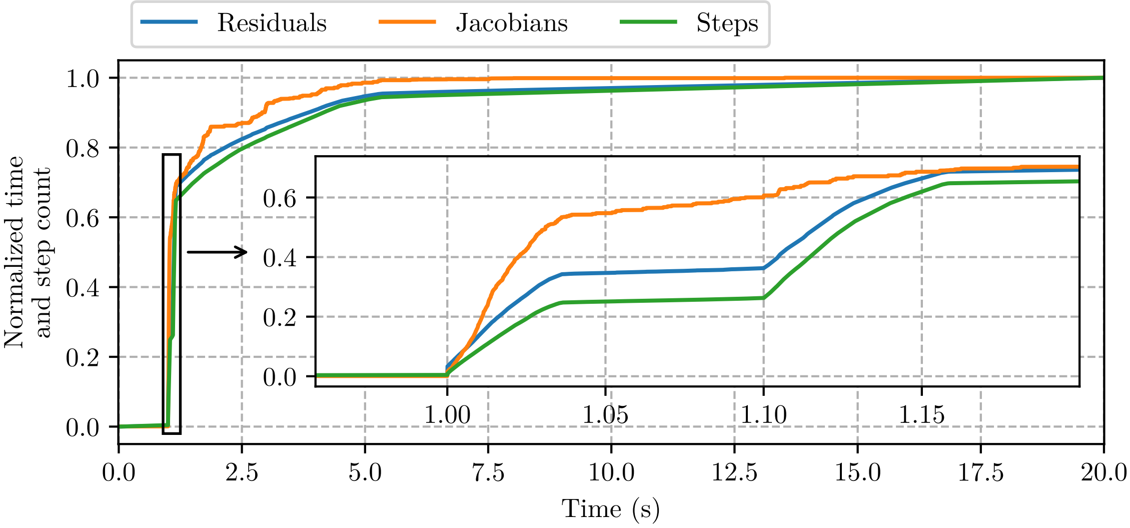

Figure 6.20: Normalized residual and Jacobian evaluations for simulation of a fault on bus 297 in the ACTIVSg500 network

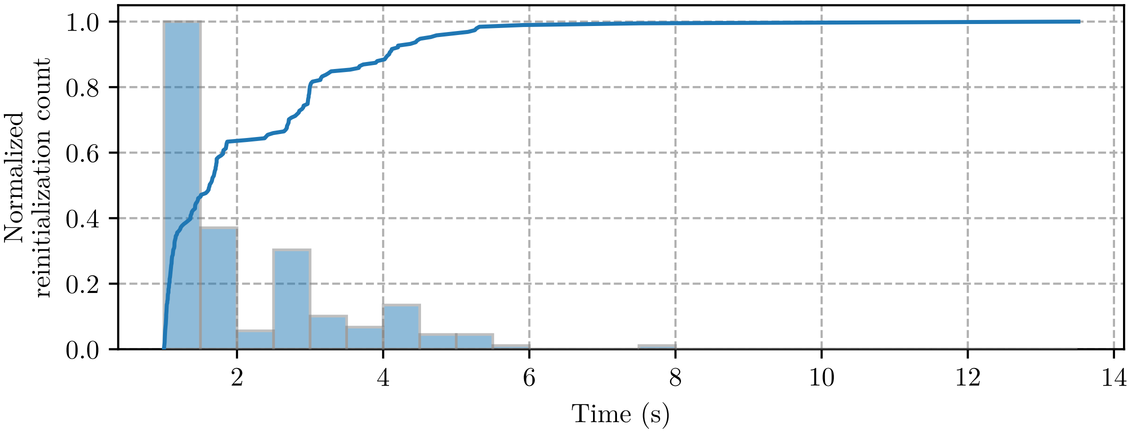

Figure 6.21: Reinitializations for simulation of a fault on bus 297 in the ACTIVSg500 network

Figure 6.22 illustrates the time step over time for the simulated disturbance with the first 0.2 s following the disturbance expanded for additional details. The solver determined for this case that a time step in the range 1 µs to 150 µs is required to accurately pass through network transients. The next 4 s of simulation time are conducted using a time step in the range 0.3 ms to 1 ms, with short drops to handle events requiring reinitialization. The last 15 s of the simulation are carried out using the maximum time step of a half cycle as the only remaining dynamics are electromechanical oscillations. These results illustrate the importance of variable step solvers in dynamic phasor-based simulators. A high degree of accuracy comparable with EMT simulation for network transients can be achieved with microsecond scale time steps near the disturbance. As the dynamics progress towards steady state the time step is automatically increased towards large fractions of a cycle taking advantage of carrier demodulation to complete long duration simulations of large systems.

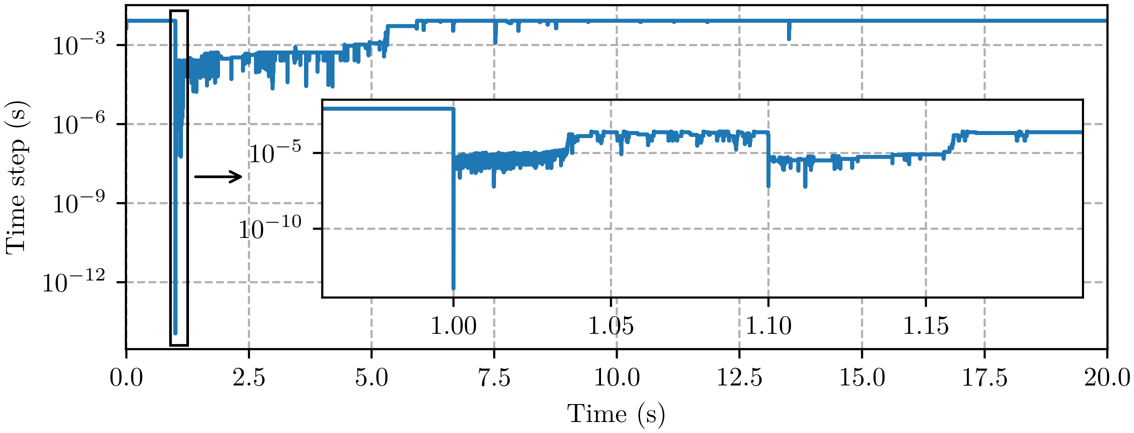

Figure 6.22: Simulation time step for a simulation of a fault on bus 297 in the ACTIVSg500 network

6.3 ACTIVSg500-HVdc

The ACTIVSg500 case used in Section 6.2 was modified to include two HVdc transmission systems. The 345 kV ac transmission line located between buses 386 and 440 was converted into a dc transmission line with the rectifier connected to bus 386. The 345 kV ac transmission line located between buses 220 and 162 was converted into a dc transmission line with the rectifier at bus 220. The placement of the rectifiers was selected based on the direction of active power flow in the original ac transmission lines. The line between buses 386 and 440 is referred to as HVdc line 1 and the line between buses 220 and 162 is referred to as HVdc line 2. This modified case is referred to as the ACTIVSg500-HVdc network.

A 20 s second simulation of the ACTIVSg500-HVdc was conducted in both the DP and EMT programs where a 0.1 s three-phase short circuit fault was applied at 1 s on bus 39. This location was selected as it induces a strong response from both HVdc lines without destabilizing their controllers. The DP program requires 4.6 s of CPU time to simulate this disturbance while the EMT program requires 1422 s of CPU time. Therefore, the DP program is 309 times faster than the EMT program for this disturbance.

The timing results in this case illustrate an advantage of dynamic phasors for modern transient stability simulation of power systems. Switching devices in EMT simulators require interpolation to ensure switching events are accurately located [15]. Interpolation is essential to maintain accuracy involving periodic switching devices but it has an impact on performance. Furthermore, interpolation must be used for an entire simulation so its impact is not localized to disturbances. On the other hand, the dynamic phasor-based HVdc model predominantly impacts only the period of time nearest a disturbance when network variables are actively changing. Its variables return to rest in steady state like the demodulated ac variables. The relative change in CPU time for a 20 second simulation of the ACTIVSg500 network illustrates the impact that HVdc models has on the EMT and DP methods. The required CPU time for the EMT simulation increased by 250 % when the HVdc transmission lines are included in its ACTIVSg500 network model while the required CPU time for the DP program only increased by 31 %.

Figure 6.23 illustrates HVdc line 1's capacitor voltage and Figures 6.24 and 6.25 illustrate HVdc line 1's line currents at the rectifier and inverter, respectively. The inverter quickly enters into commutation failure following the disturbance as demonstrated by the inverter current's sudden rise to nearly 5 kiloamps following the disturbance. The inverter shuts down momentarily following commutation failure and the entire system begins to recover before the fault is removed and continues recovery through the fault's removal. The results for HVdc line 1 illustrate good agreement between the DP and EMT programs through the disturbance. The DP program demonstrates that it is capable of accurately modeling the general trend through the adverse operating conditions illustrated in HVdc line 1's response.

Figure 6.28 illustrates line 2's dc capacitor voltage and Figures 6.26 and 6.27 illustrate line 2's line currents at the rectifier and inverter, respectively. The dc line currents illustrate that both the rectifier and inverter remain in normal operating mode for the duration of the disturbance. Line 2's response to the fault is less severe than line 1 because it does not enter into commutation failure. However, the dc line voltage and currents experience large excursions from their steady state values. The DP and EMT programs maintain excellent agreement

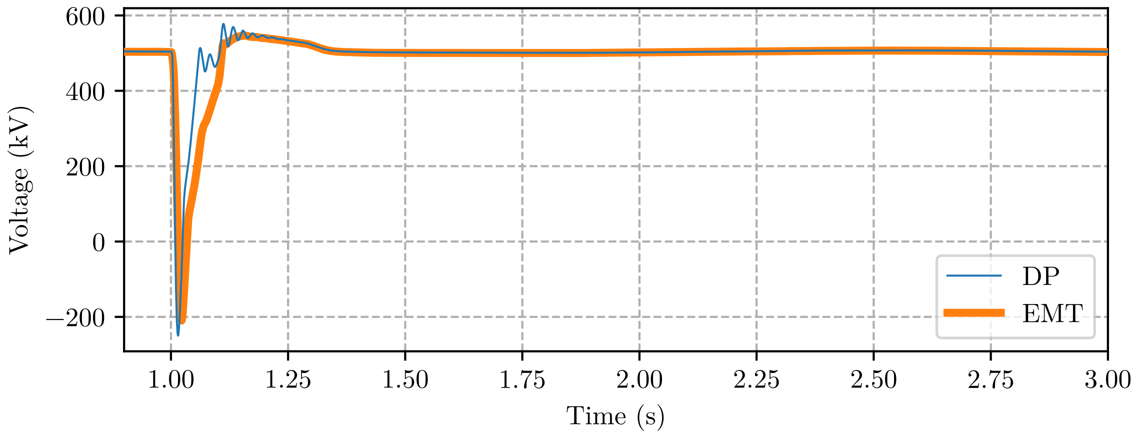

Figure 6.23: HVdc line 1 dc capacitor voltage response to a fault on bus 39 in the ACTIVSg500-HVdc network

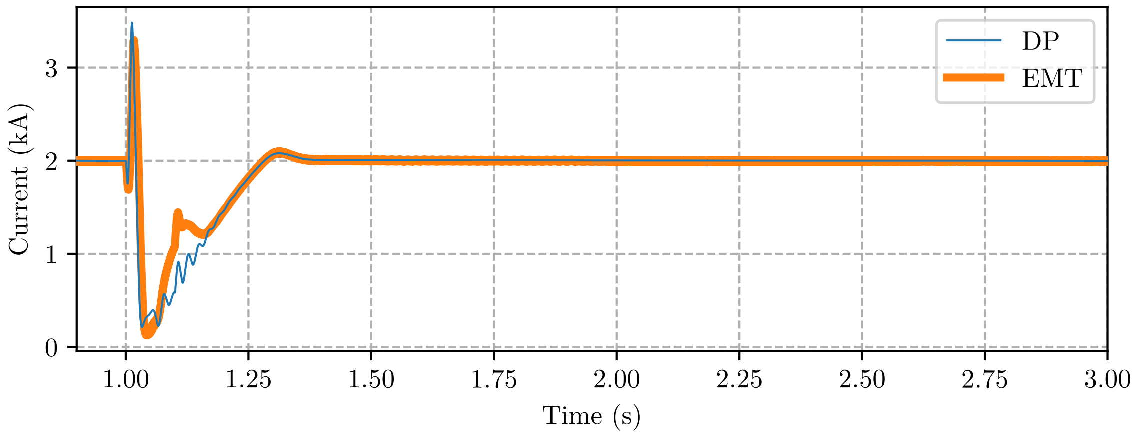

Figure 6.24: HVdc line 1 rectifier dc current response to a fault on bus 39 in the ACTIVSg500-HVdc network

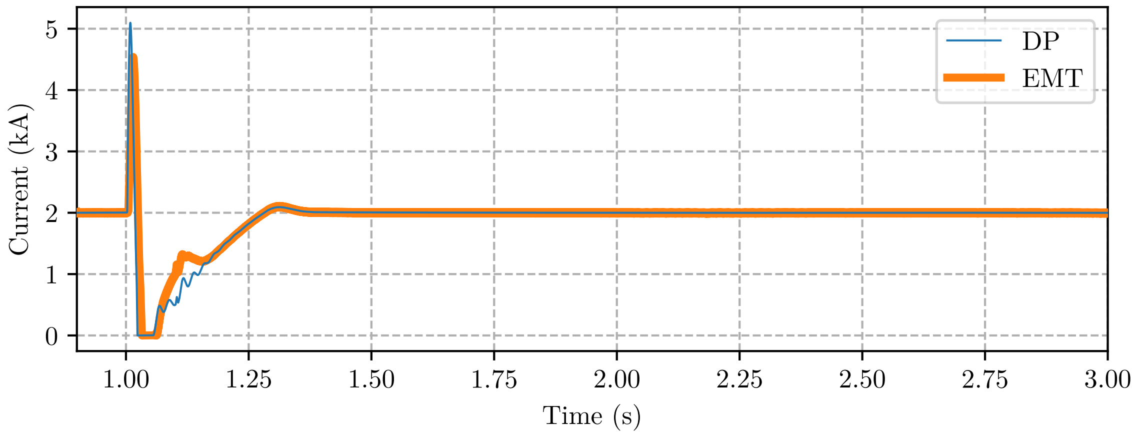

Figure 6.25: HVdc line 1 inverter dc current response to a fault on bus 39 in the ACTIVSg500-HVdc network

through these transients. These results together with the responses of line 1 demonstrate that the overall behaviour of dc side transients are accurately modeled using the DP program even through commutation failure and converter shut down.

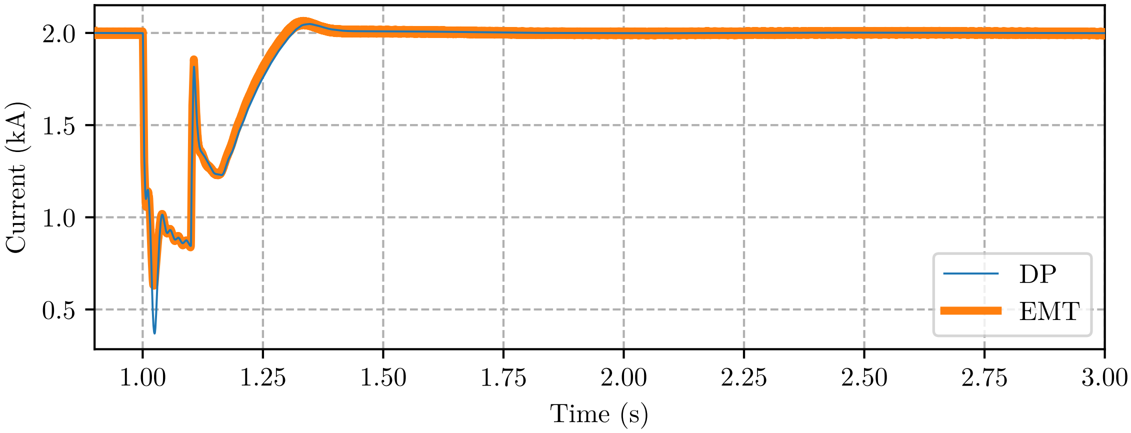

Figure 6.26: HVdc line 2 rectifier current response to a fault on bus 39 in the ACTIVSg500-HVdc network

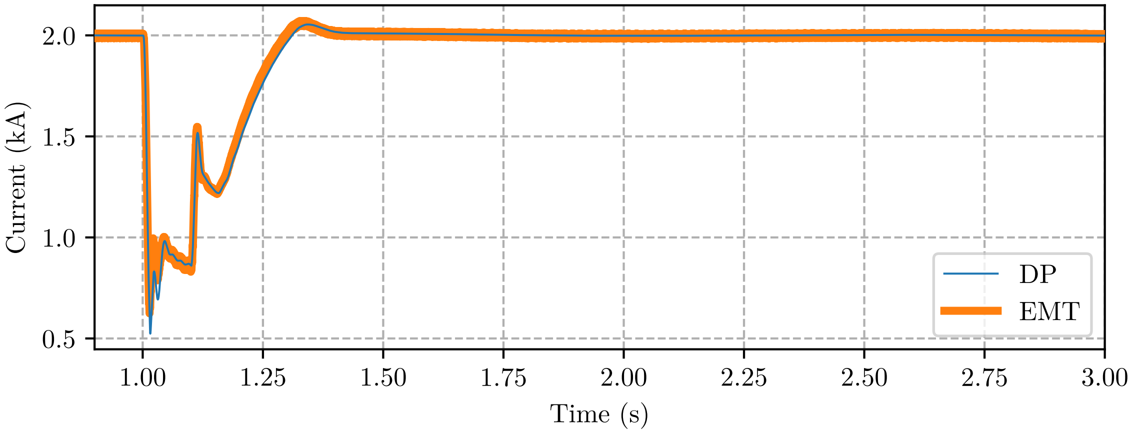

Figure 6.27: HVdc line 2 inverter current response to a fault on bus 39 in the ACTIVSg500-HVdc network

Figures 6.29 and 6.30 illustrate the line 1 rectifier and inverter ac currents, respectively. The rectifier current simulated by the DP program accurately captures the overall trend of the EMT current even though there are some noticeable differences during the disturbance. The inverter current simulated by the DP program deviates significantly during the commutation period, but it accurately models the inverter's recovery period. It is not surprising that the ac currents, particularly on the inverter side where commutation failure occurs, demonstrate

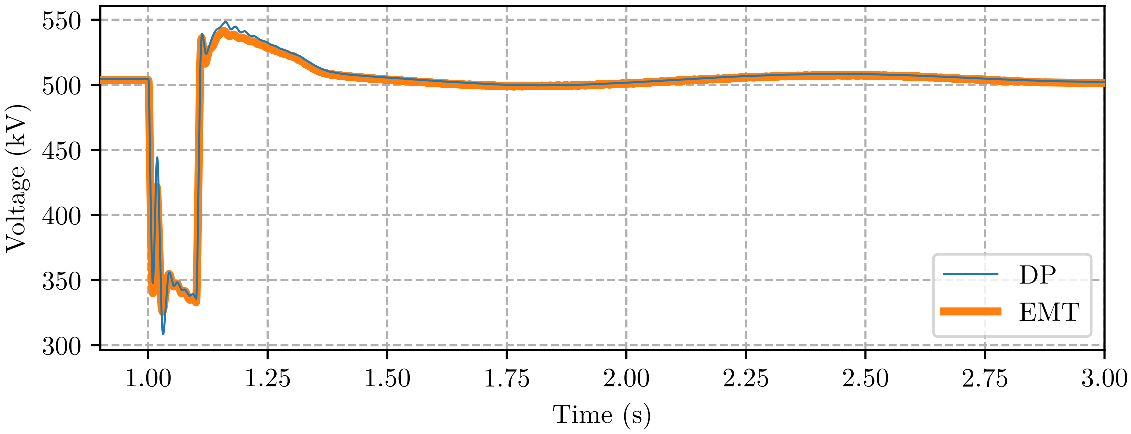

Figure 6.28: HVdc line 2 dc capacitor voltage response to a fault on bus 39 in the ACTIVSg500-HVdc network

some differences during adverse operating conditions. Commutation failure is by its nature a phenomena that is both unbalanced and instantaneous. These characteristics make it difficult to accurately model using dynamic phasors. Enforcing the requirement that the extinction angle remain at zero results in the large non-zero current observed in Figure 6.30 as the two cannot be simultaneously satisfied. There are dynamic phasor model extensions that have been researched to address the issues of unbalanced behaviour [84]. However, these extensions were deemed beyond the scope of this research given the match observed in more critical characteristics such as the dc line currents and voltages.

Figures 6.31 and 6.32 illustrate the line 2 rectifier and inverter ac currents, respectively. The ac currents for line 2 also demonstrate a less severe response when compared to the ac currents of line 1. The EMT and DP programs demonstrate excellent agreement for both the rectifier and inverter ac currents during the disturbance and the system's recovery following removal of the fault. These results along with the ac current responses of line 1 demonstrate that the DP program accurately captures the ac side behaviour of line commutated converters during transient but balanced operating conditions. However, there are some issues during commutation failure that could be improved with model extensions that are capable of predicting unbalanced behaviours.

The ac system response to this disturbance in the ACTIVSg500-HVdc network was observed using two generators. The first is the generator at bus 430, which is close to the faulted bus. The other is the generator at bus 222, which is connected to bus 220 or the rectifier of line 2

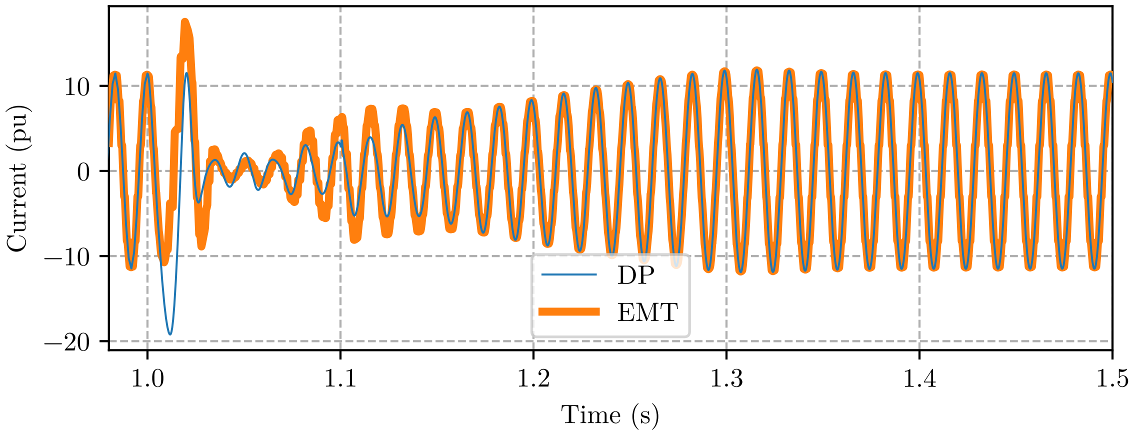

Figure 6.29: HVdc line 1 rectifier ac current response to a fault on bus 39 in the ACTIVSg500-HVdc network

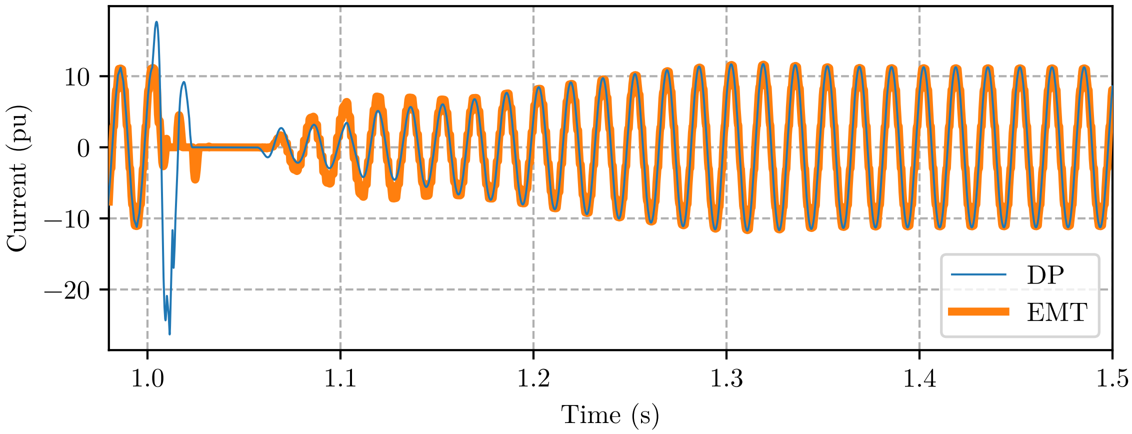

Figure 6.30: HVdc line 1 inverter ac current response to a fault on bus 39 in the ACTIVSg500-HVdc network

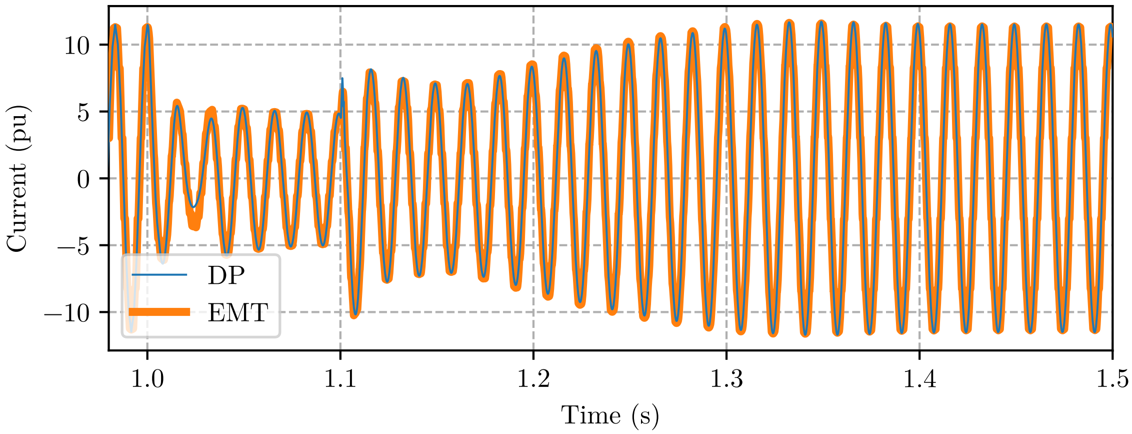

Figure 6.31: HVdc line 2 rectifier ac current response to a fault on bus 39 in the ACTIVSg500-HVdc network

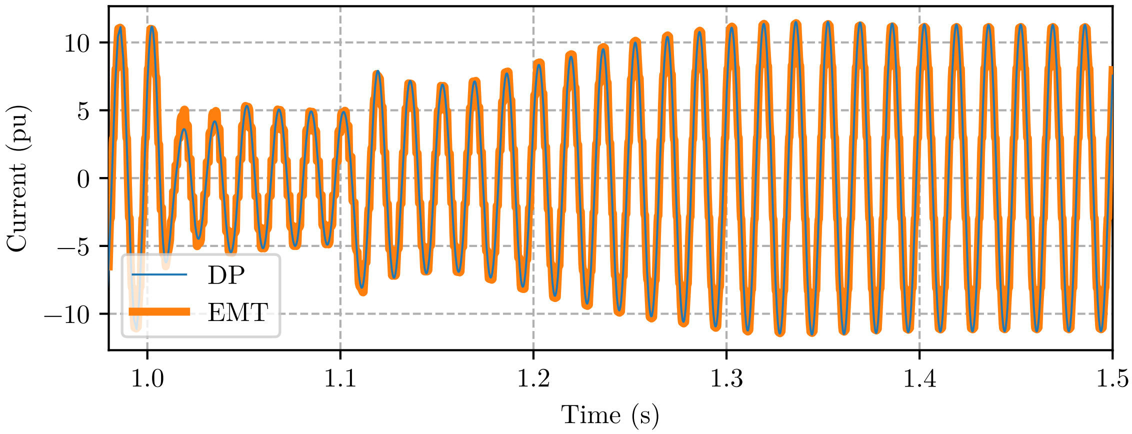

Figure 6.32: HVdc line 2 inverter ac current response to a fault on bus 39 in the ACTIVSg500-HVdc network

through a step-up transformer. Figures 6.33 and 6.34 illustrates the stator current of generators 430 and 222, respectively, for the 0.6 s period following the fault. These results demonstrate good agreement between the two programs for generator stator and network transients. The dc-offset and amplitude fluctuations are well matched in both generators.

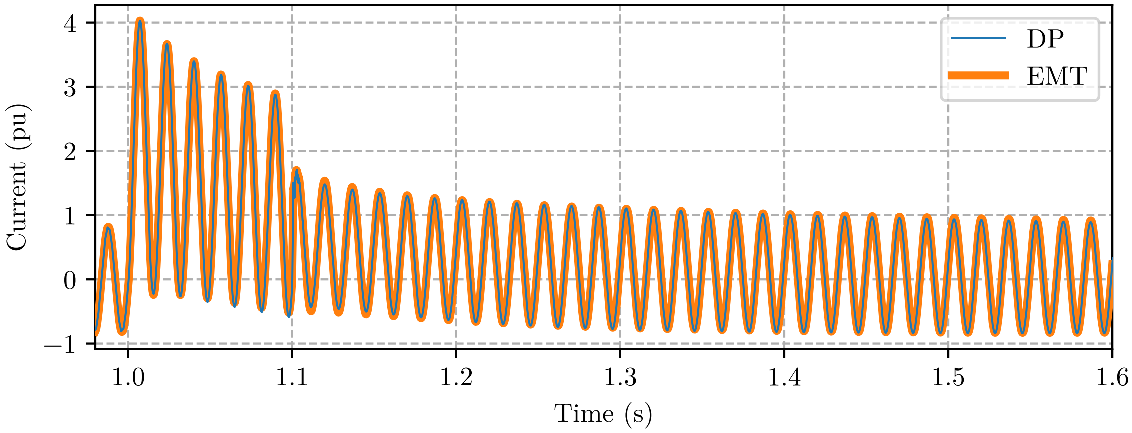

Figure 6.33: Generator 430 stator current response to a fault on bus 39 in the ACTIVSg500-HVdc network

Figures 6.35 and 6.36 illustrate the air-gap torque of generators 430 and 222, respectively, for the first second following the fault. These results also illustrate good agreement between the two programs for generator stator transients. The damping of the fundamental frequency oscillations and the first second of electromechanical oscillations are well matched in both generators.

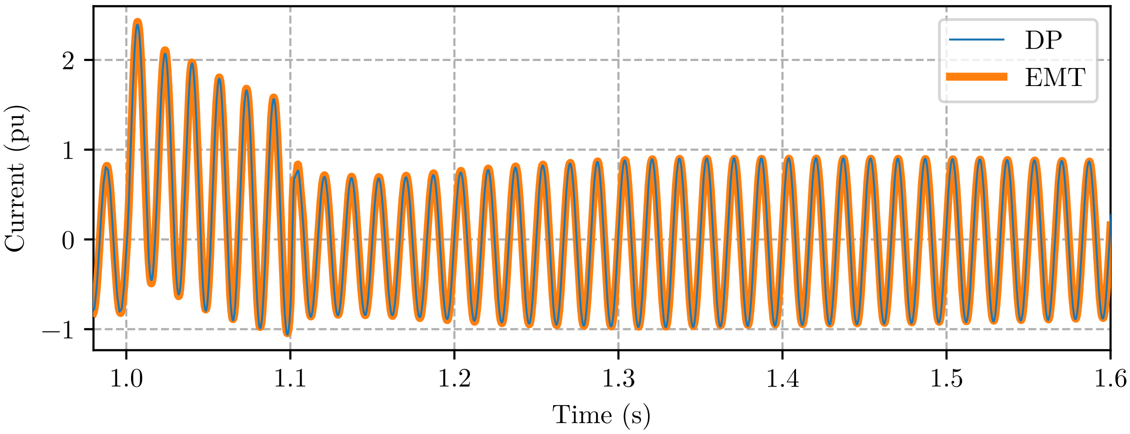

Figure 6.34: Generator 222 stator current response to a fault on bus 39 in the ACTIVSg500-HVdc network

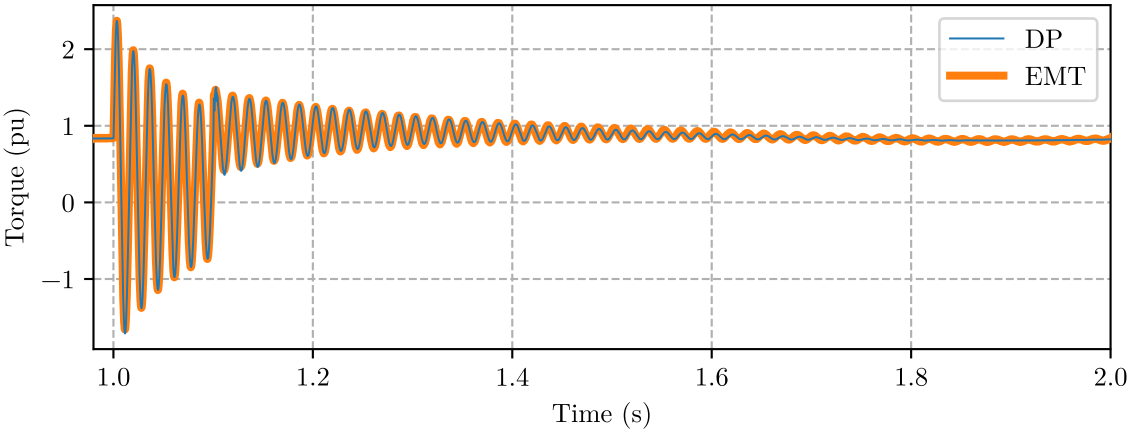

Figure 6.35: Generator 430 air-gap torque response to a fault on bus 39 in the ACTIVSg500-HVdc network

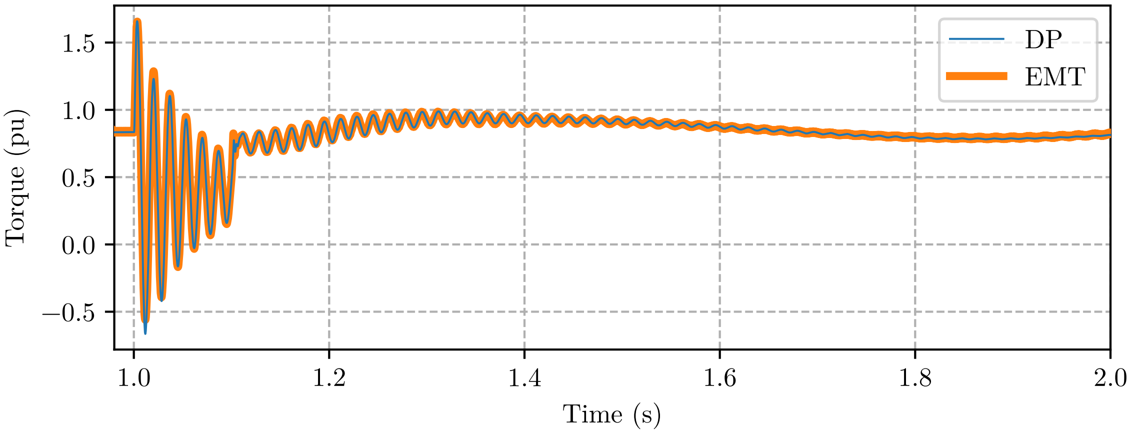

Figure 6.36: Generator 222 air gap torque response to a fault on bus 39 in the ACTIVSg500-HVdc network

Figures 6.37 and 6.38 illustrate the rotor speed deviation of generators 430 and 222, respectively, for the entire simulation. These results demonstrate that the two programs are in excellent agreement for electromechanical transients. The generator results demonstrate that the DP program accurately predicts ac system and generator behaviour in systems involving HVdc transmission. The two programs are well matched for ac network, stator, and electromechanical transients.

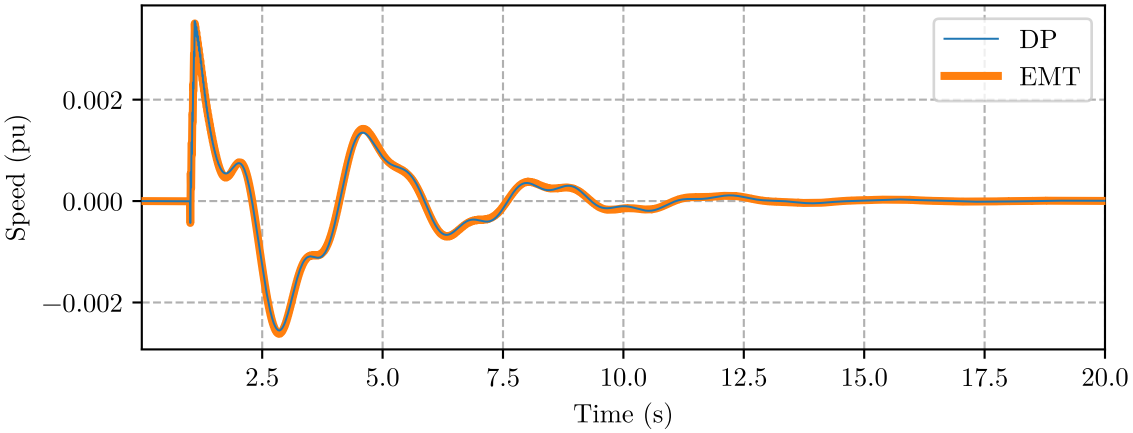

Figure 6.37: Generator 430 rotor speed deviation response to a fault on bus 39 in the ACTIVSg500-HVdc network

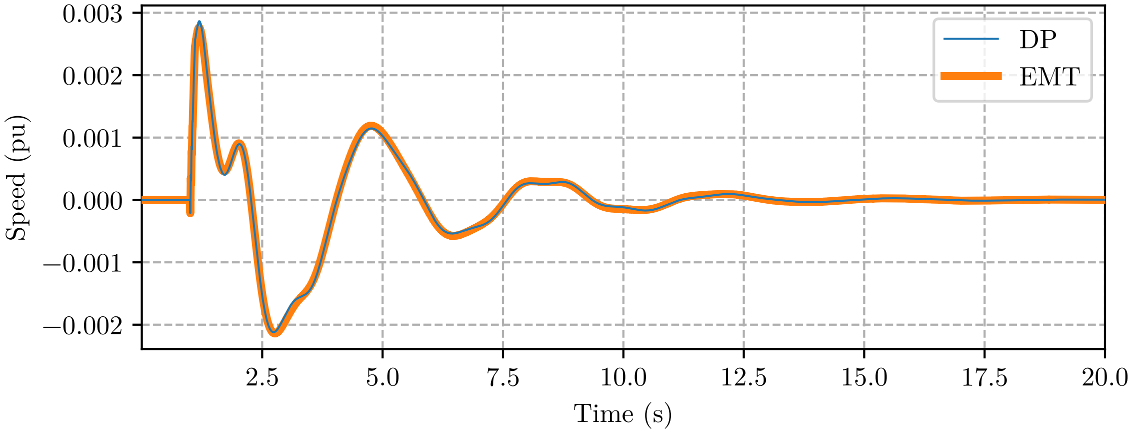

Figure 6.38: Generator 222 rotor speed deviation response to a fault on bus 39 in the ACTIVSg500-HVdc network

6.4 ACTIVSg500-SSTI

The ACTIVSg500 case with embedded HVdc discussed in Section 6.3 was further modified to include a multimass turbine to investigate subsynchronous torsional interactions (SSTI) between HVdc controls and generator turbines. The generator at bus 222 was adjusted to include a multimass model since it is located near the rectifier terminal of HVdc line 2. The generator model was adjusted to use the generator and multimass turbine data presented in the single machine infinite bus multimass case discussed in Section 4.4. The governor located at this generator was also removed from service but its exciter and power system stabilizer were left in service. The rectifier control parameters for HVdc line 2 were adjusted to use a proportional gain, K_{p}, of 0.11 and an integral time constant, T_{i}, of 4.5 ms to introduce unstable subsynchronous oscillations into the network [33].

A 0.1 s three-phase short-circuit fault was applied at bus 39 and the simulation was run for 20 s. Simulations of this case were run with a relative error tolerances 10^{-4}, 10^{-5}, and 10^{-6} to examine the effect of torsional modes on the DP program's performance and accuracy. The DP program requires 11.4 s, 20.9 s, and 38.6 s to complete these simulations with relative error tolerances of 10^{-4}, 10^{-5}, and 10^{-6}, respectively. These results show that the DP program requires between 250 % and 840 % more CPU time when compared to the ACTIVSg500-HVdc case due to the presence of torsional oscillations. The ACTIVSg500-SSTI case was also run on the EMT simulator, which required 2088 s to simulate the disturbance. Therefore, the DP program is between 54 and 183 times faster than the EMT program for this disturbance, which illustrates that the DP program still offers a significant speed benefit when simulating torsional oscillations despite the degraded performance.

Figure 6.39 illustrates a comparison of the rotor speed for generator 222 simulated by the DP program using relative error tolerances 10^{-4}, 10^{-5} and 10^{-6}. The results produced by the DP program with a relative error tolerance of 10^{-4} appear to be better damped when compared to the results produced with an error tolerance of 10^{-5} and 10^{-6}. Figure 6.40 illustrates the rotor speed comparison for the first 3 s following the disturbance. These results demonstrate that early in the simulations there is no difference between the results. Figure 6.41 illustrates the rotor speed comparison 18 s after the disturbance. This comparison shows that the results produced using error tolerances 10^{-5} and 10^{-6} are nearly identical, but the results produced using an error tolerance 10^{-4} are significantly different. However, the difference in error tolerance only affects the damping and the frequency produced by all three results is almost identical.

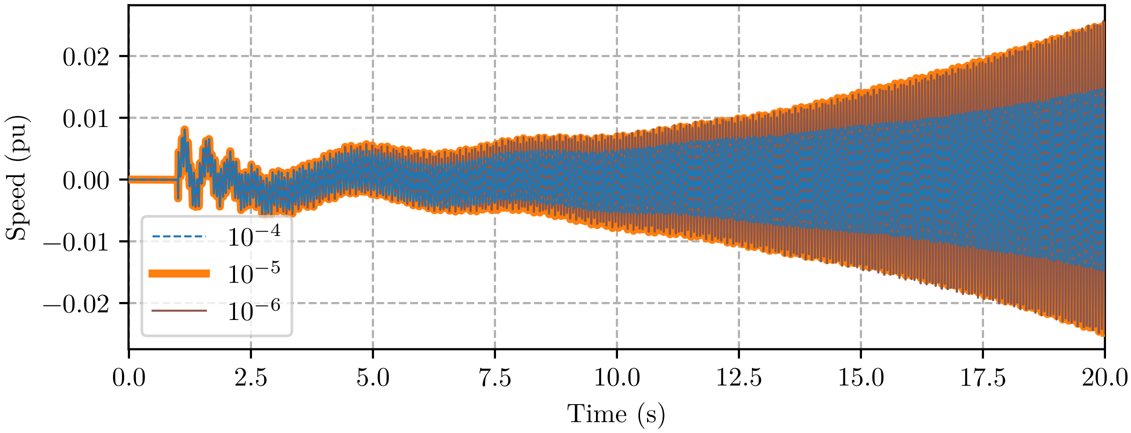

Figure 6.39: Generator 222 rotor speed deviation comparison using different relative error tolerances in the ACTIVSg500-SSTI network – entire simulation

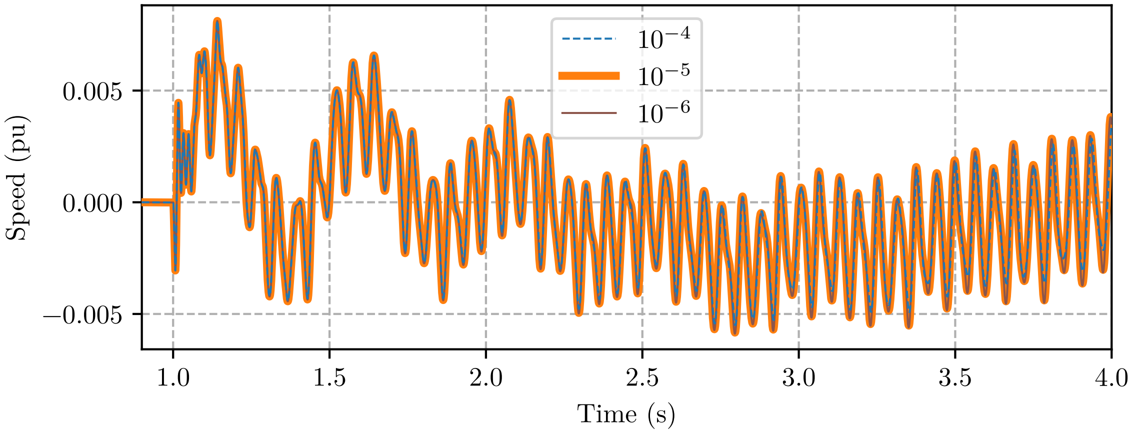

Figure 6.40: Generator 222 rotor speed deviation comparison using different relative error tolerances in the ACTIVSg500-SSTI network – first 3 s following the disturbance

The DP results obtained using an error tolerance of 10^{-5} was selected for comparison with the EMT program based on the results shown in Figure 6.39. Figure 6.42 illustrates the rotor speed response of generator 222 to the disturbance simulated using the DP and EMT programs. The results indicate that the initial transients are well matched but there is a mismatch in the electromechanical damping of the torsional oscillations that becomes apparent around 10 s following the disturbance. Figure 6.43 illustrates the rotor speed response for the first 3 s

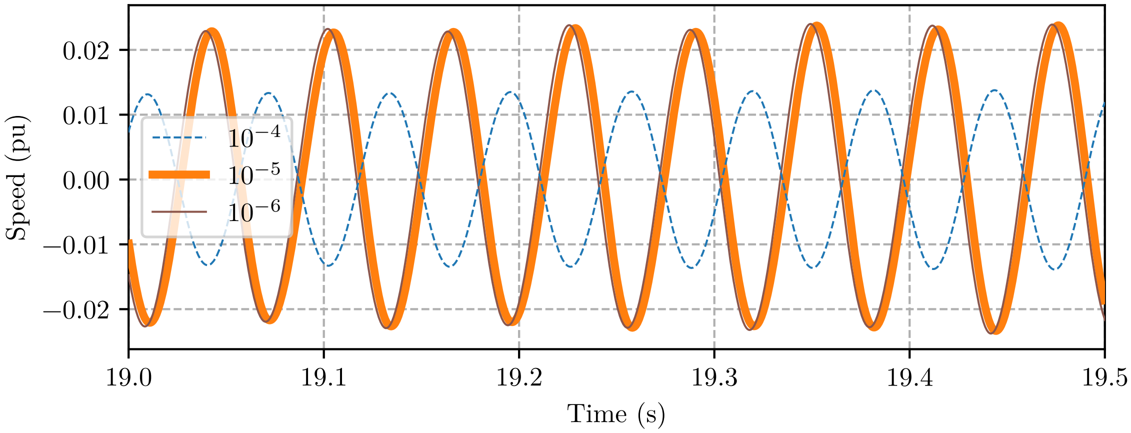

Figure 6.41: Generator 222 rotor speed deviation comparison using different relative error tolerances in the ACTIVSg500-SSTI network – 18 s after the disturbance

following following the disturbance. These results demonstrate that the DP and EMT program are in excellent agreement immediately following the disturbance. Figure 6.44 illustrates the rotor speed response after 18 s have elapsed following the disturbance. These results show that the frequency of the torsional oscillations is well matched between the two programs. The results also demonstrate at this point that the EMT oscillations have slightly less damping than the DP oscillations. However, as shown in the results from previous cases, electromechanical damping is generally difficult to match between different programs as it is sensitive to many simulator specific methods and assumptions. Therefore, the results are considered to be in good agreement.

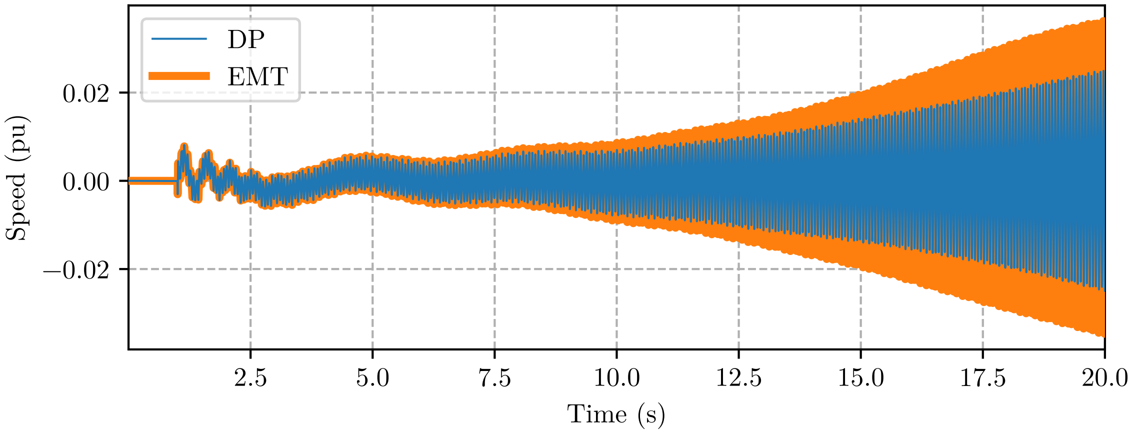

Figure 6.42: Generator 222 rotor speed deviation response to a fault on bus 39 in the ACTIVSg500-SSTI network – entire simulation

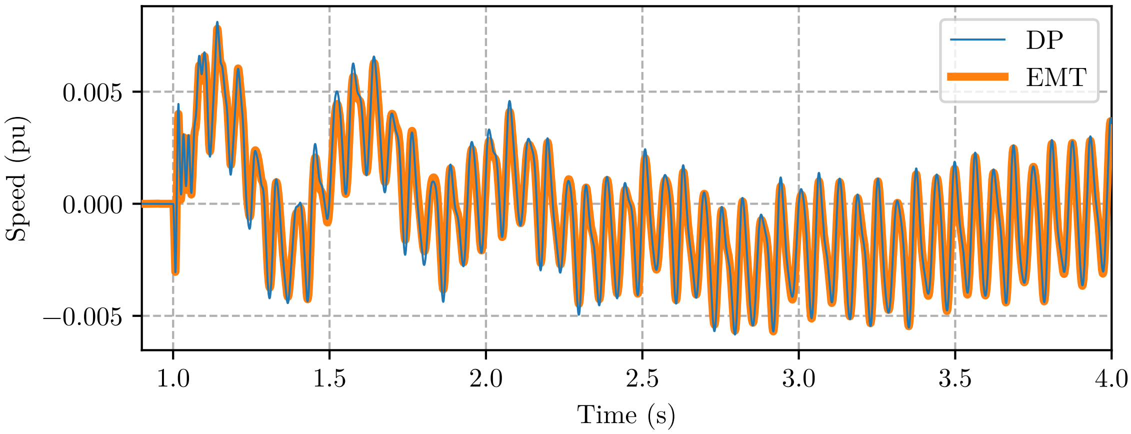

Figure 6.43: Generator 222 rotor speed deviation response to a fault on bus 39 in the ACTIVSg500-SSTI network – first 3 s following the disturbance

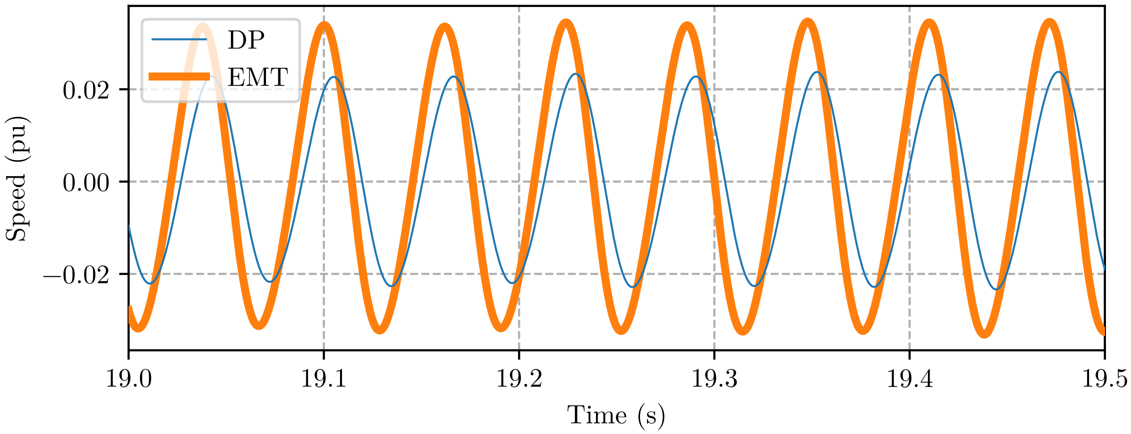

Figure 6.44: Generator 222 rotor speed deviation response to a fault on bus 39 in the ACTIVSg500-SSTI network – 18 s after the disturbance

Figure 6.45 illustrates the HVdc line 2 rectifier current response to the disturbance. These results also demonstrate that the initial transients are well matched but there is a mismatch in the damping of oscillations produced by the generator's turbine. Figure 6.46 illustrates the HVdc line 2 rectifier current for the first 3 s following the disturbance. The results illustrate excellent agreement for the disturbance and the dc system's recovery as well as the torsional oscillations that are induced on the dc current. Figure 6.47 illustrates the rectifier current after 18 s of simulation time has elapsed. These results also illustrate good agreement in the frequency of the torsional oscillations but a similar mismatch in their damping as the rotor speed devation. However, these results together with the rotor speed response of generator 222 demonstrate that the DP program can accurately capture unstable subsynchronous oscillations that are induced by the interaction of HVdc systems and multimass rotor models.

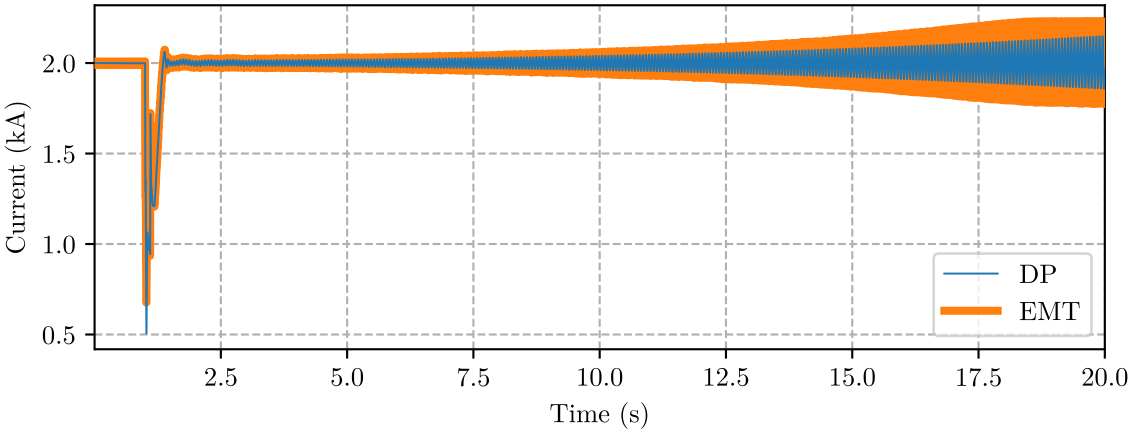

Figure 6.45: HVdc line 2 rectifier current response to a fault on bus 39 in the ACTIVSg500-SSTI network – entire simulation

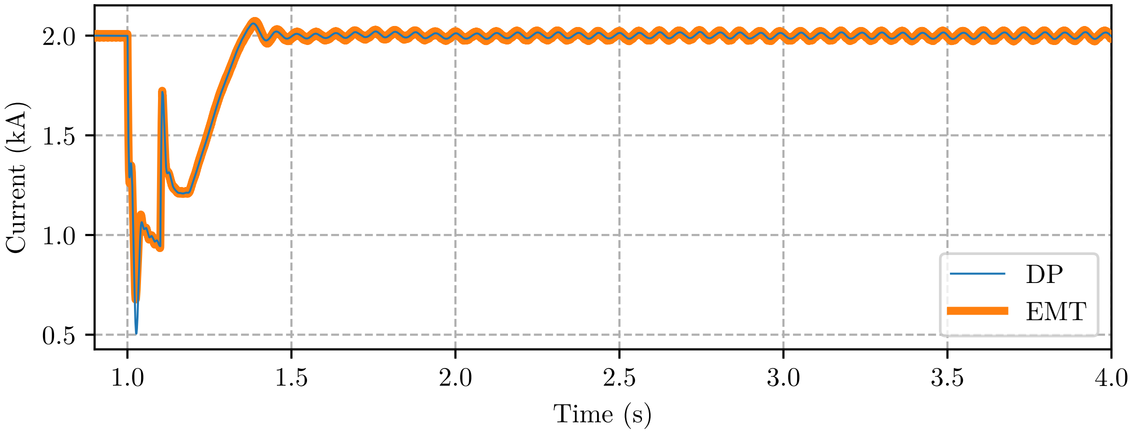

Figure 6.46: HVdc line 2 rectifier current response to a fault on bus 39 in the ACTIVSg500-SSTI network – first 3 s following the disturbance

Figure 6.48 illustrates the elapsed CPU time and step count, normalized to their final values, for the ACTIVSg500-SSTI simulation. These results illustrate that simulation of torsional dynamics has a significant effect on the DP program's CPU time and step count. The first 0.2 sfollowing the fault requires only 25 % of CPU time and steps while the period of time from 5 s to 20 s requires over 50 %. Furthermore, the step count and CPU time are almost linear in this time period. The additional CPU time and linear behaviour are consequences of simulating sustained 16 Hz oscillations associated with the multimass model.

Figure 6.49 illustrates the simulation time step used by the DP program for relative error

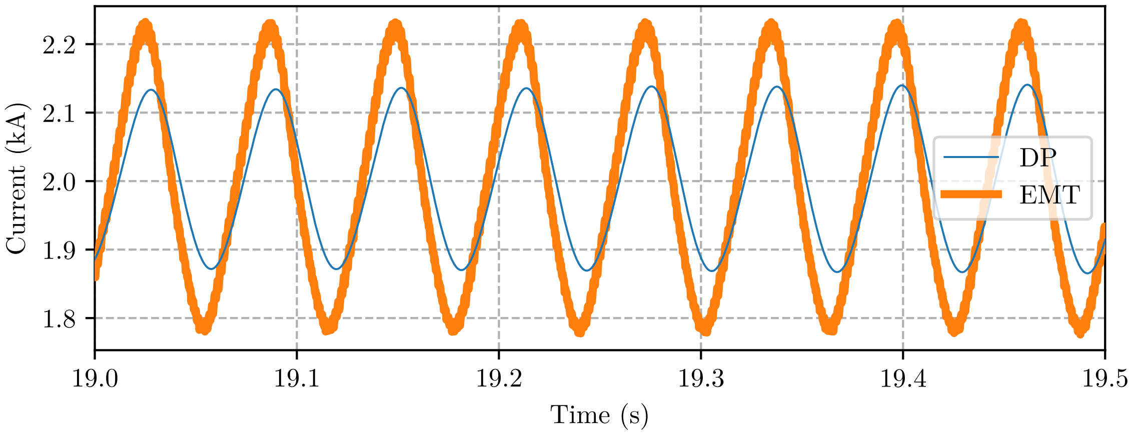

Figure 6.47: HVdc line 2 rectifier current response to a fault on bus 39 in the ACTIVSg500-SSTI network – 18 s after the disturbance

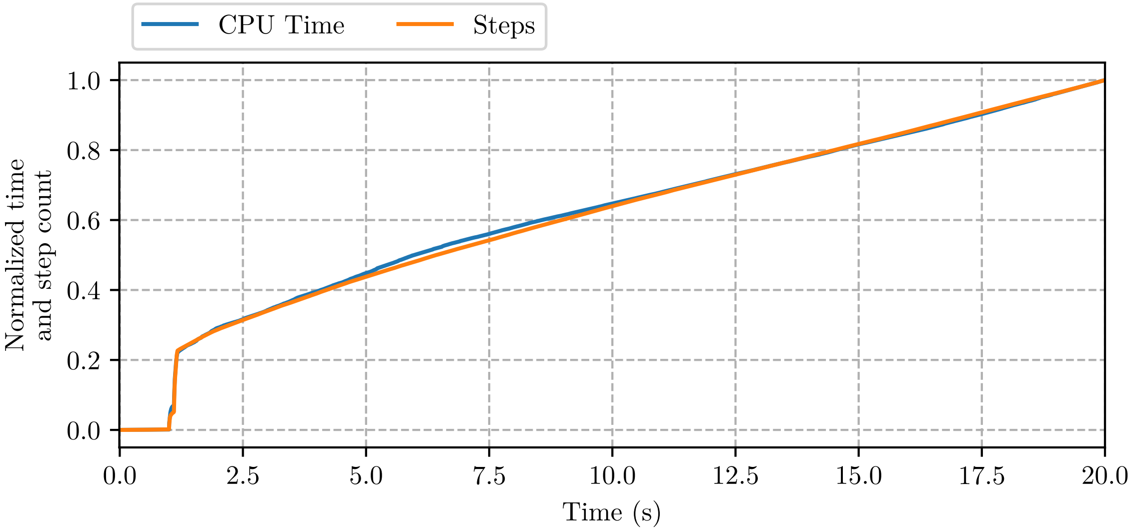

Figure 6.48: Normalized elapsed CPU time and step count for simulation of a fault on bus 39 in the ACTIVSg500-SSTI network

tolerances 10^{-4}, 10^{-5}, and 10^{-6}. The average time step in the torsional portion of the simulation for the three relative error tolerances is 575 µs, 218 µs, and 96 µs, respectively. These values are significantly smaller than the steady state time step used in cases without torsional dynamics, which is a half cycle or 8.33 ms. This result illustrates that a much smaller time step is required for the duration of simulations involving torsional oscillations. However, the ideal time step for capturing the 16 Hz oscillations lies somewhere in the range 218 µs to 575 µs, which is greater than the 50 µs fixed step used by the EMT simulator.

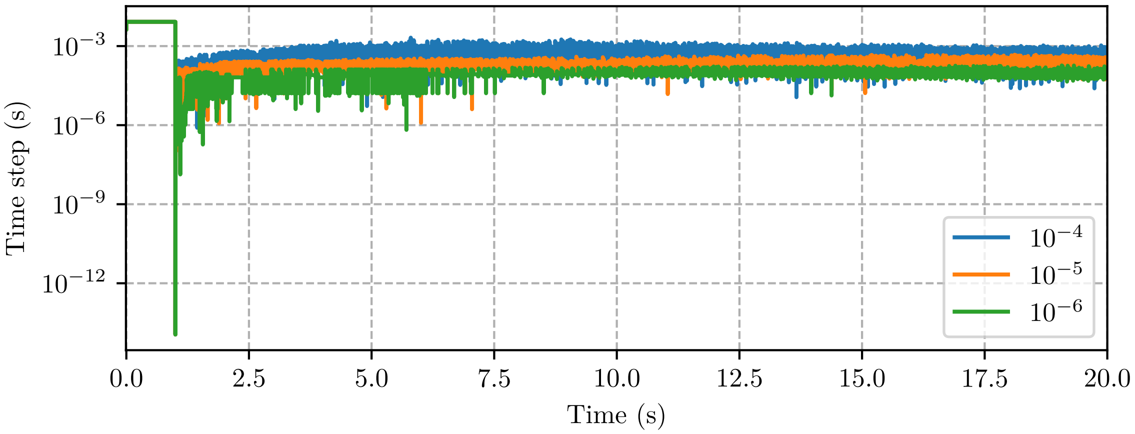

Figure 6.49: Simulation time step for simulation of a fault on bus 39 in the ACTIVSg500-SSTI network

6.5 ACTIVSg2000

The final system used to validate the DP method is a two thousand bus system known as the ACTIVSg2000 network [85]. This network is also a synthetic power system model that approximates the electric grid in Texas and is freely available for research [88]. Generators in this case were aggregated and assigned dynamic models in the same manner as the ACTIVSg500 network. There are 1703 buses and 191 generators in this model after modification. The DP model of this program has 15 373 variables of which 97.9 % are differential. The Jacobian matrix associated with the model has 64 978 nonzeros, which is only 0.028 % of the total elements.

As in the ACTIVsg500 case, a 20 s simulation was run involving a 0.1 s second three-phase short circuit fault applied at 1 s second to each bus. Figure 6.50 illustrates the distribution of CPU time required to conduct the bus fault simulations. The required CPU time for simulations in this case vary over a wide of values, with the fastest case requiring 6.4 s of CPU time and the slowest requiring 117.2 s. However, the distribution shows that the CPU times are well clustered on the lower side of this range with the average simulation requiring 23.6 s of CPU time. Half of the simulations in this case require less than 20 s of CPU time and 88 % require less than 40 s.

A three-phase short circuit fault was simulated on bus 5384 in the ACTIVSg2000 network for comparison with the TS program. This disturbance was not compared with the EMT simulation program due to the size of the system and constraints of the available resources.

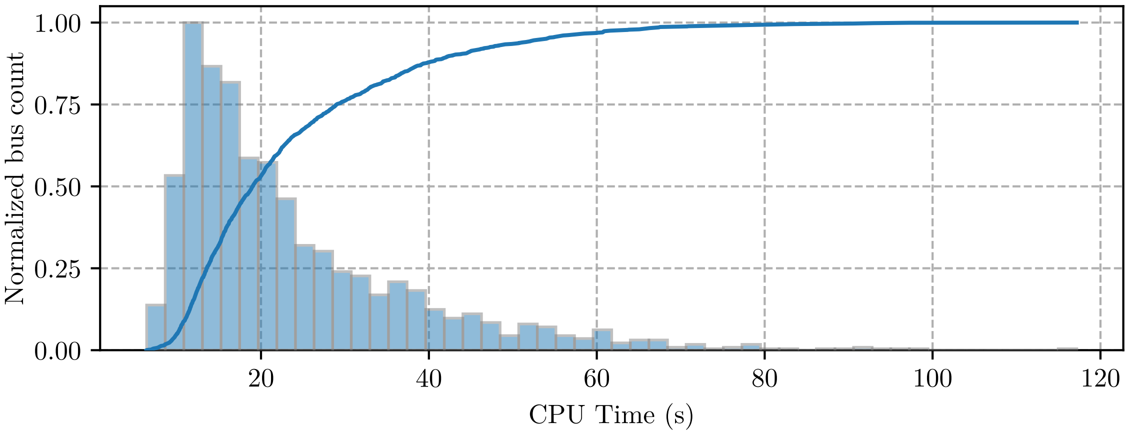

Figure 6.50: Distribution of CPU time required for bus faults in the ACTIVSg2000 network

The DP program requires 67.4 s of CPU time and 11419 steps to complete this simulation, which makes this bus one of the most expensive simulations in this model. Three generators near the location of the disturbance were chosen for comparison, which are located at buses 5324, 5382, and 5035.

Figures 6.51 to 6.53 illustrate a comparison between the DP and TS programs for the stator currents of generators 5324, 5382, and 5035, respectively. Phasor magnitude is compared in this case since EMT simulations are not included in these results. The results indicate good agreement for all three generators during the fault and for the slower electromechanical oscillations. As expected, the TS simulation results do not include the oscillations at the fundamental frequency during the disturbance as those are related to the stator's transients. However, the results show that the average value of the DP results and the TS results are in good agreement when stator oscillations are present. There is some mismatch between the DP and TS simulation results immediately following removal of the fault for generators 5382 and 5035. The stator current simulated by the TS program immediately drops to near its post-fault value while the DP stator current decreases slowly. This mismatch is a consequence of the algebraic stator model used in conventional transient stability simulation.

Figures 6.54 to 6.56 illustrate the air-gap torque comparisons for the three generators. As in the previous cases, the DP simulation results include the fundamental frequency oscillations that are neglected through the quasistationary assumption in TS simulation. However, the TS results accurately capture the average value of the DP results near the disturbance, and the two are well matched during the electromechanical oscillations. This indicates a good agreement between the two programs for dynamics associated with generator power.

Figures 6.57 to 6.59 illustrate comparisons of the rotor speed deviations responses of the three generators. As with the previous sets of simulation results, the rotor speeds in the TS program do not capture the initial drop in rotor speed that is associated with the inductive stator model included in the DP program. Furthermore, there are some slight differences in the electromechanical damping where the DP program appears to simulate the generators with slightly better damping when compared to the TS program. However, this is a common occurance since electromechanical damping between different programs is sensitive to minor differences in modeling. However, the differences in damping are minor and the frequency of the electromechanical oscillations is well matched, so the two programs are in good agreement.

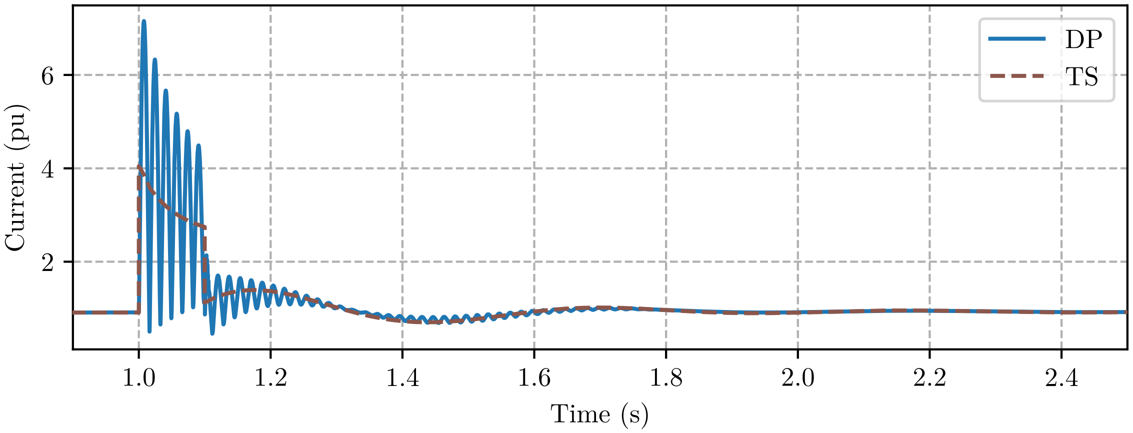

Figure 6.51: Generator 5324 stator current response to a fault at bus 5384 in the ACTIVSg2000 network

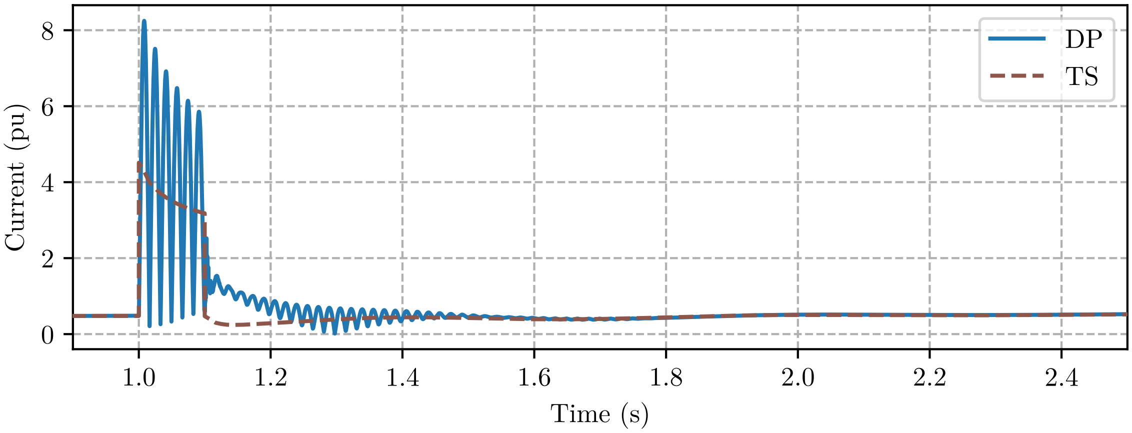

Figure 6.52: Generator 5382 stator current response to a fault at bus 5384 in the ACTIVSg2000 network

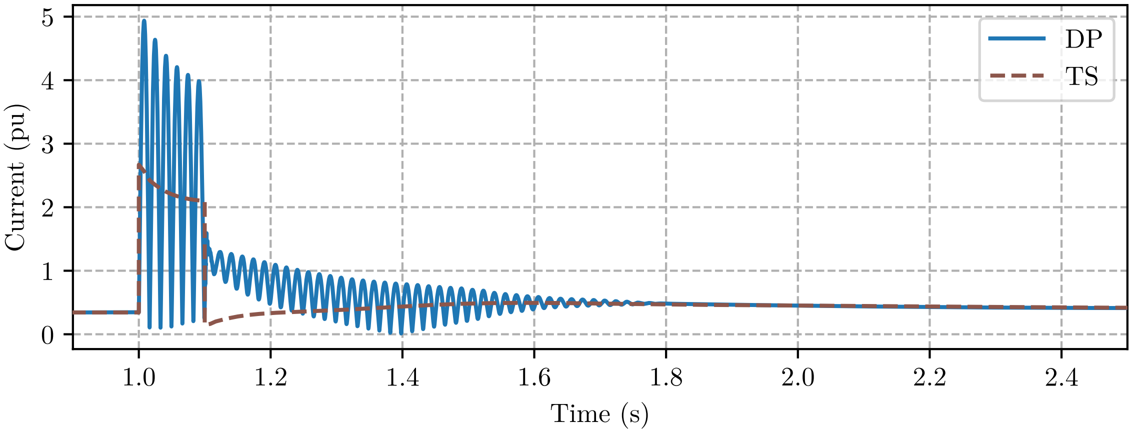

Figure 6.53: Generator 5035 stator current response to a fault at bus 5384 in the ACTIVSg2000 network

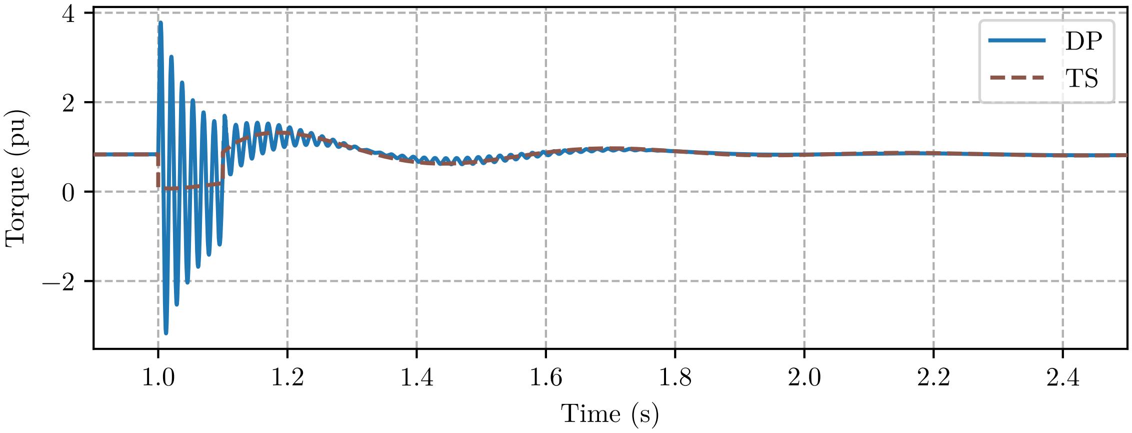

Figure 6.54: Generator 5324 air-gap torque response to a fault at bus 5384 in the ACTIVSg2000 network

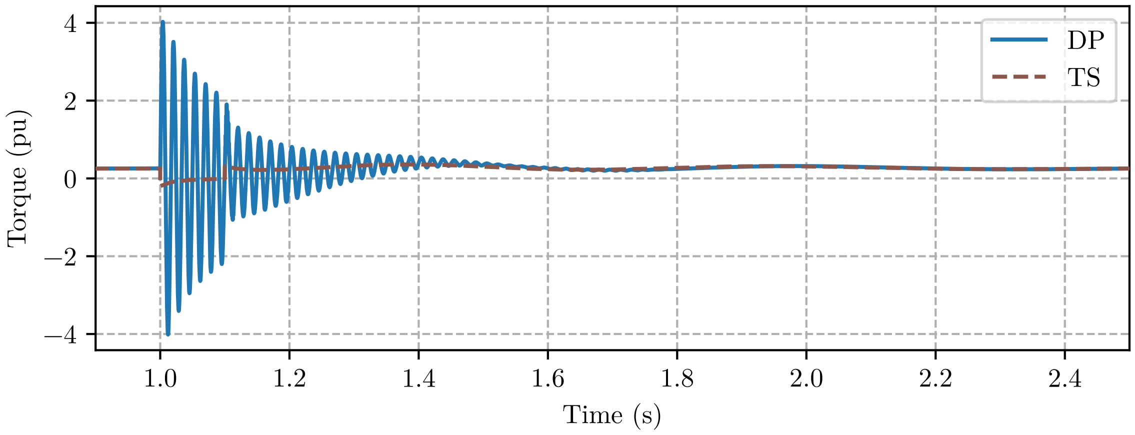

Figure 6.55: Generator 5382 air-gap torque response to a fault at bus 5384 in the ACTIVSg2000 network

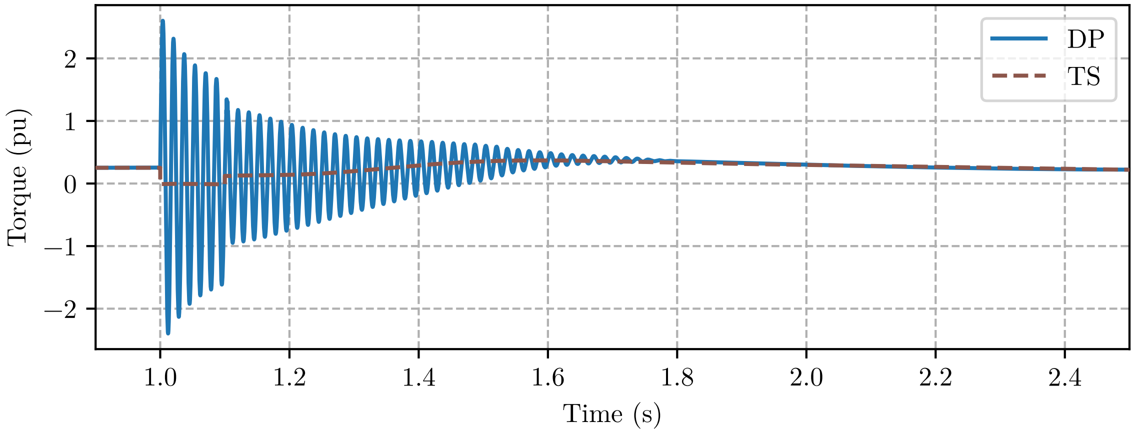

Figure 6.56: Generator 5035 air-gap torque response to a fault at bus 5384 in the ACTIVSg2000 network

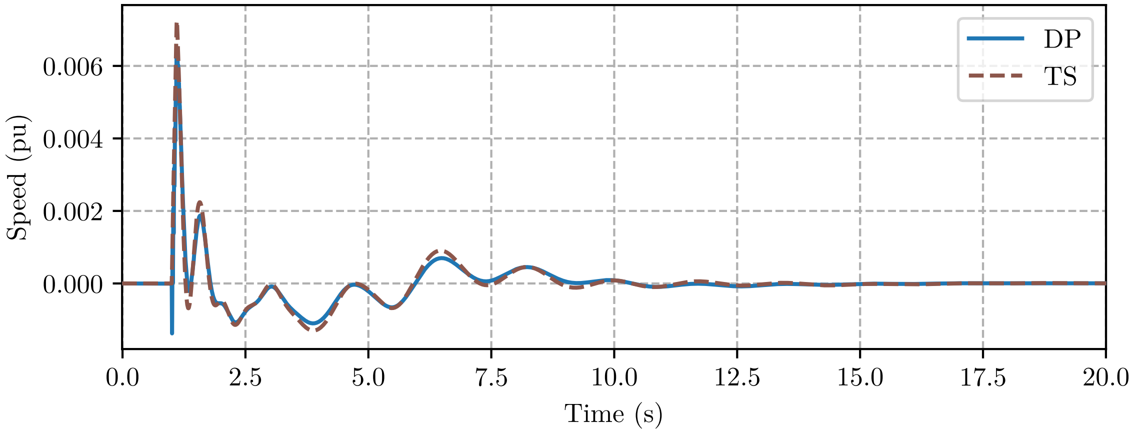

Figure 6.57: Generator 5324 rotor speed deviation response to a fault at bus 5384 in the ACTIVSg2000 network

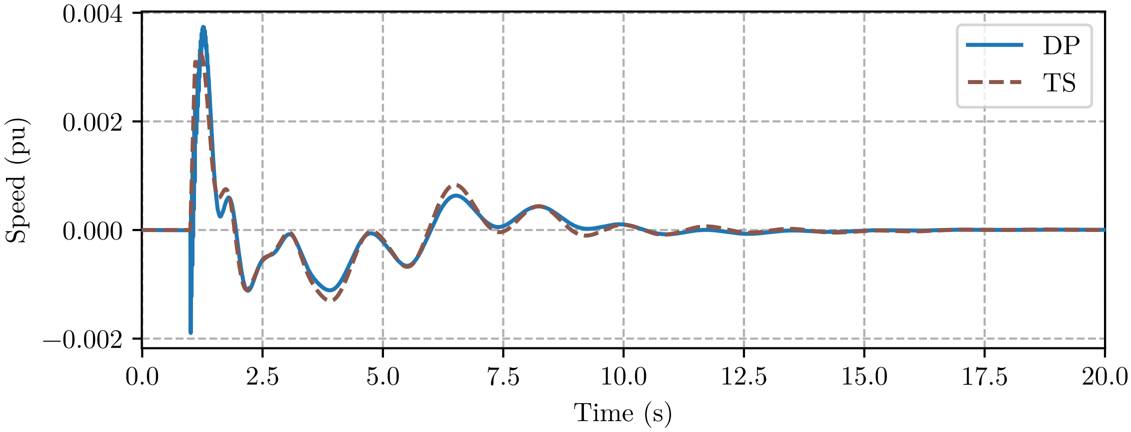

Figure 6.58: Generator 5382 rotor speed deviation response to a fault at bus 5384 in the ACTIVSg2000 network

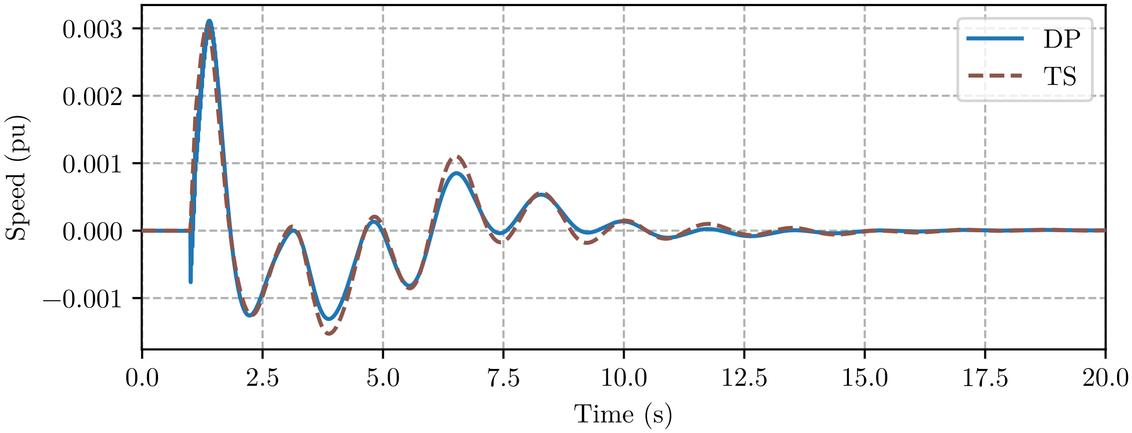

Figure 6.59: Generator 5035 rotor speed deviation response to a fault at bus 5384 in the ACTIVSg2000 network

Chapter 7

Conclusions

The method developed in this work represents an important contribution to the field of transient stability simulation. The primary outcome of this research is that it demonstrated that the frequency bandwidth of transient stability simulations is improved through dynamic phasors and that those simulations are both fast and accurate. Results were obtained with this method for realistic power system models containing HVdc systems and multimass turbines. Comparisons were provided with a commercial EMT simulator, which is considered a benchmark in power system simulation. The dynamic phasor-based simulation results were visually coherent with the EMT simulation results. Furthermore, the dynamic phasor simulations were performed with CPU times that are between thirty and three hundred times faster than EMT simulations. The following sections will examine the conclusions and contributions of this work in Section 7.1 and detail its limitations as well as possible future work in Section 7.2.

7.1 Conclusions and Contributions

The main contribution of this research work is a flexible and robust approach for conducting transient stability simulations of dynamic phasor-based power system models. The approach is based on MNA, which balances flexibility and simplicity with equation conditioning in power system models. It is flexible because new models for the method developed in this work can be defined using any number of variables and equations. The only requirement of these models is that they define an expression for any currents associated with connections to network buses. Furthermore, the MNA approach is generally well-suited to power system models and produces DAEs with a low degree of redundancy. The general structure of power systems is inductive currents connected between capacitive voltages. Therefore, the majority of variables in a network's DAEs are differential and would appear in a state space model as well, which is significantly more complicated to construct.

This work identified several numerical issues related to the solution of dynamic phasor-based DAEs. The DAEs developed by the MNA approach are solved using the BDF family of methods. The general purpose library `IDA` was selected to numerically integrate the DAEs, which uses a variable step and variable order algorithm. Several issues were identified during the course of this research even though the equations developed by MNA are generally well-conditioned and the solver's components are well-designed. The default error estimator in `IDA` assumes that each variable is independent of the other variables in the system. This assumption does not hold when complex variables are split into two real variables. The error in either the real or the imaginary component must take into account the value of its partner in this case. A simple method for correcting this assumption was introduced in this work, which computes the error in both components relative to the magnitude of their combined complex value. Furthermore, variables must be identified as either differential or algebraic for error control and initialization purposes. The membership of auxiliary variables is determined by their associated constraint equation. However, the membership of a network node voltage depends on the components connected to that node. A method to estimate node voltage memberships was proposed in this work that uses the differential portion of the system's Jacobian.

The concept of floating subnetworks was identified in this work. Floating subnetworks are a generalization of network configurations that appear frequently in power system models. For example, synchronous machines are often connected to power systems through a step up transformer. The terminal bus in this case represents a floating subnetwork consisting of a single node. It was shown that floating subnetworks result in numerical deficiencies in a system's DAEs. These deficiencies lead to poorly conditioned Jacobians when the simulation time step is small, which often result in numerical failures. A simple approach was used to solve this issue, which involves applying a small numerical bias to the diagonal of a system's differential coefficient matrix, $\mathbf{T}$.

A new synchronous machine stator model that is compatible with the proposed method was developed in this work. This model is based on a direct reference frame transformation of the stator's differential equations. This approach illustrates that a synchronous machine is an inductance with nonlinear coupling to its rotor from the perspective of ac networks. This model is useful because it can be used to diagnose numerical deficiencies in ac system models such as floating subnetworks. The new stator model was combined with rotor and control models used in existing EMT and conventional transient stability programs. Control components in realistic power system models come in many different types and configurations. A generic control block system was developed as part of this work, which allows new control system components to be rapidly developed and implemented. Finally, an optional multimass component based on existing multimass models was also implemented as part of the synchronous machine subsystem, which is important for simulating subsynchronous oscillations.

An HVdc transmission system model that is compatible with the proposed method was also implemented in this work. This model is based on the CIGRE HVdc benchmark case to provide a structure and component values. However, the individual components of the HVdc system were developed independent of each other and only coupled as required. This approach illustrates that the components can be used in different HVdc system arrangements. The converter model implemented in this work is based on dynamic phasor-based modeling approaches developed in the literature and uses steady state dc voltage and ac current expressions to construct a dynamic phasor model. The ac current dynamic phasor equation developed using this procedure includes an ac voltage contribution, which was found to be an important factor for accurately simulating low impedance faults near a converter's terminal. Part of the HVdc converter model's mathematical description includes an inverse cosine, which implies that there are domain limitations on the model. These limitations were shown to be related to adverse operating conditions, which must be simulated using different modes of operation. The converter model developed in this work includes alternate states and events to handle transitions between those states. It was found that these operating states are essential to reliably simulate the HVdc model during ac system faults.

The value of this work was demonstrated through simulation of realistic power system models. Results are presented of the IEEE 68 bus test system, which was used as a base case.

The results demonstrated that the dynamic phasor program was in excellent agreement with commercial simulators for both electromechanical and electromagnetic transients. Simulations of a five hundred bus test system were also included in this work. The first iteration of this system was a purely ac network model, used as a benchmark of the case. The simulation results indicate that the CPU time required for simulations is sensitive to location, with an order of magnitude separating the slowest and fastest simulation times. However, the majority of times were found to be well clustered and approximately 75 % of cases required less than four seconds of CPU time. These results further demonstrated that 75 % of cases simulated with the dynamic phasor program were 143 times faster than EMT simulations. Finally, comparisons of various network quantities illustrated excellent agreement between the dynamic phasor program and EMT simulations for both electromechanical and network transients.

The five hundred bus test system was modified to illustrate the dynamic phasor program's ability to simulate systems with multiple HVdc transmission lines. Two ac lines were replaced with HVdc transmission systems. The simulation results indicate that the dynamic phasor program is approximately three hundred times faster than the EMT simulator for the selected disturbance. The results also show that the dynamic phasor simulator is capable of accurately reproducing both dc system dynamics and the ac system dynamics of nearby generators. This result illustrates that the dynamic phasor method is suitable for simulation of models that include both synchronous generators and HVdc systems.

The five hundred bus test system was further modified to include a multimass turbine model. The multimass turbine was modeled on a generator near one of the rectifier terminals. The control system of the nearby HVdc system was purposely tuned to induce subsynchronous oscillations in the multimass generator model. The results illustrate that the damping of unstable torsional modes simulated by the dynamic phasor program is sensitive to relative error tolerance. The tolerance used in previous simulations results in error that grows over time. However, the results also show that the frequency of unstable torsional modes is not sensitive relative error tolerance and was accurately predicted in all three trials. Comparisons between the dynamic phasor program's results and the EMT program also demonstrate some error in the damping of the unstable mode. However, the frequency is once again identical and the initial oscillations show an identical match for both the multimass turbine speed and the rectifier current nearest the unstable generator. Therefore, the two programs were considered to be in good agreement. Furthermore, the dynamic phasor method was still between 36 and 122 times faster than the EMT program.

The final test system presented in this thesis is a purely ac system with approximately two thousand buses. EMT simulations were not conducted for this system given the system size and resource limitations. A fault was conducted at each bus and the results demonstrated that the majority of simulations required less than 40 s of CPU time. Comparing these results with the five hundred bus timing information illustrates that the dynamic phasor program is significantly faster than the EMT simulations for a system that is four times larger. The simulation results also demonstrate good agreement between the two programs for both stator and electromechanical transients.

7.2 Limitations and Future Work

At this time it is unknown how well this method will scale for extremely large models such as those that represent a large geographic region such as North America. It is difficult to predict how the program will scale given the complexity of the factors influencing CPU time. The simulation results of the two thousand bus system demonstrate that the method is promising for large systems. However, it is expected that modifications will be required to simulate extremely large network models. An area that can be explored is the use of hybrid ac network models, which have been explored in both dynamic phasor-based small signal analysis and co-simulation using EMT and transient stability simulations [28], [34]. Both techniques define a detailed region modeled either using dynamic phasors or an EMT simulator, which is where disturbances are simulated and high resolution results are required. The remainder of the network where detailed results are not required is modeled or simulated using conventional transient stability methods. This approach tends to yield accurate results provided the study area is large enough. A hybrid ac model has the potential to be a major performance enhancement for the dynamic phasor method because quasistationary network models can be reduced to simulate only dynamic nodes [2]. This reduction procedure is not possible for differential nodes in a dynamic phasor network model and is complicated for algebraic nodes.

An interesting area of future work is to conduct a more thorough investigation into sparse iterative solvers for the linear system solution step of the Newton-Raphson process. Preliminary tests illustrated that iterative solvers did not perform as well as direct solvers as discussed in Section 3.2.2. These tests were conducted with a general purpose incomplete *LU* preconditioner, which do not take advantage of any specific structural information available from the system model. However, a preconditioning technique specifically designed for dynamic phasor-based power system models could improve convergence rates. Iterative solvers could be a valuable option if it is found that linear solve times are a bottleneck in scaling the program up to large power system models.

Application of high performance computing methods represents an interesting and exciting area of future work on this project. Investigating parallel processing will be an important area of research in applying the dynamic phasor-based method for extremely large systems. The performance of operations such as the residual and Jacobian computations can be improved through threading models. The high performance multithreaded and GPU-based linear solvers packaged with `IDA` should be investigated for larger system models [53]. Finally, traveling wave models are used in EMT simulation to introduce natural network partitions, which produce subnetworks that can be solved in parallel [15]. It is possible to develop a dynamic phasor-based traveling wave model [89]. Therefore, the method presented in this work could be extended to include a traveling wave transmission line model. Implementing a traveling wave model represents an interesting challenge in this area of research as it introduces periodic events and time step limitations that must be balanced with the advantages offered by network partitioning.

Modern power systems consist of a diverse set of components from many different machine control models to new power electronic-based renewable generation. The performance of the machine and HVdc system model developed in this work suggests that dynamic phasor-based simulations are capable of accurately modeling and simulating different controllers and power electronic components. However, simulations with a wider variety of components is required to examine the method with more realistic set of models. Therefore, there is a significant amount of interesting future work to develop new models for solar and wind generation, a wide range of FACTS devices, and complex loads such as induction motors.

Bibliography

[1] P. Kundur, N. Balu, and M. Lauby, *Power System Stability and Control.* McGraw-Hill Education, 1994.

[2] W. Stevenson, *Elements of Power System Analysis* (Electrical and Electronic Engineering Series). McGraw-Hill, 1955.

[3] D. Shankle, C. Murphy, R. Long, and E. Harder, “Transient stability studies – I synchronous and induction machines,” *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 73, no. 2, pp. 1563–1580, Jan. 1954.

[4] M. Dyrkacz and D. Lewis, “A new digital transient stability program,” *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 78, no. 3, pp. 913–918, 1959.

[5] M. Dyrkacz, C. Young, and F. Maginniss, “A digital transient stability program including the effects or regulator, exciter, and governor response,” *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 79, no. 3, pp. 1245–1254, Apr. 1960.

[6] V. Venkatasubramanian, H. Schättler, and J. Zaborszky, “Fast time-varying phasor analysis in the balanced three-phase large electric power system,” *IEEE Transactions on Automatic Control*, vol. 40, no. 11, pp. 1975–1982, Nov. 1995.

[7] H. Rustebakke and C. Concordia, “Self-excited oscillations in a transmission system using series capacitors,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 7, pp. 1504–1512, Sep. 1970.

[8] L. Kilgore, L. Elliott, and E. Taylor, “The prediction and control of self-excited oscillations due to series capacitors in power systems,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, no. 3, pp. 1305–1311, May 1971.

[9] J. Undrill and T. Kostyniak, “Subsynchronous oscillations part 1 – comprehensive system stability analysis,” *IEEE Transactions on Power Apparatus and Systems*, vol. 95, no. 4, pp. 1446–1455, Jul. 1976.

[10] M. Bahrman, E. Larsen, R. Piwko, and H. Patel, “Experience with hvdc - turbine-generator torsional interaction at square butte,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-99, no. 3, pp. 966–975, May 1980.

[11] K. Mortensen, E. Larsen, and R. Piwko, “Field tests and analysis of torsional interaction between the coal creek turbine-generators and the cu hvdc system,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 1, pp. 336–344, Jan. 1981.

[12] M. Parniani and M. Iravani, “Computer analysis of small-signal stability of power systems including network dynamics,” *IEE Proceedings - Generation, Transmission and Distribution*, vol. 142, no. 6, pp. 613–617, Nov. 1995.

[13] J. Zaborszky, H. Schättler, and V. Venkatasubramanian, “Error estimation and limitation of the quasi stationary phasor dynamics,” in *Proc. Power Systems Computation Conference*, 1993, pp. 721–729.

[14] R. Farmer, A. Schwalb, and E. Katz, “Navajo project report on subsynchronous resonance analysis and solutions,” *IEEE Transactions on Power Apparatus and Systems*, vol. 96, no. 4, pp. 1226–1232, Jul. 1977.

[15] N. Watson and J. Arrillaga, *Power Systems Electromagnetic Transients Simulation* (IET Power and Energy Series). The Institution of Engineering and Technology, 2003.

[16] Z. Zhou and V. Dinavahi, “Fine-grained network decomposition for massively parallel electromagnetic transient simulation of large power systems,” *IEEE Power and Energy Technology Systems Journal*, vol. 4, no. 3, pp. 51–64, Sep. 2017.

[17] D. Shu, X. Xie, V. Dinavahi, C. Zhang, X. Ye, and Q. Jiang, "Dynamic phasor based interface model for EMT and transient stability hybrid simulations," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3930–3939, Jul. 2018.

[18] K. Padiyar and M. Geetha, "Study of torsional interactions in multi-terminal dc systems through small signal stability analysis," in *International Conference on AC and DC Power Transmission*, Sep. 1991, pp. 411–413.

[19] P. Sauer, B. Lesieutre, and M. Pai, "Transient algebraic circuits for power system dynamic modelling," *International Journal of Electrical Power & Energy Systems*, vol. 15, no. 5, pp. 315–321, 1993.

[20] C. L. DeMarco and G. C. Verghese, "Bringing phasor dynamics into the power system load flow," in *1993 North American Power Symposium*, Oct. 1993, pp. 463–471.

[21] V. Venkatasubramanian, "Tools for dynamic analysis of the general large power system using time-varying phasors," *International Journal of Electrical Power & Energy Systems*, vol. 16, no. 6, pp. 365–376, 1994.

[22] S. Henschel, "Analysis of electromagnetic and electromechanical power system transients with dynamic phasors," Ph.D. dissertation, University of British Columbia, Feb. 1999.

[23] K. Strunz, R. Shintaku, and F. Gao, "Frequency-adaptive network modeling for integrative simulation of natural and envelope waveforms in power systems and circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 12, pp. 2788–2803, Dec. 2006.

[24] S. Sanders, J. Noworolski, X. Liu, and G. Verghese, "Generalized averaging method for power conversion circuits," *IEEE Transactions on Power Electronics*, vol. 6, no. 2, pp. 251–259, Apr. 1991.

[25] E. Zhijun, K. Chan, and D. Fang, "A practical dynamic phasor model of static VAR compensator," in *2006 2nd International Conference on Power Electronics Systems and Applications*, Nov. 2006, pp. 23–27.

[26] M. Daryabak, S. Filizadeh, J. Jatskevich, *et al.*, “Modeling of LCC-HVDC systems using dynamic phasors,” *IEEE Transactions on Power Delivery*, vol. 29, no. 4, pp. 1989–1998, Aug. 2014.

[27] F. Jusan, S. Gomes, and G. Taranto, “SSR results obtained with a dynamic phasor model of SVC using modal analysis,” *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 6, pp. 571–582, 2010.

[28] K. Mudunkotuwa, S. Filizadeh, and U. Annakkage, “Development of a hybrid simulator by interfacing dynamic phasors with electromagnetic transient simulation,” *IET Generation, Transmission Distribution*, vol. 11, no. 12, pp. 2991–3001, 2017.

[29] H. Zhu, Z. Cai, H. Liu, Q. Qi, and Y. Ni, “Hybrid-model transient stability simulation using dynamic phasors based HVDC system model,” *Electric Power Systems Research*, vol. 76, no. 6, pp. 582–591, 2006.

[30] C. Karawita, D. Suriyaarachchi, M. Mohaddes, D. Kell, R. Ostash, and T. Maguire, “Generic VSC models for project planning studies,” in *2014 CIGRE Session*, Aug. 2014, pp. 1–11.

[31] E. Zhijun, D. Fang, K. Chan, and S. Yuan, “Hybrid simulation of power systems with SVC dynamic phasor model,” *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 5, pp. 175–180, 2009.

[32] P. Mattavelli, A. Stankovic, and G. Verghese, “SSR analysis with dynamic phasor model of thyristor-controlled series capacitor,” *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 200–208, Feb. 1999.

[33] C. Karawita and U. Annakkage, “Multi-infeed HVDC interaction studies using small-signal stability assessment,” *IEEE Transactions on Power Delivery*, vol. 24, no. 2, pp. 910–918, Apr. 2009.

[34] C. Karawita and U. D. Annakkage, “A hybrid network model for small signal stability analysis of power systems,” *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 443–451, Feb. 2010.

[35] U. Ascher and L. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1998.

[36] J. Noworolski and S. Sanders, “Generalized in-plane circuit averaging,” in *[Proceedings] APEC ’91: Sixth Annual Applied Power Electronics Conference and Exhibition*, Mar. 1991, pp. 445–451.

[37] R. Shintaku and K. Strunz, “Branch companion modeling for diverse simulation of electromagnetic and electromechanical transients,” *Electric Power Systems Research*, vol. 77, no. 11, pp. 1501–1505, 2007.

[38] M. Kulasza, “Generalized dynamic phasor-based simulation for power systems,” M.S. thesis, University of Manitoba, Jan. 2015.

[39] M. Mirz, S. Vogel, G. Reinke, and A. Monti, “DPsim – a dynamic phasor real-time simulator for power systems,” *SoftwareX*, vol. 10, pp. 1–8, 2019.

[40] T. Noda, T. Kikuma, T. Nagashima, and R. Yonezawa, “A dynamic-phasor simulation method with sparse tableau formulation for distribution system analysis: A preliminary result,” in *2018 Power Systems Computation Conference (PSCC)*, Jun. 2018, pp. 1–7.

[41] T. Demiray, “Simulation of power system dynamics using dynamic phasor models,” Ph.D. dissertation, Swiss Federal Institute of Technology, 2008.

[42] T. Demiray, G. Andersson, and L. Busarello, “Evaluation study for the simulation of power system transients using dynamic phasor models,” in *2008 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, Aug. 2008, pp. 1–6.

[43] E. Kuh and R. Rohrer, “The state-variable approach to network analysis,” *Proceedings of the IEEE*, vol. 53, no. 7, pp. 672–686, Jul. 1965.

[44] R. Riaza, “DAEs in circuit modelling: A survey,” in *Surveys in Differential-Algebraic Equations I*, A. Ilchmann and T. Reis, Eds. Springer Berlin Heidelberg, 2013, pp. 97–136.

[45] B. Petersen, “Investigating solvability and complexity of linear active networks by means of matroids,” *IEEE Transactions on Circuits and Systems*, vol. 26, no. 5, pp. 330–342, May 1979.

[46] A. Recski, “Unique solvability and order of complexity of linear networks containing memoryless n-ports,” *International Journal of Circuit Theory and Applications*, vol. 7, no. 1, pp. 31–42, 1979.

[47] G. Reißig, “Extension of the normal tree method,” *International Journal of Circuit Theory and Applications*, vol. 27, no. 2, pp. 241–265, 1999.

[48] R. Newcomb and B. Dziurla, “Some circuits and systems applications of semistate theory,” *Circuits, Systems and Signal Processing*, vol. 8, no. 3, pp. 235–260, Sep. 1989.

[49] R. Rohrer, “Circuit simulation – the early years,” *IEEE Circuits and Devices Magazine*, vol. 8, pp. 32–37, May 1992.

[50] J. Astic, A. Bihain, and M. Jerosolimski, “The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems,” *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 929–935, May 1994.

[51] B. Haut, V. Savcenco, and P. Panciatici, “A multirate approach for time domain simulation of very large power systems,” in *2012 45th Hawaii International Conference on System Sciences*, Jan. 2012, pp. 2125–2132.

[52] P. Gibert, P. Panciatici, R. Losseau, A. Guironnet, D. Tromeur-Dervout, and J. Erhel, “Speedup of EMT simulations by using an integration scheme enriched with a predictive Fourier coefficients estimator,” in *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Oct. 2018, pp. 1–6.

[53] A. Hindmarsh, P. Brown, K. Grant, *et al.*, “SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[54] R. C. Dorf and J. A. Svoboda, *Introduction to Electric Circuits*, 7th ed. John Wiley & Sons, 2010.

[55] C. Ho, A. Ruehli, and P. Brennan, “The modified nodal approach to network analysis,” *IEEE Transactions on Circuits and Systems*, vol. 22, no. 6, pp. 504–509, Jun. 1975.

[56] D. Estévez Schwarz and C. Tischendorf, "Structural analysis of electric circuits and consequences for MNA," *International Journal of Circuit Theory and Applications*, vol. 28, no. 2, pp. 131–162, 2000.

[57] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York, NY, USA: John Wiley & Sons, Inc., 1983.

[58] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, "Using Krylov methods in the solution of large-scale differential-algebraic systems," *SIAM Journal on Scientific Computing*, vol. 15, no. 6, pp. 1467–1488, 1994.

[59] G. Mao and L. Petzold, "Efficient integration over discontinuities for differential-algebraic systems," *Computers & Mathematics with Applications*, vol. 43, pp. 65–79, Jan. 2002.

[60] B. Bramas and P. Kus, "Computing the sparse matrix vector product using block-based kernels without zero padding on processors with AVX-512 instructions," *PeerJ Computer Science*, 2018, 4.

[61] A. Kahn, "Topological sorting of large networks," *Communications of the ACM*, vol. 5, no. 11, pp. 558–562, Nov. 1962.

[62] C. Sanderson and R. Curtin, "Armadillo: A template-based C++ library for linear algebra," *Journal of Open Source Software*, vol. 1, no. 2, p. 26, 2016.

[63] X. Li, "An overview of superlu: Algorithms, implementation, and user interface," *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 302–325, Sep. 2005.

[64] E. Natarajan, "KLU – a high performance sparse linear solver for circuit simulation problems," M.S. thesis, University of Florida, 2005.

[65] L. Razik, L. Schumacher, A. Monti, A. Guironnet, and G. Bureau, "A comparative analysis of LU decomposition methods for power system simulations," in *2019 IEEE Milan PowerTech*, Jun. 2019, pp. 1–6.

[66] C. Tischendorf, "Coupled systems of differential algebraic and partial differential equations in circuit and device simulation," Habilitationsschrift, Humboldt-Universtät zu Berlin, Mar. 2003.

[67] P. Brown, A. Hindmarsh, and L. Petzold, "Consistent initial condition calculation for differential-algebraic systems," *SIAM Journal on Scientific Computing*, vol. 19, no. 5, pp. 1495–1512, Jun. 1998.

[68] L. Mandache, D. Topan, and I. Sirbu, "Improved modified nodal analysis of nonlinear analog circuits in the time domain," in *Proceedings of the World Congress on Engineering 2010 Vol II*, Jun. 2010, pp. 1–4.

[69] C. Karawita, "HVDC interaction studies using small signal stability assessment," Ph.D. dissertation, University of Manitoba, Apr. 2009.

[70] K. Corzine, B. Kuhn, S. Sudhoff, and H. Hegner, "An improved method for incorporating magnetic saturation in the q–d synchronous machine model," *IEEE Transactions on Energy Conversion*, vol. 13, no. 3, pp. 270–275, Sep. 1998.

[71] E. Allen and M. Ilić, "Interaction of transmission network and load phasor dynamics in electric power systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 11, pp. 1613–1620, Nov. 2000.

[72] P. Zhang, J. Martí, and H. Dommel, "Synchronous machine modeling based on shifted frequency analysis," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1139–1147, Aug. 2007.

[73] Y. Huang, M. Chapariha, F. Therrien, J. Jatskevich, and J. Martì, "A constant-parameter voltage-behind-reactance synchronous machine model based on shifted-frequency analysis," *IEEE Transactions on Energy Conversion*, vol. 30, no. 2, pp. 761–771, Jun. 2015.

[74] "IEEE recommended practice for excitation system models for power system stability studies," *IEEE Std 421.5-2016 (Revision of IEEE Std 421.5-2005)*, pp. 1–207, 2016.

[75] "Dynamic models for turbine-governors in power system studies," IEEE PES Task force on Turbine-Governor Modeling, Tech. Rep., Jun. 2013.

[76] "First benchmark model for computer simulation of subsynchronous resonance," *IEEE Transactions on Power Apparatus and Systems*, vol. 96, no. 5, pp. 1565–1572, 1977.

[77] M. Szechtman, T. Weiss, and C. Thio, "First benchmark model for HVDC control studies," *Electra*, no. 135, Apr. 1991.

[78] M. Szechtman, T. Wess, and C. Thio, “A benchmark model for HVDC system studies,” in *International Conference on AC and DC Power Transmission*, 1991, pp. 374–378.

[79] J. Kassakian, M. Schlecht, and G. Verghese, *Principles of Power Electronics* (Addison-Wesley series in electrical engineering). Addison-Wesley, 1991.

[80] W. Hammer, “Dynamic modeling of line and capacitor commutated converters for HVDC power transmission,” Ph.D. dissertation, Swiss Federal Institute of Technology, Dec. 1972.

[81] M. Daryabak, “Modeling line-commutated converter HVDC transmission systems using dynamic phasors,” Ph.D. dissertation, University of Manitoba, Apr. 2016.

[82] M. Faruque, Y. Zhang, and V. Dinavahi, “Detailed modeling of CIGRE HVDC benchmark system using PSCAD/EMTDC and PSB/SIMULINK,” *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 378–387, 2006.

[83] A. Vandaei, S. Filizadeh, K. Mudunkotuwa, and E. Tara, “Real-time implementation of an enhanced dynamic phasor-based three-phase phase-locked loop for line-commutated converters,” in *2018 IEEE International Conference on Industrial Technology (ICIT)*, 2018, pp. 812–817.

[84] M. Daryabak, S. Filizadeh, and A. Vandaei, “Dynamic phasor modeling of LCC-HVDC systems: Unbalanced operation and commutation failure,” *Canadian Journal of Electrical and Computer Engineering*, vol. 42, no. 2, pp. 121–131, 2019.

[85] A. Birchfield, T. Xu, K. Gegner, K. Shetye, and T. Overbye, “Grid structural characteristics as validation criteria for synthetic networks,” *IEEE Transactions on power systems*, vol. 32, no. 4, pp. 3258–3265, 2016.

[86] T. Xu, A. Birchfield, K. Shetye, and T. Overbye, “Creation of synthetic electric grid models for transient stability studies,” in *The 10th Bulk Power Systems Dynamics and Control Symposium (IREP 2017)*, 2017, pp. 1–6.

[87] “SouthCarolina 500-bus system: ACTIVSg500.” (2021), [Online]. Available: `https://electricgrids.engr.tamu.edu/electric-grid-test-cases/activsg500` (visited on 10/20/2021).

[88] "ACTIVSg2000: 2000-bus synthetic grid on footprint of Texas." (2021), [Online]. Available: `https://electricgrids.engr.tamu.edu/electric-grid-test-cases/activsg2000` (visited on 10/20/2021).

[89] V. Venkatasubramanian, H. Schättler, and J. Zaborszky, "A time-delay differential algebraic phasor formulation of the large power system dynamics," in *1994 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 6, May 1994, pp. 49–52.