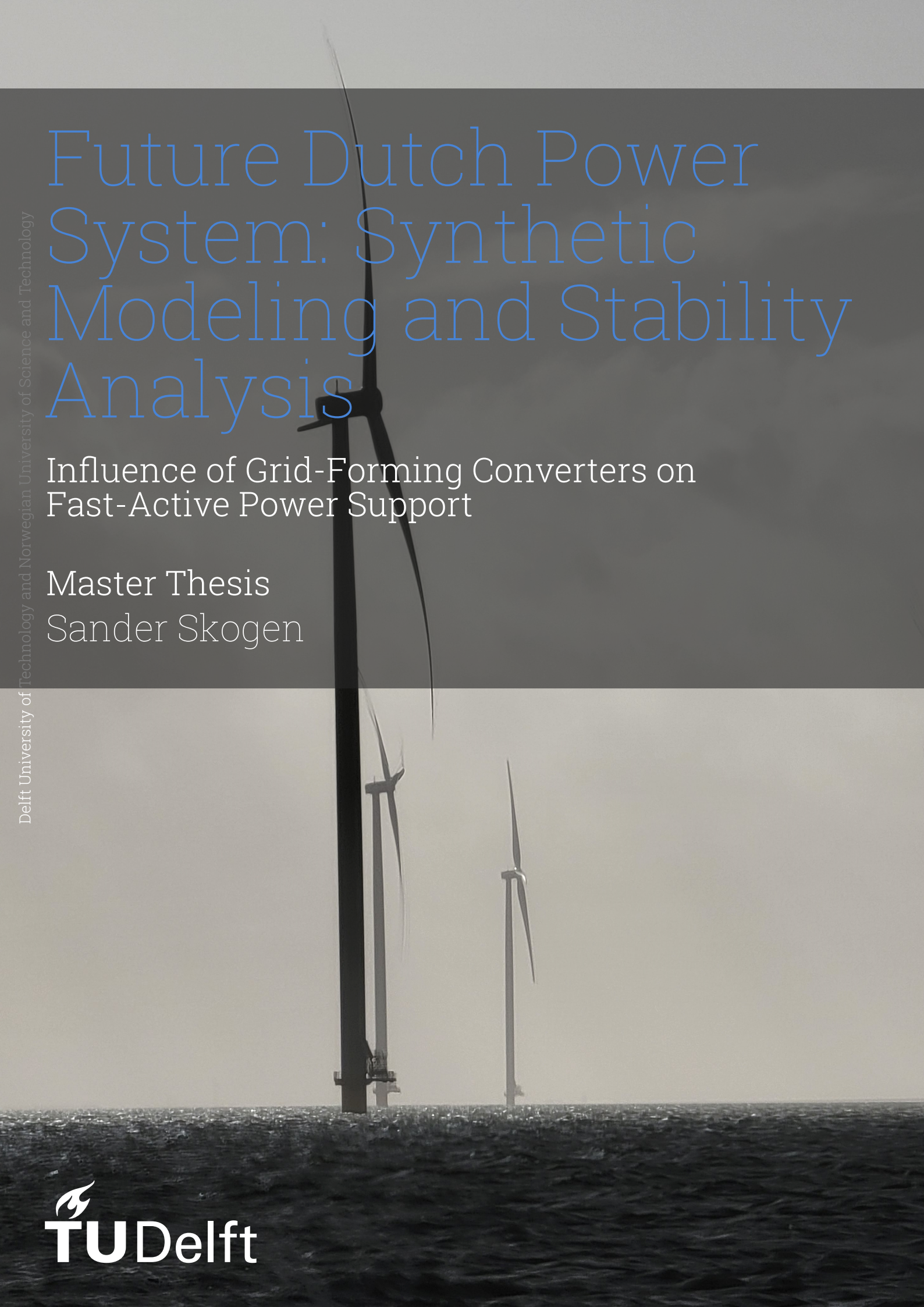
Future Dutch Power System: Synthetic Modeling and Stability Analysis

Influence of Grid-Forming Converters on Fast-Active Power Support

Master Thesis

Sander Skogen

Delft University of Technology and Norwegian University of Science and Technology



TU Delft

Future Dutch Power System: Synthetic Modeling and Stability Analysis

Influence of Grid-Forming Converters on Fast-Active Power Support

by

Sander Skogen

To obtain the degree of Master of Science at the Delft University of Technology and Norwegian University of Science and Technology.
to be defended publicly on 25.06.2024.

Student number:	517300 (NTNU), 5829593 (TU Delft)	
Project Duration:	6 November, 2023 - 1 July, 2024	
Faculty:	EEMCS, TU Delft	
	IE, NTNU	
Thesis committee:	Dr. Ir. José L. Rueda Torres	TU Delft
	Dr. Zian Qin	TU Delft
	H. (Hani) Vahedi	TU Delft
	Gilbert Bergna-Diaz	NTNU

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft NTNU

Acknowledgements

As my educational journey ends, I would like to acknowledge some important people who helped me along the way.

I would like to express my deepest gratitude to my main professor, Dr.ir. Jose Rueda Torres, for his invaluable guidance, insightful feedback, and support. His mentorship has been instrumental in shaping this research, and he has significantly contributed to my overall learning experience over the past two years.

This Master's program has given me the incredible opportunity to live and study in Delft, Copenhagen, and Trondheim. I am grateful for the experiences, knowledge, and friendships I have gained along the way.

I would also like to thank my co-students for their camaraderie and support throughout this journey.

Finally, I am deeply thankful for my family, friends, and girlfriend's unwavering support. Their encouragement and belief in me have been a constant source of strength, and this would not have been possible without them.

Thank you.

Sander Skogen
Delft, June 2024

Abstract

As the global shift towards renewable energy accelerates, it is crucial to address the inherent challenges and fully leverage its potential. Electric power systems have undergone significant transformations, transitioning from traditional generation methods to those incorporating renewable energy sources and power electronic interfaces (PEI). This transition necessitates extensive research to ensure future power system stability and reliability. This thesis work aimed to enhance a Root Mean Square (RMS) synthetic model of the future 380 kV Dutch power system and conducted comprehensive research using this model.

A scenario analysis was conducted to evaluate various future scenarios for the Dutch power system, highlighting deviations and uncertainties. The national scenario from the II3050-2 project was selected as the baseline for further work. This approach ensured a comprehensive understanding of potential future developments and their implications for policy and planning.

The synthetic model was developed by updating the existing model with new dynamic models and parameters as defined by IEEE and CIGRE standards and aligning it with future public projections and investment plans for the Dutch power system towards 2050, rooted in the scenario analysis. This included upgrading infrastructure such as transmission lines and substations and incorporating new generation and flexibility resources. The model was designed to enhance the dynamic characteristics and facilitate extensive research by integrating DigSilent PowerFactory, Python, and Excel.

Three different case studies were performed using the enhanced synthetic model to analyze the system's response to various perturbations. These studies evaluated the impact of varying levels of renewable generation and load, the role of grid-forming converters, and the effects of different kinetic energy and inertia constant levels on system stability with respect to frequency and rotor angle stability. The findings underscored the significant potential of grid-forming converters to enhance frequency stability and system damping, the influence of controller parameters on system performance, and the critical role of kinetic energy and inertia in maintaining system resilience.

By leveraging the capabilities of PowerFactory, Python, and Excel, the research demonstrated the utility of the enhanced synthetic model in facilitating a broad range of stability studies and other analyses. This work provides a valuable foundation for future research and regulatory planning, contributing to the ongoing efforts of Dr. J.L. Rueda Torres' team. The findings have been submitted to international journals and conferences, emphasizing the importance of continued technological adaptation and research to ensure stable and resilient power systems.

Future work should focus on developing smart control strategies, optimizing dynamic control settings, investigating the impacts of kinetic energy and inertia constants, enhancing the synthetic dynamic model with additional features, and assessing the role of flexibility resources in future scenarios. This research offers crucial insights into the evolving dynamics of power systems and sets the stage for further advancements to ensure their stability and resilience.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
BESS	Battery Energy Storage Systems
DFIG	Doubly Fed Induction Generators
EMT	Electromagnetic Transient
ESS	Energy Storage Systems
PEI	Power Electronic Interface
PSS	Power System Stabilizers
P2X	Power to X
RES	Renewable Energy Sources
RMS	Root Mean Square
WOZ	Offshore Wind
WOL	Onshore Wind
PV	Solar PV
STATCOM	Static Synchronous Compensators
TSO	Transmission System Operator
VRES	Variable Renewable Energy Sources
VRG	Variable Renewable Generation
VSC	Voltage Source Converter
MMC	Modular Multilevel Converters
HVDC	High Voltage DC
DSL	Digsilent Simulation language

1 Introduction

1.1. Background

Electric power systems are a key element of human existence. They play a part in all aspects of life, from basic necessities for survival, such as food supply and heating, to excessive acts, such as travel and information. Power systems and electricity have changed how humans exist on Earth and have been a key part of history's largest technological advancement period. This makes upgrading and maintaining power systems a crucial element for future life.

Electric power systems have seen a resurgence of technological advancements in the last decade. The shift from traditional generation and load sources, such as coal and gas, to renewable generation, introducing power electronic interfaced generation (PEI), has transformed system operation from its traditional use to a more advanced and dynamic setting. This shift has significantly changed power system engineering, spanning all power system domains to account for the change. Forecasting, control, and assessment become inherently important as systems move from traditional, slow, and predictable to more dynamic and unpredictable operations. In the upcoming decades, it will be vital for engineers and researchers to keep up with the necessary developments to ensure a sustainable future with a reliable power system.

Transitioning to renewable energy production offers numerous benefits, but it has challenges. The landscape of renewable generation can be partitioned into two distinct categories: PEI and non-PEI generation.
Non-PEI Generation: This group encompasses renewable sources like hydropower, biomass, and geothermal. These sources are directly connected to the power system and resonate with characteristics similar to traditional fossil fuel-based generators. When a power system is dominated by non-PEI renewables, its operation largely mirrors that of a conventional system. Nonetheless, they come with unique challenges. A quintessential example is hydropower, which, despite its stability, is contingent upon adequate rainfall patterns. A consistent downpour ensures optimal power production, but discrepancies in rainfall can disrupt its reliability.
PEI Generation: This category signifies a different set of challenges. Power electronics and storage devices are necessary for power sources with pronounced variability, often termed variable renewable generation (VRG). The most prevalent is wind and solar energy. Their variability stands in stark contrast to the seasonal consistency of hydropower. For wind and solar energies, variability is multifaceted, spanning from annual cycles down to momentary fluctuations [1].

The variability of PEI generation has resulted in these sources being dependent upon additional control through PEI to connect to the power system, while non-PEI generation is typically directly connected to the system using synchronous machines, similar to traditional generation units. This change from dominantly directly connected to PEI-connected generation leads to a decrease in the overall inertia levels of the system, which has traditionally been seen as one of the most important parameters for stability [2].

Power electronics' rising dominance and complexities underline the significance of power system operation and stability. This evolving landscape necessitated IEEE/Cigre to update their standards, transitioning from a traditional three-part stability classification to a more nuanced five-part one due to the changing dynamics and phenomena associated with PEI dominant generation [3].

Due to the significant evolution anticipated in future power systems, extensive research is being conducted to address forthcoming challenges and devise solutions. The stability and functionality of future power systems will hinge on rigorous research and interdisciplinary collaboration among different vendors, agencies, and countries. Major advancements have been implemented to account for the changing power system. Among them are the increased potential of large-scale energy storage units for power systems, improved communication and information regarding wide-area control systems and intelligent system management, and a general improvement in the power electronic domain, which serves as an increasing key for multiple aspects of power systems [4, 5, 6]. However, major advancements are still necessary to ensure the reliable operation of future power systems.

A key innovation under investigation is grid-forming converters, which differ from traditional grid-following converters by providing voltage and power control independently of the power system. These converters are considered a potential solution to the diminishing number of traditional generators, critical for maintaining system stability, and are thus a focal point of ongoing research [7]. Despite their promise, grid-forming converters require further study. The academic community has been urged to establish benchmark test systems for these converters and develop suitable testing and validation procedures [7]. Future research directions include developing new simulation tools to adapt to new system dynamics, examining the real-world impacts and interactions of grid-forming converters, and understanding their influence on system stability at different penetration levels of grid-following and grid-forming technologies [8, 9]. This thesis project aims to address some of these gaps.

Overall, the evolving landscape of power systems necessitates ongoing research and adaptability, particularly with the increasing penetration of power electronic interfaces (PEIs), which introduce dynamic challenges such as time-varying frequencies and oscillation damping [10, 11]. To tackle these challenges, this thesis project employs a three-step approach to integrate and analyze the theoretical futuristic synthetic Dutch power system and assess grid-forming technologies as a stability resource for future systems.

As the global shift towards renewable energy accelerates, addressing the inherent challenges and fully leveraging its potential is crucial. The rapid transition necessitates research efforts to integrate renewables into national power systems, ensuring system stability and efficiency. The project aims to bridge the gap between renewable energy goals and the realities of power system operations by utilizing advanced tools such as Digsilent PowerFactory 2023 version SP6, combined with Python version 3.11 and Excel Version 2404, hereafter referred to as PowerFactory, Python and Excel respectively. Utilizing these resources, different case studies will be conducted with the aim of addressing some of the gaps in literature by publishing scientific papers in international journals and conferences on these issues.

1.2. Literature Review

This literature review delves into the evolving landscape of power systems, specifically focusing on grid-forming converters and the simulation of power systems. As power systems transition towards increased integration of renewable energy sources (RES) and PEI generation, there is a critical need to develop innovative solutions and advanced simulation techniques to maintain system stability and reliability. This review highlights recent advancements and pilot projects and identifies gaps for future research in these areas.

1.2.1. Grid-Forming Converters

Grid-forming converters have emerged as a pivotal technology in modern power systems, offering the capability to provide voltage and frequency control independently of the power system. This section reviews the different types of grid-forming converters, recent advancements, pilot projects, and gaps for future research.

The primary distinction between grid-forming and grid-following technologies lies in their controllability. Grid-following converters have power and current control loops, enabling control over active and reactive power [12]. They rely on an AC signal from the externally connected power system, typically using a phase-locked loop (PLL) for control. This dependency makes them susceptible to errors during transient events or in weak-grid and islanded modes [12, 13]. In contrast, grid-forming converters can operate independently of the power system, providing voltage and frequency control without relying on an external AC signal. This capability is essential for maintaining system stability, especially as RES and PEI generation increasingly dominate the power system. This essential difference allows for the grid-forming converters' ancillary services during transient operation, such as contributing to fast-active power balancing [14].

Three types of grid-forming controllers have been identified and are subject to further analysis: Virtual Synchronous Machine (VSM), Synchroverter, and Droop. These controllers, representing different control approaches, are analyzed using generic models available in the extensive PowerFactory library. This chosen simulation software significantly enhances the availability of these controllers and allows for a robust comparative analysis.

Virtual Synchronous Machine (VSM): Virtual Synchronous Machines (VSM) are designed to mimic traditional synchronous generators, ranging from simple droop control models to complex 7th-order synchronous machine models. The first VSM was introduced by Beck and Hesse in 2007 [13, 15]. Although high-order models theoretically offer detailed control, lower-order models have proven more stable and controllable, particularly during fault events [16]. VSMs provide synthetic inertia and damping using control techniques combined with power electronics, bridging the gap left by the reduced inherent inertia in PEI generation. They are effective in various applications, including microgrids and islanded modes, but still face limitations, especially in short-circuit fault scenarios [17]. The VSM controller has shown promise in smaller IEEE test systems and large-scale studies on the European power system [18]. Current research focuses on integrating energy storage, tuning converters, and developing higher-order models [16, 19]. Recent developments in VSM controllers, particularly for battery energy storage systems (BESS), have demonstrated promising results in real-life implementations, such as in southern Australia [20, 21]. However, further research is necessary, including analyzing different grid-forming controller generation sources, large-scale implementation, and establishing general policy and guidelines to better align and converge on the path forward for these controllers [20, 21, 22].

Synchroverter: The Synchroverter mimics synchronous machines using torque control instead of power, offering enhanced mechanical signal accuracy and stability under certain conditions [23, 24]. This approach also allows for the simulation and inclusion of electromagnetic torque [25]. It uses frequency and voltage control loops for active and reactive power, respectively, and has an integrated damping correction loop to improve stability [26]. While it provides better reactive power control and adapts well to changing power system dynamics, its increased complexity and reliance on advanced control systems pose implementation challenges [27]. Another inherent drawback identified for the Synchroverter is the poor damping of oscillations, which could impact system stability [28]. Various

research efforts are underway to address these issues, including adding new first-order control blocks to the control structure and improving the damping structure [28, 29, 30]. Current research focuses on integrating energy storage and optimizing control systems to make Synchroverters a viable future solution [31, 32]. Future research is expected to extend current efforts to large-scale system models, various resource implementation schemes, and real-life integration and testing.

Droop Control: Traditional Droop control is a well-established method used in microgrids and synchronous machine controls, leveraging frequency and voltage droop to regulate active and reactive power [33, 34]. Its communication-less design allows for easy implementation and adjustment, but it results in slower transient responses and less effective transient power-sharing [12]. Proven effective in both grid-connected and stand-alone microgrids, droop control continues to evolve with various proposed types, including single-loop, multi-loop, and dual-loop techniques [35, 36, 37]. However, Droop control also exhibits limitations, such as overshoot and settling time issues during transient conditions, potential stability issues or oscillations due to interactions with other power system control dynamics, and influence by network impedance, system inertia, and load dynamics [38]. Various research efforts are underway to address these challenges, focusing on different controller characteristics, grid-connected services, and operations in weak-grid and islanded modes [39, 40]. Future research needs to implement large-scale system studies, real-life implementation and testing, and enhanced controller approaches based on these findings [41]. Even though Droop control is a well-known methodology for power system control, its relevance and integration with grid-forming converters remain areas of interest for future research.

In general, grid-forming converters have been identified as crucial components in integrating large-scale PEI generation units concerning system operation and stability. Their vast potential in ancillary services and their mimicry of traditional synchronous generators label them as critical. However, while various grid-forming control approaches exist, grid-forming controllers still depend on extensive future research, pilot projects, integration, and policy determination. These controllers remain largely undeveloped and untested on large-scale system integration. However, multiple new real-life pilot projects are underway to analyze different aspects of grid-forming controllers regarding their control approach and different generation units. More information regarding these current projects can be found in [42].

Multiple gaps in the current literature indicate a significant necessity in multiple facets of grid-forming converter operation and analysis. The Global Power Systems Transformation Consortium identified four ancillary power system needs of importance concerning stability and power quality: 1) Synchronization and Angle Stability Needs, 2) Frequency Regulation Needs, 3) Voltage Regulation Needs, and 4) Damping Needs [43]. All of these topics are critical to ensuring the proper operation of future systems with high penetration of renewable resources.

Another resource suggests focusing on investigating the interplay between grid-following and grid-forming inverters in a mixed environment, including analyzing the impact of various ratios of grid-forming and following on system stability, frequency control, and overall system performance [38]. It highlights the importance of future research to prioritize transient stability studies to carefully evaluate the impact of grid-forming converters and their interactions with different power system resources, such as synchronous generators, loads, and transmission lines under dynamic perturbations [38]. Furthermore, it highlights the necessity for incorporating detailed representations of inverter control algorithms, system dynamics, and power system infrastructure into simulation models to ensure realistic results [38].

A comparative analysis of current and future trends also suggests various important focus points for future research. Four main branches were identified: 1) Frequency Control, 2) Voltage Regulation, 3) System Strength, and 4) Regulatory Framework [42]. For the frequency control branch, two fundamental research questions need to be answered: can grid-forming converters achieve frequency regulation in systems comprising grid-following and synchronous generators, and are there any limitations regarding the share of grid-forming in power systems [42]? For voltage regulation, the increased share of grid-forming and grid-following converters makes it crucial to understand how various distributed voltage/reactive power control affects the overall voltage regulation of the system. Identifying the optimal locations for different resources and the suitability of different voltage control systems is

important [42].

For system strength, future research questions lie in the contribution of different grid-forming resources to the system's overall strength on large-scale integration. The effects of individual components, such as virtual inertia and synthetic impedance, have not been fully explored for various systems, and it is critical to ensure system stability and strength.

Furthermore, the regulatory framework requires research to demonstrate the key functionalities of grid-forming converters, especially frequency and voltage regulation [42]. This involves research on different systems and practical implementations, helping to establish technical standards and regulatory frameworks to push the technological advancements of grid-forming converter technologies in a similar converging technological development to maximize their potential [42].

Another recent study highlights various challenges grid-forming converters face and their respective prognosis for future developments and integration into future systems [44]. It emphasizes that grid-forming converters are a relatively new technology, requiring extensive research and cooperation between different vendors to achieve stability in future systems. An overall 9-step guideline is proposed for future deployment, with certain time ranges to successfully integrate these resources into large-scale power systems [44].

Four main research areas are highlighted. First, the deployment of grid-forming converters from traditional microgrids to large-scale power systems. Grid-forming converters have shown promise in microgrid integration, both small and large, such as in various Hawaiian islands with different islanded microgrids. The report suggests that replacing synchronous machines with grid-forming converters has a timeline between 10-30 years, requiring robust standards and extensive research. Current technological findings regarding the operation of microgrids need to be extended to large-scale power systems with increasing penetration and capacity rates [44].

Another critical aspect is the development of technical standards for grid-forming converters. The unique behavior of the converters, such as their voltage source characteristics, tailored standards, and grid codes, are critical to successfully integrating these resources and having a convergent idea of the direction of research for the different grid-forming converter controllers [44].

Further, the report highlights the importance of accurate models and simulation tools for both grid-forming converters and high levels of PEI generation testing. This is vital to better understanding the intrinsic dynamics related to these resources and their impact on system stability during different operational and system characteristics. Priority should be given to research on accurate models and simulation tools for real-world scenarios [44].

While grid-forming converters show significant promise for power systems' future stability and reliability, ongoing research is essential to address their current limitations and optimize their performance. This technology is still very fresh, and extensive research and development are necessary to successfully integrate future systems with large-scale PEI-based generation. Multiple research questions need to be answered, ranging from the impact of different integration levels of these resources to the implementation of international standards. Collaboration among multiple vendors is crucial to making grid-forming converters a practical solution for future power system stability. The evolution of these technologies will play a crucial role in the transition towards more sustainable and resilient power systems.

1.2.2. Simulation of Future Power Systems

The emergence of power electronics has profoundly transformed the operation and analysis of power systems, influencing all facets of power system planning, modeling, and analysis, particularly in the realm of simulations. Historically, power systems were analyzed using simplified mathematical models, which evolved with the development of computers and simulation software, leading to more intricate modeling and simulation capabilities [45]. Power system simulation models are broadly categorized into static and dynamic models [46]. Among dynamic models, Root Mean Square (RMS) has been the

predominant method for analyzing classical power systems' performance in planning and operational phases [46]. RMS models were favored due to their ability to analyze synchronous generators' physics using simplified equations to model system dynamics [46]. However, with the fast-acting dynamics introduced by power electronics, traditional RMS models sometimes fall short of accurately capturing these new dynamics, necessitating updates to these models and stability assessments using Electromagnetic Transient (EMT) simulations as areas of active research.

While RMS simulations primarily model the fundamental frequency components of voltage and currents, focusing on the system's electromechanical transient behavior, EMT modeling utilizes instantaneous values to represent electromagnetic transients [47]. Historically, due to their high computational complexity, EMT simulations were limited to small system sections and short durations. However, the rapid dynamics induced by power electronics have elevated the role of EMT modeling in accurately representing power systems, particularly during this transitional period with many unknowns concerning stability phenomena [46]. The timeframe for EMT simulations, traditionally between 20 ms and 1 second, has recently expanded to durations ranging from 0.5 to 60 seconds, necessitating smaller time steps to capture all frequency phenomena and thus exponentially increasing computational complexity [48]. Despite this and current computational limitations, there remains a need for RMS modeling [46, 47, 49, 50].

Co-simulation emerges as a promising strategy to address the limitations of RMS modeling and the computational demands of EMT modeling. This technique integrates EMT simulations, specifically for power electronic components or sections, with conventional RMS modeling for the remaining parts of the system. This hybrid approach effectively balances computational load while preserving the critical frequency details in areas impacted by power electronics [49]. However, this method may not fully capture the harmonic components introduced by power electronics throughout the system, and its commercial usage is still limited due to availability and applicability challenges [49]. Despite these challenges, ongoing research aims to advance co-simulation techniques to adapt to the evolving system dynamics within current computational constraints [51, 52, 53]. Some software, like DigSilent PowerFactory, supports hybrid approaches for co-simulation, utilizing region definitions for different modeling complexities (EMT, balanced RMS, unbalanced RMS) and leveraging modern multicore processing for detailed modeling within a single program [48]. It is crucial to note that while EMT simulations offer a more detailed system analysis, they do not inherently guarantee accuracy. Validation against real-world measurements is essential for confirming their representational accuracy [48].

As the landscape of EMT-RMS simulations continues to evolve, the quest to find the optimal balance between computational complexity and accuracy is ongoing. Both EMT and RMS simulations are critical during this transitional phase, each playing a pivotal role in adapting to these emerging changes. This situation signals a substantial shift in methodologies for power system simulations, set to unfold in the upcoming years.

Challenges related to the influx of power electronics extend beyond simulation accuracy and speed for transient phenomena, particularly in addressing the variability introduced by PEI generation units, primarily in renewable energy sectors like solar and wind power. This generation style, dependent on natural elements such as sunlight and wind, inherently has variability in its energy production capabilities. As a result, energy generation can fluctuate based on daily, hourly, or even minute-to-minute changes in weather conditions, making forecasting a crucial component in the modern and future power system as these generation units will play a major role [54].

This variability poses a significant challenge in synchronizing generation with load. While traditional systems benefited from well-established load patterns and the ability to modulate generation capacities, the rise of variable energy sources coupled with evolving load patterns—driven by factors like the integration of electric vehicles and localized energy generation—demands a more sophisticated approach to forecasting. This has been identified as a challenge in the literature [50, 55, 56]. Such complexities are already evident in systems heavily reliant on hydropower, like Norway, where seasonal challenges occur, albeit on a smaller scale [57].

Historically, balancing generation with load was more straightforward, as load patterns were well-

documented and generation could be adjusted almost at will. Exceptions include countries like Norway, heavily reliant on hydropower, presenting seasonal challenges [57]. However, the shift towards more variable generation sources and changing load patterns due to the influx of electrical devices and lower hierarchy power generation alters the traditionally known load and demand system. This transformation places significant emphasis on accurate forecasting, both for weather-related generation capabilities and for anticipating changes in load demand [50, 55, 56].

Extensive research over the past two decades, especially with the rise of the wind power industry, has addressed these challenges. Current models rely on historical data for long-term forecasting and meteorological data for short-term projections [58, 59]. Operational methods, despite their efficacy, have identifiable limitations: intrinsic model constraints, reliance on the quality of input data, data confidentiality issues, and a lack of standardization [50, 59, 58]. These constraints limit the system's ability to provide services effectively and capitalize on renewable energy benefits.

To bridge this gap, emerging research is exploring the integration of machine learning, deep learning, and AI with traditional forecasting methods [58, 59, 60, 61, 62, 63, 64, 65, 66]. These approaches utilize advancements in computer science, pairing them with established and novel power system tools to enhance forecasting models. Embracing these technologies could be strategic, considering the rapid advancements in the computer science domain.

Forecasting has vast potential, with current developments in machine learning, AI, deep learning, and other advanced technologies potentially playing a crucial role in achieving efficient system stability. Cross-disciplinary collaboration and knowledge sharing are essential to maximize these technologies' potential.

This variability from renewable sources affects power system planning and operation in several ways. It adds complexity to system modeling, as generation units can no longer be relied upon to provide full capacity whenever needed. Interconnections between regions with diverse weather patterns and generation types are becoming increasingly vital, allowing for a more distributed generation pattern. This approach can reduce reliance on individual weather cycles, exploiting a broader range of conditions [67, 68]. Examples include Norway's interconnections with neighbors to mitigate seasonal variations and address supply-demand challenges for hydropower [69]. However, such interconnections also introduce challenges like fluctuating energy prices due to dependencies on external factors [70]. Effective utilization of interconnections requires improved communication and cooperation across borders and companies, ensuring power system stability and enhancing simulation effectiveness [71].

Due to these variability concerns, simulation software that allows for both short-term and long-term simulation planning is critical to better understand the intrinsic nature of these resources and to analyze them with respect to multiple different time steps and forecasts. While this is not a necessity, a major benefit is seen for software that can integrate multiple different aspects of power systems and the changing simulation nature of these future systems. This allows for a better overall analysis and makes the model better suited for future work. This applies to dynamic transient analysis, long-term forecasting, and load flow analysis.

Similar to utilizing simulation software for different time horizons based on forecasting and operational characteristics, it is also beneficial to use software that allows for both RMS and EMT simulation and/or is capable of co-simulation procedures. This severely enhances the operational capabilities of the simulations. By doing these steps, different power system representations in system modeling can be utilized for multiple analyses, overall increasing their potential impact for research and further study.

It is also beneficial to use simulation software that produces or has available multiple templates to allow for fast integration of different resources and to focus on simulation rather than the setup of different resources.

The literature review for the grid-forming converters also calls for the necessity of large-scale system representation in modeling software to allow for successful analysis and future integration of the differ-

ent PEI generation resources and their controllers. By creating new large-scale system representation simulation models, combined with generic templates utilizing software that tackles various analyses and co-simulation possibilities, the first gap in identifying challenges related to these resources can be addressed. Thus, while there are no specific current research gaps regarding the simulation of future power systems, better and easier integration of future resources is necessary. Combined with the possibilities of co-simulation, different time-scale simulations, and different models available, future analysis can be more effectively conducted. This is a crucial part of converging future directions with regard to future power systems and their intrinsic nature.

One of these resources, DigSilent PowerFactory, has been identified as a potential resource with all these different capabilities and thus serves as the benchmark for the rest of this study.

In summary, the rise of power electronics in power systems influences simulations in various ways, from altering the timescale of interest to transitioning towards co-simulation and EMT-RMS simulations. It also elevates the need for advanced forecasting to mitigate the impacts of variability. This variability introduces new challenges in system expansions, planning, and operation, all of which significantly affect the simulation landscape. However, these challenges can be addressed by utilizing simulation software with extensive short-term and long-term simulation potential, co-simulation capabilities, and various analyses available. The future of power systems relies heavily on research, cross-domain collaborations, and learning from ongoing changes, underscoring the critical role of adapting traditional simulations and models to ensure stability and security in the evolving power landscape.

1.3. Research Objective and Scope

The objective of this thesis project is to analyze the performance of the future Dutch power system using a synthetic stability model, focusing on the fast-active power support of grid-forming converters.
To achieve the overall research objective, the following research questions are sequentially defined:

1. Under what operating conditions and disturbances can the stability of fast-active power balancing be compromised in theoretical future power systems?
2. What are the implications of varying locations, structures, and parameterizations of grid-forming converters on the fast-active power balancing in power systems?
3. How do theoretical projections of future inertia constants and kinetic energy levels impact the fast-active power balancing stability of future systems?

To answer these questions, a three-stage approach is used to conduct the research:

1. **Scenario Analysis**: Future scenarios for the Dutch power system towards 2050 are identified and analyzed by comparing various reports and studies. A baseline scenario is selected for integration into the synthetic model for further analysis.
2. **Model Enhancement**: The theoretical Root Mean Square (RMS) synthetic model of the future 380 kV Dutch power system, created by [72], is updated to facilitate extensive future research. This includes incorporating IEEE and CIGRE standard parameters, implementing dynamic models representing future energy sources, and aligning the model with projections and investment plans towards 2050, such as upgrading transmission lines, creating new substations, and distributing new generation and flexibility resources.
3. **Comprehensive Study**: Using the enhanced model, a detailed study of the system and its components is conducted. This involves analyzing different levels of generation and load, the impacts of grid-forming converters, and varying future projections of kinetic energy and inertia. These elements are tested under various system perturbations to perform a dynamic fast-active power balancing stability assessment and sensitivity analysis based on frequency and eigenvalue analysis.

The research aims to bridge the gap between theoretical developments and future trends in power systems. By building and improving a large-scale synthetic model representation of the future Dutch power system, this study enables various research efforts concerning the impact of grid-forming converters. The analysis focuses on rotor angle and frequency stability. Analyzed through eigenvalue and frequency analysis, emphasizing fast-active power balancing operations and system damping. Ultimately, this project aims to enhance the understanding of future power electronics interfaced (PEI) systems and their impact on system characteristics, contributing to scientific advancements in international journals and conferences.

1.4. Methodology Overview

First, the thesis work examines predictions regarding the structure of the future Dutch power system, analyzing various scenarios based on national and international policy guidelines from key scenario reports. These scenarios reveal substantial deviations between different predictions based on policies and within similar predictions in different reports. Thus, future system development will heavily depend on ongoing research and policy guidance, making these areas critical for future research. The analysis focuses on key reports related to the future Dutch power system, highlighting structural differences and deviations between scenarios, leading to selecting one scenario for further implementation and investigation.

Next, a generic synthetic dynamic model in PowerFactory, developed in [72], was upgraded to represent the Dutch power system in 2050. This involved implementing the selected scenario from the analysis, integrating future investment plans, and enhancing the model with dynamic properties to reflect future system characteristics. The model also implemented representations of the interconnected Continental European power system, enabling large-scale system analysis for predictions of the future Dutch power system and its interconnected characteristics. The Dutch power system is represented using generic models and public information on future predictions. This allows the synthetic model to be used for future research requiring different operational characteristics regarding generation share, converter technologies, and interconnected inertia levels for large-scale systems to study the impact of future systems on fast-acting power balancing.

Finally, the enhanced dynamic model was used to analyze critical topics related to future systems. This included examining the impact of varying PEI generation shares on power production, evaluating grid-forming converter technologies, and assessing the effects of inertia and kinetic energy levels on dynamic stability. The analysis employed dynamic simulations evaluated by frequency and eigenvalue analysis to assess the large-scale synthetic model representation of the future Dutch power system, handled through PowerFactory combined with Python and Excel.

Following these steps, the goal is to bridge current gaps in understanding future power system stability, providing a comprehensive scenario analysis, a synthetic dynamic power system model, and insights into future challenges and solutions.

1.5. Thesis Structure

The structure of the thesis project can be seen in Figure 1.1, with chapters and appendix abbreviated as chp and app, respectively.

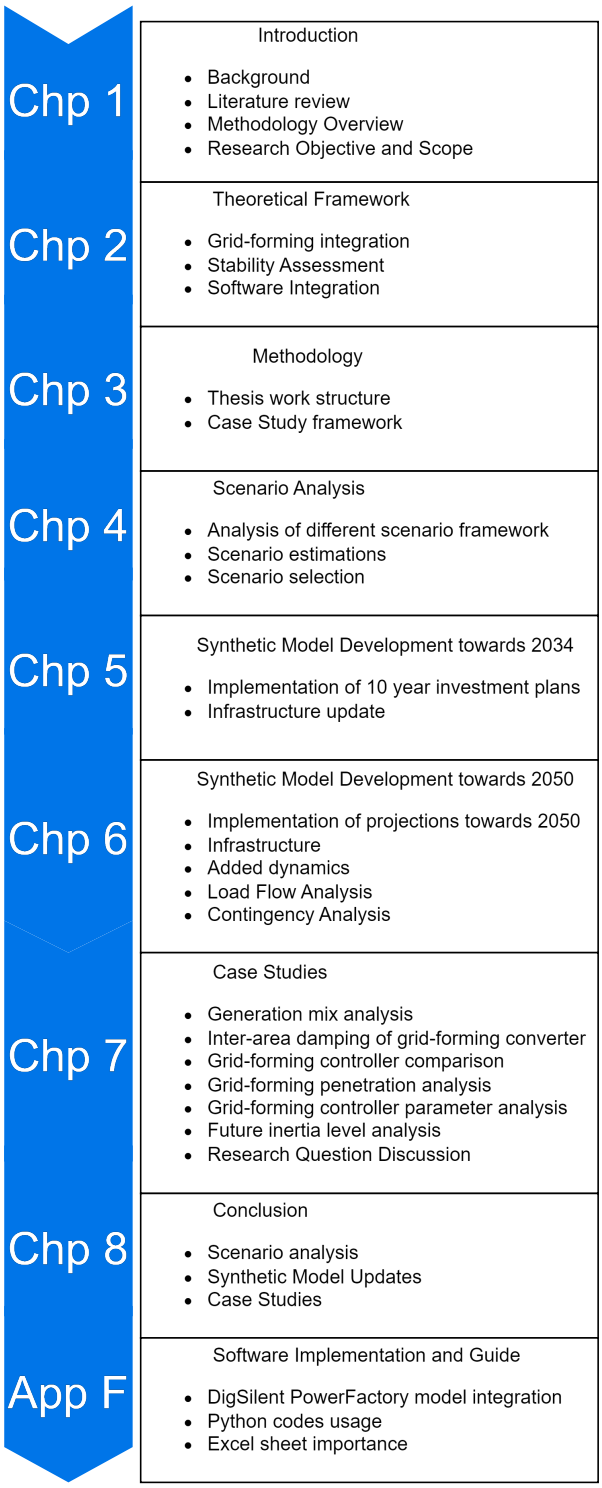


Figure 1.1: Processchart for structure and methodology of the thesis work.

2

Theoretical Framework

2.1. DigSilent PowerFactory

DigSilent PowerFactory was chosen as the preferred simulation software due to its high versatility and user-friendly interface. This section details the usage and role of the program concerning power system simulation, including key features, modeling techniques, dynamic simulation capabilities, data handling, visualization, and relevant studies and resources necessary for a comprehensive understanding of the program's application to the current study.

2.1.1. Dynamic Models

This section explores the dynamic models in PowerFactory, which are essential tools for simulating dynamic responses in power systems. These models facilitate the analysis of power system components under various disturbances. The following subsections will detail the hierarchy of dynamic model implementation in PowerFactory and their setup to provide a comprehensive understanding of their utilization.

An overview of the dynamic model setup in PowerFactory is presented in Figure 2.1. The primary dynamic setup comprises built-in dynamic models and dynamic controllers. The built-in dynamic models, native to PowerFactory, can be simulated across multiple domains, including load flow, RMS, and EMT simulations. These models can be directly selected from the PowerFactory drawing menu and include standard representations of synchronous generators, static generators, transformers, and transmission lines. While these models possess dynamic properties, they do not include direct control mechanisms.

Dynamic controllers, specified by the user, address this gap by providing control functionalities. These controllers can be either external or internally interfaced representations. The ongoing development and creation of new controllers reflect the continuous research in this field. The availability of both external and internal controller implementations offers significant flexibility, allowing for diverse studies and the adaptation of controllers to meet specific needs.

For this study, new internal dynamic controllers will be integrated into the system to enhance the representation of system dynamics. These controllers were developed using the built-in DigSilent Simulation Language (DSL) and applied to various resources such as synchronous and static generators, flexibility resources, and load representations. The focus will particularly be on the implementation of grid-forming controllers, as illustrated in section 2.2.

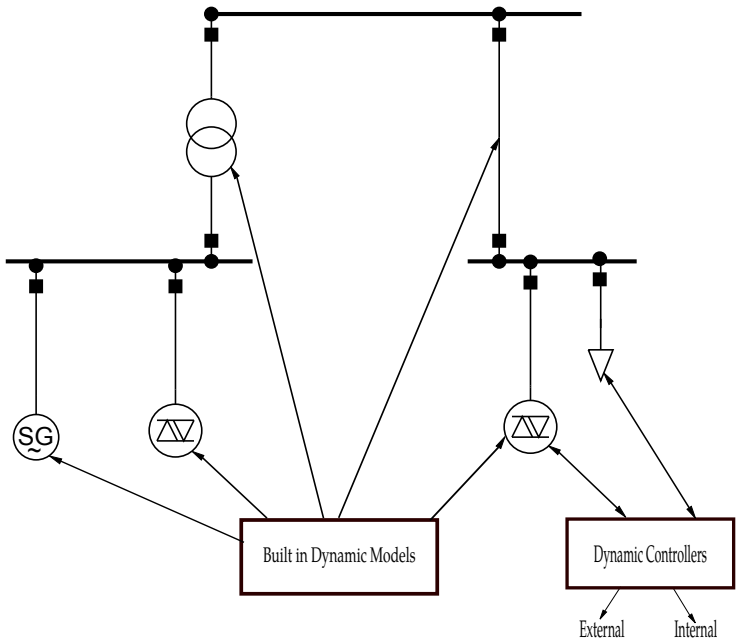


Figure 2.1: Overview of Dynamic Model setup in DigSilent PowerFactory.

DigSilent Simulation Language (DSL) is a powerful tool, and understanding it allows for further expansion of already made usable templates. This allows for fully controllable components in the system, controlled using a combination of transfer functions and equations to achieve the desired dynamic response to mimic real-life scenarios. The DSL language allows for flexibility in terms of modeling power system components; it allows for customization from already-made templates or the creation of independent ones. It also allows for interaction with other useful tools inside PowerFactory and has a user-friendly interface that makes it fairly easy to understand. When analyzing the system's dynamic response, the DSL equations that describe the dynamics of the controls are combined with the equations describing the dynamic behavior of the power system components, allowing for analysis based on an integrated transient analysis [73].

The DSL models are structured using a hierarchy system based on four DSL components: Composite Frame, Composite Model, Model Definition, and Common Model. How these four models are interlinked is illustrated in Figure 2.2, with an overview and remarks of the individual model definitions given in Table 2.1.
The core structural model is defined using composite frames and model definitions, serving as the blueprint for each network element and connected device. This hierarchy is analogous to object-oriented programming classes.
Composite models and common models are objects derived from the composite frame and model definition. The composite frame creates placeholders for the model definitions, describing how components are connected, while the model definition specifies the functionality of each component.

Consider two synchronous machines in a network, each with auxiliary control components such as Automatic Voltage Regulators (AVR), Power System Stabilizers (PSS), and governors. In both systems, these components are connected to the synchronous generator similarly. This connection methodology is defined in the composite frame, which is located in the user-defined models section of the project library.
The AVR, PSS, and governor systems have schematics defining their behavior. These behaviors are specified using model definitions, which are stored separately in the user-defined models section of the

project library, similar to composite frames.
Each machine can have different parameter values. For example, one generator might react slower but more significantly to a disturbance. To implement this, a common model is created from the model definition. The properties of the common model are defined by adding different parameters and coefficients to each auxiliary component. The common models are located in the network data folder under the network model.

As with the model definition and common model, a composite model is required for each machine. This allows each machine's separate common model instances to be connected and different parameters to be defined for each machine. Here, the network element and the synchronous machine are also implemented. The composite models are also located in the network data folder under the network model.
This hierarchical approach ensures a structured and detailed representation of the power system, facilitating accurate simulations and analyses.

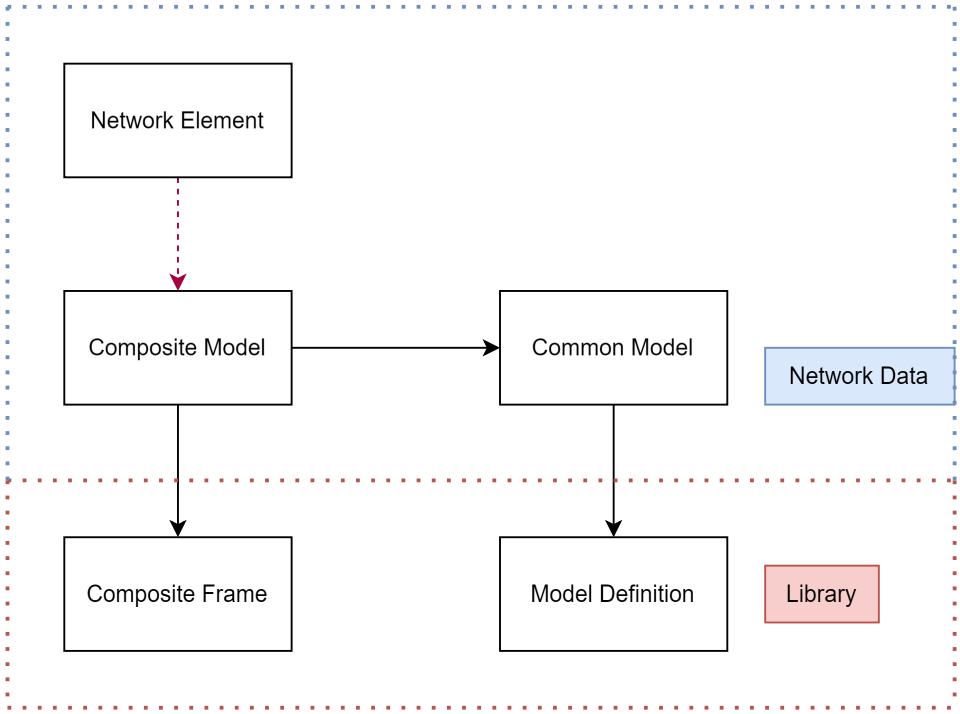


Figure 2.2: DSL model hierarchy.

Table 2.1: Overview and remarks on DSL components [73]

Component	Remarks
Composite Frame	– Composite Frame is a structure that contains slots for components (the generator, exciter, governor, PSS, etc.) – Only one per network element (e.g., synchronous generator) type – Located inside the User Defined Models folder in the Library.
Composite Model	– Object derived from the Composite Frame – Common Model objects are contained inside this Composite Model object. – One Composite Model per network element instance (one for each synchronous generator) – Located inside the Network Data folder (Grid).
Model Definition	– Contains the component's model by transfer function blocks (describing its functionality). – Only one per different component (AVR, PSS, and governor) – Located inside the User Defined Models folder in the Library.
Common Model	– Object derived from the Model Definition – Common Model object contains the parameter values of each network element instance (e.g., gain and time constant of transfer function). – One per component of each network element instance (AVR of gen 1, AVR of gen 2, PSS of gen 1, etc.) – Located inside Network Data Folder (inside respective Composite Model).

Dynamic models can be created or modified using DSL to achieve the desired dynamic response. For further details, refer to the following literature [73, 74].

2.1.2. Dynamic Simulation

Simulations must be conducted to effectively analyze system response to dynamic disturbances and respective power system characteristics. Various simulation options are available for power system analysis, but RMS and EMT simulations are preferred for power system dynamics. This section will focus on RMS simulations.

RMS simulations model the fundamental frequency components of voltage and currents, emphasizing electro-mechanical transient behavior [47]. The RMS simulation function in PowerFactory supports both balanced and unbalanced system conditions and is suitable for simulating both small and large power systems [73]. These simulations operate within a time frame of milliseconds to seconds, representing network elements using algebraic equations [73].

To simulate dynamic responses properly, a hierarchical process is followed. The first step involves executing a load flow calculation based on the system's steady-state operational characteristics. This step ensures that the system operates under normal conditions with the given characteristics and includes selecting the type of load flow calculation for initial conditions, the type of simulation (RMS/EMT), the step size, the solver selection, and other specific simulation characteristics.

The next step involves defining the variables. During dynamic simulations, multiple variables change over time, including voltages, currents, power values, and other signals of interest across the system, such as controller signals and other parameters. These variables are selected before proceeding to the next step.

Event definition follows, involving selecting and defining events relevant to the dynamic analysis. These events can include outages, short circuits, and load events. Each event's performance can be adjusted through various parameters, such as the time of occurrence and duration. Multiple events can occur simultaneously or at different stages in the simulation, allowing for comprehensive control over

the analysis.

After completing the event definition, the simulation setup is finalized, and the simulation can be executed. The simulation begins, and the disturbances occur at the defined times. The overall simulation extends according to the set simulation window. Post-simulation, the desired plots are selected. PowerFactory offers multiple plot types, including single or multiple curve plots and vector plots.

The final steps involve processing the plots and analyzing the results. PowerFactory's internal plotting function generates plots that can be directly processed as PNG/SVG images. Alternatively, result variables can be exported as CSV files for use in other programs, such as Python, for different plotting structures.

After the dynamic simulation, a modal/eigenvalue analysis in PowerFactory is also possible. This analysis enables the examination of system modes and their respective dampings, speeds, and other characteristics. This analysis facilitates the creation of plots highlighting the different modes of participation. This integrated approach between dynamic simulations and modal/eigenvalue analysis enhances the overall simulation process.

A flowchart of the aforementioned process is illustrated in Figure 2.3.

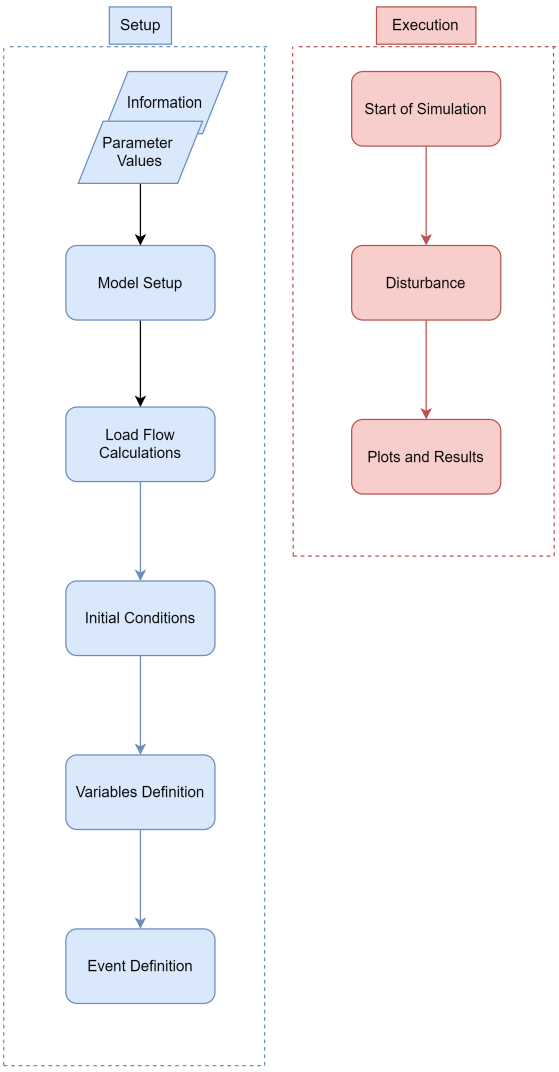


Figure 2.3: Flowchart of dynamic simulation process in PowerFactory [73].

PowerFactory offers extensive parameter customization based on simulation scenarios and desired inputs and outputs. Its user-friendly visual interface facilitates quick learning and a thorough understanding of its features.

2.1.3. Scripting

Another key feature of PowerFactory is scripting, which allows the automation of tasks to enhance the efficiency and accuracy of simulations. Scripting provides access to internal software functionalities, further improving controllability and accessibility. Although PowerFactory's interface is intuitive and easy to use, managing larger systems and running multiple scenarios can be time-consuming and prone to errors. Scripting mitigates this human element.

PowerFactory supports scripting in two ways: using the DIgSILENT Programming Language (DSL) or Python scripting. This section will focus on Python scripting, which will be utilized further. Python can be used in two ways: either through an internal Python object (ComPython) within PowerFactory, where the script can be written directly, or by using an external Python editor and linking it to the internal Python object. The latter option is preferable for larger Python scripts. This integration leverages Python's powerful features, enabling full external control of PowerFactory and enhancing its efficiency.

Utilizing Python scripts in combination with PowerFactory significantly increases the efficiency of various studies and automates numerous tasks, including simulation tasks, resource integration, parameter adjustments, and result processing. A simple flowchart illustrating the PowerFactory and Python integration is presented in Figure 2.4.

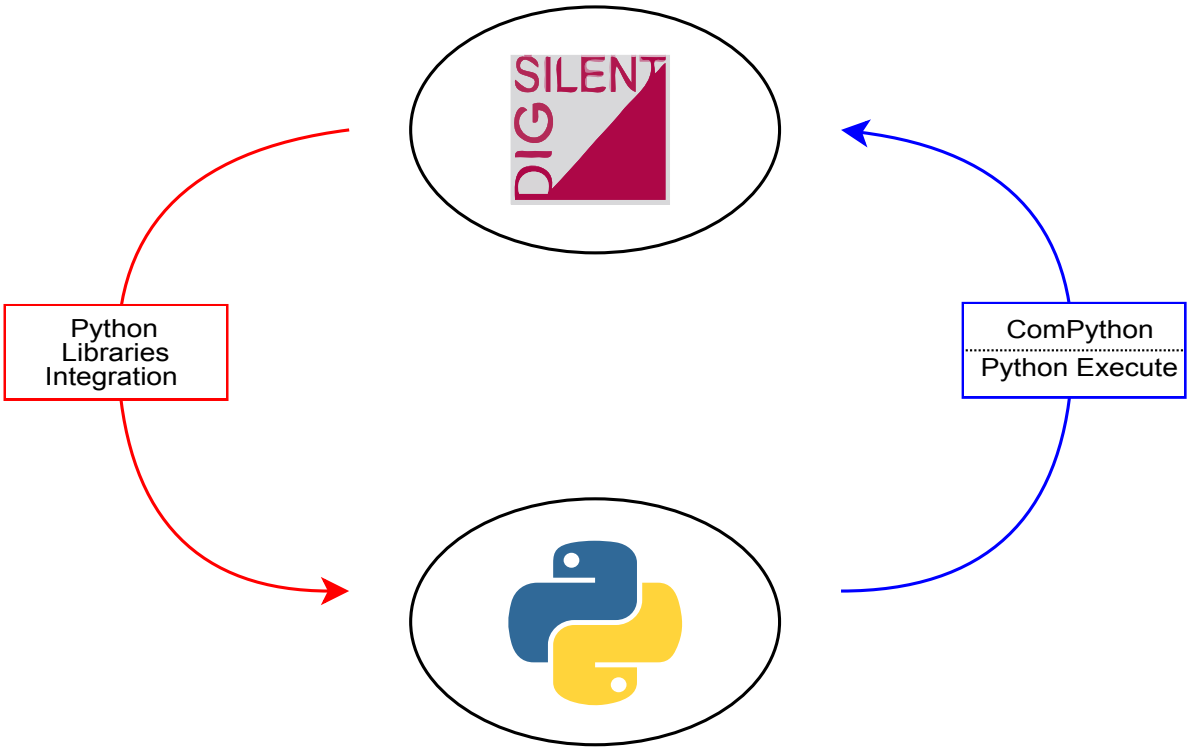


Figure 2.4: Flowchart of PowerFactory and Python integration.

2.2. Grid-forming Converters Implementation in PowerFactory

This section discusses the implementation of three different grid-forming converters—the Virtual Synchronous Machine (VSM), Synchroverter, and Droop Controller. The main equations supporting each controller are examined, along with an overview of the control blocks and key parameters. All models are available through DigSilent PowerFactory's library, as detailed in [75].

2.2.1. Virtual Synchronous Machine (VSM)

First, the implementation of the VSM controller is discussed. The converter frame represented as a composite model in DigSilent PowerFactory can be seen in Figure A.1. The model consists of current and voltage measurements, virtual impedance, voltage controller, output voltage calculations, and a slot for the converter operation. Here, the active and reactive power are calculated using the following equations:

$$p_{\text{mea}} = ur \cdot ir + ui \cdot ii \tag{2.1}$$

$$q_{\text{mea}} = ui \cdot ir - ur \cdot ii \tag{2.2}$$

Here, the active and reactive power is found using components of the measured real and imaginary voltage and currents (ur, ui) and (ir, ui), respectively. Active power is further used for this model.
The overview of the grid-forming control block as a DSL model is given in Figure A.2. The DSL model of the grid-forming control block is the block that primarily makes the converter operate as a VSM. The highlighted red area is the part that mimics the swing equation and makes the VSM act as a synchronous generator. This block is shown in Figure 2.5. Here, $P_{\text{measurement}}$ is the same as p_{mea} in Equation 2.1. This block emulates the swing equation by forming the following equation:

$$T_a s\omega = (P_{\text{setpoint}} - P_{\text{measurement}}) - D_p(\omega_{\text{lpf}} - \omega_{\text{ref}}) \tag{2.3}$$

Two important parameters and the main drivers behind the inertia emulation are the acceleration time constant T_a and the damping coefficient D_p. Here, the acceleration time constant acts similarly to the inertia time constant, and the damping coefficient mimics the inherent damping of synchronous machines, effectively replicating the swing equation. These two parameters are critical. Note, the T seen in Figure A.2 is not the torque as seen in the Synchroverter but rather the initialization of the acceleration time constant T_a. The rotating speed of the VSM is given by ω_{lpf} with the frequency set point transformed to speed as ω_{ref}. Another important note is that the damping term is passed through a low-pass filter, seen as the ω_c in the block diagram. This allows for damping during transient conditions while eliminating it for steady-state operation. If steady-state damping is wanted, ω_c must be changed to zero.

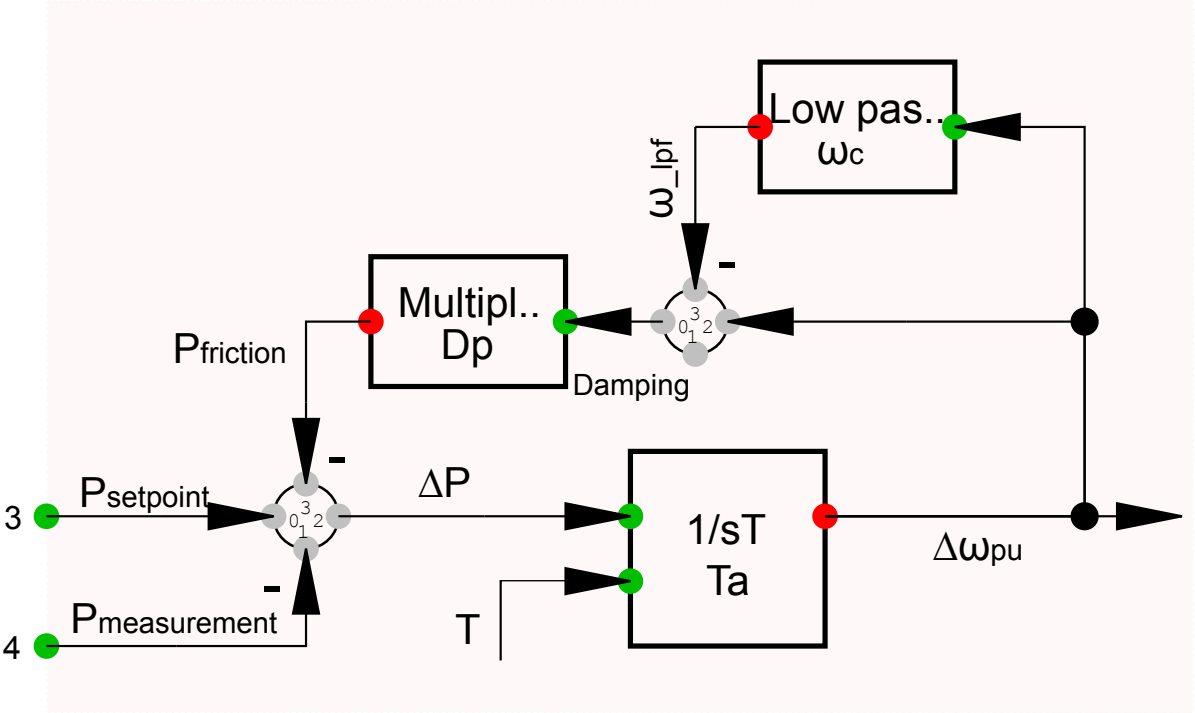


Figure 2.5: Frequency control loop of the VSM controller, representing the swing equation [76].

The model also contains a voltage controller, acting as a proportionate voltage controller, and the DSL frame can be seen in Figure A.3. Proportionate controllers give an output that is proportionate to the error; they respond to the error [77]. Here, K is the controller gain, and T_r is the measurement filter time constant. The voltage error du is found using the reference voltages u_{ref} and du_{ref} and the terminal voltage u. This error is multiplied by the controller gain. After this, the limits of the signals are determined before the output voltage is given and used in subsequent controls.

Notably, the VSM uses power to mimic the swing equation. Therefore, an extra power calculation block is also implemented based on the measured current and voltage profiles. The other blocks in the control system are general control blocks and will not be covered here; however, the virtual impedance and its characteristics are explained in subsection 2.2.4.

2.2.2. Synchroverter

The model of the Synchroverter represented as a composite model can be seen in Figure A.4. The structure here is similar to that of the VSM, with the measurement slots, virtual impedance, output voltage calculations, and the slot for the converter being similar. Additionally, the Synchroverter contains a current conversion slot that converts the measured current into magnitude and angle for further usage. The model does not contain a voltage control slot, which is integrated into the Synchroverter block.

As a DSL model, the internal Synchroverter grid-forming control block is given in Figure A.5. Here, the red part highlights the frequency loop, and the blue part highlights the voltage loop. This is more detailed in Figure 2.6. The frequency control loop is derived from Equation 2.4 and utilized in the control system as Equation 2.5. Here, ΔT represents a change in torque, while $\Delta\omega$ represents a change in mechanical speed. Thus, this equation represents a change in torque warranting a change in mechanical speed and vice versa.

$$D_p = -\frac{\Delta T}{\Delta \omega} \tag{2.4}$$

$$T_a s\omega = (T_{\text{setpoint}} - T_{\text{electrical}}) - D_p(\omega_{\text{lpf}} - \omega_{\text{ref}}) \tag{2.5}$$

Notably, the equation has parameters similar to those of the VSM. The difference comes from the Synchroverter utilizing torques ($T_{setpoint}$, $T_{electrical}$) instead of powers for the VSM. This is done internally in the Synchroverter control frame, using three equations in the block labeled "Zhong synchroverter eqn 7-8-9". These three equations are the following:

$$t_e = Mf\_if \cdot i_{\text{abs}} \cdot \sin(\theta - \theta_{\text{i}}) \tag{2.6}$$

$$u_{\text{magnitude}} = \omega \cdot Mf\_if \tag{2.7}$$

$$q_e = -u_{\text{magnitude}} \cdot i_{\text{abs}} \cdot \cos(\theta - \theta_{\text{i}}) \tag{2.8}$$

Here, the three different equations calculate the torque t_e, the voltage magnitude $u_{magnitude}$, and the reactive power q_e. Further, Mf_if represents the simplified excitation current, θ represents the rotor angle, ω the speed, and i_{abs} the absolute value of the measured current found through the current conversion block.

θ_{i} is found using the following equation:

$$\theta_{\text{i}} = \arctan\left(\frac{\sin(\theta_{\text{i}})}{\cos(\theta_{\text{i}})}\right) \tag{2.9}$$

These three equations allow for the three outputs t_e, marked as $T_{\text{electrical}}$ in the frequency loop, $u_{\text{magnitude}}$ the output voltage magnitude in the output voltage calculations, and q_e marked as $Q_{\text{electrical}}$ in the voltage control loop.

The voltage control loop, highlighted in blue in Figure 2.6, has a similar structure to the frequency loop. This structure is based on Equation 2.10 and is implemented as seen in Equation 2.11. Here, ΔQ represents a change in reactive power, while ΔV represents a change in voltage, describing the relationship between changes in reactive power and corresponding changes in voltage.

$$D_q = -\frac{\Delta Q}{\Delta V} \tag{2.10}$$

$$K_q = D_q(u_{\text{setpoint}} - u_{\text{measurement}}) + (Q_{\text{setpoint}} - Q_{\text{electrical}}) \tag{2.11}$$

Here, K_q represents the voltage control gain, dependent on the damping coefficient D_q, the set and measured voltages (u_{setpoint}, $u_{\text{measurement}}$) and the set and measured reactive powers (Q_{setpoint} and $Q_{\text{electrical}}$).

Thus, four parameters are of main interest when determining the control function of the Synchroverter: the acceleration time constant T_a, the damping coefficient D_p, the voltage control gain K_q, and the reactive power droop coefficient D_q. The acceleration time constant acts similarly to the same parameter for the VSM, functioning as the inertia constant in traditional synchronous machines. The voltage control gain acts similarly to the proportional voltage controller gain in the VSM, scaling the controller's response to voltage and reactive power loop deviations. The damping coefficient functions like the damping coefficient in the VSM, and the reactive power droop coefficient determines the amount of reactive power in response to a voltage deviation.

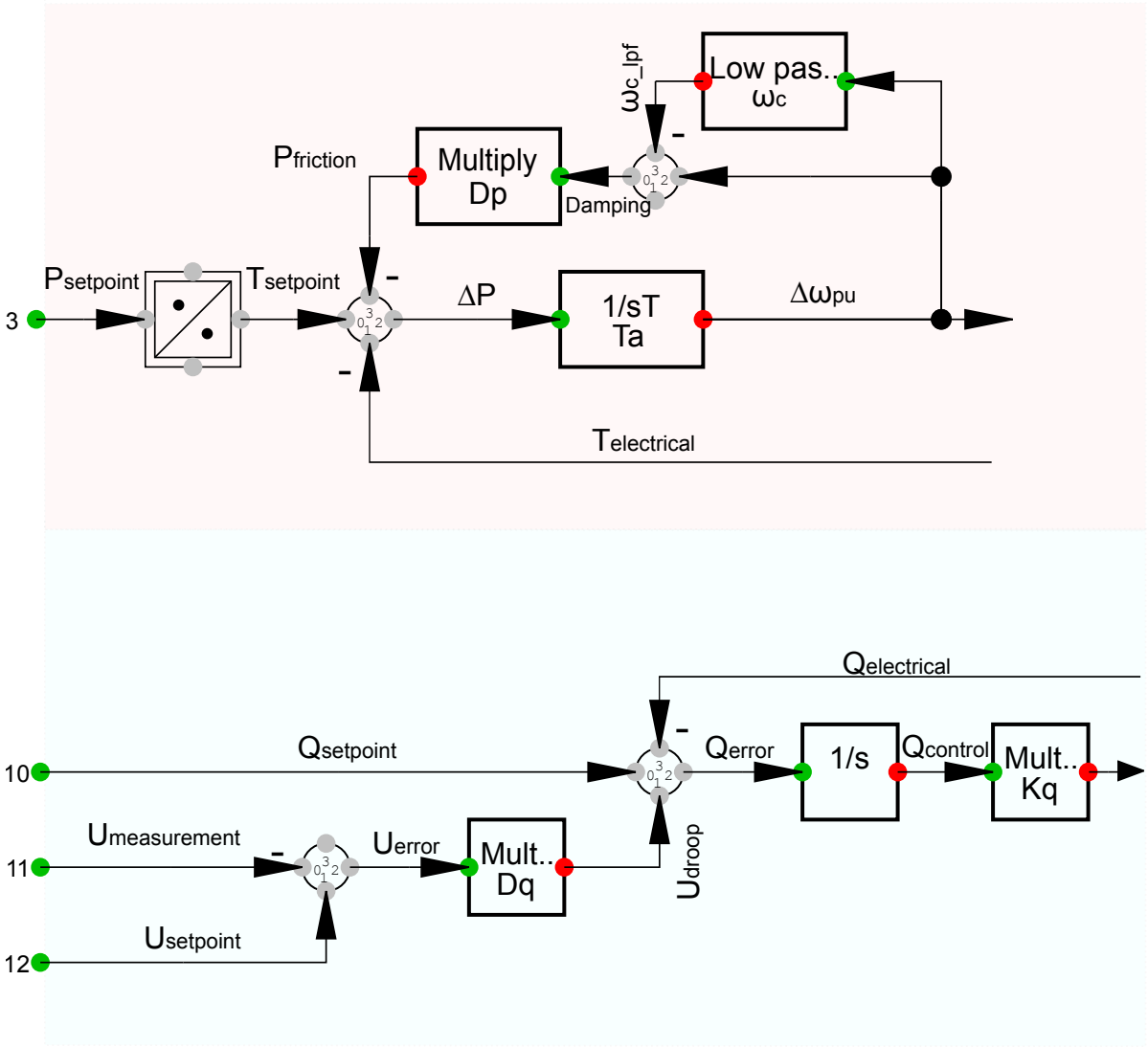


Figure 2.6: Frequency (top) and voltage (bottom) control loops for the Synchroverter controller [76].

The Synchroverter also contains a virtual impedance block, explained in subsection 2.2.4.

2.2.3. Droop Control

The model of the Droop controller represented as a composite model can be seen in Figure A.6. The structure is identical to the VSM; however, no voltage controller is specified in the voltage control block, as this is handled through the internal droop control similar to the Synchroverter.

The internal droop-control block as a dsl frame is illustrated in Figure A.7, highlighting the active power loop in red and the voltage loop in blue. This is further detailed in Figure 2.7.

The equation utilized for the active power-frequency control is given as:

$$\Delta\omega = m_p \Delta P \tag{2.12}$$

Representing the change in speed $\Delta\omega$, with the active power droop coefficient mp and the active power change ΔP. This translates into the control system as:

$$\omega_{gen} = \omega_{ref} - mp(P_{\text{setpoint}} - P_{\text{measurement}}) \tag{2.13}$$

Here, the speed is split into the Droop controller and reference speed (ω_{gen}, ω_{ref}), and the active power is split into the setpoint and measured (P_{setpoint}, $P_{\text{measurement}}$).

For the reactive power-voltage control, the equation is:

$$\Delta V = mq\Delta Q \tag{2.14}$$

Representing the voltage change ΔV, the reactive power droop coefficient mq, and the change in reactive power ΔQ.
Translating to:

$$u = u_{\text{ref}} - m_q(Q_{\text{setpoint}} - Q_{\text{measured}}) \tag{2.15}$$

Here, similar to the active power-frequency control, the voltage is split into measured and referenced (u, u_{ref}), and the reactive power is split into setpoint and measured (Q_{setpoint}, $Q_{\text{measurement}}$).

This relationship leverages the close correlation between active power and frequency and between reactive power and voltage. The Droop controller also utilizes a low-pass filter to operate only in the transient state rather than the steady state.

The active power droop coefficient mp and the reactive power droop coefficient mq are the key variables in the controller control. These coefficients control the amount of active and reactive power output as a function of the change in frequency and voltage, respectively. Notably, ω_{ref} is equivalent to F_{setpoint}, as all units are in per unit.

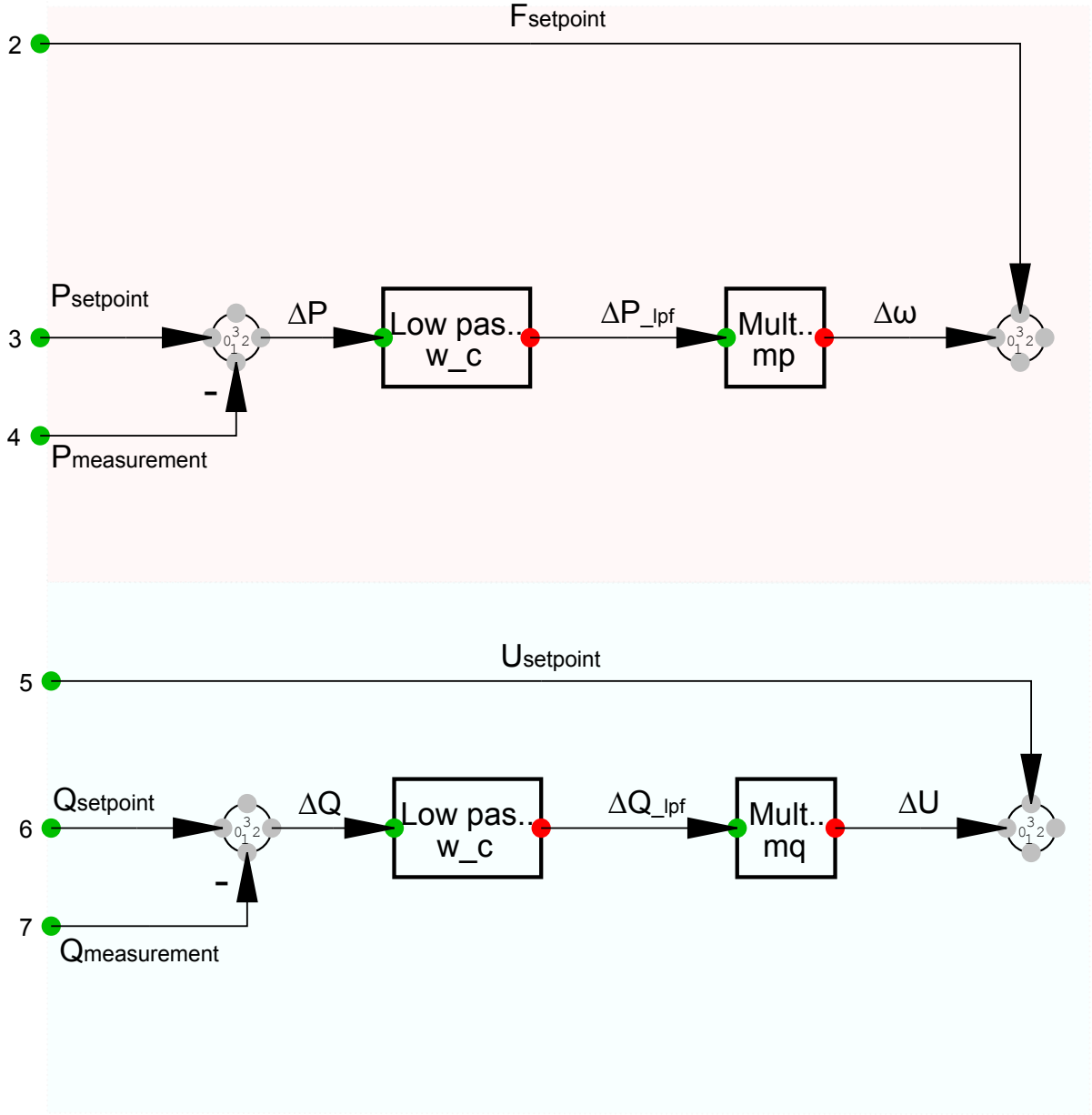


Figure 2.7: Frequency (top) and voltage (bottom) control loops for the Droop controller [76].

The Droop controller also implements a virtual impedance, which will be discussed in subsection 2.2.4.

2.2.4. Virtual Impedance

The virtual impedance block is combined with the grid-forming converters to improve damping and allow current limitation in the event of a short circuit [75]. This utilizes the three outputs of the current measurement: i_r (the real part), i_i (the imaginary part), and i (the magnitude of the current). Within the block, a voltage drop is calculated that is given as the output and added to the output voltage control calculation outlined in subsection 2.2.5.

The voltage drop over the virtual impedance is found using the following equations:

$$\Delta u_{vi,r} = r_{vi} i_{vi,r} - x_{vi} i_{vi,i} \tag{2.16}$$

$$\Delta u_{vi,i} = r_{vi} i_{vi,i} + x_{vi} i_{vi,r} \tag{2.17}$$

Here, r_{vi} and x_{vi} represent the resistance and reactance of the virtual impedance, respectively, and i_{vi} represents the current supplied by the converter, found using the aforementioned current measurements. The virtual impedance can be controlled in two ways: either as a constant or a proportional over-current factor. In the proportional over-current mode, the virtual impedance changes during transient stages to help with current protection during these stages. This is done by adjusting the virtual impedance according to a proportion of the low-pass filtered current absolute value i_{vi} when a certain i_{lim} threshold is exceeded [75]. Meanwhile, the constant control mode has a constant virtual impedance during transient events and thus can result in currents that can damage the controller. The low-pass filter allows the control system to act only during transient events, not in a steady state. The low-pass filter current uses the original current magnitude i. The controller mode operation for the proportional over-current mode can be seen in the following equation:

$$z_{vi} = \begin{cases} r_{vi} + jx_{vi}, & \text{if } |i_{vi}| \leq i_{\text{lim}} \\ (k_{pr}r_{vi} + jk_{px}x_{vi})(|i_{vi}| - i_{\text{lim}}) + r_{vi} + jx_{vi}, & \text{otherwise} \end{cases} \tag{2.18}$$

Here, the factors k_{pr} and k_{px} are defined as proportional factors of the virtual resistance and reactance, respectively. This allows the voltage drop to be calculated according to Equation 2.16 and Equation 2.17 to be further used in the output voltage control. The parameters of interest in the virtual impedance are:

r: Virtual resistance
x: Virtual reactance
i_lim: Over-current threshold
kpr: Proportional factor for additional resistance
kpx: Proportional factor for additional reactance
T_lpf: Time constant of low-pass filter

2.2.5. Output Voltage Control

The Output Voltage Control block is the last block of interest in the control schemes. Here, the grid-forming controller's voltage and the virtual impedance's voltage drop are utilized to calculate the final output voltage of the converter [75]. A current limitation, similar to the virtual impedance, is added to this block and can be enabled or disabled as needed.

First, the real and imaginary parts from the magnitude and angle of the calculated voltage of the individual block are calculated, and the drop over the virtual impedance is subtracted using the following equations:

$$u_{r\_unltd} = u_{\text{mag}} \cdot \cos(u_\theta) - u_{vi\_r} \tag{2.19}$$

$$u_{i\_unltd} = u_{\text{mag}} \cdot \sin(u_\theta) - u_{vi\_i} \tag{2.20}$$

These are marked as unlimited as there has been no voltage restriction yet.

Next, the voltage drop over the series reactance over the series converter impedance is calculated using the following equations:

$$u_{r\_drop\_unltd} = u_{r\_unltd} - u_{r\_terminal} \tag{2.21}$$

$$u_{i\_drop\_unltd} = u_{i\_unltd} - u_{i\_terminal} \tag{2.22}$$

After this, the voltage drop is found when the current limitation is enabled. The maximum absolute value of the voltage drop is found utilizing the converter series impedance z_{series} and the maximum converter current $i_{\text{con,lim}}$ using the following equation:

$$u_{drop,ltd} = |z_{\text{series}}| i_{\text{con,lim}} \tag{2.23}$$

Then, the real and imaginary voltage drops can be found using the following equations:

$$u_{r\_drop\_ltd} = u_{drop,ltd} \cdot \cos(\theta_{u_{drop}}) \tag{2.24}$$

$$u_{i\_drop\_ltd} = u_{drop,ltd} \cdot \sin(\theta_{u_{drop}}) \tag{2.25}$$

After this, the limited voltages can be found using the following equations:

$$u_{r\_ltd} = u_{r\_terminal} + u_{r\_drop\_ltd} \tag{2.26}$$

$$u_{i\_ltd} = u_{i\_terminal} + u_{i\_drop\_ltd} \tag{2.27}$$

Then, the output converter voltages can be computed using the following equations:

$$u_r = \begin{cases} u_{r\_ltd}, & \text{if Mode} = 1 \\ u_{r\_unltd}, & \text{if Mode} = 0 \end{cases} \tag{2.28}$$

$$u_i = \begin{cases} u_{i\_ltd}, & \text{if Mode} = 1 \\ u_{i\_unltd}, & \text{if Mode} = 0 \end{cases} \tag{2.29}$$

Here, mode 1 indicates an enabled current limitation, while mode 0 indicates a disabled limitation.

Thus, the converter's final output voltages can be found using internal computation in the output voltage controller block, and an additional measure of current limitation is utilized as well.

2.3. Power System Stability

Power system stability can be generally described as the ability of a power system to maintain operational equilibrium under normal conditions and to return to an acceptable state of equilibrium following a disturbance [78]. The definition of power system stability has evolved from a three-branch format to a five-branch format, shown in Figure 2.8, reflecting the increasing penetration of power electronics and changing power systems. Consequently, stability in power systems must be reconsidered for current and future systems to account for these new factors.

Traditionally, the dynamic performance of synchronous generators, their controls, and the dynamic performance of the loads primarily determined the dynamic performance of power systems [79]. Stability issues were typically related to slow electromechanical phenomena, while fast transients related to the network and other fast-response devices were often disregarded because they decayed rapidly [78, 80]. However, the increasing penetration of power electronics in power systems has altered these dynamic properties. With significant penetration of PEI generation, loads, and transmission devices, the dynamic responses of power systems have become more dependent on complex fast-response power electronic devices, thereby altering the dynamic behavior [79].

Power electronics exhibit a very quick dynamic range from a few microseconds to multiple milliseconds, accounting for both wave and electromagnetic phenomena. As power electronics increasingly penetrate various areas of power systems, the overall dynamic characteristic will shift from slower to faster dynamics, necessitating a change in how stability is addressed.

This section will discuss the two main stability classifications relevant to this thesis: small-signal rotor angle stability and short-term frequency stability, highlighted in green in Figure 2.8. The different concepts will be briefly explained, followed by a discussion of how the paradigm shift impacts these stability criteria and how they are analyzed using different metrics and operations.

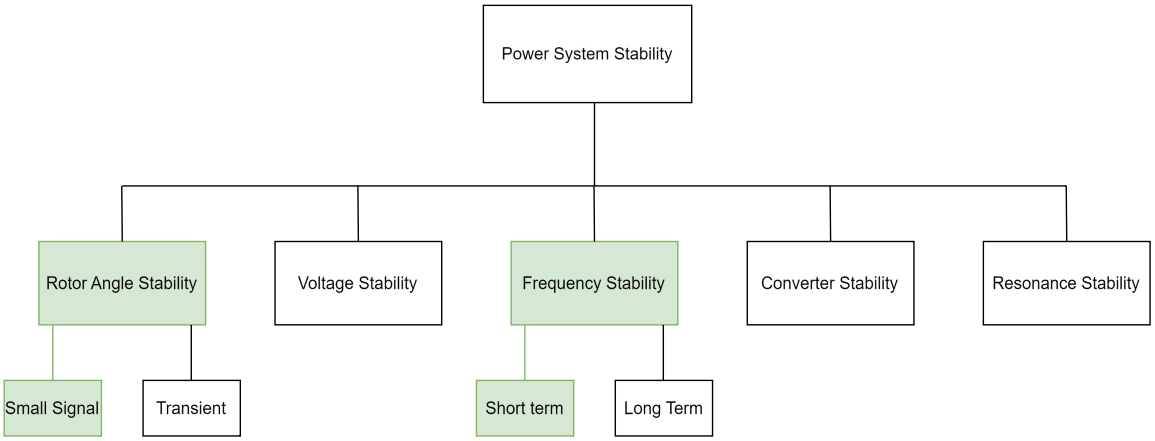


Figure 2.8: Stability Classification Overview.

2.3.1. Rotor Angle Stability

Rotor angle stability is defined as the ability of interconnected synchronous machines in a power system to maintain synchronism [78]. This stability is crucial both during normal operation and after a disturbance. In an interconnected system, all connected machines' stator voltages and currents must operate at the same frequency, and the rotor mechanical speed must be synchronized to this frequency [78]. This synchronization implies that all rotors of interconnected machines must remain in synchronism. Achieving this requires the electromagnetic torque to be equal and opposite to the mechanical torque delivered by the prime mover [79]. Therefore, the overall stability depends on the ability of different machines to maintain or restore this torque relationship [79].

Rotor angle stability is divided into small-signal (small-disturbance) stability and transient stability [78]. This discussion focuses on small-signal stability, which refers to the power system's ability to maintain synchronism under small disturbances [78]. These disturbances occur continuously in the system and include events such as small load and/or generation variations. The disturbance is considered small if it is possible to linearize the system equation for analysis [78]. Two parameters can lead to small-disturbance instability: a steady increase in rotor angle due to insufficient synchronizing torque or rotor oscillations of increasing amplitude due to a lack of sufficient damping torque [78]. The system's ability to respond to disturbances depends on multiple factors of the pre-fault system, including the generators' loading, the transmission system's strength, and the generators' excitation system [78]. The damping torques of the system will be further analyzed and studied here.

The primary cause of instability in traditional systems is identified as a lack of damping torque [80]. Small-signal stability is often analyzed based on the system response to a disturbance, with a load increase typically used to represent the disturbance for analysis. More details regarding the different disturbances can be found in section 2.4. Disturbances in the power system result in various oscillatory behaviors between different resources, significantly affecting system stability. These oscillatory behaviors are often classified as system modes, which can have a large impact dependent on the speed and damping.

These modes can broadly be defined as local-area modes and inter-area modes [81]. Local-area modes are associated with one synchronous generator or a small group of generators swinging against the rest of the power system, while inter-area modes involve a part of the power system swinging against other parts [78, 81]. A general classification of the different frequency ranges for these modes is shown in Table 2.2.

The stability of local-area modes depends on the strength of the transmission system as perceived by the power plant, the excitation control systems of the generator, and the plant output [78, 80]. In contrast, the characteristics of inter-area modes are significantly different and highly complex, with load characteristics having a major effect on the stability of these modes, highlighting the intrinsic nature of the different modes [78, 80]. Small-signal studies typically consider a time frame of interest of 10-20

seconds following a disturbance [80].

Table 2.2: Overview of different oscillation modes and their frequency ranges [82]

Mode	Frequency Range	Description
Inter-area (low)	0.1 – 0.3 Hz	Very low-frequency oscillations involving all generators in the system.
Inter-area (high)	0.4 – 0.7 Hz	Oscillations where sub-groups of generators swing against each other.
Intra-area	0.4 – 1 Hz	Oscillations between groups of generators within a single area of the power system.
Local	0.7 – 2 Hz	Oscillations of a single generator or power plant against the rest of the power system.
Intra-plant	1.5 – 3 Hz	Oscillations between generation units within a multi-unit power station.
High-frequency oscillations	2.5 – 97.5 Hz	Oscillations induced by power electronics such as VSC-HVDC, particularly in large-scale wind power plants.

The analysis of system modes is often represented using a complex conjugate pair of eigenvalues [79]. The eigenvalues can be classified as shown in Figure 2.9. The x-axis represents the real part of the eigenvalue, correlating to the damping, while the y-axis represents the imaginary part correlating to the speed. Generally, if the eigenvalues are in the left-hand plane, the system is considered stable; if they are in the right-hand plane, the system is considered unstable.

The eigenvalue findings are often analyzed using a damping ratio (ζ), which indicates the damping levels (σ) with respect to the oscillatory speed (ω), representing how well the different modes are damped. This is calculated using Equation 2.30.

$$\zeta = \frac{-\sigma}{\sqrt{\sigma^2 + \omega^2}} \tag{2.30}$$

Thus, if the eigenvalues are to the right of this line, they are classified as stable but not sufficiently damped for the system and thus are identified as poorly damped.

Traditionally, specific criteria have not been applied to damping ratios. However, a damping ratio of 3 % is mentioned as a stability limit indicating insufficient damping, while at least 5 % damping is considered acceptable [83, 84]. Studies, such as those for the Nordic transmission system, have confirmed this, identifying a security criterion of a 3 % damping ratio, while 3-5 % was found to be adequate damping [84, 85].

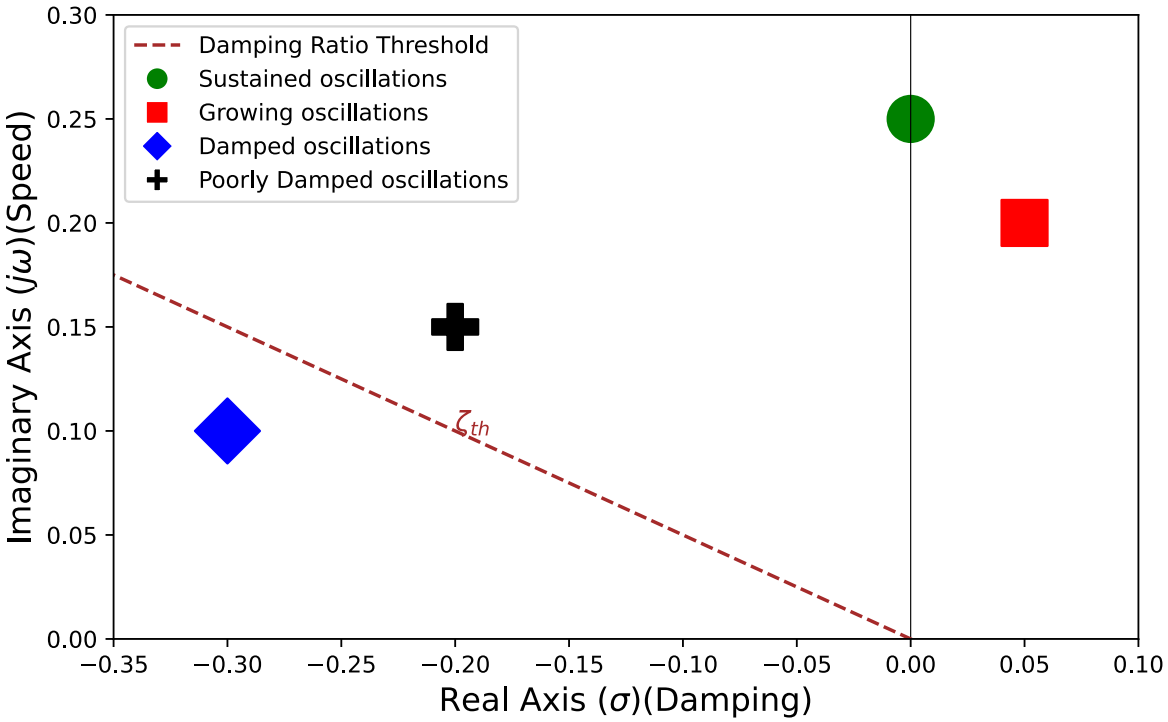


Figure 2.9: Eigenvalue analysis classification.

Due to PowerFactory's extensive library, these different modes and damping ratios can be found directly using the integrated modal analysis. These values can be extracted as CSV files for improved plotting and integration with other resources, such as Python.

While rotor angle stability primarily focuses on synchronous machines and their characteristics, the penetration of power electronics and the displacement of synchronous generators can influence this stability. The updated stability criteria found in [79] highlight four points for small-signal stability:

1. Changed power flows in major tie-lines: The change in the overall generation and load structure of modern systems can alter the power flow in traditional major tie-lines, impacting the damping of inter-area modes and transient stability margins [86, 87].
2. Displacement of large synchronous generators: This can affect the mode shape, modal frequency, and damping of different modes associated with rotor oscillations [86].
3. Damping torque effect on nearby synchronous generators: This effect is similar to how FACTS devices influence damping [88, 89].
4. Displacement of synchronous machines with crucial power system stabilizers.

However, there is potential for power electronic devices to help alleviate power oscillations in the future, similar to the concept of power oscillation dampers on FACTS devices [79, 88, 89].

Overall, while traditional factors influencing rotor angle stability related to synchronous machines are well-researched and understood, the changing power system characteristics due to the penetration of power electronics can significantly influence and complicate this stability. Although significant research has been conducted on these issues, more work is needed to fully understand and prepare for the upcoming challenges that power system planners and operators face.

2.3.2. Frequency Stability

The next aspect of power system stability is frequency stability. Frequency stability refers to the power system's ability to maintain a stable frequency after a significant disturbance that causes a substantial imbalance between generation and load [80]. This depends upon the system's ability to maintain or

restore equilibrium between load and generation in the system, resulting in minimum loss of load [80]. Sustained frequency swings eventually leading to tripping of generation and/or loads is one of the possible results of instability [80].

Frequency stability is usually defined in terms of maximum acceptable deviation, often referred to as the frequency nadir indicating the lowest or highest frequency value, and the speed at which the frequency changes, also called the rate of change of frequency (ROCOF) [90]. An example of the frequency nadir/maximum allowed deviation is seen in Table 2.3, by the European Network of Transmission System Operators (TSO) for Electricity (ENTSOE) [91]. Here, it can be seen that a given frequency range is assigned a certain time period for operation. This means that for frequencies within that range, the components should be able to withstand the frequency for the specified period. So, for a range of 1 Hz deviation from 50 Hz, it should be connected indefinitely, but for other ranges, the time periods will vary. These are the set frequency deviations and their corresponding operating times. Some are set on a continental basis, such as continental Europe, while some of the frequency ranges will depend upon individual TSOs.

Table 2.3: Frequency ranges and time periods for operation in Continental Europe [91]

Synchronous Area	Frequency Range	Time period for operation
Continental Europe	47.5 Hz – 48.5 Hz	To be defined by each TSO while respecting the provisions of Article 4(3), but not less than 30 minutes
	48.5 Hz – 49.0 Hz	To be defined by each TSO while respecting the provisions of Article 4(3), but not less than the period for 47.5 Hz – 48.5 Hz
	49.0 Hz – 51.0 Hz	Unlimited
	51.0 Hz – 51.5 Hz	30 minutes

The ROCOF is calculated by taking the derivative of the frequency (frequency change (Δf) to the time change (Δt) during an event, as shown in Equation 2.31.

$$\text{ROCOF} = \frac{\Delta f}{\Delta t} \tag{2.31}$$

This is usually calculated over a set time period. Typically set to a time period of 500 ms to avoid noise and more local oscillations impacting the overall ROCOF for the system [92, 93]. However, recent research has suggested that a smaller time window could be beneficial to better capture the overall dynamics associated with future power systems and the associated ROCOF for lower inertia systems [94]. Still, ENTSO-E guidelines suggest that a maximum allowed 1 Hz/s ROCOF should be permitted with a 500 ms sampling rate because of the limitations in current equipment [90, 95]. It is also expected that future systems need to withstand a ROCOF of up to 2 Hz/s [96], while much of standard technology can only cope with a ROCOF up to 1 Hz/s [97]. As for the maximum allowed frequency deviation, this is typically defined by the time period, with the first requirement set at 49 Hz for a two-second time period, as seen in Table 2.3 [98].

In order to analyze the frequency performance of the system during different operational characteristics, different perturbations are implemented into the system. Here, the frequency response will be analyzed for four different events: a load increase and decrease, an outage event, and a short circuit. For more information regarding the specific events, refer to section 2.4.

The frequency analysis utilizes the PowerFactory library to use the different plotting functions and extract plots as CSV files. This allows for external plotting, which is done using Python to better adapt the plotting of the various results.

An example from plotting in Python can be seen in Figure 2.10. Here, two different signals can be seen, with different frequency nadir and ROCOFs marked on the diagram. The ROCOF has a similar starting value for both signals, but when the next measure point occurs after a given time, there is a different resulting value. This indicates a different rate of change for the signal, highlighting the

better ROCOF of the blue (dashed) signal. Similarly, there are different levels for the frequency nadir, with the blue (dashed) signal again showing the overall better performance, with a lower allowed deviation from the original 50 Hz value. This is a simple strategy to identify the different signals' patterns and characteristics in relation to the ratings of the two indexes. Frequency nadir and maximum allowed frequency deviation are often used for the same purpose, indicating a similar performance index.

The oscillatory behavior of the two different signals indicates different damping levels in the system. For example, the red (solid) line has a worse initial system performance but a better settling with quicker, smaller oscillations. When analyzing the signal's frequency response, different damping characteristics are evident. Thus, the frequency stability, rotor angle small signal stability, and damping characteristics are closely interlinked and can be combined to better understand the system's different parameter characteristics.

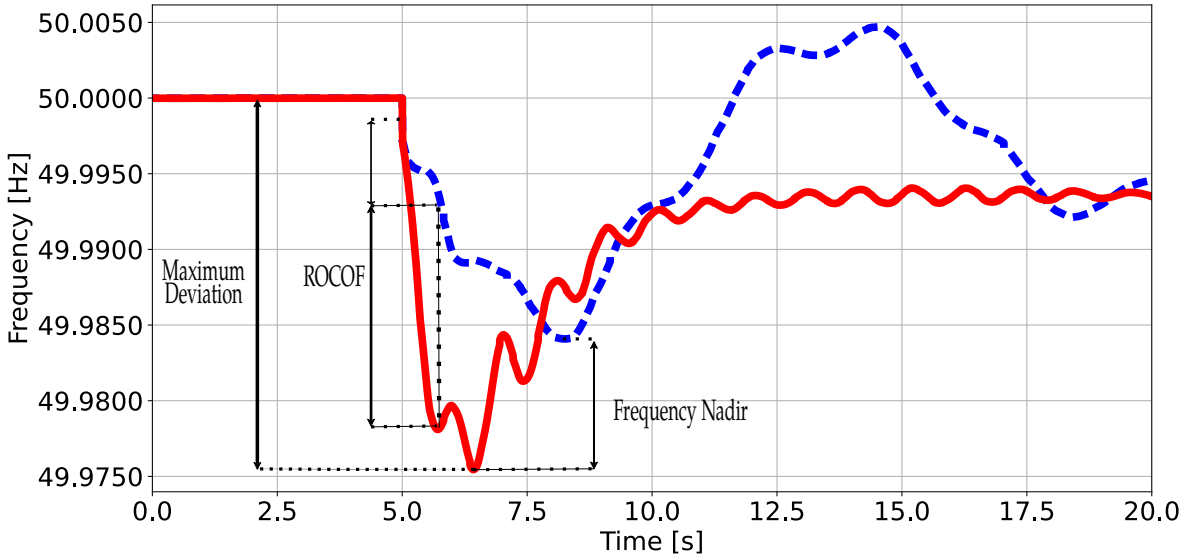


Figure 2.10: Frequency response with different damping characteristics and calculation parameters.

Frequency stability was part of the original three-step stability classification before integrating power electronics, and the definition of frequency stability has not changed [79]. However, frequency stability itself is impacted by the penetration of power electronics. Inertia plays a pivotal part in the frequency response of power systems and results from kinetic energy in large synchronous generators. Thus, as these generation units are replaced by power-electronic interfaced generation, the total inertia in the system will decrease [99]. This causes a problem as the "first line of defense" in combating frequency deviation is severely altered, making the system more susceptible to frequency issues. The lack of inertia causes frequency changes to occur faster, increasing the likelihood of instabilities [79]. However, since the limiting factors in providing frequency response in PEI generation primarily come from the response time of electronics and electrical equipment instead of mechanical systems like traditional synchronous generation, the primary frequency response can be quicker if adequate controller tuning is applied [79, 81]. Highlighting the importance of proper controller tuning. Research has shown promise in adequate frequency response control using PEI generation [99, 100, 101, 102]. Some suggestions have been made to minimize the reduction in inertia through a de-commitment and re-dispatch structure that leads to more spinning generation reserve to combat these challenges, as seen in the Western Wind and Solar Integration study [103]. However, further studies are necessary to ensure the reliable operation of future power systems, especially as further improvements are made to power electronics and their surrounding control strategies [79, 81].

Frequency stability is one of the most vital parts of power system operation. It is one of the main characteristics of power systems and their operation, and a deviation outside allowed values can result in cascading effects, damaging components, and risking large-scale system blackouts. Additionally, the frequency is highly impacted by the paradigm shift for future power systems, which highlights the

importance of proper analysis for future systems to better understand the future of frequency stability.

2.4. Perturbations

Different perturbations are often introduced into dynamic studies of power systems to analyze their impacts on system stability and operation. These perturbations result in different characteristics impacting frequency and rotor angle stability analysis, forming the basis for future studies in these areas.

The different perturbations will be briefly described before explaining their specific characteristics and utilization through PowerFactory. Different perturbations are implemented for further analysis: a load increase for rotor angle analysis and a load increase and decrease, short-circuit, and outage event for frequency stability.

Short-circuit Event: Short-circuits can occur in various combinations, such as phase-phase, phase-ground, or phase-phase-ground faults, each with distinct characteristics and impacts. According to IEC 60909-0, a short-circuit is an accidental or intentional connection between conductors through low resistance or impedance [104]. These faults can be caused by high temperatures, overvoltages, arcing due to moisture, or external factors like storms. They result in large currents that can interrupt power supply, damage components, and stress equipment. The severity of a short circuit depends on factors such as fault location, system topology, generator strength, and grounding setup. Analyzing different short circuits under various conditions is crucial for accurate results. With the increase of power electronics in power systems, the short-circuit capacity is typically reduced, impacting protection systems and requiring further investigation into dynamic responses.

In PowerFactory, the short circuit is modeled as occurring at the tie line between the 380 kV and 150 kV areas. It occurs at one of the tie lines and lasts 0.1 seconds before being cleared. This short circuit occurs at the midpoint between the different voltage levels and is represented as a three-phase fault.

Load Event: Compared to short-circuit events, load events involve smaller changes due to natural cycles and varying customer demands. These can be characterized using linearization techniques and are classified as small-transient or small-signal when analyzing rotor angle stability. Load events are typically measured in percentage points and can involve both increases and decreases. Factors such as total load increase, location, system loading, and topology influence these events. Traditionally, load changes have been predictable, but the rise of renewables and variable loads has made operating conditions more dynamic and impactful. Therefore, analyzing load events under different conditions is important to assess system resiliency.

The impact of power electronics has made load events more unpredictable. Historically, load patterns were well understood, but traditional patterns are changing with the shift to power electronic interfaced loads and increased consumer-based generation like solar PV. This transition requires updated analysis and dynamic simulations to maintain the load-generation equilibrium.

For this analysis, both a load increase and decrease are implemented as perturbations. A load increase of 5.6 GW and a load decrease of 10 GW are introduced and occur in one step, spread across the loads in the system. This is distributed across all load centers represented in the system, scaled based on their respective individual scaling. PowerFactory allows for direct control of the size and duration of these perturbations.

Outage Event: Outage events involve the failure of system components such as transmission lines, cables, generators, and transformers, leading to changes in topology, generation capacity, or transmission capacity. These changes can significantly impact the system's dynamic stability. While short circuits often lead to outages due to safety measures, outages can also result from various factors, including maintenance and external influences. Outages are categorized into planned, semi-forced, and forced scenarios [13]. External factors, such as environmental conditions, play a major role in these failures, making them difficult to predict. Therefore, it is crucial to test different types of outages to understand

their effects on system performance.

For this analysis, the outage event implementation involves one of the nuclear power plants due to their similar generation dispatches in all the different cases and events covered. This allows for a better representation of events across all scenarios. The nuclear power plant located at MVL380, with a constant dispatch of 1 GW, is chosen. This setup is directly configured in PowerFactory, allowing full controllability of system events.

Different perturbations are introduced into the system to assess its overall stability and analyze its response. This approach enables a comprehensive comparative analysis of various events under different operational characteristics rooted in both frequency and rotor angle stability.
This methodology ensures the creation of a safe and robust power system capable of withstanding multiple disturbances. This is particularly important in evolving power systems, where the increase in power electronics can affect how disturbances impact system performance.

3

Methodology

This chapter discusses the methodology of the thesis. The thesis project consisted of three major stages: a scenario analysis, synthetic model adaptation, and a set of case studies utilizing the aforementioned work based on research areas identified in the literature review. By following this three-part approach, a comprehensive examination of key factors essential for future power system analysis and evaluation was performed. The general overview of the thesis work can be seen in Figure 3.1. The respective sections will give a more detailed overview of the different methodologies.

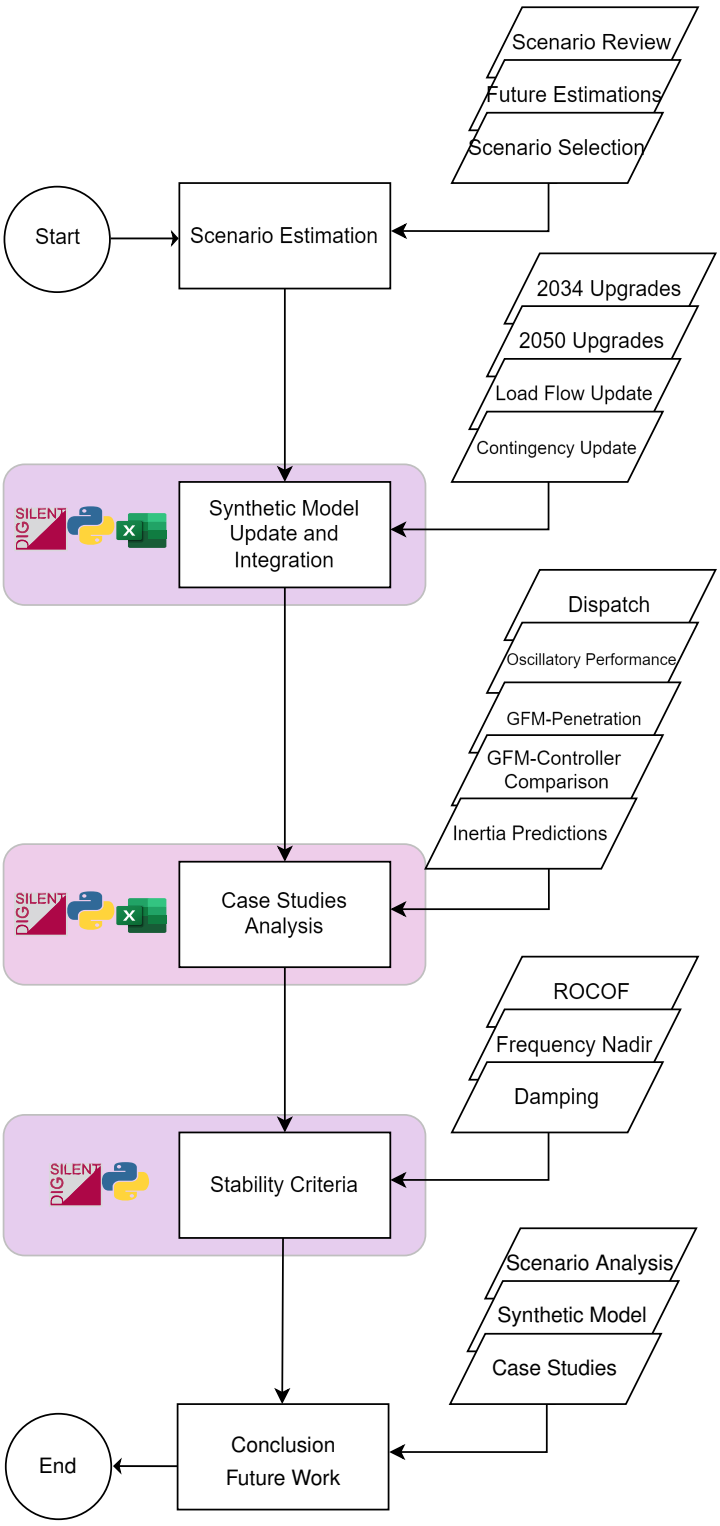


Figure 3.1: Flowchart of the general thesis work structure.

3.1. Scenario Analysis

The first stage comprised the scenario analysis found in chapter 4, which investigated different reports and projects highlighting scenarios for the future Dutch power system towards 2050. A thorough understanding of the future Dutch power system was developed by comparing and analyzing multiple reports and scenarios within each report, serving as the basis for subsequent work. The focus was on significant scientific research and credible future analyses. A critical evaluation of the relevance of different reports and their findings was conducted, aligning with specific future projections from the Dutch government and their policy-making. Three scenarios were highlighted and suggested for future analysis. One of these was then selected and implemented into the synthetic model for further analysis.

A flowchart depicting the main steps of the scenario analysis can be seen in Figure 3.2.

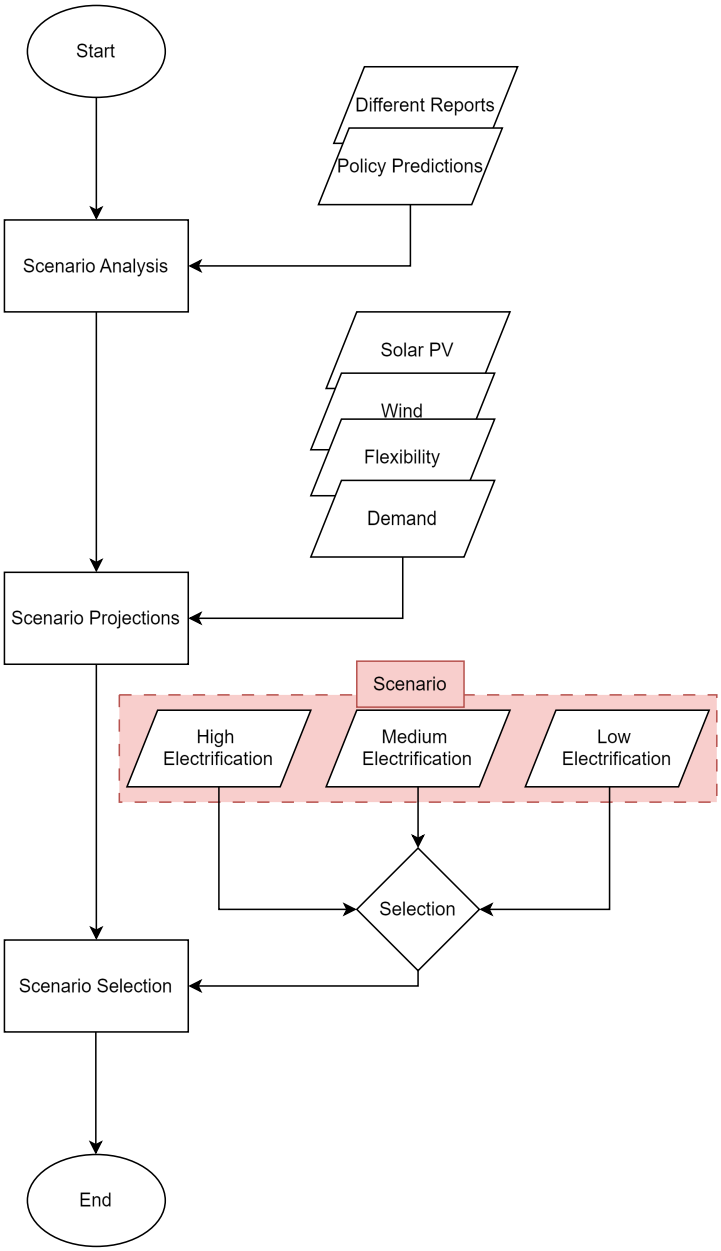


Figure 3.2: Flowchart of the scenario estimation process.

The first part of the analysis was done by analyzing and comparing the different projections from the reports highlighted in Table 3.1. Here, multiple projects with projections towards 2050 were analyzed

and compared to better understand the future projections of the Dutch power system. The different reports and their relevance to future Dutch policy guidance were compared and analyzed based on the respective policy predictions with regard to electrification levels in the future Dutch power system. This included different projections based on various policy adaptations, such as a national or international energy emphasis. A similar analysis was done for the 2030 representation of the synthetic model in [105] and was here extended towards 2050 projections. New reports that were published after the initial analysis and updates of certain reports were added to the analysis. The different reports utilized for the study can be seen in Table 3.1.

Table 3.1: Comparative Overview of Energy Infrastructure Studies and Projections for the Netherlands (2023-2050)

Project Title	**Abbreviation**	**Organization**	**Published**	**Scenario Year**	**Citation**
Integrale Infrastructuurverkenning 2030-2050	II3050-2	Netbeheer Nederland	2023 (Updated)	2050	[106, 107]
The Energy System of the Future	ESOTF2050	Netbeheer Nederland	2023	2050	[108, 109]
Infrastructure Outlook 2050	IO-2050	Tennet, Gasunie	2019	2050	[110]
Phase II - Pathways to 2050	IO-2050-2	Tennet, Gasunie	2020	2050	[111]
Net voor de Toekomst	NVDT	CE Delft	2017	2050	[112]
Nederland klimaatneutraal in 2050: vier scenarios	BS2050	Berenschot	2020	2050	[113]
Synthetic Steady-State Model of the Dutch EHV Network: Study of the impact of future additions of VRES and electrolysers	Roos2021	deRoos, Anouk(Thesis)	2021	2030/2050	[105]
Target Grid	TG	TenneT	Ongoing	2045	[114]
Ontwerp investeringsplan Net op land 2024-2033	Invest-land-2024	TenneT	2024	2033	[115]
Ontwerp investeringsplan Net op zee 2024-2033	Invest-sea-2024	TenneT	2024	2033	[116]

After the initial analysis highlighting different policy predictions and comparing the relevancy of different work, the different projections of renewable energy sources, flexibility resources, and demand levels in the power system were highlighted and compared between the various reports and policies. These were compared based on the installed levels of these resources to analyze how different scenarios differ between different reports and policy projections.

After the analysis, one of the scenario projections was identified as a prominent candidate due to its relevant work and projections, and this was utilized for implementation and representation in the synthetic model. This was represented using estimated values for different load and generation units in the system. The analysis also listed two alternative projections to allow for easier implementation of these estimations for future research if necessary. These two alternative projections made the three different projections range from the highest to lowest electrification, thus representing the whole range for future projections from highest to lowest.

The analysis aimed to gain more insights into the future projections of the Dutch power system and understand the different projection levels for various resources such as solar PV, offshore and onshore wind, and flexibility resources. This allowed for a better understanding of the intrinsic nature of future power systems and facilitated easier implementation and integration with the synthetic model.

3.2. Synthetic Model Development

Following the scenario estimation, the 2050 synthetic Dutch power system model was built upon the existing Dutch model for 2030 from [105]. The goal was to implement the necessary changes to expand the model from a 2030 to a 2050 scenario and introduce new dynamic characteristics. An overview of the flow of the model development can be seen in Figure 3.3. This was done in two parts, for 2034 and 2050.

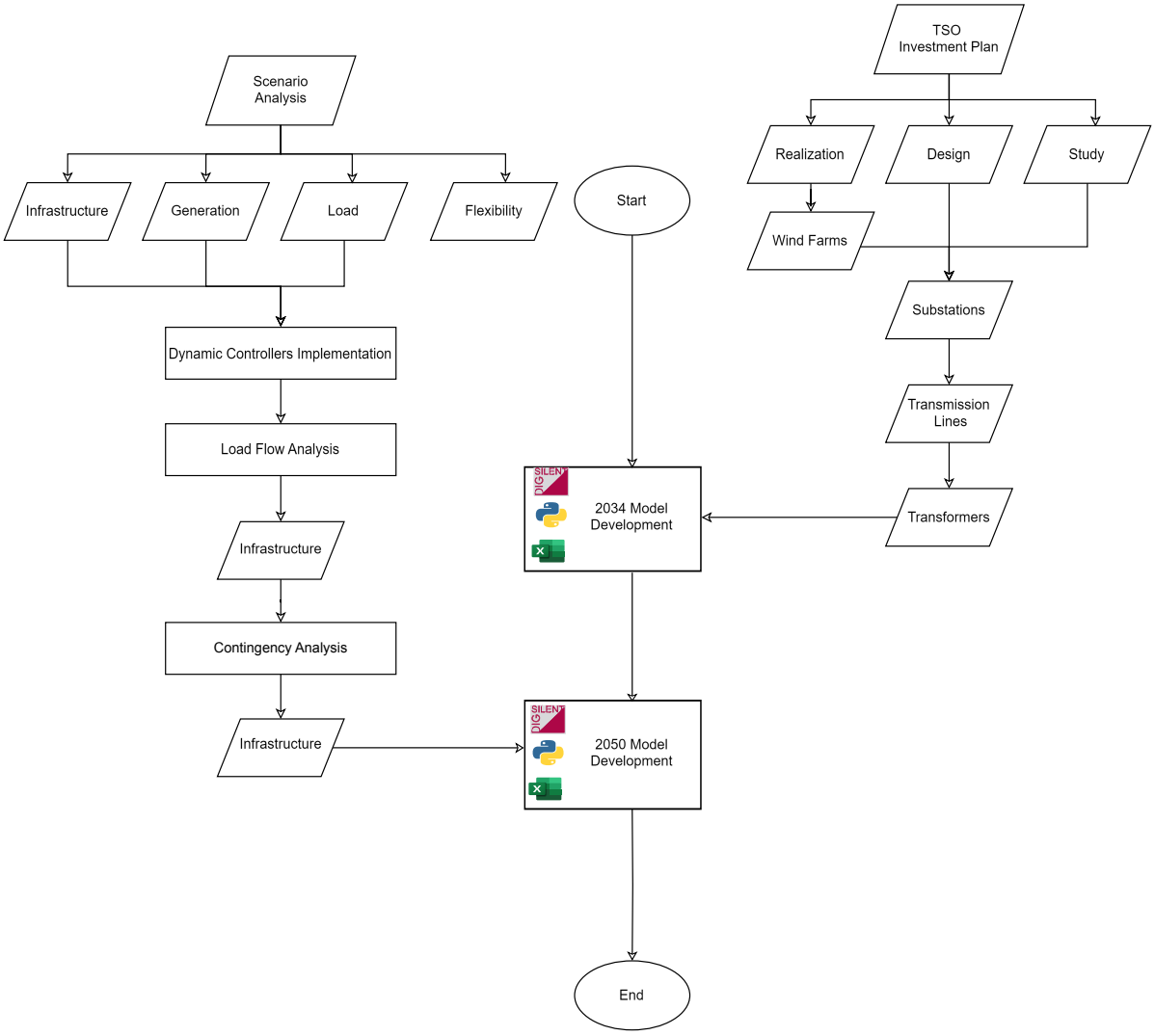


Figure 3.3: Detailed flowchart of the Synthetic Model Development process.

3.2.1. 2034 Model Development

The first part, model development towards 2034, updated the infrastructure in the synthetic model based on additional projects found in the Dutch TSO's 10-year investment plans. The original model had already implemented multiple projects in the 10-year investment plan. However, these were updated to account for all the different projects. The 10-year investment plans are divided into three different project stages: realization, design, and study, indicating the progress of the different projects.

Realization projects were in the realization stage, meaning that the project was studied, designed, and in motion to be realized. Projects in the design phase had undergone the initial study phase and were now in the process of design in terms of locations and connections. Finally, projects in the study phase were barely in motion and were still undergoing studies to assess their applicability and feasibility. The three different project stages represent the progress of certain projects, and the further along the projects were, the more public information was available. The realization stage projects typically had specific

information with regard to locations and connections, while the projects in the design and study phases were much more reliant on simplifications and estimations of the different implementations into the model. The projects ranged from substations, transmission lines, and transformer projects for all three project stages and consisted of removing, improving, and implementing different infrastructures. The realization stage projects also had specific projects regarding future wind farm projects, which were also implemented here.

3.2.2. 2050 Model Development

After the 2034 model development, the model was further developed toward 2050 scenarios. The first stage in this development was implementing the selected scenario from the scenario analysis. Here, the report and scenario analysis of the specific scenario were utilized, and different generation, load, and flexibility resources were implemented and/or scaled based on the given levels. Due to very little specific information regarding these levels, major simplifications and estimations were needed concerning the capacities and locations of the different resources implemented. All of the different resources were implemented based on the projected installed capacities in 2050, with simplifications done to implement and standardize the different capacities of the different resources in their respective locations. Similar was true for the infrastructure development. Similar to the 2034 model development, this consisted of different substations, transmission lines, and transformer upgrades. As before, there was very little information regarding the project, but rather suggestions of probable bottlenecks and areas suggested for infrastructural improvement.

The next part of the model update involved enhancing the dynamic properties of the model to better align simulations with real-life dynamic properties. This was achieved by adding more generic controllers representing grid-forming converters, changing the interconnections to dynamic representations instead of constant loads, and introducing dynamic representations of other constant demands in the system. Adding these dynamic properties allowed for a model that better highlighted interactions between different dynamic system components, providing a more accurate representation of system dynamics in future systems and allowing for a better understanding of how these resources could be adjusted to suit future systems, thus easing integration. Overall, this increased the expandability and usability of the synthetic model, enabling large-scale stability analyses.

After this, the model underwent a load flow and contingency study to allow for further improvement. This was done based on the different cases of different load and generation levels in the system, identified in Case Study A. Here, all six identified cases underwent a load flow and contingency analysis. An 80 % and 90 % loading limit were set for the load flow and contingency analysis, respectively. If any components did not adhere to these limits, these components were upgraded to withstand the power flow in the region. The contingency analysis was done based on the N-2 contingency, similar to today's Dutch power system.

The synthetic model was developed using DigSilent PowerFactory with ancillary services from Python and Excel. Because of this, certain changes and implementations added to the model in PowerFactory also had to be included and adjusted for Python and Excel. This allows for easier customization and implementation of different characteristics in the synthetic model, overall increasing adaptability. This integration between the infrastructural synthetic model development is not explicitly explained in the model development chapters, but more information can be found in Appendix C.

3.3. Case Studies

After the model development, the next step was to utilize the model for research on current and relevant topics identified in the literature review. Three state-of-the-art case studies were conducted. These case studies encompassed a wide range of analyses, including different dispatch analyses, grid-forming converter studies, and inertia studies regarding dynamic stability in the future Dutch power system.

Each case study has its own methodology section, but the general approach will be highlighted here. Refer to the respective case studies for a more detailed analysis of the different case studies. An overview of the different case studies is provided in Figure 3.4.

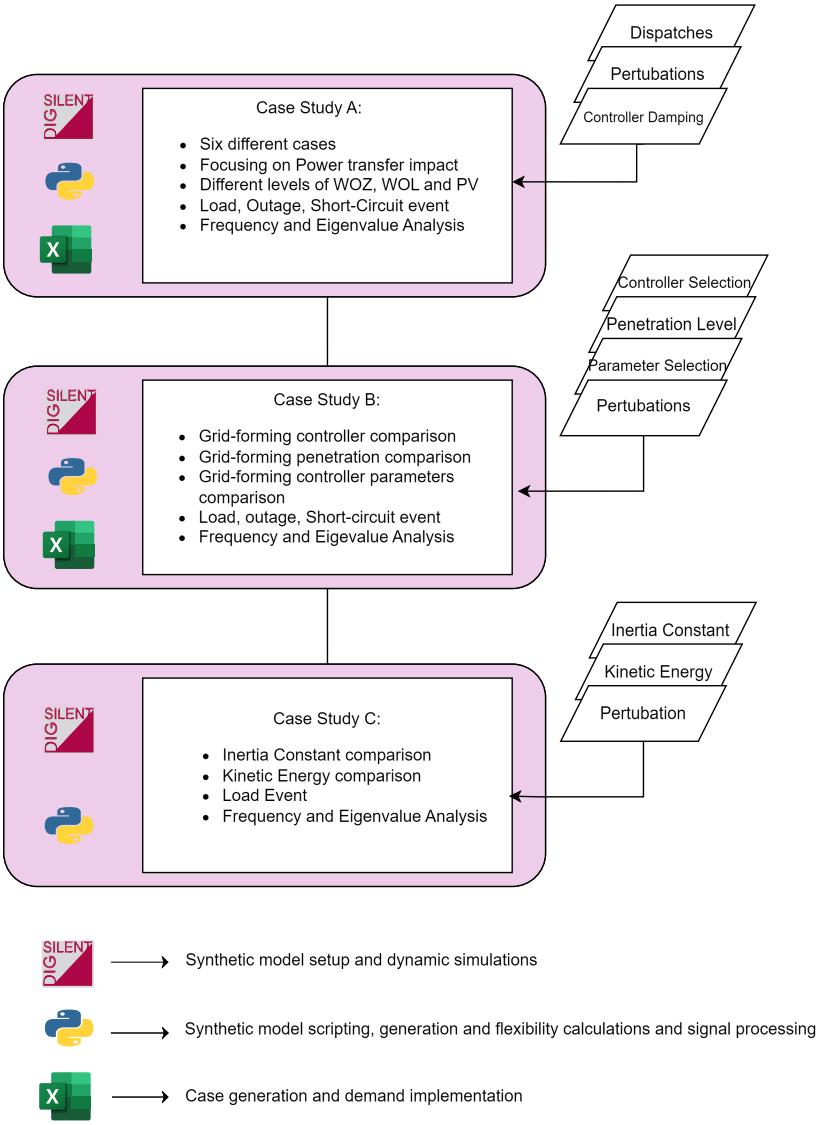


Figure 3.4: Flowchart highlighting the different case studies.

The first case study analyzed the impact of different high levels of renewable energy sources (RES) on system stability. The three sources under scrutiny were offshore wind (WOZ), onshore wind (WOL), and solar PV (PV). Six different cases were implemented, each with high levels of one of the three sources, and tested under peak and minimum demand conditions. This approach provided a comprehensive view of system stability and the impact of varying demand levels on system and flexibility loading. All six cases were examined for three different perturbations and analyzed using a frequency analysis.

The case study also examined the oscillatory performance of the system using an extended frequency analysis simulation window and an eigenvalue analysis to understand the intrinsic damping of different oscillatory modes in the system. This analysis was further expanded by testing the impact of grid-forming controller parameter selection on the damping of certain modes, with the load increase utilized as the perturbation.

The second case study focused on the characteristics associated with grid-forming controllers. All analyses were conducted for one of the cases identified in Case Study A, specifically the case with high offshore wind penetration levels and peak demand (Case A). Three main analyses were performed: the impact of different grid-forming controllers (VSM, Droop, and Synchroverter), the impact of grid-forming penetration levels in the system (25 %, 50 %, and 85 %), and the impact of different parameter values for the most critical control parameters for the three aforementioned controllers. These analyses were performed using the three perturbations for frequency analyses, while eigenvalue analyses were done for the load increase.

The third case study analyzed the impact of different inertia values on the stability of the future Dutch system. Similar to Case Study B, Case A was utilized here as well. Different inertia predictions, in the form of inertia constant and kinetic energy levels, were identified and implemented into the synthetic dynamic model using the interconnection representation to analyze the different predictions. Three different levels of kinetic energy and three different levels of inertia constant were used and simulated to analyze future projections. A frequency and eigenvalue analysis was conducted for the different levels, using the load increase as the perturbation of the system.

For Case Study A and C, the grid-forming penetration levels were set to 50 %. Similarly, for Case Study A and B, the inertia levels were set to the maximum levels. These simplifications allowed for a better general overview and similarity across different case studies and events. If the grid-forming penetration or inertia levels are changed for certain simulations, this is specified in the relevant case study.

An overview of the different perturbations and their utilization for the different analyses and case studies can be found in Table 3.2. Furthermore, a map showing the location of the various events and the location of the frequency measurement unit can be seen in Figure 3.5. Here, the 2030 model is depicted because of its zone division and easier identification of the different locations.

Table 3.2: Perturbations, locations, impacts, and descriptions for various events, along with the type of analysis and case studies considered

Pertubation	Location	Impact	Description	Frequency Analysis	Eigenvalue Analysis	Case Study
Outage Event	MVL380	Nuclear Power Plant (2 GW)	Outage of the nuclear power plant at MVL380. Dispatch levels of 1 GW in all scenarios	x		A, B
Short-circuit Event	B150	Tie-line	Short-circuit at one of the tie-lines between 380 kV and 150 kV level at Borsselle. Lasting 0.1 seconds before being cleared	x		A, B
Load Increase	System-Wide	5.6 GW	System-wide load increase, scaled based on the respective ratings of each load.	x	x	A, B, C
Load Decrease	System-Wide	10 GW	System-wide load decrease, scaled based on the respective ratings of each load.	x		A

Another important aspect of the work involved the different ancillary services provided for the synthetic model. This was achieved by using the sophisticated integration of Python and Excel resources in PowerFactory, allowing for seamless integration with open-access resources and enabling the automated execution of various operations in PowerFactory. These Python code structures ranged from simple integration of dispatches found in Excel to PowerFactory to automated tasks such as changing system parameters and performing different calculations for grid-forming and flexibility resources. The overall Python framework was highly integrated, allowing for automated calculations and integration of different system scenarios and characteristics, and signal processing. This integration was a major part

of the work conducted in the study. The work and a tutorial for integrating and further using these resources can be found in Appendix C.

The case studies build upon the foundational research and work done for the scenario analysis and the synthetic model adaptation. They address the research questions outlined in the introduction, allowing for sophisticated research to bridge some of the literature gaps regarding fast-active power balancing in large-scale systems. This was achieved using a state-of-the-art synthetic dynamic model rooted in real scenario estimations towards 2050.

Findings from the case studies will also be transformed into journal and conference research publications, highlighting their importance and significance in the field of power systems.

Finally, a conclusion and suggestions for future work are provided. The synthetic grid model, combined with the scenario analysis and various case studies, offered multiple perspectives for future research. The main findings of the thesis work are presented, along with recommendations to streamline future research.

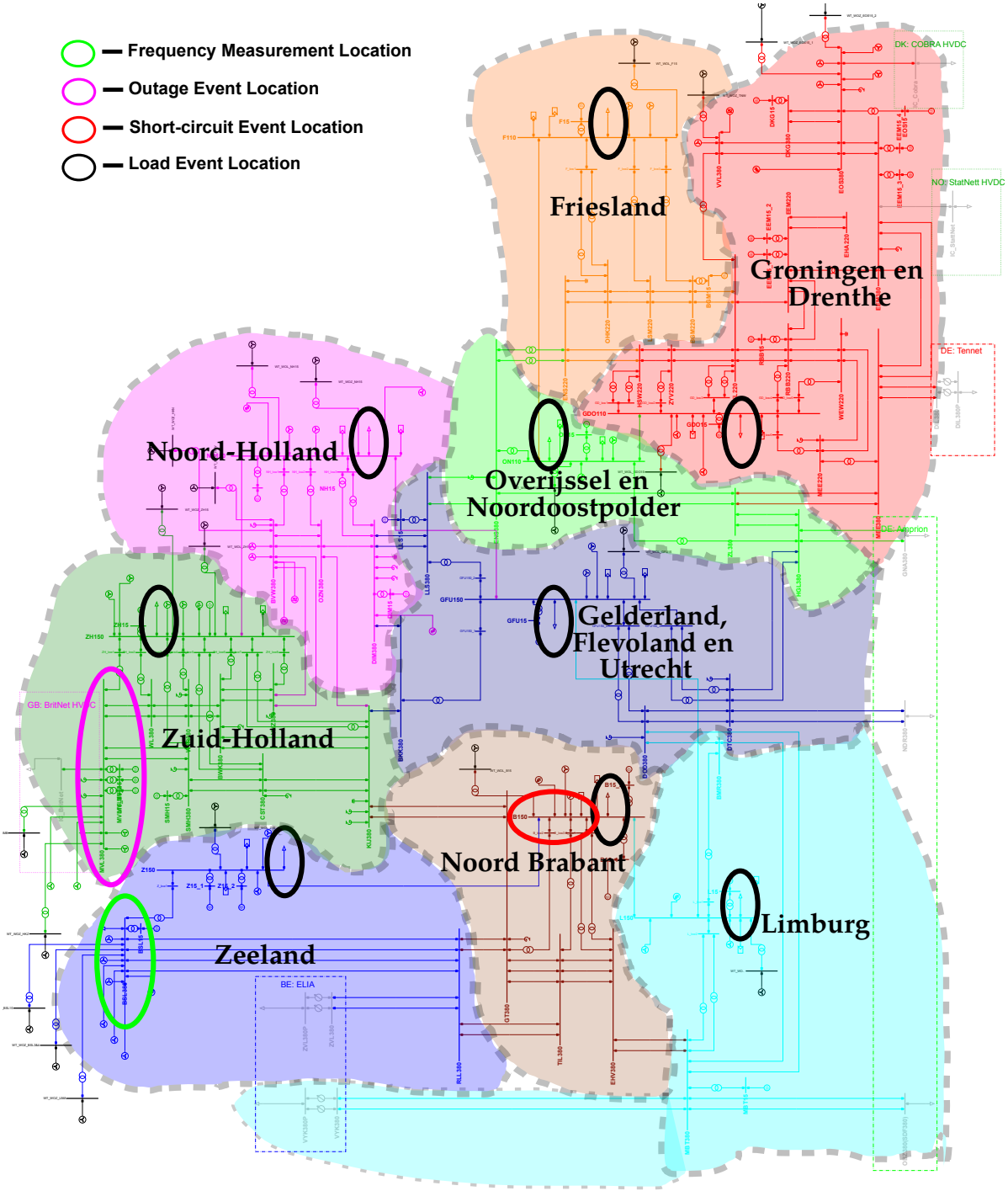


Figure 3.5: Overview of different event and measurement locations in the 2030 Synthetic power system model.

4

Future Scenarios Analysis

4.1. Introduction

In this section, the analysis and comparison of different scenarios for the Dutch power system towards 2050 will be conducted. This work will examine the traditional groupings of future scenario estimations, typically categorized according to policy-making guidelines. It will compare these various scenarios across multiple reports, highlighting their differences. Subsequently, it will present the main estimations for the most significant future scenarios and display the results in graphical diagrams. This approach facilitates easy replication and integration of different scenarios into individual models, thereby streamlining the process of analyzing the impacts of these future scenarios on various system parameters.

One of the scenarios presented will be selected as the baseline scenario. This baseline will be used for further work on upgrading the dynamic model and analyzing the case studies.

4.2. Analysis of Future Scenarios

This section will delve into identifying future Dutch power system scenarios, including the share of renewables and future operational scenarios towards 2050. Various scenarios from different vendors offer diverse projections of the generation mix, many of which are of significant interest.

The main projects and reports of interest were highlighted in Table 3.1, with a comparison provided in Table 4.1. Multiple reports analyze projections towards 2050; however, due to the disparities in future system trajectories and policies, these reports often include multiple scenarios for different policies. There is no unanimous agreement on the direction the Netherlands will take by 2050. Consequently, a detailed analysis and comparison are essential for properly implementing scenarios for future systems.

One issue with using 2050 as a benchmark is the lack of detailed analysis regarding specific projects of generation and demand and the future infrastructure. Therefore, these details will be supplemented based on more detailed models using 2030-2034 scenarios and further scaled towards 2050 scenarios.

A similar analysis was conducted in [105]. However, due to ongoing developments in power systems and policies both nationally and internationally, this work will serve as an update to that analysis, highlighting various new characteristics and future scenarios.

Table 4.1: Detailed Explanation of Abbreviations Used in Energy Infrastructure Studies

Abbreviation	Description
II3050-2 [106]	Infrastructure outlook for the Netherlands between 2030 and 2050, aimed at developing insights into the needs for infrastructure, flexibility, and system integration for the future power system. Introduces four different scenarios. Produced by Netbeheer Nederland, comprising multiple energy operators in the Netherlands.
ESOTF2050 [108]	Interim step detailing scenarios presented in II3050, providing a foundation for better understanding the projections in II3050-2.
IO-2050 [110]	Infrastructure outlook for the Netherlands from 2030 to 2050, developed by Gasunie and TenneT. Focuses on future systems in the Netherlands and Germany for the power and gas system as an integrated energy infrastructure. Introduces three different scenarios.
IO-2050-2 [111]	Phase two of the IO-2050 study, expanding on the previous study with more detailed projections of renewable energy sources and emphasizing significant changes in future systems. Introduces three updated scenarios based on the previous report and additional analyzed scenarios.
NVDT [112]	Outlook for the Dutch power system towards 2050. An older outlook compared to II3050-2 and IO-2050-2 provides valuable regional projections. Produced by Netbeheer Nederland.
BS2050 [113]	Study presenting different scenarios for the Dutch power system towards 2050. Introduces four scenarios and serves as the final report for the II3050 phase 1 scenario. Commissioned by TenneT.
Roos2021 [105]	A master's thesis from TU Delft that introduces a steady-state model in PowerFactory for analyzing the Dutch power system towards 2030. The report also examines scenarios towards 2050 and proposes future scenarios.
TG [114]	The Target Grid project by TenneT, visualizing the Dutch power system towards 2045. This project is currently ongoing.
Invest-land-2024 [115]	TenneT's 10-year investment plan for onshore investments between 2024 and 2033. Although it has a shorter timeframe, it provides a detailed analysis of TenneT's plans.
Invest-sea-2024 [116]	Similar to Invest-land-2024, but focuses on TenneT's offshore power activities during the same period.

Many of the reports establish multiple scenarios in their projections to account for various approaches by the Dutch government and the EU/international trade organizations. This enables analysis of a wider range of scenarios. However, this also introduces additional uncertainty as the individual scenarios between reports, while similar in approach, propose different levels of both generation and demand.

The different scenarios, although varied, are typically based on similar directions of governmental and/or technological advancements. Some reports, such as [110, 111], consist of three scenarios, while others, such as [106, 112], consist of four. Many studies build on previous scenarios, advancing them with new data and projections.

1. **National Scenario:** National Scenarios typically emphasize a centralized approach where the government plays a pivotal role in shaping the energy landscape. This involves a strategic push towards enhancing energy efficiency and steering the energy mix from a national perspective. Governmental actions may include regulations and support for technological innovations, financial backing for critical infrastructure projects, and fostering the growth of new industries like synthetic fuel production. Such scenarios often advocate for electrifying existing sectors and developing integrated heating solutions powered by renewable sources. Large-scale energy projects, including the expansive use of offshore wind and the strategic deployment of nuclear power, are common. Additionally, alternative energy carriers like green hydrogen are considered vital for system balance and industrial applications, emphasizing a transition to more sustainable practices [105,

106, 110].

This scenario typically estimates large amounts of offshore and onshore wind capacity. Offshore wind capacity ranges from 53 to 72 GW, onshore wind capacity ranges from 14 to 20 GW, while solar PV capacity ranges from 34 to 173 GW. Flexibility resources are also considered necessary to accommodate these high levels of renewable generation. Battery capacity is projected to range from 50 to 60 GW, and electrolysis capacity (hydrogen) is estimated to be between 39-45 and 60 GW. The scenarios also estimate different ranges of interconnection capacity, from 15 to 18.8 GW [106, 110, 112, 113].

2. **Regional Scenario:** Regional energy scenarios often emphasize a decentralized approach, focusing on climate-neutral solutions driven by local businesses and community initiatives. This approach is exemplified by the report [106], which refers to this scenario as the decentralized initiatives scenario. In such frameworks, a significant emphasis is placed on the autonomy of local entities in making energy-related decisions. This could involve leveraging cost-effective, region-specific options or adopting measures grounded in environmental idealism. Local governments frequently employ incentive programs to promote sustainable practices, which might lead to increased projects that efficiently use local resources. However, in these scenarios, energy-intensive industries may face stringent controls or even shutdowns as the transition toward sustainable energy sources progresses, possibly leading to a decline in certain traditional industrial sectors. Adopting various local heating solutions, such as heat pumps, green hydrogen, and geothermal energy, is encouraged, alongside a substantial increase in onshore renewable energy generation [105, 106, 110, 112, 113].

   The ranges in the regional scenario offer various levels of different renewables, flexibility, and interconnections. Offshore wind capacity ranges from 26 to 45 GW, onshore wind from 16 to 25 GW, and solar PV from 84 to 183 GW. Battery flexibility is projected to range from 6 to 60 GW, and hydrogen capacity from 33 to 75 GW. Interconnection capacity ranges from 15 to 18.8 GW [106, 110, 112, 113].
3. **European Integration Scenario:** The European Integration Scenario envisions a cohesive and streamlined energy framework across European nations, achieved by harmonizing national energy policies and pooling collective resources. The aim is to establish energy independence from non-European regions and establish a unified energy strategy. This scenario foresees widespread production and utilization of green gas and robust solar and wind energy expansion alongside nuclear power. It highlights the significant use of the North Sea for offshore wind in collaboration with neighboring countries. The scenario advocates for sustainable industrial practices through electrification and the strategic use of biomass and hydrogen. It also envisions comprehensive carbon capture and storage applications, supplemented by resource recycling and minimal use of new fossil materials. Neighborhood strategies are key in this scenario, driving the sustainability of built environments through regional heating networks and hybrid energy systems, which help to balance power demand. A broad enhancement of trans-European networks, including electric vehicle charging and high-speed rail, supports widespread electrification of transport [105, 106, 113].

   Not all reports include a specific European scenario, thus the variations are not as extensive. The primary sources utilizing this scenario are II3050-2 [106] and BS2050 [113]. In this scenario, offshore wind capacity ranges from 38 to 42 GW, onshore wind capacity is projected at 10 GW, solar PV capacity ranges from 42 to 126 GW, battery capacity ranges from 4 to 38 GW, and hydrogen capacity is projected at 16 GW. Interconnection capacity ranges from 15 to 28.8 GW [106, 113].
4. **International Trade Scenario:** The International Trade Scenario strategically focuses on integrating the Dutch economy into the global energy and raw materials markets. This approach leverages the competitive nature of international trade to procure cost-effective solutions, emphasizing free trade principles. The scenario supports importing economically produced climate-neutral energy carriers, transforming the Netherlands into a key transit hub, particularly for hydrogen. Domestic industries adapt through electrification and sustainable resource use, with some energy-intensive processes relocating overseas. The Netherlands concurrently invests in green hydrogen production linked to its offshore wind energy projects, reducing the imperative for self-sufficiency in energy

production [105, 106, 110, 112, 113].

The variations in this scenario are also significant. Offshore wind capacity ranges from 6 to 46 GW, onshore wind from 5 to 10 GW, and solar PV from 16 to 100 GW. Battery capacity is projected to range from 6 to 70 GW, and hydrogen capacity from 1 to 41 GW. Interconnection capacity ranges from 15 to 28.8 GW [106, 110, 113, 112].

The scenarios display various generation, demand, flexibility, and interconnection levels. Additional reports also present other scenarios, such as [111]. These scenarios build on the points discussed in [110], which included national, international, and local scenarios. The phase 2 project [111] introduces three new scenarios: EL & RES, GAS & RES+, and EL & RES+. These scenarios are based on different supply and demand levels and the scenarios described in the phase 1 project [110]. The different supply scenarios are categorized by the level of renewables implemented, specifically high RES and ambitious RES levels. The high RES scenario uses the same parameters as the RES levels in NVDT [112], while the ambitious RES scenario uses the maximum reasonable levels provided in IO-2050 [110].

The demand scenarios are based on two distinct scenarios: high electrification and high gas. In the high electrification scenario, a large number of applications are electrified, and the heating sector has a high share of electric heat pumps. The transport sector is also highly electrified, but hydrogen still plays a significant role. Many processes in the industrial sector are electrified, and power-to-x (P2X) technologies, which use renewables to create other energy sources, are applied in industrial plants to decarbonize processes. Other processes, such as those in the chemical industry or steel production, also utilize hydrogen-based methods to increase energy efficiency.

The high gas demand scenario employs a broader mix of technologies, not relying on electricity as the primary source. Here, hydrogen, methane, and other renewable energy sources constitute a significant portion of the demand mix. This scenario applies across all sectors, with thermal technologies used in the heating sector, various technologies in the transport sector, and hydrogen and methane dominating end-use energy demand. These scenarios are combined in three ways: EL & RES combines the electrification scenario with the high RES scenario, GAS & RES+ combines the gas demand scenario with ambitious RES levels, and EL & RES+ combines the electrification and ambitious RES levels.

Another significant project for the future power system is the Target Grid by the Dutch TSO TenneT [114]. This power system vision for 2045 provides a robust and realistic outlook, with detailed planning for different regions. The project aims to create a future power system scenario that the TSO can actively work towards. Having a clear target makes the goals more attainable than multiple scenarios in a report. The project is based on findings from the II3050 [106] project and uses the scenario that assumes the highest level of electrification in society to prepare for all potential electricity demand scenarios, known as the national scenario. Planning for this scenario allows for easier scaling if future demand is lower than expected. The scenario assumes a doubling of electricity demand, with electricity's share in energy systems growing from around 20 % today to between 40 % and 60 %. It also anticipates generation growing faster than demand and a strong increase in flexibility resources in electricity generation and use. The Target Grid project, still under development, lacks full details but remains a valuable resource for different simulation scenarios. However, this project becomes vital because the Target Grid is based on the II3050 project. The Target Grid plans also highlight various transmission upgrades needed to transfer power from different regions and offshore wind farms, calling for investments in DC links, upgraded AC transmission lines, and energy hubs to better utilize variable renewable generation across regions and countries. They also highlight current and future interconnection developments with other countries.

Other important resources include Invest-land-2024 [115] and Invest-sea-2024 [116]. These are TenneT's 10-year development plans for power system investments. While they do not reflect the levels of generation and demand for 2050, they provide detailed plans for investments over the next 10 years. These plans offer more accurate projections and can be used as a baseline for further integration of the 2050 scenarios.

Multiple reports include multiple scenarios with significant variations in generation capacity, type, and demand. This variability makes it challenging to create accurate simulation scenarios. It is always

recommended to account for "worst-case" scenarios, making the highest electrification scenarios the most important. Although they do not provide complete accuracy for 2050, they offer estimates for future scenarios regarding generation levels and placements when used alongside other resources and serve as a good benchmark for future system analysis.

4.3. Estimates of Future Scenarios

This section provides a detailed comparative analysis of the projected levels of generation and demand across various future scenarios in different projects. This analysis aims to identify similarities between different studies and scenarios to establish realistic and feasible scenarios. It covers the projected capacities for solar PV, offshore and onshore wind, various flexibility resources such as P2X and batteries, and future demand projections.

4.3.1. Solar PV

The analysis of different scenarios for solar PV levels reveals a significant range of projections, from 16 to 183 GW, as illustrated in Figure 4.1. This wide range indicates substantial differences between scenarios and highlights notable gaps within similar scenarios. For instance, the NVDT and IO-2050 national scenarios estimate 34 GW of installed solar PV, whereas the II3050-2 and BS2050 national scenarios project up to 173 GW.

Deciding on precise scenarios is complex, necessitating the consideration of a broad spectrum of possibilities. The scenarios with the lowest solar capacity installations are the international ones, which makes sense as they do not emphasize a self-sufficient energy system but rather view the Netherlands as part of a larger international energy system reliant on various vendors and countries. This reduces the need for domestic energy production, resulting in lower solar PV installations, with estimates ranging from 16 GW for the NVDT and IO-2050 projects to 100 GW for the II3050-2 project.

The EU scenarios fall in the middle range, with the BS2050 estimating 42 GW and the II3050-2 estimating 126 GW of installed capacity. The highest solar PV installations are projected in the regional scenarios, which focus heavily on regional energy infrastructure developments. In these scenarios, the installed capacity ranges from 84 GW for the NVDT and IO-2050 to 89 GW for the BS2050 and up to 183 GW for the II3050-2 scenarios.

The IO-2050-2 scenarios, based on the levels found in IO-2050, show consistent patterns. For example, the EL & RES+ scenario aligns with the national levels found in NVDT and IO-2050, while the Gas & RES+ and El & RES scenarios match the regional levels in NVDT and IO-2050, differing by only 1 GW. The scenarios for 2035 also exhibit a wide range of capacity installations, from 52.6 GW to 98.2 GW, indicating that even near-term projections can vary significantly.

This variability underscores the importance of considering a range of potential future scenarios to effectively prepare for various possible outcomes.

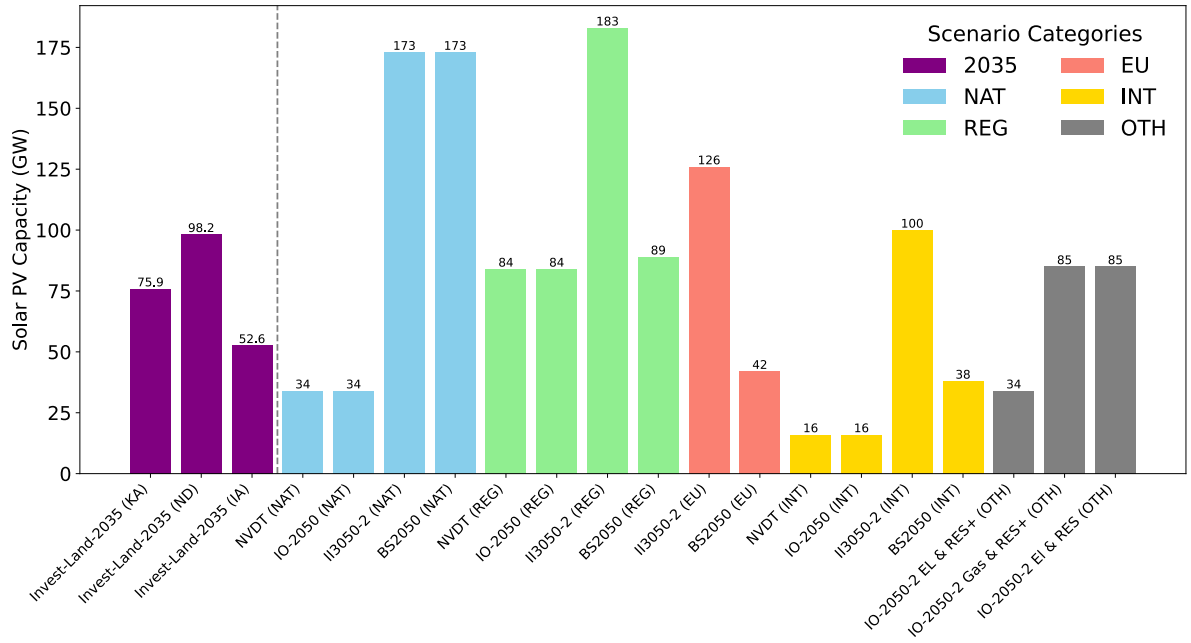


Figure 4.1: Projected installed solar PV capacity for different scenarios in 2035 and 2050 in the Netherlands.

4.3.2. Offshore Wind

The projections for offshore wind capacity show similarities to those of solar PV installations, with a wide variation between scenarios and different studies, ranging from 6 to 72 GW, as shown in Figure 4.2. The lowest values are from the international scenarios of NVDT and IO-2050, which estimate only 6 GW. In contrast, the international scenarios of II3050-2 and BS2050 predict much higher values, with 46 GW and 38 GW, respectively. The highest projections come from the national scenarios, with NVDT and IO-2050 estimating 53 GW, and II3050-2 and BS2050 projecting 72 GW.

The regional scenarios project significantly lower values, similar to the case for solar PV, but without an extremely high value for the II3050-2 scenario. These projections are 26 GW for the NVDT and IO-2050 scenarios, while II3050-2 and BS2050 predict 45 GW and 43 GW, respectively. The European scenarios align closely with the regional values, with II3050-2 and BS2050 both projecting 45 GW and 43 GW. All three IO-2050-2 scenarios predict the same installed capacity of 53 GW. The 2035 scenarios are more consistent than the solar PV projections, ranging from 25.5 GW to 29.5 GW.

The international scenarios show the widest range of estimates, from 6 GW in the NVDT and IO-2050 reports to 46 GW and 38 GW in the II3050-2 and BS2050 reports. The NVDT and IO-2050 reports, published in 2017 and 2019, respectively, did not prioritize large-scale offshore wind integration for the international scenario, focusing instead on other aspects. However, the IO-2050-2 report, published in 2020, emphasized the Dutch government's significant potential and strategic policy toward future power systems, focused on the maximum projected levels of 53 GW for all three scenarios.

Due to the longer implementation processes associated with offshore wind projects, the disparities between the 2035 scenarios are much smaller than those for solar PV, ranging from 25.5 GW to 29.5 GW. This indicates a general consensus on projections towards 2035, although there are still large deviations between different scenarios.

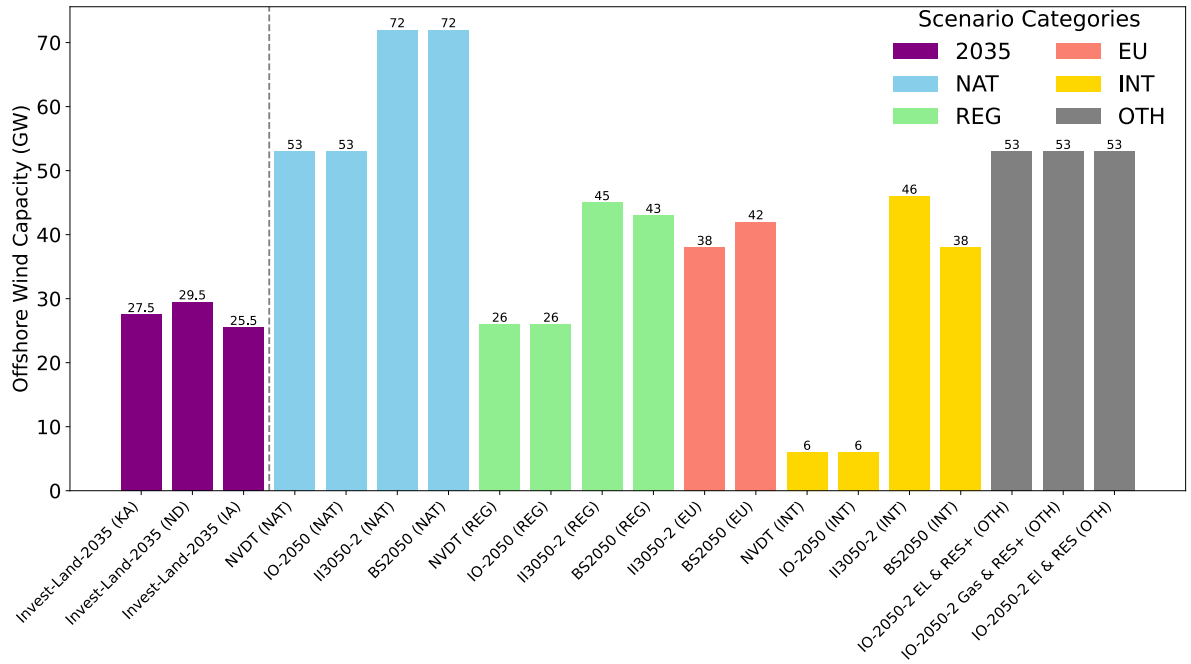


Figure 4.2: Projected installed offshore wind capacity for different scenarios in 2035 and 2050 in the Netherlands.

4.3.3. Onshore Wind

The onshore wind capacity scenarios exhibit similar trends to those observed for solar PV and offshore wind. The different installed capacities for each scenario are presented in Figure 4.3. The range of projections varies from 5 GW to 20 GW, highlighting deviations between different reports and scenarios.

The international scenarios have the lowest installed capacities, particularly in the NVDT and IO-2050 projects, while the highest capacities are projected in the II3050-2 and BS2050 national scenarios. The EU scenarios show similar levels to the international scenarios, with both the II3050-2 and BS2050 scenarios estimating 10 GW of installed capacity. In contrast, the national scenarios of IO-2050 and NVDT estimate 14 GW, compared to 20 GW in the II3050-2 and BS2050 scenarios.

Regional scenarios display similar values, with the maximum projection of 20 GW in the BS2050 scenario, while NVDT and IO-2050 estimate 16 GW, and II3050-2 projects 15 GW of installed capacity. The IO-2050-2 scenarios show consistency again, with the EL & RES+ scenario estimating 14 GW and the other two projecting 16 GW of installed capacity.

There is a notable disparity in the projections for the 2035 scenarios, which is larger than offshore wind. Onshore wind projects have a shorter commission time than offshore wind, resulting in a greater margin of error influenced by future policy decisions. Generally, national and regional scenarios project higher capacities than European and international scenarios. This difference is due to the reduced necessity for domestic power production in the European and international scenarios, which rely more on the broader European and international community.

Overall, the total installed capacity for onshore wind is lower than both offshore wind and solar PV, resulting in less variation between scenarios. However, variation is still evident.

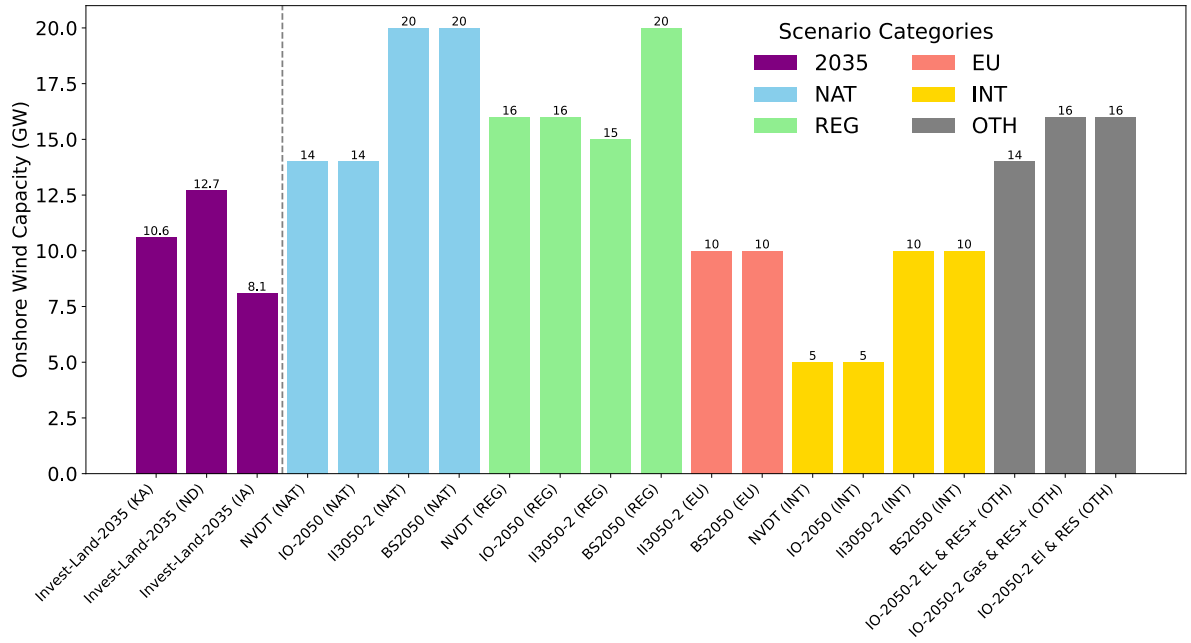


Figure 4.3: Installed Onshore Wind capacity for different scenarios in 2035 and 2050 in the Netherlands.

4.3.4. Flexibility Resources

The flexibility capacity in different scenarios exhibits considerable variation, depending on both the scenario and the project. Figure 4.4 illustrates the levels of installed flexibility, with battery, hydrogen, and interconnection capacities represented individually in different colors, along with an additional category labeled "other." This category encompasses various resources, including power-to-heat and adjustable power, depending upon different reports. The specific installed capacities of the different resources are indicated numerically on the respective graphs.

Notably, one of the most significant variations occurs within the same scenario, specifically the international scenario, and between the II3050-2 and BS2050 projects. The BS2050 project proposes minimal flexibility resources due to its international focus and policy, whereas the II3050-2 project has the highest overall flexibility resources among all scenarios. Compared to the other international scenarios in NVDT and IO-2050, the II3050-2 international scenario also shows a substantially larger installed capacity.

For most scenarios, the primary flexibility resources are batteries and hydrogen. These two resources constitute the majority of installed capacity in all scenarios except for the international and European scenarios of BS2050, where the majority of flexibility comes from interconnections and "other" resources. Not all reports include interconnections and other resources in their flexibility scenarios, such as the NVDT, IO-2050, and IO-2050-2 scenarios.

Estimating the levels of flexibility is challenging and highly dependent on ongoing research and governmental policies regarding these resources and future investments. This uncertainty contributes to the wide range of variable results in these scenarios. However, a common theme is that national and regional scenarios project larger flexibility capacities due to a greater emphasis on self-sustainability than regional and international scenarios. An exception is the II3050-2 international scenario, which emphasizes utilizing flexibility on the international market as a valuable resource that the Netherlands can provide to its neighboring countries.

The projections for 2035 also show significant differences, particularly in the installment of battery and hydrogen resources. In the short term, most scenarios do not foresee hydrogen playing a major role in the flexibility system compared to batteries. However, this trend reverses for the 2050 scenarios, where many cases include substantial amounts of hydrogen. The IO-2050-2 report suggests that while batteries will

significantly impact flexibility in the coming years, their total installed capacity will decrease towards 2050, surpassed by hydrogen as a superior alternative. For instance, in the EL & RES scenario, batteries are the primary flexibility resource, whereas the other two scenarios emphasize hydrogen production.

Various flexibility resources are mentioned across different reports, with differing levels of importance depending on the time horizon and political influences. However, it is evident that batteries and hydrogen are the predominant resources in most scenarios. Other flexibility resources, such as interconnections and those labeled "other," also play crucial roles in adapting the power system to its more variable nature.

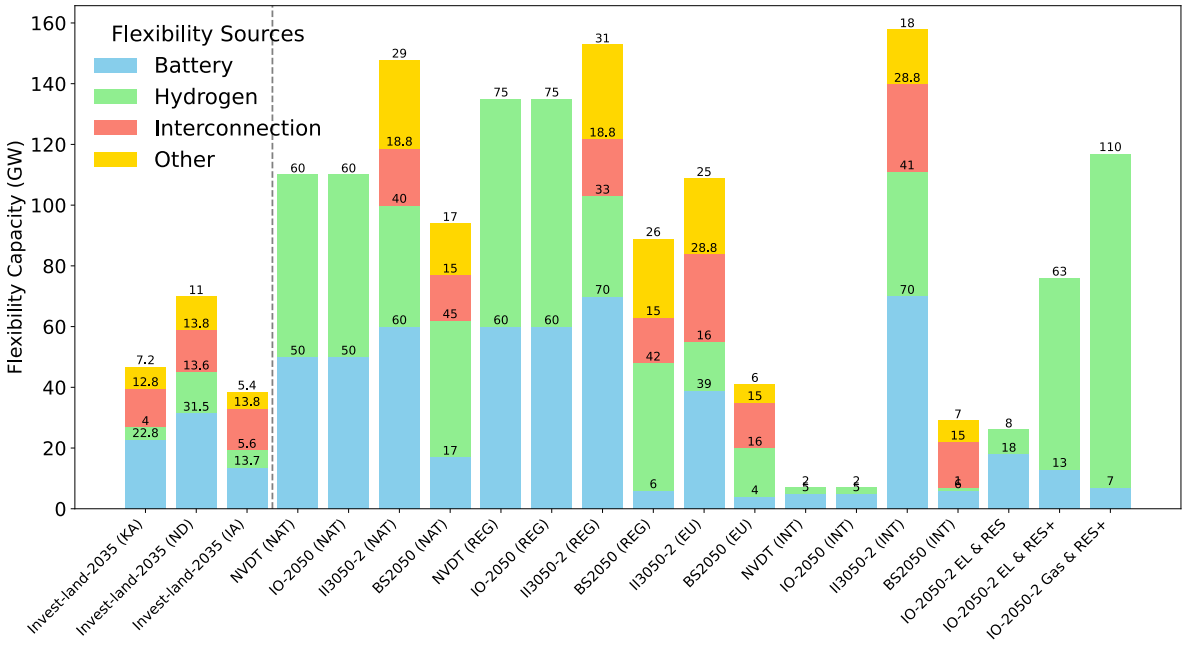


Figure 4.4: Installed Flexibility capacity for different scenarios in 2050 in the Netherlands.

4.3.5. Power Demand

Some of the reports do not provide specific information about the peak power demand of different systems, but analyzing the total power demands can help differentiate the power demand across different scenarios for further analysis.

Another interesting aspect of the different scenarios is the total energy demand per year, as shown in Figure 4.5. While the total energy demand does not provide insights into hourly demand variations, which are crucial for assessing the dynamic stability of the system during operation, it does help explain the wide margins in installed capacities across different scenarios. The graph clearly shows that the II3050-2 scenario has the highest estimated energy demand, followed by BS2050. The NVDT, IO-2050, and IO-2050-2 scenarios project lower energy demands than the first two reports. This highlights the significant differences in projections and how similar projects can yield varying outcomes, emphasizing the uncertainty facing the future power system.

Interestingly, some of the 2035 scenarios have higher energy demands than several 2050 scenarios. This initially counterintuitive observation can be explained by the additional levels of energy efficiency accounted for in the 2050 scenarios compared to the 2035 scenarios. Many industries and processes that are currently highly energy-consuming are expected to become more efficient, thus reducing their overall impact on energy demand. The international and European scenarios generally show lower energy demands than other scenarios within the same report. This is because these scenarios often relocate energy-intensive industries outside national borders to increase efficiency and reduce costs through highly functional international markets.

This wide range in energy demand corresponds to the previous graphs showing installed capacities of different renewable energy generation and flexibility units. Typically, projects with higher installed capacities, such as the II3050-2 and BS2050 projects, also correspond to the highest energy demands in all comparable scenarios.

However, as mentioned, many reports do not provide information about peak power demand, which is crucial for understanding the dynamic stability of the future system. While energy demand helps explain some of the reasoning behind different installed capacities, it is essential to consider various peak and variable demands during different operating scenarios to fully assess dynamic stability. Peak power demand in the Netherlands typically occurs during cold months. Between 2030 and 2050, peak power demand is expected to exceed power generation by 35-50 GW during certain periods, while power generation will exceed power demand by 65-95 GW during others [106]. This highlights the importance of flexibility installations and managing limitations in renewable energy production during specific hours to maintain a balanced power system.

Regarding peak power demand, the first edition of the II3050-2 report estimates a peak power demand of around 40 GW for all four scenarios. The II3050-2 report then further details that the demand in the regional scenario ranges from 16-40 GW, the national scenario from 18-48 GW, the European scenario from 20-46 GW, and the international scenario from 20-46 GW. In these scenarios, the regional and national scenarios predict a sharp increase in peak supply surpluses, reaching levels of 120 GW, while the European and international scenarios predict surpluses of up to 80 GW.

Another source of peak power demand information is the IO2050 report, which estimates a peak power demand of around 38 GW for the regional scenario, 31 GW for the national scenario, and 25 GW for the international scenario.

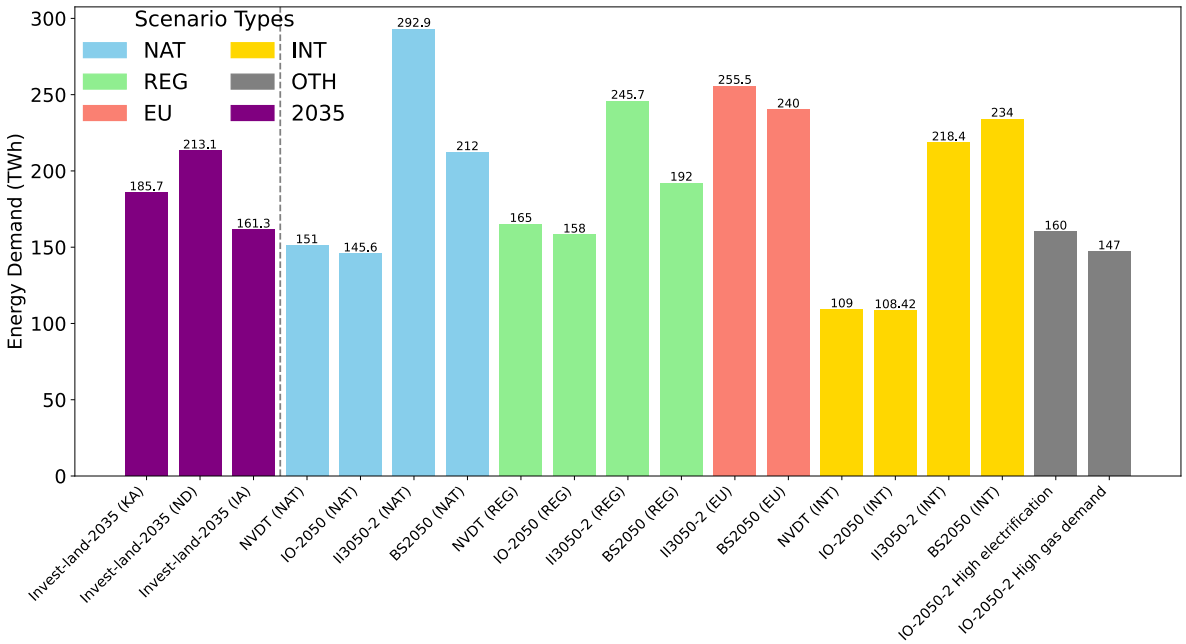


Figure 4.5: Energy Demand for Different Scenarios in 2035 and 2050 in the Netherlands.

The combined capacity of various renewable energy sources and flexibility installations across different scenarios is depicted in Figure 4.6. This figure provides a comprehensive overview of the different scenarios and their respective installations of renewable energies and flexibility resources.

Notably, the national scenarios in II3050-2 and BS2050 exhibit the highest combined installed ca-

pacity of renewables, totaling 265 GW. This is followed by the regional scenario of II3050-2, with 243 GW installed. The variation in total installed generation between different scenarios is significant. For instance, in the national scenarios, the IO-2050 and NVDT scenarios have a combined installation of 101 GW, compared to the 265 GW installed in the II3050-2 and BS2050 scenarios.

This trend is also evident in the regional, European, and international scenarios, where large variations exist between different projections. Such differences highlight the uncertainty and variability in future energy system planning, emphasizing the need for flexible and adaptive strategies to accommodate these potential changes.

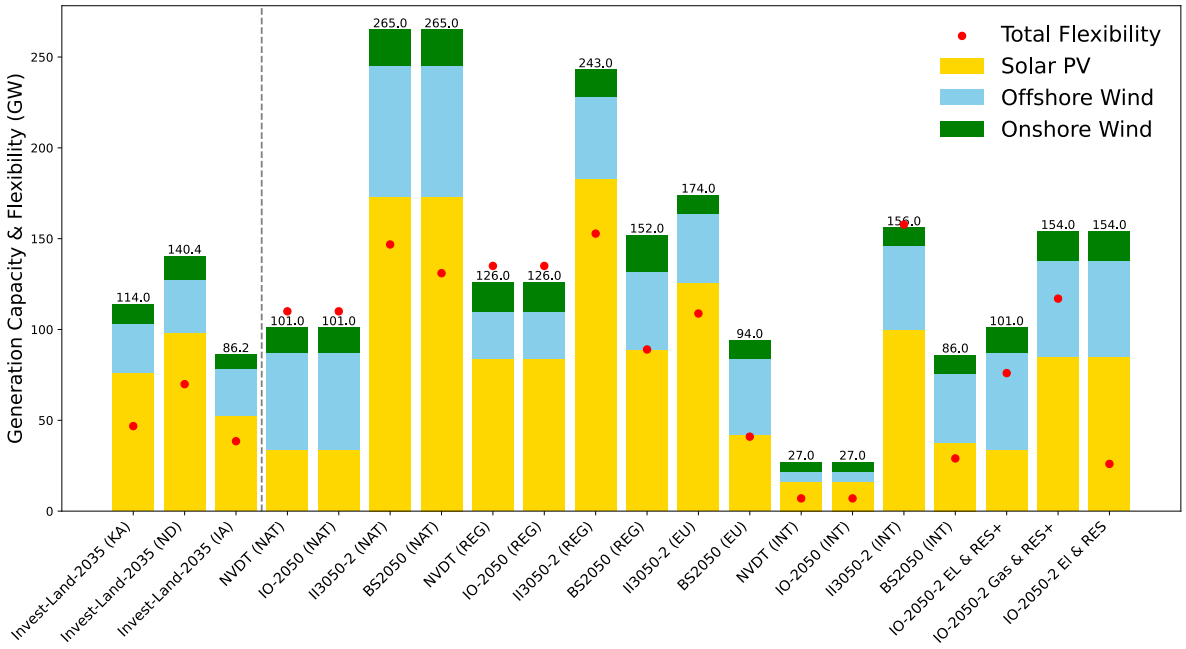


Figure 4.6: Combined generation and flexibility for different scenarios in 2050 in the Netherlands.

4.3.6. Final Estimation of Scenarios

Analyzing different scenarios across various projects reveals significant variation between scenarios and reports. Consequently, the scenarios considered in this study for further work need to account for the most extreme cases. Thus, the national scenario from II3050-2 will be used as the baseline scenario due to its projection of the highest installed capacity. This scenario is also the basis for TenneT's Target Grid project, making it realistic despite the high degree of electrification. It includes a peak power demand of 48 GW, as discussed earlier.

Two additional scenarios were identified for future work to capture the full range of projections and ensure system stability and operation across different projections.

The first of these scenarios is based on the NVDT and IO-2050 international scenarios, which suggest the lowest amount of installed generation and flexibility. Here, the peak demand was found to be around 25 GW.

The third and final scenario is based on a middle line between the two scenarios mentioned that accounts for the highest and lowest installed capacities, largely based on the IO-2050-2 El & RES+ scenario. Here, the demand side was based on the national scenario of the NVDT report, the same as the IO-2050 report. The regional scenario was estimated to have a peak demand of 38 GW.

The three different scenarios and their respective installed renewable capacities, flexibility capacities, and peak power demand can be seen in Figure 4.7.

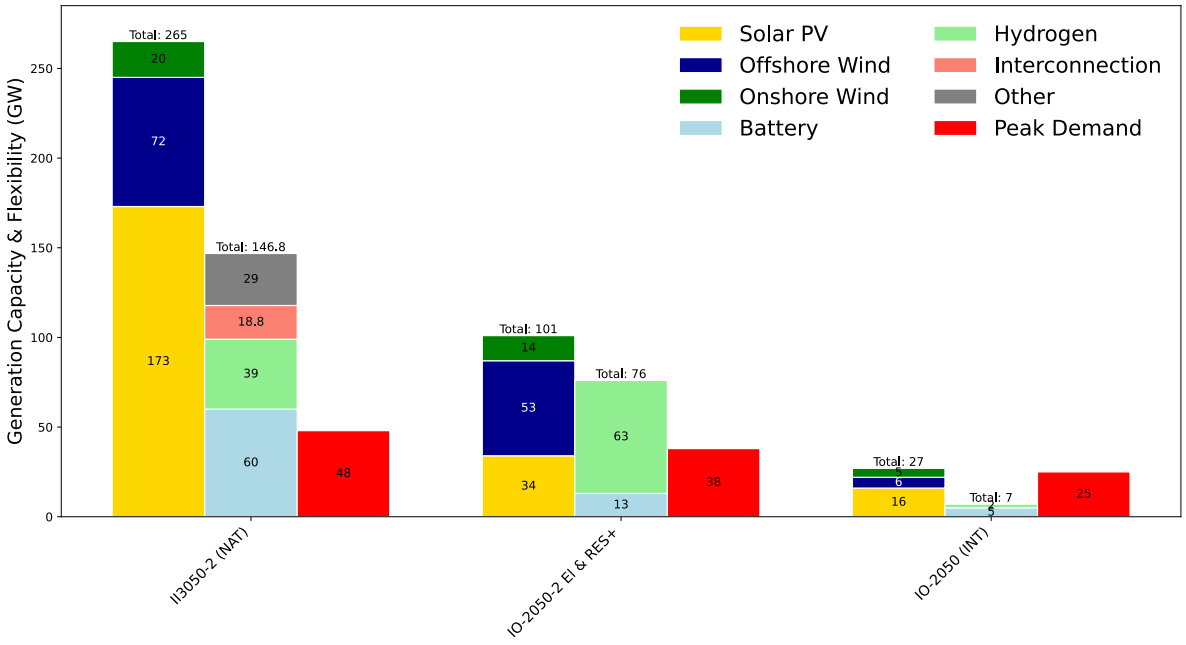


Figure 4.7: Three scenarios of interest including renewable and flexibility capacity as well as peak power demand.

5

Model Development towards 2034 Scenario

5.1. Introduction

This section outlines the necessary steps and modifications to adapt the Synthetic Dynamic Model in PowerFactory to the 2034 scenario estimations, highlighted in TenneT's 10-year investment plans [115, 116].

As given in TenneT's investment plans, the adaptation process is divided into three project phases: realization, design, and study phase projects.

- **Realization Phase**: These projects have been fully studied, designed, and are ready for construction.
- **Design Phase**: These projects have completed the feasibility study, and the design work is ongoing.
- **Study Phase**: These projects are under investigation to assess their impact and feasibility.

While the realization phase projects have detailed plans and information, assumptions are necessary for the design and study phase projects. This approach follows the methodology utilized in reports such as II3050-2 [106], where future scenarios are implemented and simulated despite project uncertainties.

The implementation in PowerFactory uses different network variations to allow for easy modifications based on changes to future investment plans. The sections are categorized by the project completion status and the primary component, which is either added or altered. Substation modifications usually involve alterations to the surrounding transmission lines, which will be referenced in the respective substation sections, while general capacity improvements of transmission lines will be covered in a separate section.

The transmission network in the Netherlands is divided into nine different zones, seen in Figure 5.1. The different projects and investments in the various zones will be implemented here. An overview of the original synthetic model for the 2030 scenarios, highlighting the different voltage levels, can be seen in Figure 5.2. For a more detailed transmission network analysis, refer to [105].

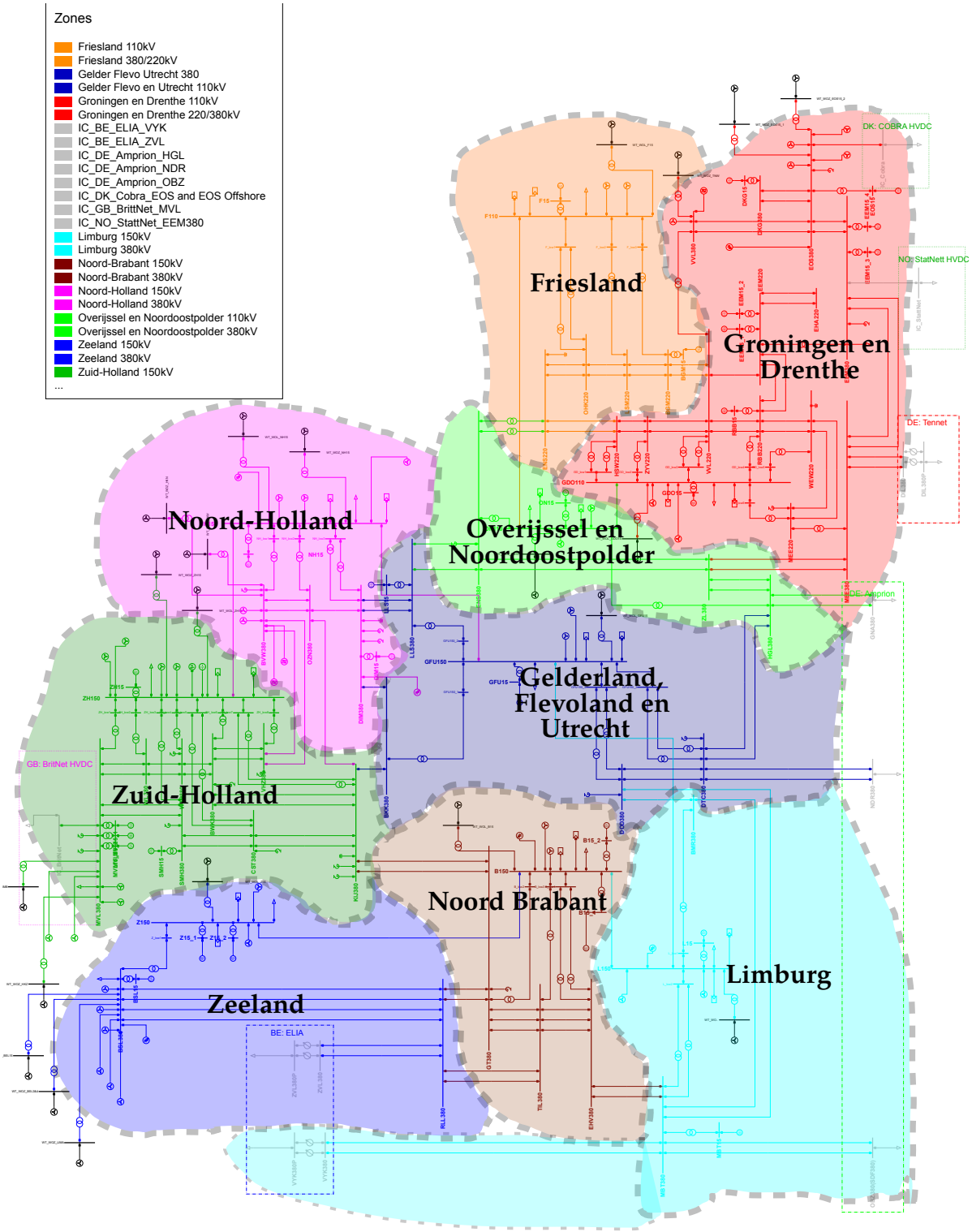


Figure 5.1: Overview of the different zones in the Synthetic model.

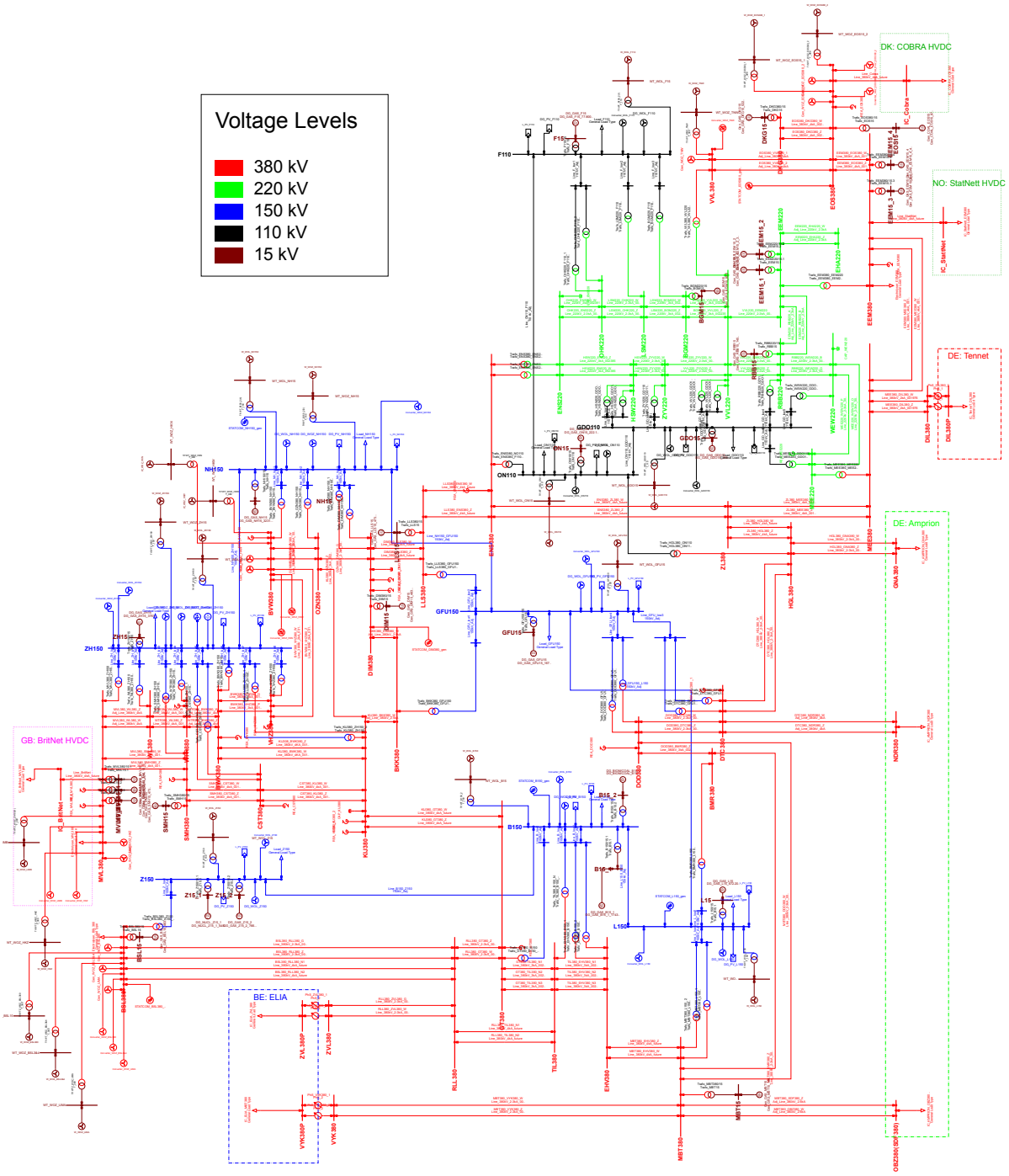


Figure 5.2: Overview of single-line diagram of the Synthetic model.

5.2. Upgrades until 2034 (Realization)

First, the infrastructure upgrades in the realization phase will be discussed. These upgrades are well underway in their planning and integration into the power system and thus contain specific public project details.

5.2.1. Substation Developments

First, the changes to TenneT's 380 kV substations were implemented. These changes are according to Invest-Land-2024 [115], and only changes not already implemented in [105] were implemented. Some of the upgrades in Invest-Land-2024 carry over from previous versions, and thus, some projects that are

still under construction have already been implemented in the PowerFactory model.

The updated investment plan currently includes two new substation projects in the realization phase. These are the substations of Amaliehaven 380 kV (AMH380) and Musselkanaal 380 kV (MSK380) in the regions of Zuid-Holland and Groningen en Drenthe, respectively.

The MSK380 project is part of a larger network reinforcement program by TenneT in the north-east of the Netherlands [117]. This also includes another 380 kV substation in the design phase, which will be covered later. The MSK380 station is located in the town of Musselkanaal and will be connected by the line between Zwolle 380 kV (ZL380) and Meeden 380 kV (MEE380) stations. The new substation will also be connected to the corresponding 110 kV power system in the Musselkanaal area. This is done by a 380/110 kV transformer, using the same specifications as in the connection between ENS380-NO110, with 3 additional parallel transformers due to the higher power flow between the areas. Previously, the geographical representation of the single-line diagram showed a straight line between ZL380 and MEE380, while in practice, this already goes through Musselkanaal without a substation. However, when the lines were upgraded, this was reflected in the geographic representation, and the corresponding line lengths were updated automatically based on the geographical coordinates.

First, the MSK380 project were implemented. The changes made to the system can be seen in Figure 5.3. The corresponding lines removed from the system can be seen in Table 5.1, the removed lines in Table 5.2, and the transformer added in Table 5.3. All figures are fully zoomable to allow for an easier understanding of the location of the new implementations.

Table 5.1: Added transmission lines for the MSK380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
ZL380_MSK380 _Future_1	Line_380kV _4kA_001388R_2	ZL380	MSK380	Overijssel /Groningen	71.841
ZL380_MSK380 _Future_2	Line_380kV _4kA_001388R_2	ZL380	MSK380	Overijssel /Groningen	71.841
MSK380_MEE380 _Future_1	Line_380kV _4kA_001388R_2	MSK380	MEE380	Overijssel /Groningen	23.818
MSK380_MEE380 _Future_2	Line_380kV _4kA_001388R_2	MSK380	MEE380	Overijssel /Groningen	23.818

Table 5.2: Removed transmission lines for the MSK380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
ZL380_MEE380 _W	Line_380kV _4kA_001388R_2	ZL380	MEE380	Overijssel /Groningen	107.85
ZL380_MEE380 _Z	Line_380kV _4kA_001388R_2	ZL380	MEE380	Overijssel /Groningen	107.85

Table 5.3: Added transformer for the MSK380 project

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_MSK380 -GDO110_Future	MSK380	GDO110	Groningen	6	2220	370

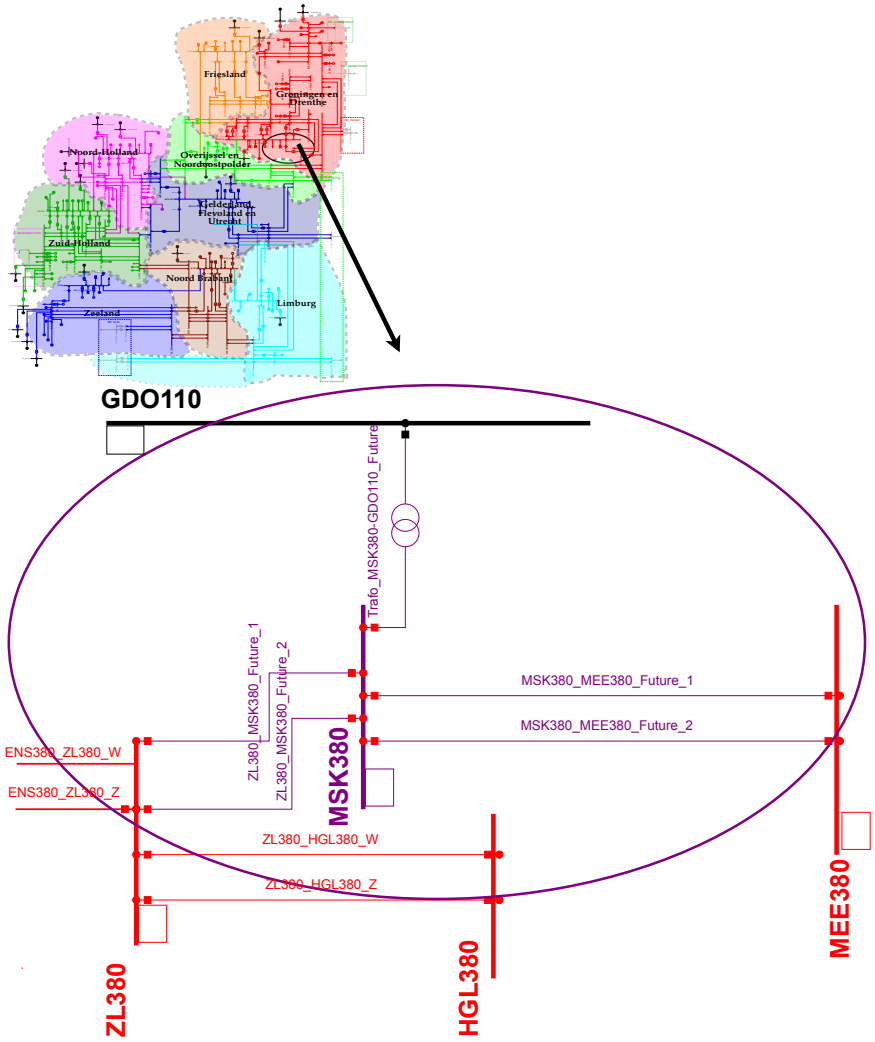


Figure 5.3: Implementation of substation MSK380 and corresponding changes to diagram highlighted in purple.

Next, the AMH380 project was implemented. A substation will be constructed adjacent to the current substation in Maasvlakte port in Rotterdam, the 380 kV Maasvlakte (MVL380) station [118]. Originally, there was a connection between MVL380 and both Simonshaven 380 kV (SVH380) and Westerlee 380 kV (WL380). With the addition of AMH380, the original connection between MVL380 and SVH380 will be moved to AMH380, and a new connection between AMH380 and MVL380 will be established. The connection between MVL380 and WL380 will remain unchanged. As before, the corresponding line lengths due to the change are measured using the geographical feature of PowerFactory.

The changes made to the system can be seen in Figure 5.4, with the added and removed transmission lines given in Table 5.4 and Table 5.5 respectively.

Table 5.4: Added transmission lines for the AMH380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
MVL380_AMH380 _Future_1	Line_380kV _4kA_001539R	MVL380	AMH380	Zuid-Holland	0.085
MVL380_AMH380 _Future_2	Line_380kV _4kA_001539R	MVL380	AMH380	Zuid-Holland	0.085
AMH380_SMH380 _Future_1	Line_380kV _4kA_001539R	AMH380	SMH380	Zuid-Holland	22.357
AMH380_SMH380 _Future_2	Line_380kV _4kA_001539R	AMH380	SMH380	Zuid-Holland	22.357

Table 5.5: Removed transmission lines for the AMH380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
MVL380_SMH380 _W	Line_380kV _4kA_001539R	MVL380	SMH380	Zuid-Holland	26
MVL380_SMH380 _Z	Line_380kV _4kA_001539R	MVL380	SMH380	Zuid-Holland	26

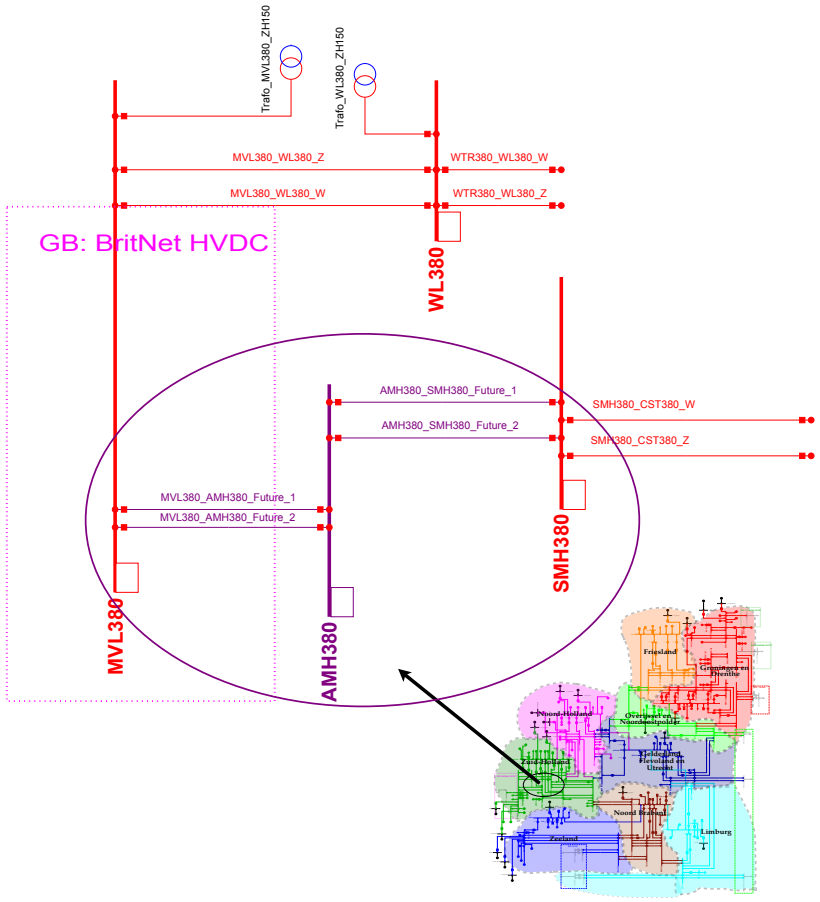


Figure 5.4: Implementation of substation AMH380 and corresponding changes to diagram highlighted in purple.

5.2.2. Wind Farm Integration

The next step involved integrating additional offshore wind farms as outlined in Invest-sea-2024 [116] that were not yet implemented in the PowerFactory model. The implemented wind farms are listed in Table 5.6. While numerous offshore wind projects are still under consideration or in various stages of development, this section solely focuses on those confirmed in Invest-sea-2024 [116].

Noteworthy points include the Hollandse Kust West project, designated as (Converter_WOZ_HKW – 2) in the synthetic model, which has not yet officially determined a connection point. Nevertheless, Beverwijk (BVW380) is mentioned as a viable option and was thus used in this implementation [116, 119]. Similarly, for Nederviek 3 designated as (Converter_WOZ_NDW – 3), the connection point is not definitively determined, reports suggest Geertruidenberg (GT380) as a feasible alternative, and was used here [116, 119]. Lastly, the Ten Noorden van de Waddeneilanden wind farm, designated as (Converter_WOZ_TNW), required a connection point change. This connection was moved from the original location at Vierverlaten (VVL380) to Eemshaven Oudeship (EOS380) [116, 119]. The specific details are provided in Table 5.7.

Table 5.6: Overview of different Offshore Wind farms implemented throughout 2035 for the realization project phase

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Converter _WOZ_HKW-2	TBD	Hollandse Kust West, kavel VIII	BVW380	Noord-Holland	380	700
Converter _WOZ_IJMG	2029	Ijmuiden Ver Beta	MVL380	Zuid-Holland	380	2000
Converter _WOZ_NDW_1	2030	Nederwiek 1	BSL380	Zeeland	380	2000
Converter _WOZ_NDW_2	2030	Nederwiek 2	MVL380	Zuid-Holland	380	2000
Converter _WOZ_NDW_3	2030	Nederwiek 3	GT380	Noord-Brabant	380	2000
Converter _WOZ_DDW_1	2030	Doordewind 1	EOS380	Groningen	380	2000
Converter _WOZ_DDW_2	2030	Doordewind 2	EEM380	Groningen	380	2000

Table 5.7: Overview of Offshore Wind Farm connection change

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Converter _WOZ_TNW-2	2030	Ten Noorden van de Waddeneilanden	EOS380	Groningen	380	700

5.2.3. Transmission Line Developments

Next, the different transmission line developments planned in Invest-Land-2024 [115] will be discussed. These developments typically involve upgrading or removing existing lines. Therefore, the changes will not be discussed in as much detail as the substation scenarios; rather, an overview of the changes will be provided using tables.

All the upgraded transmission lines were upgraded to the future 4 kA type used in PowerFactory. Their lengths were based on the length of the existing lines or the geographical feature in PowerFactory for new lines. The numbers in the name indicate the additional circuit number that is connected. As previously explained for the substation developments, all of the transmission line changes were

implemented using varying network variations in PowerFactory.

The different projects implement varying improvements; some involve upgrading existing lines, while others involve entirely new lines. This is given in different tables.

The lines in the system that were upgraded to the 4 kA line type can be seen in Table 5.8, the additional lines implemented in Table 5.9, and the lines removed in Table 5.10.

Table 5.8: Transmission lines upgraded for the realization project phase

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
GT380_TIL380 _N1	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	30
GT380_TIL380 _N2	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	30
GT380_TIL380 _N3	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	30
TIL380_EHV380 _N1	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	35
TIL380_EHV380 _N2	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	35
TIL380_EHV380 _N3	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	35
BSL380_RLL380 _G	Line_380kV _4kA_future	BSL380	RLL380	Zeeland	38.924
BSL380_RLL380 _W	Line_380kV _4kA_future	BSL380	RLL380	Zeeland	38.924

Table 5.9: Transmission lines added for the realization project phase

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
ENS380_VVL380 _Future_1	Line_380kV _4kA_future	ENS380	VVL380	Overijssel	73.973
ENS380_VVL380 _Future_2	Line_380kV _4kA_future	ENS380	VVL380	Overijssel	73.973

Table 5.10: Transmission lines removed for the realization project phase

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
VVL220_EEM220	Line_220kV _2-3kA_002474R	VVL220	EEM220	Groningen	40.02

5.2.4. Transformer Developments

Next, the transformer upgrades in the realization stage of the investment plan are discussed. These were handled similarly to the transmission line developments, with a table indicating the upgrades and additions made. All transformers were upgraded with one additional parallel transformer unless otherwise stated. The original MVA capacity and number of parallel transformers are highlighted in red. The resulting transformer upgrades can be seen in Table 5.11.

Table 5.11: Upgraded Transformers in the realization project phase (original numbers highlighted in red)

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_BVW380 _NH150	BVW380	NH150	Noord-Holland	3 (2)	1500 (1000)	500
Trafo_MEE220 _GDO110	MEE220	GDO110	Groningen	3 (2)	1125 (750)	375

5.3. Upgrades until 2034 (Design)

Next, the infrastructure upgrades in the design phase will be discussed. As mentioned, these do not have specific plans, and assumptions were made to implement them. These upgrades do not contain information regarding power plants and load centers but rather focus on infrastructure developments in the 380/220 kV power system.

5.3.1. Substation Developments

First, the additional substation developments will be discussed. Eight substations are in the basic design phase.

The first is the new Veenoord-Boerdijk 380 kV station (VOB380). This is part of the larger network reinforcement project in Northeast Netherlands, which included the new substation of MSK380 earlier implemented. The new substation will be located north of the A37 at the Veenoord exit (Boerdijk, municipality of Emmen on the border of the municipality of Coevorden) [117]. This new substation was implemented similarly to the MSK380 substation, rerouting lines in the same manner. The same procedure was followed regarding installing new lines and removing old lines using the geographical feature in PowerFactory. The changes made for the project can be seen in Figure 5.5, with the corresponding lines added and removed in Table 5.12 and Table 5.13, respectively.

Table 5.12: Added transmission lines for the VOB380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
ZL380_VOB380 _Future_1	Line_380kV _4kA_001388R_2	ZL380	VOB380	Overijssel /Groningen	49.71
ZL380_VOB380 _Future_2	Line_380kV _4kA_001388R_2	ZL380	VOB380	Overijssel /Groningen	49.71
VOB380_MSK380 _Future_1	Line_380kV _4kA_001388R_2	VOB380	MSK380	Overijssel	25.541
VOB380_MSK380 _Future_2	Line_380kV _4kA_001388R_2	VOB380	MSK380	Overijssel	25.541

Table 5.13: Removed transmission lines for the VOB380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
ZL380_MSK380 _Future_1	Line_380kV _4kA_001388R_2	ZL380	MSK380	Overijssel	71.841
ZL380_MSK380 _Future_2	Line_380kV _4kA_001388R_2	ZL380	MSK380	Overijssel	71.841

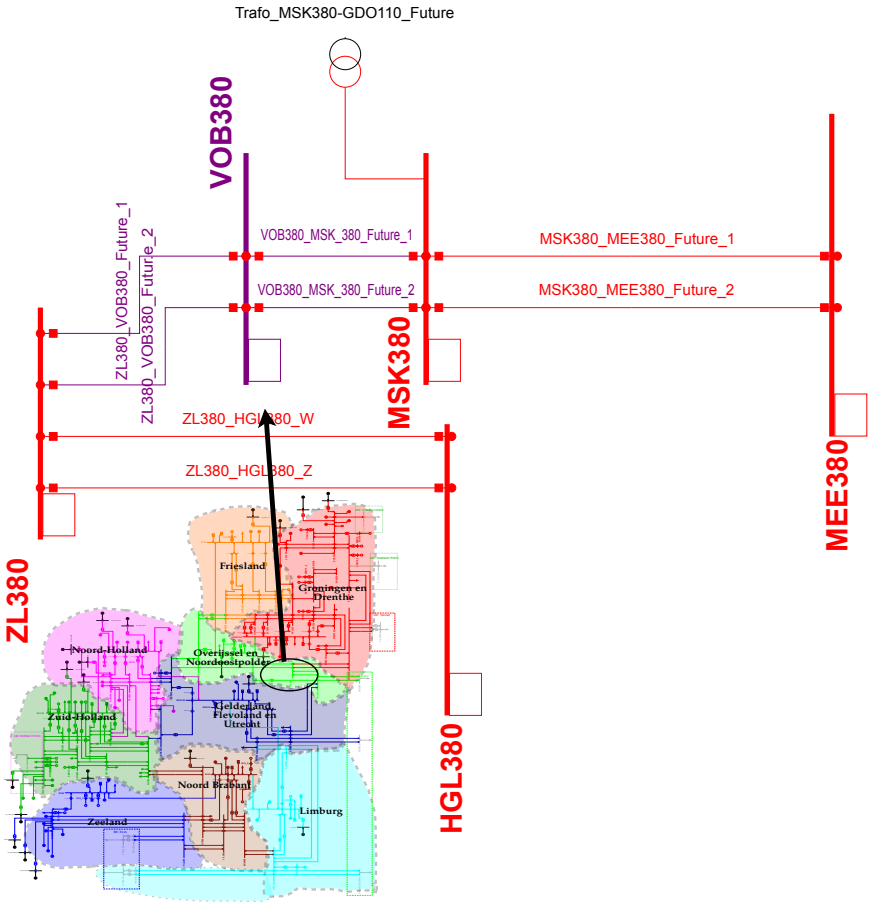


Figure 5.5: Implementation of substation VOB380 and corresponding changes to diagram highlighted in purple.

Next, the substation Farmsum Oosterlaan 220 kV (FSO220) was implemented. Very limited details regarding the project have been published as of now. However, one of the reasons for the extra substation is due to the projected electrification of industry in the area and the increased amount of renewable energy [120]. It is indicated that the new substation will be in the same area as the previous substation, Weiwerd 220 kV (WEW220) in Farmsum. Thus, the new substation was placed in Farmsum and connected to the existing line between WEW220 and Meeden 220 kV (MEE220). As before, this was done using the geographical function in PowerFactory, providing accurate estimates regarding placement and corresponding line lengths. The changes made for the project in PowerFactory can be seen in Figure 5.6, with the added and removed lines in Table 5.14 and Table 5.15, respectively.

Table 5.14: Added transmission lines for the FSO220 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
MEE220_FSO220 _Future_1	Line_220kV _2-3kA_00166R	MEE220	FSO220	Groningen	19.996
MEE220_FSO220 _Future_2	Line_220kV _2-3kA_00166R	MEE220	FSO220	Groningen	19.996
FSO220_WEW220 _Future_1	Line_220kV _2-3kA_00166R	FSO220	WEW220	Groningen	0.826
FSO220_WEW220 _Future_2	Line_220kV _2-3kA_00166R	FSO220	WEW220	Groningen	0.826

Table 5.15: Removed transmission lines for the FSO220 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
MEE220_WEW220_O	Line_220kV_2-3kA_00166R	MEE220	WEW220	Groningen	29.6
MEE220_WEW220_P	Line_220kV_2-3kA_00166R	MEE220	WEW220	Groningen	29.6

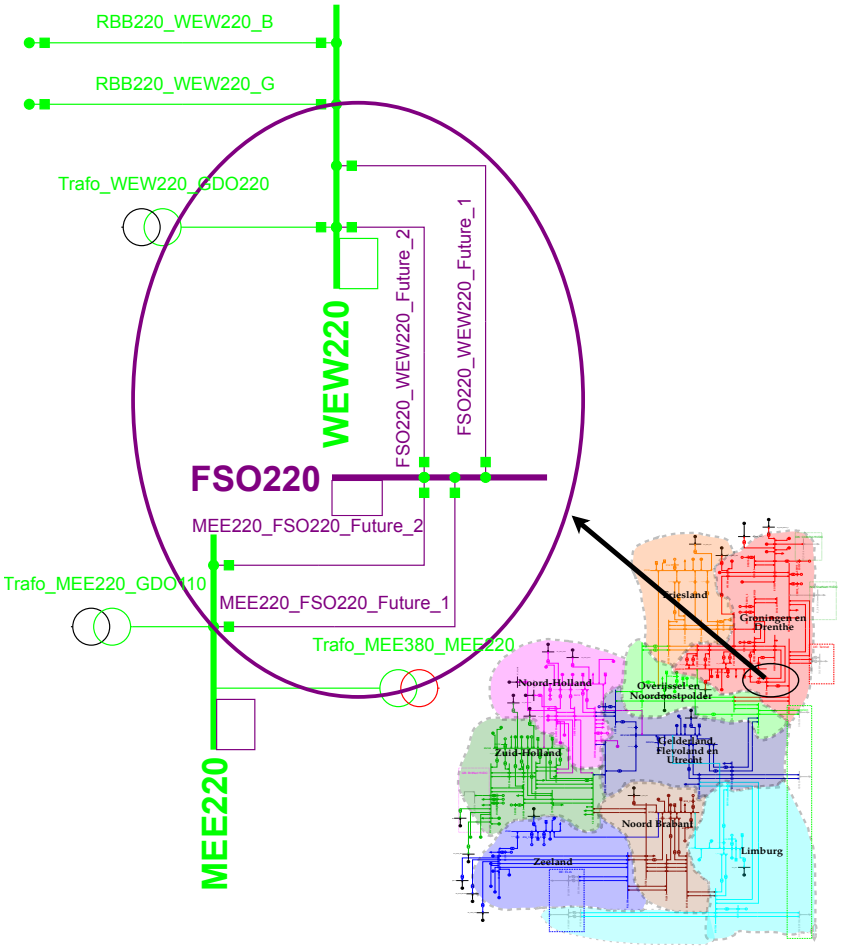


Figure 5.6: Implementation of substation FSO220 and corresponding changes to diagram highlighted in purple.

Next, the new station Graetheide 380 kV (GRTH380) and an upgrade to the 380 kV transmission line between Graetheide and Maasbracht, which currently has a 150 kV connection, were implemented. The PowerFactory model includes sections of the 110 and 150 kV power systems modeled as single substations to reduce computation. Consequently, as here, the connections between the 380 kV and these systems will not always be properly representative.

As mentioned, the 150 kV connection between Graetheide and Maasbracht will be upgraded to 380 kV. However, there is no current 150 kV connection between these in the model, but rather a connection between Maasbracht 380 kV (MBT380) and the 150 kV substation representing the 150 kV power system of the Limburg region (L150). A new substation (GRTH380) was created in the model and connected to MBT380 with two circuits. This was done using the future 4 kA transmission line type from previous projects due to the lack of specific public information regarding this project. GRTH380 was also connected to the L150 station through three parallel 500 MVA transformers using a standard type. This was then connected through a 50 km transmission line with nine parallel lines, containing

one circuit using the adjusted line type from previous projects. The transmission line, transformer options, and line length were chosen based on the previous representations between the 380 kV and 150 kV power systems. As before, the line lengths between the 380 kV stations were determined using the geographical map through PowerFactory.

The changes made to the system in PowerFactory can be seen in Figure 5.7, with the added transmission line and transformer in Table 5.16 and Table 5.17, respectively.

Table 5.16: Added transmission lines for the GRTH380 project

Name	**Type in PowerFactory**	**Terminal I**	**Terminal J**	**Zone**	**Length (km)**
MBT380_GRTH380 _Future_1	Line_380kV _4kA_Future	MBT380	GRTH380	Limburg	16.315
MBT380_GRTH380 _Future_2	Line_380kV _4kA_Future	MBT380	GRTH380	Limburg	16.315
Line_L_low3_a	150kV_Adj	L_low3	L150	Limburg	50

Table 5.17: Added transformer for the GRTH380 project

Name	**HV Terminal**	**LV Terminal**	**Zone**	**Parallel Trafos**	**S_total (MVA)**	**S_rated (MVA)**
Trafo_GRTH380 _L150_Future	GRTH380	L150	Limburg	3	1500	500

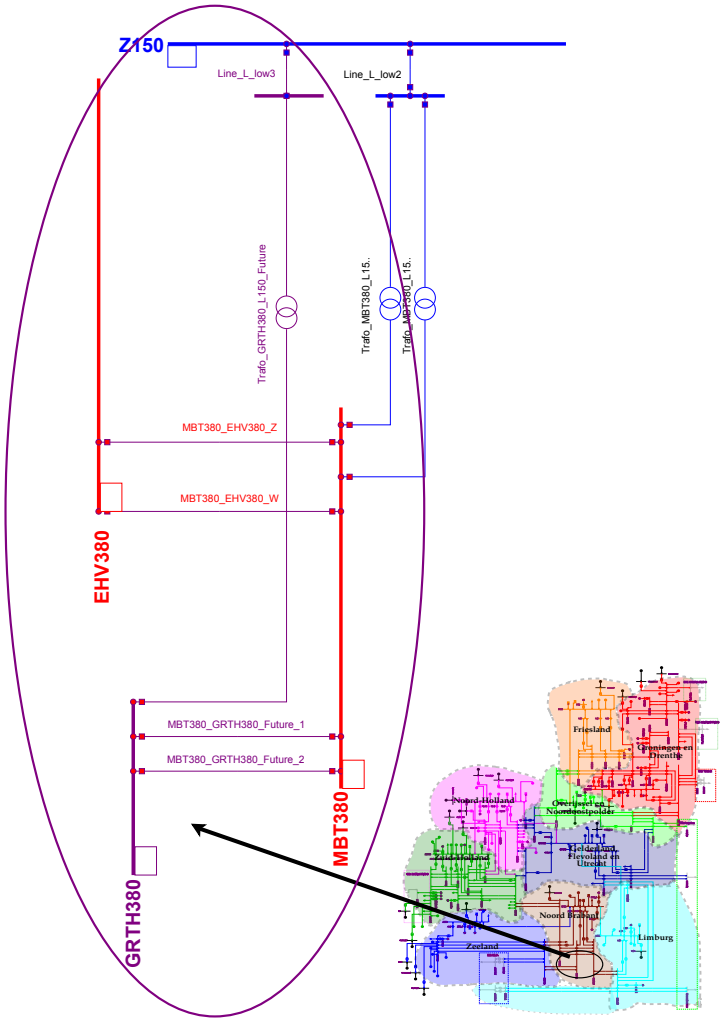


Figure 5.7: Implementation of substation GRTH380 and corresponding changes to diagram highlighted in purple.

Next, the new 380 kV substation project at Terneuzen (TNZ380) will be discussed. Here, the placement of the substation has not yet been fully decided, but two areas have been designated for further study. The same goes for the transmission connection to the 380 kV station. Two alternatives are being considered, with a direct connection to the existing 380 kV line between Borssele (BSL380) and Rilland (RLL380) [121, 122]. The substation will also be connected to the surrounding 150 kV power system, and a demand increase of up to six-fold is expected in the area until 2050 [123]. Because of this, the current 150 kV power system in the area is insufficient to keep up with the change in demand, and reinforcement is necessary. The region is also being studied regarding future interconnection towards Belgium [122].

Because neither the exact placement of the substation nor the connection path to the 380 kV line between Borssele (BSL380) and Rilland (RLL380) nor the transmission line path is decided, assumptions were made. Here, a substation was placed in the center of Terneuzen for simplicity's sake. The transmission connection was placed approximately in the middle of RLL380 and BSL380, and the crossing was made straight to the substation. A four-circuit connection is expected, and it was assumed that the 4 kA transmission lines used for future scenarios in previous expansions will be used.

The connection between the 380 kV station and the 150 kV power system followed the same procedure as in the previous case, with the same transformer type, quantity, and corresponding line and length.

The changes made to the system in PowerFactory can be seen in Figure 5.8, with the added transmission line and transformer in Table 5.18 and Table 5.19. The CP stands for connection point on the existing

transmission line.

Table 5.18: Added transmission lines for the TNZ380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
Line_L_low3_a	150kV_Adj	L_low3	L150	Limburg	50
BSL380_TNZ380 _Future_1	Line_380kV _4kA_future	CP	TNZ380	Zeeland	11
BSL380_TNZ380 _Future_2	Line_380kV _4kA_future	CP	TNZ380	Zeeland	11
BSL380_TNZ380 _Future_3	Line_380kV _4kA_future	CP	TNZ380	Zeeland	11
BSL380_TNZ380 _Future_4	Line_380kV _4kA_future	CP	TNZ380	Zeeland	11
Line_Z_low2	150kV_Adj	Z_low2	TNZ380	Zeeland	50

Table 5.19: Added transformer for the TNZ380 project

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_TNZ380 _Z150_Future	TNZ380	Z150	Zeeland	3	1500	500

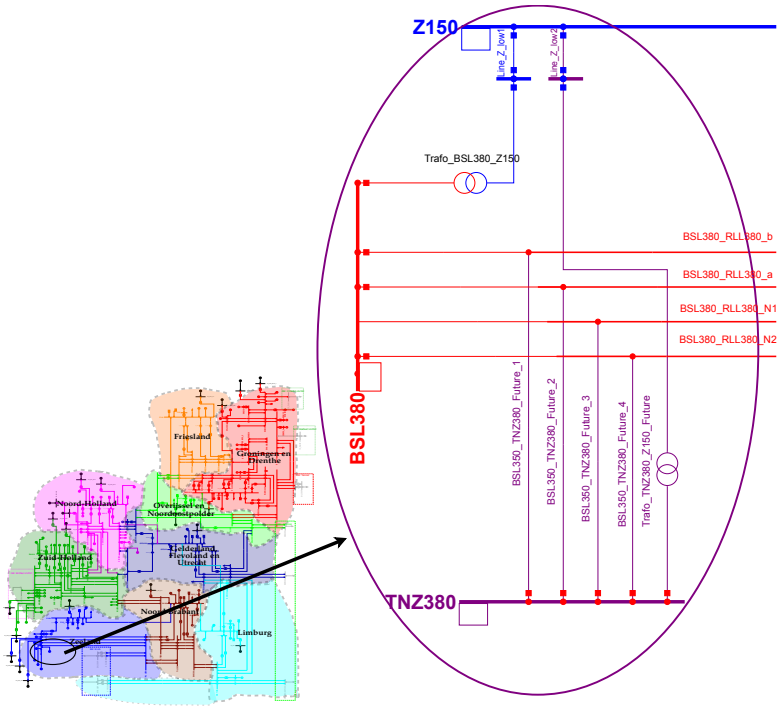


Figure 5.8: Implementation of substation TNZ380 and corresponding changes to diagram highlighted in purple.

Next, the new 380 kV substation project at Haven Vlissingen (HVL380) will be discussed. This substation is connected adjacent to the 380 kV station of Borssele (BSL380) as a result of the increased power demand and generation in the area. This is the same area as the previously discussed TNZ380, and the same reasons apply here [124, 125, 126]. However, this new substation also helps alleviate the increased offshore wind generation in the area. The capacity at Borssele has already been capped, necessitating

additional capacity [124, 125, 126].

As before, a four-circuit connection was utilized between the HVL380 and BSL380 stations, using the 4 kA line type designated for future scenarios. The length of the lines was determined using the geographical function in PowerFactory.

The resulting schematic in PowerFactory can be seen in Figure 5.9 with the corresponding transmission lines added in Table 5.20.

Table 5.20: Added transmission lines for the HVL380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
BSL380_HVL380 _Future_1	Line_380kV _4kA_future	BSL380	HVL380	Zeeland	0.364
BSL380_HVL380 _Future_2	Line_380kV _4kA_future	BSL380	HVL380	Zeeland	0.364
BSL380_HVL380 _Future_3	Line_380kV _4kA_future	BSL380	HVL380	Zeeland	0.364
BSL380_HVL380 _Future_4	Line_380kV _4kA_future	BSL380	HVL380	Zeeland	0.364

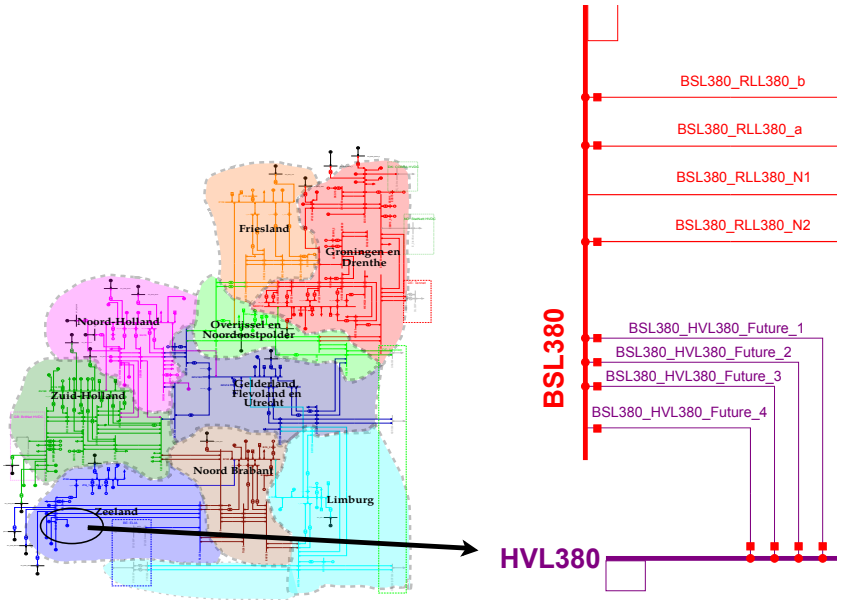


Figure 5.9: Implementation of substation HVL380 and corresponding changes to diagram highlighted in purple.

Next, the project involving a new substation Halsteren 380 kV (HST380) will be discussed. The new substation will be connected to the existing line between Rilland (RLL380) and Geertruidenberg (GTB380) [127]. The connection is part of a larger upgrade project in the Halsteren area [128]. The substation will also connect to the local 150 kV network. This will help grow renewable energy and increase electrification for the region, allowing the region of Schouwen-Duiveland to be connected to the 380 kV power system.

The substation was connected to the RLL380 - GTB380 connection and was implemented using a direct connection to the transmission line, as was the case for the project of TNZ380. As before, the connection was done according to the geographical function in PowerFactory to allow for accurate estimations of line lengths. Four circuits will be connected in pairs, with a small distance between two connection points [129]. However, the connection between RLL380-GTB380 is yet to be upgraded to

four circuits. Thus, the two in use now were utilized and upgraded later with the specific transmission line upgrades. The transmission line used the 4 kA type designated for future lines. The connection point is closest to RLL380; thus, this was used as the reference bus for the terminal connection in the line name to simplify identifying the connection.

A transformer was also connected to the existing 150 kV power system of Noord-Brabant (B150). As before, a 50 km line connection was used to connect the transformer to the 150 kV power system. The same transformer and line type were used. The transformer has three parallel transformers, and the transmission line has four circuits.

The resulting schematic in PowerFactory can be seen in Figure 5.10 with the corresponding transmission lines and transformer added in Table 5.21 and Table 5.22, respectively.

Table 5.21: Added transmission lines for the HST380 project

Name	**Type in PowerFactory**	**Terminal I**	**Terminal J**	**Zone**	**Length (km)**
RLL380_HST380 _Future_1	Line_380kV _4kA_future	CP	HST380	Zeeland	10.527
RLL380_HST380 _Future_2	Line_380kV _4kA_future	CP	HST380	Zeeland	10.527
Line_B_low3	150kV_Adj	Z_low3	B150	Noord-Brabant	50

Table 5.22: Added transformer for the HST380 project

Name	**HV Terminal**	**LV Terminal**	**Zone**	**Parallel number of Trafos**	**S_total (MVA)**	**S_rated (MVA)**
Trafo_TNZ380 _Z150_Future	TNZ380	Z150	Zeeland	3	1500	500

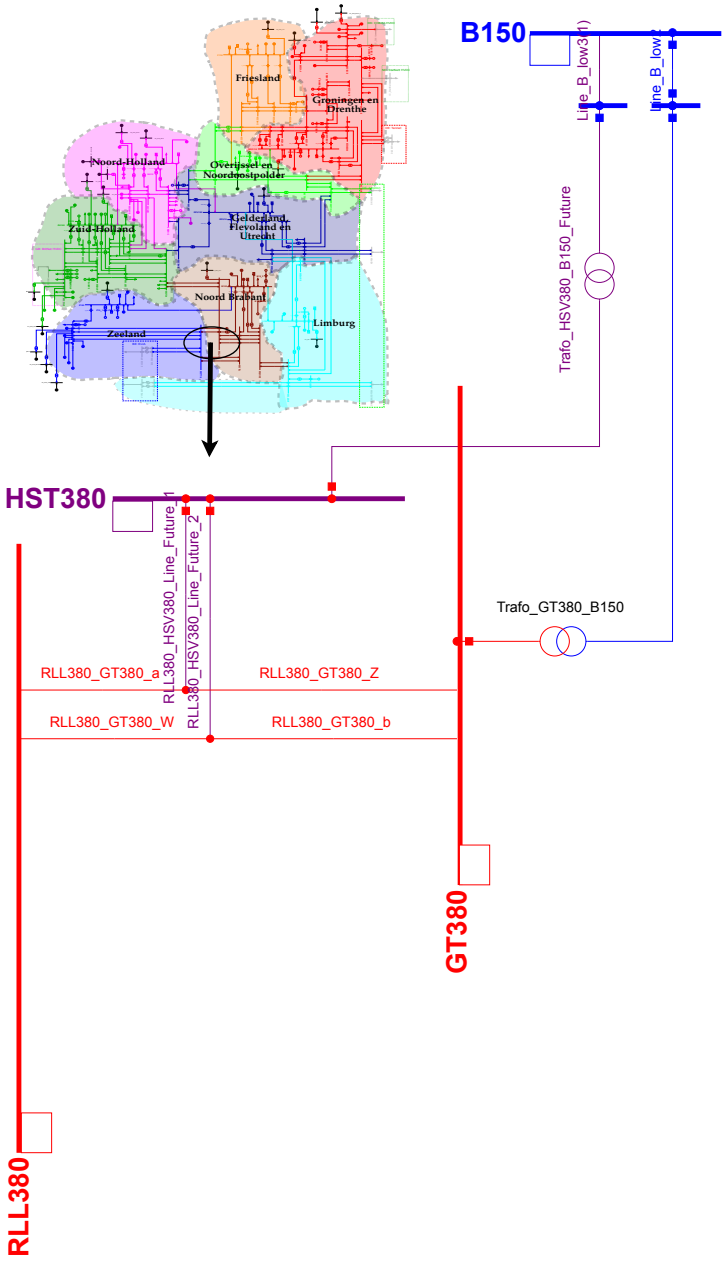


Figure 5.10: Implementation of substation HST380 and corresponding changes to diagram highlighted in purple.

Next, the new 380 kV substation in Oostpolder (OOP380) will be discussed. This is part of the larger project in the Eemshaven area to accommodate the significant increase in renewable generation and interconnections. Due to this, more substations, connection points, transmission lines, and overall infrastructural developments are necessary. The 380 kV substation will be placed adjacent to the existing 110 kV substation with the same name [120]. No detailed information is available regarding the connection to the 380 kV substation. Therefore, it was assumed that the connection will be made to the nearest connection point in Eemshaven Oudeship (EOS380), which has already received upgrades. Due to the lack of information, there will not be a connection to the 110 kV power system in this scenario.

A four-circuit transmission line was assumed to connect OOP380 and EOS380 to allow for further expansions in the area in future scenarios, utilizing the 4 kA future line type. This new substation aids in integrating the large amounts of renewable energy planned in future scenarios.

The implementation of the project in PowerFactory can be seen in Figure 5.11, with the corresponding

lines added in Table 5.23.

Table 5.23: Added transmission lines for the OOP380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
EOS380_OOP380 _Line_Future_1	Line_380kV _4kA_future	EOS380	OOP380	Groningen	4.266
EOS380_OOP380 _Line_Future_2	Line_380kV _4kA_future	EOS380	OOP380	Groningen	4.266
EOS380_OOP380 _Line_Future_3	Line_380kV _4kA_future	EOS380	OOP380	Groningen	4.266
EOS380_OOP380 _Line_Future_4	Line_380kV _4kA_future	EOS380	OOP380	Groningen	4.266

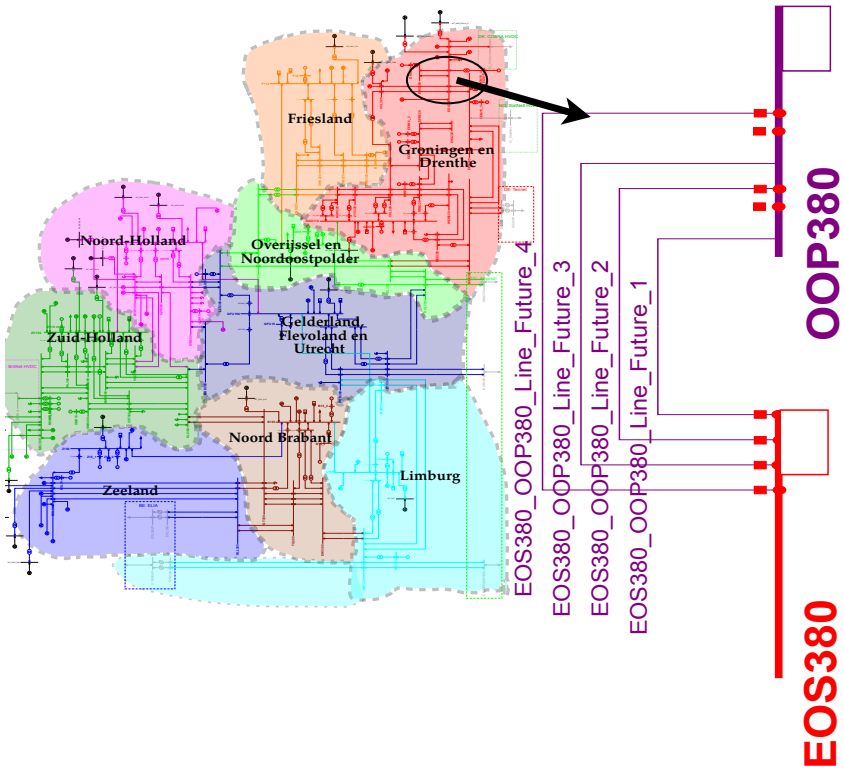


Figure 5.11: Implementation of substation OOP380 and corresponding changes to diagram highlighted in purple.

This concludes all the substation developments planned in Invest-Land-2024 [115] that are under the basic design building stage. All the changes were made using a network variation in PowerFactory to allow for easy removal of variations due to future uncertainties.

5.3.2. Transmission Line Developments

Next, the transmission line developments in the design phase were implemented. A similar procedure as for the realization phase developments was followed, with similar assumptions. The lines added for this phase are shown in Table 5.24, and the lines upgraded are shown in Table 5.25.

Table 5.24: PowerFactory Transmission Lines Added for the 2034 Basic Design project phase

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
DIM380_LLS380 _Future_3	Line_380kV _4kA_future	DIM380	LLS380	Utrecht	50.43
DIM380_LLS380 _Future_4	Line_380kV _4kA_future	DIM380	LLS380	Utrecht	50.43
LLS380_ENS380 _Future_3	Line_380kV _4kA_future	LLS380	ENS380	Overijssel	21
LLS380_ENS380 _Future_4	Line_380kV _4kA_future	LLS380	ENS380	Overijssel	21
KIJ380_GT380 _Future_3	Line_380kV _4kA_future	KIJ380	GT380	Noord-Brabant	34.02
ENS380_VVL380 _Future_1	Line_380kV _4kA_future	ENS380	VVL380	Overijssel	73.973
ENS380_VVL380 _Future_2	Line_380kV _4kA_future	ENS380	VVL380	Overijssel	73.973
MBT380_EHV380 _Future_3	Line_380kV _4kA_future	MBT380	EHV380	Noord-Brabant	48.742

Table 5.25: PowerFactory Transmission Lines Improved for the 2034 Basic Design project phase

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
BSL380_RLL380 _G	Line_380kV _4kA_future	BSL380	RLL380	Zeeland	26.6
BSL380_RLL380 _Z	Line_380kV _4kA_future	BSL380	RLL380	Zeeland	26.6

5.3.3. Transformer Developments

Next, the transformer upgrades in the basic design stage of the Investment plan are given. This was done in a similar manner as for the realization phase, and the resulting transformer upgrades can be seen in Table 5.26.

Table 5.26: Upgraded Transformers in the basic design project phase

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_MEE380 _MEE220	MEE380	MEE220	Groningen	2 (1)	1500 (750)	750
Trafo_GT380 _B150	GT380	B150	Noord-Brabant	3 (2)	1350 (900)	450
Trafo_SMH380 _ZH150	SMH380	ZH150	Zuid-Holland	2 (1)	100 (500)	500
Trafo_VVL380 _VVL220	VVL380	VVL220	Groningen	10 (7)	7500 (5250)	750

Table 5.27: New Transformer Added in the basic design project phase

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_ENS380 _ENS220_3	ENS380	ENS220	Overijssel	1	750	750

5.4. Upgrades until 2034 (Study)

Next, the infrastructure updates in the study phase of the investment plan will be discussed. Due to the early project phase, public information about the projects is limited. This results in larger assumptions necessary to integrate them into the power system. A new network variation in PowerFactory was created, similar to the implementation of the design phase.

5.4.1. Substation Developments

The first project is the new 380 kV substation in the port of Moerdijk. The substation will have connections to the existing 150 kV system in the area. This substation will facilitate landings/connections of offshore wind and support the electrification of the area [130, 131].

The project lacks public information regarding connection points, substation placements, and overall system infrastructure, making assumptions and simplifications necessary. The project description mentions two connections: one to the existing connection between Rilland (RLL380) and Geertruidenberg (GT380) and another between RLL380 and Tilburg (TIL380) [130]. An assumption was made that the double connection to each set of transmission lines occurs at the midpoint; this should be updated once the proper location is determined. The substation was connected directly to the line, and the 4 kA-rated future line model used for previous projects was also utilized here. The line connections were separated to provide a similar connection type observed in previous projects with similar characteristics.

The changes made in PowerFactory can be seen in Figure 5.12 with the corresponding additional lines implemented in Table 5.28.

Table 5.28: Added transmission lines for the MDK380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
GT380_MDK380 _Line_Future_1	Line_380kV _4kA_future	GT380	MDK380	Zeeland	9.926
GT380_MDK380 _Line_Future_2	Line_380kV _4kA_future	GT380	MDK380	Zeeland	9.926
TIL380_MDK380 _Line_Future_1	Line_380kV _4kA_future	TIL380	MDK380	Zeeland	20.580
TIL380_MDK380 _Line_Future_2	Line_380kV _4kA_future	TIL380	MDK380	Zeeland	20.371

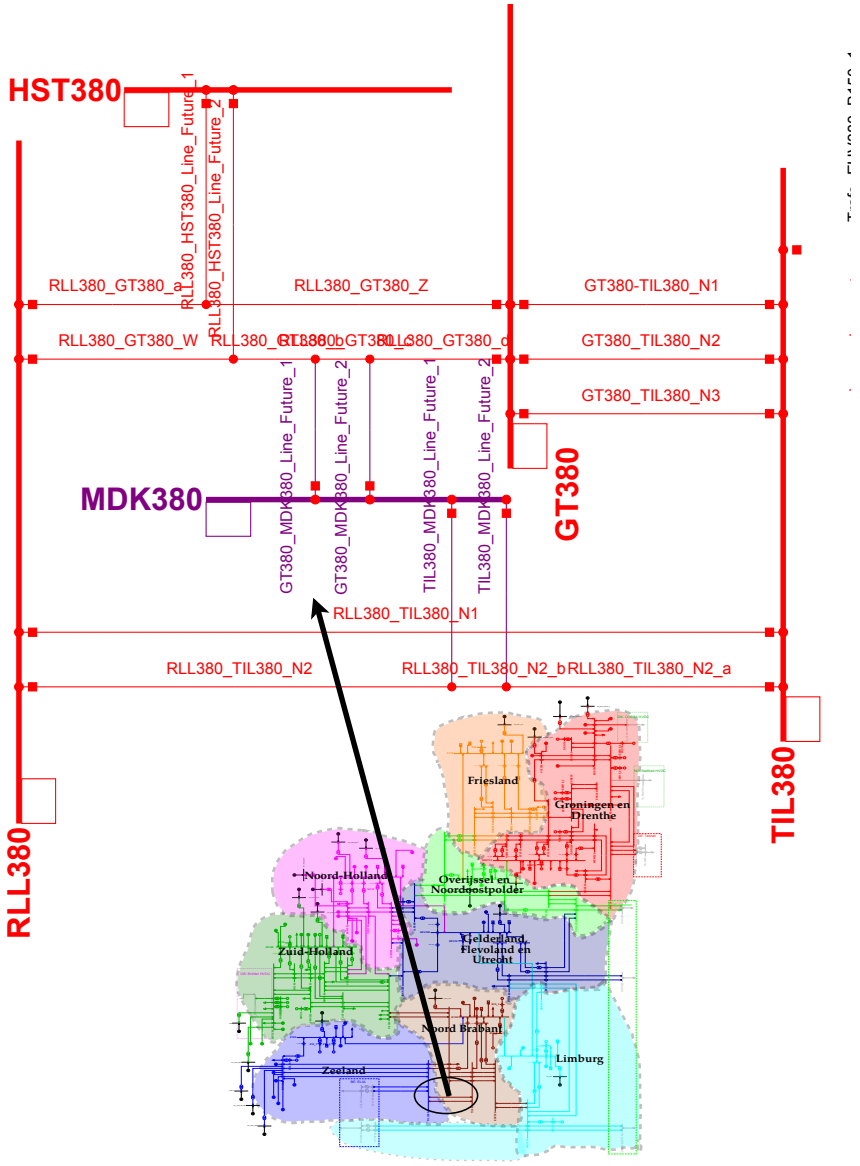


Figure 5.12: Implementation of substation MDK380 and corresponding changes to diagram highlighted in purple.

Next, the new 380 kV substation project at Europoort (ERP380) will be discussed. This is part of the larger infrastructural development occurring in the port of Rotterdam, including the aforementioned projects, such as the new 380 kV substation at Amaliehaven. The connection will allow for more landing connections of offshore wind and overall help with the region's electrification in terms of demand and supply for different scenarios [132].

The new substation was placed in Europoort in Rotterdam, between the existing substations of Maasvlakte (MVL380) and Westerlee (WL380). The substation was placed as a connection point between the two existing substations. The transmission lines already installed between MVL380 and WL380 were utilized, and new line lengths were determined using the geographical function in PowerFactory.

The changes made in PowerFactory can be seen in Figure 5.13 with the corresponding additional lines implemented in Table 5.29.

Table 5.29: Added transmission lines for the ERP380 project

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
MVL380_ERP380_Future_1	Adj_Line_380kV_4kA	MVL380	ERP380	Zuid-Holland	8.446
MVL380_ERP380_Future_2	Adj_Line_380kV_4kA	MVL380	ERP380	Zuid-Holland	8.446
WL380_ERP380_Future_1	Adj_Line_380kV_4kA	WL380	ERP380	Zuid-Holland	5.828
WL380_ERP380_Future_2	Adj_Line_380kV_4kA	WL380	ERP380	Zuid-Holland	5.828

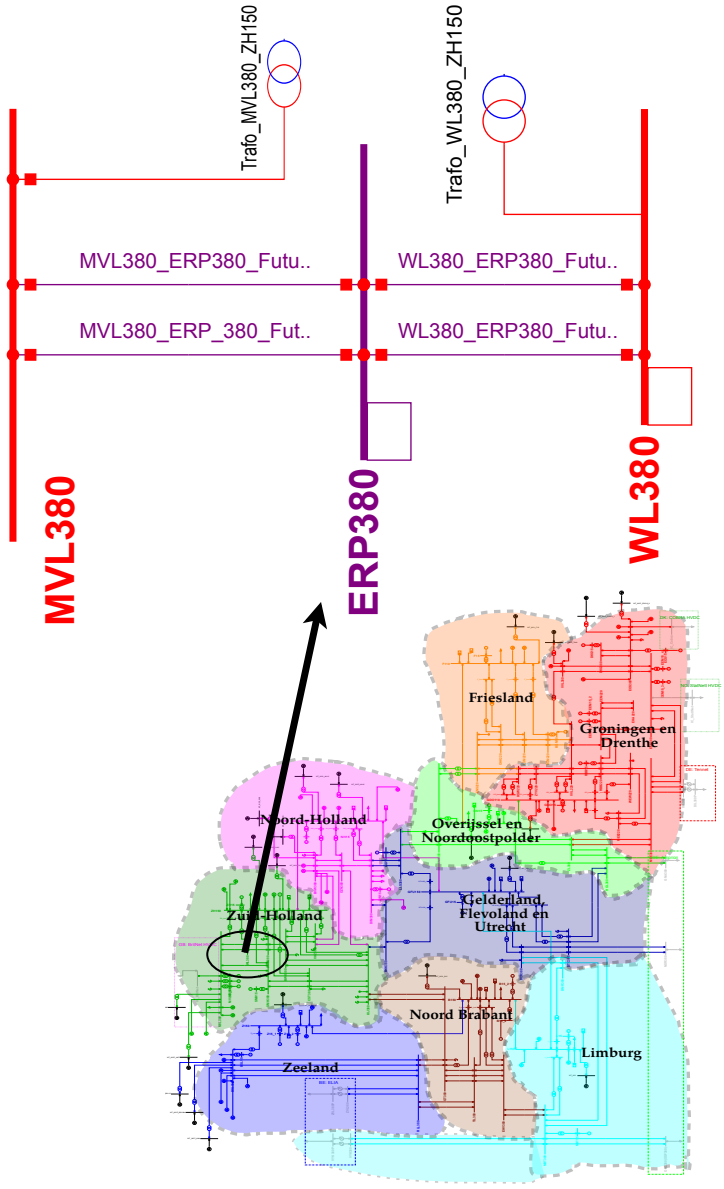


Figure 5.13: Implementation of substation ERP380 and corresponding changes to diagram highlighted in purple.

Next, the new 380 kV substation project in Almere (ALR380) will be discussed. This project is an

extension of existing plans regarding an upgrade in the transmission line infrastructure between Diemen (DIM380) and ENS (ENS380) [133]. This will be discussed later, and the corresponding infrastructure changes will be updated accordingly.

The region has a growing amount of renewable generation, particularly from large solar farms in the area [134]. The new substation will better prepare the region for the increasing need for high-voltage infrastructure.

Due to the project's study phase, there is a lack of public information regarding the development, necessitating assumptions. Here, it was assumed that the new substation will be placed between the two substations of DIM380 and Lelystad (LLS380). Similar to previous substation implementations, it was connected using the existing line and serve as a connection point between DIM380 and LLS380. As before, the line lengths and placement of the substation was determined using the geographical function of PowerFactory.

The changes made in PowerFactory can be seen in Figure 5.14 with the corresponding additional lines implemented in Table 5.30.

Table 5.30: Added transmission lines for the ALR380 project

Name	**Type in PowerFactory**	**Terminal I**	**Terminal J**	**Zone**	**Length (km)**
DIM380_ALR380 _Future_1	Line_380kV _4kA_future	DIM380	ALR380	Noord-Holland	13.343
DIM380_ALR380 _Future_2	Line_380kV _4kA_future	DIM380	ALR380	Noord-Holland	13.343
LLS380_ALR380 _Future_1	Line_380kV _4kA_future	LLS380	ALR380	Noord-Holland	30.062
LLS380_ALR380 _Future_2	Line_380kV _4kA_future	LLS380	ALR380	Noord-Holland	30.062

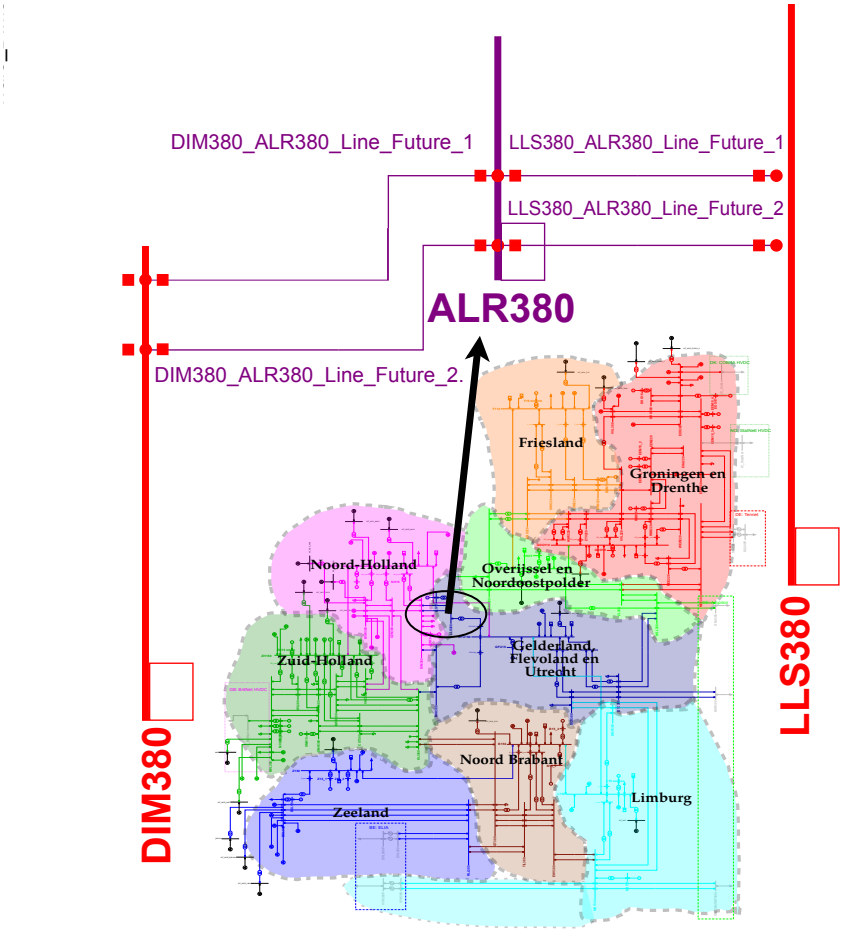


Figure 5.14: Implementation of substation ALR380 and corresponding changes to diagram highlighted in purple.

No information was found regarding the three remaining substation projects at the 380 kV level, nWCN380, nSDM380, and nWSP380. Thus, these were included in the upgrades, and future work should incorporate these when information becomes available.

5.4.2. Transmission Line Developments

Next, the transmission line developments in the study phase were implemented. The same procedure was followed as for the realization part, and similar assumptions were made. The transmission lines upgraded for this project phase are shown in Table 5.31, and lines added shown in Table 5.32

Table 5.31: PowerFactory Transmission Lines Upgraded for the Study project phase

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
MBT380_DOD380_W	Line_380kV_4kA_future	MBT380	DOD380	Limburg	99.6
MBT380_BMR380_Z	Line_380kV_4kA_future	MBT380	BMR380	Limburg	57.9
DOD380_BMR380_Z	Line_380kV_4kA_future	DOD380	BMR380	Limburg	41.7
DOD380_DTC380_W	Line_380kV_4kA_future	DOD380	DTC380	Utrecht	45.4
DOD380_DTC380_Z	Line_380kV_4kA_future	DOD380	DTC380	Utrecht	45.4
KIJ380_BKK380_W	Line_380kV_4kA_future	KIJ380	BKK380	Utrecht	20
BKK380_DIM380_W	Line_380kV_4kA_future	BKK380	DIM380	Utrecht	37.52
OZN380_DIM380_G	Line_380kV_4kA_future	OZN380	DIM380	Utrecht	15.2
OZN380_BVW380_W	Line_380kV_4kA_future	OZN380	BVW380	Noord-Holland	17
OZN380_BVW380_Z	Line_380kV_4kA_future	OZN380	BVW380	Noord-Holland	17
RLL380_GT380_W	Line_380kV_4kA_future	RLL380	GT380	Zeeland	61.213
RLL380_GT380_Z	Line_380kV_4kA_future	RLL380	GT380	Zeeland	61.213
DTC380_HGL380_W	Line_380kV_4kA_future	DTC380	HGL380	Utrecht	58.65
DTC380_HGL380_Z	Line_380kV_4kA_future	DTC380	HGL380	Utrecht	58.65
BGM220_VVL220_W	Line_220kV_4kA_005000R	BGM220	VVL220	Friesland	32.48
VVL220_ZYV220_W	Line_220kV_4kA_005000R	VVL220	ZYV220	Friesland	22.71
VVL220_ZYV220_Z	Line_220kV_4kA_005000R	VVL220	ZYV220	Friesland	22.71

Table 5.32: PowerFactory Transmission Lines Added for the Study project phase

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
GT380_TIL380 _Future_4	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	30
TIL380_EHV380 _Future_4	Line_380kV _4kA_future	TIL380	EHV380	Noord-Brabant	35
BSL380_RLL380 _Future_6	Line_380kV _4kA_future	BSL380	RLL380	Zeeland	38.9
BSL380_TNZ380 _Future_5	Line_380kV _4kA_future	BSL380	TNZ380	Zeeland	11.05
MBT380_GRTH380 _Future_3	Line_380kV _4kA_future	MBT380	GRTH380	Limburg	16.315
EEM380_EOS380 _Future_3	Line_380kV _4kA_future	EEM380	EOS380	Groningen	1.44

5.4.3. Transformer Developments

Finally, the transformer upgrades in the study phase of the investment plan are presented. This was done in a similar manner as for the realization and basic design phases. The resulting transformer upgrades are given in Table 5.33.

Table 5.33: Upgraded Transformers for the Study project phase, red numbers in parenthesis highlight the original values

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_BVW380 _NH150	BVW380	NH150	Noord-Holland	4 (2)	2000 (1000)	500
Trafo_BSL380 _Z150	BSL380	Z150	Zeeland	4 (3)	2000 (1500)	500
Trafo_OZN380 _NH150	OZN380	NH150	Noord-Holland	4 (3)	2000 (1500)	500
Trafo_LSM220 _F110	LSM220	F110	Friesland	3 (2)	600 (400)	200
Trafo_KIJ380 _ZH150_2	KIJ380	ZH150	Zuid-Holland	4 (2)	1800 (900)	450
Trafo_CST380 _ZH150_1	CST380	ZH150	Zuid-Holland	4 (3)	2000 (1500)	500
Trafo_CST380 _ZH150_2	CST380	ZH150	Zuid-Holland	4 (3)	2000 (1500)	500
Trafo_TIL380 _B150_N	TIL380	B150	Noord-Brabant	4 (3)	2000 (1500)	500
Trafo_BWK380 _ZH150	BWK380	ZH150	Zuid-Holland	4 (3)	2000 (1500)	500
Trafo_WTR380 _ZH150	WTR380	ZH150	Zuid-Holland	4 (3)	2000 (1500)	500

6

Model Development towards 2050 Scenario

6.1. Introduction

To fully assess the different scenarios and their respective dynamic stability, it is necessary to update the synthetic dynamic model to better fit the selected scenario for future work found in chapter 4. This includes varying levels of renewable generation, flexibility, demand, synchronous generation, and plans for improved or decommissioned infrastructure. This section delves deeper into the the scenario and the model's necessary adaptations.

The scenario utilized consists of the scenario found in II3050-2, specifically the national (NAT) scenario. This is the scenario that the TargeT Grid by TenneT is based on, and it serves as a "worst case" scenario for system expansion towards 2045 [114]. As a worst-case scenario, it serves as the projected highest degree of installed electrification and necessary upgrades to the high-voltage system. The highest degree of electrification is used as the benchmark because it is easier to scale down rather than up if the future outlook of the Dutch power system changes.

As mentioned, significant upgrades to the infrastructure are necessary due to the large amounts of renewable generation, flexibility measures, and high load peaks in this scenario. This will be highlighted in the following sections.

Because the timeline of 2050 is far in the future, no detailed plans are available for the specifics of the changes to transmission lines, generators, and substations. However, information is available regarding the location of different generation units, load centers, and general plans for infrastructure investments in specific regions. After implementing the plans specified in various resources, an additional assessment of the power system will be conducted using load flows and contingency analysis to identify possible bottlenecks. If bottlenecks are identified, they will either be improved or addressed as areas for future planning, depending on the severity of the bottleneck.

This allows the model to accurately represent the future power system and thus serve as a benchmark model for future analysis of the Dutch power system towards 2050.

6.2. Infrastructure Developments

With the high levels of renewable generation and flexibility, the need for a strengthened power system with increased transmission capabilities is essential. Because renewable generation depends on factors such as wind speed and sun hours, generation units are typically centered where the conditions are optimal for each type of generation. Consequently, when these factors are low, there may be a need for large power flows between regions to ensure a sufficient supply level for each region. This creates the possibility of significant bottlenecks in the system as some of the generated power needs to travel longer

distances. This presents a challenge for the existing power system, necessitating the implementation of new infrastructure. This was also identified as a weak point for the 2030 scenario by DeRoos [105].

The different reports offer varying perspectives on the need for future power system investments. However, they do not all provide the same level of detail. II3050-2 provides a detailed explanation of the predicted bottlenecks and necessary infrastructure developments. Thus, this serves as the main report for the upgrades towards 2050, coinciding with the selected National scenario implementation.

II3050-2 and IO-2050 identified similar bottlenecks when analyzing the system, primarily due to the power system's transit flow [106, 110]. The different scenarios showed effective handling of bottlenecks when electrolysers were placed close to large-scale renewable generation farms. However, during periods requiring transit flow between regions due to low renewable generation, many bottlenecks were observed in most scenarios [106, 110]. It is important to remember that identified bottlenecks depend on various factors, such as the placement of new large generation units, offshore wind landings, demand and international cooperation on new projects, and interconnections between countries. Therefore, there is no one-size-fits-all solution, and additional infrastructure may be necessary to improve the system in response to changes.

II3050-2 identifies similar bottlenecks for both national and international scenarios. One significant bottleneck is between the north and south of Holland to the south of the Netherlands. II3050-2 predicts that this must be strengthened with an additional high-voltage line. This spans from the stations of Beverwijk (BVW380) and Diemen (DIM380) towards Maasvlakte (MVL380) to Geertruidenberg (GT380) and further south to Rilland (RLL380) and Eindhoven (EHV380). This is a major concern due to the large amounts of offshore wind farm landings in northern Noord-Holland, making it critical to implement changes to account for these projects and allow power to flow to other regions.

First, the connection between Beverwijk (BVW380) and Diemen (DIM380) towards the south was upgraded. An additional transmission line was implemented between BVW380 and DIM380 and between BVW380 and Oostzan (OZN380). Other connections were implemented between BVW380 and Vijfhuizen (VHZ380), between OZN380 and Krimpen a/d IJssel (KIJ380), and between DIM380 and Breukelen Kortrijk (BKK380), extending to KIJ380.

Next, the connection from KIJ380 to EHV380 was upgraded. This was done with an additional transmission line similar to the previous case. First, the connection between KIJ380 and GT380 was upgraded, followed by the connection from GT380 to Tilburg (TIL380) and from TIL380 to EHV380.

Next, the connection from GT380 to Rilland was strengthened by adding an additional transmission line. First, the connection between GT380 and RLL380 was upgraded, which also meant upgrading the two connected substations on this line, namely the HST380 and MDK380 substations. Further, the line connecting one of the interconnections to Belgium, the RLL380 to ZVL380 connection, was upgraded.

Next, the connection from KIJ380 to MVL380 was upgraded with an additional transmission line. First, the connection between KIJ380 and CST380 was upgraded, then the connection between CST380 and SMH380, followed by SMH380 to Amaliehaven (AMH380), and finally AMH380 to MVL380.

II3050-2 also identified that the connection between Ens (ENS380) and Hengelo (HGL380) must be doubled, passing through Zwolle (ZL380). Here, two new transmission lines were implemented between the substations of ENS380 to ZL380 and ZL380 to HGL380.

The port of Borssele also needs further research into the supply and demand of the region. Here, upgrades will be considered if necessary during the initial load flows when the scenarios are complete [106].

Another bottleneck is found in the connections in Zeeland during the European integration scenario [106]. However, this occurs due to the influx of nuclear power in that area for the specific scenario and is not relevant to the three scenarios of interest.

There is also mention of DC connections being utilized for the onshore power system to alleviate these bottlenecks by allowing large power flows to be rerouted through the high-capacity DC line. However, this was not considered in this report.

If any of the transmission lines that underwent upgrades during this stage were of an older lower-rated type, such as 2 or 3 kA, these were upgraded to 4 kA lines. The resulting upgraded lines can be seen in Table 6.1, and the added lines in Table 6.2. The resulting single-line diagram of the entire system can be seen in Figure 6.1.

Table 6.1: 2050 Scenario - Lines Upgraded

Name	**Type in PowerFactory**	**Terminal I**	**Terminal J**	**Zone**	**Length (km)**
BVW380_VHZ380_W	Line_380kV_4kA_future	BVW380	VHZ380	Noord-Holland	14.31
BVW380_VHZ380_Z	Line_380kV_4kA_future	BVW380	VHZ380	Noord-Holland	14.31
KIJ380_OZN380_Z	Line_380kV_4kA_future	KIJ380	OZN380	Noord-Holland	72.529
RLL380_ZVL380_G	Line_380kV_4kA_future	RLL380	ZVL380	Zeeland	6.841
RLL380_ZVL380_W	Line_380kV_4kA_future	RLL380	ZVL380	Zeeland	6.841

Table 6.2: 2050 Scenario - Lines Added

Name	Type in PowerFactory	Terminal I	Terminal J	Zone	Length (km)
OZN380_BVW380 _Future_3	Line_380kV _4kA_future	OZN380	BVW380	Noord-Holland	17
OZN380_DIM380 _Future_2	Line_380kV _4kA_future	OZN380	DIM380	Noord-Holland	15.2
BVW380_VHZ380 _Future_3	Line_380kV _4kA_future	BVW380	VHZ380	Noord-Holland	14.31
KIJ380_OZN380 _Future_2	Line_380kV _4kA_future	KIJ380	OZN380	Noord-Holland	72.529
BKK380_DIM380 _Future_2	Line_380kV _4kA_future	BKK380	DIM380	Utrecht	37.52
KIJ380_BKK380 _Future_2	Line_380kV _4kA_future	KIJ380	BKK380	Utrecht	20.01
KIJ380_GT380 _Future_4	Line_380kV _4kA_future	KIJ380	GT380	Noord-Brabant	34.02
GT380_TIL380 _Future_5	Line_380kV _4kA_future	GT380	TIL380	Noord-Brabant	30
TIL380_EHV380 _Future_5	Line_380kV _4kA_future	TIL380	EHV380	Noord-Brabant	35
GT380_MDK380 _Line_Future_3	Line_380kV _4kA_future	CP	MDK380	Zeeland	9.95
RLL380_HST380 _Line_Future_3	Line_380kV _4kA_future	CP	HST380	Zeeland	10.519
RLL380_ZVL380 _Future_3	Line_380kV _4kA_future	RLL380	ZVL380	Zeeland	6.841
CST380_KIJ380 _Future_3	Line_380kV _4kA_future	CST380	KIJ380	Zuid-Holland	14.835
SMH380_CST380 _Future_3	Line_380kV _4kA_future	SMH380	CST380	Zuid-Holland	40.43
AMH380_SMH380 _Future_3	Line_380kV _4kA_001539R	AMH380	SMH380	Zuid-Holland	22.357
MVL380_AMH380 _Future_3	Line_380kV _4kA_001539R	MVL380	AMH380	Zuid-Holland	0.085
ENS380_ZL380 _Future_3	Line_380kV _4kA_future	ENS380	ZL380	Overijssel	25.266
ENS380_ZL380 _Future_4	Line_380kV _4kA_future	ENS380	ZL380	Overijssel	25.266
ZL380_HGL380 _Future_3	Line_380kV _4kA_future	ZL380	HGL380	Overijssel	47.528
ZL380_HGL380 _Future_4	Line_380kV _4kA_future	ZL380	HGL380	Overijssel	47.528

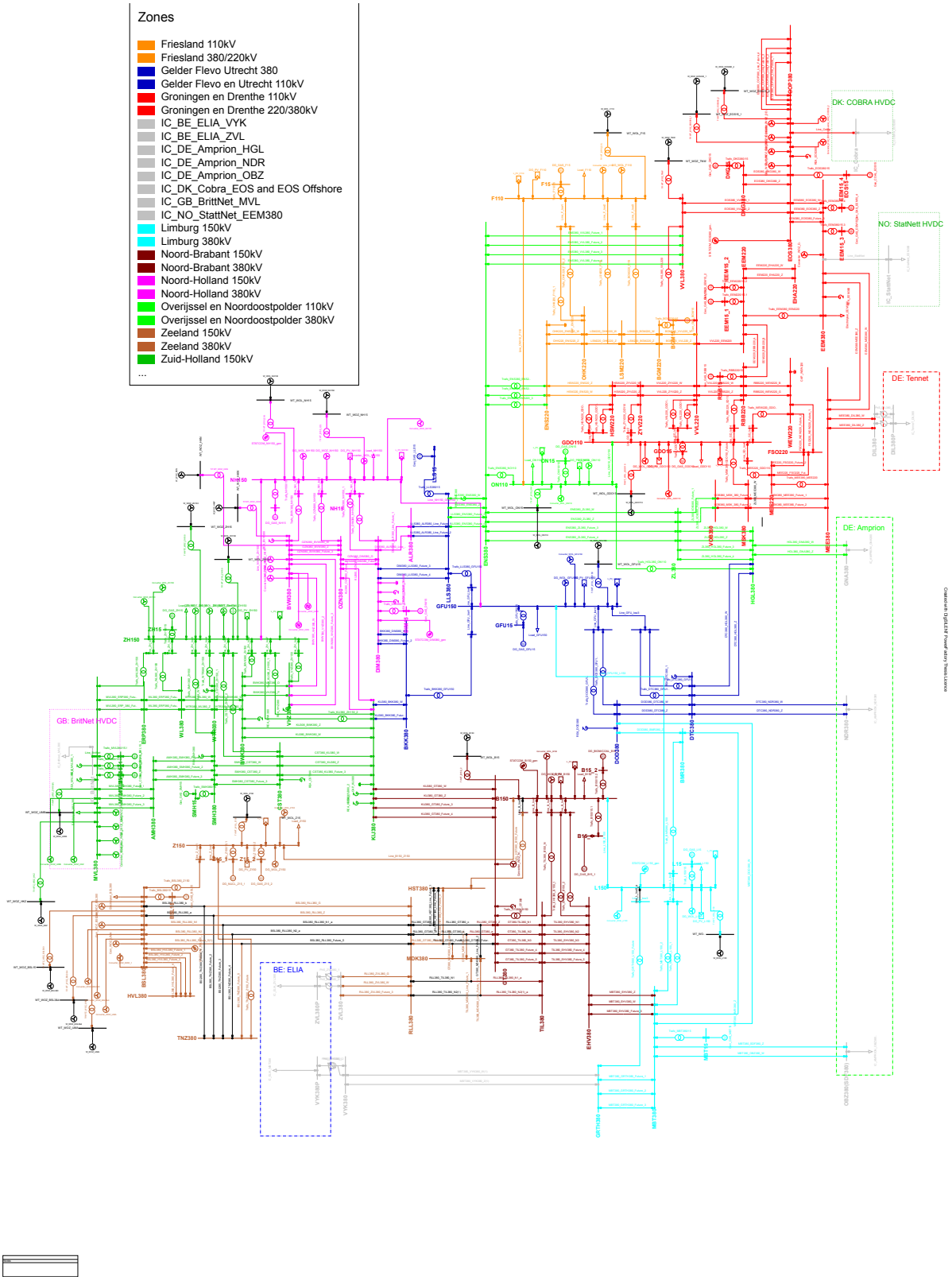


Figure 6.1: Overview of single line diagram with implemented changes for 2050, highlighting the different regions.

6.3. Generation Units

As mentioned, large amounts of Solar PV and wind energy are projected for this scenario. However, some traditional synchronous generation is still in play. How the different generation units are modeled and implemented according to scenario II3050-2 (NAT) will be further discussed.

6.3.1. Offshore Wind

First, offshore wind is discussed. As seen in Figure 4.7, the installed capacity of offshore wind in this scenario is 72 GW. With the planned offshore wind implementation until 2034 in Invest-Land-2024 [115], a total of 22.8 GW was installed. However, 20 GW of the projected 72 GW will be utilized exclusively for hydrogen production and will not be considered here [106]. Thus, a remaining 29.2 GW of offshore wind was installed between 2034 and 2050 for this scenario.

Due to the specific locations available for offshore wind projects, their connections to the main power system are also limited to specific regions and substations. It is desirable to spread these connections out as much as possible to avoid overloading any single area with injected power. TenneT mentions a maximum of three 2 GW offshore wind connections at a single substation to maintain stability [115]. This also suggests that flexibility resources should be prioritized in areas where multiple wind farms are connected to maximize the benefits and limit the need for transferring large amounts of power over long distances. These flexibility resources can include electrolysers, battery charging, and power to heat, and will be discussed later. By placing them close to the offshore wind farms, the efficiency of the systems will be maximized. II3050-2 also suggests deeper connections of offshore wind to substations further inland to avoid exclusively connecting to coastal substations and to distribute the generation more evenly [106]. All regions near the coast are susceptible to large integration of offshore wind farms.

The II3050-2 report does not specify offshore wind landing zones for their different scenarios. Thus, arbitrary substations spread across the available connection points on the coast were done. Ongoing work will highlight the offshore wind farms and landings projected towards 2040 and 2050, but this has not been published yet and should be updated accordingly at a later stage [135]. All wind farms will connected using a standard 2 GW DC connection after 2028 by TenneT, similar to the latest projects in the investment plan [136].

Two deep landing zones were identified as potential landings in Invest-land-2024 [115] at the substations at Moerdijk (MDK380) and Geertruidenberg (GT380). These was each maximized with three landings, totaling 6 GW individually and 12 GW in total. A 2 GW wind farm named Nederwiek 3 was already connected at GT380, so an additional 4 GW was connected. Another connection was at the new 380 kV station of Terneuzen (TNZ380) in Zeeland, which was also fully loaded with three individual 2 GW offshore stations, totaling 6 GW. Another connection was at the Oostpolder 380 kV (OOP380) station, which was upgraded with two 2 GW farms, totaling 4 GW. Another connection was at Lelystad (LLS380), with three 2 GW connections totaling 6 GW. The remaining 3.2 GW to fully implement the projected capacity was connected at the new 380 kV station of Halsteren (HST380), with one 2 GW and one 1.2 GW station. The newly implemented offshore wind farms can be seen in Table 6.3.

All of the offshore wind generators were added twice: once as a static generator connected directly to the power system terminals, used to represent the grid-forming converters with different types of controllers, and once through a 15 kV transformer using a grid-following controller type. This way, the dispatch of grid-forming and grid-following can be regulated to analyze the impact of different levels of grid-forming generation. The grid-forming converters are given the prefix "Converter," while the grid-following converters are given the prefix "W."

The newly implemented offshore wind farms are given in Table 6.3. Here, the wind farms utilizing grid-forming control are given the same names given to the grid-following controls.

Table 6.3: Overview of different Offshore Wind farms implemented throughout 2050 for scenario II3050-2 (NAT)

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Converter _WOZ_MDK380_1	2050	Future	MDK380	Zeeland	380	2000
Converter _WOZ_MDK380_2	2050	Future	MDK380	Zeeland	380	2000
Converter _WOZ_MDK380_3	2050	Future	MDK380	Zeeland	380	2000
Converter _WOZ_GT380_1	2050	Future	GT380	Noord-Brabant	380	2000
Converter _WOZ_GT380_2	2050	Future	GT380	Noord-Brabant	380	2000
Converter _WOZ_TNZ380_1	2050	Future	TNZ380	Zeeland	380	2000
Converter _WOZ_TNZ380_2	2050	Future	TNZ380	Zeeland	380	2000
Converter _WOZ_TNZ380_3	2050	Future	TNZ380	Zeeland	380	2000
Converter _WOZ_OOP380_1	2050	Future	OOP380	Groningen	380	2000
Converter _WOZ_OOP380_2	2050	Future	OOP380	Groningen	380	2000
Converter _WOZ_LLS380_1	2050	Future	LLS380	Utrecht	380	2000
Converter _WOZ_LLS380_2	2050	Future	LLS380	Utrecht	380	2000
Converter _WOZ_LLS380_3	2050	Future	LLS380	Utrecht	380	2000
Converter _WOZ_HST380_1	2050	Future	HST380	Zeeland	380	2000
Converter _WOZ_HST380_2	2050	Future	HST380	Zeeland	380	1200

6.3.2. Onshore Wind

Next, the installations of onshore wind for the II3050-2 National scenario will be discussed. The scenario predicts an installed capacity of 20 GW of onshore wind [106]. A total of 7.28 GW was already implemented in the synthetic model for the 2034 scenario. Thus, an additional 12.72 GW was installed. II3050-2 does not provide specific information regarding connection points for onshore wind farms, but a map indicating zones for future generation is provided. This was used to estimate connection points and sizes for the different wind farms. The largest areas suited for onshore wind are North Holland (Noord-Holland), North Gelderland (Gelder Flevo Utrecht), and West Friesland. There are also suitable zones in the northern parts of South Holland (Zuid-Holland), as well as in the region surrounding North Brabant (Noord-Brabant), Limburg, and Gelderland (Gelder Flevo Utrecht). However, these are not as suitable as the larger areas mentioned and are better suited for smaller wind farms. There are multiple regions in the Netherlands that are well-suited for onshore wind due to the country's high wind characteristics. However, onshore wind farms require large spaces for construction, and the high influx of variable offshore wind in some regions makes spreading the onshore wind to other places more beneficial. This is accounted for in II3050-2 [106].

The already installed onshore wind farms were connected at 150 and 110 kV voltage levels. The new additional wind farms were connected to 380, 220, 150, and 110 kV levels, with the largest wind farms connected at the 380 kV level.

Onshore wind farms are typically smaller in capacity due to the large amounts of space required

and the limitations in the size of wind turbines due to noise and other external factors that affect the ecosystem [137]. Because of this, the remaining 12.72 GW of onshore wind was distributed through 15 new onshore wind farm connections in the regions mentioned earlier, with 7 of these being larger wind farms connected at the 380 kV level and the remaining 8 connected at lower voltage levels. A more accurate representation of onshore wind farms should be implemented once more detailed projections are available.

As before, the wind farms were implemented both as grid-forming and grid-following in the same manner as earlier. An overview of the connected wind farms can be seen in Table 6.4, making the total onshore wind connected in the system 20 GW.

Table 6.4: Overview of different Onshore Wind farms implemented throughout 2050 for scenario II3050-2 (NAT)

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Converter _WOL_ALR380	2050	Future	ALR380	Noord-Holland	380	1000
Converter _WOL_OZN380	2050	Future	OZN380	Noord-Holland	380	1000
Converter _WOL_DOD380	2050	Future	DOD380	Utrecht	380	1000
Converter _WOL_ENS380	2050	Future	ENS380	Overijssel	380	1000
Converter _WOL_VVL380	2050	Future	VVL380	Groningen	380	1000
Converter _WOL_BKK380	2050	Future	BKK380	Utrecht	380	1000
Converter _WOL_VHZ380	2050	Future	VHZ380	Zuid-Holland	380	1000
Converter _WOL_HSW220	2050	Future	HSW220	Groningen	220	715
Converter _WOL_LSM220	2050	Future	LSM220	Friesland	220	715
Converter _WOL_ZYV220	2050	Future	ZYV220	Groningen	220	715
Converter _WOL_GDO110	2050	Future	GDO110	Groningen	110	715
Converter _WOL_GFU150_2	2050	Future	GFU150	Utrecht	150	715
Converter _WOL_NH150_2	2050	Future	NH150	Noord-Holland	150	715
Converter _WOL_ZH150_2	2050	Future	ZH150	Zuid-Holland	150	715
Converter _WOL_ON110_2	2050	Future	ON110	Overijssel	110	715

6.3.3. Solar PV

Next, the implementation of Solar PV generation for the II3050-2 (NAT) scenario will be discussed. II3050-2 predicts 173 GW of Solar PV installed by 2050 in the power system. Of these, 58 GW is expected to come from installations on land and water, while the remaining 115 GW comes from installations on buildings and dwellings.

As before, II3050-2 does not provide specifics about the placements of these Solar PV generation units. The 58 GW expected to come from installations on land and water was connected to the 380 kV part of the power system to represent large-scale projects. These installations are also expected to primarily be located in regions with a low population density, thus avoiding connections close

to large cities. However, the remaining 115 GW, representing Solar PV on buildings and dwellings, was connected to the lower voltage levels of 220 kV, 150 kV, and 110 kV, as well as the 380 kV power system near large cities to represent urban installations. The capacity of the different installations was standardized for all connections within the two specific categories of installation.

A total of 24.75 GW was already installed in the synthetic model; thus, a remaining 148.25 GW was installed. The type of Solar PV already installed is not specified, and thus, it was estimated as a mix of both types and implemented as half on buildings and dwellings and half on land and water. Therefore, a total of 148.25 GW remained, with 49.7 GW connected as installations on land and water and 98.55 GW installed on buildings and dwellings, for a total of 173 GW installed. The 49.7 GW connected as installations on land and water were represented by 10 different installations on the 380 kV power system in rural areas, each with a capacity of 4.97 GW. The 98.55 GW installed on buildings and dwellings was represented by connections to 21 substations at voltage levels of 220 kV, 150 kV, and 110 kV, and 5 substations at 380 kV representing the 5 largest cities in the Netherlands [138]. These had a capacity of 3.79 GW each. As with the wind farms, all installations was implemented as both grid-forming and grid-following installations.

The grid-forming converters were given a prefix of "Converter", while the grid-following converters were given the suffix L_PV. The grid-following converters were modeled using the ElmPVsys component in PowerFactory. This allows for control similar to a static generator with active power control but also allows for a more accurate representation of the power output of the converter based on the geographical location and placement of the converter as a solar PV resource [139]. However, the grid-forming converter frame could not connect to this component type and was modeled as a standard static generator. For future work, it will be important to allow grid-forming connections for these models and to represent Solar PV through more accurate estimations of power generation for specific locations and sizes of the panels to account for the variabilities experienced with Solar PV generation. The installed Solar PV installations can be seen in Table 6.5.

Table 6.5: Overview of different Solar PV converters implemented throughout 2050 for scenario II3050-2 (NAT). The green input represents installations marked as "on land and water"

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Converter _PV_BMR380	2050	Future	BMR380	Limburg	380	4970
Converter _PV_DOD380	2050	Future	DOD380	Utrecht	380	4970
Converter _PV_DTC380	2050	Future	DTC380	Utrecht	380	4970
Converter _PV_HGL380	2050	Future	HGL380	Overijssel	380	4970
Converter _PV_HST380	2050	Future	HST380	Zeeland	380	4970
Converter _PV_ZL380	2050	Future	ZL380	Overijssel	380	4970
Converter _PV_TIL380	2050	Future	TIL380	Noord-Brabant	380	4970
Converter _PV_VOB380	2050	Future	VOB380	Groningen	380	4970
Converter _PV_VVL380	2050	Future	VVL380	Groningen	380	4970
Converter _PV_ZL380_2	2050	Future	ZL380	Overijssel	380	4970
Converter _PV_BKK380	2050	Future	BKK380	Utrecht	380	3790

Continued on next page

Table 6.5 – continued from previous page

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Converter _PV_DIM380	2050	Future	DIM380	Noord-Holland	380	3790
Converter _PV_EHV380	2050	Future	EHV380	Noord-Brabant	380	3790
Converter _PV_KIJ380	2050	Future	KIJ380	Zuid-Holland	380	3790
Converter _PV_WTR380	2050	Future	WTR380	Zuid-Holland	380	3790
Converter _PV_BGM220	2050	Future	BGM220	Friesland	220	3790
Converter _PV_EEM220	2050	Future	EEM220	Groningen	220	3790
Converter _PV_EHA220	2050	Future	EHA220	Groningen	220	3790
Converter _PV_ENS220	2050	Future	ENS220	Friesland	220	3790
Converter _PV_FSO220	2050	Future	FSO220	Groningen	220	3790
Converter _PV_HSW220	2050	Future	HSW220	Groningen	220	3790
Converter _PV_LSM220	2050	Future	LSM220	Friesland	220	3790
Converter _PV_MEE220	2050	Future	MEE220	Groningen	220	3790
Converter _PV_OHK220	2050	Future	OHK220	Friesland	220	3790
Converter _PV_RBB220	2050	Future	RBB220	Groningen	220	3790
Converter _PV_VVL220	2050	Future	VVL220	Groningen	220	3790
Converter _PV_WEW220	2050	Future	WEW220	Groningen	220	3790
Converter _PV_ZYV220	2050	Future	ZYV220	Groningen	220	3790
Converter _PV_B150	2050	Future	B150	Noord-Brabant	150	3790
Converter _PV_GFU150	2050	Future	GFU150	Utrecht	150	3790
Converter _PV_L150	2050	Future	L150	Limburg	150	3790
Converter _PV_NH150	2050	Future	NH150	Noord-Holland	150	3790
Converter _PV_Z150	2050	Future	Z150	Zeeland	150	3790
Converter _PV_F110	2050	Future	F110	Friesland	110	3790
Converter _PV_GDO110	2050	Future	GDO110	Groningen	110	3790
Converter _PV_ON110	2050	Future	ON110	Overijssel	110	3790

6.3.4. Nuclear Energy

Next, the implementation of nuclear energy will be discussed. Nuclear energy was once considered the frontrunner for sustainable future power systems due to its reliable power supply and zero carbon output [140]. However, due to a number of serious events involving nuclear energy, such as the Chernobyl accident and the Fukushima disaster, global skepticism rose around nuclear energy, limiting its growth [141]. These days, nuclear energy splits opinions, with some viewing it as a failed energy resource while others still consider it vital for future power systems. The II3050-2 NAT scenario accounts for some level of nuclear power, but with a mere 3 GW installed capacity compared to the high capacity of solar PV and wind energy.

There is currently one operating nuclear power plant in the Netherlands, located in the region of Borssele [142]. II3050-2 mentions that this plant will continue to operate and is included in the estimations for the different scenarios [106]. II3050-2 also mentions that the nuclear power plants assumed in the different scenarios are located in Borssele and on the Maasvlakte [106]. Therefore, two nuclear power plants were implemented at these two locations with a capacity of 1.5 GW each and connected to the buses of BSL380 and MVL380, representing the respective regions. A template for a nuclear power plant was already provided in the PowerFactory model and further utilized in the power plants. The power plants were connected to the 380 kV power system, with a power plant voltage of 15 kV through a transformer, similar to other connected generators in the synthetic model, for simplicity.

The implemented nuclear power plants can be seen in Table 6.6.

Table 6.6: Overview of Nuclear Power Plants implemented throughout 2050

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Gen_Nuclear_BSL380	2050	Future	BSL380	Zeeland	380	1500
Gen_Nuclear_MVL380	2050	Future	MVL380	Zeeland	380	1500

6.4. Flexibility Resources

Next, the implementation of flexibility resources will be discussed. The choice of location for flexibility resources plays a vital role in the characteristics of the power system. A smart allocation of flexibility resources reduces the stress that variable generation causes on the power system during times of high generation and low demand while also being able to provide power when renewable generation is low. This allows flexibility resources to limit the impact of renewable generation variability on system operation. However, if flexibility resources are not managed well, their operation can be detrimental to the system. Large power flows between areas and regions can occur by placing flexibility resources far away from renewable generation, resulting in unnecessary heavy system loading, large losses, and potentially leading to stability issues. If the placement is close to the renewable generation, the flexibility resources can be utilized in the same substation as the renewable generation. This minimizes losses as there is no need for power flow over large distances, and the efficiency of these flexibility resources becomes much higher, thus making them more viable for operation. Different flexibility resources have different operational characteristics, but they all serve to mitigate the variability of renewable generation units on system operation.

By 2050, the total load demand will exceed the total renewable generation during periods of up to 35-50 GW, depending on different scenarios. At the same time, renewable generation is expected to account for amounts ranging from 65-95 GW more than the total load demand during other periods [106]. This highlights the high need for flexibility in future power systems. The need for flexibility depends on a mix of existing and new technologies, ranging from short and medium-term resources such as battery storage and supply and demand management to long-term resources such as power-to-gas, gas storage, and gas-to-power. These resources complement each other and must be utilized in unison. For the flexibility resources in the specific II3050-2 (NAT) scenario, a total of 152.8 GW of flexible power is estimated.

It is important to note that the operation of flexibility resources depends on various factors, such

as weather conditions and region loading, so exact requirements and capacities may vary.

6.4.1. Battery

First, battery implementation is discussed. A total of 60 GW is estimated for 2050. As before, II3050-2 does not provide direct connections and sizes of various battery projects; however, a map indicating space requirements for future battery systems is given and was utilized to indicate where the installations will occur. Along with this information, the battery units were placed as close as possible to areas with high wind and/or solar installation concentrations. More accurate placements and sizes of relevant battery storage systems should implemented for future work. The map indicates that Groningen en Drenthe, Noord and Zuid-Holland, Noord-Brabant, and Zeeland will implement the largest amounts of battery systems towards 2050. These areas have large installations of renewable energy and were used as the main connection points for the battery systems. TenneT also mentions that battery systems should be connected close to large installations of Solar PV, wind energy, or industrialized areas [143].

Battery systems consist of battery energy storage systems (BESS) connected to the power system and ancillary services provided by electric vehicles acting as battery energy systems and consumer battery systems. The national scenario estimates large amounts of these electric vehicles [106, 113]. However, these was represented as a single entity in this project.

A total of 20 battery systems were implemented, each with a capacity of 3 GW. This simplifies the process and does not accurately represent the respective battery projects towards 2050 in terms of size, placement, and definite operation. However, the lack of specific plans made simplifications necessary. The 3 GW installed representation represents the battery energy storage capacity in its entirety and does not differentiate between large and small-scale projects. The 20 battery systems were distributed within the 5 regions mentioned earlier and were arbitrarily placed, emphasizing a connection close to large generation centers. The battery systems were represented using the predefined template WECC Battery Energy Storage System Templates in PowerFactory. This models the system as a static generator and allows for battery charging and discharging. It is also equipped with dynamic models that allow for a more accurate representation of battery dynamics with respect to the states of both charging and discharging. More information can be found in the technical reference for the type in PowerFactory [144].

The battery systems implemented can be seen in Table 6.7.

Table 6.7: Overview of different BESS installations implemented throughout 2050 for scenario II3050-2 (NAT)

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
BESS_EOS380	2050	Future	EOS380	Groningen	380	3000
BESS_OOP380	2050	Future	OOP380	Groningen	380	3000
BESS_EEM380	2050	Future	EEM380	Groningen	380	3000
BESS_VOB380	2050	Future	VOB380	Groningen	380	3000
BESS_BVW380	2050	Future	BVW380	Noord-Holland	380	3000
BESS_DIM380	2050	Future	DIM380	Noord-Holland	380	3000
BESS_BKK380	2050	Future	BKK380	Noord-Holland	380	3000
BESS_ALR380	2050	Future	ALR380	Noord-Holland	380	3000
BESS_MVL380	2050	Future	MVL380	Zuid-Holland	380	3000
BESS_KIJ380	2050	Future	KIJ380	Zuid-Holland	380	3000
BESS_WTR380	2050	Future	WTR380	Zuid-Holland	380	3000
BESS_ZH150	2050	Future	ZH150	Zuid-Holland	150	3000
BESS_TIL380	2050	Future	TIL380	Noord-Brabant	380	3000
BESS_B150	2050	Future	B150	Noord-Brabant	150	3000
BESS_GT380	2050	Future	GT380	Noord-Brabant	380	3000
BESS_EHV380	2050	Future	EHV380	Noord-Brabant	380	3000
BESS_TNZ380	2050	Future	TNZ380	Zeeland	380	3000
BESS_MDK380	2050	Future	MDK380	Zeeland	380	3000
BESS_BSL380	2050	Future	BSL380	Zeeland	380	3000
BESS_RLL380	2050	Future	RLL380	Zeeland	380	3000

6.4.2. Hydrogen

Next, the hydrogen flexibility resources will be discussed. Flexibility resources based on hydrogen primarily come from two different operations: using electricity to create hydrogen or using hydrogen to create electricity. This allows excess power to be utilized for hydrogen production instead of limiting excess power output by methods such as curtailment. Vice versa, this allows generation and ancillary services to be provided to the power system during low-generation periods.

A total of 40 GW of hydrogen-based flexibility services is estimated in the II3050-2 NAT scenario. Of these, 25 GW are used as electrolysers, while the remaining 15 GW are used as flexible power plants. As before, there are no specific details regarding the locations and planned integration of hydrogen. II3050-2 specifies that electrolysers should be placed in locations with high surpluses of generation, mostly occurring in coastal areas, while controllable power plants should be placed in areas with generation shortages [106].

There were three electrolysers placed in the synthetic model located at BSL380, EEM380, and MVL380. This represents the largest offshore wind ports and corresponds with the suggestions in II3050-2 regarding placement. 7 new electrolysers were implemented, totaling 10 with a rating of 2.5 GW each. This was selected as an arbitrary size for simplicity; future work should implement sizes and locations based on real-life data and experiences. The electrolysers were modeled as a general load in PowerFactory and implemented with a dynamic model. This dynamic model uses a reverse droop controller, allowing the electrolyser to adjust its active power based on the frequency, reducing its active power consumption when the frequency drops and increasing it when the frequency rises. This helps with the fast-active power balancing of the system and allows the electrolysers to provide ancillary services to the power system. The remaining 7 electrolysers were placed along the coast, similar to the first three, specifically at the substations of EOS380, HST380, BVW380, TNZ380, LLS380, OOP380, and DIM380. The electrolysers can be seen in Table 6.8.

Table 6.8: Overview of different electrolyser Installations implemented throughout 2050

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Electrolyser_BSL380	2050	Future	BSL380	Zeeland	380	2500
Electrolyser_EEM380	2050	Future	EEM380	Groningen	380	2500
Electrolyser_MVL380	2050	Future	MVL380	Zuid-Holland	380	2500
Electrolyser_EOS380	2050	Future	EOS380	Groningen	380	2500
Electrolyser_HST380	2050	Future	HST380	Zeeland	380	2500
Electrolyser_BVW380	2050	Future	BVW380	Noord-Holland	380	2500
Electrolyser_TNZ380	2050	Future	TNZ380	Zeeland	380	2500
Electrolyser_LLS380	2050	Future	LLS380	Utrecht	380	2500
Electrolyser_OOP380	2050	Future	OOP380	Groningen	380	2500
Electrolyser_DIM380	2050	Future	DIM380	Noord-Holland	380	2500

The remaining 15 GW utilized as flexible power plants was implemented as two different resources: fuel cells and traditional power plants, either converted gas power plants or state-of-the-art hydrogen power plants. Fuel cell technology provides similar ancillary services to the power system as those connected through power electronics, including fast-acting voltage and frequency support, while not being limited by an external component such as wind for wind turbines, but rather by the amount of stored hydrogen [145, 146]. This is a great resource for maintaining system stability and managing supply and demand variations concerning the large amounts of Solar PV and wind energy in future systems. Grid-forming services related to the operation of fuel cells are also under investigation and could prove to be a viable option for further controlling the future system using the benefits of grid-forming control [147, 148]. On the other hand, utilizing traditional gas turbines or new state-of-the-art hydrogen turbines can help stabilize the power system like traditional synchronous generation [149].

Much research is focused on how the power system needs to adapt to maintain stability with reduced inertia in the system. However, by utilizing hydrogen power plants, inertia levels can be maintained while still achieving net zero emission goals. Most research on hydrogen-to-power production is done on fuel cells, but research is underway for hydrogen turbines and gas-altered hydrogen turbines. One of the largest projects currently ongoing in the Netherlands is the conversion of a state-of-the-art gas-powered power plant to a hydrogen power plant based in Eemshaven [150]. This is a 1.4 GW power plant, with one of the three 440 MW generators expected to be fully operational by 2025 [150]. Thus, it might be a viable option for future scenarios and can be a vital contributor to the future power system.

To account for both different methods of producing power using hydrogen, and due to the uncertain future regarding which direction will gain the most prominence, both sources were implemented in the synthetic model to allow for simulations utilizing both. This can be used to analyze which type of power source provides the best resources to the power system, or it can be easily implemented based on the actual projects the future brings. Given the example of the 1.4 GW power plant in Eemshaven mentioned, 10 power plants was implemented for both resources with a size of 1.5 GW each. The largest fuel cell project currently ongoing worldwide is the Hinincheon Bitdream Fuel Cell Power Plant in Incheon, with a total capacity of 78.96 MW [151]. This suggests that fuel cells currently do not come in the sizes described here, and BS2050 [113] mentions that hydrogen power plants based on steam turbines similar to gas turbines will be the most prominent future option. As II3050-2 is a further stage project of BS2050, and it is not explicitly stated which technology is utilized in the II3050-2 scenarios, steam turbines (and thus synchronous generation) will be utilized in further simulation, but the fuel cell technology will be implemented in the model for future work.

The fuel cells were implemented using standard templates found in PowerFactory, while the synchronous generation was implemented using the same structure as current gas power plants. This helps maintain realistic levels of inertia from each respective power plant and assumes the control of hydrogen power plants will be similar to gas power plants. A new type in PowerFactory was created for synchronous generation, utilizing the same structure as the gas power plant found at EEM380,

the one currently being converted to hydrogen operation. These were standardized as single 1.5 GW synchronous generators for simplicity, and 10 of these were installed. Little is said about the location of these future controllable hydrogen power plants; however, it is mentioned that placing these at the locations of existing power plants minimizes the impact on the power system [106]. Thus, the 10 power plants were installed at locations with existing large synchronous power plants. Based on this, the implemented controllable hydrogen power plants modeled in the synthetic model can be seen in Table 6.9. The corresponding fuel cell implementations can be seen in Table 6.10.

Table 6.9: Overview of different Hydrogen Generation Plants implemented throughout 2050

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Gen_Hydrogen_MVL380	2050	Future	MVL380	Zuid-Holland	380	1500
Gen_Hydrogen_BSL380	2050	Future	BSL380	Zeeland	380	1500
Gen_Hydrogen_DIM380	2050	Future	DIM380	Noord-Holland	380	1500
Gen_Hydrogen_EOS380	2050	Future	EOS380	Groningen	380	1500
Gen_Hydrogen_EEM380	2050	Future	EEM380	Groningen	380	1500
Gen_Hydrogen_ZH150	2050	Future	ZH150	Zuid-Holland	150	1500
Gen_Hydrogen_NH150	2050	Future	NH150	Noord-Holland	150	1500
Gen_Hydrogen_MBT380	2050	Future	MBT380	Limburg	380	1500
Gen_Hydrogen_GFU150	2050	Future	GFU150	Utrecht	150	1500
Gen_Hydrogen_ZH150_2	2050	Future	ZH150_2	Zuid-Holland	150	1500

Table 6.10: Overview of different Fuel Cell Installations implemented throughout 2050

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Fuelcell_MVL380	2050	Future	MVL380	Zuid-Holland	380	1500
Fuelcell_BSL380	2050	Future	BSL380	Zeeland	380	1500
Fuelcell_DIM380	2050	Future	DIM380	Noord-Holland	380	1500
Fuelcell_EOS380	2050	Future	EOS380	Groningen	380	1500
Fuelcell_EEM380	2050	Future	EEM380	Groningen	380	1500
Fuelcell_ZH150	2050	Future	ZH150	Zuid-Holland	150	1500
Fuelcell_NH150	2050	Future	NH150	Noord-Holland	150	1500
Fuelcell_MBT380	2050	Future	MBT380	Limburg	380	1500
Fuelcell_GFU150	2050	Future	GFU150	Utrecht	150	1500
Fuelcell_ZH150_2	2050	Future	ZH150_2	Zuid-Holland	150	1500

6.4.3. Interconnection

Interconnection between countries will play a significant role in balancing and regulating future power systems. Due to the higher dependency on intermittent generation, utilizing interconnections between different countries and regions allows for the stabilization of these intermittent resources by leveraging diverse resources across different regions. Since wind and solar PV power generation have different cycles in various regions across Europe, interconnecting these regions and countries creates a more diverse power system less dependent on local external factors.

However, a dependency on interconnection can also pose risks. As future power systems become more reliant on intermittent resources such as solar and wind, these resources can be scarce during extreme weather years. If interconnected countries experience similar shortages, the interconnection capacity can be drastically reduced, making it a non-viable flexibility resource. Nonetheless, the chances of this are significantly reduced when interconnected to multiple countries with power systems of diverse characteristics, allowing the utilization of different countries' resources. Interconnections can also effectively address excess generation during high wind or solar PV generation periods. In these times, the excess wind or solar PV can be exported and sold to other countries, effectively utilizing the excess power instead of resorting to curtailment techniques [68, 152].

For the II3050-2 scenario, the interconnection capacity is estimated to be around 18.8 GW for the national scenario [106] towards 2050. The interconnection level is estimated by TenneT to be 9.2 GW for 2025, indicating that a doubling is necessary by 2050 [115]. Additional interconnections are in the planning stage, such as the Lion cable connecting the Netherlands to the UK via a DC connection [153]. However, since this is a hybrid interconnection project connected through a 2 GW offshore wind farm in the North Sea, it was implemented in the synthetic model at this stage and must be considered for future work [154].

Here, the current interconnections were updated, with active power limits reflecting the levels estimated for 2050. This was achieved by upgrading their capacity based on a percentage scale relative to their current capacities, aligned with the future scenario.

The interconnections and their updated active power ratings can be seen in Table 6.11.

Table 6.11: Overview of different interconnection capacities implemented throughout 2050, red numbers in parenthesis highlight the original values

Name	Year	Type	Country	Zone	Voltage (kV)	Capacity (MW)
IC_AMPRION _GNA380_1	2050	AC	Germany	Overijssel	380	1577 (650)
IC_AMPRION _NDR380_1	2050	AC	Germany	Utrecht	380	1577 (650)
IC_AMPRION _OBZ380	2050	AC	Germany	Limburg	380	1577 (650)
IC_BritNed _MVL380	2050	DC	United Kingdom	Zuid-Holland	380	2426 (1000)
IC_COBRA _EOS380	2050	DC	Denmark	Groningen	380	1698 (700)
IC_ELIA _MBT380_1	2050	AC	Belgium	Limburg	380	1698 (700)
IC_ELIA _ZVL380_1	2050	AC	Belgium	Zeeland	380	1698 (700)
IC_StatNett _EEM380	2050	DC	Norway	Groningen	380	1698 (700)
IC_TenneT _DIL380_1	2050	AC	Germany	Groningen	380	4852 (2000)

The interconnections were changed to dynamic models to enhance their dynamic characteristics. The previous scenarios modeled the interconnections as regular loads, resulting in a constant active and reactive power output regardless of faults and load changes in the system. Synchronous generators were utilized to model the AC interconnections to incorporate a more realistic dynamic response from these units for simulation purposes, and an HVDC VSC-MMC static generator was utilized for the DC connections.

The DC interconnection, being modeled as a static generator, allows for more dynamic similarities to HVDC connections, with a built-in template further improving the realistic operational characteristics found in PowerFactory. The model is available in PowerFactory's built-in model library and is called "DIgSILENT HVDC MMC Terminal." This model also allows for a greater dynamic representation based on the VSC-MMC elements included in the template calculations, although the DC circuit representation is not implemented. PowerFactory does not provide a technical reference for the model, but more information regarding the operation of VSC-HVDC connections using MMC can be found in [155], and its application in PowerFactory in [156]. Three different control modes are available in the model: V/f, P/AC, and Q/Vac. The different controllers utilize different variables to control the connection in a specific manner, and the controller selection will be specified in the simulation setup. The controller

can also differentiate its operating conditions based on whether the connection is islanded or parallel with other power systems; in this scenario, it will be parallel. While some of the interconnections in operation today do not utilize VSC-HVDC MMC-based technology, due to its prominence in research and applicability for many different operations, it is assumed that all new interconnections will be implemented with this technology, while the older converters have upgraded their systems [157].

The AC interconnection was modeled using a general synchronous generator representing the connected synchronous power system. A standard control system containing a governor, exciter, and PSS was utilized. Different levels of inertia will be introduced for the interconnections, with inertia levels based on a system needs study done by ENTSO-E for future scenarios of the power system and their respective inertia levels, found for continental Europe, which the Netherlands is part of [158]. These levels will be implemented and analyzed in the simulation case studies. The estimated levels for the inertia constant H (s) in 2040 range from around 4.1 to around 0.4 [s] for the three different scenarios [158]. Future predictions by ENTSO-E also estimate the kinetic energy for continental Europe to be in the region of 1800 to 250 GVA · s in 2040 based on different scenarios for different yearly operating hours [159]. By using this kinetic energy and calculating the resulting rated power of the system using equation 3.6 in [159], the interconnected systems can be represented using a synchronous machine in PowerFactory with the rated power and inertia constant. This calculation is given in Equation 6.1, and a similar approach will be used for different estimations of kinetic energy and inertia constant to simulate different levels.

$$H = \frac{E_{\text{kin}}}{S} \qquad S = \frac{E_{\text{kin}}}{H} \qquad S = \frac{1800\ \text{GVA} \cdot \text{s}}{4.1\ \text{s}} = 439,024\ \text{MVA} \tag{6.1}$$

Because of this change, the interconnections' active and reactive power limits are handled using a per-unit conversion for the maximum rating through the governor model. This also has to be changed if the constant kinetic energy and inertia levels change. The constant kinetic energy and inertia levels must be updated to 2050 estimations once they are available. Due to the uncertainties during different operational scenarios throughout the year as well as for different scenario predictions in 2040, the inertia constant and kinetic energy levels in the system need to be changed during different simulations to fully understand the impact and register the effect of different inertia levels on system behavior for the future Dutch power system.

By modeling the interconnections as synchronous machine connections instead of loads, the dynamic response of the interconnections to events will be included, creating a more realistic representation of the interconnections. One of the larger benefits of interconnection on power system stability, namely inertia, is implemented in the dynamic studies with this solution, providing a more realistic power system.

While there are multiple benefits to AC interconnected systems, they also exhibit drawbacks. By interconnecting, other power systems can greatly impact other interconnected systems, meaning that shortages and events occurring in different parts of the system will affect all interconnected systems. Despite rigorous development of grid codes and agreements to allow the interconnection of zones, issues may still arise due to dependencies on other power systems. An example occurred in the 1970s in the United States when interconnections were implemented through AC connections instead of DC connections, leading to dynamic oscillations in the western part of the country, resulting in power system instability with large inadvertent power exchanges [160].

6.4.4. Demand Response

II3050-2 also estimates the amount of demand response in the industry to be around 10 GW. This falls under the "other flexibility resources" definition given in chapter 4. Demand response represents an operation where industrial operators have agreements with the TSO regarding flexible demand, responding to frequency deviations in the power system [161]. This means that if the frequency deviates beyond a certain limit, specific actors in the industry will automatically reduce some of their operations, limiting their load output to retain or restore system frequency if necessary. This is a valuable asset for system operation and stability due to the minimal operational changes needed to accommodate the flexibility. However, lowering the industrial sector's output can negatively impact the economy, making industrial demand response a last resort to prevent major outages in the power system [162, 163]. Demand response is often referred to as load shedding and has been used as a flexibility option

for decades. However, due to changes in future power systems, new faster load shedding operations are necessary [164].

Demand response models in PowerFactory are available but require a license. Because of this, a direct load-shedding model is not implemented here; for future work, it can be found in [165]. Rather, the dynamic model used to represent electrolysers was utilized. The electrolyser model has quicker dynamics than the traditional demand response currently available. However, it is assumed that the technical progress in the horizon for all power system domains, with respect to the increased dynamics in the system, also applies to the load response technology. To account for the usage of this technology primarily as a last resort, the frequency wideband, or the amount of frequency deviation needed before the demand response is triggered, is increased. The frequency wideband for the electrolyser model was 0.1 Hz, set to 0.15 Hz for the demand response. Thus, the demand response responds 0.05 Hz later than the electrolysers. This was chosen as an arbitrary number to represent a later involvement in the dynamic response; however, for better accuracy, similar projects in the future should be compared, and the model parameters updated accordingly.

New loads were implemented at the same substations as those already included in the synthetic model. The load structure was not changed but rather scaled according to future load projections in II3050-2 [106]. These loads will be explained in more detail in a later subsection. Nine regular loads were already implemented in the model, and nine new demand response representations were implemented, each with a maximum value of 1.11 GW, totaling 10 GW for the model. The demand response will be set to 10 GW for all simulations, regardless of overall system loading. This assumes that the system has a full capacity for demand response during all operational characteristics. In real-life operations, this could vary depending on the respective agreements between TSOs and the industry and could be modeled in further detail. This could also be used in further analysis to evaluate the effect of load response characteristics on system stability.

The implemented load responses can be seen in Table 6.12.

Table 6.12: Overview of Load Response Projects implemented throughout 2050

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Load_B150_LR	2050	Future	B150	Noord-Brabant	150	1110
Load_F110_LR	2050	Future	F110	Friesland	110	1110
Load_GDO110_LR	2050	Future	GDO110	Groningen	110	1110
Load_GFU150_LR	2050	Future	GFU150	Utrecht	150	1110
Load_L150_LR	2050	Future	L150	Limburg	150	1110
Load_NH150_LR	2050	Future	NH150	Noord-Holland	150	1110
Load_ON110_LR	2050	Future	ON110	Overijssel	110	1110
Load_Z150_LR	2050	Future	Z150	Zeeland	150	1110
Load_ZH150_LR	2050	Future	ZH150	Zuid-Holland	150	1110

6.4.5. Power to Heat

Next, power to heat as a flexibility option is discussed. Similar to Power-to-Gas technology and demand response, power-to-heat utilizes flexible load demand as a resource to provide ancillary services for system stability. For power to heat, this utilization comes from targeting heating infrastructure as a scalable, controllable load [166]. This is achieved by either increasing or decreasing the production used for heating to regulate power imbalances. Most of the Netherlands currently utilizes gas networks as a heating resource for residential and commercial heating infrastructure, but for a sustainable future, the heating supplied through electric power is increasing [167]. Because of this, the possibility of using power to heat as a flexible resource is also growing, and II3050-2 estimates that the national scenario will have 11 GW of power to heat flexible power available in 2050 [106]. Adjusting the power needed to regulate heat results in either excess or too little heat for the users dependent on heat and can result in uneven temperature cycles in the heating networks. Here, the solution comes in the form of thermal inertia [166]. Because heating typically occurs in places where some sort of insulation exists, in the form

of building material or water for boilers, heat does not change immediately but is dependent upon the thermal inertia to respond to changes in temperature [166], similar to inertia in power systems. Because of this, power to heat can still provide flexible resources without significantly impacting the heating network.

Parallels can be drawn to other flexibility resources that try to allow the utilization of resources already in operation, such as trains [168]. Here, by braking the trains, lots of kinetic energy became available for the power system, and by limiting the amount of kinetic energy available, an acceptable delay could be imposed on the train system to provide ancillary services, similar to power to heat resources. II3050-2 assumes that power to heat resources do not need major additional space to be implemented for the high voltage network; however, on a lower scale, the change from natural gas towards alternative heating options will require major infrastructural changes in the heating network [106, 169]. On the high voltage scale, power to heat usually refers to large-scale boilers that provide flexible electrical demand based on similar principles discussed and use the thermal inertia of the boilers and water to store heat that can be utilized as a flexible resource depending on the objective [169].

For future scenarios, it is assumed that dynamic control of the power to heat resource is implemented, allowing for a similar control setup as those of demand response and power to gas. Similar to the demand response, the power to heat was implemented at all the load centers in the system for a total of 11 GW over the 9 locations, or 1.22 GW individually. This allows for an adaptive control system that allows for flexibility modeling; for future systems, this should be updated according to specifics for certain projects. Similar to the demand response, it is estimated that all 11 GW of adjustable power is available during the simulations. This can be utilized for future work to analyze the direct impact of various power-to-heat schemes on power systems' overall stability and oscillatory performance. No hierarchy of the power-to-heat response system is mentioned in II3050-2, and the response was set to use the same bandwidth as the electrolysers, with a response occurring at 0.1 Hz frequency deviation.

The various implemented power-to-heat projects can be seen in Table 6.13.

Table 6.13: Overview of Power To Heat Projects implemented throughout 2050

Name	Year	Project	Terminal	Zone	Voltage (kV)	Capacity (MW)
Load_B150_PTH	2050	Future	B150	Noord-Brabant	150	1110
Load_F110_PTH	2050	Future	F110	Friesland	110	1110
Load_GDO110_PTH	2050	Future	GDO110	Groningen	110	1110
Load_GFU150_PTH	2050	Future	GFU150	Utrecht	150	1110
Load_L150_PTH	2050	Future	L150	Limburg	150	1110
Load_NH150_PTH	2050	Future	NH150	Noord-Holland	150	1110
Load_ON110_PTH	2050	Future	ON110	Overijssel	110	1110
Load_Z150_PTH	2050	Future	Z150	Zeeland	150	1110
Load_ZH150_PTH	2050	Future	ZH150	Zuid-Holland	150	1110

6.5. Loads

Finally, the system loads are discussed. II3050-2 does not mention any specifics regarding a change in the overall load infrastructure in future scenarios [106]. Because of this, the load setup that was already implemented in the model was further utilized. Here, the loads are connected to the lower voltage networks, with 150 and 110 kV voltage ratings. For the national scenario in II3050-2, a peak load of 48 GW was found [106]. Because the loads do not have a power rating but are rather scaled as needed in PowerFactory. The loads were scaled using a percentage scale based on the current load levels in the system, found for the 2030 scenario in previous work [105]. This was done because there is no specific mention of load levels in the different regions for future scenarios in II3050-2. Thus, once more information regarding the load infrastructure becomes available, the model should be updated accordingly. The 11 GW of Power to Heat and 10 GW of load response are also included in the total loading of the system. The different loads will be scaled based on the different loading percentages for the different cases in the simulations.

A total of 27 GW was implemented through regular loads, in addition to the 21 GW of loads represented by load response and power to heat. A total of 26.43 GW was implemented in the 2030 scenario and thus only needs to be scaled by 0.57 GW. For simplicity, the peak load and average load percentage distribution are not accounted for. Thus, the peak load will be scaled down uniformly for different levels.
The new peak demand levels of the different regions can be seen in Table 6.14.

Table 6.14: Peak Demand and share of total demand assumed for each region for 2050

Region	Peak Demand (MW)	Distribution (%)
F110	794.55	2.84
GDO110	1695.04	6.05
ON110	1800.98	6.43
GFU150	4555.43	16.27
Z150	1250.09	4.46
B150	4131.67	14.76
L150	2330.68	8.32
NH150	5826.71	20.81
ZH150	5614.83	20.05
Total	28000	100

6.6. Load Flow Analysis

After the implementation of the different resources and adaptations to the different scenarios, various analyses were conducted to examine network operation and make necessary adjustments. The first was a simple load flow analysis performed for all the scenarios found in Case Study A. Components that were heavily loaded (above 80 %) were upgraded to reduce congestion in the power system. This was done intuitively for all transmission lines and transformers. If there were voltage issues at buses after upgrading the heavily loaded components, reactive power compensation units were placed in adjacent areas to maintain a desired voltage at all buses during usual load flows.

The necessary upgrades to comply with these maximum loadings can be seen in Table 6.15 to Table 6.17. Here, the necessary transformers and transmission lines were added and upgraded. The original values are highlighted in parentheses, and where the "Rated (kA)" is changed, the transmission line type was also upgraded.

Table 6.15: Transformer added to account for the contingency analysis on the 2050 power system

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_EEM380 _EEM220_2	EEM380	EEM220	Groningen	12	9000	750

Table 6.16: Transformers upgraded to account for the load flow analysis on the 2050 power system, red numbers in parenthesis highlight the original values

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_GRTH380 _L150_Future	GRTH380	L150	Limburg	10 (3)	5000 (1500)	500 (500)
Trafo_MBT380 _L150_2	MBT380	L150	Limburg	10 (3)	4500 (1350)	450 (450)
Trafo_MBT380 _L150_1	MBT380	L150	Limburg	5 (1)	2500 (500)	500 (500)
Trafo_BVW380 _NH150	BVW380	NH150	Noord-Holland	20 (4)	20000 (2000)	500 (500)
Trafo_OZN380 _NH150	OZN380	NH150	Noord-Holland	8 (4)	4000 (2000)	500 (500)
Trafo_DIM380 _NH150_1	DIM380	NH150	Noord-Holland	18 (1)	9000 (500)	500 (500)
Trafo_DIM380 _NH150_2	DIM380	NH150	Noord-Holland	20 (3)	9000 (1350)	450 (450)
Trafo_MVL380 _ZH150	MVL380	ZH150	Zuid-Holland	12 (2)	5400 (900)	450 (450)
Trafo_WL380 _ZH150	WL380	ZH150	Zuid-Holland	6 (3)	3000 (1500)	500 (500)
Trafo_SMH380 _ZH150	SMH380	ZH150	Zuid-Holland	5 (2)	2500 (1000)	500 (500)
Trafo_WTR380 _ZH150	WTR380	ZH150	Zuid-Holland	6 (4)	3000 (2000)	500 (500)
Trafo_TNZ380 _Z150_Future	TNZ380	Z150	Zeeland	8 (3)	4000 (1500)	500 (500)
Trafo_BSL380 _Z150	BSL380	Z150	Zeeland	10 (4)	5000 (2000)	500 (500)
Trafo_GT380 _B150	GT380	B150	Noord-Brabant	6 (3)	2700 (1350)	450 (450)
Trafo_DOD380 _GFU150_1	DOD380	GFU150	Utrecht	20 (1)	10000 (500)	500 (500)
Trafo_DOD380 _GFU150_2	DOD380	GFU150	Utrecht	22 (2)	9900 (900)	450 (450)
Trafo_DTC380 _GFU150_1	DTC380	GFU150	Utrecht	30 (1)	15000 (500)	500 (500)
Trafo_DTC380 _GFU150_2	DTC380	GFU150	Utrecht	34 (1)	15300 (450)	450 (450)
Trafo_BMR380 _L150	BMR380	L150	Limburg	5 (1)	2500 (500)	500 (500)
Trafo_BKK380 _GFU150	BKK380	GFU150	Utrecht	14 (1)	7000 (500)	500 (500)
Trafo_LLS380 _GFU150	LLS380	GFU150	Utrecht	6 (2)	3000 (1000)	500 (500)
Trafo_ENS380 _ENS220_1	ENS380	ENS220	Overijssel	16 (1)	12000 (750)	750 (750)
Trafo_ENS380 _ENS220_2	ENS380	ENS220	Overijssel	24 (1)	12000 (750)	500 (500)

Continued on next page

Table 6.16 – continued from previous page

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_ENS380 _ENS220_3	ENS380	ENS220	Overijssel	16 (1)	12000 (750)	750 (750)
Trafo_OHK220 _F110_1	OHK220	F110	Friesland	8 (2)	1600 (400)	200 (200)
Trafo_OHK220 _F110_2	OHK220	F110	Friesland	5 (2)	1850 (740)	740 (740)
Trafo_LSM220 _F110	LSM220	F110	Friesland	8 (3)	1600 (600)	200 (200)
Trafo_BGM220 _F110	BGM220	F110	Friesland	8 (2)	1600 (400)	200 (200)
Trafo_VVL380 _VVL220	VVL380	VVL220	Friesland	26 (2)	19500 (1500)	750 (750)
Trafo_HGL380 _ON110	HGL380	ON110	Overijssel	50 (3)	17500 (1050)	350 (350)
Trafo_ENS380 _NO110	ENS380	ON110	Overijssel	40 (3)	14800 (1110)	370 (370)
Trafo_TIL380 _B150_N	TIL380	B150	Noord-Brabant	8 (4)	4000 (2000)	500 (500)
Trafo_GT380 _B150	GT380	B150	Noord-Brabant	9 (6)	4050 (2700)	450 (450)
Trafo_MEE380 _MEE220	MEE380	MEE220	Groningen	16 (2)	12000 (1500)	750 (750)
Trafo_VHZ380 _ZH150	VHZ380	ZH150	Zuid-Holland	24 (3)	12000 (1500)	500 (500)
Trafo_MSK380 _GD0110	MSK380	GD0110	Groningen	40 (7)	14800 (3150)	370 (370)

Table 6.17: Transmission lines upgraded to account for the load flow analysis on the 2050 power system, red numbers in parenthesis highlight the original values

Name	Terminal i	Terminal j	Zone	Parallel Lines	Total (kA)	Rated (kA)
MBT380_EHV380_Z	MBT380	EHV380	Noord-Brabant	6 (1)	24 (4)	4 (4)
MBT380_EHV380_W	MBT380	EHV380	Noord-Brabant	6 (1)	24 (4)	4 (4)
MBT380_EHV380 _Future	MBT380	EHV380	Noord-Brabant	6 (1)	24 (4)	4 (4)
RLL380_TIL380_N1	RLL380	TIL380	Noord-Brabant	5 (1)	20 (4)	4 (4)
RLL380_TIL380_N1	RLL380	TIL380	Noord-Brabant	5 (1)	20 (4)	4 (4)
HSW220_ENS20_Z	HSW220	ENS220	Friesland	6 (1)	24 (2.5)	4 (2.5)
HSW220_ENS20_W	HSW220	ENS220	Friesland	6 (1)	24 (2.5)	4 (2.5)
HSW220_ZYV220_W	HSW220	ZYV220	Friesland	2 (1)	8 (2.5)	4 (2.5)
HSW220_ZYV220_Z	HSW220	ZYV220	Friesland	2 (1)	8 (2.5)	4 (2.5)
RBB220_WEW220_B	RBB220	WEW220	Groningen	3 (1)	12 (2.3)	4 (2.3)
RBB220_WEW220_G	RBB220	WEW220	Groningen	3 (1)	12 (2.3)	4 (2.3)
EEM220_RBB220_B	EEM220	RBB220	Groningen	5 (1)	20 (2.3)	4 (2.3)
EEM220_RBB220_G	EEM220	RBB220	Groningen	5 (1)	20 (2.3)	4 (2.3)
EEM220_EHA220_W	EEM220	EHA220	Groningen	2 (1)	8 (2.3)	4 (2.3)
EEM220_EHA220_Z	EEM220	EHA220	Groningen	2 (1)	8 (2.3)	4 (2.3)

Continued on next page

Table 6.17 – continued from previous page

Name	Terminal i	Terminal j	Zone	Parallel Lines	Total (kA)	Rated (kA)
EEM380_EOS380_W	EEM380	EOS380	Groningen	2 (1)	8 (4)	4 (4)
EEM380_EOS380_Z	EEM380	EOS380	Groningen	2 (1)	8 (4)	4 (4)
EEM380_EOS380 _Future_3	EEM380	EOS380	Groningen	2 (1)	8 (4)	4 (4)
MEE220_FSO220 _Future_1	MEE220	FSO220	Groningen	1 (1)	4 (2.3)	4 (2.3)
MEE220_FSO220 _Future_2	MEE220	FSO220	Groningen	1 (1)	4 (2.3)	4 (2.3)
DOD380_DTC380_W	DOD380	DTC380	Utrecht	10 (1)	40 (4)	4 (4)
DOD380_DTC380_Z	DOD380	DTC380	Utrecht	10 (1)	40 (4)	4 (4)
DOD380_BMR380_Z	DOD380	BMR380	Utrecht	6 (1)	24 (4)	4 (4)
MBT380_DOD380_W	MBT380	DOD380	Limburg	8 (1)	32 (4)	4 (4)
LLS380_ENS380_W	LLS380	ENS380	Overijssel	7 (1)	28 (4)	4 (4)
LLS380_ENS380_Z	LLS380	ENS380	Overijssel	7 (1)	28 (4)	4 (4)
LLS380_ENS380 _Future_3	LLS380	ENS380	Overijssel	7 (1)	28 (4)	4 (4)
LLS380_ENS380 _Future_4	LLS380	ENS380	Overijssel	7 (1)	28 (4)	4 (4)
LLS380_ALR380 _Future_1	LLS380	ALR380	Overijssel	9 (1)	36 (4)	4 (4)
LLS380_ALR380 _Future_2	LLS380	ALR380	Overijssel	9 (1)	36 (4)	4 (4)
DIM380_ALR380 _Future_1	DIM380	ALR380	Noord-Holland	9 (1)	36 (4)	4 (4)
DIM380_ALR380 _Future_2	DIM380	ALR380	Noord-Holland	9 (1)	36 (4)	4 (4)
OZN380_DIM380_G	OZN380	DIM380	Noord-Holland	7 (1)	28 (4)	4 (4)
OZN380_DIM380_W	OZN380	DIM380	Noord-Holland	7 (1)	28 (4)	4 (4)
DIM380_LLS380 _Future_3	DIM380	LLS380	Utrecht	5 (1)	20 (4)	4 (4)
DIM380_LLS380 _Future_4	DIM380	LLS380	Utrecht	5 (1)	20 (4)	4 (4)
OZN380_BVW380_W	OZN380	BVW380	Noord-Holland	3 (1)	12 (4)	4 (4)
OZN380_BVW380_Z	OZN380	BVW380	Noord-Holland	3 (1)	12 (4)	4 (4)
OZN380_BVW380 _Future_3	OZN380	BVW380	Noord-Holland	3 (1)	12 (4)	4 (4)
BKK380_DIM380_W	BKK380	DIM380	Utrecht	5 (1)	20 (4)	4 (4)
BKK380_DIM380_Z	BKK380	DIM380	Utrecht	5 (1)	20 (4)	4 (4)
MEE380_DIL380_W	MEE380	DIL380	Groningen	2 (1)	8 (2)	4 (2)
MEE380_DIL380_Z	MEE380	DIL380	Groningen	2 (1)	8 (2)	4 (2)
ENS380_VVL380 _Future_1	ENS380	VVL380	Overijssel	3 (1)	12 (4)	4 (4)
ENS380_VVL380 _Future_2	ENS380	VVL380	Overijssel	3 (1)	12 (4)	4 (4)
ENS380_VVL380 _Future_3	ENS380	VVL380	Overijssel	3 (1)	12 (4)	4 (4)
ENS380_VVL380 _Future_4	ENS380	VVL380	Overijssel	3 (1)	12 (4)	4 (4)
ENS380_ZL380_W	ENS380	ZL380	Overijssel	3 (1)	12 (4)	4 (4)
ENS380_ZL380_Z	ENS380	ZL380	Overijssel	3 (1)	12 (4)	4 (4)

Continued on next page

Table 6.17 – continued from previous page

Name	Terminal i	Terminal j	Zone	Parallel Lines	Total (kA)	Rated (kA)
ENS380_ZL380 _Future_3	ENS380	ZL380	Overijssel	3 (1)	12 (4)	4 (4)
ENS380_ZL380 _Future_4	ENS380	ZL380	Overijssel	3 (1)	12 (4)	4 (4)

6.7. Contingency Analysis

Finally, a contingency analysis was performed for all the different scenarios to ensure the power system remains functional in the event of outages at any location. A contingency analysis consisting of an $N-2$ analysis was done for all system components and scenarios. As before, all components heavily loaded (now above 90 %) for these scenarios were also upgraded to maintain system operation during different outages. The selection of an $N-2$ contingency was chosen as this is the same stipulation that the current 380 kV ring was designed after, thus maintaining a similar level of robustness in the future power systems as for current systems [170].

While this type of contingency analysis, defined as a deterministic analysis, has been the typical approach for planning and building power systems, new methods have been discussed as alternatives for future systems due to deterministic planning often leading to over-designed power systems while simultaneously neglecting risk and economics [170]. Some of these new methods include probabilistic methods that take into account the probability of certain failures in the analysis, methods that combine deterministic and probabilistic approaches, and also new deterministic methods that include economic factors [171, 172, 173, 174]. Thus, using a traditional deterministic $n-2$ contingency analysis might not be the ideal representation of the future system. However, because this is the method used to build the current 380 kV network, it is the best current approach to accurately represent how the future power system could look. This could be altered for future work if the design characteristic is changed from the traditional one.

The necessary upgrades of the system can be seen in Table 6.18 and Table 6.19. Similar to the load flow, the necessary transformers and transmission lines. The original values are highlighted in parentheses, and where the "Rated (kA)" is changed, the transmission line type was also upgraded.

Table 6.18: Transformers upgraded to account for the contingency analysis on the 2050 power system, red numbers in parenthesis highlight the original values

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_BMR380 _L150	BMR380	L150	Limburg	25 (5)	1250 (2500)	500 (500)
Trafo_BVW380 _NH150	BVW380	NH150	Noord-Holland	35 (8)	12500 (5000)	500 (500)
Trafo_OZN380 _NH150	OZN380	NH150	Noord-Holland	20 (8)	17500 (5000)	500 (500)
Trafo_BGM220 _F110	BGM220	F110	Friesland	20 (8)	4000 (1600)	200 (200)
Trafo_LSM220 _F110	BGM220	F110	Friesland	20 (8)	4000 (1600)	200 (200)
Trafo_OHK220 _F110_1	OHK220	F110	Friesland	20 (8)	4000 (1600)	200 (200)
Trafo_OHK220 _F110_2	OHK220	F110	Friesland	10 (5)	3700 (1850)	370 (370)

Continued on next page

Table 6.18 – continued from previous page

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_DIM380 _NH150_1	DIM380	NH150	Noord-Holland	25 (5)	12500 (2500)	500 (500)
Trafo_DIM380 _NH150_2	DIM380	NH150	Noord-Holland	26 (6)	11700 (2700)	250 (250)
Trafo_VVL220 _GDO110_1	VVL220	GDO110	Groningen	10 (1)	3700 (370)	370 (370)
Trafo_VVL220 _GDO110_2	VVL220	GDO110	Groningen	18 (1)	3600 (200)	200 (200)
Trafo_RBB220 _GD110_new	RBB220	GDO110	Groningen	16 (2)	5920 (740)	370 (370)
Trafo_HGL380 _ON110	HGL380	ON110	Overijssel	25 (8)	8750 (2800)	350 (350)
Trafo_LLS380 _GFU150	LLS380	GFU150	Utrecht	30 (6)	15000 (3000)	500 (500)
Trafo_WEW220 _GDO110	RBB220	GDO110	Groningen	15 (2)	5550 (740)	370 (370)
Trafo_EHV380 _B150_1	EHV380	B150	Noord-Braabant	6 (1)	3000 (500)	500 (500)
Trafo_EHV380 _B150_2	EHV380	B150	Noord-Braabant	7 (3)	3150 (1350)	450 (450)
Trafo_TIL380 _B150_N	TIL380	B150	Noord-Braabant	10 (8)	5000 (4000)	500 (500)
Trafo_GT380 _B150	GT380	B150	Noord-Braabant	12 (10)	5400 (4500)	450 (450)
Trafo_HST380 _B150	HST380	B150	Noord-Braabant	12 (10)	5400 (4500)	450 (450)
Trafo_BKK380 _GFU150	BKK380	GFU150	Utrecht	40 (7)	20000 (3500)	500 (500)
Trafo_MBT380 _L150_1	MBT380	L150	Limburg	7 (5)	3500 (2500)	500 (500)
Trafo_ENS380 _NO110	ENS380	NO110	Overijssel	27 (6)	9990 (2220)	370 (370)
Trafo_ZYV220 _GD110	ZYV220	GDO110	Groningen	4 (2)	800 (400)	200 (200)
Trafo_KIJ380 _ZH150_1	KIJ380	ZH150	Zuid-Holland	3 (1)	1500 (500)	500 (500)
Trafo_EEM380 _EEM220	EEM380	EEM220	Groningen	12 (3)	9000 (2250)	750 (750)
Trafo_MSK380 _GDO110_Future	MSK380	GDO110	Groningen	27 (6)	9990 (2220)	370 (370)
Trafo_MEE220 _GDO110	MEE220	GDO110	Groningen	15 (3)	5550 (1110)	370 (370)
Trafo_GRTH380 _L150_Future	GRTH380	L150	Limburg	11 (10)	5500 (5000)	500 (500)
Trafo_TNZ380 _Z150 _Future	TNZ380	ZH150	Zeeland	10 (8)	5000 (4000)	500 (500)

Continued on next page

Table 6.18 – continued from previous page

Name	HV Terminal	LV Terminal	Zone	Parallel number of Trafos	S_total (MVA)	S_rated (MVA)
Trafo_HSW220 _GDO110_1	HSW220	GDO110	Groningen	20 (1)	7400 (370)	370 (370)
Trafo_HSW220 _GDO110_2	HSW220	GDO110	Groningen	20 (2)	7000 (700)	350 (350)
Trafo_ZYV220 _GD110	ZYV220	GDO110	Groningen	25 (2)	5000 (400)	200 (200)
Trafo_VVL220 _GDO110_1	VVL220	GDO110	Groningen	5 (1)	1850 (370)	370 (370)
Trafo_VVL220 _GDO110_2	VVL220	GDO110	Groningen	9 (2)	1800 (370)	200 (200)
Trafo_BWK380 _ZH150	BWK380	ZH150	Zuid-Holland	12 (4)	6000 (2000)	500 (500)
Trafo_KIJ380 _ZH150_1	KIJ380	ZH150	Zuid-Holland	6 (3)	3000 (1500)	500 (500)
Trafo_KIJ380 _ZH150_2	KIJ380	ZH150	Zuid-Holland	7 (3)	3150 (1350)	450 (450)
Trafo_EEM380 _EEM220_1	EEM380	EEM220	Groningen	16 (8)	12000 (6000)	750 (750)
Trafo_EEM380 _EEM220_2	EEM380	EEM220	Groningen	16 (8)	12000 (6000)	750 (750)

Table 6.19: Transmission lines upgraded to account for the contingency analysis on the 2050 power system, red numbers in parenthesis highlight the original values

Name	Terminal i	Terminal j	Zone	Parallel Lines	Total (kA)	Rated (kA)
OHK220_ENS220_W	OHK220	ENS220	Friesland	7 (1)	28 (2.5)	4 (2.5)
OHK220_ENS220_Z	OHK220	ENS220	Friesland	7 (1)	28 (2.5)	4 (2.5)
VVL220_EEM220	VVL220	EEM220	Groningen	4 (1)	16 (2.3)	4 (2.3)
MBT380_BMR380_Z	MBT380	BMR380	Limburg	10 (1)	40 (4)	4 (4)
KIJ380_GT380_W	KIJ380	GT380	Noord-Brabant	3 (1)	12 (4)	4 (4)
KIJ380_GT380_Z	KIJ380	GT380	Noord-Brabant	3 (1)	12 (4)	4 (4)
KIJ380_GT380 _Future_3	KIJ380	GT380	Noord-Brabant	3 (1)	12 (4)	4 (4)
KIJ380_GT380 _Future_4	KIJ380	GT380	Noord-Brabant	3 (1)	12 (4)	4 (4)
KIJ380_BKK380 _Future_2	KIJ380	BKK380	Noord-Brabant	7 (1)	28 (4)	4 (4)
KIJ380_BKK380_W	KIJ380	BKK380	Noord-Brabant	7 (1)	28 (4)	4 (4)
EOS380_VVL380_1	EOS380	VVL380	Groningen	4 (1)	16 (4)	4 (4)
EOS380_VVL380_2	EOS380	VVL380	Groningen	4 (1)	16 (4)	4 (4)
LSM220_OHK220_W	LSM220	OHK220	Friesland	4 (1)	16 (2.5)	4 (2.5)
LSM220_OHK220_Z	LSM220	OHK220	Friesland	4 (1)	16 (2.5)	4 (2.5)
LSM220_BGM220_W	LSM220	BGM220	Friesland	2 (1)	8 (2.5)	4 (2.5)
LSM220_BGM220_Z	LSM220	BGM220	Friesland	2 (1)	8 (2.5)	4 (2.5)
LSM220_BGM220_a	LSM220	BGM220	Friesland	2 (1)	8 (2.5)	4 (2.5)
MSK380_MEE380 _Future_1	MSK380	MEE380	Groningen	5 (1)	20 (4)	4 (4)

Continued on next page

Table 6.19 – continued from previous page

Name	Terminal i	Terminal j	Zone	Parallel Lines	Total (kA)	Rated (kA)
MSK380_MEE380 _Future_2	MSK380	MEE380	Groningen	5 (1)	20 (4)	4 (4)
GT380_MDK380 _Future_1	GT380	MDK380	Zeeland	2 (1)	8 (4)	4 (4)
GT380_MDK380 _Future_2	GT380	MDK380	Zeeland	2 (1)	8 (4)	4 (4)
GT380_MDK380 _Future_3	GT380	MDK380	Zeeland	2 (1)	8 (4)	4 (4)
EEM380_MEE380_W	EEM380	MEE380	Groningen	5 (1)	20 (4)	4 (4)
EEM380_MEE380_Z	EEM380	MEE380	Groningen	5 (1)	20 (4)	4 (4)
DTC380_HGL380_W	DTC380	HGL380	Utrecht	10 (1)	40 (4)	4 (4)
DTC380_HGL380_Z	DTC380	HGL380	Utrecht	10 (1)	40 (4)	4 (4)
ZL380_VOB380 _Future_1	ZL380	VOB380	Overijssel	7 (1)	28 (4)	4 (4)
ZL380_VOB380 _Future_2	ZL380	VOB380	Overijssel	7 (1)	28 (4)	4 (4)
VOB380_MSK380 _Future_1	VOB380	MSK380	Groningen	7 (1)	28 (4)	4 (4)
VOB380_MSK380 _Future_2	VOB380	MSK380	Groningen	7 (1)	28 (4)	4 (4)
FSO220_WEW220 _Future_1	FSO220	WEW220	Groningen	2 (1)	8 (2.3)	4 (2.3)
FSO220_WEW220 _Future_2	FSO220	WEW220	Groningen	2 (1)	8 (2.3)	4 (2.3)
ZL380_HGL380_W	ZL380	HGL380	Overijssel	5 (1)	20 (4)	4 (4)
ZL380_HGL380_Z	ZL380	HGL380	Overijssel	5 (1)	20 (4)	4 (4)
ZL380_HGL380 _Future_3	ZL380	HGL380	Overijssel	5 (1)	20 (4)	4 (4)
ZL380_HGL380 _Future_4	ZL380	HGL380	Overijssel	5 (1)	20 (4)	4 (4)
KIJ380_OZN380_Z	KIJ380	OZN380	Noord-Holland	4 (1)	16 (4)	4 (4)
KIJ380_OZN380 _Future_2	KIJ380	OZN380	Noord-Holland	4 (1)	16 (4)	4 (4)
MEE220_FSO220 _Future_1	MEE220	FSO220	Groningen	4 (1)	16 (4)	4 (4)
MEE220_FSO220 _Future_2	MEE220	FSO220	Groningen	4 (1)	16 (4)	4 (4)
BWK380_VHZ380_O	BWK380	VHZ380	Zuid-Holland	3 (1)	12 (3)	4 (3)
BWK380_VHZ380_P	BWK380	VHZ380	Zuid-Holland	3 (1)	12 (3)	4 (3)
BVW380_VHZ380_W	BVW380	VHZ380	Noord-Holland	3 (1)	12 (4)	4 (4)
BVW380_VHZ380_Z	BVW380	VHZ380	Noord-Holland	3 (1)	12 (4)	4 (4)
BVW380_VHZ380 _Future_3	BVW380	VHZ380	Noord-Holland	3 (1)	12 (4)	4 (4)
TIL380_EHV380_N1	TIL380	EHV380	Noord-Brabant	3 (1)	12 (4)	4 (4)
TIL380_EHV380_N2	TIL380	EHV380	Noord-Brabant	3 (1)	12 (4)	4 (4)
TIL380_EHV380_N3	TIL380	EHV380	Noord-Brabant	3 (1)	12 (4)	4 (4)

Continued on next page

Table 6.19 – continued from previous page

Name	Terminal i	Terminal j	Zone	Parallel Lines	Total (kA)	Rated (kA)
TIL380_EHV380 _Future_4	TIL380	EHV380	Noord-Brabant	3 (1)	12 (4)	4 (4)
TIL380_EHV380 _Future_5	TIL380	EHV380	Noord-Brabant	3 (1)	12 (4)	4 (4)
EEM380_EOS380_W	EEM380	EOS380	Groningen	3 (1)	12 (4)	4 (4)
EEM380_EOS380_Z	EEM380	EOS380	Groningen	3 (1)	12 (4)	4 (4)
EEM380_EOS380 _Future_3	EEM380	EOS380	Groningen	3 (1)	12 (4)	4 (4)
KIJ380_BWK380_Z	KIJ380	BWK380	Zuid-Holland	3 (1)	12 (4)	4 (4)
KIJ380_BWK380_W	KIJ380	BWK380	Zuid-Holland	3 (1)	12 (4)	4 (4)
BSL380_RLL380_G	BSL380	RLL380	Zeeland	3 (1)	12 (4)	4 (4)
BSL380_RLL380_Z	BSL380	RLL380	Zeeland	3 (1)	12 (4)	4 (4)
BSL380_RLL380_N1	BSL380	RLL380	Zeeland	3 (1)	12 (4)	4 (4)
BSL380_RLL380_N2	BSL380	RLL380	Zeeland	3 (1)	12 (4)	4 (4)
BSL380_RLL380 _Future_5	BSL380	RLL380	Zeeland	3 (1)	12 (4)	4 (4)
BGM220_VVL220_Z	BGM220	VVL220	Friesland	4 (1)	16 (4)	4 (4)
BGM220_VVL220_W	BGM220	VVL220	Friesland	4 (1)	16 (4)	4 (4)
VVL220_ZYV220_W	VVL220	ZYV220	Groningen	3 (1)	8 (4)	4 (4)
VVL220_ZYV220_Z	VVL220	ZYV220	Groningen	3 (1)	8 (4)	4 (4)
MBT380_SDF380_Z	MBT380	SDF380	Limburg	2 (1)	8 (2.6)	4 (2.6)
MBT380_OBZ380_W	MBT380	SDF380	Limburg	2 (1)	8 (2.6)	4 (2.6)
GT380_TIL380_N1	GT380	TIL380	Noord-Brabant	2 (1)	8 (4)	4 (4)
GT380_TIL380_N2	GT380	TIL380	Noord-Brabant	2 (1)	8 (4)	4 (4)
GT380_TIL380_N3	GT380	TIL380	Noord-Brabant	2 (1)	8 (4)	4 (4)
GT380_TIL380 _Future_4	GT380	TIL380	Noord-Brabant	2 (1)	8 (4)	4 (4)
GT380_TIL380 _Future_5	GT380	TIL380	Noord-Brabant	2 (1)	8 (4)	4 (4)
TIL380_MDK380 _Future_1	TIL380	MDK380	Zeeland	2 (1)	8 (4)	4 (4)
TIL380_MDK380 _Future_2	TIL380	MDK380	Zeeland	2 (1)	8 (4)	4 (4)
CST380_KIJ380_W	CST380	KIJ380	Zuid-Holland	2 (1)	8 (4)	4 (4)
CST380_KIJ380_Z	CST380	KIJ380	Zuid-Holland	2 (1)	8 (4)	4 (4)
CST380_KIJ380 _Future_3	CST380	KIJ380	Zuid-Holland	2 (1)	8 (4)	4 (4)
MBT380_VYK380_W	MBT380	VYK380	Limburg	1 (1)	4 (2.5)	4 (2.5)
MBT380_VYK380_Z	MBT380	VYK380	Limburg	1 (1)	4 (2.5)	4 (2.5)
RLL380_GT380_Z	RLL380	GT380	Noord-Brabant	3 (1)	12 (4)	4 (4)
RLL380_GT380_W	RLL380	GT380	Noord-Brabant	3 (1)	12 (4)	4 (4)
RLL380_GT380 _Future_3	RLL380	GT380	Noord-Brabant	3 (1)	12 (4)	4 (4)
EOS380_OOP380 _Future_1	EOS380	OOP380	Groningen	2 (1)	8 (4)	4 (4)
EOS380_OOP380 _Future_2	EOS380	OOP380	Groningen	2 (1)	8 (4)	4 (4)
EOS380_OOP380 _Future_3	EOS380	OOP380	Groningen	2 (1)	8 (4)	4 (4)

Continued on next page

Table 6.19 – continued from previous page

Name	Terminal i	Terminal j	Zone	Parallel Lines	Total (kA)	Rated (kA)
EOS380_OOP380 _Future_4	EOS380	OOP380	Groningen	2 (1)	8 (4)	4 (4)
WTR380_BWK380_W	WTR380	BWK380	Zuid-Holland	2 (1)	8 (4)	4 (4)
WTR380_BWK380_Z	WTR380	BWK380	Zuid-Holland	2 (1)	8 (4)	4 (4)
DOD380_BMR380_W	DOD380	BMR380	Limburg	12 (1)	48 (4)	4 (4)

The resulting Synthethic model, fully updated and utilized for future studies, can be seen in Figure 6.2.

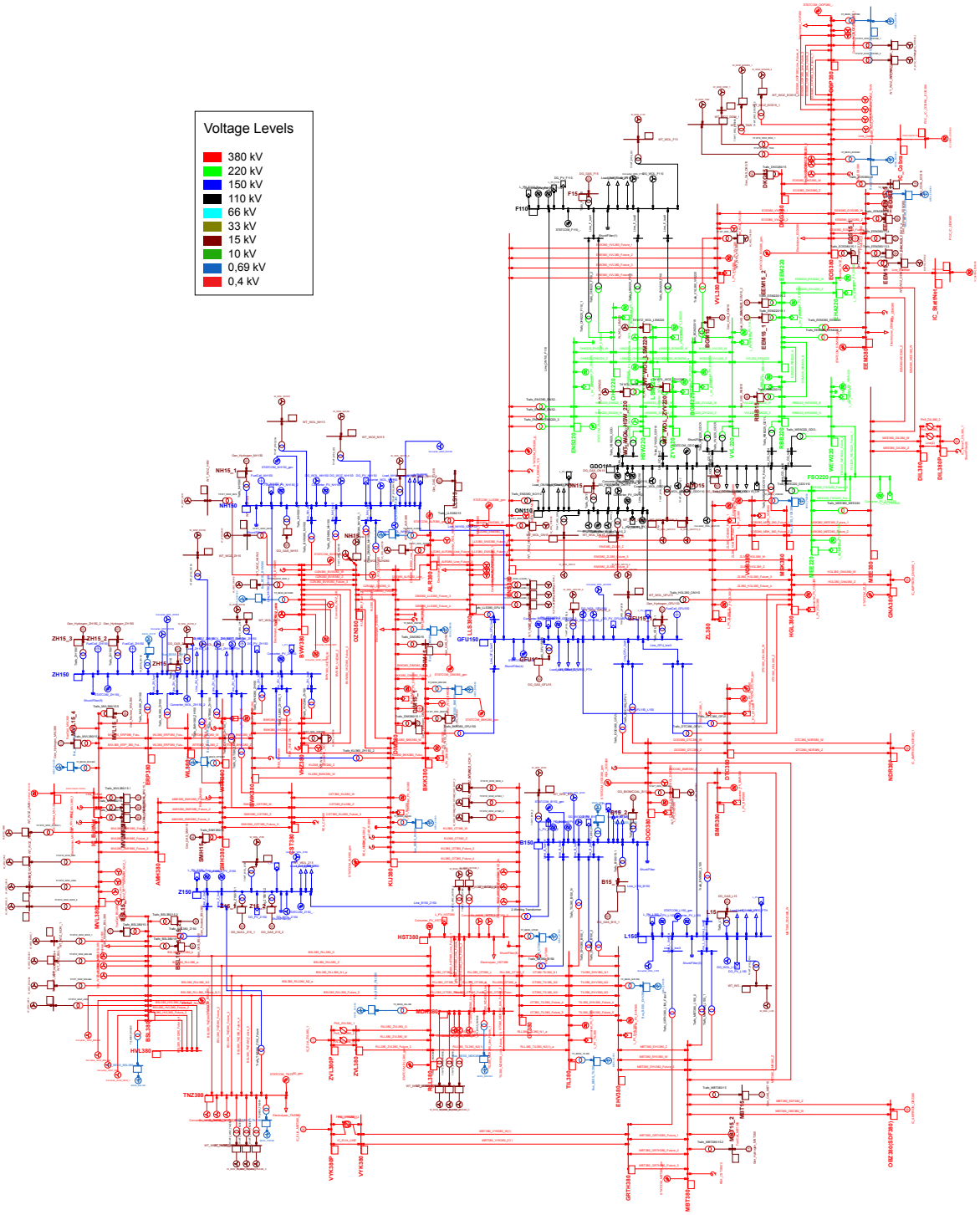


Figure 6.2: Overview of the complete Synthetic model highlighting the different voltage levels.

7

Case Studies

This chapter discusses the three case studies done utilizing the synthetic dynamic model. Each case study follows a traditional research structure, including an introduction, methodology, results, and discussion. This approach facilitates easier adaptation for publication in international journals and conferences.

First, the results of the case studies are discussed to address the specific contributions to the research questions derived from the literature review. Then, the complete details of the different case studies are provided.

7.1. Discussion of Research Questions

The research scope of the thesis included three different research questions. These questions were based on the literature review, highlighting areas that needed further research. Here, they will be discussed directly in relation to the results found in the different case studies.

1. Under what operating conditions and disturbances can the stability of fast-active power balancing be compromised in theoretical future power systems? The different renewable generation shares of onshore wind, offshore wind, and solar PV were analyzed in Case Study A, yielding several interesting insights regarding power system stability. Overall, it was evident that the power flows resulting from the different generation topologies due to the various renewable resources did not significantly impact system stability. The synthetic model's adherence to an N-2 contingency analysis ensured minimal impact due to system loading. Almost no differentiation was found concerning the large shares of different renewable resources. Generally, the different flexibility resources were distributed throughout the system based on their relevance and compatibility with the various renewable generation units, resulting in an efficiently allocated system.

However, there were small deviations in system response among the different cases for the various events. The different events demonstrated varied responses depending on the renewable generation shares, indicating some impact from the differing load flow patterns. Nonetheless, these impacts were minimal due to the system's effective contingency planning. If a different contingency planning approach is used in future systems, this could potentially have a larger impact.

The largest frequency deviations among the different generation shares were identified in cases with the maximum levels of solar PV; similar results were found for the ROCOF values. In these cases, offshore wind generation was set to 0 %, indicating no ancillary service provision from these resources. This differed from other cases, which had generation from all three resources. It was evident that system performance regarding frequency stability was worse when ancillary services from these generation units were unavailable. While somewhat self-explanatory, this highlights the importance of accounting for the variability in renewable generation sources. If there are times in the system with low or no

input from some of the resources, dynamic stability can be severely impacted. This underscores the importance of proper forecasting for these resources and leveraging them so that ancillary services are available when needed, even with the variability of these resources as a limiting factor. This can, for example, be achieved through flexible storage systems in combination with these resources, especially in systems with a high penetration of variable renewable generation sources.

It is worth noting that these case studies were conducted using scenario estimations accounting for the highest degree of electrification. This results in a larger share of generation and flexibility resources, which could reduce the impact of variability on system performance as the likelihood of insufficient generation is reduced. The system also depends on various synchronous generation resources, such as hydrogen and nuclear power plants, as well as multiple interconnections, leveraging possibilities and ancillary services in certain cases.

Overall, due to the system's strength, different renewable generation shares did not dramatically impact frequency stability, and the system performed well in all cases. However, the study still highlights the importance of ancillary services and the intrinsic nature of these resources due to their variability.

2. What are the implications of varying penetration levels, structures, and parameterizations of grid-forming converters on the fast-active power balancing in power systems? The impact of different grid-forming controller types on the frequency stability of the system was dependent on the perturbations, as investigated in Case Study B. Overall, the three controllers—Virtual Synchronous Machine (VSM), Synchroverter, and Droop controller—exhibited almost identical responses for the load and outage events. While results from [76] comparing these controllers for a two-area, four-generator system indicated performance differences for load increases, these differences were negligible in both outage and load increase scenarios in this large-scale system study. This suggests that the overall differences among grid-forming controllers are minimized during such perturbations in large-scale systems. This was consistent for the different controllers' maximum deviation and the ROCOF values, with only minimal deviations between cases. These minor deviations slightly favored the VSM and Droop controllers. This conclusion held true for the controller comparison across all three grid-forming penetration levels of 25 %, 50 %, and 85 %.

However, comparing controller performance during a short circuit yielded different results. The Synchroverter demonstrated the poorest overall performance, showing the largest frequency deviations and ROCOF values across all three penetration levels. Additionally, with an 85 % grid-forming penetration level, the system could not converge simulations during the short-circuit event when using the Synchroverter, indicating an inability to handle the short-circuit event effectively. This issue was not observed with either the VSM or Droop controller. To make the simulations converge for the 85 % Synchroverter case, the maximum allowed current had to be increased from 1.2 to 1.3 pu. Despite this adjustment, the Synchroverter still exhibited the highest frequency deviations and ROCOF values, indicating inferior short-circuit performance. This short-circuit performance issue with the Synchroverter warrants further study to improve its current layout. The Droop and VSM controllers generally had similar maximum deviation and ROCOF values, with slight differences in settling time and oscillatory response, favoring the VSM.

Comparing the controllers regarding oscillatory performance and the damping of the system's most critical modes yielded results similar to the outage and load event analysis for frequency stability. No significant differences were found in the damping capabilities of the different controllers, consistent across all three grid-forming penetration levels. Although performance differences in damping were identified in [76], these differences did not significantly impact large-scale system performance.

Overall, the different grid-forming controller types exhibited similar performance for both frequency and damping capabilities. Differences were evident for the short-circuit perturbations, with the Synchroverter showing a poorer overall short-circuit capability than the VSM and Droop controllers. This was consistent across all three grid-forming penetration levels.

Furthermore, the different grid-forming converter penetration levels of 25 %, 50 %, and 85 %, analyzed in Case Study B, were assessed for all three controllers. Here, similar results between controllers were observed across all perturbations and penetration levels, except for the short-circuit event at the highest penetration level for the Synchroverter.

Overall, higher grid-forming penetration levels showed improved stability in terms of both frequency and damping performance. The results demonstrated lower overall frequency deviations and ROCOF values for higher penetration levels, with the worst performance observed at the lowest penetration level. This was consistent across all three perturbation events, except for the Synchroverter at the highest penetration level. The higher penetration levels exhibited better frequency performance, with lower maximum deviations, reduced ROCOF values, and better settling times. This indicates that, in terms of frequency stability, greater grid-forming penetration levels result in better performance due to the ancillary services provided by grid-forming converters, especially their fast-active power response capabilities. This was true for all three grid-forming controllers.

A similar trend was observed for the system's damping performance, particularly for inter-area oscillations. Higher penetration levels resulted in increased damping ratios for the inter-area modes. One inter-area mode exhibited a slightly lower damping ratio, approximately 0.18 %, while the highest improved damping ratio was around 3.5 %. Four of the five identified inter-area modes showed improved damping ratios, indicating overall enhanced damping performance for higher penetration levels. The damping improvement was evident for both higher penetration levels compared to the lowest. However, a local mode experienced a significant reduction in damping ratio at the highest penetration level. This mode had a damping ratio of around 14 % for the 25 % and 50 % penetration levels, which reduced to approximately 6 % for the 85 % level. This occurred for all three controllers. The reduced damping was linked to one of the grid-following onshore wind converters. When this converter was set out of service, the lowered damping did not appear in any other grid-following converters, suggesting that the local mode was influenced by the grid-following converter's location and the system's oscillatory behavior. Although the mode did not propagate to other system areas, it highlights that high grid-forming penetration levels can cause oscillatory damping issues in large-scale power systems, depending on factors such as topology and resource location. While no definitive conclusions can be drawn from this mode's analysis, it is an interesting finding that should be considered in future research to better understand the intrinsic nature of high grid-forming penetrated power systems.

Overall, higher grid-forming penetration levels showed increased damping and frequency stability performance in the system. However, while the highest penetration levels exhibited enhanced performance in most metrics, they also revealed some inherent weaknesses. This was evident through the occurrence of reduced damping in one local mode and the lack of short-circuit capability at the 85 % penetration level combined with the Synchroverter controller. Thus, while the overall positive impact of high grid-forming converter penetration is significant, some challenges arise that warrant further study.

The main controller parameters of the three grid-forming controllers underwent a sensitivity analysis to determine the parameterization impact on system stability, both for frequency and damping performance. Two parameters were analyzed for the VSM (D_p, T_a), two for the Droop (mp, m_q), and four for the Synchroverter (D_p, T_a, K_q, D_q). These parameters were increased and decreased relative to their base values and simulated for all three levels. This was done in Case Study B with a 50 % grid-forming penetration level, done for a load increase, analyzing the frequency stability and damping of the system.

A similar impact was found for the D_p of the VSM and Synchroverter as the mp for the Droop controller. In PowerFactory, the mp parameter is modeled as an inverse droop, thus having a direct opposite response to an increase or decrease. An increase in D_p and a decrease in mp significantly improved the allowed maximum frequency deviation and ROCOF. This resulted in an approximately fivefold decrease in the maximum allowed frequency between the original and new levels, with improved ROCOF values as well. The results showed an improved initial frequency deviation, quicker settling, and lower oscillatory behavior during the simulation window. Conversely, the lowered D_p and increased mp resulted in worse oscillatory performance, showing the worst performance in all calculation metrics.

The damping performance also showed improvement with increased D_p and decreased mp values. Four out of the five inter-area modes had improved damping ratios, ranging up to an increase of 8 %, with a decrease in one mode of around 0.2 %. Overall, the inter-area damping in the system significantly improved with new parameter values. Conversely, reduced D_p and increased mp values led to decreased system performance. Interestingly, the decreased D_p levels resulted in multiple controller interaction modes with reduced damping ratios for the grid-forming controllers, which were not found in the original or increased parameter values, showing critical damping levels. This was not true for the Droop controller's decreased mp values. No similar modes were found, although the full range of possible mp values was not conducted, and thus, the study cannot guarantee similar results for lower levels. The results indicate that controller parameter tuning can positively and negatively impact system stability, highlighting the potential for dynamic controller parameter methods and the risks of poorly adjusted parameters.

No significant deviation in system performance was found for the T_a parameter value analysis for the VSM and Synchroverter. Some small differences were observed for the damping of higher-speed oscillatory modes, favoring higher T_a values.

Finally, the K_q and D_q parameters of the Synchroverter and the m_q of the Droop controller had no impact on system performance, indicating a successfully divided active and reactive power control loop in the controllers. This highlights the potential for further studies without concerns that these parameters will impact these aspects of power system performance.

Overall, the different parameter analyses showed significant variation in system performance. Some parameters had a major impact on both frequency and system damping, while others had a negligible effect. Parameters with a high impact on system performance highlight the possibilities for future control systems with the potential to adjust controller performance dynamically, allowing for adaptable control schemes. Conversely, parameters with negligible impact allow for further research with confidence that these parameters do not affect this section of power system performance.

3. How do theoretical projections of future inertia constants and kinetic energy levels impact the fast-active power balancing stability of future systems? To analyze the impact of future inertia levels on system stability, Case Study C utilized the AC synchronous machine representations in the synthetic model to represent different kinetic energy and inertia constant levels representing interconnected continental Europe. Three values of inertia constants (4.1 s, 2.25 s, 0.5 s) and three different values of kinetic energy (1800 GVAs, 1025 GVAs, 250 GVAs) were utilized for the study. This represented future predictions for continental Europe towards 2040 and covered both values' predicted range. This analysis was conducted for both frequency and damping performance, using a load increase perturbation.

The results showed that higher inertia constants (4.1 s, 2.25 s) provided a more stable system response with better damping. This generally resulted in slower responses but reduced oscillatory behavior, as indicated by the better settling of the frequency after the perturbation. High inertia constants combined with high kinetic energy levels improved system stability. There was minimal difference in the maximum frequency deviation across the different levels. For the damping of the different modes, the high inertia constants allowed the inter-area modes to have a damping ratio within their respective limits, with no significant impact found for the other modes in the system.

A faster system response was evident for the low inertia constant (0.5 s) but with reduced damping. This generally resulted in increased oscillatory behavior, exhibiting critical stability issues. The system was capable of faster frequency changes, but the response typically resulted in more sustained, quicker oscillations. The damping of the different modes became critical for the low inertia constant simulations. In one simulation, combining the 0.5 s inertia constant with the 1025 GVAs kinetic energy, one of the inter-area modes had a negative real eigenvalue, indicating system instability. The inter-area modes had positive real eigenvalues for the other kinetic energy values, but the damping ratios were well below the stability limits, indicating a critical stability limit.

The highest kinetic energy value showed the overall best system performance. These levels exhibited better maximum allowed frequency deviation and ROCOF values, with slower settling times due to the larger rotational forces in the system. The middle kinetic energy levels showed slightly reduced stability compared to the higher levels, with a similar response pattern but increased oscillatory behavior. The lowest kinetic energy levels resulted in faster-acting oscillations in the system response, leading to larger ROCOF and maximum deviation levels.

Interestingly, the lowest kinetic energy levels had the best system damping. Here, the inter-area modes' oscillatory speed increased, rendering them outside the traditional definition of inter-area modes but with similar participation factors. The overall oscillatory modes of the system changed significantly, allowing the fast-acting grid-forming controllers to dampen the system more effectively due to their compatible fast-acting nature. This indicates that system damping improved for the lower kinetic energy levels, suggesting an intrinsic relationship between frequency stability and damping for these projected levels.

Overall, the different inertia levels projected for continental Europe towards 2040 indicate a complex impact on system stability for the future power system. While the frequency analysis favored the high kinetic energy levels, the damping performance was better for the low kinetic energy levels. At the same time, the middle kinetic energy levels exhibited an unstable system for the lowest inertia constant level and poorly dampened modes for the other inertia constants compared to the other kinetic energy levels. This suggests that future kinetic energy and inertia constant levels will significantly impact power system stability, necessitating further research and policy guidance to ensure safe levels for system operation. The good damping performance of the lowest kinetic energy levels indicates that a fast-acting natured system can thrive under the right circumstances. Future research should investigate the impact of grid-forming penetration and controller parameter values on these different projections.

7.2. Case Study A

7.2.1. Introduction

Case Study A investigates the upgraded dynamic model developed in DIgSILENT PowerFactory, tailored for dynamic studies of the Dutch power system projected for 2050. The model's enhancement enables the evaluation of the national scenario as outlined in [106], which is also envisioned in TenneT's Target Grid for 2045 [114]. Detailed descriptions of the scenario setup are provided in chapter 4, with specific model modifications documented in chapter 5 and 6. This study assesses the system's response to three disturbances: an outage, short-circuit, and load event across six renewable generation and demand cases.

The primary objective of this case study is to validate the synthetic Dutch model created and to showcase its applicability for various analyses. The model's robustness is tested using the extensive functionalities available in DIgSILENT PowerFactory, complemented by tools in Python and Excel. Each of the six generation and demand cases is subjected to three types of disturbances, which are used to scrutinize and delineate the impact of these configurations on system performance. System performance is examined by analyzing the frequency response to each disturbance, supplemented by an eigenvalue analysis for a deeper understanding of the system's oscillatory performance during a load event.

A significant focus of the study is the role of grid-forming converters in enhancing system stability and performance. These converters are critical for maintaining stability in future power systems with high levels of renewable integration. Their performance is critically evaluated in the context of large-scale systems under varied disturbances, providing essential insights into their potential contributions. The efficacy of these converters in damping different inter-area modes is also highlighted, pointing to promising avenues for future enhancements. Further details on grid-forming converters control and their implementation in PowerFactory are presented in section 2.2.

This case study sets the stage for a comprehensive evaluation that serves as a regulatory benchmark concerning stability and the regulatory framework for future systems. The cases explored offer varied generation and demand levels, different sources of generation, and the impact of grid-forming converters on system dynamics. Given the evolving nature of system characteristics, which will be shaped by future research and regulatory assessments, this case study provides a solid foundation for ongoing and future analyses enabled by the synthetic model and its extensive ancillary services.

7.2.2. Methodology

In Case Study A, the dynamic model of the Dutch power system, projected for the year 2050, was employed to simulate and analyze the stability under various cases characterized by distinct levels of generation, demand, and flexibility. The comprehensive flowchart illustrating the system setup is presented in Figure 7.1. The system configuration relied heavily on Python scripts and Excel configurations, detailed in Appendix C, which collectively orchestrated the operational system. These tools were utilized to integrate a diverse array of system characteristics, including the proportion of various renewable energy resources, flexibility options, and the penetration levels of grid-forming converters.

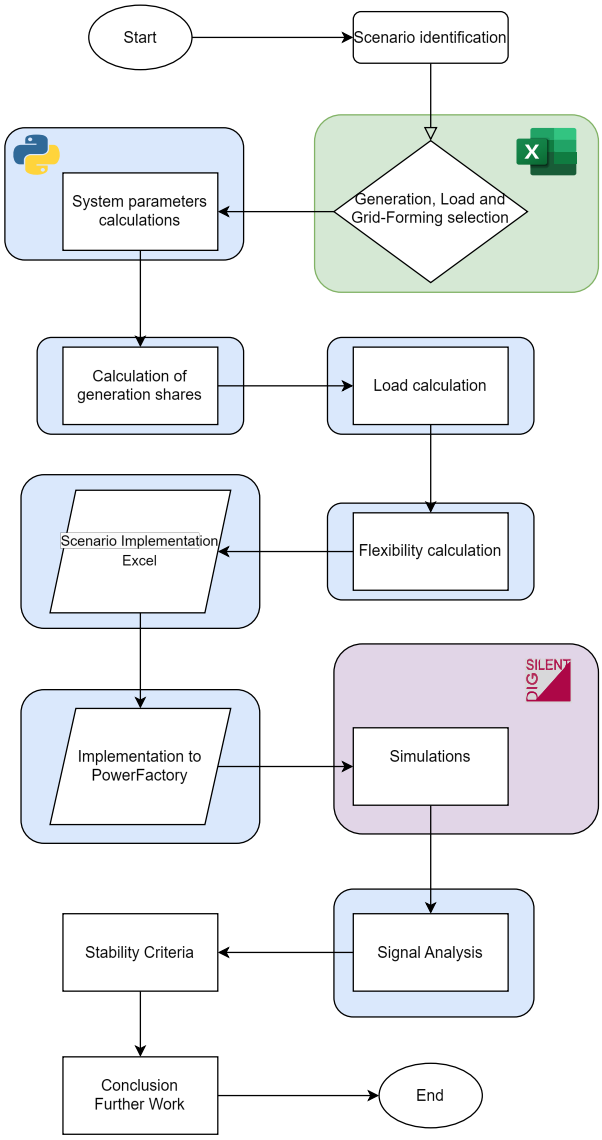


Figure 7.1: Detailed flowchart of the system setup for assessing the stability of the Dutch power system in 2050 under various generation, demand, and flexibility cases. The pink color indicates operation performed by PowerFactory, green by Excel, and blue by Python.

The foundational scenario for the simulations is detailed in chapter 4, incorporating the II3050-2 National scenario and TenneT's Target Grid as a proxy for future system configurations. Additionally, six distinct cases were formulated for this analysis. These cases encompass varied onshore wind, offshore wind, and solar PV levels, each with peak or minimum load conditions. A brief overview of these cases is provided below:

1. Case A: High offshore wind with peak demand

2. Case B: High onshore wind with peak demand
3. Case C: High solar PV with peak demand
4. Case D: High offshore wind with minimum demand
5. Case E: High onshore wind with minimum demand
6. Case F: High solar PV with minimum demand

The various cases investigated how high levels of generation from renewable energy resources affect the dynamic stability of the system. These high levels can result in large power flows across large distances. ENTSO-E recognizes such extensive power transfers as critical factors that can compromise system stability during certain outages and faults due to significant generation loss at specific locations [175].

Cases A and D focused on analyzing the effects of high offshore wind penetration under both peak and minimum demand conditions. Cases B and E assess the impact of high onshore wind, while cases C and F explored the consequences of high solar PV penetration. The distribution of generation among the three main renewable energy sources—offshore wind (WOZ), onshore wind (WOL), and solar PV (PV)—as well as the system load, is presented in Table 7.1. These proportions were selected to represent a range of potential future cases for the Dutch power system, reflecting periods of significantly high contributions from each resource to examine how such dynamics influence system stability under peak and minimum load conditions.

Table 7.1: Distribution of renewable energy sources and demand levels across cases, illustrating diverse configurations of offshore wind (WOZ), onshore wind (WOL), solar PV (PV), and corresponding demand percentages to assess their impacts on system stability.

Case	WOZ (%)	WOL (%)	PV (%)	Demand (%)
A	90	10	5	100
B	5	90	5	100
C	0	10	65	100
D	90	10	5	45
E	5	90	5	45
F	0	10	65	45

An overview of the dispatched generation and demand levels corresponding to the different percentage shares is presented in Table 7.2. In all cases, conventional generation was provided by nuclear power plants at MVL380 and BSL380, each operating at a constant dispatch of 1 GW. Due to these uniform generation levels, the BSL380 plant was utilized for frequency signal analysis unless otherwise specified. The generation distribution from various renewable energy sources was evenly allocated across grid-forming and grid-following converters for all three converter types, achieving a 50 % grid-forming penetration level. This distribution was facilitated by a Python script designed to allocate generation based on specified percentages and grid-forming penetration levels, ensuring adaptability to varying configurations of the renewable mix and grid-forming levels within system constraints.

While the script ensured even distribution across converters, utilizing randomized functions from Python libraries means that a perfectly localized generation distribution among the different renewable resources could not be guaranteed. Instead, it ensured an equitable distribution system-wide, which may result in zones with varying degrees of grid-forming penetration. Despite this, the overall distribution aligns with a 50 % grid-forming target for the entire system. Future enhancements to this model could improve the precision of resource distribution. However, a system-wide approach remains effective for a broad study such as this. The developed code offers a robust foundation for further studies and projects aiming to refine grid-forming penetration level calculations and implementations. The codes can be found in Appendix C.

Table 7.2: Generation distribution across cases, detailing the allocation of renewable and conventional energy sources in megawatts (MW).

Cases	WOZ	WOL	PV	Flexibility	Conv. Generation	Demand
A	47272.5	2068.5	8649.5	12020.29256	2000	47979.9
B	2626.25	18616.5	8649.5	11754.41622	2000	47979.9
C	0	2068.5	112443.5	67800	2000	47979.9
D	47272.5	2068.5	8649.5	38519.6922	2000	21591.0
E	2626.25	18616.5	8649.5	10550	2000	21591.0
F	0	2068.5	112443.5	75950	2000	21591.0

Flexibility levels were tailored to balance the excess or deficit in generation and load for each case, aiding in maintaining system stability. These settings are dynamically adjustable in the Excel model, facilitating modifications and comparisons for future analysis cases (Appendix C). This flexibility in adjustment supports easier integration and evaluation of different future system cases and allows for a systematic analysis of how varying flexibility resources impact system stability and system operation. The flexibility allocations for the different cases can be seen in Table 7.3. Here, a negative BESS value indicates charging, while a positive value indicates discharging.

The interconnection, typically a critical flexibility resource, was strategically underutilized and reserved instead as a last resource to emphasize national resource management over-reliance on cross-border interconnections. However, cases B and F required active utilization of interconnections to ensure operational stability due to insufficient domestic flexibility resources. In case B, where generation was notably less than demand, additional resources such as fuel cells and hydrogen power plants were employed. Conversely, case F had an excess of generation requiring interconnection capabilities to manage the surplus effectively. In the other cases, while interconnections remained operational, they did not engage in active power dispatch, thus providing only inertia and active and reactive power support.

All cases incorporated load response and power-to-heat dynamics, set to their maximum capacities to represent the dynamic capabilities of future system loads (section 6.5). The flexibility resources were configured with a wideband of 0.15 and 0.1 Hz, respectively, triggering load shedding after deviations from these frequency thresholds, employing a standard dynamic Droop response similar to that used for electrolysers. For further details on the deployment and implications of the flexibility resources, see section 6.4. Despite the comprehensive setup, the primary focus of this case study remains on examining the impact of grid-forming converters and the diversity of generation shares, with a less extensive analysis of flexibility resource impact.

Table 7.3: Detailed allocation of flexibility resources across cases, expressed in megawatts (MW)

Case	Total	Hydrogen Plants	electrolysers	BESS	Fuel-Cell	Interconnections
A	6020.3	0	6020.3	-6000	0	0
B	11754.4	5300	0	4454.4	2000	4454
C	67800	0	24800	-43000	0	0
D	38519.7	0	12900	-25619.7	0	0
E	10550	0	2300	-8000	0	0
F	75950	0	23420	-52200	0	-18799

Moving on, a flowchart representing the various simulation setups for Case Study A is displayed in Figure 7.1. This flowchart delineates two principal investigative pathways within the case study. Initially, the pathway on the left examined the six different cases, simulating their response to three distinct disturbances: a load event, an outage event, and a short circuit. These disturbances include an outage at the 2 GW nuclear power plant in Maasvlakte (MVL380), a short circuit on a tie-line to the 150 kV power system of Noord-Brabant (B150), and variations in load—specifically, a 5.6 GW increase for peak load cases and a 10 GW decrease for minimum load cases. The frequency responses from these disturbances were analyzed using Python to assess the system frequency stability.

Subsequently, case A was specifically chosen to examine the system's oscillatory behavior in-depth. In this instance, the load event serves as the initiating disturbance, leading to an eigenvalue analysis conducted via PowerFactory, which facilitated the exploration of oscillatory modes and participation factor analysis. Utilizing a combination of frequency response and eigenvalue analysis allowed for a comprehensive stability assessment concerning frequency, rotor angle, and converter-driven stability, providing a detailed evaluation of these parameters across varying system characteristics and operational conditions.

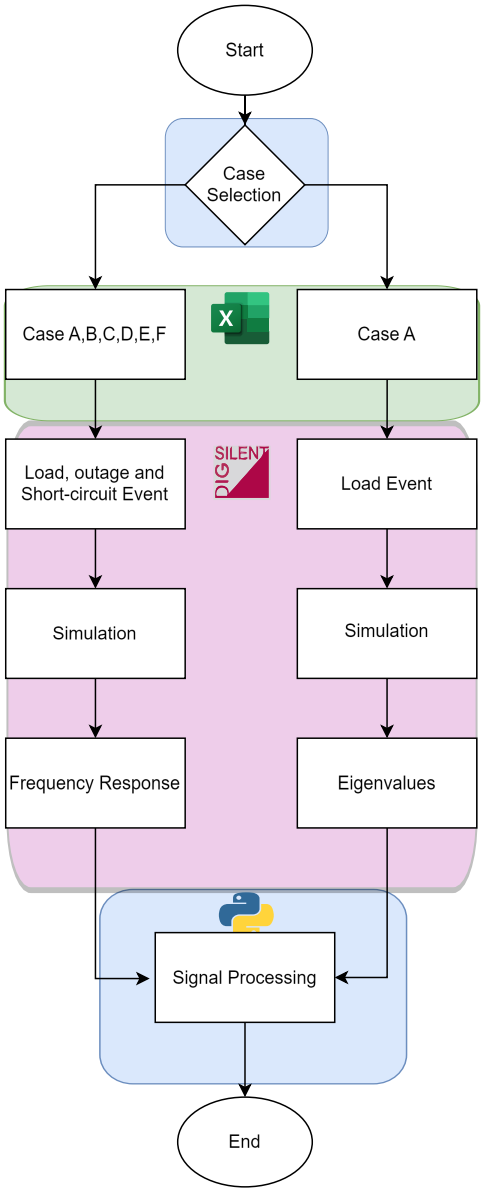


Figure 7.2: Detailed flowchart of simulation methodologies used in Case Study A, with components involving PowerFactory distinctly highlighted for clarity. The pink color indicates operation performed by PowerFactory, green by Excel, and blue by Python.

Finally, following the initial two analyses, an additional part focused on the damping of specific modes using grid-forming converters. This analysis involved modifying the damping parameter, D_p, of a strategically positioned grid-forming converter to evaluate its influence on each mode's eigenvalues and damping ratios. Three different values of D_p were tested to assess their impact: 100, 10, and 1000, with 100 being the original value.

The integration of system simulations and calculations, utilizing Python and Excel, enhanced the ease with which the model can be adapted to future cases, including various estimations and policy-dependent changes. This case study demonstrates the synthetic model's capability to integrate seamlessly with different configurations, providing a comprehensive system stability assessment tool. This tool facilitates a deeper understanding of the inherent dynamics of future power systems and establishes a benchmark for future policy formulation and system operations.

7.2.3. Results

7.2.3.1. Outage Event

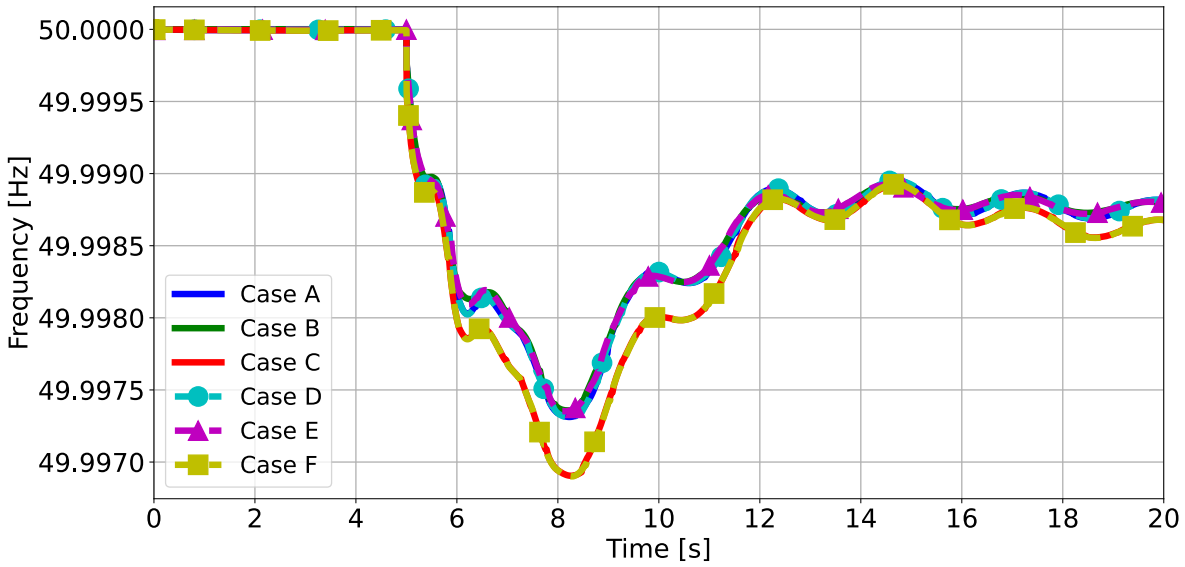


Figure 7.3: Frequency responses of the 2 GW Gen Nuclear BSL380 following a 5-second post-outage event at Gen Nuclear MVL380, across various cases.

7.2.3.2. Short-Circuit Event

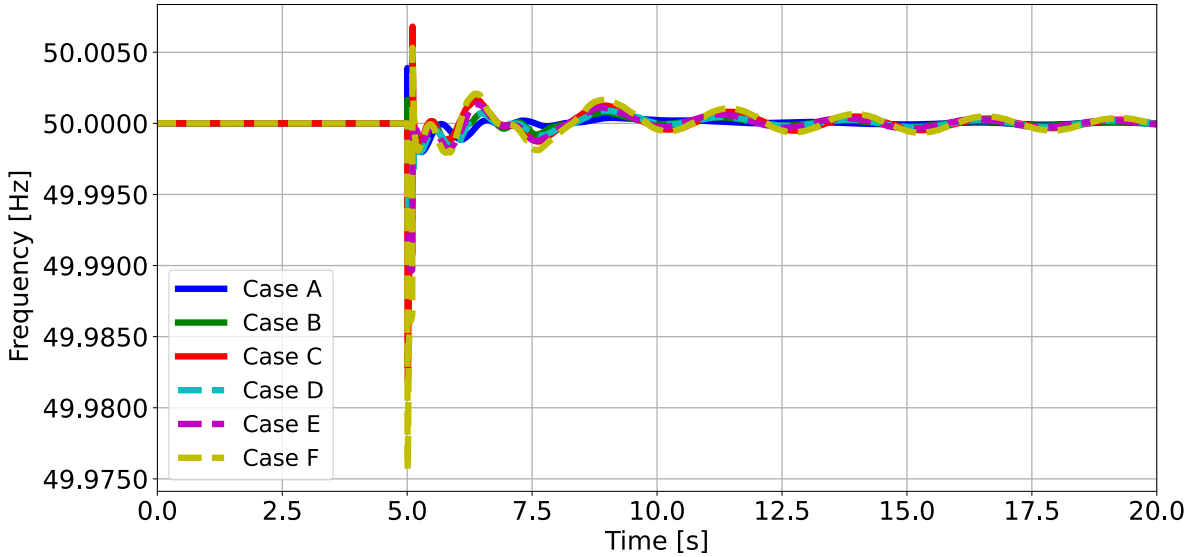


Figure 7.4: Frequency responses of the 2 GW Gen Nuclear BSL380 following a short-circuit at 5 seconds across various cases.

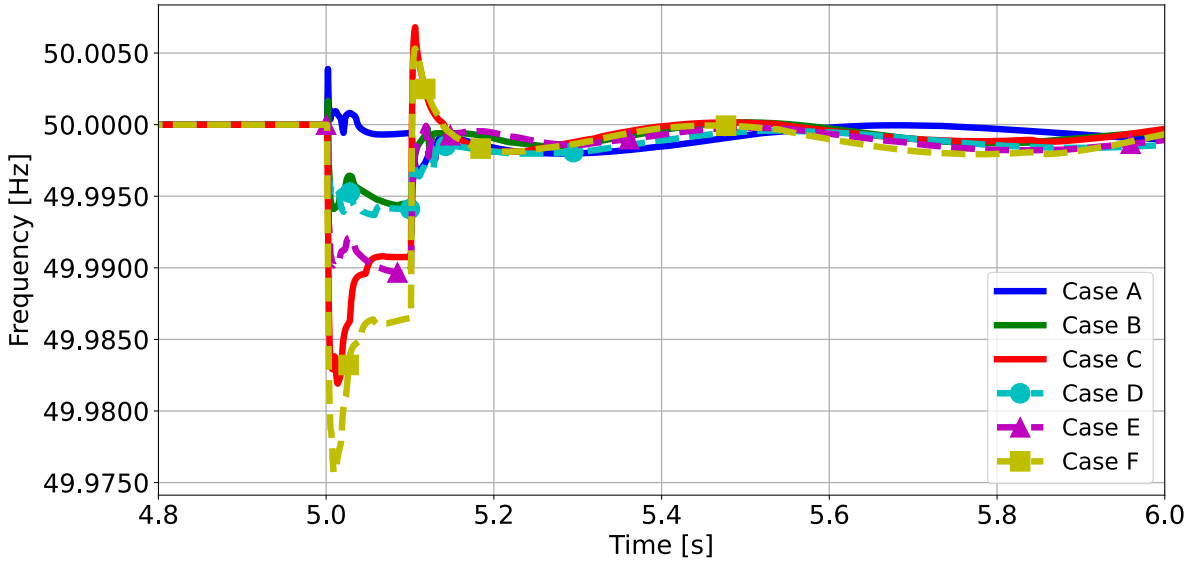


Figure 7.5: Detailed frequency response of the 2 GW Gen Nuclear BSL380 post-short-circuit at 5 seconds, illustrated for different cases with a zoomed view.

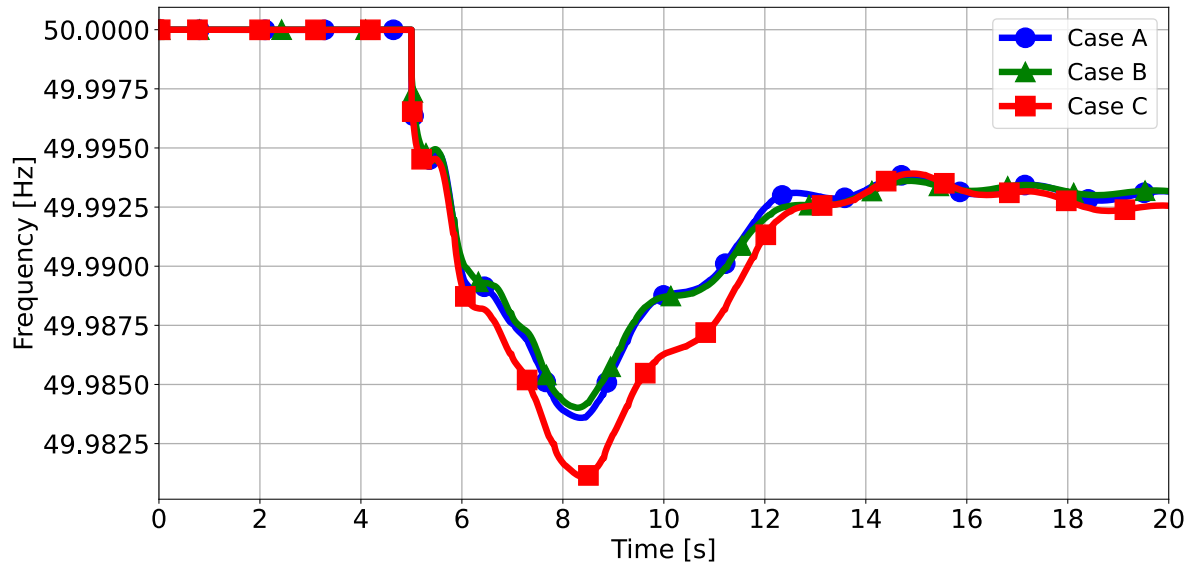


Figure 7.6: Frequency responses of the 2 GW Gen Nuclear BSL380 following a load increase event, specifically illustrated for cases A, B, and C with high load conditions.

7.2.3.3. Load Events

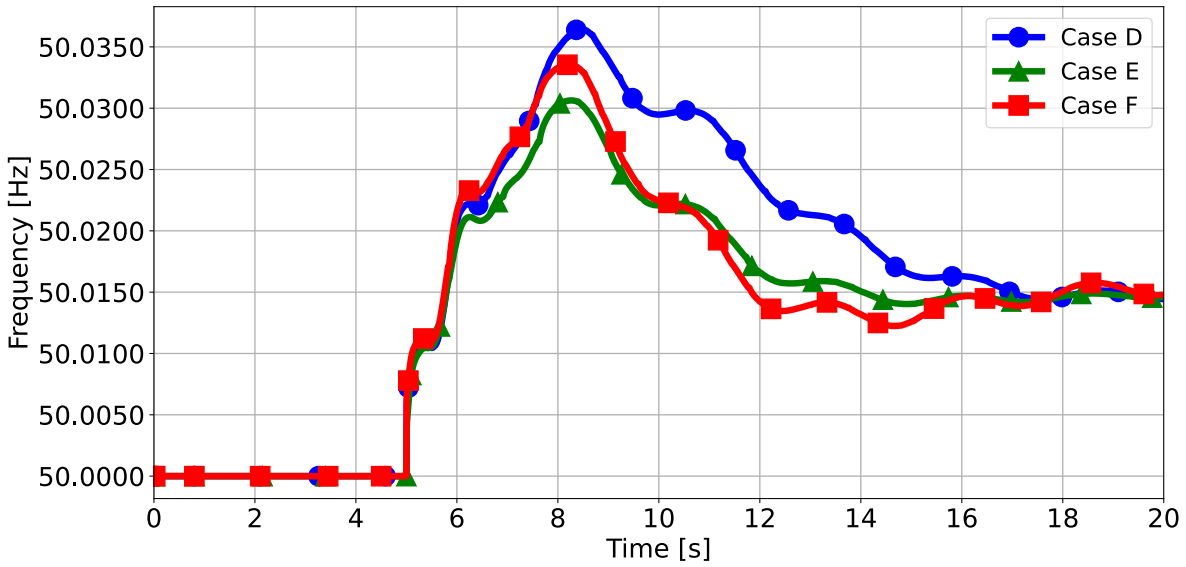


Figure 7.7: Frequency responses of the 2 GW Gen Nuclear BSL380 following a load decrease, specifically depicted for cases D, E, and F with reduced load conditions.

Table 7.4: Rate of Change of Frequency (ROCOF), frequency deviation, and frequency nadir values for each of the three disturbance events analyzed in the study.

Outage Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
A	49.9973	1.5	1.0	2.7
B	49.9974	1.1	0.9	2.6
C	49.9969	1.2	1.0	3.1
D	49.9973	1.4	1.0	2.7
E	49.9974	1.5	0.9	2.6
F	49.9969	1.3	1.0	3.1

Short-Circuit Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
A	49.9961	16	6.3	3.9
B	49.9941	48.7	12	5.9
C	49.9819	237.7	35.9	18.1
D	49.9937	47.6	11.6	6.3
E	49.9896	97.4	19.5	10.4
F	49.9757	289.5	48.1	24.3

Load Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
A	49.9836	6.6	4.8	16.4
B	49.984	5.6	4.2	16
C	49.9811	6.3	4.8	18.9
D	49.9635	89.4	22	36.5
E	49.9694	81.7	20.9	30.6
F	49.9665	96.9	22.3	33.5

7.2.3.4. Oscillatory Analysis

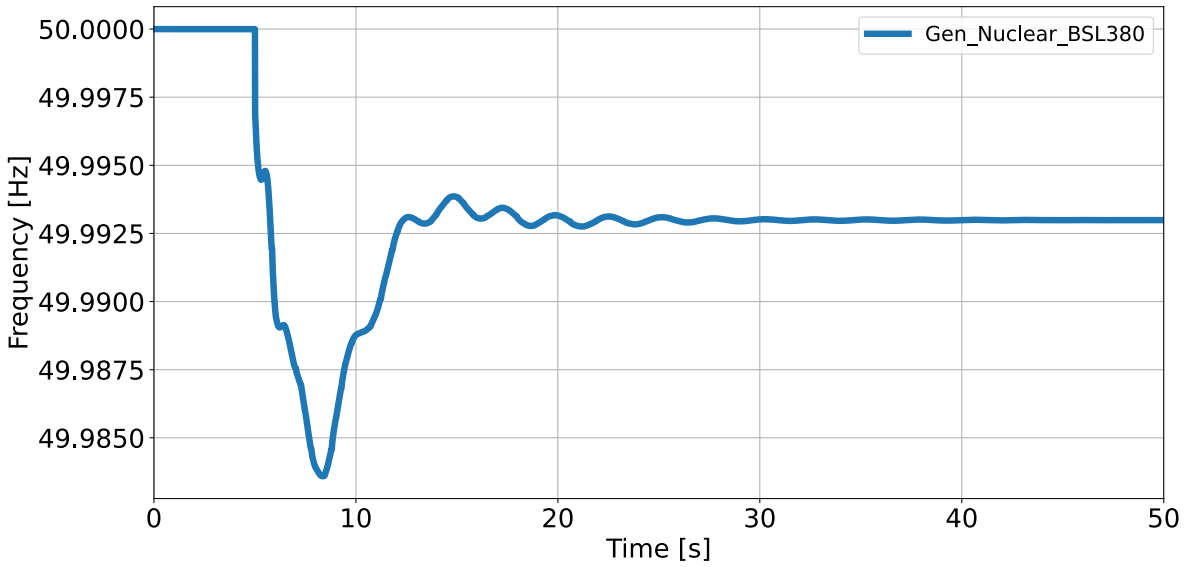


Figure 7.8: Extended frequency response of the 2 GW Gen Nuclear BSL380 following a 5-second post-load increase across different cases over a 50-second simulation period.

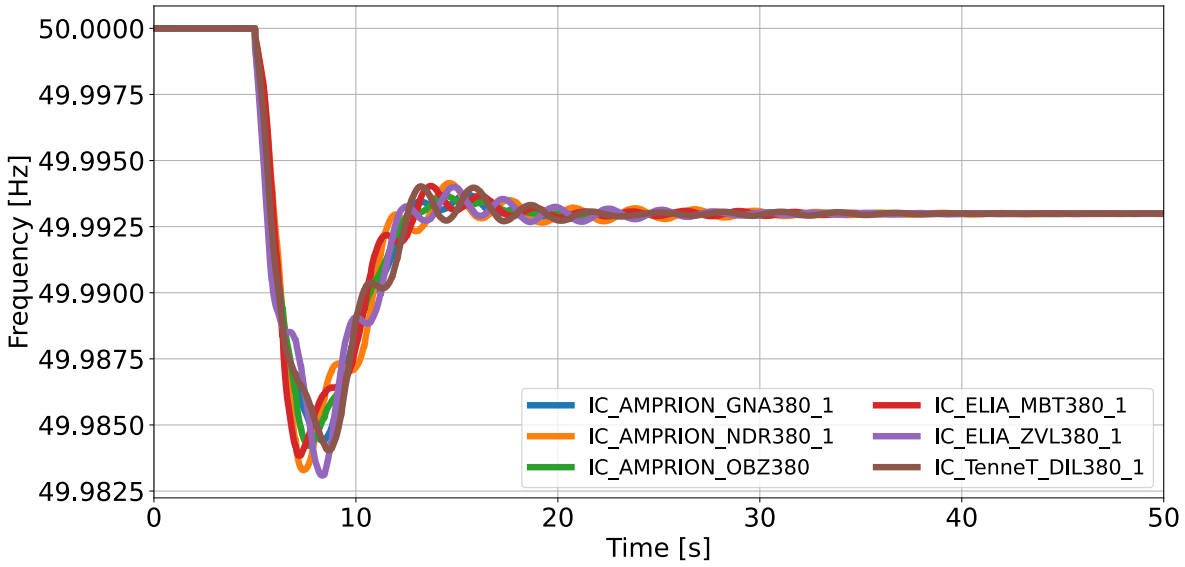


Figure 7.9: Extended frequency response of AC interconnections following a 5-second post-load increase, depicted across various cases over a 50-second simulation window.

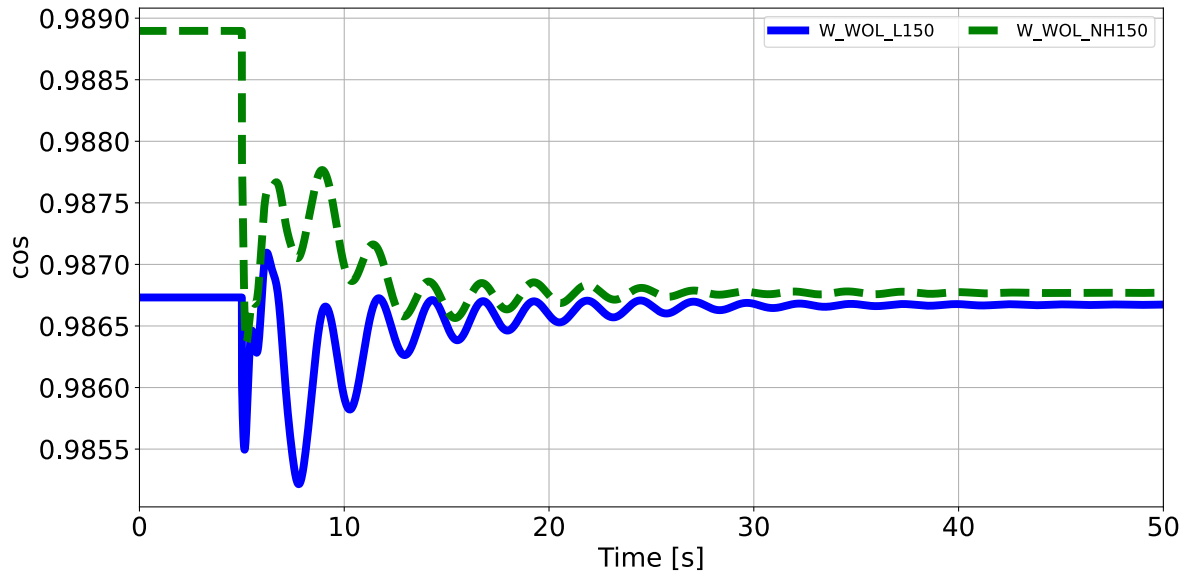


Figure 7.10: Cosine of the reference angle for grid-following converters at L150 and NH150 following a load increase, presented over an extended simulation period.

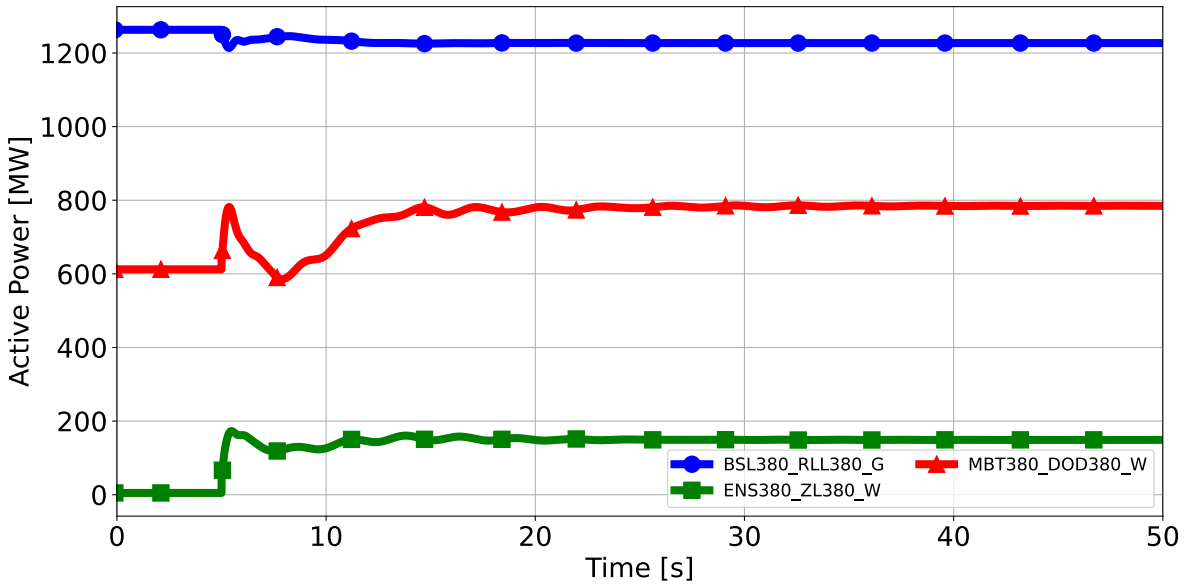


Figure 7.11: Active power flow dynamics in selected transmission lines following a load increase, visualized over a 50-second extended simulation window.

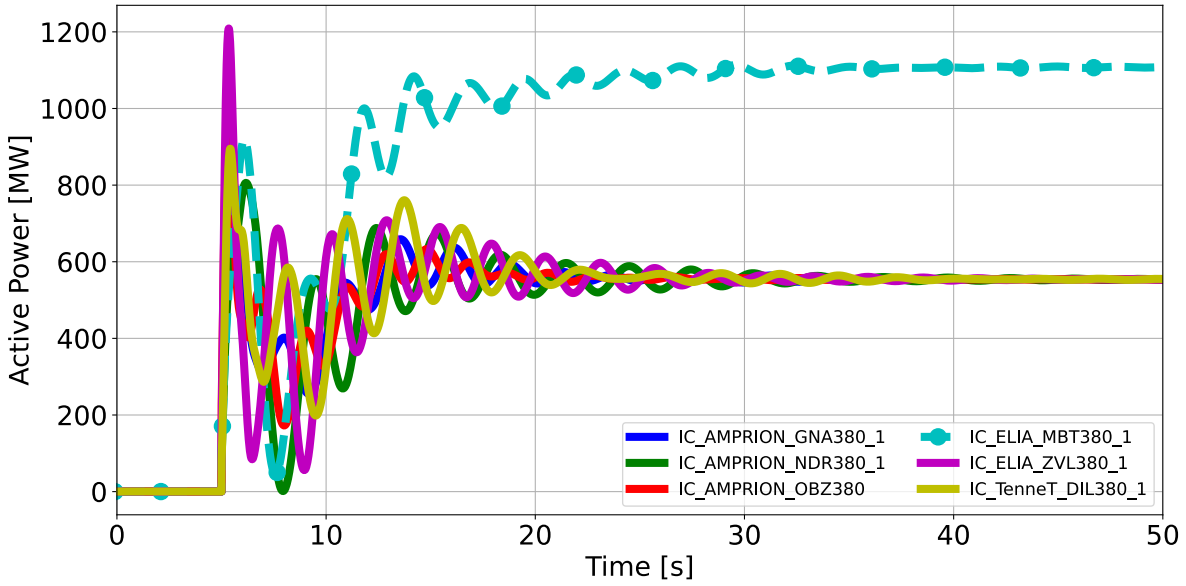


Figure 7.12: Active power dynamics in AC interconnections following a 5-second post-outage event at Gen Nuclear MVL380, analyzed across various cases during a load increase over an extended 50-second simulation window.

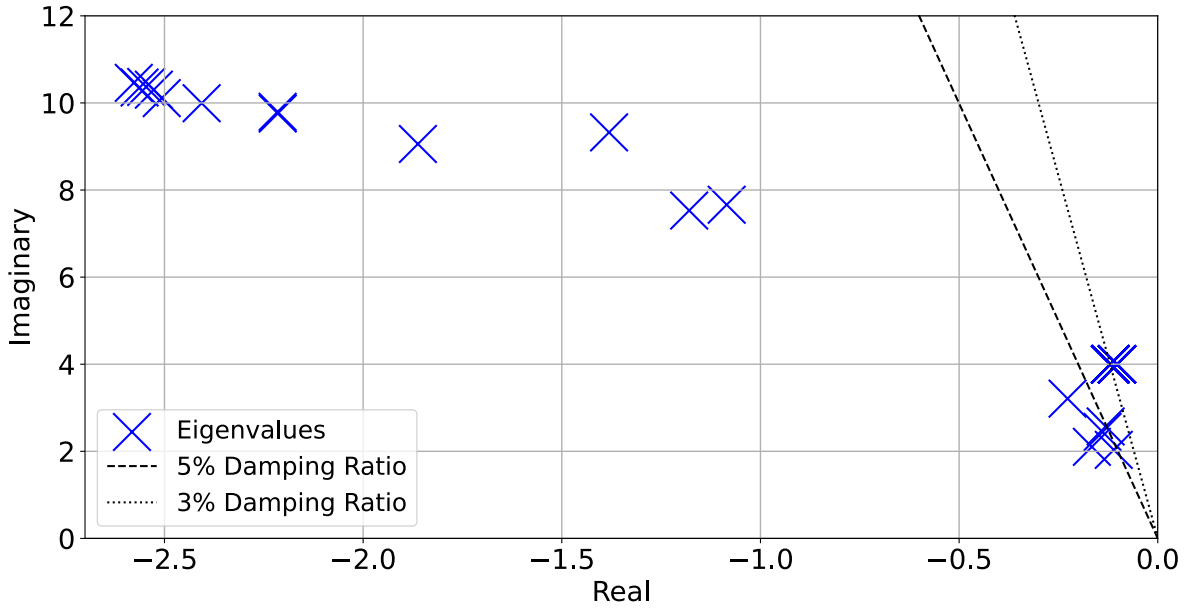


Figure 7.13: Eigenvalue analysis from Case Study A's load event case, highlighting eigenvalues with less than 50 % damping ratio over a 50-second analysis period.

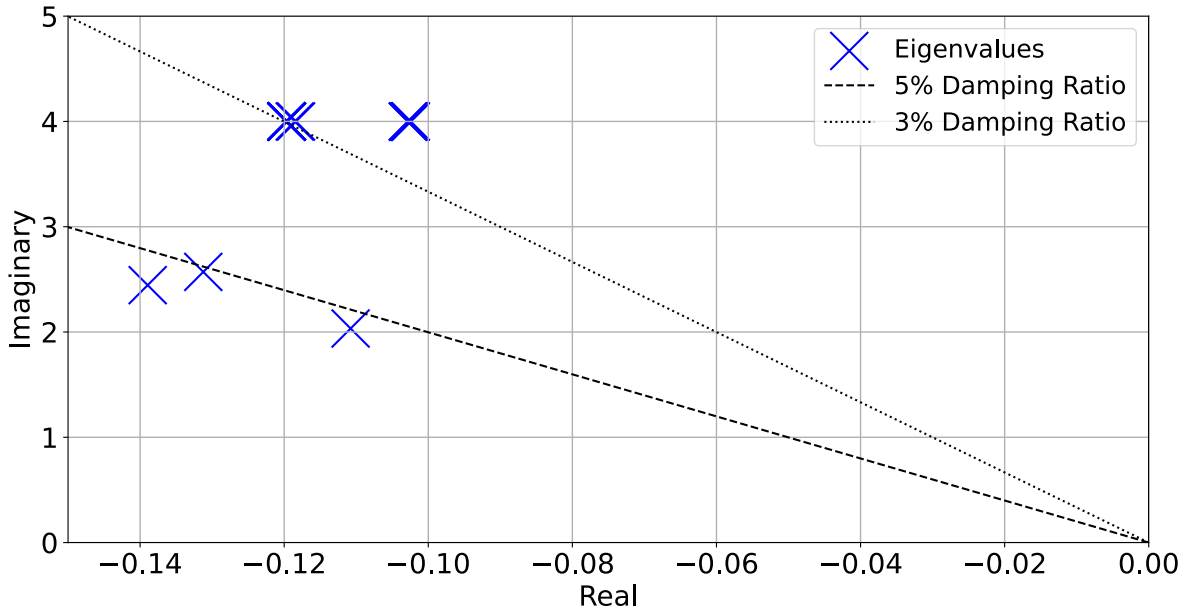


Figure 7.14: Detailed view of the most critical eigenvalues from Case Study A's load event, emphasizing those with the highest impact on system stability.

Table 7.5: Comprehensive Table of Mode Indices, detailing Eigenvalues, Frequencies, and Damping Ratios for Case Study A's eigenvalue analysis.

Mode Index	Eigenvalue	Frequency	Damping Ratio
62	$-0.1024 - 3.9989i$	0.6364	2.5610
121	$-0.1197 + 3.9996i$	0.6366	2.9906
122	$-0.1312 - 2.5709i$	0.4092	5.0978
86	$-0.1108 - 2.0332i$	0.3236	5.4405
124	$-0.1390 - 2.4449i$	0.3891	5.6740
192	$-0.2271 - 3.2098i$	0.5109	7.0568
129	$-0.1658 - 2.1029i$	0.3347	7.8621
254	$-1.0850 - 7.6626i$	1.2195	14.0198

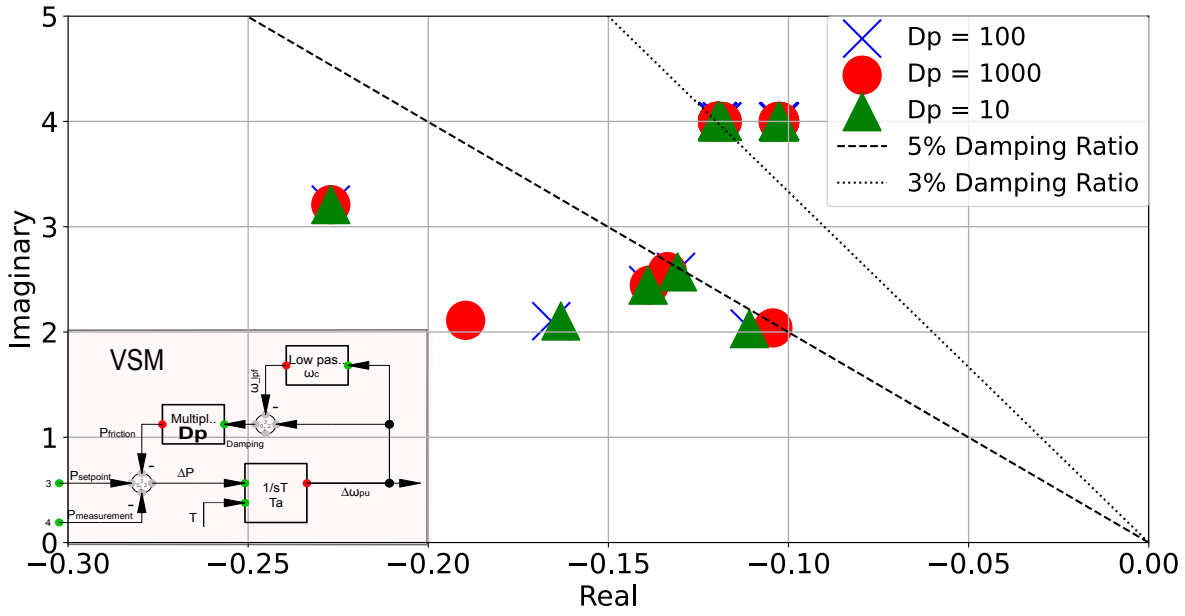


Figure 7.15: Analysis of the most critical eigenvalues during the load event in Case Study A, for the parameter analysis of the VSM converter

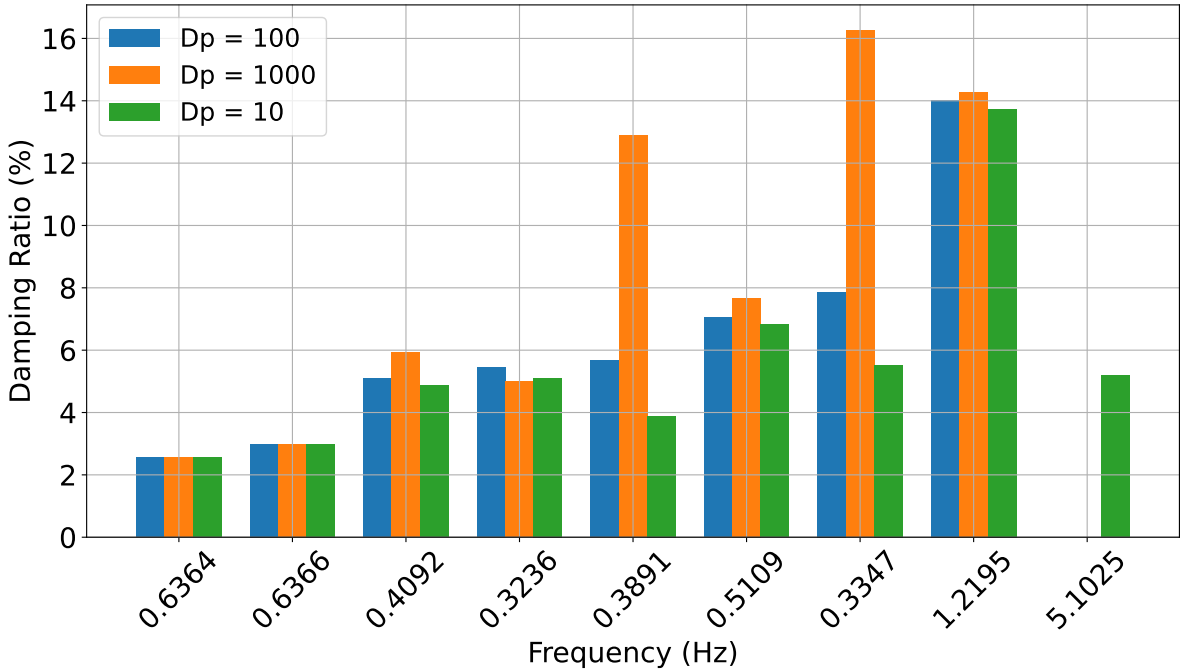


Figure 7.16: Damping ratio and frequency for the most critical modes for the parameter analysis of the VSM converte

7.2.4. Discussion

7.2.4.1. Fast Active Power Response

A comparative plot of the frequency response for the nuclear power plant at Borselle (BSL380) after an outage is illustrated in Figure 7.3. This plot aggregates the responses of the generator across all six cases. BSL380 was chosen as the single comparison point due to its consistent dispatch level of 1 GW across all cases, simplifying the comparison by focusing on one machine. The plot reveals that while all cases exhibit similar response patterns, they differ in their levels of frequency deviation. Cases A, B, D, and E show comparable frequency deviations, whereas cases C and F display higher deviations. The differences are about 0.15 mHz, with overall frequency deviations ranging from 2.6 to 3.1 mHz. This indicates that the system's frequency remains within regulatory bounds during the outage of the 1 GW generator across all cases. However, cases dominated by Solar PV (C and F), which feature no offshore wind, show greater deviations. This is attributed to the lack of available power from offshore wind converters to assist with fast-active power balancing, resulting in a lower overall contribution from solar-dominant cases. Nonetheless, the permitted deviation remains low across all cases, supported by the high levels of kinetic energy and high inertia constants provided by the interconnections with continental Europe, set to maximum estimations for these cases.

The frequency responses, though minor across all cases are notably fast. The initial response to the frequency drop is almost instantaneous, followed by a slight increase before further decline. This rapid response is characteristic of the converter-based resources in the system, contrasting with the traditional sinusoidal response typical of synchronous dominant power systems. Instead, the initial response appears more erratic, a common outcome when inverter resources react quickly compared to slower traditional synchronous machines. While this enables quicker response times, it may also introduce issues such as increased harmonics and additional strain on older, slower components. These aspects are not explored in this study but present valuable directions for future research. Notably, at around 13 seconds, the overall frequency stabilizes, demonstrating good system damping despite minor oscillations as the simulation concludes. The overall system stability remains robust with respect to frequency deviations, and the dynamic response effectively minimizes the impact of the outage event on stability limits. However, the higher inertia of the system, which typically leads to slower response times, results in more prolonged oscillatory responses due to the larger system time constant.

The frequency response due to a short-circuit at the line connecting the 380 kV power system to the 150 kV power system at B150 is depicted in Figure 7.4 and Figure 7.5. While voltage levels are generally critical in assessing the impact of short circuits on system stability, the increasing contribution of power electronic inverter (PEI) generation makes the analysis of short-circuit impacts increasingly essential. This is due to the limited short-circuit current that converters can provide, constrained by the thermal limits of power electronic components. Typically, these are in the range of about 1.1-1.2 pu, compared to 3-6 pu for low impedance short-circuits from synchronous generators [176]. Consequently, short circuits have a pronounced effect on various system components. When a short-circuit occurs, the local voltage drop prompts generators to increase their output current to supply short-circuit current, reducing synchronous speed. This deceleration affects the entire interconnected power system, with its severity influenced by factors such as power system loading, short-circuit current levels, and the system's protection schemes. Consequently, the overall system frequency deviates, with the inertia and kinetic energy of the system playing significant roles. Lower total inertia and kinetic energy under short-circuit conditions result in a more substantial initial frequency drop due to the more significant impact of localized generator deviations on system variables. This is particularly crucial when synchronous machines are replaced with PEI generation, with reduced inherent short-circuit capabilities. The lower short-circuit current contributions from PEI generation significantly affect overall system stability concerning short-circuits.

Comparing different cases, it is evident that the lowest permissible frequency deviations occur in cases with high solar PV levels and no offshore wind penetration. Even though converter-based resources provide lower short-circuit current compared to synchronous generators, they still offer more current than disconnected converters. However, in cases C and F, where no offshore wind plants are available to contribute to the power system, the overall short-circuit current contribution is significantly reduced, highlighting the impact of short-circuits on system stability. This difference is most pronounced

during the initial swing of the frequency response, leading to poorer system stability performance in restoring the system to steady state. The responses in onshore and offshore wind-dominated cases (A, D, B, and E) initially show lower allowable frequencies for the onshore wind cases during the short-circuit, which stabilize to nearly identical amplitudes once the short-circuit is cleared, indicating that the same resources are contributing during the event. This initial variation is likely due to the short-circuit location—B150 being inland, closer to areas with high onshore wind penetration but limited offshore wind. Therefore, when a short-circuit occurs at B150, the local converter voltage levels drop, reducing power levels in the power system and thus significantly impacting system stability. This is also relevant for cases C and E, where high levels of solar PV are also inland, closer to the short circuit. These cases exhibit much larger peak-to-peak amplitude responses compared to other cases, as detailed in Table 7.4. A difference of 14.2 mHz is observed between the two extreme cases among the high load cases (A-C), compared to an 18 mHz deviation for the low load cases (D-F). The most considerable frequency deviations and overall permissible variations occur in the low load cases, attributed to higher stress levels on flexibility resources due to the greater generation levels in the system, resulting in a more strained system.

Further analysis of the lack of short-circuit strength provided by offshore wind farms shows that in case C, no offshore wind was engaged to provide ancillary support during the short-circuit. The comparative response between cases with and without additional converter support is illustrated in Figure B.1. This analysis indicates that including additional converters significantly improves the response, with deviations much lower and now comparable to other cases with better outcomes. While expected, as the short-circuit current potential increases along with the ancillary services from grid-forming converters, this finding underscores the variability associated with renewable resources. In cases where the wind is absent, or the sun is not shining, converters connected to the power system can become ineffective. This variability suggests that including as many converters as possible for ancillary support could be beneficial, potentially supported by battery storage systems to ensure power availability for ancillary services. This concept requires further research but could be viable for enhancing overall system stability.

In summary, the impact of a short-circuit depends significantly on the location of the event, the impedance at the fault, the available short-circuit currents, and the system's load conditions. Systems that are highly loaded with lower short-circuit current capabilities tend to respond worse to short-circuits. A more extensive short-circuit analysis is necessary to fully validate the system's performance under various short-circuit conditions. While not conducted in this project due to data simulation limitations, this remains a promising area for future research. The findings from the short-circuit analysis are crucial, demonstrating that short-circuits can substantially affect fast-active power and frequency balancing and thus are critical with regard to stability analysis.

The system's frequency response to a 5.6 GW load increase occurring 5 seconds post-event is depicted in Figure 7.6. This response shows that the largest deviations occur in case C, while cases A and B exhibit similar responses. The absence of WOZ converters in these cases, and thus the lack of ancillary services from these resources, is apparent in the frequency response. Although cases A and B show slight differences, these do not significantly affect system stability, and responses are nearly identical across these cases. However, some oscillations are still evident in all cases at the end of the simulation window.

A load decrease was also introduced in the low-load cases to analyze a similar impact to that observed during the load increase. Here, a 10 GW load decrease was introduced. The results are displayed in Figure 7.7, showing that case D experiences the highest permissible frequency deviation, while case E shows the lowest. The objective after the perturbation is to reduce power generation or increase power demand in the system, thus decreasing system frequency, which means the implementation of WOZ grid-forming converters and their ancillary services plays a less critical role. All results eventually settle around a similar level by the end of the simulation window, but it is clear that the response varies across different cases. Comparing the overall frequency deviation levels to the A-C cases in Table 7.4, it is evident that the load decrease has a more significant impact than the load decrease. Naturally, the load decrease had a higher magnitude than the load increase. The most significant deviations among all three events are observed for the load decrease. However, all responses remain within acceptable limits as defined in subsection 2.3.2. Although deviations exist between respective load increase and

decrease cases, these are likely due to differences in topologies and their respective loading, proximity to flexibility resources, and load characteristics.

Overall, the system exhibits adequate stability concerning frequency stability for all three highlighted events: the outage, short-circuit, and load events. The most significant deviations are observed for the load decrease in cases D-F. The highest median deviations occur during the load event, with greater disparities between cases during the short-circuit event. On the other hand, the outage event shows excellent response characteristics, with only minor deviations across different cases. This study underscores various important perturbations concerning frequency stability and further highlights the intrinsic nature of stability concerning future systems, the penetration of grid-forming converters, and different levels of PEI-generation resources. It is clear that the various cases do not create extreme differences concerning the overall loading of the system. This is largely due to the extensive infrastructure requirements associated with the N-2 contingency design phase of the system as discussed in section 6.7. If a different approach is taken regarding the future development of the Dutch power system, this should be considered in future research as it could play a more significant role than currently modeled. The future of these systems will depend on various policies concerning the implementation of renewable energy resources and the selection of controllers based on studies and research into future systems. This analysis highlights that the implementation of grid-forming converters can play a significant role in ensuring the frequency stability of future systems, and the scenario and case implementation demonstrate excellent stability performance.

7.2.4.2. Oscillatory Analysis

To comprehensively assess the oscillatory performance of the system, an eigenvalue and extended frequency response analysis was conducted for Case A during a load increase event. The simulation window was extended to 50 seconds to identify any sustained oscillations over longer periods. A similar pattern was observed for all cases and events; thus, only the results from Case A and the load event are presented for clarity. The frequency response for the nuclear generator at BSL380 is depicted in Figure 7.8, with all synchronous machines in the system shown in Figure B.2. These figures illustrate that oscillations persist beyond the initial 20-second window, settling around the 30-second mark.

An eigenvalue analysis was also performed, with the results displayed in Figure 7.13. The eigenvalues with a damping ratio below 50 % are given for further analysis, representing the lowest damped modes in the system. Most modes exhibit a damping ratio at or above 5 %, as indicated. A closer examination of modes with lower damping ratios is provided in Figure 7.14, revealing some modes with damping ratios around or below 3 %. These modes' frequency and damping ratios are detailed in Table 7.5. For clarity, only one mode associated with each frequency is mentioned despite the presence of multiple similar modes. Notably, modes 62 and 121 have damping ratios at or below 3 %, with a frequency of 0.636 Hz.

Following the definition provided in Table 2.2, the first two modes are identified as intra-area modes, involving either a subgroup of generators swinging against each other or generators swinging within the same area. A participation factor analysis for mode 62 is shown in Figure B.3, indicating that this mode is dominated by the onshore wind grid-following converters of VHZ380 and OZN380, with the highest contributions from OZN380. These oscillations are influenced by grid-following converters, particularly the two-mass control block, which mimics the dynamic performance of the wind turbine's drive train (rotor and stator) [177] and conforms to the IEC61400-27-1:2020 standard [178]. This mode includes various sub-modes with identical frequencies and varying damping ratios from modes 62 to 121, all impacted by different grid-following wind converters, both onshore and offshore. The close proximity of VHZ380 and OZN380 confirms that these oscillations arise from grid-following converters connected in close proximity swinging against each other. While the term "swinging" typically describes synchronous machines, a similar assessment applies to grid-following converters, which inherently "follow" the power system. Power system oscillations can affect these converters' performance, causing further instability. This is corroborated by analyzing the cosine reference angle for the two grid-following converters associated with mode 121 in Figure 7.10, which shows sustained oscillations post-event. The oscillations in the reference angle, crucial for converter output, highlight the influence of inter-area oscillations. Consequently, oscillations in the power system can destabilize grid-following converters, potentially leading to cascading effects where converters exacerbate system instability.

Literature has documented oscillations due to grid-following converter interactions, often linked to phase-locked loops (PLLs) under weak-grid conditions [179, 180, 181, 182]. Such oscillations have been observed in real power systems, including the ERCOT system in the U.S. [183]. Mitigation strategies include parameter adjustments to converter control, adding damping units, strengthening the transmission network, implementing control strategies for Battery Energy Storage Systems (BESS), using grid-forming converters, and eliminating alternative oscillations [184, 185, 186, 187]. These oscillations will not be further analyzed in this case study as they are outside the scope of this thesis. However, the impact of grid-forming converters will be examined in Case Study B.

Mode 86, categorized as an inter-area oscillation (typically below 0.4 Hz), was also analyzed. A participation factor analysis for mode 86 is provided in Figure B.7, showing that the majority of oscillations originate from Amprion NDR380, an AC interconnection to Germany in the Utrecht region. Contributions from other AC interconnections suggest this is primarily an inter-area oscillation between AC interconnections, with a damping ratio of 5.4405 %. Inter-area oscillations are well-documented in literature and, if not properly damped, can significantly impact power systems [188, 189]. Such oscillations are prevalent in the interconnected power system of continental Europe, causing large power flow and voltage deviations [190, 191, 192]. The frequency response for the AC interconnections, shown in Figure 7.9, reveals sustained oscillations post-20 seconds, with some interconnections swinging against each other. The active power transfer for the AC interconnections, depicted in Figure 7.12, shows oscillations similar to the frequency response, indicating power flow oscillations. The initial response to the load event causes substantial power transfers, with significant deviations observed in the connection to Belgium (ZVL380). Here, oscillations ranging between 1200 to 150 MW are evident in the initial response, indicating large power transfer variations. Oscillations after the 20-second mark are also evident, with oscillations of around 60 MW for the MBT380 interconnection to Belgium after the 20-second simulation window. Overall, it is evident that there are oscillations with large power transfers for the system during and after the load increase. These oscillations can disrupt protection schemes and cause cascading outages.

Further analysis of the Dutch power system's impact is shown by the active power response for three transmission lines in Figure B.11. Oscillations are evident but less pronounced than in the interconnections, particularly on the transmission line between ENS380 and ZL380 and between BSL380 and RLL380. Larger oscillations are observed on the MBT380 to DOD380 line due to its proximity to interconnections OBZ380 and NDR380 to Germany. Here, oscillations ranging from active power values of 775 to 580 MW are evident in the first two swings, and oscillations ranging from 790 to 760 MW after the 20-second simulation window. The majority of generation in Case A comes from the western Dutch power system, while the AC interconnections are in the east, creating significant dispatches and increasing inter-area oscillation risks.

Modes 122, 124, 192, and 129 are also influenced by interconnections, as shown from Figure B.5 to B.9. Though above the 3 % and 5 % damping ratio thresholds, they pose no immediate threat to system stability. Mode 192, dominated by interconnections, falls above the typical 0.4 Hz inter-area oscillation range yet is identified as an inter-area mode due to its participators, highlighting the evolving system stability challenges with fast-acting control types and the changing nature of power system modes as a result.

Mode 254, with a frequency of 1.2195 Hz and a 14 % damping ratio, is identified as an intra-area mode involving nuclear power plants primarily at BSL380 and MVL380, with contributions from a hydrogen plant at BSL380. Despite being well-damped, it indicates localized events in the system. The participation factors can be seen in Figure B.10.

In summary, sustained oscillations due to fast-acting power balancing are primarily due to interconnections, which, despite providing increased inertia and kinetic energy, pose oscillatory risks. Accurate interconnection estimations are crucial for understanding oscillations, which may vary based on the connected grid's operational characteristics and control systems. Further research should incorporate more precise interconnected system characteristics. Research on damping inter-area oscillations

has been extensive [188, 193, 194], with methods including FACTS devices, HVDC links, and wide-area monitoring control, alongside power system stabilizer improvements [195].
Low-damping intra-area modes between grid-forming converters, with dampings as low as 2.56 %, present stability risks, necessitating further analysis to fully understand their impact under various operational conditions.

Next, the potential for damping these inter-area modes using grid-forming converters was analyzed, focusing specifically on the Virtual Synchronous Machine (VSM) controller. For this analysis, the second PV converter located at ZL380 was chosen due to its proximity to the interconnection GNA380, which shows the highest participation in the inter-area mode 122. This converter is the closest to the interconnection, though it is not directly connected to the same bus, with one bus separating them.

The damping coefficient D_p of the VSM was initially set to 100. Simulations were subsequently conducted with D_p values of 1000 and 10 to evaluate the impact of different damping levels. The results, showing the most critical eigenvalues for the three levels of grid-forming damping, are presented in Figure 7.15. A comparison of the most critical modes for different damping levels is provided in Figure 7.16. The mode index in PowerFactory is sorted based on the real eigenvalue level, allowing modes to change the index between different simulations. The mode index was kept consistent with the original scenario to facilitate easier identification and analysis, with results identified using the frequency instead.

The results indicate that increasing the damping of the grid-forming converter generally has a positive impact on most inter-area modes, with no impact observed on the intra-area grid-following oscillation modes. A slightly improved damping is noted for the intra-area mode at 1.2195 Hz. For inter-area oscillations, a slight decrease in the damping ratio is observed for the mode at 0.326 Hz (mode 86), with the best damping ratio found at the original D_p level and the worst at D_p = 1000. The differences between the damping changes range around 0.45 %, indicating a minor variation, yet critical as it approaches the 5 % threshold discussed earlier, thus requiring careful observation.

However, while a small deviation is noted in the damping ratios for mode 86, larger differences are evident for modes 124 and 129. The differences between the best and worst damping ratios range from 9 % to 10.5 % for the respective modes. For mode 122, which is closest to the converter with the parameter change, a small difference of 1.05 % is observed, favoring the highest D_p value. Overall, the total damping of the inter-area oscillations improved with an increased damping parameter in one of the grid-forming values and decreased with a reduced D_p. Differences ranging from -0.45 % to 10.5 % in the damping ratios were found, with only the lowest difference favoring the lower D_p value. All values remained within the 3 % limit mark of the damping ratios. However, the lowest D_p value resulted in some modes beneath the 5 % limit.
This variability suggests that these grid-forming converters do not immediately induce inter-area oscillations but can effectively mitigate dangerously low damping modes. Interestingly, the lowest D_p value resulted in a newly created poorly damped local mode. This was attributed to the grid-forming converter undergoing the sensitivity analysis, which highlighted controller issues for low D_p values and warranted further research. This is more explicitly highlighted in case study B.

The analysis demonstrates the significant potential for improving damping ratios with grid-forming converters and their controller parameters concerning inter-area oscillations. While one mode saw a minor reduction of 0.45 %, increases of up to 10 % were observed, indicating substantial impacts from controller adjustments. These possibilities should be investigated further, with a deeper analysis of the different controller parameters of grid-forming converters undertaken in Case Study B.

7.2.5. Results and Findings

The initial analysis compared six cases characterized by varying levels of onshore and offshore wind and solar PV integration under different system load conditions. Three different perturbations were implemented and analyzed: a generator outage, a short-circuit on a tie line between different voltage levels, and an approximate 5.6 and 10 GW load increase and decrease, respectively. The analysis revealed that all cases exhibited good fast-active power balancing responses, with the inclusion of grid-forming converters contributing significantly to the overall response.

The frequency response comparison for the six cases showed that the two PV cases without any offshore wind converter penetration resulted in worse responses for all three events, leading to higher overall allowed frequency deviations when not all converters were included for ancillary services. The overall frequency response was similar across all cases relative to their respective system load levels. Lower system loading and subsequent load decreases led to higher frequency deviations compared to highly loaded systems with load increases. This indicates the system's dependence on flexibility resources, particularly in low-loaded scenarios. The analysis also demonstrated relatively good damping for the different cases, although small oscillations were evident after the 20-second simulation window. Overall, the system fulfilled the margins associated with frequency deviations in fast-active power balancing scenarios, indicating a well-adjusted system even with lower overall inertia.

An oscillatory behavior analysis was also conducted, identifying different oscillatory modes through eigenvalue analysis in PowerFactory. The modes associated with grid-following converters and their control system representations of the stator and rotor showed relatively poor damping ratios below 3 %. Intra-area oscillations were identified with damping ratios ranging from approximately 2.56 % to 2.99 % and a frequency of around 0.636 Hz, highlighting potential stability issues if grid-following converters are not properly analyzed and integrated.

Five inter-area oscillations were identified for the associated interconnections, with frequencies ranging from 0.3236 to 0.5109 Hz and damping ratios between 5.0978 % and 7.8621 %. These inter-area modes are typically associated with large interconnected power systems and are likely to be present in future power systems if interconnections continue to increase. An analysis was performed to assess the impact of varying the VSM controller's damping parameter on the associated modes' damping. The results showed significant potential for grid-forming converter damping, with simple control adjustments leading to substantial improvements in damping ratios. Conversely, lowering the parameter resulted in a minor decrease in damping ratios for most modes. The benefits of increasing the damping parameter far outweighed the negative impacts of reducing it.

Overall, the potential to enhance system damping using grid-forming converters is substantial and will be further explored in Case Study B. With advancements in power system technology, such as wide-area monitoring, protection, control, real-time data analysis from PMUs, and integration with artificial intelligence and deep learning, the implementation of smart, dynamic controls using flexible grid-forming converters could play a crucial role in the evolving power system environment. Therefore, the impact of grid-forming converter controllers warrants further exploration, given their significant potential advantages over traditional slow-acting controllers. Continued research is essential to fully explore this potential.

7.3. Case Study B

7.3.1. Introduction

As the share of PEI generation in the power system increases, large-scale stability studies of future systems are vital to ensuring safe operation. Grid-forming converters have been identified as key components in bridging the stability gap with an increased share of PEI generation. This case study serves as an important part of research into grid-forming converters for future systems, analyzing different aspects of these converters and their characteristics in system operation.

Case study B utilizes the synthetic model of the Dutch power system towards 2050, as documented in chapter 5 and chapter 6. This study emphasizes the critical role of these converters in enhancing future stability as power systems increasingly integrate renewable energy sources. The use of this model allows for extensive analysis concerning frequency and eigenvalue analysis, as well as a multitude of other studies facilitated through the use of PowerFactory, Python, and Excel.

The first part of the study compares three grid-forming technologies: Virtual Synchronous Machines (VSMs), Synchroverters, and Droop controllers. It evaluates their responses to three disturbances: an outage, a load change, and a short circuit, at three levels of grid-forming penetration. As the share of the grid-forming penetration levels increases, identifying differences between different grid-forming converter controller technologies is vital to better understand the intrinsic nature of controller selection and identifying each respective controller's positive and negative impacts. A similar study was done for the three controllers in [76]; however, this study is extended here with respect to the large-scale system nature of the synthetic model, while the research was done on a two-area four-generator model, representing a smaller system. Insights from [76] enrich this comparative analysis, offering a deeper understanding of the impacts on system stability and a comparative analysis with respect to the difference in these controllers on large- and small-scale system stability studies.

Building upon foundational research, such as [196], which examined the balance between grid-forming and grid-following converters in wind farms and their respective stability criteria, this study extends the analysis to broader system scales, specifically the Dutch power system. It analyzes penetration levels ranging from 25 %, 50 %, and 85 %, assessing their effects on system robustness and stability. As the reduction of traditional synchronous generation decreases in the system, the impact of different levels of grid-forming penetration in large-scale studies is vital to guarantee stability and safe operation in future systems. By analyzing three different grid-forming penetration levels, a better understanding of how these levels impact power system stability in large-scale systems is achieved, allowing for a better understanding of future systems.

Additionally, the study examines the intrinsic impact of grid-forming converters through a detailed analysis of their main controller parameters. The study highlights potential benefits and uncertainties that could influence future stability enhancements by assessing these parameters across all three controllers. Given the estimated growth of grid-forming converters in future systems, a better understanding of their critical parameters and how these influence system characteristics is crucial for fully optimizing these converters. A small-scale analysis conducted for Case Study A highlighted interesting findings regarding the damping of inter-area modes by tuning one converter, underscoring the impact these converters can have. Extending this to a large-scale system model and comparing the most important parameters for all three controllers, a thorough and significant analysis is conducted, again extending the research done in [76] and case study A.

Both frequency and eigenvalue analyses were conducted to evaluate stability, with frequency analysis applied to all three event types and eigenvalue analysis focused on the load event. This dual-method approach provides comprehensive insights into these systems' fast-active power-balancing capabilities.

Detailed system and simulation setup models are provided in the methodology for easier reenactment and an overview of the study at hand.

Overall, the detailed examination of grid-forming parameters in this large-scale setting offers valuable

insights into future stability scenarios for the Dutch power system and illustrates broader implications for grid-forming converter parameters on system performance. This study serves as a benchmark in future large-scale grid-forming converter analysis and the synthetic power system model, with its valuable resources utilized through PowerFactory, Python, and Excel, allowing for a model that can be utilized for important research across various system characteristics.

7.3.2. Methodology

This section outlines the methodologies and setups utilized to investigate the dynamic stability of power systems influenced by different penetration levels, controllers, and controller parameter sensitivities. The focus was particularly on scenarios integrating high levels of renewable energy sources controlled by different controllers, including offshore and onshore wind as well as solar PV. The study leveraged the "Dutch power system towards 2050 model," as detailed in chapter 5 and chapter 6, which is essential for understanding the dynamic stability under different operational characteristics. This dynamic model utilizes the extensive dynamic properties enabled by PowerFactory, combined with automated tasks through Python and Excel, facilitating easy adjustments of varying system parameters. This setup allows for easy recreation of the system simulations conducted in this case study and provides a solid foundation for future work.

A flowchart depicting the simulation setup for the different analyses conducted in this case study is presented in Figure 7.17. The setup begins by selecting the case utilized for the studies. Due to the high relevance of offshore wind and its role in the transition towards more renewable energy resources, Case A, as identified in Case Study A, was utilized. This case comprises large-scale offshore wind generation levels and includes a peak demand scenario, further used for all simulations conducted.

Next, the grid-forming penetration levels were selected, with three different levels under scrutiny: 25 %, 50 %, and 85 % penetration. Following this, one of the three grid-forming converters (VSM, Droop, and Synchroverter) were selected. Subsequently, two options were available: either a controller parameter adjustment or the base values for the controller were selected and implemented from the Python and Excel scripts to the synthetic model, which was used to execute the simulations. Three alternate routes were possible from the simulations: a new selection of grid-forming penetration level, a new controller selection, or a controller parameter selection change. Each of these options was then implemented again before new simulations were run. By following this procedure, the overall impact of various grid-forming operational characteristics was simulated and analyzed.

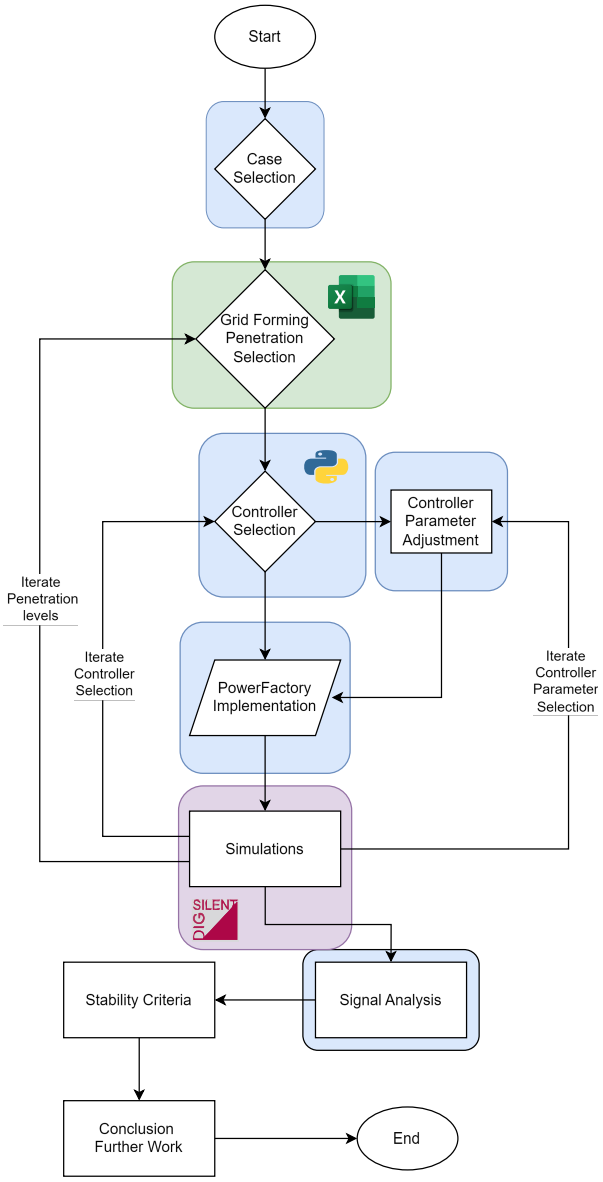


Figure 7.17: Flowchart illustrating the comprehensive simulation setup process for evaluating the impact of different grid-forming technologies, penetration levels, and controller parameters on power system stability in Case Study B. The pink color indicates operation performed by PowerFactory, green by Excel, and blue by Python.

The first analysis compared the three different controllers: the VSM, Synchroverter, and Droop controller. These were implemented and evaluated for the three disturbances, with results analyzed through frequency and eigenvalue analysis. This study further explored the impact of different grid-forming converters on system stability. A similar comparison was made in [76], but this research extends to a large-scale model, representing an advancement in the analysis. The implementation of different controllers to allow for easy comparison was facilitated through the Python code associated with the synthetic model. More information regarding this can be found in Appendix C. This controller comparison was done for all three events and for the three respective grid-forming penetration levels of 25 %, 50 %, and 85 %.

The second analysis explored the effects of varying penetration levels of grid-forming converters, specifically 25 %, 50 %, and 85 %, representing low, medium, and high penetration levels. This analysis compared the three different penetration levels for all three controller types. The distinction between the first and second parts of the study lies in the comparative nature. In the first part, the controllers themselves were compared with varying levels of grid-forming penetration. The second part compared

the grid-forming penetration levels for the respective controllers. A Python script automated the differentiation of grid-forming percentages, controllers, and parameters, enabling rapid and efficient analysis. This script is documented and accessible in Appendix C, ensuring reproducibility and further exploration of the methodologies employed.

The third analysis examined the different main controller parameters of the grid-forming converters identified in section 2.2. These parameters were analyzed using a sensitivity analysis approach, comparing increased and decreased values of these parameters on overall system stability. The parameters were changed for all the grid-forming converters in the system, highlighting the systematic impact these converters can have. The different parameter values under scrutiny are highlighted in Table 7.6. These parameter values were selected randomly, with a representative increase and decrease from the baseline values utilized to accurately depict the differences associated with parameter changes. Another part of the Python script was utilized to automate the integration of different controller parameters for the three controllers, as highlighted in Appendix C. A similar analysis was also done in [76], comparing important parameters of the three controllers. However, this study stands out by allowing a large-scale system impact analysis of these parameters. For the parameter sensitivity analysis, a grid-forming penetration level of 50 % was used for simplicity.

Table 7.6: Controller Parameters for the three Grid-forming converters, new values highlighted in parentheses

Controller	T_a	D_p	K_q	D_q	mp	m_q
VSM	3 (1, 10)	100 (10, 1000)	-	-	-	-
Synchroverter	3 (1, 10)	100 (10, 1000)	1000 (500, 2000)	20 (10, 40)	-	-
Droop	-	-	-	-	0.01 (0.001, 0.1)	0.05 (0.01, 0.1)

A flowchart depicting the simulation process for the three different analyses is given in Figure 7.18.

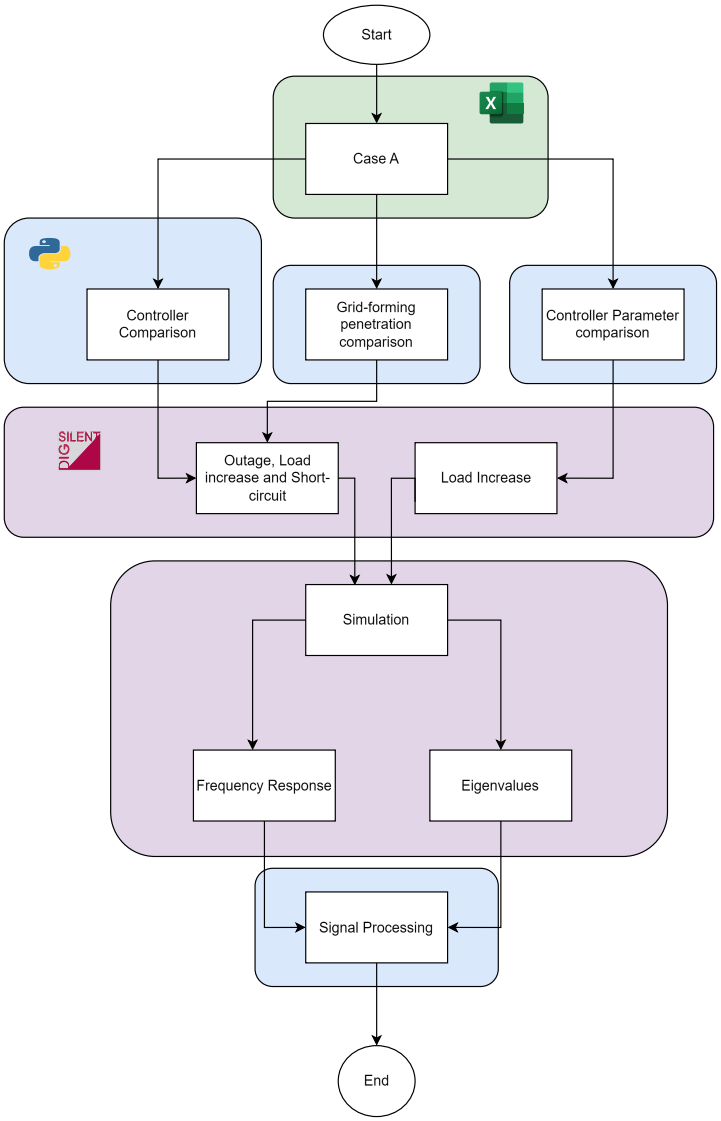


Figure 7.18: Detailed flowchart of simulation process used in Case Study B. The pink color indicates operation performed by PowerFactory, green by Excel, and blue by Python.

The dynamic simulations were conducted using the synthetic model created in DIgSILENT PowerFactory, as outlined in chapter 5 and chapter 6, with Python and Excel integrated to adjust various system characteristics such as grid-forming penetration levels and controller selection and settings. This integration facilitated the manipulation and analysis of data exported from Python and Excel into PowerFactory, which aids in adjusting stability parameters easily.

Frequency and eigenvalue analyses were employed to assess system stability. For frequency response analysis, metrics such as maximum frequency deviation, Rate of Change of Frequency (ROCOF), and settling time were considered, alongside damping ratios and oscillatory performance of critical modes for the eigenvalue analysis. The handling and presentation of results and the setup of different system parameters were primarily done through Python, utilizing PowerFactory's advanced export features.

This study included simulations of three distinct disturbance events—an outage, a short-circuit, and a load increase—each designed to test the system's resilience under different stress conditions. Here, the same disturbance configurations used in Case Study A were implemented: an outage event consisting of the disconnection of the 2 GW nuclear power plant at MVL380, a short-circuit on the tie-line of the connection between the 380 kV and 150 kV grid in Noord-Brabant (B150), and a load increase of 5.6 GW.

Insights into the implementation of various grid-forming converters and their control systems are provided in section 2.2, discussing the operational characteristics and settings for each converter type. This enhances the understanding of their differential impacts on system stability.

Extensive cross-referencing was performed between original data from PowerFactory and processed results in Python to ensure data integrity and the validity of simulation results. Measures are implemented to maintain consistent system parameters across different simulation scenarios, enhancing the reliability of the study's findings.

This comprehensive approach seamlessly integrated each aspect of the methodology, facilitating a clear understanding of how the study was conducted and ensuring the reproducibility of the results.

7.3.3. Results

7.3.3.1. Comparative Analysis of Grid-Forming Controllers

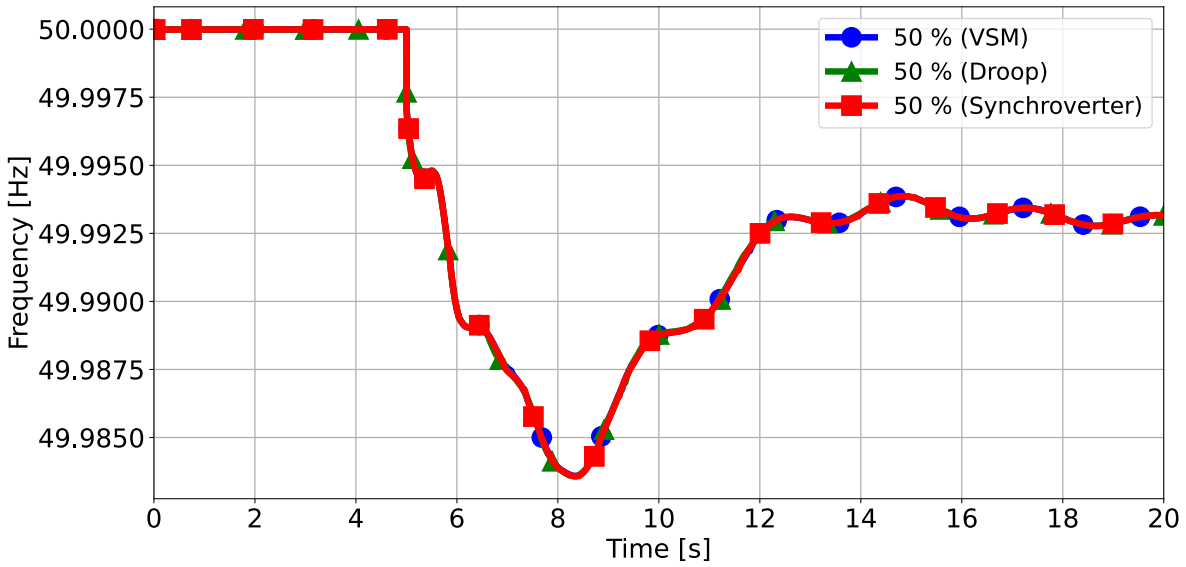


Figure 7.19: Frequency response for different grid-forming controllers with 50 % grid-forming penetration following a load event occurring at 5 seconds.

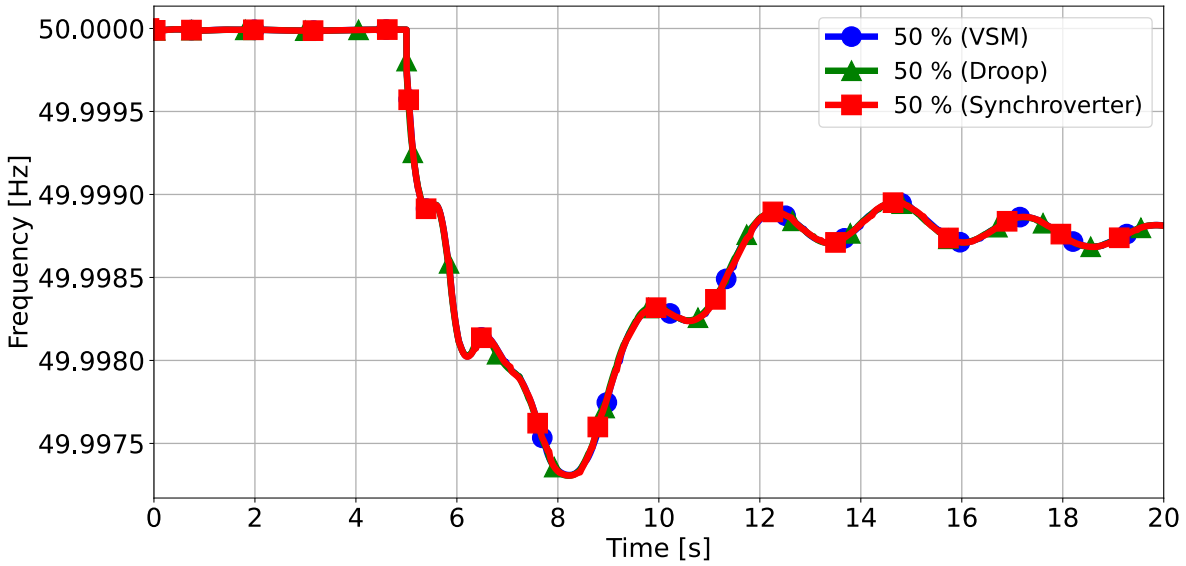


Figure 7.20: Frequency response for different grid-forming controllers with 50 % grid-forming penetration following an outage event occurring at 5 seconds.

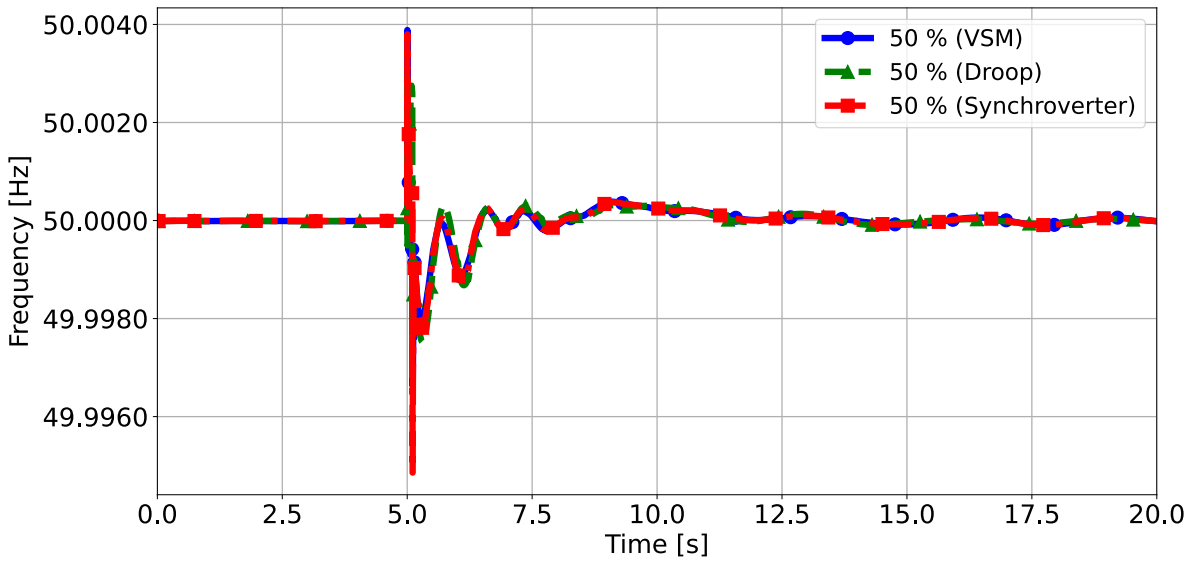


Figure 7.21: Frequency response for different grid-forming controllers with 50 % grid-forming penetration following a short-circuit event occurring at 5 seconds.

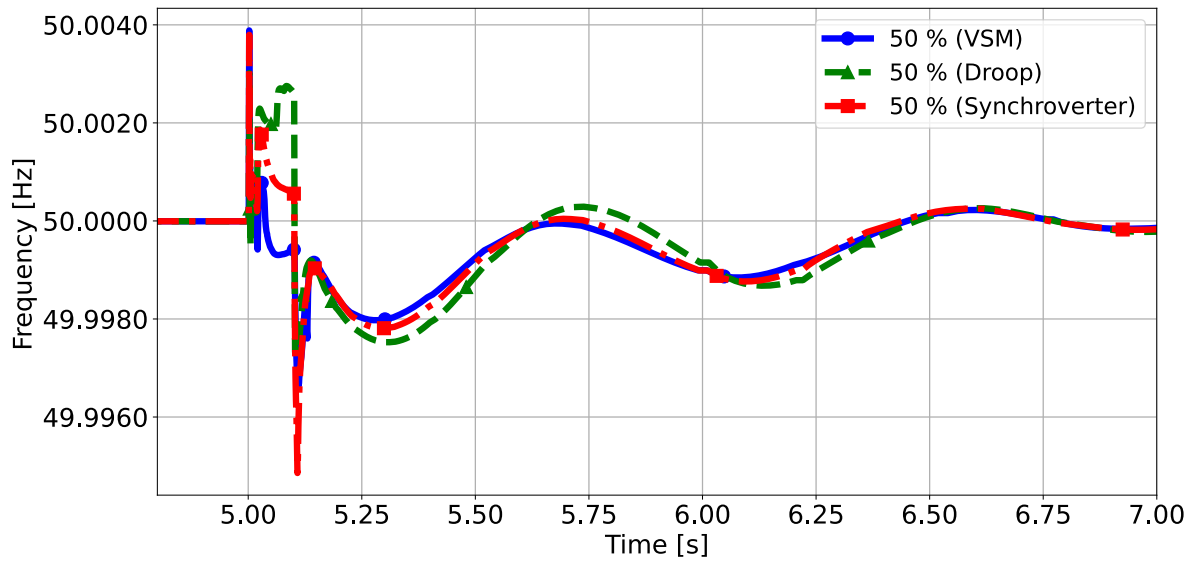


Figure 7.22: Frequency response for different grid-forming controllers with 50 % grid-forming penetration following a short-circuit event occurring at 5 seconds (zoomed view).

Table 7.7: ROCOF, Frequency deviation, and Frequency Nadir for the three controller comparisons with 50 % grid-forming penetration.

Outage Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (50 %)	49.9973	1.4	1.0	2.7
Droop (50 %)	49.9973	1.4	1.0	2.7
Synchroverter (50 %)	49.9973	1.2	1.0	2.7

Short-Circuit Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (50 %)	49.9961	16.1	6.3	3.9
Droop (50 %)	49.9970	26.6	5.5	3.0
Synchroverter (50 %)	49.9949	33.1	9.8	5.1

Load Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (50 %)	49.9836	6.6	4.9	16.4
Droop (50 %)	49.9836	5.8	4.7	16.4
Synchroverter (50 %)	49.9836	6.2	4.8	16.4

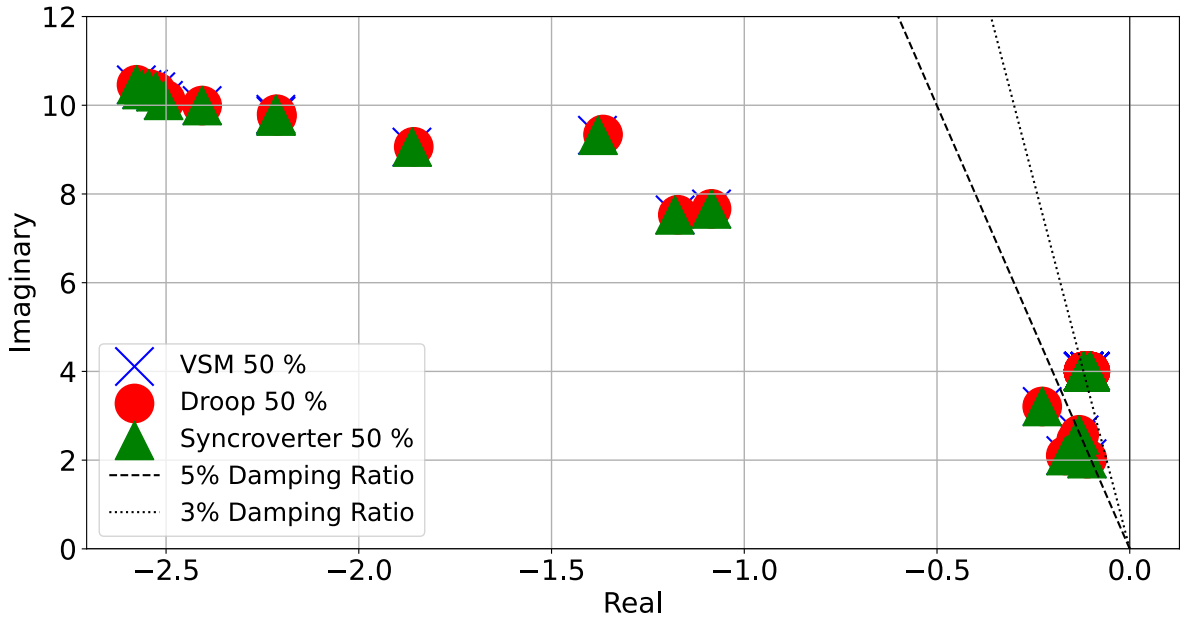


Figure 7.23: Eigenvalue plot comparing the three different controllers at a 50 % grid-forming penetration level, including all modes under a 50 % damping ratio.

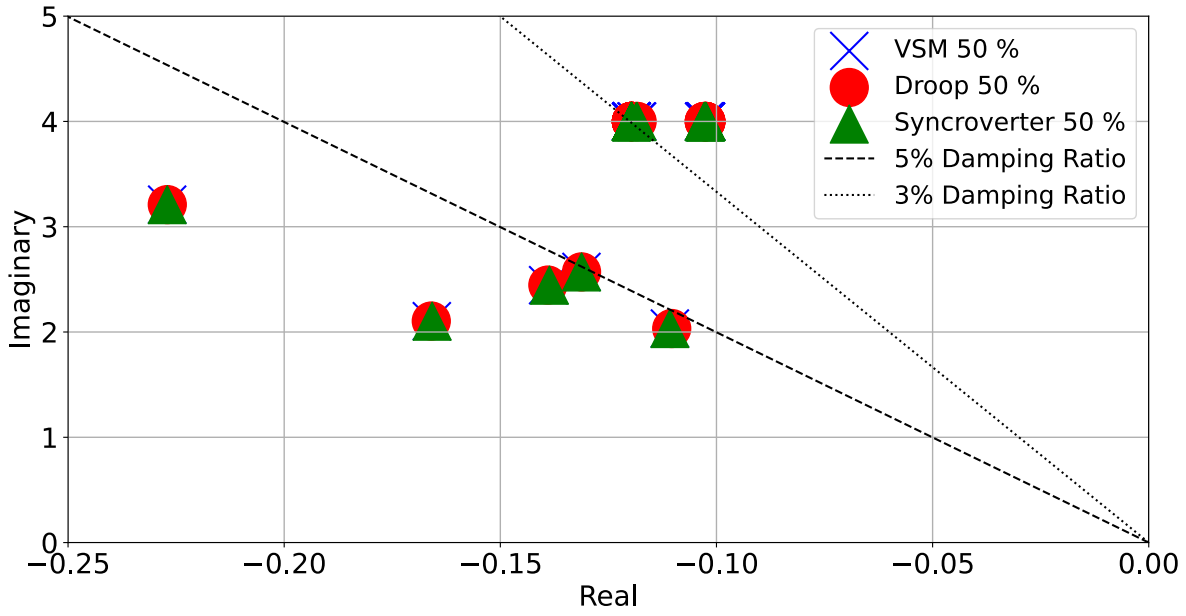


Figure 7.24: Eigenvalue plot comparing the three different controllers at a 50 % grid-forming penetration level, zoomed in on the most critical modes.

Table 7.8: Comparison of Eigenvalues, Frequencies, and Damping Ratios for VSM, Droop, and synchroverter with 50 % grid-forming penetration.

Mode index	Eigenvalue	Frequency (Hz)	Damping Ratio (%)		
			VSM	Droop	Sync
62	$-0.1024 - 3.9989i$	0.6364	2.5610	2.5610	2.5610
121	$-0.1197 + 3.9996i$	0.6366	2.9906	2.9906	2.9906
122	$-0.1312 - 2.5709i$	0.4092	5.0978	5.0969	5.0967
86	$-0.1108 - 2.0332i$	0.3236	5.4405	5.4198	5.4402
124	$-0.1390 - 2.4449i$	0.3891	5.6739	5.6742	5.6618
192	$-0.2271 - 3.2098i$	0.5109	7.0568	7.0554	7.0563
129	$-0.1658 - 2.1029i$	0.3347	7.8622	7.8632	7.8587
254	$-1.0850 - 7.6626i$	1.2195	14.0196	14.0160	14.0176

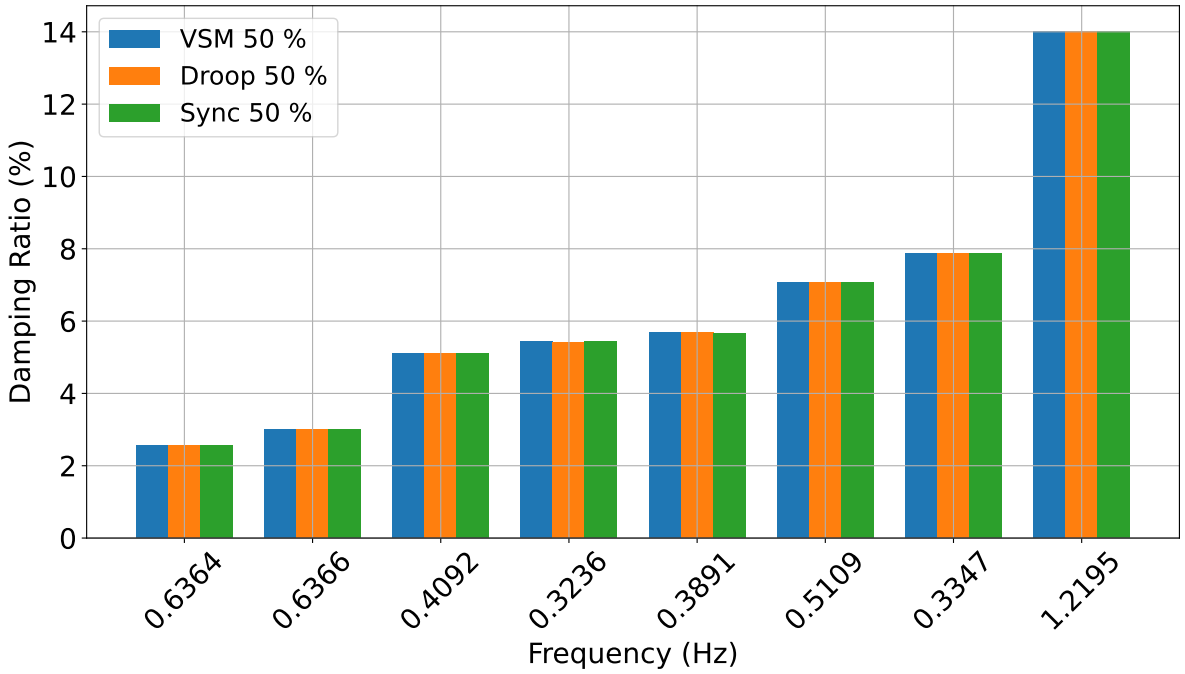


Figure 7.25: Damping ratio and frequency for the most critical modes for the three different controllers comparison.

7.3.3.2. Comparison of Grid-Forming Penetration Levels

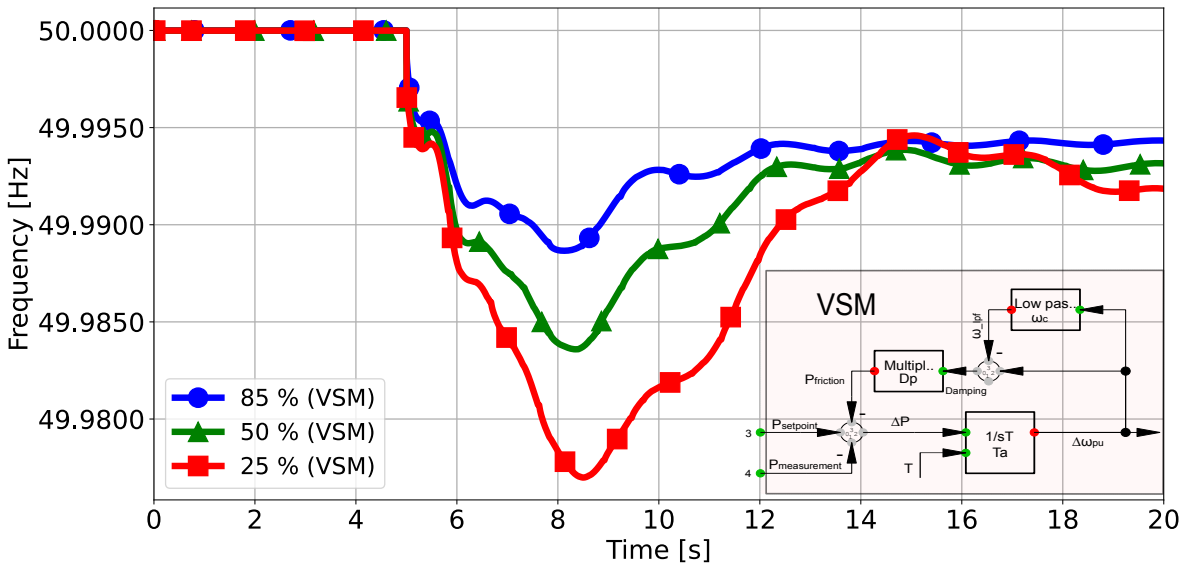


Figure 7.26: Frequency response for different grid-forming penetration levels with the VSM controller following a load event occurring at 5 seconds, The VSM frequency control loop is highlighted in pink.

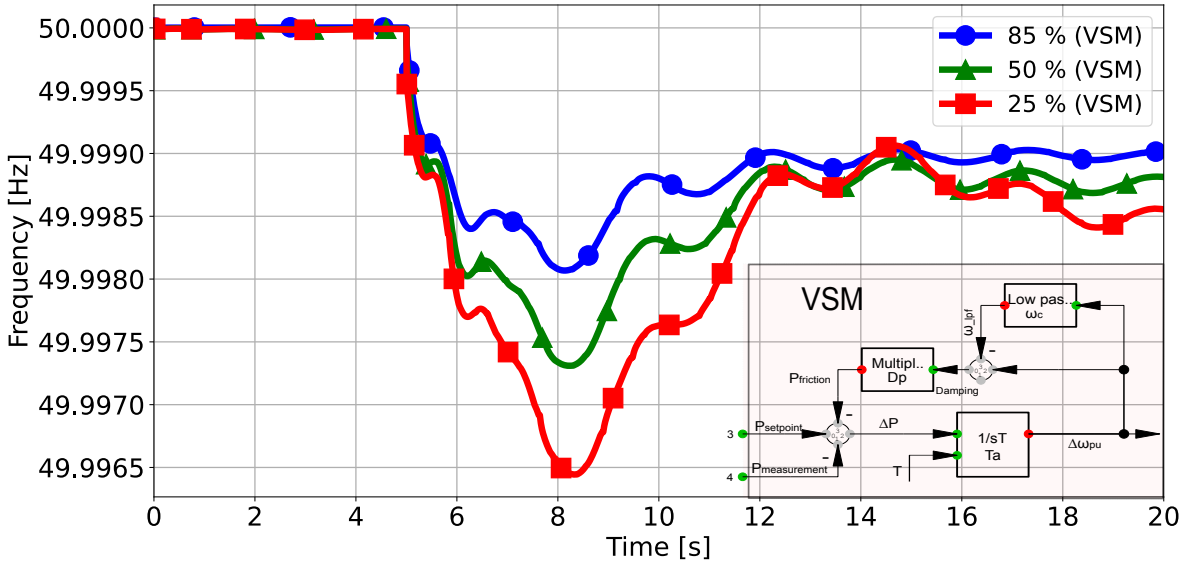


Figure 7.27: Frequency response for different grid-forming penetration levels with the VSM controller following an outage event occurring at 5 seconds. The VSM frequency control loop is highlighted in pink.

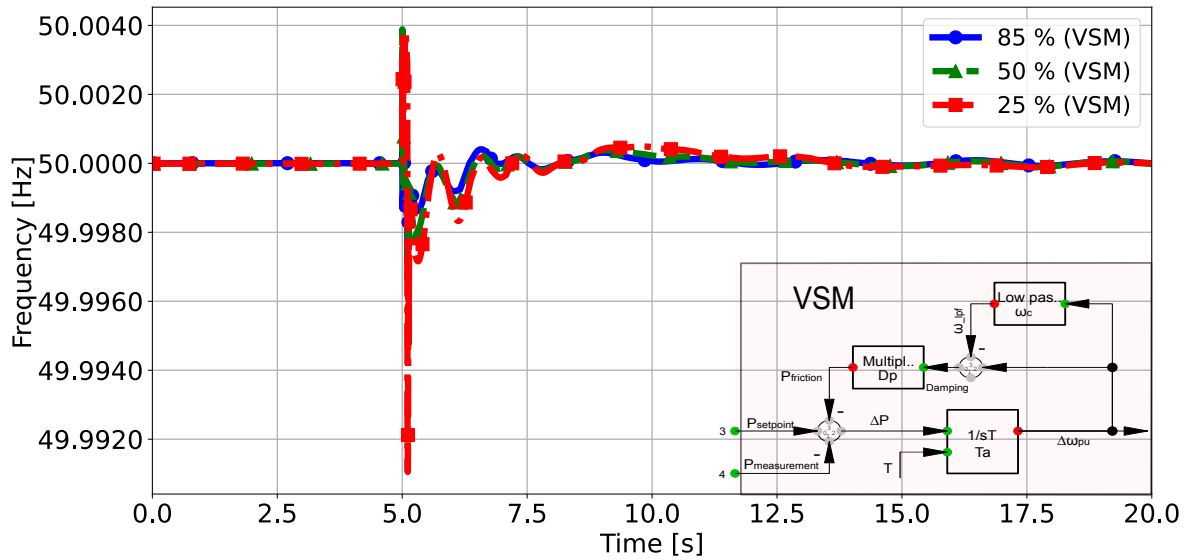


Figure 7.28: Frequency response for different grid-forming penetration levels with the VSM controller following a short-circuit event occurring at 5 seconds. The VSM frequency control loop is highlighted in pink.

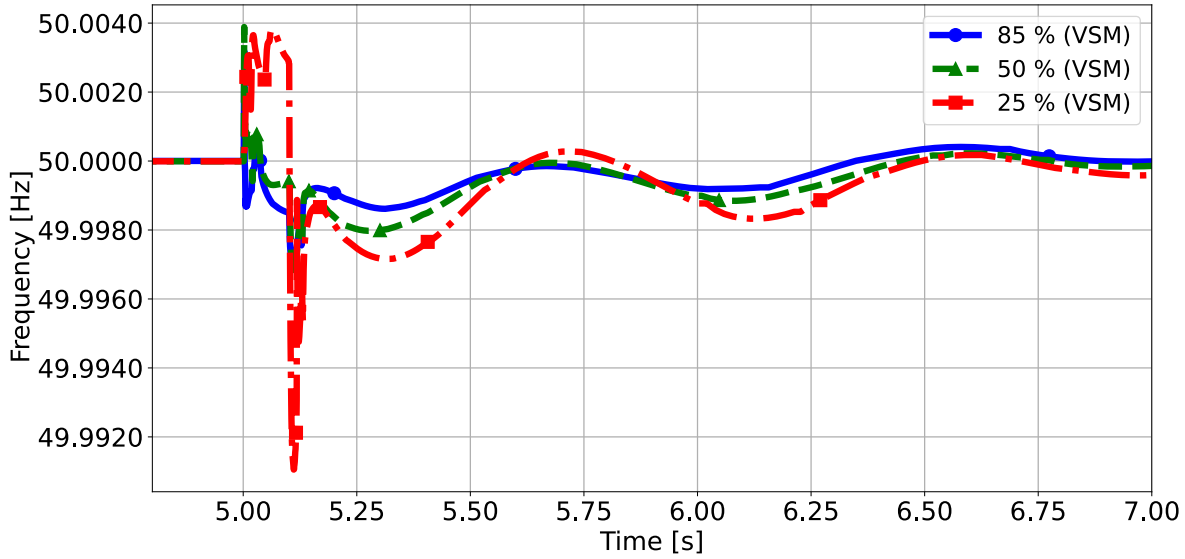


Figure 7.29: Frequency response for different grid-forming penetration levels with the VSM controller following a short-circuit event occurring at 5 seconds (zoomed view).

Table 7.9: ROCOF, Frequency Deviation, and Frequency Nadir for different levels of grid-forming penetration with the VSM controller.

Outage Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85 %	49.9981	1.0	0.7	1.9
50 %	49.9973	1.4	1.0	2.7
25 %	49.9964	1.6	1.1	3.6

Short-Circuit Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85 %	49.9968	18.8	5.3	3.2
50 %	49.9961	16.1	6.3	3.9
25 %	49.9911	68	17.5	8.9

Load Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85 %	49.9887	4.4	3.7	11.3
50 %	49.9836	6.6	4.9	16.4
25 %	49.9770	7.2	5.8	23.0

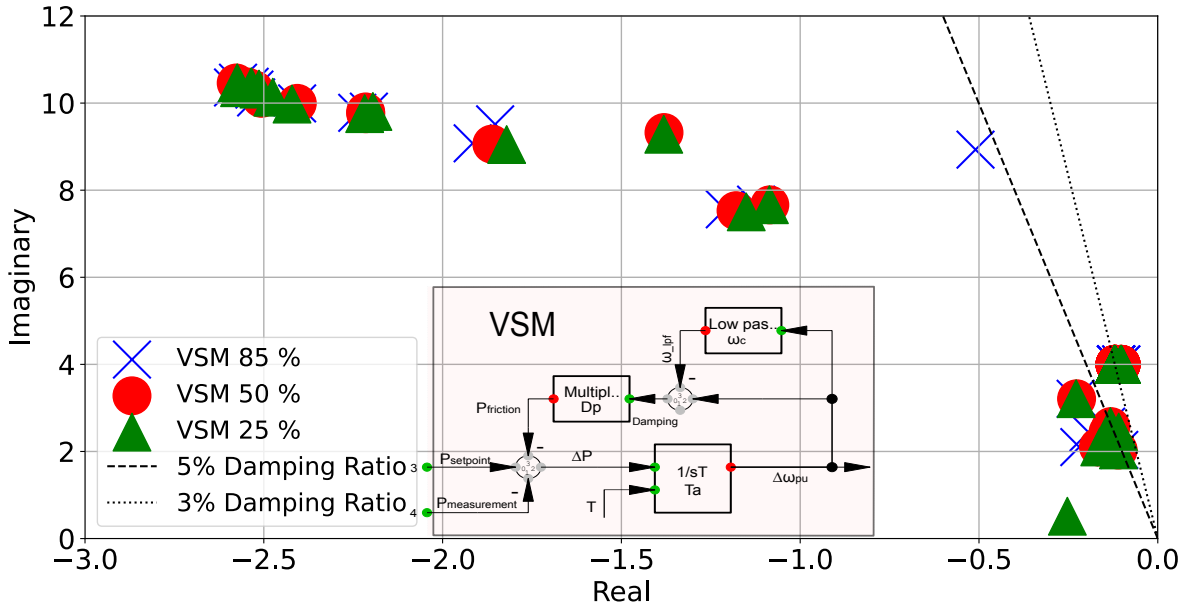


Figure 7.30: Eigenvalue plot comparing the three grid-forming penetration levels with the VSM controller, including all modes under a 50 % damping ratio. The VSM frequency control loop is highlighted in pink.

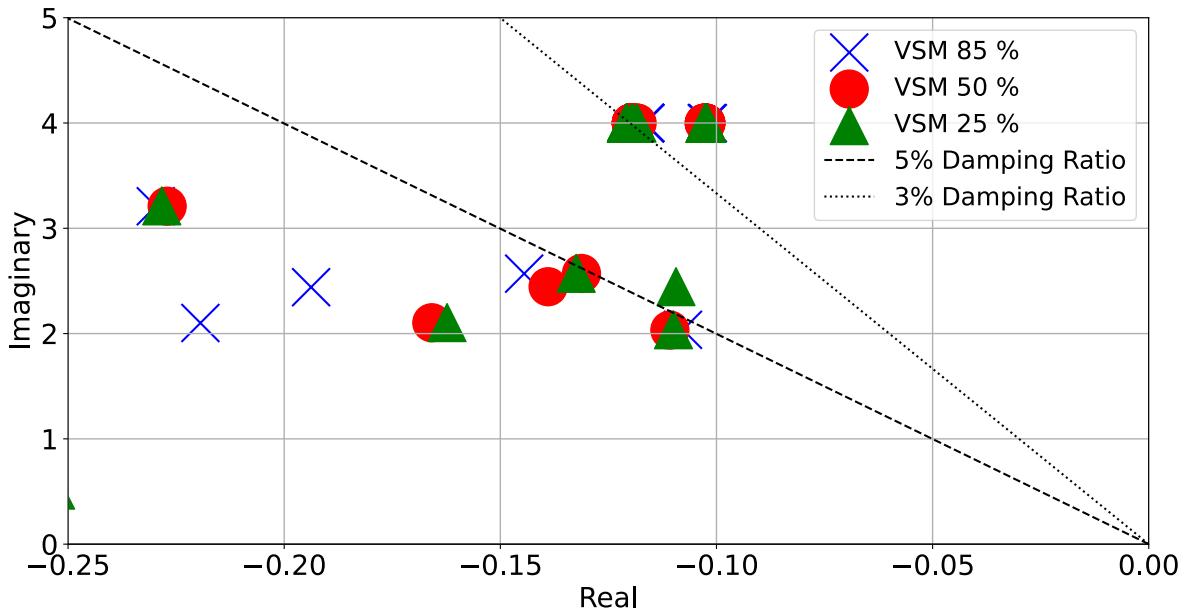


Figure 7.31: Eigenvalue plot comparing the three grid-forming penetration levels with the VSM controller, zoomed in on the most critical modes.

Table 7.10: Comparison of Eigenvalues, Frequencies, and Damping Ratios for different grid-forming penetration levels with the VSM controller.

Mode index	Eigenvalue	Frequency (Hz)	Damping Ratio (%)		
			85%	50%	25%
62	$-0.1024 - 3.9989i$	0.6364	2.5485	2.5610	2.5565
121	$-0.1197 + 3.9996i$	0.6366	2.9125	2.9906	3.0190
122	$-0.1312 - 2.5709i$	0.4092	5.6146	5.0978	5.1421
86	$-0.1108 - 2.0332i$	0.3236	5.2854	5.4405	5.3907
124	$-0.1390 - 2.4449i$	0.3891	7.9148	5.6739	4.4667
192	$-0.2271 - 3.2098i$	0.5109	7.1388	7.0568	7.0878
129	$-0.1658 - 2.1029i$	0.3347	10.3876	7.8622	7.6904
128	$-0.5097 - 8.9315i$	1.4215	5.6974	20.1386	20.1386

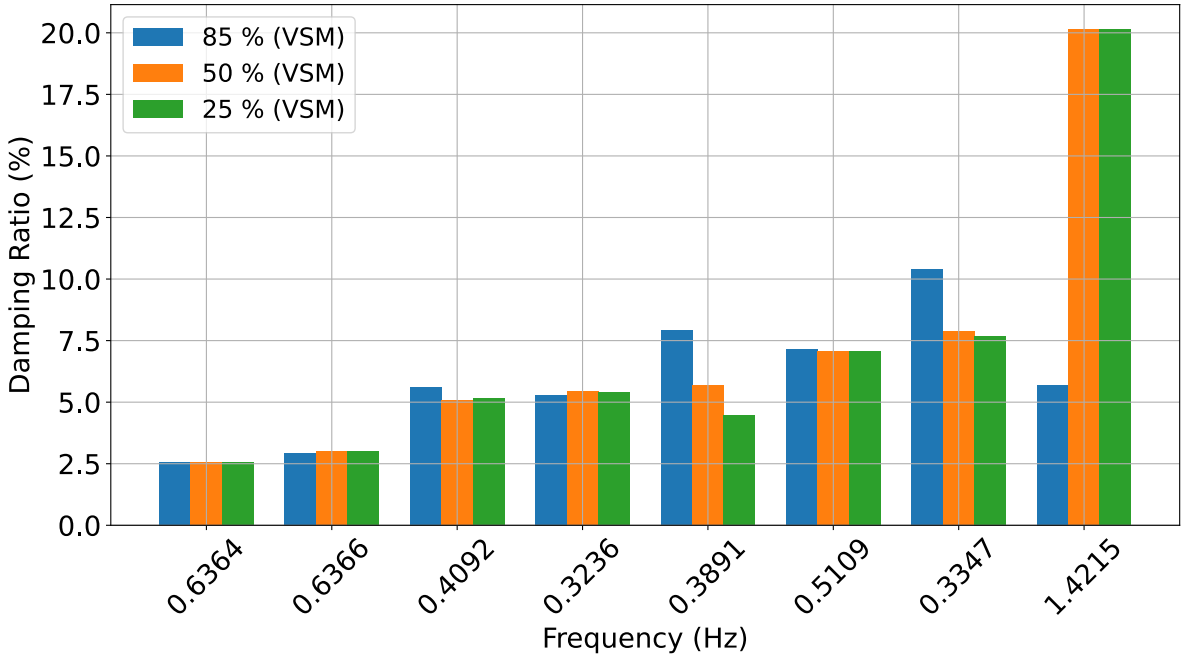


Figure 7.32: Damping ratio and related frequency for the most critical modes for the three levels of grid-forming penetration with the VSM controller.

7.3.3.3. Controller Parameter Sensitivity Analysis

VSM Controller l

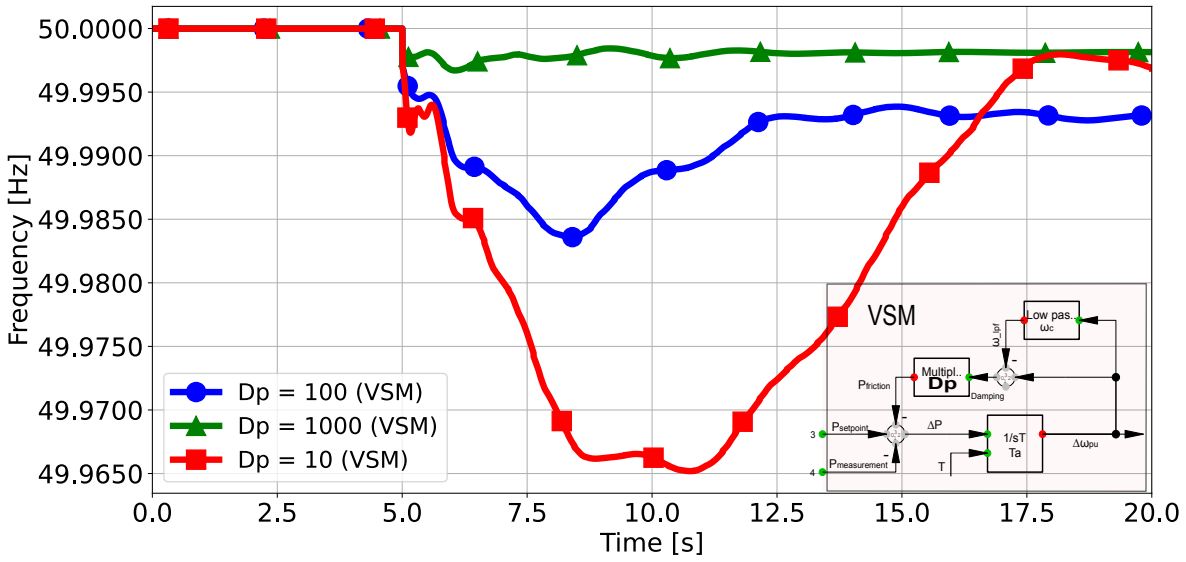


Figure 7.33: Frequency response for different D_p values for the VSM controller following a load event occurring at 5 seconds. The VSM frequency control loop is highlighted in pink.

Table 7.11: ROCOF, Frequency deviation, and Frequency Nadir for different D_p values for the VSM controller.

D_p (VSM)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
D_p = 100	49.9836	6.6	4.9	16.4
D_p = 1000	49.9967	7.4	2.2	3.3
D_p = 10	49.9652	17.6	7.4	34.8

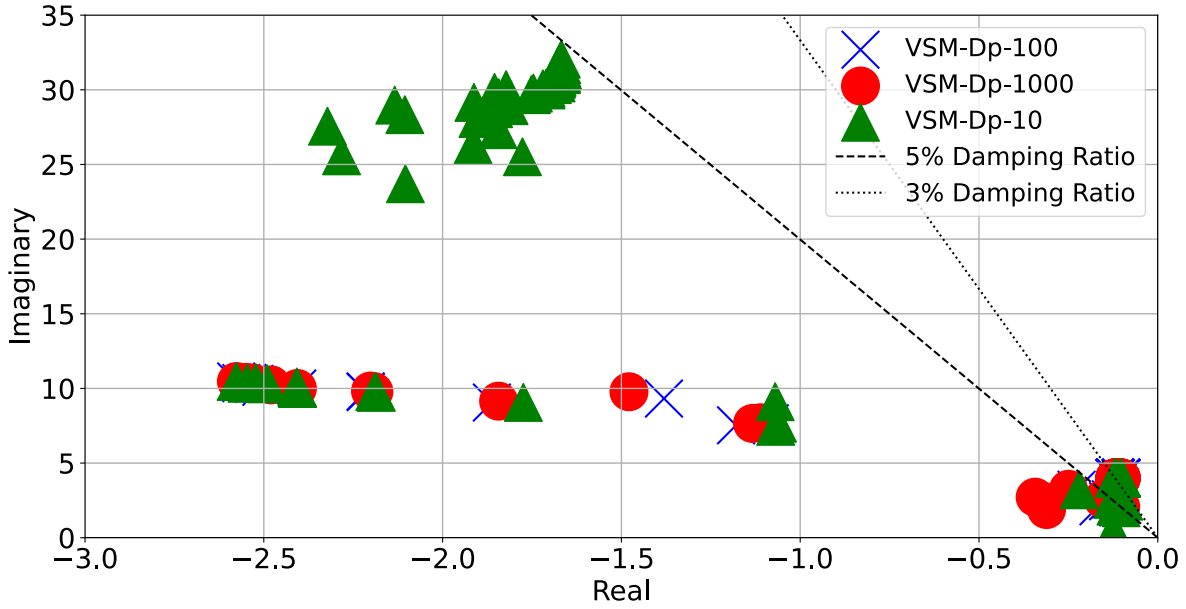


Figure 7.34: Eigenvalues for different D_p values for the VSM controller.

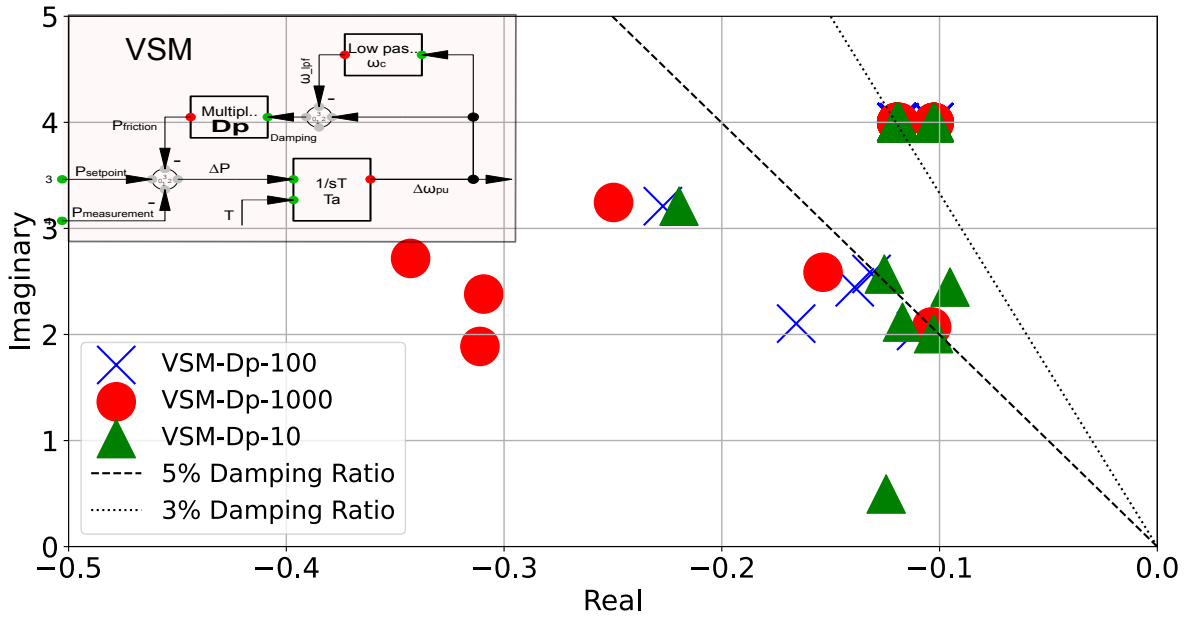


Figure 7.35: Eigenvalues for different D_p values for the VSM controller, zoomed in on the most critical modes. The VSM frequency control loop is highlighted in pink.

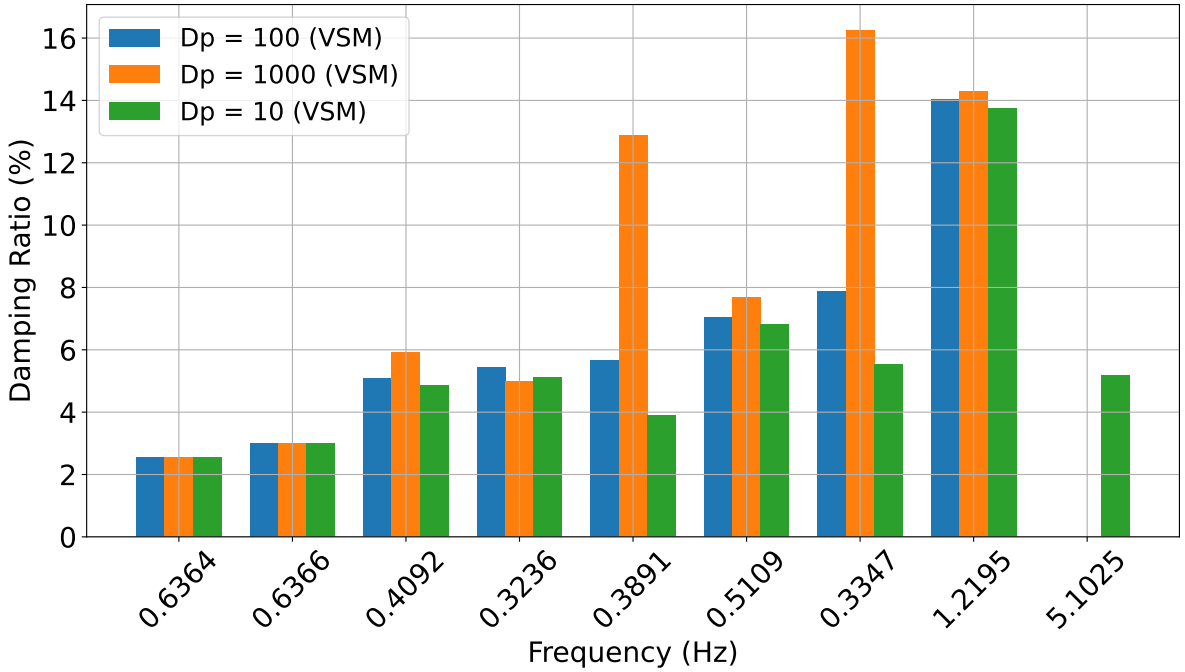


Figure 7.36: Damping ratio and corresponding frequency for different D_p values for the VSM controller.

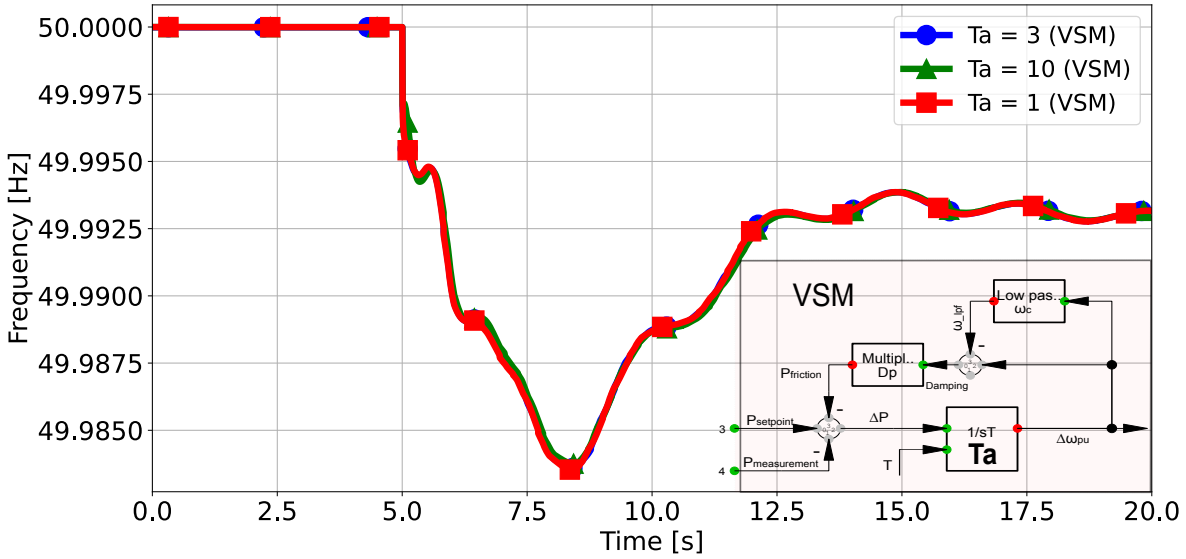


Figure 7.37: Frequency response for different T_a values for the VSM controller following a load event occurring at 5 seconds. The VSM frequency control loop is highlighted in pink.

Table 7.12: ROCOF, Frequency deviation, and Frequency Nadir for different T_a values for the VSM controller.

T_a (VSM)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
$T_a = 3$	49.9836	6.6	4.9	16.4
$T_a = 10$	49.9838	5.6	4.6	16.2
$T_a = 1$	49.9838	5.6	4.6	16.2

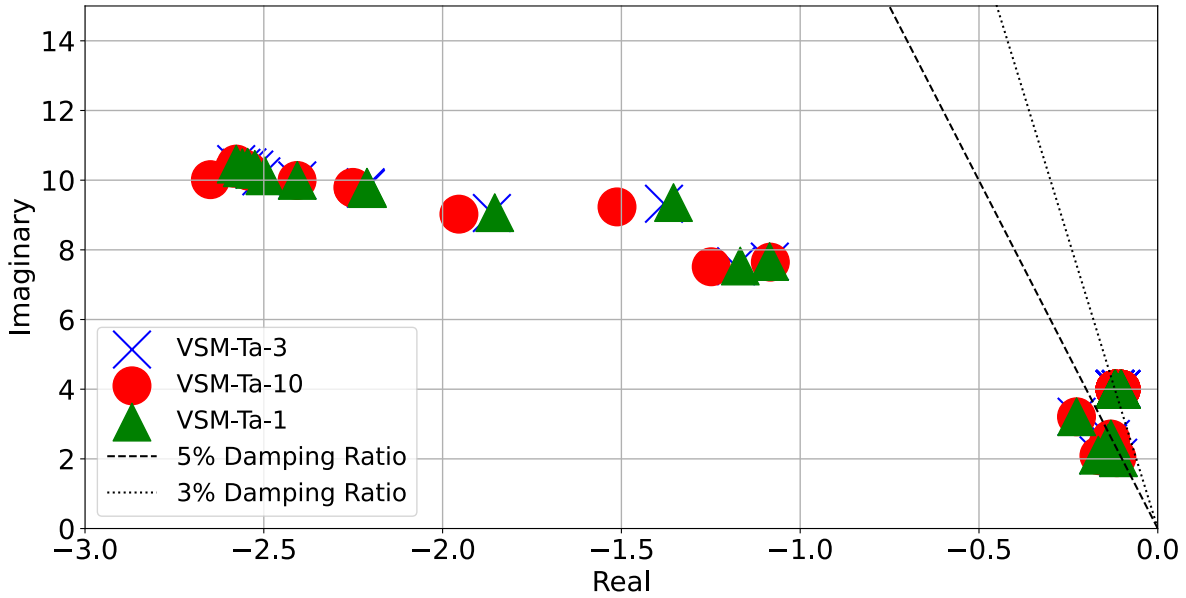


Figure 7.38: Eigenvalues for different T_a values for the VSM controller.

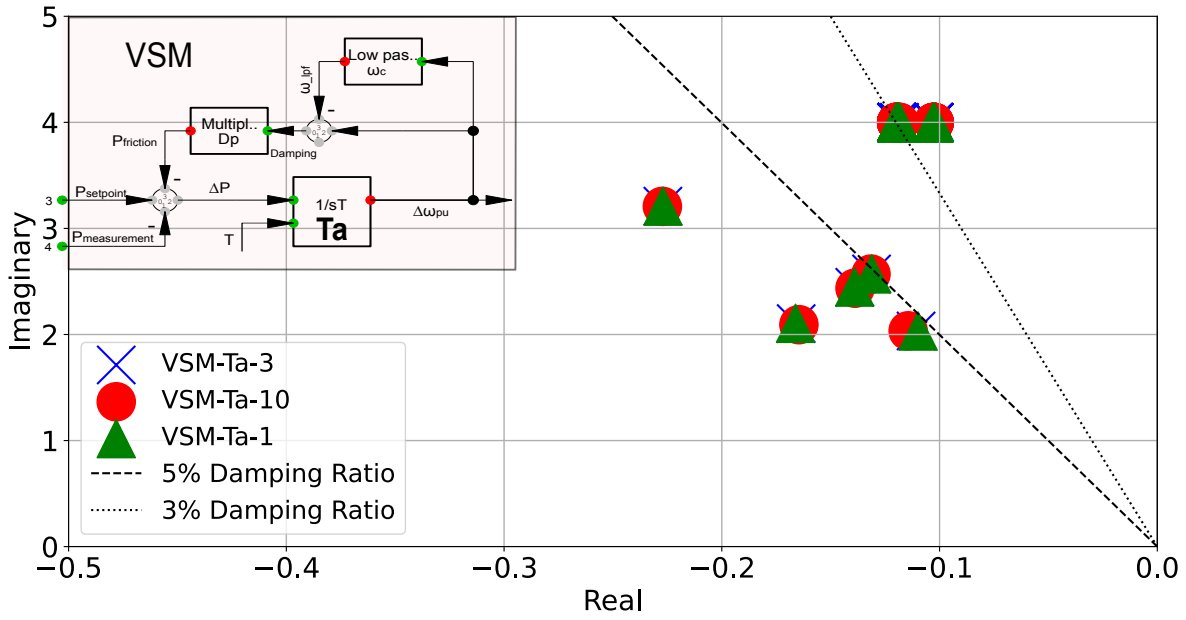


Figure 7.39: Eigenvalues for different T_a values for the VSM controller, zoomed in on the most critical modes.

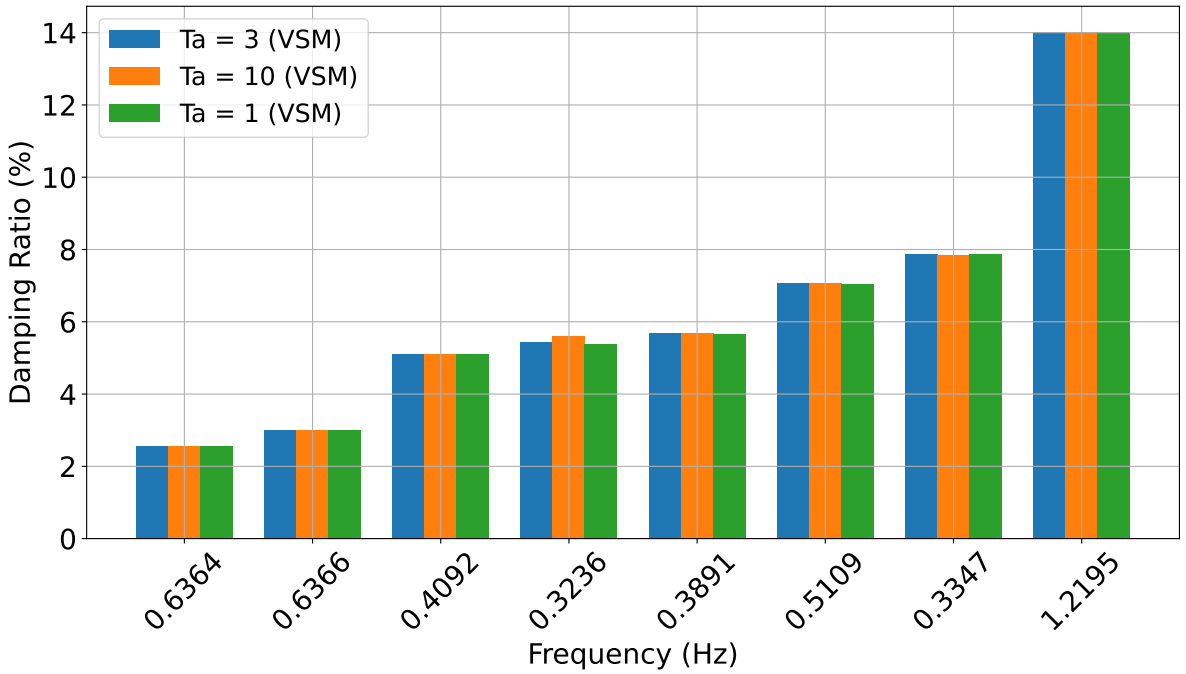


Figure 7.40: Damping ratio and corresponding frequency for different T_a values for the VSM controller. The VSM frequency control loop is highlighted in pink.

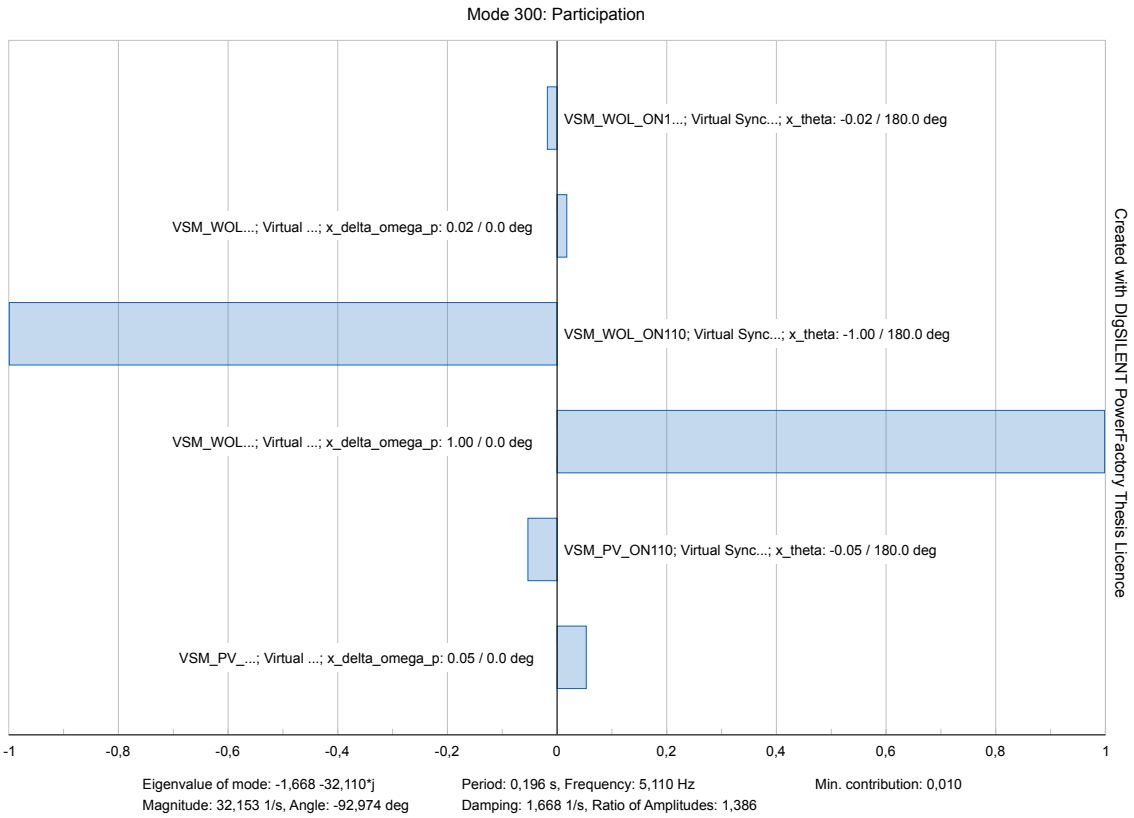


Figure 7.41: Overview of participation factors for mode 300 with $D_p = 10$ for the VSM controller.

Synchroverter Controller 1

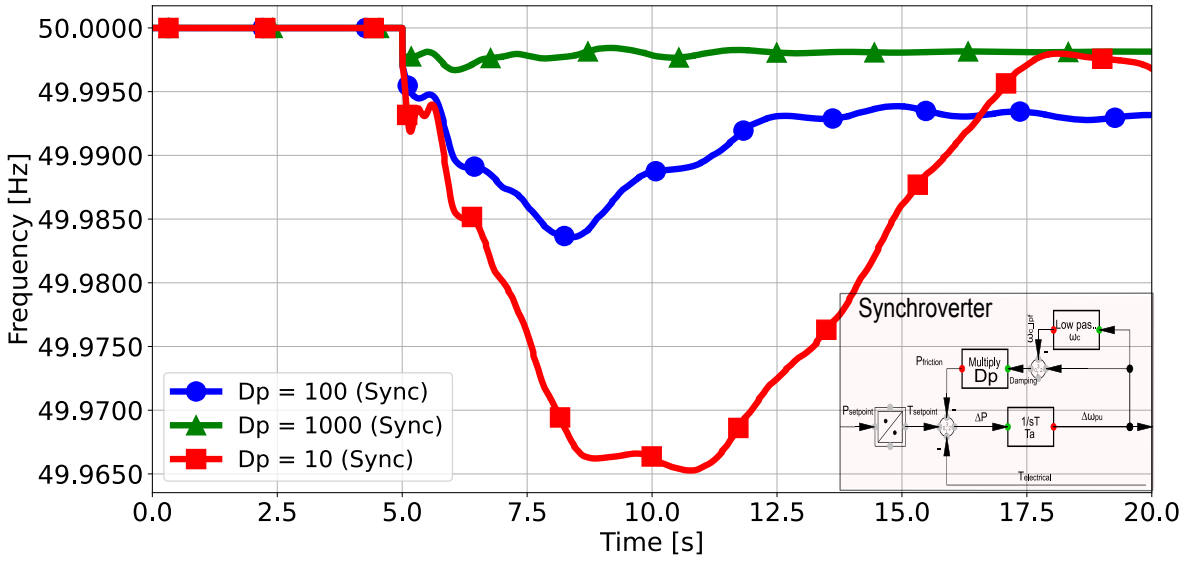


Figure 7.42: Frequency response for different D_p values for the Synchroverter controller following a load event occurring at 5 seconds. The Synchroverter frequency control loop is highlighted in pink.

Table 7.13: ROCOF, Frequency deviation, and Frequency Nadir for different D_p values for the Synchroverter controller.

D_p (Synchroverter)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
D_p = 100	49.9836	6.2	4.8	16.4
D_p = 1000	49.9967	7.4	2.2	3.3
D_p = 10	49.9653	16.9	7.3	34.7

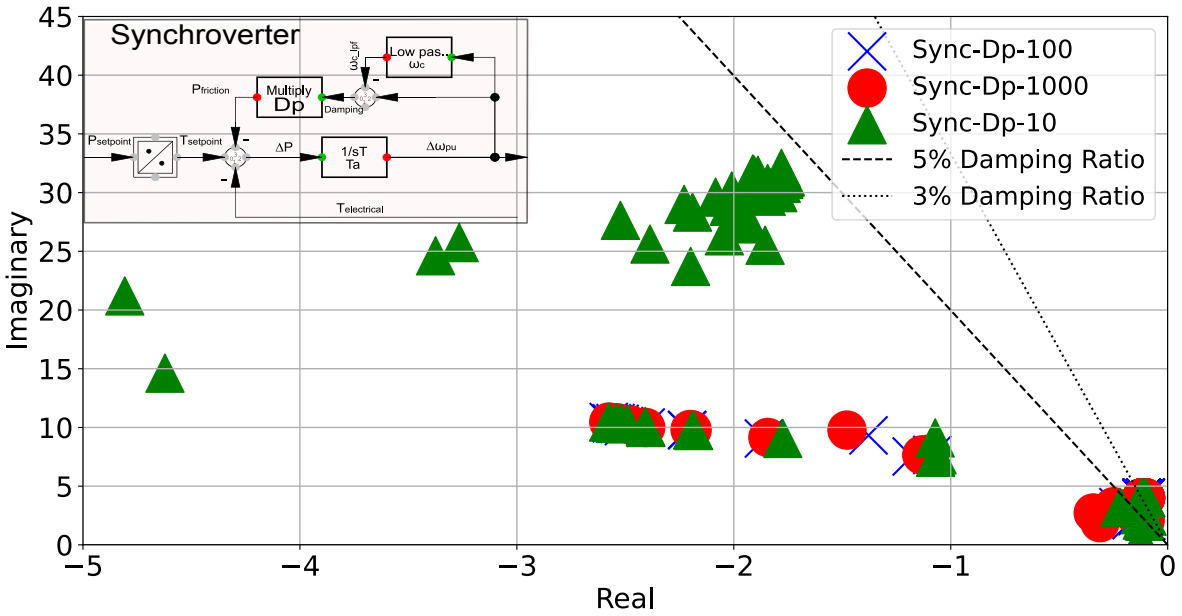


Figure 7.43: Eigenvalues for different D_p values for the Synchroverter controller. The Synchroverter frequency control loop is highlighted in pink.

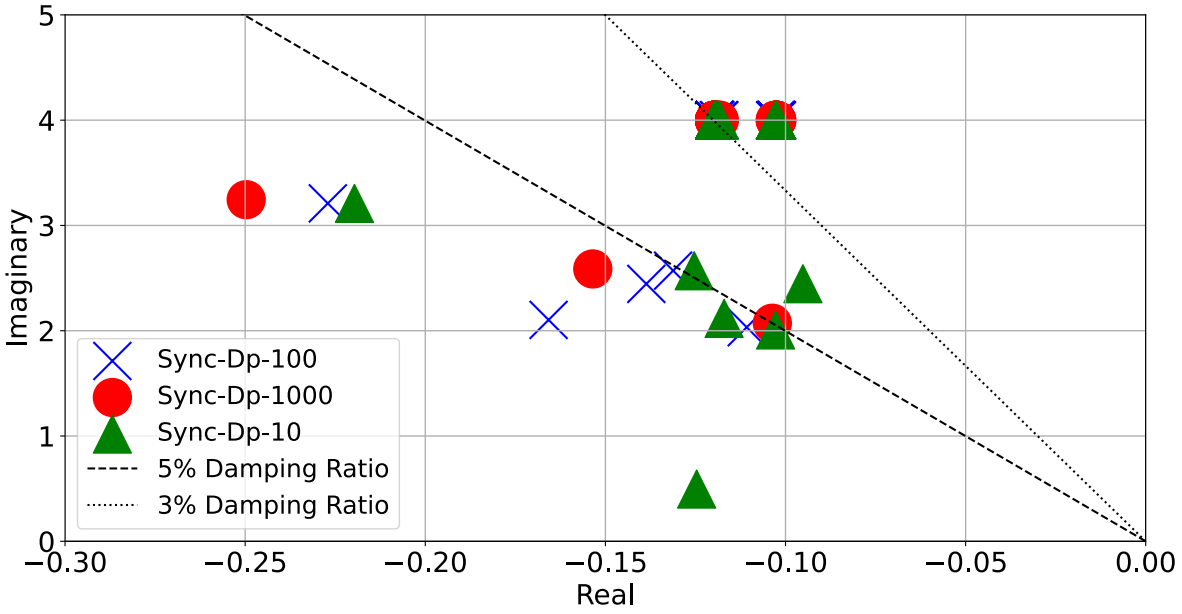


Figure 7.44: Eigenvalues for different D_p values for the Synchroverter controller, zoomed in on the most critical modes.

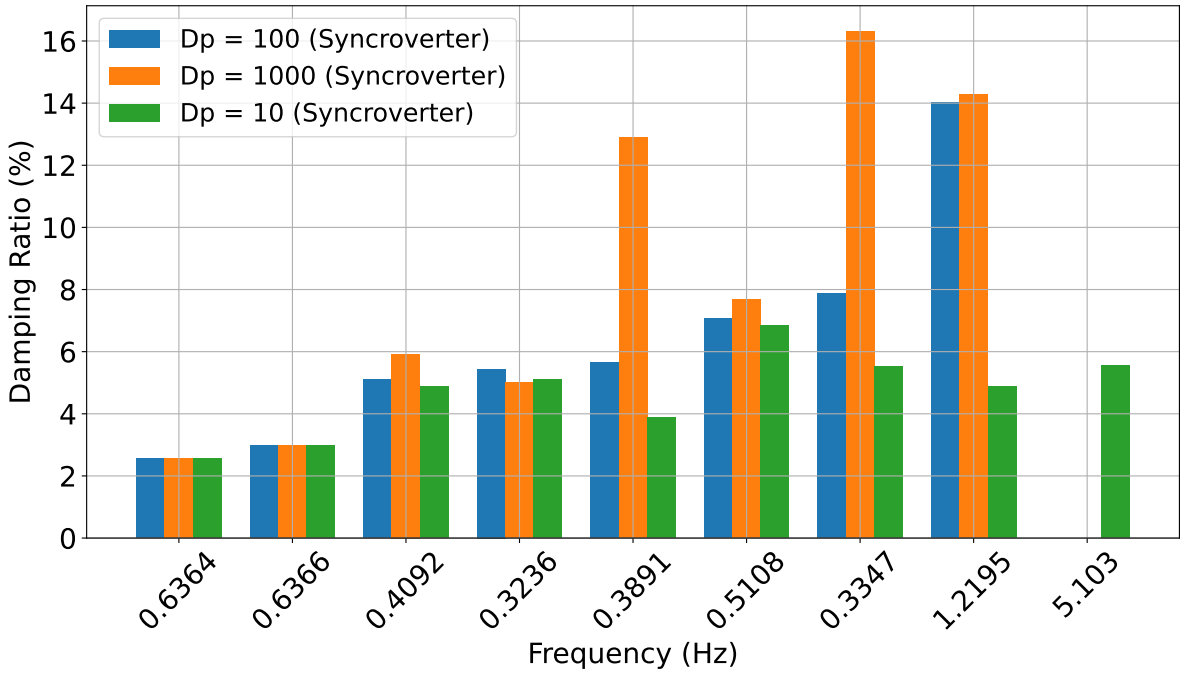


Figure 7.45: Damping ratio and corresponding frequency for different D_p values for the Synchroverter controller.

Droop Controller 1

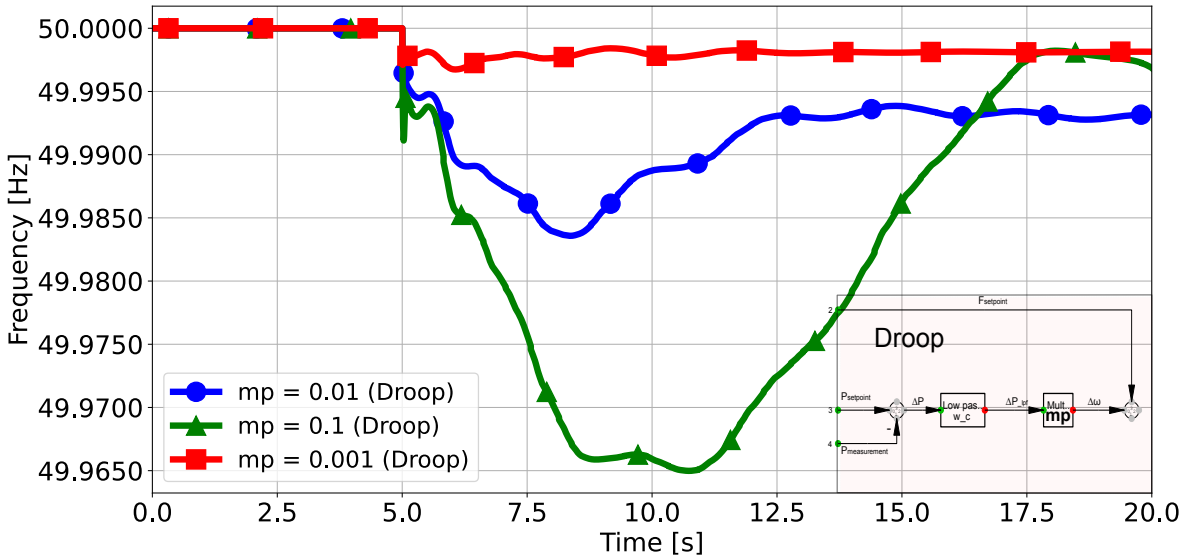


Figure 7.46: Frequency response for different mp values for the Droop controller following a load event occurring at 5 seconds. The Droop frequency control loop is highlighted in pink.

Table 7.14: ROCOF, Frequency deviation, and Frequency Nadir for different mp values for the Droop controller.

mp (Droop)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
$mp = 0.01$	49.9836	5.8	4.7	16.4
$mp = 0.1$	49.965	25.7	7.9	35
$mp = 0.001$	49.9967	7.3	2.1	3.3

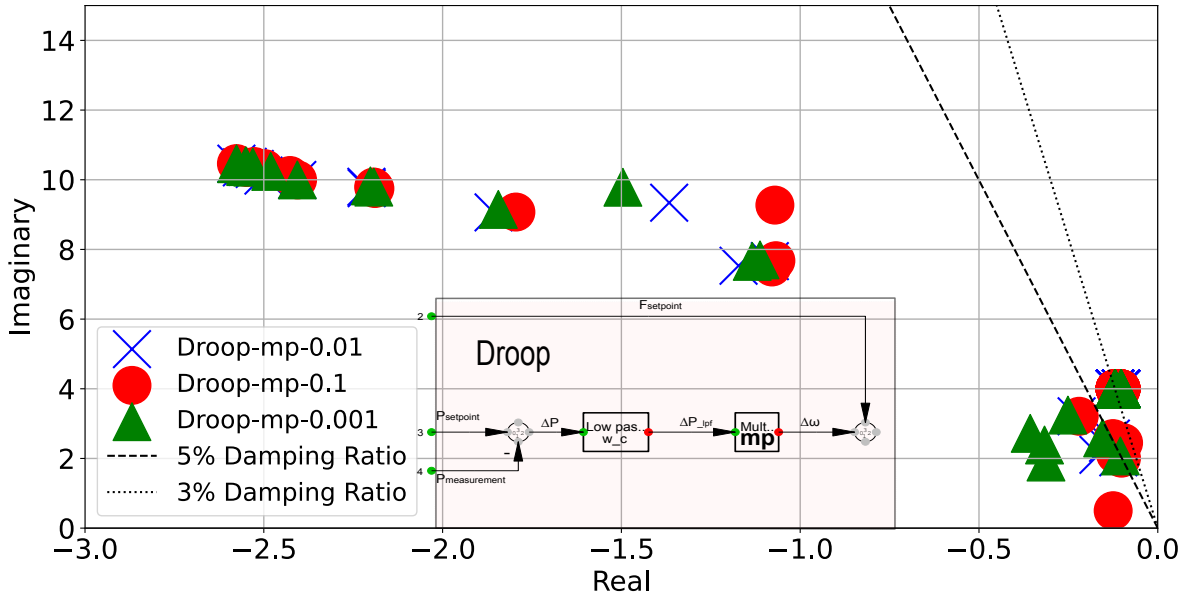


Figure 7.47: Eigenvalues for different mp values for the Droop controller. The Droop frequency control loop is highlighted in pink.

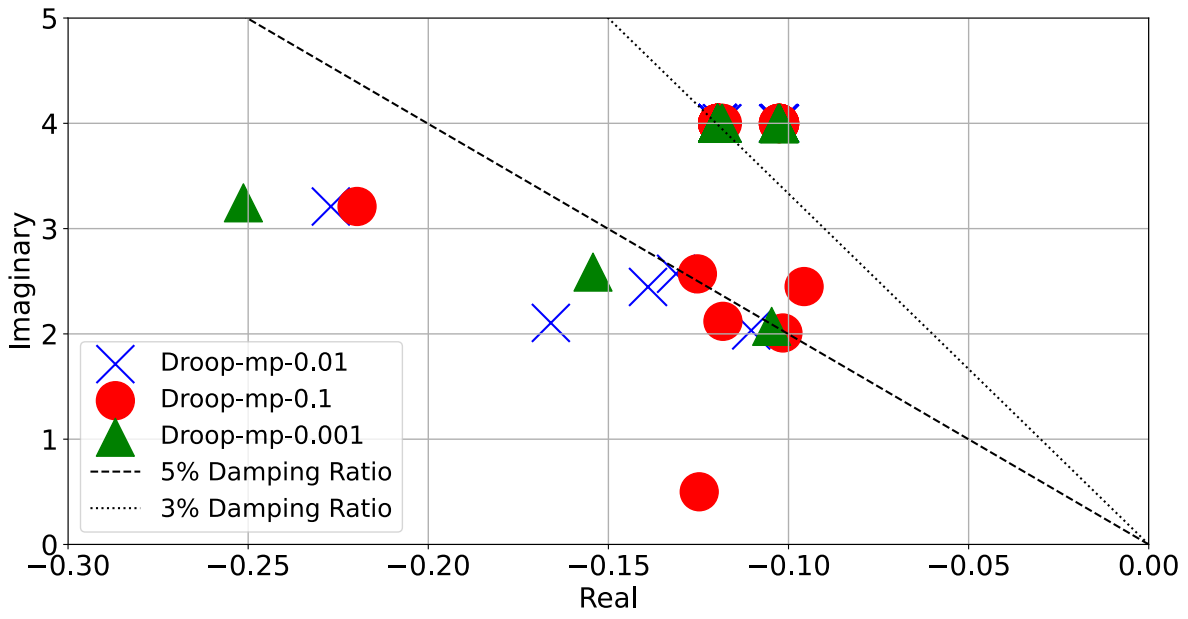


Figure 7.48: Eigenvalues for different mp values for the Droop controller, zoomed in on the most critical modes.

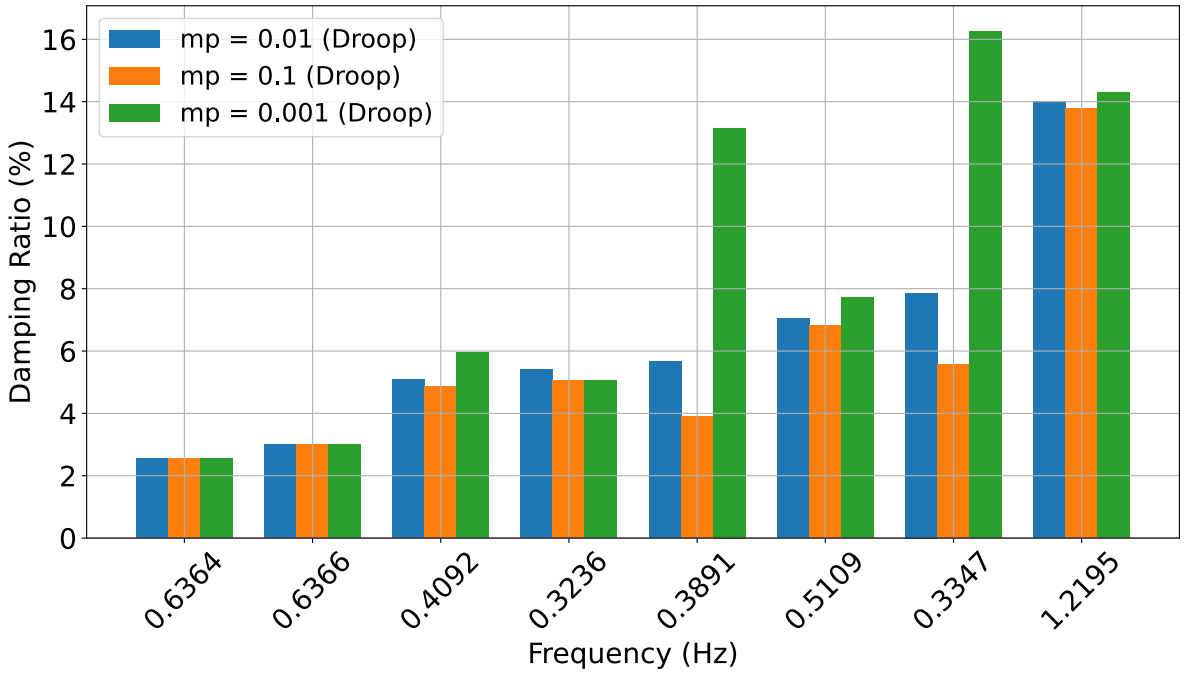


Figure 7.49: Damping ratio and corresponding frequency for different mp values for the Droop controller.

7.3.4. Discussion

7.3.4.1. Comparative Analysis of Grid-Forming Controllers

The first part of the study focused on comparing the three controllers, VSM, Droop, and Synchroverter, across three events: a load increase, an outage, and a short circuit. The results for scenarios with a 50 % grid-forming penetration are shown in Figure 7.19 to Figure 7.22, with ROCOF, frequency nadir, and maximum frequency deviation provided in Table 7.7. Similar analyses for grid-forming penetration levels of 25 % and 85 % are shown in Figure B.12 to Figure B.14 and Figure B.15 to Figure B.17, with corresponding metrics in Table B.1 and Table B.2.

All three controllers exhibit nearly identical frequency responses for both the load and outage events. This is evidenced by overlapping curves and minimal differences in frequency nadir, ROCOF, and maximum deviation across all penetration levels. These findings align with [76], which noted minor differences in frequency responses for load increases, suggesting that controller differences' impact is minimized in larger systems. The study found slightly lower allowed frequency deviations for the Droop controller and Synchroverter compared to the VSM, but these differences were negligible. Furthermore, the level of grid-forming penetration does not significantly affect the differentiation of frequency responses comparing the different controllers. This indicates that in large systems, the choice of controller type has minimal impact on the frequency response for load increases and outage events for all three analyzed grid-forming penetration levels.

The analysis extends to the system-wide performance, as shown in Figure B.18, indicating consistent performance across all machines at a 50 % penetration level. However, local phenomena and influences on other variables cannot be ruled out, necessitating further analysis of specific key elements to fully understand the system impact of each respective controller.

In contrast, the frequency response to a short-circuit event, illustrated in Figure 7.21 and Figure 7.22, shows notable differences. All three controllers produce similar oscillatory responses with varying magnitudes of deviation. The VSM controller exhibits the lowest allowed frequency deviation during the first swing, followed by the Droop controller, with the Synchroverter having the highest deviation. When the fault is cleared, the second swing shows smaller differences, with the Synchroverter having the highest allowed deviation. After these initial swings, the VSM and Synchroverter responses align more closely, while the Droop controller shows larger deviations with regard to amplitude and phase shift in

the frequency response. The Synchroverter consistently has the highest frequency deviation and ROCOF.

A similar pattern is observed for 25 % and 85 % grid-forming penetration, as shown in Figure B.14 and Figure B.17. For the 85 % penetration, the Synchroverter initially failed to provide enough short-circuit power for simulation convergence, requiring an increase in the maximum short-circuit current from 1.2 pu to 1.3 pu. Even with this adjustment, the Synchroverter exhibited the highest frequency deviation allowed. This indicates a lower overall short-circuit capability compared to the VSM and Droop controllers.

Overall, the analysis shows minimal differences among controller types concerning large-scale systems' outages and load increase events. However, differences are significant in their short-circuit capabilities. The VSM and Droop controllers perform better in short-circuit scenarios, with the Synchroverter showing the worst performance. The VSM and Droop controllers demonstrate similar and superior frequency responses for short circuits, with the lowest allowed frequency deviations for all penetration levels. Conversely, the Synchroverter shows the highest deviations and the poorest performance, especially at the 85 % level, which required increasing the short-circuit current limit for simulation. This highlights the Synchroverter's lower short-circuit capability compared to the VSM and Droop controllers.

In conclusion, while the frequency responses of all three controllers are similar, with similar oscillations and allowed deviations, differences in short-circuit performance warrant a thorough analysis to guide controller selection. The system exhibits a quick response to disturbances, akin to the response in Case Study A. However, the evident differences, particularly in short-circuit scenarios, necessitate further research to optimize controller selection for varying system configurations and scenarios.

Next, the load response was analyzed using an eigenvalue analysis to understand the impact of different controllers on the system's small-signal stability and oscillatory performance. The eigenvalues for the three controllers, considering all modes with a damping ratio below 50 %, at a 50 % grid-forming penetration level, are presented in Figure 7.23, with the most critical modes highlighted in Figure 7.24. These eigenvalues indicate that the overall oscillatory performance is nearly identical for all three controllers. Minor deviations are observed in some modes, but the majority remain consistent across scenarios, reflecting similar frequency responses for all three controllers.

Detailed information on the eigenvalues, their respective frequencies, and damping ratios can be found in Table 7.8 and Figure 7.25. It is important to note that mode indexes and minor changes in eigenvalues and frequencies could change with each controller simulation. However, for simplicity and clear comparison, VSM values were used as the benchmark, and modes were found in other simulations by associated frequencies. The analysis reveals that all modes exhibit almost identical damping ratios, with deviations smaller than 0.1 %, indicating negligible differences.

The modes consist of inter-area and intra-area modes, similar to those identified in Case Study A. Modes 122, 86, 124, 192, and 129 represent inter-area modes, while modes 62 and 121 are intra-area modes, akin to mode 254. These are identified according to the definitions given in Table 2.2. Inter-area modes are primarily dominated by AC interconnections, and intra-area modes by grid-following converters, with an additional intra-area mode dominated by hydrogen and nuclear power plants within the system. Participation factors for these modes are shown in Figure B.3 to Figure B.8. These modes are similar to those found in case study A, and a more detailed analysis is found in subsubsection 7.2.4.2.

Similar patterns are observed at a 25 % grid-penetration level, as seen in Figure B.19 and Figure B.20, along with Table B.3. Small differences in damping ratios, mostly within very small ranges, represent minor system impact. At the 85 % level, while the overall damping remains similar across different modes, slightly larger differences are evident. These differences are still relatively small and have minimal impact on the system's overall damping ratio. The overall differences balance out as each controller has different modes with the best damping ratio. For instance, VSM shows the best damping ratio for mode 192 with a frequency of 0.5109 Hz but also exhibits lower values in other modes. Similar patterns are seen for Droop and Synchroverter controllers.

Comparing these findings with those in [76], which found larger differences in oscillatory responses

across controllers, this analysis shows that in large-scale systems, controller differences are once more minimized, maintaining overall system stability across all grid-forming penetration levels for the three individual controllers.

The different converters exhibit varying impacts on system stability under different perturbations. Frequency responses were nearly identical for load and outage events across all three controller types. However, differences were evident for short-circuit perturbations, with the Synchroverter showing the highest frequency deviation and ROCOF values. The Droop performed the best, followed by the VSM controller. ROCOF values were similar across controllers, with the Synchroverter slightly underperforming.

The eigenvalue analysis showed almost identical performance across different controllers, with minor deviations in damping ratios not significantly impacting overall stability. Thus, while theoretical differences exist, the practical impact of these controllers on system stability is minimal for load increases and outage events. However, the Synchroverter's performance is inferior for short-circuits, especially at higher penetration levels where dynamic simulation issues arose, requiring an increased maximum short-circuit current.

These findings emphasize the need for thorough analysis when selecting controllers for future systems. More research is necessary to fully understand the advantages and drawbacks of each controller type, ensuring optimal system stability and performance.

7.3.4.2. Comparison of Grid-Forming Penetration Levels

Next, the impact of different levels of grid-forming penetration is analyzed and discussed. Previously, the three controllers (VSM, Droop, and Synchroverter) were compared for various simulations with different penetration levels. Here, the penetration levels themselves are the focus of the analysis.

The results for the VSM are shown in Figure 7.26 to Figure 7.29, with ROCOF, frequency nadir, and maximum deviation presented in Table 7.9. These figures provide frequency responses for outage, load increase, and short-circuit events, comparing the three penetration levels for the VSM controller.

For the outage event, the level of grid-forming penetration significantly impacts the allowed frequency deviation and system response. Higher grid-forming penetration results in a better frequency response with a lower frequency deviation. The most significant difference is observed when the frequency starts increasing after reaching its lowest value. Although all signals have a similar oscillatory response, the magnitude of frequency differences is much larger, favoring higher grid-forming levels. After this increase, the frequency responses converge around the 15-second mark. The 85 % and 50 % grid-forming penetration levels exhibit a more settled response, while the 25 % level shows another deviation and a decrease towards the end of the 20-second simulation window. Small oscillations are also evident for the 85 % and 50 % levels but with much lower magnitudes.

The ROCOF, frequency nadir, and maximum deviation for the outage event show differences; however, these are minimal. Frequency deviations are in the low mHz ranges, with the 25 % penetration level having a maximum deviation of 3.6 mHz and the 85 % level at 1.9 mHz. Thus, while higher grid-forming penetration levels improve the response and settling of outage events, the overall system response is excellent for all scenarios.

A similar pattern is observed for the load response. Higher grid-forming penetration levels result in better frequency response with lower frequency deviations. The same oscillatory pattern is seen with the outage event, with minor deviations in the first two swings and larger ones in the secondary response. The responses converge around the 15-second mark, with the 85 % and 50 % penetration levels settling, while the 25 % level shows another deviation. The load response analysis, including frequency nadir, ROCOF, and maximum allowed deviation, shows that higher penetration levels perform better. The difference between the load event penetration levels is larger than the outage event, with a deviation range between 11.3 and 23 mHz, once more favoring the higher penetration levels. The load event has the largest impact on frequency stability, with higher grid-forming percentages handling disturbances

more effectively.

For the short-circuit event, the highest frequency deviation occurs with the lowest grid-forming penetration. While the oscillatory response is similar, larger differences are observed. For the first swing, the 50 % penetration level has the highest deviation, followed by the 25 % level. After clearing the short circuit at 5.1 seconds, the 25 % penetration level allows the highest frequency deviation and worst performance. The 85 % level has the smoothest response and best-settled state at the end of the 20-second simulation. The 50 % level performs slightly worse, with more oscillations post-short-circuit. Overall, higher grid-forming penetration levels result in better frequency performance. Here, the maximum frequency deviation ranges from 3.2 to 8.9 mHz.

The grid-forming penetration levels impact the frequency nadir, ROCOF, and maximum frequency deviation. Higher penetration levels consistently provide the best frequency response. Compared to the outage and short-circuit events, the load event shows the largest impact on these parameters, followed by the short-circuit and outage events.

The Synchroverter and Droop controllers exhibit similar responses. Results for the Synchroverter are shown in Figure B.23 to Figure B.26, and for the Droop controller in Figure B.30 to Figure B.33. ROCOF, frequency nadir, and maximum deviation are detailed in Table B.5 and Table B.7. All three grid-forming converters show similar impacts on system stability, with small deviations across penetration levels. A higher share of grid-forming penetration is beneficial for all controllers, except for the necessary adjustment of short-circuit current limits for the Synchroverter at high penetration levels.

Next, the oscillatory response of the three converters was further analyzed using an eigenvalue analysis for different levels of grid-forming penetration. The results for the VSM controller are presented in Figure 7.30 and Figure 7.31, as well as Table 7.10, with a chart comparing the damping ratios for different modes in Figure 7.32.

The analysis reveals that while the overall modes and damping ratios are relatively similar across the three penetration levels, larger differences are observed compared to the controller comparisons. The damping ratios vary based on the mode and penetration level. Modes 62, 121, 86, and 192, which include inter-area modes dominated by AC interconnections and intra-area modes dominated by grid-following wind converters, show minimal differences across the penetration levels. However, significant improvements in damping ratios are evident for modes 124 and 129 at the 85 % penetration level compared to the 50 % and 25 % levels, indicating better damping of these inter-area oscillations at higher penetration levels.

Despite the improved damping in some modes, all penetration levels exhibit well-damped inter-area oscillatory modes above the 5 % mark. Controller parameters, as seen in Case Study A, also influence these damping levels and will be further investigated. Overall, higher grid-forming penetration levels enhance the damping ratio of most oscillatory modes, but some modes remain below the required level, mainly the grid-following converter modes. This indicates that further analysis of controller parameters and locations is necessary to ensure proper damping on large-scale system implementation of grid-forming and following converters.

An additional vulnerability is observed at the highest grid-forming penetration level in mode 128, which has a frequency of 1.4215 Hz. Here, the 85 % penetration level shows a damping ratio of 5.69742 %, significantly lower than the 20.1386 % for the 50 % and 25 % levels. A participation factor analysis, shown in Figure B.37, indicates that this mode is dominated by grid-following wind turbines and associated H-mass control units. Due to only one converter evident in the participation analysis, a local mode is indicated. This local mode, identified as such due to its frequency and participation factor, suggests that higher grid-forming penetration may cause grid-following converters to struggle to keep up with the fast-acting power-balancing operations. While not confirmed, this hypothesis highlights the need for a detailed examination of this phenomenon.

Similar results are found for the Droop and Synchroverter controllers, as shown in Figure B.34, Figure B.35, Figure B.36, and Table B.8 for the Droop controller, and Figure B.27, Figure B.28, Figure B.29,

and Table B.6 for the Synchroverter. This indicates that the observed phenomena are not specific to any one controller type.

A hypothesis is that higher grid-forming penetration increases fast-acting power balancing, making it challenging for grid-following converters to keep up due to their dependence on an external signal through the PLL. This time delay can cause oscillations between the stator and rotor of wind turbines. The converter found in the participation analysis, the onshore wind converter at ENS380, was removed from the system, and the simulation was run again. Here, the mode was removed from the system and did not exist elsewhere. This indicates that the mode might be dependent upon the topology and placement of the grid-following converters. However, the findings still indicate that these modes can occur in highly penetrated grid-forming systems and should be grounds for further analysis of this phenomenon.

Overall, while higher grid-forming penetration levels improve the damping of various modes, challenges remain, particularly with local modes in grid-following converters. This finding is consistent with stability issues associated with high PEI generation levels and suggests that high grid-forming penetration can cause local mode oscillations in grid-following converters, highlighting the need for careful planning and analysis.

A thorough analysis should be conducted for projects regarding grid-forming penetration levels. Although the damping ratio for the local mode was above 5 %, indicating adequate damping, the findings suggest the need for scenario analysis to identify potential oscillatory modes early.

The comparative analysis of grid-forming penetration levels highlights mixed results regarding system damping and stability. Higher grid-forming penetration levels result in better frequency responses for all events and parameters across all controller types, including frequency nadir, ROCOF, and frequency deviation. However, the Synchroverter showed the lowest short-circuit capability, requiring an increase in maximum output current for simulations at the 85 % penetration level.

The oscillatory analysis showed mixed results. Increased penetration levels improved damping for inter-area modes, but the 85 % level showed a drastic reduction in damping for one local mode, indicating that higher penetration levels can impact the damping of other converter types. This finding underscores the importance of strategic planning and further research into grid-forming converter implementation.

Consistent with the findings in [196], this analysis shows that while higher grid-forming penetration levels improve overall stability, high penetration can also lead to issues. Here, with local modes in grid-following converters, emphasizing the need for detailed stability analysis in large-scale systems.

7.3.4.3. Controller Parameter Sensitivity Analysis

Finally, the impact of different controller parameters will be discussed. This analysis was conducted through frequency and eigenvalue analysis, similar to the two previous analyses, but focused on comparing different values of the most important controller parameters and their impact on stability. As grid-forming converters become more prominent in future power systems, it is crucial to fully understand how different controller parameters affect system stability in large-scale systems. This understanding is essential for regulatory assessments concerning grid-forming controller settings and the importance of tuning these parameters. In this analysis, a load increase was used as the disturbance to evaluate both the frequency response and eigenvalues in the system.

The parameters of interest, identified in section 2.2, vary for the three different controllers. For the VSM, the primary parameters of interest were the acceleration time constant (T_a) and the damping coefficient (D_p). The active and reactive power coefficients (mp and mq) were identified for the Droop controller. For the Synchroverter, four parameters were considered: the acceleration time constant (T_a), damping coefficient (D_p), voltage control gain (K_q), and reactive power droop coefficient (D_q). The controller parameters for the different simulations are detailed in Table 7.6, where the parameter changes for the simulations are highlighted in parentheses. The load response was used to compare

frequency and eigenvalue simulations. For a more detailed section regarding these parameters, refer to section 2.2. All these simulations were done with a grid-forming penetration level of 50 %; future work should analyze the impact of different levels of penetration as well.

VSM Controller Analyzing the results from the VSM controller, the impact of the two controller parameter adjustments shows significant deviations in system stability. First, considering the impact of the D_p parameter on the system's frequency response, the results are shown in Figure 7.33. Here, the 10, 100, and 1000 values substantially affect the overall system response. The highest D_p levels show the best frequency response, characterized by extremely fast power balancing. The highest level has the lowest initial frequency deviation during the first swing and does not allow large oscillations afterward, demonstrating good damping within the 20-second time window. In contrast, the lowest D_p value shows the largest initial frequency deviation. For the two lower D_p values, an additional oscillatory response occurs after the initial swing, with the lowest frequencies evident in the 10 D_p value around the 11-second mark, indicating a slower and overall worse frequency response.

The 100 D_p value settles around the 15-second mark, whereas the 10 D_p value does not properly settle until the end of the simulation window. Overall, the two lower D_p values exhibit similar responses in terms of oscillatory behavior but with significant differences in frequency deviations, with the lowest D_p values showing the worst values. Compared to the highest D_p values, the system responds extremely quickly, with fast damping, allowing only one oscillatory response before settling. This is also confirmed by the ROCOF, frequency deviation, and frequency nadir for the different values, as seen in Table 7.11. The highest D_p values result in the lowest ROCOF values and the lowest overall frequency deviation compared to the other values, with significant differences. These differences range from a maximum deviation of 3.3 mHz for the 1000 D_p value, while the 10 D_p value shows a maximum deviation of 34.8 mHz, over a 10-fold increase.

Comparing these results with the eigenvalue analysis depicted in Figure 7.34 and Figure 7.35, similar conclusions can be drawn. The three levels' lowest damped modes and their respective damping ratios are shown in Figure 7.36. The highest values of D_p result in better overall damping of the most critical modes. This improvement is evident for most inter-area oscillatory modes, with some modes experiencing significantly increased damping ratios. For instance, the damping ratio improvements can be up to 3.3 times better between the highest and lowest D_p levels. All inter-area modes have a damping ratio of at least 5 % for the D_p values of 1000 and 100, while the value of 10 exhibited a damping ratio below 5 % for modes 122 and 124, indicating overall lower damping of critical inter-area modes. However, all of these inter-area modes have a damping ratio above the 3 % margin, indicating adequate damping for all parameter values for these modes.

A new mode with a frequency of 5.1025 Hz and a damping ratio of around 5.2 % was also discovered. A participation factor analysis was conducted, and the results from mode 300, representing a similar mode, can be seen in Figure 7.41. The analysis show that the $x_{delta_{omega}}$ and x_{theta} parameters of the VSM controller create a new and low-damped mode, impacting the overall slow-interaction converter-driven stability of the system. Similar slow-interaction modes have been observed in literature for VSC-HVDC connected systems, primarily affected by tuning the outer-loop parameters [79]. This highlights that similar results are found for high grid-forming penetration levels with a low damping parameter D_p, emphasizing that controller parameters significantly impact system stability. This mode does have a damping ratio above 5 %; however, due to the difference with respect to the other parameter values, the suddenly lowered damping is of interest and highlights that converter-driven stability issues can arise for grid-forming integrated systems.

The different levels of D_p show no significant impact on the intra-area and local modes associated with the grid-following converters.

The analysis of the D_p parameter underscores the importance of parameter tuning in future systems with high grid-forming penetration levels, as these parameters significantly affect system performance.

The analysis shows that damping levels too low, results in low inter-area damping that approaches stability margins. Conversely, high D_p levels ensured that all inter-area modes had a damping ratio above 5 %, reinforcing the grid-forming potential regarding inter-area oscillations and general damping studies, as discussed in Case Study A. The lowest damping settings also result in local modes associated with the grid-forming converters, exhibiting fast oscillations and low damping. Although above the 5 % mark, this still indicates generalized instability across all grid-forming converters, underscoring that controller tuning impacts both the grid-forming converter itself and large-scale system stability.

Analyzing the results for the T_a parameter, seen in Figure 7.37, the frequency analysis shows an almost identical response for all three parameter levels. Small differentiations are observed, with higher T_a values resulting in a slightly slower response and higher frequency deviation in the first and second swings. However, these differences are minimal, and the overall impact is low. This is also evident in Table 7.12, highlighting the frequency nadir, ROCOF, and maximum frequency deviation for all three controller parameters. The overall levels are nearly identical, with slight improvements evident for the lower and increased T_a values.

A similar pattern is found in the eigenvalue analysis, with the eigenvalues shown in Figure 7.38 and Figure 7.39, and the damping ratios of the most critical modes in Figure 7.40. There is a clear similarity among the three controller parameter values for all critical modes, with small deviations. The highest T_a values show slightly improved damping for inter-area modes, supporting the frequency response findings. Slight differences are also evident in higher-speed modes, with higher T_a values indicating slightly better damping. However, these modes already have high damping ratios that do not significantly impact system stability. Nonetheless, this could differ in systems with poor damping or different topologies, warranting further consideration of T_a in stability analyses for future projects using the VSM controller.

Overall, the T_a parameter analysis results show minimal differential impacts across different parameter values. Small differences in frequency response and critical system modes are observed but within sub-1 % ranges. Differences in higher-speed oscillations favor higher T_a values, though these are highly damped and do not raise immediate concerns. However, this could vary for cases with poor damping or different topologies, suggesting further research on these controller parameters is warranted, as they can impact system stability.

Synchroverter Controller Four parameters were examined for the Synchroverter: D_p, T_a, K_q, and D_q. First, analyzing the results from the D_p parameter, the frequency response shown in Figure 7.42 is very similar to that of the VSM controller. Small deviations between the two responses are evident when comparing the frequency nadir, ROCOF, and maximum deviation as seen in Table 7.13. However, these differences are minimal, and the overall impact of D_p on the Synchroverter is similar to its impact on the VSM controller. As before, the highest damping leads to the best frequency response and quicker settling of the signal, while the two lower levels exhibit more oscillatory behavior, slower signal settling, and worse maximum frequency deviations and ROCOF values. The eigenvalue analysis in Figure 7.43 and Figure 7.44, with the most critical mode differences shown in Figure 7.45, supports these findings.

Similar to the VSM controller, a new mode was discovered for the lowest rated D_p parameter, with a frequency of 5.103 Hz. This mode has a damping ratio above 5 %, still indicating stability, but due to its emergence, it warrants close observation and further research to fully understand its potential impact on stability in future systems. This mode was not evident with a damping ratio below 50 % for the other D_p levels. As before, the inter-area oscillations are significantly better damped for higher D_p levels, with improvements up to 3.3 times for individual inter-area modes. The D_p parameter findings are almost identical to those of the VSM controller, highlighting the similarities in controller structures and the impact of parameter changes on system performance.

The similarity between the Synchroverter and the VSM controller structures is also evident when analyzing the T_a parameter. The frequency response and calculations show similar values to the VSM, as seen in Figure B.38 and Table B.9. There are slight differences in the response between different T_a levels. The highest parameter value shows a slightly delayed response in the first and second swings

but with minimal impact on overall system stability. The eigenvalue analysis, shown in Figure B.39 and Figure B.40, indicates that the most critical modes are nearly identical across T_a levels, with a slight advantage for the highest value in inter-area oscillations, similar to the VSM. Slightly larger differences are found in modes with higher oscillatory speed, indicated by the imaginary axis, where higher T_a values improve damping. These modes are well-damped and do not pose immediate threats to system stability but could be impacted by factors like system topology and strength. This change is noteworthy as it could negatively impact stability under certain operational characteristics. Overall, T_a is not the most critical parameter for the Synchroverter, but it does cause some changes and could be important in future system tuning and analysis.

Next, the results for the frequency and eigenvalue analysis of K_q are shown in Figure B.41 to Figure B.43, and Table B.10. The findings for D_q are in Figure B.44 to Figure B.46 and Table B.11. No significant differences are found in these analyses, indicating that K_q and D_q do not impact the system's oscillatory performance with changes in their values. These parameters primarily handle reactive power control, which is less critical in these studies focused on active power control. Similar conclusions were drawn from a smaller test system analysis in [76]. The results suggest that K_q and D_q do not affect system oscillatory performance, implying that further studies can explore their potential to improve system performance without compromising oscillatory stability.

The Synchroverter results mirror the performance patterns of the VSM. The two main parameters affecting system stability, D_p and T_a, have a similar impact on both controllers. The additional parameters in the Synchroverter, K_q and D_q, show no impact on the system's oscillatory performance and should be further examined for their potential to enhance other system parameters as they can be adjusted without affecting oscillatory stability.

Droop Controller Moving on to the Droop controller, two parameters of interest were analyzed: mp and mq. These are the primary variables concerning the Droop controller's active and reactive power droop control functions. This is an inverse droop function, meaning lower values result in a larger power increase.

A similar analysis to the other controllers was performed, and the results for the frequency response of the different mp values can be seen in Figure 7.46, with the frequency nadir, ROCOF, and maximum frequency deviation provided in Table 7.14. It is evident that the lowest values of mp result in the best system performance in terms of frequency response, impacting the system similarly to the high D_p parameter values of the VSM and Synchroverter. The lowest level results in a system that is dampened after the initial swing, with only minor oscillations during the remaining 20-second simulation window. In contrast, the higher levels allow more oscillations after the load increase, with the highest value not fully settling until the end of the 20-second window and the medium value settling around 15 seconds. The performance metrics—frequency nadir, ROCOF, and maximum frequency deviation—are significantly worse for higher mp values, with maximum frequency deviations ranging from 3.3 mHz to 35 mHz, a tenfold difference between the best and worst performance levels. Overall, it is evident that reducing the mp values increases the performance of the frequency response of the system

Comparing this with the eigenvalues, seen in Figure 7.47 and 7.48, with the most critical modes and their respective damping ratios highlighted in Figure 7.49, shows that the lowest mp values once again exhibit the best performance, particularly in inter-area oscillatory modes. All inter-area modes have a damping ratio above 3 %, but the mp value of 0.1 shows two modes with damping below 5 %. The differences in these modes range up to a 3.37-fold damping ratio improvement from the worst to the best parameter values, similar to findings for the D_p parameter for the VSM and Synchroverter. The higher mp levels show slightly lower damping in higher oscillatory speed modes, but these modes are already well-damped and do not pose a risk to system stability. The mp parameter does not significantly impact intra-area and local modes associated with the grid-following converters. Notably, unlike the other VSM and Synchroverter controllers, none of the parameter values resulted in a low damping grid-forming dominant mode in the 5 Hz frequency range, suggesting that the Droop controller may be less prone to negative impacts from parameter tuning compared to the VSM and Synchroverter.

Overall, the response of the mp parameter in the Droop controller is very similar to the D_p variable in the VSM and Synchroverter controllers, with similar impacts on frequency oscillatory response and system stability. Higher values of D_p correspond to higher active power levels for the same signal, compared to the inverse nature of mp. The Droop controller shows similar damping performance to both the VSM and Synchroverter and exhibits better performance by not creating a poorly damped grid-forming controller interaction mode associated with the lowest D_p levels in the VSM and Synchroverter, indicating a higher margin of error in the Droop controller with regards to system stability limits concerning the primary active power parameter.

Next, the findings regarding the frequency response and oscillatory performance of the mq parameter adjustment are discussed. The results are shown in Figure B.47 to Figure B.49, as well as in Table B.12. Similar to the reactive power control loop main parameters of the Synchroverter, no major differences in system performance are observed based on different mq values. This confirms that the reactive power control parameters do not significantly impact the system's frequency and oscillatory performance, highlighting that this parameter is well-separated from the active power loop. Further studies can focus on the impact of mq on other system parameters, knowing it does not affect oscillatory performance.

The controller parameter analysis demonstrates significant controllability with respect to the frequency response, associated parameters, and oscillatory analysis. Different parameter values lead to considerable variations in performance indexes across controllers. The main active-power controller loop parameters, D_p and mp, significantly impact system damping. Slight differences are evident for the acceleration time constant T_a in the VSM and Synchroverter controllers, with lower values resulting in quicker responses, but overall performance is similar. Higher T_a values perform better in higher oscillatory speed modes related to controller interactions but do not pose immediate risks to system stability. Reactive power control loop parameters K_q and D_q in the Synchroverter and mq in the Droop controller do not impact frequency and oscillatory performance, indicating they can be adjusted to enhance other system performance aspects without affecting oscillatory stability.

The comparative analysis was facilitated by using Python and Excel functions to automate tasks and significantly reduce system configuration setup time. These tools automated controller selections, grid-forming penetration level calculations, and necessary controller parameter changes, enabling comprehensive system-wide analyses. With minor adjustments, these tools can be further utilized to analyze other characteristics of grid-forming converters and components in PowerFactory, providing a highly functional model for large-scale system studies.

7.3.5. Results and Findings

This case study comprehensively analyzed the impact of different grid-forming converters based on controller comparisons, varying levels of grid-forming penetration, and the influence of grid-forming converter parameters on system stability. Each part of the study offers an in-depth examination rooted in frequency and oscillatory behavior, utilizing key metrics for comparison.

The controller parameter study found that the performance of VSM, Droop, and Synchroverter controllers on system stability is nearly identical for frequency and oscillatory behavior for a load and outage event. Negligible differences were observed in Frequency Nadir, ROCOF, maximum allowed frequency deviations, and damping ratios for the respective modes. However, significant differences emerged in the short-circuit analysis, with larger deviations evident in the frequency response for all three controllers across different levels of grid-forming penetration. The Droop controller exhibited the best performance, with the VSM controller close behind, but the Synchroverter showed the worst performance, particularly at the 85 % penetration level, where short-circuit strength was non-viable without increasing the maximum current. These findings suggest that while controller performance is similar for load response and outage events, short-circuit events could serve as a critical differentiator in selecting grid-forming controllers.

The comparison of different levels of grid-forming penetration (25 %, 50 %, and 85 %) revealed that higher penetration levels generally lead to better frequency response and oscillatory performance for all three controllers. Higher penetration levels allowed for lower maximum frequency deviations and ROCOF values across all perturbation events, with the most significant differences observed in short-circuit events. However, the Synchroverter's limitations with increased short-circuit current levels should be noted once more. In terms of oscillatory performance, higher penetration levels improved the damping ratios of inter-area modes, although a local mode in a grid-following converter exhibited low damping at the highest penetration level, highlighting potential stability issues. These findings underscore the need for further research to understand the dynamics of high grid-forming penetration levels and their impact on system stability.

The controller parameter analysis demonstrated significant variations in results based on different controllers and their respective parameters. The main active power parameters (D_p for VSM and Synchroverter and mp for Droop) were identified as the most critical for frequency and oscillatory performance. Increasing these parameters improved inter-area modes' damping ratios, with all modes achieving damping ratios above 5 % for the highest parameter values. Conversely, poor parameter values led to lower damping ratios and the emergence of low-damped modes associated with the grid-forming converters, highlighting the importance of proper parameter tuning. The acceleration time constant T_a had minimal impact, though slightly better damping in high-speed oscillatory modes was observed for higher T_a values. Reactive power control parameters (K_q and D_q for Synchroverter and mq for Droop) showed no significant impact on system stability, indicating that the active and reactive power control loops are well separated.

The case study offers valuable insights into the grid-forming converters, their respective controller selection, penetration levels, and parameter adjustments. Grid-forming converters demonstrate significant potential in enhancing system stability for large-scale applications. However, certain issues, particularly related to high penetration levels and specific controller parameters, indicate that the full potential of grid-forming converters in large-scale systems is not yet fully understood. Future research will be crucial for the successful integration of large-scale PEI generation in future power systems.

The study bridges the gap between research questions and practical implementation, providing guidelines for further analysis of the three controllers, various penetration levels, and important controller parameters. These findings will be instrumental in guiding the development and deployment of grid-forming converters in future power systems.

7.4. Case Study C

7.4.1. Introduction

As power systems increasingly integrate Power Electronic Interfaced (PEI) generation sources, the levels of inertia and kinetic energy within these systems undergo significant changes. Inertia and kinetic energy have traditionally been critical parameters for system stability and control mechanisms. As these parameters evolve, it is imperative to develop new control methods to maintain the security and stability previously ensured by conventional systems. The previous case studies highlighted the potential of grid-forming converter technologies in enhancing system stability. However, these simulations were conducted using the highest estimated inertia and kinetic energy values projected for 2040, as discussed in subsection 6.4.3 for simplicity. These projections vary considerably throughout the year and exhibit significant uncertainty, reflecting the unpredictable nature of the energy transition and associated policy and technological advancements.

This case study aims to investigate how different projections for 2040 will impact system stability, focusing on the estimated levels of inertia constants and kinetic energy. This study is crucial for informing future power system planning and policymaking, particularly regarding the potential decline in inertia and kinetic energy and its implications for system stability. The analysis will help identify realistic opportunities for future stability or highlight the need for alternative measures.

The analysis is based on the synthetic model of the future Dutch power system and its interconnections with Continental Europe. These interconnections serve as reference points for determining the system's inertia and kinetic energy.

Case A from Case Study A, representing a system predominantly powered by offshore wind, will be utilized. The perturbation used for system analysis is a load increase of 5.6 GW, consistent with previous studies. The results will be evaluated based on frequency and oscillatory analysis, similar to those conducted in Case Studies A and B.

7.4.2. Methodology

This section outlines the methodology used to simulate and analyze the impact of different inertia constants and kinetic energy values on a large-scale interconnected system. The system under scrutiny is the Dutch power system, using a synthetic model created based on scenarios for 2050. More detailed information can be found in chapter 5 and chapter 6. By analyzing this, future policy-making and directions can be informed with respect to the differentiating stability characteristics.

A flowchart highlighting the main steps taken for the system setup with respect to the case study can be seen in Figure 7.50.

The first step involved selecting the case for analysis. Different cases have distinct dispatch levels regarding load and generation. Case Study A provides a detailed analysis of these cases. For this analysis, Case A was chosen. Case A represents a scenario with high levels of offshore wind, low levels of onshore wind and solar PV, and the maximum load possible in the system.

The analysis was conducted with a grid-penetration level set at 50 %, serving as a baseline scenario. Further research can explore various scenarios involving generation demand, grid-forming controllers, and grid-forming penetration levels. The VSM controller was selected as the grid-forming converter controller.

subsection 6.4.3 identified different levels of inertia and kinetic energy estimated for Continental Europe in 2040 by ENTSO-E in [158] and [159]. Based on various estimations, the kinetic energy levels were projected to range from 1800 to 250 GVAs, with inertia constants estimated from 4.1 to 0.4 seconds. The study was conducted by simulating different ranges for both values and analyzing system metrics accordingly.

The inertia constants were directly implemented into a Python code that automatically calculated

the necessary conversions and inputs for later use in the PowerFactory model. These levels were represented using the AC interconnections as the source, reflecting the interconnections to Continental Europe, which are the dominant inertia constant characteristics in the system. Detailed structuring of these interconnections and their relevance can be found in subsection 6.4.3. The inertia constants were set by adjusting the inertia constants of the synchronous machines representing the different AC interconnections. Simulations were conducted for three levels: 4.1, 2.25, and 0.5 seconds.

Next, three different levels of kinetic energy—1800, 1025, and 250 GVAs—were selected and implemented to capture the full range of estimations. Since the interconnections are represented as synchronous machines, the kinetic energy was translated into an estimated rated power of the machine using Equation 6.1 for the three different cases. This resulted in levels of 439,025 MVA, 250,000 MVA, and 60,976 MVA. MVA values are used due to their similar output index in PowerFactory. These values were automatically calculated using the associated Python script, allowing for direct input in GVAs for future analyses and streamlining the process. Detailed information regarding the Python script utilized is found in Appendix C.

These values were then implemented in PowerFactory, and simulations were run. This included a combination of different values for inertia constants and kinetic energy levels, representing an analysis that studies best and worst-case estimations.

By following these steps, the methodology ensured a comprehensive analysis of the impact of varying inertia constants and kinetic energy values on the stability of large-scale interconnected systems.

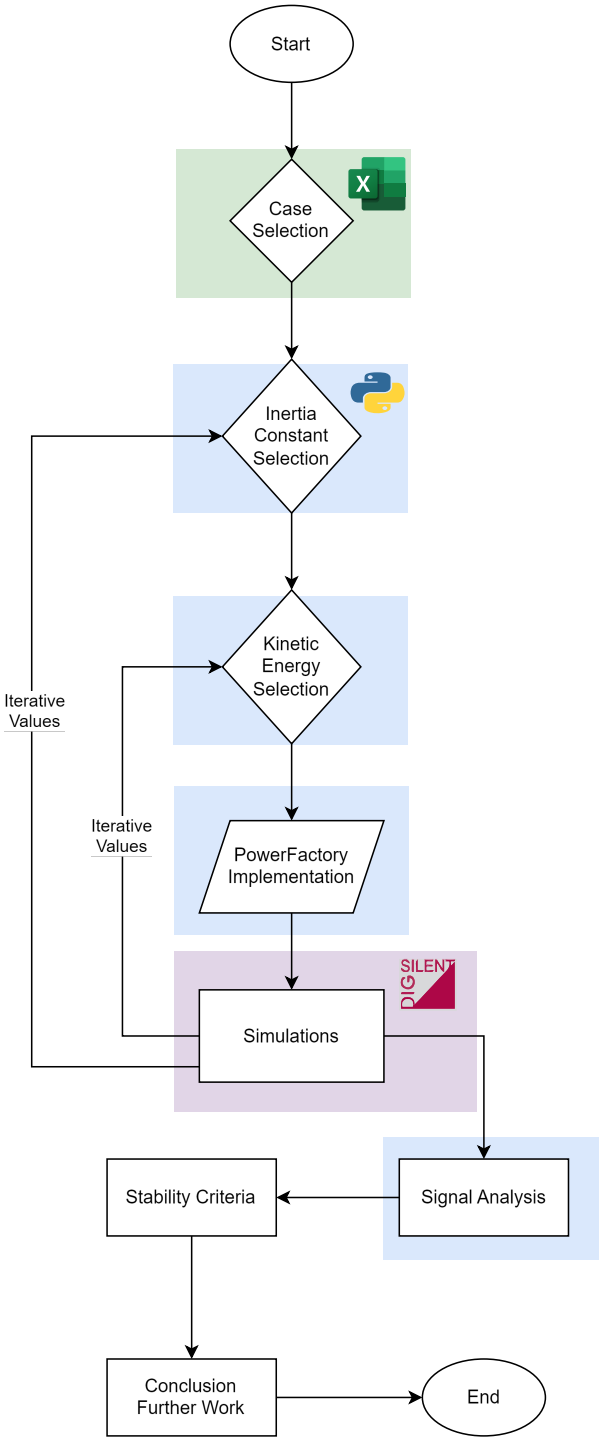


Figure 7.50: Detailed flowchart of the system setup for assessing the stability of the Dutch power system in 2050 under various inertia constants and kinetic energy values. The pink color indicates operation performed by PowerFactory, green by Excel, and blue by Python.

After this, a flowchart of the simulations and analysis can be seen in Figure 7.51. As discussed, Case A was selected as the simulation's base case. Two different simulations were conducted, comparing different results of the inertia constant and the kinetic energy. The levels used and analyzed can be seen in Table 7.15. All the different values and parameters were simulated for the other corresponding parameter values, resulting in a total of 9 main simulations with varying kinetic energy and inertia constant values. All simulations were conducted with a load increase of 5.6 GW, similar to previous studies. These simulations were then analyzed using a frequency response and eigenvalue analysis

for all the different cases. The overall structure here is similar to previous case studies, as the model and associated Python tools allow for a streamlined analysis of important system characteristics. More information about the model can be found in Appendix C.

Table 7.15: Simulation Values for Inertia Constant and Kinetic Energy

Simulation Type	Levels	Fixed Parameter
Inertia Constant [s]	4.1, 2.25, 0.5	Kinetic Energy [GVA]
Kinetic Energy [GVA]	1800, 1025, 250	Inertia Constant [s]

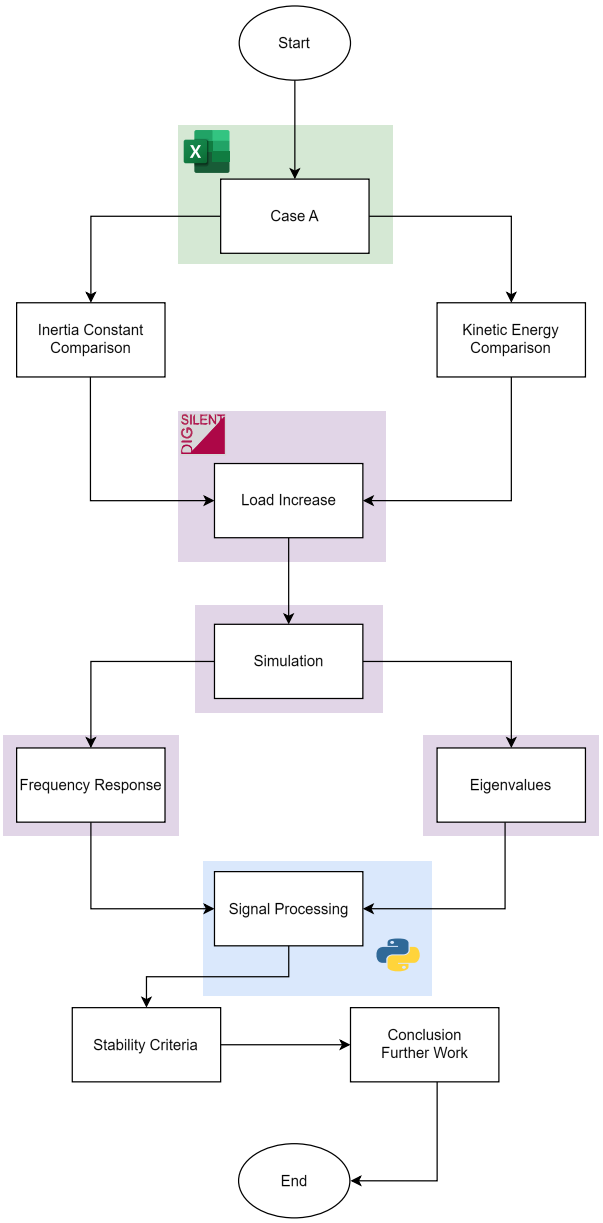


Figure 7.51: Detailed flowchart of the simulation process used in Case Study C. The pink color indicates operation performed by PowerFactory, green by Excel, and blue by Python.

7.4.3. Results

7.4.3.1. Inertia Constant Analysis

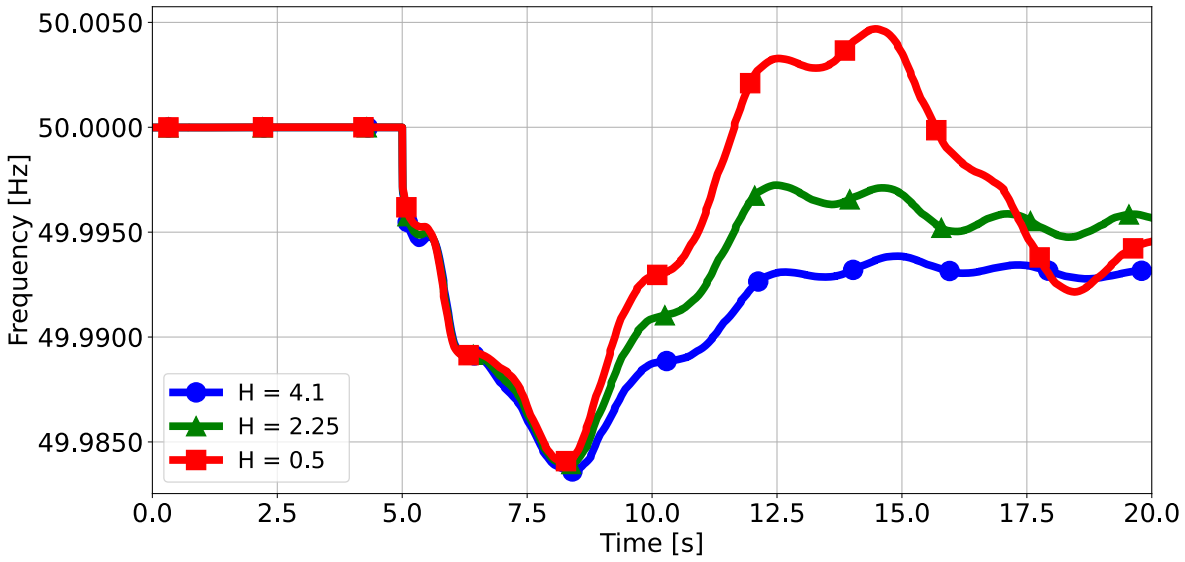


Figure 7.52: Frequency response for different levels of inertia constant H with a kinetic energy level of 1800 GVAs

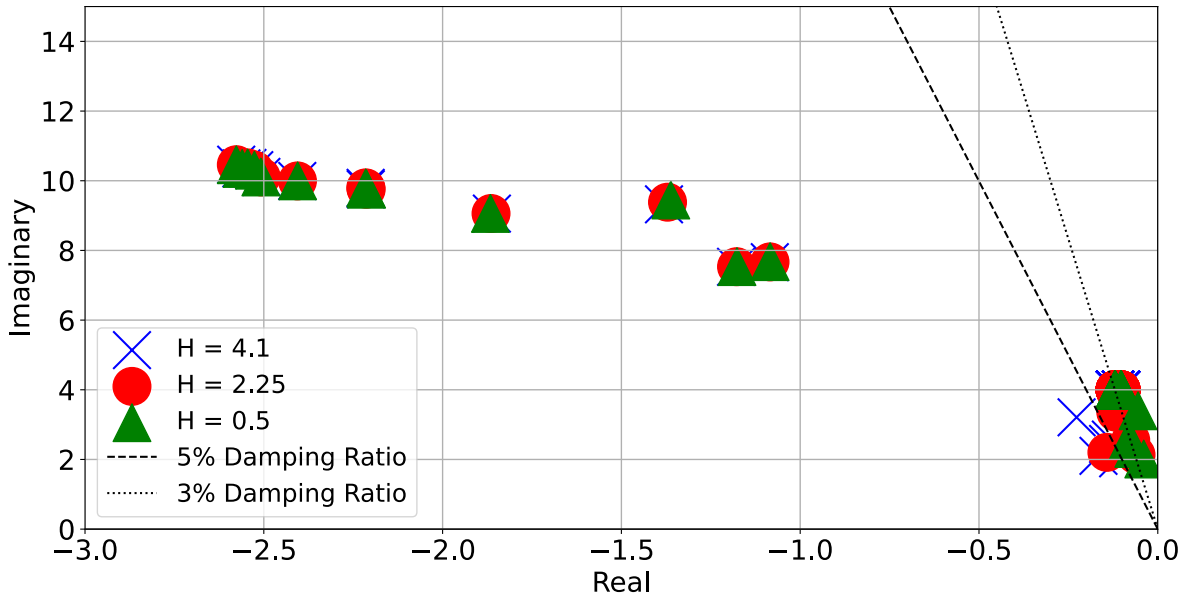


Figure 7.53: Eigenvalues for different levels of inertia constant H with a kinetic energy level of 1800 GVAs

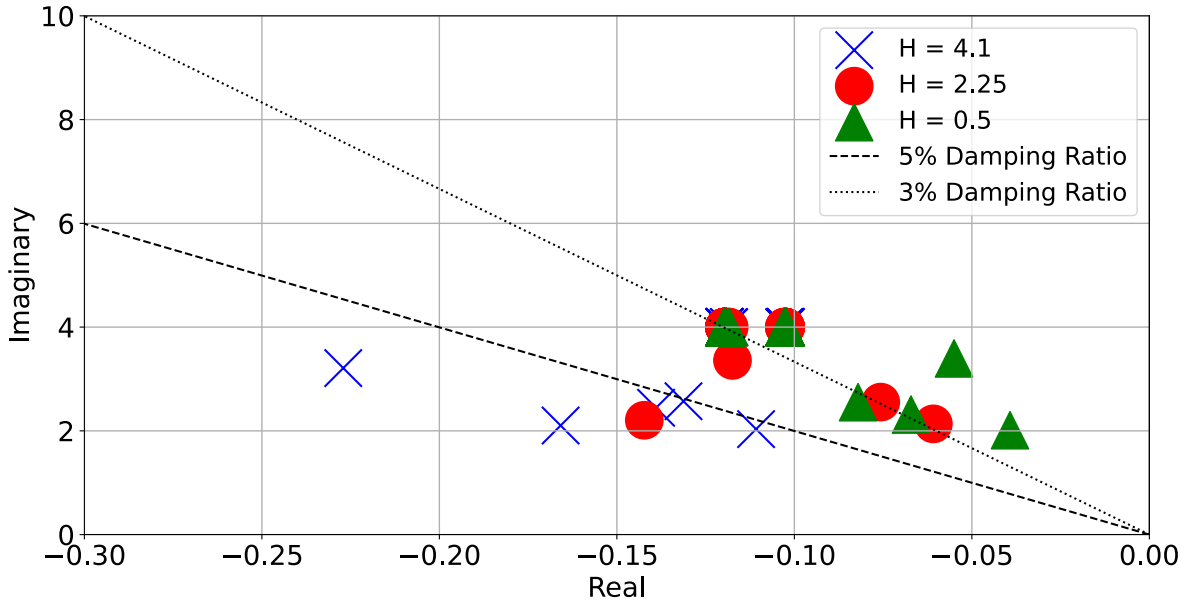


Figure 7.54: Eigenvalues for different levels of inertia constant H with a kinetic energy level of 1800 GVAs, zoomed on the most critical modes

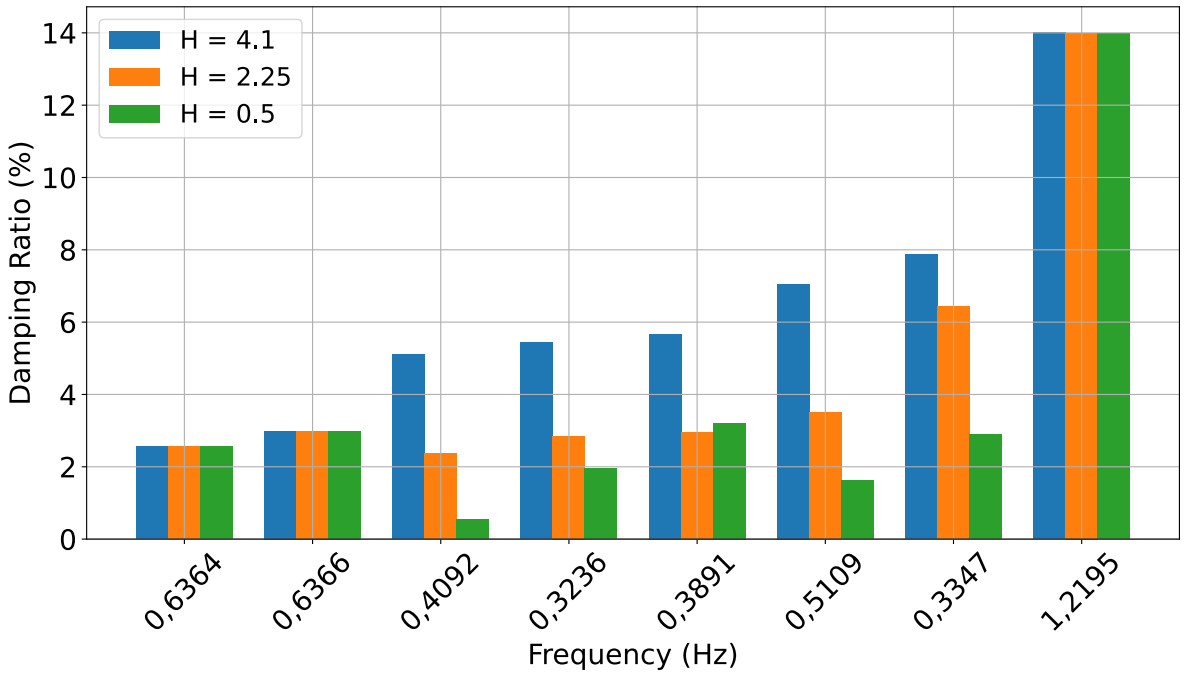


Figure 7.55: Damping ratio versus frequency for different levels of inertia constant H with a kinetic energy level of 1800 GVAs

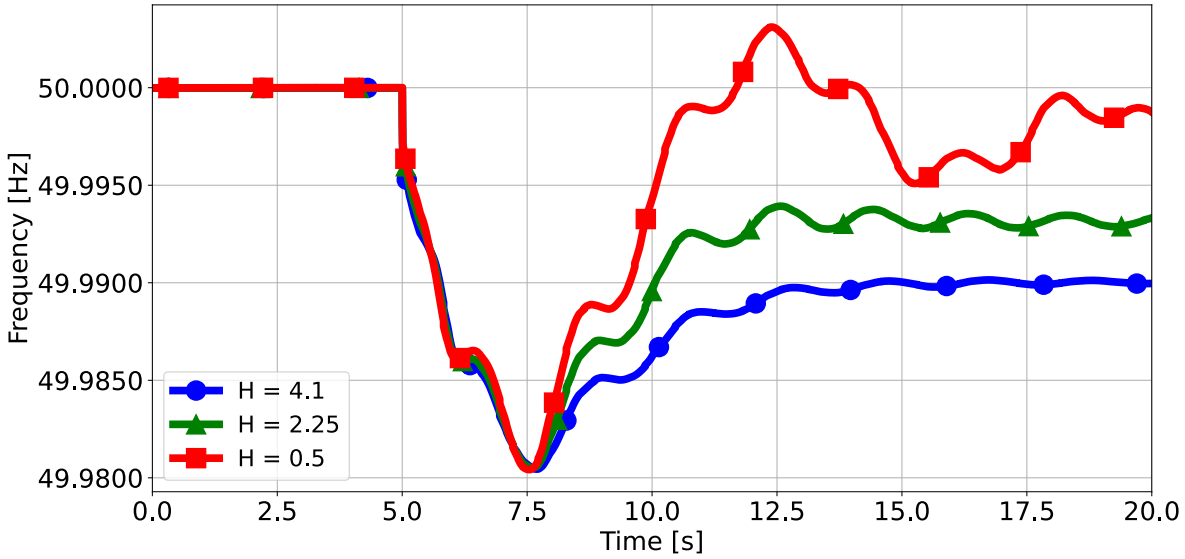


Figure 7.56: Frequency response for different levels of inertia constant H with a kinetic energy level of 1025 GVAs

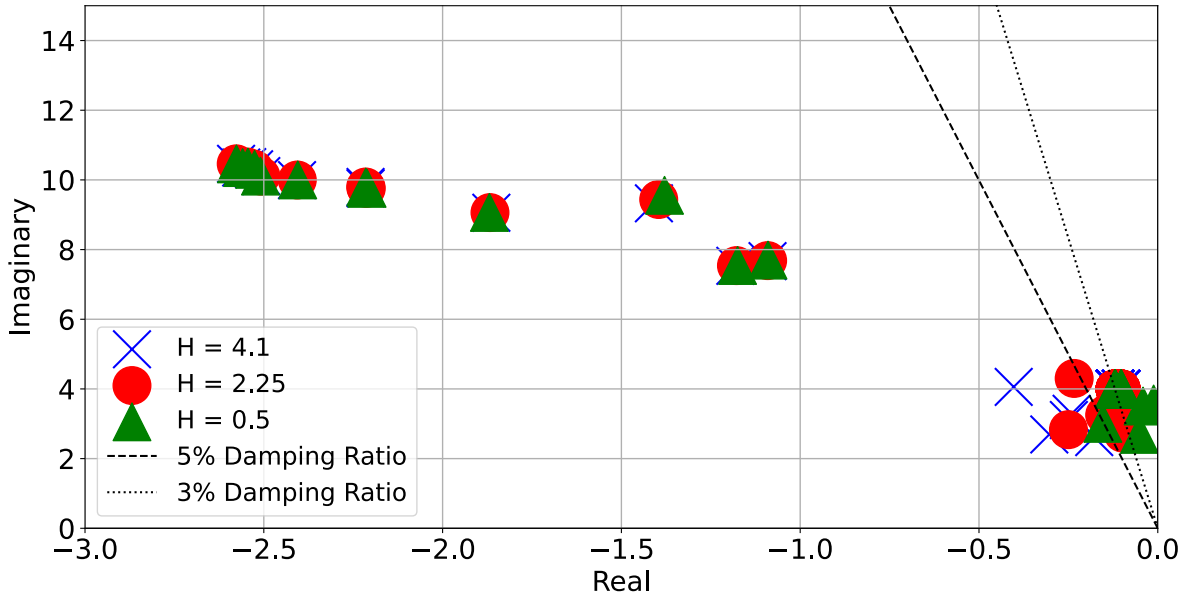


Figure 7.57: Eigenvalues for different levels of inertia constant H with a kinetic energy level of 1025 GVAs

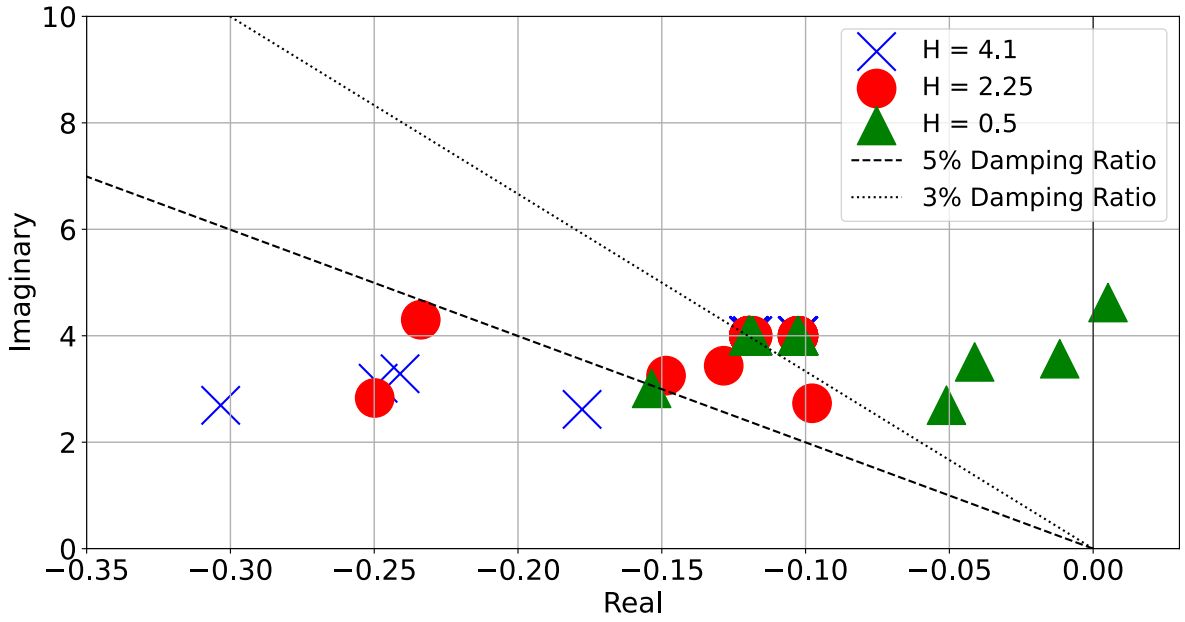


Figure 7.58: Eigenvalues for different levels of inertia constant H with a kinetic energy level of 1025 GVAs, zoomed on the most critical modes

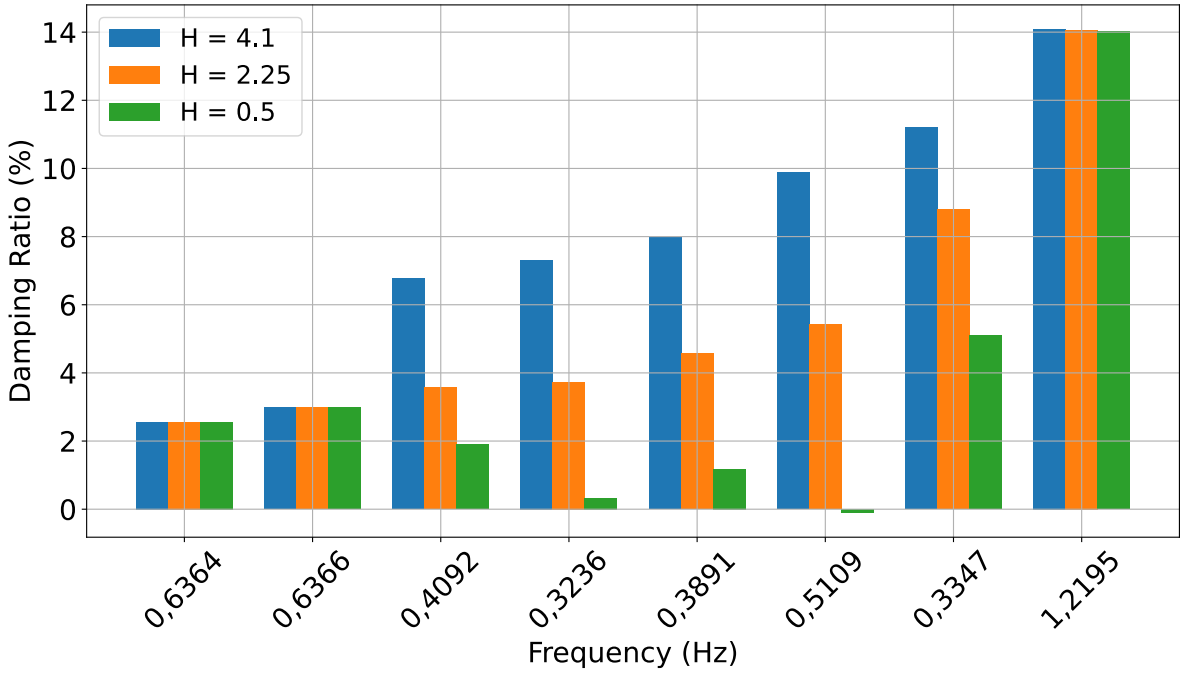


Figure 7.59: Damping ratio versus frequency for different levels of inertia constant H with a kinetic energy level of 1025 GVAs

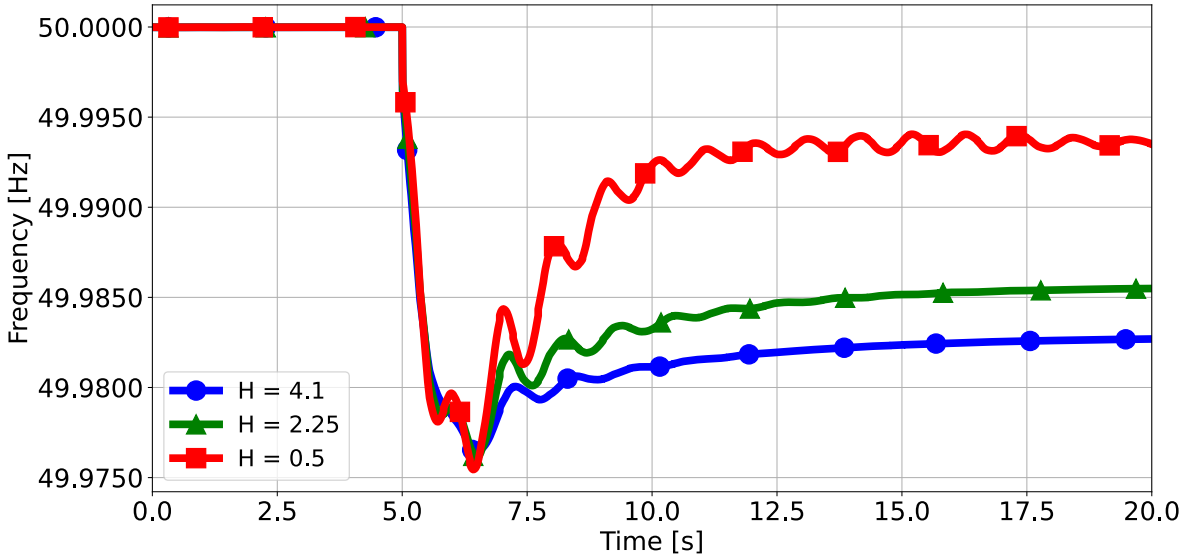


Figure 7.60: Frequency response for different levels of inertia constant H with a kinetic energy level of 250 GVAs

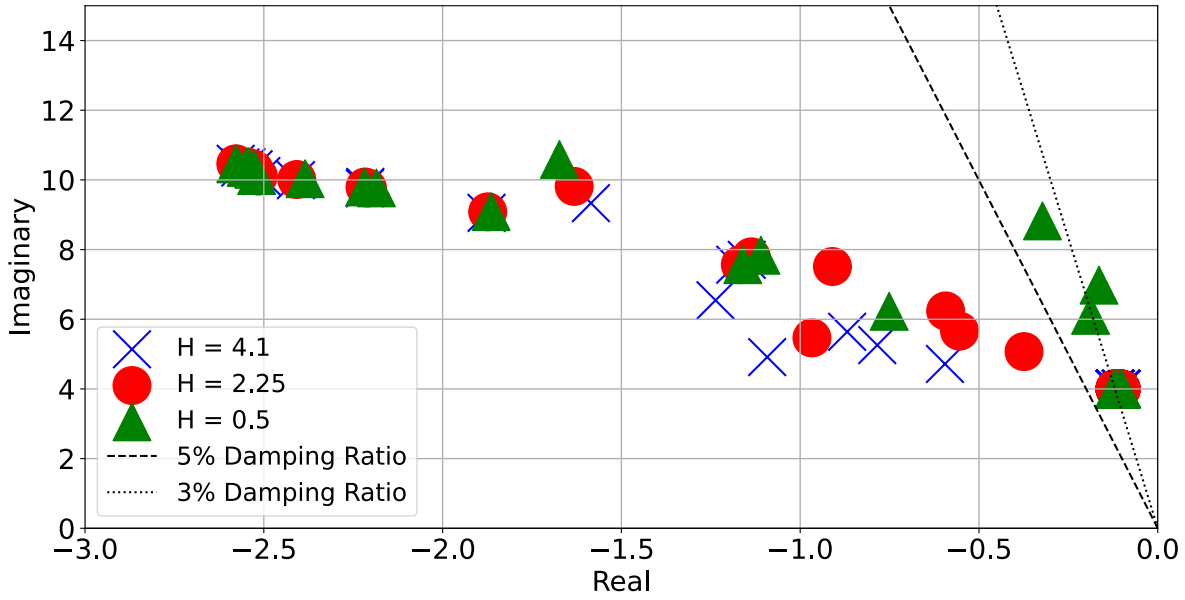


Figure 7.61: Eigenvalues for different levels of inertia constant H with a kinetic energy level of 250 GVAs

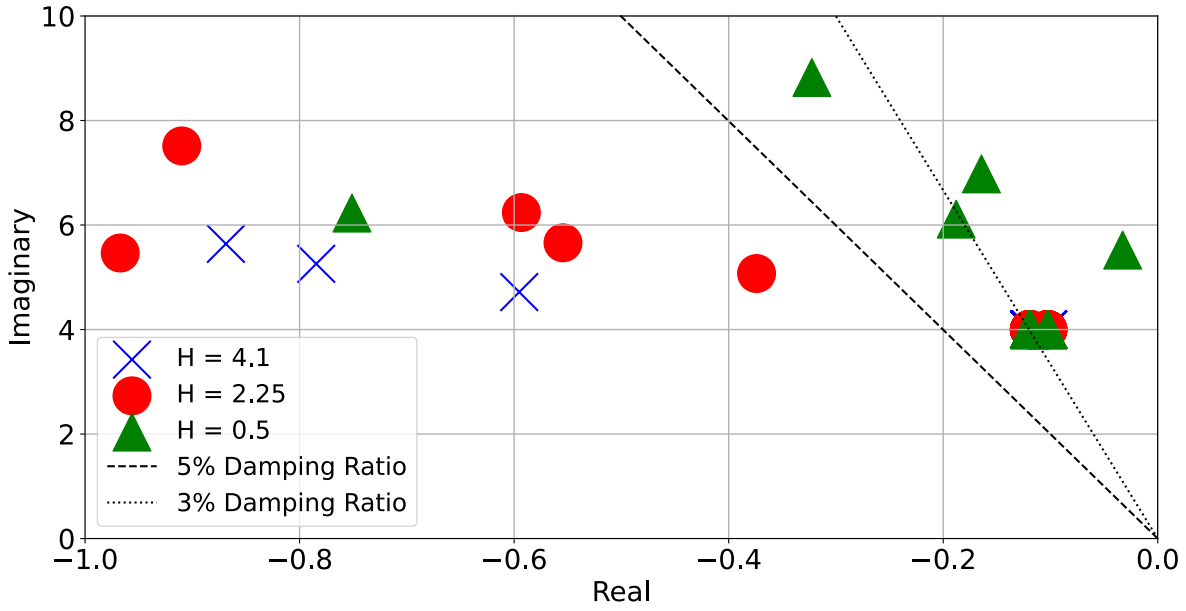


Figure 7.62: Eigenvalues for different levels of inertia constant H with a kinetic energy level of 250 GVAs, zoomed on the most critical modes

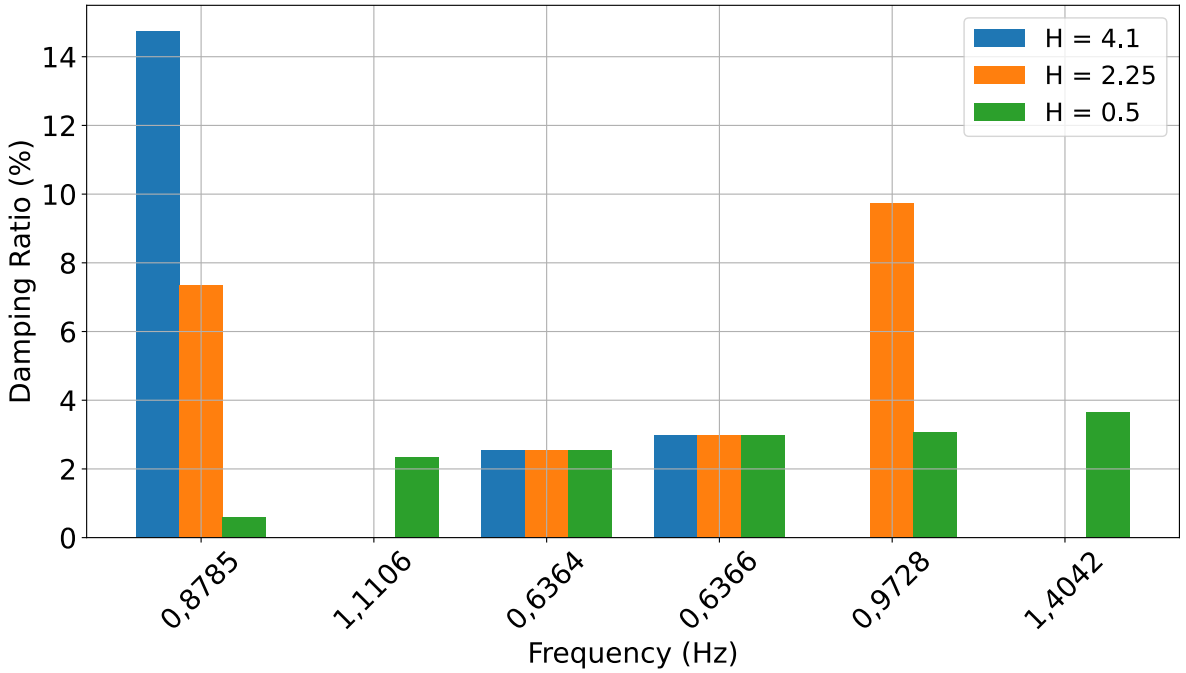


Figure 7.63: Damping ratio versus frequency for different levels of inertia constant H with a kinetic energy level of 250 GVAs

Table 7.16: ROCOF, Frequency Deviation, and Frequency Nadir for Different Levels of Kinetic Energy and Respective Inertia Constants

KE = 1800				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Maximum Deviation (mHz)
		100 ms	500 ms	
H = 4.1	49.9836	6.6	4.9	16.4
H = 2.25	49.9839	8.3	6.4	16.1
H = 0.5	49.9841	9.7	8.1	15.9

KE = 1025				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Maximum Deviation (mHz)
		100 ms	500 ms	
H = 4.1	49.9804	6.1	5.4	19.6
H = 2.25	49.9805	8.2	7.5	19.5
H = 0.5	49.9804	13.3	10.7	19.6

KE = 250				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Maximum Deviation (mHz)
		100 ms	500 ms	
H = 4.1	49.9764	7.5	6.2	23.6
H = 2.25	49.9760	14.1	10.6	24.0
H = 0.5	49.9754	23.1	17.3	24.6

7.4.3.2. Kinetic Energy Analysis

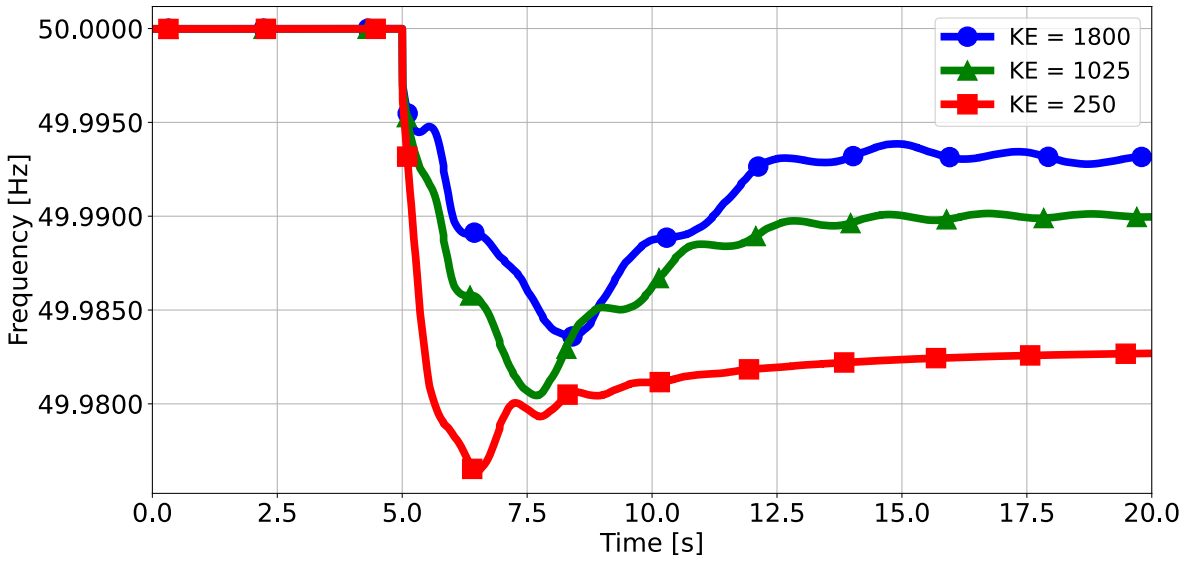


Figure 7.64: Frequency response for different levels of kinetic energy, with a fixed inertia constant H at 4.1 s

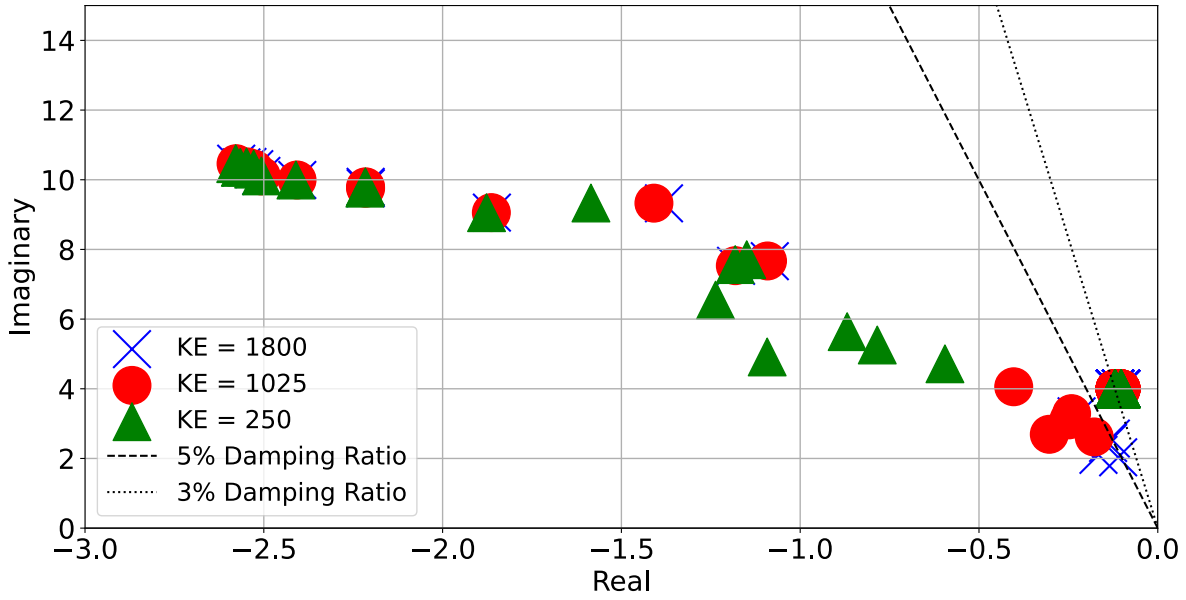


Figure 7.65: Eigenvalues for different levels of kinetic energy, with a fixed inertia constant H at 4.1 s

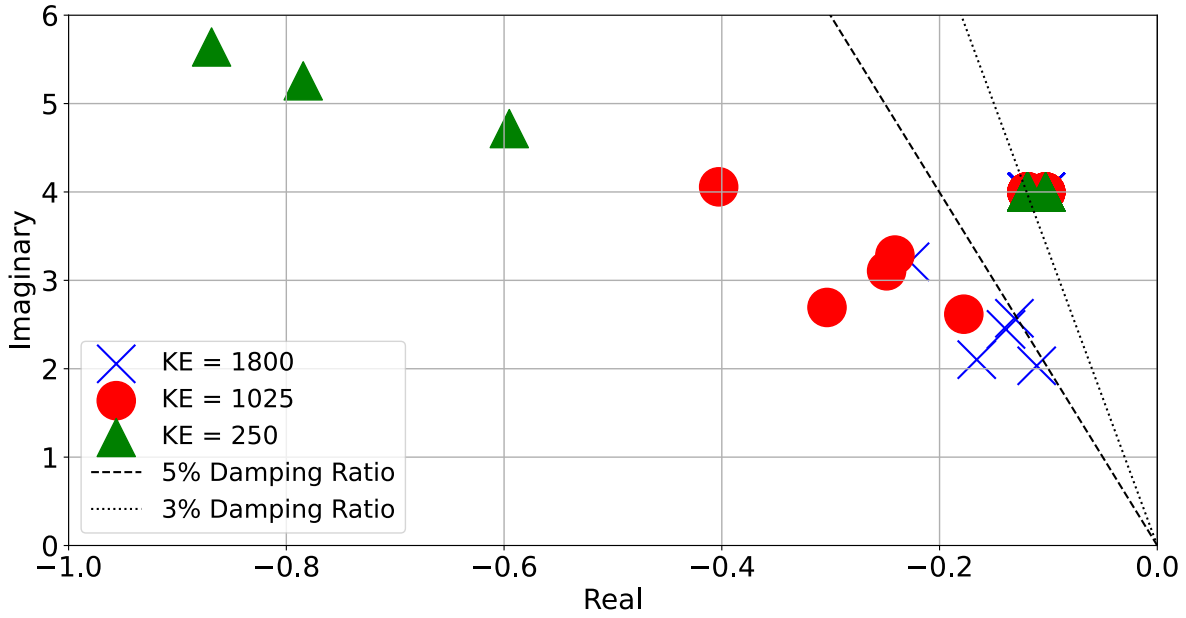


Figure 7.66: Eigenvalues for different levels of kinetic energy, with a fixed inertia constant H at 4.1 s, zoomed on the most critical modes

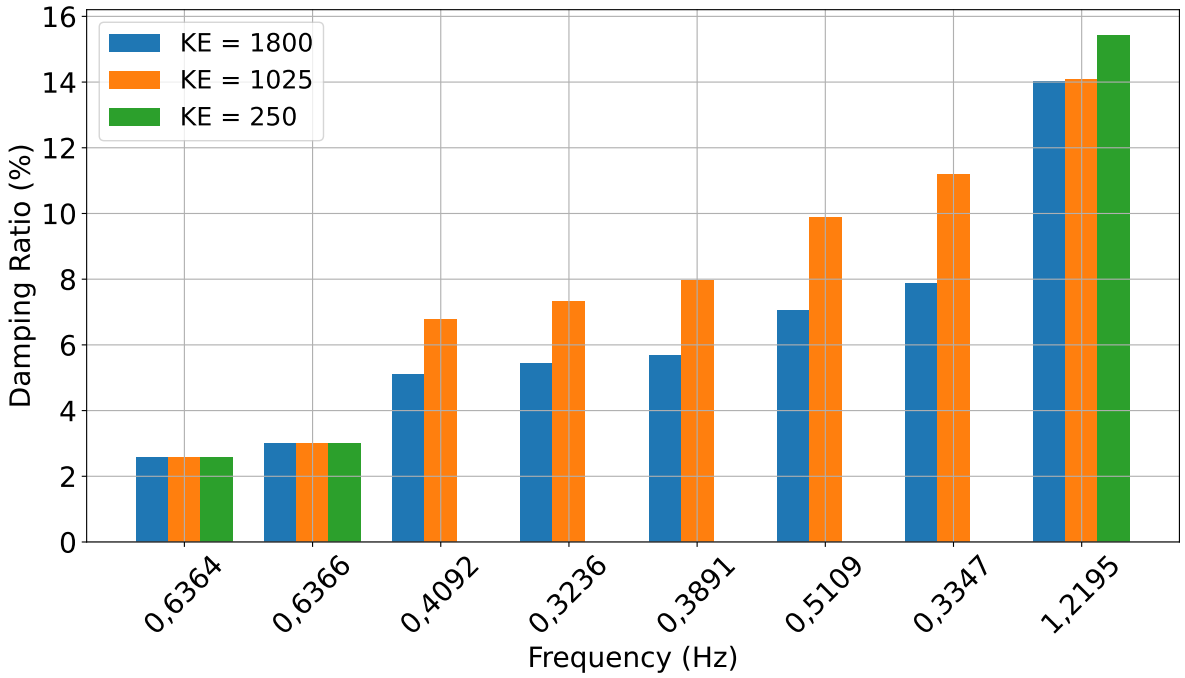


Figure 7.67: Damping ratio versus frequency for different levels of kinetic energy (KE), with a fixed inertia constant H at 4.1 s

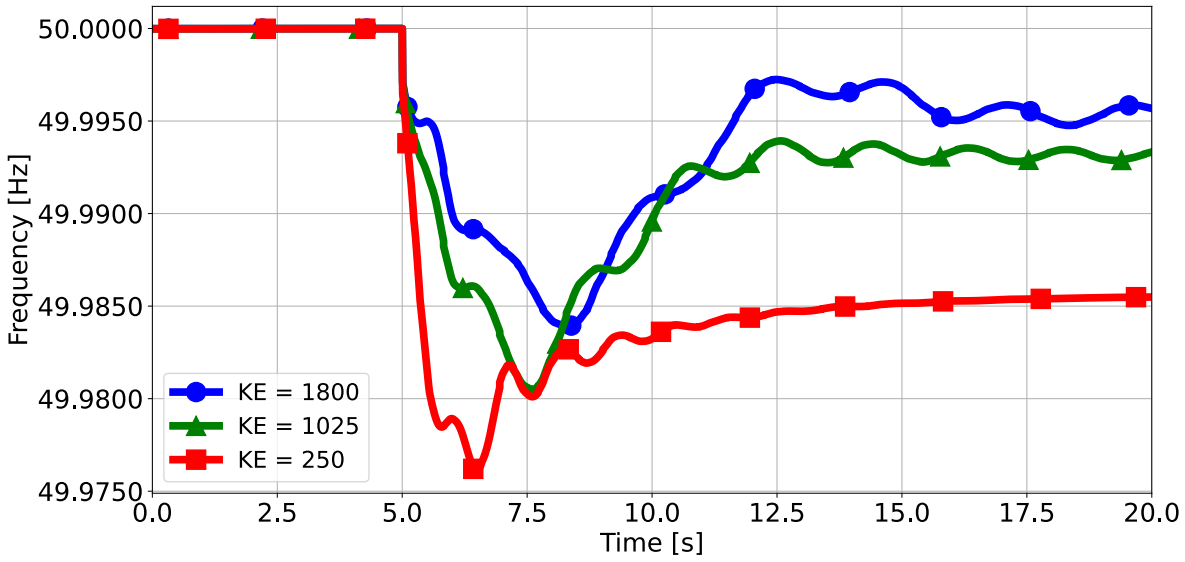


Figure 7.68: Frequency response for different levels of kinetic energy, with a fixed inertia constant H at 2.25 s

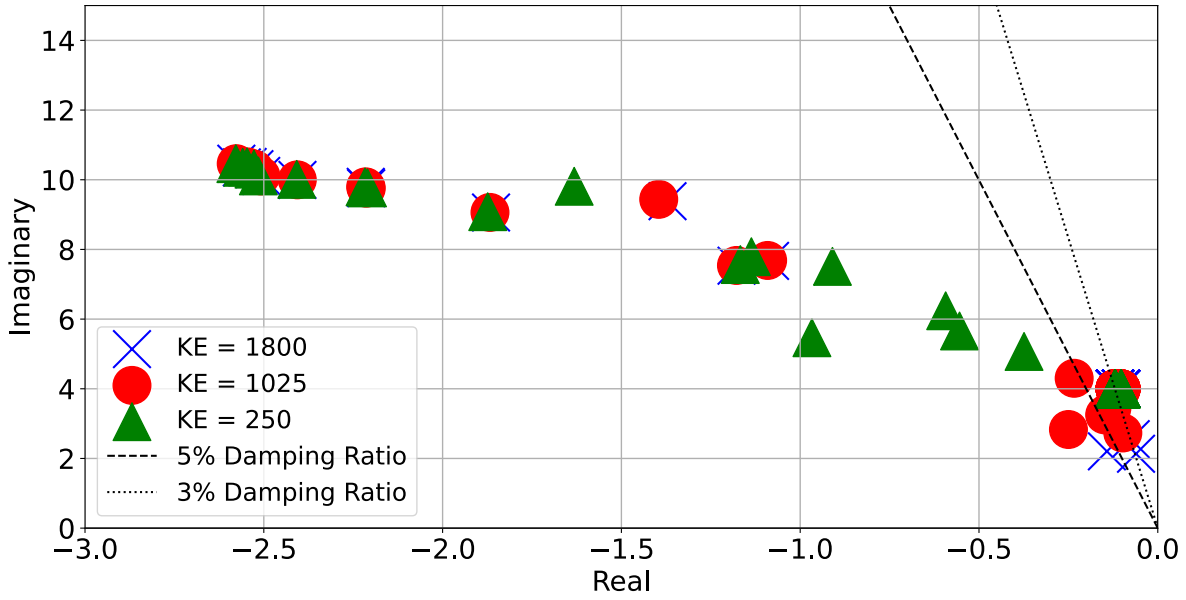


Figure 7.69: Eigenvalues for different levels of kinetic energy, with a fixed inertia constant H at 2.25 s

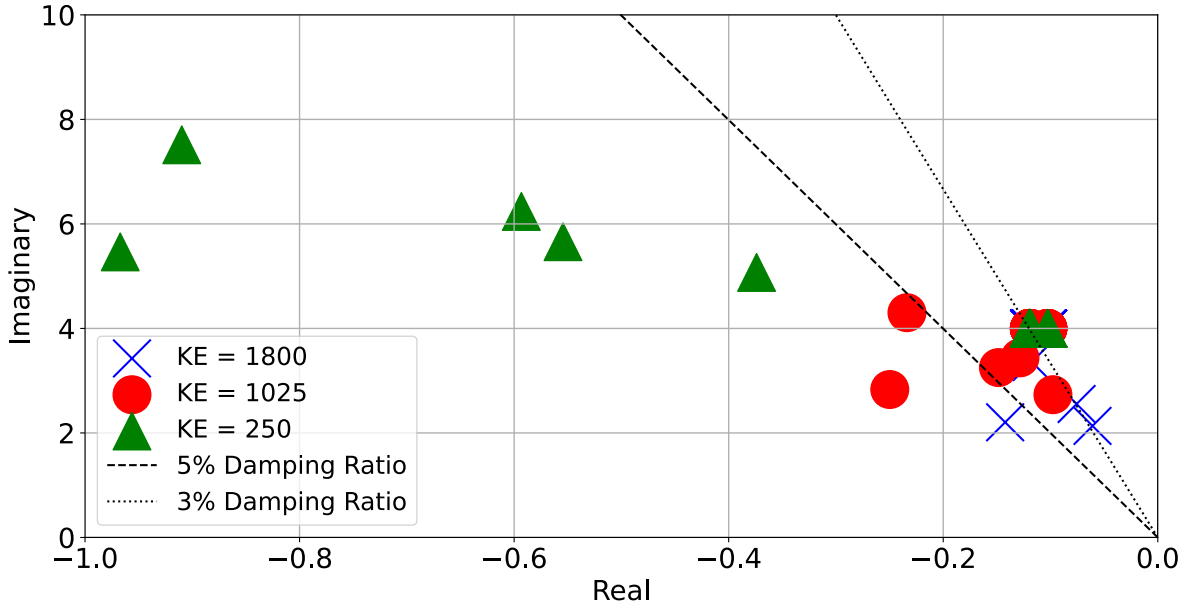


Figure 7.70: Eigenvalues for different levels of kinetic energy, with a fixed inertia constant H at 2.25 s, zoomed on the most critical modes

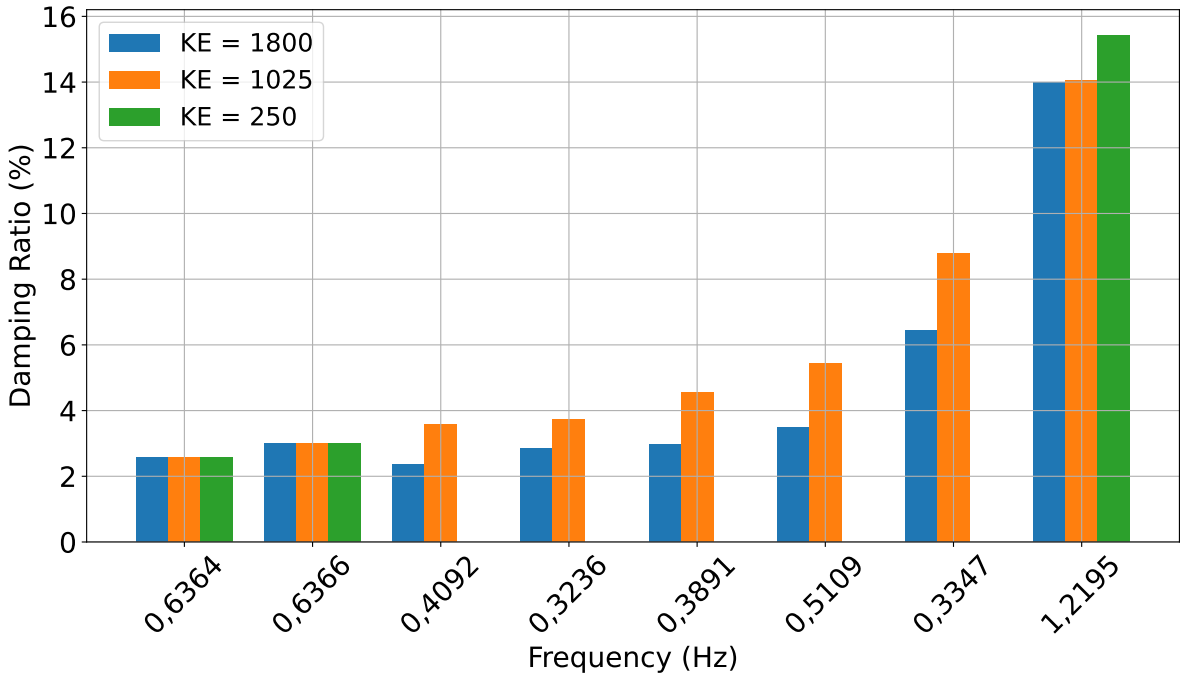


Figure 7.71: Damping ratio versus frequency for different levels of kinetic energy (KE), with a fixed inertia constant H at 2.25 s

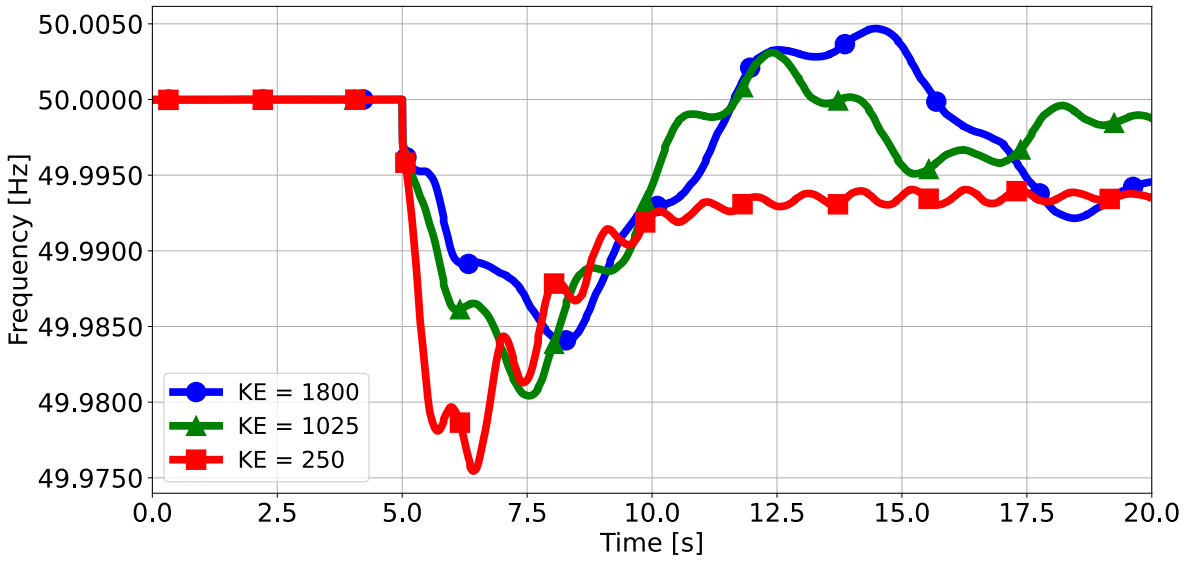


Figure 7.72: Frequency response for different levels of kinetic energy, with a fixed inertia constant H at 0.5 s

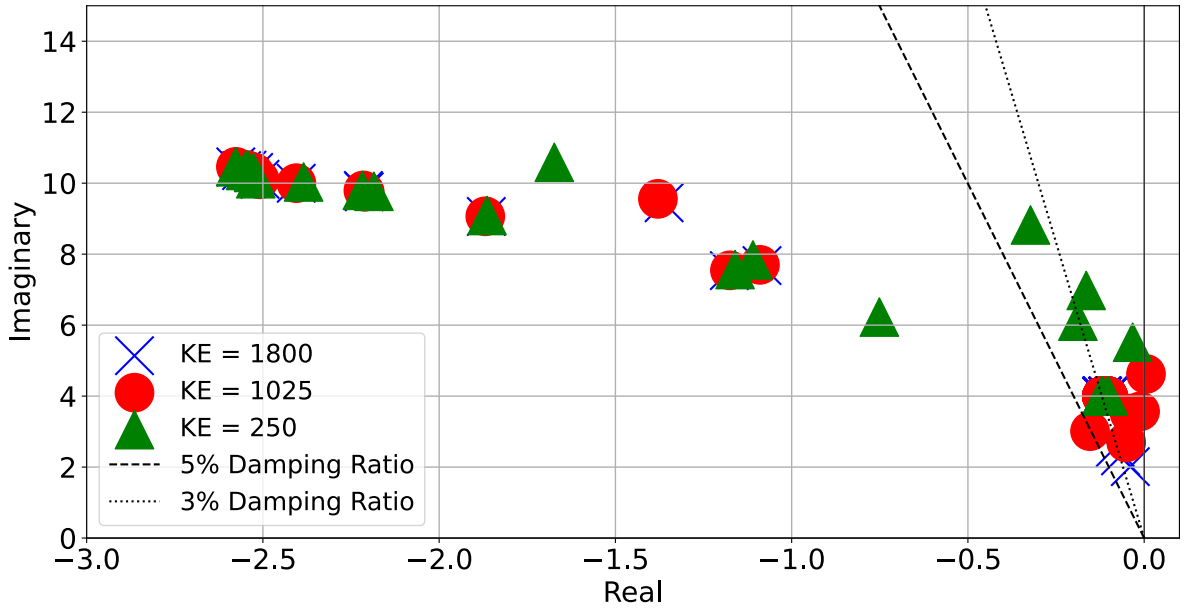


Figure 7.73: Eigenvalues for different levels of kinetic energy, with a fixed inertia constant H at 0.5 s

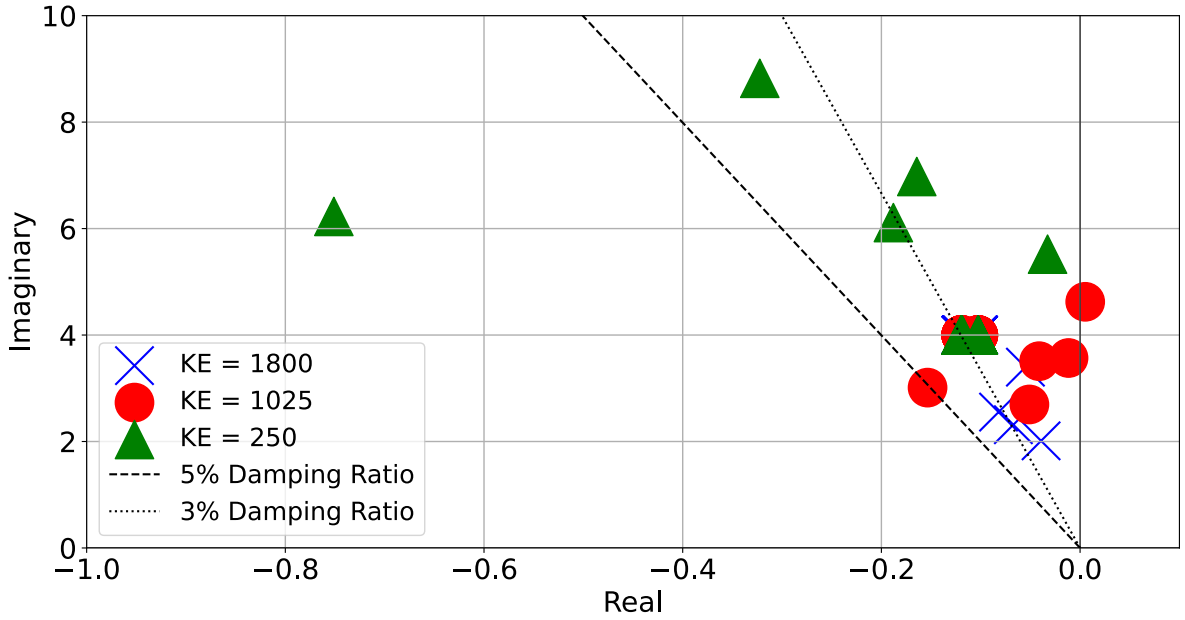


Figure 7.74: Eigenvalues for different levels of kinetic energy, with a fixed inertia constant H at 0.5 s, zoomed on the most critical modes

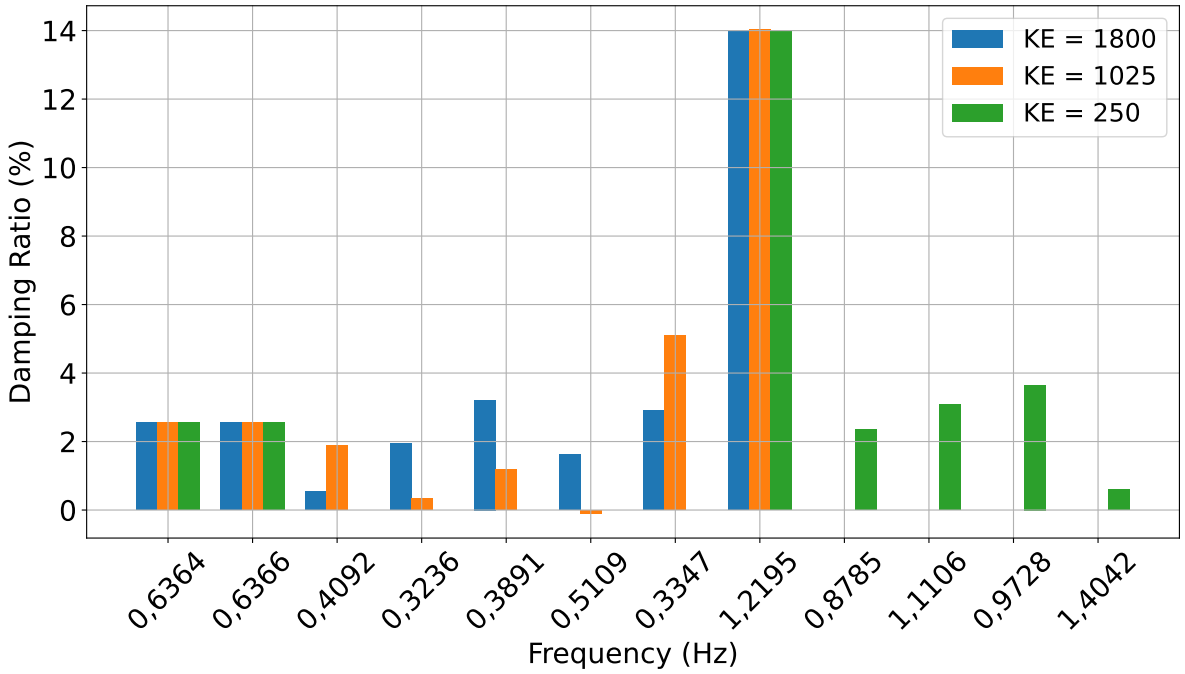


Figure 7.75: Damping ratio versus frequency for different levels of kinetic energy (KE), with a fixed inertia constant H at 0.5 s

Table 7.17: ROCOF, Frequency Deviation, and Frequency Nadir for Different Levels of Inertia Constant, and Respective Kinetic Energy Levels

H = 4.1				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
KE = 1800	49.9836	6.6	4.9	16.4
KE = 1025	49.9804	6.1	5.4	19.6
KE = 250	49.9764	7.5	6.2	23.6

H = 2.25				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
KE = 1800	49.9839	8.3	6.4	16.1
KE = 1025	49.9805	8.2	7.5	19.5
KE = 250	49.9760	14.1	10.6	24.0

H = 0.5				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
KE = 1800	49.9841	9.7	8.1	15.9
KE = 1025	49.9804	13.3	10.7	19.6
KE = 250	49.9754	23.1	17.3	24.6

7.4.4. Discussion

7.4.4.1. Inertia Constant Analysis

First, the results from the changing inertia constant levels are discussed. The results were analyzed with respect to dynamic changes in the system's inertia constant and constant kinetic energy. To comprehensively understand the potential changes, the impact of varying the inertia constant was examined through both frequency and eigenvalue analyses for all three kinetic energy levels.

First, the frequency response for the highest level of kinetic energy is depicted in Figure 7.52. It is evident that all three levels of inertia constant exhibit a similar downward slope from the start of the load increase until around the 8-second mark when the frequency starts to rise again. Small differences are observed, with higher inertia constants leading to a slightly slower response, resulting in higher allowed frequency deviations for each oscillatory response. However, this difference is minuscule, as shown in Table 7.17, with a difference of around 0.5 mHz between the 4.1 and 0.5 s inertia constants. The largest difference appears when the frequency starts to increase after the load increases. Lower inertia constant values result in a much faster-acting response compared to higher values. However, this faster response is not as well-damped, causing the frequency to exceed pre-disturbance levels and not properly settle within the 20-second time window. In contrast, the 2.25 and 4.1 s values show a slower response once the frequency rises but settles better, with only small oscillations evident by the end of the 20-second window. Overall, the frequency response for different inertia constants for the highest kinetic energy levels shows a similar maximum frequency deviation but a significant difference in the system's damping and settling time, favoring higher inertia constants.

Analyzing the results for the two lower levels of kinetic energy, seen in Figure 7.56 and Figure 7.60, similar patterns emerge. All three kinetic energy levels exhibit a similar maximum deviation for all inertia constants, but differences appear in the response where the frequency starts to rise. At an inertia constant of 0.5 s, the kinetic energy level of 1025 GVAs shows an overshoot above the initial steady-state frequency and does not settle within the 20-second simulation window. In contrast, the 250 GVAs level does not reach above the original steady-state frequency and settles within the 20-second period, although with more pronounced local oscillations. These oscillations persist almost until the end of the 20-second window for the 0.5 s inertia constant but stop around 12.5 seconds for the 2.25 s and around 10 seconds for the 4.1 s constants, indicating that higher inertia constants provide a more stable settling signal. Lower inertia constants allow for faster frequency changes, which, combined with the fast-acting nature of future systems, can cause more sustained local oscillations. This behavior is both positive and negative: while lower inertia constant values result in a primary frequency restoration closer to the pre-disturbance value, they also exhibit more oscillatory behavior.

The analysis shows that only small maximum deviation changes occur with different inertia constants, while larger deviations are evident between kinetic energy levels, which is expected since kinetic energy levels are the first line of defense against changes to the frequency. Lowering the inertia constant allows for a faster overall system response due to the lower time constant, but the fast-acting nature of converters combined with this rapid response can cause more oscillations. This could lead to fast damping and a better overall response for a perfectly tuned system. However, given the various changes in system characteristics and operations, combined with the unknown features of future systems, achieving such a system is still under investigation. It is clear that reducing the inertia constant generally leads to a more oscillatory, poorly damped frequency response, evident across all three kinetic energy levels.

Moving on to the eigenvalue analysis for dynamic inertia constants and set kinetic energy levels, the results for 1800 GVAs are presented in Figure 7.53 to Figure 7.55. It is observed that well-damped modes have nearly identical damping ratios across all inertia values, but significant differences are found in the most critical modes. Higher inertia constant values exhibit better damping. The 0.5 s inertia constant results in multiple modes with damping ratios below 3 %, while the 4.1 s inertia constant has all inter-area modes damped above 5 %. The lowest damped mode at 0.4092 Hz has a damping ratio below 1 % for the 0.5 s constant, compared to around 5 % for the 4.1 s constant. This pattern is consistent across all inter-area modes, significantly improving damping ratios with increased inertia constants. No significant changes is observed in intra-area modes associated with grid-following converters.

Analyzing the 1025 GVAs kinetic energy levels in Figure 7.57 to Figure 7.59, similar findings are noted. The higher inertia constant levels significantly enhance the stability of the most critical modes. For 0.5 s levels, positive real parts of the eigenvalues indicate system instability, with damping ratios below 3 % for several modes. The 2.25 s and 4.1 s constants provide much better damping, with all inter-area modes having damping ratios above 3 %. No differences were found for the intra-area modes of the grid-following converters.

Moving on to the kinetic energy levels of 250 GVAs, a similar pattern is again observed, where the lower H values result in decreased damping. However, it is interesting to note that the overall damping of the higher inertia constants is better than that of the previous levels. Here, it is seen that most of the modes of the 4.1 and 2.25 s values have damping ratios well within the 5 % range, with an exception for the inter-area grid-following modes. This overall favors the higher inertia constant values with respect to damping. Comparing the modes to the two previous levels, it is also evident that the overall oscillations have a higher oscillatory speed, which is to be expected as the kinetic energy in the system decreases and a more fast-acting system comes through.

When analyzing the most critical modes, it is seen that the 0.5 s value has some modes beneath the 3 % damping ratio limit, and one of them is as low as 0.59 %, indicating that system stability is in critical limits and cannot be guaranteed at this level. When examining the frequencies of the different modes, due to the faster-acting nature of the system now, new modes are created, which are different from those for the 1800 and 1025 GVAs levels. Instead, new poorly damped, faster-acting modes are evident. While these new modes now have an increased oscillatory speed, they still primarily involve interconnections. The participation factors for the newly discovered modes can be seen in Figure B.50 to Figure B.53.

Mode 131, with a frequency of 0.973 Hz, is still primarily dominated by the interconnections, but there is now some participation by the nuclear and hydrogen power plants at BSL380 as well, although small compared to the two primary interconnections. A similar story is true for mode 124; here again, the main participation factors are the interconnections, with a small participation from the BSL380 hydrogen power plant, but primarily dominated by the interconnections, same as for mode 193. For mode 13, only the interconnections participate. This mode also exhibits the same modes as previously with regard to the intra-area modes for the grid-following converters with a frequency of around 0.6366 Hz. As before, the kinetic energy and inertia constant do not play a part in damping these modes. Interestingly, none of the modes exhibit a low damping of any of the other intra-area modes associated with this, with a frequency of around 1.2 Hz. This indicates that this single phenomenon did not translate to lower kinetic energy levels and is most likely due to the lower damped inter-area oscillations found in the original simulations.

Interestingly, while the interconnection modes include all the different interconnections, it is seen that for all the modes, the interconnection at MBT380 is the first or second largest participation factor for all modes except for mode 124. The location of the MBT380 interconnection is of interest. While the other AC interconnections are all lined up on the far east side of the power system, in a line, the MBT380 is placed more inland and southern towards Belgium. Because of this, it has a slightly different placement than the other interconnections, contributing to the different modes. A similar pattern was evident in the original eigenvalue analysis regarding the modes, but here, the frequencies and participations largely suggest inter-area modes influenced by all the different interconnections. Analyzing the frequency of the three modes with heavy participation by MBT380, the frequencies of interest are 0.878, 0.973, and 1.404 Hz, respectively. According to the earlier definition with regards to mode identification, these fall into local modes with ranges from 0.7 to 2 Hz or intra-area oscillations with frequencies between 0.4 to 1 Hz. However, as inter-area oscillations typically stop at around 0.7 Hz, due to the changing system dynamics associated with these low kinetic energy and inertia constant system parameters, it could be true that the inter-area oscillatory mode definition has changed as well, representing faster-acting inter-area oscillations.

Based on this, it is assumed that modes 13 and 131 are still evident as inter-area modes but with increased oscillatory speed. This is due to the nature of the different areas represented in the participation modes,

indicating an oscillatory dynamic between different areas. For mode 121, with a frequency of 1.111 Hz, only the interconnections on the eastern side of the country are part of the participation for this mode. Due to the nature of the oscillations within the same "area" part of the power system, this is classified as an intra-area mode. As before, while the intra-area mode typically falls within frequencies between 0.4 to 1 Hz, due to the faster-acting nature of the system now, it is assumed that even frequencies up to 1.111 Hz can exhibit intra-area oscillatory modes. Finally, mode 193, with a frequency of 1.404 Hz, is identified. Here, the main participation comes from the interconnection at OBZ380, with the secondary coming from the MBT380 interconnection. These generators are closely located, suggesting an intra-area oscillation with an even higher frequency.

As seen in this analysis, the definition of these modes is expected to change as the overall dynamics of future power systems change. As the kinetic energy drops in combination with different inertia constants, new faster-acting modes are observed in the system, impacting the system differently than the original inter-area modes found for higher kinetic energy levels. The frequency response findings also discussed the faster-acting oscillations, which showed a quick settling time but with high amounts of smaller, faster oscillatory modes in the response.

It is evident that the combination of low inertia constant and kinetic energy levels shifts the frequencies of the different modes, complicating their identification with respect to oscillatory behavior. While some of these modes were evident for different inertia constants, others were not. Analyzing the different inertia constants, it is evident that even with low kinetic energy levels, there are well-damped modes for these system characteristics. For example, with an inertia constant of 2.25 s, modes with frequencies of 0.8785 and 0.9728 Hz were found, ignoring the grid-following modes that did not change. These two modes are the lowest damped for the system, with damping ratios of 7.35 % and 9.75 % respectively. Thus, with these two being the lowest damped modes in the system (ignoring the grid-following modes), it can be concluded that this combination actually has a well-damped oscillatory behavior, which was also evident in the frequency response analysis. A similar analysis was found for the 4.1 s inertia constant. Here, only the mode with a frequency of 0.8785 was identified, which is similar to the two other lower-level inertia constant modes. However, all the modes with this system characteristic had excellent damping, with none of the modes exhibiting a damping ratio below 10 %.

Once again, the worst damping performance occurs for the system with the lowest inertia constant. However, unlike the 1025 GVAs simulation results, it can be seen here that while the system has very low and poor damping, no unstable modes are observed in the analysis. This indicates that the smaller inertia constant has improved damping for the lowest levels of kinetic energy compared to the medium levels. This is likely due to the new fast-acting nature of both the kinetic energy and inertia constant levels working closer in harmony, compared to medium responsiveness for medium kinetic energy levels and fast responsiveness for the low inertia constant. While the low inertia constant did not show any unstable modes for the highest level of kinetic energy either, the kinetic energy levels were probably high enough to keep the system stable alone, maintaining its limit even with the slow-acting nature of the system. Thus, while there are new interesting mode behaviors for the lowest level of kinetic energy and with a properly tuned control system, it is possible to combine lower inertia constants with lower kinetic energy values to create a fast-acting, properly balanced power system.

In conclusion, the analysis shows that different kinetic energy and inertia constant values significantly impact the system's stability, both in terms of frequency stability and rotor angle oscillatory stability. Lower inertia constants had worse performance for all three kinetic energy levels, indicating a preference for higher inertia constants. However, comparing different inertia constants across different kinetic energy levels revealed a more complex analysis. This will be further analyzed in the next section. Overall, the future interconnected power system and internal inertia constant and kinetic energy levels will play a vital role in future system stability, and further research with respect to the tuning of controller parameters should be conducted to address the stability issues associated with these system characteristics.

7.4.4.2. Kinetic Energy Analysis

Next, the simulations performed for a dynamic change in the kinetic energy levels with respect to a constant inertia constant will be discussed. These simulations were conducted with a constant inertia constant and varying kinetic energy levels, providing an inverse analysis compared to the inertia constant study.

Starting with the inertia constant of 4.1 s, the frequency response analysis results for the different kinetic energy levels can be seen in Figure 7.64, with the frequency nadir, ROCOF, and maximum deviation provided in Table 7.16. It is evident that the highest kinetic energy levels result in the best system performance. The highest kinetic energy level shows a frequency deviation of 16.4 mHz compared to 23.6 mHz for the lowest kinetic energy level, a difference of 30.5 %. The higher kinetic energy levels also display the best ROCOF values for the 500 ms simulation window, while the 1025 GVAs level performs best for the 100 ms window. The lowest kinetic energy levels perform the worst in both cases.

When the initial load response occurs, the lower kinetic energy levels exhibit a faster and larger initial deviation before the response occurs. This results in a larger maximum deviation but allows the response of the signal to occur faster. The maximum deviation for the 1800 GVAs occurs around the 8-second mark, at approximately 7.5 seconds for the 1025 GVAs, and around 6-6.5 seconds for the 250 GVAs. It is known that kinetic energy levels serve as the "first line of defense" in the system, and thus, the initial frequency deviation is largely dependent on the kinetic energy levels, as highlighted here.

After the initial deviation, the frequency increases for all responses. The order is similar to the frequency deviation, with higher kinetic energy levels allowing for a higher frequency when the simulation window is up compared to the lower values. The difference between the minimum frequency nadir and the new settling time is small for all three responses, but it is evident that higher kinetic energy levels result in a better frequency nadir and a better new settled frequency level after the initial event. However, the oscillatory nature of the lower kinetic energy levels dampens quickly, and the overall oscillatory nature stops around the 10-second mark. For the 1025 GVAs, this is not properly dampened until the 20-second window is up, but it starts to become a smooth signal, while the 1800 GVA still has the largest oscillations occurring. Although these oscillations are not large, they indicate the damping characteristics of the different levels. Typically, while higher kinetic energy levels provide better system defense against initial frequency deviation, they are known to have a slower settling time due to the large amounts of rotational forces necessary to reach new steady-state operation, potentially causing oscillatory performance issues, often influenced by larger inter-area oscillations as highlighted in Case Study A.

Moving on to the eigenvalue analysis, as seen in Figure 7.65 to 7.67, a similar pattern is observed. Major deviations in the system's overall damping and oscillatory performance favor lower kinetic energy levels. The lower kinetic energy levels shift the modes to have a faster oscillatory response and improved damping. Analyzing the most critical modes, the same grid-following inter-area converter modes are still evident and unchanged. For the other critical modes, the 1800 GVAs kinetic energy levels have modes closer to the critical limit. There are no modes, except for the grid-following modes, with damping ratios below 5 % and overall worse damping ratios than the 1025 and 250 GVAs levels.

Comparing the damping ratios for these critical modes, it is seen that all three levels have the same damping ratio for the 0.63 Hz intra-area grid-following converter modes. However, the lower kinetic energy levels have increased damping for all the inter-area modes associated with the system. The comparisons are based on the results from the 1800 GVAs scenario, as these have the lowest damping ratios. The same inter-area oscillations are not found for the 250 GVAs kinetic energy levels; results similar to the low kinetic energy levels with varying inertia constants are found. Importantly, even though these modes were found for the low kinetic energy levels, none had damping ratios below 10 %, indicating greatly improved damping compared to the other levels. Therefore, it can be concluded that for a higher inertia constant, lower kinetic energy levels improve system damping but allow larger frequency deviations, ROCOF values, and a lower steady-state frequency value after the disturbance.

Next, the results for an inertia constant of 2.25 s are discussed, with the frequency response shown in Figure 7.68 and the calculation indices in Table 7.16. Similar to the 4.1 s inertia constant, the overall frequency deviation is higher for lower kinetic energy levels, ranging from 16.1 to 24 mHz. The settling times of the signal are also different but decreased for all kinetic energy levels as the inertia constant is reduced. The maximum frequency deviation occurs at 8, 7.5, and 6 seconds for the highest to lowest kinetic energy levels. The higher kinetic energy levels also result in higher settled frequency levels after the disturbance. However, it is clear that the settling of the frequency is worse for the lower inertia constant. The higher kinetic energy levels also exhibit sustained oscillations compared to the lowest kinetic energy value, highlighting a similar analysis to the higher inertia constant but with slightly more oscillations evident. For the lowest kinetic energy levels, the oscillations settle around the 14-second mark compared to the 10-second mark identified in the higher inertia constant simulations. The ROCOF values are higher than those of the 4.1 s inertia constant results, even though the frequency deviations are relatively similar for all scenarios. The largest difference is seen in the 100 ms simulation window for the lowest kinetic energy values, with an 88 % difference favoring higher inertia values. A similar difference is true for the other results but on a smaller scale. Overall, the frequency response is similar to the higher inertia constant analysis, still exhibiting both positive and negative impacts with increased and decreased inertia constant levels, not providing a clear performance enhancement between the different values.

Finally, the results for the inertia constant of 0.5 s are discussed. The frequency response and frequency calculation parameters are shown in Figure 7.72 and 7.16. A similar pattern is observed when comparing the three different levels of kinetic energy, with a later maximum frequency deviation time and a better maximum deviation. However, the overall oscillations in the frequency response are different. The maximum deviations occur after 8, 7.5, and 6 seconds for 1800, 1025, and 250 GVAs, respectively, similar to the 2.25 s inertia constant, with maximum frequency deviations ranging from 15.9 to 24.6 mHz. The ROCOF values are higher than the other inertia constant values for the three levels of kinetic energy, generally worse for the low kinetic energy values, as expected.

The higher kinetic energy levels reach new heights compared to the low kinetic energy levels, going above the original steady-state frequency before decreasing again, indicating a poorly damped response with an overshoot. The previously high levels of kinetic energy had some sustained oscillations, now increased with the 0.5 s inertia constant, and the overall oscillatory response is much higher. The two levels of 1800 and 1025 GVAs still experience oscillations and do not reach steady state until the 20-second simulation window is up. The 250 GVAs kinetic energy levels now exhibit a higher frequency value at the end of the 20-second simulation window but with multiple smaller oscillations, indicating that the fast-acting nature of the system still causes oscillations, indicating poor damping for certain modes. None of the different kinetic energy levels exhibit a well-damped frequency response, indicating general instability for the 0.5 s inertia constant.

Analyzing the eigenvalue results for similar values, seen in Figure 7.73 to 7.75, many modes for all three kinetic energy levels are beneath the 5 % and 3 % markers, with one mode for the 1025 GVAs level having a positive real eigenvalue, indicating a negatively damped mode. This is seen for the mode with a 0.5109 Hz frequency, one of the inter-area modes. A similar pattern is seen with the change in oscillatory speed for the modes related to the 250 GVAs kinetic energy levels, and they cannot be directly compared. The most critical modes of the 250 GVAs kinetic energy levels are also included in the damping ratio comparison for simplicity. As before, the modes related to the grid-following converters remain unchanged for the different kinetic energy values, but the damping of the different inter-area modes varies. Two modes have improved damping for the 1025 GVAs level, while three modes have improved damping for the 1800 GVAs level. Poor damping is also evident for the kinetic energy levels of 250 GVAs, with four modes having damping ratios beneath 5 % and two modes below 3 %. Only the kinetic energy levels of 1025 GVAs have a negatively damped mode. Overall, all three levels of kinetic energy show poorly damped systems for the 0.5 s inertia constant, with one being unstable.

The unstable mode associated with 1025 GVAs, not the higher or lower values, probably results from a bad combination of kinetic energy and inertia constant. The reduced inertia constant causes fast-acting changes in the system, and the kinetic energy levels are not strong enough to provide proper

damping, nor are they small enough to benefit from the fast-acting nature of the grid-forming converters. This results in a poorly damped combination. However, it is worth noting that both the higher and lower kinetic energy values result in almost unstable systems. Thus, the only real conclusion from this analysis is that none of the systems are stable with the current operational characteristics and an inertia constant as low as 0.5 s.

Overall, the analysis shows complex results with respect to changes in the different kinetic energy levels and the corresponding damping and frequency response of the system. Kinetic energy is a major component in determining the maximum initial frequency deviation, which results in a slower initial decrease. However, the lower kinetic energy levels generally result in better system damping. Thus, the different kinetic energy and inertia constant levels are a double-edged sword in their current state. Decreasing the kinetic energy levels results in higher maximum frequency deviations and higher new frequency levels after the disturbance, while the lower kinetic energy levels result in a quicker damped system with well-damped modes. The largest issues arose in the analysis for the low inertia constant value of 0.5 s, resulting in one of the modes with negative damping for one of the kinetic energy levels and very poorly damped oscillatory modes for all three kinetic energy levels, with each having at least one mode below a 1 % damping ratio.

The results suggest that there is no ideal level of kinetic energy and inertia constant, and complex response and damping of these modes are evident with respect to changes in the different values. As these future estimations depend on technological advancements and policies, it is clear that further research is needed to maintain system stability during all future kinetic energy and inertia constants predictions, especially focusing on low inertia constant scenarios. Future research should build upon this study and the studies on grid-forming penetration, controller, and controller parameter analysis to further explore the potential of grid-forming converters in addressing these challenges.

7.4.5. Results and Findings

This study delved into the dynamic interplay between kinetic energy, inertia constants, and system stability within a power system. Examining the frequency response and eigenvalue analysis across varying inertia constants and kinetic energy levels revealed several critical insights.

Firstly, it was evident that higher inertia constants (e.g., 4.1 s and 2.25 s) consistently provide a more stable system response. These higher values enhance the damping of oscillations, ensuring the frequency settles more smoothly after disturbances. Conversely, lower inertia constants (e.g., 0.5 s) lead to a system that reacts more quickly but suffers from increased oscillatory behavior and poorer damping, potentially compromising stability.

Secondly, kinetic energy levels play a pivotal role in determining the system's initial frequency deviation and overall stability. Higher kinetic energy levels (e.g., 1800 GVAs) offer the best performance, minimizing frequency deviations and enhancing system resilience. Medium kinetic energy levels (e.g., 1025 GVAs) show similar patterns but with slightly reduced stability. Lower kinetic energy levels (e.g., 250 GVAs), while settling within acceptable time frames, tend to exhibit more localized oscillations, indicating less stable behavior.

The eigenvalue analysis further corroborates these findings, revealing that systems with higher inertia constants possess significantly better damping ratios for critical modes. Lower inertia constants, particularly when combined with medium kinetic energy levels, often result in modes with critically low or even negative damping ratios, signaling potential instability.

Overall, the study underscores the intricate balance required between kinetic energy and inertia constants to maintain system stability. While higher inertia constants and kinetic energy levels generally foster a more stable environment, they do so at the cost of slower response times. Conversely, lower values promote quicker responses but at the expense of increased oscillatory behavior and potential instability.

Ongoing research is essential to ensure future power systems remain robust. This includes tuning control systems for low inertia scenarios and leveraging technologies like grid-forming converters. These efforts are crucial for achieving a well-balanced, stable power system capable of handling the dynamic demands of future energy landscapes.

In summary, while there is no one-size-fits-all solution, this study highlights the need for a nuanced approach to managing kinetic energy and inertia constants. The stability and resilience of power systems in an evolving energy landscape can be ensured by striking the right balance and continuing to innovate in control strategies and technologies.

8

Conclusion

This thesis work presents extensive research and analysis on future power systems, including scenario analysis, synthetic model development, and case studies. A brief summary of the work, reflections on the research questions, and suggestions for future work are further detailed.

The scenario analysis highlighted deviations and uncertainties, underscoring the importance of understanding various scenarios for future policy guidance. Using the national scenario from the II3050-2 project as a baseline, the study integrated renewable generation capacity, flexibility resources, and demand estimations. The upgraded synthetic model of the future Dutch power system, based on the work of De Roos and Midhuna [72, 105], incorporated new infrastructure, generation, and demand components informed by Dutch TSO 10-year investment plans, governmental plans, and II3050-2 projections. Furthermore, more dynamic characteristics were implemented, and a load flow and contingency analysis to meet N-2 criteria was done. Python scripts and Excel sheets were integrated, enabling semi-automated processing. Conducting analyses in PowerFactory allowed for extensive examinations, while Python and Excel improved efficiency. Three case studies demonstrated the model's capabilities, examining the impact of different shares of renewable generation units, grid-forming converter controller parameters, and future projections of kinetic energy and inertia constants on stability. The findings highlight the model's potential for future research and policy adaptation.

8.1. Reflections of the Research Questions

The research addressed three critical questions concerning the stability of future power systems with high renewable penetration and varying grid-forming converter configurations. Firstly, the analysis revealed that while different renewable generation shares had minimal impact on overall system stability due to robust contingency planning, the absence of ancillary services from certain renewable sources, particularly solar PV, could compromise frequency stability. This highlights the necessity for accurate forecasting and the strategic use of flexibility storage systems to ensure stability despite the variability of renewable resources.

Secondly, the study found that different grid-forming converter types (VSM, Synchroverter, Droop) exhibited similar performance in frequency stability and damping capabilities under most conditions. However, the Synchroverter showed inferior performance during short-circuit events, especially at high penetration levels, indicating a need for further refinement. Higher grid-forming converter penetration generally improved system stability, although it also introduced specific challenges, such as reduced damping in certain local modes. The sensitivity analysis of controller parameters revealed that adjustments in key parameters, such as the damping coefficient, significantly affected system performance. Improved frequency stability and damping were observed with optimized parameter settings, suggesting the potential for dynamic control strategies to enhance stability. This underscores the importance of both the penetration levels and precise tuning of controller parameters in achieving optimal system performance.

Lastly, future projections of inertia levels showed that higher inertia constants and kinetic energy levels enhanced frequency stability but could lead to more oscillatory behavior. Conversely, lower kinetic energy levels improved system damping but posed risks for frequency stability. These findings suggest a complex interplay between inertia and stability, emphasizing the need for adaptive control strategies and further research to optimize future power system configurations.

Overall, the study bridges the gap between theoretical developments and future trends in power systems by building and improving a large-scale synthetic model representation of the future Dutch power system and performing state-of-the-art case studies. This research enhances the understanding of grid-forming converters, focusing on rotor angle and frequency stability, and evaluates system performance through eigenvalue and frequency analysis. The project's emphasis on fast-active power balancing operations and system damping provides valuable insights into future power electronics interfaced (PEI) systems and their impact on system characteristics. These contributions advance scientific knowledge, offering significant value for future research efforts and policy development, and are expected to influence international journals and conferences.

8.2. Future Work

The work done in the thesis opens various areas for future research utilizing the vast potential of the synthetic dynamic model. Leveraging PowerFactory allows for multiple different types of studies and research. Integrated with Python and Excel, the research opportunities are extensive.

Future research should focus on:

- Developing and validating smart control strategies for grid-forming converters.
- Exploring dynamic control settings to optimize stability under varying conditions.
- Further investigating the impact of different kinetic energy and inertia constant levels on stability and control tactics.
- Enhancing the synthetic dynamic model with additional features and scenarios for broader applicability.
- Assessing the impact of different flexibility resources in future scenarios.

References

[1] Frank Kaspar et al. "A climatological assessment of balancing effects and shortfall risks of photovoltaics and wind energy in Germany and Europe". In: *Advances in Science and Research* 16 (July 2019), pp. 119–128. DOI: 10.5194/asr-16-119-2019.

[2] Pieter Tielens and Dirk Van Hertem. "The relevance of inertia in power systems". In: *Renewable and Sustainable Energy Reviews* 55 (2016), pp. 999–1009. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.11.016. URL: https://www.sciencedirect.com/science/article/pii/S136403211501268X.

[3] Jan Shair et al. "Power system stability issues, classifications and research prospects in the context of high-penetration of renewables and power electronics". In: *Renewable and Sustainable Energy Reviews* 145 (2021). DOI: 10.1016/j.rser.2021.111111. URL: https://www.sciencedirect.com/science/article/pii/S1364032121003993.

[4] Subrat Sahoo and Pascal Timmann. "Energy Storage Technologies for Modern Power Systems: A Detailed Analysis of Functionalities, Potentials, and Impacts". In: *IEEE Access* 11 (2023), pp. 49689–49729. DOI: 10.1109/ACCESS.2023.3274504.

[5] Saghar Vahidi et al. "Security of Wide-Area Monitoring, Protection, and Control (WAMPAC) Systems of the Smart Grid: A Survey on Challenges and Opportunities". In: *IEEE Communications Surveys & Tutorials* 25.2 (2023), pp. 1294–1335. DOI: 10.1109/COMST.2023.3251899.

[6] Jose Rodriguez, Frede Blaabjerg, and Marian P. Kazmierkowski. "Energy Transition Technology: The Role of Power Electronics". In: *Proceedings of the IEEE* 111.4 (2023), pp. 329–334. DOI: 10.1109/JPROC.2023.3257421.

[7] Shahid Aziz Khan et al. "Grid-Forming Converters for Stability Issues in Future Power Grids". In: *Energies* 15.14 (2022). ISSN: 1996-1073. DOI: 10.3390/en15144937. URL: https://www.mdpi.com/1996-1073/15/14/4937.

[8] Hesam Pishbahar, Frede Blaabjerg, and Hedayat Saboori. "Emerging grid-forming power converters for renewable energy and storage resources integration – A review". In: *Sustainable Energy Technologies and Assessments* 60 (2023), p. 103538. ISSN: 2213-1388. DOI: https://doi.org/10.1016/j.seta.2023.103538. URL: https://www.sciencedirect.com/science/article/pii/S2213138823005313.

[9] Lasantha Meegahapola et al. "Power System Stability with Power-Electronic Converter Interfaced Renewable Power Generation: Present Issues and Future Trends". In: *Energies* 13.13 (2020), p. 3441. DOI: 10.3390/en13133441. URL: https://doi.org/10.3390/en13133441.

[10] Solomon Feleke et al. "Damping of Frequency and Power System Oscillations with DFIG Wind Turbine and DE Optimization". In: *Sustainability* 15.6 (2023). ISSN: 2071-1050. DOI: 10.3390/su15064751. URL: https://www.mdpi.com/2071-1050/15/6/4751.

[11] José Luis Rodríguez-Amenedo and Santiago Arnaltes Gómez. "Damping Low-Frequency Oscillations in Power Systems Using Grid-Forming Converters". In: *IEEE Access* (Dec. 2021). Received November 5, 2021, accepted November 20, 2021, date of publication November 23, 2021, date of current version December 7, 2021. Supported by the Spanish Research Agency under Project PID2019-106028RB-I00/ AEI/10.13039/501100011033. DOI: 10.1109/ACCESS.2021.3130333.

[12] Francisco González-Longatt and José Rueda. *Stability and Control for Higher Integration of Variable Renewable Power Generation*. Slide presentation in EE4545 Electric Power Systems of the Future. Accessed on [01.12.2023]. 2023.

[13] José Rueda. *Stability and Control for Higher Integration of Variable Renewable Power Generation*. Reader in EE4545 Electric Power Systems of the Future. Accessed on [01.12.2023]. 2023.

[14] Behrooz Bahrani et al. "Grid-Forming Inverter-Based Resource Research Landscape: Understanding the Key Assets for Renewable-Rich Power Systems". In: *IEEE Power and Energy Magazine* 22 (Mar. 2024), pp. 18–29. DOI: 10.1109/MPE.2023.3343338.

[15] Ralf Hesse, Dirk Turschner, and Hans-Peter Beck. "Microgrid stabilization using the Virtual Synchronous Machine (VISMA)". In: *Proceedings of the International Conference on Renewable Energies and Power Quality (ICREPQ'09)*. European Association for the Development of Renewable Energies, Environment and Power Quality. Valencia, Spain, Apr. 2009, pp. 15–17.

[16] Hasan Alrajhi Alsiraji and Ramadan El-Shatshat. "Comprehensive assessment of virtual synchronous machine based voltage source converter controllers". In: *IET Generation, Transmission & Distribution* 11 (2017), pp. 1762–1769. DOI: 10.1049/iet-gtd.2016.1423. URL: https://doi.org/10.1049/iet-gtd.2016.1423.

[17] Md Asif Uddin Khan et al. "Comparative Evaluation of Dynamic Performance of a Virtual Synchronous Machine and Synchronous Machines". In: *The 9th Renewable Power Generation Conference (RPG Dublin Online 2021)*. Vol. 2021. 2021, pp. 366–371. DOI: 10.1049/icp.2021.1362.

[18] Andrew Tuckey and Simon Round. "Practical application of a complete virtual synchronous generator control method for microgrid and grid-edge applications". In: *2018 IEEE 19th Workshop on Control and Modeling for Power Electronics (COMPEL)*. 2018, pp. 1–6. DOI: 10.1109/COMPEL.2018.8459987.

[19] Alisher B. Askarov et al. "A review and comparison of current trends in virtual synchronous generator's models". In: *IFAC-PapersOnLine* 55.9 (2022). 11th IFAC Symposium on Control of Power and Energy Systems CPES 2022, pp. 350–355. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2022.07.061. URL: https://www.sciencedirect.com/science/article/pii/S2405896322004463.

[20] Stephen Sproul, Stanislav Cherevatskiy, and Hugo Klingenberg. *Grid Forming Energy Storage: Provides Virtual Inertia, Interconnects Renewables and Unlocks Revenue*. Tech. rep. © Hitachi ABB Power Grids 2020. All rights reserved. Hitachi ABB Power Grids and ElectraNet, July 2020. URL: https://www.electranet.com.au/wp-content/uploads/2021/01/Grid-Forming-Energy-Storage-Webinar-ESCRI-SA-July-2020.pdf.

[21] S. Sproul et al. "System strength support using grid-forming energy storage to enable high penetrations of inverter-based resources to operate on weak networks". In: *Challenges and Advances in Power System Dynamics, C4 System Technical Performance*. *1 Hitachi Energy, *2 Australian Energy Market Operator, Australia. CIGRE. 2022. URL: https://library.e.abb.com/public/3dfcf71b95334ad78d8e658367e2dbe4/CIGRE%202022_C4%20PS3%20Paper%2010585_GFM%20support%20weak%20grid_S.Sproul,%20N.Modi.pdf?x-sign=b2R5IUqrsR85l5iI0Cg673c255QjXb4U6kAfKNoeuOOpXf+Ot61ICOhPow4W5q4G.

[22] SINTEF Energy Research. *Virtual Synchronous Machines*. Accessed: 2024-05-28. 2024. URL: https://www.sintef.no/en/expertise/sintef-energy-research/virtual-synchronous-machines/.

[23] Praveen Varghese Thomas. *Relation Between Torque and Power*. Accessed: 2 January 2024. 2022. URL: https://www.tutorialspoint.com/relation-between-torque-and-power (visited on 12/21/2022).

[24] *Power and Torque - Essential Concepts*. Accessed: 2 January 2024. 2011. URL: http://www.epi-eng.com/piston_engine_technology/power_and_torque.htm (visited on 03/11/2011).

[25] Wenju Sang et al. "Virtual Synchronous Generator, a Comprehensive Overview". In: *Energies* 15.17 (2022). ISSN: 1996-1073. DOI: 10.3390/en15176148. URL: https://www.mdpi.com/1996-1073/15/17/6148.

[26] Krishnakumar R. Vasudevan et al. "Synchronverter: A Comprehensive Review of Modifications, Stability Assessment, Applications and Future Perspectives". In: *IEEE Access* 8 (2020), pp. 131565–131589. DOI: 10.1109/ACCESS.2020.3010001.

[27] Luis Santiago Azuara-Grande et al. "Real-Time Implementation of Two Grid-Forming Power Converter Controls to Emulate Synchronous Generators". In: *2022 IEEE Biennial Congress of Argentina (ARGENCON)*. 2022, pp. 1–6. DOI: 10.1109/ARGENCON55245.2022.9940076.

[28] Georgios C. Kryonidis et al. "A new perspective on the synchronverter model". In: *International Journal of Electrical Power & Energy Systems* 140 (2022), p. 108072. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2022.108072. URL: https://www.sciencedirect.com/science/article/pii/S0142061522001144.

[29] Zeev Kustanovich et al. "Synchronverters With Fast Current Loops". In: *IEEE Transactions on Industrial Electronics* 70.11 (2023), pp. 11357–11367. DOI: 10.1109/TIE.2022.3229275.

[30] José Dickson Araújo-de-Oliveira et al. "Stability analysis and application of a synchronverter-based control approach to DFIG-based wind energy conversion systems". In: *Electric Power Systems Research* 232 (2024), p. 110403. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2024.110403. URL: https://www.sciencedirect.com/science/article/pii/S0378779624002918.

[31] Muamar Mohamed et al. "Combination of the Synchronverter with Droop Control for Hybrid OFF-grid System". In: *2022 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*. 2022, pp. 1–4. DOI: 10.1109/PEDES56012.2022.10080562.

[32] Krishnakumar R. Vasudevan et al. "Synergizing pico hydel and battery energy storage with adaptive synchronverter control for frequency regulation of autonomous microgrids". In: *Applied Energy* 325 (2022), p. 119827. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2022.119827. URL: https://www.sciencedirect.com/science/article/pii/S0306261922010984.

[33] P. Piagi and R.H. Lasseter. "Autonomous control of microgrids". In: *2006 IEEE Power Engineering Society General Meeting*. 2006. DOI: 10.1109/PES.2006.1708993.

[34] J. Grainger and W. Stevenson. *Power System Analysis*. New York, NY, USA: McGraw-Hill, 1994.

[35] Sana Fazal et al. "Droop Control Techniques for Grid Forming Inverter". In: *2022 IEEE PES 14th Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. 2022, pp. 1–6. DOI: 10.1109/APPEEC53445.2022.10072251.

[36] Anshul Awasthi, Souvik Bera, and Ragavan K. "A Modified Dual Loop Droop Control for Parallel Operation of Grid-forming Converters". In: *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*. 2023, pp. 1–6. DOI: 10.1109/I2CT57861.2023.10126279.

[37] Hui Hwang Goh et al. "Comparative assessment of single-loop droop controlled grid-forming converter and its damping enhancement". In: *Energy Reports* 9 (2023). Selected papers from 2022 International Conference on Frontiers of Energy and Environment Engineering, pp. 1048–1056. ISSN: 2352-4847. DOI: https://doi.org/10.1016/j.egyr.2023.04.108. URL: https://www.sciencedirect.com/science/article/pii/S2352484723004729.

[38] Khaliqur Rahman et al. "Reviewing Control Paradigms and Emerging Trends of Grid-Forming Inverters—A Comparative Study". In: *Energies* 17.10 (2024). ISSN: 1996-1073. DOI: 10.3390/en17102400. URL: https://www.mdpi.com/1996-1073/17/10/2400.

[39] Federico Ibanez et al. "Improving the power sharing transients in droop-controlled inverters with the introduction of an angle difference limiter". In: *International Journal of Electrical Power & Energy Systems* 153 (2023), p. 109371. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2023.109371. URL: https://www.sciencedirect.com/science/article/pii/S0142061523004283.

[40] Sana Fazal et al. "Droop Control Techniques for Grid Forming Inverter". In: Nov. 2022, pp. 1–6. DOI: 10.1109/APPEEC53445.2022.10072251.

[41] Walid Issa, Sulieman Sharkh, and Mohammad Abusara. "A review of recent control techniques of drooped inverter-based AC microgrids". In: *Energy Science & Engineering* 12.1 (Jan. 2024), pp. 1–25. DOI: 10.1002/ese3.1670. URL: https://doi.org/10.1002/ese3.1670.

[42] Dayan B. Rathnayake et al. "Grid Forming Inverter Modeling, Control, and Applications". In: *IEEE Access* 9 (2021), pp. 114781–114807. DOI: 10.1109/ACCESS.2021.3104617.

[43] J. Bialek et al. *System needs and services for systems with High IBR penetration*. Tech. rep. G-PST Tech. Workstream Rep. The Global Power System Transformation Consortium, 2021. URL: https://globalpst.org/wp-content/uploads/GPST-IBR-Research-Team-System-Services-and-Needs-for-High-IBR-Networks.pdf.

[44] Musa Khan, Wenchuan Wu, and Li Li. "Grid-forming control for inverter-based resources in power systems: A review on its operation, system stability, and prospective". In: *IET Renewable Power Generation* (Apr. 2024). DOI: 10.1049/rpg2.12991. URL: https://doi.org/10.1049/rpg2.12991.

[45] Isao Iyoda and Jean Belanger. *History of Power System Simulators to Analyze and Test of Power Electronics Equipment*. Tech. rep. 2017, pp. 117–120. DOI: 10.1109/HISTELCON.2017.8535780.

[46] B. Badrzadeh and Z. Emin. "The need for enhanced power system modelling techniques and simulation tools". In: C4.56 Convenor (AU) and SC C4 Chair (GB). Feb. 2020.

[47] John Pierce. *Power System Engineering and Software*. 2017. URL: https://www.aemc.gov.au/sites/default/files/content/d7a7ae4d-d792-4b5b-ad73-1d32aa0b34a7/RuleChange-Submission-ERC0219-DIgSILENT-Pacific-170801.PDF.

[48] Scott Hagamann. *What Future for RMS Simulation?* 2019. URL: https://digsilent.com.au/en/publications.html?file=files/publications/2020/slides/What%20future%20for%20RMS%20simulation.pdf%5C&cid=13840.

[49] S. Subedi et al. "Review of Methods to Accelerate Electromagnetic Transient Simulation of Power Systems". In: *IEEE Access* 9 (2021), pp. 89714–89731. DOI: 10.1109/ACCESS.2021.3090320.

[50] H. Holttinen et al. "System Impact Studies for Near 100 % Renewable Energy Systems Dominated by Inverter Based Variable Generation". In: *IEEE Transactions on Power Systems* 37.4 (2022), pp. 3249–3258. DOI: 10.1109/TPWRS.2020.3034924.

[51] *Co-Simulation*. https://smarter.energynetworks.org/projects/nia2_nget0020/. Status: Live. Project Reference Number: NIA2_NGET0020. Strategy Theme: Net zero and the energy system transition. Funding Mechanism: NIA_RIIO-2. Technology: Various. Expenditure: £300,000. July 2023–2025.

[52] Bernd Schweinshaut et al. "Transformation, Comparison and Co-Simulation of a RMS-EMT Multi-Domain Test Network". In: *NEIS 2022; Conference on Sustainable Energy Supply and Energy Storage Systems*. 2022, pp. 1–6.

[53] Christian Scheibe et al. "Real Time Co-Simulation of Electromechanical and Electromagnetic Power System Models". In: *2022 International Conference on Smart Energy Systems and Technologies (SEST)*. 2022, pp. 1–6. DOI: 10.1109/SEST53650.2022.9898468.

[54] D. Koolen, M. De Felice, and S. Busch. *Flexibility requirements and the role of storage in future European power systems*. Tech. rep. European Commission, Joint Research Centre, 2023. DOI: 10.2760/384443. URL: https://data.europa.eu/doi/10.2760/384443.

[55] D. Flynn et al. "Technical impacts of high penetration levels of wind power on power system stability". In: *WIREs Energy and Environment* (2016). DOI: 10.1002/wene.216. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wene.216.

[56] Sumit. "Forecasting Issues in Present Day Power Systems". PhD Synopsis. Mullana, Haryana: Maharishi Markandeshwar Engineering College, Maharishi Markandeshwar University, 2014.

[57] Statnett. *Measures to Ensure Power Supply This Winter*. https://www.statnett.no/en/about-statnett/news-and-press-releases/news-archive-2022/measures-to-ensure-power-supply-this-winter/. Accessed: 29.11.2023. Sept. 2022.

[58] Wen-Chang Tsai et al. "A Review of Modern Wind Power Generation Forecasting Technologies". In: *Sustainability* 15.14 (2023). ISSN: 2071-1050. DOI: 10.3390/su151410757. URL: https://www.mdpi.com/2071-1050/15/14/10757.

[59] ENTSO-E. *Enhanced RES Infeed Forecasting - Wind*. https://www.entsoe.eu/Technopedia/techsheets/enhanced-res-infeed-forecasting-wind. Accessed: 2023-11-09. 2023.

[60] Mohammad Abdul Baseer et al. "Electrical Power Generation Forecasting from Renewable Energy Systems Using Artificial Intelligence Techniques". In: *Energies* 16.18 (2023). ISSN: 1996-1073. DOI: 10.3390/en16186414. URL: https://www.mdpi.com/1996-1073/16/18/6414.

[61] Freddie Harvey. "Artificial Intelligence in Power System Operation and Optimization: Current Trends and Future Directions". In: *Journal of Electrical & Electronic Systems* 12 (1 2023). Delft University of Technology, Delft, The Netherlands. URL: https://www.hilarispublisher.com/open-access/artificial-intelligence-in-power-system-operation-and-optimization-current-trends-and-future-directions-99142.html#:~:text=Artificial%20Intelligence%20is%20revolutionizing%20power,scheduling%2C%20and%20grid%20stability%20analysis..

[62] B. Rajanarayan Prusty et al. "Chapter Ten - Machine learning application to power system forecasting". In: *Smart Electrical and Mechanical Systems*. Ed. by Rakesh Sehgal et al. Academic Press, 2022, pp. 225–236. ISBN: 978-0-323-90789-7. DOI: https://doi.org/10.1016/B978-0-323-90789-7.00013-0. URL: https://www.sciencedirect.com/science/article/pii/B9780323907897000130.

[63] Gang Chen et al. "Machine-Learning-Based Electric Power Forecasting". In: *Sustainability* 15.14 (2023). ISSN: 2071-1050. DOI: 10.3390/su151411299. URL: https://www.mdpi.com/2071-1050/15/14/11299.

[64] Natei Ermias Benti, Mesfin Diro Chaka, and Addisu Gezahegn Semie. "Forecasting Renewable Energy Generation with Machine Learning and Deep Learning: Current Advances and Future Prospects". In: *Sustainability* 15.9 (2023). ISSN: 2071-1050. DOI: 10.3390/su15097087. URL: https://www.mdpi.com/2071-1050/15/9/7087.

[65] D A Widodo et al. "Renewable energy power generation forecasting using deep learning method". In: *IOP Conference Series: Earth and Environmental Science* 700.1 (2021), p. 012026. DOI: 10.1088/1755-1315/700/1/012026. URL: https://dx.doi.org/10.1088/1755-1315/700/1/012026.

[66] D Widodo et al. "Renewable energy power generation forecasting using deep learning method". In: *IOP Conference Series: Earth and Environmental Science* 700 (Mar. 2021), p. 012026. DOI: 10.1088/1755-1315/700/1/012026.

[67] Imdadullah et al. "Electric Power Network Interconnection: A Review on Current Status, Future Prospects and Research Direction". In: *Electronics* 10.17 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10172179. URL: https://www.mdpi.com/2079-9292/10/17/2179.

[68] Ahmad Alzakkar et al. "The impact of electrical interconnection between countries on the stability of electrical power systems". In: *2022 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*. 2022, pp. 1–6. DOI: 10.1109/REEPE53907.2022.9731442.

[69] Jonas Skaare Amundsen et al. *Kraftmarkedsanalyse 2017 - 2030: Underlagsrapport med detaljerte forutsetninger*. Tech. rep. Middelthunsgate 29, 0301 OSLO: Norges vassdrags- og energidirektorat, 2017. URL: https://publikasjoner.nve.no/rapport/2017/rapport2017_78.pdf.

[70] Nordic Energy Research. *The Nordic Energy Trilemma: Security of Supply, Prices, and the Just Transition*. Tech. rep. Nordic Energy Research, 2023. URL: https://pub.norden.org/nordicenergyresearch2023-02/nordicenergyresearch2023-02.pdf.

[71] Matthew Wittenstein, Jesse Scott, and Noor Miza Muhamad Razali. *Electricity Security Across Borders: Case Studies on Cross-Border Electricity Security in Europe*. Tech. rep. International Energy Agency, 2016. URL: https://iea.blob.core.windows.net/assets/a015167b-867d-4ff9-bd57-e53d0f52c17b/ElectricitySecurityAcrossBorders.pdf.

[72] M. Garapati. "Synthetic Digital Model for Stability Studies in the Future Dutch Power System". MSc thesis dissertation. Delft University of Technology, June 2024.

[73] J. L. Rueda Torres and Rohan Kamat Tarcar. *Lab Session 1: PowerFactory Tutorial for T4113 Power System Dynamics*. Brightspace TU Delft. Accessed: 2024-01-12. May 2023. URL: https://brightspace.tudelft.nl/d2l/le/content/499919/viewContent/2926716/View.

[74] DIgSILENT GmbH. *PowerFactory 2023 User Manual*. Version 2023, Online Edition. Heinrich-Hertz-Straße 9, 72810 Gomaringen, Germany: DIgSILENT GmbH, 2023.

[75] DIgSILENT GmbH. *PowerFactory 2023 Technical Reference: DIgSILENT Grid-forming Converter Templates. Droop Controlled Converter, Synchronverter, Virtual Synchronous Machine*. Heinrich-Hertz-Straße 9, 72810 Gomaringen, Germany: DIgSILENT GmbH, 2023.

[76] Sander Skogen and Jose Rueda Torres. "Fundamental Assessment of Oscillatory Performance of Grid-Forming Dominated Systems". In press. Delft, The Netherlands, 2023.

[77] Nicolae Lobontiu. "Chapter 11 - Block Diagrams and Feedback Control System Modeling". In: *System Dynamics for Engineering Students (Second Edition)*. Ed. by Nicolae Lobontiu. Second Edition. Boston: Academic Press, 2018, pp. 541–592. ISBN: 978-0-12-804559-6. DOI: https://doi.org/10.1016/B978-0-12-804559-6.00011-7. URL: https://www.sciencedirect.com/science/article/pii/B9780128045596000117.

[78] P. Kundur J. Neal N. Balu and M. G. Lauby. *Power System Stability and Control*. New York, NY, USA: McGraw-Hill, 1994.

[79] Nikos Hatziargyriou et al. "Definition and Classification of Power System Stability Revisited & Extended". In: *Power Systems, IEEE Transactions on* 36 (July 2021), pp. 3271–3281. DOI: 10.1109/TPWRS.2020.3041774.

[80] Prabha Kundur et al. "Definition and Classification of Power System Stability IEEE/CIGRE Joint Task Force on Stability Terms and Definitions". In: *Power Systems, IEEE Transactions on* 19 (Sept. 2004), pp. 1387–1401. DOI: 10.1109/TPWRS.2004.825981.

[81] Nikos Hatziargyriou et al. *Stability Definitions and Characterization of Dynamic Behavior in Systems with High Penetration of Power Electronic Interfaced Technologies*. Technical Report. IEEE Power & Energy Society, May 2020.

[82] Dr.ir. J.L. (Jose) Rueda Torres. *Lecture 6 - Rotor angle stability*. ET4113 - Power System Dynamics (course). Accessed: 28.10.2023. 2023. URL: https://brightspace.tudelft.nl/d2l/le/content/499919/viewContent/2926627/View.

[83] O. Ruhle. *Eigenvalue Analysis – All Information on Power System Oscillation Behavior Rapidly Analyzed*. 5 p. 2006.

[84] Jani Stenroos. "Evaluation of Criteria for Transmission Capacity Calculation". Master of Science Thesis. Faculty of Information Technology and Communication Sciences, Oct. 2019.

[85] Kjetil Uhlen et al. *Application of linear analysis for stability improvements in the Nordic power transmission system*. Aug. 2003. DOI: 10.1109/PES.2003.1270938.

[86] Durga Gautam, Vijay Vittal, and Terry Harbour. "Impact of Increased Penetration of DFIG-Based Wind Turbine Generators on Transient and Small Signal Stability of Power Systems". In: *IEEE Transactions on Power Systems* 24.3 (2009), pp. 1426–1434. DOI: 10.1109/TPWRS.2009.2021234.

[87] G. Tsourakis, B. M. Nomikos, and C. D. Vournas. "Effect of wind parks with doubly fed asynchronous generators on small-signal stability". In: *Electr. Power Syst. Res.* 79.1 (2009), pp. 190–200.

[88] Slobodan N. Vukosavic. *Grid-Side Converters Control and Design: Interfacing Between the AC Grid and Renewable Power Sources*. Belgrade, Serbia: Springer, 2018.

[89] M. J. Gibbard, P. Pourbeik, and D. J. Vowles. *Small-Signal Stability, Control and Dynamic Performance of Power Systems*. [Online]. Available: https://www.adelaide.edu.au/press/titles/small-signal. Adelaide, Australia: Univ. Adelaide Press, 2015.

[90] ENTSO-E. *Project Inertia – Phase II: Updated Frequency Stability Analysis in Long Term Scenarios, Relevant Solutions and Mitigation Measures*. Tech. rep. For Publication. Brussels: European Network of Transmission System Operators for Electricity (ENTSO-E), Nov. 2023. URL: https://www.entsoe.eu.

[91] European Network of Transmission System Operators (ENTSOE) for Electricity. *Network Code for Requirements for Grid Connection Applicable to All Generators: Requirements in the Context of Present Practices*. Tech. rep. European Network of Transmission System Operators for Electricity, 2012.

[92] ENTSO-E. *Rate of Change of Frequency (RoCoF) withstand capability ENTSO-E guidance document for national implementation for network codes on grid connection*. Guidance document. Nov. 2017. URL: https://www.entsoe.eu.

[93] Gert Rietveld et al. "Evaluation report on the problem of ROCOF measurement in the context of actual use cases and the "wish list" of accuracy and latency from an end-user point of view". In: *Pre-normative research project "ROCOF"* (2017). Funded by the EMPIR programme co-financed by the Participating States and the European Union's Horizon 2020 research and innovation programme.

[94] Francisco Gonzalez-Longatt et al. "Investigation of Inertia Response and Rate of Change of Frequency in Low Rotational Inertial Scenario of Synchronous Dominated System". In: *Electronics* 10.18 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10182288. URL: https://www.mdpi.com/2079-9292/10/18/2288.

[95] Gert Rietveld et al. "15NRM04 ROCOF: Evaluation report on the problem of ROCOF measurement in the context of actual use cases and the "wish list" of accuracy and latency from an end-user point of view". In: *Pre-normative research project "ROCOF"* (2017). Funded by the EMPIR programme co-financed by the Participating States and the European Union's Horizon 2020 research and innovation programme. URL: https://www.empir.eu.

[96] RG-CE System Protection & Dynamics Sub Group. *Frequency Stability Evaluation Criteria for the Synchronous Zone of Continental Europe – Requirements and Impacting Factors*. Tech. rep. Technical Report. Regional Group Continental Europe, Mar. 2016. URL: https://www.entsoe.eu.

[97] ENTSO-E RDI Committee Thematic WG 2. *Secure operations of tomorrow Internal Workshop on Fast Control Reserves for Future Frequency Control*. Vienna: APG & ENTSO-E – RDIC, Oct. 2019.

[98] ENTSO-E StG CNC. *Maximum Admissible Active Power Reduction at Low Frequencies*. Guidance Document. Revised Clean Version. ENTSO-E, Apr. 2021. URL: https://www.entsoe.eu.

[99] Jingyang Fang et al. "The Role of Power Electronics in Future Low Inertia Power Systems". In: *2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)*. 2018, pp. 1–6. DOI: 10.1109/PEAC.2018.8590632.

[100] Zhimin Gao et al. "Virtual Inertia Control Strategy for Photovoltaic Power Generation Based on Improved Incremental Conductance Method". In: *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*. 2022, pp. 1265–1269. DOI: 10.1109/ACPEE53904.2022.9784080.

[101] Aref Pouryekta and Vigna K. Ramachandaramurthy. *Virtual Inertia Control to Enhance Frequency Stability of Power Systems*. Accessed: 16.11.2023. 2019. URL: https://smartgrid.ieee.org/bulletins/september-2019/virtual-inertia-control-to-enhance-frequency-stability-of-power-systems.

[102] Wenyuan Wang et al. "Analysis of Active Power Control for VSC–HVDC". In: *IEEE Transactions on Power Delivery* 29.4 (2014), pp. 1978–1988. DOI: 10.1109/TPWRD.2014.2322498.

[103] N. W. Miller et al. "Frequency response of California and WECC under high wind and solar conditions". In: *2012 IEEE Power and Energy Society General Meeting*. 2012, pp. 1–8. DOI: 10.1109/PESGM.2012.6345564.

[104] Ismail Kasikci. *Short Circuits in Power Systems: A Practical Guide to IEC 60909-0*. Weinheim: Wiley-VCH, 2017.

[105] Anouk de Roos. "Synthetic Steady-State Model of the Dutch EHV Network: Study of the impact of future additions of VRES and electrolysers". Master's thesis. Delft, The Netherlands: Delft University of Technology, 2021.

[106] Tennet. *Integrale Infrastructuurverkenning 2030-2050*. https://www.tennet.eu/nl/over-tennet/publicaties/integrale-infrastructuurverkenning-2030-2050. Main report, Updated October 2023. 2023. (Visited on 01/18/2024).

[107] Tennet. *II3050 Bijlagen*. https://www.tennet.eu/nl/over-tennet/publicaties/integrale-infrastructuurverkenning-2030-2050. Appendix of the main report, Updated October 2023 (Updated). 2023. (Visited on 01/18/2024).

[108] Netbeheer Nederland. *Het Energiesysteem van de Toekomst Integrale Infrastructuurverkenning 2030-2050*. 2023. URL: %5Curl%7Bhttps://www.netbeheernederland.nl/_upload/Files/Rapport_II3050_Scenario's_280.pdf%7D.

[109] Netbeheer Netherland. *Summary: The Energy System of the Future 2030-2050 Integral Infrastructure Survey*. https://www.tennet.eu/nl/over-tennet/publicaties/summary-the-energy-system-of-the-future-2030-2050. Based on the reports "Integrale Infrastructuurverkenning 2030-2050" and "II3050 Bijlagen". 2021. (Visited on 01/18/2024).

[110] Gasunie and Tennet. *Infrastructure Outlook 2050: A joint study by Gasunie and TenneT on integrated energy infrastructure in the Netherlands and Germany*. https://www.gasunie.nl/en/expertise/energy-system/infrastructure-outlook-2050. 2019. (Visited on 01/18/2024).

[111] Gasunie and Tennet. *Phase II - Pathways to 2050: A joint follow-up study by Gasunie and TenneT of the Infrastructure Outlook 2050*. https://www.gasunie.de/en/news/gasunie-and-tennet-climate-goals-can-only-be-achieved-with-an-integrated-european-energy-system. Extension of the report "Infrastructure Outlook 2050". 2020. (Visited on 01/18/2024).

[112] CE Delft. *Net voor de Toekomst*. 2017. URL: https://www.ce.nl/publicaties/2030/netvoor-de-toekomst.

[113] Berenschot. *Nederland klimaatneutraal in 2050: vier scenario's*. 2020. URL: https://www.berenschot.nl/nieuws/2020-april-nederland-klimaatneutraal-in-2050.

[114] TenneT. *Target Grid*. Project description on the official TenneT website. Accessed: 2024-01-25. 2024. URL: https://www.tennet.eu/target-grid.

[115] TenneT. *Ontwerpinvesteringsplan Net op land 2024-2033*. https://tennet-drupal.s3.eu-central-1.amazonaws.com/default/2024-01/IP2024_Netopland_01-1-2024_0.pdf. Accessed: 2024-01-25. 2024.

[116] TenneT. *Ontwerpinvesteringsplan Net op zee 2024-2033*. https://tennet-drupal.s3.eu-central-1.amazonaws.com/default/2024-01/IP2024_Netopzee_01-1-2024.pdf. Accessed: 2024-01-25. 2024.

[117] TenneT. *Netversterking Noordoost Nederland*. https://www.tennet.eu/nl/projecten/provincies/drenthe/netversterking-noordoost-nederland. Accessed: 2024-02-29. 2024.

[118] *Hoogspanningsstation Amaliahaven*. https://www.tennet.eu/nl/projecten/provincies/zuid-holland/hoogspanningsstation-amaliahaven. Accessed: 2024-02-29. 2024.

[119] TenneT. *Offshore Projecten Nederland*. https://www.tennet.eu/nl/projecten/offshore-projecten-nederland. Accessed: 2024-02-26. 2024.

[120] Provincie Groningen. *Provinciaal Meerjarenprogramma Infrastructuur Energie en Klimaat*. https://www.ipo.nl/media/h10led5b/provinciaal-miek-groningen_def.pdf. Accessed: 2024-02-29. June 2023.

[121] *Zeeuws-Vlaanderen 380kV*. https://ten.projectatlas.app/zeeuws_vlaanderen_380kv/page/home?map=51.405324,3.830767,11.01,0,0. Accessed: 2024-02-29.

[122] *MIEK Overzicht 2022: Meerjarenprogramma Infrastructuur Energie en Klimaat*. https://open.overheid.nl/documenten/ronl-7b2edb5f512b6885ff9cd12e5cc5c2fd9bd1b12c/pdf. Accessed: 2024-02-29.

[123] *Aanleg 380 kV verbinding Zeeuws-Vlaanderen*. https://www.tennet.eu/nl/projecten/aanleg-380-kv-verbinding-zeeuws-vlaanderen. Accessed: 2024-02-29. TenneT, 2024.

[124] TenneT. *Bouw Hoogspanningsstation omgeving Sloegebied*. https://www.tennet.eu/nl/projecten/bouw-hoogspanningsstation-omgeving-sloegebied. Accessed: 2024-02-29.

[125] Rijksdienst voor Ondernemend Nederland (RVO). *Hoogspanningsstation omgeving Sloegebied*. https://www.rvo.nl/onderwerpen/bureau-energieprojecten/lopende-projecten/hss-omgeving-sloegebied. Accessed: 2024-02-29.

[126] Commissie voor de milieueffectrapportage. *Hoogspanningsstation omgeving Sloegebied: Advies over reikwijdte en detailniveau van het milieueffectrapport*. https://www.rvo.nl/sites/default/files/2023-05/Advies-Commissie-mer.pdf. Accessed: 2024-02-29. Mar. 2023.

[127] *Station Halsteren*. https://www.tennet.eu/nl/projecten/provincies/noord-brabant/station-halsteren. Accessed: 2024-02-29. TenneT, 2024.

[128] *Netversterking Schouwen-Duiveland, Tholen en omgeving Bergen op Zoom*. https://www.tennet.eu/nl/projecten/netversterking-schouwen-duiveland-tholen-en-omgeving-bergen-op-zoom. Accessed: 2024-02-29. TenneT, 2024.

[129] *Schouwen-Duiveland Tracékaart Straks*. https://ten.projectatlas.app/schouwen-duiveland/tracekaart/straks?map=51.537315,4.256263,13,0,0&textfield=false. Accessed: 2024-02-29. 2024.

[130] *Port Moerdijk 380/150/20kV*. https://www.tennet.eu/nl/projecten/provincies/noord-brabant/port-moerdijk-38015020kv. Accessed: 2024-02-29. TenneT, 2024.

[131] *POM 380/150/20 kV*. https://www.rvo.nl/onderwerpen/bureau-energieprojecten/lopende-projecten/pom-380-150-20-kv. Accessed: 2024-02-29. Rijksdienst voor Ondernemend Nederland (RVO), 2024.

[132] Port of Rotterdam. *Verzwaring Elektriciteitsnet Rotterdamse Haven Kan Efficiënter En Goedkoper Door Aanpak Met Langetermijnfocus*. Port of Rotterdam. Available online: https://www.portofrotterdam.com/sites/default/files/2021-05/verzwaring-elektriciteitsnet-haven-rotterdam.pdf. Sept. 2019.

[133] Adviescombinatie Tauw/Witteveen en Bos. *Notitie reikwijdte en detailniveau 380 kV-hoogspanningsverbinding Diemen-Ens*. Technical Report. Feb. 2024.

[134] Rijksdienst voor Ondernemend Nederland (RVO). *Jaaroverzicht Energie-Infrastructuurprojecten 2023*. https://www.rvo.nl/sites/default/files/2024-02/Jaaroverzicht%20energie-infrastructuurprojecten%202023.pdf. Accessed: 2024-02. Feb. 2024.

[135] Netherlands Enterprise Agency (RVO). *Offshore Wind Energy Plans 2030-2050*. https://english.rvo.nl/topics/offshore-wind-energy/plans-2030-2050. Accessed: 2024-03-06. Netherlands Enterprise Agency, 2021.

[136] TenneT. *The 2GW Program*. https://www.tennet.eu/about-tennet/innovations/2gw-program. Accessed: 2024-03-06. TenneT, 2023.

[137] National Grid. *Onshore vs Offshore Wind Energy: What's the Difference?* 2022. URL: https://www.nationalgrid.com/stories/energy-explained/onshore-vs-offshore-wind-energy (visited on 03/08/2024).

[138] Statista. *Largest Cities in the Netherlands by Number of Inhabitants*. https://www.statista.com/statistics/993709/largest-cities-in-the-netherlands-by-number-of-inhabitants/. Accessed: insert-date-here. 2023.

[139] *PowerFactory 2023 Technical Reference*. PV System - ElmPvsys, TypPvpanel. DIgSILENT. 2023.

[140] N.L. Char and Bela J. Csik. "Nuclear power development: History and outlook". In: *Bulletin* 29.3 (Sept. 1987).

[141] OECD Nuclear Energy Agency. *Public Attitudes to Nuclear Power*. NEA No. 6859. OECD, 2010. ISBN: 978-92-64-99111-8.

[142] *Nuclear Energy*. https://www.government.nl/topics/renewable-energy/nuclear-energy. Accessed: 16 March 2024.

[143] TenneT. *TenneT's Position on Battery Energy Storage Systems (BESS)*. Available online. C1 - Public Information. June 2023. URL: https://tennet-drupal.s3.eu-central-1.amazonaws.com/default/2023-06/TenneT_s_position_large_BESS_-_Public_Info_-_update.pdf.

[144] *PowerFactory 2023 Technical Reference: WECC Battery Energy Storage System Templates*. Templates for modeling and simulation of battery energy storage systems in PowerFactory software. DIgSILENT. 2023.

[145] Feras Alshehri et al. "Generic Modelling of PEM Technologies for Power System Stability Studies Based on PowerFactory". English. In: *Power Systems*. Power Systems. Springer, 2021, pp. 269–283. DOI: 10.1007/978-3-030-54124-8_12.

[146] Feras A. Alshehri. "Ancillary services from Hydrogen Based Technologies to Support Power System Frequency Stability". This thesis is confidential and cannot be made public until March 31, 2020. Master's thesis. Delft, The Netherlands: Delft University of Technology, Sept. 2018.

[147] Edgar Diego Gomez Anccas, Johannes Blanz, and Detlef Schulz. "Grid-forming fuel cell system for a multi-energy-microgrid in islanding operation". In: *PESS + PELSS 2022; Power and Energy Student Summit*. 2022, pp. 1–6.

[148] Kumaraguru Prabakar et al. *Enhancing Distribution System Resiliency Using Grid-Forming Fuel Cell Inverter: Preprint*. Golden, CO: National Renewable Energy Laboratory. 2022. URL: https://www.nrel.gov/docs/fy22osti/82111.pdf.

[149] *Zero Emission Hydrogen Turbine Center*. https://www.siemens-energy.com/global/en/home/products-services/solutions-usecase/hydrogen/zehtc.html. Accessed: 2024-03-10. 2024.

[150] Sonal Patel. "RWE to Take Over Vattenfall's 1.4-GW 'Hydrogen-Ready' Dutch Gas-Fired Plant". In: *POWER Magazine* (July 2022). Accessed: 2024-03-10. URL: https://www.powermag.com/rwe-to-take-over-vattenfalls-1-4-gw-hydrogen-ready-dutch-gas-fired-plant/.

[151] "World's largest hydrogen fuel cell power plant". In: *Fuel Cells Bulletin* 2021.11 (2021), pp. 5–6. ISSN: 1464-2859. DOI: https://doi.org/10.1016/S1464-2859(21)00602-7. URL: https://www.sciencedirect.com/science/article/pii/S1464285921006027.

[152] Muntathir Al Talaq. "Frequency Stability Enhancement for a System with Integrated Renewable Energy Sources and HVDC Link". In: *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*. 2020, pp. 391–395. DOI: 10.1109/iSemantic50169.2020.9234218.

[153] Tennet. *Tennet Delivers 2023 Results: Record High EUR 7.7 Billion Grid Investments for Coping with the Grid*. Accessed: 2024-03-14. 2023. URL: https://www.tennet.eu/news/tennet-delivers-2023-results-record-high-eur-77-billion-grid-investments-coping-grid.

[154] Tennet and National Grid Ventures. *LionLink: A Vital Step Towards an Integrated North Sea Grid*. Web Page. Accessed: 2024-03-14. 2023. URL: https://www.tennet.eu/lionlink.

[155] Ngoc-Tuan Trinh et al. "Generic Model of MMC-VSC-HVDC for Interaction Study With AC Power System". In: *IEEE Transactions on Power Systems* 31.1 (2016), pp. 27–34. DOI: 10.1109/TPWRS.2015.2390416.

[156] Francisco Gonzalez-Longatt and José Rueda. *PowerFactory Applications for Power System Analysis*. Jan. 2015. ISBN: ISBN 978-3-319-12958-7. DOI: 10.1007/978-3-319-12958-7.

[157] Andrei Stan, Sorina Costinaș, and Georgiana Ion. "Overview and Assessment of HVDC Current Applications and Future Trends". In: *Energies* 15.3 (2022). ISSN: 1996-1073. DOI: 10.3390/en15031193. URL: https://www.mdpi.com/1996-1073/15/3/1193.

[158] European Network of Transmission System Operators for Electricity (ENTSO-E). *TYNDP 2022 System Needs Study: System Dynamic and Operational Challenges*. Tech. rep. Final Version. European Network of Transmission System Operators for Electricity (ENTSO-E), May 2023. URL: https://eepublicdownloads.blob.core.windows.net/public-cdn-container/tyndp-documents/TYNDP2022/public/syst-dynamic-operational-challenges.pdf.

[159] European Network of Transmission System Operators for Electricity (ENTSO-E). *Inertia and Rate of Change of Frequency (RoCoF) Version 17 SPD – Inertia TF*. System Protection Document. Version 17. European Network of Transmission System Operators for Electricity (ENTSO-E), Dec. 2020.

[160] Komal S. Shetye et al. "Considerations for Interconnection of Large Power Grid Networks". In: *2021 IEEE Power and Energy Conference at Illinois (PECI)*. 2021, pp. 1–8. DOI: 10.1109/PECI51586.2021.9435208.

[161] A.A.M. Zin, H.M. Hafiz, and W.K. Wong. "Static and dynamic under-frequency load shedding: a comparison". In: *2004 International Conference on Power System Technology, 2004. PowerCon 2004.* Vol. 1. 2004, 941–945 Vol.1. DOI: 10.1109/ICPST.2004.1460129.

[162] The Staff of the South African Reserve Bank. *South African Reserve Bank Occasional Bulletin of Economic Notes*. Economic Notes. OBEN/23/01. South African Reserve Bank, June 2023.

[163] Calvin Naidoo. "The Impact of Load Shedding on the South Africa Economy". In: *Journal Title* 58.1 (Mar. 2023). URL: https://hdl.handle.net/10520/ejc-jpad_v58_n1_a2.

[164] Kevin W. Jones, Katriana Webber, and Krithika Bhuvaneshwaran. "The Need for Faster Underfrequency Load Shedding". In: *Proceedings of the 2021 Conference on Underfrequency Load Shedding*. TP7006-01. Xcel Energy and Schweitzer Engineering Laboratories, Inc. Mar. 2021. URL: https://prorelay.tamu.edu/wp-content/uploads/sites/3/2022/02/TheNeedFaster_7006_20210310.pdf.

[165] DIGSILENT. *How can I setup an Under-Frequency Load Shedding (UFLS) scheme?* Accessed on: yyyy-mm-dd. 2023. URL: https://www.digsilent.de/index.php/en/faq-reader-powerfactory/how-can-i-setup-an-under-frequency-load-shedding-ufls-scheme.html.

[166] Zhengguang Liu et al. "Power to heat: Opportunity of flexibility services provided by building energy systems". In: *Advances in Applied Energy* 11 (2023), p. 100149. ISSN: 2666-7924. DOI: https://doi.org/10.1016/j.adapen.2023.100149. URL: https://www.sciencedirect.com/science/article/pii/S2666792423000288.

[167] Emma Koster et al. *The Natural Gas Phase-Out in the Netherlands*. Briefing. Publication code: 22.210381.016. CE Delft, Feb. 2022.

[168] Callum Henderson et al. "Inertia and Frequency Support From Britain's AC Powered Trains". In: *IEEE Transactions on Sustainable Energy* 14.2 (2023), pp. 1259–1268. DOI: 10.1109/TSTE.2022.3221192.

[169] Sebastiaan Hers et al. *Potential for Power-to-Heat in the Netherlands*. Tech. rep. Publication code: 15.3E04.65. CE Delft, Aug. 2015.

[170] José Rueda. *Probabilistic Reliability Analysis of Sustainable Power Systems*. Reader in EE4545 Electric Power Systems of the Future. Accessed on [28.03.2024]. 2024.

[171] Pei Zhang, Ke Meng, and Z.Y. Dong. "Probabilistic vs Deterministic Power System Stability and Reliability Assessment". In: Jan. 2010, pp. 117–145. ISBN: 978-3-642-04281-2. DOI: 10.1007/978-3-642-04282-9_5.

[172] Juan Carlos Beltrán et al. "Comparative analysis of deterministic and probabilistic methods for the integration of distributed generation in power systems". In: *Energy Reports* 6 (2020). Technologies and Materials for Renewable Energy, Environment and Sustainability, pp. 88–104. ISSN: 2352-4847. DOI: https://doi.org/10.1016/j.egyr.2019.10.025. URL: https://www.sciencedirect.com/science/article/pii/S2352484719308534.

[173] Hyeon Yang and Seong Gon Choi. "Deterministic System Analysis to Guarantee Worst Case Performance for Optimal ESS and PV Sizing". In: *IEEE Access* 7 (2019). Received February 7, 2019, accepted February 22, 2019, date of publication March 6, 2019, date of current version August 7, 2019. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2019R1A2C1006167. Corresponding author: Seong Gon Choi (choisg@chungbuk.ac.kr), p. 2903313. DOI: 10.1109/ACCESS.2019.2903313.

[174] Umair Shahzad. *A Comprehensive Review on Power System Risk-Based Transient Stability*. Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA. umair.shahzad@huskers.unl.edu. 2024.

[175] Task Force from REE, Terna, TransnetBW, 50Hertz Transmission, RTE, Swissgrid, Energinet.dk. *Frequency Stability Evaluation Criteria for the Synchronous Zone of Continental Europe – Requirements and impacting factors*. Tech. rep. This report was prepared by a task force with members from REE, Terna, TransnetBW, 50Hertz Transmission, RTE, Swissgrid, and Energinet.dk. RG-CE System Protection & Dynamics Sub Group, Mar. 2016.

[176] ENTSO-E. *Short Circuit Contribution of New Generating Units Connected with Power Electronics and Protection Behaviour*. Accessed: date-of-access. Apr. 2019. URL: https://eepublicdownloads.entsoe.eu/clean-documents/SOC%20documents/190304_SOC_TOP_7.4_Short%20Circuit%20Contribution%20of%20Power%20Electronics%20Connecting%20Generators%20and%20Protection%20report.pdf.

[177] National Renewable Energy Laboratory. *Modeling of Type 4 Wind Turbine Generators*. https://www.esig.energy/wiki-main-page/modeling-of-type-4-wind-turbine-generators/. Accessed: [06.04.2024].

[178] *IEC 61400-27-1:2020 (Edition 2.0, 2020) - Wind turbines - Part 27-1: Electrical simulation models - Wind turbines*. International Electrotechnical Commission, 2020.

[179] Chong Zhang et al. "The influence of phase-locked loop on the stability of single-phase grid-connected inverter". In: *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2015, pp. 4737–4744. DOI: 10.1109/ECCE.2015.7310329.

[180] Gaoxiang Li et al. "PLL phase margin design and analysis for mitigating sub/super-synchronous oscillation of grid-connected inverter under weak grid". In: *International Journal of Electrical Power & Energy Systems* 151 (2023), p. 109124. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2023.109124. URL: https://www.sciencedirect.com/science/article/pii/S0142061523001813.

[181] Wenjuan Du, Xubin Wang, and Haifeng Wang. "Sub-synchronous interactions caused by the PLL in the grid-connected PMSG for the wind power generation". In: *International Journal of Electrical Power & Energy Systems* 98 (2018), pp. 331–341. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2017.11.018. URL: https://www.sciencedirect.com/science/article/pii/S0142061517324201.

[182] Yin Li, Lingling Fan, and Zhixin Miao. "Wind in Weak Grids: Low-Frequency Oscillations, Subsynchronous Oscillations, and Torsional Interactions". In: *IEEE Transactions on Power Systems* 35.1 (2020), pp. 109–118. DOI: 10.1109/TPWRS.2019.2924412.

[183] Shun-Hsien Huang et al. "Voltage control challenges on weak grids with high penetration of wind generation: ERCOT experience". In: July 2012, pp. 1–7. ISBN: 978-1-4673-2727-5. DOI: 10.1109/PESGM.2012.6344713.

[184] Xumeng Cui et al. "Design of Subsynchronous Oscillation Damping Controller of Grid-Connected VSC Based on Selective Modal Analysis Method". In: *2022 4th International Conference on Smart Power & Internet Energy Systems (SPIES)*. 2022, pp. 1516–1522. DOI: 10.1109/SPIES55999.2022.10082019.

[185] Lin Zhu et al. "A Comprehensive Method to Mitigate Forced Oscillations in Large Interconnected Power Grids". In: *IEEE Access* 9 (2021), pp. 22503–22515. DOI: 10.1109/ACCESS.2021.3056123.

[186] Felipe Arraño-Vargas et al. "Mitigation of power system oscillations in weak grids with battery energy storage systems: A real-world case study". In: *Energy* 283 (2023), p. 128648. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2023.128648. URL: https://www.sciencedirect.com/science/article/pii/S036054422302042X.

[187] José Luis Rodríguez-Amenedo and Santiago Arnaltes Gómez. "Damping Low-Frequency Oscillations in Power Systems Using Grid-Forming Converters". In: *IEEE Access* 9 (2021), pp. 158984–158997. DOI: 10.1109/ACCESS.2021.3130333.

[188] M. Klein, G.J. Rogers, and P. Kundur. "A fundamental study of inter-area oscillations in power systems". In: *IEEE Transactions on Power Systems* 6.3 (1991), pp. 914–921. DOI: 10.1109/59.119229.

[189] Lajos Kiss and Jozsef Zerenyi. "The Impact of a Power Plant on the Inter Area Oscillations". In: *2007 IEEE Lausanne Power Tech*. 2007, pp. 74–79. DOI: 10.1109/PCT.2007.4538295.

[190] ENTSO-E SG SPD REPORT. *Analysis of CE Inter-Area Oscillations of 19 and 24 February 2011*. Tech. rep. Aug. 2011. URL: https://eepublicdownloads.entsoe.eu/clean-documents/pre2015/publications/entsoe/RG_SOC_CE/Top7_110913_CE_inter-area-oscil_feb_19th_24th_final.pdf.

[191] ENTSO-E SG SPD REPORT. *Analysis of CE Inter-Area Oscillations of 1st December 2016*. Tech. rep. July 2017. URL: https://eepublicdownloads.entsoe.eu/clean-documents/SOC%20documents/Regional_Groups_Continental_Europe/2017/CE_inter-area_oscillations_Dec_1st_2016_PUBLIC_V7.pdf.

[192] ENTSO-E AISBL. *Oscillation Event 03.12.2017 System Protection and Dynamics WG*. Tech. rep. Mar. 2018. URL: https://eepublicdownloads.entsoe.eu/clean-documents/SOC%20documents/Regional_Groups_Continental_Europe/OSCILLATION_REPORT_SPD.pdf.

[193] A.R. Messina et al. "Coordinated application of FACTS controllers to damp out inter-area oscillations". In: *Electric Power Systems Research* 62.1 (2002), pp. 43–53. ISSN: 0378-7796. DOI: https://doi.org/10.1016/S0378-7796(02)00034-2. URL: https://www.sciencedirect.com/science/article/pii/S0378779602000342.

[194] Zuan Zhang, Yanchang Liang, and Xiaowei Zhao. "Adaptive inter-area power oscillation damping from offshore wind farm and MMC-HVDC using deep reinforcement learning". In: *Renewable Energy* 224 (2024), p. 120164. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2024.120164. URL: https://www.sciencedirect.com/science/article/pii/S0960148124002295.

[195] Keerthy Mullakkal Pradeep and V. Mini. "Mitigation of inter area oscillations in a two area system using PSS and STATCOM". In: *2018 International Conference on Power, Instrumentation, Control and Computing (PICC)*. 2018, pp. 1–5. DOI: 10.1109/PICC.2018.8384797.

[196] Callum Henderson, Agusti Egea-Alvarez, and Lie Xu. "Analysis of optimal grid-forming converter penetration in AC connected offshore wind farms". In: *International Journal of Electrical Power & Energy Systems* 157 (2024), p. 109851. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2024.109851. URL: https://www.sciencedirect.com/science/article/pii/S0142061524000723.

A

Grid-forming Converters Implementation in PowerFactory

A.1. Additional Material

A.1.1. Virtual Synchronous Machine

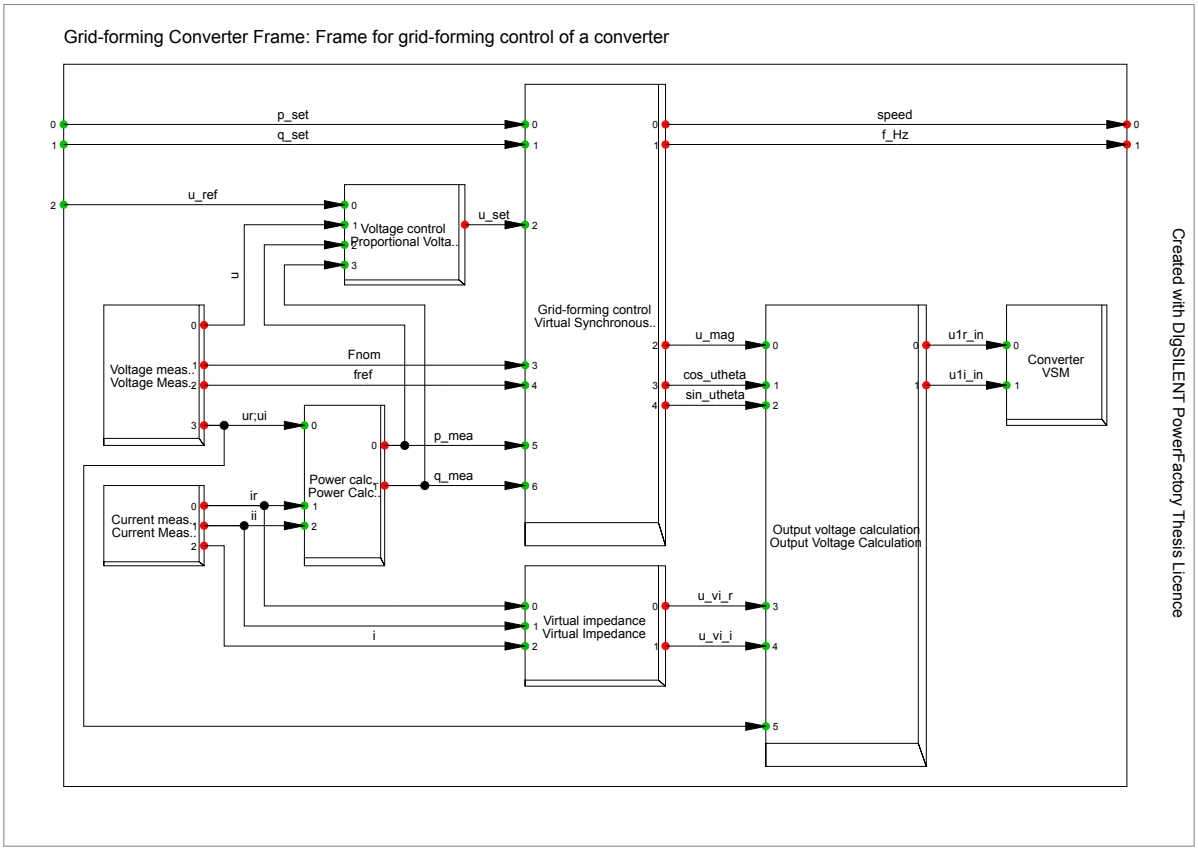


Figure A.1: VSM converter frame composite model.

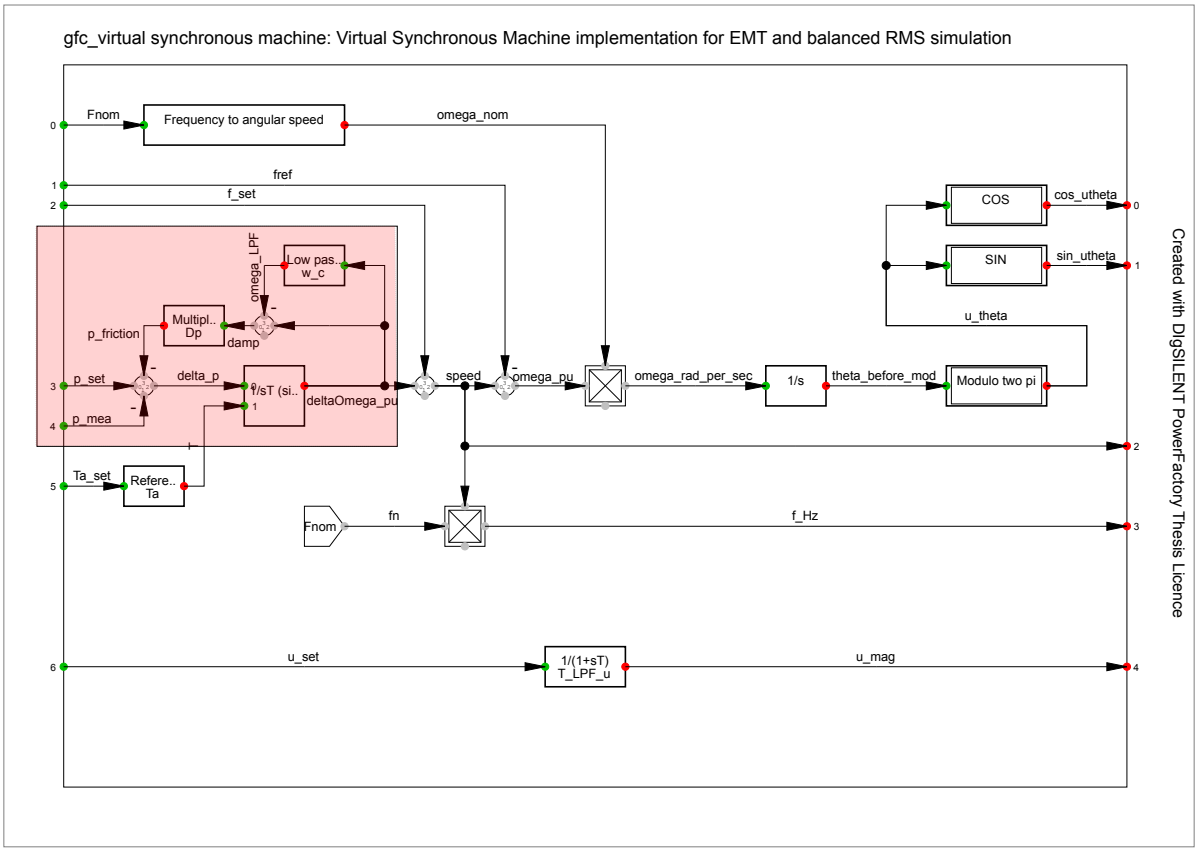


Figure A.2: VSM block DSL frame.

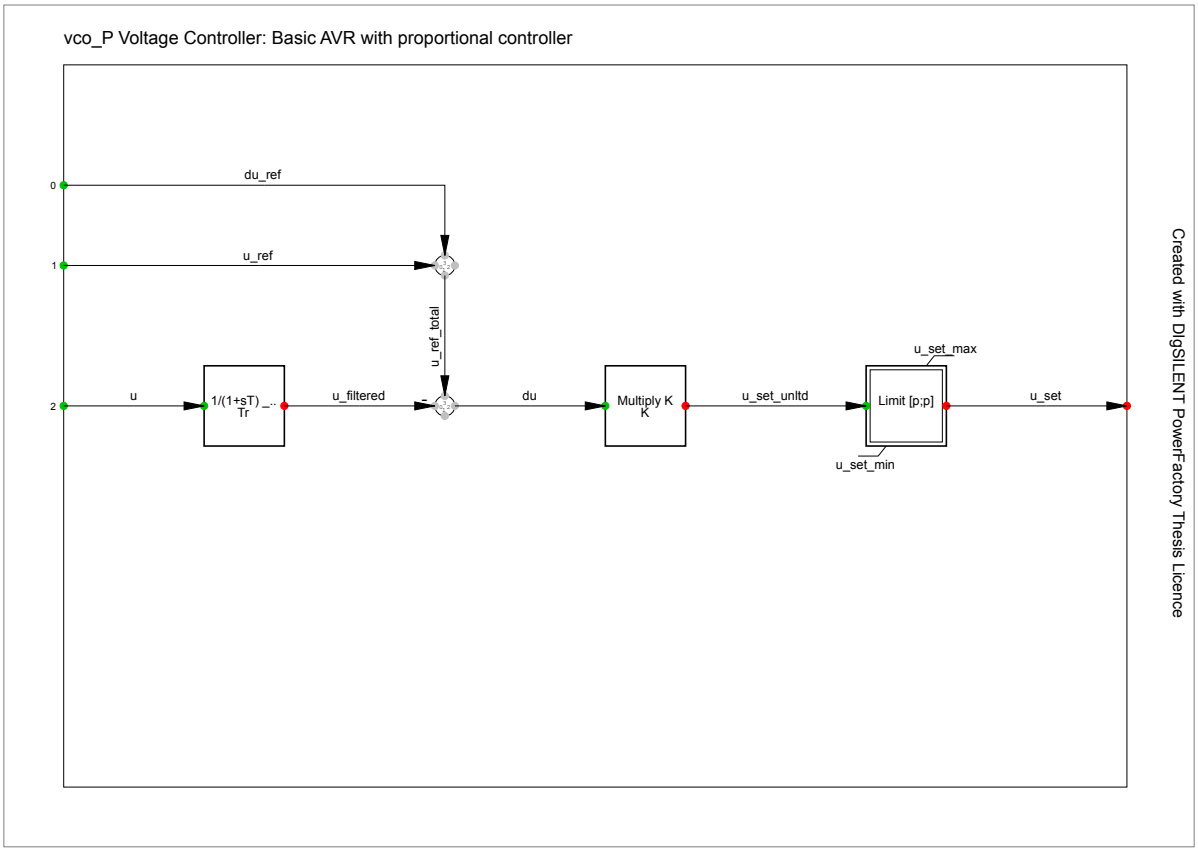


Figure A.3: Proportionate voltage controller DSL frame.

A.1.2. Synchroverter

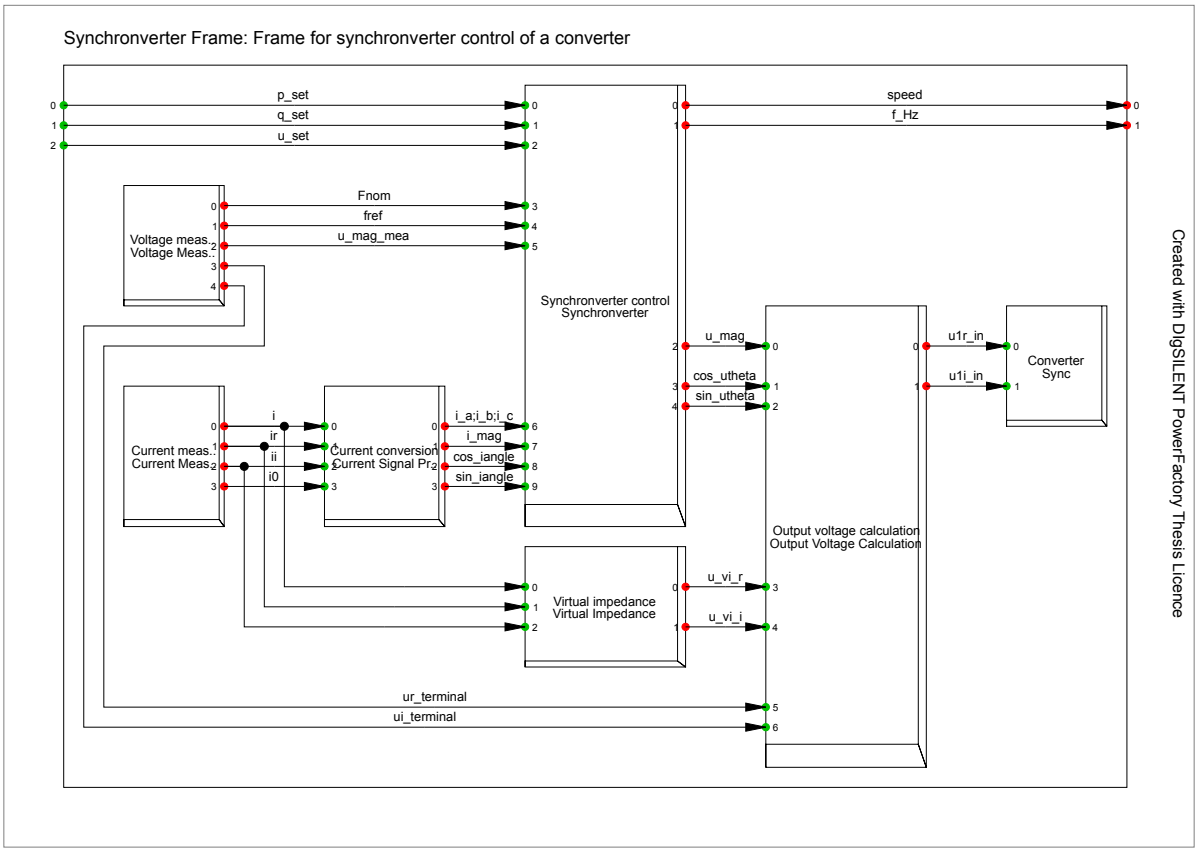


Figure A.4: Synchroverter converter frame composite model.

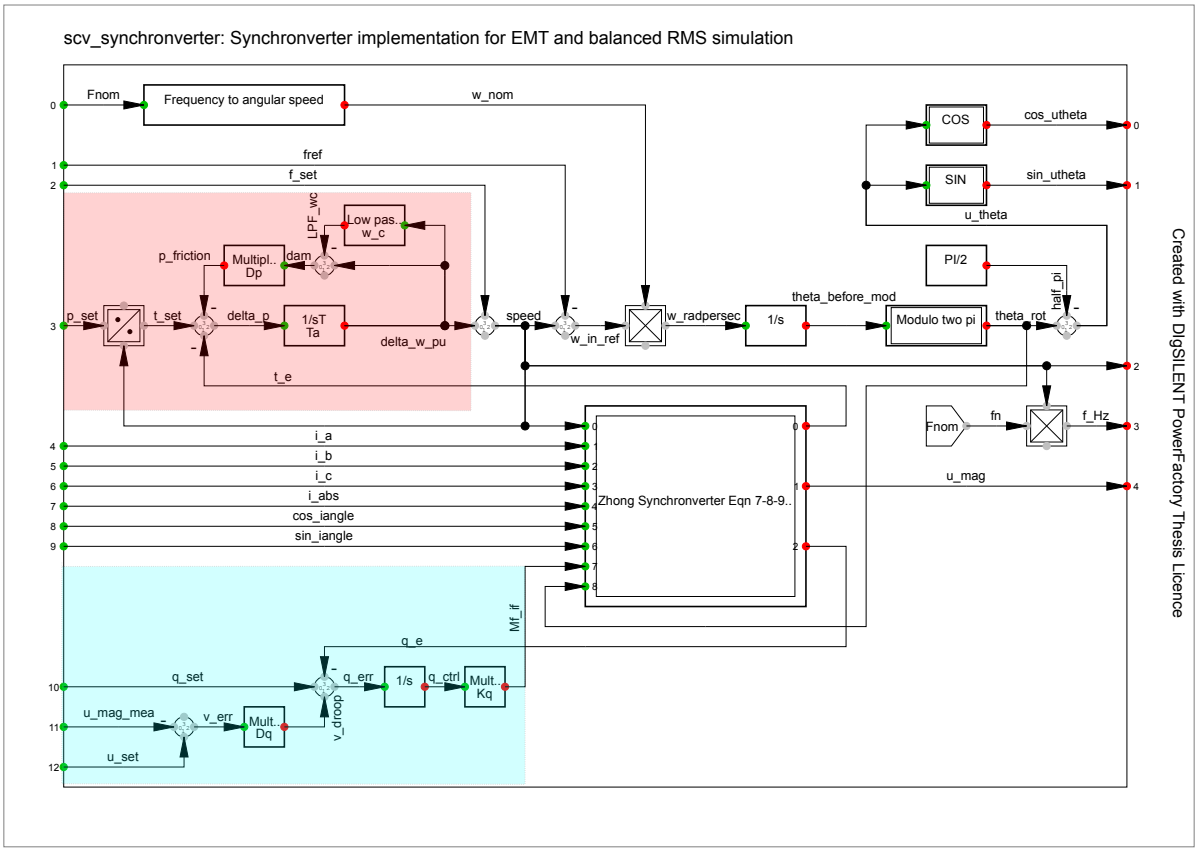


Figure A.5: Synchroverter block DSL frame.

A.1.3. Droop Control

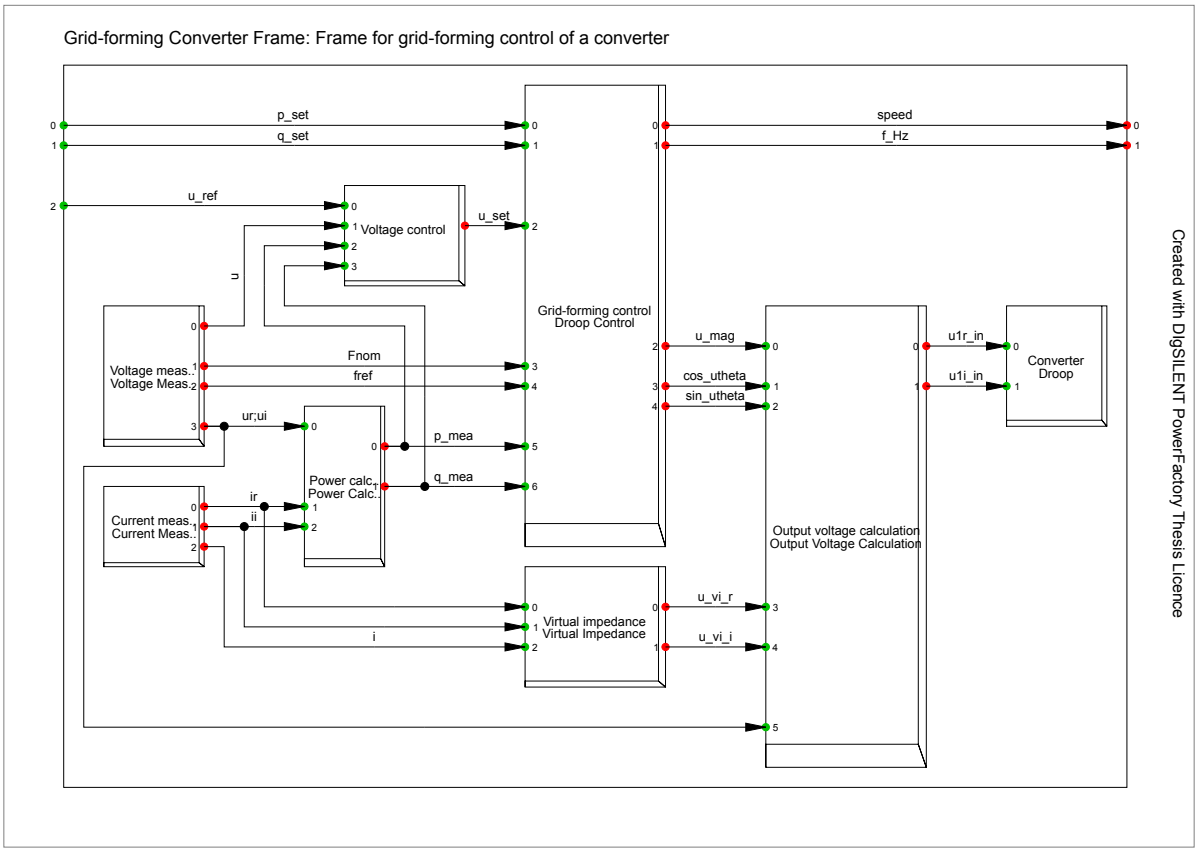


Figure A.6: Droop converter frame composite model.

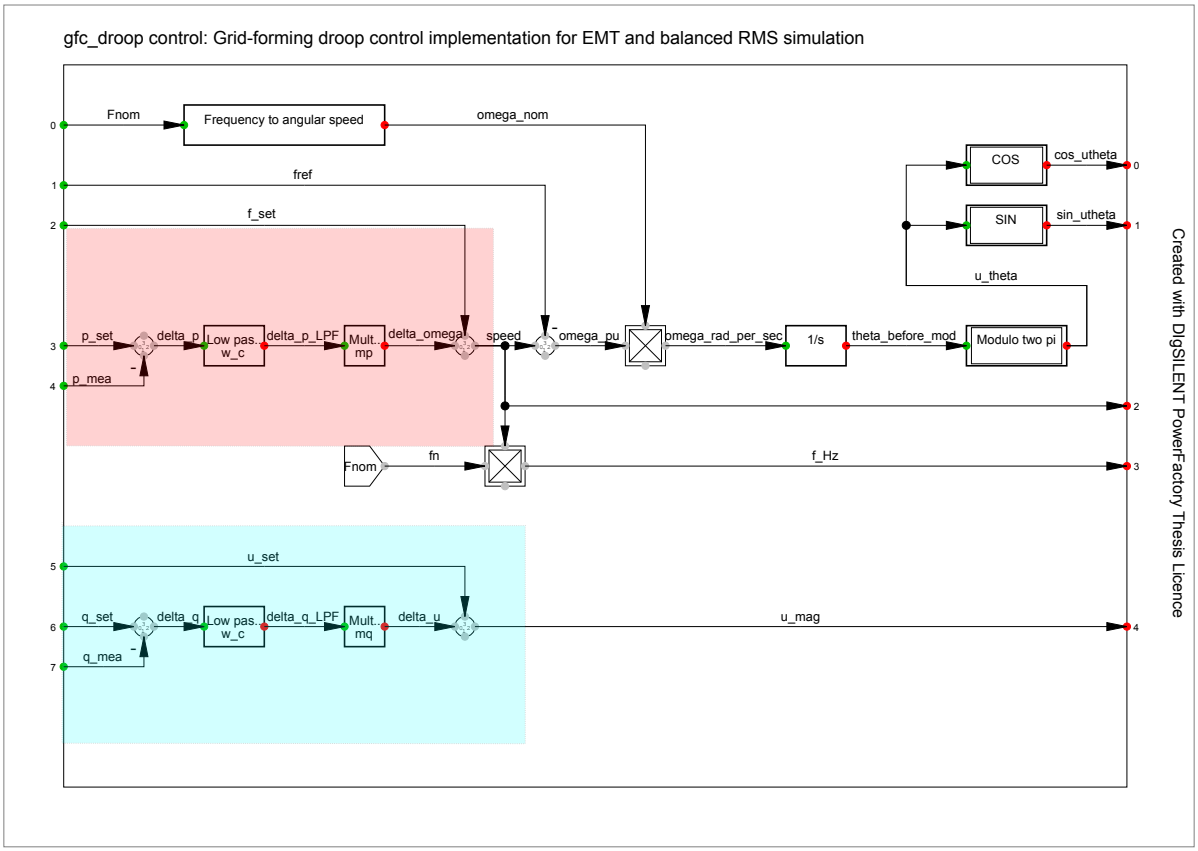


Figure A.7: Droop control block DSL frame.

B

Case Studies Results

B.1. Case Study A

B.1.1. Short-Circuit event

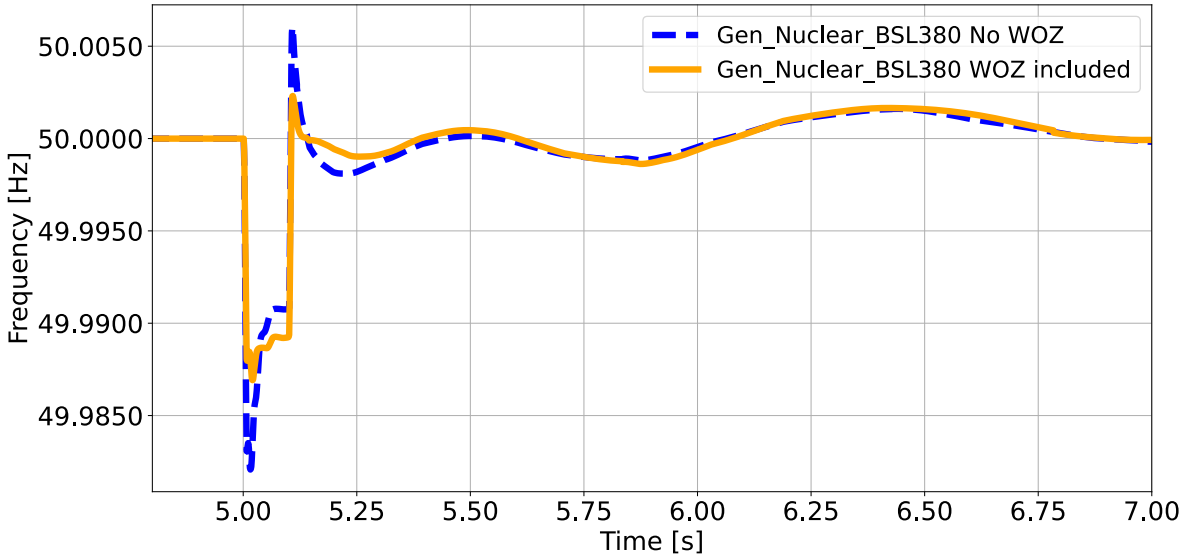


Figure B.1: Frequency response of 2 GW generator Gen Nuclear BSL380 after a short circuit occurring after 5 seconds for Case C, with comparison of WOZ converters implemented.

B.1.2. Load Event

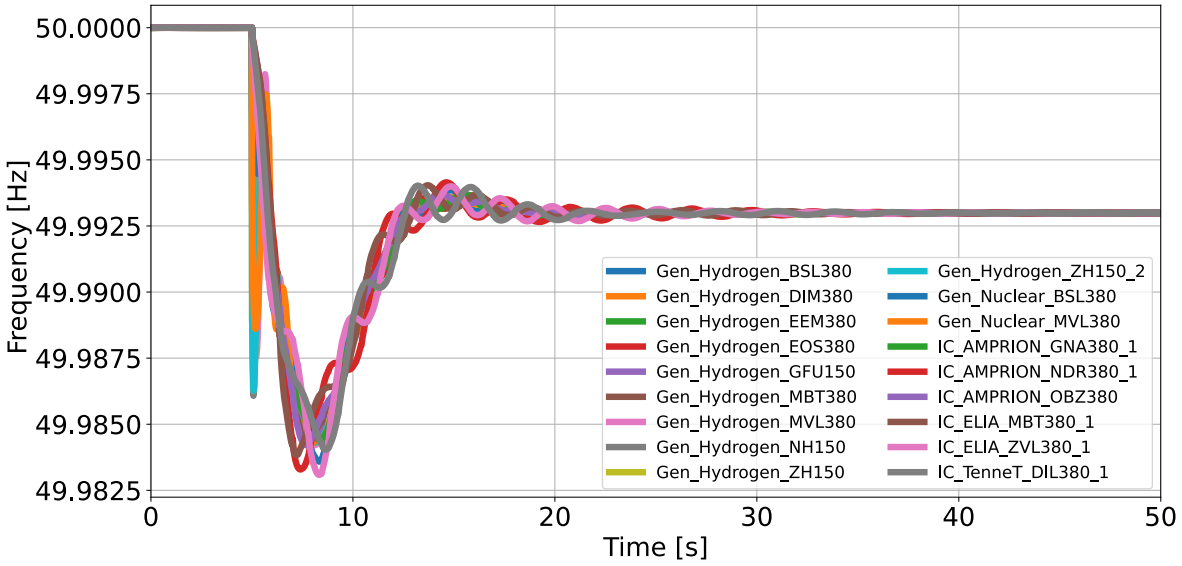


Figure B.2: Frequency response of all synchronous machines after a load event occurring after 5 seconds for Case A.

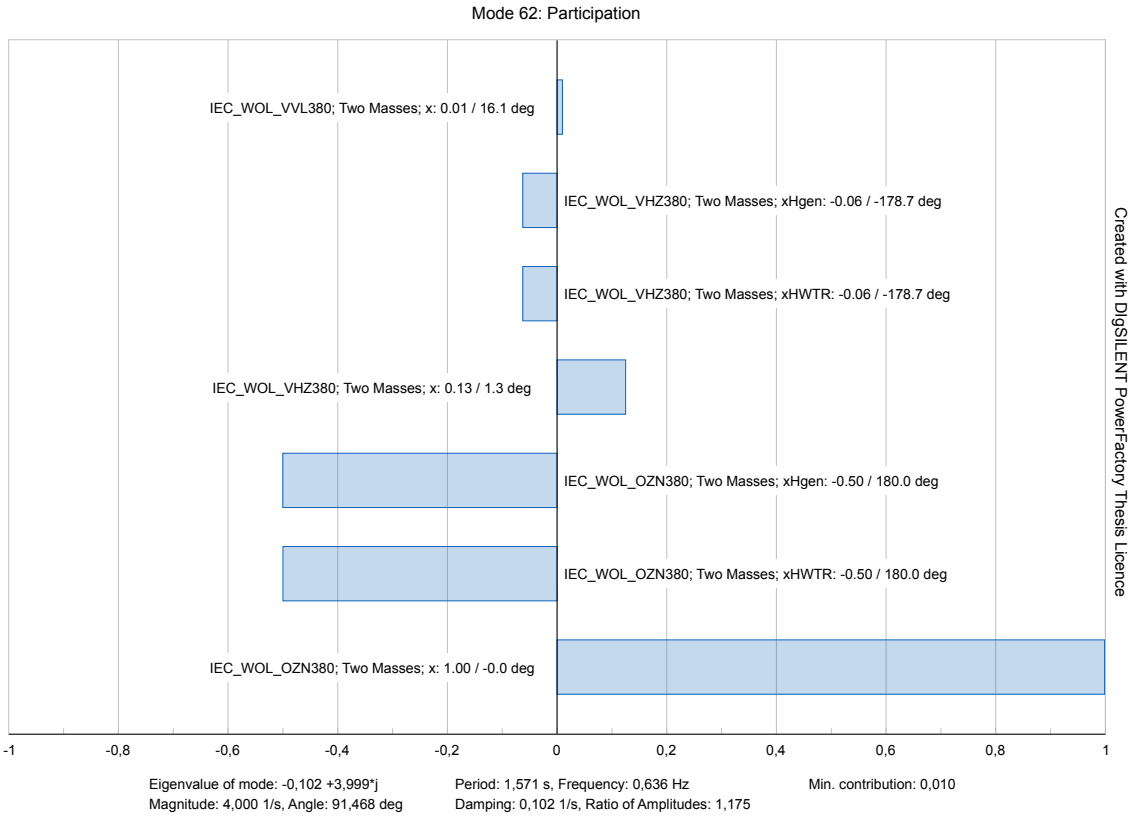


Figure B.3: Overview of participation factors for mode 62.

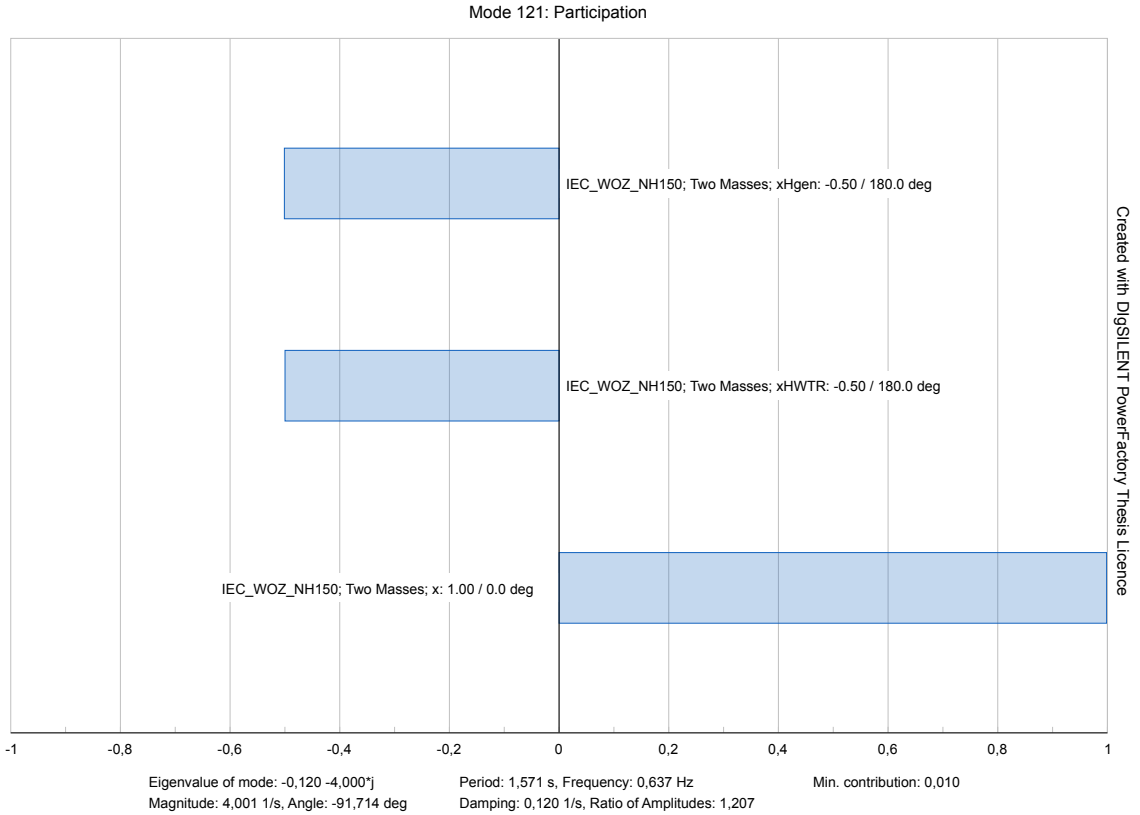


Figure B.4: Overview of participation factors for mode 121.

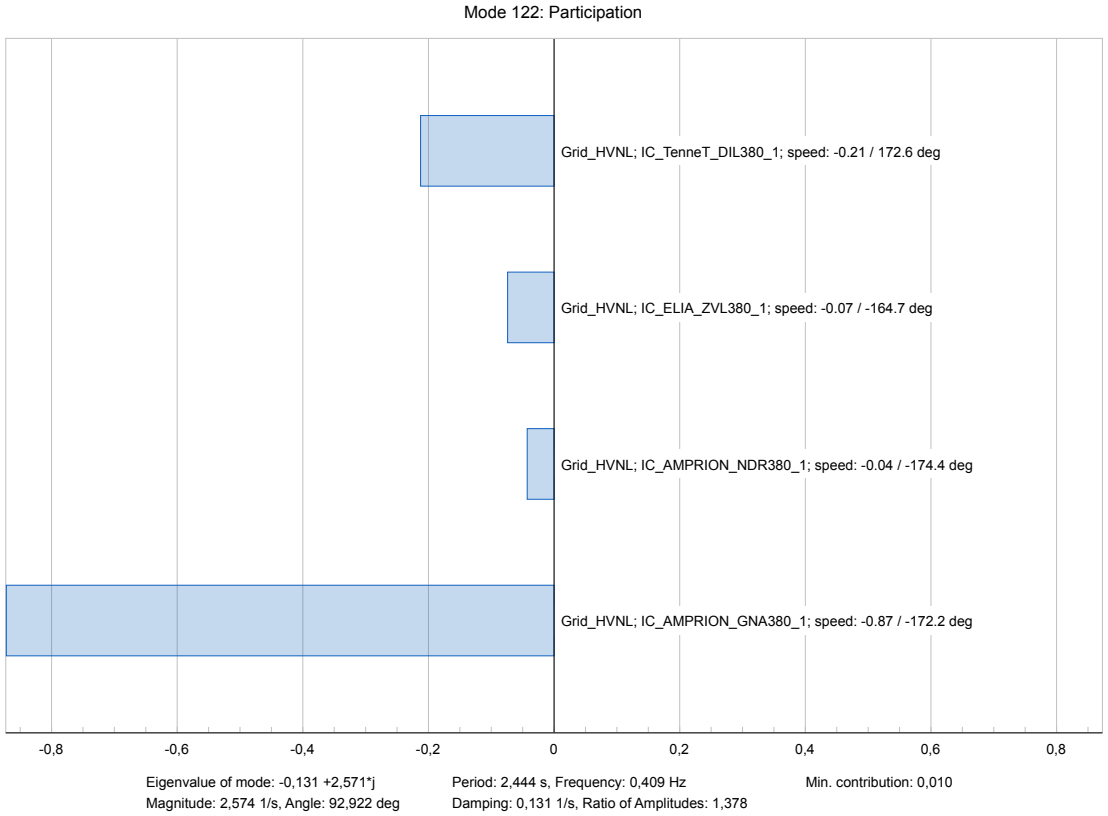


Figure B.5: Overview of participation factors for mode 122.

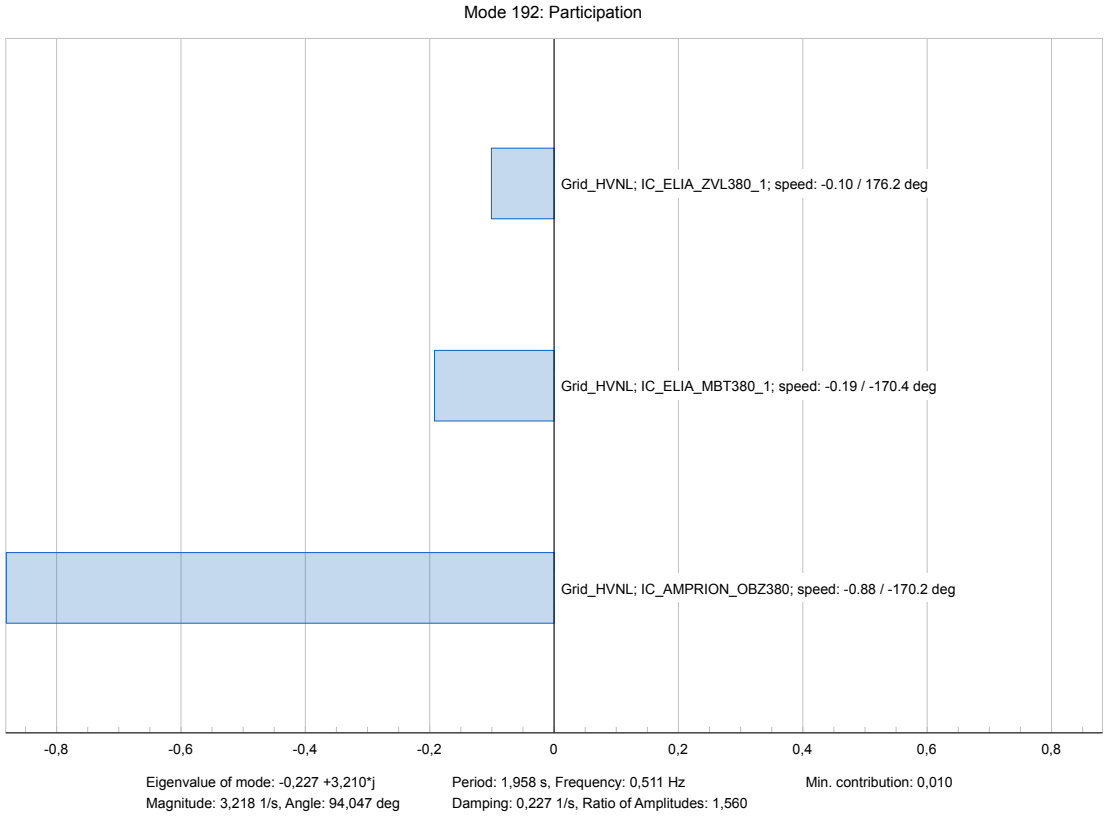


Figure B.6: Overview of participation factors for mode 192.

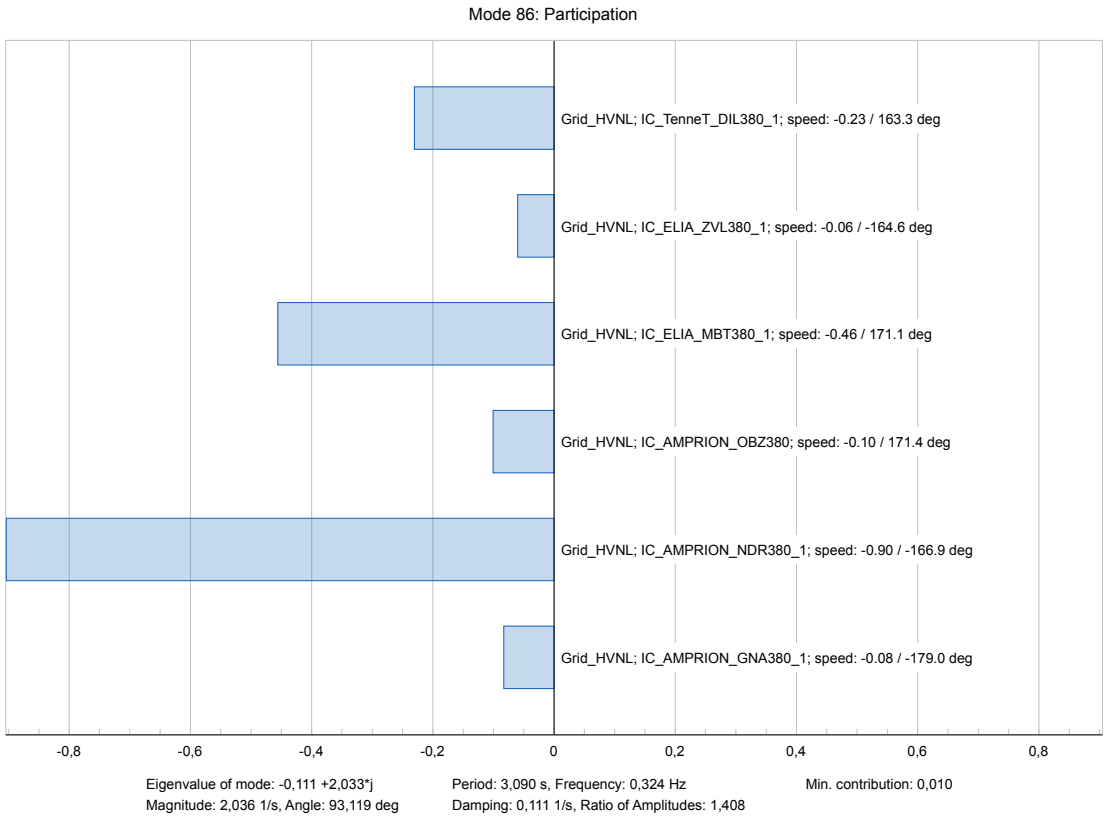


Figure B.7: Overview of participation factors for mode 86.

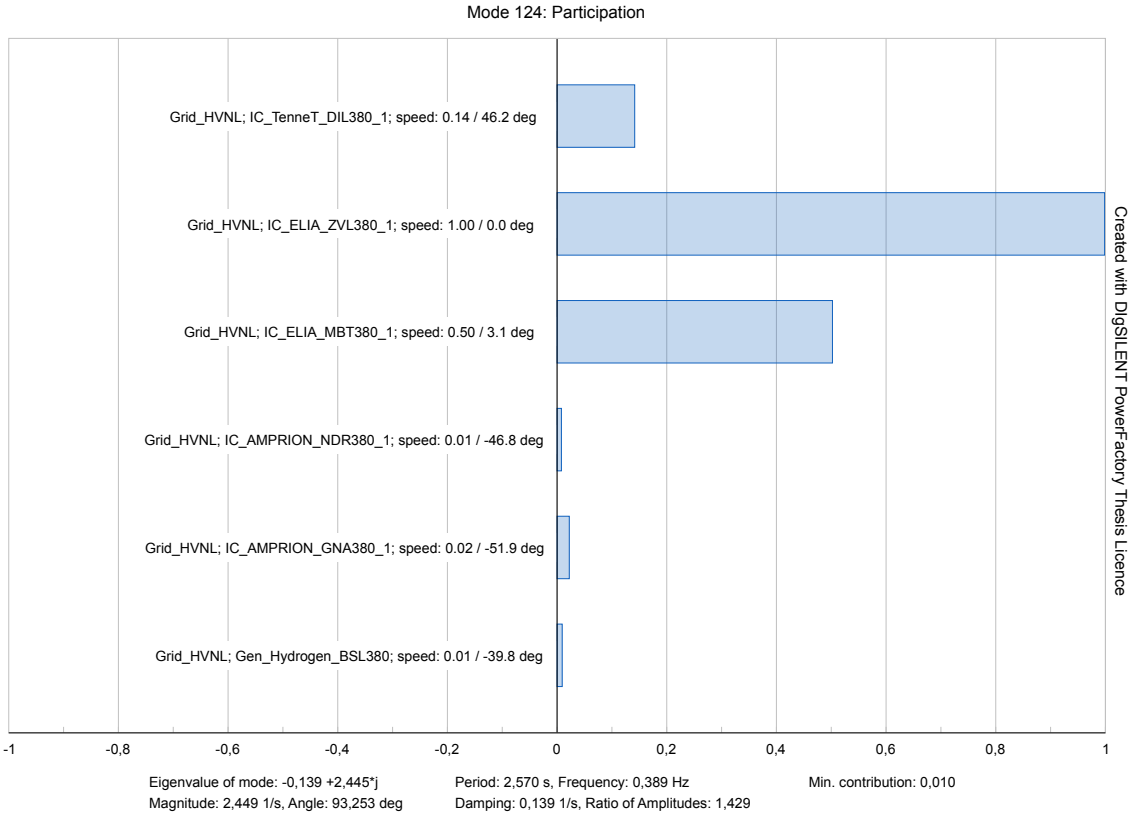


Figure B.8: Overview of participation factors for mode 124.

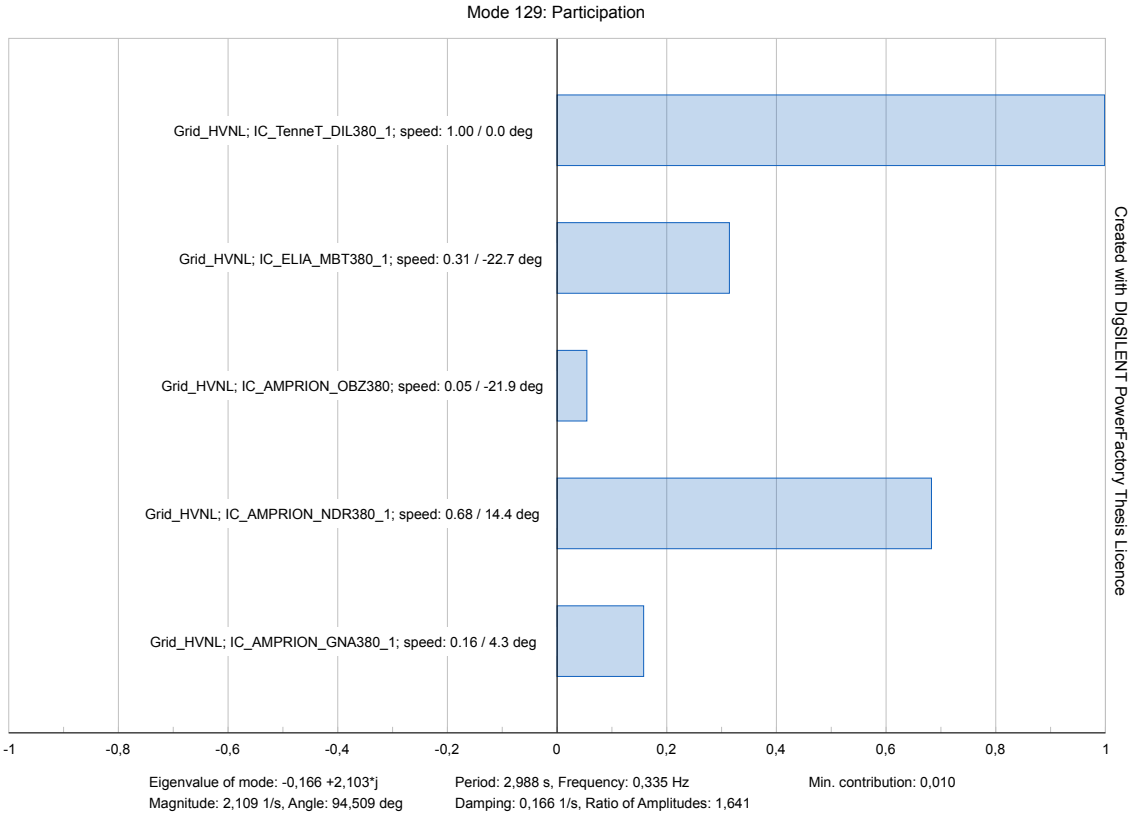


Figure B.9: Overview of participation factors for mode 129.

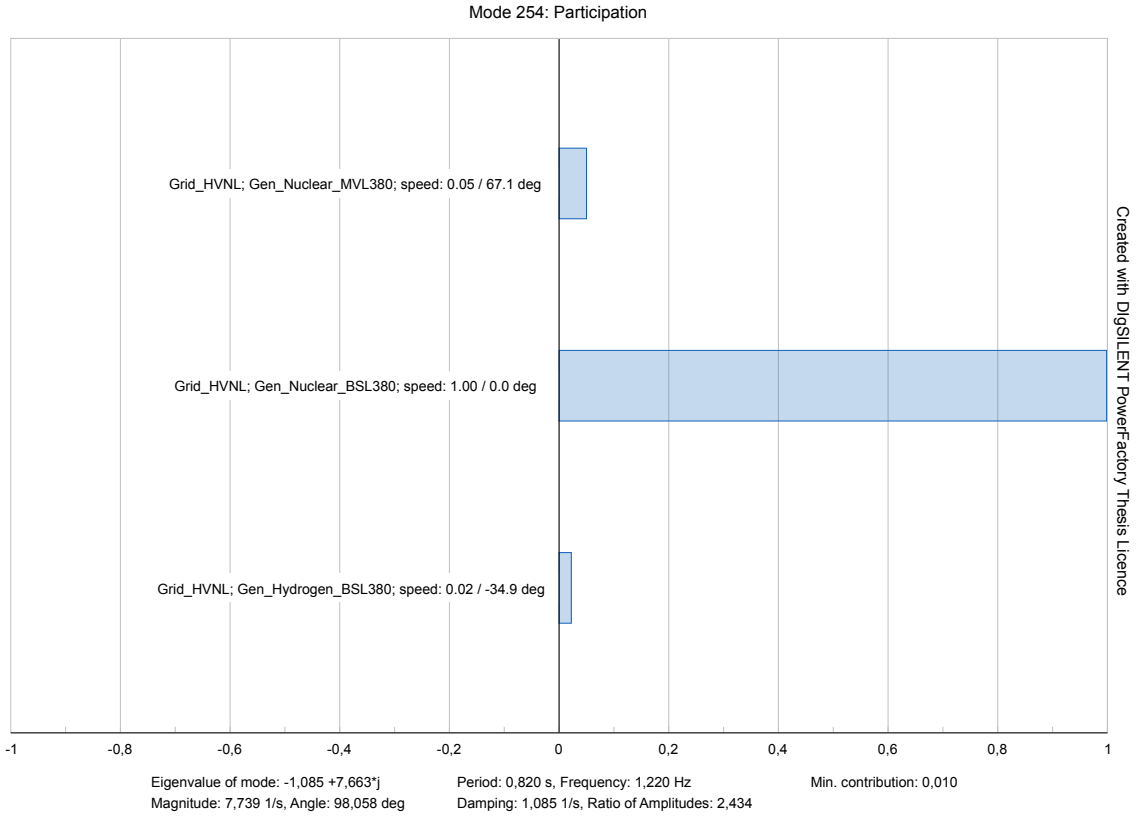
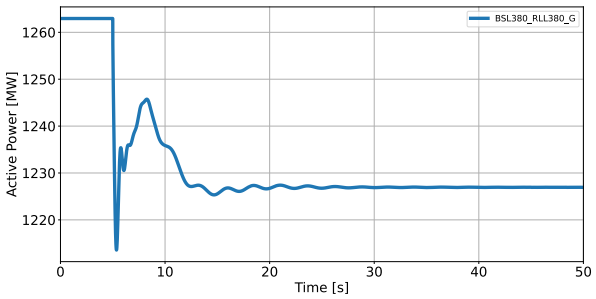
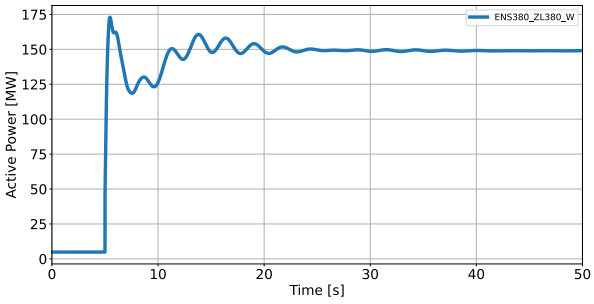


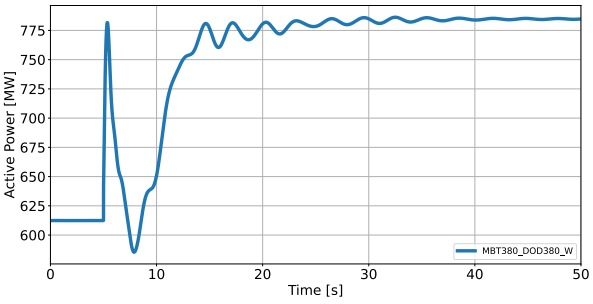
Figure B.10: Overview of participation factors for mode 254



(a) Active power transfer for transmission line BSL380-RLL380-W.



(b) Active power transfer for transmission line ENS380-ZL380-W.



(c) Active power transfer for transmission line MBT380-DOD380-W.

Figure B.11: Active power transfer for different transmission lines for a load increase, extended simulation window.

B.2. Case Study B

B.2.1. Comparative Analysis of Grid-Forming Controllers

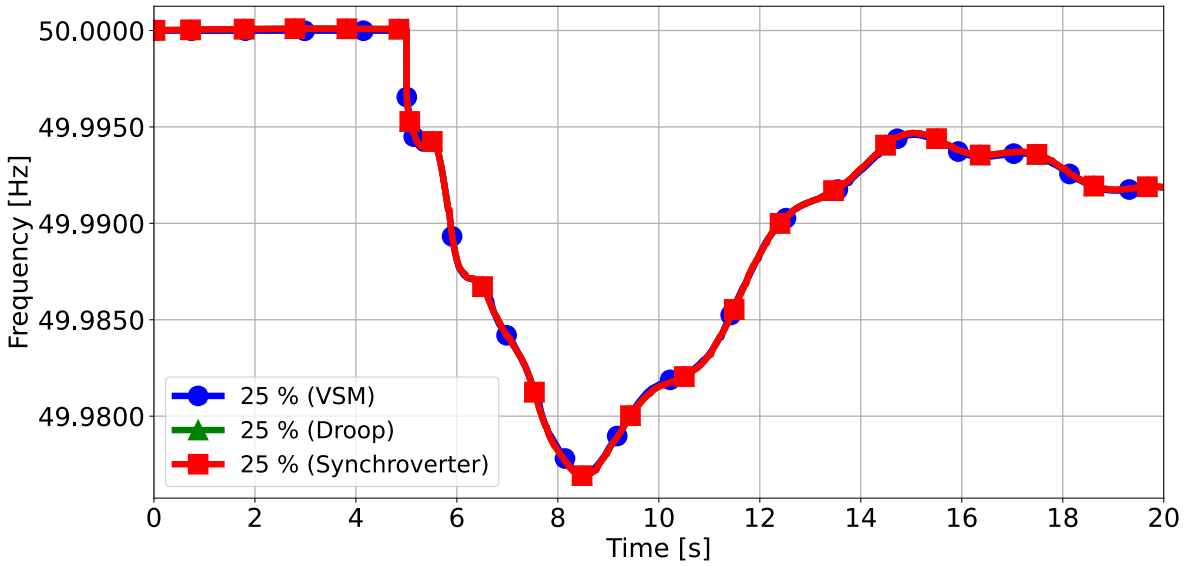


Figure B.12: Frequency response for different grid-forming controls at 25 % penetration after a load response occurring at 5 seconds.

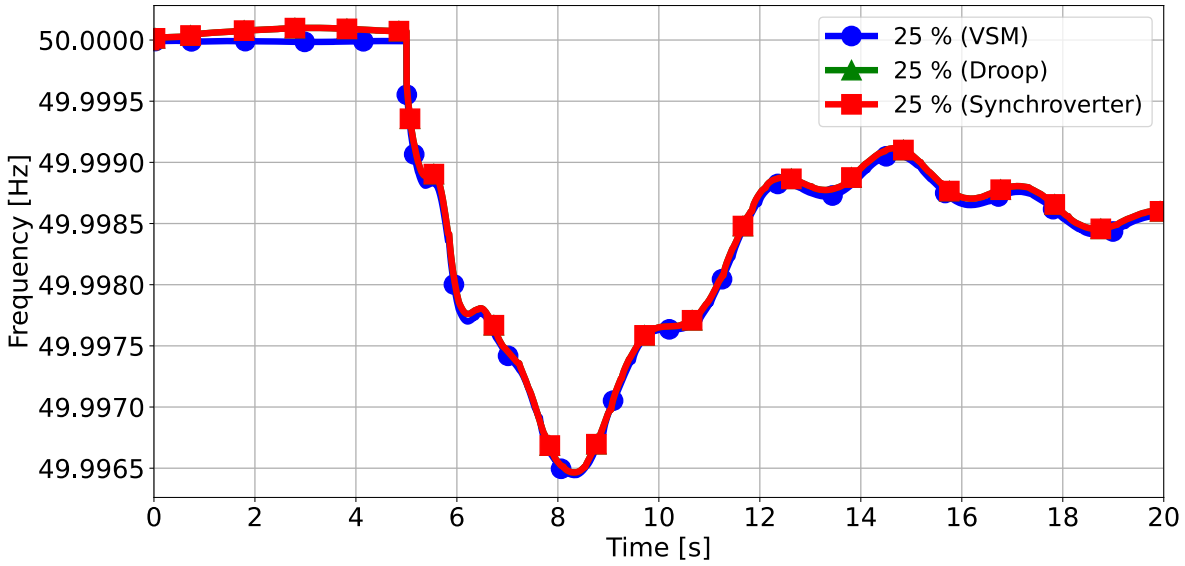


Figure B.13: Frequency response for different grid-forming controls at 25 % penetration after an outage event occurring at 5 seconds.

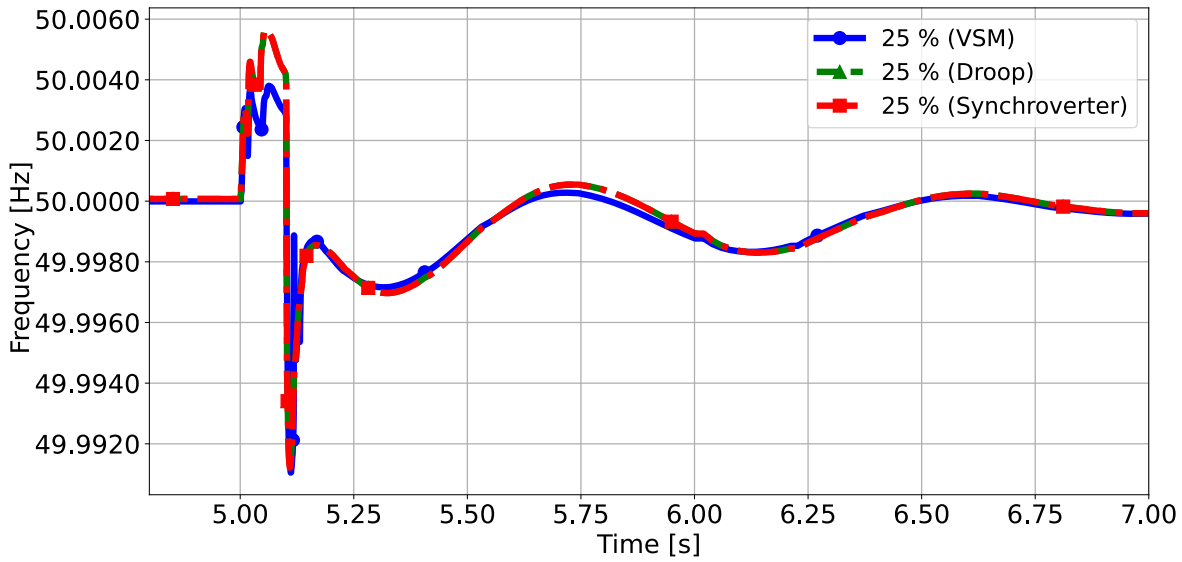


Figure B.14: Frequency response for different grid-forming controls at 25 % penetration after a short circuit occurring at 5 seconds.

Table B.1: ROCOF, Frequency deviation, and Frequency Nadir for the three controllers compared at 25 % grid-forming penetration.

Outage Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (25%)	49.9964	1.6	1.1	3.6
Droop (25%)	49.9965	1.5	1.1	3.5
Synchroverter (25%)	49.9965	1.5	1.1	3.5

Short-Circuit Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (25%)	49.9911	68.0	17.5	8.9
Droop (25%)	49.9912	68.4	17.3	8.8
Synchroverter (25%)	49.9912	68.4	17.3	8.8

Load Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (25%)	49.9770	7.2	5.8	23.0
Droop (25%)	49.9769	7.3	5.9	23.1
Synchroverter (25%)	49.9769	7.3	5.9	23.1

Table B.2: ROCOF, Frequency deviation, and Frequency Nadir for the three controllers compared at 85 % grid-forming penetration.

Outage Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (85%)	49.9981	1.0	0.7	1.9
Droop (85%)	49.9981	0.9	0.7	1.9
Synchroverter (85%)	49.9981	1.0	0.8	1.9

Short-Circuit Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (85%)	49.9968	18.8	5.3	3.2
Droop (85%)	49.9967	13.8	4.6	3.3
Synchroverter (85%)	49.9958	29.3	7.9	4.2

Load Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
VSM (85%)	49.9887	4.4	3.7	11.3
Droop (85%)	49.9886	5.1	3.8	11.4
Synchroverter (85%)	49.9886	5.3	3.9	11.4

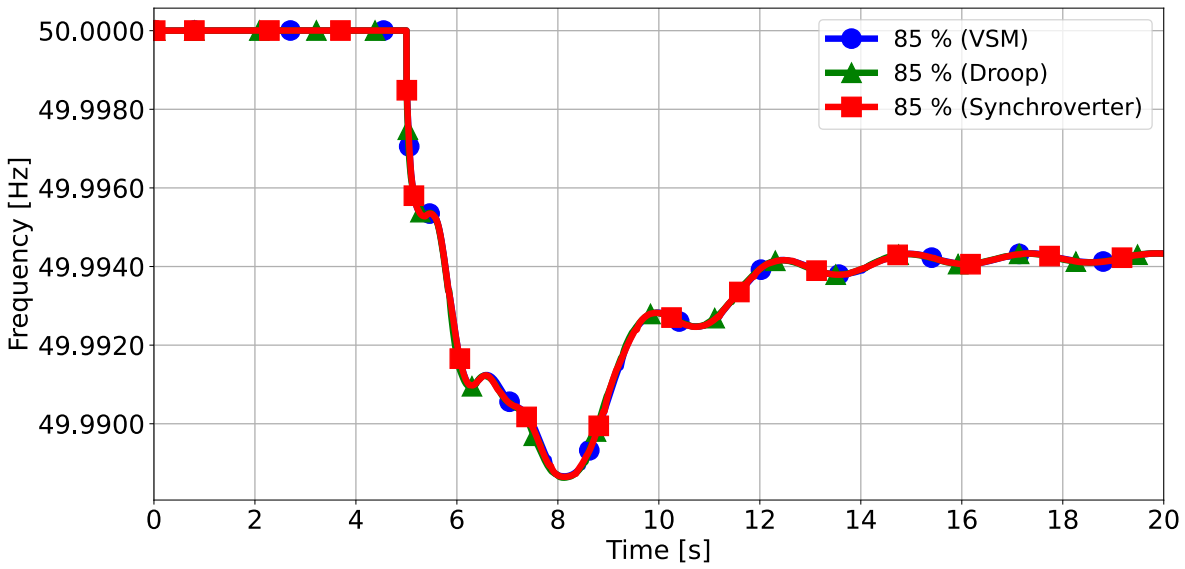


Figure B.15: Frequency response for different grid-forming controls at 85 % penetration after a load response occurring at 5 seconds.

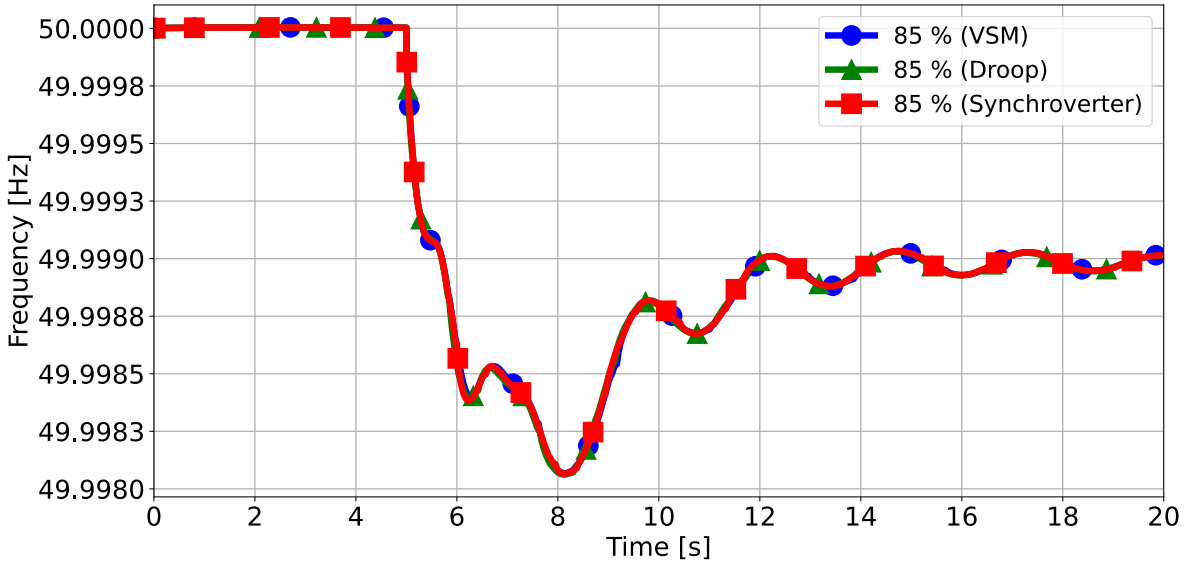


Figure B.16: Frequency response for different grid-forming controls at 85 % penetration after an outage event occurring at 5 seconds.

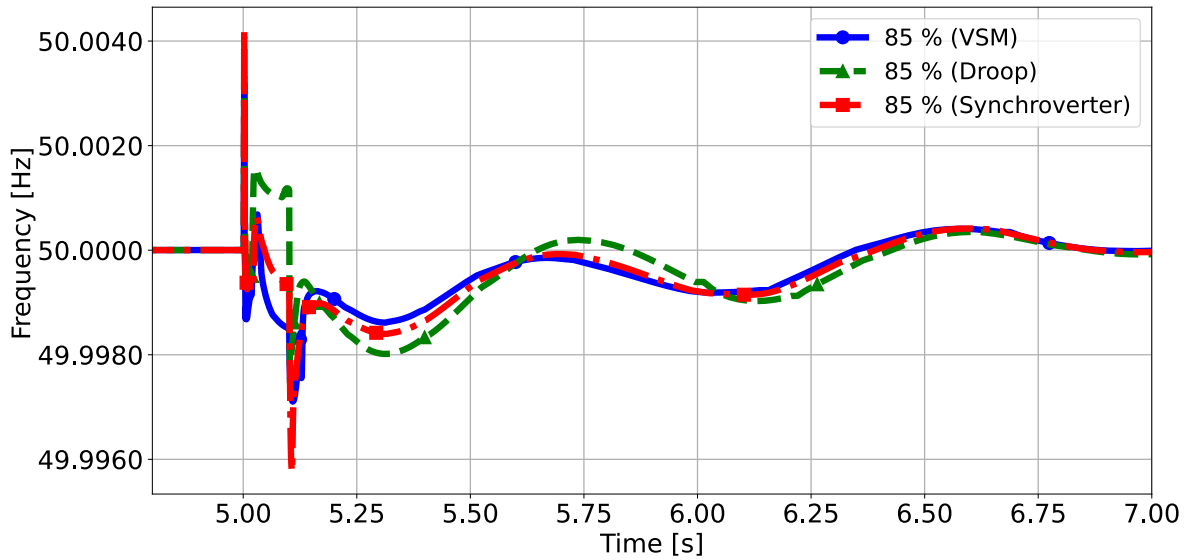


Figure B.17: Frequency response for different grid-forming controls at 85 % penetration after a short circuit occurring at 5 seconds.

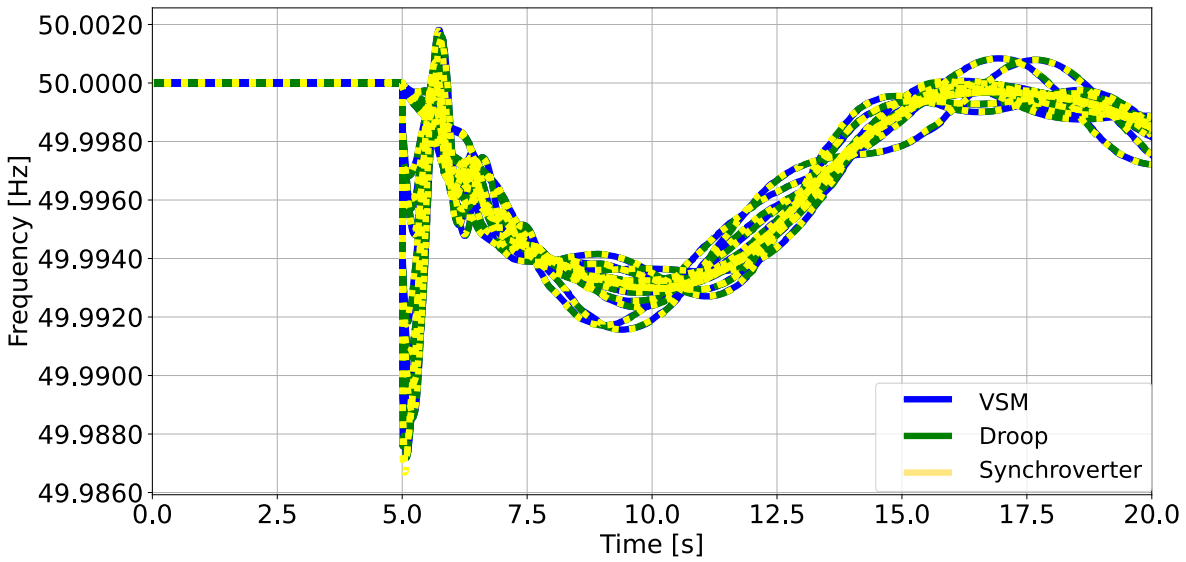


Figure B.18: Frequency response for all synchronous machines for different grid-forming controls at 50 % penetration after a load response occurring at 5 seconds.

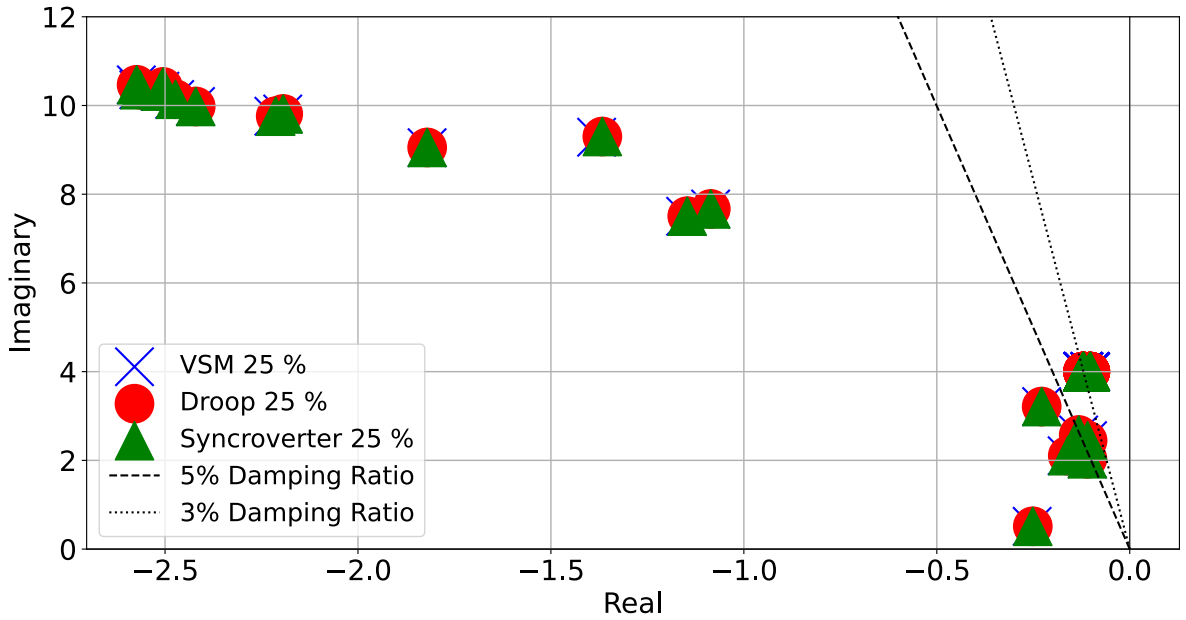


Figure B.19: Eigenvalue plot comparing the three different controllers at 25 % grid-forming penetration, all modes under 50 % damping ratio included.

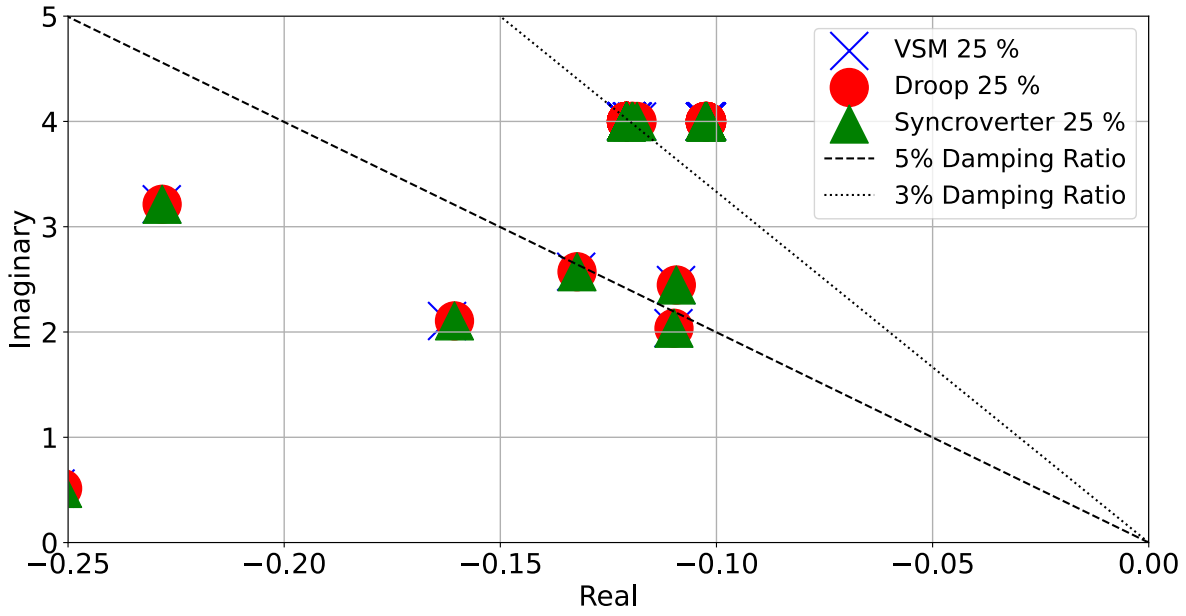


Figure B.20: Eigenvalue plot comparing the three different controllers at 25 % grid-forming penetration, zoomed in on the most critical modes.

Table B.3: Comparison of Eigenvalues, Frequencies, and Damping Ratios for VSM, Droop, and Synchroverter at 25 % grid-forming penetration.

Mode index	Eigenvalue	Frequency (Hz)	Damping Ratio (%)		
			VSM	Droop	Sync
92	$-0.1023 - 3.9989i$	0.6364	2.5565	2.5565	2.5565
183	$-0.1208 + 3.9996i$	0.6366	3.0190	3.0190	3.0190
128	$-0.1094 - 2.4459i$	0.3893	4.4667	4.4625	4.4625
184	$-0.1325 - 2.5725i$	0.4094	5.1421	5.1348	5.1348
130	$-0.1099 - 2.0365i$	0.3241	5.3907	5.3877	5.3877
252	$-0.2283 - 3.2132i$	0.5114	7.0878	7.0852	7.0852
189	$-0.1623 - 2.1045i$	0.3349	7.6904	7.6049	7.6049
313	$-1.0861 - 7.6667i$	1.2202	14.0266	14.0188	14.0188

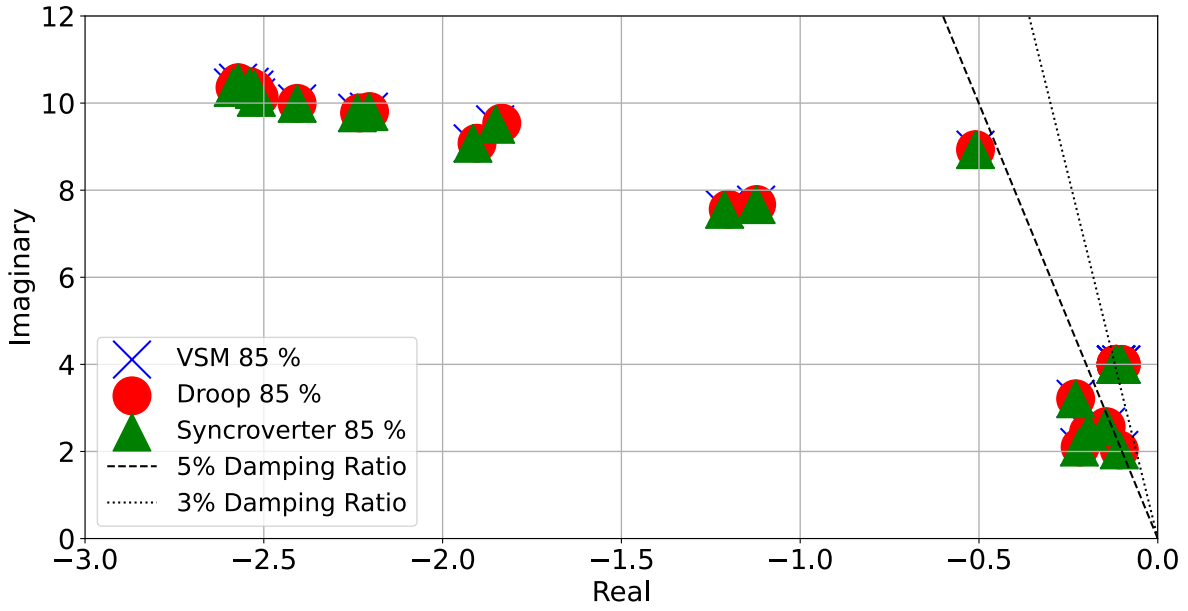


Figure B.21: Eigenvalue plot comparing the three different controllers at 85 % grid-forming penetration, all modes under 50 % damping ratio included.

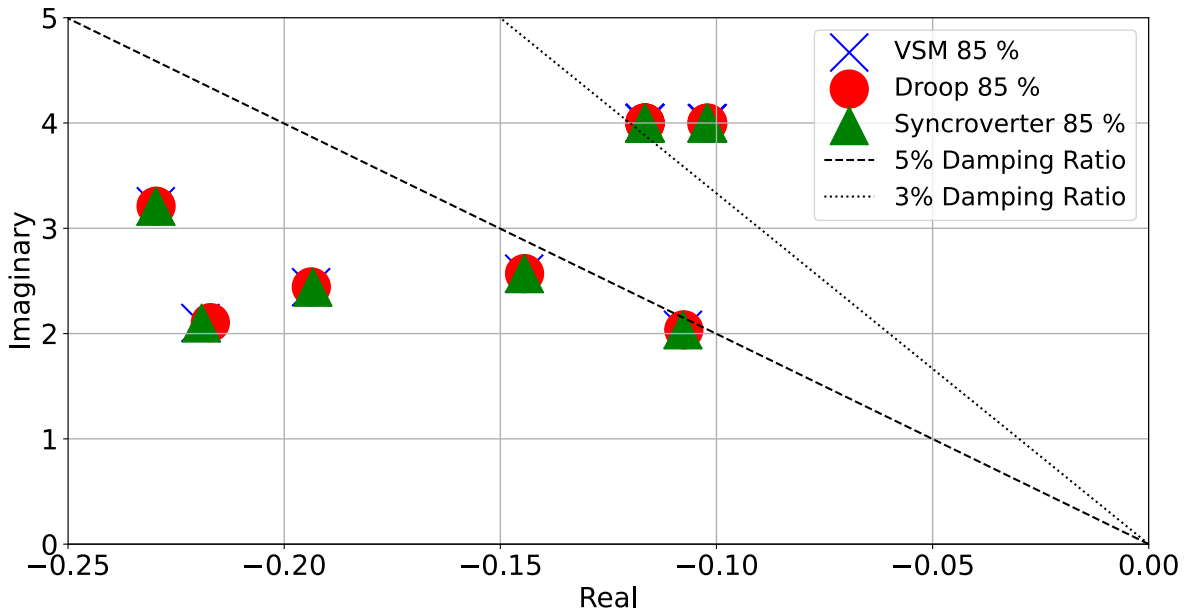


Figure B.22: Eigenvalue plot comparing the three different controllers at 85 % grid-forming penetration, zoomed in on the most critical modes.

Table B.4: Comparison of Eigenvalues, Frequencies, and Damping Ratios for VSM, Droop, and Synchroverter at 85 % grid-forming penetration.

Mode index	Eigenvalue	Frequency (Hz)	Damping Ratio (%)		
			VSM	Droop	Sync
24	$-0.1019 - 3.9989i$	0.6364	2.5485	2.5485	2.5485
43	$-0.1165 + 3.9995i$	0.6365	2.9125	2.9125	2.9125
30	$-0.1078 - 2.0358i$	0.3240	5.2854	5.2768	5.2858
44	$-0.1445 - 2.5693i$	0.4089	5.6146	5.6103	5.6125
128	$-0.5097 - 8.9315i$	1.4215	5.6974	5.6974	5.6974
114	$-0.2297 - 3.2095i$	0.5108	7.1388	7.1360	7.1380
58	$-0.1938 - 2.4406i$	0.3884	7.9148	7.9045	7.8969
94	$-0.2194 - 2.1005i$	0.3343	10.3876	10.2634	10.3748

B.2.1.1. Comparison of Grid-Forming Penetration Levels

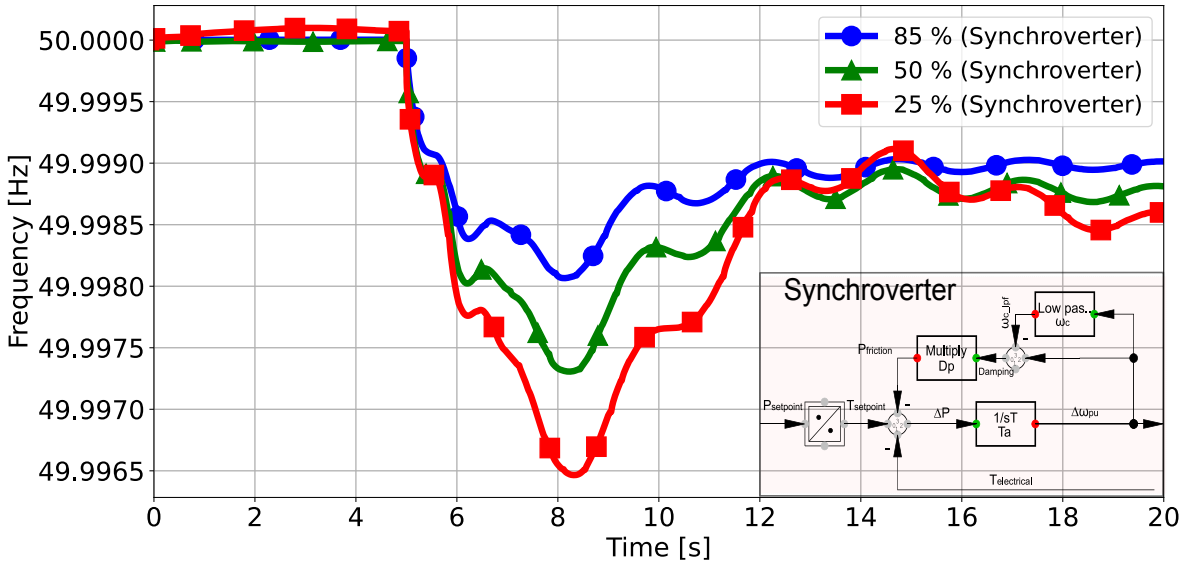


Figure B.23: Frequency response for different grid-forming penetration levels for the Synchroverter controller during an outage event occurring at 5 seconds. The Synchroverter frequency control loop is highlighted in pink.

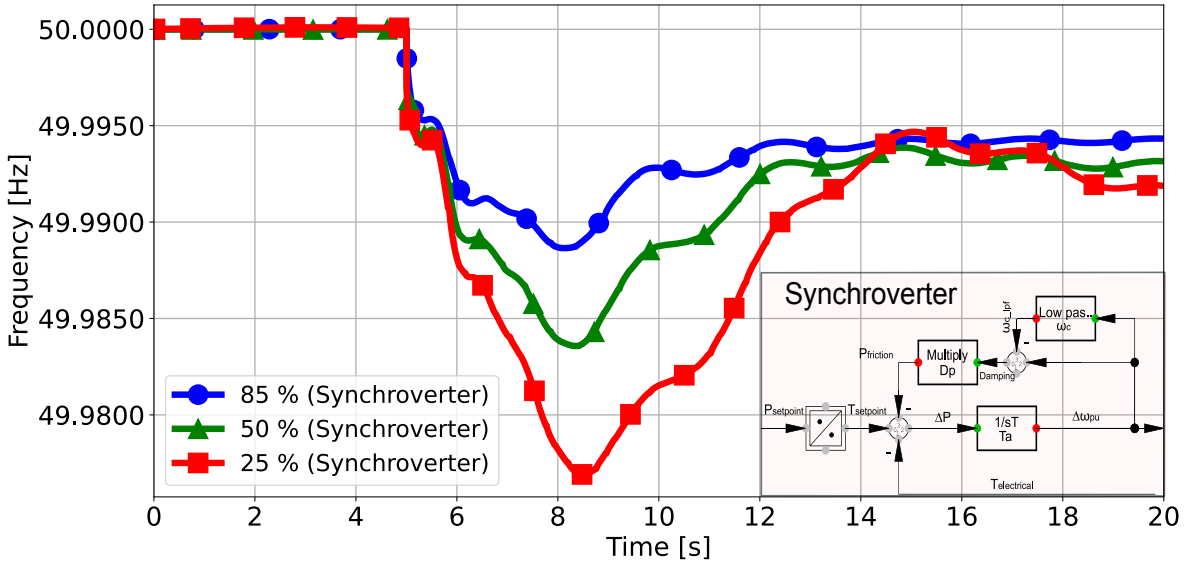


Figure B.24: Frequency response for different grid-forming penetration levels for the Synchroverter controller during a load event occurring at 5 seconds, Synchroverter frequency control loop highlighted in red.

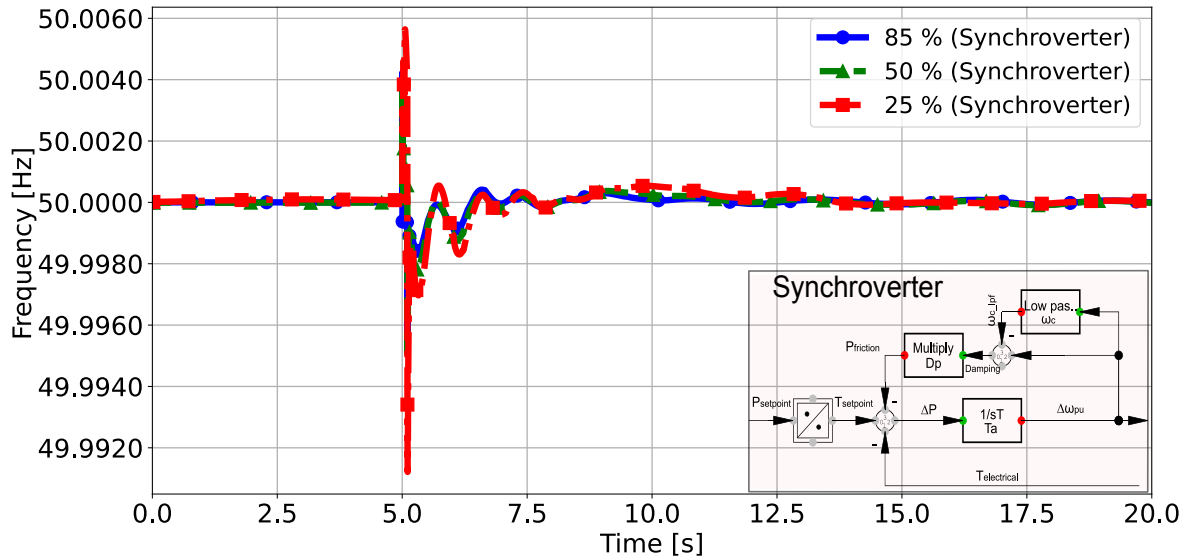


Figure B.25: Frequency response for different grid-forming penetration levels for the Synchroverter controller during a short-circuit event occurring at 5 seconds. The Synchroverter frequency control loop is highlighted in pink.

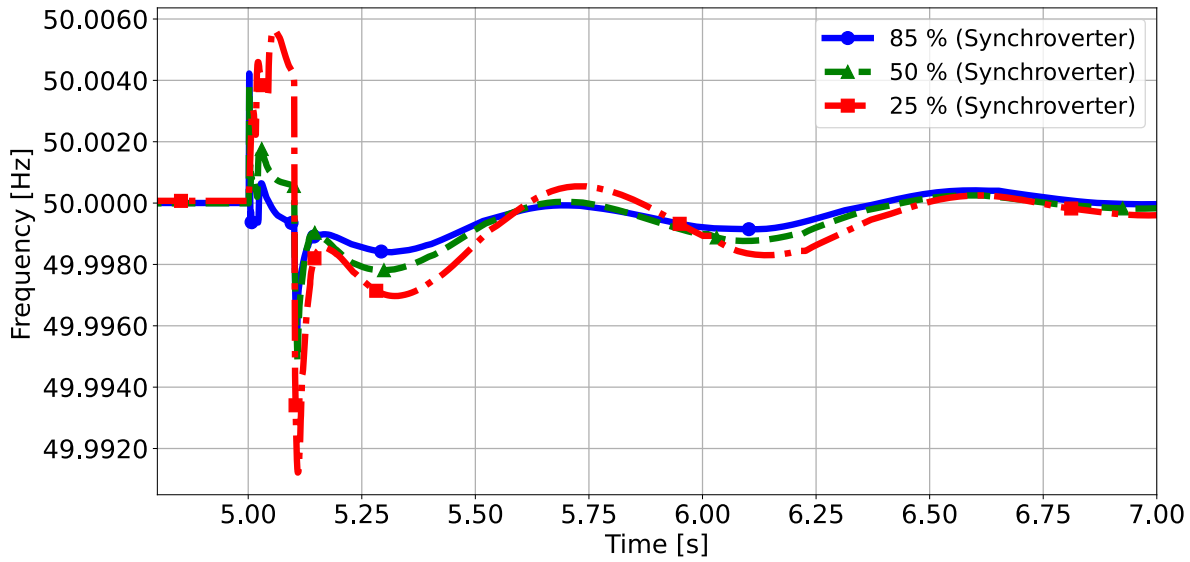


Figure B.26: Frequency response for different grid-forming penetration levels for the Synchroverter controller during a short-circuit event occurring at 5 seconds (zoomed).

Table B.5: ROCOF, Frequency deviation, and Frequency Nadir for different grid-forming penetration levels with the Synchroverter controller.

Outage Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85 %	49.9981	1.0	0.8	1.9
50 %	49.9973	1.2	1.0	2.7
25 %	49.9965	1.5	1.1	3.5

Short-Circuit Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85 %	49.9958	29.3	7.9	4.2
50 %	49.9949	33.1	9.8	5.1
25 %	49.9912	68.4	17.3	8.8

Load Event				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85 %	49.9886	5.3	3.9	11.4
50 %	49.9836	6.2	4.8	16.4
25 %	49.9769	7.3	5.9	23.1

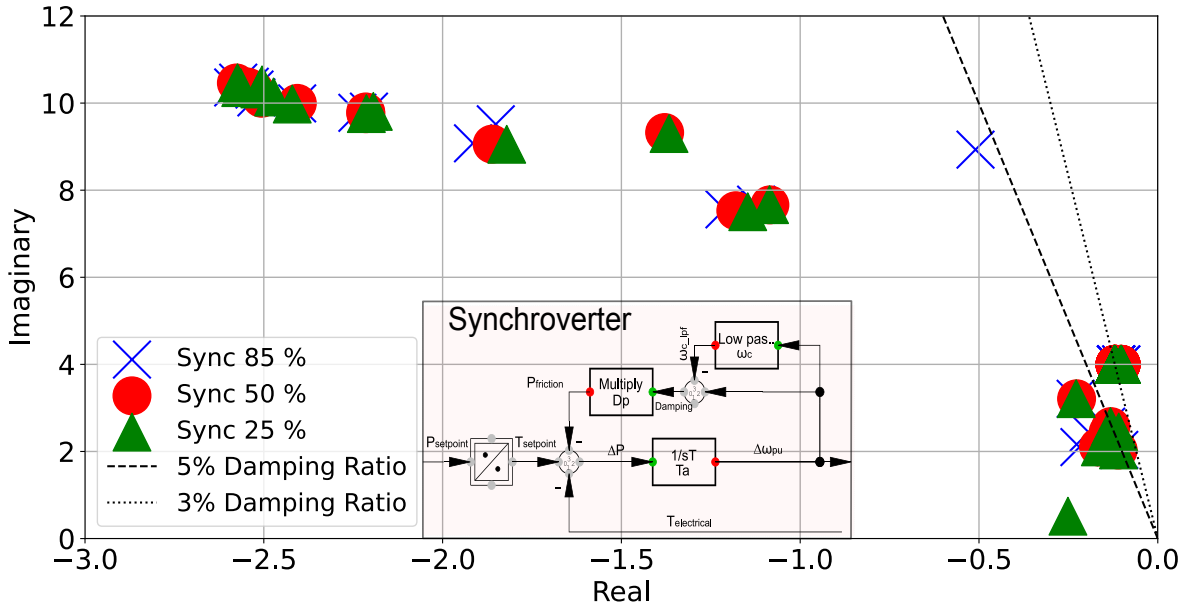


Figure B.27: Eigenvalue plot comparing the three grid-forming penetration levels for the Synchroverter controller, all modes under 50% damping ratio included. The Synchroverter frequency control loop is highlighted in pink.

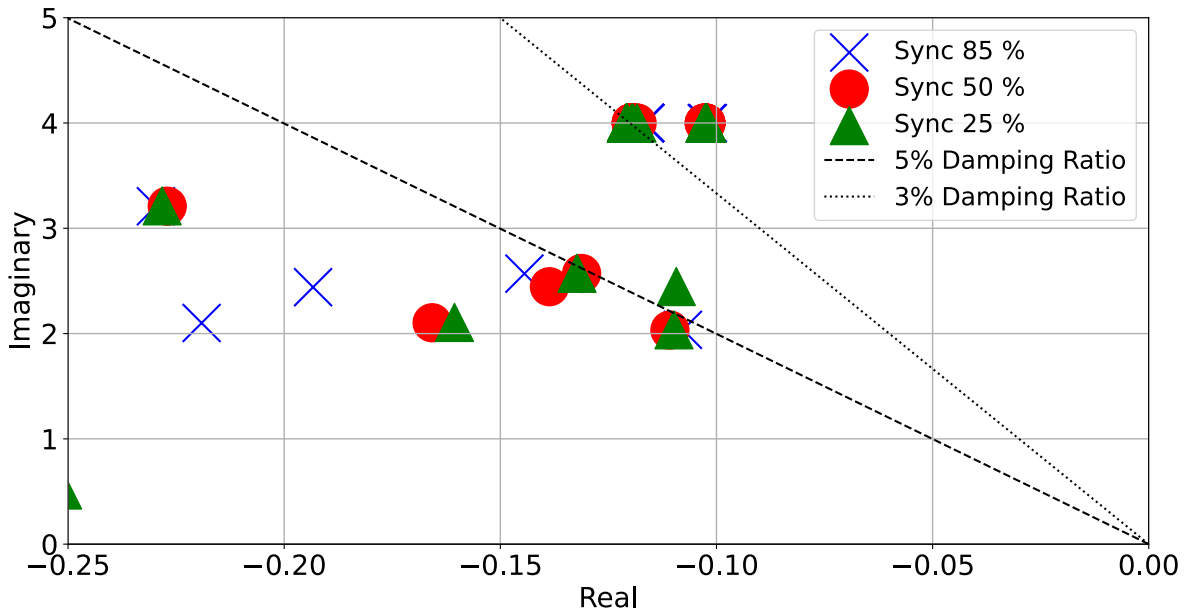


Figure B.28: Eigenvalue plot comparing the three grid-forming penetration levels for the Synchroverter controller, zoomed on the most critical modes.

Table B.6: Comparison of Eigenvalues, Frequencies, and Damping Ratios for different grid-forming penetration levels with the Synchroverter controller.

Mode index	Eigenvalue	Frequency (Hz)	Damping Ratio (%)		
			85%	50%	25%
113	$-0.1024 - 3.9989i$	0.6364	2.5485	2.5610	2.5565
172	$-0.1197 + 3.9996i$	0.6366	2.9124	2.9906	3.0190
173	$-0.1312 - 2.5708i$	0.4092	5.6125	5.0967	5.1348
137	$-0.1108 - 2.0331i$	0.3236	5.2858	5.4402	5.3877
175	$-0.1386 - 2.4448i$	0.3891	7.8969	5.6618	4.4625
243	$-0.2271 - 3.2097i$	0.5108	7.1380	7.0563	7.0852
182	$-0.1658 - 2.1029i$	0.3347	10.3748	7.8587	7.6049
130	$-0.5097 - 8.9315i$	1.4215	5.6974	14.6317	14.5388

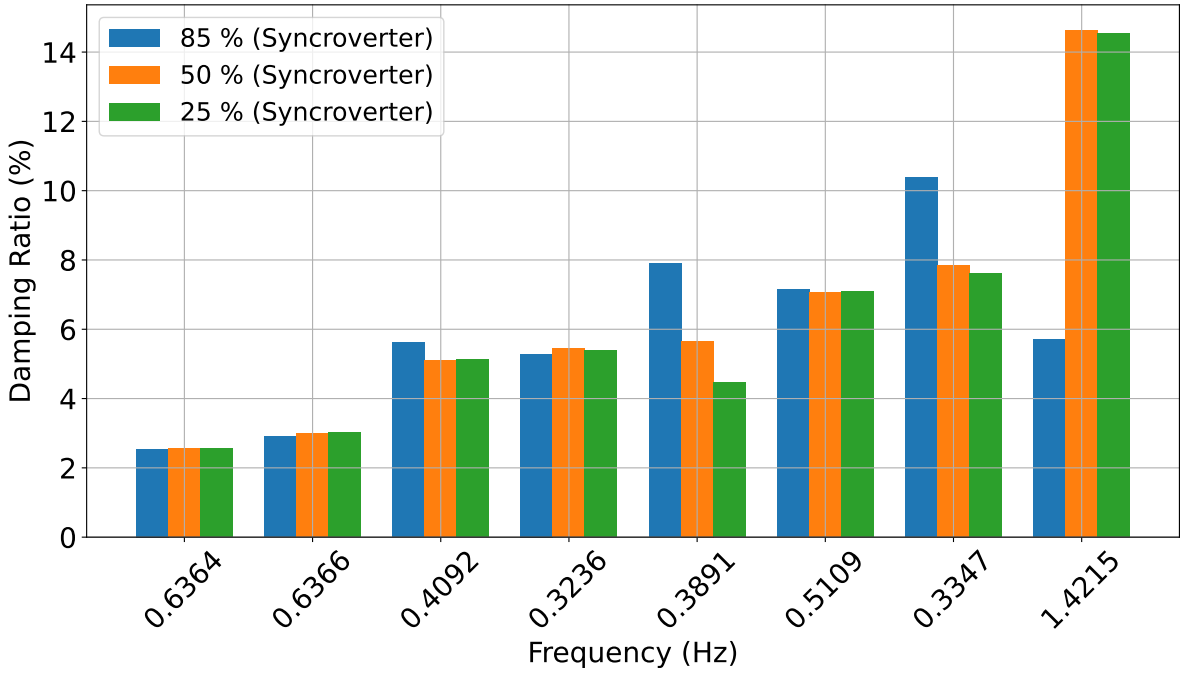


Figure B.29: Damping ratio for each individual mode-related frequency for the three levels of grid-forming penetration with the Synchroverter controller.

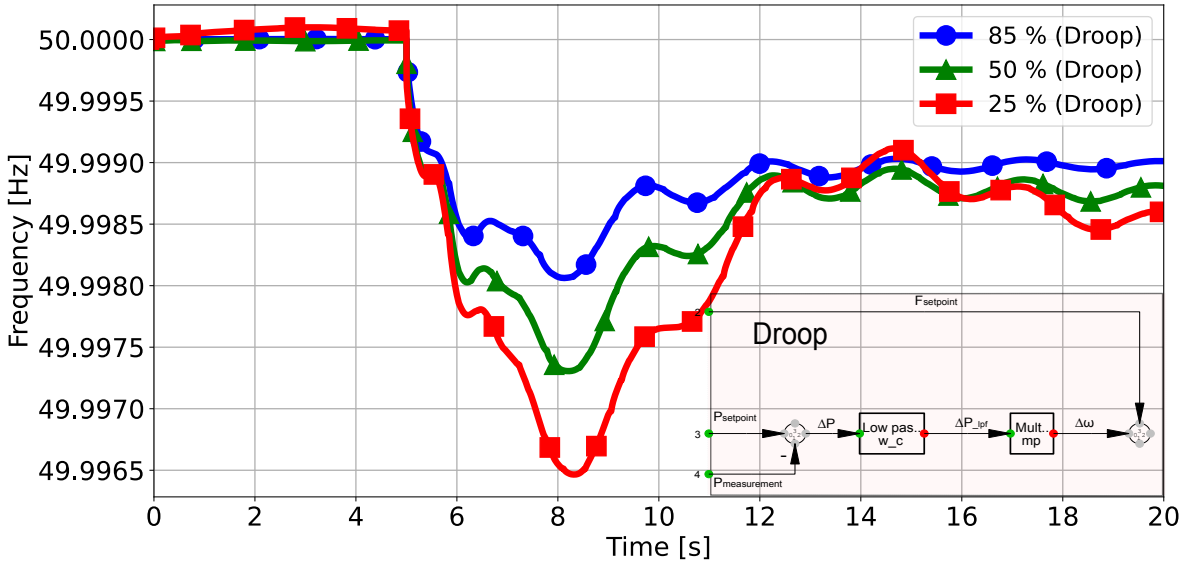


Figure B.30: Frequency response for different grid-forming penetration levels with the Droop controller during an outage event occurring at 5 seconds. The Droop frequency control loop is highlighted in pink.

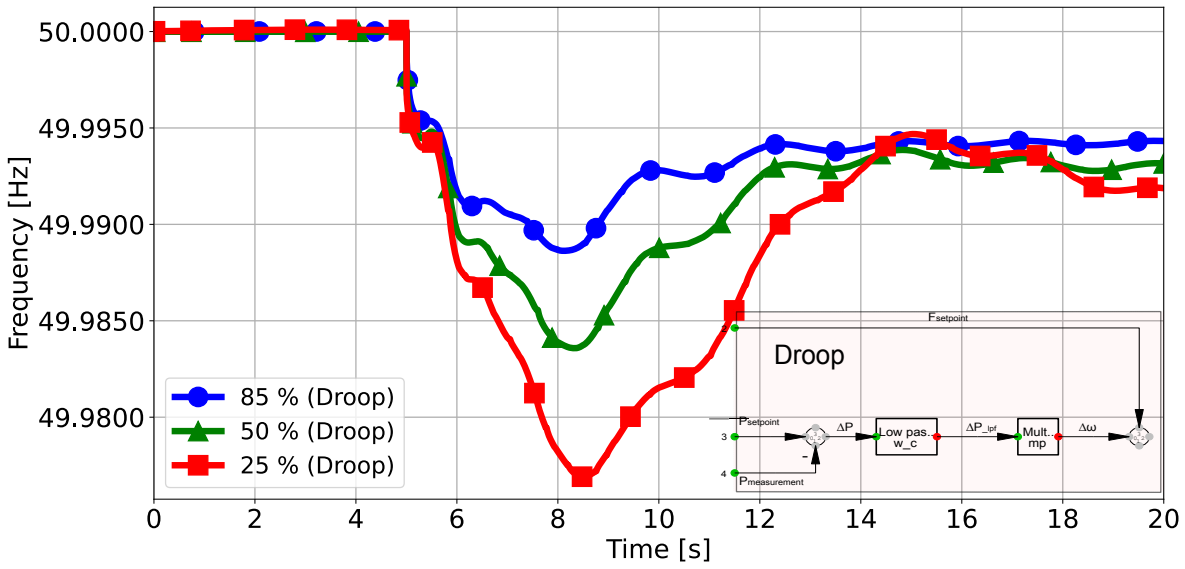


Figure B.31: Frequency response for different grid-forming penetration levels with the Droop controller during a load event occurring at 5 seconds. The Droop frequency control loop is highlighted in pink.

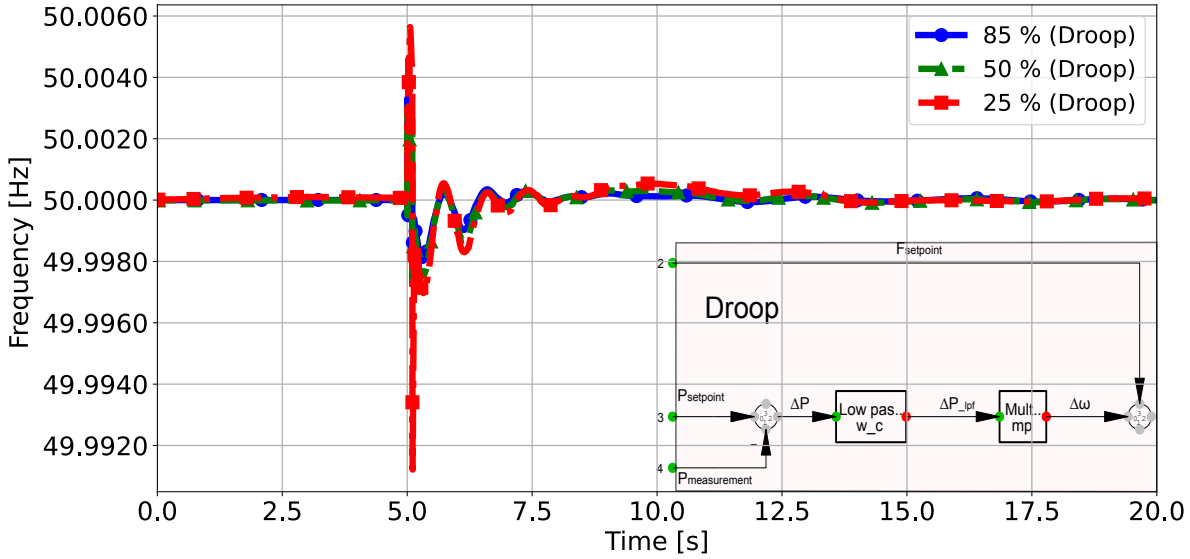


Figure B.32: Frequency response for different grid-forming penetration levels with the Droop controller during a short-circuit event occurring at 5 seconds. The Droop frequency control loop is highlighted in pink.

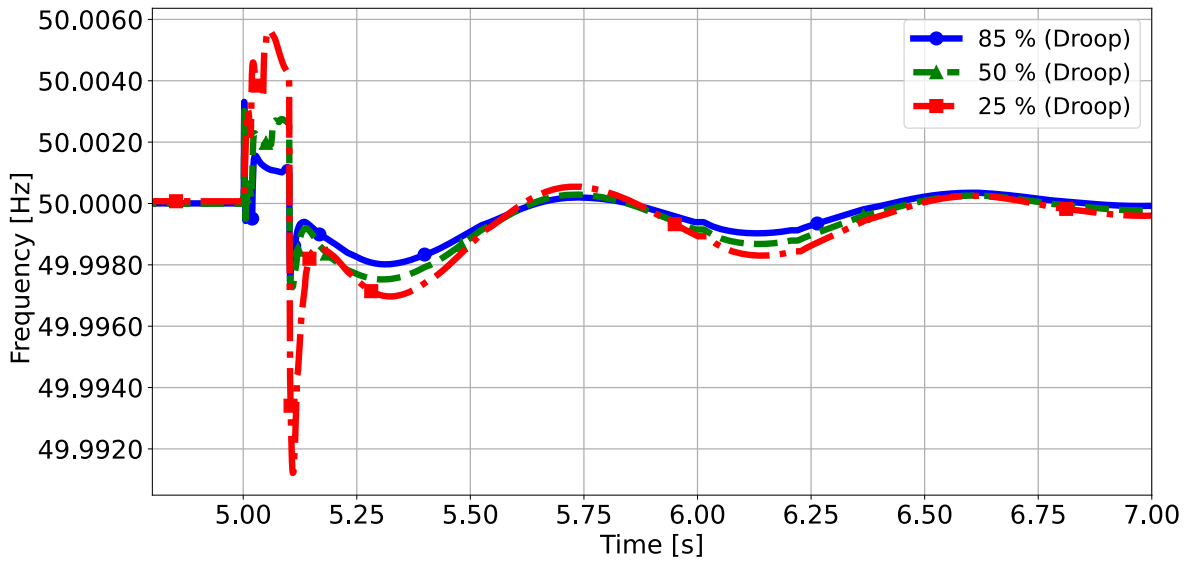


Figure B.33: Frequency response for different grid-forming penetration levels with the Droop controller during a short-circuit event occurring at 5 seconds (zoomed).

Table B.7: ROCOF, Frequency deviation, and Frequency Nadir for different levels of grid-forming penetration with the Droop controller.

Outage Event				
Case Index	Frequency Nadir (Hz)	ROCOF (Hz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85%	49.9981	0.0009	0.0007	1.9
50%	49.9973	0.0014	0.001	2.7
25%	49.9965	0.0015	0.0011	3.5

Short-Circuit Event				
Case Index	Frequency Nadir (Hz)	ROCOF (Hz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85%	49.9967	0.0138	0.0046	3.3
50%	49.9970	0.0266	0.0055	3.0
25%	49.9912	0.0684	0.0173	8.8

Load Event				
Case Index	Frequency Nadir (Hz)	ROCOF (Hz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
85%	49.9886	0.0051	0.0038	11.4
50%	49.9836	0.0058	0.0047	16.4
25%	49.9769	0.0073	0.0059	23.1

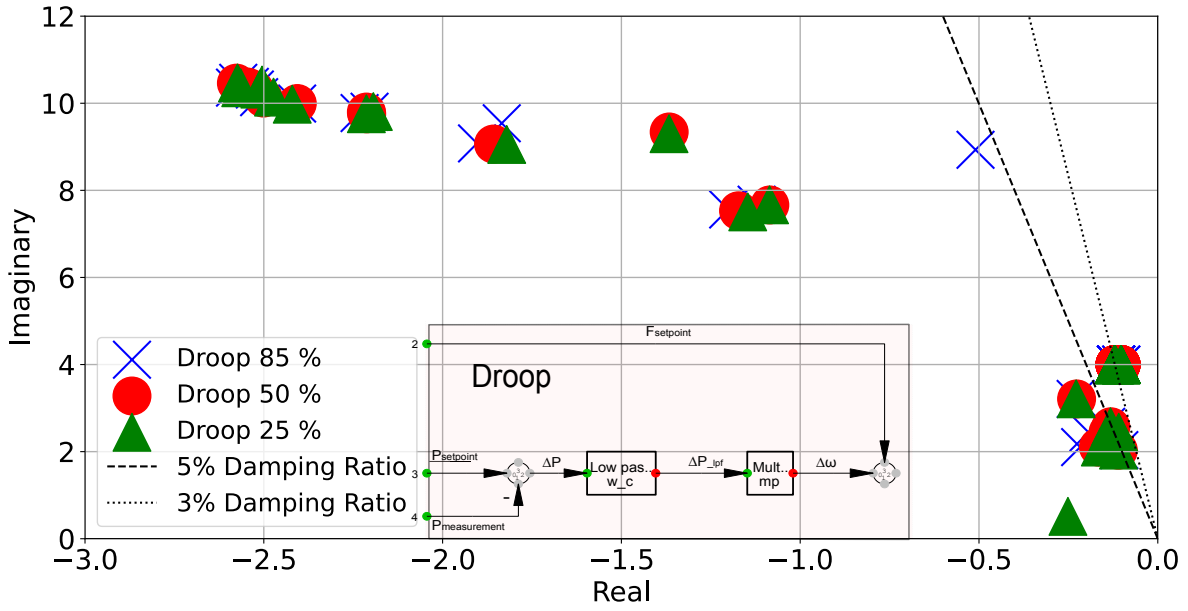


Figure B.34: Eigenvalue plot comparing the three grid-forming penetration levels for the Droop controller, all modes under 50% damping ratio included. The Droop frequency control loop is highlighted in pink.

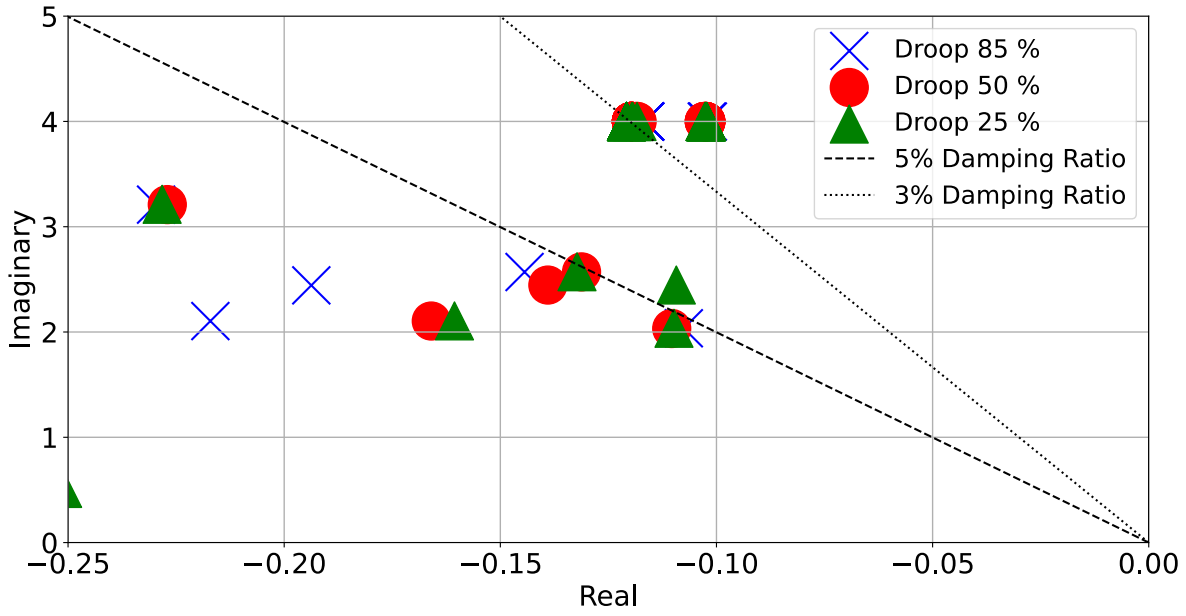


Figure B.35: Eigenvalue plot comparing the three grid-forming penetration levels for the Droop controller, zoomed on the most critical modes.

Table B.8: Comparison of Eigenvalues, Frequencies, and Damping Ratios for different grid-forming penetration levels with the Droop controller.

Mode index	Eigenvalue	Frequency (Hz)	Damping Ratio (%)		
			85%	50%	25%
62	$-0.1024 - 3.9989i$	0.6364	2.5485	2.5610	2.5565
121	$-0.1197 + 3.9996i$	0.6366	2.9125	2.9906	3.0190
122	$-0.1312 - 2.5710i$	0.4092	5.6103	5.0969	5.1348
86	$-0.1104 - 2.0331i$	0.3236	5.2768	5.4198	5.3877
124	$-0.1390 - 2.4458i$	0.3893	7.9045	5.6742	4.4625
192	$-0.2270 - 3.2100i$	0.5109	7.1360	7.0554	7.0852
129	$-0.1660 - 2.1042i$	0.3349	10.2634	7.8632	7.6049
130	$-0.5097 - 8.9315i$	1.4215	5.6974	14.4747	14.5388

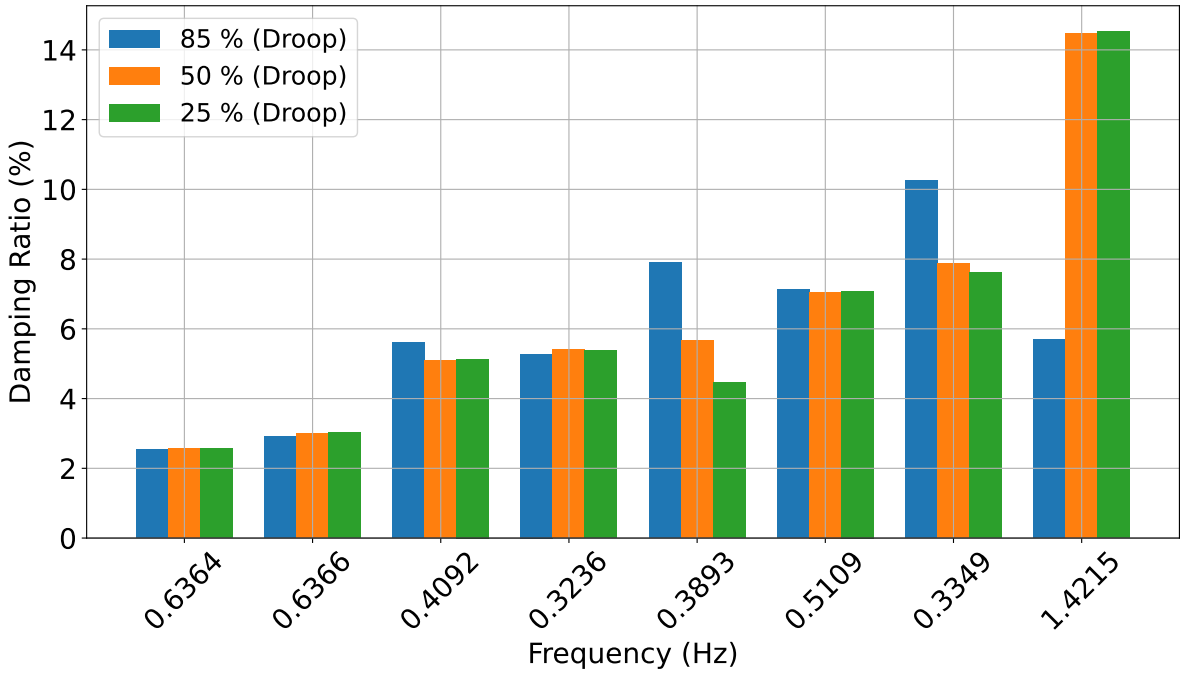


Figure B.36: Damping ratio for each individual mode-related frequency for the three levels of grid-forming penetration with the Droop controller.

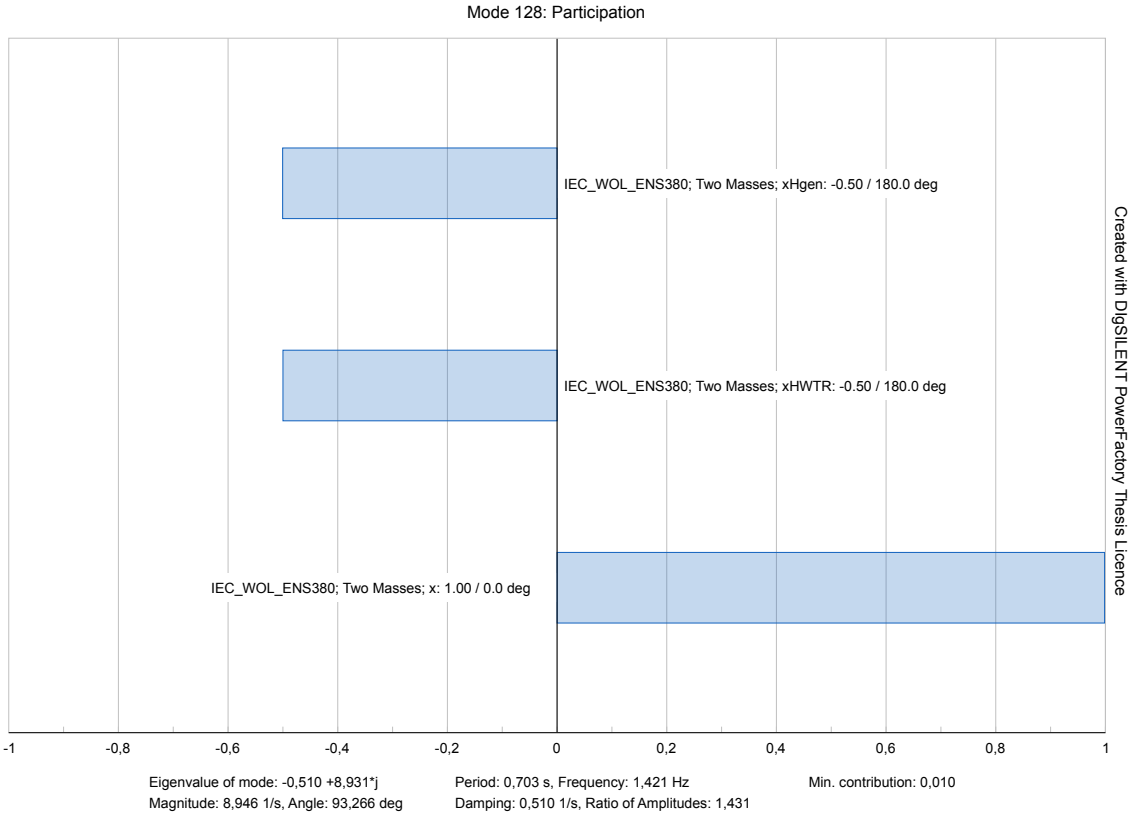


Figure B.37: Overview of participation factors for mode 128.

B.2.1.2. Controller Parameter Sensitivity Analysis
B.2.1.3. Synchroverter Results

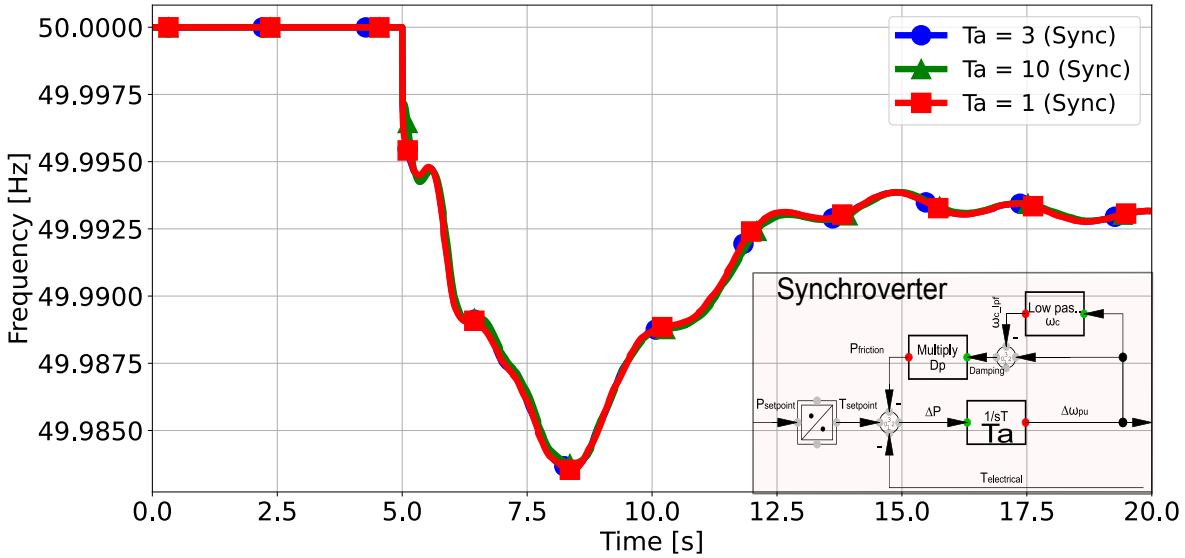


Figure B.38: Frequency response for different T_a values for the Synchroverter controller. The Synchroverter frequency control loop is highlighted in pink.

Table B.9: ROCOF, Frequency deviation, and Frequency Nadir for different values of T_a for the Synchroverter controller.

T_a (Synchroverter)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
$T_a = 3$	49.9836	6.2	4.8	16.4
$T_a = 10$	49.9837	6.4	4.7	16.3
$T_a = 1$	49.9837	6.4	4.7	16.3

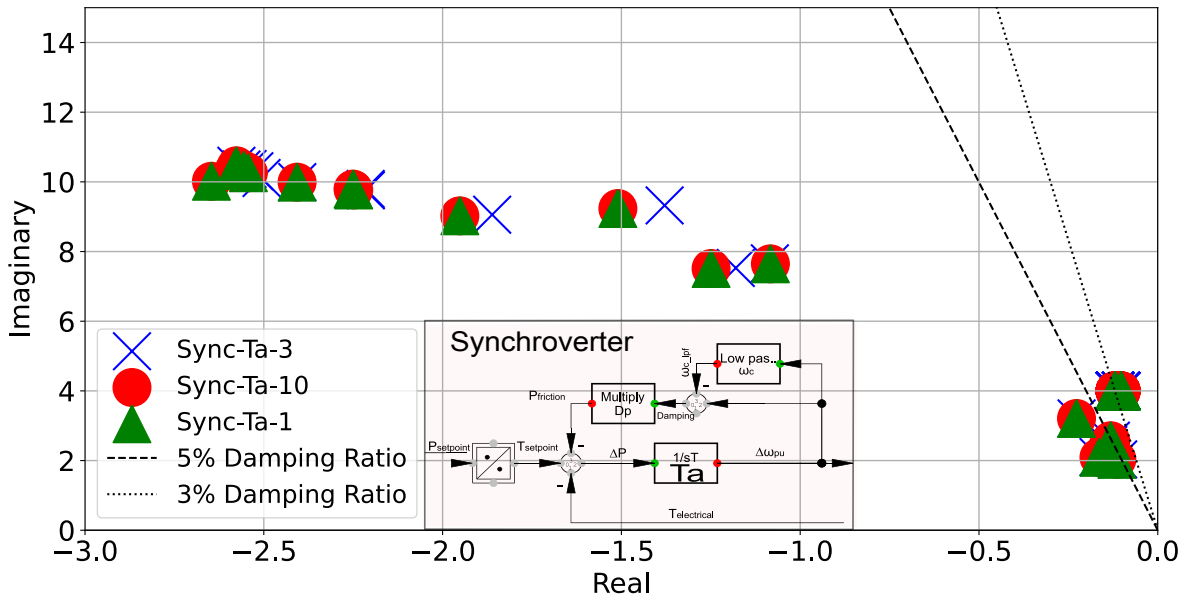


Figure B.39: Eigenvalues for different T_a values for the Synchroverter controller. The Synchroverter frequency control loop is highlighted in pink.

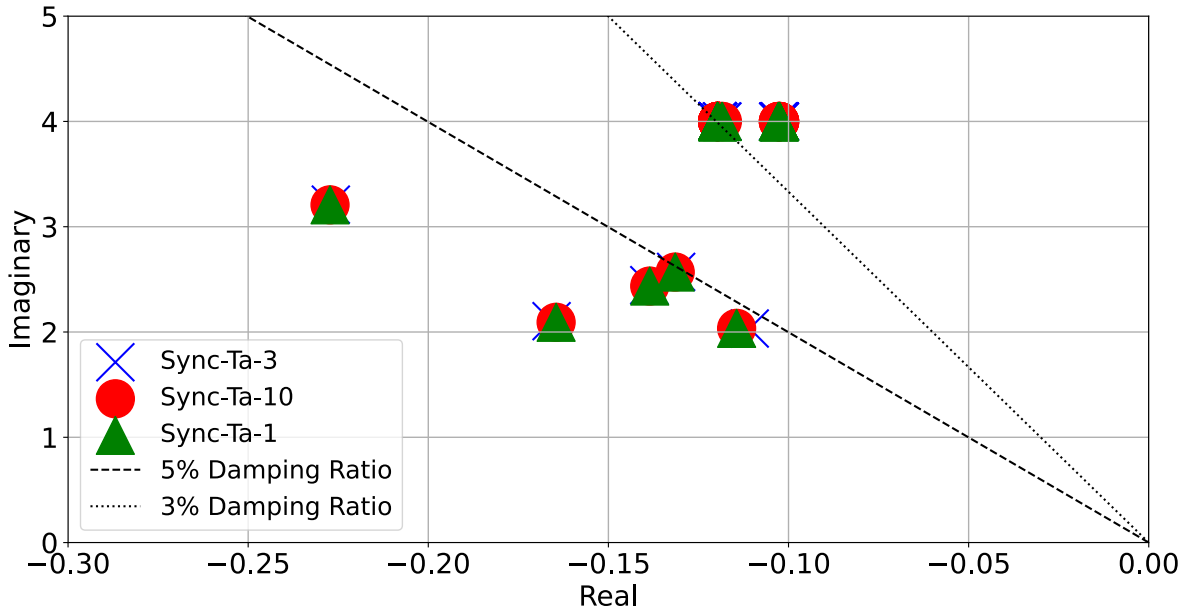


Figure B.40: Eigenvalues for different T_a values for the Synchroverter controller, zoomed on the most critical modes.

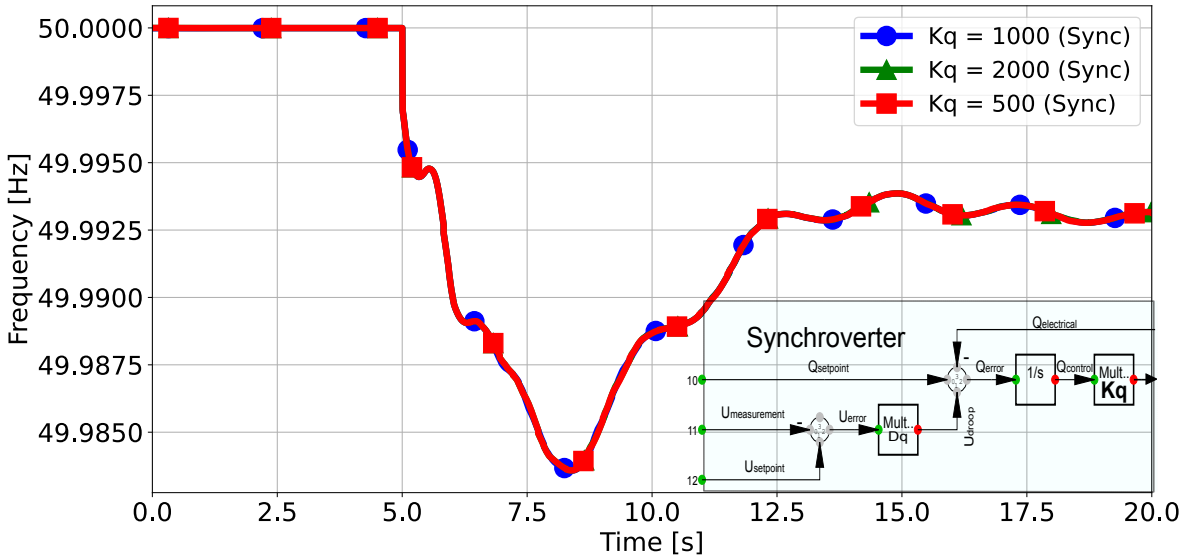


Figure B.41: Frequency response for different K_q values for the Synchroverter controller. Synchroverter voltage control loop highlighted in blue.

Table B.10: ROCOF, Frequency deviation, and Frequency Nadir for different values of K_q for the Synchroverter controller.

K_q (Synchroverter)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
$K_q = 1000$	49.9836	6.2	4.8	16.4
$K_q = 2000$	49.9836	5.8	4.5	16.4
$K_q = 500$	49.9836	5.9	4.5	16.4

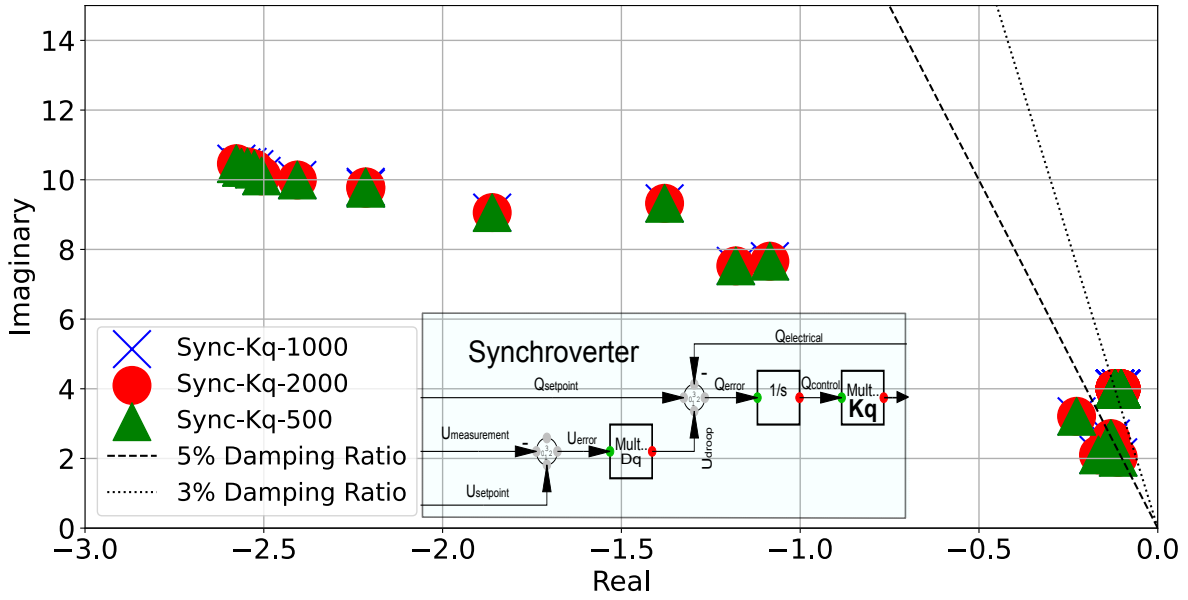


Figure B.42: Eigenvalues for different K_q values for the Synchroverter controller. Synchroverter voltage control loop highlighted in blue.

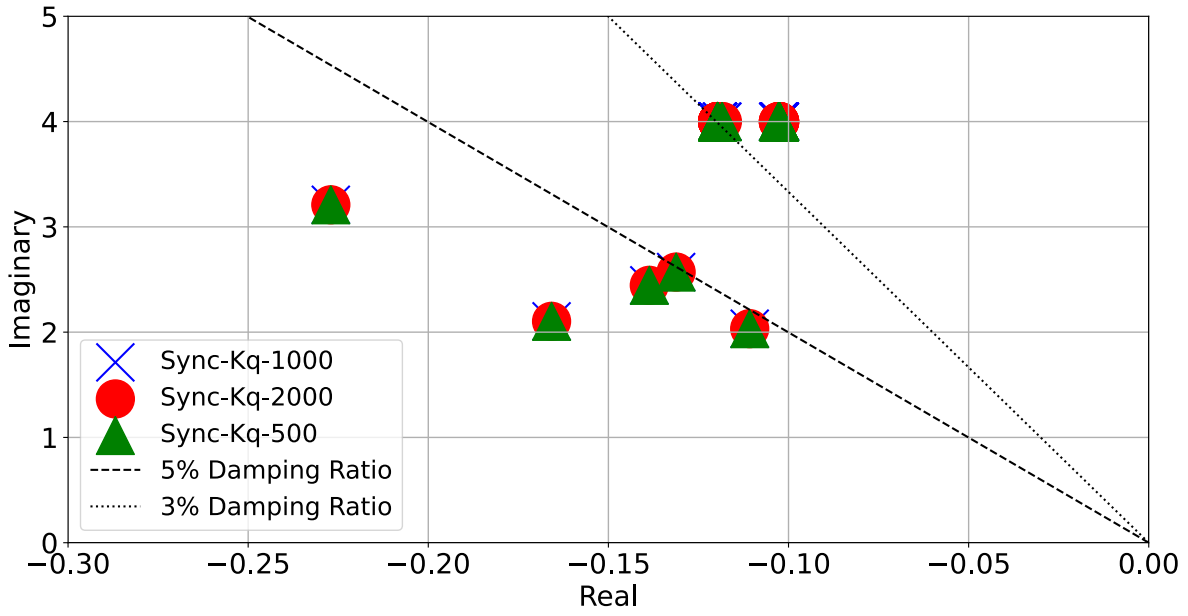


Figure B.43: Eigenvalues for different K_q values for the Synchroverter controller, zoomed on the most critical modes.

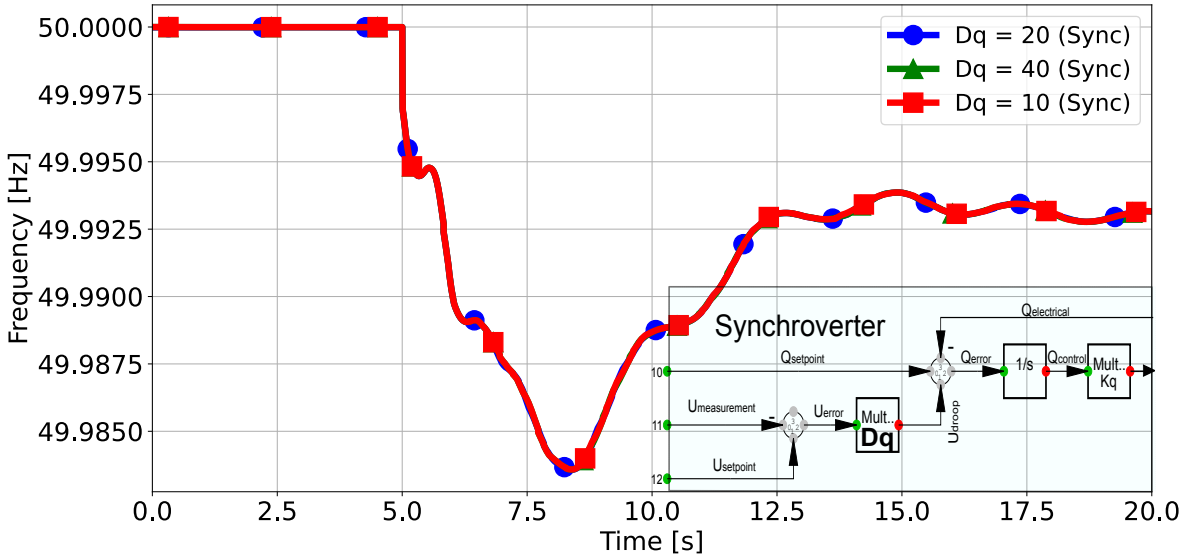


Figure B.44: Frequency response for different D_q values for the Synchroverter controller. Synchroverter voltage control loop highlighted in blue.

Table B.11: ROCOF, Frequency deviation, and Frequency Nadir for different values of D_q for the Synchroverter controller.

D_q (Synchroverter)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
$D_q = 20$	49.9836	6.2	4.8	16.4
$D_q = 40$	49.9836	5.9	4.5	16.4
$D_q = 10$	49.9836	5.7	4.6	16.4

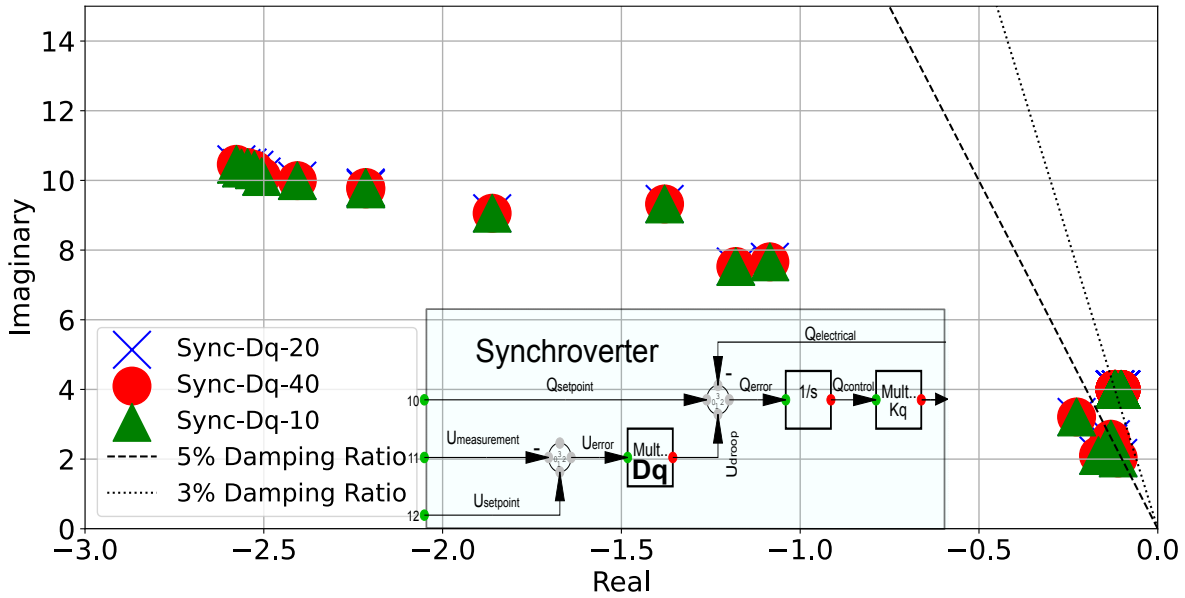


Figure B.45: Eigenvalues for different D_q values for the Synchroverter controller.

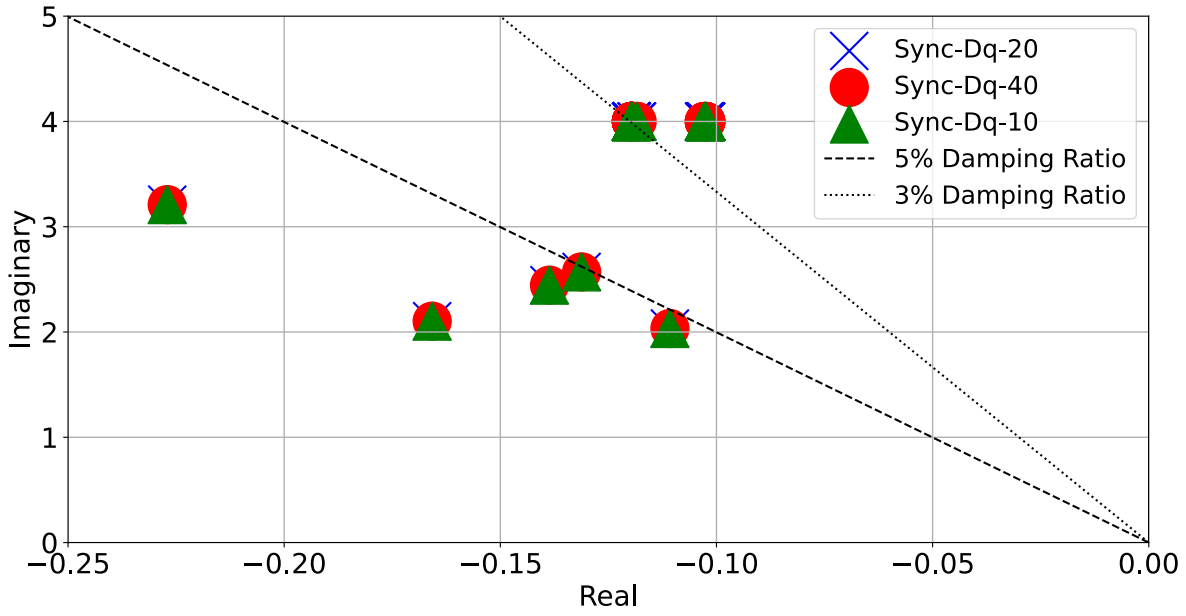


Figure B.46: Eigenvalues for different D_q values for the Synchroverter controller, zoomed on the most critical modes.

B.2.1.4. Droop Controller

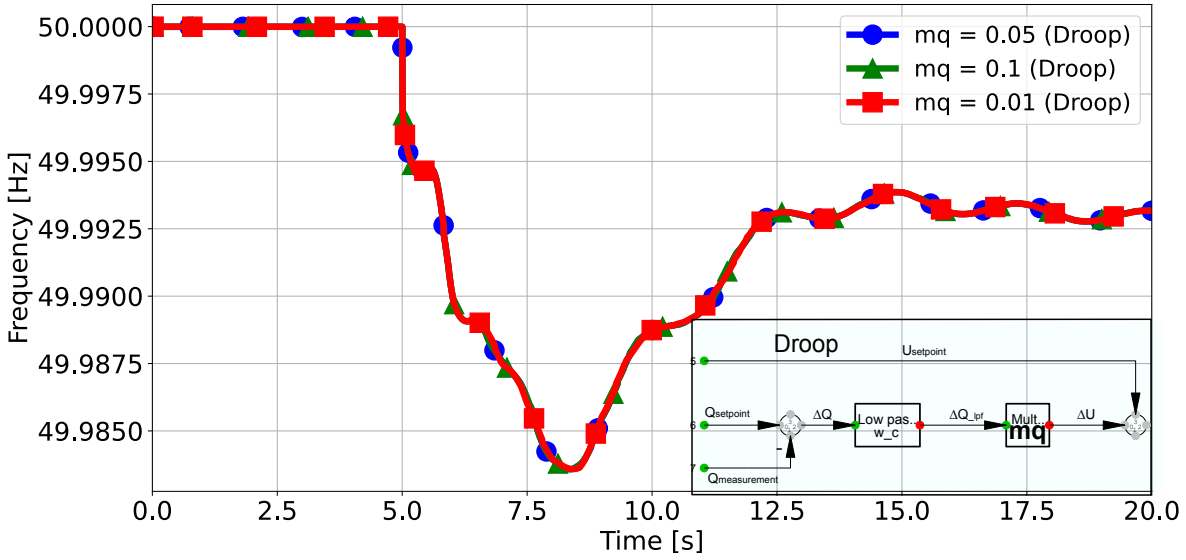


Figure B.47: Frequency response for different m_q values for the Droop controller. Droop voltage control loop highlighted in blue.

Table B.12: ROCOF, Frequency deviation, and Frequency Nadir for different values of m_q for the Droop controller.

m_q (Droop)				
Case Index	Frequency Nadir (Hz)	ROCOF (mHz/s)		Frequency Deviation (mHz)
		100 ms	500 ms	
$m_q = 0.05$	49.9836	5.8	4.7	16.4
$m_q = 0.1$	49.9836	5.8	4.7	16.4
$m_q = 0.01$	49.9836	5.9	4.7	16.4

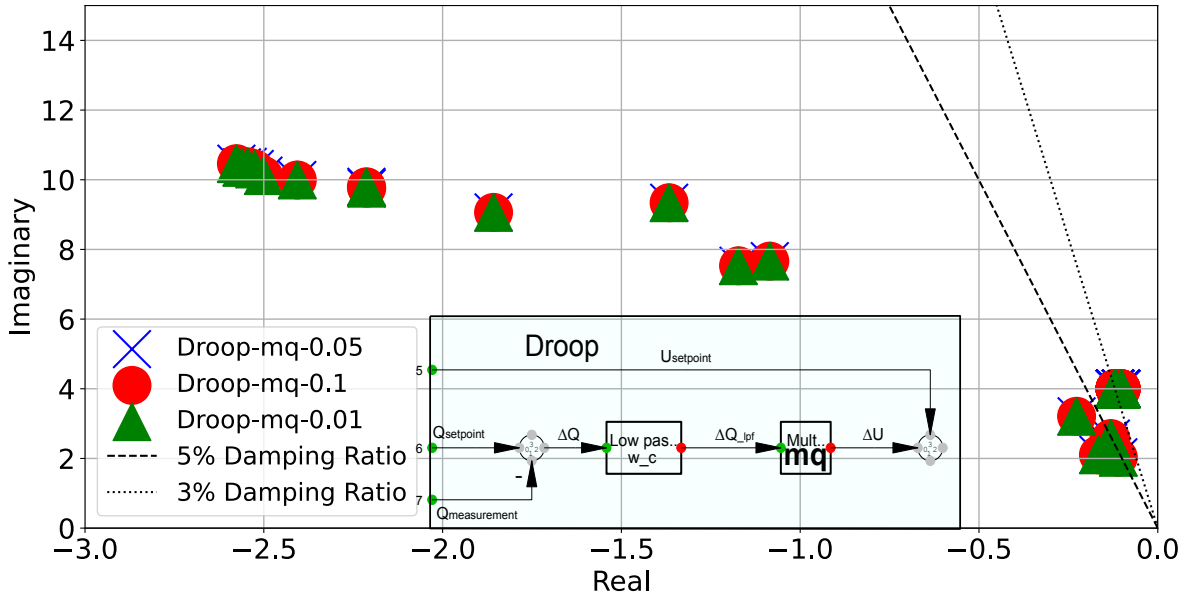


Figure B.48: Eigenvalues for different m_q values for the Droop controller. Droop voltage control loop highlighted in blue.

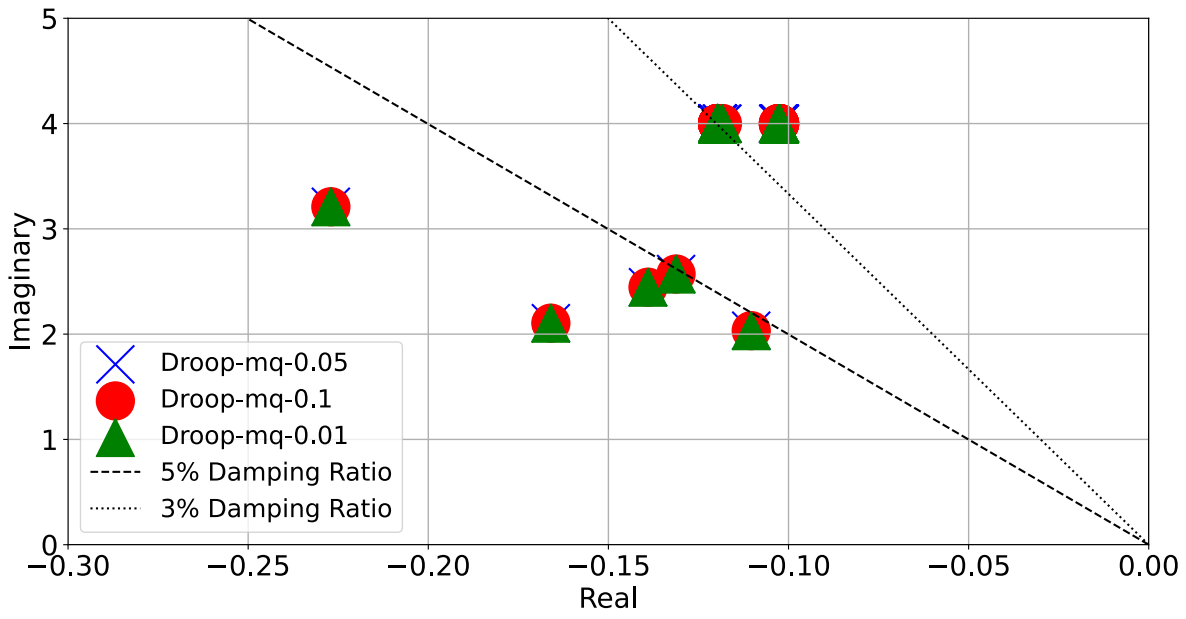


Figure B.49: Eigenvalues for different m_q values for the Droop controller, zoomed on the most critical modes.

B.3. Case Study C

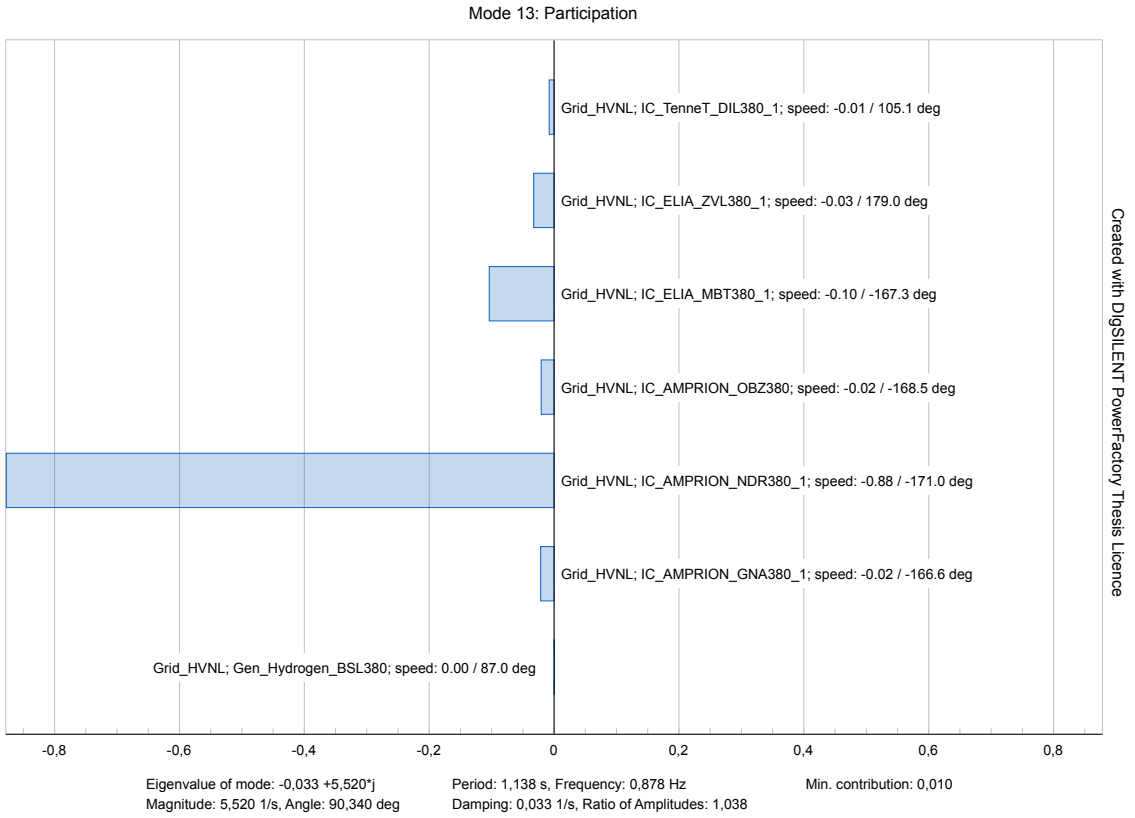


Figure B.50: Overview of participation factors for mode 13

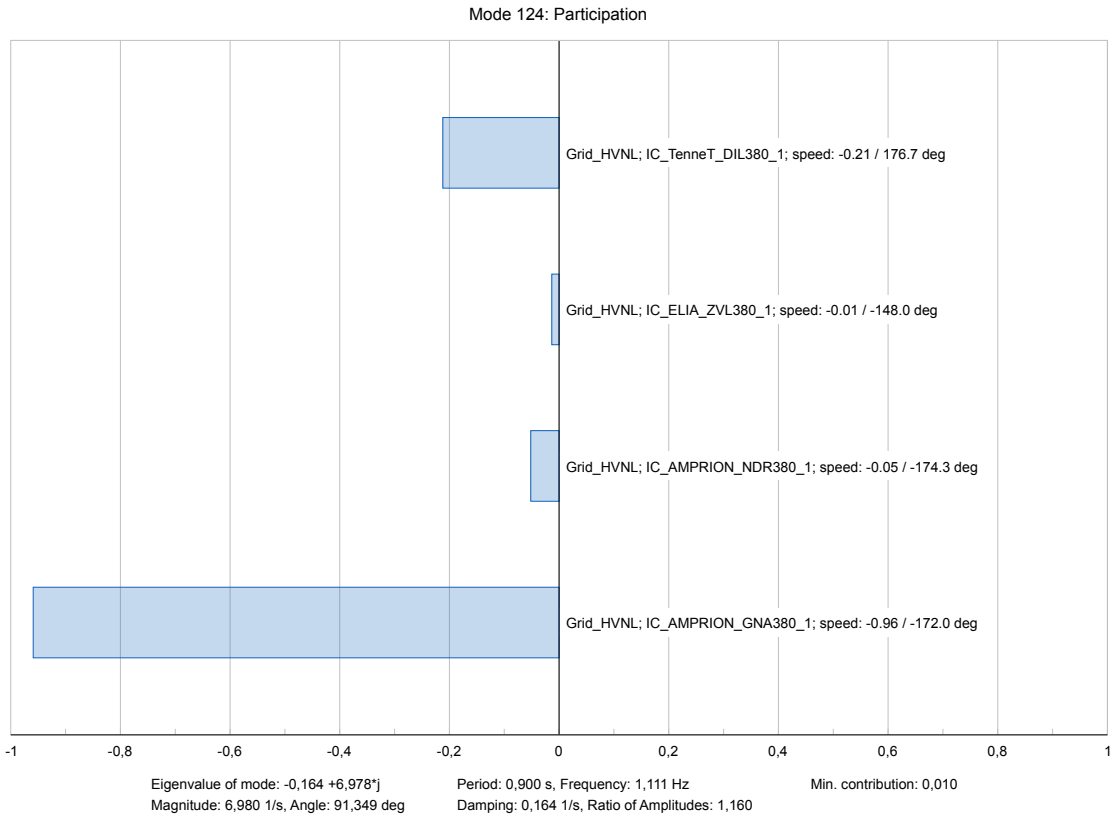


Figure B.51: Overview of participation factors for mode 124.

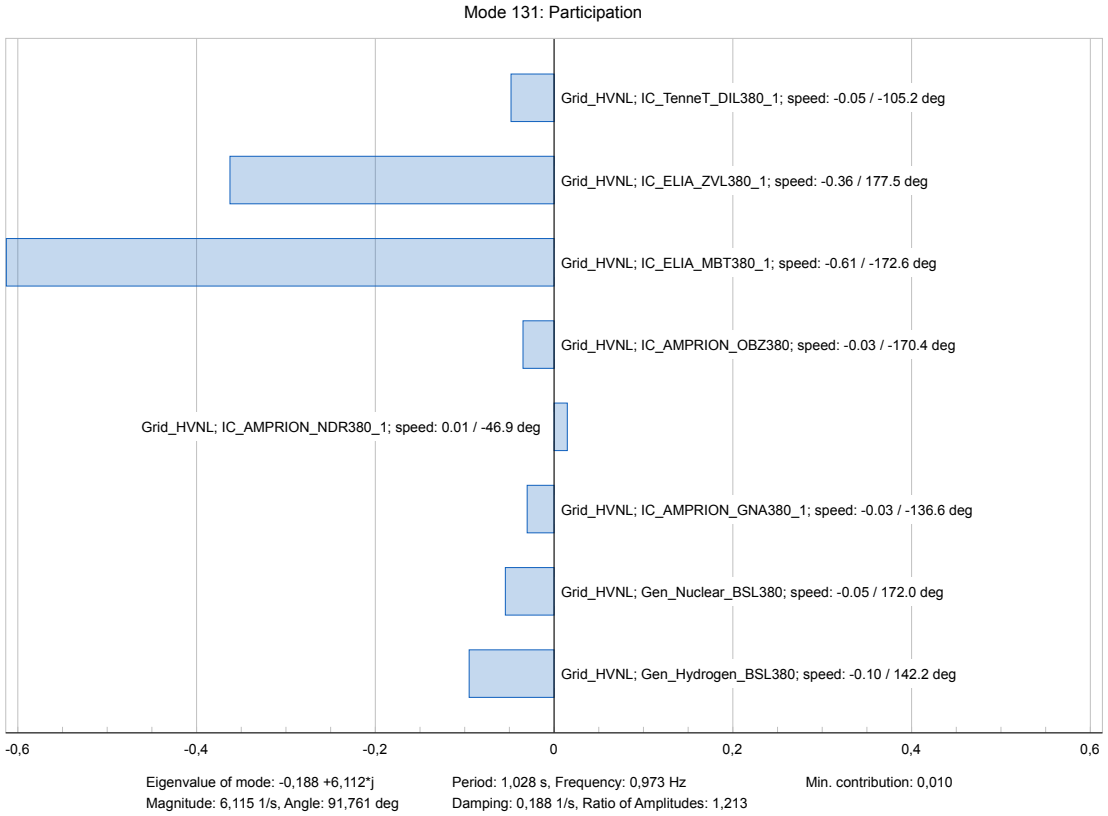


Figure B.52: Overview of participation factors for mode 131

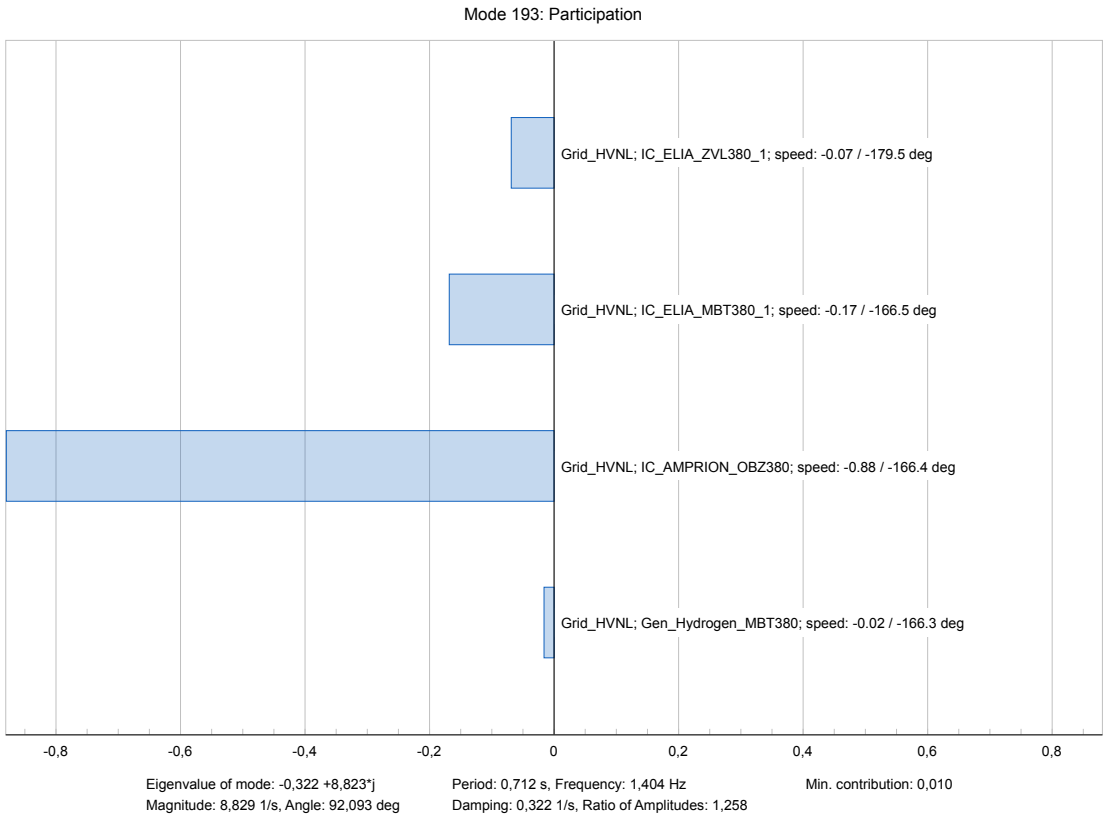


Figure B.53: Overview of participation factors for mode 193.

C

Software Implementation and Guide

C.1. Introduction

This appendix provides a comprehensive explanation of the Python and Excel resources utilized as ancillary services for the Synthetic Dynamic power system Model, which has been upgraded in chapter 5 and 6. The integration involves two primary Python scripts that work in conjunction with two Excel spreadsheets. The Python scripts are responsible for calculating and distributing the dispatches of various sources and implementing these calculations into PowerFactory. The Excel spreadsheets offer a clear overview of the installed capacities, facilitate the placement of various dispatches, and summarize system parameters for further integration into Python.

Initially, the setup process for these resources will be outlined, followed by a detailed explanation of the codes and their functionalities. This structure aims to ensure seamless integration into complementary work, thus making the codes and model a vital resource for advancing related projects.

Some aspects of the work in Excel, Python, and PowerFactory are based on prior research and have been extended to better suit this assignment. This foundational work is primarily documented in [72], hereafter referred to as "Midhuna." The relevant sections will detail the contributions from this source and the modifications made.

All models used within the PowerFactory model are based on generic representations provided as templates within PowerFactory. For more detailed information on these models, please refer to the PowerFactory technical reference.

Case A should already be implemented with the proper operational characteristics in the synthetic model. However, the updates highlighted in the following sections must be made if the different cases are to be used.
A flowchart depicting the general process of integrating Python, Excel, and PowerFactory resources in the synthetic model can be seen in Figure C.1.

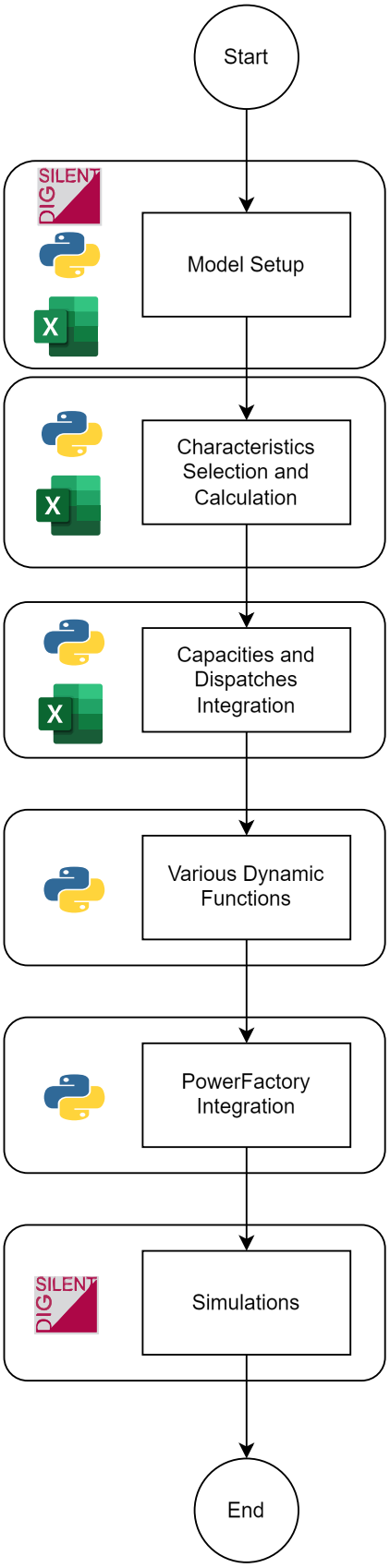


Figure C.1: Flowchart highlighting the Python and Excel setup and integration with PowerFactory.

C.2. Model Setup

This section provides the necessary information to set up the model in PowerFactory, Python, and Excel.

C.2.1. PowerFactory

Starting with the PowerFactory model, several important implementations are made to enable the use of the model and its ancillary scripts and Excel files. Here, the PowerFactory version "2023 SP6" is utilized. The same version should be used to ensure the smooth operation of the model.

The first step in the PowerFactory setup is importing the dynamic model from the zip file attached to the thesis. Navigate to file → import → .PFD file, as highlighted in the red square in Figure C.2. Then, select the PowerFactory PFD file from the thesis's zip folder.

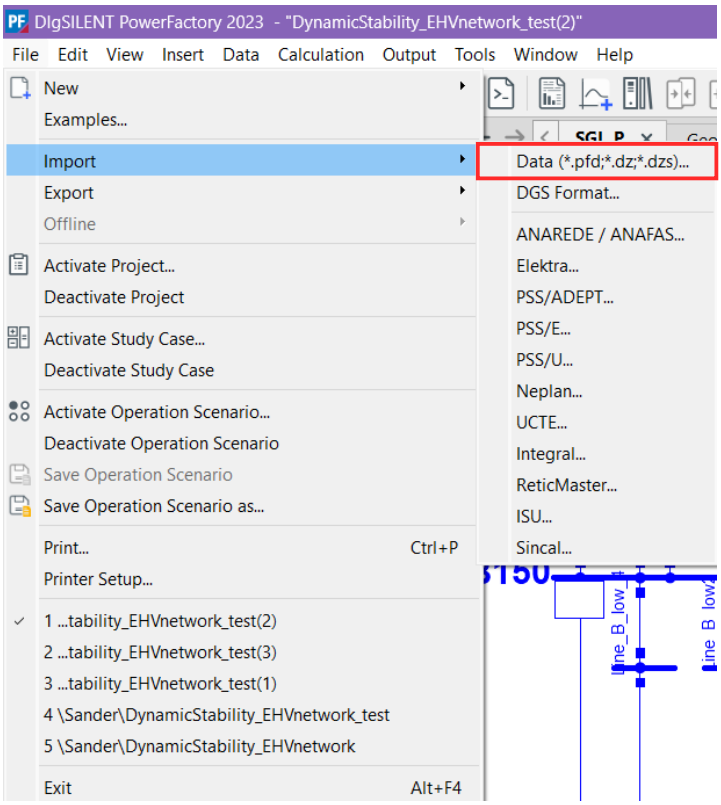


Figure C.2: Snapshot PowerFactory model import.

The second part of the setup involves ensuring that PowerFactory utilizes the correct Python version for external scripting. Begin by selecting Tools and Configuration, as indicated by the blue and pink squares in Figure C.3. This action will open the window shown in Figure C.4, where the external application option should be selected, as highlighted in the yellow square. Next, choose the appropriate Python version, indicated by the red square. For this work, Python version 3.11 is recommended. While different versions may be used, this can affect the functionality of certain features. Therefore, it is advisable to use Python version 3.11 or ensure that the necessary adjustments for other versions are made.

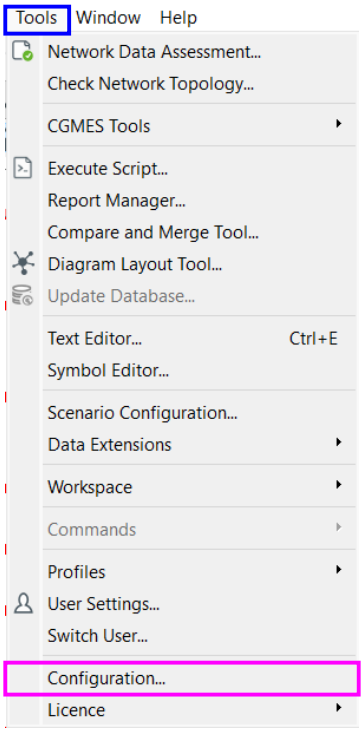


Figure C.3: Snapshot PowerFactory External Python Version setup.

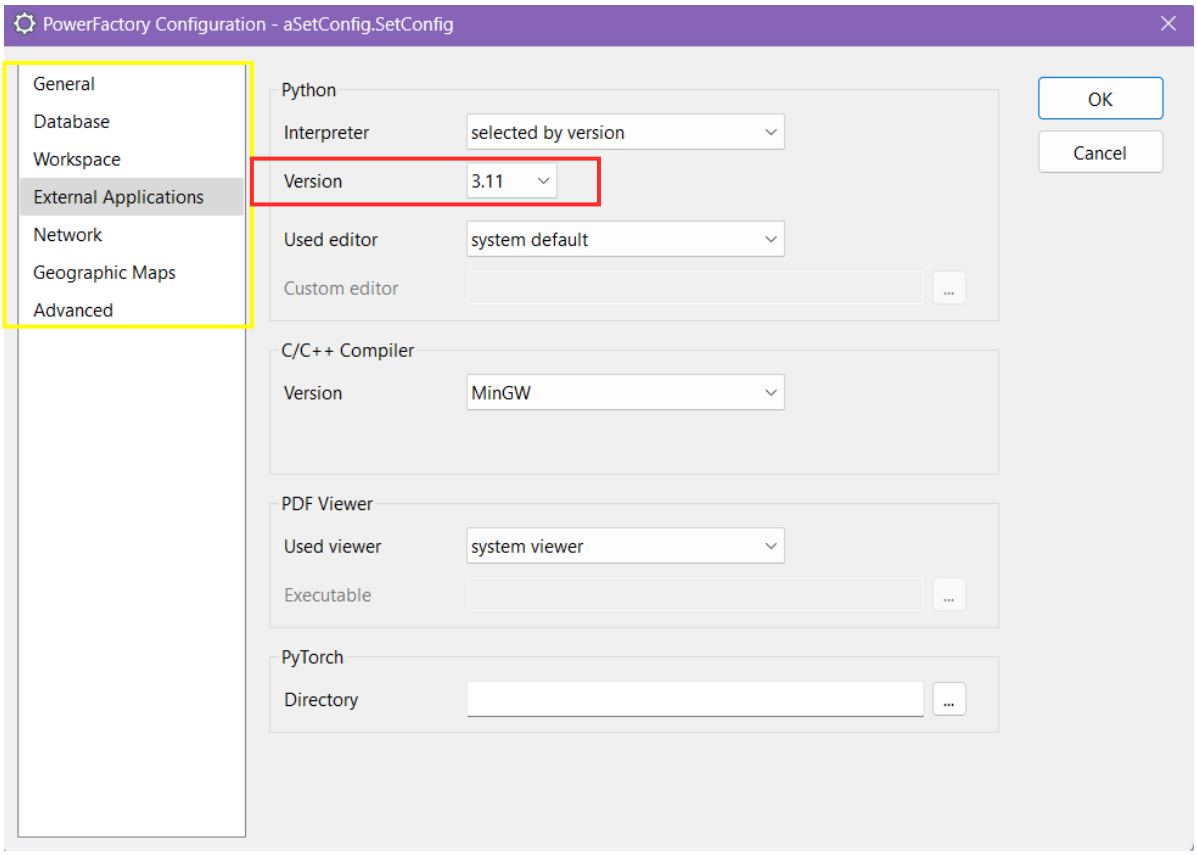


Figure C.4: Snapshot PowerFactory External Python Version setup.

C.2.2. Python Setup

As mentioned in the previous subsection, Python version 3.11 is recommended to ensure the model's proper operation.

The first part of the Python setup is to ensure the correct packages are installed. The primary package required is the "win32" package. The installation process for this package will depend on the environment in which Python is used. For a detailed installation guide, please refer to: Tutorial.

The overall setup changes necessary in Python are minimal. They involve primarily adapting folder paths and the name of the PowerFactory model.

The initialization of the two Python scripts, "Scenario-2050" and "Calculations," is depicted in Figure C.5 and Figure C.6, respectively. The initialization of the "Scenario-2050" script consists of two parts: integration with Excel and integration with PowerFactory. The setup for integration with Excel is shown in the red square. Three different Excel sheets are initialized: the input, installed capacities, and output files. The input file contains the dispatches, the installed capacities file includes the installed capacities and the output file is used for output purposes. Although no output is printed to the output file, it is included to ensure consistency with previous work by De Roos. All three Excel sheets must have the correct folder paths, which must be updated for the specific computer used.

A similar initialization for the dispatches file in Excel is required for the "Calculations" script, as shown in Figure C.6. This setup needs to be adjusted similarly.

In the blue square in Figure C.5, the integration with PowerFactory is shown. Line 31, highlighted in yellow, indicates the PowerFactory module and the respective version of Python for scripting. This must be updated to match the Python version used, as detailed in subsection C.2.1. Line 35, highlighted in the green square, represents the specific project in PowerFactory to be activated and used for simulations. This should correspond to the name of the ".PFD" file found in the zip file and should be adjusted accordingly. This setup facilitates easy integration with test systems for specific model work, as shown in the example.

```
16  #%% File paths for the relevant Excel files (Needs to be adjusted for other paths)
17
18  # change the following commands for the location of the excel files on your computer: input_excel, installedcap_excel and output_excel.
19  input_excel = r'G:\Andre datamaskiner\Min bærbare datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Scenario-Input-2-with-dispatched-case-b.xlsx'                    # This is the exce
20
21  # input_excel = r'G:\Andre datamaskiner\Min bærbare datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Scenario-Input-2.xlsx'                    # This is the excel file displaying mul
22  installedcap_excel = r'G:\Andre datamaskiner\Min bærbare datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Installed-Capacities.xlsx'           # This is the excel file with an overview
23  # installedcap_excel = r'G:\Andre datamaskiner\Min bærbare datamaskin\Master\Thesis\DynamicStability_thesis\Scenario Installed Cap_DynamicStability.xlsx'
24  output_excel = r'G:\Andre datamaskiner\Min bærbare datamaskin\Master\Thesis\DynamicStability_thesis\output_excel.xlsx'
25
26
27  #%% Initialization of the PowerFactory scripting application and the relevant tools to effectively use Excel applications
28
29  # #Getting the Powerfactory Application
30  import sys
31  sys.path.append('C:\\Program Files\\DIgSILENT\\PowerFactory 2023 SP6\\Python\\3.11')
32
33  import powerfactory as pf                           # Imports the PowerFactory module
34  app=pf.GetApplication()                             # Calls the application
35  app.ActivateProject('DynamicStability_EHVnetwork_test(2)')   # Activates the correct project, if not already activated
36  app.ResetCalculation()                              # Resets all calculations
37  app.ClearOutputWindow()                             # Clears the Output Window in PowerFactory
38
39  import win32com.client as win32                     # Imports Win32
40  excel=win32.gencache.EnsureDispatch('Excel.Application')   # Checks the dispatching of the excel application
41  excel.Visible=True                                  # Makes the excel file visible (can be turned off, however might be needed to click "don't update" when the excel files are ope
42  import math                                         # Calculation tool
43
44
45  #open the needed excel files
46  wbin = excel.Workbooks.Open(input_excel)            # Function to open the input excel file
47  wbout = excel.Workbooks.Open(output_excel)          # Function to open the result excel file
48  wbcap = excel.Workbooks.Open(installedcap_excel)    # Function to open the installed capacity excel file
49  excel.Visible = True                                # Activates application visibility
50
51  #studycase files
52  studycasefolder = app.GetProjectFolder('study')     # Locates the study folders in the PowerFactory module
53  studycases = studycasefolder.GetContents()          # Locates the study cases in the PowerFactory module
54
55  #other input values
56  Genstatmax = 1                                      # Sets level for maximum power dispatch [p.u]
57
58
```

Figure C.5: Snapshot Python Initialization (Scenario-2050).

```
# %% Initialization
excel = win32.gencache.EnsureDispatch('Excel.Application')
excel.Visible = True

# Define paths to your Excel files
# input_excel = r"G:\Andre datamaskiner\Min bærbare datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Scenario-Input-2-with-dispatched-case-b.xlsx"

input_excel = r"G:\Andre datamaskiner\Min bærbare datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Scenario-Input-2-with-dispatched-case-b-test.xlsx"
# Add paths for output_excel and installedcap_excel as necessary

# Open the Excel workbooks
wbin = excel.Workbooks.Open(input_excel)

sheet_II30502 = wbin.Worksheets("II3050")
```

Figure C.6: Snapshot Python Initialization (Calclations).

C.2.3. Excel

No specific setup is required to use the Excel sheets. The key consideration is ensuring that the Excel sheets' correct names and locations are accurately referenced in the Python codes. Additionally, avoid making changes in Excel without verifying their impact on the various Python scripts, as these scripts rely on specific cells in Excel. The Excel version used for this work is 2404.

C.3. Python and Excel Functions

This section explains the different functions of the Excel and Python resources. There are primarily two Excel sheets and two Python scripts of interest.

The two Excel sheets are used to determine the system capacities and dispatches of different generation and load sources for various cases.

The two Python scripts include one for the main calculation of dispatches in the system, considering grid-forming penetration levels and the generation percentages of different sources, as well as flexibility calculations and load determination. The other script is used with PowerFactory to integrate the Excel sheets into PowerFactory, containing various functions for easy integration and adaptation of dynamic models in PowerFactory.

C.3.1. Setting Capacities of the Model

First, the Excel sheet used to set the different capacities of the system, called "Installed-Capacities," is highlighted. This sheet determines the capacities for the static and synchronous machines in the system. The capacities are set manually, and new resources added to the dynamic model should also be implemented in this Excel sheet.

An overview of the Excel setup can be seen in Figure C.7. The red circle highlights the columns and the structure of the setup, the blue circle displays the different sheets available in the Excel structure, and the green square highlights the new implementations done for this project. This general overview shows that multiple characteristics are available for selection.

As previously mentioned, this Excel file is largely based on the work from Midhuna. The work has been adjusted to fit the newly implemented resources while maintaining a similar overall structure for easier integration.

It can be seen that the synchronous machines are chosen based on the "1) SymGen list" Excel sheet. New additions like hydrogen and nuclear power plants and the interconnections represented as synchronous machines are highlighted in the green square. New synchronous machines should be included here if they are added to the system. The corresponding transformers connected to the synchronous machines for the hydrogen and nuclear power plants are also implemented, enabling easy adjustment based on necessary increases or decreases of these sources for different projections. The AC interconnections are directly connected to the 380 kV level and do not require any transformer. It is important to note that the interconnection ratings are properly determined in one of the Python codes by using the synchronous machine's capacity to represent the kinetic energy and inertia levels in the AC interconnections. Therefore, they can remain as is in the Excel sheet, but it is essential to run the exact Python function to properly integrate the AC interconnections. This specific Python function will be explained later.

Additionally, it can be seen that the installed capacities are easily integrated into different scenarios, making it easy to control and compare various scenarios. It is also possible to change the number of parallel machines if necessary.

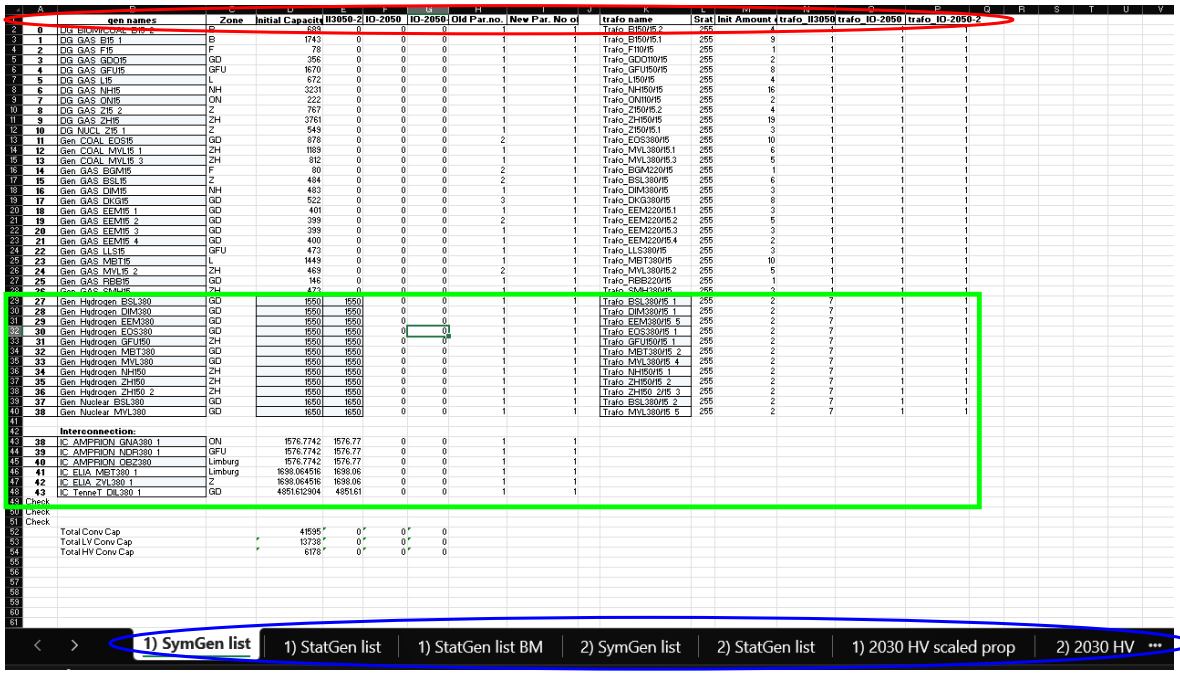
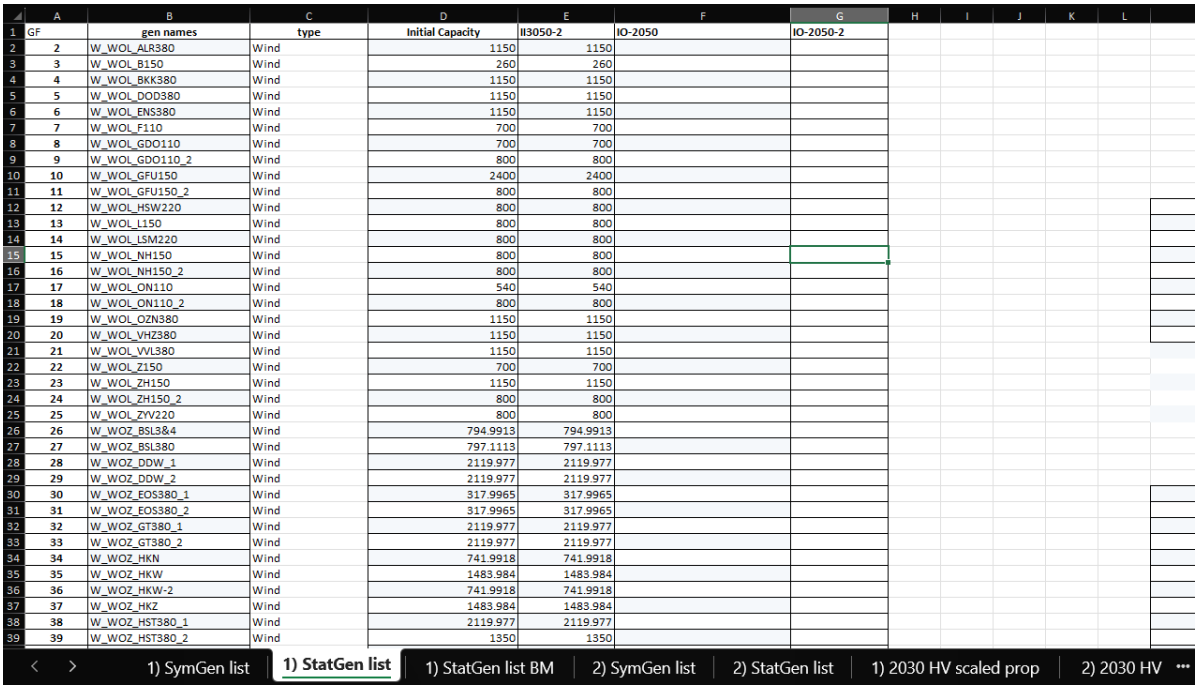


Figure C.7: Snapshot Installed-Capacities Excel.

Next, a snapshot of the static generator sheet in the installed-capacities Excel file is provided in Figure C.8. Here, a similar setup to that of the synchronous generator list is evident. The grid-following converters are represented in this figure. Further down, the BESS (Battery Energy Storage Systems), fuel cell, and interconnection capacities are also implemented.

Following these initializations, an overview of the different sources can be seen in Figure C.9. This overview presents the total ratings for various resources, including a comparison of renewable capacities representing grid-forming and grid-following converter integration. This comparison is useful for an easy overview and ensures that the total installed capacities in the system match.

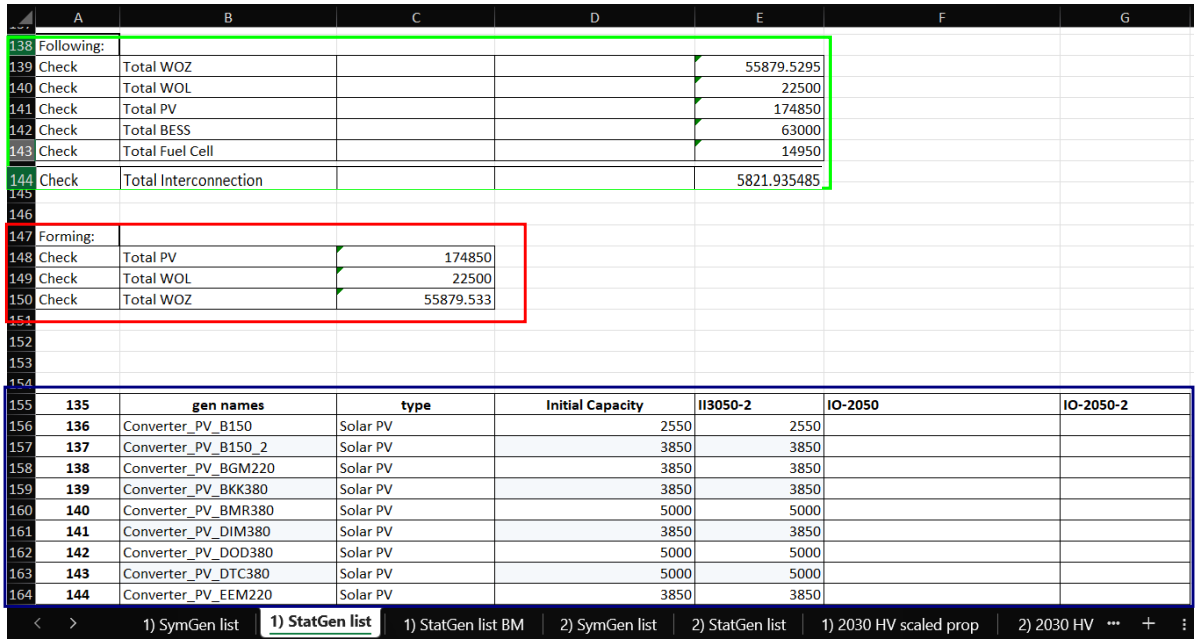
After this, the grid-forming converters are initialized in the same manner as the grid-following. Overall, this setup provides an easy overview, facilitates comparisons, and ensures smooth integration into the model.



GF	gen names	type	Initial Capacity	II3050-2	IO-2050	IO-2050-2
2	W_WOL_ALR380	Wind	1150	1150		
3	W_WOL_B150	Wind	260	260		
4	W_WOL_BKK380	Wind	1150	1150		
5	W_WOL_DOD380	Wind	1150	1150		
6	W_WOL_ENS380	Wind	1150	1150		
7	W_WOL_F110	Wind	700	700		
8	W_WOL_GDO110	Wind	700	700		
9	W_WOL_GDO110_2	Wind	800	800		
10	W_WOL_GFU150	Wind	2400	2400		
11	W_WOL_GFU150_2	Wind	800	800		
12	W_WOL_HSW220	Wind	800	800		
13	W_WOL_L150	Wind	800	800		
14	W_WOL_LSM220	Wind	800	800		
15	W_WOL_NH150	Wind	800	800		
16	W_WOL_NH150_2	Wind	800	800		
17	W_WOL_ON110	Wind	540	540		
18	W_WOL_ON110_2	Wind	800	800		
19	W_WOL_OZN380	Wind	1150	1150		
20	W_WOL_VHZ380	Wind	1150	1150		
21	W_WOL_VVL380	Wind	1150	1150		
22	W_WOL_Z150	Wind	700	700		
23	W_WOL_ZH150	Wind	1150	1150		
24	W_WOL_ZH150_2	Wind	800	800		
25	W_WOL_ZYV220	Wind	800	800		
26	W_WOZ_BSL3&4	Wind	794.9913	794.9913		
27	W_WOZ_BSL380	Wind	797.1113	797.1113		
28	W_WOZ_DDW_1	Wind	2119.977	2119.977		
29	W_WOZ_DDW_2	Wind	2119.977	2119.977		
30	W_WOZ_EOS380_1	Wind	317.9965	317.9965		
31	W_WOZ_EOS380_2	Wind	317.9965	317.9965		
32	W_WOZ_GT380_1	Wind	2119.977	2119.977		
33	W_WOZ_GT380_2	Wind	2119.977	2119.977		
34	W_WOZ_HKN	Wind	741.9918	741.9918		
35	W_WOZ_HKW	Wind	1483.984	1483.984		
36	W_WOZ_HKW-2	Wind	741.9918	741.9918		
37	W_WOZ_HKZ	Wind	1483.984	1483.984		
38	W_WOZ_HST380_1	Wind	2119.977	2119.977		
39	W_WOZ_HST380_2	Wind	1350	1350		

1) SymGen list | 1) StatGen list | 1) StatGen list BM | 2) SymGen list | 2) StatGen list | 1) 2030 HV scaled prop | 2) 2030 HV

Figure C.8: Snapshot Installed-Capacities Excel, statgen sheet.



Following:						
Check	Total WOZ			55879.5295		
Check	Total WOL			22500		
Check	Total PV			174850		
Check	Total BESS			63000		
Check	Total Fuel Cell			14950		
Check	Total Interconnection			5821.935485		
Forming:						
Check	Total PV	174850				
Check	Total WOL	22500				
Check	Total WOZ	55879.533				

135	gen names	type	Initial Capacity	II3050-2	IO-2050	IO-2050-2
136	Converter_PV_B150	Solar PV	2550	2550		
137	Converter_PV_B150_2	Solar PV	3850	3850		
138	Converter_PV_BGM220	Solar PV	3850	3850		
139	Converter_PV_BKK380	Solar PV	3850	3850		
140	Converter_PV_BMR380	Solar PV	5000	5000		
141	Converter_PV_DIM380	Solar PV	3850	3850		
142	Converter_PV_DOD380	Solar PV	5000	5000		
143	Converter_PV_DTC380	Solar PV	5000	5000		
144	Converter_PV_EEM220	Solar PV	3850	3850		

1) SymGen list | 1) StatGen list | 1) StatGen list BM | 2) SymGen list | 2) StatGen list | 1) 2030 HV scaled prop | 2) 2030 HV

Figure C.9: Snapshot Installed-Capacities Excel, Overview of different capacities.

It can be seen that multiple other sheets are available in the Excel file. These sheets were brought over from previous projects and were left uninterrupted in the current project to allow for future integration if necessary. Thus, only the first two sheets are relevant for this project. Refer to [72] for further details on these sheets.

To integrate these capacities into the PowerFactory model, a script was updated in Python. A snapshot of this function can be seen in Figure C.10. This function was originally available in previous work by De Roos and has been extended to integrate the newly implemented resources and capacities. It can be seen that the first two Excel sheets for synchronous and static generators are utilized. The different

inputs are classified into ranges for the various resources, allowing for easy implementation and further extension in future work.

The resources are split into separate lists and then implemented into the PowerFactory model. Each list containing the different resources is processed in a for loop, facilitating easy integration into PowerFactory. The implementation depends on the resource type in PowerFactory, such as a synchronous generator for the first list, represented as a ".ElmSym" type. The number of parallel units is determined by row 7. The for loop sets the nominal apparent power of the generator based on row 3 if the power rating is above zero and implements the active and reactive power limits. If row 3 is zero, the generators are set out of service. For different scenarios, the rows must be adjusted to fit accordingly based on their placement in the Excel sheet. This adjustment can be done within the specified function, or a new capacities function can be created similarly, with different rows representing the capacities. This implementation approach continues for all resources in the Excel sheets, including transformers, as shown in the figure. The entire code and function are provided in section C.5. Each resource type has slight differences in logical implementation, but the overall structure is similar, with special differences noted as comments in the code.

To utilize the function, the function call "capacities2050()" must be uncommented in the overall function call structure, as highlighted in subsubsection C.3.4.8. This should be done for all cases where the capacities are necessary, and it only needs to be done during the model initialization. The model speed is significantly enhanced if this is not implemented each time the script is run.

After this, the system's installed capacities should be set. The overall structure allows for easy adaptation and implementation of different resources, using Python and Excel in combination to facilitate quick integration with PowerFactory.

```
60  #%% Definition of function that allows to set the capacities of the system in PowerFactory based on the relevant excel file
61
62  def capacities2050():
63      wbcap1=wbcap.Worksheets("1) SymGen list")                # Activates the sheet with the Synchronous generators
64      wbcap2 = wbcap.Worksheets("1) StatGen list")             # Activates the sheet with the static generators
65      symgenlist = wbcap1.Range("B2:I40").Value                # Finds the range of information regarding Synchronuos generators in the sheet
66      trafolist = wbcap1.Range("K2:P12").Value                 # Finds the same range for transformers
67      statgenlistfollowing = wbcap2.Range("B2:G59").Value      # Finds the range for static generators marked with a "following" control system
68      statgenPVlistfollowing = wbcap2.Range("B60:G104").Value  # Finds the range for PV systems marked with a "following" control system
69      statgenforming = wbcap2.Range("B156:G258").Value         # Finds the range for static generators marked with a "forming" control system
70      statgenflexibility = wbcap2.Range("B105:G134").Value     # Finds the range for static generators labeled as a flexibility resource (batteries etc)
71      interconnectionDC = wbcap2.Range("B264:G266").Value      # Finds the range for HVDC interconnections
72      interconnectionAC = wbcap1.Range("B43:I48").Value        # Finds the range for AC interconnections
73
74
75
76      #Synchronous generation
77      for row in symgenlist:                                   # For loop for the rows in the list containing synchronous generator information
78          name = row[0]+ '.Elmsym'                             # Defines the name of the component, with the type in PowerFactory added (For the names in excel)
79          variable = app.GetCalcRelevantObjects(name)[0]       # Locates the object in PowerFactory with the same name
80          variable.ngnum = int(row[7])                         # Determines the numbers of paralell generators in PowerFactory
81          if float(row[3])>0:                                  # For loop that activates if the active power is higher than 0
82              variable.typ_id.sgn = float(row[3])              # Determines the nominal apparent power of the generator
83              variable.Pmax_ucPU = 1                           # Active Power limit in p.u.
84              variable.Pmin_ucPU = 0                           # Reactive Power limit in p.u.
85          if float(row[3])==0:
86              variable.outserv = 1                             # Sets the generator out of service if the active power = 0
87
88                                                               # This is further done for all the other components
89
90      # Transformers
91      for row in trafolist:
92          name = row[0]+'.ElmTr2'
93          variable = app.GetCalcRelevantObjects(name)[0]
94          variable.ntnum = math.ceil(row[3])
```

Figure C.10: Snapshot of installed capacities function in Python code (Scenario-2050).

C.3.2. Setting Dispatches of the System

Next, the structure for setting the dispatches for various scenarios and cases is discussed. This is accomplished using Excel and Python in combination with scripting through PowerFactory. The Excel sheet used is the "Scenario-input" sheet, and the corresponding PowerFactory functions are called "dispatches."

As before, De Roos had already established much of the structure of the Excel sheets, and the implementation of the new sources was incorporated here. This includes different grid-following and grid-forming resources, new fuel cells, hydrogen and nuclear power plants, and BESS systems.

An overview of the Excel sheet is provided in Figure C.11. The snapshot shows the start of the Excel sheet, highlighting the synchronous machines in the system. Note that the AC interconnections are

located further down in the sheet. The newly implemented synchronous machines, such as hydrogen and nuclear power plants, are highlighted in the blue square. While the older synchronous power plants above are not used in any of the cases for this scenario, they are still included in the Excel sheet to facilitate easy implementation if necessary. Additionally, the totals of each dispatch are given at the bottom, providing a simple overview of the total synchronous generation in the system. The figure also highlights the active, reactive, and apparent power for Case 1 and Case 2. This extends to Case 6, covering the different cases made and implemented for Case Study A.

9	gen names	Zone	S base model	New Pmax	PF dispatch	P_Case-1	S_Case-1	Q_Case-1	P_Case-2	S_Case-2	Q_Case-2
10	DG_BIOMCOAL_B15_2	B	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
11	DG_GAS_B15_1	B	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
12	DG_GAS_F15	F	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
13	DG_GAS_GDO15	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
14	DG_GAS_GFU15	GFU	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
15	DG_GAS_L15	L	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
16	DG_GAS_NH15	NH	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
17	DG_GAS_ON15	ON	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
18	DG_GAS_Z15_2	Z	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
19	DG_GAS_ZH15	ZH	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
20	DG_NUCL_Z15_1	Z	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
21	Gen_COAL_EOS15	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
22	Gen_COAL_MVL15_1	ZH	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
23	Gen_COAL_MVL15_3	ZH	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
24	Gen_GAS_BGM15	F	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
25	Gen_GAS_BSL15	Z	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
26	Gen_GAS_DIM15	NH	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
27	Gen_GAS_DKG15	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
28	Gen_GAS_EEM15_1	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
29	Gen_GAS_EEM15_2	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
30	Gen_GAS_EEM15_3	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
31	Gen_GAS_EEM15_4	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
32	Gen_GAS_LLS15	GFU	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
33	Gen_GAS_MBT15	L	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
34	Gen_GAS_MVL15_2	ZH	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
35	Gen_GAS_RBB15	GD	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
36	Gen_GAS_SMH15	ZH	0	0	0.96	0.00	0.00	0.00	0.00	0.00	0.00
37	Gen_Hydrogen_BSL380	GD	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
38	Gen_Hydrogen_DIM380	GD	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
39	Gen_Hydrogen_EEM380	GD	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
40	Gen_Hydrogen_EOS380	GD	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
41	Gen_Hydrogen_GFU150	ZH	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
42	Gen_Hydrogen_MBT380	GD	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
43	Gen_Hydrogen_MVL380	GD	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
44	Gen_Hydrogen_NH150	ZH	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
45	Gen_Hydrogen_ZH150	ZH	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
46	Gen_Hydrogen_ZH150_2	ZH	1550	1500	0.96	0	0	0	530	552.0833333	154.5833333
47	Gen_Nuclear_BSL380	GD	1550	1500	0.96	1000	1041.666667	291.6666667	1000	1041.666667	291.6666667
48	Gen_Nuclear_MVL380	GD	1550	1500	0.96	1000	1041.666667	291.6666667	1000	1041.666667	291.6666667
49			0	0		0	0	0	0	0	0
50						0	0	0	0	0	0
51						0	0	0	0	0	0
52						0	0	0	0	0	0
53						0	0	0	0	0	0
54						0	0	0	0	0	0
55	Totals		0	0		2000.00	2083.33	583.33	7300.00	7604.17	2129.17

Figure C.11: Snapshot Scenario-input Excel, Overview of dispatches of synchronous machine.

Further, a similar structure is observed for the implementation of the grid-following converters, as shown in Figure C.12. The main difference is that the type of converter is highlighted instead of the zone of the generator, as in the synchronous dispatches. However, the overall S base of the model, the maximum active power, and the power factor (PF) are similar and easily adjustable. Once again, the different dispatches span across the six different cases, similar to the synchronous generator section.

This grid-following section implements the three different grid-following converters for offshore and onshore wind, solar PV, BESS, and fuel cells, all following a similar structure. The same structure is applied to the grid-forming converters, which are implemented beneath these sections. An overview of each source's totals is provided for both the grid-following and grid-forming sections, allowing for easy comparison between the different dispatches.

313	Input for Loads total		26430			16000	0	0	0	0	0
314	load names	type	Max Capacity		PF dispatch	P_Case-1	S_Case-1	Q_Case-1	P_Case-2	S_Case-2	Q_Case-2
315	Electrolyser_BSL380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
316	Electrolyser_BVW380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
317	Electrolyser_DIM380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
318	Electrolyser_EEM380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
319	Electrolyser_EOS380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
320	Electrolyser_HST380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
321	Electrolyser_LLS380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
322	Electrolyser_MVL380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
323	Electrolyser_OOP380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
324	Electrolyser_TNZ380	Electrolyser	2500		1	602.0292562	602.0292562	0	0	0	0
325	Load_B150	Load	4131.66856		0.985	4131.66856	4194.587371	723.7942522	4131.66856	4194.587371	723.7942522
326	Load_B150_LR	Load Response	1110		0.985	1110	1126.903553	194.4520981	1110.0	1126.903553	194.4520981
327	Load_B150_PTH	Power To Heat	1110		0.985	1110.0	1126.903553	194.4520981	1110.0	1126.903553	194.4520981
328	Load_F110	Load	794.6		0.985	794.6	806.6514173	139.1912023	794.6	806.6514173	139.1912023
329	Load_F110_LR	Load Response	1110		0.985	1110.0	1126.903553	194.4520981	1110.0	1126.903553	194.4520981
330	Load_F110_PTH	Power To Heat	1110		0.985	1110.0	1126.903553	194.4520981	1110.0	1126.903553	194.4520981
331	Load_GDO110	Load	1695.0		0.985	1695.0	1720.856355	296.9412314	1695.04351	1720.856355	296.9412314
332	Load_GDO110_LR	Load Response	1110		0.985	1110.0	1126.903553	194.4520981	1110.0	1126.903553	194.4520981
333	Load_GDO110_PTH	Power To Heat	1110		0.985	1110.0	1126.903553	194.4520981	1110.0	1126.903553	194.4520981
334	Load_GFU150	Load	4555.4		0.985	4555.4	4624.801462	798.0295605	4555.4	4624.801462	798.0295605
335	Load_GFU215_LR	Load Response	1110		0.985	1110.0	1126.903553	194.4520981	1110.0	1126.903553	194.4520981
336	Load_GFU150_PTH	Power To Heat	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
337	Load_L150	Load	2330.68483		0.985	2330.68483	2366.177492	408.2941938	2330.68483	2366.177492	408.2941938
338	Load_L150_LR	Load Response	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
339	Load_L150_PTH	Power To Heat	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
340	Load_NH150	Load	5826.71207		0.985	5826.71207	5915.443726	1020.735484	5826.71207	5915.443726	1020.735484
341	Load_NH150_LR	Load Response	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
342	Load_NH150_PTH	Power To Heat	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
343	Load_ON110	Load	1800.983731		0.985	1800.983731	1828.409879	315.5000586	1800.983731	1828.409879	315.5000586
344	Load_ON110_LR	Load Response	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
345	Load_ON110_PTH	Power To Heat	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
346	Load_Z150	Load	1250.009459		0.985	1250.009459	1269.045136	218.9792449	1250.009459	1269.045136	218.9792449
347	Load_Z150_LR	Load Response	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
348	Load_Z150_PTH	Power To Heat	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
349	Load_ZH150	Load	5614.83163		0.985	5614.83163	5700.33668	983.6178295	5614.83163	5700.33668	983.6178295
350	Load_ZH150_LR	Load Response	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
351	Load_ZH150_PTH	Power To Heat	1110		0.985	1110	1126.903553	194.4520981	1110	1126.903553	194.4520981
352	Totals:		47979.9			47979.91488			47979.91488		

Figure C.12: Snapshot Scenario-input Excel, Overview of dispatches of grid-following converter.

Moving on, a similar structure is observed for the implementation of the different load resources, as shown in Figure C.13. Here, all the loads in the system are represented, including electrolyzers, regular loads, power to heat, and load response. The type, maximum capacity, and power factor (PF) dispatch are set, along with the active, reactive, and apparent powers for all six different cases.

It is important to note that the electrolyzers, load response, and power to heat are also implemented by a dynamic model that requires manual setting of the dispatches to respond during dynamic simulations. This process is further elaborated on in subsubsection C.3.4.4 and is accomplished using Python scripts.

58	Grid-following										
59	**gen names**	**type**	**S base model**	**Pmax**	**PF dispatch**	**P_Case-1**	**S_Case-1**	**Q_Case-1**	**P_Case-2**	**S_Case-2**	**Q_Case-2**
60	W_WOL_ALR380	Wind	1150	1000	1.00	0	0	0	0	0	0
61	W_WOL_B150	Wind	260	260	1.00	0	0	0	0	260	0
62	W_WOL_BKK380	Wind	1150	1000	1.00	0	0	0	924	0	0
63	W_WOL_DDD380	Wind	1150	1000	1.00	0	0	0	924	1000	0
64	W_WOL_ENS380	Wind	1150	1000	1.00	0	0	0	924	1000	0
65	W_WOL_F110	Wind	700	700	1.00	78.84863124	78.84863124	0	647	0	0
66	W_WOL_GDO110	Wind	700	700	1.00	0	0	0	0	700	0
67	W_WOL_GDO110_2	Wind	800	715	1.00	80.53824477	80.53824477	0	0	715	0
68	W_WOL_GFU150	Wind	2400	2400	1.00	0	0	0	0	0	0
69	W_WOL_GFU150_2	Wind	800	715	1.00	0	0	0	661	715	0
70	W_WOL_HSW220	Wind	800	715	1.00	0	0	0	661	0	0
71	W_WOL_L150	Wind	800	715	1.00	80.53824477	80.53824477	0	0	0	0
72	W_WOL_LSM220	Wind	800	715	1.00	80.53824477	80.53824477	0	0	0	0
73	W_WOL_NH150	Wind	800	800	1.00	90.11272142	90.11272142	0	739	800	0
74	W_WOL_NH150_2	Wind	800	715	1.00	80.53824477	80.53824477	0	661	715	0
75	W_WOL_ON110	Wind	540	540	1.00	60.82608696	60.82608696	0	0	540	0
76	W_WOL_ON110_2	Wind	800	715	1.00	0	0	0	0	0	0
77	W_WOL_OZN380	Wind	1150	1000	1.00	112.6409018	112.6409018	0	324	1000	0
78	W_WOL_VHZ380	Wind	1150	1000	1.00	112.6409018	112.6409018	0	324	0	0
79	W_WOL_VVL380	Wind	1150	1000	1.00	112.6409018	112.6409018	0	0	0	0
80	W_WOL_Z150	Wind	700	700	1.00	78.84863124	78.84863124	0	0	700	0
81	W_WOL_ZH150	Wind	1150	1150	1.00	0	0	0	0	0	0
82	W_WOL_ZH150_2	Wind	800	715	1.00	80.53824477	80.53824477	0	661	0	0
83	W_WOL_ZVV220	Wind	800	715	1.00	0	0	0	661	715	0
84	W_WOZ_BSL364	Wind	794.9913	700	1.00	584.6422261	584.6422261	0	39	0	0
85	W_WOZ_BSL380	Wind	797.1113	700	1.00	0	0	0	0	42.31012658	0

Figure C.13: Snapshot Scenario-input Excel, Overview of dispatches of different load resources.

Further, the overview of the interconnections and STATCOM dispatches can be seen in Figure C.14. Here, the AC and DC interconnections are grouped together. However, similar to the installed capacities, the AC interconnections are set dynamically based on the kinetic energy and inertia constant values as a representation of the apparent power. These are calculated and implemented using another part of the Python setup, as detailed in subsubsection C.3.4.7. A similar setup is used for the STATCOMs, where the reactive power is the critical factor. For both interconnections and STATCOMs, it is possible to adjust the dispatches for all six cases.

354	**Interconnections**		**S base model**	**New Pmax**	**PF dispatch**	**P_Case-1**	**S_Case-1**	**Q_Case-1**	**P_Case-2**	**S_Case-2**	**Q_Case-2**
355	IC_BritNed_MVL380	Interconnection	2500	2425.8	0.98	0	0	0	574.8016014	586.5322463	116.7184433
356	IC_COBRA_EOS380	Interconnection	1750	1698.45	0.98	0	0	0	402.4535328	410.6668702	81.72167534
357	IC_StatNetr_EEM380	Interconnection	1750	1698.45	0.98	0	0	0	402.4535328	410.6668702	81.72167534
358	IC_AMPRION_GNA380_1	Interconnection	1650	1576	0.98	0	0	0	373.4385868	381.0597824	75.82393326
359	IC_AMPRION_NDR380_1	Interconnection	1650	1576	0.98	0	0	0	373.4385868	381.0597824	75.82393326
360	IC_AMPRION_OBZ380	Interconnection	1650	1576	0.98	0	0	0	373.4385868	381.0597824	75.82393326
361	IC_ELIA_MBT380_1	Interconnection	1750	1698	0.98	0	0	0	402.3469038	410.5580651	81.70002339
362	IC_ELIA_ZVL380_1	Interconnection	1750	1698	0.98	0	0	0	402.3469038	410.5580651	81.70002339
363	IC_TenneT_DIL380_1	Interconnection	5000	4852	0.98	0	0	0	1149.697984	1173.161208	233.4561328
364											
365											
366	**Compensation**	**type**	**Max Capacity**		**PF dispatch**	**P_Case-1**	**S_Case-1**	**Q_Case-1**	**P_Case-2**	**S_Case-2**	**Q_Case-2**
367	STATCOM_ALR380_gen	Reactive Power Compensa	2500		0	0	0	-600	0	0	0
368	STATCOM_B150_gen	Reactive Power Compensa	2000		0	0	0	1400	0		1340
369	STATCOM_BKK380_gen	Reactive Power Compensa	3000		0	0	0	-2000	0		-2850
370	STATCOM_BSL380_gen	Reactive Power Compensa	2500		0	0	0	-700	0		0
371	STATCOM_BVW380_gen	Reactive Power Compensa	3500		0	0	0	-1157	0		0
372	STATCOM_DIM380_gen	Reactive Power Compensa	3000		0	0	0	-700	0		-1500
373	STATCOM_DDD380_gen	Reactive Power Compensa	3000		0	0	0	-2600	0		-1300
374	STATCOM_DTC380_gen	Reactive Power Compensa	2000		0	0	0	0	0		-1500
375	STATCOM_EEM380_gen	Reactive Power Compensa	2500		0	0	0	-2000	0		0
376	STATCOM_ENS220_gen	Reactive Power Compensa	2000		0	0	0	0	0		0
377	STATCOM_ENS380_gen	Reactive Power Compensa	3000		0	0	0	-2500	0		0
378	STATCOM_EOS380_gen(1)	Reactive Power Compensa	2500		0	0	0	1030	0		800
379	STATCOM_F110_gen	Reactive Power Compensa	2000		0	0	0	686	0		800
380	STATCOM_GDO110_gen	Reactive Power Compensa	2000		0	0	0	0	0		-750
381	STATCOM_GFU150_gen	Reactive Power Compensa	2000		0	0	0	-100	0		-1000
382	STATCOM_KIJ380_gen	Reactive Power Compensa	3000		0	0	0	600	0		0
383	STATCOM_L150_gen	Reactive Power Compensa	2500		0	0	0	420	0		0
384	STATCOM_LLS380_gen	Reactive Power Compensa	2500		0	0	0	-500	0		-2400
385	STATCOM_MBT380_gen	Reactive Power Compensa	2500		0	0	0	1600	0		1300
386	STATCOM_NH150_gen	Reactive Power Compensa	2500		0	0	0	1600	0		350
387	STATCOM_ON110_gen	Reactive Power Compensa	2000		0	0	0	300	0		70
388	STATCOM_OOP380_gen	Reactive Power Compensa	2500		0	0	0	0	0		0
389	STATCOM_RLL380_gen	Reactive Power Compensa	2500		0	0	0	0	0		0
390	STATCOM_TNZ380_gen	Reactive Power Compensa	2500		0	0	0	70	0		300
391	STATCOM_Z150_gen	Reactive Power Compensa	2000		0	0	0	1012	0		826
392	STATCOM_ZH150_gen	Reactive Power Compensa	2000		0	0	0	710	0		-176

Figure C.14: Snapshot Scenario-input Excel, Overview of dispatches of interconnections and STATCOMs.

The Excel sheet also contains an overview of the system's different active, reactive, and apparent power levels, as seen in Figure C.15. The different rows represent power levels for various resources in the system, including renewable and conventional generation, total demand, and the different levels of demand and flexibility resources. These rows also include various calculations to help identify gaps in the power levels within the system, such as the amount of flexibility needed. These calculations are performed using Python codes, as detailed in subsubsection C.3.3.1.

						P_Case-1	S_Case-1	Q_Case-1	P_Case-2	S_Case-2	Q_Case-2
401	Totals										
402	Total Conv Generation					2000.00	2083.33	583.33	2000.00	2083.33	583.33
403	Total VRES Generation					58020.5	58020.5	0.0	29892.3	29451.5	0.0
404	Total Demand					47979.9	48710.6	8405.2	47979.9	48710.6	8405.2
405	Total Load Response Demand					9990.0	10142.1	1750.1	9990.0	10142.1	1750.1
406	Total Power to heat demand					9990.0	10142.1	1750.1	9990.0	10142.1	1750.1
407	Total IC					0	0	0	4454	4545	905
408	Gen-Demand					**12040.59**	**11393.26**	**-7821.89**	**-16087.66**	**-17175.74**	**-7821.89**
409	Total STATCOM							-3429			-5690
410	Total Hydrogen Plant:					0	0	0	5300	5520.833333	1545.833333
411	Total Electrolyzers					**6020.292562**	**6020.292562**	**0**	**0**	**0**	**0**
412	Total BESS					**-6000**	**-6000**	**0**	**4454.416219**	**4454.416219**	**0**
413	Total Fuel-Cell					**0**	**0**	**0**	**2000**	**2000**	**0**
414	Gen-Demand including flexibility					20.29	-627.03	-4392.89	121.17	-655.17	318.45
415					BESS: Negative = load, positive = generation						
416					Electrolyzer: Positive = load						
417					Hydrogen: Positive = generation						
418					Fuel cell: Positive = generation						
419					Interconnection: Positive = generation, negative = load						

Figure C.15: Snapshot Scenario-input Excel, Overview of different active, reactive, and apparent power levels in the system.

The Excel sheet also contains an overview of the different dispatches separated based on the regions of the system, as seen in Figure C.16. However, this section was not updated to reflect the newly implemented changes; it was retained to facilitate easy implementation for future work. Therefore, if these regional dispatches are to be used, they need to be updated with the appropriate cell ranges in Excel.

426	Totals	Load	LV Gas +WKK	HV Gas	Tot Gas	WOL/WOZ LV	PV	Left Cap
427	B	1110.0	0.00	0	0.00	#REF!	#REF!	
428	F	794.6	0.00	0	0.00	#REF!	#REF!	
429	GD	#REF!	0.00	0	0.00	#REF!	#REF!	
430	GFU	1110.0	0.00	0	0.00	#REF!	#REF!	
431	L	1695.0	0.00	0	0.00	#REF!	#REF!	
432	NH	1110.0	0.00	0	0.00	#REF!	#REF!	
433	ON	1110.0	0.00	0	0.00	#REF!	#REF!	
434	Z	4555.4	0.00	0	0.00	#REF!	#REF!	
435	ZH	1110.0	0.00	0	0.00	#REF!	#REF!	
436	Totals	#REF!	0.0	0.0	0.00	#REF!	#REF!	
437								
438	Regional Dispatches	Load	Gas/WKK	PV	WOL/WOZ	Export (+)/Import (-)	Trafo Cap	Trafo Left Cap
439	B	1110.0	0.00	1000	715	605.00	4250	3645.00
440	F	794.6	0.00	260	715	180.45	1570	1389.55
441	GD	1110.0	0.00	1000	715	605.00	4260	3655.00
442	GFU	1110.0	0.00	1000	715	605.00	3850	3245.00
443	L	1695.0	0.00	1000	800	104.96	2350	2245.04
444	NH	1110.0	0.00	700	1715	1305.00	4350	3045.00
445	ON	1110.0	0.00	700	540	130.00	2160	2030.00
446	Z	4555.4	0.00	715	715	-3125.43	1500	-1625.43
447	ZH	1110.0	0.00	2400	2000	3290.00	11800	8510.00
448	Totals	13705.0	0.0	8775.0	8630.0	3700.0	36090.0	26139.2
449								
450	Regional Dispatches	Load	Gas/WKK	PV	WOL/WOZ	Export (+)/Import (-)	Trafo Cap	Trafo Left Cap
451	B	1110.0	0.00	0	660.5854839	-449.41	4250	3800.59
452	F	1110.0	0.00	0	660.5854839	-449.41	1570	1120.59
453	GD	794.6	0.00	923.8957816	0	129.34	4260	4130.66
454	GFU	1110.0	0.00	923.8957816	0	-186.10	3850	3663.90
455	L	1110.0	0.00	923.8957816	739.1166253	553.01	2350	1796.99
456	NH	1110.0	0.00	646.7270471	1584.481266	1121.21	4350	3228.79
457	ON	1110.0	0.00	0	0	-1110.00	2160	1050.00
458	Z	4555.4	0.00	0	0	-4555.43	1500	-3055.43
459	ZH	1110.0	0.00	0	923.8957816	-186.10	11800	11613.90
460	Totals	13120.0	0.0	3418.4	4568.7	-5132.9	36090.0	27350.0
461								
462	Regional Dispatches	Load	Gas/WKK	PV	WOL/WOZ	Export (+)/Import (-)	Trafo Cap	Trafo Left Cap
463	B	1110.0	0.00	109.1263651	78	-922.85	4250	3327.15
464	F	1110.0	0.00	0	78	-1031.97	1570	538.03
465	GD	794.6	0.00	109.1263651	78	-607.40	4260	3652.60
466	GFU	1110.0	0.00	109.1263651	0	-1000.87	3850	2849.13
467	L	1110.0	0.00	0	87	-1022.70	2350	1327.30
468	NH	1110.0	0.00	76.38845554	109	-924.49	4350	3425.51
469	ON	1110.0	0.00	0	59	-1051.07	2160	1108.93
470	Z	4555.4	0.00	78.02535101	0	-4477.40	1500	-2977.40
471	ZH	1110.0	0.00	0	0	-1110.00	11800	10690.00
472	Totals	13120.0	0.0	481.8	489.4	-12148.8	36090.0	23941.2

Figure C.16: Snapshot Scenario-input Excel, Overview of regional dispatches.

At the bottom of the Excel sheet are lists containing dispatches for the different grid-forming and grid-following converters and the STATCOM levels used for the grid-forming penetration comparison conducted in Case Study B. This is shown in Figure C.17. Due to the Python code calculating the different dispatches based on a random function, these values will vary each time the code is run. Therefore, the levels for the three different penetration levels were stored in the Excel sheet to allow replication of the results found in Case Study B. This is also important to remember for the different dispatches across the six cases. Thus, they should not be altered if a reenactment of the study is intended.

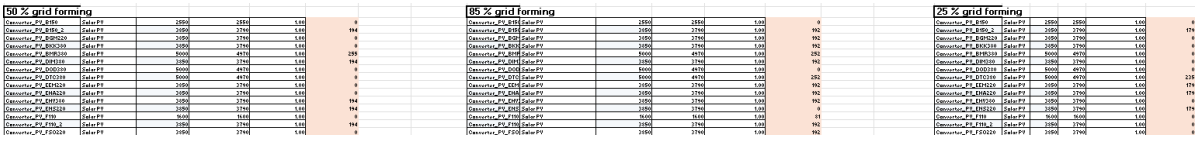


Figure C.17: Snapshot Scenario-input Excel, dispatches used for grid-forming penetration case study.

Finally, the Excel sheet contains a section representing different characteristics of the system as percentages, shown in Figure C.18. The first table provides information on the six different cases and their respective levels of various resources, such as renewable generation and demand levels. The second table contains information regarding the different flexibility resources, which depend on whether there is an excess or lack of generation in the system. These two tables are also accompanied by a grid-forming penetration level percentage. All of these tables represent different operational characteristics of the system's dispatches and allow for significant customizability concerning different dispatches for various systems. By adjusting these tables, it is possible to quickly adapt the system to different levels of generation and/or flexibility resources, among other factors, enabling easy analysis of the impact of certain resources. More information on how these calculations are performed can be found in subsection C.3.3 and subsubsection C.3.3.1.

Finally, there is a table representing the different levels of the various resources for the six different cases. This table was used in the report to explain the operational characteristics of the different cases and is provided here for an easy overview of the given setup.

	AH	AI	AJ	AK	AL	AM	AN
19							
20	WOZ	WOL	PV	Demand	LR	Power 2 Heat (%)	
21	90	10	5	100	100	100	
22	5	90	5	100	100	100	
23	0	10	65	100	100	100	
24	90	10	5	45	100	100	
25	5	90	5	45	100	100	
26	0	10	65	45	100	100	
27							
28	Flexibility:	IC	BESS	Electrolyzer	Hydrogen PP	Fuel Cell	Total
29	Surplus generation	25	25	50	0	0	100
30	Surplus Load	25	25	0	25	25	100
31							
32							
33	Grid forming %	50					
34							
35							
36							
37							
38	Scenario	WOZ	WOL	PV	Flexibility	Conv. Generation	Demand
39	A	47272.5	2068.5	8649.5	12020.29256	2000	47979.9
40	B	2626.25	18616.5	8649.5	11754.41622	2000	47979.9
41	C	0	2068.5	112443.5	67800	2000	47979.9
42	D	47272.5	2068.5	8649.5	38519.6922	2000	21591.0
43	E	2626.25	18616.5	8649.5	10300	2000	21591.0
44	F	0	2068.5	112443.5	75620	2000	21591.0
45							

Figure C.18: Snapshot Scenario-input Excel, tables for different calculations, and table for different system characteristics.

All the different rows and columns are referenced in various parts of the Excel sheet, as well as the Python scripts in this Excel sheet. Therefore, if any cells are moved, it is crucial to cross-reference to ensure that the relocated cells are not connected to other factors. Failing to do so could result in incomplete implementation of resources or miscalculations based on the Python codes.

Next, the implementation of the dispatches to PowerFactory using Python is explained. The function utilized is shown in Figure C.19. This function is largely based on a pre-existing similar function made by De Roos and has been upgraded to better fit the additions made to the Excel sheet and the different Python functions that utilize the dispatch levels. Similar to the installed-capacities function, this function first defines the different ranges for the various resources in Excel before calling them through for loops. These for loops are further classified and implemented using the type of the different resources, as before. The active and reactive power dispatches are then transferred from the Excel sheet to the different resources in PowerFactory. It can be seen that rows 5 and 7 are used for the dispatches and

need to be changed for the different cases due to the Excel structure. This has already been done, and six different dispatch functions are available in the Python script. Each of these functions has already been adjusted to fit the correct rows of dispatches. In Python, these are classified as rows because of the float structure; however, in Excel, these are technically columns representing the different case dispatches found in the Excel sheet. It is also possible to set the maximum and minimum active power limits based on either a number or the Genstatmax, which can be dynamically set at the beginning of the Python script.

Additionally, it is possible to set the different resources out of service by changing the "variable.outserv" to a value of 1. The for loops cover all the different resources implemented. The different values for the loads are also found and then called using a function to set the dispatched load levels for the different load dynamic models, seen in the function "update dynamic model parameters," which utilizes the variable (Load 1).

Similar to the installed-capacities, this function is called by uncommenting the different "dispatches" functions seen in the different cases from the function call part of the script, as shown in subsubsection C.3.4.8. The different dispatches, ranging from 1-6, account for the cases ranging from A-F.

If new models are added to the Excel sheet or existing ones are moved, adjusting the cell ranges for the different variables representing the various resources in the Python function is important. Otherwise, there will not be an accurate translation between the different calculations and dispatches in the Excel sheet and the PowerFactory model. The same applies if new additional resources are added, such as different types, in which case a new for loop needs to be created to implement them and a new variable representing the range in Excel.

```
160 def dispatches1():                                   # Set dispatches for the first Case, Case A
161     w1 = wbin.Worksheets("II3050")                   # Name of the worksheet
162     SymG_1 = w1.Range("A37:Z48").Value               # Synchronous generators
163     StatG_following = w1.Range("A60:Z117").Value     # All static generators with a "following" control mode
164     StatG_PV_following = w1.Range("A118:Z162").Value # All PV systems with a "following" control mode
165     StatG_battery = w1.Range("A163:Z182").Value      # Battery systems
166     StatG_fuelcell = w1.Range("A183:Z192").Value     # Fuel cells
167     StatG_forming = w1.Range("A199:Z301").Value      # All converters with a "forming" control mode
168     Load_1 = w1.Range("A315:Z351").Value             # Loads (Includes Power to heat and load response)
169     StatG_interconnection = w1.Range("A355:Z357").Value # HVDC interconnections
170     SymG_interconnection = w1.Range("A358:Z363").Value  # AC interconnections
171     Reactive_compensation = w1.Range("A367:Z388").Value
172     update_dynamic_model_parameters(Load_1)          # Updates the dynamic model parameters for all units using electrolyser droop control.
173
174
175
176     for row in SymG_1:                               # Updates the list for synchronous generators:
177         name = row[0]+ '.ElmSym'                     # Defines the name of the component, with the type in PowerFactory added
178         variable = app.GetCalcRelevantObjects(name)[0] # Locates the object in PowerFactory with the same name
179         variable.pgini = float(row[5])               # Active Power Dispatch based on dispatch scenario in Excel sheet
180         variable.qgini = float(row[7])               # Reactive Power Dispatch
181         variable.Pmax_ucPU = 1                       # Active Power limit in p.u.
182         variable.Pmin_ucPU = 0                       # Minimum Active Power limit in p.u.
183         variable.outserv = 0                         # Determines the operation of the components  1 (Sets the components out of service)
184
185     for row in StatG_following:                      # Similar strucutre is followed for all components
186         name = row[0]+'.ElmGenstat'
187         variable = app.GetCalcRelevantObjects(name)[0]
188         variable.pgini = float(row[5])
189         variable.qgini = float(row[7])
190         variable.Pmax_ucPU = Genstatmax              # Sets the maximum Active power based on ratings from the capacities function
191         variable.Pmin_ucPU = 0                       # Sets the minimum Active power
192         variable.outserv = 0
193     for row in StatG_PV_following:
194         name = row[0]+'.ElmPvsys'
195         variable = app.GetCalcRelevantObjects(name)[0]
196         variable.pgini = float(row[5])
197         variable.qgini = float(row[7])
```

Figure C.19: Snapshot of dispatches function in Python code (Scenario-2050).

It is worth noting that small changes are typically made between the different cases regarding overlapping control of busbar voltage and voltage values in the system. This usually results in minor adjustments being necessary for the STATCOM values presented in the system to achieve the desired voltage levels and proper initialization. The STATCOMs are set manually, and minor changes can have a significant impact. It is recommended to analyze whether the different dynamic loads in the system have the same active power levels for the dynamic model initialization in PowerFactory. If not, the STATCOM values at the specific busbar need to be adjusted, or a change in the voltage control is necessary.

C.3.3. Dispatch Calculation and Grid-forming Distribution

Next, the Python script used to calculate the different dispatches and grid-forming distribution is highlighted. The entire code can be seen in section C.4, and it is part of the Python script named "Grid-Forming-Flexibility-implementation." The most important parts of the code, the lines necessary to change certain calculation parameters, and the calculation steps will be presented. For easier identification of which Python script is represented in the figures, snapshots from this script will be labeled with (Calculations) in the figure captions.

The overview of the first function in the setup can be seen in Figure C.20. Here, a few key aspects are important. First, the different generation forms based on forming and following are split into three separate parts; the PV following is split by itself because of the usage of the PV system component for the grid-following PV units instead of a static generator representation.

For these codes, three similar codes are created representing calculations for cases A, B, and C. These were not extended toward cases D, E, and F because they have the same dispatch percentages as the first three cases. However, if this is to be extended, this structural difference between these different case representations can be followed for extension with respect to changing columns and cells in the Excel file, etc.

Next, it can be seen that the different resources undergo scaling, utilizing a float from Excel. These Excel cells correspond to those shown in Figure C.18, which decide the percentage of generation for each of the three renewable sources. Thus, this Python script depends on the Excel sheet "Scenario-input." After scaling, the total installed capacities of the sources are found and summed for each of the different initialized sources. This extends to all the grid-following and grid-forming resources.

After these resources are added together for the overall system capacity, the system utilizes the grid-forming percentage, once more found in Figure C.18, to distribute the generation from the different resources concerning the share of grid-forming and grid-following percentage. This can be seen in Figure C.21, highlighted in blue. This is the "Grid-forming %" highlighted in Figure C.18. This then scales the different resources based on the grid-forming and capacity percentages. It can also be seen that row 3 here is used as the maximum capacity, and thus, these scenario-input Excel values need to be updated if the maximum capacity values of the system change.

```python
# %% Case A Grid-forming integration

def calculate_and_set_capacities_with_scaling_CaseA(sheet, case_row):
    # Read grid-forming and grid-following capacities from Excel ranges
    statgen_forming = sheet.Range("A199:E301").Value
    statgen_following = sheet.Range("A60:E117").Value
    statgen_PV_following = sheet.Range("A118:E162").Value

    # Initialize total capacities for different converter types
    total_Converter_PV = 0
    total_Converter_WOL = 0
    total_Converter_WOZ = 0
    total_W_WOZ = 0
    total_W_WOL = 0
    total_PV_following = 0

    # Read the scaling factors from Excel for the specified case row and convert percentages to decimals
    scaling_factors_WOZ = float(sheet.Range("AH21").Value) / 100
    scaling_factors_WOL = float(sheet.Range("AI21").Value) / 100
    scaling_factors_PV = float(sheet.Range("AJ21").Value) / 100

    # Sum capacities for grid-forming converters based on their type
    for row in statgen_forming:
        if row[0] and row[0].startswith("Converter_PV"):
            total_Converter_PV += row[3]
        elif row[0] and row[0].startswith("Converter_WOL"):
            total_Converter_WOL += row[3]
        elif row[0] and row[0].startswith("Converter_WOZ"):
            total_Converter_WOZ += row[3]

    # Sum capacities for grid-following converters based on their type
    for row in statgen_following:
        if row[0] and row[0].startswith("W_WOZ"):
            total_W_WOZ += row[3]
        elif row[0] and row[0].startswith("W_WOL"):
            total_W_WOL += row[3]
```

Figure C.20: Snapshot of dispatches and grid-forming distribution (Calculations).

```python
    # Read the grid-forming percentage from Excel cell AI33 and convert to decimal
    grid_forming_percentage = float(sheet.Range("AI33").Value) / 100

    # Apply scaling factors to the total capacities
    scaled_WOL_total = total_Converter_WOL * scaling_factors_WOL
    scaled_WOZ_total = total_Converter_WOZ * scaling_factors_WOZ
    scaled_solar_total = total_Converter_PV * scaling_factors_PV

    # Calculate new levels of grid-forming and following capacities
    grid_forming_WOL = scaled_WOL_total * grid_forming_percentage
    grid_following_WOL = scaled_WOL_total * (1 - grid_forming_percentage)
    grid_forming_WOZ = scaled_WOZ_total * grid_forming_percentage
    grid_following_WOZ = scaled_WOZ_total * (1 - grid_forming_percentage)
    grid_forming_solar = scaled_solar_total * grid_forming_percentage
    grid_following_solar = scaled_solar_total * (1 - grid_forming_percentage)
```

Figure C.21: Snapshot of dispatches and grid-forming distribution, slide 2 (Calculations).

Next, the distribution of the aforementioned dispatch calculations is transferred to the Excel sheet. The four functions utilized are shown in Figure C.22. In blue, the first function is highlighted. This is the main function, with the other functions compiled within it. First, the maximum load percentage and the column to update are chosen. This maximum load percentage is based on the maximum capacity columns in Excel, previously represented as "row 3" in Python. This can be adjusted based on the maximum allowable load percentage of the different resources. Next, the column update is selected as column 5 because this corresponds to the calculations for Case A.

The pink square highlights the zero-out function. This function simply zeros out the ranges for the different resources to avoid any overlap when running the script with earlier cases.

The green and yellow squares represent the remaining functions that distribute and update the Excel file with the newly calculated dispatches. This is done concerning the grid-forming and grid-following capacity distribution and the maximum loading percentage. Here, the maximum capacity column in Excel is used. Nothing needs to be changed between the different case calculations.

A snapshot representing the function calls of the aforementioned functions is seen in Figure C.23. Here, all the distributions and calculations of the aforementioned functions are performed. The only new and informative part comes from the randomized distribution. This allows for a random distribution concerning which converters are allocated as grid-following and grid-forming. This ensures there is no overlap. However, it is important to remember that since this distribution is randomized, the allocations of the different resources will vary each time the script is run. Therefore, it is crucial to save the dispatches if you want to maintain an accurate representation of the calculations for later analysis. This is the reason for the storage of the three different grid-forming penetration levels, as seen in Figure C.17.

Various print statements are also included throughout the code to facilitate easier understanding and comparison of different levels during system calculations.

```
def distribute_capacity_based_on_percentage(sheet, grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar, grid_forming_percentage):
    max_load_percentage = 1
    update_column = 5  # Column "F" in Excel, adjusted for Python indexing

    # Zero out the grid-forming and grid-following capacities in Excel
    def zero_out_ranges():
        ranges = [
            "F244:F267",  # grid_forming_range_WOL
            "F60:F83",    # grid_following_range_WOL
            "F268:F301",  # grid_forming_range_WOZ
            "F84:F117",   # grid_following_range_WOZ
            "F199:F301",  # grid_forming_range_solar
            "F118:F162"   # grid_following_range_solar
        ]
        for cell_range in ranges:
            for cell in sheet.Range(cell_range):
                cell.Value = 0

    zero_out_ranges()

    # Function to manage and update the distribution of capacities
    def distribute_and_update(forming_converters, following_converters, forming_capacity, following_capacity, converter_type):
        # Determine the number of grid-forming converters based on a predefined percentage
        num_forming = int(len(forming_converters) * grid_forming_percentage)
        # Select the subset of grid-forming converters
        forming_subset = forming_converters[:num_forming]

        # Identify names of unused grid-forming converters
        unused_forming_names = {conv[1][0].replace("Converter_", "") for conv in forming_converters[num_forming:]}

        # Select eligible grid-following converters that are not used in forming
        eligible_following_subset = [conv for conv in following_converters if conv[1][0].split("_", 1)[1] in unused_forming_names]

        # Distribute the grid-forming capacity among the forming subset
        distribute_capacity(forming_subset, forming_capacity, "forming", converter_type)

        # Distribute the grid-following capacity among the eligible following subset
        distribute_capacity(eligible_following_subset, following_capacity, "following", converter_type)

    # Function to distribute a specified capacity among a subset of converters
    def distribute_capacity(converters_subset, total_capacity, mode, converter_type):
        # Calculate the total rating by summing the fourth element (index 3) of each converter's row in the subset
        total_rating = sum(row[1][3] for row in converters_subset)

        # Iterate over each converter in the subset along with its index
        for index, row in converters_subset:
            # Calculate the maximum capacity for the current converter based on its rating and a predefined max load percentage
            max_capacity = row[3] * max_load_percentage

            # Calculate the proportion of the total rating this converter contributes
            proportion = row[3] / total_rating if total_rating > 0 else 0
```

Figure C.22: Snapshot of dispatches and grid-forming distribution, slide 3 (Calculations).

```
185         # Prepare lists of converters and shuffle them to randomize distribution
186         forming_WOL_converters = [(index + 199, row) for index, row in enumerate(sheet.Range("A199:E301").Value) if "Converter_WOL" in row[0]]
187         forming_WOZ_converters = [(index + 199, row) for index, row in enumerate(sheet.Range("A199:E301").Value) if "Converter_WOZ" in row[0]]
188         forming_solar_converters = [(index + 199, row) for index, row in enumerate(sheet.Range("A199:E301").Value) if "Converter_PV" in row[0]]
189         following_WOL_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("A60:E117").Value) if "W_WOL" in row[0]]
190         following_WOZ_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("A60:E117").Value) if "W_WOZ" in row[0]]
191         following_solar_converters = [(index + 118, row) for index, row in enumerate(sheet.Range("A118:E162").Value) if "L_PV" in row[0]]
192
193         import random
194         random.shuffle(forming_WOL_converters)
195         random.shuffle(forming_WOZ_converters)
196         random.shuffle(forming_solar_converters)
197         random.shuffle(following_WOL_converters)
198         random.shuffle(following_WOZ_converters)
199         random.shuffle(following_solar_converters)
200
201         # Distribute capacities according to the adjusted strategy
202         distribute_and_update(forming_WOL_converters, following_WOL_converters, grid_forming_WOL, grid_following_WOL, "WOL")
203         distribute_and_update(forming_WOZ_converters, following_WOZ_converters, grid_forming_WOZ, grid_following_WOZ, "WOZ")
204         distribute_and_update(forming_solar_converters, following_solar_converters, grid_forming_solar, grid_following_solar, "Solar")
205
206     # Main execution logic for Case A
207     grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar = calculate_and_set_capacities_with_scaling_CaseA(sheet_II30502, 20)
208     grid_forming_percentage = float(sheet_II30502.Range("AI33").Value) / 100
209     distribute_capacity_based_on_percentage(sheet_II30502, grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar, grid_forming_percentage)
210     print("END CASE A")
211
```

Figure C.23: Snapshot of dispatches and grid-forming distribution, slide 4 (Calculations).

Overall, the code highlighted here calculates the total installed capacities of the three different renewable energy sources: offshore wind (WOZ), onshore wind (WOL), and solar PV (PV). It then calculates the allocated dispatches for these resources based on the percentages given in the Excel sheet, representing the different cases. After this, the grid-forming penetration level is determined and implemented into the calculations, as the Excel sheet indicates. These newly calculated overall dispatches for the system are then divided into grid-following or grid-forming representations based on the aforementioned grid-forming percentage levels. These resources are then randomly distributed across the different grid-forming and grid-following converters without overlap to account for the total generation dispatch percentage and grid-forming percentage of the system. This straightforward approach allows for easy customization and comparison concerning different levels of these resources while respecting the various capacities and overlaps.

It should be noted that the randomized function can be improved to better represent the converter controller type distribution. This could also be extended to serve as a control system for different grid-forming penetration levels and distributions. Thus, while this is a simple implementation, a more complex extension of the code can be made to allow for further research and innovation.

C.3.3.1. Flexibility Calculation and Implementation

Next, the codes for the flexibility calculation and allocation will be discussed. An overview of the function can be seen in Figure C.24. This function calculates and distributes the flexibility resources based on the set parameters.

This calculation is based on the remaining demand left in the system after the distribution of the generation sources and is calculated as a percentage of each flexibility resource. This allows for easy implementation of the necessary flexibility in the system while also enabling customization regarding the use of different resources.

In the figure, the blue square represents the function and the allocation of different flexibility resources based on their location in Excel. These are then distributed according to certain percentages, as seen in Figure C.18. This allows for customization of the different distributions of the flexibility resources, making it easier to compare the impact of the various resources. These percentages can be adjusted in Excel for easy modifications.

In the green square, more information is extracted from the Excel sheet, including the amount of demand or generation excess in the system, as seen in Figure C.15. This provides an easy estimation of the necessary level of flexibility in the system for the set operational characteristics. It is important to note that this is just an estimation, as the calculations do not account for the locations of the resources or power system losses. Therefore, adjustments might be necessary after the initial calculations. This can be done manually by adjusting the set ranges in the Excel sheet. Thus, despite the imperfect Python codes, their complementary nature with Excel allows for a simple and customizable model.

Then, as before, there is a zeroing sequence to avoid overlap between different scenario dispatches.

The purple square also relies on the Excel sheet, realigning the setup of the flexibility resources with respect to different parameters found in the Excel sheet. This makes it easier to utilize and read the flexibility resources in Python.

Furthermore, a similar maximum rating calculation is done for the different flexibility resources, as was done for the different renewable generation resources.

The final part of the code can be seen in Figure C.25. Here, the different resources are once again distributed based on their maximum rating. A condition regarding the available resources is determined depending on whether there is excess generation or demand in the system. This depends on certain resources that are only available for one of the scenarios, such as the fuel cells or batteries that are available for both but need an inversion to serve the correct operational characteristics.

After this determination, the different amounts of flexibility calculated as necessary are spread across the different available resources. This is based on the previously mentioned scaling factors of the different resources. Most of the resources are distributed evenly across all the locations, except for the interconnections, which are set independently. This is because a similar rating approach was used for all the other flexibility resources to streamline system efficiency, except for the interconnections, where a more accurate representation was necessary to better capture the related dynamics.

```
# %% Case A Flexibility resources

def manage_grid_flexibility_CaseA(sheet):
    # Fetch the scaling factors for surplus generation and surplus load directly from the cells
    scaling_factors = {
        'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, 'surplus_load': sheet.Range("AI30").Value / 100},
        'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load': sheet.Range("AJ30").Value / 100},
        'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only surplus generation is relevant
        'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load is relevant
        'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load is relevant
    }

    # Get the total generation left in the system
    total_generation_left = sheet.Range("F408").Value
    print(f"Total generation left in the system: {total_generation_left}")

    # Null out all dispatches before starting the allocation
    dispatch_ranges_to_clear = ['F37:F46', 'F163:F182', 'F183:F192', 'F315:F324', 'F306:F308', 'F49:F54', 'F355:F363']  # Proper ranges to clear
    for dispatch_range in dispatch_ranges_to_clear:
        num_instances = len(sheet.Range(dispatch_range).Value)
        sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]

    # Adjust resource definitions to handle interconnections as one, with updated range
    resource_info_columns = {
        'hydrogen': ('A37:E46', 'F37:F46', 4),
        'batteries': ('A163:E182', 'F163:F182', 4),
        'fuel_cell': ('A183:E192', 'F183:F192', 4),
        'electrolyzers': ('A315:E324', 'F315:F324', 3),
        'interconnections': ('A355:E363', 'F355:F363', 4),  # Updated range
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    # Calculate total maximum ratings for each resource and store dispatch ranges
    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Determine the condition based on total generation left
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in scaling_factors if condition in scaling_factors[resource])

    # Select resources based on grid condition
    available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
    resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left > 0 else []

    print("Allocating remaining generation to flexibility resources and updating Excel...")
```

Figure C.24: Snapshot of flexibility calculation and distribution function (Calculations).

```python
    # Allocate generation to each available resource
    for resource in available_resources:
        if resource not in total_max_ratings:
            continue

        scaling_factor = scaling_factors[resource].get(condition, 0)
        max_rating = total_max_ratings[resource]
        allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if total_scaling_factor > 0 else 0)

        # Invert allocation for certain resources under surplus generation condition
        if condition == 'surplus_generation' and resource in resources_needing_inversion:
            allocation = -min(abs(allocation), max_rating)
        else:
            allocation = min(abs(allocation), max_rating)

        ranges_to_update = dispatch_ranges[resource]

        # Update the Excel sheet with the calculated allocations
        for dispatch_range in ranges_to_update:
            resource_instances = sheet.Range(dispatch_range).Value
            num_instances = len(resource_instances)
            allocation_per_instance = allocation / num_instances if num_instances > 0 else 0

            if resource == 'interconnections':
                # Handle interconnections separately
                interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value if row[3] is not None]
                total_rating = sum(interconnection_ratings)
                print(f"Interconnection ratings: {interconnection_ratings}")
                base_row = 355  # Starting row for interconnections in Excel
                for i, rating in enumerate(interconnection_ratings):
                    # Calculate individual allocation
                    individual_allocation = min((allocation * rating) / total_rating, rating) if total_rating > 0 else 0
                    # Apply this allocation directly to the Excel cell
                    cell_address = f"F{base_row + i}"  # Construct the cell address dynamically
                    sheet.Range(cell_address).Value = individual_allocation
                    print(f"Updated interconnection {i+1} dispatch with allocation: {individual_allocation}")
            else:
                # Update the dispatch values for other resources
                updated_dispatch_values = [[allocation_per_instance] for _ in range(num_instances)]
                sheet.Range(dispatch_range).Value = updated_dispatch_values
                print(f"Updated {resource} dispatches in range {dispatch_range} with allocation: {allocation_per_instance}")

    print("Allocation process completed. Updated Excel with new dispatch values.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
manage_grid_flexibility_CaseA(sheet_II30502)
```

Figure C.25: Snapshot of flexibility calculation and distribution function, slide 2 (Calculations).

Overall, the code is very straightforward. However, due to the different allocation percentages directly implemented in the calculations, the code offers various services for different operational scenarios under investigation. This accounts for the desired percentages for the different flexibility resources, ensuring they do not overlap and are calculated individually to avoid errors. This can also be easily extended to investigate future control modes concerning smart-grid control of the different flexibility resources and the diverse impacts this can have on system stability and operation.

The overall function is easily replicated across different scenarios and cases. However, six different functions were implemented for the six cases in this study to allow for quick and easy adjustments. Simple adjustments, such as selecting different columns in the Excel sheet, were necessary to adapt to different cases. This can be further extended if necessary.

As before, the script depends on the Excel sheet "scenario-input," any changes made to either resource should be double-checked to confirm they do not impact the other.

C.3.3.2. Load Calculation and Implementation Code

Next, the function for calculating the system's loads is highlighted. The function utilized can be seen in Figure C.26 and Figure C.27, and it is part of the larger Grid-Forming-Flexibility-Implementation script, found in section C.4.

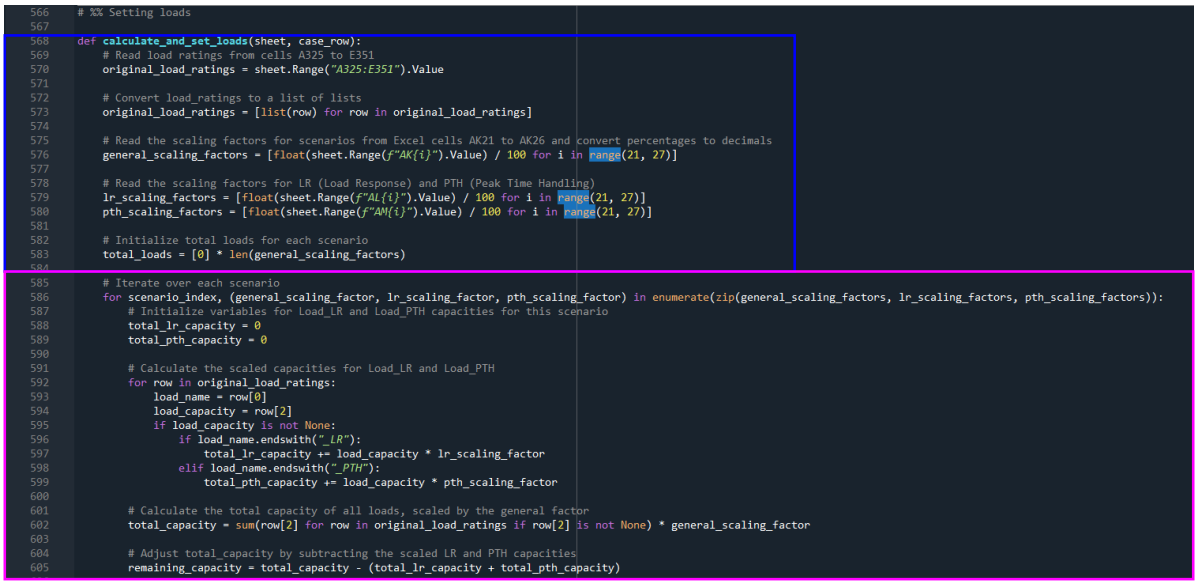
This code is very simple. It once again takes the loading distribution percentages as seen in Figure C.18. Here, they are represented by the "Demand," "Lr," and "Power 2 Heat" columns. All of these are given in percentages, allowing for easy scaling concerning the desired percentages for the different resources. This facilitates easy implementation for the analysis of different resources and load demands in the system.

The main connection between the load calculation and the determination of the load percentages

on the Excel sheet is seen in the blue square. They are directly intercepted from the Excel sheet before further calculations, and any necessary changes should be made here. The different resources are scaled based on their value, so if differentiation between the resources is desired while maintaining the same load percentage of the system, this needs to be manually accounted for in the other sources. The pink square, and the rest of the code seen in Figure C.27, represents the remaining implementation of the system. These are dependent on the different maximum ratings taken from row 2 instead of row 3 for the renewable resources due to the setup in Excel.

It should be noted that if the work involves analyzing different resources and their impacts, setting the loading above the maximum capacity in percentage might intervene with this part of the code. If this occurs, it can be easily adjusted by setting the maximum load of the different resources.

The overall load calculations are simple yet effective and very scalable concerning different demand levels and resource allocation. This allows for easier integration of different demand scenarios in the system and analysis of the impact of said resources.



```
# %% Setting loads

def calculate_and_set_loads(sheet, case_row):
    # Read load ratings from cells A325 to E351
    original_load_ratings = sheet.Range("A325:E351").Value

    # Convert load_ratings to a list of lists
    original_load_ratings = [list(row) for row in original_load_ratings]

    # Read the scaling factors for scenarios from Excel cells AK21 to AK26 and convert percentages to decimals
    general_scaling_factors = [float(sheet.Range(f"AK{i}").Value) / 100 for i in range(21, 27)]

    # Read the scaling factors for LR (Load Response) and PTH (Peak Time Handling)
    lr_scaling_factors = [float(sheet.Range(f"AL{i}").Value) / 100 for i in range(21, 27)]
    pth_scaling_factors = [float(sheet.Range(f"AM{i}").Value) / 100 for i in range(21, 27)]

    # Initialize total loads for each scenario
    total_loads = [0] * len(general_scaling_factors)

    # Iterate over each scenario
    for scenario_index, (general_scaling_factor, lr_scaling_factor, pth_scaling_factor) in enumerate(zip(general_scaling_factors, lr_scaling_factors, pth_scaling_factors)):
        # Initialize variables for Load_LR and Load_PTH capacities for this scenario
        total_lr_capacity = 0
        total_pth_capacity = 0

        # Calculate the scaled capacities for Load_LR and Load_PTH
        for row in original_load_ratings:
            load_name = row[0]
            load_capacity = row[2]
            if load_capacity is not None:
                if load_name.endswith("_LR"):
                    total_lr_capacity += load_capacity * lr_scaling_factor
                elif load_name.endswith("_PTH"):
                    total_pth_capacity += load_capacity * pth_scaling_factor

        # Calculate the total capacity of all loads, scaled by the general factor
        total_capacity = sum(row[2] for row in original_load_ratings if row[2] is not None) * general_scaling_factor

        # Adjust total_capacity by subtracting the scaled LR and PTH capacities
        remaining_capacity = total_capacity - (total_lr_capacity + total_pth_capacity)
```

Figure C.26: Snapshot of load calculation and implementation, (Calculations).

```
        # Calculate the sum of the unscaled regular loads for the proportional distribution
        sum_unscaled_loads = sum(row[2] for row in original_load_ratings if not row[0].endswith(("_LR", "_PTH")) and row[2] is not None)

        # Determine the column letter for scaled loads (column F for the first scenario)
        output_column_letter = chr(70 + scenario_index * 3)

        # Zero out the output column in Excel
        sheet.Range(f"{output_column_letter}325:{output_column_letter}351").Value = [[0]] * 27

        # Write the scaled load ratings to the Excel sheet
        for index, row in enumerate(original_load_ratings, start=325):
            load_name = row[0]
            load_capacity = row[2]
            if load_capacity is not None:
                scaled_capacity = load_capacity
                if load_name.endswith("_LR"):
                    scaled_capacity *= lr_scaling_factor
                elif load_name.endswith("_PTH"):
                    scaled_capacity *= pth_scaling_factor
                else:
                    # Scale the individual load proportionally to its part of the sum of unscaled loads
                    load_proportional_scale = (load_capacity / sum_unscaled_loads) * remaining_capacity
                    scaled_capacity = load_proportional_scale

                # Write the scaled capacity to the appropriate cell in Excel
                sheet.Cells(index, ord(output_column_letter) - 64).Value = scaled_capacity

        # Add the total scaled loads (including LR and PTH) to the total loads
        total_loads[scenario_index] = remaining_capacity + total_lr_capacity + total_pth_capacity

    return total_loads

# Assuming sheet_II30502 is a valid reference to your Excel sheet
total_loads = calculate_and_set_loads(sheet_II30502, 20)
print("Total Loads for all scenarios:", total_loads)
```

Figure C.27: Snapshot of load calculation and implementation, slide 2 (Calculations).

C.3.3.3. Reactive and Apparent Power Calculations

Finally, different reactive and apparent power values are calculated and distributed for the various resources. The code can be seen in Figure C.28. This script calculates the reactive and apparent power based on the different cases and resources' active power and power factor. While this currently does not have further implementation in the PowerFactory model, it serves as a structuring tool for the Excel sheet and allows for easy integration if these levels are needed in future work.

```python
# %% Finding apparent and reactive power


def calculate_multiple_scenarios(sheet):
    # Constants for the first scenario
    initial_active_power_column = 6  # Column F for active power dispatch
    initial_apparent_power_column = 7  # Column G for apparent power (S)
    initial_reactive_power_column = 8  # Column H for reactive power (Q)
    power_factor_column = 5  # Column E for power factor, remains constant

    # Loop through 6 scenarios
    for scenario in range(6):
        # Calculate column indices for current scenario
        active_power_column = initial_active_power_column + 3 * scenario
        apparent_power_column = initial_apparent_power_column + 3 * scenario
        reactive_power_column = initial_reactive_power_column + 3 * scenario

        # Retrieve active power dispatch from the current scenario's column
        active_power_dispatch_range = sheet.Range(sheet.Cells(10, active_power_column), sheet.Cells(363, active_power_column))
        active_power_dispatch_values = active_power_dispatch_range.Value

        # Power factor range remains constant
        power_factor_range = sheet.Range(sheet.Cells(10, power_factor_column), sheet.Cells(363, power_factor_column))
        power_factor_values = power_factor_range.Value

        # Calculate and write S and Q for each row in the current scenario
        for row_index, (active_power_dispatch, power_factor) in enumerate(zip(active_power_dispatch_values, power_factor_values), start=10):
            if isinstance(active_power_dispatch[0], (int, float)) and isinstance(power_factor[0], (int, float)):
                # Extract values from Excel as floats
                active_power_dispatch = float(active_power_dispatch[0])
                power_factor = float(power_factor[0])

                # Calculate apparent power (S) using the formula S = P / pf
                apparent_power = active_power_dispatch / power_factor

                # Calculate reactive power (Q) using the formula Q = (S^2 - P^2)^0.5
                if apparent_power ** 2 - active_power_dispatch ** 2 >= 0:
                    reactive_power = (apparent_power ** 2 - active_power_dispatch ** 2) ** 0.5
                else:
                    reactive_power = 0

                # Write calculated values to the scenario's S and Q columns in Excel
                sheet.Cells(row_index, apparent_power_column).Value = apparent_power
                sheet.Cells(row_index, reactive_power_column).Value = reactive_power

        print(f"Scenario {scenario + 1}: Calculations completed.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
calculate_multiple_scenarios(sheet_II30502)
```

Figure C.28: Snapshot of apparent and reactive power (Calculations).

C.3.4. PowerFactory Implementation Codes

Next, the codes utilized to implement and change different resources in PowerFactory are discussed. This is part of the two different functions presented in the Excel sheet presentation regarding the installed capacities and the dispatch integration into PowerFactory, as discussed in subsection C.3.1 and C.3.2.

These parts of the codes will not be represented again, as the necessary information regarding this structure is found in the respective sections. However, the remaining codes vary from different control system implementations in the model to various dynamic model changes. The code includes many ancillary services that were utilized to make the case studies possible. Most of these are easily changed to adapt to the parameters and characteristics required for certain analyses.
The full code can be found in section C.5.

The overall script is based on the work of Midhuna, with the addition of multiple new functions to allow for customizable options for the Dynamic models.

In the figure captions, snapshots from this Python script will be labeled with (Calculations) for easier identification of which Python script is represented in the figures.

C.3.4.1. Reactive Power Control

The first function in the Scenario-2050 script after the capacities and dispatches function is the reactive power control function, seen in Figure C.29. This was primarily taken from Midhunas' previous work. However, ranges in Excel and the setup of the different resources implemented in the work were also adapted for the control modes, making it more flexible with respect to the updated model.

Here, the different control modes of the system are selected. This is mainly separated by which generation units in the system will control voltage and which will have a constant reactive power control, known as "Constant V" or "Constant Q" in PowerFactory.

It is possible to select either PV or Wind as a control mode. The PV control mode sets the grid-forming PV converters to operate with a constant V, while all other static generators are set to have a constant Q, except for a few PV-following units.

The same is true for the Wind control mode. Here, the grid-forming onshore and offshore wind converter units are set with a constant V and some singular grid-following converters. The rest are set to constant Q.

For both control modes, the synchronous machines represented as nuclear and hydrogen power plants are not included in the control modes and are set as constant V. This can simply be integrated into the script if desired. Overall, the script is designed so that the specific controls of the different machines can be switched easily. The control modes are not fine-tuned, and based on the different dispatches in the system, some overlap in voltage control at certain busbars is possible. This can be manually changed, utilizing the code as a generalized setup, or further work could ensure no overlap occurs, making it a more flexible code.

In the figure, the parameters setup with respect to the information in Excel is seen in the pink square. Once again, if changes are made in the function or the Excel sheet, necessary adjustments must be made in the other program. Here, the different resources included in the control mode for the decision of control parameters are listed. Similar components can be included in future work if necessary, allowing for the easy inclusion of new resources.

Similar to the previous dispatches and capacities functions, the control mode is also called in the same function call setup. The "reactivepowercontrol()" is uncommented, and the control mode is selected inside the parentheses, either "'Wind'" or "'PV'." The setup can be seen in subsubsection C.3.4.8.

```
726 #%% function for setting reactive power controls
727
728
729     # Functions for setting different control modes in PowerFactory
730     # PV: The grid is primarily dominated by PV generation, and these resources will focus on controlling the voltage
731     # Wind: Similar to PV control, but now wind is dominant
732     # It is possible to change the different control modes directly in the function to alter and try different control modes
733     # The control mode is selected at the end of the script, as a function call
734
735
736 def reactivepowercontrol(Controlmode):
737
738                                                     # Set the range of cells for the different sources
739     w1 = wbin.Worksheets("II3050")                  # Name of the worksheet
740     SymG_1 = w1.Range("A10:Z48").Value              # Synchronous generators
741     StatG_following = w1.Range("A60:Z117").Value    # All static generators with a "following" control mode
742     StatG_PV_following = w1.Range("A118:Z162").Value # All PV systems with a "following" control mode
743     StatG_battery = w1.Range("A163:Z182").Value     # Battery systems
744     StatG_fuelcell = w1.Range("A183:Z192").Value    # Fuel cells
745     StatG_forming = w1.Range("A199:Z301").Value     # All converters with a "forming" control mode
746     Load_1 = w1.Range("A315:Z351").Value            # Loads (Includes Power to heat and load response)
747     StatG_interconnection = w1.Range("A355:Z357").Value # HVDC interconnections
748     SymG_interconnection = w1.Range("A358:Z363").Value  # AC interconnections
749
750
751
752     if Controlmode == 'PV':                         # Control mode set to: PV (System is dominantly PV generation)
753
754         # Grid-Forming Converters
755         for row in StatG_forming:                   # For all the static generators with "forming" control
756             name = row[0]+'.ElmGenstat'
757             variable = app.GetCalcRelevantObjects(name)[0]
758             if variable.loc_name.startswith("Converter_PV"):  # Index to find the PV converters
759                 variable.av_mode = 'constv'         # Sets control mode to Constant V in PowerFactory
760             else:
761                 variable.av_mode = 'constq'         # If the converter is not a "Converter_PV", a control mode tof Constant Q is set
762
```

Figure C.29: Snapshot of reactive power control function, (Scenario-2050).

C.3.4.2. Grid-forming Controller Selection

Next, the script includes a function to select the grid-forming controllers for the system. This function selects the grid-forming controllers for the entire system, creating a uniform controller structure across the different grid-forming controllers. For future work, the impact of different shares of each controller could be explored to see if it has any significant effects. Currently, the function simply decides which of the three types of grid-forming controllers is selected and utilized for the system. The function can be seen in Figure C.30. The blue square highlights the different control options, and the rest of the code sets

each dynamic controller representation either in or out of service, depending on the controller selection.

Similar to the other functions, the controller selection is specified in the function call part of the script, as seen in subsubsection C.3.4.8. Here, the "gridformingcontrol()" function is uncommented, and one of the three controllers is specified inside the parentheses: ('VSM'), ('Synchroverter'), or ('Droop'). This allows for easy implementation of the different controllers, which was crucial for this study.

However, not all grid-forming converters should be in service simultaneously due to the distribution concerning different grid-forming and grid-following converter representations. This is managed with a separate function that also handles the grid-following converters. These two functions need to be executed in conjunction with each other to ensure successful integration of the different grid-forming and grid-following converters. This is further elaborated in the coming subsection.

```python
# %% Control selection of Grid-forming control type

# This function controls the selection of the grid-forming control type for composite models in the power system simulation.
# Available control modes include VSM, Droop, and Synchroverter. Alternatively, "none" can be selected to put all controllers in service.
# The desired control selection is defined at the end of the script, as a function call.

def grid_forming_control(control_mode):
    """
    Control the selection of the grid-forming control type for composite models.

    Parameters:
        control_mode (str): The desired control mode ('VSM', 'Droop', 'Synchroverter', or 'none').
    """

    # Get all composite models (.ElmComp*) in the project
    composite_models = app.GetCalcRelevantObjects("*.ElmComp*")

    # Define the available control modes
    control_modes = ['VSM', 'Droop', 'Synchroverter']

    # Check if the provided control_mode is in the list of control modes
    if control_mode not in control_modes:
        app.PrintInfo(f"Invalid control mode: {control_mode}. Please choose from 'VSM', 'Droop', 'Synchroverter'.")
        return

    # Iterate over each composite model
    for model in composite_models:
        model_name = model.loc_name

        # Check if the model name starts with the control mode prefix to be enabled
        if model_name.startswith(control_mode):
            # Set the chosen control mode in service
            model.outserv = 0
            app.PrintInfo(f"Control enabled for model: {model_name}")
        # Check if the model name starts with another control mode's prefix and disable it
        elif any(model_name.startswith(other_mode) for other_mode in control_modes if other_mode != control_mode):
            # Set the other control modes out of service
            model.outserv = 1
            app.PrintInfo(f"Control disabled for model: {model_name}")

```

Figure C.30: Snapshot of grid-forming controller selection function, (Scenario-2050).

C.3.4.3. Controller Operation Selection

Next, the controller operation selection function is explained. The goal is to ensure that the correct converters in the system are in operation and that converter controllers with a dispatch of 0 MW are set out of service to avoid incorrect interactions with the dynamic models. While the static generator representation of the converters is set to be out of service when the active power is zero, the dynamic models require a different approach. This function addresses that need.

The function can be seen in Figure C.31 and Figure C.32. The blue square represents the first operation of the function, which involves setting all the different grid-following converters in operation. This step ensures there is no overlap between previous scenarios, allowing for an accurate function each time. This is done for the grid-following WOZ and WOL units, where the controller starts with "IEC," and for the grid-following solar PV units.

Then, the different ranges representing the different converters in the Excel sheet are given. This allows for easy identification of the converters with zero dispatch. The three squares in Figure C.32 show this identification and subsequent controller operation selection. A representation is provided for all the grid-forming and grid-following converters. The selection is based on the dispatch levels of Case A, represented as "row" 5 in Python or column 5 in Excel. If the controller operation selection

is to be performed for different cases or scenarios, this variable needs to change to the corresponding dispatches, typically located by a 3-step increase. This is highlighted in the yellow square.

Similar to the previous functions, this function is called in the function call part of the script, as seen in subsubsection C.3.4.8. The function "setconvertercontrolsoutofservice()" is uncommented in the correct case calling. It is important to uncomment the previous function that selects the grid-forming controller as well to avoid overlap between different dispatches.

```
# %% Converter selection for operation


def set_converter_controls_out_of_service():
    """
    Here, all the different converters are set out of service, based on if they have a dispatch or not.
    Converters without a dispatch is set out of service.
    All the grid forming converters are set in service in the "Control selection of Grid-forming control type" Cell. (Right above). Thus, these dont need to be set in service here.
    """
    app.PrintPlain("Setting all 'Grid-following' converters in service...")
    composite_models = app.GetCalcRelevantObjects("*.ElmComp*")
    for model in composite_models:
        if model.loc_name.startswith("IEC_"):
            model.outserv = 0  # Set in service (assuming 0 is in service)
            app.PrintPlain(f"Converter '{model.loc_name}' set in service.")
    for model in composite_models:
        if model.loc_name.startswith("WECC Large-scale PV Plant_"):
            model.outserv = 0  # Set in service (assuming 0 is in service)
            app.PrintPlain(f"Converter '{model.loc_name}' set in service.")

    # Name of the worksheet
    w1 = wbin.Worksheets("II3050")

    # All static generators with a "following" control mode
    StatG_following = w1.Range("A60:Z117").Value

    # All converters with a "forming" control mode
    StatG_forming = w1.Range("A199:Z301").Value

    # All converters with a "following" control mode for PV
    StatG_PV_following = w1.Range("A118:Z162").Value

    composite_models = app.GetCalcRelevantObjects("*.ElmComp*")
```

Figure C.31: Snapshot of controller operation selection function, (Scenario-2050).

```python
# Find converters in StatG_forming with zero output
app.PrintPlain("Checking converters in StatG_forming...")
for row in StatG_forming:
    name = row[0]
    if float(row[5]) == 0:  # Check if dispatch is zero
        app.PrintPlain(f"Converter '{name}' identified with zero output.")
        # Set associated controllers out of service
        control_name_prefix = name.split("_", 1)[1]  # Extract the name after "Converter_"
        for control_type in ["VSM", "Synchroverter Control", "Droop Control System"]:
            control_name = f"{control_type}_{control_name_prefix}"
            control = [c for c in composite_models if c.loc_name == control_name]
            if control:
                control[0].outserv = 1  # Set control mode out of service
                app.PrintPlain(f"Control mode '{control_name}' set out of service.")

# Find converters in StatG_following with zero output
app.PrintPlain("Checking converters in StatG_following...")
for row in StatG_following:
    name = row[0]
    if name.startswith("W_") and float(row[5]) == 0:  # Check if dispatch is zero
        converter_name = name.split("_", 1)[1]  # Extract the name after "W_"
        app.PrintPlain(f"Converter '{converter_name}' identified with zero output.")
        control_name = f"IEC_{converter_name}"
        control = [c for c in composite_models if c.loc_name == control_name]
        if control:
            control[0].outserv = 1  # Set control mode out of service
            app.PrintPlain(f"Control mode '{control_name}' set out of service.")

# Check PV converters in StatG_PV_following with zero output and set controls out of service
app.PrintPlain("Checking PV converters in StatG_PV_following...")
for row in StatG_PV_following:
    name = row[0]
    if name.startswith("L_PV_") and float(row[5]) == 0:  # Check if dispatch is zero
        # Extract the unique part of the converter's name
        unique_name = name.replace("L_PV_", "")  # Removes the prefix "L_PV_"
        app.PrintPlain(f"PV Converter '{unique_name}' identified with zero output.")
        # Construct the control name using the unique part
        control_name = f"WECC Large-scale PV Plant_{unique_name}"
        # Find the control by name
        control = [c for c in composite_models if c.loc_name == control_name]
        if control:
            control[0].outserv = 1  # Set control mode out of service
            app.PrintPlain(f"Control mode '{control_name}' set out of service.")
```

Figure C.32: Snapshot of controller operation selection function, slide 2 (Scenario-2050).

When using the function, it is important to remember the specifics of setting some of the controllers in service. This applies to certain grid-following PV units. At the start of the work, there were issues with the dynamic models of some solar PV representations. This was resolved by creating a new representation of the controller, typically indexed with an additional (1) at the end of the controller names. Because of this indexing, there are slight issues when these controllers are set in service. Therefore, only the controllers with an indexed (1) should be set in operation for certain scenarios. An example of this is given in Figure C.33.

In the figure, the yellow square represents a controller where an additional controller had to be made. During the selection process of the controllers to be set in service, this needs to be double-checked. Either only the one with a (1) needs to be set in service, or both need to be set out of service. However, for the controllers indicated with a "2", this is not relevant. These represent an actual second converter at the specific location and do not correlate to the controller issues. These are seen in the blue square and can be left untouched.

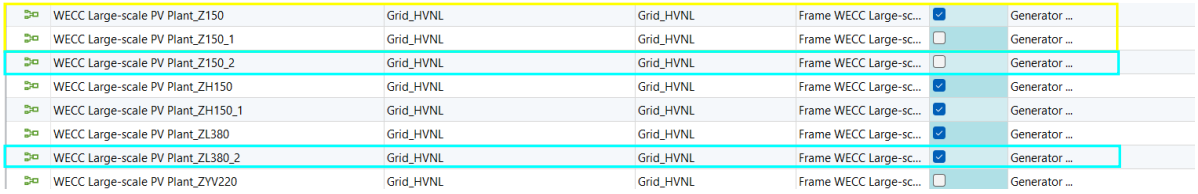


Figure C.33: Snapshot of the list of grid-following PV controllers in PowerFactory.

Overall, the function allows for a straightforward setup of the correct controllers' operational selection. However, it is important to remember the small adjustments necessary when running the grid-forming converter selection function simultaneously and being aware of the correct PV grid-following controllers to be set out of service. This structure could benefit from a simpler and more dynamic setup, and the different controllers in PowerFactory could be cleared properly.

C.3.4.4. Dynamic Load Model Update

Next, the function used to update the dynamic model representation of the different demand units is discussed. This includes load response, power to heat, and electrolysers utilizing a dynamic droop control controller.

The function can be seen in Figure C.34 and Figure C.35.

The goal is to update the dynamic models of the loads to allow for dynamic interplay with the model. Since these are not directly taken from the dispatches in PowerFactory, they are manually scripted through Python. The various parameters are given in the blue square, where values can be set directly. For more information regarding the different parameters, refer to subsection 6.4.4. These values can be changed easily, allowing for quick adaptation to different scenarios, enabling analysis of the impact of different parameters, and facilitating adjustments for future developments and new respective values.

The yellow square indicates where the dynamic update takes its values from, once again, the dispatches' Excel sheet. It takes the maximum capacity from row 2 and the active power levels from the dispatches and one of the rows in the cases, specifically case A, represented by row 5 in Python or column 5 in Excel. This needs to be changed for different case implementations of the dynamic models, moving three columns to the right between each case.

This function is called similarly to the others in this script in the function call section. The function "update dynamic model parameters()" is uncommented, and the script can be run. If this is altered, it is important to remember to change the dispatch column of the respective case.

```python
# %% Dynamic Load models update
# Updating the model parameters of the inverse droop controls (Electrolyser, Load response (LR) and Power to heat (PTH))

# For dynamic modeling, it does not use the load flow inputs, thus this needs to be added by its own. This is what the script does
# The different parameters are calculated based on the same scalings as the example control parameter values, these can be changed as necessary

def update_dynamic_model_parameters(Load_1):
    """
    Update dynamic model parameters for Electrolyser inverse droop controls based on specified load data.
    Load_1 is defined in the dispatch implementation, it is the excel range that describes the dispatches of the loads in the system (including the electrolysers, regular loads, power to heat and load response)
    """
    for row in Load_1:
        name_base = row[0]
        plini_value = float(row[5])       # Active power setpoint for the load. This needs to be changed based on the case simulated. Row [5] is the active power column for Case A in excel. This moves 3 columns to
        p_max_from_excel = float(row[2])  # Fetching P_max from the third column

        # Determine the model type
        model_type = ""
        if "Electrolyser" in name_base:
            model_type = "Electrolyser"
        elif "_LR" in name_base:
            model_type = "LR"
        elif "_PTH" in name_base:
            model_type = "PTH"

        name_base = name_base.replace("Electrolyser_", "").replace("Load_", "")
        dynamic_model_name = "Electrolyser Droop_" + name_base + ".ElmDsl"
        dynamic_models = app.GetCalcRelevantObjects(dynamic_model_name)

        if dynamic_models:  # If the model is found
            dynamic_model = dynamic_models[0]

            # Correctly call update_parameter for each parameter to be updated
            update_parameter(dynamic_model, 'P_ref', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'FCR_db', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'FCR_bid', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'grad', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'Bid_min', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'P_min', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'Bid_max', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'P_max', plini_value, model_type, p_max_from_excel, dynamic_model_name)
            # Repeat for other parameters as needed...
        else:
            app.PrintWarn(f"No dynamic model found with the name {dynamic_model_name}")
```

Figure C.34: Snapshot of dynamic load model implementation function(Scenario-2050).

```python
def update_parameter(dynamic_model, parameter_name, plini_value, model_type, p_max_from_excel, dynamic_model_name):
    """
    Update a specific parameter of a dynamic model.

    Parameters:
        dynamic_model (object): The dynamic model object to be updated.
        parameter_name (str): The name of the parameter to be updated.
        plini_value (float): The active power setpoint value.
        model_type (str): The type of the model (e.g., "Electrolyser", "LR", "PTH").
        p_max_from_excel (float): The maximum active power from the Excel sheet.
        dynamic_model_name (str): The name of the dynamic model.
    """
    # Conditional logic to handle each parameter's specific update requirements
    if parameter_name == 'FCR_db':
        value = 0.1 if model_type in ['Electrolyser', 'PTH'] else 0.15   # Different operational limits based on the type of dynamic load
    elif parameter_name == 'FCR_bid':
        value = 0.3 * plini_value
    elif parameter_name == 'grad':
        value = plini_value
    elif parameter_name == 'Bid_min':
        value = -plini_value
    elif parameter_name == 'P_min':
        value = 0
    elif parameter_name == 'Bid_max':
        value = plini_value
    elif parameter_name == 'P_max':
        value = p_max_from_excel
    else:
        value = plini_value  # Default case, for P_ref and potentially other straightforward updates

    setattr(dynamic_model, parameter_name, value)
    app.PrintInfo(f"Updated {parameter_name} to {value} for model {dynamic_model.Loc_name} ({model_type})")
```

Figure C.35: Snapshot of dynamic load model implementation function, slide 2 (Scenario-2050).

C.3.4.5. Grid-forming Current Limitation Function

Next, the grid-forming current limitation function is presented. The function can be seen in Figure C.36. This function allows for the adjustment of the maximum current limitation of the virtual impedance and output voltage controllers, enabling the analysis of the impact of changing the maximum current limit of the converters.

The virtual impedance and output voltage controllers are the main components used to limit the maximum current of the converters in the event of a disturbance. While the current limits are low for the existing brand of converters, this could change with future technology, making it important to analyze the impact. This function allows for easy integration of different characteristics for future systems.

Two different options are available for both functions, determining the mode in which the current limitation should operate.

The two options for the virtual impedance shown in the blue square are a constant impedance mode or a proportional impedance mode. For a more detailed explanation of these modes, refer to subsection 2.2.4. A value of 1 represents proportional impedance, while 0 represents constant impedance.

For the output voltage calculation, the two modes consist of the current limitation being enabled, represented by a value of 1, or disabled, represented by a value of 0.

The two models also allow for a change in the maximum current, seen right below the mode selection. This allows the script to easily adapt and modify the models with direct values selected in Python. The function is structured simply and can be extended easily to change other parameters in the respective controls.

Similar to the other functions, this function is called in the function call section, seen in subsubsection C.3.4.8. The function "update all virtual impedance modes" is uncommented, and the script is run. The only important thing to remember is to change the values as desired and to change them back after usage.

```
# %% Virtual impedance and output voltage calculation parameters adjustment
# Here, the different virtual impedance and output voltage calculation parameters are adjusted

def update_all_virtual_impedance_modes():
    """
    Update parameters for all Virtual Impedance and Output Voltage Calculation models in the project.
    The primary focus here is the maximum short-circuit current, and the current limitation mode of the controllers
    It is  also possible to test for different impedance values etc, but this needs to be implemented. This can be done in the same manner.
    """
    # Retrieve all DSL model objects in the project.
    all_dsl_models = app.GetCalcRelevantObjects("*.ElmDsl")

    for model in all_dsl_models:
        try:
            # Update parameters for "Virtual Impedance" models
            if "Virtual Impedance" in model.loc_name:
                # model.SetAttribute('e:params:r', 0.006)
                model.SetAttribute('e:params:Mode', 1)
                model.SetAttribute('e:params:i_lim', 1.1)  # 1.1 originally
                app.PrintInfo(f"Virtual Impedance model updated: {model.loc_name}")

            # Update parameters for "Output Voltage Calculation" models
            elif "Output Voltage Calculation" in model.loc_name:
                model.SetAttribute('e:params:Mode', 1)
                model.SetAttribute('e:params:i_max', 1.2)  # 1.2 originally
                app.PrintInfo(f"Output Voltage Calculation model updated: {model.loc_name}")

        except AttributeError as e:
            app.PrintWarn(f"Could not update model {model.loc_name}: {e}")
```

Figure C.36: Snapshot of current limitation function for grid-forming converters (Scenario-2050).

C.3.4.6. Grid-forming Controller Parameter Selection Function

Next, the three functions used to change the different parameters for the three grid-forming converters are explained. The first function, representing the changes for the VSM controller, can be seen in Figure C.37. The same functions are available for the Synchroverter and the Droop controller, with their respective parameters available for selection.

For the VSM controller update, two different values can be changed: the T_a and the D_p, as seen in the pink square. These are the two most vital parameters for the operation of the VSM controller. Similar parameters are selected for the Synchroverter and the Droop controller, with the different parameters better explained in section 2.2. These different parameter values can be changed as needed depending on the necessary simulation characteristics. These functions were critical in Case Study B to allow for easy integration of different parameters for the parameter sensitivity analysis. This can be extended to other parameters for the three different controllers.

The parameters for all the different grid-forming controllers are changed, even if they are set in service or not. This allows for easy integration if some of these controllers are to be set in service, as they do not pose any threat to different system characteristics when set out of service.

Similar to the previous functions, these three functions are also available to call in the system function call section seen in subsubsection C.3.4.8. The three different functions (updateallVSMcontrolsettings()), (updateallSynchrovertercontrolsettings()), and (updateallDroopcontrolsettings()) are uncommented, and the script can be run depending on which controller parameters are to be changed.

```python
# %% Update grid-forming controller parameters

"""
Here, the main parameters of the controllers are selected and are able to be adjusted to what value is wanted.
Here, all the parameters of the controllers can be added easily, by doing it in the same manner as done here.
The original values are also given, to easier allow for a reset.
"""

def update_all_VSM_control_settings():
    """
    Update parameters for all Virtual Synchronous Machine control models in the project.
    """
    # Retrieve all DSL model objects in the project.
    all_dsl_models = app.GetCalcRelevantObjects("*.ElmDsl")

    for model in all_dsl_models:
        try:
            # Update parameters for "Droop Control" models
            if "Virtual Synchronous Machine" in model.loc_name:
                model.SetAttribute('e:params:Ta', 3)     # Original: 3
                model.SetAttribute('e:params:Dp', 100)   # Original: 100
                app.PrintInfo(f"VSM Control model updated: {model.loc_name}")

        except AttributeError as e:
            app.PrintWarn(f"Could not update model {model.loc_name}: {e}")
```

Figure C.37: Snapshot of VSM parameter update function (Scenario-2050).

C.3.4.7. Kinetic Energy and Inertia Constant Update Function

Next, the function utilized to determine the inertia constants and kinetic energy levels represented by the AC interconnections in the system is explained. Having this represented in a Python script allows for quick and easy integration of different characteristics, facilitating efficient research on the projected levels of kinetic energy and inertia constants in future systems.

Starting off, the first part of the function is seen in Figure C.38. The interconnections and their respective ratings are initialized here, as shown in the pink square. Furthermore, the different kinetic energy levels given in GVA and inertia constants given in seconds, which are desired for system representation, are provided in the blue square. These values are easy to change, and the rest of the script handles the necessary integration and calculations. The new apparent power rating of the synchronous machine for the different interconnections is found by dividing the kinetic energy by the inertia constant.

The first function that updates the values is shown in Figure C.39. The green square represents the values changed in the different interconnections: the rated apparent power and the inertia constant. This is done for the synchronous machine representing the different interconnections, with the parameters being those of the synchronous machine.

Next, the calculations and implementation of the synchronous machine controller are seen in Figure C.40. Here, the different interconnections' limitations are managed, as the synchronous machines are now represented with a large apparent power rating. This is controlled in the dynamic model. The interconnections and their capacities are listed in the red square. Based on these values, the various parameters of the synchronous machine controllers are calculated and implemented into the dynamic models, as seen in the yellow square. These updated values represent the active power dispatches of the synchronous machine controllers, determining the maximum valve opening time constant of the synchronous machine controller dispatch. This is based on the ratings of the machines instead of the kinetic energy representation, allowing for a more accurate estimation. The maximum active power dispatch is based on the active power ratings of the interconnections, while the maximum valve opening time constant was set to 0.25 of that value. This was the standard value found for the controller and was thus kept for the study. It can be changed if necessary.

For more information on the interconnection representations, refer to subsection 6.4.3.

These functions are called in the function call section, as seen in subsubsection C.3.4.8. The two functions called "updateselectedsynchronousmachines()" and "updateieeeg1dslmodels()" are uncom-

mented in the function call, and the parameters will be updated accordingly.

Having this easy integration of different kinetic energy and inertia constant levels in the system allows for a good analysis of these parameters' different values. This can be used to analyze the system performance under different operational characteristics and their impact on system performance. This was done in Case Study B, but the work can be extended further by simply defining the different values in the blue square in Figure C.38.

```
# %% Update the inertia and kinetic energy levels

"""
Here, the inertia constant and kinetic energy levels of the AC interconnections can be
Select the kinetic energy and inertia constant wanted for the system, and this will be
The dynamic models of the AC interconnections also needs to be updated, to be responsi
"""

# Initialize known interconnection levels for each specific synchronous machine
interconnection_levels = {
    "IC_AMPRION_GNA380_1": 1650,
    "IC_AMPRION_NDR380_1": 1650,
    "IC_AMPRION_OBZ380": 1650,
    "IC_ELIA_MBT380_1": 1750,
    "IC_ELIA_ZVL380_1": 1750,
    "IC_TenneT_DIL380_1": 5000
}

# Set the kinetic energy and inertia constant values directly
kinetic_energy = 1800000  # Replace with the desired value in GVA
inertia_constant = 4.1  # Replace with the desired value in seconds
new_sgn_value = kinetic_energy / inertia_constant
```

Figure C.38: Snapshot of Kinetic energy and inertia constant function, (Scenario-2050).

```
def update_selected_synchronous_machines():
    """
    Update selected synchronous machine parameters.
    """
    # Retrieve all relevant TypSym models in the project
    all_typ_sym = app.GetCalcRelevantObjects("*.TypSym")

    # Names of interest for filtering
    target_names = [
        "Gen_Interconnection_1650",
        "Gen_Interconnection_1750",
        "Gen_Interconnection_5000"
    ]

    # Loop through all TypSym models and filter by name
    for typ_sym in all_typ_sym:
        try:
            # Check if the object's name matches any in the target list
            if typ_sym.loc_name in target_names:
                # Update the 'sgn' attribute (rated apparent power)
                typ_sym.SetAttribute('sgn', new_sgn_value)

                # Update the 'h' attribute (inertia constant)
                typ_sym.SetAttribute('h', inertia_constant)

                app.PrintInfo(f"Updated {typ_sym.loc_name} with sgn={new_sgn_value} and h={inertia_constant}")
        except AttributeError as e:
            app.PrintWarn(f"Could not update {typ_sym.loc_name}: {e}")
```

Figure C.39: Snapshot of Kinetic energy and inertia constant function, slide 2 (Scenario-2050).

```python
# Function to update the IEEEG1 DSL models within each relevant ElmComp
def update_ieeeg1_dsl_models():
    """
    Update parameters for all IEEEG1 models in the project.
    """
    # Base interconnection level to model name mapping
    interconnection_levels = {
        "Plant_Gen_Interconnection_DIL380": 5000,
        "Plant_Gen_Interconnection_GNA380": 1650,
        "Plant_Gen_Interconnection_MBT380_1": 1750,
        "Plant_Gen_Interconnection_NDR380_1": 1650,
        "Plant_Gen_Interconnection_OBZ380": 1650,
        "Plant_Gen_Interconnection_ZVL380_1": 1750
    }

    # Gather all ElmComp objects in the project
    all_elmcomps = app.GetCalcRelevantObjects("*.ElmComp")

    # Log the total number of ElmComp objects found
    app.PrintInfo(f"Total ElmComp components found: {len(all_elmcomps)}")

    for comp in all_elmcomps:
        # Check if component matches any relevant interconnection plant
        if comp.loc_name in interconnection_levels.keys():
            app.PrintInfo(f"Processing IEEEG1 models in {comp.loc_name}")

            # Fetch the interconnection value based on the component name
            base_value = interconnection_levels[comp.loc_name]
            app.PrintInfo(f"Using base value: {base_value} for calculations.")

            # Calculate parameters based on the given kinetic energy and inertia
            uc = -((0.25 * base_value) / new_sgn_value)
            pmin = -((base_value) / new_sgn_value)
            uo = abs(uc)
            pmax = abs(pmin)

            # Find all IEEEG1 DSL models within this component
            ieeeg1_models = comp.GetContents("*.ElmDsl", recursive=True)

            if ieeeg1_models:
                app.PrintInfo(f"Found {len(ieeeg1_models)} IEEEG1 models in {comp.loc_name}")
                for model in ieeeg1_models:
                    if "IEEEG1" in model.loc_name:
                        # Update the model parameters
                        model.SetAttribute("e:params:Uc", uc)
                        model.SetAttribute("e:params:Pmin", pmin)
                        model.SetAttribute("e:params:Uo", uo)
                        model.SetAttribute("e:params:Pmax", pmax)
                        app.PrintInfo(f"Updated IEEEG1 model: {model.loc_name} with Uc={uc}, Pmin={pmin}, Uo={uo}, Pmax={pmax}")
            else:
                app.PrintInfo(f"No IEEEG1 models found in {comp.loc_name}")
```

Figure C.40: Snapshot of Kinetic energy and inertia constant function, slide 3 (Scenario-2050).

C.3.4.8. Function Call

Next, the function call section of the script is highlighted and explained. This section allows for the selection and operation of different functions.

An overview of the function call for Case A can be seen in Figure C.41. Similar function calls exist for all six cases with the same possible function calls. Some function calls are case-dependent, such as the different dispatches with six different functions, while other functions are standardized across all the different cases. Some functions require specific changes within the functions themselves, such as choosing the correct column in Excel to utilize. For specific instructions regarding this, refer to the descriptions of the respective functions.

To run the different functions, they are simply uncommented in the specific case, and the script is ready to be run. It is possible to initialize all the different cases simultaneously; however, it is recommended to do this individually to allow for better control of the different system characteristics and initialization. Four different rows are uncommented for Case A in the function call section seen in Figure C.41. The first uncommented line is the activation of the specific case, ensuring that all the functions are only called for that specific case in PowerFactory. Next, the capacities2050() function and the two functions used to initialize the kinetic energy and inertia constant levels of the interconnections are uncommented. If this script is run, it will activate Case A, set the system capacities, and update the interconnections to the specified kinetic energy and inertia constant levels.
If the "simulations()" function is uncommented, it will simply perform a load flow at the end of the function call.

A similar approach can be used for the different cases. To avoid any overlap in the script, it is

recommended to comment out the activation of the case studies not in use. After this, it is simply a matter of uncommenting the functions to be used, and the script is ready to run.

The function call section allows for an easy overview of the specific functions utilized each time the script is run and facilitates easy extension with the addition of new functions.

```
#%%% Simulations for 2050
# This section activates different scenarios for the year 2050, setting capacities, controls, and dispatches accordingly.
# If a scenario needs to be commented out/in, make sure the whole relevant section is commented out, as shown in the guidelines.

# Retrieve operational scenario folders
OSfolder = app.GetProjectFolder('scen')
OS = OSfolder.GetContents()

# Loop through each operational scenario folder
for opscenfold in OS:
    app.PrintPlain(opscenfold)
    # Check if the folder corresponds to the desired scenario
    if opscenfold.loc_name == 'II3050-2':
        sheetname = "II30502"  # Define the sheet name in the file "Scenario Input Data.xlsx"
        opscen = opscenfold.GetContents()  # Get the contents of the operational scenario folder
        # Loop through each scenario in the folder
        for scen in opscen:
            # Activate the scenario and perform relevant updates
            if scen.loc_name == 'CaseA':
                scen.Activate()
                capacities2050()
                # dispatches1()
                # reactivepowercontrol('Wind')
                # grid_forming_control('VSM')
                # set_converter_controls_out_of_service()
                # update_all_droop_control_settings()
                # update_all_VSM_control_settings()
                # update_all_Syncroverter_control_settings()
                # update_all_virtual_impedance_modes()
                update_selected_synchronous_machines()
                update_ieeeg1_dsl_models()
                # simulations()
```

Figure C.41: Snapshot of function call section (Scenario-2050).

C.3.4.9. Running the Script

Finally, a quick guide on how to utilize the script in PowerFactory is provided. The first step in running the script involves correctly utilizing the system function call as previously described. The next part involves running the script in PowerFactory, as shown here.

The first step is selecting the script execution function in PowerFactory, marked in the pink square in Figure C.42. After this is selected, the menu shown in Figure C.43 should appear. The "Scenario 2050" script is identified and selected here, as seen in the blue square. After this, the menu shown in Figure C.44 should appear. Note that the correct settings should be selected as the "Script," shown in the yellow square. Additionally, selecting the correct path for the Python script is important, as this automatically changes for different computers with different script locations. This is changed by selecting the three dots in the green square and selecting the correct script. After this is done, the script is ready to be executed, as indicated in the red square.

It is important to remember that all of the different resources of Excel, Python, and PowerFactory should be in the same model as given in section C.2. Otherwise, problems can occur.

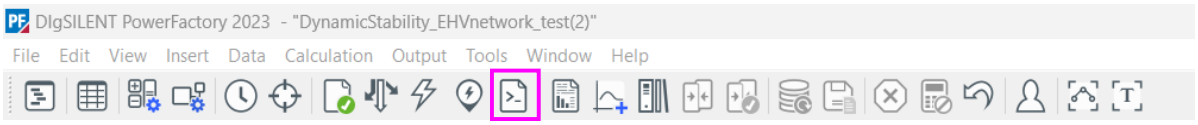


Figure C.42: Snapshot of execute script location in PowerFactory.

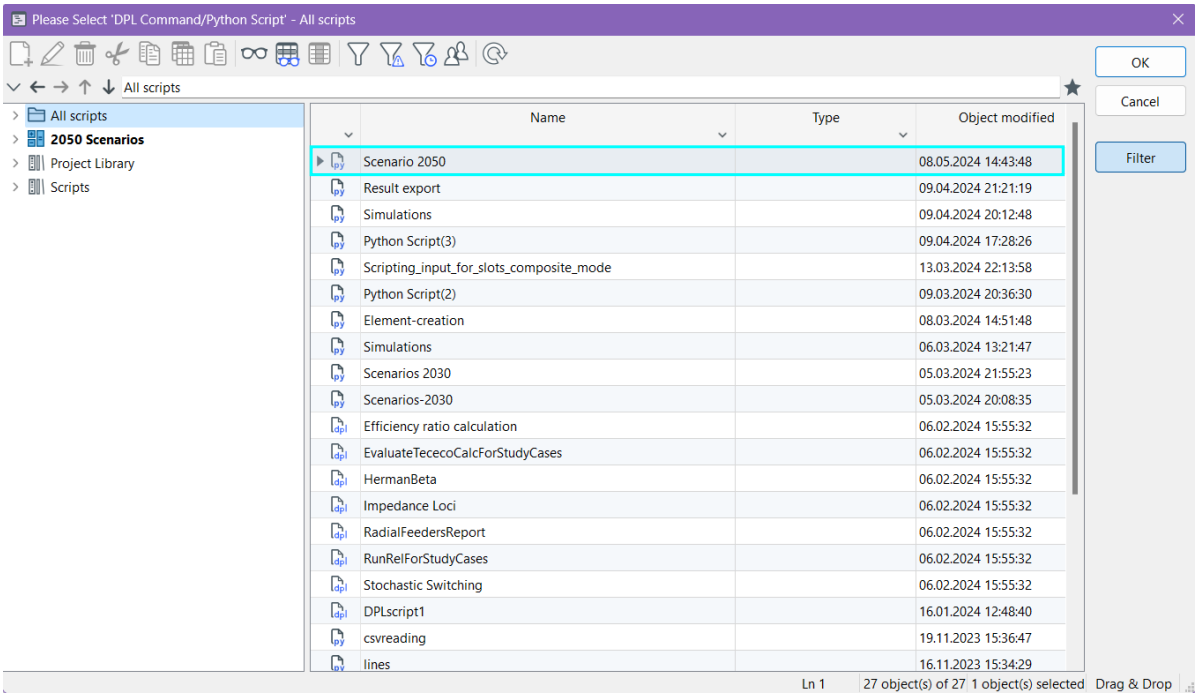


Figure C.43: Snapshot of specific script location in PowerFactory.

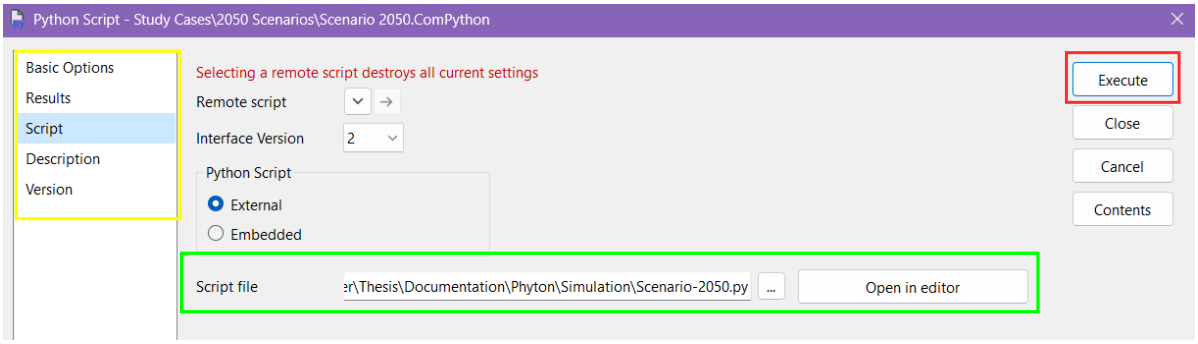


Figure C.44: Snapshot of running the script in PowerFactory.

C.4. Grid-forming and Flexibility Calculation Codes

Listing C.1: Grid-forming and Flexibility implementation codes

```
# -*- coding: utf-8 -*-
"""
Created on Tue Feb 25 19:37:42 2024

@author: Sander
"""
import win32com.client as win32
import math
import random


# %% Initialization
excel = win32.gencache.EnsureDispatch('Excel.Application')
excel.Visible = True

# Define paths to your Excel files
input_excel = r"G:\Andre datamaskiner\Min berbare datamaskin\Master\Thesis\Documentation\
    Phyton\Simulation\Scenario-Input-2-with-dispatched-case-b.xlsx"

# input_excel = r"G:\Andre datamaskiner\Min berbare datamaskin\Master\Thesis\Documentation\
    Phyton\Simulation\Scenario-Input-2-with-dispatched-case-b-test.xlsx"
# Add paths for output_excel and installedcap_excel as necessary

# Open the Excel workbooks
wbin = excel.Workbooks.Open(input_excel)

sheet_II30502 = wbin.Worksheets("II3050")


# %% Case A Grid-forming integration

def calculate_and_set_capacities_with_scaling_CaseA(sheet, case_row):
    # Read grid-forming and grid-following capacities from Excel ranges
    statgen_forming = sheet.Range("A199:E301").Value
    statgen_following = sheet.Range("A60:E117").Value
    statgen_PV_following = sheet.Range("A118:E162").Value

    # Initialize total capacities for different converter types
    total_Converter_PV = 0
    total_Converter_WOL = 0
    total_Converter_WOZ = 0
    total_W_WOZ = 0
    total_W_WOL = 0
    total_PV_following = 0

    # Read the scaling factors from Excel for the specified case row and convert percentages
        to decimals
    scaling_factors_WOZ = float(sheet.Range("AH21").Value) / 100
    scaling_factors_WOL = float(sheet.Range("AI21").Value) / 100
    scaling_factors_PV = float(sheet.Range("AJ21").Value) / 100

    # Sum capacities for grid-forming converters based on their type
    for row in statgen_forming:
        if row[0] and row[0].startswith("Converter_PV"):
            total_Converter_PV += row[3]
        elif row[0] and row[0].startswith("Converter_WOL"):
            total_Converter_WOL += row[3]
        elif row[0] and row[0].startswith("Converter_WOZ"):
            total_Converter_WOZ += row[3]

    # Sum capacities for grid-following converters based on their type
    for row in statgen_following:
        if row[0] and row[0].startswith("W_WOZ"):
            total_W_WOZ += row[3]
        elif row[0] and row[0].startswith("W_WOL"):
            total_W_WOL += row[3]

```

```
    # Sum capacities for PV grid-following converters
    for row in statgen_PV_following:
        if row[0]:  # Check if the identifier is not None
            total_PV_following += row[3]

    # Ensure totals are numerical values (floats)
    total_Converter_PV = float(total_Converter_PV)
    total_Converter_WOL = float(total_Converter_WOL)
    total_Converter_WOZ = float(total_Converter_WOZ)
    total_W_WOZ = float(total_W_WOZ)
    total_W_WOL = float(total_W_WOL)
    total_PV_following = float(total_PV_following)

    # Read the grid-forming percentage from Excel cell AI33 and convert to decimal
    grid_forming_percentage = float(sheet.Range("AI33").Value) / 100

    # Apply scaling factors to the total capacities
    scaled_WOL_total = total_Converter_WOL * scaling_factors_WOL
    scaled_WOZ_total = total_Converter_WOZ * scaling_factors_WOZ
    scaled_solar_total = total_Converter_PV * scaling_factors_PV

    # Calculate new levels of grid-forming and following capacities
    grid_forming_WOL = scaled_WOL_total * grid_forming_percentage
    grid_following_WOL = scaled_WOL_total * (1 - grid_forming_percentage)
    grid_forming_WOZ = scaled_WOZ_total * grid_forming_percentage
    grid_following_WOZ = scaled_WOZ_total * (1 - grid_forming_percentage)
    grid_forming_solar = scaled_solar_total * grid_forming_percentage
    grid_following_solar = scaled_solar_total * (1 - grid_forming_percentage)

    # Print the original and scaled totals for verification
    print("Case A")
    print("Original and Scaled Totals for Wind and Solar PV:")
    print(f"Total WOL (Grid Forming): {total_Converter_WOL}")
    print(f"Total WOL (Grid Following): {total_W_WOL}")
    print(f"Total WOZ (Grid Forming): {total_Converter_WOZ}")
    print(f"Total WOZ (Grid Following): {total_W_WOZ}")
    print("-" * 20)  # Separator for clarity in the output
    print(f"WOL scale: {scaling_factors_WOL}")
    print(f"WOZ scale: {scaling_factors_WOZ}")
    print(f"Scaled WOL Total: {scaled_WOL_total}")
    print(f"Scaled WOZ Total: {scaled_WOZ_total}")
    print("-" * 20)  # Separator for clarity in the output
    print(f"Total Solar PV (Grid Forming): {total_Converter_PV}")
    print(f"Total Solar PV (Grid Following): {total_PV_following}")
    print(f"PV scale: {scaling_factors_PV}")
    print(f"Scaled Solar PV Total: {scaled_solar_total}")
    print("-" * 20)  # Separator for clarity in the output
    print("Original and Scaled Totals for Wind and Solar PV with Grid Forming/Following
        Distribution:")
    print(f"Grid forming percentage: {grid_forming_percentage*100} % ")
    print(f"Grid Forming WOL: {grid_forming_WOL}")
    print(f"Grid Following WOL: {grid_following_WOL}")
    print(f"Grid Forming WOZ: {grid_forming_WOZ}")
    print(f"Grid Following WOZ: {grid_following_WOZ}")
    print(f"Grid Forming Solar PV: {grid_forming_solar}")
    print(f"Grid Following Solar PV: {grid_following_solar}")

    # Return the distributed totals if needed for further processing
    return grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ,
        grid_forming_solar, grid_following_solar

def distribute_capacity_based_on_percentage(sheet, grid_forming_WOL, grid_following_WOL,
    grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar,
    grid_forming_percentage):
    max_load_percentage = 1
    update_column = 5  # Column "F" in Excel, adjusted for Python indexing

    # Zero out the grid-forming and grid-following capacities in Excel
    def zero_out_ranges():
        ranges = [
            "F244:F267",  # grid_forming_range_WOL
```

```
            "F60:F83",    # grid_following_range_WOL
            "F268:F301",  # grid_forming_range_WOZ
            "F84:F117",   # grid_following_range_WOZ
            "F199:F301",  # grid_forming_range_solar
            "F118:F162"   # grid_following_range_solar
        ]
        for cell_range in ranges:
            for cell in sheet.Range(cell_range):
                cell.Value = 0

    zero_out_ranges()

    # Function to manage and update the distribution of capacities
    def distribute_and_update(forming_converters, following_converters, forming_capacity,
        following_capacity, converter_type):
        # Determine the number of grid-forming converters based on a predefined percentage
        num_forming = int(len(forming_converters) * grid_forming_percentage)
        # Select the subset of grid-forming converters
        forming_subset = forming_converters[:num_forming]

        # Identify names of unused grid-forming converters
        unused_forming_names = {conv[1][0].replace("Converter_", "") for conv in
            forming_converters[num_forming:]}

        # Select eligible grid-following converters that are not used in forming
        eligible_following_subset = [conv for conv in following_converters if conv[1][0].
            split("_", 1)[1] in unused_forming_names]

        # Distribute the grid-forming capacity among the forming subset
        distribute_capacity(forming_subset, forming_capacity, "forming", converter_type)

        # Distribute the grid-following capacity among the eligible following subset
        distribute_capacity(eligible_following_subset, following_capacity, "following",
            converter_type)

    # Function to distribute a specified capacity among a subset of converters
    def distribute_capacity(converters_subset, total_capacity, mode, converter_type):
        # Calculate the total rating by summing the fourth element (index 3) of each
            converter's row in the subset
        total_rating = sum(row[1][3] for row in converters_subset)

        # Iterate over each converter in the subset along with its index
        for index, row in converters_subset:
            # Calculate the maximum capacity for the current converter based on its rating
                and a predefined max load percentage
            max_capacity = row[3] * max_load_percentage

            # Calculate the proportion of the total rating this converter contributes
            proportion = row[3] / total_rating if total_rating > 0 else 0

            # Determine the desired load for this converter based on its proportion of the
                total capacity
            desired_load = total_capacity * proportion

            # Set the cell value to the smaller of the desired load or the maximum capacity
                for the converter
            sheet.Cells(index, update_column + 1).Value = min(desired_load, max_capacity)

        # Print the total capacity distributed for the given converter type and mode
        print(f"Total {converter_type} {mode} Capacity Distributed: {total_capacity}")

    # Prepare lists of converters and shuffle them to randomize distribution
    forming_WOL_converters = [(index + 199, row) for index, row in enumerate(sheet.Range("
        A199:E301").Value) if "Converter_WOL" in row[0]]
    forming_WOZ_converters = [(index + 199, row) for index, row in enumerate(sheet.Range("
        A199:E301").Value) if "Converter_WOZ" in row[0]]
    forming_solar_converters = [(index + 199, row) for index, row in enumerate(sheet.Range("
        A199:E301").Value) if "Converter_PV" in row[0]]
    following_WOL_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("
        A60:E117").Value) if "W_WOL" in row[0]]
```

```
    following_WOZ_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("A60:E117").Value) if "W_WOZ" in row[0]]
    following_solar_converters = [(index + 118, row) for index, row in enumerate(sheet.Range("A118:E162").Value) if "L_PV" in row[0]]

    import random
    random.shuffle(forming_WOL_converters)
    random.shuffle(forming_WOZ_converters)
    random.shuffle(forming_solar_converters)
    random.shuffle(following_WOL_converters)
    random.shuffle(following_WOZ_converters)
    random.shuffle(following_solar_converters)

    # Distribute capacities according to the adjusted strategy
    distribute_and_update(forming_WOL_converters, following_WOL_converters, grid_forming_WOL, grid_following_WOL, "WOL")
    distribute_and_update(forming_WOZ_converters, following_WOZ_converters, grid_forming_WOZ, grid_following_WOZ, "WOZ")
    distribute_and_update(forming_solar_converters, following_solar_converters, grid_forming_solar, grid_following_solar, "Solar")

# Main execution logic for Case A
grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar = calculate_and_set_capacities_with_scaling_CaseA(sheet_II30502, 20)
grid_forming_percentage = float(sheet_II30502.Range("AI33").Value) / 100
distribute_capacity_based_on_percentage(sheet_II30502, grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar, grid_forming_percentage)
print("END CASE A")


# %% Case B Grid-forming integration

def calculate_and_set_capacities_with_scaling_CaseB(sheet, case_row):
    # Read grid-forming and grid-following capacities from Excel ranges
    statgen_forming = sheet.Range("A199:E301").Value  # Adjusted range for forming converters
    statgen_following = sheet.Range("A60:E117").Value
    statgen_PV_following = sheet.Range("A118:E162").Value

    # Initialize total capacities for different converter types
    total_Converter_PV = 0
    total_Converter_WOL = 0
    total_Converter_WOZ = 0
    total_W_WOZ = 0
    total_W_WOL = 0
    total_PV_following = 0

    # Read the scaling factors from Excel for the specified case row and convert percentages to decimals
    scaling_factors_WOZ = float(sheet.Range("AH22").Value) / 100  # Moved one row down
    scaling_factors_WOL = float(sheet.Range("AI22").Value) / 100  # Moved one row down
    scaling_factors_PV = float(sheet.Range("AJ22").Value) / 100  # Same row

    # Sum capacities for grid-forming converters based on their type
    for row in statgen_forming:
        if row[0] and row[0].startswith("Converter_PV"):
            total_Converter_PV += row[3]
        elif row[0] and row[0].startswith("Converter_WOL"):
            total_Converter_WOL += row[3]
        elif row[0] and row[0].startswith("Converter_WOZ"):
            total_Converter_WOZ += row[3]

    # Sum capacities for grid-following converters based on their type
    for row in statgen_following:
        if row[0] and row[0].startswith("W_WOZ"):
            total_W_WOZ += row[3]
        elif row[0] and row[0].startswith("W_WOL"):
            total_W_WOL += row[3]

    # Sum capacities for PV grid-following converters
```

```
    for row in statgen_PV_following:
        if row[0]:  # Check if the identifier is not None
            total_PV_following += row[3]

    # Ensure totals are numerical values (floats)
    total_Converter_PV = float(total_Converter_PV)
    total_Converter_WOL = float(total_Converter_WOL)
    total_Converter_WOZ = float(total_Converter_WOZ)
    total_W_WOZ = float(total_W_WOZ)
    total_W_WOL = float(total_W_WOL)
    total_PV_following = float(total_PV_following)

    # Read the grid-forming percentage from Excel cell AI33 and convert to decimal
    grid_forming_percentage = float(sheet.Range("AI33").Value) / 100

    # Apply scaling factors to the total capacities
    scaled_WOL_total = total_Converter_WOL * scaling_factors_WOL
    scaled_WOZ_total = total_Converter_WOZ * scaling_factors_WOZ
    scaled_solar_total = total_Converter_PV * scaling_factors_PV

    # Calculate new levels of grid-forming and following capacities
    grid_forming_WOL = scaled_WOL_total * grid_forming_percentage
    grid_following_WOL = scaled_WOL_total * (1 - grid_forming_percentage)
    grid_forming_WOZ = scaled_WOZ_total * grid_forming_percentage
    grid_following_WOZ = scaled_WOZ_total * (1 - grid_forming_percentage)
    grid_forming_solar = scaled_solar_total * grid_forming_percentage
    grid_following_solar = scaled_solar_total * (1 - grid_forming_percentage)

    # Print the original and scaled totals for verification
    print("Case B")
    print("Original and Scaled Totals for Wind and Solar PV:")
    print(f"Total WOL (Grid Forming): {total_Converter_WOL}")
    print(f"Total WOL (Grid Following): {total_W_WOL}")
    print(f"Total WOZ (Grid Forming): {total_Converter_WOZ}")
    print(f"Total WOZ (Grid Following): {total_W_WOZ}")
    print("-" * 20)  # Separator for clarity in the output
    print(f"WOL scale: {scaling_factors_WOL}")
    print(f"WOZ scale: {scaling_factors_WOZ}")
    print(f"Scaled WOL Total: {scaled_WOL_total}")
    print(f"Scaled WOZ Total: {scaled_WOZ_total}")
    print("-" * 20)  # Separator for clarity in the output
    print(f"Total Solar PV (Grid Forming): {total_Converter_PV}")
    print(f"Total Solar PV (Grid Following): {total_PV_following}")
    print(f"PV scale: {scaling_factors_PV}")
    print(f"Scaled Solar PV Total: {scaled_solar_total}")
    print("-" * 20)  # Separator for clarity in the output
    print("Original and Scaled Totals for Wind and Solar PV with Grid Forming/Following
        Distribution:")
    print(f"Grid forming percentage: {grid_forming_percentage*100} % ")
    print(f"Grid Forming WOL: {grid_forming_WOL}")
    print(f"Grid Following WOL: {grid_following_WOL}")
    print(f"Grid Forming WOZ: {grid_forming_WOZ}")
    print(f"Grid Following WOZ: {grid_following_WOZ}")
    print(f"Grid Forming Solar PV: {grid_forming_solar}")
    print(f"Grid Following Solar PV: {grid_following_solar}")

    # Return the distributed totals if needed for further processing
    return grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ,
        grid_forming_solar, grid_following_solar

def distribute_capacity_based_on_percentage_CaseB(sheet, grid_forming_WOL, grid_following_WOL
    , grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar,
    grid_forming_percentage):
    max_load_percentage = 1
    update_column = 8  # Adjusted to column 8 (Column "I" in Excel, adjusted for Python
        indexing)

    # Zero out the grid-forming and grid-following capacities in Excel
    def zero_out_ranges():
        ranges = [
            "I244:I267",  # Adjusted ranges for Case B
```

```
            "I60:I83",    # Adjusted ranges for Case B
            "I268:I301",  # Adjusted ranges for Case B
            "I84:I117",   # Adjusted ranges for Case B
            "I199:I301",  # Adjusted ranges for Case B
            "I118:I162"   # Adjusted ranges for Case B
        ]
        for cell_range in ranges:
            for cell in sheet.Range(cell_range):
                cell.Value = 0

    zero_out_ranges()

    # Function to manage and update the distribution of capacities
    def distribute_and_update(forming_converters, following_converters, forming_capacity, following_capacity, converter_type):
        # Determine the number of grid-forming converters based on a predefined percentage
        num_forming = int(len(forming_converters) * grid_forming_percentage)
        # Select the subset of grid-forming converters
        forming_subset = forming_converters[:num_forming]

        # Identify names of unused grid-forming converters
        unused_forming_names = {conv[1][0].split("_", 1)[1] for conv in forming_converters[num_forming:]}
        eligible_following_subset = [conv for conv in following_converters if conv[1][0].split("_", 1)[1] in unused_forming_names]

        # Distribute the grid-forming capacity among the forming subset
        distribute_capacity(forming_subset, forming_capacity, "forming", converter_type)

        # Distribute the grid-following capacity among the eligible following subset
        distribute_capacity(eligible_following_subset, following_capacity, "following", converter_type)

    def distribute_capacity(converters_subset, total_capacity, mode, converter_type):
        # Calculate the total rating by summing the fourth element (index 3) of each converter's row in the subset
        total_rating = sum(row[1][3] for row in converters_subset)
        # Iterate over each converter in the subset along with its index
        for index, row in converters_subset:
            # Calculate the maximum capacity for the current converter based on its rating and a predefined max load percentage
            max_capacity = row[3] * max_load_percentage
            # Calculate the proportion of the total rating this converter contributes
            proportion = row[3] / total_rating if total_rating > 0 else 0
            # Determine the desired load for this converter based on its proportion of the total capacity
            desired_load = total_capacity * proportion
            # Set the cell value to the smaller of the desired load or the maximum capacity for the converter
            sheet.Cells(index, update_column + 1).Value = min(desired_load, max_capacity)
        # Print the total capacity distributed for the given converter type and mode
        print(f"Total {converter_type} {mode} Capacity Distributed: {total_capacity}")

    # Prepare lists of converters and shuffle them to randomize distribution
    forming_WOL_converters = [(index + 200, row) for index, row in enumerate(sheet.Range("A200:E301").Value) if "Converter_WOL" in row[0]]  # Adjusted range
    forming_WOZ_converters = [(index + 200, row) for index, row in enumerate(sheet.Range("A200:E301").Value) if "Converter_WOZ" in row[0]]  # Adjusted range
    forming_solar_converters = [(index + 200, row) for index, row in enumerate(sheet.Range("A200:E301").Value) if "Converter_PV" in row[0]]  # Adjusted range
    following_WOL_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("A60:E117").Value) if "W_WOL" in row[0]]
    following_WOZ_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("A60:E117").Value) if "W_WOZ" in row[0]]
    following_solar_converters = [(index + 118, row) for index, row in enumerate(sheet.Range("A118:E162").Value) if "L_PV" in row[0]]

    import random
    random.shuffle(forming_WOL_converters)
    random.shuffle(forming_WOZ_converters)
    random.shuffle(forming_solar_converters)
```

```
    random.shuffle(following_WOL_converters)
    random.shuffle(following_WOZ_converters)
    random.shuffle(following_solar_converters)

    # Distribute capacities according to the adjusted strategy
    distribute_and_update(forming_WOL_converters, following_WOL_converters, grid_forming_WOL,
        grid_following_WOL, "WOL")
    distribute_and_update(forming_WOZ_converters, following_WOZ_converters, grid_forming_WOZ,
        grid_following_WOZ, "WOZ")
    distribute_and_update(forming_solar_converters, following_solar_converters,
        grid_forming_solar, grid_following_solar, "Solar")

# Main execution logic for Case B
grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ,
    grid_forming_solar, grid_following_solar =
    calculate_and_set_capacities_with_scaling_CaseB(sheet_II30502, 20)
grid_forming_percentage = float(sheet_II30502.Range("AI33").Value) / 100
distribute_capacity_based_on_percentage_CaseB(sheet_II30502, grid_forming_WOL,
    grid_following_WOL, grid_forming_WOZ, grid_following_WOZ, grid_forming_solar,
    grid_following_solar, grid_forming_percentage)
print("END CASE B")



# %% Case C Grid-forming integration
def calculate_and_set_capacities_with_scaling_CaseC(sheet, case_row):
    # Read grid-forming and grid-following capacities from Excel ranges
    statgen_forming = sheet.Range("A199:E301").Value  # Adjusted range for forming converters
    statgen_following = sheet.Range("A60:E117").Value
    statgen_PV_following = sheet.Range("A118:E162").Value

    # Initialize total capacities for different converter types
    total_Converter_PV = 0
    total_Converter_WOL = 0
    total_Converter_WOZ = 0
    total_W_WOZ = 0
    total_W_WOL = 0
    total_PV_following = 0

    # Read the scaling factors from Excel for the specified case row and convert percentages
        to decimals
    scaling_factors_WOZ = float(sheet.Range("AH23").Value) / 100  # Moved one row down
    scaling_factors_WOL = float(sheet.Range("AI23").Value) / 100  # Moved one row down
    scaling_factors_PV = float(sheet.Range("AJ23").Value) / 100  # Same row

    # Sum capacities for grid-forming converters based on their type
    for row in statgen_forming:
        if row[0] and row[0].startswith("Converter_PV"):
            total_Converter_PV += row[3]
        elif row[0] and row[0].startswith("Converter_WOL"):
            total_Converter_WOL += row[3]
        elif row[0] and row[0].startswith("Converter_WOZ"):
            total_Converter_WOZ += row[3]

    # Sum capacities for grid-following converters based on their type
    for row in statgen_following:
        if row[0] and row[0].startswith("W_WOZ"):
            total_W_WOZ += row[3]
        elif row[0] and row[0].startswith("W_WOL"):
            total_W_WOL += row[3]

    # Sum capacities for PV grid-following converters
    for row in statgen_PV_following:
        if row[0]:  # Check if the identifier is not None
            total_PV_following += row[3]

    # Ensure totals are numerical values (floats)
    total_Converter_PV = float(total_Converter_PV)
    total_Converter_WOL = float(total_Converter_WOL)
    total_Converter_WOZ = float(total_Converter_WOZ)
    total_W_WOZ = float(total_W_WOZ)
```

```
    total_W_WOL = float(total_W_WOL)
    total_PV_following = float(total_PV_following)

    # Read the grid-forming percentage from Excel cell AI33 and convert to decimal
    grid_forming_percentage = float(sheet.Range("AI33").Value) / 100

    # Apply scaling factors to the total capacities
    scaled_WOL_total = total_Converter_WOL * scaling_factors_WOL
    scaled_WOZ_total = total_Converter_WOZ * scaling_factors_WOZ
    scaled_solar_total = total_Converter_PV * scaling_factors_PV

    # Calculate new levels of grid-forming and following capacities
    grid_forming_WOL = scaled_WOL_total * grid_forming_percentage
    grid_following_WOL = scaled_WOL_total * (1 - grid_forming_percentage)
    grid_forming_WOZ = scaled_WOZ_total * grid_forming_percentage
    grid_following_WOZ = scaled_WOZ_total * (1 - grid_forming_percentage)
    grid_forming_solar = scaled_solar_total * grid_forming_percentage
    grid_following_solar = scaled_solar_total * (1 - grid_forming_percentage)

    # Print the original and scaled totals for verification
    print("Case C")
    print("Original and Scaled Totals for Wind and Solar PV:")
    print(f"Total WOL (Grid Forming): {total_Converter_WOL}")
    print(f"Total WOL (Grid Following): {total_W_WOL}")
    print(f"Total WOZ (Grid Forming): {total_Converter_WOZ}")
    print(f"Total WOZ (Grid Following): {total_W_WOZ}")
    print("-" * 20)  # Separator for clarity in the output
    print(f"WOL scale: {scaling_factors_WOL}")
    print(f"WOZ scale: {scaling_factors_WOZ}")
    print(f"Scaled WOL Total: {scaled_WOL_total}")
    print(f"Scaled WOZ Total: {scaled_WOZ_total}")
    print("-" * 20)  # Separator for clarity in the output
    print(f"Total Solar PV (Grid Forming): {total_Converter_PV}")
    print(f"Total Solar PV (Grid Following): {total_PV_following}")
    print(f"PV scale: {scaling_factors_PV}")
    print(f"Scaled Solar PV Total: {scaled_solar_total}")
    print("-" * 20)  # Separator for clarity in the output
    print("Original and Scaled Totals for Wind and Solar PV with Grid Forming/Following
        Distribution:")
    print(f"Grid forming percentage: {grid_forming_percentage*100} % ")
    print(f"Grid Forming WOL: {grid_forming_WOL}")
    print(f"Grid Following WOL: {grid_following_WOL}")
    print(f"Grid Forming WOZ: {grid_forming_WOZ}")
    print(f"Grid Following WOZ: {grid_following_WOZ}")
    print(f"Grid Forming Solar PV: {grid_forming_solar}")
    print(f"Grid Following Solar PV: {grid_following_solar}")

    # Return the distributed totals if needed for further processing
    return grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ,
        grid_forming_solar, grid_following_solar

def distribute_capacity_based_on_percentage_CaseC(sheet, grid_forming_WOL, grid_following_WOL
    , grid_forming_WOZ, grid_following_WOZ, grid_forming_solar, grid_following_solar,
    grid_forming_percentage):
    max_load_percentage = 1
    update_column = 11  # Adjusted to column 11 (Column "L" in Excel, adjusted for Python
        indexing)

    # Zero out the grid-forming and grid-following capacities in Excel
    def zero_out_ranges():
        ranges = [
            "L244:L267",  # Adjusted ranges for Case C
            "L60:L83",    # Adjusted ranges for Case C
            "L268:L301",  # Adjusted ranges for Case C
            "L84:L117",   # Adjusted ranges for Case C
            "L199:L301",  # Adjusted ranges for Case C
            "L118:L162"   # Adjusted ranges for Case C
        ]
        for cell_range in ranges:
            for cell in sheet.Range(cell_range):
                cell.Value = 0
```

```
503
504     zero_out_ranges()
505
506     # Function to manage and update the distribution of capacities
507     def distribute_and_update(forming_converters, following_converters, forming_capacity,
            following_capacity, converter_type):
508         # Determine the number of grid-forming converters based on a predefined percentage
509         num_forming = int(len(forming_converters) * grid_forming_percentage)
510         # Select the subset of grid-forming converters
511         forming_subset = forming_converters[:num_forming]
512
513         # Identify names of unused grid-forming converters
514         unused_forming_names = {conv[1][0].split("_", 1)[1] for conv in forming_converters[
                num_forming:]}
515         eligible_following_subset = [conv for conv in following_converters if conv[1][0].
                split("_", 1)[1] in unused_forming_names]
516
517         # Distribute the grid-forming capacity among the forming subset
518         distribute_capacity(forming_subset, forming_capacity, "forming", converter_type)
519
520         # Distribute the grid-following capacity among the eligible following subset
521         distribute_capacity(eligible_following_subset, following_capacity, "following",
                converter_type)
522
523     def distribute_capacity(converters_subset, total_capacity, mode, converter_type):
524         # Calculate the total rating by summing the fourth element (index 3) of each
                converter's row in the subset
525         total_rating = sum(row[1][3] for row in converters_subset)
526         # Iterate over each converter in the subset along with its index
527         for index, row in converters_subset:
528             # Calculate the maximum capacity for the current converter based on its rating
                    and a predefined max load percentage
529             max_capacity = row[3] * max_load_percentage
530             # Calculate the proportion of the total rating this converter contributes
531             proportion = row[3] / total_rating if total_rating > 0 else 0
532             # Determine the desired load for this converter based on its proportion of the
                    total capacity
533             desired_load = total_capacity * proportion
534             # Set the cell value to the smaller of the desired load or the maximum capacity
                    for the converter
535             sheet.Cells(index, update_column + 1).Value = min(desired_load, max_capacity)
536         # Print the total capacity distributed for the given converter type and mode
537         print(f"Total␣{converter_type}␣{mode}␣Capacity␣Distributed:␣{total_capacity}")
538
539     # Prepare lists of converters and shuffle them to randomize distribution
540     forming_WOL_converters = [(index + 200, row) for index, row in enumerate(sheet.Range("
            A200:E301").Value) if "Converter_WOL" in row[0]]  # Adjusted range
541     forming_WOZ_converters = [(index + 200, row) for index, row in enumerate(sheet.Range("
            A200:E301").Value) if "Converter_WOZ" in row[0]]  # Adjusted range
542     forming_solar_converters = [(index + 200, row) for index, row in enumerate(sheet.Range("
            A200:E301").Value) if "Converter_PV" in row[0]]  # Adjusted range
543     following_WOL_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("
            A60:E117").Value) if "W_WOL" in row[0]]
544     following_WOZ_converters = [(index + 60, row) for index, row in enumerate(sheet.Range("
            A60:E117").Value) if "W_WOZ" in row[0]]
545     following_solar_converters = [(index + 118, row) for index, row in enumerate(sheet.Range(
            "A118:E162").Value) if "L_PV" in row[0]]
546
547     import random
548     random.shuffle(forming_WOL_converters)
549     random.shuffle(forming_WOZ_converters)
550     random.shuffle(forming_solar_converters)
551     random.shuffle(following_WOL_converters)
552     random.shuffle(following_WOZ_converters)
553     random.shuffle(following_solar_converters)
554
555     # Distribute capacities according to the adjusted strategy
556     distribute_and_update(forming_WOL_converters, following_WOL_converters, grid_forming_WOL,
            grid_following_WOL, "WOL")
557     distribute_and_update(forming_WOZ_converters, following_WOZ_converters, grid_forming_WOZ,
            grid_following_WOZ, "WOZ")
```

```
    distribute_and_update(forming_solar_converters, following_solar_converters,
        grid_forming_solar, grid_following_solar, "Solar")

# Main execution logic for Case C
grid_forming_WOL, grid_following_WOL, grid_forming_WOZ, grid_following_WOZ,
    grid_forming_solar, grid_following_solar =
    calculate_and_set_capacities_with_scaling_CaseC(sheet_II30502, 20)
grid_forming_percentage = float(sheet_II30502.Range("AI33").Value) / 100
distribute_capacity_based_on_percentage_CaseC(sheet_II30502, grid_forming_WOL,
    grid_following_WOL, grid_forming_WOZ, grid_following_WOZ, grid_forming_solar,
    grid_following_solar, grid_forming_percentage)
print("END CASE C")

# %% Setting loads

def calculate_and_set_loads(sheet, case_row):
    # Read load ratings from cells A325 to E351
    original_load_ratings = sheet.Range("A325:E351").Value

    # Convert load_ratings to a list of lists
    original_load_ratings = [list(row) for row in original_load_ratings]

    # Read the scaling factors for scenarios from Excel cells AK21 to AK26 and convert
        percentages to decimals
    general_scaling_factors = [float(sheet.Range(f"AK{i}").Value) / 100 for i in range(21,
        27)]

    # Read the scaling factors for LR (Load Response) and PTH (Peak Time Handling)
    lr_scaling_factors = [float(sheet.Range(f"AL{i}").Value) / 100 for i in range(21, 27)]
    pth_scaling_factors = [float(sheet.Range(f"AM{i}").Value) / 100 for i in range(21, 27)]

    # Initialize total loads for each scenario
    total_loads = [0] * len(general_scaling_factors)

    # Iterate over each scenario
    for scenario_index, (general_scaling_factor, lr_scaling_factor, pth_scaling_factor) in
        enumerate(zip(general_scaling_factors, lr_scaling_factors, pth_scaling_factors)):
        # Initialize variables for Load_LR and Load_PTH capacities for this scenario
        total_lr_capacity = 0
        total_pth_capacity = 0

        # Calculate the scaled capacities for Load_LR and Load_PTH
        for row in original_load_ratings:
            load_name = row[0]
            load_capacity = row[2]
            if load_capacity is not None:
                if load_name.endswith("_LR"):
                    total_lr_capacity += load_capacity * lr_scaling_factor
                elif load_name.endswith("_PTH"):
                    total_pth_capacity += load_capacity * pth_scaling_factor

        # Calculate the total capacity of all loads, scaled by the general factor
        total_capacity = sum(row[2] for row in original_load_ratings if row[2] is not None) *
             general_scaling_factor

        # Adjust total_capacity by subtracting the scaled LR and PTH capacities
        remaining_capacity = total_capacity - (total_lr_capacity + total_pth_capacity)

        # Calculate the sum of the unscaled regular loads for the proportional distribution
        sum_unscaled_loads = sum(row[2] for row in original_load_ratings if not row[0].
            endswith(("_LR", "_PTH")) and row[2] is not None)

        # Determine the column letter for scaled loads (column F for the first scenario)
        output_column_letter = chr(70 + scenario_index * 3)

        # Zero out the output column in Excel
        sheet.Range(f"{output_column_letter}325:{output_column_letter}351").Value = [[0]] *
            27

        # Write the scaled load ratings to the Excel sheet
        for index, row in enumerate(original_load_ratings, start=325):
```

```
            load_name = row[0]
            load_capacity = row[2]
            if load_capacity is not None:
                scaled_capacity = load_capacity
                if load_name.endswith("_LR"):
                    scaled_capacity *= lr_scaling_factor
                elif load_name.endswith("_PTH"):
                    scaled_capacity *= pth_scaling_factor
                else:
                    # Scale the individual load proportionally to its part of the sum of unscaled loads
                    load_proportional_scale = (load_capacity / sum_unscaled_loads) * remaining_capacity
                    scaled_capacity = load_proportional_scale

                # Write the scaled capacity to the appropriate cell in Excel
                sheet.Cells(index, ord(output_column_letter) - 64).Value = scaled_capacity

        # Add the total scaled loads (including LR and PTH) to the total loads
        total_loads[scenario_index] = remaining_capacity + total_lr_capacity + total_pth_capacity

    return total_loads

# Assuming sheet_II30502 is a valid reference to your Excel sheet
total_loads = calculate_and_set_loads(sheet_II30502, 20)
print("Total Loads for all scenarios:", total_loads)









# %% Case A Flexibility resources

def manage_grid_flexibility_CaseA(sheet):
    # Fetch the scaling factors for surplus generation and surplus load directly from the cells
    scaling_factors = {
        'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, 'surplus_load': sheet.Range("AI30").Value / 100},
        'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load': sheet.Range("AJ30").Value / 100},
        'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only surplus generation is relevant
        'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load is relevant
        'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load is relevant
    }

    # Get the total generation left in the system
    total_generation_left = sheet.Range("F408").Value
    print(f"Total generation left in the system: {total_generation_left}")

    # Null out all dispatches before starting the allocation
    dispatch_ranges_to_clear = ['F37:F46', 'F163:F182', 'F183:F192', 'F315:F324', 'F306:F308', 'F49:F54', 'F355:F363']  # Proper ranges to clear
    for dispatch_range in dispatch_ranges_to_clear:
        num_instances = len(sheet.Range(dispatch_range).Value)
        sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]

    # Adjust resource definitions to handle interconnections as one, with updated range
    resource_info_columns = {
        'hydrogen': ('A37:E46', 'F37:F46', 4),
        'batteries': ('A163:E182', 'F163:F182', 4),
        'fuel_cell': ('A183:E192', 'F183:F192', 4),
```

```
        'electrolyzers': ('A315:E324', 'F315:F324', 3),
        'interconnections': ('A355:E363', 'F355:F363', 4),  # Updated range
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    # Calculate total maximum ratings for each resource and store dispatch ranges
    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[
            column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Determine the condition based on total generation left
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in
        scaling_factors if condition in scaling_factors[resource])

    # Select resources based on grid condition
    available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if
        condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
    resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left
        > 0 else []

    print("Allocating␣remaining␣generation␣to␣flexibility␣resources␣and␣updating␣Excel...")

    # Allocate generation to each available resource
    for resource in available_resources:
        if resource not in total_max_ratings:
            continue

        scaling_factor = scaling_factors[resource].get(condition, 0)
        max_rating = total_max_ratings[resource]
        allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if
            total_scaling_factor > 0 else 0)

        # Invert allocation for certain resources under surplus generation condition
        if condition == 'surplus_generation' and resource in resources_needing_inversion:
            allocation = -min(abs(allocation), max_rating)
        else:
            allocation = min(abs(allocation), max_rating)

        ranges_to_update = dispatch_ranges[resource]

        # Update the Excel sheet with the calculated allocations
        for dispatch_range in ranges_to_update:
            resource_instances = sheet.Range(dispatch_range).Value
            num_instances = len(resource_instances)
            allocation_per_instance = allocation / num_instances if num_instances > 0 else 0

            if resource == 'interconnections':
                # Handle interconnections separately
                interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value
                    if row[3] is not None]
                total_rating = sum(interconnection_ratings)
                print(f"Interconnection␣ratings:␣{interconnection_ratings}")
                base_row = 355  # Starting row for interconnections in Excel
                for i, rating in enumerate(interconnection_ratings):
                    # Calculate individual allocation
                    individual_allocation = min((allocation * rating) / total_rating, rating)
                        if total_rating > 0 else 0
                    # Apply this allocation directly to the Excel cell
                    cell_address = f"F{base_row␣+␣i}"  # Construct the cell address
                        dynamically
                    sheet.Range(cell_address).Value = individual_allocation
```

```
                    print(f"Updated interconnection {i+1} dispatch with allocation: {
                        individual_allocation}")
            else:
                # Update the dispatch values for other resources
                updated_dispatch_values = [[allocation_per_instance] for _ in range(
                    num_instances)]
                sheet.Range(dispatch_range).Value = updated_dispatch_values
                print(f"Updated {resource} dispatches in range {dispatch_range} with 
                    allocation: {allocation_per_instance}")

    print("Allocation process completed. Updated Excel with new dispatch values.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
manage_grid_flexibility_CaseA(sheet_II30502)

# %% Case B Flexibility resources

def manage_grid_flexibility_CaseB(sheet):
    # Fetch the scaling factors for surplus generation and surplus load directly from the
        cells
    scaling_factors = {
        'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, '
            surplus_load': sheet.Range("AI30").Value / 100},
        'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load':
            sheet.Range("AJ30").Value / 100},
        'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only
            surplus generation is relevant
        'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load
            is relevant
        'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load
            is relevant
    }

    # Get the total generation left in the system
    total_generation_left = sheet.Range("i408").Value
    print(f"Total generation left in the system: {total_generation_left}")

    # Null out all dispatches before starting the allocation
    dispatch_ranges_to_clear = ['I37:I46', 'I163:I182', 'I183:I192', 'I315:I324', 'I306:I308'
        , 'I49:I54', 'I355:I363']
    for dispatch_range in dispatch_ranges_to_clear:
        num_instances = len(sheet.Range(dispatch_range).Value)
        sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]

    # Adjust resource definitions to handle interconnections as one, with updated range
    resource_info_columns = {
        'hydrogen': ('A37:E46', 'I37:I46', 4),
        'batteries': ('A163:E182', 'I163:I182', 4),
        'fuel_cell': ('A183:E192', 'I183:I192', 4),
        'electrolyzers': ('A315:E324', 'I315:I324', 3),
        'interconnections': ('A355:E363', 'I355:I363', 4),  # Updated range
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    # Calculate total maximum ratings for each resource and store dispatch ranges
    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[
            column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Determine the condition based on total generation left
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in
        scaling_factors if condition in scaling_factors[resource])
```

```
800
801     # Select resources based on grid condition
802     available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if
            condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
803     resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left
            > 0 else []
804
805     print("Allocating remaining generation to flexibility resources and updating Excel...")
806
807     # Allocate generation to each available resource
808     for resource in available_resources:
809         if resource not in total_max_ratings:
810             continue
811
812         scaling_factor = scaling_factors[resource].get(condition, 0)
813         max_rating = total_max_ratings[resource]
814         allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if
                total_scaling_factor > 0 else 0)
815
816         # Invert allocation for certain resources under surplus generation condition
817         if condition == 'surplus_generation' and resource in resources_needing_inversion:
818             allocation = -min(abs(allocation), max_rating)
819         else:
820             allocation = min(abs(allocation), max_rating)
821
822         ranges_to_update = dispatch_ranges[resource]
823
824         # Update the Excel sheet with the calculated allocations
825         for dispatch_range in ranges_to_update:
826             resource_instances = sheet.Range(dispatch_range).Value
827             num_instances = len(resource_instances)
828             allocation_per_instance = allocation / num_instances if num_instances > 0 else 0
829
830             if resource == 'interconnections':
831                 # Handle interconnections separately
832                 interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value
                        if row[3] is not None]
833                 total_rating = sum(interconnection_ratings)
834                 print(f"Interconnection ratings: {interconnection_ratings}")
835                 base_row = 355  # Starting row for interconnections in Excel
836                 for i, rating in enumerate(interconnection_ratings):
837                     # Calculate individual allocation
838                     individual_allocation = min((allocation * rating) / total_rating, rating)
                            if total_rating > 0 else 0
839                     # Apply this allocation directly to the Excel cell
840                     cell_address = f"I{base_row + i}"  # Construct the cell address
                            dynamically
841                     sheet.Range(cell_address).Value = individual_allocation
842                     print(f"Updated interconnection {i+1} dispatch with allocation: {
                            individual_allocation}")
843             else:
844                 # Update the dispatch values for other resources
845                 updated_dispatch_values = [[allocation_per_instance] for _ in range(
                        num_instances)]
846                 sheet.Range(dispatch_range).Value = updated_dispatch_values
847                 print(f"Updated {resource} dispatches in range {dispatch_range} with
                        allocation: {allocation_per_instance}")
848
849     print("Allocation process completed. Updated Excel with new dispatch values.")
850
851 # Assuming sheet_II30502 is a valid reference to your Excel sheet
852 manage_grid_flexibility_CaseB(sheet_II30502)
853
854
855 # %% Case C Flexibility resources
856 def manage_grid_flexibility_CaseC(sheet):
857     # Fetch the scaling factors for surplus generation and surplus load directly from the
            cells
858     scaling_factors = {
859         'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, '
                surplus_load': sheet.Range("AI30").Value / 100},
```

```
        'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load': sheet.Range("AJ30").Value / 100},
        'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only surplus generation is relevant
        'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load is relevant
        'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load is relevant
    }

    # Get the total generation left in the system
    total_generation_left = sheet.Range("L408").Value
    print(f"Total generation left in the system: {total_generation_left}")

    # Null out all dispatches before starting the allocation
    dispatch_ranges_to_clear = ['L37:L46', 'L163:L182', 'L183:L192', 'L315:L324', 'L306:L308', 'L49:L54', 'L355:L363']
    for dispatch_range in dispatch_ranges_to_clear:
        num_instances = len(sheet.Range(dispatch_range).Value)
        sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]

    # Adjust resource definitions to handle interconnections as one, with updated range
    resource_info_columns = {
        'hydrogen': ('A37:E46', 'L37:L46', 4),
        'batteries': ('A163:E182', 'L163:L182', 4),
        'fuel_cell': ('A183:E192', 'L183:L192', 4),
        'electrolyzers': ('A315:E324', 'L315:L324', 3),
        'interconnections': ('A355:E363', 'L355:L363', 4),  # Updated range
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    # Calculate total maximum ratings for each resource and store dispatch ranges
    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Determine the condition based on total generation left
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in scaling_factors if condition in scaling_factors[resource])

    # Select resources based on grid condition
    available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
    resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left > 0 else []

    print("Allocating remaining generation to flexibility resources and updating Excel...")

    # Allocate generation to each available resource
    for resource in available_resources:
        if resource not in total_max_ratings:
            continue

        scaling_factor = scaling_factors[resource].get(condition, 0)
        max_rating = total_max_ratings[resource]
        allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if total_scaling_factor > 0 else 0)

        # Invert allocation for certain resources under surplus generation condition
        if condition == 'surplus_generation' and resource in resources_needing_inversion:
            allocation = -min(abs(allocation), max_rating)
        else:
```

```
            allocation = min(abs(allocation), max_rating)

        ranges_to_update = dispatch_ranges[resource]

        # Update the Excel sheet with the calculated allocations
        for dispatch_range in ranges_to_update:
            resource_instances = sheet.Range(dispatch_range).Value
            num_instances = len(resource_instances)
            allocation_per_instance = allocation / num_instances if num_instances > 0 else 0

            if resource == 'interconnections':
                # Handle interconnections separately
                interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value
                    if row[3] is not None]
                total_rating = sum(interconnection_ratings)
                print(f"Interconnection ratings: {interconnection_ratings}")
                base_row = 355  # Starting row for interconnections in Excel
                for i, rating in enumerate(interconnection_ratings):
                    # Calculate individual allocation
                    individual_allocation = min((allocation * rating) / total_rating, rating)
                         if total_rating > 0 else 0
                    # Apply this allocation directly to the Excel cell
                    cell_address = f"L{base_row + i}"  # Construct the cell address
                        dynamically
                    sheet.Range(cell_address).Value = individual_allocation
                    print(f"Updated interconnection {i+1} dispatch with allocation: {
                        individual_allocation}")
            else:
                # Update the dispatch values for other resources
                updated_dispatch_values = [[allocation_per_instance] for _ in range(
                    num_instances)]
                sheet.Range(dispatch_range).Value = updated_dispatch_values
                print(f"Updated {resource} dispatches in range {dispatch_range} with
                    allocation: {allocation_per_instance}")

    print("Allocation process completed. Updated Excel with new dispatch values.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
manage_grid_flexibility_CaseC(sheet_II30502)


# %% Case D Flexibility resources
def manage_grid_flexibility_CaseD(sheet):
    # Fetch the scaling factors for surplus generation and surplus load directly from the
        cells
    scaling_factors = {
        'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, '
            surplus_load': sheet.Range("AI30").Value / 100},
        'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load':
            sheet.Range("AJ30").Value / 100},
        'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only
            surplus generation is relevant
        'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load
            is relevant
        'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load
            is relevant
    }

    # Get the total generation left in the system
    total_generation_left = sheet.Range("O408").Value  # Changed to column O
    print(f"Total generation left in the system: {total_generation_left}")

    # Null out all dispatches before starting the allocation
    dispatch_ranges_to_clear = ['O37:O46', 'O163:O182', 'O183:O192', 'O315:O324', 'O306:O308'
        , 'O49:O54', 'O355:O363']  # Changed to column O
    for dispatch_range in dispatch_ranges_to_clear:
        num_instances = len(sheet.Range(dispatch_range).Value)
        sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]

    # Adjust resource definitions to handle interconnections as one, with updated range
    resource_info_columns = {
```

```
        'hydrogen': ('A37:E46', 'O37:O46', 4),  # Changed to column O
        'batteries': ('A163:E182', 'O163:O182', 4),  # Changed to column O
        'fuel_cell': ('A183:E192', 'O183:O192', 4),  # Changed to column O
        'electrolyzers': ('A315:E324', 'O315:O324', 3),  # Changed to column O
        'interconnections': ('A355:E363', 'O355:O363', 4),  # Changed to column O
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    # Calculate total maximum ratings for each resource and store dispatch ranges
    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[
            column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Determine the condition based on total generation left
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in
        scaling_factors if condition in scaling_factors[resource])

    # Select resources based on grid condition
    available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if
        condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
    resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left
        > 0 else []

    print("Allocating remaining generation to flexibility resources and updating Excel...")

    # Allocate generation to each available resource
    for resource in available_resources:
        if resource not in total_max_ratings:
            continue

        scaling_factor = scaling_factors[resource].get(condition, 0)
        max_rating = total_max_ratings[resource]
        allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if
            total_scaling_factor > 0 else 0)

        # Invert allocation for certain resources under surplus generation condition
        if condition == 'surplus_generation' and resource in resources_needing_inversion:
            allocation = -min(abs(allocation), max_rating)
        else:
            allocation = min(abs(allocation), max_rating)

        ranges_to_update = dispatch_ranges[resource]

        # Update the Excel sheet with the calculated allocations
        for dispatch_range in ranges_to_update:
            resource_instances = sheet.Range(dispatch_range).Value
            num_instances = len(resource_instances)
            allocation_per_instance = allocation / num_instances if num_instances > 0 else 0

            if resource == 'interconnections':
                # Handle interconnections separately
                interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value
                    if row[3] is not None]
                total_rating = sum(interconnection_ratings)
                print(f"Interconnection ratings: {interconnection_ratings}")
                base_row = 355  # Starting row for interconnections in Excel
                for i, rating in enumerate(interconnection_ratings):
                    # Calculate individual allocation
                    individual_allocation = min((allocation * rating) / total_rating, rating)
                        if total_rating > 0 else 0
                    # Apply this allocation directly to the Excel cell
                    cell_address = f"O{base_row + i}"  # Changed the column to "O"
```

```
                    sheet.Range(cell_address).Value = individual_allocation
                    print(f"Updated interconnection {i+1} dispatch with allocation: {
                        individual_allocation}")
            else:
                # Update the dispatch values for other resources
                updated_dispatch_values = [[allocation_per_instance] for _ in range(
                    num_instances)]
                sheet.Range(dispatch_range).Value = updated_dispatch_values
                print(f"Updated {resource} dispatches in range {dispatch_range} with 
                    allocation: {allocation_per_instance}")

    print("Allocation process completed. Updated Excel with new dispatch values.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
manage_grid_flexibility_CaseD(sheet_II30502)  # Call the function with the appropriate sheet
    name

# %% Case E Flexibility resources
def manage_grid_flexibility_CaseE(sheet):
    # Fetch the scaling factors for surplus generation and surplus load directly from the
        cells
    scaling_factors = {
        'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, '
            surplus_load': sheet.Range("AI30").Value / 100},
        'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load':
            sheet.Range("AJ30").Value / 100},
        'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only
            surplus generation is relevant
        'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load
            is relevant
        'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load
            is relevant
    }

    # Get the total generation left in the system
    total_generation_left = sheet.Range("R408").Value  # Changed to column R
    print(f"Total generation left in the system: {total_generation_left}")

    # Null out all dispatches before starting the allocation
    dispatch_ranges_to_clear = ['R37:R46', 'R163:R182', 'R183:R192', 'R315:R324', 'R306:R308'
        , 'R49:R54', 'R355:R363']  # Changed to column R
    for dispatch_range in dispatch_ranges_to_clear:
        num_instances = len(sheet.Range(dispatch_range).Value)
        sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]

    # Adjust resource definitions to handle interconnections as one, with updated range
    resource_info_columns = {
        'hydrogen': ('A37:E46', 'R37:R46', 4),  # Changed to column R
        'batteries': ('A163:E182', 'R163:R182', 4),  # Changed to column R
        'fuel_cell': ('A183:E192', 'R183:R192', 4),  # Changed to column R
        'electrolyzers': ('A315:E324', 'R315:R324', 3),  # Changed to column R
        'interconnections': ('A355:E363', 'R355:R363', 4),  # Changed to column R
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    # Calculate total maximum ratings for each resource and store dispatch ranges
    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[
            column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Determine the condition based on total generation left
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
```

```
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in scaling_factors if condition in scaling_factors[resource])

    # Select resources based on grid condition
    available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
    resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left > 0 else []

    print("Allocating remaining generation to flexibility resources and updating Excel...")

    # Allocate generation to each available resource
    for resource in available_resources:
        if resource not in total_max_ratings:
            continue

        scaling_factor = scaling_factors[resource].get(condition, 0)
        max_rating = total_max_ratings[resource]
        allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if total_scaling_factor > 0 else 0)

        # Invert allocation for certain resources under surplus generation condition
        if condition == 'surplus_generation' and resource in resources_needing_inversion:
            allocation = -min(abs(allocation), max_rating)
        else:
            allocation = min(abs(allocation), max_rating)

        ranges_to_update = dispatch_ranges[resource]

        # Update the Excel sheet with the calculated allocations
        for dispatch_range in ranges_to_update:
            resource_instances = sheet.Range(dispatch_range).Value
            num_instances = len(resource_instances)
            allocation_per_instance = allocation / num_instances if num_instances > 0 else 0

            if resource == 'interconnections':
                # Handle interconnections separately
                interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value if row[3] is not None]
                total_rating = sum(interconnection_ratings)
                print(f"Interconnection ratings: {interconnection_ratings}")
                base_row = 355  # Starting row for interconnections in Excel
                for i, rating in enumerate(interconnection_ratings):
                    # Calculate individual allocation
                    individual_allocation = min((allocation * rating) / total_rating, rating) if total_rating > 0 else 0
                    # Apply this allocation directly to the Excel cell
                    cell_address = f"R{base_row + i}"  # Changed the column to "R"
                    sheet.Range(cell_address).Value = individual_allocation
                    print(f"Updated interconnection {i+1} dispatch with allocation: {individual_allocation}")
            else:
                # Update the dispatch values for other resources
                updated_dispatch_values = [[allocation_per_instance] for _ in range(num_instances)]
                sheet.Range(dispatch_range).Value = updated_dispatch_values
                print(f"Updated {resource} dispatches in range {dispatch_range} with allocation: {allocation_per_instance}")

    print("Allocation process completed. Updated Excel with new dispatch values.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
manage_grid_flexibility_CaseE(sheet_II30502)  # Call the function with the appropriate sheet name



# %% Case E Flexibility resources
def manage_grid_flexibility_CaseE(sheet):
    # Fetch the scaling factors for surplus generation and surplus load directly from the cells
```

```
    scaling_factors = {
        'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, '
            surplus_load': sheet.Range("AI30").Value / 100},
        'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load':
            sheet.Range("AJ30").Value / 100},
        'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only
            surplus generation is relevant
        'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load
            is relevant
        'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load
            is relevant
    }

    total_generation_left = sheet.Range("R408").Value  # Changed to column R
    print(f"Total generation left in the system: {total_generation_left}")

    # Null out all dispatches before starting the allocation
    dispatch_ranges_to_clear = ['R37:R46', 'R163:R182', 'R183:R192', 'R315:R324', 'R306:R308'
        , 'R49:R54', 'R355:R363']  # Changed to column R
    for dispatch_range in dispatch_ranges_to_clear:
        num_instances = len(sheet.Range(dispatch_range).Value)
        sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]

    # Adjust resource definitions to handle interconnections as one, with updated range
    resource_info_columns = {
        'hydrogen': ('A37:E46', 'R37:R46', 4),  # Changed to column R
        'batteries': ('A163:E182', 'R163:R182', 4),  # Changed to column R
        'fuel_cell': ('A183:E192', 'R183:R192', 4),  # Changed to column R
        'electrolyzers': ('A315:E324', 'R315:R324', 3),  # Changed to column R
        'interconnections': ('A355:E363', 'R355:R363', 4),  # Changed to column R
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[
            column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Compute total scaling factor for the available resources
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in
        scaling_factors if condition in scaling_factors[resource])

    # Select resources based on grid condition
    available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if
        condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
    resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left
        > 0 else []

    print("Allocating remaining generation to flexibility resources and updating Excel...")
    for resource in available_resources:
        if resource not in total_max_ratings:
            continue

        scaling_factor = scaling_factors[resource].get(condition, 0)
        max_rating = total_max_ratings[resource]

        allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if
            total_scaling_factor > 0 else 0)

        if condition == 'surplus_generation' and resource in resources_needing_inversion:
            allocation = -min(abs(allocation), max_rating)
        else:
            allocation = min(abs(allocation), max_rating)
```

```
1221
1222        ranges_to_update = dispatch_ranges[resource]
1223
1224        for dispatch_range in ranges_to_update:
1225            resource_instances = sheet.Range(dispatch_range).Value
1226            num_instances = len(resource_instances)
1227            allocation_per_instance = allocation / num_instances if num_instances > 0 else 0
1228
1229            if resource == 'interconnections':
1230                interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value
                        if row[3] is not None]
1231                total_rating = sum(interconnection_ratings)
1232                print(f"Interconnection ratings: {interconnection_ratings}")
1233                base_row = 355  # Starting row for interconnections in Excel
1234                for i, rating in enumerate(interconnection_ratings):
1235                    # Calculate individual allocation
1236                    individual_allocation = min((allocation * rating) / total_rating, rating)
                            if total_rating > 0 else 0
1237
1238                    # Apply this allocation directly to the Excel cell
1239                    cell_address = f"R{base_row + i}"  # Changed the column to "R"
1240                    sheet.Range(cell_address).Value = individual_allocation
1241                    print(f"Updated interconnection {i+1} dispatch with allocation: {
                            individual_allocation}")
1242
1243            else:
1244                updated_dispatch_values = [[allocation_per_instance] for _ in range(
                        num_instances)]
1245                sheet.Range(dispatch_range).Value = updated_dispatch_values
1246                print(f"Updated {resource} dispatches in range {dispatch_range} with
                        allocation: {allocation_per_instance}")
1247
1248    print("Allocation process completed. Updated Excel with new dispatch values.")
1249
1250 manage_grid_flexibility_CaseE(sheet_II30502)  # Call the function with the appropriate sheet
        name
1251
1252 # %% Case F Flexibility resources
1253 def manage_grid_flexibility_CaseF(sheet):
1254     # Fetch the scaling factors for surplus generation and surplus load directly from the
            cells
1255     scaling_factors = {
1256         'interconnections': {'surplus_generation': sheet.Range("AI29").Value / 100, '
                surplus_load': sheet.Range("AI30").Value / 100},
1257         'batteries': {'surplus_generation': sheet.Range("AJ29").Value / 100, 'surplus_load':
                sheet.Range("AJ30").Value / 100},
1258         'electrolyzers': {'surplus_generation': sheet.Range("AK29").Value / 100},  # Only
                surplus generation is relevant
1259         'hydrogen': {'surplus_load': sheet.Range("AL30").Value / 100},  # Only surplus load
                is relevant
1260         'fuel_cell': {'surplus_load': sheet.Range("AM30").Value / 100},  # Only surplus load
                is relevant
1261     }
1262
1263     # Get the total generation left in the system
1264     total_generation_left = sheet.Range("U408").Value  # Changed to column U
1265     print(f"Total generation left in the system: {total_generation_left}")
1266
1267     # Null out all dispatches before starting the allocation
1268     dispatch_ranges_to_clear = ['U37:U46', 'U163:U182', 'U183:U192', 'U315:U324', 'U306:U308'
            , 'U49:U54', 'U355:U363']  # Changed to column U
1269     for dispatch_range in dispatch_ranges_to_clear:
1270         num_instances = len(sheet.Range(dispatch_range).Value)
1271         sheet.Range(dispatch_range).Value = [[0] for _ in range(num_instances)]
1272
1273     # Adjust resource definitions to handle interconnections as one, with updated range
1274     resource_info_columns = {
1275         'hydrogen': ('A37:E46', 'U37:U46', 4),  # Changed to column U
1276         'batteries': ('A163:E182', 'U163:U182', 4),  # Changed to column U
1277         'fuel_cell': ('A183:E192', 'U183:U192', 4),  # Changed to column U
1278         'electrolyzers': ('A315:E324', 'U315:U324', 3),  # Changed to column U
```

```
        'interconnections': ('A355:E363', 'U355:U363', 4),  # Changed to column U
    }

    # Initialize total maximum ratings and dispatch ranges
    total_max_ratings = {}
    dispatch_ranges = {}

    # Calculate total maximum ratings for each resource and store dispatch ranges
    for resource, details in resource_info_columns.items():
        info_range, dispatch_range, column_index = details
        resource_info = sheet.Range(info_range).Value
        column_index_adjusted = column_index - 1
        total_max = sum(row[column_index_adjusted] for row in resource_info if row[
            column_index_adjusted] is not None)
        total_max_ratings[resource] = total_max
        dispatch_ranges[resource] = [dispatch_range]

    # Determine the condition based on total generation left
    condition = 'surplus_generation' if total_generation_left > 0 else 'surplus_load'
    total_scaling_factor = sum(scaling_factors[resource][condition] for resource in
        scaling_factors if condition in scaling_factors[resource])

    # Select resources based on grid condition
    available_resources = ['interconnections', 'batteries', 'fuel_cell', 'hydrogen'] if
        condition == 'surplus_load' else ['electrolyzers', 'interconnections', 'batteries']
    resources_needing_inversion = ['batteries', 'interconnections'] if total_generation_left
        > 0 else []

    print("Allocating remaining generation to flexibility resources and updating Excel...")

    # Allocate generation to each available resource
    for resource in available_resources:
        if resource not in total_max_ratings:
            continue

        scaling_factor = scaling_factors[resource].get(condition, 0)
        max_rating = total_max_ratings[resource]

        allocation = (total_generation_left * (scaling_factor / total_scaling_factor) if
            total_scaling_factor > 0 else 0)

        if condition == 'surplus_generation' and resource in resources_needing_inversion:
            allocation = -min(abs(allocation), max_rating)
        else:
            allocation = min(abs(allocation), max_rating)

        ranges_to_update = dispatch_ranges[resource]

        for dispatch_range in ranges_to_update:
            resource_instances = sheet.Range(dispatch_range).Value
            num_instances = len(resource_instances)
            allocation_per_instance = allocation / num_instances if num_instances > 0 else 0

            if resource == 'interconnections':
                interconnection_ratings = [row[3] for row in sheet.Range("A355:E363").Value
                    if row[3] is not None]
                total_rating = sum(interconnection_ratings)
                print(f"Interconnection ratings: {interconnection_ratings}")
                base_row = 355  # Starting row for interconnections in Excel
                for i, rating in enumerate(interconnection_ratings):
                    # Calculate individual allocation
                    individual_allocation = min((allocation * rating) / total_rating, rating)
                         if total_rating > 0 else 0

                    # Apply this allocation directly to the Excel cell
                    cell_address = f"U{base_row + i}"  # Changed the column to "U"
                    sheet.Range(cell_address).Value = individual_allocation
                    print(f"Updated interconnection {i+1} dispatch with allocation: {
                        individual_allocation}")

            else:
```

```
                updated_dispatch_values = [[allocation_per_instance] for _ in range(
                    num_instances)]
                sheet.Range(dispatch_range).Value = updated_dispatch_values
                print(f"Updated {resource} dispatches in range {dispatch_range} with
                    allocation: {allocation_per_instance}")

    print("Allocation process completed. Updated Excel with new dispatch values.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
manage_grid_flexibility_CaseF(sheet_II30502)  # Call the function with the appropriate sheet
    name


# %% Finding apparent and reactive power


def calculate_multiple_scenarios(sheet):
    # Constants for the first scenario
    initial_active_power_column = 6  # Column F for active power dispatch
    initial_apparent_power_column = 7  # Column G for apparent power (S)
    initial_reactive_power_column = 8  # Column H for reactive power (Q)
    power_factor_column = 5  # Column E for power factor, remains constant

    # Loop through 6 scenarios
    for scenario in range(6):
        # Calculate column indices for current scenario
        active_power_column = initial_active_power_column + 3 * scenario
        apparent_power_column = initial_apparent_power_column + 3 * scenario
        reactive_power_column = initial_reactive_power_column + 3 * scenario

        # Retrieve active power dispatch from the current scenario's column
        active_power_dispatch_range = sheet.Range(sheet.Cells(10, active_power_column), sheet
            .Cells(363, active_power_column))
        active_power_dispatch_values = active_power_dispatch_range.Value

        # Power factor range remains constant
        power_factor_range = sheet.Range(sheet.Cells(10, power_factor_column), sheet.Cells
            (363, power_factor_column))
        power_factor_values = power_factor_range.Value

        # Calculate and write S and Q for each row in the current scenario
        for row_index, (active_power_dispatch, power_factor) in enumerate(zip(
            active_power_dispatch_values, power_factor_values), start=10):
            if isinstance(active_power_dispatch[0], (int, float)) and isinstance(power_factor
                [0], (int, float)):
                # Extract values from Excel as floats
                active_power_dispatch = float(active_power_dispatch[0])
                power_factor = float(power_factor[0])

                # Calculate apparent power (S) using the formula S = P / pf
                apparent_power = active_power_dispatch / power_factor

                # Calculate reactive power (Q) using the formula Q = (S^2 - P^2)^0.5
                if apparent_power ** 2 - active_power_dispatch ** 2 >= 0:
                    reactive_power = (apparent_power ** 2 - active_power_dispatch ** 2) **
                        0.5
                else:
                    reactive_power = 0

                # Write calculated values to the scenario's S and Q columns in Excel
                sheet.Cells(row_index, apparent_power_column).Value = apparent_power
                sheet.Cells(row_index, reactive_power_column).Value = reactive_power

        print(f"Scenario {scenario + 1}: Calculations completed.")

# Assuming sheet_II30502 is a valid reference to your Excel sheet
calculate_multiple_scenarios(sheet_II30502)
```

C.5. Integration and Simulation Code

Listing C.2: Simulation 2050 code

```
# # -*- coding: utf-8 -*-
# """
# Created on Mon Mar 18 15:24:08 2024

# @author: Sander
# """

# # -*- coding: utf-8 -*-
# """

# Script for automation of different scenarios in DigSilent PowerFactory

# Before running, install the win32 package (pip install win32)

# """
#%% File paths for the relevant Excel files (Needs to be adjusted for other paths)

# change the following commands for the location of the excel files on your computer: input_excel, installedcap_excel and output_excel.
input_excel = r'G:\Andre␣datamaskiner\Min␣berbare␣datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Scenario-Input-2-with-dispatched-case-b.xlsx'                        # This is the excel file displaying multiple scenarios and their load flow configurations

# input_excel = r'G:\Andre datamaskiner\Min berbare datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Scenario-Input-2.xlsx'                        # This is the excel file displaying multiple scenarios and their load flow configurations
installedcap_excel = r'G:\Andre␣datamaskiner\Min␣berbare␣datamaskin\Master\Thesis\Documentation\Phyton\Simulation\Installed-Capacities.xlsx'            # This is the excel file with an overview of the capacities of the different components
# installedcap_excel = r'G:\Andre datamaskiner\Min berbare datamaskin\Master\Thesis\DynamicStability_thesis\Scenario Installed Cap_DynamicStability.xlsx'
output_excel = r'G:\Andre␣datamaskiner\Min␣berbare␣datamaskin\Master\Thesis\DynamicStability_thesis\output_excel.xlsx'


#%% Initialization of the PowerFactory scripting application and the relevant tools to effectively use Excel applications

# #Getting the Powerfactory Application
import sys
sys.path.append('C:\\Program␣Files\\DIgSILENT\\PowerFactory␣2023␣SP6\\Python\\3.11')

import powerfactory as pf                                           # Imports the PowerFactory module
app=pf.GetApplication()                                             # Calls the application
app.ActivateProject('DynamicStability_EHVnetwork_test(2)')          # Activates the correct project, if not already activated
app.ResetCalculation()                                              # Resets all calculations
app.ClearOutputWindow()                                             # Clears the Output Window in PowerFactory

import win32com.client as win32                                     # Imports Win32
excel=win32.gencache.EnsureDispatch('Excel.Application')            # Checks the dispatching of the excel application
excel.Visible=True                                                  # Makes the excel file visible (can be turned off, however might be needed to click "don't update" when the excel files are opened for the first time, otherwise simulations will not start)
import math                                                         # Calculation tool


#open the needed excel files
wbin = excel.Workbooks.Open(input_excel)                            # Function to open the input excel file
wbout = excel.Workbooks.Open(output_excel)                          # Function to open the result excel file
wbcap = excel.Workbooks.Open(installedcap_excel)                    # Function to open the installed capacity excel file
```

```
49 excel.Visible = True                                          # Activates application
       visibility
50
51 #studycase files
52 studycasefolder = app.GetProjectFolder('study')                # Locates the study folders
       in the PowerFactory module
53 studycases = studycasefolder.GetContents()                    # Locates the study cases in
        the PowerFactory module
54
55 #other input values
56 Genstatmax = 1                                                # Sets level for maximum
       power dispatch [p.u]
57
58
59
60 #%% Definition of function that allows to set the capacities of the system in PowerFactory
       based on the relevant excel file
61
62 def capacities2050():
63     wbcap1=wbcap.Worksheets("1) SymGen list")                  # Activates the sheet with
           the Synchronous generators
64     wbcap2 = wbcap.Worksheets("1) StatGen list")               # Activates the sheet with
           the static generators
65     symgenlist = wbcap1.Range("B2:I40").Value                  # Finds the range of
           information regarding Synchronuos generators in the sheet
66     trafolist = wbcap1.Range("K2:P12").Value                   # Finds the same range for
           transformers
67     statgenlistfollowing = wbcap2.Range("B2:G59").Value        # Finds the range for static
           generators marked with a "following" control system
68     statgenPVlistfollowing = wbcap2.Range("B60:G104").Value    # Finds the range for PV
           systems marked with a "following" control system
69     statgenforming = wbcap2.Range("B156:G258").Value           # Finds the range for static
           generators marked with a "forming" control system
70     statgenflexibility = wbcap2.Range("B105:G134").Value       # Finds the range for static
           generators labeled as a flexibility resource (batteries etc)
71     interconnectionDC = wbcap2.Range("B264:G266").Value        # Finds the range for HVDC
           interconnections
72     interconnectionAC = wbcap1.Range("B43:I48").Value          # Finds the range for AC
           interconnections
73
74
75
76     #Synchronous generation
77     for row in symgenlist:                                     # For loop for the rows in
           the list containing synchronous generator information
78         name = row[0]+ '.Elmsym'                               # Defines the name of the
               component, with the type in PowerFactory added (For the names in excel)
79         variable = app.GetCalcRelevantObjects(name)[0]         # Locates the object in
               PowerFactory with the same name
80         variable.ngnum = int(row[7])                           # Determines the numbers of
               paralell generators in PowerFactory
81         if float(row[3])>0:                                    # For loop that activates if
               the active power is higher than 0
82             variable.typ_id.sgn = float(row[3])                # Determines the nominal
                   apparent power of the generator
83             variable.Pmax_ucPU = 1                             # Active Power limit in p.u.
84             variable.Pmin_ucPU = 0                             # Reactive Power limit in p.u
                   .
85         if float(row[3])==0:
86             variable.outserv = 1                               # Sets the generator out of
                   service if the active power = 0
87
88                                                                # This is further done for
                                                                       all the other components
89
90     # Transformers
91     for row in trafolist:
92         name = row[0]+'.ElmTr2'
93         variable = app.GetCalcRelevantObjects(name)[0]
94         variable.ntnum = math.ceil(row[3])
95
```

```
96      # Static Generators (Following)
97      for row in statgenlistfollowing:
98          name = row[0]+'.ElmGenstat'
99          variable = app.GetCalcRelevantObjects(name)[0]
100         if float(row[3])>0:
101             variable.sgn = float(row[3])
102             variable.Pmax_ucPU = 1
103             variable.Pmin_ucPU = 0
104
105     # PV System (Following)
106     for row in statgenPVlistfollowing:
107         name = row[0]+'.ElmPvsys'
108         variable = app.GetCalcRelevantObjects(name)[0]
109         if float(row[3])>0:
110             variable.sgn = float(row[3])
111             variable.Pmax_ucPU = 1
112             variable.Pmin_ucPU = 0
113
114     # Static Generators (Forming)
115     for row in statgenforming:
116         name = row[0]+'.ElmGenstat'
117         variable = app.GetCalcRelevantObjects(name)[0]
118         if float(row[3])>0:
119             variable.sgn = float(row[3])
120             variable.Pmax_ucPU = 1
121             variable.Pmin_ucPU = 0
122
123     # Static Generators (Flexibility resources)
124     for row in statgenflexibility:
125         name = row[0]+'.ElmGenstat'
126         variable = app.GetCalcRelevantObjects(name)[0]
127         if float(row[3])>0:
128             variable.sgn = float(row[3])
129             variable.Pmax_ucPU = 1
130             variable.Pmin_ucPU = -1
131
132
133     # HVDC Interconnections
134     for row in interconnectionDC:
135         name = row[0]+'.ElmGenstat'
136         variable = app.GetCalcRelevantObjects(name)[0]
137         if float(row[3])>0:
138             variable.sgn = float(row[3])
139             variable.Pmax_ucPU = 1
140             variable.Pmin_ucPU = -1
141
142     # AC Interconnections
143     for row in interconnectionAC:
144         name = row[0]+ '.Elmsym'
145         variable = app.GetCalcRelevantObjects(name)[0]
146         variable.ngnum = int(row[7])
147         if float(row[3])>0:
148             variable.typ_id.sgn = float(row[3])/0.9
149             variable.Pmax_ucPU = 1
150             variable.Pmin_ucPU = -1
151         if float(row[3])==0:
152             variable.outserv = 1
153
154
155 #%% define functions to set the dispatches using the excelfiles
156
157
158
159
160 def dispatches1():                                          # Set dispatches for the
        first Case, Case A
161     w1 = wbin.Worksheets("II3050")                          # Name of the worksheet
162     SymG_1 = w1.Range("A37:Z48").Value                      # Synchronous generators
163     StatG_following = w1.Range("A60:Z117").Value            # All static generators with
            a "following" control mode
164     StatG_PV_following = w1.Range("A118:Z162").Value        # All PV systems with a "
```

```
        following" control mode
    StatG_battery = w1.Range("A163:Z182").Value                 # Battery systems
    StatG_fuelcell = w1.Range("A183:Z192").Value                # Fuel cells
    StatG_forming = w1.Range("A199:Z301").Value                 # All converters with a "
        forming" control mode
    Load_1 = w1.Range("A315:Z351").Value                        # Loads (Includes Power to
        heat and load response)
    StatG_interconnection = w1.Range("A355:Z357").Value         # HVDC interconnections
    SymG_interconnection = w1.Range("A358:Z363").Value          # AC interconnections
    Reactive_compensation = w1.Range("A367:Z388").Value
    update_dynamic_model_parameters(Load_1)                     # Updates the dynamic model
        parameters for all units using electrolyser droop control.


    for row in SymG_1:                                          # Updates the list for
        synchronous generators:
        name = row[0]+ '.Elmsym'                                # Defines the name of the
            component, with the type in PowerFactory added
        variable = app.GetCalcRelevantObjects(name)[0]          # Locates the object in
            PowerFactory with the same name
        variable.pgini = float(row[5])                          # Active Power Dispatch based
             on dispatch scenario in Excel sheet
        variable.qgini = float(row[7])                          # Reactive Power Dispatch
        variable.Pmax_ucPU = 1                                  # Active Power limit in p.u.
        variable.Pmin_ucPU = 0                                  # Minimum Active Power limit
            in p.u.
        variable.outserv = 0                                    # Determines the operation of
             the components  1 (Sets the components out of service)

    for row in StatG_following:                                 # Similar strucutre is
        followed for all components
        name = row[0]+'.ElmGenstat'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[5])
        variable.qgini = float(row[7])
        variable.Pmax_ucPU = Genstatmax                         # Sets the maximum Active
            power based on ratings from the capacities function
        variable.Pmin_ucPU = 0                                  # Sets the minimum Active
            power
        variable.outserv = 0
    for row in StatG_PV_following:
        name = row[0]+'.ElmPvsys'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[5])
        variable.qgini = float(row[7])
        variable.Pmax_ucPU = Genstatmax
        variable.Pmin_ucPU = 0
        variable.outserv = 0
    for row in StatG_battery:
        name = row[0]+'.ElmGenstat'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[5])
        variable.qgini = float(row[7])
        variable.Pmax_ucPU = Genstatmax
        variable.Pmin_ucPU = -Genstatmax
        variable.outserv = 0
    for row in StatG_fuelcell:
        name = row[0]+'.ElmGenstat'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[5])
        variable.qgini = float(row[7])
        variable.Pmax_ucPU = Genstatmax
        variable.Pmin_ucPU = 0
        variable.outserv = 1
    for row in StatG_forming:
        name = row[0]+'.ElmGenstat'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[5])
        variable.qgini = float(row[7])
        variable.Pmax_ucPU = Genstatmax
```

```
        variable.Pmin_ucPU = 0
        variable.outserv = 0
    for row in SymG_interconnection:
        name = row[0]+ '.Elmsym'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[5])
        variable.qgini = float(row[7])
        variable.Pmax_ucPU = 1
        variable.Pmin_ucPU = -1
        variable.outserv = 0
    for row in StatG_interconnection:
         name = row[0]+ '.ElmGenstat'
         variable = app.GetCalcRelevantObjects(name)[0]
         variable.pgini = float(row[5])
         variable.qgini = float(row[7])
         variable.Pmax_ucPU = 1
         variable.Pmin_ucPU = -1
         variable.outserv = 0
    for row in Reactive_compensation:
         name = row[0]+ '.ElmGenstat'
         variable = app.GetCalcRelevantObjects(name)[0]
         variable.pgini = float(row[5])
         variable.qgini = float(row[7])
         variable.outserv = 0
    for row in Load_1:
        name = row[0]+'.ElmLod'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.plini = float(row[5])
        variable.qlini = float(row[7])
        variable.Pmax_ucPU = 1
        variable.Pmin_ucPU = 0


def dispatches2():                                          # Set dispatches for the
    second Case, Case B
    w1 = wbin.Worksheets("II3050")                          # Name of the worksheet
    SymG_1 = w1.Range("A37:Z48").Value                      # Synchronous generators
    StatG_following = w1.Range("A60:Z117").Value            # All static generators with
        a "following" control mode
    StatG_PV_following = w1.Range("A118:Z162").Value        # All PV systems with a "
        following" control mode
    StatG_battery = w1.Range("A163:Z182").Value             # Battery systems
    StatG_fuelcell = w1.Range("A183:Z192").Value            # Fuel cells
    StatG_forming = w1.Range("A199:Z301").Value             # All converters with a "
        forming" control mode
    Load_1 = w1.Range("A315:Z351").Value                    # Loads (Includes Power to
        heat and load response)
    StatG_interconnection = w1.Range("A355:Z357").Value     # HVDC interconnections
    SymG_interconnection = w1.Range("A358:Z363").Value      # AC interconnections
    Reactive_compensation = w1.Range("A367:Z388").Value
    update_dynamic_model_parameters(Load_1)                 # Updates the dynamic model
        parameters for all units using electrolyser droop control.

    for row in SymG_1:                                      # Updates the list for
        synchronous generators:
        name = row[0]+ '.Elmsym'                            # Defines the name of the
            component, with the type in PowerFactory added
        variable = app.GetCalcRelevantObjects(name)[0]      # Finds the object in
            PowerFactory with the same name
        variable.pgini = float(row[8])                      # Active Power Dispatch based
             on dispatch scenario in Excel sheet
        variable.qgini = float(row[10])                     # Reactive Power Dispatch
        variable.Pmax_ucPU = 1                              # Active Power limit in p.u.
        variable.Pmin_ucPU = 0                              # Minimum Active Power limit
            in p.u.
        variable.outserv = 0                                # Determines the operation of
             the components  1 (Sets the components out of service)

    for row in StatG_following:                             # Similar strucutre is
        followed for all components
        name = row[0]+'.ElmGenstat'
```

```
281         variable = app.GetCalcRelevantObjects(name)[0]
282         variable.pgini = float(row[8])
283         variable.qgini = float(row[10])
284         variable.Pmax_ucPU = Genstatmax                          # Sets the maximum Active
                power based on ratings from the capacities function
285         variable.Pmin_ucPU = 0                                   # Sets the minimum Active
                power based
286         variable.outserv = 0
287     for row in StatG_PV_following:
288         name = row[0]+'.ElmPvsys'
289         variable = app.GetCalcRelevantObjects(name)[0]
290         variable.pgini = float(row[8])
291         variable.qgini = float(row[10])
292         variable.Pmax_ucPU = Genstatmax
293         variable.Pmin_ucPU = 0
294         variable.outserv = 0
295     for row in StatG_battery:
296         name = row[0]+'.ElmGenstat'
297         variable = app.GetCalcRelevantObjects(name)[0]
298         variable.pgini = float(row[8])
299         variable.qgini = float(row[10])
300         variable.Pmax_ucPU = Genstatmax
301         variable.Pmin_ucPU = -Genstatmax
302         variable.outserv = 0
303     for row in StatG_fuelcell:
304         name = row[0]+'.ElmGenstat'
305         variable = app.GetCalcRelevantObjects(name)[0]
306         variable.pgini = float(row[8])
307         variable.qgini = float(row[10])
308         variable.Pmax_ucPU = Genstatmax
309         variable.Pmin_ucPU = 0
310         variable.outserv = 0
311     for row in StatG_forming:
312         name = row[0]+'.ElmGenstat'
313         variable = app.GetCalcRelevantObjects(name)[0]
314         variable.pgini = float(row[8])
315         variable.qgini = float(row[10])
316         variable.Pmax_ucPU = Genstatmax
317         variable.Pmin_ucPU = 0
318         variable.outserv = 0
319     for row in SymG_interconnection:
320         name = row[0]+ '.Elmsym'
321         variable = app.GetCalcRelevantObjects(name)[0]
322         variable.pgini = float(row[8])
323         variable.qgini = float(row[10])
324         variable.Pmax_ucPU = 1
325         variable.Pmin_ucPU = -1
326         variable.outserv = 0
327     for row in StatG_interconnection:
328          name = row[0]+ '.ElmGenstat'
329          variable = app.GetCalcRelevantObjects(name)[0]
330          variable.pgini = float(row[8])
331          variable.qgini = float(row[10])
332          variable.Pmax_ucPU = 1
333          variable.Pmin_ucPU = -1
334          variable.outserv = 0
335     for row in Reactive_compensation:
336          name = row[0]+ '.ElmGenstat'
337          variable = app.GetCalcRelevantObjects(name)[0]
338          variable.pgini = float(row[8])
339          variable.qgini = float(row[10])
340          variable.outserv = 0
341     for row in Load_1:
342         name = row[0]+'.ElmLod'
343         variable = app.GetCalcRelevantObjects(name)[0]
344         variable.plini = float(row[8])
345         variable.qlini = float(row[10])
346         variable.Pmax_ucPU = 1
347         variable.Pmin_ucPU = 0
348
349
```

```
350
351 def dispatches3():                                          # Set dispatches for the
        third case, Case C
352     w1 = wbin.Worksheets("II3050")                         # Name of the worksheet
353     SymG_1 = w1.Range("A37:Z48").Value                     # Synchronous generators
354     StatG_following = w1.Range("A60:Z117").Value           # All static generators with
            a "following" control mode
355     StatG_PV_following = w1.Range("A118:Z162").Value       # All PV systems with a "
            following" control mode
356     StatG_battery = w1.Range("A163:Z182").Value            # Battery systems
357     StatG_fuelcell = w1.Range("A183:Z192").Value           # Fuel cells
358     StatG_forming = w1.Range("A199:Z301").Value            # All converters with a "
            forming" control mode
359     Load_1 = w1.Range("A315:Z351").Value                   # Loads (Includes Power to
            heat and load response)
360     StatG_interconnection = w1.Range("A355:Z357").Value    # HVDC interconnections
361     SymG_interconnection = w1.Range("A358:Z363").Value
362     Reactive_compensation = w1.Range("A367:Z388").Value    # AC interconnections
363     update_dynamic_model_parameters(Load_1)                # Updates the dynamic model
            parameters for all units using electrolyser droop control.
364
365     for row in SymG_1:                                     # Updates the list for
            synchronous generators:
366         name = row[0]+ '.Elmsym'                           # Defines the name of the
                component, with the type in PowerFactory added
367         variable = app.GetCalcRelevantObjects(name)[0]     # Finds the object in
                PowerFactory with the same name
368         variable.pgini = float(row[11])                    # Active Power Dispatch based
                 on dispatch scenario in Excel sheet
369         variable.qgini = float(row[13])                    # Reactive Power Dispatch
370         variable.Pmax_ucPU = 1                             # Active Power limit in p.u.
371         variable.Pmin_ucPU = 0                             # Minimum Active Power limit
                in p.u.
372         variable.outserv = 0                               # Determines the operation of
                 the components  1 (Sets the components out of service)
373
374     for row in StatG_following:                            # Similar strucutre is
            followed for all components
375         name = row[0]+'.ElmGenstat'
376         variable = app.GetCalcRelevantObjects(name)[0]
377         variable.pgini = float(row[11])
378         variable.qgini = float(row[13])
379         variable.Pmax_ucPU = Genstatmax                    # Sets the maximum Active
                power based on ratings from the capacities function
380         variable.Pmin_ucPU = 0                             # Sets the minimum Active
                power based
381         variable.outserv = 0
382     for row in StatG_PV_following:
383         name = row[0]+'.ElmPvsys'
384         variable = app.GetCalcRelevantObjects(name)[0]
385         variable.pgini = float(row[11])
386         variable.qgini = float(row[13])
387         variable.Pmax_ucPU = Genstatmax
388         variable.Pmin_ucPU = 0
389         variable.outserv = 0
390     for row in StatG_battery:
391         name = row[0]+'.ElmGenstat'
392         variable = app.GetCalcRelevantObjects(name)[0]
393         variable.pgini = float(row[11])
394         variable.qgini = float(row[13])
395         variable.Pmax_ucPU = Genstatmax
396         variable.Pmin_ucPU = -Genstatmax
397         variable.outserv = 0
398     for row in StatG_fuelcell:
399         name = row[0]+'.ElmGenstat'
400         variable = app.GetCalcRelevantObjects(name)[0]
401         variable.pgini = float(row[11])
402         variable.qgini = float(row[13])
403         variable.Pmax_ucPU = Genstatmax
404         variable.Pmin_ucPU = 0
405         variable.outserv = 1
```

```
406     for row in StatG_forming:
407         name = row[0]+'.ElmGenstat'
408         variable = app.GetCalcRelevantObjects(name)[0]
409         variable.pgini = float(row[11])
410         variable.qgini = float(row[13])
411         variable.Pmax_ucPU = Genstatmax
412         variable.Pmin_ucPU = 0
413         variable.outserv = 0
414     for row in SymG_interconnection:
415         name = row[0]+ '.Elmsym'
416         variable = app.GetCalcRelevantObjects(name)[0]
417         variable.pgini = float(row[11])
418         variable.qgini = float(row[13])
419         variable.Pmax_ucPU = 1
420         variable.Pmin_ucPU = -1
421         variable.outserv = 0
422     for row in StatG_interconnection:
423          name = row[0]+ '.ElmGenstat'
424          variable = app.GetCalcRelevantObjects(name)[0]
425          variable.pgini = float(row[11])
426          variable.qgini = float(row[13])
427          variable.Pmax_ucPU = 1
428          variable.Pmin_ucPU = -1
429          variable.outserv = 0
430     for row in Load_1:
431         name = row[0]+'.ElmLod'
432         variable = app.GetCalcRelevantObjects(name)[0]
433         variable.plini = float(row[11])
434         variable.qlini = float(row[13])
435         variable.Pmax_ucPU = 1
436         variable.Pmin_ucPU = 0
437     for row in Reactive_compensation:
438          name = row[0]+ '.ElmGenstat'
439          variable = app.GetCalcRelevantObjects(name)[0]
440          variable.pgini = float(row[11])
441          variable.qgini = float(row[13])
442          variable.outserv = 0
443
444
445 def dispatches4():                                          # Set dispatches for the
        fourth case, Case D
446     w1 = wbin.Worksheets("II3050")                          # Name of the worksheet
447     SymG_1 = w1.Range("A37:Z48").Value                      # Synchronous generators
448     StatG_following = w1.Range("A60:Z117").Value            # All static generators with
            a "following" control mode
449     StatG_PV_following = w1.Range("A118:Z162").Value        # All PV systems with a "
            following" control mode
450     StatG_battery = w1.Range("A163:Z182").Value             # Battery systems
451     StatG_fuelcell = w1.Range("A183:Z192").Value            # Fuel cells
452     StatG_forming = w1.Range("A199:Z301").Value             # All converters with a "
            forming" control mode
453     Load_1 = w1.Range("A315:Z351").Value                    # Loads (Includes Power to
            heat and load response)
454     StatG_interconnection = w1.Range("A355:Z357").Value     # HVDC interconnections
455     SymG_interconnection = w1.Range("A358:Z363").Value      # AC interconnections
456     update_dynamic_model_parameters(Load_1)                 # Updates the dynamic model
            parameters for all units using electrolyser droop control.
457     Reactive_compensation = w1.Range("A367:Z388").Value
458
459     for row in SymG_1:                                      # Updates the list for
            synchronous generators:
460         name = row[0]+ '.Elmsym'                            # Defines the name of the
                component, with the type in PowerFactory added
461         variable = app.GetCalcRelevantObjects(name)[0]      # Finds the object in
                PowerFactory with the same name
462         variable.pgini = float(row[14])                     # Active Power Dispatch based
                 on dispatch scenario in Excel sheet
463         variable.qgini = float(row[16])                     # Reactive Power Dispatch
464         variable.Pmax_ucPU = 1                              # Active Power limit in p.u.
465         variable.Pmin_ucPU = 0                              # Minimum Active Power limit
                in p.u.
```

```
466         variable.outserv = 0                                    # Determines the operation of
                the components  1 (Sets the components out of service)
467
468     for row in StatG_following:                                 # Similar strucutre is
            followed for all components
469         name = row[0]+'.ElmGenstat'
470         variable = app.GetCalcRelevantObjects(name)[0]
471         variable.pgini = float(row[14])
472         variable.qgini = float(row[16])
473         variable.Pmax_ucPU = Genstatmax                         # Sets the maximum Active
                power based on ratings from the capacities function
474         variable.Pmin_ucPU = 0                                  # Sets the minimum Active
                power based
475         variable.outserv = 0
476     for row in StatG_PV_following:
477         name = row[0]+'.ElmPvsys'
478         variable = app.GetCalcRelevantObjects(name)[0]
479         variable.pgini = float(row[14])
480         variable.qgini = float(row[16])
481         variable.Pmax_ucPU = Genstatmax
482         variable.Pmin_ucPU = 0
483         variable.outserv = 0
484     for row in StatG_battery:
485         name = row[0]+'.ElmGenstat'
486         variable = app.GetCalcRelevantObjects(name)[0]
487         variable.pgini = float(row[14])
488         variable.qgini = float(row[16])
489         variable.Pmax_ucPU = Genstatmax
490         variable.Pmin_ucPU = -Genstatmax
491         variable.outserv = 0
492     for row in StatG_fuelcell:
493         name = row[0]+'.ElmGenstat'
494         variable = app.GetCalcRelevantObjects(name)[0]
495         variable.pgini = float(row[14])
496         variable.qgini = float(row[16])
497         variable.Pmax_ucPU = Genstatmax
498         variable.Pmin_ucPU = 0
499         variable.outserv = 1
500     for row in StatG_forming:
501         name = row[0]+'.ElmGenstat'
502         variable = app.GetCalcRelevantObjects(name)[0]
503         variable.pgini = float(row[14])
504         variable.qgini = float(row[16])
505         variable.Pmax_ucPU = Genstatmax
506         variable.Pmin_ucPU = 0
507         variable.outserv = 0
508     for row in SymG_interconnection:
509         name = row[0]+ '.Elmsym'
510         variable = app.GetCalcRelevantObjects(name)[0]
511         variable.pgini = float(row[14])
512         variable.qgini = float(row[16])
513         variable.Pmax_ucPU = 1
514         variable.Pmin_ucPU = -1
515         variable.outserv = 0
516     for row in StatG_interconnection:
517          name = row[0]+ '.ElmGenstat'
518          variable = app.GetCalcRelevantObjects(name)[0]
519          variable.pgini = float(row[14])
520          variable.qgini = float(row[16])
521          variable.Pmax_ucPU = 1
522          variable.Pmin_ucPU = -1
523          variable.outserv = 0
524     for row in Load_1:
525         name = row[0]+'.ElmLod'
526         variable = app.GetCalcRelevantObjects(name)[0]
527         variable.plini = float(row[14])
528         variable.qlini = float(row[16])
529         variable.Pmax_ucPU = 1
530         variable.Pmin_ucPU = 0
531     for row in Reactive_compensation:
532          name = row[0]+ '.ElmGenstat'
```

```
533             variable = app.GetCalcRelevantObjects(name)[0]
534             variable.pgini = float(row[14])
535             variable.qgini = float(row[16])
536             variable.outserv = 0
537 def dispatches5():                                          # Set dispatches for the
        fifth case, Case E
538     w1 = wbin.Worksheets("II3050")                          # Name of the worksheet
539     SymG_1 = w1.Range("A37:Z48").Value                      # Synchronous generators
540     StatG_following = w1.Range("A60:Z117").Value            # All static generators with
            a "following" control mode
541     StatG_PV_following = w1.Range("A118:Z162").Value        # All PV systems with a "
            following" control mode
542     StatG_battery = w1.Range("A163:Z182").Value             # Battery systems
543     StatG_fuelcell = w1.Range("A183:Z192").Value            # Fuel cells
544     StatG_forming = w1.Range("A199:Z301").Value             # All converters with a "
            forming" control mode
545     Load_1 = w1.Range("A315:Z351").Value                    # Loads (Includes Power to
            heat and load response)
546     StatG_interconnection = w1.Range("A355:Z357").Value     # HVDC interconnections
547     SymG_interconnection = w1.Range("A358:Z363").Value      # AC interconnections
548     update_dynamic_model_parameters(Load_1)                 # Updates the dynamic model
            parameters for all units using electrolyser droop control.
549     Reactive_compensation = w1.Range("A367:Z388").Value
550     for row in SymG_1:                                      # Updates the list for
            synchronous generators:
551         name = row[0]+ '.Elmsym'                            # Defines the name of the
                component, with the type in PowerFactory added
552         variable = app.GetCalcRelevantObjects(name)[0]      # Finds the object in
                PowerFactory with the same name
553         variable.pgini = float(row[17])                     # Active Power Dispatch based
                 on dispatch scenario in Excel sheet
554         variable.qgini = float(row[19])                     # Reactive Power Dispatch
555         variable.Pmax_ucPU = 1                              # Active Power limit in p.u.
556         variable.Pmin_ucPU = 0                              # Minimum Active Power limit
                in p.u.
557         variable.outserv = 0                                # Determines the operation of
                 the components  1 (Sets the components out of service)
558
559     for row in StatG_following:                             # Similar strucutre is
            followed for all components
560         name = row[0]+'.ElmGenstat'
561         variable = app.GetCalcRelevantObjects(name)[0]
562         variable.pgini = float(row[17])
563         variable.qgini = float(row[19])
564         variable.Pmax_ucPU = Genstatmax                     # Sets the maximum Active
                power based on ratings from the capacities function
565         variable.Pmin_ucPU = 0                              # Sets the minimum Active
                power based
566         variable.outserv = 0
567     for row in StatG_PV_following:
568         name = row[0]+'.ElmPvsys'
569         variable = app.GetCalcRelevantObjects(name)[0]
570         variable.pgini = float(row[17])
571         variable.qgini = float(row[19])
572         variable.Pmax_ucPU = Genstatmax
573         variable.Pmin_ucPU = 0
574         variable.outserv = 0
575     for row in StatG_battery:
576         name = row[0]+'.ElmGenstat'
577         variable = app.GetCalcRelevantObjects(name)[0]
578         variable.pgini = float(row[17])
579         variable.qgini = float(row[19])
580         variable.Pmax_ucPU = Genstatmax
581         variable.Pmin_ucPU = -Genstatmax
582         variable.outserv = 0
583     for row in StatG_fuelcell:
584         name = row[0]+'.ElmGenstat'
585         variable = app.GetCalcRelevantObjects(name)[0]
586         variable.pgini = float(row[17])
587         variable.qgini = float(row[19])
588         variable.Pmax_ucPU = Genstatmax
```

```
        variable.Pmin_ucPU = 0
        variable.outserv = 1
    for row in StatG_forming:
        name = row[0]+'.ElmGenstat'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[17])
        variable.qgini = float(row[19])
        variable.Pmax_ucPU = Genstatmax
        variable.Pmin_ucPU = 0
        variable.outserv = 0
    for row in SymG_interconnection:
        name = row[0]+ '.Elmsym'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.pgini = float(row[17])
        variable.qgini = float(row[19])
        variable.Pmax_ucPU = 1
        variable.Pmin_ucPU = -1
        variable.outserv = 0
    for row in StatG_interconnection:
         name = row[0]+ '.ElmGenstat'
         variable = app.GetCalcRelevantObjects(name)[0]
         variable.pgini = float(row[17])
         variable.qgini = float(row[19])
         variable.Pmax_ucPU = 1
         variable.Pmin_ucPU = -1
         variable.outserv = 0
    for row in Load_1:
        name = row[0]+'.ElmLod'
        variable = app.GetCalcRelevantObjects(name)[0]
        variable.plini = float(row[17])
        variable.qlini = float(row[19])
        variable.Pmax_ucPU = 1
        variable.Pmin_ucPU = 0
    for row in Reactive_compensation:
         name = row[0]+ '.ElmGenstat'
         variable = app.GetCalcRelevantObjects(name)[0]
         variable.pgini = float(row[17])
         variable.qgini = float(row[19])
         variable.outserv = 0

def dispatches6():                                          # Set dispatches for the sixt
     case, Case F
    w1 = wbin.Worksheets("II3050")                          # Name of the worksheet
    SymG_1 = w1.Range("A37:Z48").Value                      # Synchronous generators
    StatG_following = w1.Range("A60:Z117").Value            # All static generators with
        a "following" control mode
    StatG_PV_following = w1.Range("A118:Z162").Value        # All PV systems with a "
        following" control mode
    StatG_battery = w1.Range("A163:Z182").Value             # Battery systems
    StatG_fuelcell = w1.Range("A183:Z192").Value            # Fuel cells
    StatG_forming = w1.Range("A199:Z301").Value             # All converters with a "
        forming" control mode
    Load_1 = w1.Range("A315:Z351").Value                    # Loads (Includes Power to
        heat and load response)
    StatG_interconnection = w1.Range("A355:Z357").Value     # HVDC interconnections
    SymG_interconnection = w1.Range("A358:Z363").Value      # AC interconnections
    update_dynamic_model_parameters(Load_1)                 # Updates the dynamic model
        parameters for all units using electrolyser droop control.
    Reactive_compensation = w1.Range("A367:Z388").Value
    for row in SymG_1:                                      # Updates the list for
        synchronous generators:
        name = row[0]+ '.Elmsym'                            # Defines the name of the
            component, with the type in PowerFactory added
        variable = app.GetCalcRelevantObjects(name)[0]      # Finds the object in
            PowerFactory with the same name
        variable.pgini = float(row[20])                     # Active Power Dispatch based
             on dispatch scenario in Excel sheet
        variable.qgini = float(row[22])                     # Reactive Power Dispatch
        variable.Pmax_ucPU = 1                              # Active Power limit in p.u.
        variable.Pmin_ucPU = 0                              # Minimum Active Power limit
            in p.u.
```

```
649         variable.outserv = 0                                    # Determines the operation of
                the components  1 (Sets the components out of service)
650
651     for row in StatG_following:                                 # Similar strucutre is
            followed for all components
652         name = row[0]+'.ElmGenstat'
653         variable = app.GetCalcRelevantObjects(name)[0]
654         variable.pgini = float(row[20])
655         variable.qgini = float(row[22])
656         variable.Pmax_ucPU = Genstatmax                         # Sets the maximum Active
                power based on ratings from the capacities function
657         variable.Pmin_ucPU = 0                                  # Sets the minimum Active
                power based
658         variable.outserv = 0
659     for row in StatG_PV_following:
660         name = row[0]+'.ElmPvsys'
661         variable = app.GetCalcRelevantObjects(name)[0]
662         variable.pgini = float(row[20])
663         variable.qgini = float(row[22])
664         variable.Pmax_ucPU = Genstatmax
665         variable.Pmin_ucPU = 0
666         variable.outserv = 0
667     for row in StatG_battery:
668         name = row[0]+'.ElmGenstat'
669         variable = app.GetCalcRelevantObjects(name)[0]
670         variable.pgini = float(row[20])
671         variable.qgini = float(row[22])
672         variable.Pmax_ucPU = Genstatmax
673         variable.Pmin_ucPU = -Genstatmax
674         variable.outserv = 0
675     for row in StatG_fuelcell:
676         name = row[0]+'.ElmGenstat'
677         variable = app.GetCalcRelevantObjects(name)[0]
678         variable.pgini = float(row[20])
679         variable.qgini = float(row[22])
680         variable.Pmax_ucPU = Genstatmax
681         variable.Pmin_ucPU = 0
682         variable.outserv = 1
683     for row in StatG_forming:
684         name = row[0]+'.ElmGenstat'
685         variable = app.GetCalcRelevantObjects(name)[0]
686         variable.pgini = float(row[20])
687         variable.qgini = float(row[22])
688         variable.Pmax_ucPU = Genstatmax
689         variable.Pmin_ucPU = 0
690         variable.outserv = 0
691     for row in SymG_interconnection:
692         name = row[0]+ '.Elmsym'
693         variable = app.GetCalcRelevantObjects(name)[0]
694         variable.pgini = float(row[20])
695         variable.qgini = float(row[22])
696         variable.Pmax_ucPU = 1
697         variable.Pmin_ucPU = -1
698         variable.outserv = 0
699     for row in StatG_interconnection:
700          name = row[0]+ '.ElmGenstat'
701          variable = app.GetCalcRelevantObjects(name)[0]
702          variable.pgini = float(row[20])
703          variable.qgini = float(row[22])
704          variable.Pmax_ucPU = 1
705          variable.Pmin_ucPU = -1
706          variable.outserv = 0
707     for row in Load_1:
708         name = row[0]+'.ElmLod'
709         variable = app.GetCalcRelevantObjects(name)[0]
710         variable.plini = float(row[20])
711         variable.qlini = float(row[22])
712         variable.Pmax_ucPU = 1
713         variable.Pmin_ucPU = 0
714     for row in Reactive_compensation:
715          name = row[0]+ '.ElmGenstat'
```

```
            variable = app.GetCalcRelevantObjects(name)[0]
            variable.pgini = float(row[20])
            variable.qgini = float(row[22])
            variable.outserv = 0

    # Add more of these if more scenarios are added




#%% function for setting reactive power controls


    # Functions for setting different control modes in PowerFactory
    # PV: The grid is primarily dominated by PV generation, and these resources will focus on controlling the voltage
    # Wind: Similar to PV control, but now wind is dominant
    # It is possible to change the different control modes directly in the function to alter and try different control modes
    # The control mode is selected at the end of the script, as a function call


def reactivepowercontrol(Controlmode):

                                                        # Set the range of cells for the different sources
    w1 = wbin.Worksheets("II3050")                      # Name of the worksheet
    SymG_1 = w1.Range("A10:Z48").Value                  # Synchronous generators
    StatG_following = w1.Range("A60:Z117").Value        # All static generators with a "following" control mode
    StatG_PV_following = w1.Range("A118:Z162").Value    # All PV systems with a "following" control mode
    StatG_battery = w1.Range("A163:Z182").Value         # Battery systems
    StatG_fuelcell = w1.Range("A183:Z192").Value        # Fuel cells
    StatG_forming = w1.Range("A199:Z301").Value         # All converters with a "forming" control mode
    Load_1 = w1.Range("A315:Z351").Value                # Loads (Includes Power to heat and load response)
    StatG_interconnection = w1.Range("A355:Z357").Value # HVDC interconnections
    SymG_interconnection = w1.Range("A358:Z363").Value  # AC interconnections



    if Controlmode == 'PV':                             # Control mode set to: PV (System is dominantly PV generation)

        # Grid-Forming Converters
        for row in StatG_forming:                       # For all the static generators with "forming" control
            name = row[0]+'.ElmGenstat'
            variable = app.GetCalcRelevantObjects(name)[0]
            if variable.loc_name.startswith("Converter_PV"):    # Index to find the PV converters
                variable.av_mode = 'constv'                     # Sets control mode to Constant V in PowerFactory
            else:
                variable.av_mode = 'constq'                     # If the converter is not a "Converter_PV", a control mode tof Constant Q is set

        # Grid-forming Converters
        for row in StatG_forming:
            name = row[0]+'.ElmGenstat'
            variable = app.GetCalcRelevantObjects(name)[0]
            if variable.loc_name.endswith("2"):
                    variable.av_mode = 'constq'

        # Grid-forming Converters
        for row in StatG_forming:
            name = row[0]+'.ElmGenstat'
            variable = app.GetCalcRelevantObjects(name)[0]
            if variable.loc_name.endswith("3"):
```

```
775                     variable.av_mode = 'constq'
776
777
778                                                                 # The same structure is used
                                                                        for all the different
                                                                        converters
779         # Grid-following PV Systems
780         for row in StatG_PV_following:
781             name = row[0]+'.ElmPvsys'
782             variable = app.GetCalcRelevantObjects(name)[0]
783             if variable.loc_name.startswith("L_PV"):
784                 variable.av_mode = 'constq'
785             else:
786                 variable.av_mode = 'constq'
787
788         # Grid-following PV Systems
789         for row in StatG_PV_following:
790                name = row[0] + '.ElmPvsys'
791                variable = app.GetCalcRelevantObjects(name)[0]
792                if variable.loc_name.endswith("2"):
793                    variable.av_mode = 'constq'
794
795         for row in StatG_PV_following:
796                name = row[0] + '.ElmPvsys'
797                variable = app.GetCalcRelevantObjects(name)[0]
798                if variable.loc_name.endswith("3"):
799                    variable.av_mode = 'constq'
800
801         # Grid-following Static generators
802         for row in StatG_following:
803             name = row[0]+'.ElmGenstat'
804             variable = app.GetCalcRelevantObjects(name)[0]
805             if variable.loc_name == "W_WOZ_HKW" or variable.loc_name ==  "W_WOZ_IJMA" or
                    variable.loc_name == "W_WOZ_IJMB" or variable.loc_name == "W_WOZ_TNW" or
                    variable.loc_name ==  "W_WOZ_EOS380_2":
806                 variable.av_mode = 'constv'                         # if no wind, some WOZ only
                        act as STATCOMS
807
808             else:
809                 variable.av_mode = 'constq'
810
811         # Grid-following Static generators
812         for row in StatG_following:
813             name = row[0]+'.ElmGenstat'
814             variable = app.GetCalcRelevantObjects(name)[0]
815             if variable.loc_name.endswith("2"):
816                    variable.av_mode = 'constq'
817
818        # Grid-following Static generators
819         for row in StatG_following:
820             name = row[0]+'.ElmGenstat'
821             variable = app.GetCalcRelevantObjects(name)[0]
822             if variable.loc_name.endswith("3"):
823                    variable.av_mode = 'constq'
824
825         # Synchronous Generators
826         for row in SymG_1:
827             name = row[0]+'.ElmSym'
828             variable = app.GetCalcRelevantObjects(name)[0]
829             if variable.loc_name == "Gen_GAS_MBT15":
830                 variable.av_mode = 'constv'
831
832             else:
833                 variable.av_mode = 'constv'
834
835
836         # Flexibility resources
837         for row in StatG_battery:
838                 name = row[0]+'.ElmGenstat'
839                 variable = app.GetCalcRelevantObjects(name)[0]
840                 if variable.loc_name.startswith("BESS_"):
```

```
                    variable.av_mode = 'constq'
        for row in StatG_fuelcell:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.startswith("FuelCell_"):
                    variable.av_mode = 'constq'


        # HVDC Interconnections
        for row in StatG_interconnection:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.startswith("IC_"):
                    variable.av_mode = 'constq'
        # AC Interconnections
        for row in SymG_interconnection:
                name = row[0]+'.ElmSym'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.startswith("IC_"):
                    variable.av_mode = 'constv'

    if Controlmode == 'Wind':                                   # Control mode set to: Wind (
        System is dominantly Wind generation)

            # Grid-forming Static generators
            for row in StatG_forming:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.startswith("Converter_W"): # For all wind converters
                        variable.av_mode = 'constv'
                else:
                        variable.av_mode = 'constq'
            # Grid-forming Static generators
            for row in StatG_forming:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.endswith("2"):
                        variable.av_mode = 'constq'
            # Grid-forming Static generators
            for row in StatG_forming:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.endswith("3"):
                        variable.av_mode = 'constq'
            # Grid-forming Static generators
            for row in StatG_forming:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.startswith("Converter_PV"):
                        variable.av_mode = 'constq'


            # Grid-following PV systems
            for row in StatG_PV_following:
                name = row[0] + '.ElmPvsys'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.startswith("L_PV"):
                    variable.av_mode = 'constq'
            # Grid-following PV systems
            for row in StatG_PV_following:
                name = row[0] + '.ElmPvsys'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.endswith("2"):
                    variable.av_mode = 'constq'
            # Grid-following PV systems
            for row in StatG_PV_following:
                name = row[0] + '.ElmPvsys'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.endswith("3"):
                    variable.av_mode = 'constq'

```

```
            # Grid-following Static generator
            for row in StatG_following:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.startswith("W_"):
                        variable.av_mode = 'constq'             # Grid forming is primarily
                            set to control V.
                else:
                        variable.av_mode = 'constq'
            # Grid-following Static generator
            for row in StatG_following:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.endswith("2"):
                        variable.av_mode = 'constq'
            # Grid-following Static generator
            for row in StatG_following:
                name = row[0]+'.ElmGenstat'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name.endswith("3"):
                        variable.av_mode = 'constq'

            # Synchronous generators
            for row in SymG_1:
                name = row[0]+'.ElmSym'
                variable = app.GetCalcRelevantObjects(name)[0]
                if variable.loc_name == "Gen_GAS_MBT15":
                        variable.av_mode = 'constv'
                else:
                        variable.av_mode = 'constv'

            # Static generators (Flexibility resources)
            for row in StatG_battery:
                 name = row[0]+'.ElmGenstat'
                 variable = app.GetCalcRelevantObjects(name)[0]
                 if variable.loc_name.startswith("W_"):
                          variable.av_mode = 'constq'
                 else:
                          variable.av_mode = 'constq'

            for row in StatG_fuelcell:
                 name = row[0]+'.ElmGenstat'
                 variable = app.GetCalcRelevantObjects(name)[0]
                 if variable.loc_name.startswith("Fuelcell_"):
                          variable.av_mode = 'constq'
                 else:
                          variable.av_mode = 'constq'

            # HVDC Interconnections
            for row in StatG_interconnection:
                 name = row[0]+'.ElmGenstat'
                 variable = app.GetCalcRelevantObjects(name)[0]
                 if variable.loc_name.startswith("IC_"):
                          variable.av_mode = 'constq'
                          variable.ictpg = 0
                 else:
                          variable.av_mode = 'constq'

            # AC Interconnections
            for row in SymG_interconnection:
                 name = row[0]+'.ElmSym'
                 variable = app.GetCalcRelevantObjects(name)[0]
                 if variable.loc_name.startswith("IC_"):
                          variable.av_mode = 'constv'
                 else:
                          variable.av_mode = 'constq'


# %% Control selection of Grid-forming control type

# This function controls the selection of the grid-forming control type for composite models
```

```
    in the power system simulation.
# Available control modes include VSM, Droop, and Synchroverter. Alternatively, "none" can be selected to put all controllers in service.
# The desired control selection is defined at the end of the script, as a function call.

def grid_forming_control(control_mode):
    """
    Control the selection of the grid-forming control type for composite models.

    Parameters:
        control_mode (str): The desired control mode ('VSM', 'Droop', 'Synchroverter', or 'none').
    """

    # Get all composite models (.ElmComp*) in the project
    composite_models = app.GetCalcRelevantObjects("*.ElmComp*")

    # Define the available control modes
    control_modes = ['VSM', 'Droop', 'Synchroverter']

    # Check if the provided control_mode is in the list of control modes
    if control_mode not in control_modes:
        app.PrintInfo(f"Invalid control mode: {control_mode}. Please choose from 'VSM', 'Droop', 'Synchroverter'.")
        return

    # Iterate over each composite model
    for model in composite_models:
        model_name = model.loc_name

        # Check if the model name starts with the control mode prefix to be enabled
        if model_name.startswith(control_mode):
            # Set the chosen control mode in service
            model.outserv = 0
            app.PrintInfo(f"Control enabled for model: {model_name}")
        # Check if the model name starts with another control mode's prefix and disable it
        elif any(model_name.startswith(other_mode) for other_mode in control_modes if other_mode != control_mode):
            # Set the other control modes out of service
            model.outserv = 1
            app.PrintInfo(f"Control disabled for model: {model_name}")




# %% Converter selection for operation


def set_converter_controls_out_of_service():
    """
    Here, all the different converters are set out of service, based on if they have a dispatch or not.
    Converters without a dispatch is set out of service.
    All the grid forming converters are set in service in the "Control selection of Grid-forming control type" Cell. (Right above). Thus, these dont need to be set in service here.
    """
    app.PrintPlain("Setting all 'Grid-following' converters in service...")
    composite_models = app.GetCalcRelevantObjects("*.ElmComp*")
    for model in composite_models:
        if model.loc_name.startswith("IEC_"):
            model.outserv = 0  # Set in service (assuming 0 is in service)
            app.PrintPlain(f"Converter '{model.loc_name}' set in service.")
    for model in composite_models:
        if model.loc_name.startswith("WECC Large-scale PV Plant_"):
            model.outserv = 0  # Set in service (assuming 0 is in service)
            app.PrintPlain(f"Converter '{model.loc_name}' set in service.")

    # Name of the worksheet
    w1 = wbin.Worksheets("II3050")

```

```
    # All static generators with a "following" control mode
    StatG_following = w1.Range("A60:Z117").Value

    # All converters with a "forming" control mode
    StatG_forming = w1.Range("A199:Z301").Value

    # All converters with a "following" control mode for PV
    StatG_PV_following = w1.Range("A118:Z162").Value

    composite_models = app.GetCalcRelevantObjects("*.ElmComp*")

    # Find converters in StatG_forming with zero output
    app.PrintPlain("Checking converters in StatG_forming...")
    for row in StatG_forming:
        name = row[0]
        if float(row[5]) == 0:  # Check if dispatch is zero
            app.PrintPlain(f"Converter '{name}' identified with zero output.")
            # Set associated controllers out of service
            control_name_prefix = name.split("_", 1)[1]  # Extract the name after "Converter_
                "
            for control_type in ["VSM", "Synchroverter Control", "Droop Control System"]:
                control_name = f"{control_type}_{control_name_prefix}"
                control = [c for c in composite_models if c.loc_name == control_name]
                if control:
                    control[0].outserv = 1  # Set control mode out of service
                    app.PrintPlain(f"Control mode '{control_name}' set out of service.")

    # Find converters in StatG_following with zero output
    app.PrintPlain("Checking converters in StatG_following...")
    for row in StatG_following:
        name = row[0]
        if name.startswith("W_") and float(row[5]) == 0:  # Check if dispatch is zero
            converter_name = name.split("_", 1)[1]  # Extract the name after "W_"
            app.PrintPlain(f"Converter '{converter_name}' identified with zero output.")
            control_name = f"IEC_{converter_name}"
            control = [c for c in composite_models if c.loc_name == control_name]
            if control:
                control[0].outserv = 1  # Set control mode out of service
                app.PrintPlain(f"Control mode '{control_name}' set out of service.")

    # Check PV converters in StatG_PV_following with zero output and set controls out of
        service
    app.PrintPlain("Checking PV converters in StatG_PV_following...")
    for row in StatG_PV_following:
        name = row[0]
        if name.startswith("L_PV_") and float(row[5]) == 0:  # Check if dispatch is zero
            # Extract the unique part of the converter's name
            unique_name = name.replace("L_PV_", "")  # Removes the prefix "L_PV_"
            app.PrintPlain(f"PV Converter '{unique_name}' identified with zero output.")
            # Construct the control name using the unique part
            control_name = f"WECC Large-scale PV Plant_{unique_name}"
            # Find the control by name
            control = [c for c in composite_models if c.loc_name == control_name]
            if control:
                control[0].outserv = 1  # Set control mode out of service
                app.PrintPlain(f"Control mode '{control_name}' set out of service.")



# %% Dynamic Load models update
# Updating the model parameters of the inverse droop controls (Electrolyser, Load response (
    LR) and Power to heat (PTH))

# For dynamic modeling, it does not use the load flow inputs, thus this needs to be added by
    its own. This is what the script does
# The different parameters are calculated based on the same scalings as the example control
    parameter values, these can be changed as necessary

def update_dynamic_model_parameters(Load_1):
    """
    Update dynamic model parameters for Electrolyser inverse droop controls based on
```

```
        specified load data.
    Load_1 is defined in the dispatch implementation, it is the excel range that describes
        the dispatches of the loads in the system (including the electrolysers, regular loads
        , power to heat and load response)
    """
    for row in Load_1:
        name_base = row[0]
        plini_value = float(row[5])         # Active power setpoint for the load. This needs
            to be changed based on the case simulated. Row [5] is the active power column for
             Case A in excel. This moves 3 columns to the right for each case.
        p_max_from_excel = float(row[2])    # Fetching P_max from the third column

        # Determine the model type
        model_type = ""
        if "Electrolyser" in name_base:
            model_type = "Electrolyser"
        elif "_LR" in name_base:
            model_type = "LR"
        elif "_PTH" in name_base:
            model_type = "PTH"

        name_base = name_base.replace("Electrolyser_", "").replace("Load_", "")
        dynamic_model_name = "Electrolyser␣Droop_" + name_base + ".ElmDsl"
        dynamic_models = app.GetCalcRelevantObjects(dynamic_model_name)

        if dynamic_models:  # If the model is found
            dynamic_model = dynamic_models[0]

            # Correctly call update_parameter for each parameter to be updated
            update_parameter(dynamic_model, 'P_ref', plini_value, model_type,
                p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'FCR_db', plini_value, model_type,
                p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'FCR_bid', plini_value, model_type,
                p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'grad', plini_value, model_type, p_max_from_excel
                , dynamic_model_name)
            update_parameter(dynamic_model, 'bid_min', plini_value, model_type,
                p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'P_min', plini_value, model_type,
                p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'Bid_max', plini_value, model_type,
                p_max_from_excel, dynamic_model_name)
            update_parameter(dynamic_model, 'P_max', plini_value, model_type,
                p_max_from_excel, dynamic_model_name)
            # Repeat for other parameters as needed...
        else:
            app.PrintWarn(f"No␣dynamic␣model␣found␣with␣the␣name␣{dynamic_model_name}")

def update_parameter(dynamic_model, parameter_name, plini_value, model_type, p_max_from_excel
    , dynamic_model_name):
    """
    Update a specific parameter of a dynamic model.

    Parameters:
        dynamic_model (object): The dynamic model object to be updated.
        parameter_name (str): The name of the parameter to be updated.
        plini_value (float): The active power setpoint value.
        model_type (str): The type of the model (e.g., "Electrolyser", "LR", "PTH").
        p_max_from_excel (float): The maximum active power from the Excel sheet.
        dynamic_model_name (str): The name of the dynamic model.
    """
    # Conditional logic to handle each parameter's specific update requirements
    if parameter_name == 'FCR_db':
        value = 0.1 if model_type in ['Electrolyser', 'PTH'] else 0.15   # Different
            operational limits based on the type of dynamic load
    elif parameter_name == 'FCR_bid':
        value = 0.3 * plini_value
    elif parameter_name == 'grad':
        value = plini_value
    elif parameter_name == 'Bid_min':
```

```
        value = -plini_value
    elif parameter_name == 'P_min':
        value = 0
    elif parameter_name == 'Bid_max':
        value = plini_value
    elif parameter_name == 'P_max':
        value = p_max_from_excel
    else:
        value = plini_value  # Default case, for P_ref and potentially other straightforward updates

    setattr(dynamic_model, parameter_name, value)
    app.PrintInfo(f"Updated {parameter_name} to {value} for model {dynamic_model.loc_name} ({model_type})")









# %% Virtual impedance and output voltage calculation parameters adjustment
# Here, the different virtual impedance and output voltage calculation parameters are adjusted

def update_all_virtual_impedance_modes():
    """
    Update parameters for all Virtual Impedance and Output Voltage Calculation models in the project.
    The primary focus here is the maximum short-circuit current, and the current limitation mode of the controllers
    It is  also possible to test for different impedance values etc, but this needs to be implemented. This can be done in the same manner.
    """
    # Retrieve all DSL model objects in the project.
    all_dsl_models = app.GetCalcRelevantObjects("*.ElmDsl")

    for model in all_dsl_models:
        try:
            # Update parameters for "Virtual Impedance" models
            if "Virtual Impedance" in model.loc_name:
                # model.SetAttribute('e:params:r', 0.006)
                model.SetAttribute('e:params:Mode', 1)
                model.SetAttribute('e:params:i_lim', 1.1)  # 1.1 originally
                app.PrintInfo(f"Virtual Impedance model updated: {model.loc_name}")

            # Update parameters for "Output Voltage Calculation" models
            elif "Output Voltage Calculation" in model.loc_name:
                model.SetAttribute('e:params:Mode', 1)
                model.SetAttribute('e:params:i_max', 1.2)  # 1.2 originally
                app.PrintInfo(f"Output Voltage Calculation model updated: {model.loc_name}")

        except AttributeError as e:
            app.PrintWarn(f"Could not update model {model.loc_name}: {e}")



# %% Update grid-forming controller parameters

"""
Here, the main parameters of the controllers are selected and are able to be adjusted to what value is wanted.
Here, all the parameters of the controllers can be added easily, by doing it in the same manner as done here.
The original values are also given, to easier allow for a reset.
"""

def update_all_VSM_control_settings():
    """
    Update parameters for all Virtual Synchronous Machine control models in the project.
    """
```

```
    # Retrieve all DSL model objects in the project.
    all_dsl_models = app.GetCalcRelevantObjects("*.ElmDsl")

    for model in all_dsl_models:
        try:
            # Update parameters for "Droop Control" models
            if "Virtual Synchronous Machine" in model.loc_name:
                model.SetAttribute('e:params:Ta', 3)     # Original: 3
                model.SetAttribute('e:params:Dp', 100)   # Original: 100
                app.PrintInfo(f"VSM Control model updated: {model.loc_name}")

        except AttributeError as e:
            app.PrintWarn(f"Could not update model {model.loc_name}: {e}")

def update_all_Syncroverter_control_settings():
    """
    Update parameters for all Synchronverter control models in the project.
    """
    # Retrieve all DSL model objects in the project.
    all_dsl_models = app.GetCalcRelevantObjects("*.ElmDsl")

    for model in all_dsl_models:
        try:
            # Update parameters for "Droop Control" models
            if "Synchronverter" in model.loc_name:
                model.SetAttribute('e:params:Ta', 3)     # Original: 3
                model.SetAttribute('e:params:Dp', 100)   # Original: 100
                model.SetAttribute('e:params:Kq', 1000)  # Original: 1000
                model.SetAttribute('e:params:Dq', 20)    # Original: 20
                app.PrintInfo(f"VSM Control model updated: {model.loc_name}")

        except AttributeError as e:
            app.PrintWarn(f"Could not update model {model.loc_name}: {e}")



def update_all_droop_control_settings():
    """
    Update parameters for all Droop Control models in the project.
    """
    # Retrieve all DSL model objects in the project.
    all_dsl_models = app.GetCalcRelevantObjects("*.ElmDsl")

    for model in all_dsl_models:
        try:
            # Update parameters for "Droop Control" models
            if "Droop Control" in model.loc_name:
                model.SetAttribute('e:params:mp', 0.01) #Original: 0.01
                model.SetAttribute('e:params:mq', 0.1) # Original: 0.05
                app.PrintInfo(f"Droop Control model updated: {model.loc_name}")

        except AttributeError as e:
            app.PrintWarn(f"Could not update model {model.loc_name}: {e}")




# %% Update the inertia and kinetic energy levels

"""
Here, the inertia constant and kinetic energy levels of the AC interconnections can be
    changed
Select the kinetic energy and inertia constant wanted for the system, and this will be
    transformer to PowerFactory values. This conversion can be seen in the "interconnection"
    section in the 2050 scenario implementation chapter in the thesis.
The dynamic models of the AC interconnections also needs to be updated, to be responsible for
     the maximum power output of the machines
"""

# Initialize known interconnection levels for each specific synchronous machine
```

```
interconnection_levels = {
    "IC_AMPRION_GNA380_1": 1650,
    "IC_AMPRION_NDR380_1": 1650,
    "IC_AMPRION_OBZ380": 1650,
    "IC_ELIA_MBT380_1": 1750,
    "IC_ELIA_ZVL380_1": 1750,
    "IC_TenneT_DIL380_1": 5000
}

# Set the kinetic energy and inertia constant values directly
kinetic_energy = 1800000  # Replace with the desired value in GVA
inertia_constant = 4.1  # Replace with the desired value in seconds
new_sgn_value = kinetic_energy / inertia_constant

def update_selected_synchronous_machines():
    """
    Update selected synchronous machine parameters.
    """
    # Retrieve all relevant TypSym models in the project
    all_typ_sym = app.GetCalcRelevantObjects("*.TypSym")

    # Names of interest for filtering
    target_names = [
        "Gen_Interconnection_1650",
        "Gen_Interconnection_1750",
        "Gen_Interconnection_5000"
    ]

    # Loop through all TypSym models and filter by name
    for typ_sym in all_typ_sym:
        try:
            # Check if the object's name matches any in the target list
            if typ_sym.loc_name in target_names:
                # Update the 'sgn' attribute (rated apparent power)
                typ_sym.SetAttribute('sgn', new_sgn_value)

                # Update the 'h' attribute (inertia constant)
                typ_sym.SetAttribute('h', inertia_constant)

                app.PrintInfo(f"Updated␣{typ_sym.loc_name}␣with␣sgn={new_sgn_value}␣and␣h={
                    inertia_constant}")
        except AttributeError as e:
            app.PrintWarn(f"Could␣not␣update␣{typ_sym.loc_name}:␣{e}")

# Function to update the IEEEG1 DSL models within each relevant ElmComp
def update_ieeeg1_dsl_models():
    """
    Update parameters for all IEEEG1 models in the project.
    """
    # Base interconnection level to model name mapping
    interconnection_levels = {
        "Plant_Gen_Interconnection_DIL380": 5000,
        "Plant_Gen_Interconnection_GNA380": 1650,
        "Plant_Gen_Interconnection_MBT380_1": 1750,
        "Plant_Gen_Interconnection_NDR380_1": 1650,
        "Plant_Gen_Interconnection_OBZ380": 1650,
        "Plant_Gen_Interconnection_ZVL380_1": 1750
    }

    # Gather all ElmComp objects in the project
    all_elmcomps = app.GetCalcRelevantObjects("*.ElmComp")

    # Log the total number of ElmComp objects found
    app.PrintInfo(f"Total␣ElmComp␣components␣found:␣{len(all_elmcomps)}")

    for comp in all_elmcomps:
        # Check if component matches any relevant interconnection plant
        if comp.loc_name in interconnection_levels.keys():
            app.PrintInfo(f"Processing␣IEEEG1␣models␣in␣{comp.loc_name}")

            # Fetch the interconnection value based on the component name
```

```
            base_value = interconnection_levels[comp.loc_name]
            app.PrintInfo(f"Using base value: {base_value} for calculations.")

            # Calculate parameters based on the given kinetic energy and inertia
            uc = -((0.25 * base_value) / new_sgn_value)
            pmin = -((base_value) / new_sgn_value)
            uo = abs(uc)
            pmax = abs(pmin)

            # Find all IEEEG1 DSL models within this component
            ieeeg1_models = comp.GetContents("*.ElmDsl", recursive=True)

            if ieeeg1_models:
                app.PrintInfo(f"Found {len(ieeeg1_models)} IEEEG1 models in {comp.loc_name}")
                for model in ieeeg1_models:
                    if "IEEEG1" in model.loc_name:
                        # Update the model parameters
                        model.SetAttribute("e:params:Uc", uc)
                        model.SetAttribute("e:params:Pmin", pmin)
                        model.SetAttribute("e:params:Uo", uo)
                        model.SetAttribute("e:params:Pmax", pmax)
                        app.PrintInfo(f"Updated IEEEG1 model: {model.loc_name} with Uc={uc}, 
                            Pmin={pmin}, Uo={uo}, Pmax={pmax}")
            else:
                app.PrintInfo(f"No IEEEG1 models found in {comp.loc_name}")



#%% Load flow functions for initialization and results

def loadflow():
      Ldf = app.GetFromStudyCase('ComLdf')
      Ldf.Execute()


#%% Define whether simulations are run from python, and which results need to be exported

def simulations():
        loadflow()                                  # execute a load flow


#%%% Simulations for 2050
# This section activates different scenarios for the year 2050, setting capacities, controls, 
    and dispatches accordingly.
# If a scenario needs to be commented out/in, make sure the whole relevant section is 
    commented out, as shown in the guidelines.

# Retrieve operational scenario folders
OSfolder = app.GetProjectFolder('scen')
OS = OSfolder.GetContents()

# Loop through each operational scenario folder
for opscenfold in OS:
    app.PrintPlain(opscenfold)
    # Check if the folder corresponds to the desired scenario
    if opscenfold.loc_name == 'II3050-2':
        sheetname = "II30502"  # Define the sheet name in the file "Scenario Input Data.xlsx"
        opscen = opscenfold.GetContents()  # Get the contents of the operational scenario 
            folder
        # Loop through each scenario in the folder
        for scen in opscen:
            # Activate the scenario and perform relevant updates
            if scen.loc_name == 'CaseA':
                scen.Activate()
                capacities2050()
                # dispatches1()
                # reactivepowercontrol('Wind')
                # grid_forming_control('VSM')
                # set_converter_controls_out_of_service()
                # update_all_droop_control_settings()
                # update_all_VSM_control_settings()
```

```
                # update_all_Syncroverter_control_settings()
                # update_all_virtual_impedance_modes()
                update_selected_synchronous_machines()
                update_ieeeg1_dsl_models()
                # simulations()

            # Uncomment and adapt for each scenario (B to F) as needed
            # if scen.loc_name == 'CaseB':
            #     scen.Activate()
            #     dispatches2()
            #     reactivepowercontrol('Wind')
            #     grid_forming_control('VSM')
            #     set_converter_controls_out_of_service()
            #     update_selected_synchronous_machines()
            #     update_ieeeg1_dsl_models()
            #     update_all_VSM_control_settings()
            #     simulations()
            # if scen.loc_name == 'CaseC':
            #     scen.Activate()
            #     dispatches3()
            #     # reactivepowercontrol('PV')
            #     grid_forming_control('VSM')
            #     set_converter_controls_out_of_service()
            #     update_selected_synchronous_machines()
            #     update_ieeeg1_dsl_models()
            #     simulations()
            # if scen.loc_name == 'CaseD':
            #     scen.Activate()
            #     dispatches4()
            #     # reactivepowercontrol('Wind')
            #     grid_forming_control('VSM')
            #     set_converter_controls_out_of_service()
            #     update_selected_synchronous_machines()
            #     update_ieeeg1_dsl_models()
            #     simulations()
            # if scen.loc_name == 'CaseE':
            #     scen.Activate()
            #     dispatches5()
            #     # reactivepowercontrol('Wind')
            #     grid_forming_control('VSM')
            #     set_converter_controls_out_of_service()
            #     simulations()
            # if scen.loc_name == 'CaseF':
            #     scen.Activate()
            #     dispatches6()
            #     update_selected_synchronous_machines()
            #     update_ieeeg1_dsl_models()
            #     # reactivepowercontrol('PV')
            #     grid_forming_control('VSM')
            #     set_converter_controls_out_of_service()
            #     simulations()
```