Data-Driven Methods for Distribution Grid Modeling and Topological Reduction of Transmission Grid Models

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Moritz Weber
geb. in Marburg

Tag der mündlichen Prüfung	10.07.2025
Hauptreferent	Prof. Dr. Veit Hagenmeyer Karlsruher Institut für Technologie
Korreferent	Univ.-Prof. Antonello Monti, PhD Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen

Moritz Weber
Institute for Automation and Applied Informatics (IAI)
Karlsruhe Institute of Technology (KIT)
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen
Germany

DOI: 10.5445/IR/1000191400

Acknowledgement

This thesis was a lot of work for me, but I didn't get there alone, and I want to thank the people who supported me on the way. First, I wish to thank my supervisors: Veit Hagenmeyer for his guidance in pursuing this degree and for helping me tie this work together. Uwe Kühnapfel for sharing his vast knowledge about power systems, his invaluable models that are the basis for much of my work, and the opportunity to become a visiting scholar in the US. Hüseyin Kemâl Çakmak for his endless stream of ideas and advice, for his guidance and the freedom to pursue my ideas, and for always having an open ear for my opinions, enabling me to grow. I would also like to thank Antonello Monti for being a reviewer of my dissertation.

I thank my colleagues, who make working from home a tough choice. To just name a few, my ESA colleagues, Alex, Burak, Erfan, Hao, and Jovana, the ones I mostly see during breaks, Dorina, Jakob, Julian, Marvin, and Simon, and my office mates Moritz & Moritz. For making the non-scientific aspects of this work as painless as possible, I thank the infrastructure team and the administrative team, especially Bernadette and Andreas. For making the project work a breeze, I thank my partners in ENSURE and ESD, especially the colleagues from FAU. I thank the students who made substantial contributions to this thesis: Alex (again), Jona, and Luc. Special thanks go to a couple of colleagues: to Anselm for his precision and for supporting me in all steps of the process. To Chang for the countless on- and off-topic conversations during our breaks, often providing a different perspective on the world and challenging my views—and of course, the excursions to Yangda. To Marian for fundamentally shaping my writing process and being an invaluable advisor. To Michael for getting me started in the real-time simulation world. To Richard for sharing his experiences with the doctoral process, power systems, and academia.

Finally, I would like to thank my friends and family. My friends for the great times together, keeping my work-life balance in check. My parents and my sister for laying the foundation for everything I am today, providing me with all the necessary support and so many tools to pursue this endeavor: independence, patience, critical thinking, confidence, teamwork, and the unshakable belief that everything will be alright. This would not have been possible without you.

Karlsruhe, November 2024 *Moritz Weber*

Abstract

With the transition from fossil fuels as a primary energy source to renewable energy sources, power grids worldwide face drastic changes. Centralized and controllable energy generation with traditional power plants is gradually replaced with decentralized and volatile generation from renewable sources. Additionally, the electrification of various sectors, such as transportation and heating, significantly changes the demand for electric energy. These changes pose significant challenges to all voltage levels of the electric power grid, from high-voltage transmission grids to low-voltage distribution grids, and require innovative solutions. Developing these solutions requires a wide variety of simulations of the relevant power systems. Therefore, this thesis proposes methods to create the necessary power system models for distribution grids as well as transmission grids.

For the simulation of distribution grids, this thesis introduces an imputation method to fill missing values in energy time series and a method to automatically generate distribution grid models from openly available data sources. The imputation method exploits periodicities often found in demand and generation time series and copy-pastes matching chunks of data into the gaps considering energy quantities. Filling these gaps helps with using previously incomplete time series for grid analysis and modeling. The method to automatically generate distribution grid models utilizes open map data to estimate the demands of buildings and creates power grids based on street layouts in a two-stage optimization process for medium-voltage and underlying low-voltage grids.

To tackle the computational burden of dynamic simulations of large transmission grids, this thesis presents two new model reduction methods. The first method performs a static reduction that, in contrast to conventional methods, preserves the basic topology of the grid and allows the tracing of power system components from the detailed model to the reduced model. This traceability allows, for example, to handle the geographic information of the system components adequately, which is crucial for modern power systems with a high share of renewable energy generation. To also replicate the dynamic behavior of detailed models in a simplified way, the second method uses a genetic algorithm to optimize the controllers of synchronous machines in the reduced system.

Kurzfassung

Mit dem Umstieg von fossilen Brennstoffen auf erneuerbare Energien stehen die Stromnetze weltweit vor großen Veränderungen. Die zentrale und kontrollierbare Energieerzeugung mit traditionellen Kraftwerken wird durch eine dezentrale und volatile Erzeugung aus erneuerbaren Quellen ersetzt. Die Elektrifizierung unterschiedlicher Sektoren wie Verkehr und Wärme hat einen erheblichen Einfluss auf die Nachfrage nach elektrischer Energie. Diese Veränderungen stellen alle Spannungsebenen des Stromnetzes vor große Herausforderungen und erfordern innovative Lösungen, wofür umfangreiche Simulationen der relevanten Energiesysteme nötig sind. In dieser Arbeit werden die dazu notwendigen Methoden zur Erstellung der erforderlichen Modelle für Verteil- und Übertragungsnetze vorgestellt.

Zur Simulation von Verteilnetzen führt diese Arbeit eine Imputationsmethode für Energiezeitreihen ein, sowie eine Methode zur automatischen Generierung von Verteilnetzmodellen aus öffentlichen Datenquellen. Die Imputationsmethode nutzt Regelmäßigkeiten in Last- und Erzeugungszeitreihen und fügt passende Datenblöcke unter Berücksichtigung der tatsächlichen Energiemengen in die Lücken ein. Dadurch wird die Verwendung ursprünglich unvollständiger Zeitreihen für die Netzanalyse und -modellierung ermöglicht. Die Methode zur automatischen Generierung von Verteilnetzmodellen nutzt freie Kartendaten, um den Energieverbrauch von Gebäuden zu schätzen, und erstellt Stromnetze auf der Grundlage von Straßenverläufen in einem zweistufigen Optimierungsprozess für Mittel- und Niederspannungsnetze.

Zur Bewältigung des Rechenaufwands für dynamische Simulationen großer Übertragungsnetze stellt diese Arbeit zwei neue Methoden zur Modellreduktion vor. Die erste Methode führt eine statische Reduktion durch, die—im Gegensatz zu herkömmlichen Methoden—die grundlegende Topologie des Netzes erhält und die Zuordnung von Stromsystemkomponenten aus dem detaillierten Modell in das reduzierte Modell ermöglicht. Diese Zuordnung ermöglicht es beispielsweise, die geografischen Informationen der Systemkomponenten angemessen zu verarbeiten, was für Stromsysteme mit einem hohen Anteil an erneuerbarer Energieerzeugung von entscheidend ist. Um auch das dynamische Verhalten detaillierter Modelle vereinfacht nachzubilden, optimiert die zweite Methode mithilfe eines genetischen Algorithmus die Regler von Synchronmaschinen im reduzierten System.

Contents

Acronyms

AC	alternating current
ANN	artificial neural network
AoI	area of interest
BW	Baden-Württemberg
CCE	cable capacity estimation
CHiL	control hardware-in-the-loop
CIM	Common Information Model
CPI	copy-paste imputation
DC	direct current
DG	distributed generation
DSO	distribution system operator
EM	electricity meter
EMT	electromagnetic transient
EV	electric vehicle
FDNE	frequency-dependent network equivalent
GA	genetic algorithm
GDF	generic data format
GUI	graphical user interface
HiL	hardware-in-the-loop
HV	high-voltage
HVDC	high-voltage direct current
LMP	local marginal price
LOCF	last observation carried forward
LV	low-voltage

MAE mean absolute error

MAPE mean absolute percentage error

MILP mixed integer linear program

MRE mean relative error

MV medium-voltage

MXE maximum error

MXPE maximum percentage error

NSGA-II improved non-dominated sorting genetic algorithm

OPF optimal power flow

OSM OpenStreetMap

OWA optimally weighted average

PF power flow

PSS power system stabilizer

PTDF power transfer distribution factor

PV photovoltaics

REI radial, equivalent, and independent

RMS root mean square

RNM reference network model

RNN recurrent neural network

SEXS simplified excitation system

SPA singular perturbations analysis

TPNR topology-preserving network reduction

TSA transient stability assessment

TSP travelling salesman problem

WAPE weighted absolute percentage error

Part I

Introduction

1 Motivation

To combat climate change on a global level, the United Nations adopted the Paris Agreement in 2015 with the goal to "limit the temperature increase to $1.5\,^{\circ}\mathrm{C}$ above pre-industrial levels" [1]. Reducing the emission of greenhouse gases is a key to limiting the increase of the global average temperature. Consequently, the transition from fossil fuel-based to carbon-neutral energy generation is a cornerstone of limiting global warming. The transition to renewable energy sources, however, results in massive changes to the world's power grids. The German power grid is no exception to this: Germany's last nuclear power plants were shut down in 2023, coal-based energy generation is planned to phase out until 2038, and the installed capacity of wind parks and photovoltaics (PV) systems—in large PV farms, but also on rooftops and balconies—increases constantly. Simultaneously, other sectors become increasingly electrified, such as the transport sector with the adoption of electric vehicles (EVs) and the heating sector with an increasing number of installed heat pumps replacing gas and oil heating. These developments change the shape of the power grid dramatically: from a few large, centralized, and controllable power plants that supply the grid from the high-voltage transmission grid to the lower voltage levels, to numerous, decentralized, and volatile generation units, located on all voltage levels of the grid. Electrical **distribution grids** now include an increasing amount of energy generation themselves and have to integrate new consumers due to the electrification of various energy-consuming sectors. As sufficient renewable energy generation might not be available in regions with a high energy demand, this energy might need to be transported over long distances, potentially requiring changes in the **transmission grids** [2].

1.1 Challenges in Distribution Grids

The distribution grid faces several challenges due to the changes on the road to carbon neutrality. Traditionally designed to supply consumers with energy, distribution grids nowadays contain an increasing share of prosumers, which not only consume energy, but also produce energy, injecting any surplus into the grid. This energy is largely generated by wind turbines and PV systems—in low-voltage (LV) grids

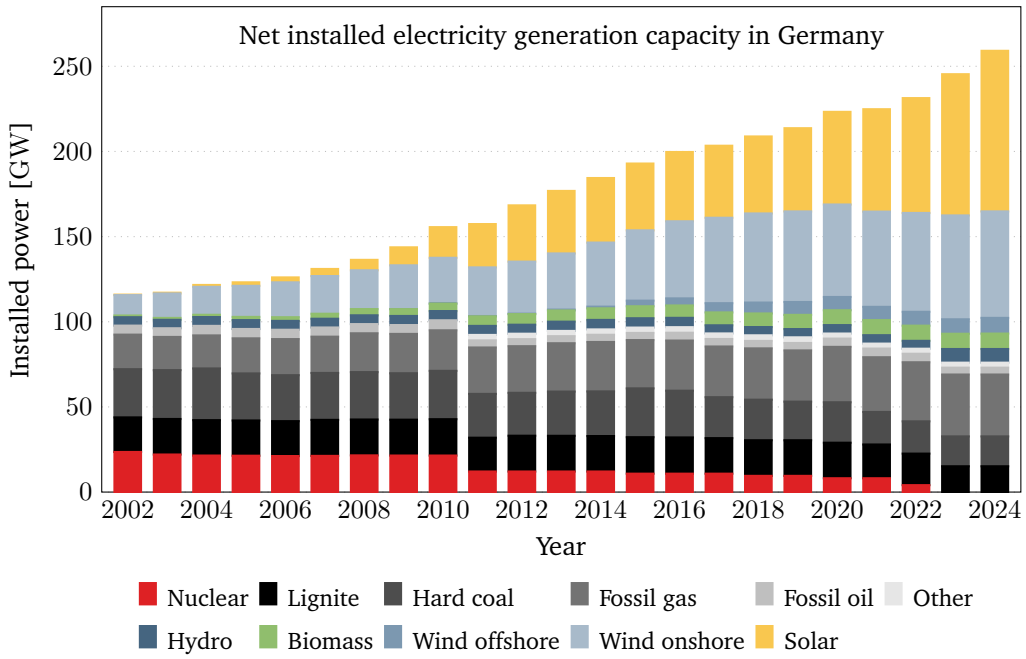


Fig. 1.1.: The installed electricity generation capacity in Germany shows a decline of conventional generation from 98 GW in the year 2002 to 76 GW in 2024. In the same time, the installed wind and solar generation capacity increased from 12 GW to 166 GW [3].

especially rooftop PV. In some scenarios, these new generation units might lead to an inversion of power flow: more power can be generated in distribution grids than is consumed. This leads to entirely new considerations regarding the adherence to voltage limits and the loading of lines and transformers.

On the other hand, new consumers are installed in distribution grids constantly, e.g., due to the electrification of energy-intensive sectors, such as heating and transportation. The increasing number of installed heat pumps puts a strain on distribution grids, especially in seasons with less generation through PV. The increasing adoption of electric vehicles poses a further challenge for distribution grids. With potentially high charging power, uncontrolled simultaneous charging of EVs can overload distribution grids [4, 5]. However, with the potential of bidirectional power flow, the batteries of EVs can also act as a source for flexibility in the grid and thus be part of a solution [6, 7]. Considering the size of the distribution grids, these challenges lead to the estimation that the largest part of required grid expansions lies in the medium-voltage (MV) and LV distribution grids [8].

Due to the challenges and subsequently high cost of adapting the distribution grids, it is imperative to have a good understanding of these grids, e.g., to evaluate potential solutions and identify arising problems. Relevant questions include: How much renewable generation can be integrated into the grid [9]? Analogous questions

arise, e.g., for EV charging stations and heat pumps. Are there scenarios and grid configurations that lead to voltage band violations or overloading of grid components, such as transformers [10, 11, 12]? If so, how can these problems be mitigated? How can the grid operation be optimized, e.g., by developing improved algorithms [13, 14]?

Answering such questions requires realistic models of distribution grids. However, detailed models of real distribution grids are not readily available. Furthermore, distribution system operators (DSOs) are often reluctant to share datasets about their grids, as they often pose a security threat or at least a trade secret [10, 11, 14]. For research purposes, there are reference models available that mitigate the lack of models of real systems to some extent [9, 15, 16]. These reference models, however, have two considerable downsides: first, they have only limited significance when trying to answer questions about a specific real network, as the results achieved with a reference network might not be valid for a real network [10]. Second, the available reference networks are typically not large and complex enough to test new algorithms and solutions thoroughly enough [13].

Given this, there is a need for realistic distribution grid models that are based on real infrastructure. It is logical to use publicly available data, such as the widely available open map data of the OpenStreetMap (OSM) project, to create such models and to automate the modeling process, as manual modeling is tedious and error-prone.

1.2 Challenges in Transmission Grids

The transition from conventional power plants to renewable energy generation poses several challenges for transmission grids. Traditionally, power plants are located close to regions with high consumption, such as large cities and energy-intensive industries. Renewable energy sources, however, might not be as available in these regions and thus require the energy transmission over long distances [2]. Without adapting the transmission grids to these new circumstances, transmission capacities can easily be reached. In Germany, this is often the case on days with high wind energy generation, which is intensified when combined with high solar generation [17]. When transmission capacities are fully utilized, countermeasures to ensure grid stability have to be taken, such as redispatch and feed-in management [18]. These measures can become costly, and calculating adequate pricing for the involved market participants is not trivial [19]. To avoid deploying these countermeasures, usually two approaches can be applied: optimization of the grid operation and,

if necessary, grid expansion. Both of these approaches require accurate models of transmission grids to identify the potentials for optimization, to develop and improve algorithms, and to plan adequate expansion strategies. Long-term grid studies are crucial to cover the different seasons and weather conditions, as they have a strong influence on the grids, especially in systems with a large share of renewable energy generation [20]. These long-term studies usually require many power flow calculations. Using a fully detailed model of a country's transmission grid is often infeasible and would result in extremely long computation times. To gather insights for the detailed system, it is thus common to utilize models with a reduced complexity for these studies.

In addition to long-term studies, the development of new solutions to optimize or expand the grid often requires dynamic grid simulations of shorter time periods. One of these solutions to optimize the operation of the transmission grids includes so-called grid boosters. These grid boosters incorporate some form of energy storage, e.g., battery- or hydrogen-based, and allow a higher capacity utilization of existing lines as they can support the grid in the case of a failing transmission line [21]. A common solution for grid expansions is building additional transmission lines. Especially to transport large amounts of energy over long distances, typically high-voltage direct current (HVDC) transmission lines are used. Transporting direct current (DC), these lines are connected to the alternating current (AC) grid via converter stations. To ensure the stability of the grid, the different solutions for optimization and expansion need to be carefully evaluated before integrating them into the grid. This is especially important for grids without the inertia of classic rotating machines, such as large thermal power plants [2], as is the case for many grids around the globe with the transition to converter-based generation from renewable sources. The evaluations for these solutions include dynamic simulations of different fault scenarios. As dynamic grid simulations are computationally expensive, the size and complexity of the simulated grid become a deciding factor. For simulations that incorporate real hardware components, so-called hardware-in-the-loop (HiL) simulations, the computation times of individual time steps are crucial. Converter-based units often undergo so-called control hardware-in-the-loop (CHiL) testing to validate the controllers that determine the behavior of the system. These tests require electromagnetic transient (EMT) simulations with a resolution in the range of usually $10\,\mu s$ to $50\,\mu s$ in real time and are performed on specialized simulation hardware. Utilizing fully detailed transmission grid models for this type of dynamic simulation is often infeasible due to limitations of the computational capacity of current simulation hardware. One solution is expanding the available computational capacity by cooperating with other research teams in so-called distributed

real-time co-simulations [22, 23, 24]. While these collaborations help to concentrate knowledge and resources to tackle large projects, these long-distance connections do not have the same fidelity as local simulator couplings and might not always be a suitable solution for the problem at hand [25].

In conclusion, solving the challenges faced in transmission grids requires comprehensive grid studies and dynamic simulations with a high temporal resolution, in some cases combined with real-time requirements. The sheer size of transmission grids, however, makes a detailed simulation of the systems infeasible and thus requires the reduction of complexity in modeling these systems. The reduction of transmission grid models is thus a cornerstone in the evaluation of new solutions for the transition to renewable energy sources, with applications in various grid study types.

1.3 Research Questions and Contributions

As outlined in the previous section, solving the challenges of the renewable energy transition in distribution and transmission grids requires models to, e.g., gather insights, optimize algorithms, and evaluate hardware solutions. With the high demand for these grid models, the modeling of these systems is an active research topic with a history of multiple decades. However, this thesis addresses open research questions regarding the data-driven modeling of distribution grids and the reduction of transmission grid models that have not been answered satisfactorily in the literature.

1.3.1 Data-Driven Distribution Grid Modeling

There is a clear demand to understand current developments in distribution grids as well as potential future scenarios. However, due to the lack of accurate models created by the grid operators, new methods for investigating distribution grids need to be found. Data-driven approaches seem to offer solutions to mitigate these issues using available data sources, such as open geographic data or time series describing the demand and generation of grid participants. Using these data-driven approaches to gain insights into distribution grids raises several questions that shall be answered in this thesis.

Research Question 1: How to handle missing values in energy time series? Time series of measured demand and generation can be an important source to generate insights into distribution grids, e.g., in forecasting, simulation, modeling or analysis of the time series directly. In the automated grid modeling, for example, actual energy time series could significantly improve the modeling accuracy of load and generation, which is a crucial part for most approaches. With a widespread rollout of new metering devices, these time series also become increasingly available. However, the infrastructure collecting these measurements—from sensors to communication networks—is prone to errors, which might result in the loss of data. Handling these missing values is a crucial preprocessing step for many applications.

While there are existing methods for the imputation of missing values in time series, none of the existing methods focus on energy time series. This thesis contributes to the research area by introducing an imputation method specifically for energy time series, with the goal to fill gaps in time series with realistic data while preserving the overall measured energy.

Research Question 2: How can realistic grid models be generated with minimal available data sources? The lack of distribution grid models is a well-known problem, and the utilization of data-driven methods to create such models is an active research topic. However, existing approaches for the data-driven generation of power grid models often require multiple data sources that might be proprietary or are only available for very limited areas or regions. These requirements often limit the applicability of existing approaches to certain regions. Through the OpenStreetMap (OSM) project, open map data is widely available and, thus, provides a good basis for a generic model creation approach—even though it does not contain the exact routes for underground cables.

To overcome these limitations, this thesis proposes a method to create distribution grid models with minimal data requirements applicable for large-scale power system modeling. Our approach aims to provide a basis for distribution grid modeling that can be extended with further input to improve the resulting model accuracy if additional data sources are available for the area of interest.

Research Question 3: How does the availability of data influence the accuracy of the generated grid models? While utilizing various data sources to automatically create power grid models is an ongoing research topic, little research has been done to assess how the availability and quality of the input data affect the resulting grid models. As the availability and quality likely vary from region to region, it is

essential to examine their effect on the accuracy of the generated models to gauge the reliability of the results and insights derived from these models.

Any attempt to quantify the accuracy and realism of grid models raises the question of finding appropriate measures to compare grid models [26, 27]. As the existing literature does not provide a clear answer to this question, this thesis contributes by introducing several key measures to compare the topological and electrical properties of grid models. Furthermore, we use these measures to compare models created using various data sources in order to assess minimal data requirements for the creation of realistic distribution grid models.

1.3.2 Transmission Grid Model Reduction

Solving the challenges that transmission grids are facing, requires thorough grid studies and simulations to identify weaknesses and evaluate potential solutions. However, detailed transmission grid models are often prohibitively complex for the required studies, considering the realistically available computing capacity. Thus, simplified models of the expansive transmission systems are required to enable gaining relevant insights. Nevertheless, the reduction of detailed grid models is not trivial and raises questions that this thesis addresses.

Research Question 4: How can transmission grid models with considerable amounts of renewable generation be reduced, while retaining crucial geographic information? With their increasing share in the total energy generation, renewable energy sources play a crucial role in the simulation of power systems. As these renewable sources depend heavily on the weather, their geographic locations are vital for an accurate model of the system. This importance is recognized in some model reduction techniques, focusing mostly on the market aspects of the power grid [28]. Existing model reduction techniques with a focus on electrical properties, however, ignore these aspects of the system and lose this valuable information.

Considering the importance of geographic information in power system models, this thesis introduces a new model reduction method with a focus on static network simulations that preserves relevant geographic and structural information in the reduced models.

Research Question 5: How can computationally expensive transmission grid models be simplified to support dynamic simulations, enabling hardware-in-the-loop (HiL) testing for crucial new grid components? The reduction of power system models is not an ongoing effort, as these models have been pushing the boundaries of computational viability for decades. Thus, a multitude of methods to perform such model reductions exist for various applications, including the dynamic simulations required for HiL setups. As stated in the previous research question, existing methods with a focus on electrical properties—especially dynamic simulations—do not consider relevant information, such as the geographic locations of loads and generation.

Geographic information and topological properties have so far only been considered in static model reduction methods. This thesis introduces a new network reduction method for dynamic simulations that considers these crucial properties of transmission grids.

1.4 Thesis Outline

The introduction is concluded by Chapter 2, which introduces some basic concepts that are used throughout the rest of this thesis. Following the introduction, the identified challenges and research questions form the frame for this thesis, dividing it into parts and chapters. Part II focuses on the challenges in distribution grids and answers the corresponding research questions. In this part, Chapter 3 answers research question **RQ1**, addressing the issue of handling missing values in energy time series by introducing a new imputation method. Chapter 4 follows with answering research questions **RQ2** and **RQ3**, addressing the generation and assessment of distribution grid models. To answer these questions, we introduce a new two-stage optimization-based model generation method and compare the models generated using this method from various data sources. Part III addresses the challenges regarding the transmission grid. As the corresponding research questions revolve around reduced network models, Chapter 5 starts this part with general background information and an overview of the state of the art for network equivalencing methods. Chapter 6 follows with answering research question **RQ4** by introducing a new topology-preserving network reduction method for static calculations and comparing it to existing network reduction methods. Extending on this work, Chapter 7 answers research question **RQ5** and presents an optimization-based method for approximating the dynamic behavior of the reduced system. Finally, Part IV concludes this thesis with a summary and an outlook for future developments.

2 Preliminaries

Parts of this chapter are reproduced from

Moritz Weber, Alexander Kocher, Hüseyin K. Çakmak, and Veit Hagenmeyer. "ePowCoRe: A Novel Generic Representation of Power Grids Enabling Open-Source Model Conversion Modules". In: *2024 Open Source Modelling and Simulation of Energy Systems (OSMSES)*. Sept. 2024. DOI: 10.1109/OSMSES62085.2024.10668981.

Moritz Weber, Jona Enzinger, Hüseyin K. Çakmak, Uwe Kühnapfel, and Veit Hagenmeyer. "PyAPI-RTS: A Python-API for RSCAD Modeling". In: *2023 Open Source Modelling and Simulation of Energy Systems (OSMSES)*. Mar. 2023. DOI: 10.1109/OSMSES58477.2023.10089671.

In this chapter, we briefly describe background information that is used and referenced in multiple chapters in the rest of this thesis. This information includes basics on power system simulation types in Section 2.1. The literature uses a wide range of software and hardware solutions to model power systems and perform these different types of simulations. In Section 2.2, we provide a brief overview of these solutions. Many of these solutions maintain their own unique data formats to represent their models, which often hinders collaboration between research teams. To mitigate this issue, we briefly introduce two supporting works on the topic of model representation and conversion in Section 2.3. Lastly, we provide central definitions for error measures used for the evaluations of multiple solutions in the main parts of this thesis.

2.1 Power System Simulation Types

Power system simulations are used for a multitude tasks, including planning, design, development, operation, and control [29]. These simulations serve to answer a wide range of questions: from short-term system responses to events, such as short circuits or load changes, to long-term scenario exploration spanning multiple years

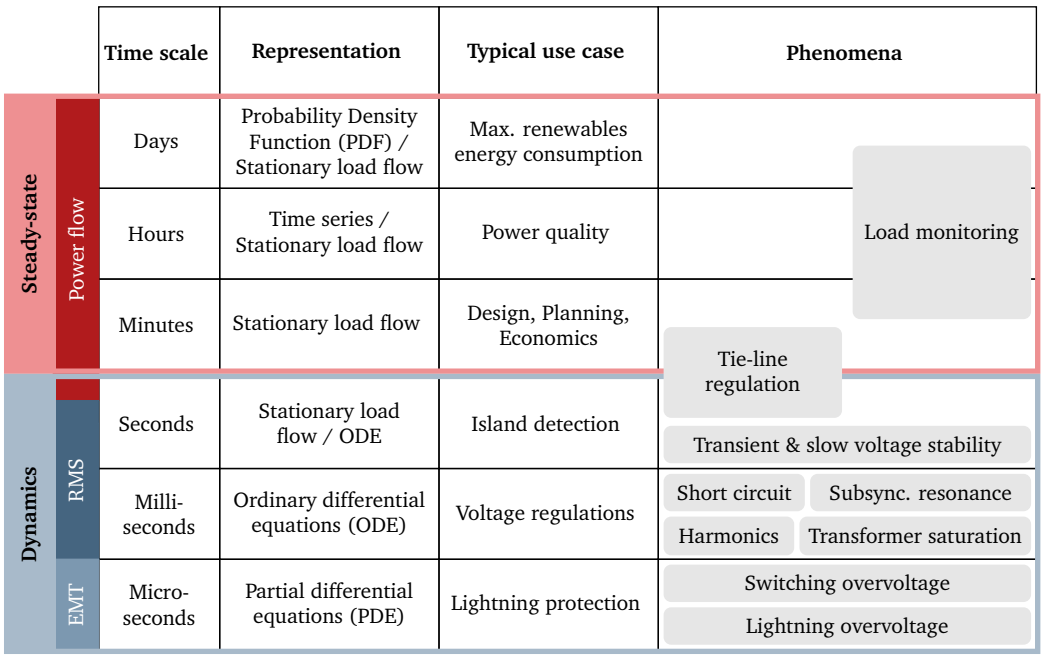


Fig. 2.1.: An overview of power system modeling and simulation types divided into steady-state and dynamic approaches. The wide range of simulation time scales and mathematical representations for the models highlights the diversity of methods that is required to perform the different types of grid studies. Adapted from [30].

of simulated time. Figure 2.1 provides an overview of common use cases, representations, and timescales of energy system models and the phenomena examined with these models. In general, these simulations can be divided into steady-state, most commonly power flow (PF), and dynamic simulations, root mean square (RMS) and electromagnetic transient (EMT). This section aims to provide a brief overview of these types of power system simulations that are relevant throughout this thesis.

The power flow calculation is the most basic of the three simulation types and yields an overview of the distribution of power flowing from generators to consumers and the resulting voltage profiles [31]. It can be used, for example, to identify grid congestions, component overloading, and violations of voltage level standards. In combination with time series that describe system properties, such as load, generation, and switch states, PF calculations can be used as sequential steady-state simulations for long-term studies. In common simulation tools, this type of simulation is referred to as “quasi-dynamic” (PowerFactory) and “quasi-static” (OpenDSS). However, power grids are in constant change: loads change, renewable generation is volatile, errors, and faults might occur in every system component. Therefore, dynamic simulations are required to examine the system behavior between those stationary system states and ensure a stable operation.

Dynamic simulations can be most significantly distinguished by their temporal resolutions. While RMS simulations usually have a time step size around $10\,\mathrm{ms}$, EMT simulations usually operate in the range of $10\,\mu\mathrm{s}$ to $50\,\mu\mathrm{s}$. With their lower temporal resolution, RMS simulations are used to examine the electromechanical dynamics of the system that are dominated by rotating machines, such as turbines and generators in thermal power plants. RMS simulations make the assumption that voltages and currents are sinusoidal variables with a basically constant frequency, which allows representing them as so-called phasors, that consist of the magnitude and the phase angle. EMT simulations, on the other hand, perform their calculations with the instantaneous values of voltages and currents. This becomes important, when these variables deviate from the assumed sinusiodal form, as is the case with inverters, and power electronics in general, that are commonly used for renewable energy generation and HVDC links. With an increasing presence of these components in modern power grids, EMT simulations play an increasingly important role in an accurate system representation. On one hand, the use of instantaneous values and the high temporal resolution allow the integration of real hardware into systems simulated with EMT, if these simulations run in real-time. This HiL testing of equipment under extreme conditions is not possible with real grids and marks a crucial step in the development of new power system components. On the other hand, these properties make EMT simulations very computationally expensive and, in contrast to the other simulation types, unsuitable for long-term studies.

2.2 Common Solutions for Modeling and Simulation

With the different simulation types and the wide range of use cases from planning to operation, naturally there are many software and hardware solutions to meet the modeling and simulation needs of the power system industry. While some tools specialize on a specific use case or simulation type, others offer more general purpose solutions. In the following, we present a brief overview of commonly used modeling and simulation tools. This list is by no means complete and only aims to provide a sense of the landscape of modeling and simulation tools as well as a common reference point for tools that are mentioned throughout this thesis.

DIgSILENT PowerFactory [32] is a very versatile tool for power system modeling and simulation. It supports a variety of simulation types, such as PF and optimal power flow (OPF), short-circuit calculations (SC), and RMS and EMT simulations. In addition to comprehensive modeling capabilities, PowerFactory also provides

extensive visualization and organization tools, including geographic information, operators, and the definition of zones. This allows, for example, using the coordinates of PV systems to calculate the irradiation and resulting power output with common formulas if no irradiation data is available.

PSCAD [33] is an example for a highly specialized software tool that solely focuses on EMT simulations. While offering extensive modeling and simulation capabilities, it does not support real-time EMT simulation, which is often desired in order to integrate actual hardware into simulated systems for HiL simulations.

One solution that offers such real-time EMT simulations is **RSCAD FX** [34], which is the modeling software for real-time simulators by RTDS Technologies Inc. These simulators are highly specialized on EMT simulations and are equipped with various digital and analog interfaces, enabling the integration of hardware into the simulation. This is a highly integrated software and hardware solution that results in a seamless usability but also a fairly closed ecosystem.

Another solution for real-time simulations is offered by **OPAL-RT** [35], providing specialized simulation hardware that can be combined with various software tools covering different use cases. These use cases range from RMS simulations with a temporal resolution of about $10\,\mathrm{ms}$ and tens of thousands of nodes (ePHASORsim) to EMT simulations with down to $10\,\mu\mathrm{s}$ time steps and ten to hundreds of nodes (eMEGAsim / eFPGAsim). Additionally, these systems also offer a multi-domain software solution, RT-LAB, that is fully integrated with MATLAB/Simulink and thus more flexible in its application.

Simscape Electrical [36] with the Specialized Power Systems library for the dynamic simulation of power systems is another MATLAB/Simulink-based solution. It supports simulations with phasors (RMS) and instantaneous values (EMT) with varying temporal resolutions. Through its tight integration with MATLAB/Simulink and the independence from specialized simulation hardware, Simscape is a very flexible tool for all sorts of simulations.

All solutions introduced so far are proprietary, commercial products. However, there are also open-source solutions available for power system simulation. **MATPOWER** [37] is an open-source project, that is widely used by industry and academia for steady-state power flow and optimization problems. MATPOWER itself does not offer a graphical user interface (GUI) or visualization features. Its basic model structure is based on MATLAB and results in a minimalistic data format consisting of matrices.

OpenDSS [38] open-source solution that focuses on the simulation of electric distribution grids. In contrast to MATPOWER, it supports dynamic RMS simulations and offers a basic GUI and visualization features. Similar to PowerFactory, it offers so-called "quasi-static" simulations, which are basically a series of steady-state simulations using time-series.

2.3 Generic Model Representation and Conversion

The wide variety of proprietary and open-source tools for power system modeling and simulation comes with one major downside: all tools have their own unique data format to represent the power system models. This results in a significant problem when cooperating with partners that use other tools or trying to reuse existing models for use cases better supported by another modeling and simulation tool. To mitigate this issue to some extent, many tools include import and less often export methods for third-party data formats. For example, PowerFactory includes several import and export options to formats, including Common Information Model (CIM) and various proprietary formats that are widely used in the industry. While these options are very useful in many cases, the configuration options are limited and the conversion process is not transparent, hindering understanding the resulting models in other tools. Similarly, RSCAD FX supports importing various widely used power system model formats. However, these imports often require manual corrections, as the positioning of components on the model canvas is error-prone, and relevant details might get lost in the conversion process.

Recognizing this issue and to support the wide range of simulation types and studies that are required to solve the challenges identified in Chapter 1, we developed **ePowCoRe**[1]. It is an open-source generic model representation with conversion methods to and from several common modeling tools. In this thesis, ePowCoRe plays a supporting role. It ties in with the main contributions when converting the models generated in Chapter 4 to MATPOWER, opening the models to more researchers, and converting transmission grid models reduced in Part III from PowerFactory to RSCAD FX, enabling real-time EMT simulations with the potential for HiL setups.

ePowCoRe consists of a generic data format (GDF) for power system components and their relations to each other and various import and export methods to convert between this GDF and third-party formats. Table 2.1 provides a brief overview of the tools supported by ePowCoRe: It implements the import of PowerFactory

[1] `https://github.com/KIT-IAI/ePowCoRe`

Tab. 2.1.: An overview of the capabilities of modeling and simulation tools for which ePowCoRe implements conversion methods.

	Power Flow	Optimal PF	Short Circuit	RMS	EMT	HiL	ePowCoRe Import	ePowCoRe Export
PowerFactory	✔	✔	✔	✔	✔		✔	
MATPOWER	✔	✔						✔
Simscape	✔		✔	✔	✔			✔
RSCAD FX	✔		✔		✔	✔		✔

models to its GDF, utilizing PowerFactory's Python API to create GDF components for the supported PowerFactory objects and building the graph, which describes the relations between the components, in the process. ePowCoRe also implements the export from its generic model representation to MATPOWER. The main challenge of the conversion to this minimalistic format is breaking down the more complex component models of other formats to basic models supported by MATPOWER. For Simscape, ePowCoRe implements the export function using the MATLAB Engine API for Python. While this API enables creating and modifying Simulink models programmatically, positioning and connecting the component blocks that comprise the system is a challenging task. For real-time simulations, ePowCoRe implements an export method from its GDF to RSCAD FX, utilizing the **PyAPI-RTS**[2] package, which is another supporting development for this thesis and enables creating and modifying RSCAD FX models. Lastly, ePowCoRe includes an export to GeoJSON [39], which is a data format for shapes with geographic coordinates and can be useful for visualizations of the grid models.

2.4 Common Error Measures

In this section, we define common error measures that are used throughout the rest of this thesis in order to quantitatively evaluate solutions. The mean absolute error (MAE) describes the average of the absolute errors between two sets of values and is defined as

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{x}_i - x_i| \,, \tag{2.1}$$

where x_i represents the true value, $\hat{x}_i$ the corresponding value of the solution achieved by the method under test, and n is the number of total values. In order to

[2] `https://github.com/KIT-IAI/PyAPI-RTS`

represent extreme error values, we utilize the maximum error (MXE), that is defined as

$$\mathrm{MXE} = \arg\max_{\mathrm{e}_i} |\mathrm{e}_i| \,, \quad \text{with } \mathrm{e}_i = \hat{x}_i - x_i. \tag{2.2}$$

This measure returns the error with its corresponding sign that has the highest absolute value. In this case, positive error values mean that the evaluated result is an overestimate, while negative error values indicate underestimates. While these two error measures provide valuable insights into the quality of the evaluated results, it is often easier to assess errors relative to the correct values. Therefore, we consider the mean absolute percentage error (MAPE) and maximum percentage error (MXPE) that are defined analogous to Equations (2.1) and (2.2) with a percentage error instead of the absolute error, which results in

$$\mathrm{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{x}_i - x_i}{x_i} \right| \cdot 100\%, \tag{2.3}$$

$$\mathrm{MXPE} = \arg\max_{\mathrm{e}_i^p} |\mathrm{e}_i^p| \,, \quad \text{with } \mathrm{e}_i^p = \frac{\hat{x}_i - x_i}{x_i} \cdot 100\%. \tag{2.4}$$

While these percentage errors are often more intuitive than the absolute errors, they are impractical in cases where the true values x_i are close to zero, resulting in unreasonably large error values. In these cases, it is useful to consider the absolute error values for more context. For $x_i = 0$, the error value for index i is undefined and thus ignored in the calculation of the mean or maximum error.

In addition to the error definitions above, there is an alternative indexing approach for the evaluated values that is used in some parts of this thesis. The previous definitions evaluate all values from indices 1 to n. However, it is sometimes more suitable to only evaluate certain indices, such as defined by a set S. As an example, the corresponding definition of the MAPE is

$$\mathrm{MAPE} = \frac{1}{|S|} \sum_{i \in S} \left| \frac{\hat{x}_i - x_i}{x_i} \right| \cdot 100\%, \tag{2.5}$$

where $|S|$ is the number of indices compared.

Part II

Data-Driven Methods for Distribution Grids

Overview Part II

Distribution grids face significant changes in their energy consumption and generation behavior. These changes are driven by newly installed renewable energy generation units in the form of wind turbines and PV systems and new consumers, such as heat pumps and EV charging stations. These new consumers pose challenges to distribution grids, such as grid stability concerns and inversion of power flow. Overcoming these challenges calls for a more profound understanding of demand, generation, and the resulting equipment loading and voltages in existing distribution grids and in possible future scenarios to identify problems and evaluate potential solutions. This understanding requires reliable data input and accurate models of distribution grids.

In this part, we address the research questions regarding the previously identified challenges in distribution grids. Time series of measured demand and generation are an important source for gaining insights into distribution grids. However, the infrastructure that generates these time series, from sensors to communication networks, is error-prone, which often causes missing values in such time series. In answering research question **RQ1**, we address this issue. We contribute to this question by proposing a new data-driven imputation method for energy time series in Chapter 3.

Gaining the required understanding of current distribution grids alone, however, cannot solve the identified challenges. Simulations of potential future grid conditions are required to identify possible future issues and to develop and evaluate solutions to overcome these challenges. These simulations require realistic models of distribution grids that are generally not readily available. Thus, we answer the research questions **RQ2** and **RQ3** that address the automated modeling of distribution grids. To answer **RQ2**, in Chapter 4, we propose a new two-stage optimization-based grid modeling method with minimal data requirements, solely relying on openly available map data. This is followed by answering research question **RQ3**, addressing the accuracy of generated models, by proposing metrics to compare distribution grid models and using these metrics to compare models generated using various data sources.

3 Imputation of Energy Time-Series

Parts of this chapter are reproduced from

Moritz Weber, Marian Turowski, Hüseyin K. Çakmak, Ralf Mikut, Uwe Kühnapfel, and Veit Hagenmeyer. "Data-Driven Copy-Paste Imputation for Energy Time Series". In: *IEEE Transactions on Smart Grid* 12.6 (Nov. 2021), pp. 5409–5419. DOI: 10.1109/TSG.2021.3101831.

Distribution grids face several challenges in the wake of the transition to renewable energy sources. Overcoming these challenges requires more profound insights into distribution grids, in order to identify potential problems and solutions. In the form of data, these insights can be gathered from smart meters that are installed in increasing numbers in distribution grids worldwide. They record and transmit a variety of data, such as voltage, reactive power, or the electricity consumption of consumers [40]. The collected data is an essential input to various applications supporting and enabling the transition to renewable energy sources. For example, the collected consumption and generation data enables grid operators to perform grid simulations [41] for stability analysis, grid expansion and digitalization, fault detection, and efficiency improvements. The collected data is also needed for load forecasting [42], load analysis, and load management [43]. Moreover, the collected data allows research facilities to develop new technologies and solutions for the grid of the future.

The results of these applications highly depend on the quality of the input data. The data quality, in turn, is highly influenced by two key challenges in the smart grid infrastructure: the accuracy of data acquisition and the reliability of data transmission and data storage [44]. The accuracy of data acquisition refers to the correctness of the recorded data. It is reduced by problems causing, for example, noise and outliers in the data [45, 46]. For further processing, outliers in particular are often detected and labeled as missing values as a first step, e.g., in [40, 47]. The reliability of data transmission and data storage, however, mainly relates to the completeness of the recorded data. In implemented smart meter systems, recorded

data contains between 3 and 4 % of missing values, for example, because of outages, maintenance, and software updates [44, 48]. With accuracy and reliability as two key challenges of smart grids, missing values in recorded data are a common issue. Although some applications can handle incomplete data [49], most applications require that missing values are handled by data preprocessing.

Thus, in this chapter, we introduce the copy-paste imputation (CPI) method for univariate energy time series. It uses an energy time series as input and copies blocks of data with similar characteristics into gaps, exploiting the periodicity in the data. By copying blocks of matching data, the inherent patterns of the time series are preserved, even in time series with pattern changes. For this purpose, the CPI method utilizes the information about the total energy of each gap that energy time series contain in contrast to power time series. It can, therefore, guarantee that the total recorded energy remains unchanged during the imputation. To the best of our knowledge, no other method in literature has used this property of energy time series for imputation before. By this imputation, the CPI method achieves complete energy time series, which also allows deriving a complete power time series. Both energy and power time series can then serve as input for all the aforementioned applications that support and enable the transition to renewable energy sources.

With this proposed CPI method, we answer research question **RQ1** that addresses the problem of missing values in energy time series. By answering this research question, the proposed imputation method contributes to the understanding of distribution grids, by providing better data for subsequent applications and thus enabling, e.g., more accurate analyses, modeling, and forecasting.

The remainder of this chapter is structured as follows: Section 3.1 presents the state of the art for time series imputation. The proposed method is explained in detail in Section 3.2 and evaluated against three benchmark methods on a real-world dataset in Section 3.3. Section 3.4 summarizes the contributions of this chapter and outlines potential directions for future development.

3.1 State of the Art

Imputation is a common method to handle missing data. It replaces missing values with values that should resemble the actual data [50]. Since missing values are a common problem in real-world datasets, many imputation methods exist for time

series: They range from very basic methods such as linear interpolation and last observation carried forward (LOCF) [50] over time series analysis-based methods [47, 51] to learning-based methods [52, 53].

To further improve the imputation, it is a common approach to focus on time series from a particular domain and to consider their characteristics as additional information. In the context of smart meters, the recorded time series of electricity consumption or generation are typically influenced by factors such as weather, human routines, social norms (e.g., weekends or holidays) and many others [54, 55]. These factors often lead to characteristic patterns with daily, weekly, and yearly periodicities, which can be utilized by imputation methods. For example, daily and weekly patterns are exploited in [56]. The pattern frequency of a power time series is determined with the auto-correlation function and the mean values of the estimated pattern frequency are used to impute missing values. In another work [57], the similarity between days is used by filling larger gaps in a power time series with the average values of validated reference days. Very short gaps with a length of two hours or less are imputed with a linear interpolation, as this often fits the very short-term characteristics of smart meter time series. The optimally weighted average (OWA) approach in [54] utilizes daily and weekly patterns as well as seasonality to select appropriate historical values. With these values, the historical averages of a power time series are calculated, before they are combined with a linear interpolation for smooth transitions between actual and imputed values.

Other methods utilize even more additional data or information to impute missing values in smart meter time series. An example is the method for imputation, denoising, and outlier removal based on principal component pursuit in [58]. It utilizes the spatial correlations in the power load profiles of adjacent substations. In [59], the time series measured by smart meters in a factory are used to impute missing values in other time series from smart meters located in the same factory with clustering and k-nearest neighbors. In [60], the imputation of substation data is formulated as a forecasting problem. The forecast uses the collected power data of nearby substations as well as weather data, which often has an impact on power consumption and generation.

While all of these imputation approaches are specifically designed for smart meter time series, all of them except [59] are limited to the imputation of power time series and none of the approaches utilizes the inherent properties of energy time series. In a power time series P, every entry p_t contains the average power consumption or generation between two time steps $t-1$ and t. However, smart meters typically provide energy time series by default. In an energy time series E, every entry e_t

contains the meter reading, i.e., the energy that has been consumed or generated up to time step t. Therefore,—unlike in power time series—if, for example, entries between the entries e_t to e_{t+3} are missing in an energy time series, the next existing entry e_{t+4} still contains the information about the total energy, which was consumed or produced between $t-1$ and $t+4$. As a consequence, a power time series can be derived from an energy time series with missing values but not vice versa.

3.2 Method: Data-Driven Copy-Paste Imputation for Energy Time-Series

In this section, we describe the proposed copy-paste imputation (CPI) method[1]. As illustrated in the simplified process overview in Figure 3.1, the CPI method uses an energy time series with gaps, i.e., one or multiple consecutive missing values, as input and imputes the missing values by filling them with the best matching days of the same time series. In the following, we describe each step of the CPI method in detail and reference the corresponding lines in Algorithm 1.

3.2.1 Linear Interpolation of Single Missing Values

In the first step of the CPI method, single missing values, i.e., individual meter readings, are imputed in the given energy time series (Line 2). For this imputation, a linear interpolation is used because, assuming a sufficiently high temporal resolution, it provides reasonably accurate estimates for individual missing values. We consider only single missing values in this step to limit the number of consecutively linearly interpolated values and thus potentially unrealistic imputations while still benefiting from these easily imputable values. Indeed, the resulting imputed values are considered correct in the subsequent steps to increase the number of days without missing values that are available for copying.

[1] A Python implementation of the CPI method is available on `https://github.com/KIT-IAI/CopyPasteImputation`.

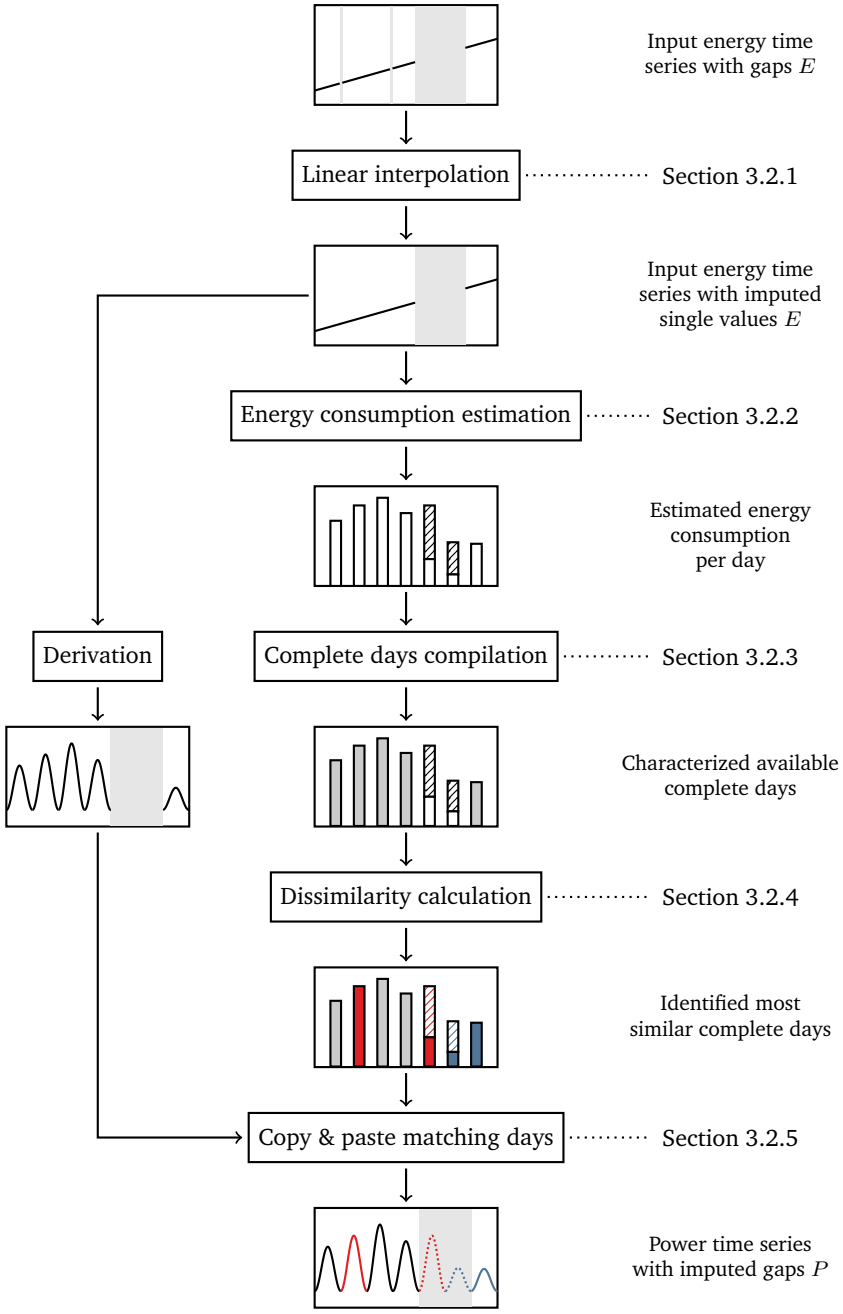


Fig. 3.1.: A simplified illustration of the proposed imputation procedure: An energy time series E is the input for the copy-paste imputation (CPI) method. The first step of the process is the imputation of individual missing values via linear interpolation. This is followed by the energy estimation of the remaining days with missing values. The complete days, i.e., those without missing values, are then compiled into a list that includes the defining characteristics of the days. These characteristics are the weekday, day of the year, and the total energy. Using these characteristics, for every day with missing values, the dissimilarity to the complete days is calculated to determine the most similar day. Finally, from a derived power time series P, the values of the most similar days are pasted into the gaps of the incomplete time series.

Algorithm 1 Copy-Paste Imputation (CPI)

```
1:  procedure CPI(
        ets ▷ energy time series with missing values (i.e. NaNs)
    )
2:      ets ← SINGLE_VALUE_LINEAR_INTERPOLATION(ets)
3:      energy_per_day ← CALCULATE_ENERGY_PER_DAY(ets)

4:      ▷ each entry in this list describes whether a day of ets has missing values
5:      non_complete_days ← DETERMINE_DAYS_WITH_MISSING_VALUES(ets)

6:      ▷ only consider the daily energy consumption of the days without missing values
7:      weekly_pattern ← ESTIMATE_WEEKLY_PATTERN_WITH_PROPHET(
            energy_per_day, non_complete_days
        )

8:      missing_energy_per_day ← ESTIMATE_MISSING_ENERGY_PER_DAY(ets, weekly_pattern)
9:      estimated_energy_per_day ← energy_per_day+missing_energy_per_day
10:     complete_days ← COMPILE_LIST_OF_COMPLETE_DAYS(
            ets.time, energy_per_day, non_complete_days
        )

11:     pts ← DERIVE_POWER_TS_FROM_ENERGY_TS(ets)
12:     for all day ∈ days with missing values do
13:         best_matching_day ← FIND_DAY_WITH_MIN_DISSIMILARITY(day, complete_days)
14:         pts[day] ← pts[best_matching_day]

15:     for all gap in ets do
16:         scaling_factor ← actual_energy_of_gap / imputed_energy_of_gap
17:         pts[gap] ← pts[gap] · scaling_factor

18:     ets ← CALCULATE_ENERGY_TS_FROM_POWER_TS(pts)
19:     return ets without missing values, optionally power time series pts
```

3.2.2 Energy Consumption Estimation

The second step of the CPI method is the energy consumption estimation for days with gaps (Lines 3 to 9). The total energy consumption[2] during gaps can be determined because the CPI method uses an energy time series as input. In energy time series, the first entry after a gap still contains the information about the total energy consumed during the gap (as illustrated by the strictly increasing energy time series in Figure 3.1). For this reason, to obtain the total energy consumption E_i of the gap i from time step t to time step $t + k$, we calculate the energy difference, i.e.,

$$E_i = e_{t+k+1} - e_{t-1}, \tag{3.1}$$

[2]In the following, we refer to consumption data only, but the same principles apply to generation data.

where e_{t+k+1} and e_{t-1} are the energy consumption at the time steps $t+k+1$ and $t-1$ respectively. This equation is the basis of the estimation of the missing energy in Algorithm 1 of the CPI algorithm.

However, for gaps longer than one day, the calculated energy consumption must be allocated to the respective days appropriately. For this purpose, firstly, the calculated energy consumption of the gap is distributed to the respective days according to their share of missing values. Secondly, we consider a weekly pattern in the daily energy consumption. For this pattern, we use the weekly pattern of the input energy time series estimated by the Prophet method [49]. It models the weekly pattern such that the values of all weekdays add up to zero. If some of these values are positive, others need to be negative. For the estimation of these values, the Prophet method only considers the daily energy consumption of the days without missing values, i.e., one value per day. Lastly, the estimated weekly pattern is added to all days of the gap. When adding the weekly pattern, the added energy is summed up. The sum of the added energy is then divided by the number of days in the gap to obtain the average. This average is subtracted from each day of the gap to preserve the total energy consumption of the gap.

3.2.3 Compilation of Available Complete Days

In the third step of the CPI method, a list of the available complete days (i.e., days without missing values) is compiled (Line 10). Assuming daily patterns, a weekly cycle, and a yearly seasonality in the energy consumption, each day is listed with its following characteristics: its total energy consumption (d_e), its weekday ($d_w \in \{1\,..\,7\}$), and its seasonal position (d_s). Under the assumption of a yearly seasonality, i.e., 365 days or 366 days for leap years, it follows that d_s is in $\{1\,..\,366\}$.

3.2.4 Calculation of Dissimilarity Between Days

In the fourth step, the CPI method calculates a dissimilarity criterion between each day with gaps and all complete days (Line 13), which is used to select the best matching days for filling gaps in the next step. For the dissimilarity criterion, the CPI method uses the three previously introduced characteristics of days: total energy, weekday, and seasonal position. Since these characteristics are already computed for all complete days, they only have to be determined for the days with missing

values in this step. More specifically, three distance measures, i.e., D_e, D_w, and D_s, are calculated for each day with gaps d_i and each available complete day d_j.

The first distance measure D_e describes the distance between the total energy consumption of a day with gaps d_i and a complete day d_j. The total energy consumption can serve as a distance measure because the CPI method uses an energy time series as input and thus can calculate the energy consumed during a gap. D_e is defined as

$$D_e(d_i, d_j) = \frac{|d_{i,e} - d_{j,e}|}{e_{max} - e_{min}}, \tag{3.2}$$

where e_{max} and e_{min} are the maximum and minimum energy consumption of a day in the time series and $d_{i,e}$ and $d_{j,e}$ are the total energy consumption of the days d_i and d_j. For the day with gaps d_i, the previously estimated energy consumption is used. Dividing by the difference between e_{max} and e_{min} ensures that the distance measure D_e is in $[0, 1]$.

The second distance measure D_w is based on the assumption of a weekly pattern in the time series and describes the distance between the weekday of a day with gaps d_i and a complete day d_j. It is defined as

$$D_w(d_i, d_j) = \begin{cases} 0.0, & \text{if } d_{i,w} = d_{j,w} \\ 0.5, & \text{if } d_{i,w} \in \{1..5\} \wedge d_{j,w} \in \{1..5\} \\ & \vee d_{i,w} \in \{6, 7\} \wedge d_{j,w} \in \{6, 7\} \\ 1.0, & \text{else,} \end{cases} \tag{3.3}$$

where $d_{i,w}$ and $d_{j,w}$ are integer representations for the weekday of days d_i and d_j. One to five represent the workdays Monday to Friday, whereas 6 and 7 represent the weekend days Saturday and Sunday. This distance measure D_w assigns smaller distances to days of the same weekday or days of the same class (i.e., workday or weekend) and higher distances to days of different classes.

The third distance measure D_s captures the underlying seasonal patterns and describes the distance between the seasonal position of a day with gaps d_i and a complete day d_j. It is defined as

$$D_s(d_i, d_j) = \begin{cases} \frac{|d_{i,s} - d_{j,s}|}{\lfloor \frac{\mathbf{s}}{2} \rfloor}, & \text{if } |d_{i,y} - d_{j,y}| \leq \lfloor \frac{\mathbf{s}}{2} \rfloor \\ \frac{\mathbf{s} - |d_{i,s} - d_{j,s}|}{\lfloor \frac{\mathbf{s}}{2} \rfloor}, & \text{else,} \end{cases} \tag{3.4}$$

where $\mathbf{s}$ is the length of the seasonal cycle and $d_{i,s}$ and $d_{j,s}$ are the position of days d_i and d_j in this cycle. For a yearly seasonality, $\mathbf{s}$ can be set to 365 or 366 to reflect the number of days in a year. This distance measure ensures that two days from the

same season are considered as more similar than two days from different seasons. For example, January 1 and December 31 of the same year are almost one year apart but have a minimal distance D_s. In contrast, January 1 and July 1 are only half a year apart and have a maximal distance D_s.

In order to determine the dissimilarity between a day with gaps and a complete day, the three individual distance measures have to be combined into a single criterion. The resulting dissimilarity criterion D is the weighted sum of the three individual distance measures D_e, D_w, and D_s. It is defined as

$$D = w_e D_e + w_w D_w + w_s D_s, \tag{3.5}$$

where w_e, w_w, and w_s are the weights and D_e, D_w, and D_s are the normalized distance measures. The individual distance measures are normalized to the interval $[0, 1]$ for an easier interpretation of these weights. The specification of these weights is necessary once before applying the CPI method. To find suitable weights, a grid search can be conducted on a representative set of time series (see Section 3.3.3 for an exemplary grid search).

3.2.5 Copy and Paste of Matching Days

In the last step, the CPI method copies the best matching days, pastes them into gaps (Line 14), and scales the imputed values to preserve the energy of the respective gaps (Lines 15 to 17). In order to determine the best matching days, the previously generated list of complete days is used. For a day with gaps d_i, the day d_j with the smallest dissimilarity $D(d_i, d_j)$ is chosen. Since the entire list of complete days is used, days from the future of the day with gaps are also considered. In the process overview in Figure 3.1, the second and last day are identified as the most similar days to the two days with missing values.

Based on the determined best matching days, the actual copying and pasting of the best matching days into gaps is done. For this purpose, the power time series serves as basis. It can be derived from the input energy time series (Line 11) by calculating the average power p_t between time steps $t-1$ and t, i.e.,

$$p_t = \frac{e_t - e_{t-1}}{\Delta t}, \tag{3.6}$$

where Δt is the time between two time steps, e_t and p_t are the energy and power at time step t, and e_{t-1} is the energy at time step $t-1$. In the derived power time series, every missing value in each day with gaps is replaced by the corresponding

value of the previously determined best matching complete day, as illustrated in the last step in Figure 3.1.

Finally, the imputed power values are scaled in order to preserve the actual energy of each gap. The scaling is based on the actual energy and the imputed energy. Both can be determined because the CPI method uses energy time series as input and thus can calculate the energy consumed during a gap. The actual energy E_i of the gap i is calculated according to Equation (3.1). The imputed energy E_i' is calculated by accumulating the imputed power values. To preserve the energy, the imputed power values of gap i are multiplied with the ratio of the actual energy and the imputed energy, i.e.,

$$\hat{p}_t = \hat{p}_t' \cdot \frac{E_i}{E_i'}, \tag{3.7}$$

where $\hat{p}_t'$ is the power value calculated by the CPI method and $\hat{p}_t$ is the scaled power value. After this scaling, the imputed power time series can be used to calculate an imputed energy time series by solving Equation (3.6) for e_t. With the calculated energy time series, the CPI method finally returns a complete energy time series whose initially missing values are completely imputed.

3.3 Evaluation

In this section, the proposed CPI method is evaluated on real-world data and its imputation accuracy and performance are compared to benchmark methods. Therefore, the used dataset is introduced followed by the selected benchmark methods. After describing the experimental setting, the results are presented.

3.3.1 Dataset

The dataset used for the evaluation is the *ElectricityLoadDiagrams20112014* dataset[3] from the UCI Machine Learning Repository [61]. The dataset consists of power time series with complete consumption data from 370 different smart meters over a period of up to four years. The time series contain quarter-hourly average power values in kW, resulting in $35\,040$ values per year. Of these 370 time series, 50 differently shaped time series with a length of one year are selected as a representative sample. The selected time series vary greatly in terms of seasonal, weekly, and daily patterns as illustrated in Figure 3.2.

[3] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`

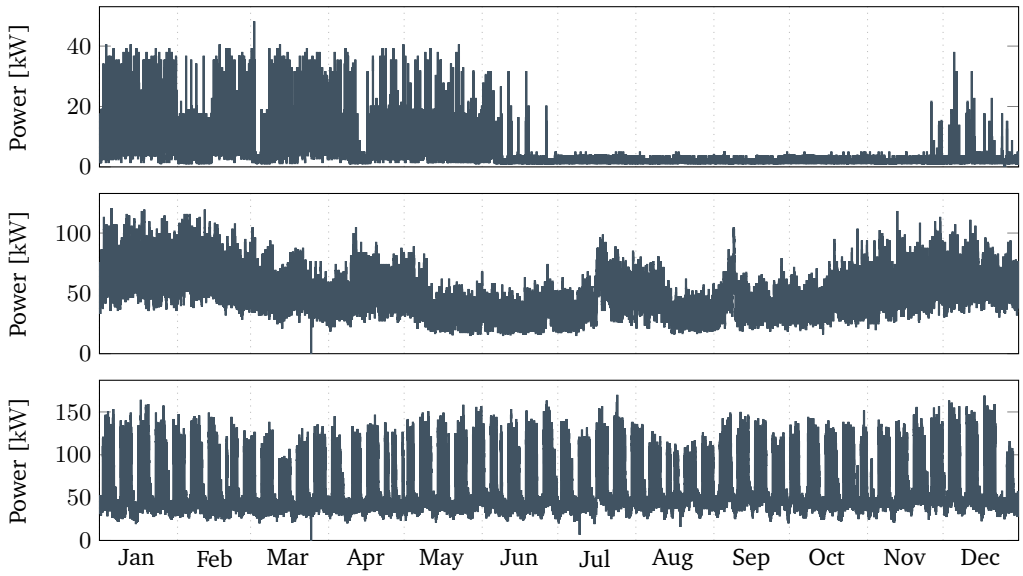


Fig. 3.2.: Three exemplary time series from the UCI dataset, containing variations of different seasonal, weekly, and daily patterns.

For the evaluation of the CPI method, those selected power time series, which do not contain any missing values, are converted to energy time series by accumulating the power values. Due to their completeness, we artificially insert missing values in the used time series by replacing values with NaNs. In order to decide which values are replaced, we follow four steps. Firstly, we determine the longest sequence of the time series without missing values T_c. Secondly, we define the number of consecutive values to be replaced in this sequence T_c by choosing uniformly between 2 and the minimum of the specified maximum number of consecutive missing values, the length of T_c, and the remaining number of values to be replaced. Thirdly, we randomly select a starting index for the determined number of consecutive values to be replaced such that all values to be replaced are contained in T_c. Lastly, we replace each selected value with NaN. These four steps are repeated until the total number of values to be replaced is reached.

For the evaluation, we consider the number of values to be replaced in the form of shares of missing values for each time series. Six shares of missing values are used, i.e., $1\,\%$, $2\,\%$, $5\,\%$, $10\,\%$, $20\,\%$ and $30\,\%$. In order to consider both larger gaps and single missing values, $5\,\%$ of each share of missing values are single missing values. The indices for the single missing values are determined randomly after creating the larger gaps.

3.3.2 Benchmark Methods

In order to compare the performance of the proposed CPI method, we apply benchmark methods to the dataset. As suitable benchmark methods, we generally consider all imputation methods for energy time series that utilize the time series and its characteristics only. Due to the lack of imputation methods for energy time series—to the best of our knowledge—, we include imputation methods for power time series and time series in general despite their disadvantage of not utilizing energy data. Methods requiring additional data or information such as weather data [47, 51] or validated reference days [57] and methods designed for multivariate time series only [52, 58, 60] are discarded due to their lack of comparability. Furthermore, for the evaluation, the method in [53] is excluded due to its excessive run-time.

In this context, we select three methods as benchmarks in view of comparison complexity and fairness. We derive these methods from literature [49, 50, 54, 56] and adapt them where necessary. To establish a fair comparison, the evaluated benchmark methods receive their data input in the same way as the CPI method. They sequentially get the 50 time series and can use each time series completely but independently of the others.

The first benchmark method is a commonly applied linear interpolation [50, 54]. This method represents a lower baseline and should be outperformed in any case. It imputes missing values $\hat{p}_t$ by linearly interpolating the first and last known power value before and after a gap, i.e.,

$$\hat{p}_t = \frac{t - t_1}{t_2 - t_1} \cdot (p_{t_2} - p_{t_1}) + p_{t_1}, \tag{3.8}$$

where t_1 and t_2 are the time steps before and after the gap. The linear interpolation is thus the only evaluated method that uses two values for imputing a gap.

The second benchmark method is the optimally weighted average (OWA) [54]. Assuming a weekly pattern, this method calculates a historical average

$$\hat{p}_t^{\mathrm{HA}} = \frac{1}{|H|} \Sigma_{i \in H} p_i, \tag{3.9}$$

where H contains all values of the hour before and after t as well as of the same two hours of the previous and of the next week. As long as H is empty, the considered weeks are iteratively extended by one in each direction to consider additional values from the same two hours in more weeks. To ensure smooth transitions between

actual and imputed values, this average is combined with a linear interpolation $\hat{p}_t^{\mathrm{LI}}$ (3.8). The combination results in

$$\hat{p}_t = w_t \hat{p}_t^{\mathrm{LI}} + (1 - w_t)\hat{p}_t^{\mathrm{HA}}, \tag{3.10}$$

where w_t weighs the influence of the two imputation methods. The weight w_t is designed to decrease with increasing distance to the actual values, i.e.,

$$w_t = e^{-\alpha d_t}, \tag{3.11}$$

where d_t describes the distance from t to the nearest actual value in time steps and α determines the rate of decay for w_t. Since α has a negligible influence on the imputation results in the present evaluation, we use a global $\alpha = 0.1387$ for the evaluation as determined in [54].

The third benchmark method is based on the Prophet method for time series forecasting [49]. Prophet uses a modular regression model that can be described as

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t, \tag{3.12}$$

where g is a model for the trend, s for seasonality, h for holidays, and ε_t for changes that are not represented in the model. The imputation method based on this model exploits Prophet's capability to estimate a time series model on irregularly spaced data [49] and imputes missing values with the corresponding values of the model. The model is learned on all values available in the time series to be imputed. In contrast to its application in the CPI method, the benchmark imputation method based on Prophet receives, like all other benchmark methods, the aforementioned quarter-hourly values as input, i.e., 96 values per day.

3.3.3 Experimental Setting

This subsection describes the used hard- and software platform, introduces the error measures to evaluate the imputation methods, and explains the calculation of the weights in the dissimilarity measure of the evaluated CPI method.

Hard- and Software Platform For the evaluation of the CPI and the benchmark methods, we compare the quality of the imputation and the required run-time. For a better comparability of the results, all methods are implemented in Python and

evaluated on the same hardware. The evaluation hardware is a desktop PC running Ubuntu 20.04 with an AMD Ryzen 5 3600 processor and $16\,\mathrm{GB}$ of memory.

Error Measures In order to evaluate the quality of an imputation in energy time series, we examine both the usage of matching patterns to fill gaps and the conservation of the total energy in the gaps. To evaluate the usage of matching patterns, we determine how well imputed patterns match the actual patterns. For this purpose, we measure the deviation between every single actual power value and the corresponding imputed power value using the MAPE as defined in Equation (2.5), only comparing the imputed indices.

To evaluate the conservation of the total energy in gaps, we measure the difference between the actual and imputed energy while ignoring the fine granular patterns that are used for the imputation. The difference is determined using the weighted absolute percentage error (WAPE), which is defined as

$$\mathrm{WAPE} = \frac{\sum_{i=1}^{N} |\hat{E}_i - E_i|}{\sum_{i=1}^{N} E_i} \cdot 100\%, \tag{3.13}$$

where E_i and $\hat{E}_i$ are the actual and imputed energy of gap i in a time series with N gaps. In contrast to the MAPE_p (2.3), the weighting of the individual absolute errors is necessary in the WAPE_E (3.13) to account for gaps of different sizes.

Weights in the CPI Dissimilarity Measure Before copying and pasting matching days, the CPI method calculates the dissimilarity measure between two days (3.5). For this calculation, it is necessary that the weights of the three distance measures regarding total energy consumption, weekday, and seasonal position are specified beforehand. To determine these weights for the CPI method applied in the evaluation, we thus perform a grid search.

This grid search is conducted before the actual evaluation on a separate dataset that is used only for calibration. To compile the calibration dataset, we consider the remaining 320 time series given in the aforementioned dataset. Based on a visual inspection, we choose five time series for the calibration dataset, which are different from each other but have similar characteristics to the 50 time series in the evaluation. Each of the five time series from the calibration dataset is evaluated with six different shares of missing values, ranging from $1\,\%$ to $30\,\%$, which results in 30 time series in total.

Using this calibration dataset, we test 1000 combinations of the three weights $w = (w_e, w_w, w_s)$ with each weight in the range $[1\,..\,10]$ and 300 combinations where one of the weights is set to zero. Each combination of weights is evaluated with both error measures MAPE_p (2.3) and WAPE_E (3.13) on all time series of the calibration dataset. Based on the results, we determine the overall minimum and maximum of both error measures. With them, we min-max normalize the values of both error measures. The min-max normalized MAPE_p is calculated with

$$\mathrm{MAPE}_{p,n}(w,i) = \frac{\mathrm{MAPE}_p(w,i) - \min \mathrm{MAPE}_p}{\max \mathrm{MAPE}_p - \min \mathrm{MAPE}_p}, \tag{3.14}$$

where w is the tested weight combination and i is the identifier of the time series from the calibration dataset. The min-max normalized WAPE_E is determined analogously to (3.14). To obtain the average sum of both normalized error measures $\overline{TE}$, we first add the normalized results of both error measures together for each time series from the calibration dataset. Afterward, we sum this total over all time series and divide it by the number of time series, i.e.,

$$\overline{TE}(w) = \frac{\sum_i^n \mathrm{MAPE}_{p,n}(w,i) + \mathrm{WAPE}_{E,n}(w,i)}{n}, \tag{3.15}$$

where n is the number of time series in the calibration dataset. This approach assumes an equal importance for both error measures. Finally, we determine the weights with the minimum average sum of both normalized error measures, i.e.,

$$w_{opt} = \arg\min_w \overline{TE}(w), \tag{3.16}$$

representing the optimal weights w_{opt} out of the 1300 tested combinations.

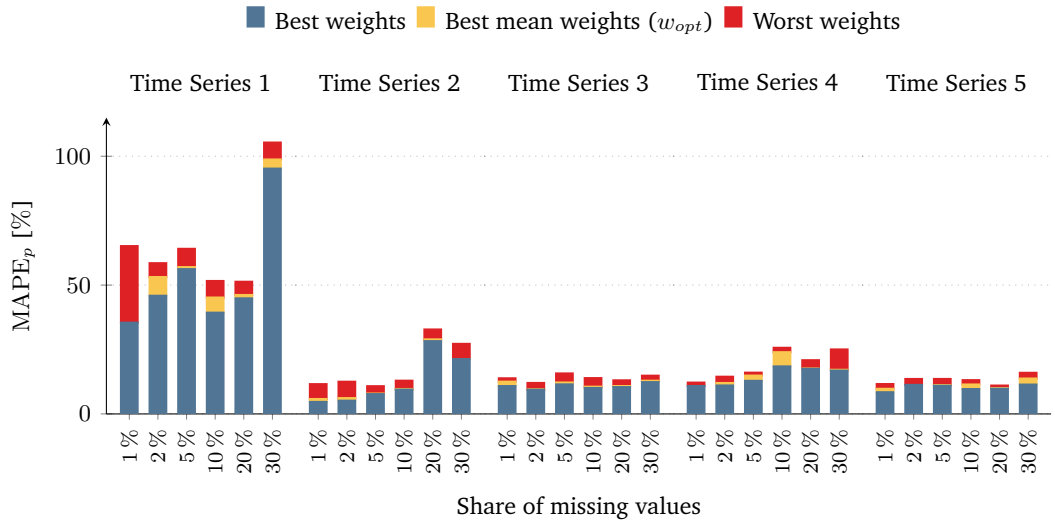
3.3.4 Results

Based on the 50 selected time series, the selected benchmark methods, and the experimental setup described above, this section presents the evaluation results. It first covers the grid search, before describing the usage of matching patterns and the conservation of energy, quantified by MAPE_p (2.3) and WAPE_E (3.13) respectively. The presented values of these error measures are the truncated means for the 50 evaluated time series, which omit the two best and worst values to obtain less outlier-sensitive results. Afterward, the run-time of the evaluated methods and its decomposition is addressed before the usage of matching patterns and the run-time are put in comparison. Lastly, the output of an exemplary imputation visually illustrates the evaluation results.

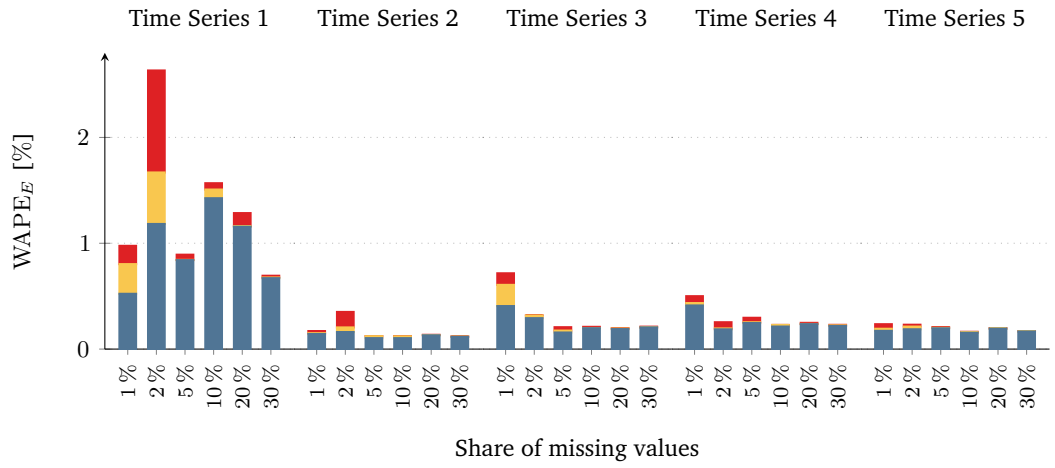
Grid Search The grid search evaluates the aforementioned 1300 weight combinations in $91\,\mathrm{min}$ with a total system memory usage of approximately $3.5\,\mathrm{GB}$. Figure 3.3 shows the MAPE_p (2.3) and WAPE_E (3.13) for the tested time series. For each time series and each share of missing values, an orange bar depicts the result with the best weights. Accordingly, the red bars show the worst results. The yellow bars describe the results with the overall best weights $w_{opt} = (5, 1, 10)$ determined with Equation (3.16). The grid search reveals that every time series has its own optimal weight combination. Nevertheless, the difference between the results with the best and the worst weights is often very small. Similarly, the difference between the results with the best weights and the overall best weights w_{opt} is often negligible. w_{opt} is, therefore, used for all 50 time series to be evaluated.

Usage of Matching Patterns The usage of matching patterns, as defined in Equation (2.3), by the evaluated methods is presented in Figure 3.4. For the six shares of artificially inserted missing values, the figure shows the MAPE_p of all evaluated methods. For most of the shares of missing values, the CPI method performs better than the OWA method as the best benchmark method. Both methods perform overall about $10\,\%$ to $12\,\%$ better than the Prophet-based method. The linear interpolation performs by far the worst for all shares of missing values. All methods tend to higher errors with higher shares of missing values. This trend is most distinct for the linear interpolation. With regard to the errors of individual time series, the benchmark methods are more prone to extreme errors with a maximum MAPE_p of $587.8\,\%$ and above while the CPI method has a maximum MAPE_p of $237.4\,\%$.

Conservation of Energy For the four evaluated methods, the conservation of energy in the gaps, as defined in Equation (3.13), is shown in Figure 3.5. The figure presents the WAPE_E for the six shares of missing values. The CPI method performs best regardless of the share of missing values. To allow a better comparability with the benchmark methods that all do not use scaling, the dashed line indicates the error values for the CPI method without scaling. Without scaling, the CPI method performs on average $4.4\,\%$ better than the OWA method, which is the second best method. We assume that this result is strongly related to using energy as a distance measure, which is possible thanks to the use of energy time series as input. The scaling even reduces the error to nearly zero for all shares of missing values, so the CPI method performs even better. In view of the already good results without scaling, the contribution of scaling is, however, relatively small. The linear interpolation again performs worst for all shares of missing values. The Prophet-based method and the OWA method perform very similarly with an average advantage of $1.6\,\%$ for the OWA method.



(a) MAPE$_p$



(b) WAPE$_E$

Fig. 3.3.: The figures show the two error measures (MAPE and WAPE) of the best, worst, and weights with the best mean error for the five time series from the calibration dataset used in the grid search. Every time series is evaluated with six different shares of missing values ranging from 1 % to 30 %, resulting in a total number of 30 tested time series. The lowest, blue-gray, bars indicate the errors of the best weights, the red bars the errors of the worst weights, and the yellow bars the errors resulting from the determined optimal weight combination w_{opt}. The bars in this chart are not stacked vertically, but rather layered from background to foreground.

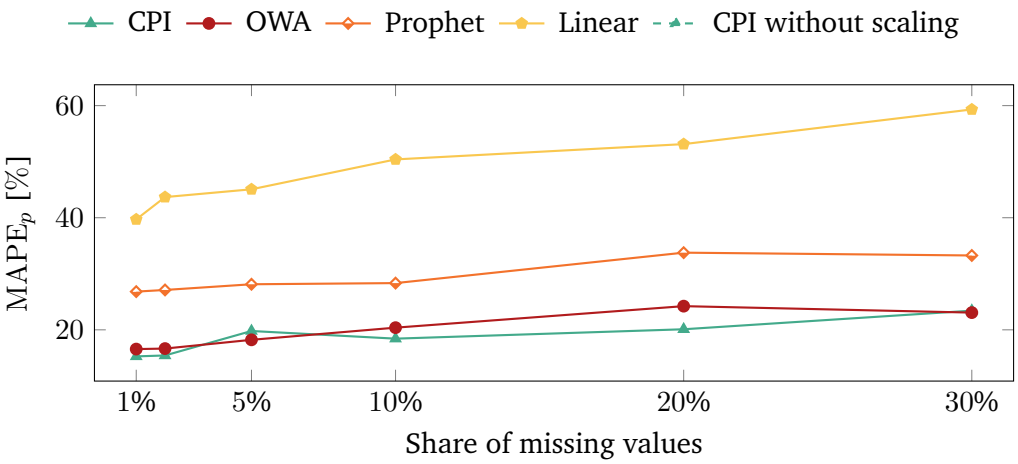


Fig. 3.4.: The MAPE_p of the CPI method and the three benchmark methods with different shares of missing values. As the scaling of imputed values does not noticeably affect the results of the CPI method, it is omitted in this figure.

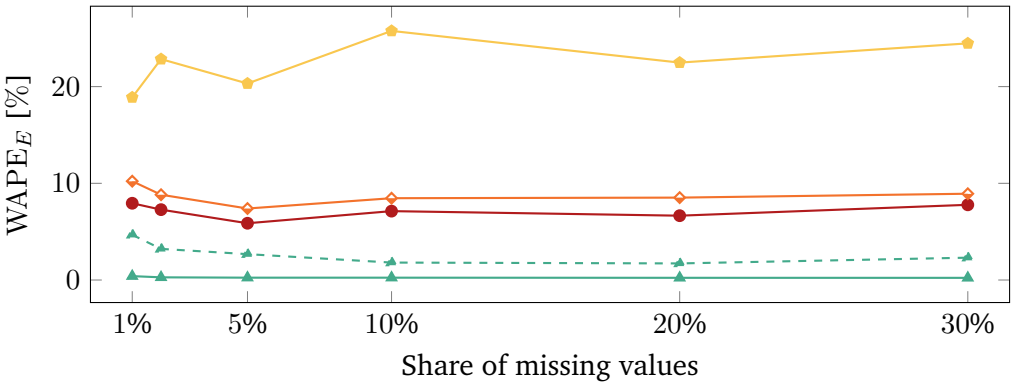


Fig. 3.5.: The WAPE_E of the CPI method and the three benchmark methods with different shares of missing values. For better comparability with the benchmark methods that all do not use scaling, the dashed line indicates the WAPE_E of the CPI method without scaling the imputed values to preserve the energy of a gap.

Run-time Figure 3.6 shows the average run-times required by the evaluated methods for the imputation of the 50 selected one-year time series with $35\,040$ values each. The linear interpolation is by far the fastest method. The OWA method is similarly fast for small shares of missing values but increases more drastically in run-time than the other methods for increasing shares of missing values. The CPI method requires about 10 to 20 times more time than the linear interpolation but is faster than the OWA method for $20\,\%$ and $30\,\%$ of missing values. The Prophet-based method requires much more time than all other methods and is 9 to 10 times slower than the CPI method. Its longer run-time compared to the CPI method, wherein the Prophet method is also used, is caused by a more time-consuming training due to its larger input. When applied as benchmark method, the Prophet method receives 96

values per day as input as opposed to only one value per day when used in the CPI method.

In addition to the run-time evaluation for the one-year time series, we briefly evaluate how the CPI method's run-time relates to the number of input values. Figure 3.7 shows the run-time of the CPI method for time series with different lengths from one quarter of a year (8832 values) up to three years (105 120 values). The CPI method scales approximately linearly to the number of input values with an average run-time of 5.56 s for time series with 105 120 values.

Run-time Decomposition Figure 3.8 decomposes the run-time of the evaluated methods for 1 % and 30 % of missing values. Model estimation including training and fitting are depicted in purple whereas the actual imputation is depicted in blue. Figure 3.8 shows that the energy consumption estimation for gaps, that relies on

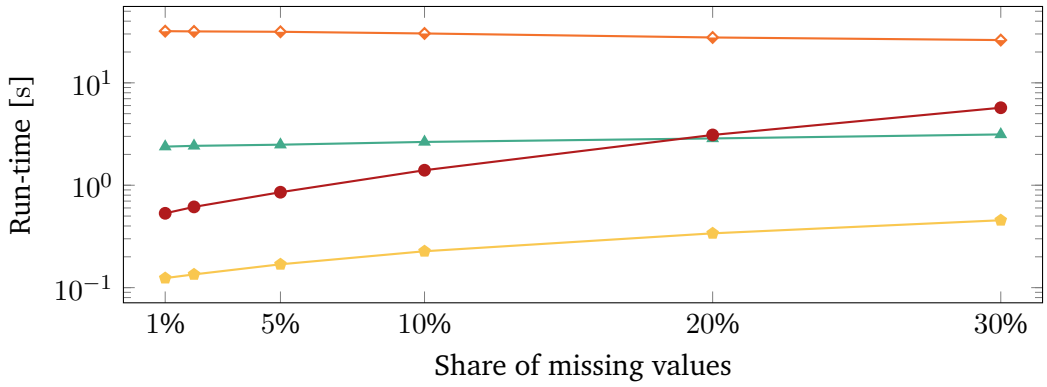


Fig. 3.6.: The average run-times required by the CPI method and the three benchmark methods for the imputation of the 50 selected one-year time series. Note the logarithmic timescale, which visually compresses Prophet's run-time decrease by 5.8 s from 1 % to 30 % of missing values.

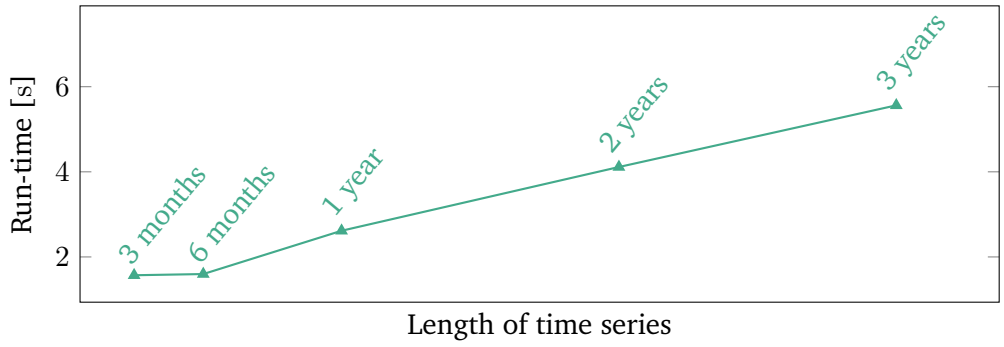


Fig. 3.7.: Run-time of the CPI method for time series with different lengths from three months (8832 values) up to three years (105 120 values).

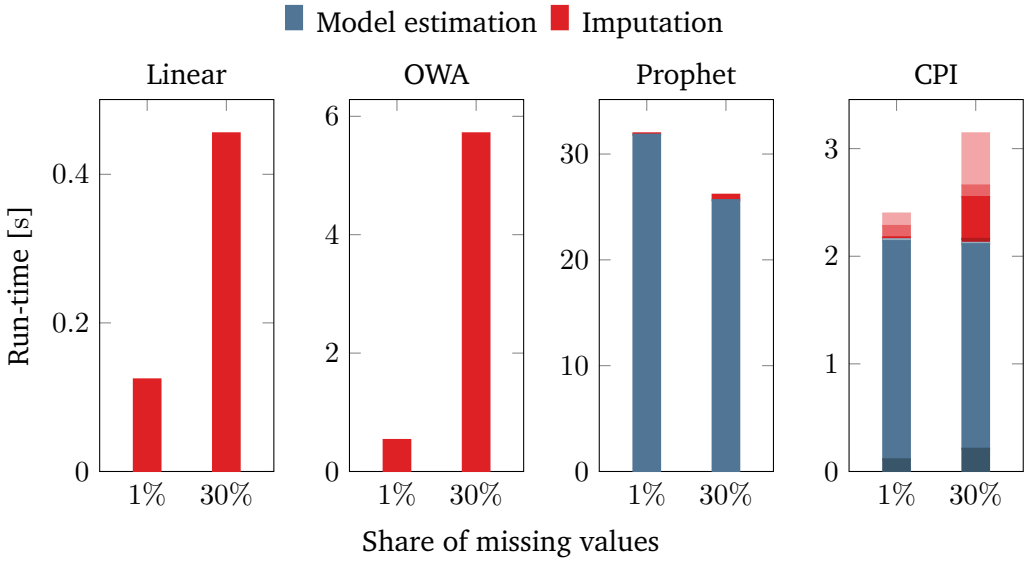


Fig. 3.8.: Run-time decomposition of the CPI method and the three benchmark methods into model estimation including training and fitting as well as imputation. For the CPI method, the run-time is even further differentiated with respect to its steps. From bottom to top, the blue colors describe the linear interpolation, the energy consumption estimation, and the compilation of available complete days. The red color shades indicate the matching of the most similar days, pasting the values into the gaps, scaling the imputed values, and calculating the completed energy time series. The individual times of the steps are stacked to make up the total run-time.

the Prophet method, is the dominating part of the CPI method's run-time. Similarly, the model estimation also dominates the run-time of the Prophet-based method. In contrast, the linear interpolation and the OWA method do not comprise any model estimation. Their run-time thus consists entirely of the imputation itself.

Usage of Matching Patterns vs. Run-time In Figure 3.9, the obtained results regarding the usage of matching patterns are put in relation to the run-time needed with a scatter plot showing the required average run-times on the x-axis and the MAPE_p on the y-axis. Smaller values indicate a better performance. While the linear interpolation provides fast and inaccurate results, the CPI method and the OWA method deliver the most accurate results with a reasonable run-time. The Prophet-based method yields mediocre results while taking much longer to calculate than the other methods.

Exemplary Imputation Results For all evaluated imputation methods, Figure 3.10 illustrates an exemplary imputation of a time series with $20\,\%$ of artificially inserted missing values, resulting in large gaps. The imputation of the linear interpolation

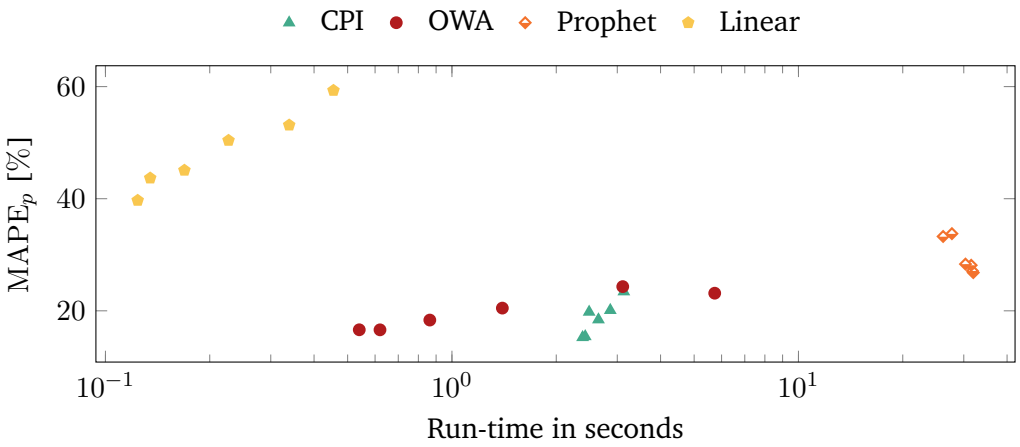


Fig. 3.9.: Comparison of the usage of matching patterns and the average run-time needed of the CPI method and the three benchmark methods for the imputation of 50 one-year time series. The x-axis shows the required average run-times on a logarithmic scale and the y-axis the MAPE_p.

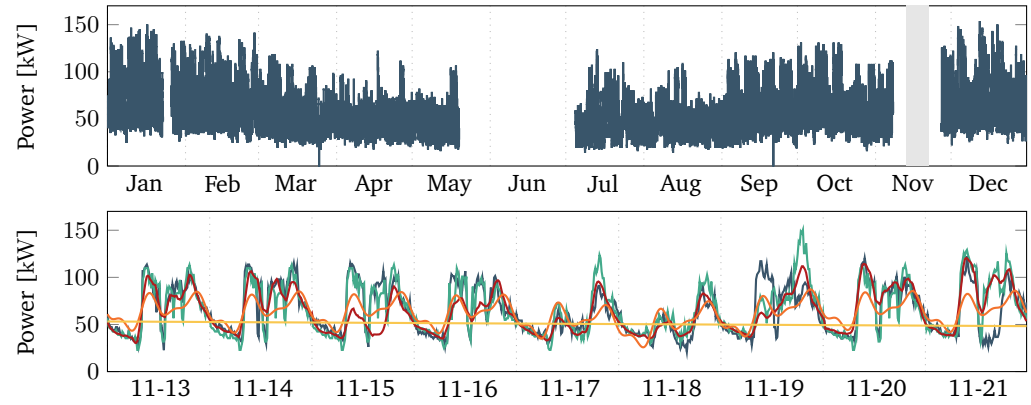


Fig. 3.10.: The upper figure shows an exemplary one-year time series with 20 % of missing values. For a multi-day excerpt of a gap in November, the lower figure presents the resulting imputations by the CPI method and the three benchmark methods in comparison to the actual values. Metrics of the CPI method and the best benchmark method for this example: MAPE_p: 19.4 % (OWA: 21.1 %), WAPE_E: 0.3 % (OWA: 6.5 %), run-time: 2.89 s (Linear: 0.35 s).

fails to capture the patterns of the time series. The imputations by the OWA method and the Prophet-based method capture the essential patterns but lack details. The imputation by the CPI method mostly fits the actual values, but it shifts and increases some peaks. Despite not explicitly addressing the transitions between existing and imputed values, the CPI method also provides smooth transitions on both ends of gaps in this case. Compared to the three benchmark methods, the imputation by the novel CPI method thus comes closest to the actual values.

3.4 Contribution and Future Work

In the present chapter we answer research question **RQ1**, addressing the handling of missing values in energy time series. In order to answer this question, we evaluate imputation methods found in the literature using a real-world dataset. We compare these methods to our proposed CPI method with a quantitative evaluation of the imputation results based on the insertion of matching patterns into gaps and the conservation of energy throughout the time series. Finally, we log the runtimes and include exemplary imputation results for visual comparisons.

In conclusion, this chapter provides the following contributions:

- We introduce a new copy-paste imputation (CPI) method for energy time series. To the best of our knowledge, it is the first method that specifically uses an energy time series as input data. Using this data, it copies blocks of data with similar characteristics and pastes them into the gaps of the time series.
- We demonstrate that utilizing the information inherent in energy time series enables a robust selection of matching blocks of data and ensures that the overall energy per gap remains unchanged while imputing the missing values with realistic patterns.
- We compare the proposed CPI method to three benchmark methods and confirm that it outperforms these methods, using matching patterns for the imputations and conserving the overall energy of every imputed gap. This is achieved with a moderate run-time that scales well with increasing shares of missing values and length of input time series.
- We conclude that the introduced CPI method enables realistic imputations even for large gaps with several weeks of consecutively missing values.

Based on these results, future work could follow three directions. First, the robustness of the CPI method could be analyzed and improved regarding aperiodic events or time series with other temporal resolutions and periodicities such as residential solar power generation or fast charging of electrical vehicles. Similarly, time series containing both power consumption and generation from renewable energy sources could be of interest for further investigation. A robustness analysis could further include the selection of the weights in the dissimilarity measure—for example in dynamic environments—and the compilation of the calibration data set used for determining the weights. Furthermore, the transitions between existing and imputed values on both ends of gaps could be further investigated. Second, a trend

analysis could enhance the CPI method's selection of matching days especially for longer gaps. Similarly, additional information such as voltage magnitude [62] and spatial temporal correlations [63] could be used to improve the matching days' selection. Third, the CPI method could be integrated in applications that rely on complete input data such as grid simulation, load forecasting, and load management. Anomaly or error detection functions could also be included in the CPI method itself to repair implausible values. Moreover, a reporting and analysis tool could use the CPI method to estimate the imputation quality based on artificially inserted missing values.

4 Automated Distribution Grid Modeling Using Open Data

Parts of this chapter are reproduced from

Moritz Weber, Luc Janecke, Hüseyin K. Çakmak, and Veit Hagenmeyer. "Open Data-Driven Automation of Residential Distribution Grid Modeling with Minimal Data Requirements". In: *IEEE Transactions on Smart Grid* 15.6 (Nov. 2024), pp. 5721–5732. DOI: 10.1109/TSG.2024.3406765.

Distribution grids face several challenges due to the transition from traditional fossil-fueled to carbon-neutral energy generation. This transition leads to more generation from renewable energy sources, such as wind farms and photovoltaic systems, in the electricity grid [64, 65]. These renewable energy sources are often distributed throughout the grid, instead of the traditionally centralized energy generation by fossil fuels or nuclear power. As such, these decentralized energy sources are typically located in the lower grid levels, which leads to an increased generation in distribution grids, especially through rooftop photovoltaics. As described in Chapter 3, measurements of load and generation, can help to gain insights into the grid. However, they are not sufficient to solve the identified challenges as they only portray current and past states of the grid. Solving the challenges distribution grids are facing, requires examining potential future scenarios, identifying problems and evaluating solutions. These tasks require realistic simulations of future grid situations. However, the required grid data and models are often not readily available due to data privacy concerns, necessitating labor-intensive modeling processes before the actual simulations can be performed.

Thus, the goal of this chapter is to further develop methods for the automated creation of grid models, building on our previous work [66], using readily available open data sources. The generated models aim to be realistic enough that they could describe the real grid, while not necessarily creating exact models of existing grids. These grid models are intended to support various use cases, e.g., machine learning applications and realistic simulations on a wide range of simulation software, such

as load flow calculations, time series-based long-term simulations, RMS and EMT simulations.

The main contribution of this chapter is answering research question **RQ2** which addresses the need for realistic distribution grid models, ideally with minimal data requirements. To this extent, we introduce a new two-stage optimization method for the automated generation of LV grids relying solely on openly available data sources, such as OSM [67] and OSM Buildings [68] that further enables the automated model generation of the $20\,\mathrm{kV}$ MV grid. This significantly reduces the data requirements compared to previous approaches [66, 10, 11, 13, 16, 69, 70, 71, 72, 73, 74], which are described in more detail in Section 4.1. In order to assess the accuracy of the generated models and answer research question **RQ3**, we present a comprehensive comparison of various data sources—open and proprietary—for the automated generation of distribution grid models. This comparison considers the topological, geographical, and electrical properties of the generated grid models.

The remainder of this chapter is structured as follows: Section 4.1 describes existing works on data-driven model generation. In Section 4.2, we describe the methodology behind our proposed model generation method, including the building load estimation with various data sources, and our optimization-based transformer placement method. We then evaluate the proposed method and compare the results using the different data sources in Section 4.3 before discussing the results in Section 4.5. Finally, we conclude with our findings and give a brief outlook on potential directions for further development in Section 4.6.

4.1 State of the Art

Since the demand for distribution grid models is apparent for many use cases centering around grid simulations and studies, the data-driven generation of such models is a well-researched topic.

The process for creating so-called reference network models (RNMs) described in [69] marks a fundamental work in the field of automated power grid modeling. This approach utilizes data on the location and demand of customers, locations and capacity of distributed generation (DG) and transmission substations, and economic and technical parameters to generate European-style power grid models from the high-voltage (HV) level down to individual LV customers. The grid models are created by a heuristic branch-exchange method to minimize the cost of the grid using a pre-defined catalog of standard equipment. Building on this, [16] creates

and publishes several models of selected test areas in the MATPOWER[1] format and compares key indicators of these models with real-world data provided by European DSOs. A further development of this process is described in [10], which focuses on the development of an online tool for the automated model generation from OSM data, area parameters (e.g., consumer density, power factor, and MV/LV transformer locations), and DSO indicators.

More enhancements to [69] are described in [13, 70, 71] with the adaption to US-style distribution grids and an appropriate validation approach for this grid type. The approach introduced in [13] is capable of creating complex US-style power grids with their typical single phase connections and voltage regulators on the LV level. While the previously mentioned methods expect the location and demand of customers as a direct input, [13] describes a method for estimating this demand based on land use data (usually from commercial vendors) and a library of reference buildings. Considering the complex nature of US-style distribution grids, this approach creates detailed models in the OpenDSS[1] and CYME format.

Other approaches, such as [11] focus on the German power grid. This approach utilizes OSM data and the known total number of LV networks in Germany to generate a total of $500\,000$ LV distribution grid topologies. The generated topologies are validated on a statistical basis with real grid data, such as the number of nodes and edges per LV grid and the total line length. Other approaches in the same research context focus on the generation of transmission grid [72] and MV grid [73] models instead of the lower voltage levels. The method described in [74] is focused on regional distribution grid structures as found in Germany and utilizes nine different data sources to classify buildings for load estimation and for generating the MV and LV grid topologies. The method generates MATPOWER models and is validated on a high-level basis against real networks.

Despite all of these approaches, the research in this domain is still open in a few key areas: Most of the methods available in the literature utilize some kind of proprietary data or very specific knowledge that prohibits a wider applicability. Furthermore, to the best of our knowledge, there has not been a comparison of models generated using different available data sources. Lastly, many available methods create models in very basic formats, such as MATPOWER. In this thesis, we address those three areas by presenting a method that requires minimal data input and generates versatile DIgSILENT PowerFactory[1] models. Furthermore, we compare the generated models under different input data circumstances.

[1] See Section 2.2 for information on common modeling and simulation tools.

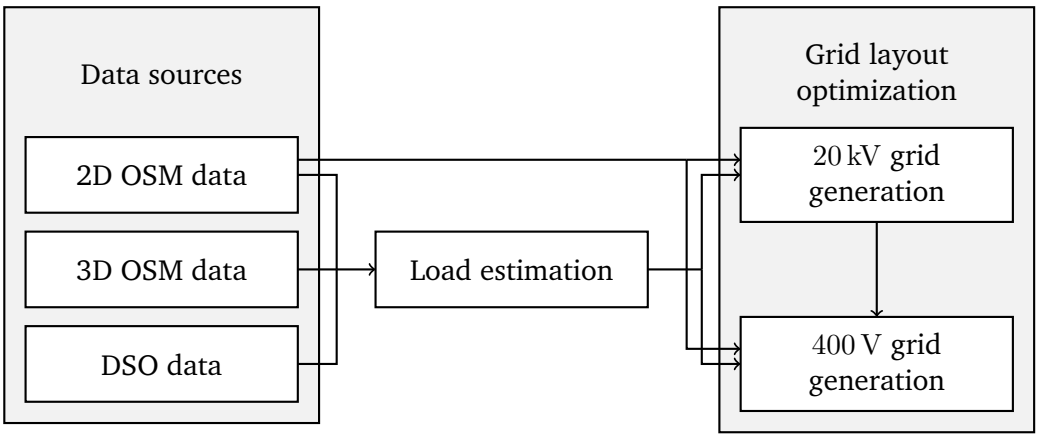


Fig. 4.1.: The two-stage optimization method for the automated distribution grid generation comprises the 20 kV and 400 V grid generation. Both stages rely on a prior load estimation that can be based on various alternative data sources.

4.2 Method: Two-Stage Optimization-based Generation of Residential Distribution Grid Models

In this section, we first describe the building load estimation for low-voltage grids before describing the two optimization stages of the grid model generation as shown in Figure 4.1. The load estimation process uses three open and proprietary data sources to estimate the load per building that is a required input of the two-stage grid layout optimization. These data sources are OSM data ($\mathbf{O_{2D}}$), 3D OSM Buildings data ($\mathbf{O_{3D}}$) and finally—provided by a local DSO—the number and location of electricity meters ($\mathbf{EM}$). In the first stage of this optimization, the 20 kV grid topology with its 20/0.4 kV substations is generated using a k-means clustering approach for the substation placement and a travelling salesman problem (TSP) optimization for the line routing between the stations. In the second stage, the underlying 400 V grid is generated by solving a variation of the minimum cost flow linear optimization problem. Since the methodology described in this thesis focuses on areas dominated by underground cables instead of overhead lines, whose routes might be available in map data, we make the common assumption that cables are laid out along roads and paths.

4.2.1 Building Load Estimation Based on Variable Data Sources

Most existing distribution grids evolved over many years and traditionally only loads were considered in the planning of low-voltage distribution grids and the

dimensioning of equipment. Thus, we disregard PV generation in our methodology for estimating the required capacities of equipment, such as transformers and cables. The impact of renewables in the low-voltage grid is rather part of studies that can be carried out with the generated grid models, see e.g., [75]. Since the network topology is generated using an optimization algorithm based on load data, the building load estimation has significant influence on the resulting network topology. The load estimation methods described in the following are, however, used for residential buildings only. Non-residential buildings are identified via tags included in OSM data and require a special treatment, for example, as described in Section 4.3.1.

In order to estimate the load of a residential building $b \in \mathcal{B}$, we utilize the household standard load profile $\mathrm{H0}$ [76] that is multiplied by an estimated yearly energy consumption E_b:

$$P_b(t) = \mathrm{H0}(t) \cdot \frac{E_b}{\sum \mathrm{H0}(t)}. \tag{4.1}$$

The estimation of the yearly consumption of a building starts with estimating the energy consumption for a single residential unit E_U,

$$E_U(A) = nR \cdot E_R + A \cdot E_A + nLA \cdot E_{LA}, \tag{4.2}$$

where nR is the average the number of residents in a household according to [77], A is the floor area of the residential unit, and nLA is the statistical number of large electrical appliances per household for the selected test area [78]. E_R and E_{LA} are the corresponding energy demands per resident and large appliance, respectively. Specific values used for the application of this method can be found in Table 4.1. With this formula for the energy demand estimation of a single residential unit, the next step is the estimation of the number and floor area of residential units in a building. These calculations differ depending on the utilized data source.

OSM data to determine the base area of buildings The OSM data source, $\mathbf{O_{2D}}$, includes the buildings shapes, which are used to calculate the floor area A_b of each building b. If only this floor area is considered, the number of residential units in a building nU_b^{2D}, is estimated as

$$nU_b^{2D} = \begin{cases} A_b/A_m, & \text{if } A_b/A_m \leq 1 \\ 1, & \text{else,} \end{cases} \tag{4.3}$$

where A_m is the median floor area of a residential unit. This median value A_m is determined over all floor areas of the buildings $\mathcal{B}$ in the target region which geographically defines an area:

$$A_m = \text{median}\left\{A_b : b \in \mathcal{B}\right\}. \tag{4.4}$$

To model the energy consumption of individual buildings E_b, which is based on the consumption per unit E_U, the base areas of the buildings are decisive and may vary greatly depending on the data basis.

Using the building floor area, the energy consumption per building E_b^{2D} is obtained by multiplying E_U with the number of residential units nU_b^{2D}, see Equation (4.3), and a scaling factor S_U,

$$E_b^{2D} = nU_b^{2D} \cdot S_U \cdot E_U(A_m). \tag{4.5}$$

The scaling factor S_U can be approximated by the average number of stories per building and can be adjusted downward to accommodate for a larger proportion of single-family homes.

OSM data and height information from OSM Buildings In this estimation, the building height H_b is considered to approximate the number of floors of a building using data provided by OSM Buildings [68], i.e., data source $\mathbf{O_{3D}}$. According to [79], the typical floor height in Baden-Württemberg is $2.5\,\mathrm{m}$ to $3\,\mathrm{m}$. In this range, an average floor height h_f is chosen for the further calculation. The corresponding energy consumption E_b^{3D} is calculated as

$$E_b^{3D} = nU_b^{2D} \cdot H_b/h_f \cdot E_U(A_m), \tag{4.6}$$

where the additional factor H_b/h_f replaces the previously introduced scaling factor S_U and is individual for each building b.

Electricity meter data supplied by public utilities The information about the number of electricity meters (EMs) for each building nU_b^{EM} provided by the public utilities (DSO) is used for the direct calculation of the year-round load data. In this data source, $\mathbf{EM}$, an electricity meter for general electricity is given for buildings with multiple households. This additional meter is considered in this context with

the fictitious electricity consumption of 0.1 households. The load for a building E_b^{EM} can be estimated as follows,

$$E_b^{EM} = \begin{cases} nU_b^{EM} \cdot E_U(A_{U,b}), & \text{if } nU_b^{EM} < 3 \\ \left(nU_b^{EM} - 0.9\right) \cdot E_U(A_{U,b}), & \text{else,} \end{cases} \tag{4.7}$$

where $A_{U,b}$ is the average floor area of a residential unit in the building b. This formula assumes that buildings with less than three EMs have one EM per residential unit, while buildings with more EMs also have one EM for general electricity, e.g., for hallways and the basement. $A_{U,b}$ is calculated as

$$A_{U,b} = A_b \cdot H_b/h_f \;/\; nU_b^{EM}, \tag{4.8}$$

which effectively divides the total floor area over all stories by the number of EMs. The generated load time series, as described in Equation (4.1), can then be utilized to perform time series-based static simulations to evaluate the models under a wide range of conditions.

4.2.2 Stage 1: Automated Generation of the 20 kV Grid

In this section, we introduce the first stage of the optimization-based grid generation that creates a $20\,\mathrm{kV}$ distribution grid. This stage is based on map data and the load estimation introduced in the previous section, and consists of two consecutive steps: First, the estimation of the number and locations of the $20/0.4\,\mathrm{kV}$ substations and second, the grid topology generation connecting these substations.

Calculation of the Number and Locations of Substations The results of the load estimation described in Section 4.2.1 are used to estimate the adequate number of transformers for the grid. This estimation is based on the peak load per building $P_{peak,b}$ which is calculated as

$$P_{peak,b} = P_{peak,U} \cdot nU_b, \tag{4.9}$$

where nU_b is the estimated number of residential units of building b, which depends on the utilized data source. In accordance with [8], each household accounts for a peak load of $2\,\mathrm{kW}$. While each individual household likely has a higher peak load, this estimation considers a certain simultaneity factor when estimating the peak for a number of households.

Tab. 4.1.: The constants used for the load and transformer estimation with their respective values and sources.

Symbol	Description	Value	Source
nR	Average number of residents per unit	1.7	[77]
E_R	Yearly consumption per resident	200 kWh	[78]
E_A	Yearly area-dependent consumption	9 kWh m^{-2}	[78]
nLA	Number of large appliances per unit	8.4	[78]
E_{LA}	Yearly consumption per large appliance	200 kWh	[78]
h_f	Average floor height	2.6 m	[79]
$P_{peak,U}$	Simultaneous peak load per residential unit	2 kW	[8]
λ	Power factor	0.95	
R_T	Transformer rating	630 kVA	
L_T	Target transformer loading	50 %	

Assuming a power factor λ for the household loads and a maximum loading L_T for each of the $20/0.4\,\mathrm{kV}$ substations with a rating of R_T, the required number of transformers nT is determined as

$$nT = \left\lceil \frac{\sum_{b \in \mathcal{B}} P_{peak,b}}{\lambda \cdot R_T \cdot L_T} \right\rceil . \tag{4.10}$$

In order to then compute the locations of the stations, a *k-means clustering* algorithm is used [80]. Each of the nT returned clusters is based on each building's geographical coordinates, uses the case-specific number of households per building nU_b as the weight for the computation, and has a cluster center that is used for the transformer location. Once this is done, a graph representation of the street layout is obtained, that includes the closest HV/MV substation and the area where the grid is to be generated. Then the calculated station positions are added as nodes to this graph structure in the same way that each building is appended.

Optimization-based 20kV Grid Topology Generation The basic requirement for the $20\,\mathrm{kV}$ grid is connecting all MV/LV substation to the supplying HV/MV station in a ring topology. To find a suitable topology, an optimization approach is applied that minimizes the length of the cables required for the grid creation. This minimization of cable lengths is modeled as a travelling salesman problem [81], with the starting and ending point being the HV/MV station. The places to visit are the $20\,\mathrm{kV}$ substations and the distances are the shortest paths on the corresponding graph that is weighted with the actual, geographical length of each road. This problem

is approximated using the Christofides algorithm [82] and can be formalized as follows:

$$\min \sum_{(i,j)\in\mathcal{P}} l_{i,j} \cdot x_{i,j} \tag{4.11}$$

$$x_{i,j} = x_{j,i} \quad \forall (i,j) \in \mathcal{P} \tag{4.12}$$

$$\sum_{(i,j)\in\mathcal{P}} x_{i,j} = 2 \quad \forall i \in \mathcal{S} \tag{4.13}$$

$$\sum_{(i\neq j)\in s} x_{i,j} \leq |s| - 1 \quad \forall s \subset \mathcal{S}. \tag{4.14}$$

The MV/HV substation together with the $20\,\mathrm{kV}$ stations form the set of nodes, $\mathcal{S}$, while $\mathcal{P}$ is the set of the shortest paths (i, j) between each of the nodes in $\mathcal{S}$. Furthermore, $l_{i,j}$ is the length of the shortest path between node i and node j. The decision variable $x_{i,j}$ equals 1 if the path between node i and node j is part of the solution, and 0 otherwise. Equation (4.12) ensures symmetry while Equation (4.13) makes sure that each station is connected to exactly two other stations. The last constraint Equation (4.14) ensures that the solution is indeed a single $20\,\mathrm{kV}$ ring and not just a union of other, smaller rings.

4.2.3 Stage 2: Automated Generation of the Low-Voltage Grid

After creating the MV grid topology in the previous stage, the goal of the second stage is creating the underlying LV subgrids. These LV grid models are derived from the street layouts available in OSM as possible cable routes. These layouts are converted to a graph representation $G = (V, E)$ with edges E representing the streets and nodes V representing the crossings between the streets. The buildings of the area found in OSM are added as additional nodes to V and connected to the streets via additional edges. For each of these buildings, the load is estimated using one of the datasets described in Section 4.2.1. In order to obtain the grid topology, a variation of the minimum cost flow optimization problem modifies the graph to comply with electrical low-voltage grid topology standards. This optimization is carried out with specified $20\,\mathrm{kV}$ substation locations—either from stage one or from prior knowledge. The optimization problem is formulated as a mixed integer

linear program (MILP) with binary decision variables and is stated in the following equations:

$$\min \sum_{(i,j)\in E} \text{cost}_{i,j} \cdot \text{install}_{i,j} \tag{4.15}$$

$$\sum_{j:(i,j)\in E} \text{flow}_{i,j} \quad - \sum_{j:(j,i)\in E} \text{flow}_{j,i} = \text{residual}_i \quad \forall i \in V \tag{4.16}$$

$$\sum_{i:(i,j)\in E} \text{install}_{i,j} \leq 1 \quad \forall j \in V \tag{4.17}$$

$$0 \leq \text{flow}_{i,j} \leq \text{cap}_{i,j}^{\max} \cdot \text{install}_{i,j} \quad \forall (i,j) \in E \tag{4.18}$$

$$\text{install}_{i,j} \in \{0,1\} \quad \forall (i,j) \in E. \tag{4.19}$$

Generally speaking, the program decides which of the edges of the graph will be used as cables for power delivery by choosing the cheapest way to supply all demand. Therefore, the objective function (4.15) minimizes the cost $\text{cost}_{i,j}$ to install a cable from node i to node j that is proportional to the potential cable route length. A binary decision variable $\text{install}_{i,j}$ is introduced in order to only account for potential routes hat are actually used or, in other words, where a cable is installed, and the route used to supply buildings with power: $\text{install}_{i,j}$ is 1 if the cable is installed between nodes i and j, and 0 in all other cases. The first flow conservation constraints (4.16) ensure that for all nodes i that are not a source, the power flowing in $\text{flow}_{j,i}$ equals the consumption residual_i of the node itself or is flowing out again. For sources that represent the secondary substations, the power flowing out equals the externally provided power that is also denoted residual_i. A positive residual_i indicates an external source, while a negative residual_i indicates consumption. For this consumption, possible PV injection is not considered in the optimization out of the aforementioned historical reasons, only the load data computed in the previous steps is used. The second constraints (4.17) ensure radiality in the obtained grid, as LV distribution grids are usually operated in a radial mode. Hence, only one cable can be used to supply a node with power. Lastly, the constraints (4.18) make sure that a certain loading limit $\text{cap}_{i,j}^{\max}$ for the cables is not exceeded and that power can only flow over a cable that is also installed. The algorithm repeats this optimization while lowering the maximum cable loading limit permitted in the optimization for a cable that is installed alongside a public way, in order to obtain a topology that uses all the secondary substations equally. This approach requires a method to determine a low and still feasible cable loading limit—or cable capacity. While our previous work [66] describes a cable capacity estimation (CCE) method using the Newton bisection method (NB-CCE), we find that this method is ineffective in finding a

suitable capacity, which results in long run-times. To mitigate this issue, we propose the new inverse proportional estimation (IP-CCE):

$$\mathrm{cap}_{i+1}^{\max} = \mathrm{round}\left(\mathrm{cap}_{i}^{\max}\left(1 - \frac{1}{i+N}\right)\right), \tag{4.20}$$

where $\mathrm{cap}_i^{\max}$ describes the maximum cable capacity in the i-th iteration of the optimization and N is an adjustable parameter for the step size of the estimation process. Once the capacity restriction of an iteration makes the problem infeasible, the capacity of the last feasible iteration is selected and the results of that optimization are used for the rest of the workflow.

This new approximation allows for fast progress at the start of the optimization method and finer steps towards the end. This ensures that the found final capacity is actually close to the theoretical optimum that is the lowest integer capacity value with which the model remains feasible and that the obtained solution is still balanced regarding secondary substation use. Nonetheless, the application still finds good results much faster because, in contrast to the Newton bisection method, very time-consuming optimization steps around the theoretical optimum are avoided. Experiments with this formula have shown that $N = 4$ yields an especially favorable balance between runtime and quality of the results.

4.2.4 PowerFactory Model Generation

In the last step, the obtained grid data structure is converted to a DIgSILENT PowerFactory model. Since system parameters such as line impedances cannot be estimated from the utilized data sources, this approach uses a library of system components that are widely used in the target area. This library can, of course, be adapted to reflect component choices of the local DSO if these are known. This also includes the automated creation of grid diagrams and an initial load flow calculation. For each individual building, a busbar system as shown in Figure 4.2 can be created. Although the present chapter only describes the methodology to estimate the loads for these house models, this approach offers the flexibility required for the analysis of future scenarios with various PV, battery and electric vehicle settings. The developed Python tool chain is fully automated, relies solely on a good OSM data coverage of the target area, and the obtained grids can be readily used for power grid analysis studies. However, due to varying grid design approaches around the world, its usage is limited to regions with European-style distribution grids.

4.3 Evaluation

In this section, we describe the study area and the evaluation criteria before presenting the results. A meaningful comparison between the proposed method and the methods found in literature is infeasible due to several factors, e.g., different voltage levels [72, 73], different grid styles [13], and missing implementation details [10, 11, 74]. Thus, we focus on the evaluation of the impact of various available data sources and the comparison with real grid topology data provided by the DSO.

For this evaluation, we consider the three data sources introduced in Section 4.2 and generate the distribution grid model either with a priori knowledge of the $20/0.4\,\mathrm{kV}$ transformer locations together with the $20\,\mathrm{kV}$ grid topology $\mathbf{T_K}$ or with a calculation of the transformer locations and the $20\,\mathrm{kV}$ network topology $\mathbf{T_C}$. This altogether results in six combinations of (d, t) tuples for data source and transformer data, as given below:

$$(d, t) = \{\mathbf{O_{2D}}, \mathbf{O_{3D}}, \mathbf{EM}\} \times \{\mathbf{T_K}, \mathbf{T_C}\} \tag{4.21}$$

With this wide variety of data source combinations, we aim to investigate the impact of data source quality on the quality of the resulting grid models. In addition to the six models generated by these combinations, we validate the generated models with

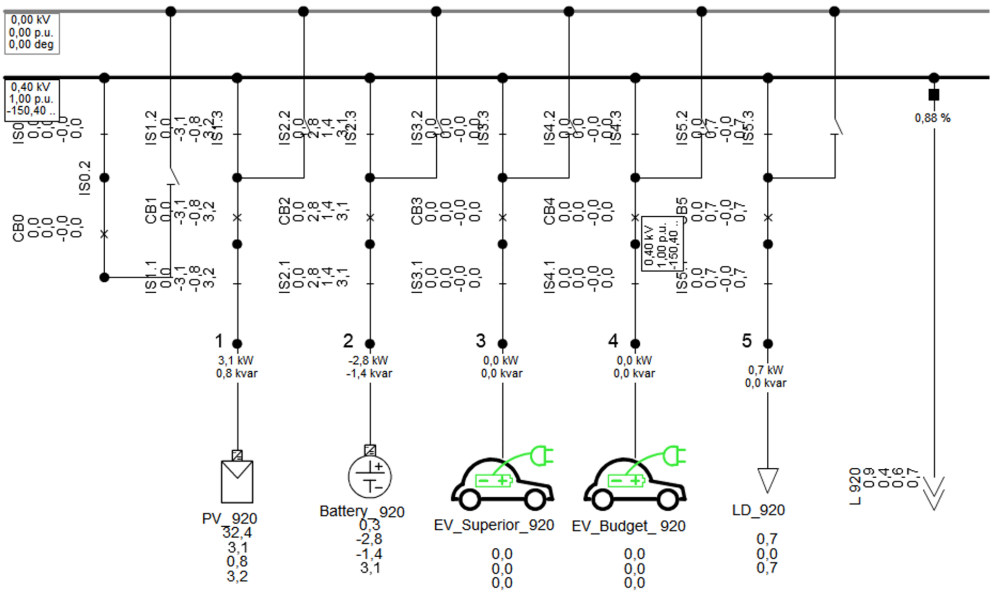


Fig. 4.2.: Detailed view of the automatically generated components in each house model as displayed by PowerFactory. Note that for the analysis in this study, only the residential load is considered. However, this approach offers the flexibility required to study various potential future scenarios.

a topological model, **DSO**, based on GIS data provided by the DSO of the target area. This model is not generated by the method described in this chapter, and has two significant differences compared to the generated models: First, it includes switching devices that allow the reconfiguration of the grid topology, and second, it contains two separate cables for most of the streets, one on each side.

For the analysis of the generated models, graph metrics are applied to the automatically generated topologies, and the results of network calculations based on voltage drops and line loadings are compared. While the **DSO** model is considered as the reference model for the topological comparison of the models, we consider the $(\mathbf{EM}, \mathbf{T_K})$ as a reference model for evaluation of the electrical properties. This is because the **DSO** model lacks information regarding the demand, whereas the $(\mathbf{EM}, \mathbf{T_K})$ is the generated model based on the most accurate available data, i.e., electricity meter data and known transformer positions. From these comparisons, statements are made regarding the quality of the automatically generated networks. In particular, an answer is given to the question which data are sufficient for an automated modeling of the power grid for reliable statements.

4.3.1 Study Area

The selected study area spans over roughly half a square kilometer and contains 241 buildings, including single houses, duplex houses, town houses as well as apartment towers. It is almost completely surrounded by forest, which simplifies the definition of a clear boundary. The selected study area is the same as in a previous work, i.e., [66], where detailed information was collected through an on-site inspection. This allows a direct comparison of the grid topologies generated in this thesis with previous results.

Even though the area is mostly residential, it also encompasses non-residential buildings such as a school, a community center and some shops. As the few shops in the area are in buildings that also encompass residential units, for this analysis they are treated as residential units for the sake of simplicity. The consumption of buildings with a purely non-residential use is calculated using the overall area available in the building multiplied by the yearly demand per floor area for the building type. For the present school and kindergarten, this value is 20 and $22\,\mathrm{kWh}$ per m^2 per year respectively, and for the community center it is $9\,\mathrm{kWh}$ per m^2 per year [83, 84]. For the overall area, the number of levels of a building is obtained either from the level tag within the OSM dataset or assumed to be two, thus $S_U = 2$ in this case.

Tab. 4.2.: Deviation of OSM-based load estimations compared to the load estimation based on DSO-supplied electricity meter data for the study area.

	$\mathbf{O_{2D}}$	$\mathbf{O_{3D}}$
Total deviation	$-58\,\%$	$-3\,\%$
Greatest negative building deviation	$-95\,\%$	$-69\,\%$
Greatest positive building deviation	$+65\,\%$	$+85\,\%$

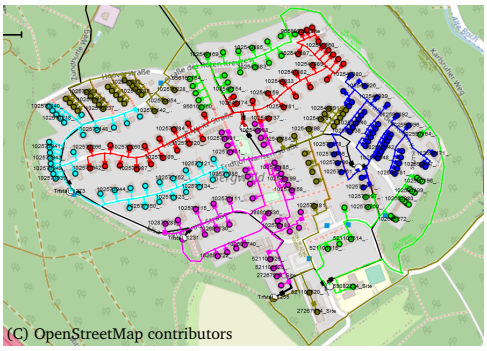
4.3.2 Load Estimation Comparison

Comparing the load estimations based on available 2D and 3D OSM data to the reference estimation based on DSO supplied EM data results in the deviations shown in Table 4.2. The table shows that the 2D-based estimations result in a significantly lower load overall, while still containing buildings with significantly higher load estimations. Thus, a simple adjustment of the scaling factor S_U would not be sufficient, as a higher value would also increase the positive deviations. On the other hand, the 3D-based estimations, while also containing some buildings with large deviations, match the overall load estimate very closely and perform better than a scaled 2D-based estimate would.

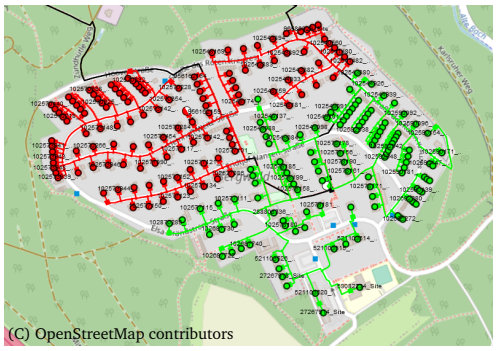
4.3.3 Topological Comparison

In this section, we perform a topological comparison of the models generated using the different data sources and generation methods. First, we describe our findings on the geographic representations of the models before quantizing them using two graph metrics, i.e., the number of nodes per transformer and the eccentricity of the transformers.

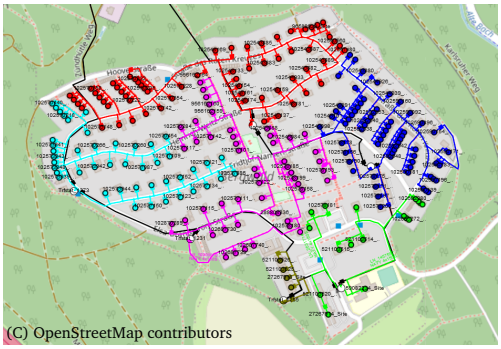
Geographic representation The model in Figure 4.3a, generated by $(\mathbf{O_{2D}}, \mathbf{T_K})$, shows six $400\,\mathrm{V}$ subgrids with similar sizes that are not very compact and in two cases (green and brown) spread across the whole study area. Especially notable are the two cables at the right edge of the area. Figure 4.3c shows the model created by $(\mathbf{O_{3D}}, \mathbf{T_K})$, which includes much more compact subgrids with larger size differences. Notably, the brown and green subgrids are significantly smaller compared to the previous case. Figure 4.3e shows the $(\mathbf{EM}, \mathbf{T_K})$ model with even smaller subgrids in the lower part of the study area. Consequently, the remaining subgrids are significantly larger. This model also includes an unrealistic connection via a footpath in the brown subgrid. Examining the models with calculated transformer positions,



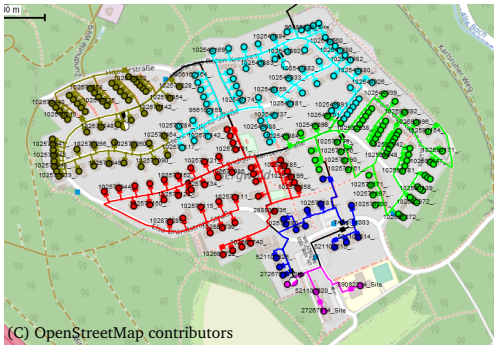
(a) Case $(\mathbf{O_{2D}}, \mathbf{T_K})$: Grid model using the OSM dataset for given number and location of transformers.



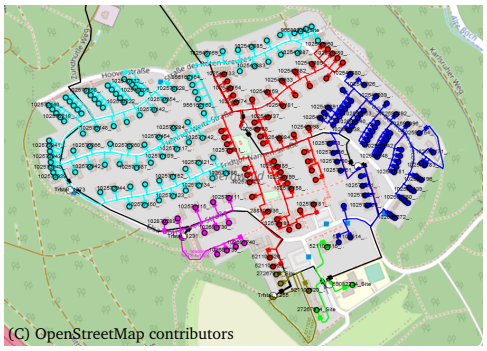
(b) Case $(\mathbf{O_{2D}}, \mathbf{T_C})$: Grid model using the OSM dataset for a calculated number and location of transformers.



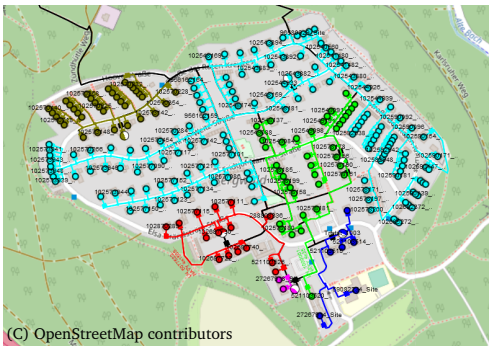
(c) Case $(\mathbf{O_{3D}}, \mathbf{T_K})$: Grid model using the OSM Buildings dataset with included height data of buildings for given number and location of transformers.



(d) Case $(\mathbf{O_{3D}}, \mathbf{T_C})$: Grid model using the OSM Buildings dataset with included height data of buildings. The number and location of transformers are calculated.



(e) Case $(\mathbf{EM}, \mathbf{T_K})$: Grid model using the DSO data on the number of electricity meters at each building for a given number and location of transformers.



(f) Case $(\mathbf{EM}, \mathbf{T_C})$: Grid model using the DSO data on the number of electricity meters at each building. The number and location of transformers are calculated.

Fig. 4.3.: The three figures on the left show the grid models with *a priori known* number and location of substations using different data sources for the load modeling, indicate with $\mathbf{T_K}$. The color of the nodes, i.e., buildings, and lines indicates the affiliation to a specific transformer, with a consistent coloring between the three variants. Analogously, the right side shows the grid models with a *calculated* number and position of substations using a k-means approach, indicated with $\mathbf{T_C}$. As there is no one-to-one relation between the transformers of the different variants, the coloring of the nodes is not consistent.

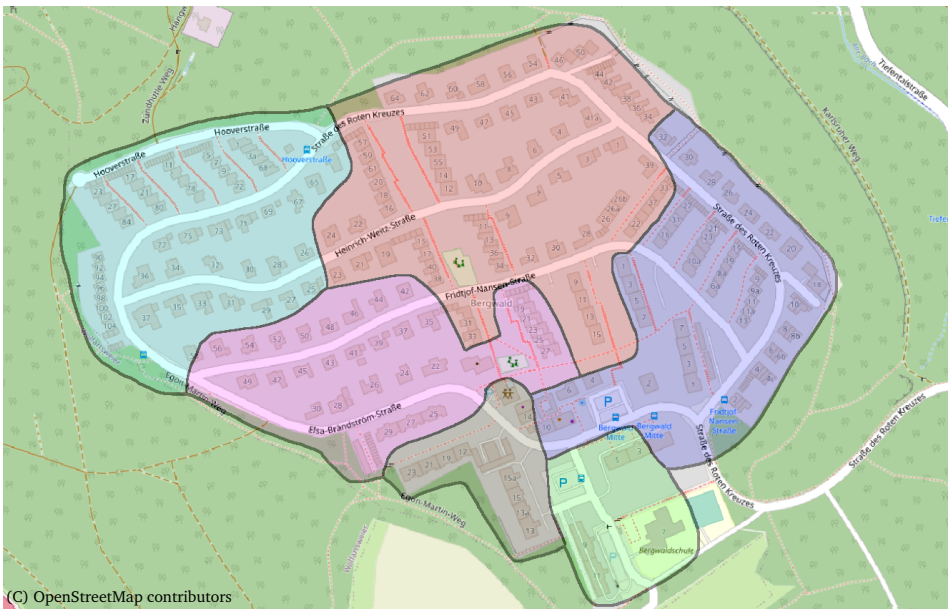


Fig. 4.4.: A simplified representation of the **DSO** model, showing the areas supplied by the six different transformers. The coloring is equivalent to Figure 4.3.

it is noticeable that the $(\mathbf{O_{2D}}, \mathbf{T_C})$ case, as shown in Figure 4.3b, contains only two transformers compared to the usual six transformers in all other models. The $(\mathbf{O_{3D}}, \mathbf{T_C})$ model in Figure 4.3d presents a realistic partitioning into subgrids, with two smaller subgrids in the lower part and four evenly distributed larger subgrids in the upper part of the study area. The $(\mathbf{EM}, \mathbf{T_C})$ in Figure 4.3f contains very unevenly distributed subgrids with a high variance in size. Compared to the six generated models, the **DSO** model offers flexibility in its topology, as it contains switchgear at 15 different locations with a variety of switching options, resulting in a very high number of interconnection variants. Since the data provided by the DSO does not include details about the switching states of the switchgear, some assumptions are required to obtain a valid configuration. The configuration that results in the partitioning shown in Figure 4.4 is based on the transformer positions and aims to balance the consumption in each area while keeping each area compact. Overall, the **DSO** model shows the closest similarity to the $(\mathbf{O_{3D}}, \mathbf{T_C})$ version of the generated models, which requires the least a priori knowledge of sensitive data.

Number of nodes per transformer In this metric, a node represents a busbar in the generated PowerFactory model. For each building, our generation methods create two busbars: One for the building itself and one at the connection of the main cable and the house cable. The number of nodes per transformer is calculated by determining the supplying transformer for each $400\,\mathrm{V}$ busbar. Figure 4.5 shows the

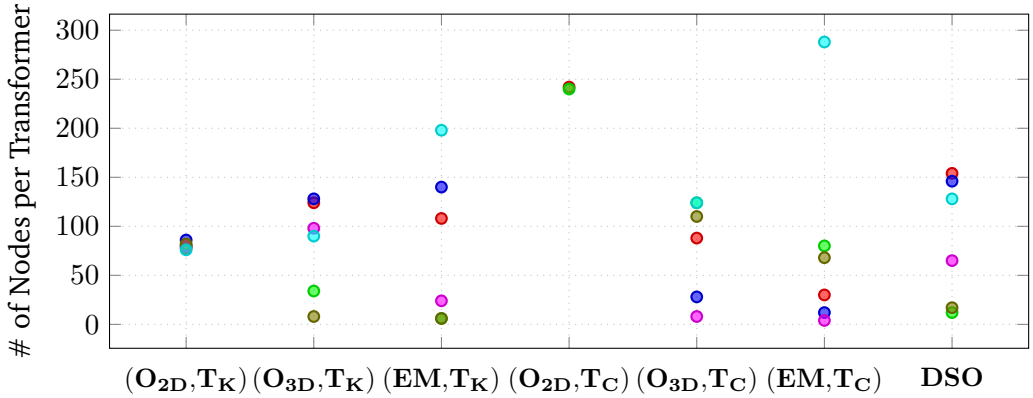


Fig. 4.5.: The number of nodes per transformer shows a very even distribution for both $\mathbf{O_{2D}}$ cases and a wider distribution for the other cases. The colors refer to the subnetworks in Figure 4.3.

number of nodes per transformer for the six generated models. The two models using only the 2D OSM data reveal a very even distribution of nodes between the different transformers, due to the homogeneity of the estimated loads. Furthermore, it is noteworthy that the $(\mathbf{O_{2D}}, \mathbf{T_C})$ case consists of only two transformers with around 250 connected nodes each, which is highly unrealistic. According to DSO data, one $630\,\mathrm{kVA}$ transformer, which is the installed transformer type in the study area, serves on average 79 residential units. The other cases show a more heterogeneous distribution of nodes due to the large differences in building load estimations, that are far more realistic for some multi-story buildings. While containing more nodes overall due to some implementation details, the **DSO** model located in between the $(\mathbf{O_{3D}}, \mathbf{T_K})$ and $(\mathbf{EM}, \mathbf{T_K})$ variants with a good matching to $(\mathbf{O_{3D}}, \mathbf{T_C})$.

Eccentricity per transformer The eccentricity of a node is defined as the maximum distance to all other nodes in a graph. Thus, when determining the eccentricity of a transformer T_i, we calculate

$$\mathrm{ecc}(T_i) = \max_{n \in N_{T_i}} (d(T_i, n)), \tag{4.22}$$

where N_{T_i} is the set of $400\,\mathrm{V}$ nodes that are supplied by T_i, and d is the cable distance between two nodes. Essentially, the eccentricity of a transformer describes the maximum cable length between the transformer and the buildings supplied by it. The comparison of eccentricities in the generated models is illustrated in Figure 4.6. This plot reveals two outliers in the $(\mathbf{O_{2D}}, \mathbf{T_C})$ case with an eccentricity of $2.1\,\mathrm{km}$ and $1.2\,\mathrm{km}$ respectively, while all other transformers have an eccentricity below $0.75\,\mathrm{km}$. The **DSO** model features very similar eccentricity values to the $(\mathbf{O_{3D}}, \mathbf{T_C})$ variant

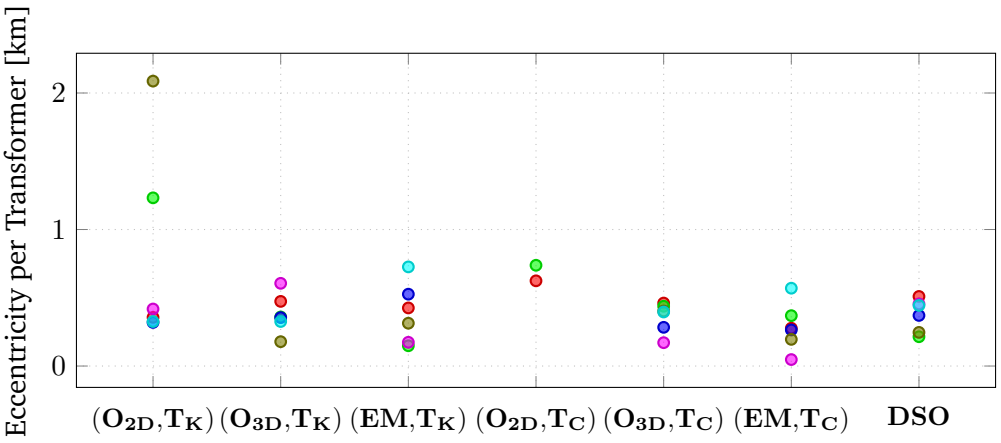
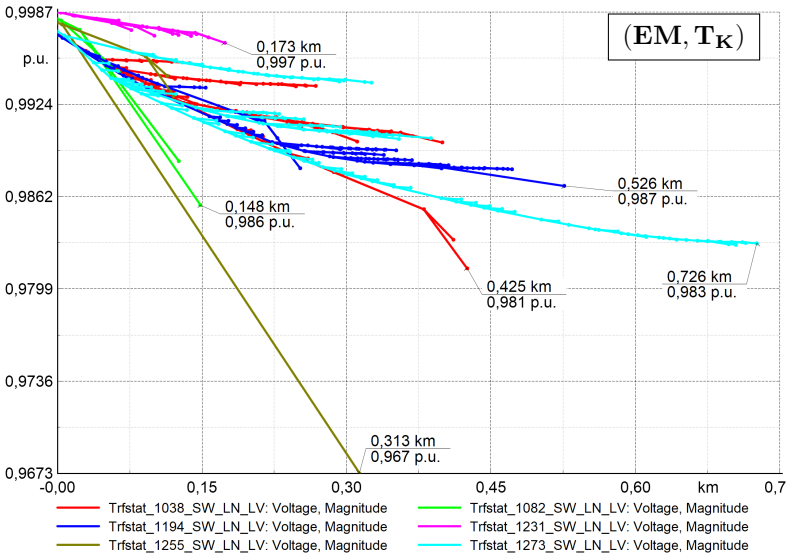


Fig. 4.6.: The eccentricity per transformer shows the maximum distance between a transformer and the connected $400\,\mathrm{V}$ nodes. This metric reveals two very large distances for the first case, while the other cases show more evenly distributed eccentricities. In general, the **EM**-based cases show a wider distribution than the 3D-data-based cases. The colors refer to the subnetworks in Figure 4.3.

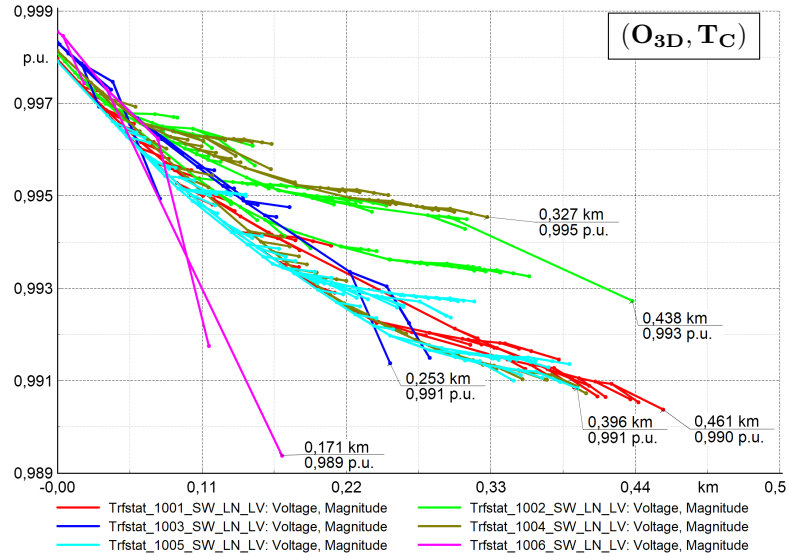
with a small range of values, because of the configuration criterion to keep the individual partitions compact. In general, the **EM**-based models exhibit a wider distribution of eccentricities than their $\mathbf{O_{3D}}$-based counterparts, whereas the $\mathbf{T_C}$-based models have overall lower eccentricities than their $\mathbf{T_K}$-based equivalents.

4.3.4 Electrical Comparison

In this section, we perform a comparison of the electrical properties of the generated distribution grid models. In a first step, we analyze the voltage profiles of all cases and verify that the voltage profiles are within the valid voltage range. In particular, we compare the voltage profiles of the reference model $(\mathbf{EM}, \mathbf{T_K})$ shown in Figure 4.7a with the proposed method $(\mathbf{O_{3D}}, \mathbf{T_C})$ shown in Figure 4.7b. The proposed method based on public data and transformer estimation outperforms regarding the voltage range. Both are in a valid range, but the case $(\mathbf{O_{3D}}, \mathbf{T_C})$ has a significantly narrower voltage band of $[0.989, 0.999]$ p.u. compared to the reference model, with $[0.967, 0.998]$ p.u. This also correlates with the shorter radial feeder length of $0.461\,\mathrm{km}$ compared to $0.726\,\mathrm{km}$ in the reference model. In both diagrams, the voltage profile with the steepest drop is associated with a transformer with high loads in multi-story buildings supplied with short length cables. Thus, this is another indicator of the appropriate choice of methodology for approximating the location and number of transformers, based on load estimation from building data, which itself also seems promising.



(a) The voltage profiles for case $(\mathbf{EM}, \mathbf{T_K})$ show steep decreases for the brown and green feeders, which contain multifamily buildings with around 70 electricity meters each. Counting one meter per residential unit yields an unrealistically high number of units for these buildings, which causes the voltage to drop below $0.97\,\mathrm{p.u.}$



(b) The voltage profiles for case $(\mathbf{O_{3D}}, \mathbf{T_C})$ also show a steep decline for the feeder containing the multifamily buildings (pink). In this case, however, the overall voltage band is much more narrow.

Fig. 4.7.: The voltage profiles show the gradient of the bus voltages from the transformer to each building supplied by the transformer. As most of the buildings consume more power than they produce, the voltages decline from the transformers to the buildings.

Further, we analyze the statistical distribution of line loadings of the $0.4\,\mathrm{kV}$ cables as shown in Figure 4.8. First, it is noticeable that the distributions for the two $\mathbf{O_{2D}}$ cases differ seriously from the others. This is due to the weak load estimation based on 2D OSM data that leads to overall lower line loadings. To further analyze the similarities of the line loading distributions, a similarity index is defined based on the Euclidean distance of the histogram distributions. For two models x and y with their respective histogram of line loading histograms $\mathcal{L}$, the similarity between the two histograms is, thus, calculated as

$$\mathrm{sim}(\mathcal{L}_x, \mathcal{L}_y) = \sqrt{\sum_{i=1}^{N_b} (\mathcal{L}_{x,i} - \mathcal{L}_{y,i})^2}. \tag{4.23}$$

N_b is the number of bins—50 in this case—and $\mathcal{L}_{x,i}$ is the number of lines in bin i of histogram $\mathcal{L}_x$. The full table of similarity indices in pairs of the cases is given in Figure 4.9. The highest similarity, i.e., smallest distance, with the reference case $(\mathbf{EM}, \mathbf{T_K})$ (shown in red in Figure 4.9) is observable for case $(\mathbf{EM}, \mathbf{T_C})$ with 15.68, where both share the same smart meter data distribution over the buildings. The second-best match with the reference is the case $(\mathbf{O_{3D}}, \mathbf{T_K})$ with 57.48, where the transformer number and positions are identical and a priori known. The very close third-best option is $(\mathbf{O_{3D}}, \mathbf{T_C})$ with 59.7, which does not need any additional input data for network topology generation at all and seems a promising approach for future use. The second finding from the similarity index table concerns the quality of the approximation of the number and location of the transformer stations. For this purpose, we compare given $(\mathbf{T_K})$ and calculated $(\mathbf{T_C})$ transformer properties, for each case. The values shown in yellow in Figure 4.9 indicate a high accordance between the $\mathbf{T_C}$ and corresponding $\mathbf{T_K}$ models. Note that this is only a statistical comparison without consideration of the real spatial distribution, which is handled in Section 4.3.5.

4.3.5 GIS-based Comparison

To be able to evaluate the approach for the automatic $20\,\mathrm{kV}$ network generation, the calculated transformer positions $\mathbf{T_C}$ are compared to the known positions $\mathbf{T_K}$ for selected cases. This includes the comparison of cases $(\mathbf{EM}, \mathbf{T_C})$ and $(\mathbf{O_{3D}}, \mathbf{T_C})$ to case $(\mathbf{EM}, \mathbf{T_K})$. In particular, each calculated position is mapped to the closest known transformer. If this mapping is not unambiguous, the closest calculated position is chosen to be mapped to the known transformer location in question.

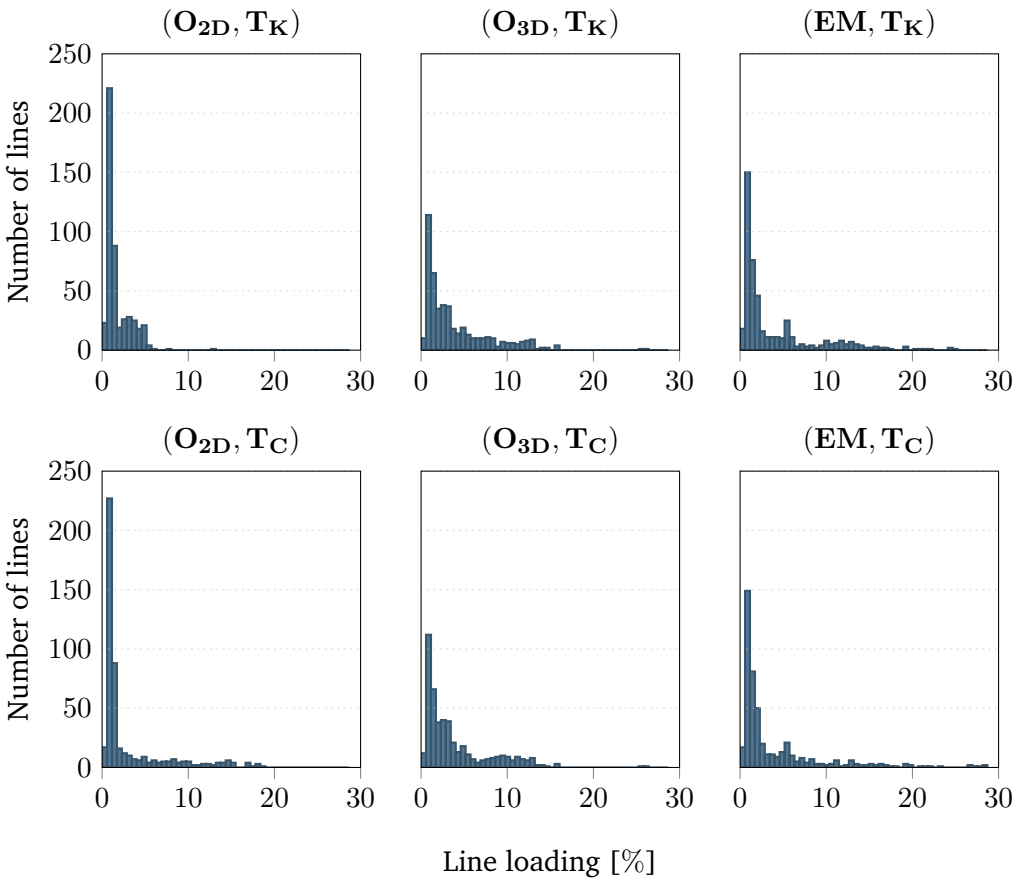


Fig. 4.8.: The histograms of line loadings at $0.4\,\mathrm{kV}$ show a high accordance between the $\mathbf{O_{3D}}$ and $\mathbf{EM}$-based models. Furthermore, the line loadings in these models appear to be only slightly affected by the transformer placement method. The histograms are created with 50 bins from zero to $28.7\,\%$, resulting in a bin width of $0.574\,\%$.

	$(\mathbf{O_{2D}}, \mathbf{T_K})$	$(\mathbf{O_{3D}}, \mathbf{T_K})$	$(\mathbf{EM}, \mathbf{T_K})$	$(\mathbf{O_{2D}}, \mathbf{T_C})$	$(\mathbf{O_{3D}}, \mathbf{T_C})$	$(\mathbf{EM}, \mathbf{T_C})$
$(\mathbf{O_{2D}}, \mathbf{T_K})$	0	116.7	87.43	39.34	118.3	87.07
$(\mathbf{O_{3D}}, \mathbf{T_K})$	116.7	0	57.48	125.7	14.14	56.57
$(\mathbf{EM}, \mathbf{T_K})$	87.43	57.48	0	87.59	59.7	15.68
$(\mathbf{O_{2D}}, \mathbf{T_C})$	39.34	125.7	87.59	0	128.5	88.51
$(\mathbf{O_{3D}}, \mathbf{T_C})$	118.3	14.14	59.7	128.5	0	58.96
$(\mathbf{EM}, \mathbf{T_C})$	87.07	56.57	15.68	88.51	58.96	0

Fig. 4.9.: The similarity indices of the cases based on histograms of line loadings show a high accordance between models with the same load estimation and different transformer placements (smaller values indicate high similarity). Furthermore, they confirm the similarity of the $\mathbf{O_{3D}}$ and $\mathbf{EM}$ models.

Tab. 4.3.: The results of the location-based comparison show very similar distances between known and calculated transformer positions for the $\mathbf{O_{3D}}$ and $\mathbf{EM}$ model.

Transformer No.	Distances from **DSO** case to $(\mathbf{O_{3D}}, \mathbf{T_C})$	to $(\mathbf{EM}, \mathbf{T_C})$
1	150 m	148 m
2	83 m	72 m
3	37 m	42 m
4	116 m	105 m
5	50 m	37 m
6	127 m	123 m

That way, there is a one-to-one comparison, the results of which are depicted in Table 4.3.

Given that the study area measures $582\,\mathrm{m}$ from the most northern to most southern point and $766\,\mathrm{m}$ from east to west, the rather low distances mean that the methodology is working well. The fact that the values change not much from the $\mathbf{EM}$ to the $\mathbf{O_{3D}}$ case means that the OSM data with height information of buildings is sufficient for good results for transformer placement.

4.3.6 Runtime Evaluation

The runtime of the introduced Inverse Proportional Cable Capacity Estimation (IPCCE) method (4.20) as part of the optimization method in the course of low voltage grid topology generation (see Section 4.2.3) is evaluated with a comparison to the Newton bisection method in [66].

To obtain the runtimes, the application was timed on a Windows machine with $32\,\mathrm{GB}$ of RAM storage and an Intel® Core™ i7-10700K CPU with 8 physical cores running at $3.8\,\mathrm{GHz}$. As can be seen in Figure 4.10, the runtime of the optimization was significantly reduced by $1307\,\mathrm{s}$, reducing the total runtime of the model generation from 1618 to $311\,\mathrm{s}$. This results in an overall speedup factor of 5.2. While most of the programs steps do not differ and therefore stay roughly the same, the difference in the optimization process can be clearly seen. This improvement is especially important because with bigger models to be created, the optimization time is expected to grow exponentially.

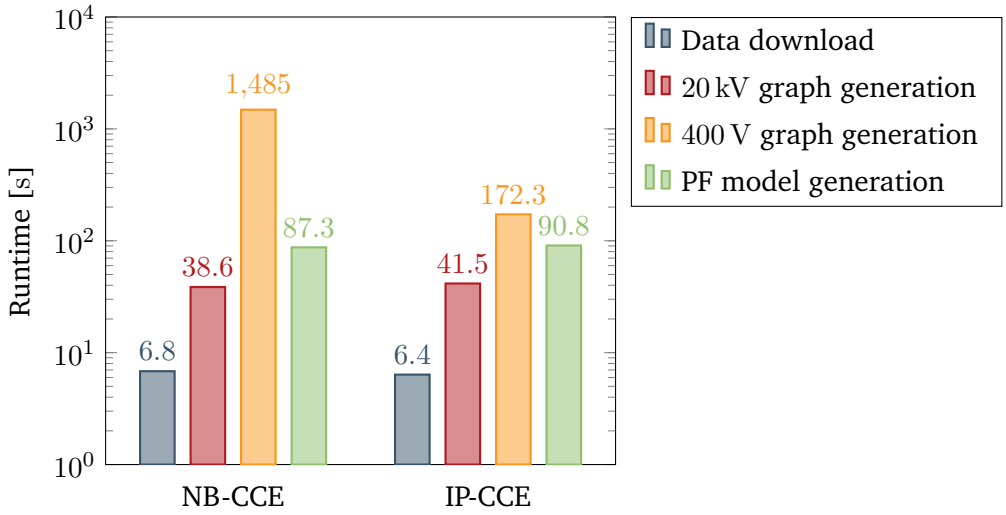
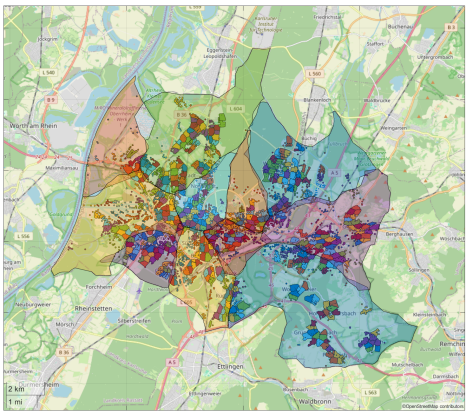


Fig. 4.10.: Runtime comparison of the introduced Inverse Proportional estimation (IP-CCE) to the Newton Bisection method (NB-CCE) from [66] in seconds. A speedup by a factor of 8.6 is achieved for the optimization with the new method for cable capacity estimation. The 20 kV graph generation includes the steps described in Section 4.2.2, while the 400 V graph generation comprises Section 4.2.3. The last step includes the generation of a PowerFactory model as described in Section 4.2.4.

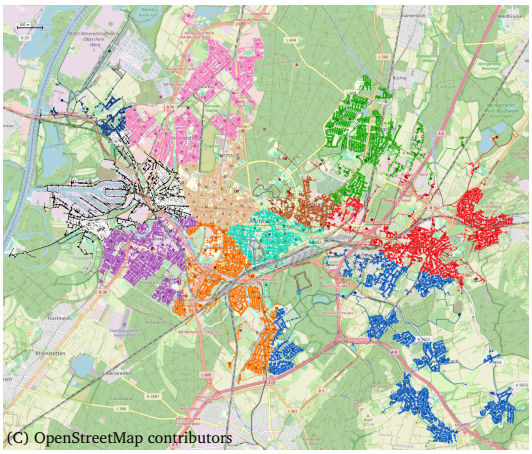
4.4 Large-Scale Demonstration

To give an outlook on the large-scale applicability of the presented model generation approach, we apply it to the whole city of Karlsruhe in Baden-Württemberg, Germany. This area contains a total of 44 917 buildings. The only additional information used to generate this large-scale model is the position of the eleven 110/20 kV transformers and the cables connecting them, and for load estimation the total number of inhabitants in the area.

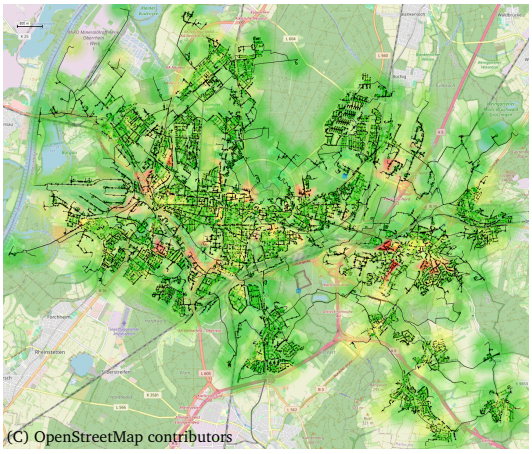
Based on the positions of the 110/20 kV transformers, Voronoi cells are calculated on the OSM graph, assigning every building that is represented as an OSM graph node in the area to one of the transformers. Approximations of the resulting cells can be seen as semi-transparent areas in Figure 4.11a. For every Voronoi cell, the number of 20/0.4 kV transformers is estimated as described in Section 4.2.2. However, instead of connecting all transformers in one ring, the number of rings is estimated based on the statistical number of 18 transformers per ring, which is derived from available grid data. Thus, the n_T 20/0.4 kV transformers of every cell are clustered using *k-means* with $\lceil n_T/18 \rceil$ clusters and connected in a ring topology for each of these clusters as previously described. For the LV grid generation, the process is exactly as described in Section 4.2.3.



(a) Cluster of buildings around 20 kV transformers within the 110 kV high-voltage grid partitions obtained with Voronoi-partitioning based on the street graph. Since the Voronoi cells are not convex and have some frayed edges, approximations are depicted for more clarity.



(b) The geographic representation of the generated PowerFactory model. The colors indicate the affiliation of buildings to the different 110/20 kV transformers.



(c) The load flow results based on 2D building and calculated transformer data $(\mathbf{O_{2D}}, \mathbf{T_C})$ with artificial PV installations based on 25 % of the building floor areas. The coloring indicates the bus voltages with values between 1.0 p.u. (green) and 1.11 p.u. (red) for a single time step around noon on a summer day with high irradiation.

Fig. 4.11.: The three figures show different stages of the large-scale modeling of a city with almost 45 000 buildings. From a purely geographical representation on the left to a working PowerFactory model and a load flow calculation.

Based on the $\mathbf{O_{2D}}$ dataset, the resulting PowerFactory model can be seen in Figure 4.11b. In this figure, the colors indicate the final affiliation of buildings and MV/LV transformers to the corresponding HV/MV transformers, as approximated by the semi-transparent cells in the previous figure. Finally, Figure 4.11c visualizes the load flow results for the case study. There, the background color indicates the bus voltages resulting from a load flow calculation with values between $1.0\,\mathrm{p.u.}$ (green) and $1.11\,\mathrm{p.u.}$ (red). In this model, every building is equipped with a PV system that covers $25\,\%$ of the building's base area. In this scenario with relatively low loads, due to the estimation based on 2D data, this results in a relatively well balanced system, with an average bus voltage of $1.02\,\mathrm{p.u.}$, which is well inside the tolerated limits. Only 26 of the roughly $631\,000$ buses are outside the valid voltage band of $\pm 10\,\%$. However, since the estimated loads are not realistic for some of the diverse consumers in the whole region, these results should be interpreted qualitatively. The results indicate that, while the load flow in generally not critical, the generated grid model might require more optimization in future work.

While the $400\,\mathrm{V}$ grid generation works as expected in most cases and demonstrates the scalability of the presented approach, the introduced method displays three clear shortcomings at this scale. First, the Voronoi partitioning does not necessarily lead to sensible supply areas, as the cell in Figure 4.11c demonstrates, with its natural division by a large forest area. Second, the load estimation for industrial buildings likely does not yield sensible results often, and in some cases it might not be appropriate to connect these loads on the $400\,\mathrm{V}$ level. Third, the method creating the MV rings follows the OSM street graph, which might not be appropriate in rural areas and should also be extended to consider overhead power lines instead of cables.

4.5 Discussion

The presented new model generation approach enables the automated generation of distribution grid models from minimal open data sources. The evaluation of the models generated using various data sources, from open 2D and 3D data to proprietary electricity meter data and GIS data including transformer positions, demonstrates the viability of open data sources for the automated generation of realistic distribution grid models. However, the evaluation also shows that the 2D data alone, as available in OSM, does not yield realistic models with the presented approach. On the other hand, combined with the height information found in OSM Buildings, the generated models closely resemble the models utilizing proprietary

electricity meter data supplied by a local DSO, which confirms the viability of the household estimation method.

While the load estimation is shown to be crucial for the subsequent estimation of the transformer number and positions and the cabling, the transformer placement itself is less important for the evaluated metrics. Especially for the loadings of the $400\,\mathrm{V}$ lines, the transformer placement is nearly negligible, as Figure 4.9 shows. In general, however, the transformers placed by the deployed optimization algorithm lead to more efficient cable layouts (see Figure 4.6).

Overall, the evaluation shows the viability of our new method to generate realistic distribution grid models from openly available building data. In some instances, the estimated household numbers might even be more accurate than the DSO data, since some buildings contain an unrealistic high number of meters.

The proposed method can be used to create new open test cases that can be shared between research partners. Compared to existing, manually created test cases, they are likely less accurate, but the automated generation enables more comprehensive testing than with existing test cases. The generated models lack details, such as PV systems, batteries, electric vehicles, which drastically impact the consumption and generation in the LV grids. As such, the generated models should be viewed as a basis that can be enhanced with artificial or estimated data to fit the needs of specific use cases.

However, a thorough comparison to a real model under various load scenarios still needs to be performed. Furthermore, the load estimation method proposed in the present work is limited to residential buildings and needs to be expanded to nonresidential and mixed buildings.

4.6 Contribution and Future Work

In this chapter, we answer two research questions. Research question **RQ2** addresses the generation of distribution grid models with minimal data requirements. As the methods found in the literature often require very specific, or even proprietary, data source, we propose a new methodology that relies solely on map data and load estimation. Having the load estimation decoupled from the grid generation, this approach can be adjusted to the given data sources and works with minimal data requirements. The second research question, **RQ3**, addresses the effect of the available data sources on the accuracy of the generated models. To answer this

question, we generate models from 2D map data, 3D data including the height of buildings, and data regarding the number of electricity meters per building. We compare the generated models using several metrics that quantify the topological and electrical properties of the grids.

In conclusion, this chapter provides the following contributions:

- We introduce a new distribution grid model generation method, that enables the automated generation of grid models relying solely on openly available data sources, i.e., OpenStreetMap (OSM) and OSM Buildings. It introduces a building load estimation method that is based on the estimation of households that utilizes open 2D and 3D building data.
- We show that the 3D-based estimation performs similarly to proprietary electricity meter data supplied by the distribution system operator (DSO) of the study area, while the household estimation based on 2D data results in a poor model accuracy. The evaluation also shows the relatively small importance of the actual $20/0.4\,\mathrm{kV}$ transformer placement for the generation of realistic models.
- We demonstrate the scalability of our approach by combining the grid optimization with clustering and Voronoi-partitioning to handle model regions with almost $45\,000$ buildings.

Given the results of this chapter, there are several areas that future work can improve on. As the evaluations shows, the load estimation is crucial for the model accuracy. To improve the accuracy for non-residential areas, future work should include a way of incorporating commercial and industrial areas and buildings into the model, as well as a way to automatically detect the kind of area that is present, e.g. residential, commercial, mixed. Furthermore, our approach has to be further developed in order to be able to accommodate more varying and challenging network topologies such as inner-city blocks with buildings inside the inner courtyard or overhead lines that are still present around our study area and thus in Germany. Moreover, as seen in the **DSO** model, real grids contain switchgear that allows the reconfiguration of the network topology to react to certain grid events. Incorporating these switching capabilities into the model generation process, and to find realistic configurations for the switchgear is important to enable evaluating grid flexibilities.

Summary Part II

In this part, we answer the three research questions motivated by challenges in distribution grids that are identified in the introduction to this thesis. The contributions of this part to answering these questions and helping to overcome the challenges can be summarized as follows:

RQ1 How to handle missing values in energy time series?
We introduce a new data-driven imputation method for energy time series and compare it with common benchmark methods on a real-world dataset, confirming the imputation quality and real-world applicability of the method.

RQ2 How can realistic grid models be generated with minimal available data sources?
We introduce a new automated two-stage optimization-based distribution grid modeling method that relies solely on openly available map data. Its modular design, consisting of a load estimation process and the subsequent grid layout optimization, allows adapting the method to the available data sources.

RQ3 How does the availability of data influence the accuracy of the generated grid models?
To assess the effect of available data sources on automated grid modeling, we first propose metrics to quantitatively compare distribution grid models. We compare models generated from different data sources and demonstrate that widely available 3D building and map data is an adequate source to automatically generate grid models of residential areas.

Part III

Network Equivalencing

Overview Part III

Changes in the consumption and especially in the generation of electric energy pose challenges to the transmission grid. Regions with a high availability of renewable energy sources might not be close to regions with high demand, such as large cities and energy-intensive industry, requiring the transmission of energy over large distances. Furthermore, the volatility of renewable energy sources can lead to grid stability and supply issues. Identifying these issues and developing solutions requires simulations of the transmission grid. However, simulating detailed models of these large systems is often too computationally expensive, requiring simplified equivalent models. Chapter 5 provides an overview of model equivalencing techniques and identifies the research gap to address the aforementioned challenges.

Static grid simulations are important for finding congestions and possible solutions, and for optimizing grid operation. With an increasing share of renewable energy sources in the generation mix, weather conditions become increasingly influential for the calculation of power flows. Weather conditions, in turn, heavily depend on geography, making geographic information of energy sources an essential factor for simulations. Established power grid reduction methods, however, do not consider this type of information. Thus, in Chapter 6, we answer research question **RQ4** that addresses retaining geographic information in reduced equivalent models for power systems.

While static grid simulations are important, they are not sufficient to investigate and solve all issues in transmission grids. Developing and implementing solutions for the identified weaknesses in the grid demands dynamic simulations. Implementing new HVDC transmission lines, for example, requires the development of converter controls, that need to be evaluated using dynamic simulations, often in HiL setups. Therefore, in Chapter 7, we focus on **RQ5** that addresses the creation of reduced equivalent models for dynamic power system simulations.

5 Background on Power System Equivalents

Models of systems are used in all kinds of domains, from various energy domains, such as electrical, mechanical, and thermal [85], to chemical and biological systems [86], and communication networks. When these systems reach a certain complexity, it might still be possible to model them accurately, but these accurate models might be too complicated and thus too computationally expensive for practical use cases [87]. Thus, it is always preferable to reduce the model complexity to a practical point at which the model still represents the behavior of the system that is crucial for the specific use case. Electrical transmission grids are one such domain of highly complex dynamic systems. Model equivalents with a reduced complexity are needed to enable the necessary simulations of transmission grids to overcome the challenges caused by the transition to carbon-neutral energy generation.

This chapter provides the background on power system equivalents and the methods and concepts to create such equivalent models. It starts by introducing general information on equivalent models in Section 5.1 before summarizing the literature and categorizing the vast research field of methods for power system equivalencing in Section 5.2. The insights of this summary are used to filter the relevant methods for the state of the art in Section 5.3. Lastly, this chapter identifies the research gap in Section 5.4, that is addressed in the subsequent chapters of this part.

5.1 Configurations of Reduced Power System Models

Two fundamental configurations for equivalents can be found in the literature [88, 28]. The first configuration divides the model into an internal and an external subsystem as shown in Figure 5.1b. This configuration is the standard for simulations that focus on the electrical properties of the system. The internal subsystem is defined by the area of interest (AoI) and contains the components that are most relevant for the research questions. While the external system plays a minor role in the simulation setup, it still influences the AoI and can not be omitted entirely. Available system information is often an important factor for using this configuration, as the

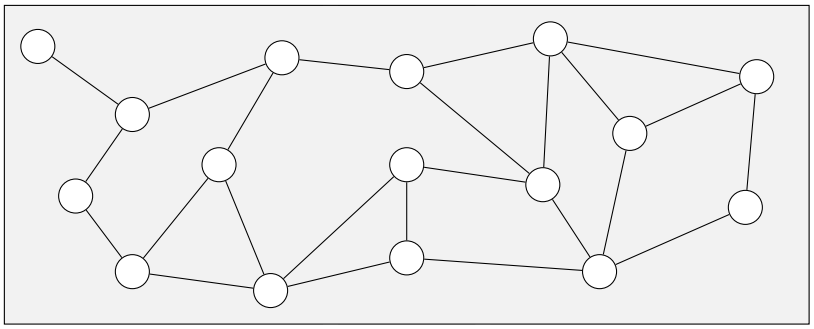
external subsystem of a model might be operated by a third party not wanting to disclose details important for the model. Thus, a simplified version of these parts of the system is often used instead. A variation of this configuration contains a buffer or boundary subsystem between the internal and external subsystems with a medium level of detail. The second fundamental configuration only consists of one system that is reduced as a whole, see Figure 5.1c. This configuration is more prominent in optimization problems for systems that do not have a particular AoI, but are prohibitively large to be optimized in detail [28].

5.2 Categories of Model Equivalencing Methods

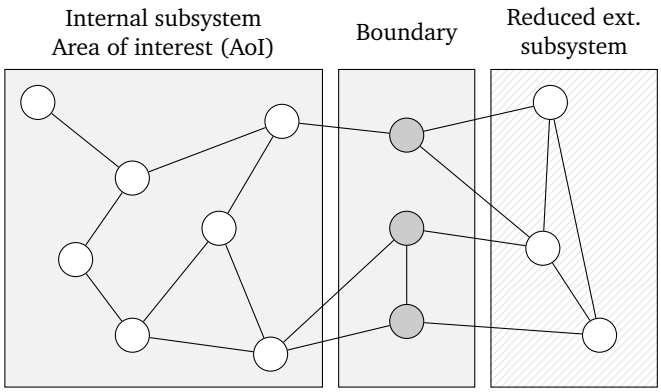
The divers landscape of power system model types and use cases, as introduced in Chapter 2, and configurations for equivalent models results in a variety of different views on reduction techniques. In this section, we briefly summarize the categories for model equivalencing described in five literature sources with varying modeling and simulation focuses [88, 87, 89, 90, 28]. From this literature overview, we derive properties of equivalencing approaches for a common classification before describing general benefits and drawbacks of equivalencing methods.

In the first literature source, Machowski et al. [88], view model equivalents from a perspective of stability and control, and thus focuses on the dynamic electrical properties of the modeled systems. On the highest level, they distinguish the equivalencing techniques by their requirement for system knowledge. While methods requiring no system knowledge are usually data-driven and utilize measurements to create an equivalent model, Machowski et al. focus on techniques that do require system knowledge. They use this criterion to define these techniques as **model reduction techniques**, dividing them further into three categories: first, **physical** reduction, which involves selecting appropriate models for the system components, such as generators and loads. Second, **topological** reduction, which involves eliminating and aggregating nodes to reduce the complexity of the network and the number of system components. Third, **modal** reduction, which linearizes the external subsystem model to ignore the unexcited modes.

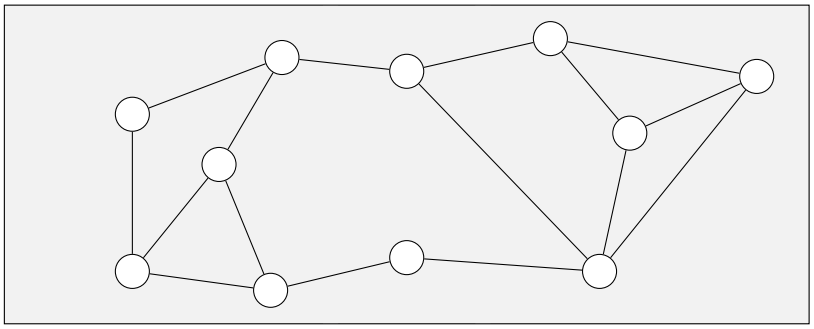
In the second source, Đukić and Sarić [87], give an overview of dynamic model reduction as a whole, not limited to power systems, before diving deeper into the specific application on power systems. In general, they distinguish the model reduction methods by their applicability on linear or nonlinear systems. For the reduction of dynamic power system models, they identify five common concepts.



(a) Full system model



(b) Focus on the AoI with a reduced external system



(c) Reduction of the whole system

Fig. 5.1.: The two basic configurations to create reduced equivalents of a fully detailed power system model Figure 5.1a. With the focus on an AoI that is kept at full detail, Figure 5.1b, reduces the external system that is less interesting for the problem at hand. An explicit definition of a boundary or buffer subsystem is optional. Figure 5.1c does not distinguish between different parts of the system. The whole system is simplified with the same reduction method.

The first concept is generator **coherency**, which describes how similar multiple generators respond to disturbances. Multiple generators are coherent under a given disturbance if they oscillate with the *same angular speed, and terminal voltages in a constant complex ratio* [87]. The second concept, **synchrony**, is a less restrictive extension coherency defined on a subset of modes v. Instead of demanding state variables to oscillate together identically, as is the case for coherency, v-synchrony only demands that the variables *swing perfectly in-phase or perfectly out-of-phase with a known constant of proportionality, whenever only v is excited* [91]. The third concept, **singular perturbations analysis (SPA)**, in the context of power system modeling, exploits the different time scales of events in a power system by creating a fast and slow subsystem. These subsystems can then be analyzed in separate time scales. The fourth concept, **modal analysis**, reduces the complex nonlinear power system model to a *linear form of state equation representation* [88]. The fifth concept, **identification** methods, assume a structure for equivalent model that aims to represent the power system and are based on data, such as measurements usually in the internal system or at the boundary of the two subsystems.

Chow [89] has a similar scope as the previous source, focusing on power system dynamics, but proposes other categories to differentiate the model reduction techniques. The first category, **coherency and aggregation**, combines the previously introduced concept of coherency with the aggregation of generators and nodes, as introduced in the topological reduction in [88]. The second category views the reduced system as an **equivalent input-output model** that can be further divided into linear and nonlinear models. This category contains methods, such as modal analysis and SPA as introduced in [87]. The last category contains **phasor measurement-based methods** that utilize these measurements to estimate the mode shapes of the system to be equivalenced.

In the fourth source, Sistermanns et al. [90] view model reduction from a perspective of applying it to speed-up OPF calculations, thus focusing on power flows and congestions in the whole system and ignoring any dynamic behavior. They identify two categories of equivalents. The first category, termed **electrical equivalencing**, is characterized by using the internal-external subsystem configuration (see Figure 5.1b) and aiming to reproduce the electrical behavior of the external subsystem. The second category, **market-based reduction**, applies aggregation based on the OPF results with the goal of preserving the power flows at the retained buses.

In the fifth source, Frysztacki et al. [28] focus on large-scale power system models for the optimization of renewable energy sources on the scale of countries and continents. They are interested in power flows and congestions in the whole modeled

area without defining any specific areas of interest. As spatial properties are crucial for the accurate modeling of renewables, they consider methods that reduce high-resolution models spatially to smaller equivalent models. They distinguish these reduction methods by the features they focus on. The first category focuses on the **network representation** and utilizes the spatial or electrical distance between nodes for aggregation. The second category focuses on **renewable variability** and applies clustering methods on databases of energy generation and demand while often using a synthetic or very simplified transmission grid model. As the present thesis focuses on the reduction of transmission grid models, this second category will not be addressed further.

With this wide range of approaches to model equivalencing and the diverse views on categorizing and classifying the methods utilizing these approaches, Table 5.1 provides an overview of methods and concepts found in the literature. The table groups specific methods by indenting them under more general methods or concepts, e.g., presenting modal truncation as a specific method of modal analysis. For a quick assessment of the methods, Table 5.1 provides three general properties:

- **Requirement of system knowledge:** Proposed by [88], this category describes whether a method requires system knowledge, i.e., a model of the system, or is data-driven, usually utilizing measurements of relevant variables in the system. Only methods requiring system knowledge are considered as **model reduction**.
- **Static or dynamic modeling:** An important factor in distinguishing equivalencing techniques is whether they are applied to static or dynamic models as these models often serve to answer completely different questions. Use cases such as system stability require dynamic models and need reduced model complexity because dynamic simulations are computationally expensive. Other use cases, such as power system optimization, perform static power flow calculation, but require reduced models as they need to compute many possible system configurations.
- **Features of interest:** Depending on the use case, equivalent models aim to represent different features of the system they replace. For dynamic simulations, these are usually **electrical** properties, such as bus voltages and current flows. Other use cases might focus more on **market-relevant** properties, such as generator dispatch, power flow, or congestions.

Naturally, the methods and concepts presented in Table 5.1 all have their benefits and drawback, that are more or less important depending on the use case. Methods

based on **coherency and aggregation** have several advantages that make them the most commonly used methods in practice [89]. These methods preserve the data structure of the model [89], as aggregated components are represented by standard models of physical components [87]. This leads to a high compatibility of the method and does not require specialized interfacing methods between the retained and reduced subsystem [87]. However, these methods are purely empirical and lack a rigorous mathematical foundation [87]. Furthermore, the coherency of generators depends on the condition of the system and might be invalid for conditions that were not considered during the reduction process [87].

Modal analysis-based methods have several drawbacks that result in their rare use in practice. These methods create equivalent models as a set of equations that might not be representable with standard models of power system components, requiring changes to the modeling software [88]. Furthermore, these equations are hard to interpret as they do not represent physical units in general [87]. These drawbacks are also shared by **surrogate models**, e.g., ANN-based equivalents, that might be difficult to integrate into common modeling software, requiring a special interface or co-simulation, and are generally not easy to interpret.

Tab. 5.1.: A review of power system equivalencing methods.

Method / Concept	System knowledge required?	Static / Dynamic	Features of interest	References
Coherency	Yes	Dynamic	Electrical	[92, 93, 94]
Synchrony	Yes	Dynamic	Electrical	[95, 96]
Singular perturbations analysis (SPA)	Yes	Dynamic	Electrical	[97, 98]
Modal Analysis	Yes	Dynamic	Electrical	
Modal Truncation	Yes	Dynamic	Electrical	[99]
Selective Modal Analysis	Yes	Dynamic	Electrical	[100]
Identification	No	Dynamic	Electrical	[101]
Krylov / Moment Matching	Yes	Dynamic	Electrical	[102]
Topological Reduction	Yes	Static	Electrical	
Node elimination by Ward	Yes	Static	Electrical	[103]
Node aggregation by Dimo (REI)	Yes	Static	Electrical	[104, 93, 94]
Node aggregation by Zhukov	Yes	Static	Electrical	[88]
Kron reduction	Yes	Static	Electrical	[105, 106]
PTDF-based reduction	Yes	Static	Market	[107]
LMP-based reduction	Yes	Static	Market	[90]
Graph clustering	Yes	Static	Various	[28, 90]
Surrogate Models	No	Both	Various	
Artificial neural network (ANN)	No	Dynamic	Electrical	[108]

A similar drawback is also present for **SPA**-based models. Simulating different subsystems in various time scales is not uncommon for power system simulation, especially when EMT simulations are required for very detailed analysis of specific parts of the system. Other parts can then be represented as a much less computationally expensive RMS simulation at a lower time resolution [109]. This is, however, mostly done in co-simulation setups and is rarely supported by a monolithic simulation of a power system.

Identification-based methods have the advantage of not requiring detailed system knowledge [87]. However, selecting an adequate model structure for the equivalent might be challenging. In the scope of this thesis, these methods are not examined further, as the thesis focuses on the reduction of available power system models.

Topological reduction methods, by default, create equivalent models for steady-state analysis [88]. However, the methods based on electrical features, such as Ward [103], radial, equivalent, and independent (REI) [104], and Zhukov [110], are often combined with coherency-based methods for dynamic equivalents. These methods are also often readily available in common modeling software, as e.g., in DIgSILENT PowerFactory. Nonetheless, all these methods come with their individual strengths and weaknesses. The branches created by the **Ward** reduction might have a poor X/R ratio that can cause convergence problems for load flow calculations [88]. The **REI** method introduces a large number of fictitious branches between the retained nodes that can also take negative admittance values, which might be problematic for some simulation tools [88]. While **Zukov's** reduction method avoids creating fictitious branches between retained nodes, the resulting admittance matrix is not necessarily symmetric, which can also be problematic for the simulation [88]. Other topological reduction methods, such as the **market-based** reductions and **clustering-based** methods, might result in very different electrical properties of the reduced models, depending on the features the methods focus on and the clustering algorithm applied [28]. While these methods can be beneficiary to preserve some aspects of the reduced models, such as renewable generation [28],—to the best of the author's knowledge—these market- or clustering-based methods have not been used for dynamic simulations.

Finally, the practical applicability of model equivalencing methods needs to be considered. While most methods based on control theory have a rigorous mathematical foundation, in practice, the empirical methods based on coherency and aggregation are generally preferred as they represent physical models and are thus easier to handle [87]. Table 5.1 distinguishes methods that do or do not require system knowledge. More in-depth considerations, however, reveal that, these requirements can be

vastly different. While some methods only require the overall model structure and parameters, others might require access to the underlying equations. For example, the Ward, REI, and Zhukov reduction methods operate directly on the admittance matrix of the system which is not easily available in many common modeling tools. Thus, might be impractical to develop on top of such a tool.

5.3 Previous Works on Power System Equivalencing

This section describes previous works and the state of the art for power system equivalencing. The methods described in this section extend, advance, specialize or combine the fundamental approaches introduced in Section 5.2 to improve the resulting equivalent models or adapt them to specific use cases.

Traditional network reduction techniques, such as the (extended) Ward reduction [103, 111] or the REI method [104], are commonly used in the industry for the generation of static network equivalents. These methods are tested thoroughly and are available in many common modeling and simulation suites. Being the industry standard for static network equivalents, these methods are also adapted for specific use cases. For example, [112], combines two Ward reductions with different sets of retained buses: one with only the internal system retained, another with retained generator buses. The second one is used to move the external generators to boundary buses of the first—more reduced—equivalent using the shortest path of electrical distance. The goal of this approach is to prevent fractioning the external generators. These traditional methods for static equivalents can be combined with other methods to create equivalents for dynamic simulations. For example, [92] introduces methods to aggregate generators and the corresponding controls based on aggregated transfer functions of coherent generator groups. In a newer development, [93, 94] combine the static REI reduction as provided by DIgSILENT PowerFactory[1] with coherency-based generator grouping and parameter estimation with genetic algorithms. This results in dynamic equivalents that can be used for real-time simulations, once converted to the RSCAD[1] model format for RTDS[1] real-time simulators.

In [113], a reduction method based on matrix reduction is introduced, that is fast enough to create and deploy equivalents for power flow calculations on the fly. Integrating this method in the power flow calculation algorithm, it could be viewed as a speed-up of the calculation algorithm itself.

[1] See Section 2.2 for information on common modeling and simulation tools.

Other methods focus more on specific types of power systems, for example distribution grids. The method introduced in [114] is focused on the reduction of distribution feeders and uses a ruin and reconstruct approach with Monte Carlo simulations in OpenDSS[2]. In [115], a method used for active distribution networks is presented, utilizing ANNs and identification for dynamic equivalents. After clustering the system states, an ANN is trained on recorded or simulated events for every cluster to estimate robust parameters for a variable-order dynamic equivalent model. An recurrent neural network (RNN) architecture representing the active components in a distribution grid while the passive components are aggregated at the boundary of the subsystems is proposed in [108]. The method uses the last N voltage values at the boundary as input and returns the corresponding current value. In [116], an RNN is combined with an autoencoder for obfuscation to enhance the privacy protection and facilitate model sharing. The method creates a static equivalent and is compared against the REI reduction integrated in PowerFactory, where it shows better results for changing system conditions.

A dynamic reduction method is proposed in [117], which yields a purely synthetic equivalent by placing a given number of generators in a reduced network and performing a parameter optimization to match the equivalent system behavior to the original. Another identification method for dynamic equivalents is presented in [118], which estimates the system's transfer function directly.

Other methods focus more on market-relevant system properties, such as congestions and generator dispatch. A method for static network equivalents that aims at preserving the congestion profiles of the grid based on power transfer distribution factors (PTDFs) is presented in [119]. Market-based information, in this case local marginal prices (LMPs), are also considered for network reduction in [90]. Additionally, this method utilizes the topological grid structure and electrical distance as criteria to reduce network models for comprehensive power flow studies, specifically avoiding the creation of fictitious branches between the retained nodes.

For EMT simulations, the frequency-dependent network equivalent (FDNE) is a common approach to create wideband multi-port equivalent models, implemented, e.g., in PSCAD[2] and RSCAD [120]. The FDNE is a multi-port admittance matrix whose elements are rational polynomials that can be determined by fitting the frequency response of the FDNE to the original model [121]. These FDNE blocks are used to represent the high-frequency characteristics of the system and can be combined with phasor-based (RMS) models for the slower electromechanical behavior [121].

[2] See Section 2.2 for information on common modeling and simulation tools.

A reduction method commonly used in circuit theory is the Kron reduction [106]. It eliminates nodes by adjusting the row and column in the admittance matrix corresponding to the connections of the node to eliminate. In [106], this method is viewed from the perspective of algebraic graph theory and applied to power flow calculations and transient stability assessment (TSA).

Lastly, [28] compares several graph clustering and aggregation methods with an application on continent-scale power flow calculations. It is motivated by unsatisfying results of the classic Ward reduction method for systems with large shares of renewable generation. The compared methods include clustering based on political borders, capacity factors, time series of capacity factors, electrical distance, and geospatial k-means.

5.4 Research Gap

Reviewing the literature on power system equivalents reveals four desirable properties of equivalents that are crucial for solving the challenges of modern transmission grids. The need for these properties mostly comes from the requirement of an adequate representation of renewable energy sources in modern grids, which are heavily influenced by the weather and, thus, geography. In the following, we describe these four properties and why they are needed in modern power system equivalents.

Physical Model Having an equivalent consisting of models of physical units, e.g., generators, controllers, lines, has several advantages compared to equations that do not represent physical units, as is often the case with modal analysis-based equivalents [87]. Firstly, in a simulation scenario that aims to retrieve insights from the equivalent model—as is usually the case for configurations without a specific area of interest—the interpretability of variables in the equivalent is crucial. Without the representation of physical units, this interpretability is significantly deteriorated. Secondly, most available modeling and simulation tools are based on physical models. An equivalent representation compatible to the prevailing data structures of common tools is important for the practical feasibility of a model reduction method.

Mapping Between Components in the Original and in the Reduced Model In addition to a representation of the equivalent with physical models, a mapping of components between the original model and the reduced model is important to gain insights from simulations involving the equivalent. While some methods purposefully obfuscate the equivalent in order to hide sensitive information when sharing

the model with other entities [116], a clear relation between components in the detailed and reduced model is often desired [90]. Such a relation is often unclear in traditional reduction methods, such as the Ward and REI reduction, that tend to introduce fictitious branches between the retained nodes without a specific relation to branches in the detailed model [88].

Preservation of Geographic Information While geographic information on power system components has no direct effect on the simulation results, they can become a deciding factor in studies of systems with large shares of renewable energy generation [28]. These studies often determine the energy generation of PV systems and wind turbines depending on weather data or general formulas depending on the plant location. Geographic properties are traditionally not considered in model reduction methods that focus on the electrical properties of the equivalent. Methods that do consider geographic properties, however, so far only focus on static simulations, such as OPF calculations and long-term power flow studies.

Support for Static and Dynamic Simulations The traditional view for dynamic equivalents is that of an external system that is either not interesting for the use case—except for its influence on the AoI—or not known in detail. In both cases, these simulations usually do not aim to gain insights into the external system and focus only on the AoI. Without this goal, the previously described equivalent properties become less important to some extent. Some reduction methods that focus on static simulations, such as [90, 28], partly do consider the previous properties. However, dynamic simulations are a crucial part in the integration of new grid components, such as HVDC lines and grid boosters, especially considering the volatile nature of renewable generation.

A review of existing model equivalencing methods shows that currently no method meets the requirements to have all four properties. This largely stems from the fact that geographic information is traditionally not considered in methods for creating equivalents that are suitable for dynamic simulations. As the combination of the described properties has significant potential for new grid simulations, the fulfillment of all four properties describes the research gap and goal for the next chapters in this part. Chapter 6 focuses on the first three properties, by creating an equivalent for static simulations of physical units, considering geographic information for the reduction process, and enabling a mapping between components by using explicit aggregation operations. Chapter 7 builds on this static reduction by introducing a method to also approximate the dynamic behavior of the original system.

6 Static Network Reduction

Parts of this chapter are reproduced from

Moritz Weber, Hüseyin K. Çakmak, Uwe Kühnapfel, and Veit Hagenmeyer. "A Novel Method for Topology Preserving Static Network Reduction". In: *2023 IEEE 11th International Conference on Smart Energy Grid Engineering (SEGE)*. Aug. 2023, pp. 58–63. DOI: 10.1109/SEGE59172.2023.10274576.

The transmission grid faces several challenges that require a wide range of different simulation types to solve them. However, the complexity of country- or even continent-sized transmission grids can be prohibitive for many studies relying on simulations. Thus, models with reduced complexity are required. As the issue of computational limitations for power system models is an active research topic, the previous chapter provides an overview of common approaches and the state of the art of creating reduced-size model equivalents. The research gap identified in Section 5.4 describes the requirements for model reduction methods that can be used to tackle the challenges of modern transmission grids with high shares of renewable energy systems. Besides the main requirement of reducing the computational demands of the model simulation, the reduced model has to meet various other criteria depending on the planned use cases. In this chapter, we focus on three of the four identified criteria:

- The reduced model has to be a physical model. Additionally, the model shall consist of standard components to ensure compatibility between different modeling formats.

- A mapping between components in the original model and the reduced model has to be possible.

- Relevant geographic information has to be preserved. This includes the preservation of the distance between boundary buses and the basic topology of the network.

While the proposed approach does aims to preserve load flow and short circuit properties at the boundary buses, i.e., the electrical properties for static simulations, it does not consider the dynamic system behavior.

In this chapter, we introduce a novel network reduction method that focuses on the preservation of the topology of the power grid. In doing so, we answer research question **RQ4** that addresses the importance of preserving geographic information in network equivalents with considerable amounts of renewable generation. Furthermore, the proposed topology-preserving network reduction (TPNR) method yields network equivalents that only consist of standard component models. These two properties result in the following benefits: The geography of the resulting models is more interpretable compared to traditional methods and can be visualized meaningfully. This interpretability enables manually editing the reduced models intuitively, which is not possible with established reduction methods. Hence, the TPNR equivalents are much more versatile, combining the benefits of a detailed AoI and a reduced external system. Preserving the structure also avoids densely connected equivalent networks that are often the result of Ward and REI reductions. Furthermore, roughly preserving the distances between nodes in the grid also preserves the line propagation times, which can become important for EMT studies.

The remainder of this chapter is structured as follows: In the next section, we describe our methodology and implementation. In Section 6.2, we evaluate the proposed method by comparing it to established reduction methods. We discuss the results of the evaluation in Section 6.3. Lastly, Section 6.4 concludes this chapter and describes possible directions for future work.

6.1 Method: Graph-based Topology Preserving Network Reduction

In this section, we start by describing the fundamental ideas behind the proposed topology-preserving network reduction method. Then we provide a more detailed look into the reduction operations, which are the core of the proposed method, and finish with the description of the resulting reduction algorithm and implementation details.

The fundamental idea behind the TPNR method is representing the power grid as an undirected multigraph $G = (V, E)$ and focusing the reduction operations on the

topology of this graph. In the network graph G, the nodes V represent the buses of the power grid, while the edges E represent the branches, e.g., overhead lines, cables, and transformers. In the power grid model, each bus can be connected with various power system components, such as loads, generators, or shunts. These components are not represented in the simplified graph model, as they are not considered relevant to the structure of the system. When referring to the components of a node n, in the following, we refer to the power system components that are connected to the bus represented by n.

Based on this simplified graph representation of the power grid, e.g., as shown in Figure 6.1, we make three observations for potential reductions of the graph complexity. From these observations, we derive operations $\mathbf{O}$ to reduce the graph while keeping its structural integrity:

- Observation 1: Parallel edges between two nodes do not contain significantly more information than a single edge between these nodes. Parallel transmission lines or cables are common in power systems. They likely have very similar properties, e.g., length and transmission capacity, and could be approximated with a single edge component. This leads to operation $\mathbf{O_1}$: replace multiple parallel edges between the same nodes with one equivalent edge.

- Observation 2: A node n_i with only one neighbor n_j affects the network at a single point—the node it is connected to. Therefore, if the buses of both nodes have the same voltage, node n_i likely does not provide significant structural information. This leads to operation $\mathbf{O_2}$: incorporate nodes n_i with only one neighbor n_j into this neighbor, i.e., remove n_i, and move the connected components from n_i to n_j.

- Observation 3: Similar to the previous observation, a node n with two neighbors does not provide much significant structural information and influences the system at only two points. This leads to the idea for operation $\mathbf{O_3}$: connect the two neighboring nodes directly and move the components from n to its neighbors appropriately.

From a graph theory perspective, operation $\mathbf{O_1}$ can be viewed as an edge elimination, as all but one of the parallel lines are removed from the graph. Operations $\mathbf{O_2}$ and $\mathbf{O_3}$ can be viewed as edge contractions, i.e., merging two adjacent nodes and deleting their connecting edges. In the case of $\mathbf{O_3}$, this contraction would be applied to the node with two neighbors (n_5 in Figure 6.1) and one of its neighbors. With these graph-theoretic operations, the reduced graph is essentially a minor of the original graph.

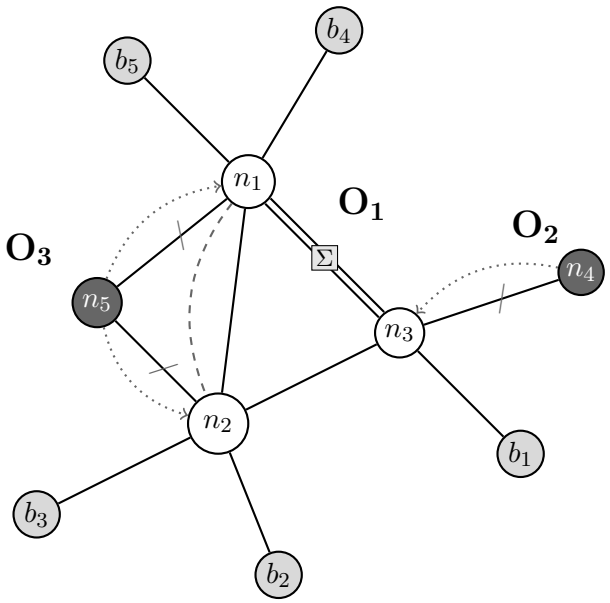


Fig. 6.1.: An example for a reduction of a simple network graph. The gray nodes represent the boundary buses, while the green nodes are internal buses. The node size indicates the number of components connected to a bus, and thicker edges indicate multiple edges between two nodes. The arrows and crosses indicate the actions performed by the reduction operations.

6.1.1 Grid Manipulation Operations

The graph representation of the power grid is a significant simplification of the whole system. Thus, we need to find equivalents of the three graph reduction operations introduced in the previous section to adapt the actual power grid model. In order to transfer the graph changes to the power grid, we identify five necessary operations: aggregation of parallel lines, removing buses and lines, moving components from one bus to its neighbors, lengthwise concatenation of lines, and aggregation of components connected to the same bus. We describe these five operations in more detail in the following paragraphs.

Aggregation of parallel lines This grid manipulation operation is a direct implementation of $\mathbf{O_1}$. Assuming a typical modeling of transmission lines with π-sections (see Figure 6.2), the electrical parameters of a transmission line are given by the following equations,

$$Z = R + j\omega L \tag{6.1}$$

$$Y = G + j\omega C, \tag{6.2}$$

where (6.1) describes the series impedance of the line, and (6.2) describes the shunt admittance. To aggregate those line parameters, we use the following equations,

$$x_{total} = \left(\sum_{l \in \text{Lines}} x_l^{-1} \right)^{-1} \tag{6.3}$$

$$x_{total} = \sum_{l \in \text{Lines}} x_l \tag{6.4}$$

where x_l is the line parameter to be aggregated. While (6.3) is used for the impedance values, R and L, (6.4) is used for the admittance values, G and C. For the length of the aggregated line, we use the mean length of all parallel lines.

Removing buses and lines In order to implement operation $\mathbf{O_2}$ in the underlying power grid model, buses and lines need to be removed from the system. When removing a bus with only one neighbor, we assume that it is not connected to any more components, which is achieved by moving the components to this neighbor first. We perform a power flow calculation to detect any flows over the transmission lines to the empty bus that is to be removed. In the case of any power flows, we place a shunt element on the neighboring bus with an equivalent power flow. In the last step, the bus and the connecting lines are removed from the system. The adjustment for power flows to or from the eliminated bus are likely not exact for all load and generation scenarios. Thus, they might introduce errors when simulating different situations.

Moving components from one bus to its neighbors In order to apply $\mathbf{O_3}$ to the grid model, grid components need to be moved from one bus to its neighbors. To move components, such as loads and generators, from a bus b_i to its neighbors N_i, we need to determine, how to split up those components. The basic idea is that neighbors connected via a higher admittance (and thus often over a shorter distance) are

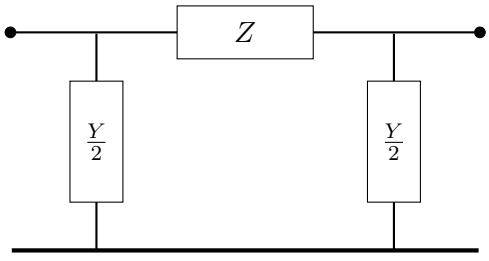


Fig. 6.2.: A so-called π-section model, which is often used to model transmission lines and cables. For simplicity, this figure shows a single-phase model, but the same aggregation principles apply to three-phase models.

affected more by the components than those connected via a lower admittance, i.e., longer distance. Therefore, we first determine the pairwise admittance values, Y_{ij} for $b_j \in N_i$, between the buses. In order to set these pairwise admittances into relation, we first sum up those individual admittance values to the total admittance,

$$Y_{total} = \sum_{b_j \in N_i} Y_{ij}. \tag{6.5}$$

Next, we calculate the relative admittance values,

$$\tilde{Y}_{ij} = Y_{ij} / Y_{total}, \tag{6.6}$$

and use these to scale the properties of the moved components accordingly. For example, the active and reactive power set points, P and Q, of a load or generator are multiplied with the relative admittance to split the component accordingly between the neighboring buses. This adjustment is generally applied to parameters that determine the scale of a component. Other parameters, such as the voltage set point of a generator, may need a different treatment. In that case, the voltage set point is changed to that of the neighboring buses. Further parameters do not need to be adjusted at all.

This operation might introduce errors into the simulation when significant line losses between bus b_i and its neighbors are present in the original model. Moving the components to the neighboring buses effectively skips the lossy transmission. This can have an effect on the power flows at those neighboring buses.

Lengthwise concatenation of lines The lengthwise concatenation of lines is essential for the incorporation of nodes with two neighbors, i.e., operation $\mathbf{O_3}$. Assuming the same line model as for the parallel aggregation, Equations (6.1) and (6.2), we use the summation (6.4) for all parameters in this case, i.e., R, L, G, and C. We sum up the individual line lengths and place the new equivalent line directly between the two neighboring buses. The bus between the two neighbors can then be removed together with the connecting original lines. While not yielding exact equivalent lines, this method is adequate in most cases with similarly parameterized lines. However, the concatenation of overhead lines and cables might lead to unwanted deviations and is therefore disabled by default.

Aggregation of components connected to the same bus While the aggregation of components does not affect the topology of the network, it reduces the overall complexity of the model. In this operation, we replace all components of the same

type that are connected to the same bus with one equivalent, e.g., all loads are aggregated to one component and all synchronous machines are aggregated to another component. In the case of loads, for example, we sum up the individual active and reactive power values and create a new equivalent load with these values. Detailed information on the aggregation procedure of multiple components of the same type are provided in Appendix A.1.

One notable exception that does change the topology is the aggregation of synchronous machines with transformers, which are considered as one entity for this aggregation. In order to consider a transformer and a synchronous machine as a single entity, there must be no other components connected to the secondary bus of the transformer. Additionally, to aggregate multiple instances of these entities, the transformers connected to the same bus need to have same voltages. The identified transformers are aggregated as well as the connected generators.

For applications that require an even lower number of components in the model, we implement a more drastic reduction, which we call **TPNR+**. This TPNR+ variant performs all the same structural reduction operations as the standard TPNR method. However, as a final step, the TPNR+ variant aggregates all consumption and generation components, e.g., loads, generators, and shunts, connected to the same bus into a single component.

Boundary Equivalencing The goal of the boundary equivalencing is matching the load flows between the reduced system and the retained system as closely as possible and is inspired by [93]. For each transmission line connecting the internal system with the external system, the active (P) and reactive power (Q) are determined for the fully detailed model on the side of the internal system. These are the target values for the equivalencing process. To achieve these target values, additional loads are placed at the corresponding buses of the external system. Iteratively, the load flow results of the reduced system ($\hat{P}$ and $\hat{Q}$) are determined and the difference between the target values and these results are added to the newly created loads. This process is repeated until the maximum number of iterations is reached, or the specified deviation is met.

6.1.2 Implementation

The graph and power grid manipulation operations described above are implemented in Python, utilizing the NetworkX package [122] for the graph operations and

Algorithm 2 Topology-preserving network reduction (TPNR) algorithm.

```
1:  procedure TPNR_REDUCTION(
        model, ▷ original model of the power system
        drastic ▷ whether to drastically aggregate components
    )
2:      AGGREGATE_ALL_BUSES_INDIVIDUALLY(model)
3:      repeat
4:          changed ← AGGREGATE_PARALLEL_LINES(model)
5:          changed ← changed ∨ REMOVE_NODES_WITH_DEGREE_1(model)
6:          changed ← changed ∨ REMOVE_NODES_WITH_DEGREE_2(model)
7:      until ¬ changed

8:      BOUNDARY_EQUIVALENCING(model)

9:      if drastic then
10:         AGGREGATE_BUSES_DRASTICALLY(model)
```

DIgSILENT PowerFactory[1] for the power system modeling. The individual operations are combined into an iterative algorithm that is described in Algorithm 2 on a high level. After aggregating the components at all buses (Line 2), the three basic graph operations $\mathbf{O_1}$ to $\mathbf{O_3}$ are executed iteratively as long as there are still any changes possible (Lines 3 to 7). Following this iterative reduction, the boundary equivalencing is performed to match the load flow results at the boundary buses more closely. Lastly, the optional drastic aggregation of loads and generators for the TPNR+ variant is performed.

6.2 Evaluation

We evaluate our method using a PowerFactory model of the German transmission grid and perform a reduction of zone 23 of said grid, which is shown in Figure 6.3. The model consists of the $380\,\mathrm{kV}$ and $220\,\mathrm{kV}$ levels and contains $78\,\mathrm{buses}$, $129\,\mathrm{edges}$ (lines and transformers), and $369\,\mathrm{components}$ in zone 23. We compare our novel TPNR method with established reduction methods that are available in DIgSILENT PowerFactory. These methods include the reduction with extended Ward [103] elements, the REI reduction [104], and PowerFactory's "regional equivalent". First, we compare the sizes of the resulting models, then perform an electrical comparison with the original model and other reduction methods, and finally a topological evaluation of the reduced model in the following section.

[1] See Section 2.2 for information on common modeling and simulation tools.

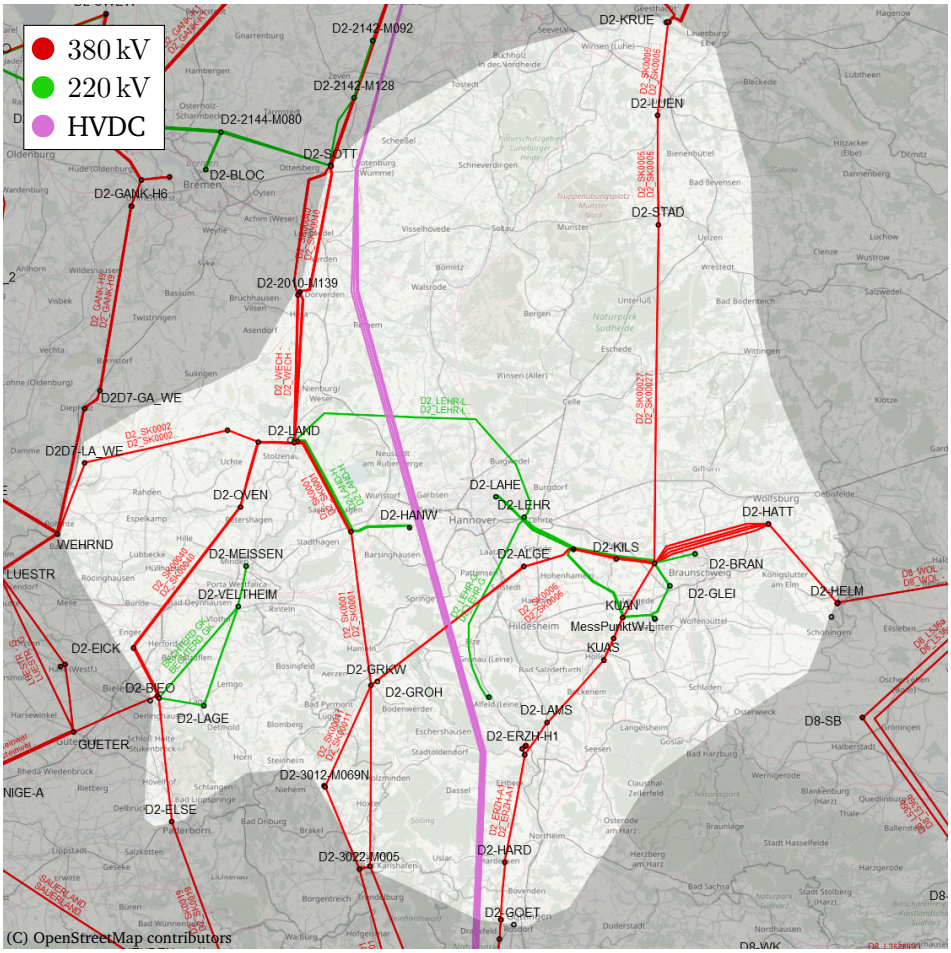


Fig. 6.3.: Geographic representation of control zone 23 of the German transmission grid, showing the substations and connecting transmission lines. It is important to note that substations can contain any number of buses and that parallel lines can be displayed as one single line. The HVDC line (pink) in the center of the grid is not connected to any of the buses in this part of the transmission grid.

Tab. 6.1.: The size comparison of the reduced models shows similar values for the regional equivalent and our TPNR method. Ward and REI result in a higher reduction of nodes and components while producing a larger number of edges.

	Original	Ward	REI	Regional Eq.	**TPNR**	**TPNR+**
Nodes	78	4	9	19	23	23
Edges	129	74	179	36	45	45
Components	369	16	19	30	88	29

6.2.1 Size Comparison

For the size comparison, we consider the number of nodes, edges, and components in the resulting models, which are shown in Table 6.1. While the Ward and REI reduction result in a very small number of nodes and fewer components compared to the other methods, they produce many edges between those nodes and the boundary. The regional equivalent and our TPNR method result in more nodes and fewer edges. The model reduced with TPNR includes 8 loads from the boundary equivalencing step that could also be aggregated with existing loads at the cost of potentially losing some transparency. In contrast to the TPNR method that retains many components, the TPNR+ variant reduced most of the consumption and generation at the same bus to one component—including the boundary equivalencing loads.

6.2.2 Electrical Comparison

To evaluate the electrical properties of the reduced models, we simulate the reduced models in PowerFactory and compare the results to the detailed original model. We examine the load flow results and three-phase short circuits according to IEC 60909 [123] at the boundary between the reduced portion of the grid and the surrounding retained model. These results include the active and reactive power flows, P and Q, over transmission lines connecting the boundary buses with the rest of the system, and the voltage magnitude and angle, U and φ, at the boundary buses. For the comparison of these results with the original values, we calculate the MAE and MXE (see Equations (2.1) and (2.2)). Additionally, to express the power flows and short circuit currents in relation to the absolute values, we consider the MAPE and MXPE as defined in Equations (2.3) and (2.4), respectively.

The results of these calculations are presented in Table 6.2, which shows the most accurate results for the Ward reduction. The load flow accuracy of the proposed TPNR method is right between the Ward and REI method for most metrics. However, the regional equivalent reveals a poor fit of the reduced model. Interestingly, the highly aggregated TPNR+ variation outperforms the more detailed default variation slightly for load flow accuracy. The analysis of the initial (I_k'') and peak (I_p) short circuit current shows that the TPNR method performs best for most metrics, only being outperformed by the Ward reduction for the $\mathrm{MAPE}_{I_k''}$ value. The main difference to the load flow results is the TPNR+ method performing worse due to the drastic aggregation, and thus the replacement of generators with loads.

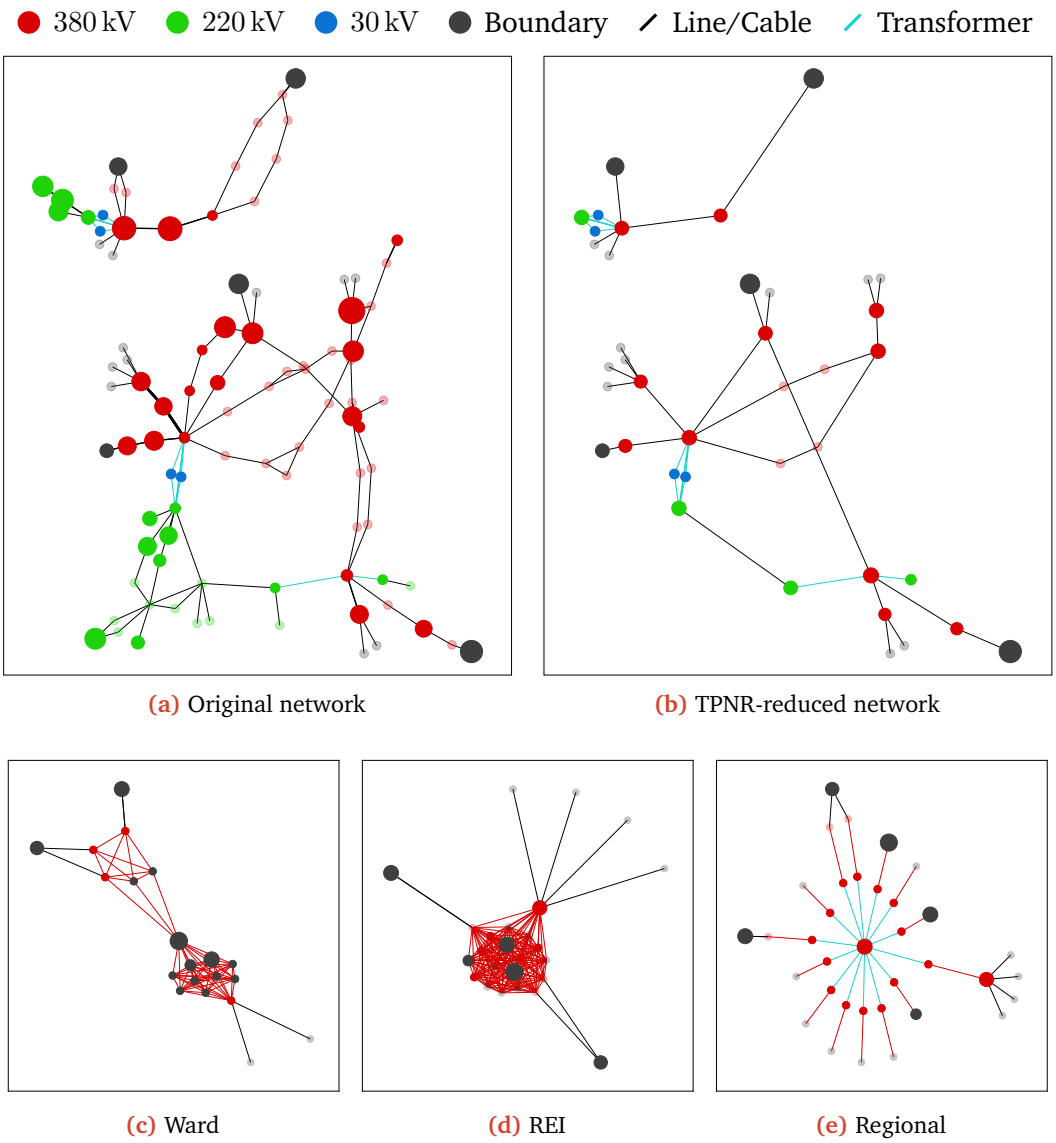
Tab. 6.2.: The errors of the equivalents at their boundaries to the rest of the transmission grid model. For most metrics, TPNR achieves results between Ward and REI, whereas the regional equivalent clearly shows the worst performance.

	Ward	REI	Regional Eq.	**TPNR**	**TPNR+**
MAE_P [MW]	0.0006	0.1822	83.9005	0.0205	0.0081
MXE_P [MW]	0.0020	0.3993	−394.5742	−0.0605	0.0167
MAE_Q [Mvar]	0.0001	0.7394	69.9561	0.1887	0.0727
MXE_Q [Mvar]	0.0003	3.2114	−254.2329	−0.7453	0.3960
MAPE_P [%]	0.0009	0.2505	150.7026	0.0574	0.0118
MXPE_P [%]	−0.0046	−0.9387	−794.0514	0.3950	−0.0617
MAPE_Q [%]	0.0018	5.9296	937.0254	2.6589	2.5401
MXPE_Q [%]	0.0405	106.0705	22379.7566	−40.9370	−63.7034
MAE_U [p.u.]	0.0000	0.0002	0.0167	0.0000	0.0000
MXE_U [p.u.]	0.0000	−0.0004	0.0365	−0.0001	0.0001
MAE_φ [°]	0.0000	0.0023	2.6627	0.0006	0.0006
MXE_φ [°]	0.0001	0.0051	5.2392	−0.0013	0.0010
$\mathrm{MAPE}_{I_k''}$ [%]	0.4311	1.0443	172.3591	0.5484	7.2034
$\mathrm{MXPE}_{I_k''}$ [%]	1.6571	1.5192	845.0421	1.2228	−12.7764
MAPE_{I_p} [%]	0.8586	1.4491	177.9390	0.6367	7.3920
MXPE_{I_p} [%]	−2.6900	5.2465	785.0565	1.3797	−13.3329

The loads created by the boundary equivalencing have a maximum active power of 1 MW, showing the good active power approximation of the reduction operations performed by the TPNR method. For the reactive power, the loads have a maximum of 7.25 Mvar, revealing a slight weakness of the reduction methods in handling reactive power. However, omitting the boundary equivalencing would not significantly alter the results overall.

6.2.3 Topological Comparison

The visual graph representation of the original and the TPNR-reduced model in Figure 6.4b shows the overall similar but highly simplified structure of the reduced model. On the other hand, the bottom row of Figure 6.4 shows the resulting graphs of the traditional methods that entirely change the topology of the network. Figure 6.5 shows the corresponding geographic representation of the five models. It illustrates that not only the structure is preserved by TPNR, but also the geographic information. The nodes and lines in the reduced system clearly resemble their



(a) Original network

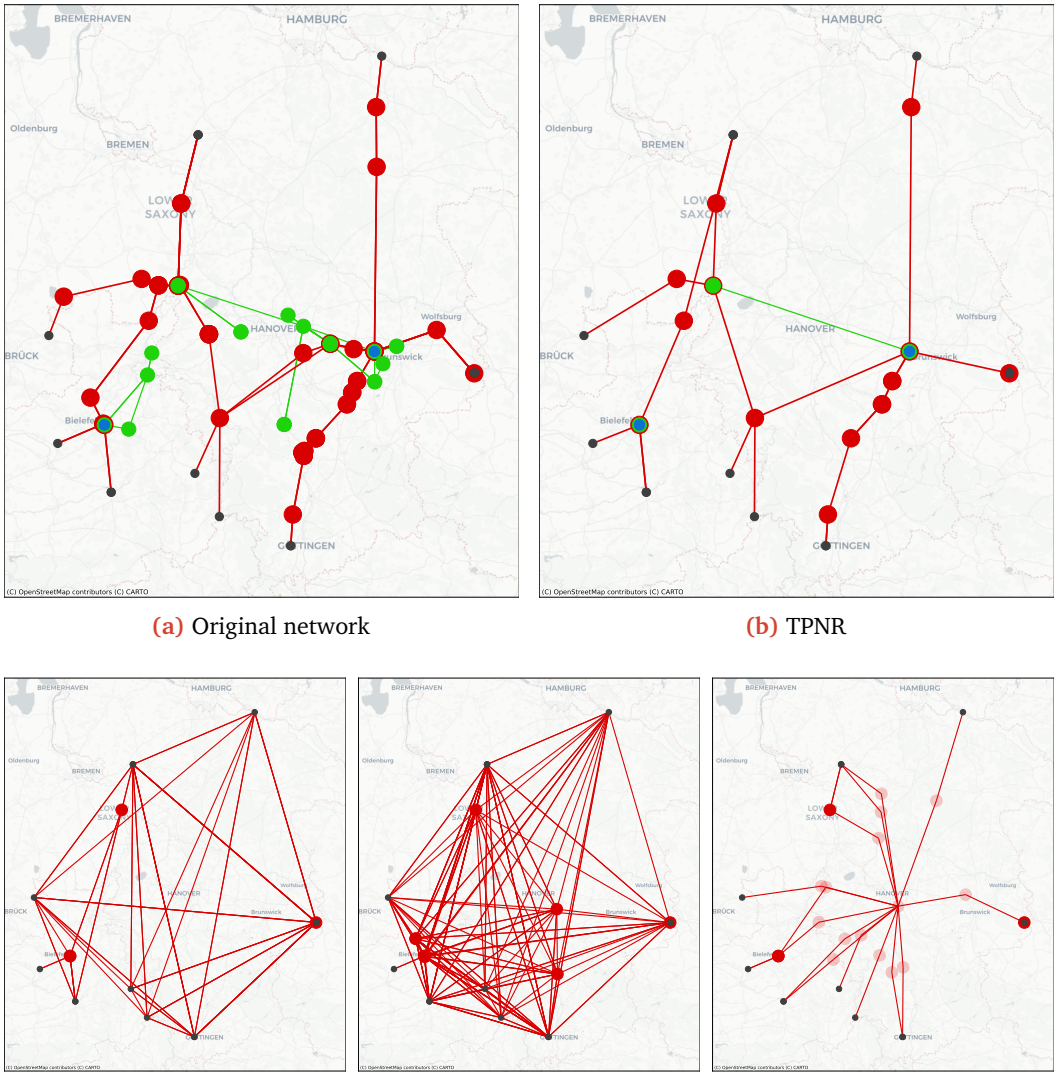
(b) TPNR-reduced network

(c) Ward

(d) REI

(e) Regional

Fig. 6.4.: The top row shows the original and TPNR-reduced graph representation of the previously introduced control zone 23. Although highly reduced in the number of nodes and edges, the equivalent on the right still clearly resembles the topology of the original model. The resulting topologies of conventional reduction methods as provided by DIgSILENT PowerFactory are shown in the bottom row. In contrast to TPNR, the Ward and REI equivalents show an extremely high interconnection between the boundary nodes, and all three methods change the topology of the network graph significantly.



(a) Original network **(b)** TPNR

(c) Ward **(d)** REI **(e)** Regional

Fig. 6.5.: The figures show the geographic representations of control zone 23 for the original network and the four reduced networks. In contrast to Figure 6.4, these figures do not show every node individually. Instead, they show the geographic locations, where one location might contain several nodes, which is evident for locations with multiple voltage levels. Transparent nodes don't have real coordinates and are placed using NetworkX's `spring_layout` method. The visual similarity between the original (a) and TPNR-reduced (b) model is obvious and allows an intuitive geographic interpretation of the reduced model. With the traditional reduction methods, (c), (d), and (e), this visual interpretation is nearly impossible.

counterparts in the detailed model. This geographic similarity is almost lost entirely with the traditional reduction methods shown in the bottom row of Figure 6.5.

To perform a quantified topological comparison of the original and reduced graph, we examine the distance between each pair of boundary buses via the connecting line lengths. Since the traditional methods utilize coupling impedances without a specified length, this can only be performed for the TPNR equivalent. We calculate the relative difference Δ_r of the distance between a pair of buses,

$$\Delta_r(d_i, \hat{d}_i) = \frac{\hat{d}_i - d_i}{d_i}, \tag{6.7}$$

where d_i is the original distance and $\hat{d}_i$ is the distance in the reduced graph. Since there are often multiple different paths between two boundary buses, we evaluate the minimum, maximum, and mean distance between buses. The results of this evaluation are visualized in Figure 6.6. It shows that the minimum distance is generally a little too high for the reduced graph, and the maximum and mean distance a little too low. Additionally, we calculate the mean relative error (MRE) for the different distances, marked by crosses in Figure 6.6. The many pairs of buses without a change in distance and the generally low MRE with absolute values between $1.2\,\%$ and $2.7\,\%$ indicate a good preservation of the spacial properties of the TPNR-reduced grid model.

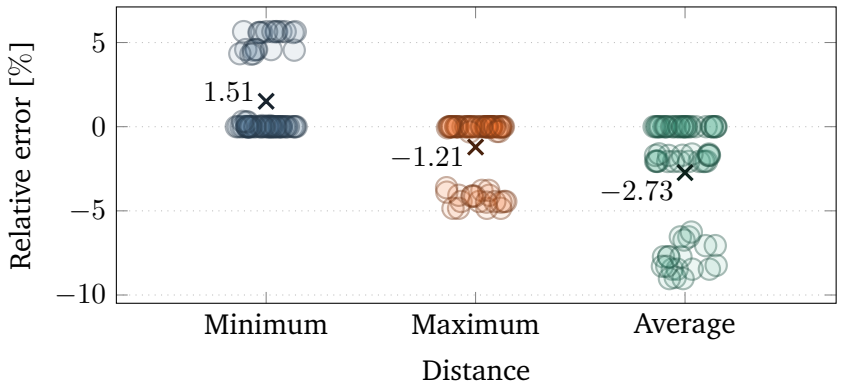


Fig. 6.6.: The figure shows the error values for distances between pairs of boundary nodes in the TPNR-reduced network compared to the original model. The crosses mark the mean relative errors for each distance. The high number of errors around zero indicate that for many pairs of nodes, the distances are preserved. However, the overall distribution of error values indicates that some minimum distances are too long, whereas the maximum and mean distances tend to be too short in the reduced model.

6.3 Discussion

In this section, we discuss the benefits and limitations of the novel TPNR method. TPNR provides a flexible usage that lets the user choose between highly reduced models (TPNR+) and more detailed versions. The evaluation shows that while TPNR and TPNR+ perform similarly well as the established Ward and REI reduction methods for load flow calculations, the standard TPNR method without the drastic component aggregation should be preferred for short circuit studies. By the preservation of the network structure and optionally of the components, TPNR yields geographically interpretable network equivalents in contrast to established methods. The individual grid manipulation operations allow tracing the components of the original model through the different aggregation steps to the reduced model. This allows, for example, calculating an average location for aggregated components or propagating property changes of detailed components to the aggregated ones. Additionally, the developed method approximately preserves the distances between boundary buses, which can be important for studies where line propagation times become relevant.

On the other hand, regarding limitations, most grid manipulation operations of the TPNR method are of a heuristic nature. They generally yield satisfying results, but are not grounded on a rigorous mathematical basis. This heuristic approach can lead to larger errors in the reactive power flow values, as observed in the evaluation, and requires further investigations. Furthermore, the aggregation and moving of components can lead to a high implementation effort, since these methods need to be implemented for all supported component types. Lastly, since the algorithm only reduces nodes with less than three neighbors, it has a limited usability for tightly meshed grids. However, the reduction of nodes by $71\,\%$ in the presented example shows that TPNR can effectively reduce moderately meshed networks.

The proposed TPNR method is a viable alternative to established reduction methods, such as Ward and REI. While the established methods yield higher reduction rates and, thus, might be preferable, in cases of very strict computational limitations, the TPNR method yields models advantageous properties: its reduced models consist of physical models, allow a mapping between components in the original and the reduced model, and preserve relevant geographic information.

6.4 Contribution and Future Work

In this chapter we answer research question **RQ4** that addresses the preservation of geographical information in reduced transmission grid models. To answer this question, we propose a new topology-preserving network reduction (TPNR) method and compare it to reduction methods found in literature and industry-leading software. This comparison includes the widely used Ward and REI methods, as well as PowerFactory's regional equivalencing method. The evaluation is performed on a real-life transmission grid model of a part of Germany and encompasses size, electrical, and topological comparisons.

In summary, this chapter provides the following contributions:

- We introduce TPNR, a reduction method for static network equivalents that preserves the overall structure of the original grid, compared to existing methods. This is accomplished by three basic graph manipulation operations, that are transferred to the underlying power grid model.
- We confirm that our proposed method has a similar, and in some cases better, accuracy as commonly used reduction methods considering steady-state electrical properties, such as load flow and short-circuit calculation.
- We demonstrate that, in contrast to existing methods, the TPNR method preserves the overall structure and important geographic information of the reduced system. This allows an easier visual interpretation of the models, helping users to understand simulation results and even adapt the reduced model to new use cases.

Considering the results of evaluating the TPNR method, further development could improve the approach in several areas. Further investigation and improvement of the heuristic grid manipulation operations, such as the length calculation for the parallel line aggregation, could lead to even better approximation results. Adding reduction operations for higher degree nodes could improve the reduction factor for more densely connected networks. Deeper investigations of the effects on renewables in the equivalent and the development methods to work with reduced models, e.g., using a mapping from original to reduced components to alter the renewable generation in the reduced model according to weather conditions, could improve the overall usefulness of the equivalents.

7 Dynamic Network Equivalencing

Solving the identified challenges the transmission grid faces—due to changes in the generation and consumption of electric energy—requires the static and dynamic simulation of these vast systems. The sheer size and complexity of transmission grids, however, is often a limiting factor especially for dynamic simulations. The topology-preserving network reduction (TPNR) method introduced in Chapter 6 fulfills all requirements identified in Section 5.4, except for the support of dynamic simulations. Being a static reduction method, TPNR does not consider the dynamic behavior of the reduced models it creates. This leaves research question **RQ5** open, that addresses supporting dynamic simulations for reduced transmission grid models, which is crucial, e.g., for HiL testing. Hence, the goal for this chapter is adapting statically equivalent reduced models to approximate the dynamic behavior of their detailed counterparts. The dynamic behavior of a transmission grid model is highly influenced by the generator controllers, e.g., exciters, governors, and power system stabilizers (PSSs). Therefore, the fundamental idea in this chapter is finding a suitable controller configuration for the generators in the reduced model to approximate the dynamic behavior of the detailed model. We describe the dynamic behavior of the model by dynamically simulating one or more events, e.g., short-circuits and load changes, and recording a selection of variables, such as bus voltages and frequencies. Thus, the goal for the reduced model is to match the curves produced by these recorded variables when simulating the same events. In order to find a controller configuration that yields similar curves, we propose a new method based on a genetic algorithm to select suitable controller models and their parameters. In addition to the *physical model* criterion identified in the research gap (Section 5.4), this method aims to approximate the dynamic behavior by only using standard controller models for maximum compatibility with common modeling software.

The remainder of this chapter is structured as follows: In Section 7.2, we introduce our new methodology in detail. Section 7.3 follows with an evaluation of the method, using a benchmark system and a real-world example. In Section 7.4, we discuss the results of the evaluation before concluding this chapter in Section 7.5.

7.1 Problem Definition and Existing Approaches

We define the original, detailed power system model as M and the reduced model as $\hat{M}$. To describe the dynamic behavior of the two systems, we define a set of events $\mathcal{E}$ and variables $\mathcal{V}$ in both models. Typical events might include short-circuits, load changes, and switch openings or closings, to represent a variety of planned and unplanned changes in the grid. Variables of interest could include voltages at the boundary buses or significant buses throughout the grid, power generation of important generators, frequencies in different areas of the grid, and power flows over crucial transmission lines. The defined events $\mathcal{E}$ are simulated and the resulting values of the variables $\mathcal{V}$ are recorded as time series. The results for all events and all variables are stored in matrices X and $\hat{X}$ for the two models, respectively. For the sake of clarity, we define the combination of an event $e \in \mathcal{E}$ and a variable $v \in \mathcal{V}$ as an objective $o = (e, v) \in \mathcal{O}$. With T time steps in each dynamic simulation, this leads to a size of $|\mathcal{O}| \times T$ for X, where $X[o, t]$ contains the t-th value of objective o. Therefore, according to [124], the problem at hand is a modeling problem: Finding a suitable model to produce the known outputs, X, for the known inputs, i.e., the defined events $\mathcal{E}$. However, this modeling problem can be transformed into an optimization problem by trying to find a suitable controller configuration in the reduced model $\hat{M}$ to minimize the difference between X and $\hat{X}$.

Eiben and Smith [124] make a clear "distinction between (search) problems—which define search spaces—and problem solvers—which are methods that tell us how to move through search spaces." For the problem at hand, the search space is the set of all possible generator control configurations, which includes the selection of controller models as well as their parameterization. Due to the size and complexity of transmission grid models, the relation between the adjustable model parameters, i.e., the controller configuration, and the model output $\hat{X}$ is prohibitively complex. It is therefore natural to view the problem of optimizing the controller configuration of the reduced model $\hat{M}$ to approximate the output X of the detailed model M as a search problem. A common method to solve this kind of problem are genetic algorithms [125].

The fundamental ideas behind genetic algorithms (GAs) are that "good solutions tend to be made up of good 'building blocks'" and that GAs "work by discovering, emphasizing, and recombining good 'building blocks' of solutions" [125]. GAs are inspired by natural evolution processes and lend many terms for their key concepts from them. According to Mitchell [125], there is no strict definition of genetic algorithms, but four key elements that most genetic algorithms have in common.

The first element is the so-called *chromosome* which describes a single solution of the genetic algorithm. Traditionally, these chromosomes are bit strings, but other representations using integer or real values are also common. A set of these solutions is called *population*. The genetic algorithm processes one population after another in so-called *generations*. The other three elements are the genetic operators that determine the processing of the populations. The first genetic operator is a *fitness-based selection*, which is used to choose chromosomes that yield good solutions to the problem as the basis for the next population. These selected chromosomes are often called *parents*. The second genetic operator, *crossover*, combines these parents to create the *offspring* which forms the next population. The third genetic operator—and last crucial element of GAs—is the *mutation*, which is used to randomly alter the offspring created by the crossover operator.

The potential to solve the problem of approximating the dynamic behavior of a reduced power grid model using genetic algorithms has been recognized in the literature. Kuri et al. [93, 94] combine the common REI reduction with a coherency-based generator aggregation and subsequent parameter tuning with a GA. They view the problem at hand as a multi-objective optimization problem, thus trying to minimize the difference between $X[o]$ and $\hat{X}[o]$ for every objective $o \in \mathcal{O}$. As such, they choose to optimize their model using the improved non-dominated sorting genetic algorithm (NSGA-II) that is specialized on multi-objective optimization [126]. They apply the NSGA-II optimization at two stages of the model equivalencing process: first to optimize the parameters of aggregated generators and then to optimize the parameters of the corresponding controllers. While the approach is described by using the NSGA-II as a framework, and the parameters of generators and controllers as a chromosome, the genetic operators used in those works are unclear.

With the formal description of the problem at hand and the recognition of GAs as an approach to solve it, as it has been applied to this problem in the literature before, we describe our solution in detail the next section.

7.2 Method: Genetic Algorithm-based Controller Optimization

This section describes our new methodology of using a genetic algorithm to optimize the generator controls of a statically reduced power system model to approximate the dynamic behavior of its detailed counterpart. As demonstrated by Kuri et al. [93, 94],

the NSGA-II is a viable option for the optimization of generator control parameters to approximate the dynamic behavior of a detailed model with a reduced-complexity equivalent. However, we identify several weaknesses in their approach that we address in our proposed method:

- Support for PSSs: While the referenced works are limited to the optimization of governors and exciters, our proposed method includes PSSs in the optimization.

- Usage of standard models only: The referenced works use a custom control structure in the form of weighted sums of transfer functions to represent the controllers. Our method uses standard controller models only for maximum compatibility with simulation software.

- Optimized selection of controller models: While the referenced works only optimize the parameters of the previously mentioned weighted transfer functions, our proposed method also includes the selection of controller models in the optimization process.

- Specialized crossover and mutation operators: The referenced works do not specify the utilized genetic operators. Our proposed method introduces crossover and mutation operators specifically designed for the optimization of generator controllers, including model selection and parameter optimization.

In the following, we introduce in detail, how our proposed method addresses these points. A brief overview of the building blocks of the proposed method and the cycle of the genetic algorithm is shown in Figure 7.1. The optimization process starts with setting up the detailed model M and the reduced model $\hat{M}$ for simulations to describe the dynamic behavior of the two models in Section 7.2.1. Section 7.2.2 follows with the genetic representation of the solution space and the creation of an initial population using this representation. In Section 7.2.3, we introduce the fitness function that is used to determine the performance of the individual solutions. Based on these fitness values, Section 7.2.4 describes the selection process for solutions to pass on their genes, i.e., parameters, and the procedure to compile the next generation of solutions. Section 7.2.5 and Section 7.2.6 describe the crossover and mutation operators, respectively, that are used to create offspring from the previously selected parent solutions. Finally, Section 7.2.7 describes the method to select a single best solution to parameterize the synchronous machine controllers of the reduced power system model.

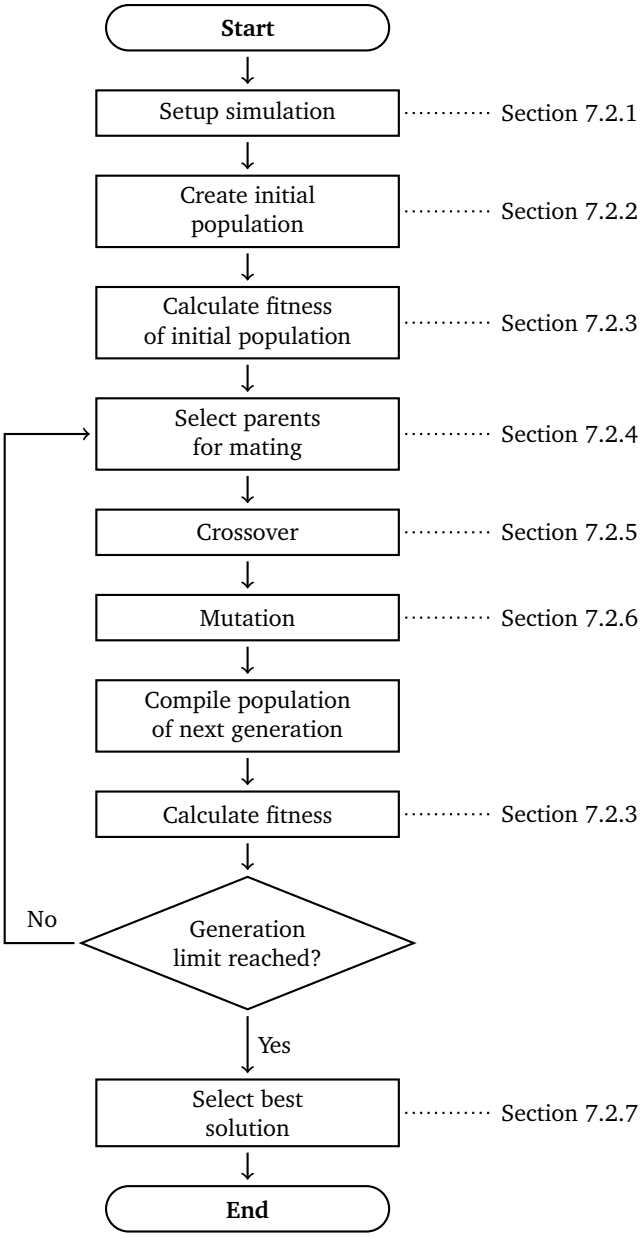


Fig. 7.1.: The procedure of optimizing the synchronous machine controllers of a reduced power system model: Starting with the setup of dynamic simulations to determine the objectives of the optimization, followed by the creation of an initial population, and the fitness calculation for this initial population. Then a loop is started with selecting individuals for mating and creating offspring through crossover and mutation. This offspring is then used to form the next generation for which the fitness is then calculated. If the completion criteria are not met, this loop is repeated, starting with the selection of new parent individuals. Otherwise, the single best solution is determined and the process terminates.

Tab. 7.1.: Parameters for tuning the optimization process with the proposed genetic algorithm.

Symbol	Description
N_P	Population size
N_G	Number of generations
p_c	Crossover probability $\in [0.0, 1.0]$
p_m	Mutation probability $\in [0.0, 1.0]$
p_{tm}	Type mutation probability $\in [0.0, 1.0]$
s_m	Mutation strength $\in [0.0, 1.0]$

The parameters in Table 7.1 are used throughout the optimization process and can be tuned to fit the problem at hand. An exemplary selection of these parameters is given in Section 7.3.2.

7.2.1 Simulation Setup: Simulated Events and Variables

As introduced in the problem definition (see Section 7.1), we describe the dynamic behavior of the detailed model M and $\hat{M}$ by simulating a number of events $\mathcal{E}$ and recording a selection of system variables $\mathcal{V}$. The optimization process starts with setting up these events and variables. The first step in the process to optimize the controllers in $\hat{M}$ to approximate the behavior of M is setting up the simulations by selecting the duration and temporal resolution. Then, a set of events $\mathcal{E}$ is defined in the retained parts of the model, i.e., in the area of interest (AoI), to simulate in both models in order to evaluate the system behavior under different situations. These simulation events usually include short-circuits, switching events dis-/connecting grid components, and changes in load or generation. The next step is the selection of system variables $\mathcal{V}$ to record. These variables can include the voltages and frequencies at the boundary buses, as in [93], generator variables, such as frequency, active and reactive power output, or any other variables of system components that are of particular interest for the specific use case. Simulating the defined events $\mathcal{E}$ with the detailed model M yields the matrix X which contains the recorded values for all variables $\mathcal{V}$ and describes the dynamic behavior of M. Approximating this matrix X with the reduced model $\hat{M}$ is the goal for the subsequent optimization procedure.

Tab. 7.2.: The optimization method utilizes controller models commonly found in literature and modeling software. The table lists all implemented controller models for the three controller types with their respective name and number of parameters in parentheses. The variance in the number of parameters per model results in the requirement for padding in the chromosome. An additional selector parameter determines the model used for the represented controller.

Selector	Exciter	Governor	PSS
0	SEXS (6)	IEEE G1 (16)	PSS1A (12)
1	IEEE AC1A (19)	TGOV1 (9)	PSS2A (23)
2	IEEE ST1A (20)	GAST (10)	No PSS (0)
3		HYGOV (12)	

7.2.2 Genetic Representation of Individual Solutions

A central part of every genetic algorithm is the representation of the solution space as a chromosome. This section describes the general structure of this representation and the methods to create valid chromosomes of individual solutions and an initial population at the start of the optimization procedure.

In order to optimize the controllers, i.e., exciter, governor, and PSS, of the aggregated generators, the chromosome needs to include two kinds of information: the specific controller model used to control the corresponding generator, and the parameters of this controller model. The genetic representation proposed in this method comprises an ordered list of all aggregated generators and a chromosome as illustrated in Figure 7.2 (simplified). This chromosome describes the controller model using a selector parameter and all relevant model parameters for the specific exciter, governor, and PSS model for every aggregated generator in the reduced power system model $\hat{M}$. Table 7.2 shows the different controller models implemented for the three controller types. These models are a selection of controller models commonly found in literature and available in common modeling software. The table also shows that not all controller models require the same number of parameters. Thus, padding with zero values is used to ensure a consistent chromosome length for all variations. This padding also ensures that each individual controller is always described by the same entries in the chromosome. The IEEE ST1A, IEEE G1, and PSS2A have the most model parameters for their controller type, respectively, and thus determine the chromosome length. Their 20, 16, and 23 parameters in addition to three type selectors result in a chromosome length of 62 genes per generator. In combination with the ordered list of generators, this enables the mapping of the parameter values in the chromosome to the corresponding controllers in the reduced system model $\hat{M}$.

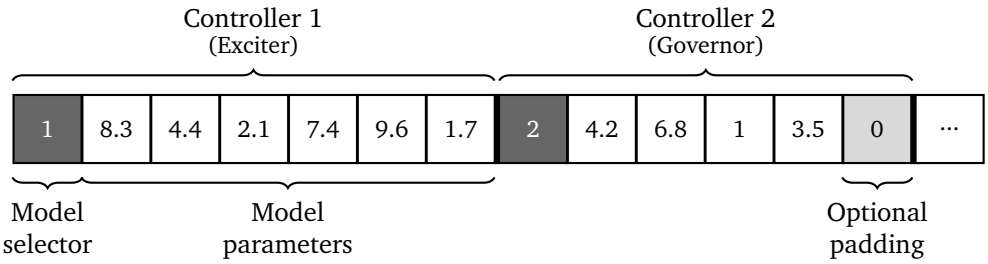


Fig. 7.2.: A simplified illustration of the genetic representation, or chromosome, used to define a solution. For every aggregated generator, it describes the model and parameter values for its exciter, governor, and PSS. As not all models for one controller have the same number of parameters, some controller descriptions include a padding with zero values to ensure a consistent chromosome length.

Generating Valid Controller Configurations

In order to gain valid configurations of controllers, the values that can be applied to the model parameters need to meet certain requirements. The controllers utilized for this optimization exhibit three types of parameter restrictions: upper and lower bounds, constraints depending on other parameters, and integer constraints.[1]

Upper and lower bounds apply to all model parameters. When generating a new individual, parameters with only this constraint are simply set to a random value selected uniformly from the valid range. For example, Table A.7 shows the valid ranges for the simplified excitation system (SEXS) whose parameters do not have any other constraint types.

Constraints depending on other parameters often describe the requirement that one parameter value is greater than another. This can be implemented by selecting the first parameter and using the resulting value as a limit for the second parameter. However, other dependencies between parameters are possible. For example, for the IEEE G1 governor the parameters K_1, K_3, K_5, and K_7 need to add up to 1. This example is implemented by randomly selecting all four parameters and dividing them by their sum afterward.

The last type of constraint are integer constraints. These constraints typically describe structural options of the controllers, such as the selection of inputs, as is the case for the I_{PB} parameter of the two PSS models. There are, however, more complex cases for integer constraints. For example, the M and N parameters of

[1] A full description of the model parameters including their constraints is included in Appendix A.2.

the PSS2A model describe exponent values in the models and exhibit all types of constraints with the following requirements:

$$0 \leq M \leq 8 \in \mathbb{Z} \tag{7.1}$$

$$N = 0, \text{ if } M = 0 \tag{7.2}$$

$$N \in \mathbb{Z} \wedge N \cdot M \leq 8, \text{ else.} \tag{7.3}$$

These constraints can be implemented by first selecting M from the valid range and then setting N accordingly depending on the relevant case.

Generating the Initial Population

For the generation of an initial population P, the proposed optimization method has two options depending on the available system information. By default, if no information about the generator controls of the detailed model is available, randomly generated control configurations are assigned to the generators in the reduced system. For each of these generators, one valid controller configuration per type is generated at random and combined with the model selectors and padding to fit the required chromosome structure.

As a second option, however, it might be beneficial to consider the controllers used in the detailed model for the initialization of the population P, if this information is available. While using the same controller models and parameters as the detailed model does not necessarily lead to a similar behavior of the reduced model, it can be a good starting point for the optimization process. In the following, we describe, how such information can be utilized when combining this genetic optimization approach for dynamic equivalents with the static TPNR method, as introduced in Chapter 6. One advantage of the TPNR method is the rule-based and discrete reduction process that describes all modifications that are applied to network components. This approach enables keeping track of how the components in the reduced model are created by moving and combining the components of the original model. Thus, for every generator in the reduced system that is created by combining generators from the detailed system, TPNR records the controls of the original generators involved in its creation. For each of the three types of controllers a list of tuples is maintained consisting of the rating of the original generator and the corresponding controller model and parameters. As an example for an aggregated generator that is created by combining three generators, the record of the used PSS controllers could look like Listing 7.1. In this case, the three original generators all have different control configurations: no PSS at all, PSS2A, and PSS1A. To generate the initial population

```
[
    (50.0, None),
    (100.0, {
        "type": "PSS2A",
        "Tw1": 2.0,
        "Tw2": 2.0,
        ...
    }),
    (50.0, {
        "type": "PSS1A",
        "Ks": 4.0,
        "Tr": 0.5,
        ...
    }),
]
```

Listing 7.1: An example record of PSS instances of the original generators. For each PSS, the list contains the rating of the corresponding generator and the type and parameters describing the controller.

P, the ratings in these records are used to perform a weighted random selection of the original controller configurations. For this random selection, the cumulative sum of ratings is calculated (50, 150, and 200 for the example in Listing 7.1). For each controller, a random value is selected in the range from zero to the total rating sum. The entry with the lowest cumulative rating greater than the random value is selected to be used to parameterize the controller for the current individual. This approach can be used to completely or partly generate the initial population P in combination with the purely random approach described above.

7.2.3 Fitness Function

A crucial part of the optimization is the fitness function F which describes the performance of an individual solution $p \in P$. The proposed optimization method performs a multi-objective optimization, meaning that it considers multiple objectives $o \in \mathcal{O}$ to determine the overall fitness of a solution. As defined in Section 7.1, these objectives are a combination of the simulated event e and the recorded system variable v, thus $\mathcal{O} \subset \mathcal{E} \times \mathcal{V}$. While X contains the values resulting from the simulations of all events $\mathcal{E}$ with the detailed model M, $\hat{X}_p$ describes these values from simulating the reduced model $\hat{M}$ using the parameters of individual p. To describe the difference between the target values and the achieved values, we use the MAE as defined in Equation (2.1). The error for objective o of solution p is thus defined as

$$\mathrm{MAE}_o(\hat{X}_p) = \frac{1}{T}\sum_{t=1}^{T}\left|X\left[o,t\right] - \hat{X}_p\left[o,t\right]\right|. \tag{7.4}$$

The goal of the genetic algorithm is to find solutions with a high fitness. Since the defined error value increases with poorer results of an individual, it cannot be used directly as a fitness function. Instead, the fitness function for objective o of individual p is defined as

$$F_o(p) = -1 \cdot \mathrm{MAE}_o(\hat{X}_p) \tag{7.5}$$

in order to produce higher fitness values with decreasing errors. The overall fitness of p, $F(p)$, is a vector consisting of the fitness values of all objective variables, $F_o(p)$, and is defined as

$$F(p) = \begin{cases} \begin{bmatrix} F_1(p) \\ F_2(p) \\ \dots \end{bmatrix} & \text{if simulation complete} \\ \begin{bmatrix} -10^{15} \\ -10^{15} \\ \dots \end{bmatrix} & \text{else,} \end{cases} \tag{7.6}$$

where simulation completeness can be determined with output flags or the number of missing values in $\hat{X}_p$. In order to qualify for overall completeness, all simulations need to be completed, which is not necessarily the case if multiple events are simulated. The value of -10^{15} is chosen as an arbitrarily small number that should be several orders of magnitude smaller than even the worst completing simulations. Mapping all failing individuals to the same vector with such a low fitness aims to ensure a fast exclusion of these individuals from further generations.

7.2.4 Selection for Mating and Compilation of the Next Population

The first genetic operator is the selection of individuals for the mating process and creation of the next generation through crossover (see Section 7.2.5) and mutation (see Section 7.2.6). The proposed method adopts parts of the NSGA-II algorithm, as this algorithm is designed for fast multi-objective optimization [126], which is well-suited for the potentially large number of objectives when optimizing, e.g., the voltages and frequencies of the boundary buses in a reduced model. A key mechanism in NSGA-II is elitism, meaning that the individuals with the highest fitness are guaranteed to move forward to the next generation. The proposed method applies elitism with $50\,\%$ of the population, i.e., with $\mathsf{N_P}$ as the population size, $\mathsf{N_P}/2$ individuals are directly used to form the next generation, and the other half is formed by selection, crossover, and mutation.

In order to determine the individuals with the highest fitness, NSGA-II introduced the crowded-comparison operator ($\prec_n$) that is based on the non-dominated rank p_{rank} (see Algorithm 3) and crowding distance p_{dist} (see Algorithm 4). The operator is defined as follows:

$$p \prec_n q :\Leftrightarrow (p_{\text{rank}} < q_{\text{rank}}) \vee ((p_{\text{rank}} = q_{\text{rank}}) \wedge (p_{\text{dist}} > q_{\text{dist}})). \tag{7.7}$$

Selecting individuals based on this crowded-comparison operator thus prefers individuals with a lower rank and individuals with a larger crowding distance, i.e., more distance to comparable solutions regarding their fitness. This aims at selecting a well-performing set of solutions with a high diversity.

One half of the next generation is created by sorting all solutions using this crowded-comparison operator and selecting the first $\mathsf{N_P}/2$ individuals. The other half is created through crossover and mutation. For this purpose, $\mathsf{N_P}$ individuals are selected from the current population P for mating, as every crossover operation combines two parent individuals to form one offspring individual. The mating individuals are selected with a tournament selection method. Two individuals are selected randomly and compared with the crowded-comparison operator and the individual with the lower value according to Equation (7.7) is selected for the mating pool. This process is repeated until $\mathsf{N_P}$ individuals are selected. Using the tournament-based selection method allows individuals from all ranks to get selected for mating, resulting in a larger diversity without risking the loss of well-performing individual because of the elitism.

7.2.5 Crossover

The crossover is the second genetic operator and a central part of every genetic algorithm, expanding the search space and creating the next generation from individuals of the current one, usually by combining two parent individuals to one offspring individual. As such, there exists a wide variety of crossover operators for general optimization problems [127]. However, it is often beneficial to adapt the crossover operator to the problem at hand. As the controller parameters and, thus, the chromosome describing each individual have very specific requirements, we propose a crossover operator specifically for this problem, as described in Algorithm 5. This crossover operator is performed piecewise for every controller encoded in the chromosome with a certain crossover probability $\mathsf{p_c}$. If a random variable is not

Algorithm 3 Fast non-dominated sorting. Recreated from [126].

procedure NON_DOMINATED_SORTING(P ▷ *population*)
    $\mathcal{F}_1 \leftarrow \emptyset$ ▷ *the first non-dominated front*
    for all $p \in P$ **do**
        $S_p \leftarrow \emptyset$ ▷ *set of solutions dominated by* p
        $n_p \leftarrow 0$ ▷ *counter of solutions dominating* p
        for all $q \in P$ **do**
            if $p \prec q$ **then** ▷ p *dominates* q
                $S_p \leftarrow S_p \cup \{q\}$
            else if $q \prec p$ **then**
                $n_p \leftarrow n_p + 1$
        if $n_p = 0$ **then** ▷ p *is not dominated by any other solution*
            $p_{\text{rank}} \leftarrow 1$ ▷ p *gets rank 1*
            $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{p\}$ ▷ p *is added to the first front*

    ▷ *determine the remaining fronts* $\mathcal{F}_i$
    $i \leftarrow 1$
    while $\mathcal{F}_i \neq \emptyset$ **do**
        $Q \leftarrow \emptyset$ ▷ *temporary set to become the next front*
        for all $p \in \mathcal{F}_i$ **do**
            ▷ *iterate over all solutions dominated by those in the current front*
            for all $q \in S_p$ **do**
                $n_q \leftarrow n_q - 1$ ▷ *eliminate* p *from domination counter of* q
                if $n_q = 0$ **then** ▷ *there are no further solutions dominating* q
                    $q_{\text{rank}} \leftarrow i + 1$ ▷ q *gets the next higher rank*
                    $Q \leftarrow Q \cup \{q\}$ ▷ q *is added to the next front*
        ▷ *assign the next front*
        $i \leftarrow i + 1$
        $\mathcal{F}_i \leftarrow Q$

Algorithm 4 Crowding distance calculation. Recreated from [126].

procedure CROWDING_DISTANCE(P ▷ *population*, $\mathcal{O}$ ▷ *objectives*)
    for all $p \in P$ **do**
        $p_{\text{dist}} \leftarrow 0$ ▷ *initialize distance*

    $l \leftarrow |P|$
    for all $o \in \mathcal{O}$ **do**
        $P \leftarrow \text{SORT}(P, o)$ ▷ *sort ascending regarding each objective*
        $P[1]_{\text{dist}} \leftarrow P[l]_{\text{dist}} \leftarrow \infty$ ▷ *to ensure that boundary points are selected*
        for $i \in \{2..l-1\}$ **do**
            $P[i]_{\text{dist}} \leftarrow P[i]_{\text{dist}} + (F_o(i+1) - F_o(i-1))/(F_o^{\max} - F_o^{\min})$

Algorithm 5 Controller-aware arithmetic crossover.

```
1: procedure CROSSOVER(parentA, parentB)
2:     offspring ← randomly selected parent
3:     for i < NUM(controllers in chromosome) do
4:         if RANDOM(0, 1) ≤ p_c then
5:             if parentA.ctrl[i].model = parentB.ctrl[i].model then
6:                 offspring.ctrl[i] ← ARITHMETIC_CROSSOVER(
                       parentA.ctrl[i], parentB.ctrl[i]
                   ) ▷ see Equation (7.8)
7:             else
8:                 offspring.ctrl[i] ← randomly select i-th controller from A or B
9:     return offspring
```

greater than p_c, the controller models of both parents are compared and, if they have the same model type, are combined via arithmetic crossover, defined as

$$C_i = A_i + r_i(B_i - A_i), \quad 0 \leq r_i \leq 1, \tag{7.8}$$

where A_i and B_i are the i-th controller of the parents and C_i is the resulting offspring controller. The variable r_i is selected randomly for every application of the operator and determines where the offspring lies between both parents. If the controller types do not match, one parent controller is selected randomly to form the offspring controller. Figure 7.3 shows a simplified application of this procedure for both cases.

One concern of the crossover as well as the mutation operator is producing valid configurations for the controllers. The arithmetic crossover that is used in both operations produces valid results regarding parameter limits and linear parameter constraints, such as the requirement of one parameter value being smaller than another one. However, it does not necessarily yield valid configurations regarding integer constraints or more complex conditional constraints, such as in Equations (7.1) to (7.3). For this reason, it might be necessary for some controller models to check and potentially correct the created configurations, e.g., by rounding parameters with integer constraints.

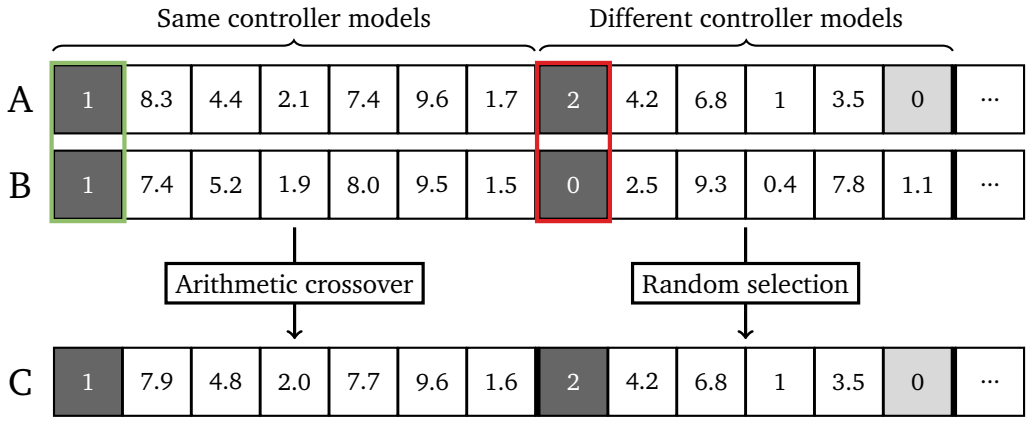


Fig. 7.3.: An exemplary application of the controller-aware crossover that performs an arithmetic crossover, as given in Equation (7.8), with $r = 0.5$ to yield the first offspring controller (with rounded values for visual clarity). As the types of the second controllers do not match, one controller is selected at random for the offspring.

7.2.6 Mutation

The mutation is the third type of genetic operator that is central to all genetic algorithms. Instead of combining two parent individuals to create offspring, this operator randomly alters only one individual to create one offspring individual [124]. For the creation of offspring via mutation, the same restrictions apply as for the crossover, mentioned in Section 7.2.5. Thus, we propose a mutation operator specifically for the problem at hand that is described in Algorithm 6. Analogous to the crossover operator, the mutation is performed by iterating over the controllers defined in the parent. With a probability of $\mathsf{p_m}$, a controller is mutated using one of two methods: With a probability of $\mathsf{p_{tm}}$, a completely new controller instance is created and used for the offspring. In the other cases, a new controller instance B_i is created, and a scaled arithmetic crossover is calculated between B_i and the single parent A_i, as follows

$$C_i = A_i + r_i \cdot s_m(B_i - A_i), \quad 0 \leq r_i \leq 1, \quad 0 \leq \mathsf{s_m} \leq 1, \tag{7.9}$$

where r_i is selected randomly for every application of the operator and $\mathsf{s_m}$ is a global parameter that defines the strength of the mutation. Lower values of $\mathsf{s_m}$ generally lead to the offspring being closer to the parent while higher values result in stronger mutations. Both mutation cases are presented in Figure 7.4 with a simplified example.

Algorithm 6 Controller-aware mutation.

```
1:  procedure MUTATION(parent)
2:      offspring ← parent
3:      for i < NUM(controllers in chromosome) do
4:          if RANDOM(0, 1) ≤ p_m then
5:              if RANDOM(0, 1) ≤ p_tm then
6:                  offspring.ctrl[i] ← CREATE_RANDOM_INSTANCE( )
7:              else
8:                  mutant_ctrl ← CREATE_RANDOM_INSTANCE(parent.ctrl[i].model)
9:                  offspring.ctrl[i] ← SCALED_ARITHMETIC_CROSSOVER(
                        parent.ctrl[i], mutant_ctrl
                    ) ▷ see Equation (7.9)
10:     return offspring
```

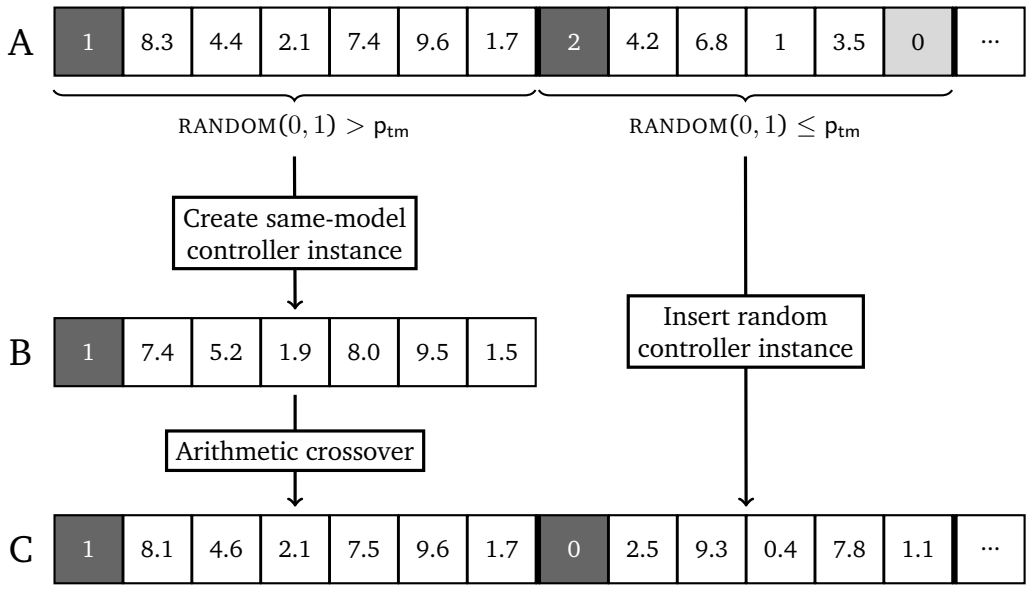


Fig. 7.4.: Both cases of the controller-aware mutation: the first controller is mutated with the scaled arithmetic crossover, as given in Equation (7.9), with $r \cdot s_m = 0.2$ (with rounded values for visual clarity). For the second controller, a type mutation is performed by selecting a completely random controller instance.

7.2.7 Selecting the Best Individual

The final step of the parameter optimization is the selection of the best individual, i.e., the set of parameters with the best results. However, the potentially high dimensionality of the fitness vectors, $F(p)$, likely results in non-dominating solutions, where no single solution has the highest fitness for all objectives. This is not an issue for the optimization with the genetic algorithm, because multiple solutions can be selected to move to the next generation (see Section 7.2.4). Still, in the end, one solution has to be selected as the parameterization of the reduced model. Thus, we define $f(p)$ to describe the fitness of an individual p with a one-dimensional value as follows:

$$F_o^{\min} = \min_{p \in P_c} F_o(p) \tag{7.10}$$

$$f_o(p) = F_o(p) / F_o^{\min} \tag{7.11}$$

$$f(p) = \sum_{o \in \mathcal{O}} f_o(p), \tag{7.12}$$

where P_c is the population of individuals that complete the simulations successfully. Dividing the fitness values of each objective by the minimum fitness achieved for this objective as in Equation (7.11), effectively normalizes the values to the interval $[0, 1]$, resulting in a better comparability of the different objectives. Failing solutions are explicitly excluded for the normalization, as these would defeat the purpose of normalization by dividing all fitness by the fixed value of -10^{15}. With this normalized fitness $f_o(p)$, originally small fitness values $F_o(p)$, i.e., large errors, result in larger normalized values. Thus, the individual with the minimal summarized fitness $f(p)$ is selected as the best solution. Depending on the use case for the reduced model, a weighted summation of the individual objectives might be beneficial. The proposed single fitness value f, however, assigns the same weight to all objectives.

7.2.8 Implementation

The genetic algorithm for optimizing the synchronous machine controls in reduced power system models described above is implemented in Python for DIgSILENT PowerFactory[2] models. For the genetic algorithm framework, the implementation uses the PyGAD package [128] that includes the basic workflow of the algorithm as well as implementations of the fast non-dominated sorting and crowding distance calculation (see Section 7.2.4).

[2] See Section 2.2 for information on common modeling and simulation tools.

7.3 Evaluation

In this section, we evaluate the proposed GA-based method to optimize controller parameters in reduced power systems $\hat{M}$ to approximate their detailed counterparts M. This evaluation is performed with the IEEE 39 Bus System as a benchmark case and a more detailed, realistic transmission grid model. These two models are described in Section 7.3.1. Since the GA requires a number of parameters (see Table 7.1) to specify its behavior, we perform a systematic evaluation of parameter sets in Section 7.3.2. Finally, the results of optimizing the reduced versions of the two models are presented in Section 7.3.3. For the IEEE benchmark case, the evaluation examines the effects of the defined objective variables and events. The resulting optimized models are compared against a controller parameterization based on expert knowledge of the detailed system model. For the more realistic use case with the large transmission grid models, we compare several approaches to perform the parameter optimization to gain insights on the practical usability of our method.

7.3.1 Validation Models

For the evaluation of the proposed optimization method, two models are utilized: The IEEE 39 Bus System is used as a benchmark model as it is widely known and used in the literature. A much larger and more detailed transmission grid model of the German state of Baden-Württemberg (BW) is used to provide insights on a more realistic use case.

IEEE 39 Bus System

The IEEE 39 Bus System, also known as the New England System, is a commonly used model of a high-voltage transmission grid and serves as a benchmark model for the method proposed in this chapter as it is also used in comparable methods [93, 94]. Specifically, this evaluation uses the model supplied by DIgSILENT PowerFactory 2022 with an adapted controller configuration as shown in Figure 7.5a. The original model contains the same controller setup for each of the ten synchronous machines in the system, consisting of an IEEE G1 governor, an IEEE ST1A exciter, and an IEEE PSS1A PSS. In order to create a more divers benchmark model, these controllers are complemented with GAST governors, IEEE AC1A exciters, and IEEE PSS2A PSSs as well as deactivated PSS. Details of the controller configuration are provided in Appendix A.3.1.

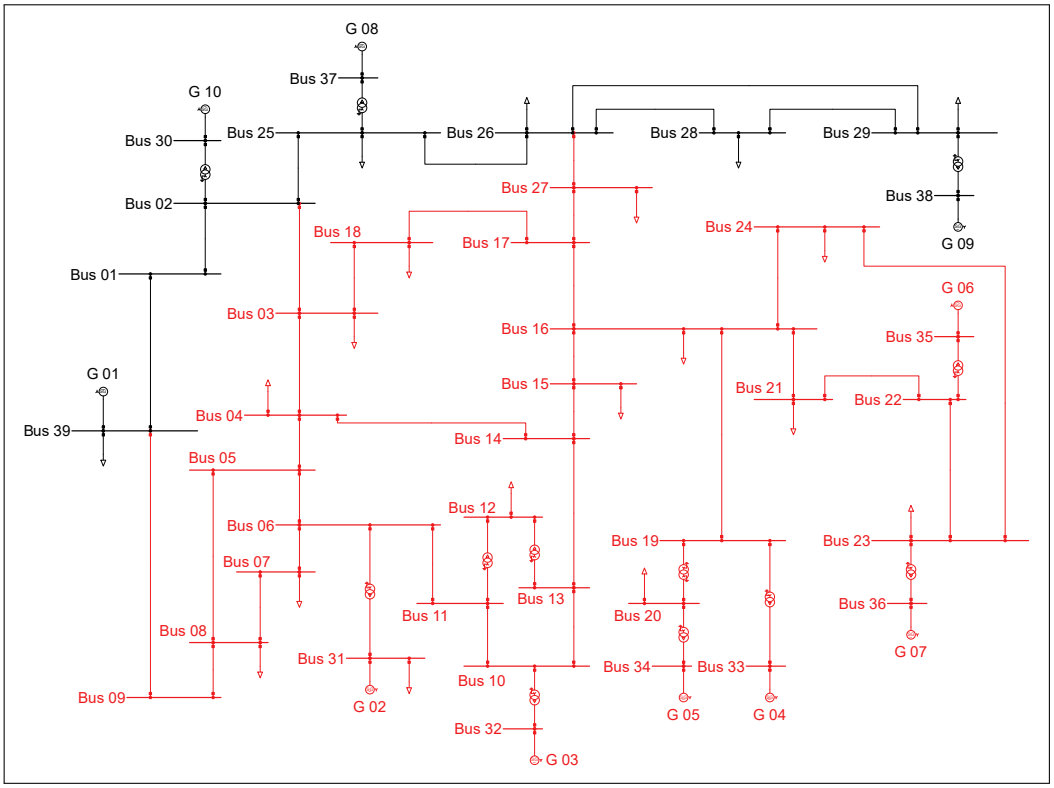
A reduced version of the IEEE 39 Bus System is obtained by reducing the red area in Figure 7.5a with the REI reduction method as provided by PowerFactory. This static reduction method is used instead of the TPNR method introduced in Chapter 6 for two reasons: First, the IEEE 39 Bus System already is a simplified model judging by its topology. Applying the TPNR method would only result in very minor reductions. Second, combining the proposed optimization method with a widely used static reduction method demonstrates its wide applicability. The resulting equivalent model is shown in Figure 7.5b. The 29 buses in the original red zone are reduced to 5 equivalent buses, and the 6 generators are reduced to 2. Apparent is the large number of coupling impedances between the newly created equivalent buses and the boundary buses, i.e., *Bus 02*, *Bus 26*, and *Bus 39*.

For evaluating the system behavior and optimizing the reduced model, we define four typical simulation events. The first event is also used in [93] and aims to provide a basic comparability between the two approaches. The other three events aim to cover typical events that could be of interest when simulating a transmission grid model. The following four events are used for the evaluation:

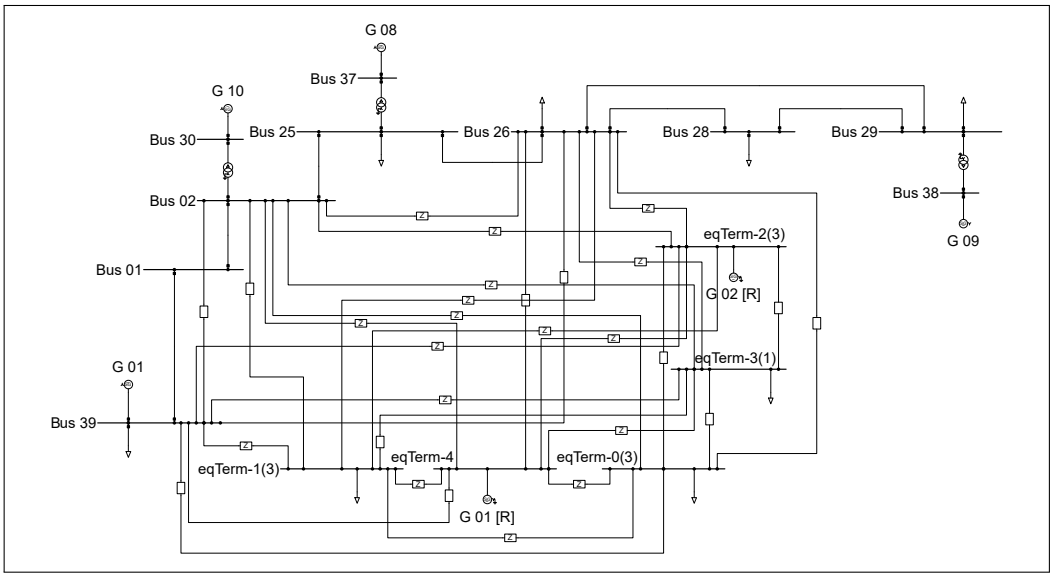
1. A three-phase short-circuit at *Bus 26* after $5\,\mathrm{s}$ with $0\,\Omega$ impedance, that is cleared after $0.1\,\mathrm{s}$.
2. Opening the switch at *Bus 01*, that connects it to *Bus 02* via a transmission line after $0.2\,\mathrm{s}$.
3. A step load increase at *Bus 25* by $50\,\%$ after $0.2\,\mathrm{s}$, increasing the load from $224\,\mathrm{MW}$ and $47.2\,\mathrm{Mvar}$ to $336\,\mathrm{MW}$ and $70.8\,\mathrm{Mvar}$.
4. Ramping up the load at *Bus 28* by $50\,\%$, starting at $1\,\mathrm{s}$ over a duration of $5\,\mathrm{s}$, increasing the load from $206\,\mathrm{MW}$ and $27.6\,\mathrm{Mvar}$ to $309\,\mathrm{MW}$ and $41.4\,\mathrm{Mvar}$.

Baden-Württemberg Transmission Grid

The second model for evaluating the proposed synchronous machine controller optimization is a transmission grid model of the German state of Baden-Württemberg (BW), comprising the $380\,\mathrm{kV}$ and $220\,\mathrm{kV}$ level. A geographic representation of the model is shown in Figure 7.6. This region consists of the two control zones D41 (blue) and D42 (red) that form the AoI and the external system for this evaluation, respectively. For this evaluation, the reduced model equivalent is obtained by applying the TPNR method to control zone D42. This static reduction reduces the number of synchronous machines in the external area from 42 to 22 without considering the dynamic behavior of the system. An overview of the main system



(a) Original network



(b) Reduced network

Fig. 7.5.: The original and reduced version of the IEEE 39 Bus System used for the evaluation depicted as a single line diagram in PowerFactory. The reduced model is obtained by applying PowerFactory's REI reduction on the red area of the original model, resulting in a reduction by 24 buses and 4 synchronous machines.

Tab. 7.3.: An overview of the numbers of system components of the BW model contained in the area of interest (AoI), the external system of the original model (red area in Figure 7.6), and the reduced equivalent of this external area. The synchronous machines relevant for this optimization are reduced by 47.6 %.

	AoI	External	Reduced	Reduction [%]
Buses	128	177	104	41.2
Synch. machines	30	42	22	47.6
Static generators	166	268	83	69.0
Loads	96	131	45	65.6
Transformers	50	90	69	23.3
Transmission lines	68	85	46	45.9

components in the AoI as well as the original and reduced external area is provided in Table 7.3. The original model uses varying governor models, depending on the type of synchronous machine, i.e., GAST for gas turbines, HYGOV for hydropower plants, and IEEE G1 for others. Otherwise, the generators all use the same SEXS exciter and IEEE PSS2A PSS. As described in Section 7.2.2, this information on the controllers used in the original model is used to partly initialize the first generation of the genetic algorithm. For the evaluation of the system behavior and the optimization of the reduced model, we define the following four simulations to cover a variety of use cases:

1. An RMS simulation without any events, i.e., a dynamic simulation of a steady system. This scenario is used because the experience with this particular model shows that a controller configuration that results in a stable simulation is not trivial to find.

2. A three-phase short-circuit from 1 s to 1.1 s with 0.01 Ω at a bus in the substation marked with the lightning symbol in Figure 7.6.

3. A load step increase at 0.2 s by 200 % at the substation marked with “L”, increasing the load from 202 MW and 37 Mvar to 606 MW and 111 Mvar.

4. A switching event of the line marked with “S”, disconnecting the line at 0.2 s and reconnecting it at 1.2 s.

7.3.2 Parameter Selection

In order to perform the synchronous machine control optimization with the genetic algorithm proposed in Section 7.2, at least the six parameters defined in Table 7.1 need to be set. For this purpose, a systematic evaluation of these parameters is

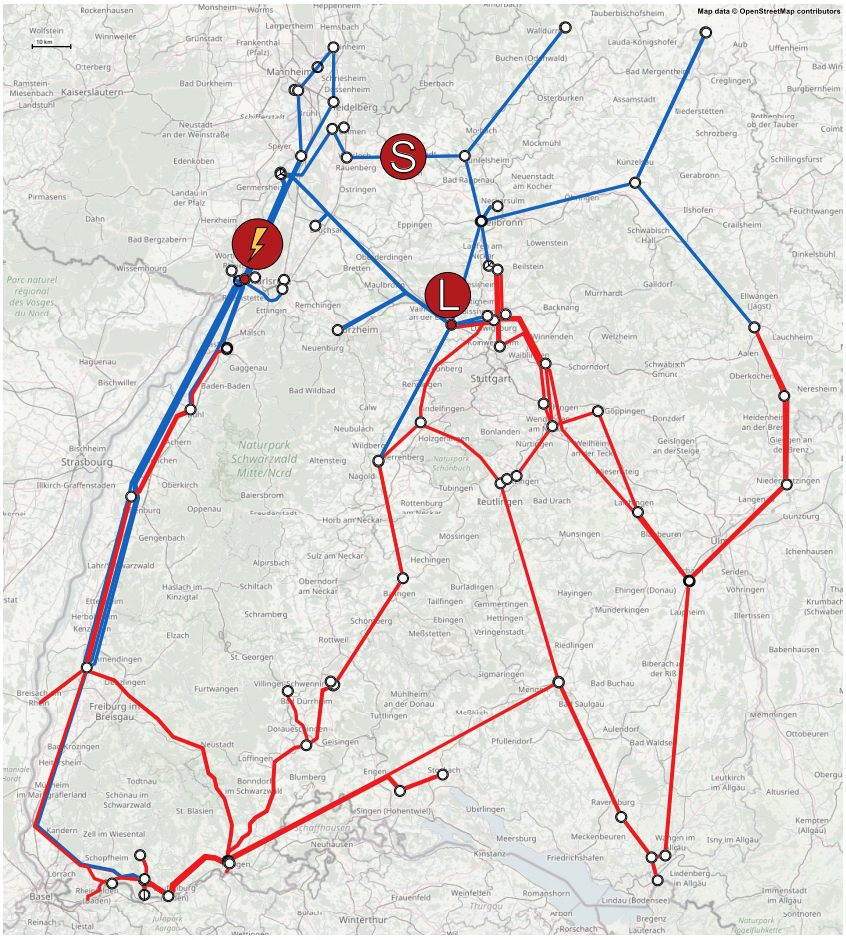


Fig. 7.6.: A geographic representation of the BW transmission grid model with the two control zones D41 (blue) and D42 (red). The reduced equivalent model is obtained by applying the TPNR method (see Chapter 6) to the red area of the system. The red symbols indicate the locations of the events used to describe the system behavior.

performed by optimizing the IEEE 39 Bus System with various parameter combinations. With an initial testing, a population size N_P of 50 and a number of generations N_G of 50 were determined to strike a good balance between runtime, diversity per generation, and improvement of generations over time. However, these parameters likely need to be adjusted for other models. Larger models with more controllers that need to be optimized likely require a larger population to achieve a suitable coverage of the solution space.

With the first two parameters set, the remaining four parameters were determined with a grid search, repeating the optimization with each combination five times. For the four parameters, the following values were evaluated:

- $p_c \in \{0.2, 0.5, 0.8\}$. To evaluate a low, moderate, and high crossover rate.

- $\mathsf{p_m} \in \{0.2, 0.5, 0.8\}$. To evaluate a low, moderate, and high mutation rate.
- $\mathsf{p_{tm}} \in \{0, 0.2\}$. As this is a drastic mutation operation, only a low probability was compared to no type mutation at all.
- $\mathsf{s_m} \in \{0.1, 0.3\}$. The mutation operator is based on a crossover with completely random controller parameters. Thus, it is assumed that this operation should lean more toward the parent controller rather than the random controller.

From the resulting 36 parameter sets, $(0.2, 0.5, 0.2, 0.1)$ was found to produce the best results on average, regarding the one-dimensional fitness f (see Equation (7.12)), with a small variance between the different optimization runs. It was also found that the 10 best combinations have type mutation probability of $\mathsf{p_{tm}} = 0.2$, while the 14 worst combinations have $\mathsf{p_{tm}} = 0$. The worst combination overall was $(0.2, 0.2, 0, 0.1)$, i.e., the combination with the smallest probability of change.

7.3.3 Results

In this section, we present the results of the dynamic equivalent optimizations with the previously described models. While the IEEE 39 Bus System is used as a benchmark to perform an in-depth quantitative evaluation of the optimization method, providing an indirect comparison to the literature, the BW transmission grid model is used for a more qualitative evaluation, representing a realistic use case for the proposed methodology.

To evaluate the optimization results quantitatively, we use the MAPE and MAE as defined in Equation (2.3) and Equation (2.1), respectively, calculating the difference of the response of the reduced model $\hat{M}$ to the detailed model M. The MAPE provides a comparable error measures for the different variables that vary significantly in their typical range of values, e.g., the active power P of large generators in the hundreds of MW and the frequency in $\mathrm{p.u.}$ close to one. However, the MAPE can be misleading when the correct values get close to zero, resulting in large relative errors even for small absolute deviations. This is often the case for the reactive power Q of the generators that might cross zero depending on the simulated event. For these cases, the MAE provides additional context to assess the errors. While the MAE is also the basis for the fitness function F, there is no direct connection between its usage in this evaluation and the fitness function. In the following, the variables that are evaluated by an error measure are indicated with a subscript to the error measure, e.g., $\mathrm{MAPE}_{B,u}$ is calculated for the voltage u of the boundary buses B and $\mathrm{MAE}_{G,P}$ is calculated for the active power P for the retained generators G.

In order to interpret the results in the following sections, we define quality criteria for the different variables. The frequency variables have the lowest margin for error since the allowed frequency band in the European grid is narrow. For example, starting with frequency deviations of $0.2\,\mathrm{Hz}$, manual countermeasures are deployed in the German grid [129]. With a $50\,\mathrm{Hz}$ base frequency, this results in deviations of $0.4\,\%$. In order to allow statements regarding frequency violations, the errors introduced by the model reduction should be an order of magnitude smaller than this critical deviation. Similar criteria apply for the voltages, however with more relaxed allowed operation ranges. The German transmission grid levels, for example, have an allowed voltage band with a width of around $7\,\%$ to $10\,\%$ for the extra high and high voltage level, respectively [129]. Again, approximation errors should be ten times smaller than half of this range, i.e., around $0.5\,\%$. For the active and reactive power of the generators, the criteria are less clear. For many applications, a low single-digit percentage error should be fine, but there might be use cases with stricter requirements. In order to assess the system stability, the reduced model $\hat{M}$ should be unstable in the same conditions, e.g., events, as the original model M.

IEEE 39 Bus System

This section describes the results of the parameter optimization for the reduced IEEE 39 Bus System. It provides insights on the selection of objectives for the optimization and the effect of the chosen events that are used to describe the system behavior.

The comparison between different equivalencing methods is very uncommon in the literature, probably due to the lack of information on the used models and missing details regarding the methods, significantly complicating a reimplementation of those methods. We establish a base level of comparability with other methods by two means:

1. We use the IEEE 39 Bus System with the same external system definition and define the same short-circuit event as in [93]. Not all details regarding the controllers used in that work are known, preventing a direct comparison. However, the similarity between the two evaluation setups gives a sense of the quality of the results in comparison to this benchmark method from the literature.

2. We utilize knowledge about the original IEEE 39 Bus System and the reduced model to determine a likely suitable controller configuration manually. This configuration is based on the insight that reduced generator *G 01 [R]* (see Figure 7.5b) is equal to the original generator *G 02* (see Figure 7.5a), thus

choosing the same controller setup for the reduced generator. The controllers for the other reduced generator are selected according to the rating of the aggregated generators. A detailed description of the resulting configuration is provided in Appendix A.3.1. In the following, this is described as the *Expert* configuration and serves as a benchmark method using the exact same evaluation as our proposed GA-based method.

Comparison of Objective Variable Selection The goal of the following comparison is determining the effect of the selection of objective variables $\mathcal{V}$ on the resulting equivalent models. For this purpose, we define five sets of objective variables that are used to perform the parameter optimization:

- *BBus*$_{u,f}$: This configuration includes the voltage and frequency at the boundary buses connecting the AoI to the external system. This configuration is used in [93].
- *BBus*$_f$: This option focuses on the frequency (f) at the boundary buses only.
- *BBus*$_u$: These objectives include the voltages (u) at the boundary buses only.
- *Gen*$_{P,Q,f}$: This option ignores the boundary and focuses on the active power (P), reactive power (Q), and frequency (f) of the generators in the retained system.
- *Gen*$_{P,Q,f}$+*BBus*$_{u,f}$: This configuration combines the objectives of the boundary buses and generators, making it the configuration with the most objectives in this setup.

These five sets of variables are evaluated by performing the parameter optimization with otherwise identical settings: All optimizations in this section are performed with a population size $\mathsf{N_P}$ of 50 and a number of generations $\mathsf{N_G}$ of 50. Furthermore, all optimizations are performed with a single event describing the system behavior, i.e., the short-circuit with the same parameters as in [93]. In order to reduce the effect of randomness in the optimization process and to gain more robust results, all optimizations were repeated ten times, using the mean values of these independent runs for the comparison.

Figures 7.7 and 7.8 show exemplary curves of the voltage and frequency at boundary *Bus 02*, and the active and reactive power as well as the frequency of generator *G 08*. These curves represent the responses to the simulated short-circuit of individual optimization runs, and are shown for the five objective variable sets, the *Expert* solution, and the original model M. Overall, the optimized equivalents produce

system responses very similar to each other and to the original, detailed model. However, larger deviations from the original model can be observed during the fault or immediately afterward for the voltage (u) and frequency (f) at boundary *Bus 02* and for the active power (P) of generator *G 08*. On the other hand, while the *Expert* solution, yields very similar results as the optimized models for the immediate fault response and long term behavior, it deviates more drastically from the other models in between. The quantitative evaluation of the different variable sets is described in Table 7.4, showing the error values averaged over the ten optimization runs and all evaluated buses and generators. The optimized models consistently outperform the *Expert* solution with errors smaller by a factor of roughly 2 to 3. The MAPE values reach a maximum of $1.25\,\%$ in the $10\,\mathrm{s}$ after the fault for all evaluated variables except the reactive power Q of the retained generators. This variable exhibits zero-crossings, depriving the percentage error of its significance. Except for Q, all variables are well within the range of acceptable errors, as defined at the start of this section. For the whole simulation, the maximum MAPE drops to $0.44\,\%$. The error values for all objective variable sets are very similar. However, $\mathit{BBus}_{u,f}$ is preferable to BBus_{f} and BBus_{u} as it results in more balanced and overall better results. Interestingly, $\mathit{Gen}_{P,Q,f}$ performs similarly to $\mathit{BBus}_{u,f}$ and outperforms $\mathit{Gen}_{P,Q,f}$+$\mathit{BBus}_{u,f}$ consistently for every variable. The overall similarity of the results, especially in the first second after the simulated events, suggests that the selection of variables for the optimization objectives has a limited effect on the model accuracy. While optimizing for more variables might result in more balanced results, the evaluation shows that simply adding more variables to optimize does not necessarily increase the accuracy of the reduced model $\hat{M}$.

Comparison of Single-Event Optimization and Multiple-Event Optimization In addition to the previously evaluated variables $\mathcal{V}$, the simulation events $\mathcal{E}$ define the objectives that are optimized by the GA. Therefore, the goal of this comparison is examining the effect of the definition and selection of simulation event $\mathcal{E}$ on the resulting optimized equivalent models $\hat{M}$. Intuitively, including a diverse set of events in the optimization procedure should result in more accurate reduced models $\hat{M}$ under a wider range of simulation scenarios. For this reason, we compare equivalent models that are optimized with a *single* simulation event and models that are optimized with *multiple* events. The single-event optimizations utilize the first event described in Section 7.3.1, i.e., the short-circuit, while the multi-event optimizations utilize all four events described in that section. In order to also evaluate the effect of multiple events with different sets of objective variables $\mathcal{V}$, we include $\mathit{BBus}_{u,f}$ and $\mathit{Gen}_{P,Q,f}$+$\mathit{BBus}_{u,f}$ in the evaluation. Tables 7.5 to 7.8 show the resulting errors for

Tab. 7.4.: The mean error values for the boundary buses (B) and generators (G) when simulating the short circuit event at 5 s. The black values depict errors for the 10 s after the event is triggered, marking the most important time frame of the simulation, whereas gray values describe the errors for the whole 30 s simulation time. The error values are in a similar range for all objective variations of the optimization method. For the *Expert* solution, the error values are roughly 2 to 3 times larger than the optimized solutions. For the optimization method, it can be observed that the variants that focus on a single objective (f or u) yield better results for their respective objective at the cost of the other non-optimized objectives. The $BBus_{u,f}$ variant achieves more balanced results outperforming both specialized options on the non-optimized generator objectives. An unexpected result is that $Gen_{P,Q,f}$, which ignores the boundary, consistently outperforms the variant that includes the boundary buses in the optimization.

	$BBus_f$	$BBus_u$	$BBus_{u,f}$	$Gen_{P,Q,f}$	$Gen_{P,Q,f}$ +$BBus_{u,f}$	*Expert*
$MAPE_{B,u}$ [%]	0.3060	0.2701	0.2747	0.2768	0.2790	0.6653
	0.1058	0.0942	0.0955	0.0965	0.0970	0.2380
$MAE_{B,u}$ [p.u.]	0.0027	0.0023	0.0023	0.0023	0.0024	0.0064
	0.0009	0.0008	0.0008	0.0008	0.0008	0.0023
$MAPE_{B,f}$ [%]	0.0087	0.0125	0.0094	0.0094	0.0094	0.0229
	0.0034	0.0047	0.0036	0.0036	0.0037	0.0083
$MAE_{B,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001	0.0001	0.0002
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001
$MAPE_{G,P}$ [%]	1.2486	1.2516	1.0162	0.9265	0.9760	2.3985
	0.4402	0.4392	0.3629	0.3297	0.3487	0.8177
$MAE_{G,P}$ [MW]	4.1242	3.9094	3.3193	2.9991	3.1865	8.4794
	1.4508	1.3736	1.1876	1.0675	1.1370	2.8900
$MAPE_{G,Q}$ [%]	101.9194	84.8410	90.7684	86.3693	81.6711	246.5924
	80.6054	63.1800	77.6439	61.7716	76.4087	190.9301
$MAE_{G,Q}$ [Mvar]	3.3343	3.4801	3.1646	3.0599	3.1655	6.9371
	1.2133	1.2570	1.1497	1.1154	1.1606	2.4651
$MAPE_{G,f}$ [%]	0.0089	0.0120	0.0089	0.0087	0.0089	0.0230
	0.0036	0.0046	0.0035	0.0034	0.0036	0.0083
$MAE_{G,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001	0.0001	0.0002
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001

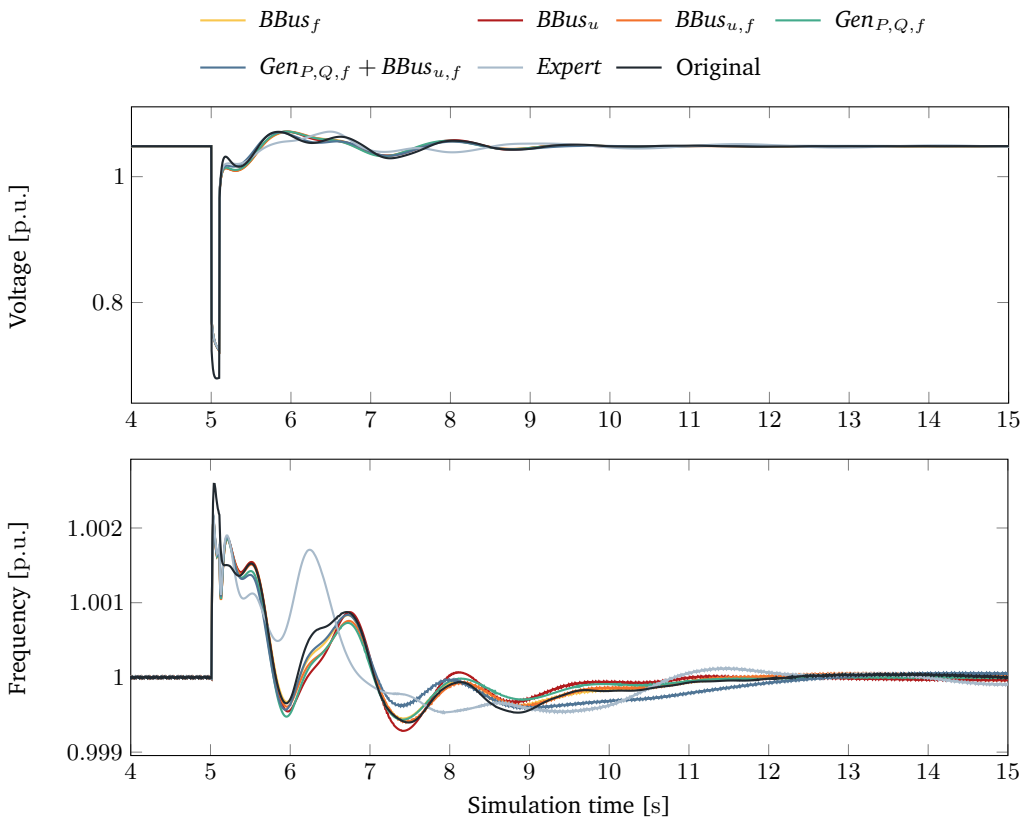


Fig. 7.7.: The voltage (u) and frequency (f) at boundary *Bus 02* produced by one of the ten optimization runs for every objective variable set compared to the *Expert* solution and the original model. For the voltage (top), all optimization variants result in very similar curves with a less drastic drop as the original model. The *Expert* solution differs slightly more from the original. For the frequency (bottom), the differences are clearer. In the first second after the fault (5 s to 6 s) and the last seconds of the shown simulation all solutions are very similar. While most solutions are also quite similar in between, the $Gen_{P,Q,f}$+$BBus_{u,f}$ option exhibits greater deviations to the rest. The *Expert* solution shows significant differences to the optimized equivalents.

the four events, respectively. These tables contain the error values for the first 10 s after the event averaged over ten optimization runs. For the first event (Table 7.5), the single-event optimization consistently achieves better results than their multi-event counterparts. The difference between the single- and multi-event optimization results is generally larger for the $Gen_{P,Q,f}$+$BBus_{u,f}$ variant. For the other three events, the multi-event optimizations yield more accurate equivalent models. This effect is less strong for the second event, however, compared to the third and fourth event. Overall, the loss in accuracy for the first event due to multi-event optimization is in a similar range as the accuracy gains for the third and fourth event. Therefore, depending on the use case for the reduced model, it might be beneficial to optimize for a specific type of event or a small set of events rather to gain more accurate

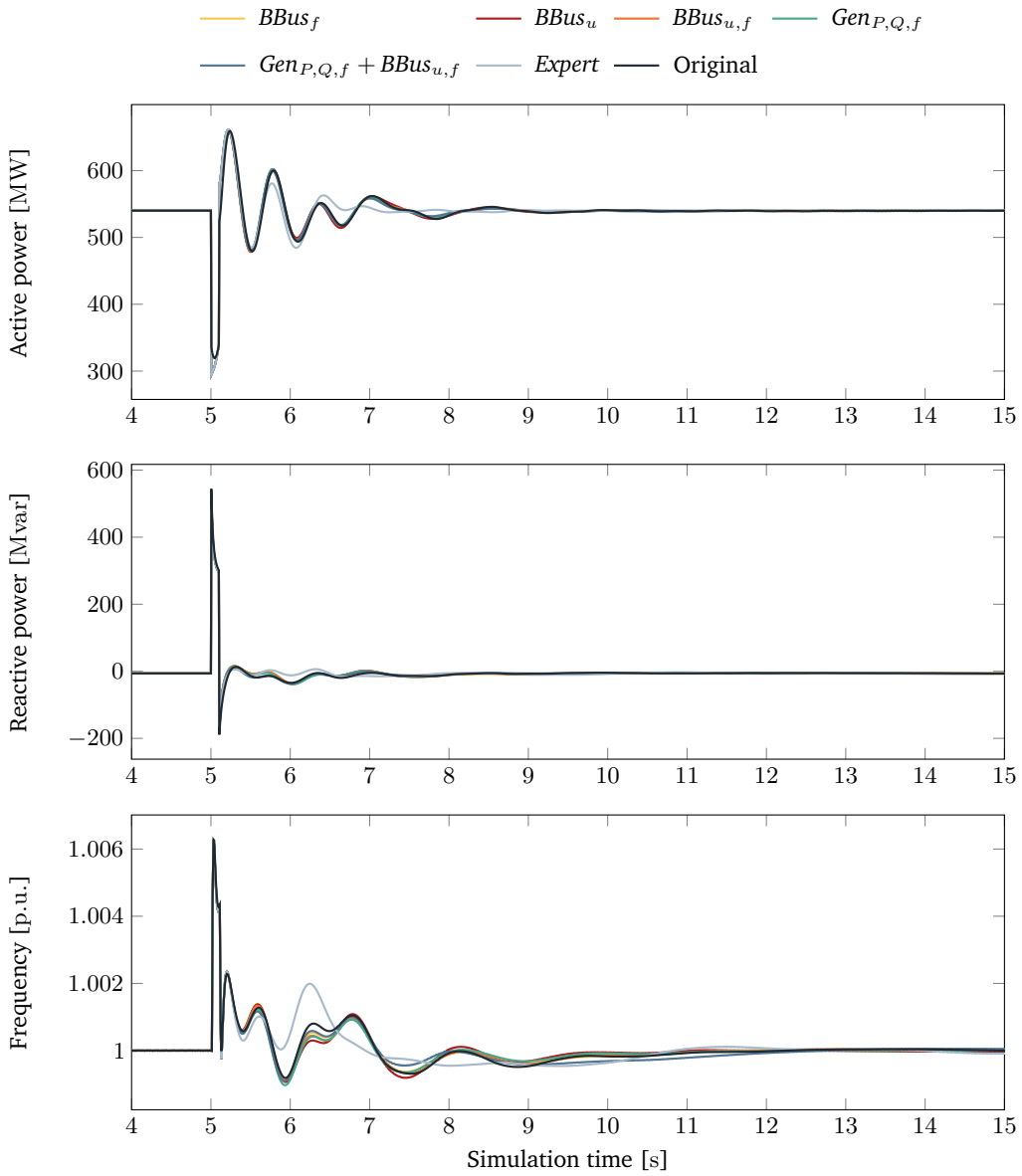


Fig. 7.8.: The active power (P), reactive power (Q) and frequency (f) of the retained generator *G 08* produced by one of the ten optimization runs for every objective variable set compared to the *Expert* solution and the original model. For the active (top) and reactive power (middle), all solutions create very similar curves that are close to the original for most of the simulation time, with only the *Expert* solution showing slightly larger deviations. The immediate response for P, however, shows a more pronounced drop and subsequent jump for all reduced variants. For the frequency (bottom), the differences between the various solutions are more visible. While the differences between the optimized solutions are still very subtle with no clearly dominant solution, the *Expert* solution differs more significantly from the rest.

Tab. 7.5.: First event: three-phase short-circuit. The *single* variations that are optimized using the short-circuit event, only, consistently outperform their *multi*-event counterparts that are optimized for all four events. On average, the *single* MAE values are $15.6\,\%$ and $24.0\,\%$ smaller compared to the *multi* values for $BBus_{u,f}$ and $Gen_{P,Q,f}+BBus_{u,f}$, respectively. All error values are averaged over ten optimization runs.

	$BBus_{u,f}$		$Gen_{P,Q,f}+BBus_{u,f}$	
	Single	Multi	Single	Multi
$\mathrm{MAPE}_{B,u}$ [%]	0.2747	0.3326	0.2790	0.3821
$\mathrm{MAE}_{B,u}$ [p.u.]	0.0023	0.0029	0.0024	0.0035
$\mathrm{MAPE}_{B,f}$ [%]	0.0094	0.0130	0.0094	0.0140
$\mathrm{MAE}_{B,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001
$\mathrm{MAPE}_{G,P}$ [%]	1.0162	1.3406	0.9760	1.3135
$\mathrm{MAE}_{G,P}$ [MW]	3.3193	4.4942	3.1865	4.5085
$\mathrm{MAPE}_{G,Q}$ [%]	90.7684	93.7493	81.6711	123.6288
$\mathrm{MAE}_{G,Q}$ [Mvar]	3.1646	3.6795	3.1655	4.1959
$\mathrm{MAPE}_{G,f}$ [%]	0.0089	0.0128	0.0089	0.0135
$\mathrm{MAE}_{G,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001

Tab. 7.6.: Second event: switch opening. For this event—and the next two events—the *multi*-event optimizations consistently outperform the *single*-event optimizations. Resulting in $6.9\,\%$ and $5.9\,\%$ smaller average MAE values for this case.

	$BBus_{u,f}$		$Gen_{P,Q,f}+BBus_{u,f}$	
	Single	Multi	Single	Multi
$\mathrm{MAPE}_{B,u}$ [%]	0.1740	0.1466	0.1724	0.1556
$\mathrm{MAE}_{B,u}$ [p.u.]	0.0018	0.0015	0.0018	0.0016
$\mathrm{MAPE}_{B,f}$ [%]	0.0061	0.0049	0.0060	0.0051
$\mathrm{MAE}_{B,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001
$\mathrm{MAPE}_{G,P}$ [%]	0.2042	0.1976	0.2023	0.1846
$\mathrm{MAE}_{G,P}$ [MW]	0.8022	0.7608	0.7954	0.7249
$\mathrm{MAPE}_{G,Q}$ [%]	91.0065	83.3476	91.8131	86.6038
$\mathrm{MAE}_{G,Q}$ [Mvar]	2.5519	2.2242	2.5051	2.2675
$\mathrm{MAPE}_{G,f}$ [%]	0.0061	0.0049	0.0059	0.0051
$\mathrm{MAE}_{G,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001

results compared to a more general optimization approach. However, no matter which events are used to optimize the controller parameters, all resulting models yield results in the range of acceptable errors.

Tab. 7.7.: Third event: load step change. For this event, the advantage of the *multi*-event optimizations is larger with 15.3 % and 23.0 % smaller average errors.

	$BBus_{u,f}$		$Gen_{P,Q,f}+BBus_{u,f}$	
	Single	Multi	Single	Multi
$MAPE_{B,u}$ [%]	0.0624	0.0472	0.0551	0.0427
$MAE_{B,u}$ [p.u.]	0.0007	0.0005	0.0006	0.0004
$MAPE_{B,f}$ [%]	0.0130	0.0085	0.0134	0.0077
$MAE_{B,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001
$MAPE_{G,P}$ [%]	0.1730	0.1533	0.2232	0.1239
$MAE_{G,P}$ [MW]	0.7330	0.6255	0.9441	0.4983
$MAPE_{G,Q}$ [%]	11.6490	8.1998	9.9218	6.7764
$MAE_{G,Q}$ [Mvar]	1.5072	1.0071	1.1588	0.7574
$MAPE_{G,f}$ [%]	0.0130	0.0085	0.0134	0.0078
$MAE_{G,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001

Tab. 7.8.: Fourth event: load ramp change. This event shows a similar effect as the third event with 12.5 % and 19.4 % smaller MAE values for the *multi*-event optimizations.

	$BBus_{u,f}$		$Gen_{P,Q,f}+BBus_{u,f}$	
	Single	Multi	Single	Multi
$MAPE_{B,u}$ [%]	0.0569	0.0436	0.0503	0.0405
$MAE_{B,u}$ [p.u.]	0.0006	0.0005	0.0005	0.0004
$MAPE_{B,f}$ [%]	0.0120	0.0078	0.0123	0.0071
$MAE_{B,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001
$MAPE_{G,P}$ [%]	0.1586	0.1431	0.2049	0.1204
$MAE_{G,P}$ [MW]	0.6687	0.5854	0.8649	0.4910
$MAPE_{G,Q}$ [%]	38.7154	21.9385	26.7607	17.0128
$MAE_{G,Q}$ [Mvar]	1.3592	0.9071	1.0393	0.6887
$MAPE_{G,f}$ [%]	0.0120	0.0078	0.0123	0.0071
$MAE_{G,f}$ [p.u.]	0.0001	0.0001	0.0001	0.0001

Tab. 7.9.: The time to simulate each event in the BW model for 120 s. While the short-circuit simulation is significantly faster with the reduced model than the original, the other two events are slightly slower with the reduced model. The short-circuit results in oscillations in the system for the whole simulation time for both models. For the other two events, the original quickly reaches a steady operation point as opposed to minor oscillations in the reduced model.

	Original [s]	Reduced [s]	Difference [s]	Rel. diff. [%]
Short-circuit	412.09	256.17	−155.92	−37.8 %
Load step	42.85	49.98	+7.13	+16.6 %
Line dis-/connection	46.98	51.45	+4.47	+9.5 %
Total	501.92	357.6	−144.32	−28.8 %

Baden-Württemberg Transmission Grid

As opposed to the IEEE 39 Bus System, the BW transmission grid does not serve as a benchmark system, but as a realistic case study to demonstrate the real-world usability of the proposed controller optimization method. The first notable difference between the systems is their size: While only two generator controls need to be parameterized in the reduced IEEE system, the reduced BW system has 22 generators in its external area. Working with the BW model has shown that finding controller configurations that lead to a stable system—even without any grid events—is much more challenging. Thus, the population size $\mathsf{N_P}$ is increased to 100, allowing for more stable solutions in the initial population. Utilizing the information on the controls of the original generators helps to increase the chance of finding suitable configurations in the initial population. Thus, we initialize half of the first population using the controller information provided by the TPNR method and the other half completely at random.

In the following, we describe our findings when optimizing the reduced BW model, using different approaches to the definition of simulation events and selection of objective variables. Due to the identified difficulty of finding controller configurations that lead to a stable system, the following three settings demonstrate our approaches to optimize this model:

- **No event**: This uses an RMS simulation without any events to optimize the reduced model with the optimization parameters determined in Section 7.3.2 for $\mathsf{N_G} = 50$ generations. Initial trial optimizations of the reduced BW model have shown that reducing the number of objectives can be beneficial to the optimization process. Thus, for the objective variables, only two boundary buses (u, f) and the slack generator in the retained system (P, Q, f) are selected.

- **Multi-event**: This approach uses the same objective variables and optimization parameters as the previous approach. However, it also uses the three other events described in Section 7.3.1 for the optimization.

- **Hybrid**: This approach combines the two approaches by using the final generation produced by the first approach as the initial population for a subsequent fine-tuning with all four simulation events. For this fine-tuning, less aggressive parameters are selected to not alter the presumably already good solutions too drastically. Specifically, the parameter set $(0.2, 0.2, 0, 0.1)$ is selected to disable the type mutation of controllers and limit all other genetic variation.

In addition to these three optimization approaches that mainly differ in the selection of events $\mathcal{E}$, two additional variations are tested with slightly different objective variables $\mathcal{V}$. The detailed quantitative results of these five approaches are provided in Appendix A.3.2. For the hybrid optimization approach, these results confirm the observation that fewer objective variables might benefit the accuracy of the resulting model. For the multi-event optimization approach, this advantage is less clear. However, as the limited number of objective variables is no disadvantage either, the following analysis focuses on the approaches using two buses for the objective variables only. The hybrid optimization approach consistently outperforms the direct multi-event optimization, often significantly. While the no-event optimization yields even better results for many of the evaluated system variables, it shows exceptionally poor results for the load step simulation. Being similarly accurate and more robust makes the hybrid approach the optimization method of choice. Excluding the $\mathrm{MAPE}_{G,Q}$ of the short-circuit simulation ($15.5\,\%$) that is skewed by zero-crossings, all MAPE values of the hybrid approach are below $8.36\,\%$, which is the value of the $\mathrm{MAPE}_{G,Q}$ for the load step change. The MAPE values for all other objectives lie below $0.37\,\%$—well in the range of acceptable errors—, indicating a very good approximation of the detailed model.

The quantitative results are averaged over the whole simulation time of $10\,\mathrm{s}$, and thus, they do not give detailed insights into the detailed differences between the optimized reduced models and the detailed model. Therefore, we analyze the response curves for selected variables of the optimizations with the different approaches that are presented in Figures 7.9 to 7.12. These plots reveal that the initial responses to the simulated events and the last seconds of the simulations are very similar for most optimization approaches and close to the original model. There are three notable exceptions to this observation: First, the multi-event optimized model shows larger deviations for the simulation without events (Figure 7.9) and, second, for the simulated line dis-/connection, especially regarding the reactive power (Figure 7.12). Third, Figure 7.11 confirms the finding in the quantitative results that the no-event optimization performs poorly for the load step event. While the initial response is similar to the other optimized models, this variant starts oscillating heavily around the $5\,\mathrm{s}$ mark. Further simulations reveal that this oscillation continues for at least $60\,\mathrm{min}$ with the frequency slowly drifting higher. The analysis of these detailed curves confirms the hybrid optimization approach as a robust choice that handles all evaluated events well. It approximates the original BW model almost perfectly after $6\,\mathrm{s}$ to $7\,\mathrm{s}$ of simulation time and exhibits only small deviations in the first seconds after the simulated events, which should be negligible for many applications.

Regarding the simulation times, the reduced number of components in the model does not necessarily translate to faster calculations in PowerFactory, as shown in Table 7.9. While the short-circuit simulation benefits massively from the reduction, the other two simulations are actually slower with the reduced model. This is likely due to the original model reaching a less dynamic state, which can be calculated faster by PowerFactory. Overall, however, performing the three simulations—each with $120\,\mathrm{s}$ of simulated time—is $144.32\,\mathrm{s}$ ($28.8\,\%$) faster with the reduced model.

7.4 Discussion

In this section, we discuss the advantages and limitations of the proposed GA-based generator control optimization for reduced equivalent models. The proposed method achieves a good approximation for the simple IEEE benchmark system as well as for the complex and more detailed BW transmission grid model with hundreds of nodes. This approximation is realized by using standard controller models only, ensuring a wide compatibility with common modeling software. The evaluation demonstrates that the proposed control optimization can be combined with different static reduction methods to fit the needs of the use case at hand.

The similarity of all results for the IEEE 39 Bus System suggests that there is a structural limitation of the reduced model that prohibits a more accurate approximation of the original system. This is especially apparent for the first second after the short-circuit is triggered, where all solutions have virtually the same response. Tests with a higher number of generations N_G have shown little improvement, supporting this assumption. This might limit the significance of the results for this benchmark case. However, the evaluations still yield tendencies for the influence of certain optimization settings on the resulting solutions and shows that the optimization yields significant improvements compared to a parameterization based on expert knowledge. Optimizing the more complex BW transmission grid model has shown that the selection of objectives for the optimization can have a significant influence on the results, especially the events, and that finding suitable settings is not entirely straightforward. In general, optimizing the system behavior to fit the response to certain selected events, does not give any guarantees for the behavior and accuracy of the reduced equivalent model. A drastic example for this is the response of the no-event optimized model to the load step shown in Figure 7.11. Including different events in the optimization can mitigate this issue to some extent. In conclusion, the large-scale demonstration with the BW system—that requires a controller configuration for 22 generators, resulting in 1364 parameters—shows the effectiveness

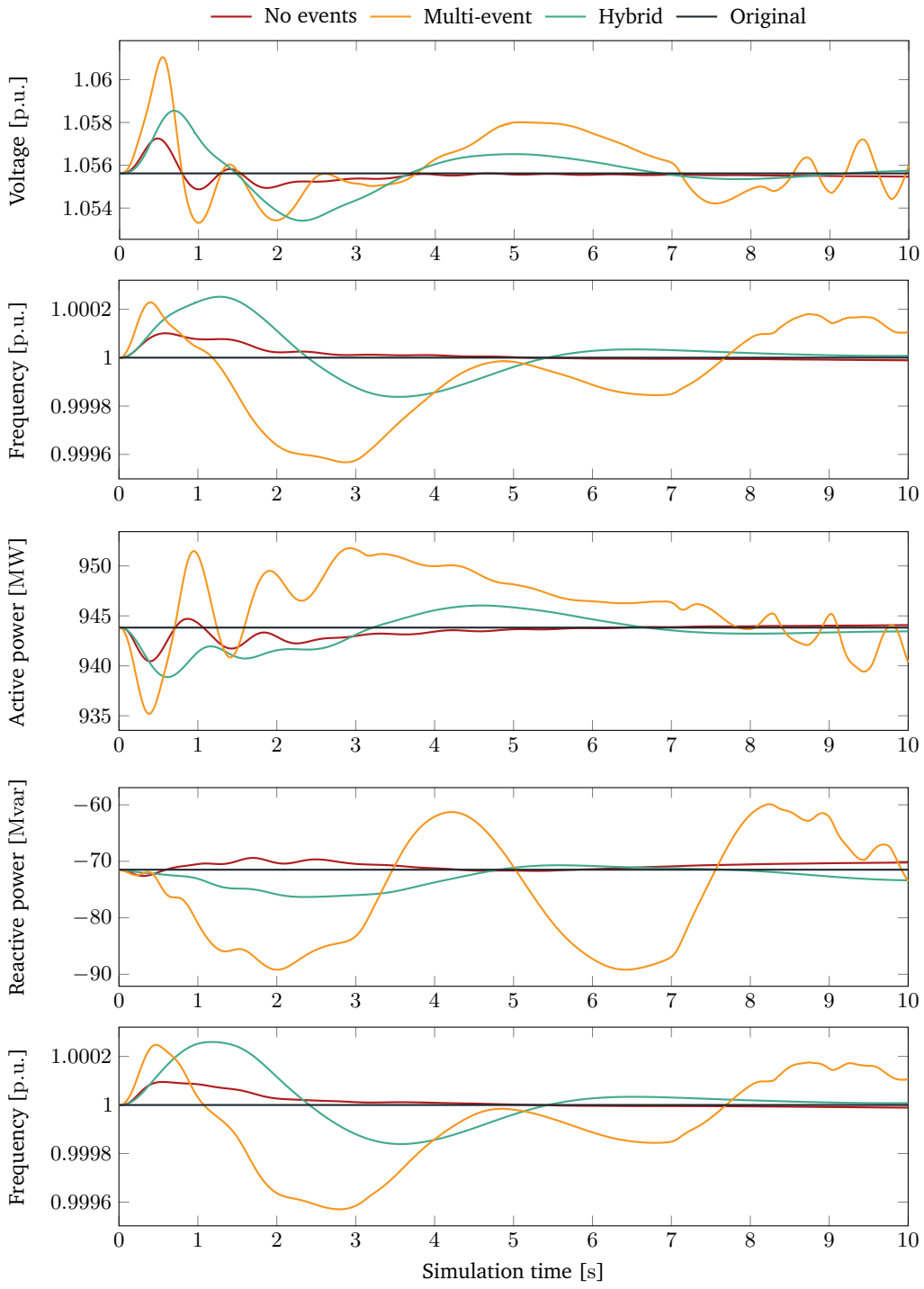


Fig. 7.9.: The simulation of the BW model without any system events shows that the original model is perfectly steady. When optimizing for no events only, the reduced system comes quite close to the original. When optimizing directly for all four events, the resulting system shows much more fluctuation. The hybrid optimization results in a much better approximation compared to the direct multi-event optimization.

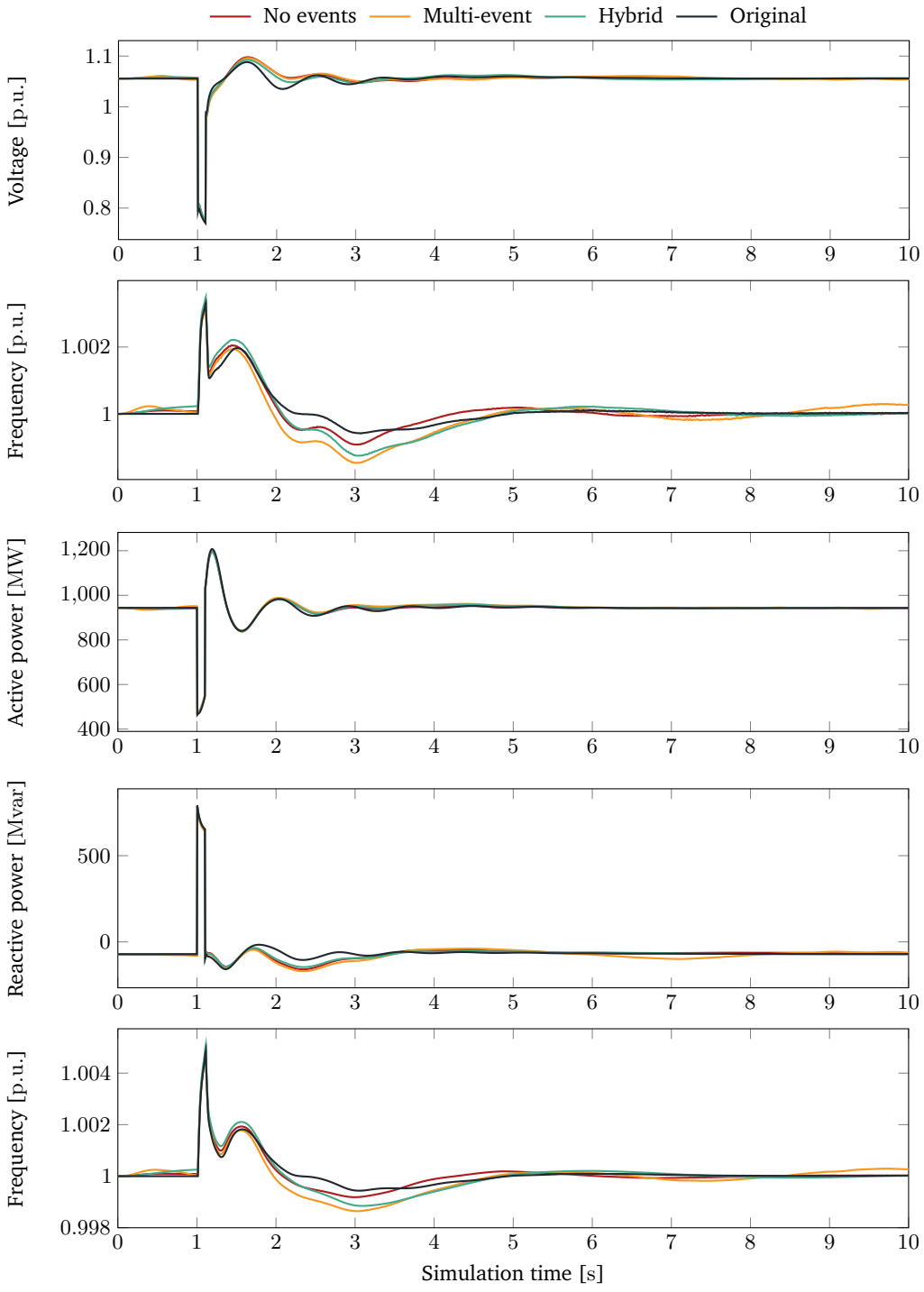


Fig. 7.10.: For the short-circuit simulation, all three reduced models produce results close to the original surprisingly with a slight advantage for the no-event optimization that did not include this event in its optimization process.

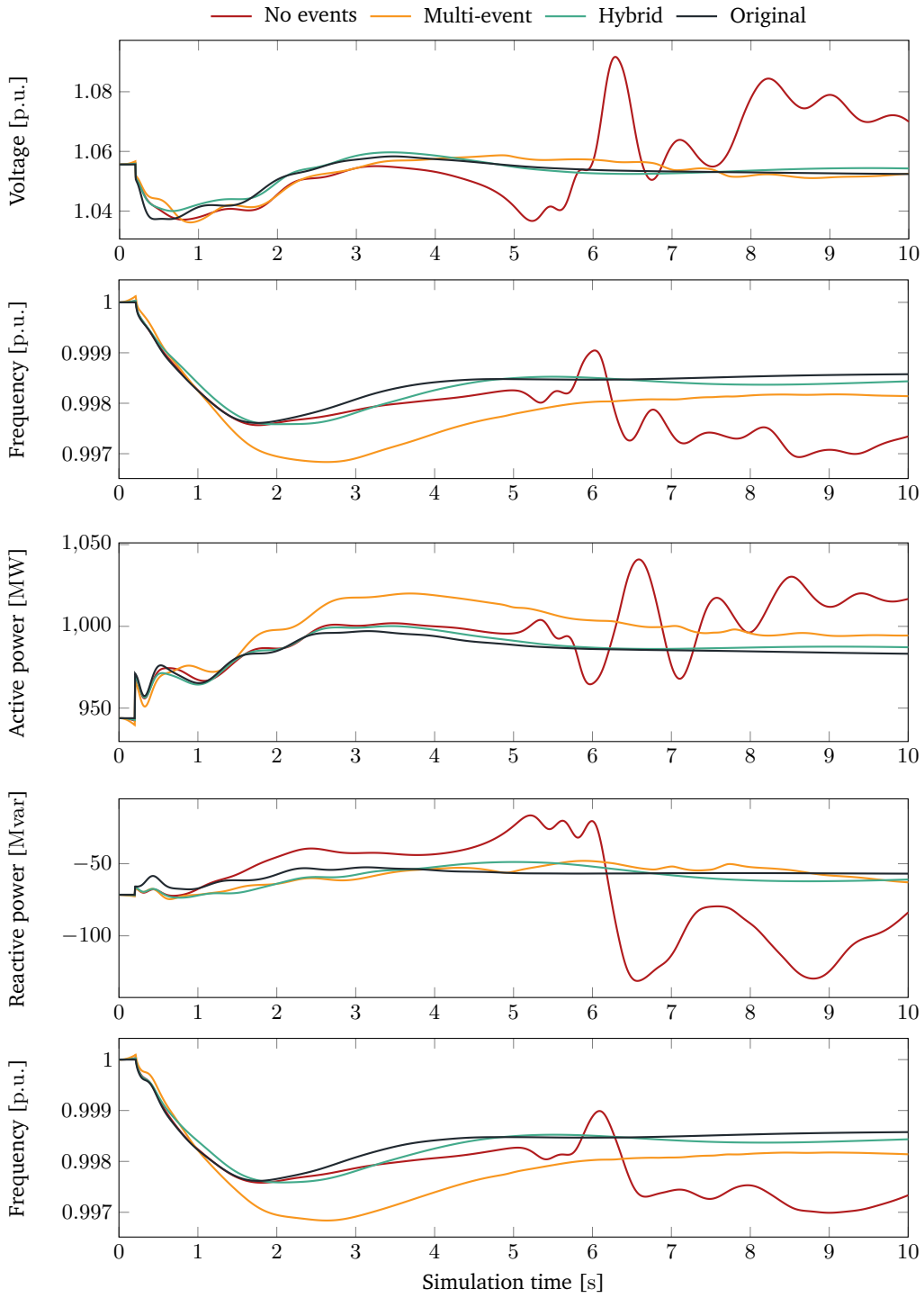


Fig. 7.11.: The simulation of the load step change shows the main disadvantage of the no-event optimization: Optimizing only the no-event simulation comes with the risk of very poor performance for unseen events in the system. A continued simulation reveals that this model keeps oscillating with a steady drift in the observed variables, e.g., reaching a frequency of $1.004\,\mathrm{p.u.}$ after roughly $60\,\mathrm{min}$. Additional fine-tuning with multiple events (Hybrid) can mitigate this issue, and in this case results in a significantly better performance than directly optimizing for multiple events.

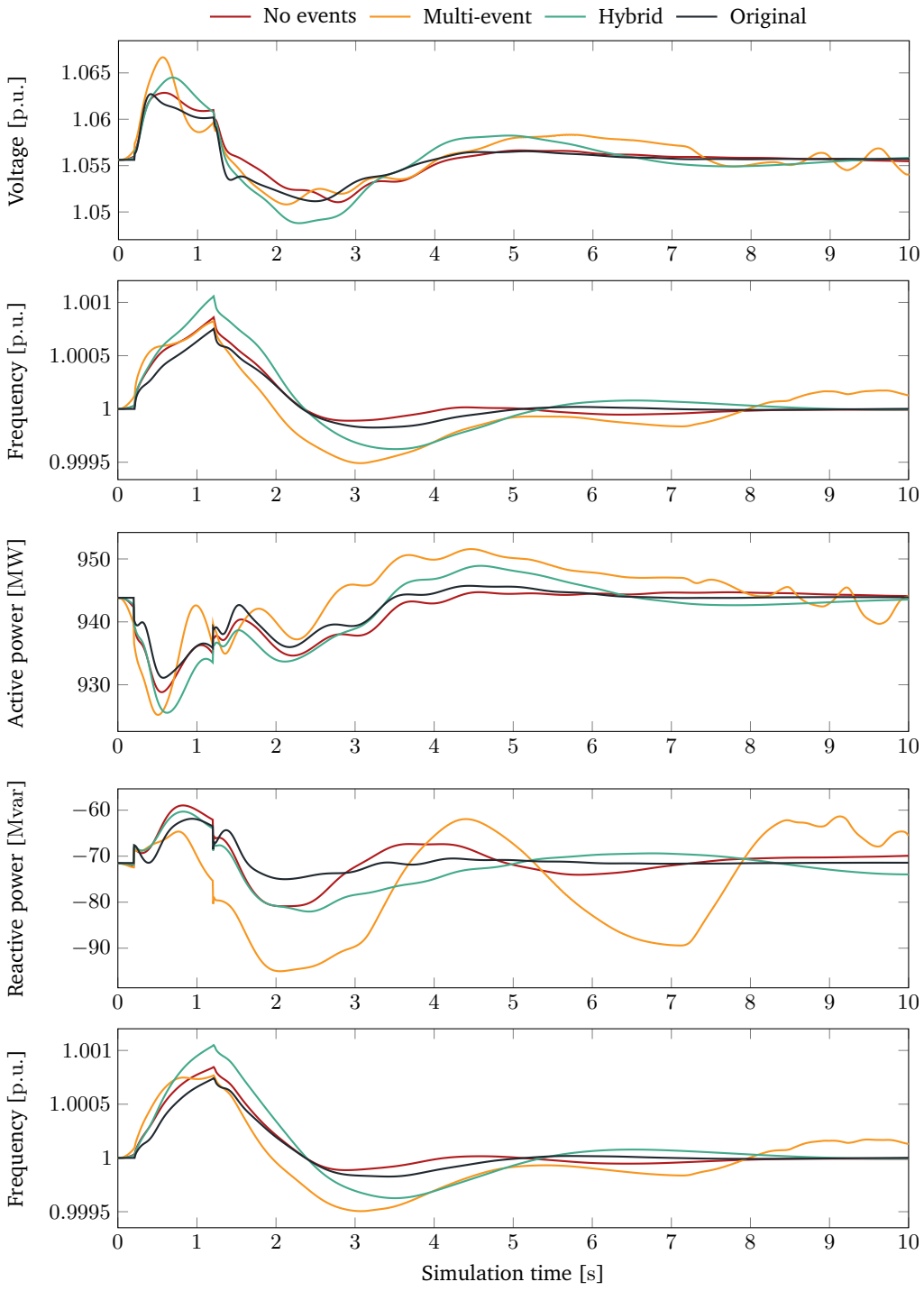


Fig. 7.12.: Disconnecting and reconnecting the transmission line is best approximated by the no-event optimization, closely followed by the hybrid approach with subsequent multi-event fine-tuning. The direct multi-event optimization performs worst for all objectives—especially for the reactive power of the generator. The last seconds of the simulations also hint at the system becoming unstable after the 10 s of simulation time.

of the proposed method to find control parameter combinations that adequately approximate the behavior of the detailed system.

The parameter optimization of both models—the simple benchmark case, and the realistic large-scale case—results in reduced models that approximate the dynamic behavior of their detailed counterparts with high accuracy. The observed errors for most variables are well in the acceptable range defined in Section 7.3, allowing to gain insights about the detailed model by simulating the reduced model. However, the proposed GA-based optimization shows difficulties approximating the reactive power of the retained generators. Introducing weighting factors into the summarizing fitness function for the selection of best solution might be able to mitigate this issue to some extent, but further evaluations are required to find the cause for these difficulties.

Table 7.9 shows that the reduction does not always result in faster simulation times using PowerFactory. The reduction of a model with the size of the BW transmission grid model can take several hours to reach satisfying results. Thus, it is not given that the reduction saves simulation time, if the number of simulations that need to be performed is low. However, for the three evaluated events, the reduction saves around $29\,\%$ of simulation time. Additionally, reducing the number of components to roughly half of the original model can be the deciding factor to enable real-time simulation, as real-time simulators, such as RTDS, have a fixed capacity that supports a limited number of components.

While the proposed method yields promising results, the parameter optimization is currently limited to three controller types for synchronous machines. However, these are not the only components that influence the dynamic behavior of a power system. With an increasing share of generation from renewable energy sources, inverter-based generation and the corresponding control becomes an increasingly important factor in the dynamic system behavior.

7.5 Contribution and Future Work

In this chapter we answer research question **RQ5** that addresses the reduction of large transmission grid models for dynamic simulations. To answer this question, we propose a new genetic algorithm (GA) for the optimization of synchronous machine controllers in statically reduced network equivalents to approximate the behavior of their detailed counterparts. While this optimization does not reduce the model complexity itself, it can be combined with static reduction methods, such as REI and

TPNR to created models with a reduced complexity that are capable of dynamic simulations. We combine our proposed optimization method with both of these static methods, and evaluate these combinations with a simple benchmark system as well as a complex, real-world transmission grid model to also cover a realistic use case of the method. The method is evaluated systematically with a series of varying settings and validated against a system-knowledge-based *Expert* solution.

In summary, this chapter provides the following contributions:

- We introduce novel genetic operators specifically designed for synchronous machine controls, including a flexible representation, crossover, and mutation. These operators are integrated into a multi-objective optimization to approximate the behavior of a detailed model with a statically reduced equivalent model.
- We perform a comprehensive evaluation using the IEEE 39 Bus System as a common benchmark model to determine the influence of several optimization parameters and choices, such as selection of objectives and definition of simulation events, and to provide comparability to methods found in the literature.
- We demonstrate the real-world applicability of the method on a realistic large-scale model of the Baden-Württemberg transmission grid in combination with the previously introduced static TPNR method.

The current optimization method is limited to a small set of synchronous machine controller models. Including more common controller models and even extending the method to other types of controllers, e.g., for converters, likely broadens the applicability and accuracy of the method. Optimizing the parameters of the BW model shows that further research on the influence of optimization settings, such as objectives, simulation events, and parameters of the genetic operators might be beneficial for the resulting equivalent models. Ideally, this would lead to an automated selection of suitable settings. The heuristic nature of the proposed method does not give any guarantees regarding the accuracy of the optimized reduced model. Thus, an automated tool for systematic testing and reporting could help to gain more insights into the model accuracy and limitations, leading to an increased confidence in the model. Lastly, the optimization method itself could be enhanced by utilizing more sophisticated techniques, such as adaptive mutation and crossover probabilities, adaptive completion criteria, and the newer NSGA-III genetic algorithm [130].

Summary Part III

In this section, we address the two research questions posed by the challenges in transmission grids highlighted in the introduction of this thesis. The contributions of this section to answering these questions and mitigating the challenges are summarized as follows:

RQ4 How can transmission grid network models with considerable amounts of renewable generation be reduced, while retaining crucial geographic information?

We introduce the novel static topology-preserving network reduction (TPNR) method that uses a rule-based reduction approach on an abstract network graph, that enables tracing the network components from the detailed model to the reduced equivalent. The TPNR method is compared to established reduction methods to confirm its validity.

RQ5 How can computationally expensive transmission grid models be simplified to support dynamic simulations, enabling HiL testing for crucial new grid components?

We propose a new genetic-algorithm-based optimization method for synchronous machine controls to approximate the dynamic behavior of detailed power system models with their reduced equivalents that can be obtained by applying different static reduction methods, such as REI and TPNR. This approach is thoroughly evaluated on a simple benchmark system as well as on a comprehensive, real-world transmission grid model.

Part IV

Conclusions

8 Summary

The goal of mitigating climate change demands a reduction of carbon emissions. This reduction requires a transition from fossil fuel-based to renewable energy sources in all sectors of society, ranging from energy to industry, transportation, agriculture, and buildings. For the energy sector, this means shutting down large coal, oil, and gas generators and replacing them with renewable energy generation, such as wind turbines and photovoltaics (PV) systems. This drastically changes the production of energy since generation from renewable sources is volatile, heavily weather-dependent, and might not be available in abundance close to large centers of consumption. For the other sectors, decarbonization often means switching to electric energy. This electrification significantly alters the consumption of electric energy, posing major changes and accompanying challenges to all levels of electric power grids, from transmission grids down to the distribution. Overcoming these challenges requires identifying issues and developing suitable solutions, which demands understanding the systems and testing the potential solutions. Since supply security is of the highest priority for power grids, identifying issues by stressing the real system and developing new solutions in the field is not an option. Instead, comprehensive simulations of the power grids are required. While power grids have been simulated for decades, solving the current challenges still leaves open questions, some of which are answered in this thesis.

Distribution grids transitioning from purely supplying consumers with energy to connecting prosumers with bidirectional power flows to the grid requires more comprehensive studies of those grids. These new circumstances require insights into the grid topology and operation, such as the voltage levels and equipment loading. Measurements of energy generation, consumption, and other variables are an important building block in understanding current distribution grids. However, due to errors and planned outages, recorded time series of those measurements are prone to missing values. This is problematic for many applications, leading to:

RQ1: How to handle missing values in energy time series? In Chapter 3, after identifying a lack of imputation methods specifically for energy time series, we introduce the new copy-paste imputation (CPI) method. This method works by copying and pasting chunks of data into gaps by exploiting periodicities, finding suitable data with similar properties, such as weekday, day of the year, and energy

demand. We compare this new methodology to several benchmark methods on a real-world dataset to confirm its effectiveness and accuracy.

Investigating current grid conditions is not sufficient to solve the challenges posed to distribution grids. Potential future scenarios need to be examined and solutions need to be evaluated, which requires models of the corresponding grids. However, these models are often only kept by operators and are not available to academia and third parties working on solutions. Thus, the need for automatically generated distribution grid models—preferably with minimal data requirements, as the available data varies from region to region. This leads to:

RQ2: How can realistic grid models be generated with minimal available data sources? In Chapter 4, we introduce a novel two-stage optimization-based method for generation of residential distribution grids. This method uses OpenStreetMap (OSM) street data as a basis for the grid layout and various data sources, from 2D and 3D building data to electricity meter (EM) data, for the load estimation, which is crucial for the grid configuration. We evaluate the method using a small residential area and demonstrate the large-scale potential of the method with a city-size case study that incorporates nearly $45\,000$ buildings.

When automatically generating distribution grid models from different data sources, the quality and availability of the data likely have a high impact on the resulting grid models, leading us to:

RQ3: How does the availability of data influence the accuracy of the generated grid models? In Chapter 4, we perform the evaluation of our grid generation method with six different combinations of data sources and compare the resulting models to each other as well as the real DSO-supplied grid layout. As the comparison of distribution grid models is not an obvious task, we propose several metrics ranging from topological to electrical and geographic properties. We find that models created using 3D building data as a basis are similar to models using the number of electricity meters per building for their load estimation. Both of these approaches come close to the actual grid layout of the evaluated area. Purely 2D building data, however, results in poor model accuracy and is not sufficient to generate realistic models.

The previous questions focus on the challenges of distribution grids. However, the transition to renewable energy sources also poses significant challenges on transmission grids, requiring simulations of these systems to develop new solutions. Transmission grids, however, are large, complex systems that are often too computationally expensive to be simulated in detail. As such, model equivalencing and specifically model reduction are research topics with a long history. However, in Chapter 5, we identify crucial issues for these topics. Traditional network reduction

methods for power grids do not consider geographic information that becomes increasingly important for the power grids transitioning to renewable energy sources. This poses the question:

RQ4: How can transmission grid network models with considerable amounts of renewable generation be reduced, while retaining crucial geographic information? In Chapter 6, we introduce the topology-preserving network reduction (TPNR) method for static equivalents that enables tracing back all components in the reduced equivalent model to their counterparts in the original detailed model. This allows, for example, to map the locations of renewable sources in the detailed model to the resulting sources in the reduced model. We evaluate this new method using a real-world transmission grid model and compare it to established reduction methods.

The TPNR method introduced in this thesis enables the reduction of large transmission grid models while considering geographic and topological properties of the grid. However, this method does not consider the dynamic behavior of the reduced system, which is essential for testing new grid components in hardware-in-the-loop (HiL) setups. These HiL are crucial for the development of new components and the reduction of complex power grid models can be the deciding factor if a new component can be evaluated under realistic grid scenarios, leading to:

RQ5: How can computationally expensive transmission grid models be simplified to support dynamic simulations, enabling HiL testing for crucial new grid components? In Chapter 7, we introduce a novel genetic-algorithm-based method to optimize the synchronous machine controllers in the reduced model to approximate the dynamic behavior of the original model. We propose new genetic operators specifically for synchronous machine controls and combine them with a variation of the common NSGA-II genetic algorithm. The method is thoroughly evaluated using a simple benchmark system, and its real-world applicability is confirmed with a large-scale transmission grid model.

In conclusion, by developing new methods to answer the five research questions discussed above, the work presented in this thesis helps researchers to better realize the potential of their current resources: whether by making incomplete energy data available for more applications by filling the gaps, or by creating distribution grid models from available data. The proposed data-driven methods for distribution grids allow researchers to make better use of current available data sources. When it comes to transmission grid simulations, the presented reduction methods enable researchers to simulate larger grids in less time. This potentially enables real-time and HiL simulations for previously prohibitively large network models with the same computational resources.

9 Outlook

This thesis contributes to solving the challenges in electric distribution and transmission grids posed by the transition from fossil fuel-based to renewable energy sources by answering the research questions defined in Section 1.3. However, answering these questions by developing new methodologies that solve identified issues in existing approaches and fill a research gap has also shown paths for further improvement.

The two-stage optimization-based method for the generation of distribution grid models yields convincing results for residential areas, that are—depending on the available data—close to the real grid. However, there are three main aspects that can be further improved: the range of applicability and accuracy. The current load estimation method, which is the basis for the network generation, focuses on residential buildings. Developing more sophisticated estimation methods, optionally using more widely available data, is an important step to expand the applicability beyond residential areas. This additional data could include building types defined in OSM, or using image recognition to identify additional properties of buildings, such as the height, age, and condition of a building, as well as its usage, i.e., residential, commercial, or industrial. Simultaneously, the grid layout optimization likely needs to be extended to properly handle large (industrial) consumers and producers with potentially multiple transformers and cables with a higher transmission capacity. For a higher accuracy and realism of the generated grids, time series of energy demand and generation could be used directly where available, e.g., when modeling a specific district. Furthermore, individually generated artificial load curves could produce more realistic models than the currently used H0 standard load curves. The grid layout optimization offers several areas to improve the accuracy and realism of the generated grids. The proposed method generates radial networks without any switching capabilities. However, in reality, multiple grid topologies are used depending on the area, e.g., meshed networks for densely populated city centers. Integrating switchgear would extend the supported simulation use cases to testing reconfigurable topologies. Another area for improvement is the generation of medium-voltage (MV) grid models, especially in rural areas, where overhead lines are used that often do not strictly follow any street layouts.

The methods for static and dynamic reduced equivalent models introduced in Part III have shown to fill an important research gap identified in the existing model reduction methods. However, the evaluation of both methods revealed various directions for further developments. Starting with the static topology-preserving network reduction (TPNR) method, increasing the configurability could significantly improve the real-world applicability of the method. Such configurable options could include the reduction of higher degree nodes and negligible lower voltage levels to gain a higher reduction rate. TPNR relies on a specific aggregation strategy for every type of power system component, e.g., loads, generators, and lines. Thus, extending the set of supported component types increases the applicability to more diverse power system models. Lastly, since most reduction operations are of heuristic nature, their application can lead to errors in the reduced model. An automated process to detect such destructive operations with a configurable error threshold, rewinding the operations and avoiding them in the further reduction process, could significantly increase accuracy and trust in the reduced models. For the optimization of dynamic equivalents, increasing the applicability is also an important direction for further development. This could be achieved by supporting more synchronous machine controller models in the genetic representation, potentially also increasing the accuracy, and supporting more controllers in general, e.g., for HVDC converters and wind turbines. An automated and potentially adaptive selection of objectives and definition of simulation events would reduce the required user input to perform the reduction and might be able to increase the accuracy of the optimization. Finally, the optimization itself offers potential for improvement, for example by using adaptive probability parameters for the genetic operators or utilizing newer genetic algorithms, such as NSGA-III.

The methods developed as part of this thesis demonstrate their contribution to answering urgent research questions regarding the current challenges in electric power grids. A direction for further development shared by all methods proposed in this thesis is extending their applicability: bringing them from benchmark use cases and selected real-world examples to researchers and industry working on solving the challenges on the path toward carbon neutrality.

Bibliography

[1] United Nations. *Paris Agreement*. Dec. 12, 2015 (cit. on p. 2).

[2] Mark R. Bülow, Andrey Litnovsky, Andrea Meyn, et al. *Helmholtz Energy Transition Roadmap : HETR*. Helmholtz, 2024. DOI: 10.5445/IR/1000172546 (cit. on pp. 2, 4, 5).

[3] *Installed Power | Energy-Charts*. URL: https://www.energy-charts.info/charts/installed_power/chart.htm?c=DE&expansion=installed_power&year=-1&l=en (visited on Nov. 4, 2024) (cit. on p. 3).

[4] Can Berk Saner, Anupam Trivedi, and Dipti Srinivasan. "A Cooperative Hierarchical Multi-Agent System for EV Charging Scheduling in Presence of Multiple Charging Stations". In: *IEEE Transactions on Smart Grid* 13.3 (May 2022), pp. 2218–2233. DOI: 10.1109/TSG.2022.3140927 (cit. on p. 3).

[5] Nadia Panossian, Matteo Muratori, Bryan Palmintier, et al. "Challenges and Opportunities of Integrating Electric Vehicles in Electricity Distribution Systems". In: *Current Sustainable/Renewable Energy Reports* 9.2 (June 1, 2022), pp. 27–40. DOI: 10.1007/s40518-022-00201-2 (cit. on p. 3).

[6] Kang Miao Tan, Vigna K. Ramachandaramurthy, and Jia Ying Yong. "Integration of Electric Vehicles in Smart Grid: A Review on Vehicle to Grid Technologies and Optimization Techniques". In: *Renewable and Sustainable Energy Reviews* 53 (Jan. 1, 2016), pp. 720–732. DOI: 10.1016/j.rser.2015.09.012 (cit. on p. 3).

[7] Agora Verkehrswende, Agora Energiewende, Regulatory Assistance Project (RAP). *Verteilnetzausbau für die Energiewende – Elektromobilität im Fokus*. 2019, p. 114 (cit. on p. 3).

[8] Annegret-Cl. Agricola, Bernd Höflich, Philipp Richard, et al. *dena-Verteilnetzstudie: Ausbau- und Innovationsbedarf der Stromverteilnetze in Deutschland bis 2030*. Ed. by Deutsche Energie-Agentur GmbH. 2012 (cit. on pp. 3, 48, 49).

[9] Georg Kerber. "Aufnahmefähigkeit von Niederspannungsverteilnetzen für die Einspeisung aus Photovoltaikkleinanlagen". PhD thesis. Technische Universität München, 2011 (cit. on pp. 3, 4).

[10] Mirna Grzanic, Marco Giacomo Flammini, and Giuseppe Prettico. "Distribution Network Model Platform: A First Case Study". In: *Energies* 12.21 (21 Jan. 2019), p. 4079. DOI: 10.3390/en12214079 (cit. on pp. 4, 43, 44, 53).

[11] Abhilash Bandam, Chloi Syranidou, Jochen Linssen, and Detlef Stolten. "Geo-Referenced Synthetic Low-Voltage Distribution Networks: A Data-Driven Approach". In: *2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*. Oct. 2021. DOI: 10.1109/ISGTEurope52324.2021.9639916 (cit. on pp. 4, 43, 44, 53).

[12] Ruchi Gupta, Alejandro Pena-Bello, Kai Nino Streicher, et al. “Spatial Analysis of Distribution Grid Capacity and Costs to Enable Massive Deployment of PV, Electric Mobility and Electric Heating”. In: *Applied Energy* 287 (Apr. 1, 2021), p. 116504. DOI: 10.1016/j.apenergy.2021.116504 (cit. on p. 4).

[13] C. Mateo, F. Postigo, F. de Cuadra, et al. “Building Large-Scale U.S. Synthetic Electric Distribution System Models”. In: *IEEE Transactions on Smart Grid* 11.6 (Nov. 2020), pp. 5301–5313. DOI: 10.1109/TSG.2020.3001495 (cit. on pp. 4, 43, 44, 53).

[14] Kathleen M. Gegner, Adam B. Birchfield, Ti Xu, Komal S. Shetye, and Thomas J. Overbye. “A Methodology for the Creation of Geographically Realistic Synthetic Power Flow Models”. In: *2016 IEEE Power and Energy Conference at Illinois (PECI)*. Feb. 2016. DOI: 10.1109/PECI.2016.7459256 (cit. on p. 4).

[15] Steffen Meinecke, Džanan Sarajlić, Simon Ruben Drauz, et al. “SimBench—A Benchmark Dataset of Electric Power Systems to Compare Innovative Solutions Based on Power Flow Analysis”. In: *Energies* 13.12 (12 Jan. 2020), p. 3290. DOI: 10.3390/en13123290 (cit. on p. 4).

[16] Joint Research Centre: Institute for Energy and Transport, Alexandre Lucas, Giuseppe Prettico, et al. *Distribution System Operators Observatory: From European Electricity Distribution Systems to Reference Network*. LU: Publications Office of the European Union, 2016 (cit. on pp. 4, 43).

[17] Veronika Grimm, Bastian Rückel, Christian Sölch, and Gregor Zöttl. “Zur Reduktion des Netzausbaubedarfs durch Redispatch und effizientes Einspeisemanagement: Eine modellbasierte Abschätzung”. In: *List Forum für Wirtschafts- und Finanzpolitik* 41.4 (Aug. 1, 2016), pp. 465–498. DOI: 10.1007/s41025-016-0027-5 (cit. on p. 4).

[18] Agora Energiewende, Agora Verkehrswende, and Stiftung Klimaneutralität. *Das Klimaschutz-Sofortprogramm. 22 Eckpunkte für die ersten 100 Tage der neuen Bundesregierung.* 2021 (cit. on p. 4).

[19] Rebecca Bauer, Xinliang Dai, and Veit Hagenmeyer. “A Shapley Value-Based Distributed AC OPF Approach for Redispatch Congestion Cost Allocation”. In: *Proceedings of the 14th ACM International Conference on Future Energy Systems*. E-Energy ’23. New York, NY, USA: Association for Computing Machinery, June 16, 2023, pp. 109–113. DOI: 10.1145/3575813.3576881 (cit. on p. 4).

[20] Marianne Zeyringer, James Price, Birgit Fais, Pei-Hao Li, and Ed Sharp. “Designing Low-Carbon Power Systems for Great Britain in 2050 That Are Robust to the Spatiotemporal and Inter-Annual Variability of Weather”. In: *Nature Energy* 3.5 (May 2018), pp. 395–403. DOI: 10.1038/s41560-018-0128-x (cit. on p. 5).

[21] TransnetBW. *Projektporträt*. Strom | Netz | Sicherheit. URL: https://www.transnetbw.de/de/netzentwicklung/projekte/netzbooster-kupferzell/projektportraet (visited on Nov. 9, 2024) (cit. on p. 5).

[22] Gert Mehlmann, Uwe Kühnapfel, Felix Wege, et al. “The Kopernikus ENSURE Co-Demonstration Platform”. In: *IEEE Open Journal of Power Electronics* 4 (2023), pp. 987–1002. DOI: 10.1109/OJPEL.2023.3332515 (cit. on p. 6).

[23] Marija Stevic, Abouzar Estebsari, Steffen Vogel, et al. “Multi-Site European Framework for Real-Time Co-Simulation of Power Systems”. In: *IET Generation, Transmission & Distribution* 11.17 (Nov. 30, 2017), pp. 4126–4135. DOI: 10.1049/iet-gtd.2016.1576 (cit. on p. 6).

[24] A. Monti, M. Stevic, S. Vogel, et al. “A Global Real-Time Superlab: Enabling High Penetration of Power Electronics in the Electric Grid”. In: *IEEE Power Electronics Magazine* 5.3 (Sept. 2018), pp. 35–44. DOI: 10.1109/MPEL.2018.2850698 (cit. on p. 6).

[25] Moritz Weber, Michael Kyesswa, Uwe Kühnapfel, Veit Hagenmeyer, and Hüseyin K. Çakmak. “Case Study of Designing a Locally Distributed Real-Time Simulation Infrastructure”. In: *2021 IEEE Electrical Power and Energy Conference (EPEC)*. Oct. 2021, pp. 310–315. DOI: 10.1109/EPEC52095.2021.9621665 (cit. on p. 6).

[26] Oriol Raventos Morera, Abhilash Bandam, Theresa Groß, and Ontje Lünsdorf. “Grid Matching Tool”. In: 1. NFDI4Energy Conference 2024. Hanover, Germany, Feb. 13, 2024. DOI: 10.5281/zenodo.10652974 (cit. on p. 8).

[27] Wilko Heitkötter, Wided Medjroubi, Thomas Vogt, and Carsten Agert. “Comparison of Open Source Power Grid Models - Combining a Mathematical, Visual and Electrical Analysis in an Open Source Tool”. In: *Energies* 12.24 (24 Dec. 11, 2019) (cit. on p. 8).

[28] Martha Maria Frysztacki, Gereon Recht, and Tom Brown. “A Comparison of Clustering Methods for the Spatial Reduction of Renewable Electricity Optimisation Models of Europe”. In: *Energy Informatics* 5.1 (May 4, 2022), p. 4. DOI: 10.1186/s42162-022-00187-7 (cit. on pp. 8, 72, 73, 75, 77, 78, 81, 82).

[29] Michael Kyesswa. “A Parallel Hybrid Method for Fast Analysis of Power System Dynamics”. PhD thesis. Karlsruher Institut für Technologie (KIT), 2021. 165 pp. DOI: 10.5445/IR/1000128495 (cit. on p. 10).

[30] Kevin Mets, Juan Aparicio Ojea, and Chris Develder. “Combining Power and Communication Network Simulation for Cost-Effective Smart Grid Analysis”. In: *IEEE Communications Surveys & Tutorials* 16.3 (2014), pp. 1771–1796. DOI: 10.1109/SURV.2014.021414.00116 (cit. on p. 11).

[31] Valentin Crastan. *Elektrische Energieversorgung 1: Netzelemente, Modellierung, stationäres Verhalten, Bemessung, Schalt- und Schutztechnik*. 4., bearbeitete Auflage. Heidelberg: Springer Vieweg, 2015. 668 pp. (cit. on p. 11).

[32] *PowerFactory - DIgSILENT*. URL: https://www.digsilent.de/de/powerfactory.html (visited on Apr. 3, 2024) (cit. on p. 12).

[33] *Overview | PSCAD*. URL: https://www.pscad.com/software/pscad/overview (visited on Oct. 29, 2024) (cit. on p. 13).

[34] *Graphical User Interface*. URL: https://www.rtds.com/technology/Graphical%20User%20Interface (visited on Apr. 3, 2024) (cit. on p. 13).

[35] *Power System Real-Time Simulation | Power System Solutions*. OPAL-RT. URL: https://www.opal-rt.com/power-systems-overview/ (visited on Oct. 29, 2024) (cit. on p. 13).

[36] *Simscape Electrical*. URL: https://www.mathworks.com/products/simscape-electrical.html (visited on Apr. 9, 2024) (cit. on p. 13).

[37] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education". In: *IEEE Transactions on Power Systems* 26.1 (Feb. 2011), pp. 12–19. DOI: 10.1109/TPWRS.2010.2051168 (cit. on p. 13).

[38] Davis Montenegro, Roger C. Dugan, and Matthew J. Reno. "Open Source Tools for High Performance Quasi-Static-Time-Series Simulation Using Parallel Processing". In: *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*. June 2017, pp. 3055–3060. DOI: 10.1109/PVSC.2017.8521538 (cit. on p. 14).

[39] *GeoJSON*. URL: https://geojson.org/ (visited on Mar. 25, 2024) (cit. on p. 15).

[40] Thamer Alquthami, Ahmed AlAmoudi, Abdullah M. Alsubaie, et al. "Analytics Framework for Optimal Smart Meters Data Processing". In: *Electrical Engineering* 102.3 (Sept. 1, 2020), pp. 1241–1251. DOI: 10.1007/s00202-020-00949-0 (cit. on p. 19).

[41] Veit Hagenmeyer, Hüseyin Kemal Çakmak, Clemens Düpmeier, et al. "Information and Communication Technology in Energy Lab 2.0: Smart Energies System Simulation and Control Center with an Open-Street-Map-based Power Flow Simulation Example". In: *Energy Technology* 4.1 (2016), pp. 145–162. DOI: 10.1002/ente.201500304 (cit. on p. 19).

[42] Benedikt Heidrich, Marian Turowski, Nicole Ludwig, Ralf Mikut, and Veit Hagenmeyer. "Forecasting Energy Time Series with Profile Neural Networks". In: *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*. E-Energy '20. New York, NY, USA: Association for Computing Machinery, June 12, 2020, pp. 220–230. DOI: 10.1145/3396851.3397683 (cit. on p. 19).

[43] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges". In: *IEEE Transactions on Smart Grid* 10.3 (May 2019), pp. 3125–3148. DOI: 10.1109/TSG.2018.2818167 (cit. on p. 19).

[44] Chris King and James Strapp. "Chapter 11 - Software Infrastructure and the Smart Grid". In: *Smart Grid: Integrating Renewable, Distributed and Efficient Energy*. Ed. by Fereidoon P. Sioshansi. Boston: Academic Press, 2012, pp. 259–288. DOI: 10.1016/B978-0-12-386452-9.00011-5 (cit. on pp. 19, 20).

[45] Wen Chen, Kaile Zhou, Shanlin Yang, and Cheng Wu. "Data Quality of Electricity Consumption Data in a Smart Grid Environment". In: *Renewable and Sustainable Energy Reviews* 75 (Aug. 1, 2017), pp. 98–105. DOI: 10.1016/j.rser.2016.10.054 (cit. on p. 19).

[46] L. Wang, M. Turowski, M. Zhang, et al. “Point and Contextual Anomaly Detection in Building Load Profiles of a University Campus”. In: *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. Oct. 2020, pp. 11–15. DOI: 10.1109/ISGT-Europe47291.2020.9248792 (cit. on p. 19).

[47] Hermine N. Akouemo and Richard J. Povinelli. “Data Improving in Time Series Using ARX and ANN Models”. In: *IEEE Transactions on Power Systems* 32.5 (Sept. 2017), pp. 3352–3359. DOI: 10.1109/TPWRS.2017.2656939 (cit. on pp. 19, 21, 30).

[48] Jouni Peppanen, Matthew J. Reno, Mohini Thakkar, Santiago Grijalva, and Ronald G. Harley. “Leveraging AMI Data for Distribution System Model Calibration and Situational Awareness”. In: *IEEE Transactions on Smart Grid* 6.4 (July 2015), pp. 2050–2059. DOI: 10.1109/TSG.2014.2385636 (cit. on p. 20).

[49] Sean J. Taylor and Benjamin Letham. “Forecasting at Scale”. In: *The American Statistician* 72.1 (Jan. 2, 2018), pp. 37–45. DOI: 10.1080/00031305.2017.1380080 (cit. on pp. 20, 25, 30, 31).

[50] Steffen Moritz and Thomas Bartz-Beielstein. “imputeTS: Time Series Missing Value Imputation in R”. In: *The R Journal* 9.1 (2017), p. 207. DOI: 10.32614/RJ-2017-009 (cit. on pp. 20, 21, 30).

[51] Hermine N. Akouemo and Richard J. Povinelli. “Time Series Outlier Detection and Imputation”. In: *2014 IEEE PES General Meeting | Conference & Exposition*. National Harbor, MD, USA: IEEE, July 2014. DOI: 10.1109/PESGM.2014.6939802 (cit. on pp. 21, 30).

[52] Wei Cao, Dong Wang, Jian Li, et al. “BRITS: Bidirectional Recurrent Imputation for Time Series”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 6775–6785 (cit. on pp. 21, 30).

[53] Neeraj Bokde, Marcus W. Beck, Francisco Martínez Álvarez, and Kishore Kulat. “A Novel Imputation Methodology for Time Series Based on Pattern Sequence Forecasting”. In: *Pattern Recognition Letters* 116 (Dec. 1, 2018), pp. 88–96. DOI: 10.1016/j.patrec.2018.09.020 (cit. on pp. 21, 30).

[54] Jouni Peppanen, Xiaochen Zhang, Santiago Grijalva, and Matthew J. Reno. “Handling Bad or Missing Smart Meter Data through Advanced Data Imputation”. In: *2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. Minneapolis, MN, USA: IEEE, Sept. 2016, pp. 1–5. DOI: 10.1109/ISGT.2016.7781213 (cit. on pp. 21, 30, 31).

[55] Jorge Á González Ordiano, Simon Waczowicz, Veit Hagenmeyer, and Ralf Mikut. “Energy Forecasting Tools and Services”. In: *WIREs Data Mining and Knowledge Discovery* 8.2 (2018), e1235. DOI: 10.1002/widm.1235 (cit. on p. 21).

[56] Martina Friese, Jorg Stork, Ricardo Ramos Guerra, et al. *UniFIeD Univariate Frequency-Based Imputation for Time Series Data*. 2013, p. 17 (cit. on pp. 21, 30).

[57] D. Matheson, Chaoying Jing, and F. Monforte. “Meter Data Management for the Electricity Market”. In: *2004 International Conference on Probabilistic Methods Applied to Power Systems*. Sept. 2004, pp. 118–122 (cit. on pp. 21, 30).

[58] Gonzalo Mateos and Georgios B. Giannakis. “Load Curve Data Cleansing and Imputation via Sparsity and Low Rank”. In: *IEEE Transactions on Smart Grid* 4.4 (Dec. 2013), pp. 2347–2355. DOI: 10.1109/TSG.2013.2259853 (cit. on pp. 21, 30).

[59] Y. Ang, Y. Qian, and S. Gao. “Factory Energy Data Imputation by Nearest Neighbor Search with Clustering”. In: *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*. Aug. 2020, pp. 302–307. DOI: 10.1109/AEECA49918.2020.9213497 (cit. on p. 21).

[60] Cruz E. Borges, Oihane Kamara-Esteban, Tony Castillo-Calzadilla, Cristina Martin Andonegui, and Ainhoa Alonso-Vicario. “Enhancing the Missing Data Imputation of Primary Substation Load Demand Records”. In: *Sustainable Energy, Grids and Networks* 23 (Sept. 1, 2020), p. 100369. DOI: 10.1016/j.segan.2020.100369 (cit. on pp. 21, 30).

[61] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2019. URL: http://archive.ics.uci.edu/ml (cit. on p. 28).

[62] Y. Gao, B. Foggo, and N. Yu. “A Physically Inspired Data-Driven Model for Electricity Theft Detection with Smart Meter Data”. In: *IEEE Transactions on Industrial Informatics* 15.9 (Sept. 2019), pp. 5076–5088. DOI: 10.1109/TII.2019.2898171 (cit. on p. 41).

[63] J. Shi, Y. Liu, and N. Yu. “Spatio-Temporal Modeling of Electric Loads”. In: *2017 North American Power Symposium (NAPS)*. Sept. 2017, pp. 1–6. DOI: 10.1109/NAPS.2017.8107311 (cit. on p. 41).

[64] OECD. *Status of Power System Transformation 2018: Advanced Power Plant Flexibility*. Paris: Organisation for Economic Co-operation and Development, 2018 (cit. on p. 42).

[65] Mackay Miller, Eric Martinot, Sadie Cox, et al. *Status Report on Power System Transformation: A 21st Century Power Partnership Report*. NREL/TP-6A20-63366. National Renewable Energy Lab. (NREL), Golden, CO (United States), May 27, 2015. DOI: 10.2172/1215069 (cit. on p. 42).

[66] Hüseyin K. Çakmak, Luc Janecke, Moritz Weber, and Veit Hagenmeyer. “An Optimization-Based Approach for Automated Generation of Residential Low-Voltage Grid Models Using Open Data and Open Source Software”. In: *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. Oct. 2022. DOI: 10.1109/ISGT-Europe54678.2022.9960483 (cit. on pp. 42, 43, 51, 54, 63, 64).

[67] OpenStreetMap Contrib. *OpenStreetMap*. OpenStreetMap. 2004. URL: https://www.openstreetmap.org/ (visited on Apr. 17, 2024) (cit. on p. 43).

[68] OSMBuildings Developers. *OSMBuildings/OSMBuildings*. Apr. 16, 2024. URL: https://github.com/OSMBuildings/OSMBuildings (visited on Apr. 17, 2024) (cit. on pp. 43, 47).

[69] Carlos Mateo Domingo, Tomas Gomez San Roman, Alvaro Sanchez-Miralles, Jesus Pascual Peco Gonzalez, and Antonio Candela Martinez. “A Reference Network Model for Large-Scale Distribution Planning With Automatic Street Map Generation”. In: *IEEE Transactions on Power Systems* 26.1 (Feb. 2011), pp. 190–197. DOI: 10.1109/TPWRS.2010.2052077 (cit. on pp. 43, 44).

[70] V. Krishnan, B. Bugbee, T. Elgindy, et al. “Validation of Synthetic U.S. Electric Power Distribution System Data Sets”. In: *IEEE Transactions on Smart Grid* 11.5 (Sept. 2020), pp. 4477–4489. DOI: 10.1109/TSG.2020.2981077 (cit. on pp. 43, 44).

[71] Bryan Palmintier, Tarek Elgindy, Carlos Mateo, et al. “Experiences Developing Large-Scale Synthetic U.S.-Style Distribution Test Systems”. In: *Electric Power Systems Research* 190 (Jan. 1, 2021), p. 106665. DOI: 10.1016/j.epsr.2020.106665 (cit. on pp. 43, 44).

[72] Wided Medjroubi, Ulf Philipp Müller, Malte Scharf, Carsten Matke, and David Kleinhans. “Open Data in Power Grid Modelling: New Approaches Towards Transparent Grid Models”. In: *Energy Reports* 3 (Nov. 1, 2017), pp. 14–21. DOI: 10.1016/j.egyr.2016.12.001 (cit. on pp. 43, 44, 53).

[73] J. Amme, G. Pleßmann, J. Bühler, et al. “The eGo Grid Model: An Open-Source and Open-Data Based Synthetic Medium-Voltage Grid Model for Distribution Power Supply Systems”. In: *Journal of Physics: Conference Series* 977.1 (Feb. 2018), p. 012007. DOI: 10.1088/1742-6596/977/1/012007 (cit. on pp. 43, 44, 53).

[74] Felix Klabunde, Christian Reinhold, and Bernd Engel. *Regionsabhängige Energiesystemanalysen auf Basis einer datengesteuerten Verteilnetzmodellierung*. Feb. 17, 2022 (cit. on pp. 43, 44, 53).

[75] Hüseyin K. Çakmak and Veit Hagenmeyer. “Using Open Data for Modeling and Simulation of the All Electrical Society in eASiMOV”. In: *2022 Open Source Modelling and Simulation of Energy Systems (OSMSES)*. Apr. 2022. DOI: 10.1109/OSMSES54027.2022.9769145 (cit. on p. 46).

[76] Hermann Meier, Christian Fünfgeld, Thomas Adam, and Bernd Schieferdecker. *Repräsentative VDEW-Lastprofile*. VDEW, 1999 (cit. on p. 46).

[77] Statistische Ämter des Bundes und der Länder. *Zensus 2011*. 2011 (cit. on pp. 46, 49).

[78] Bund der Energieverbraucher. *Die Stromformel für Verbraucher*. 2004. URL: https://www.energieverbraucher.de/de/bewertung-des-stromverbrauchs__646/ContentDetail__3449/ (visited on May 30, 2023) (cit. on pp. 46, 49).

[79] Landesrecht BW. *Gesetz zur Einsparung von Energie und zur Nutzung erneuerbarer Energien zur Wärme- und Kälteerzeugung in Gebäuden, Gebäudeenergiegesetz Anlage 5 (zu § 31 Absatz 1): Vereinfachtes Nachweisverfahren für ein zu errichtendes Wohngebäude*. 2020 (cit. on pp. 47, 49).

[80] S. Lloyd. “Least Squares Quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489 (cit. on p. 49).

[81] Stefan Nickel, Steffen Rebennack, Oliver Stein, and Karl-Heinz Waldmann. *Operations Research*. Berlin, Heidelberg: Springer, 2022. DOI: 10.1007/978-3-662-65346-3 (cit. on p. 49).

[82] Nicos Christofides. "Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem". In: *Operations Research Forum* 3.1 (Mar. 3, 2022), p. 20. DOI: 10.1007/s43069-021-00101-z (cit. on p. 50).

[83] Heike Kluttig, Andreas Dirscherl, and Hans Erhorn. "Energieverbräuche von Bildungsgebäuden in Deutschland". In: *GI-Gesundheits Ingenieur* 122.5 (2001), pp. 221–233 (cit. on p. 54).

[84] Benedikt Osiw, Sabine Jellinghaus, and Hanno Sparbier-Conradus. *Energie-Bericht Ev.-luth. Kirchengemeinde Vahrenwald für das Jahr 2019*. 2019. URL: http://vahrenwalder-kirche.de/images/stories/kirche/02_Aktuelles/Energiebericht2019_ET.pdf (visited on Jan. 20, 2023) (cit. on p. 54).

[85] Yong Ye. "Model Reduction in Physical Domain". PhD thesis. Massachusetts Institute of Technology, 2002 (cit. on p. 72).

[86] Xiaodong Cheng, Jacquelien M. A. Scherpen, and Harry L. Trentelman. "11 Reduced-order Modeling of Large-Scale Network Systems". In: *Volume 3: Applications*. Ed. by Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, et al. Berlin, Boston: De Gruyter, 2021, pp. 345–378. DOI: doi:10.1515/9783110499001-011 (cit. on p. 72).

[87] Savo D. Đukić and Andrija T. Sarić. "Dynamic Model Reduction: An Overview of Available Techniques with Application to Power Systems". In: *Serbian Journal of Electrical Engineering* 9.2 (2012), pp. 131–169 (cit. on pp. 72, 73, 75, 77, 78, 81).

[88] Jan Machowski, Zbigniew Lubosny, Janusz W Bialek, and James R Bumby. *Power System Dynamics: Stability and Control*. Wiley, 2020 (cit. on pp. 72, 73, 75–78, 82).

[89] Joe H. Chow. "Introduction". In: *Power System Coherency and Model Reduction*. Ed. by Joe H. Chow. New York, NY: Springer New York, 2013, pp. 1–14. DOI: 10.1007/978-1-4614-1803-0_1 (cit. on pp. 73, 75, 77).

[90] Julia Sistermanns, Matthias Hotz, Wolfgang Utschick, Dominic Hewes, and Rolf Witzmann. "Feature- and Structure-Preserving Network Reduction for Large-Scale Transmission Grids". In: *2019 IEEE Milan PowerTech*. June 2019. DOI: 10.1109/PTC.2019.8810704 (cit. on pp. 73, 75, 77, 80, 82).

[91] Ganesh Nachiappa Ramaswamy. "Modal Structures and Model Reduction, with Application to Power System Equivalencing". PhD thesis. Massachusetts Institute of Technology, 1995 (cit. on p. 75).

[92] A. J. Germond and R. Podmore. "Dynamic Aggregation of Generating Unit Models". In: *IEEE Transactions on Power Apparatus and Systems* PAS-97.4 (July 1978), pp. 1060–1069. DOI: 10.1109/TPAS.1978.354585 (cit. on pp. 77, 79).

[93] Ananya Kuri, Xiaoyu Zhou, Gert Mehlmann, Matthias Luther, and Piergiovanni La Seta. "Dynamic Model Reduction Based on Coherency and Genetic Optimization Methodology". In: *ETG Congress 2021*. Mar. 2021 (cit. on pp. 77, 79, 89, 101, 104, 116, 117, 122, 123).

[94] Ananya Kuri, Sebastian Gäbel, Ilya Burlakin, et al. “A Novel Concept for Dynamic Network Reduction Dedicated to Real-Time Application”. In: *Kyoto Symposium 2022 - CIGRE Japan* (Kyoto). CIGRE, Apr. 3, 2022 (cit. on pp. 77, 79, 101, 116).

[95] G.N. Ramaswamy, G.C. Verghese, L. Rouco, C. Vialas, and C.L. DeMarco. “Synchrony, Aggregation, and Multi-Area Eigenanalysis”. In: *IEEE Transactions on Power Systems* 10.4 (Nov. 1995), pp. 1986–1993. DOI: 10.1109/59.476067 (cit. on p. 77).

[96] G.N. Ramaswamy, L. Rouco, O. Fillatre, et al. “Synchronic Modal Equivalencing (SME) for Structure-Preserving Dynamic Equivalents”. In: *IEEE Transactions on Power Systems* 11.1 (Feb. 1996), pp. 19–29. DOI: 10.1109/59.485977 (cit. on p. 77).

[97] J. H. Chow, J. R. Winkelman, M. A. Pai, and P. W. Sauer. “Singular Perturbation Analysis of Large-Scale Power Systems”. In: *International Journal of Electrical Power & Energy Systems* 12.2 (Apr. 1, 1990), pp. 117–126. DOI: 10.1016/0142-0615(90)90007-X (cit. on p. 77).

[98] A. Yousefi and B. Lohmann. “Balancing & Optimization for Order Reduction of Nonlinear Systems”. In: *Proceedings of the 2004 American Control Conference*. Vol. 1. June 2004, 108–112 vol.1. DOI: 10.23919/ACC.2004.1383588 (cit. on p. 77).

[99] J. M. Undrill and A. E. Turner. “Construction of Power System Electromechanical Equivalents by Modal Analysis”. In: *IEEE Transactions on Power Apparatus and Systems* PAS-90.5 (Sept. 1971), pp. 2049–2059. DOI: 10.1109/TPAS.1971.293000 (cit. on p. 77).

[100] I. J. Pérez-Arriaga, G. C. Verghese, and F. C. Schweppe. “Selective Modal Analysis with Applications to Electric Power Systems, PART I: Heuristic Introduction”. In: *IEEE Transactions on Power Apparatus and Systems* PAS-101.9 (Sept. 1982), pp. 3117–3125. DOI: 10.1109/TPAS.1982.317524 (cit. on p. 77).

[101] J. M. Ramírez Arredondo. “Obtaining Dynamic Equivalents through the Minimization of a Line Flows Function”. In: *International Journal of Electrical Power & Energy Systems* 21.5 (June 1, 1999), pp. 365–373. DOI: 10.1016/S0142-0615(99)00005-8 (cit. on p. 77).

[102] D. Chaniotis and M.A. Pai. “Model Reduction in Power Systems Using Krylov Subspace Methods”. In: *IEEE Transactions on Power Systems* 20.2 (May 2005), pp. 888–894. DOI: 10.1109/TPWRS.2005.846109 (cit. on p. 77).

[103] J. B. Ward. “Equivalent Circuits for Power-Flow Studies”. In: *Electrical Engineering* 68.9 (Sept. 1949), pp. 794–794. DOI: 10.1109/EE.1949.6444973 (cit. on pp. 77–79, 90).

[104] Paul Dimo. *Nodal Analysis of Power Systems*. International Scholarly Book Services, Inc.,Forest Grove, OR, Jan. 1, 1975 (cit. on pp. 77–79, 90).

[105] Florian Dörfler and Francesco Bullo. “Topological Equivalence of a Structure-Preserving Power Network Model and a Non-Uniform Kuramoto Model of Coupled Oscillators”. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. Dec. 2011, pp. 7099–7104. DOI: 10.1109/CDC.2011.6160337 (cit. on p. 77).

[106] Florian Dörfler and Francesco Bullo. “Kron Reduction of Graphs With Applications to Electrical Networks”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 60.1 (Jan. 2013), pp. 150–163. DOI: 10.1109/TCSI.2012.2215780 (cit. on pp. 77, 81).

[107] Xu Cheng and T.J. Overbye. “PTDF-based Power System Equivalents”. In: *IEEE Transactions on Power Systems* 20.4 (Nov. 2005), pp. 1868–1876. DOI: 10.1109/TPWRS.2005.857013 (cit. on p. 77).

[108] A.M. Azmy and I. Erlich. “Identification of Dynamic Equivalents for Distribution Power Networks Using Recurrent ANNs”. In: *IEEE PES Power Systems Conference and Exposition, 2004.* Oct. 2004, 348–353 vol.1. DOI: 10.1109/PSCE.2004.1397544 (cit. on pp. 77, 80).

[109] Markus Mirz, Steffen Vogel, Georg Reinke, and Antonello Monti. “DPsim—A Dynamic Phasor Real-Time Simulator for Power Systems”. In: *SoftwareX* 10 (July 1, 2019), p. 100253. DOI: 10.1016/j.softx.2019.100253 (cit. on p. 78).

[110] L.A. Zhukov. “Simplified Transformation of Circuit Diagrams of Complex Electric Power Systems”. In: *Izvestia Akademii Nauk SSSR, Energetika I Transport*. Vol. 2. Akademii Nauk SSSR, 1964 (cit. on p. 78).

[111] A. Monticelli, S. Deckmann, A. Garcia, and B. Stott. “Real-Time External Equivalents for Static Security Analysis”. In: *IEEE Transactions on Power Apparatus and Systems* PAS-98.2 (Mar. 1979), pp. 498–508. DOI: 10.1109/TPAS.1979.319387 (cit. on p. 79).

[112] Di Shi, Daniel L. Shawhan, Nan Li, et al. “Optimal Generation Investment Planning: Pt. 1: Network Equivalents”. In: *2012 North American Power Symposium (NAPS)*. Sept. 2012, pp. 1–6. DOI: 10.1109/NAPS.2012.6336375 (cit. on p. 79).

[113] William F. Tinney and Joseph M. Bright. “Adaptive Reductions for Power Flow Equivalents”. In: *IEEE Transactions on Power Systems* 2.2 (May 1987), pp. 351–359. DOI: 10.1109/TPWRS.1987.4335132 (cit. on p. 79).

[114] Adarsh Nagarajan, Austin Nelson, Kumaraguru Prabakar, et al. “Network Reduction Algorithm for Developing Distribution Feeders for Real-Time Simulators”. In: *2017 IEEE Power Energy Society General Meeting*. July 2017, pp. 1–5. DOI: 10.1109/PESGM.2017.8273938 (cit. on p. 80).

[115] Eleftherios O. Kontis, Theofilos A. Papadopoulos, Mazheruddin H. Syed, et al. “Artificial-Intelligence Method for the Derivation of Generic Aggregated Dynamic Equivalent Models”. In: *IEEE Transactions on Power Systems* 34.4 (July 2019), pp. 2947–2956. DOI: 10.1109/TPWRS.2019.2894185 (cit. on p. 80).

[116] Zheng Liu, Jan-Hendrik Menke, Nils Bornhorst, and Martin Braun. “Static Grid Equivalent Models Based on Artificial Neural Networks”. In: *IEEE Access* 9 (2021), pp. 168535–168546. DOI: 10.1109/ACCESS.2021.3134373 (cit. on pp. 80, 82).

[117] Humud Said and Peter Kuffel. “Dynamic Equivalents for RTDS Applications”. In: 2020 CIGRE Canada Virtual Conference. 2020, p. 6 (cit. on p. 80).

[118] Yaosuo Xue, Yilu Liu, Zhihao Jiang, Ning Tong, and Alfonso Tarditi. *Enhanced Dynamic Equivalent Identification Method of Large-Scale Power Systems Using Multiple Event*. Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States), July 1, 2020 (cit. on p. 80).

[119] HyungSeon Oh. "Aggregation of Buses for a Network Reduction". In: *IEEE Transactions on Power Systems* 27.2 (May 2012), pp. 705–712. DOI: 10.1109/TPWRS.2011.2176758 (cit. on p. 80).

[120] Zou Xiaofeng, Yang Shihao, Zhou Desheng, and Song Jie. "Research on Application of FDNE Based on RTDS and Comparison Analysis of Fault Transient of Shanghai Sijing Power Grid". In: *The Journal of Engineering* 2019.16 (2019), pp. 1393–1397. DOI: 10.1049/joe.2018.8777 (cit. on p. 80).

[121] Yuefeng Liang, Xi Lin, Ani M. Gole, and Ming Yu. "Improved Coherency-Based Wide-Band Equivalents for Real-Time Digital Simulators". In: *IEEE Transactions on Power Systems* 26.3 (Aug. 2011), pp. 1410–1417. DOI: 10.1109/TPWRS.2010.2085456 (cit. on p. 80).

[122] Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. *Exploring Network Structure, Dynamics, and Function Using NetworkX*. LA-UR-08-05495; LA-UR-08-5495. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008 (cit. on p. 89).

[123] TC 73 Short-circuit currents. *IEC 60909-0:2016*. Jan. 28, 2016 (cit. on p. 92).

[124] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. DOI: 10.1007/978-3-662-44874-8 (cit. on pp. 100, 113).

[125] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT press, 1998 (cit. on p. 100).

[126] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197. DOI: 10.1109/4235.996017 (cit. on pp. 101, 109, 111).

[127] Gurjot Singh and Neeraj Gupta. "A Study of Crossover Operators in Genetic Algorithms". In: *Frontiers in Nature-Inspired Industrial Optimization*. Ed. by Mahdi Khosravy, Neeraj Gupta, and Nilesh Patel. Singapore: Springer, 2022, pp. 17–32. DOI: 10.1007/978-981-16-3128-3_2 (cit. on p. 110).

[128] Ahmed Fawzy Gad. "PyGAD: An Intuitive Genetic Algorithm Python Library". In: *Multimedia Tools and Applications* 83.20 (June 1, 2024), pp. 58029–58042. DOI: 10.1007/s11042-023-17167-y (cit. on p. 115).

[129] Karl Friedrich Schäfer. "Betrieb elektrischer Übertragungssysteme". In: *Systemführung: Betrieb elektrischer Energieübertragungsnetze*. Ed. by Karl Friedrich Schäfer. Wiesbaden: Springer Fachmedien, 2022, pp. 81–255. DOI: 10.1007/978-3-658-36199-0_3 (cit. on p. 122).

[130] Kalyanmoy Deb and Himanshu Jain. “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints”. In: *IEEE Transactions on Evolutionary Computation* 18.4 (Aug. 2014), pp. 577–601. DOI: 10.1109/TEVC.2013.2281535 (cit. on p. 138).

[131] NEPLAN. *Exciter Models - Standard Dynamic Excitation Systems in NEPLAN Power System Analysis Tool*. July 10, 2023. URL: https://www.neplan.ch/wp-content/uploads/2015/08/Nep_EXCITERS1.pdf (visited on July 11, 2023) (cit. on p. 168).

[132] NEPLAN. *Turbine-Governor Models - Standard Dynamic Turbine-Governor Systems in NEPLAN Power System Analysis Tool*. July 10, 2023. URL: https://www.neplan.ch/wp-content/uploads/2015/08/Nep_TURBINES_GOV.pdf (visited on July 10, 2023) (cit. on p. 168).

[133] NEPLAN. *Power System Stabilizer Models - Standard Dynamic Power System Stabilizers in NEPLAN Power System Analysis Tool*. July 10, 2023. URL: https://www.neplan.ch/wp-content/uploads/2015/08/Nep_PSSs.pdf (visited on July 11, 2023) (cit. on p. 168).

List of Figures

List of Tables

List of Listings

Appendix A

A.1 Component Aggregation Methods

This section provides details on the aggregation procedure of the TPNR method described in Section 6.1.1. These details are presented in the form of Tables A.1 to A.6. Each of these tables includes the list of relevant parameters for one component type which is identified by its PowerFactory class name, e.g., `ElmLod` for load components. The parameters are identified by their names in PowerFactory's Python API and a short description. For each parameter the aggregation method is described briefly.

The following examples can be used as instructions to interpret the descriptions of the aggregation methods:

- Table A.1, `plini`: $\sum$
  This is a straightforward aggregation of the `plini` parameter by summing up the corresponding values of all load components that are aggregated.
- Table A.2, `qrean`: $\sum \texttt{qrean}_i \cdot \texttt{ncapa}_i$
  This operation aggregates the `qrean` parameter by summing up the product of `qrean` and `ncapa` for each shunt component. The `i` subscript indicates that the two parameters of the same component are multiplied in the summation operator.
- Table A.4, `cosn`: $\sum \texttt{Pnom}_i / \texttt{sgn}_{agg}$
  This aggregation operation divides the sum of the `Pnom` parameter of the static generators by the aggegated `sgn` parameter, i.e., $\texttt{sgn}_{agg}$.

Tab. A.1.: `ElmLod`: aggregation operators for load components.

Parameter	Description	Aggregation
`plini`	Active power [MW]	$\sum$
`qlini`	Reactive power [Mvar]	$\sum$

Tab. A.2.: `ElmShnt, shtype = 1`: aggregation operators for inductive shunt components.

Parameter	Description	Aggregation
`qrean`	Rated reactive power [Mvar]	$\sum$ `qrean`$_i \cdot$ `ncapa`$_i$
`ncapx`	Maximum number of steps	1
`ncapa`	Active number of steps	1

Tab. A.3.: `ElmShnt, shtype = 2`: aggregation operators for capacitive shunt components.

Parameter	Description	Aggregation
`qcapn`	Rated reactive power [Mvar]	$\sum$ `qcapn`$_i \cdot$ `ncapa`$_i$
`ncapx`	Maximum number of steps	1
`ncapa`	Active number of steps	1

Tab. A.4.: `ElmGenstat`: aggregation operators for static generator components.

Parameter	Description	Aggregation
`pgini`	Active power [MW]	$\sum \min($`pgini`$_i$, `P_max`$_i$, `Pmax_uc`$_i)$
`qgini`	Reactive power [Mvar]	$\sum$
`cQ_min`	Capability curve minimum [Mvar]	$\sum$
`cQ_max`	Capability curve maximum [Mvar]	$\sum$
`P_max`	Active power rating [MW]	$\sum$
`sgn`	Rated apparent power [MVA]	$\sum$
`Pmin_uc`	Active power operational min [MW]	$\sum$
`Pmax_uc`	Active power operational max [MW]	$\sum$
`pmaxratf`	Active power rating factor	$\sum$ `P_max`$_i$ / $\sum$ `Pnom`$_i$
`cosn`	Rated power factor	$\sum$ `Pnom`$_i$ / `sgn`$_{agg}$
`Pnom`	Active power nominal rating [MW]	read only

Tab. A.5.: `TypTr2`: aggregation operators for two-winding transformers.

Parameter	Description	Aggregation
`strn`	Rated power [MVA]	$\sum$
`utrn_h`	Rated voltage: HV side [kV]	`strn`$_{agg}$ / $\sum$ `strn`$_i$/`utrn_h`$_i$
`utrn_l`	Rated voltage: LV side [kV]	`strn`$_{agg}$ / $\sum$ `strn`$_i$/`utrn_l`$_i$
`x1pu`	Positive sequence reactance [p.u.]	($\sum$ `x1pu`$_i \cdot$ `strn`$_i$)/`strn`$_{agg}$
`x0pu`	Zero sequence reactance [p.u.]	($\sum$ `x0pu`$_i \cdot$ `strn`$_i$)/`strn`$_{agg}$
`r1pu`	Positive sequence resistance [p.u.]	($\sum$ `r1pu`$_i \cdot$ `strn`$_i$)/`strn`$_{agg}$
`r0pu`	Zero sequence resistance [p.u.]	($\sum$ `r0pu`$_i \cdot$ `strn`$_i$)/`strn`$_{agg}$
`pfe`	No load losses [kW]	$\sum$
`curmg`	No load current [%]	($\sum$ `curmg`$_i \cdot$ `strn`$_i$)/`strn`$_{agg}$

Tab. A.6.: `ElmSym` and `TypSym`: aggregation operators for synchronous machines. The first part of the table describes the parameters of the `ElmSym` component class, while the part below the horizontal line describes the parameters of the type class, `TypSym`.

Parameter	Description	Aggregation
`pgini`	Active power [MW]	$\sum \min(\texttt{pgini}_i, \texttt{P\_max}_i, \texttt{Pmax\_uc}_i)$
`qgini`	Reactive power [Mvar]	$\sum$
`cQ_min`	Capability curve minimum [Mvar]	$\sum$
`cQ_max`	Capability curve maximum [Mvar]	$\sum$
`P_max`	Active power rating [MW]	$\sum$
`Pmin_uc`	Active power operational min [MW]	$\sum$
`Pmax_uc`	Active power operational max [MW]	$\sum$
`pmaxratf`	Active power rating factor	$\texttt{P\_max}_{agg} / \sum \texttt{Pnom}_i$
`usetp`	Voltage setpoint [p.u.]	mean
`sgn`	Rated apparent power [MVA]	$\sum$
`Q_min`	Minimum reactive power [Mvar]	$\sum$
`Q_max`	Maximum reactive power [Mvar]	$\sum$
`iturbo`	Rotor type (0: salient pole, 1: round rotor)	$\max(\texttt{iturbo}_i)$
`rstr`	Stator resistance [p.u.]	$(\sum \texttt{rstr}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xl`	Stator leakage reactance [p.u.]	$(\sum \texttt{xl}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`tags`	Acceleration time constant [s]	
`xd`	Synchronous reactance, d-axis [p.u.]	$(\sum \texttt{xd}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xq`	Synchronous reactance, q-axis [p.u.]	$(\sum \texttt{xq}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xrl`	Coupling reactance between field and damper winding [p.u.]	$(\sum \texttt{xrl}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xrlq`	Coupling reactance between q-axis damper windings [p.u.]	$(\sum \texttt{xrlq}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xds`	Transient reactance d-axis [p.u.]	$(\sum \texttt{xds}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xqs`	Transient reactance q-axis [p.u.]	$(\sum \texttt{xqs}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xdss`	Subtransient reactance d-axis [p.u.]	$(\sum \texttt{xdss}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`xqss`	Subtransient reactance q-axis [p.u.]	$(\sum \texttt{xqss}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`tds`	Short-circuit transient time constant d-axis [s]	$(\sum \texttt{tds}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`tqs`	Short-circuit transient time constant q-axis [s]	$(\sum \texttt{tqs}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`tdss`	Short-circuit subtransient time constant d-axis [s]	$(\sum \texttt{tdss}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$
`tqss`	Short-circuit subtransient time constant q-axis [s]	$(\sum \texttt{tqss}_i \cdot \texttt{sgn}_i)/\texttt{sgn}_{agg}$

A.2 Synchronous Machine Controller Models

This section provides details on the synchronous machine controller models implemented in the GA-based optimization method, introduced in Table 7.2. These details are provided in the form of Tables A.7 to A.15. These tables present the parameters of the controller models with a variable name, brief description, and the constraints for the optimization procedure. The descriptions and constraints are based on the PowerFactory documentation, as well as NEPLAN exciter [131], governor [132], and PSS documentation [133].

Tab. A.7.: Simplified Excitation System (SEXS) model parameters.

Parameter	Description	Constraints
T_b	Filter delay time [s]	$0.1 \leq T_b \leq 20$
T_a	Filter derivative time constant [s]	$0.1 \leq T_a \leq 10$
K	Controller gain [p.u.]	$0.1 \leq K \leq 400$
T_e	Exciter time constant [s]	$0.01 \leq T_e \leq 5$
E_{min}	Controller minimum output [p.u.]	$-10 \leq E_{min} \leq 0.99$
E_{max}	Controller maximum output [p.u.]	$1 \leq E_{max} \leq 10$

Tab. A.8.: IEEE AC1A exciter model parameters.

Parameter	Description	Constraints
T_b	Filter delay time [s]	$0 \leq T_b \leq 20$
T_c	Filter derivative time constant [s]	$0 \leq T_c \leq 20$
T_r	Measurement delay [s]	$0.001 \leq T_r \leq 1$
K_a	Voltage regulator gain [p.u.]	$0.001 \leq K_a \leq 1000$
T_a	Voltage regulator time constant [s]	$0.001 \leq T_a \leq 10$
K_f	Stabilization path gain [p.u.]	$0.001 \leq K_f \leq 0.3$
T_f	Stabilization path time constant [s]	$0.02 \leq T_f \leq 1.5$
T_e	Exciter time constant [s]	$0.02 \leq T_e \leq 2$
K_c	Rectifier regulation constant [p.u.]	$0 \leq K_c \leq 1$
K_e	Exciter constant [p.u.]	$0.001 \leq K_e \leq 1$
K_d	Armature reaction constant [p.u.]	$0 \leq K_d \leq 1$
E_1	Saturation factor 1 [p.u.]	$0.002 \leq E_1 \leq 20$
SE_1	Saturation factor 2 [p.u.]	$SE_2 < SE_1 \leq 1$
E_2	Saturation factor 3 [p.u.]	$0.001 \leq E_2 < E_1$
SE_2	Saturation factor 4 [p.u.]	$0 < SE_2 < 1$
$V_{a,min}$	Amplifier minimum output [p.u.]	$-15 \leq V_{a,min} \leq -0.001$
$V_{a,max}$	Amplifier maximum output [p.u.]	$1.5 \leq V_{a,max} \leq 15$
$V_{r,min}$	Controller minimum output [p.u.]	$-10 \leq V_{r,min} \leq -0.001$
$V_{r,max}$	Controller maximum output [p.u.]	$0.001 \leq V_{r,max} \leq 10$

Tab. A.9.: IEEE ST1A exciter model parameters.

Parameter	Description	Constraints
T_r	Measurement delay [s]	$0.001 \leq T_r \leq 0.1$
T_b	Filter 1st delay time constant [s]	$0.04 \leq T_b \leq 20$
T_c	Filter 1st derivative time constant [s]	$0.001 \leq T_c \leq 10$
T_{b1}	Filter 2nd delay time constant [s]	$0.04 \leq T_{b1} \leq 20$
T_{c1}	Filter 2nd derivative time constant [s]	$0 \leq T_{c1} \leq 10$
K_a	Controller gain [p.u.]	$50 \leq K_a \leq 1000$
T_a	Controller time constant [s]	$0.04 \leq T_a \leq 0.5$
K_c	Current limiter factor [p.u.]	$0 \leq K_c \leq 0.3$
K_f	Stabilization path gain [p.u.]	$0.001 \leq K_f \leq 0.3$
T_f	Stabilization path time constant [s]	$0.3 \leq T_f \leq 1.5$
K_{lr}	Current input factor [p.u.]	$0.001 \leq K_{lr} \leq 5$
I_{lr}	Current input reference [p.u.]	$0.001 \leq I_{lr} \leq 5$
V_{os}	PSS input selector [p.u.]	$V_{os} \in \{1, 2\}$
V_{el}	U_{el} input selector [s]	$V_{el} \in \{1, 2, 3\}$
$V_{i,min}$	Controller input minimum [p.u.]	$-0.3 \leq V_{i,min} \leq 0$
$V_{a,min}$	Controller minimum output [p.u.]	$-8 \leq V_{a,min} \leq -3$
$V_{r,min}$	Exciter minimum output [p.u.]	$-8 \leq V_{r,min} \leq -3$
$V_{i,max}$	Controller input maximum [p.u.]	$0.001 \leq V_{i,max} \leq 0.2$
$V_{a,max}$	Controller maximum output [p.u.]	$3 \leq V_{a,max} \leq 8$
$V_{r,max}$	Exciter maximum output [p.u.]	$3 \leq V_{r,max} \leq 8$

Tab. A.10.: IEEE G1 governor model parameters.

Parameter	Description	Constraints
K	Controller gain [p.u.]	$5 \leq K \leq 30$
T_1	Governor time constant [s]	$0.001 \leq T_1 \leq 5$
T_2	Governor derivative time constant [s]	$0 \leq T_2 \leq 10$
T_3	Servo time constant [s]	$0.04 \leq T_3 \leq 1$
K_1	High pressure turbine factor [p.u.]	$-2 \leq K_1 \leq 1$
T_5	Intermediate pressure turbine time constant [s]	$0.001 \leq T_5 \leq 10$
K_3	Intermediate pressure turbine factor [p.u.]	$0 \leq K_3 \leq 0.5$
T_6	Medium pressure turbine time constant [s]	$0.001 \leq T_6 \leq 10$
K_5	Medium pressure turbine factor [p.u.]	$0 \leq K_5 \leq 0.35$
T_4	High pressure turbine time constant [s]	$0.001 \leq T_4 \leq 1$
T_7	Low pressure turbine time constant [s]	$0.001 \leq T_7 \leq 10$
K_7	Low pressure turbine factor [p.u.]	$0 \leq K_7 \leq 0.3$
U_c	Valve closing time [p.u. s^{-1}]	$-0.3 \leq U_c \leq -0.001$
P_{min}	Minimum gate limit [p.u.]	$0 \leq P_{min} \leq 0.49$
U_o	Valve opening time [p.u. s^{-1}]	$0.01 \leq U_o \leq 0.3$
P_{max}	Maximum gate limit [p.u.]	$0.5 \leq P_{max} \leq 2$
K_2	High pressure turbine factor [p.u.]	$K_2 = 0$
K_4	Intermediate pressure turbine factor [p.u.]	$K_4 = 0$
K_6	Medium pressure turbine factor [p.u.]	$K_6 = 0$
K_8	Low pressure turbine factor [p.u.]	$K_8 = 0$
	Additional constraints	$K_1 + K_3 + K_5 + K_7 = 1$

Tab. A.11.: TGOV1 governor model parameters.

Parameter	Description	Constraints
T_3	Turbine delay time constant [s]	$2 \cdot T_2 \leq T_3 \leq 10$
T_2	Turbine derivative time constant [s]	$0.001 \leq T_2 \leq 4.99$
A_t	Turbine power coefficient [p.u.]	$A_t = 1$
D_t	Frictional losses factor [p.u.]	$0 \leq D_t \leq 0.5$
R	Controller droop [p.u.]	$0.001 \leq R \leq 0.1$
T_1	Governor time constant [s]	$0.001 \leq T_1 \leq 5$
P_N	Turbine rated power [MW]	$P_N = 0 \rightarrow$ Rating of generator
V_{min}	Minimum gate limit [p.u.]	$-5 \leq V_{min} \leq 1$
V_{max}	Maximum gate limit [p.u.]	$V_{min} < V_{max} \leq 5$

Tab. A.12.: GAST governor model parameters.

Parameter	Description	Constraints
R	Speed droop [p.u.]	$0.001 \leq R \leq 0.1$
T_1	Controller time constant [s]	$0.04 \leq T_1 \leq 0.5$
T_2	Actuator time constant [s]	$0.04 \leq T_2 \leq 0.5$
T_3	Compressor time constant [s]	$0.04 \leq T_3 \leq 5$
A_T	Ambient temperature load limit [p.u.]	$0.001 \leq A_T \leq 1$
K_t	Turbine factor [p.u.]	$0.001 \leq K_t \leq 5$
D_{turb}	Frictional losses factor [p.u.]	$0 \leq D_{turb} \leq 0.5$
P_N	Turbine rated power [MW]	$P_N = 0 \rightarrow$ Rating of generator
V_{min}	Controller minimum output [p.u.]	$0 \leq V_{min} \leq \min(V_{max}, 1)$
V_{max}	Controller maximum output [p.u.]	$0.5 \leq V_{max} \leq 1.5$

Tab. A.13.: HYGOV governor model parameters.

Parameter	Description	Constraints
r	Temporary droop [p.u.]	$0.001 \leq r \leq 2$
T_r	Governor time constant [s]	$0.04 \leq T_r \leq 30$
T_f	Filter time constant [s]	$0.04 \leq T_f \leq 0.1$
T_g	Servo time constant [s]	$0.04 \leq T_g \leq 1$
T_w	Water starting time [s]	$0.04 \leq T_w \leq 5$
A_t	Turbine gain [p.u.]	$0.01 \leq A_t \leq 1$
D_{turb}	Frictional losses factor [p.u.]	$0 \leq D_{turb} \leq 0.5$
q_{nl}	No load flow [p.u.]	$0 \leq q_{nl} \leq 1$
R	Permanent droop [p.u.]	$0.001 \leq R \leq \min(r, 0.1)$
P_N	Turbine rated power [MW]	$P_N = 0 \rightarrow$ Rating of generator
G_{min}	Minimum gate limit [p.u.]	$0 \leq G_{min} < \min(G_{max}, 1)$
V_{elm}	Gate velocity limit [p.u.]	$0.001 \leq V_{elm} \leq 0.3$
G_{max}	Maximum gate limit [p.u.]	$0.001 \leq G_{max} \leq 1$

Tab. A.14.: PSS1A power system stabilizer model parameters.

Parameter	Description	Constraints
K_s	PSS gain [p.u.]	$0 \leq K_s \leq 10$
T_r	Transducer time constant [s]	$0 \leq T_r \leq 1$
T_w	Washout time constant	$0 \leq T_w \leq 10$
T_1	Lead/lag time constant	$0 \leq T_1 \leq 10$
T_2	Lead/lag time constant	$0 \leq T_2 \leq 1$
T_3	Lead/lag time constant	$0 \leq T_3 \leq 10$
T_4	Lead/lag time constant	$0 \leq T_4 \leq 1$
A_1	PSS signal conditioning frequency filter constant	$0 \leq A_1 \leq 5$
A_2	PSS signal conditioning frequency filter constant	$0 \leq A_2 \leq 5$
I_{PB}	PSS base input selector	$I_{PB} = 1 \rightarrow$ Generator MVA
$V_{st,min}$	Controller minimum output	$-1 \leq V_{st,min} \leq -0.001$
$V_{st,max}$	Controller maximum output	$0.001 \leq V_{st,max} \leq 1$

Tab. A.15.: PSS2A power system stabilizer model parameters.

Parameter	Description	Constraints
T_{w1}	1st washout 1st time constant [s]	$1.5 \leq T_{w1} \leq 2.5$
T_{w2}	1st washout 2nd time constant [s]	$1.5 \leq T_{w2} \leq 2.5$
T_6	1st signal transducer time constant [s]	$0 \leq T_6 \leq 0.001$
T_{w3}	2nd washout 1st time constant [s]	$1.5 \leq T_{w3} \leq 2.5$
T_{w4}	2nd washout 2nd time constant [s]	$0 \leq T_{w4} \leq 0.001$
K_{s2}	2nd signal transducer factor [p.u.]	$0.001 \leq K_{s2} \leq 0.5$
T_7	2nd signal transducer time constant [s]	$1.5 \leq T_7 \leq 2.5$
K_{s3}	Washouts coupling factor [p.u.]	$0.9 \leq K_{s3} \leq 1.1$
K_{s1}	PSS gain [p.u.]	$9 \leq K_{s1} \leq 11$
T_{s1}	1st lead-lag derivative time constant [s]	$0.02 \leq T_{s1} \leq 0.3$
T_{s2}	1st lead-lag delay time constant [s]	$0.02 \leq T_{s2} \leq 0.1$
T_{s3}	2nd lead-lag derivative time constant [s]	$0.02 \leq T_{s3} \leq 0.2$
T_{s4}	2nd lead-lag delay time constant [s]	$0.002 \leq T_{s4} \leq 0.01$
T_8	Ramp tracking filter deriv. time constant [s]	$0.4 \leq T_8 \leq 0.6$
T_9	Ramp tracking filter delay time constant [s]	$0.1 \leq T_9 \leq 0.15$
N	Ramp tracking filter	$N \cdot M \leq 8$, $M = 0 \rightarrow N = 0$
M	Ramp tracking filter	$M \in \{0..8\}$
I_{c1}	1st input selector	$I_{c1} = 1$: Rotor speed dev.
I_{c2}	2nd input selector	$I_{c2} = 3$: Gen. el. power
K_d	Derivator factor [p.u.]	$0 \leq K_d \leq 2$
I_{PB}	PSS base selector (1=Gen MVA, 0=Gen MW)	$I_{PB} = 1$: Generator MVA
$V_{st,min}$	Controller minimum output [p.u.]	$-0.3 \leq V_{st,min} \leq 0$
$V_{st,max}$	Controller maximum output [p.u.]	$0.001 \leq V_{st,max} \leq 0.2$

A.3 Details on the Evaluation of the Genetic Algorithm

A.3.1 IEEE 39 Bus System

This section provides detailed information on the controller configuration of the IEEE 39 Bus System used for the evaluation of the GA-based controller optimization described in Section 7.3.1. While the IEEE 39 Bus System is a widely used benchmark model, there is no common configuration for the synchronous machine controllers. Thus, this section provides the controller models of the detailed model in Table A.16 and the controller models used in the evaluated *Expert* solution of the reduced model in Table A.17. The parameters used for these models are listed in Tables A.18 to A.23.

The model used for this evaluation is a modified version of the IEEE 39 Bus System model provided by PowerFactory. In PowerFactory, the synchronous machine controllers are interconnected using a so-called *frame*. Various frames are available in PowerFactory with different compatibilities to controller models. For many standard controller models, PowerFactory includes multiple variants that are compatible with different frame types. The model supplied by PowerFactory uses a frame type with a voltage transducer slot before the exciter. For the optimization process, we use a simpler frame type which supports the *PSSe-compatible* controller models in PowerFactory. For some standard controller models the implementations for the different frames vary slightly, which can lead to varying parameters. These cases are indicated in the tables with the concrete parameters used for the evaluation.

Tab. A.16.: The controller configuration of the detailed IEEE 39 Bus System model. The rows in gray indicate the generators in the external subsystem which are aggregated in the reduction process.

Generator	Governor	Exciter	PSS
G 01	IEEE G1	IEEE ST1A	IEEE PSS1A
G 02	IEEE GAST	IEEE AC1A	—
G 03	IEEE GAST	IEEE ST1A	IEEE PSS2A
G 04	IEEE GAST	IEEE AC1A	—
G 05	IEEE GAST	IEEE AC1A	IEEE PSS1A
G 06	IEEE GAST	IEEE ST1A	IEEE PSS2A
G 07	IEEE G1	IEEE ST1A	IEEE PSS1A
G 08	IEEE G1	IEEE ST1A	IEEE PSS1A
G 09	IEEE G1	IEEE ST1A	IEEE PSS1A
G 10	IEEE G1	IEEE ST1A	IEEE PSS1A

Tab. A.17.: The controller configuration of the *Expert* solution for the reduced IEEE 39 Bus System model. The PSS of generator G 02 [R] uses the same parameters as G 03 in the detailed model.

Generator	Governor	Exciter	PSS
G 01 [R]	IEEE GAST	IEEE AC1A	—
G 02 [R]	IEEE GAST	IEEE ST1A	IEEE PSS2A

Tab. A.18.: The parameter values for the IEEE G1 governor. The generators G 08 and G 09 use the value in parentheses for P_{max}.

Parameter	Value	Parameter	Value
K	20.0	T_7	0.0
T_1	0.0	K_7	0.0
T_2	0.0	U_c	−1.0
T_3	0.075	P_{min}	0.0
K_1	0.2	U_o	0.6786
T_5	10.0	P_{max}	0.9 (1.0)
K_3	0.4	K_2	0.0
T_6	0.6	K_4	0.0
K_5	0.4	K_6	0.0
T_4	0.3	K_8	0.0

Tab. A.19.: The parameter values for the GAST governor.

Parameter	Value	Parameter	Value
R	0.05	K_t	3.0
T_1	0.5	D_{turb}	0.18
T_2	0.5	P_N	0
T_3	3.0	V_{min}	0.0
A_T	1.0	V_{max}	1.0

Tab. A.20.: The parameter values for the IEEE ST1A exciter. The T_r parameter described in Table A.9 is not present in this variant of the model. Technically, V_{os} and V_{el} are 0, but this model is zero indexed.

Parameter	Value	Parameter	Value
T_b	10.0	I_{lr}	0.0
T_c	1.0	V_{os}	1
T_{b1}	0.0	V_{el}	1
T_{c1}	0.0	$V_{i,min}$	−0.1
K_a	200.0	$V_{a,min}$	−5.0
T_a	0.015	$V_{r,min}$	−100.0
K_c	0.0	$V_{i,max}$	0.1
K_f	0.0	$V_{a,max}$	5.0
T_f	1.0	$V_{r,max}$	100.0
K_{lr}	0.0		

Tab. A.21.: The parameter values for the IEEE AC1A exciter. The T_r parameter described in Table A.8 is not present in this variant of the model. G 05 uses values in parentheses.

Parameter	Value	Parameter	Value
T_b	10.0	K_d	0.38
T_c	1.0	E_1	4.18
K_a	200.0	SE_1	0.1
T_a	0.015	E_2	3.14
K_f	0.0	SE_2	0.03
T_f	1.0	$V_{a,min}$	-5.0 (-999.0)
T_e	0.8	$V_{a,max}$	5.0 (999.0)
K_c	0.0 (0.08)	$V_{r,min}$	-100.0
K_e	1.0	$V_{r,max}$	100.0

Tab. A.22.: The parameter values for the IEEE PSS1A. The I_{PB} parameter is replaced by $V_{si,in}$ in this variant, which offers more input options. $V_{si,in} = 1$ resolves to selection the rotor speed as input. As the different instances of the PSS1A use varying parameters, this table is divided into two parts: one for common parameters and one for individual parameters.

Common Parameters

Parameter	Value	Parameter	Value
T_r (T_6)	0.0	$V_{si,in}$	1
T_w (T_5)	10.0	$V_{st,min}$	-0.2
A_1	0.0	$V_{st,max}$	0.2
A_2	0.0		

Individual Parameters

Generator	K_s	T_1	T_2	T_3	T_4
G 01	1.0	5.0	0.6	3.0	0.5
G 05	1.0	1.5	0.2	1.0	0.1
G 07	7.5	0.2	0.02	0.5	0.1
G 08	2.0	1.0	0.2	1.0	0.1
G 09	2.0	1.0	0.5	2.0	0.1
G 10	1.0	1.0	0.05	3.0	0.5

Tab. A.23.: The parameter values for the IEEE PSS2A. As opposed to the model described in Table A.15, the second input for this model is fixed to generator power in p.u., which is calculated as `Pg/sgnn`. This is equivalent to $I_{PB} = 1$ of the other model variant. The first input is the rotor speed, which is not available in PSSe-compatible model variant. K_d is not available in this model variant. The values in parentheses are used by G 06.

Parameter	Value	Parameter	Value
T_{w1}	1.0 (10.0)	T_{s4}	0.2 (0.05)
T_{w2}	1.0 (10.0)	T_8	0.3
T_6	0.0	T_9	0.15
T_{w3}	10.0	N	4
T_{w4}	0.0	M	2
K_{s2}	1.54	I_{c1}	Rotor speed
T_7	10.0	I_{c2}	Generator power
K_{s3}	1.0	K_d	—
K_{s1}	2.0	I_{PB}	1
T_{s1}	3.0 (0.5)	$V_{st,min}$	−0.2
T_{s2}	0.2 (0.1)	$V_{st,max}$	0.2
T_{s3}	2.0 (0.5)		

Tab. A.24.: The error values for the simulation without any events show that the optimization run that specializes on this particular simulation type (first column) significantly outperforms the other variants. While the hybrid method with two buses loses some accuracy for this simulation through fine-tuning for all events, it still yields much better results than the other three variants.

	No events	Hybrid		Multi-event	
		2 Buses	3 Buses	2 Buses	5 Buses
$\text{MAPE}_{B,u}$ [%]	0.0344	0.0710	0.1810	0.1440	0.1131
$\text{MAE}_{B,u}$ [p.u.]	0.0004	0.0007	0.0019	0.0015	0.0012
$\text{MAPE}_{B,f}$ [%]	0.0019	0.0074	0.0095	0.0159	0.0277
$\text{MAE}_{B,f}$ [p.u.]	0.0000	0.0001	0.0001	0.0002	0.0003
$\text{MAPE}_{G,P}$ [%]	0.0565	0.1439	0.1737	0.3802	0.7341
$\text{MAE}_{G,P}$ [MW]	0.5328	1.3585	1.6398	3.5887	6.9282
$\text{MAPE}_{G,Q}$ [%]	1.0945	2.3387	14.2739	12.8747	5.9319
$\text{MAE}_{G,Q}$ [Mvar]	0.7824	1.6718	10.2039	9.2037	4.2405
$\text{MAPE}_{G,f}$ [%]	0.0019	0.0075	0.0095	0.0161	0.0277
$\text{MAE}_{G,f}$ [p.u.]	0.0000	0.0001	0.0001	0.0002	0.0003

A.3.2 Quantitative Results of the Reduced BW Model

This section provides the quantitative results of the reduced BW model, which are referenced in Section 7.3.3. Tables A.24 to A.27 present the quantified errors, i.e., MAPE and MAE, for the four evaluated simulation cases. Each table includes the results for the three optimization approaches described in Section 7.3.3, as well as the two additional approaches with more buses for the objective selection. Independent of the optimization settings—and thus the definition of events and selection of objective variables—all approaches are evaluated on the same basis: The error values for all approaches are calculated for the same five buses and one generator.

Tab. A.25.: The short-circuit simulation shows much closer results for all optimization runs. While the no-event optimization still performs very well, other variants perform slightly better for the bus voltage and the reactive power of the generator.

	No events	Hybrid		Multi-event	
		2 Buses	3 Buses	2 Buses	5 Buses
$\text{MAPE}_{B,u}$ [%]	0.2444	0.2269	0.2776	0.3317	0.2265
$\text{MAE}_{B,u}$ [p.u.]	0.0026	0.0024	0.0029	0.0035	0.0024
$\text{MAPE}_{B,f}$ [%]	0.0124	0.0172	0.0153	0.0244	0.0282
$\text{MAE}_{B,f}$ [p.u.]	0.0001	0.0002	0.0002	0.0002	0.0003
$\text{MAPE}_{G,P}$ [%]	0.2574	0.3678	0.3535	0.5614	0.7176
$\text{MAE}_{G,P}$ [MW]	2.3889	3.4471	3.3256	5.1899	6.7435
$\text{MAPE}_{G,Q}$ [%]	19.9918	15.5044	20.7383	31.4921	11.2297
$\text{MAE}_{G,Q}$ [Mvar]	11.6925	9.1229	14.1264	19.1898	7.4829
$\text{MAPE}_{G,f}$ [%]	0.0125	0.0171	0.0152	0.0242	0.0286
$\text{MAE}_{G,f}$ [p.u.]	0.0001	0.0002	0.0002	0.0002	0.0003

Tab. A.26.: The simulation of the load step change reveals a disadvantage of the optimization without any events. As this event was not used for its optimization, the model resulting from the no-event optimization performs much worse than all other variants for this case. The hybrid version with two buses mitigates this disadvantage, resulting in the best accuracy overall.

	No events	Hybrid		Multi-event	
		2 Buses	3 Buses	2 Buses	5 Buses
$\text{MAPE}_{B,u}$ [%]	0.7715	0.0919	0.2338	0.2000	0.1142
$\text{MAE}_{B,u}$ [p.u.]	0.0081	0.0010	0.0025	0.0021	0.0012
$\text{MAPE}_{B,f}$ [%]	0.0582	0.0118	0.0135	0.0552	0.0324
$\text{MAE}_{B,f}$ [p.u.]	0.0006	0.0001	0.0001	0.0006	0.0003
$\text{MAPE}_{G,P}$ [%]	1.4379	0.2278	0.3140	1.4592	0.8652
$\text{MAE}_{G,P}$ [MW]	14.1680	2.2470	3.0974	14.4172	8.5442
$\text{MAPE}_{G,Q}$ [%]	48.2441	8.3567	19.9076	7.5720	11.1468
$\text{MAE}_{G,Q}$ [Mvar]	27.2575	4.7998	11.2878	4.3333	6.3448
$\text{MAPE}_{G,f}$ [%]	0.0575	0.0119	0.0134	0.0554	0.0324
$\text{MAE}_{G,f}$ [p.u.]	0.0006	0.0001	0.0001	0.0006	0.0003

Tab. A.27.: For the line dis-/connecting simulation, the no-event optimization yields the best results, again. However, the hybrid version with two objective buses outperforms the remaining three variants and is a more robust choice than the no-event optimization.

	No events	Hybrid		Multi-event	
		2 Buses	3 Buses	2 Buses	5 Buses
$\text{MAPE}_{B,u}$ [%]	0.0483	0.0880	0.1342	0.1388	0.1161
$\text{MAE}_{B,u}$ [p.u.]	0.0005	0.0009	0.0014	0.0015	0.0012
$\text{MAPE}_{B,f}$ [%]	0.0045	0.0098	0.0093	0.0153	0.0282
$\text{MAE}_{B,f}$ [p.u.]	0.0000	0.0001	0.0001	0.0002	0.0003
$\text{MAPE}_{G,P}$ [%]	0.1072	0.1859	0.1875	0.3609	0.7457
$\text{MAE}_{G,P}$ [MW]	1.0095	1.7491	1.7643	3.4005	7.0285
$\text{MAPE}_{G,Q}$ [%]	3.0327	3.7019	10.7652	13.4081	6.7758
$\text{MAE}_{G,Q}$ [Mvar]	2.1556	2.6522	7.6234	9.5782	4.8116
$\text{MAPE}_{G,f}$ [%]	0.0045	0.0098	0.0092	0.0153	0.0282
$\text{MAE}_{G,f}$ [p.u.]	0.0000	0.0001	0.0001	0.0002	0.0003