

Model Identification of Power Electronic Systems for Interaction Studies and Small-Signal Analysis

Master's Thesis

Author: Lisa Reis

Submission Date: 05.04.2023

Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU)

Faculty of Electric and Computer Engineering

Department of Power Electronics

Norges teknisk-naturvitenskapelige universitet (NTNU)

Faculty of Information Technology and Electrical Engineering

Department of Engineering Cybernetics

SINTEF Energy Research

Department of Energy Systems

Examiner: PD Dr.-Ing. habil. Bernhard Hauck

Supervisors: Jon Are Wold Suul, Assoc. Prof.; Salvatore D'Arco, PhD



Abstract

The rapid growth in offshore wind brings various challenges to power system research and industry, such as the development of multi-terminal multi-vendor HVDC grids. To ensure interoperability in those power converter dominated systems, suitable models are needed to efficiently perform stability and interaction studies. With state-space based small-signal methods stability and interaction phenomena can be assessed globally for a complex system. Yet detailed models are needed. However, in multi-vendor projects most likely only black-boxed models will be available to protect the intellectual property, so that identification techniques are necessary to obtain suitable models. This thesis contributes to the research activities on state-space model identification of black-boxed power electronic systems.

In the first part of the thesis, a method was developed and tested, where the matrix elements of linearized state-space models were fitted in dependency of the operating point, based on input sweeps performed on the model of a grid forming power converter controlled as a virtual synchronous machine. It was discussed how changes in multiple inputs can be approximated by the superposition of the individual input dependencies and a fully operating point dependent state-space model approximation was created. The results were validated in time and frequency domain analyses. It was found that the method can provide a good approximation, especially for the operating range around the default operating point.

In the second part, identification of a power electronic system was performed based on measurement data which was generated experimentally from a low voltage laboratory system. A sequence of input perturbations was applied to the laboratory system and frequency response data was calculated from the corresponding output perturbations. The data served as basis for model identification with N4SID and a soon to be published vector fitting method. The identified models were validated by a visual inspection of the transfer function and by comparison of the calculated step responses to the step responses measured in the laboratory. It was found that the treatment of incomplete data sets, the generation of substitute data and the impact of time delays on the identification might be worth further investigation.

This work provides a valuable contribution to the research of state-space model identification of black-boxed power electronic systems. It points out challenges and presents promising approaches to enable state-space based methods for stability analysis and interaction studies in future multi-terminal multi-vendor HVDC grids.

Contents

1. Introduction	1
1.1. Motivation	1
1.1.1. Development of Offshore Wind	1
1.1.2. HVDC and Offshore Grids	2
1.1.3. Power Converter Dominated Grids: Stability and Interoperability	3
1.1.4. Significance of Model Identification for Interoperability Studies	3
1.2. State of the Art	4
1.2.1. Industrial Research and Demonstration Projects	4
1.2.2. Commercial Projects	5
1.2.3. Scientific Publications	7
1.3. Contribution of this Work	7
1.4. Outline of the Thesis	12
2. Theoretical fundamentals	13
2.1. Definition of Power System Stability	13
2.1.1. Classical Definition	13
2.1.2. Adapted Definition	14
2.2. Interactions and Interoperability Issues	14
2.3. Power Converter Control	15
2.3.1. Classification	15
2.3.2. Virtual Synchronous Machines	16
2.4. System Representation and Modelling	17
2.4.1. Overview	17
2.4.2. State-Space Model	17
2.4.3. Frequency Domain and Transfer Function Matrix	19
2.5. Small-Signal Analysis	19
3. Model Identification Techniques	22
3.1. General Remarks	22
3.2. Vector Fitting	22
3.3. N4SID	23
3.4. PEM	24
3.5. Application of Identification Methods in this Thesis	24
4. Fitting of an Operating Point Dependent Model	26
4.1. Overview	26
4.2. System Definition and Delimitation	27

4.3.	Selection of Operating Points	29
4.4.	System Representation	30
4.4.1.	Selection of System Representation	30
4.4.2.	Transformation into Complex Diagonal Form	30
4.4.3.	Sorting and Clustering of Eigenvalues	31
4.4.4.	Filtering of States	32
4.4.5.	Rearranging the State-Space Model	32
4.4.6.	Finding the Correct Factors for the B- and C-matrix	32
4.5.	Fitting	33
4.5.1.	Fitting of the Individual Input Sweeps	33
4.5.2.	Fitting of Pairwise Combined Input Sweeps	35
4.5.3.	Approximation of Pairwise Combination by Superposition	36
4.5.4.	Approximation of Generalized Model by Superposition	36
4.6.	Review of Fit Quality	37
4.6.1.	Overview	37
4.6.2.	Review of Fit Quality per Matrix Element of the State-Space Model	37
4.6.3.	Review of Fit Quality Based on the Frequency Response	38
4.6.4.	Review of fit Quality Based on the Step Response	39
4.7.	Fit Function Analysis	40
4.8.	Adaptive Model for Simulations	41
5.	Laboratory Experiments and Identification	43
5.1.	Overview	43
5.2.	Frequency Response Estimation	43
5.3.	Laboratory Setup	44
5.4.	Data Acquisition for Transfer Function Estimation	45
5.5.	Adaption of Perturbation Amplitudes	46
5.6.	Validation by Step Response	47
5.7.	Treatment of Transformer Phase Shift	48
5.8.	Identification and Treatment of Missing Data	49
6.	Results and Analysis Operating Point Dependently Fitted Model	51
6.1.	Individual Input Sweeps	51
6.1.1.	Eigenvalue Trajectories	51
6.1.2.	Fitting	55
6.1.3.	Validation in Frequency Domain	56
6.1.4.	Validation in Time Domain	60
6.1.5.	Discussion of Fit Functions	61

6.2. Pairwise Combination of Input Sweeps	64
6.2.1. Fit of Matrix Elements	64
6.2.2. Discussion of Fit Functions	64
6.2.3. Validation in Frequency Domain	67
6.2.4. Validation in Time Domain	73
6.3. Combination of all Inputs by Superposition	77
6.4. Summary Fitted Model and Approximation by Superposition	82
6.5. Adaptive Simulation Model	83
6.6. Fits from Model Identification	85
7. Results and Analysis Laboratory Experiments and Identification	88
7.1. General Remarks	88
7.2. Measurement Results	88
7.3. Identification Results	91
7.4. Validation by Step Response	94
7.5. Summary	96
8. Summary of Results	99
9. Discussion and critical consideration	101
9.1. Importance of efficient state-space identification methods	101
9.2. Operating point dependently fitted model	101
9.3. Laboratory Experiments and Identification	103
10.Final Conclusion	105
11.Outlook	105
Literature	VIII
Image Directory	XI
Tables Directory	XII
Code Listings Directory	XIII
List of Abbreviations	XIV
Nomenclature	XV
A. dq-Transformation	XVII
B. Additional Equations VSM	XVII

C. Similarity Transformations	XVIII
D. Conversion to Diagonal Form	XVIII
D.1. Complex Diagonal Form	XVIII
D.2. Real Block Diagonal Form	XIX
D.3. Degrees of Freedom	XIX
E. Parameters Analytical CCVSM model	XXII
F. Eigenvalues and HSVs of the Analytical Model	XXIII
G. Parameters Laboratory Investigations	XXIV
H. Pictures Laboratory Setup	XXV
I. Figures Laboratory Measurements and Identification	XXVI
J. Code for Fitting of Input Sweeps	XXXI

1. Introduction

1.1. Motivation

1.1.1. Development of Offshore Wind

In the global energy transition the renewable energy sources wind and solar power will play a central role to achieve the transformation into a sustainable energy system. The International Renewable Energy Agency (IRENA) predicts the global share of wind and solar needed in 2050 to limit the rise in global temperature to between 1.5 and 2 °C to 61%. [1] According to a report from 2019 about 17% of the globally installed wind capacity will be located offshore, meaning an installed offshore capacity of 1000 GW. [2] This would be an increase in globally installed offshore capacity by the factor 10 compared to 2018. [2]

In Europe the governments have committed to ambitious offshore wind energy targets and the North Sea area will make up a significant share of this. At the North Sea Summit in Esbjerg in May 2022 Germany, Denmark, Belgium and the Netherlands have signed a declaration on a common development of a offshore energy system. It proclaims a joined target for installed offshore wind power of at least 60 GW by 2030 and at least 150 GW by 2050 [3]. The UK has committed to 40 GW installed offshore wind capacity by 2030. [4]

Although in 2021 there were barely two offshore wind turbines installed [5] with a capacity of only 6 MW [6] in Norway, the Norwegian Government has committed to a promotion of offshore wind power to achieve 30 GW by 2040 for energy supply for the mainland, the petroleum sector and a significant share for export. [5] There is an ambition to take a leading role in floating offshore wind. In November 2022 the floating wind park Hywind Tampen started the energy production to supply to two oil and gas fields with power [7]. In March 2023 the tender phase for 3 GW offshore wind in the Norwegian North Sea was opened. [8]

WindEurope expects about 28 GW of newly installed offshore wind energy in Europe until 2026, which means that the installed capacity will be doubled compared to 2021 in only 5 years. [6] The integration of this rapidly growing offshore generation brings various challenges for power system research and industry. The development of offshore grids is one of them.

1.1.2. HVDC and Offshore Grids

The most efficient and most economical technology for the purpose of long distance bulk power transmission as needed for the transport of energy from distant offshore wind farms to shore is high voltage direct current (HVDC). It is a proven technology with the first commercial system commissioned in 1954 [9]. High voltage alternating current (HVAC) is transformed into HVDC and vice versa by a HVDC converter. The HVDC voltage source converter (VSC) technology that was developed in the last decades has a higher degree of controllability compared to the conventional line-commutated converter (LCC) and makes a simple reversal of power flow possible. This enables an easier operation in more complex configurations such as multi-terminal systems or DC grids [10].

So far and especially in Europe, HVDC has usually been built as a point-to-point connection. The development from point-to-point connections towards offshore grids will be important for the cost effective integration of the high number of upcoming offshore wind farms in Europe. This is elaborated in detail in the CIGRE feasibility study [11]. In February 2023, the German transmission system operators (TSO) and the Federal Ministry for Economic Affairs and Climate Protection has published their plans to interconnect 10 GW of wind power, including an option for further connections to the neighbouring countries Denmark and the Netherlands. [12]

Another use case of HVDC and offshore grids, which is particularly interesting for Norway, is the electrification of offshore oil and gas platforms. The country is one of the world's largest oil and gas producers. In 2022 it delivered approximately 10 % of the European Union's (EU) oil imports and became the biggest supplier of natural gas with up to 30 % of the EU's gas imports coming from Norway in the third quarter of 2022 [13]. Up to now, the oil and gas platforms are usually powered by natural gas or diesel generators. De-carbonizing the energy supply for the energy intensive processes by electrification with low emission power from shore or wind power could be a significant contribution to the reduction of climate gas emissions in Europe.

The expansion towards offshore grids will presumably be a step wise process, combining various topologies and technologies, most likely from different vendors. These offshore grids will characteristically be dominated by power converters, both from HVDC and the wind turbines. [14]

1.1.3. Power Converter Dominated Grids: Stability and Interoperability

The predominance of power converters that is observed in offshore systems will also evolve in the onshore power system. The number of conventional power plants decreases and the number of power electronic interfaced generation units as onshore wind and solar power rises. More and more HVDC converters that connect offshore wind parks to the main land will be erected. Also, the number of onshore HVDC connections that are needed for the transmission of power throughout the continent will grow. All this will lead to a power converter dominated system as a whole.

In contrast to grids with many synchronous machines with high inertia, in these power electronic dominated grids the converter control will be crucial for a stable operation of the power system. The dynamic behaviour of converters differs significantly from the one of classic generation units and high-frequency phenomena gain relevance in terms of power system stability [15].

Moreover, because of the big variety of power electronic devices a reliable inter-operation between different types of converters from multiple vendors needs to be ensured. This applies to all grids where converters operate in close vicinity and concerns all types of high power electronic devices. Interaction phenomena can appear for example:

- between the inverters of generation units, e.g. in a wind farm or in a rural area with a high penetration of solar powered generation,
- between the inverters of generation units and a HVDC converter, e.g. in an offshore wind park,
- between multiple HVDC converters, for example between parallel HVDC lines or inside a multi-terminal HVDC grid.

Control interactions are not a theoretical phenomenon but have been observed in the real world. For example in the project BorWin1, harmonic oscillations in a HVDC connected wind farm led to an outage of the HVDC system [16]. Sub-synchronous oscillations were observed in a HVDC project for wind farm integration in China [17]. More examples of oscillations caused by control interactions are listed in [18] and the interoperability of multi-terminal multi-vendor HVDC is discussed in detail in the dedicated ENTSO-E position paper [19].

1.1.4. Significance of Model Identification for Interoperability Studies

Interoperability issues concern all participants in the power system, especially the system operators, the converter manufacturers and the power plant operators. To

ensure a stable system operation, tools for analysis, planning and operation of the system are needed to identify critical operational cases and interaction modes.

For these investigations, suitable models of the converters and their control systems are needed [20]. At the same time, the technical details as the intellectual property of the manufacturers has to be protected, which is why all converter controls are so far and presumably also in future projects provided as black-boxed models [20]. These models make the analysis in time-domain possible. Yet, for small-signal analyses with impedance or state-space based methods as they are commonly applied, models need to be created from the black-boxed system first. The advantage of state-space based small-signal analyses is that stability and interoperability can be assessed globally over the whole system. However, a detailed model is necessary. So far, eigenvalue-based stability analyses with black-boxed models are not possible and can therefore not be carried out in a multi-vendor setup. This leads to the motivation of this thesis, which is to explore possibilities to identify state-space models for small-signal analyses from black-boxed systems.

1.2. State of the Art

1.2.1. Industrial Research and Demonstration Projects

There have been various research activities that deal with the operation of power converter dominated power systems. This section gives a brief overview of past and present activities in industrial research and demonstration projects.

The MIGRATE project that was completed in 2019 showed that for the test that was carried out on the Irish power system model an operation with 100 % PE-interfaced generation is possible if so called grid-forming control schemes were applied to 30 % of the generation. Sub-synchronous controller interactions were observed when converters were in close vicinity to each other. [21]

In the BestPaths project [22] that ended in 2018 the focus was put on interoperability in multi-vendor HVDC-VSC systems. Interoperability issues between two VSCs of different vendors were investigated in offline time-domain and real-time Hardware in the loop (HiL) simulations and a methodology to fix interoperability issues was developed. One result of the research activities is the statement that interoperability issues cannot be avoided completely during the design phase but that suitable and precise specifications can reduce the amount of interoperability problems. Detailed interaction studies during the planning phase will always be needed to ensure the function of the system and demanding a plug and play solution that works without

knowledge on the other parts in the system is not reasonable. The approach that was applied in this project is the following:

- A third party carries out the interoperability studies to ensure the protection of the vendors' intellectual property. The third party coordinates adaptations that the vendors need to make.
- For offline time-domain simulations the vendors deliver black-boxed models that show the same behaviour as a generic model that is used to make the specification on the expected AC and DC behaviour [20].
- Hardware in the loop simulations are carried out with replicas of the system and real control cubicles.

Although this approach proved most effective during the project, it is connected to a high effort and does not allow modal stability analyses due to the lack of detailed models. For a system with more converters, the number of operational cases that have to be considered grows exponentially and comes with an enormous computational effort. This approach is therefore not suitable for big DC grids with a high number of possible configurations to analyze.

One of the European research projects that are currently in progress is the project HVDC-WISE [23]. It started in October 2022 and includes the development of planning and analysis tools for European HVDC systems in its project description. It can be assumed, that this includes the assessment of interoperability, even though it is (as of December 2022) not explicitly stated in the available project description. The project READY4DC [24] that runs from April 2022 until October 2023 focuses on a European multi-vendor DC grid. It addresses interoperability and interaction studies in its working groups.

The project InterOPERA [25, 26] started in January 2023. There are 20 European partners involved, amongst others TSOs, wind turbine and HVDC manufacturers and universities. The aim is to enable multi-terminal multi-vendor offshore HVDC grids in Europe. So called hybrid wind farms and energy islands that interconnect multiple countries play a central role in the project description. It is planned to develop technical standards and frameworks for the realization of commercial projects.

1.2.2. Commercial Projects

In commercial HVDC projects, commonly all kinds of stability analyses are carried out (time-domain, impedance based and state-space analyses). State-space and impedance based methods are typically used to assess the small-signal behaviour. However, with black-boxed models as they are supposedly provided in multi-vendor

setups, these methods are not directly applicable. The vendors only have full knowledge about their own system, so that conventional small-signal analysis can only be performed on each converter separately.

Until now there are not many multi-vendor HVDC projects in operation and the practical experience is limited. The two HVDC multi-vendor multi-terminal connections Nanao and Zhousan that are in operation in China were built with a different approach than the one that will be needed for the projects planned in Europe. In the two named projects the control systems for all converters was delivered by an independent supplier while the converter manufacturers only provided the converter hardware. Also, the market situation and the regulations concerning the intellectual property differ from the European ones. Furthermore, these projects were designed as multi-terminal connections from the beginning of the design phase, while the European grids will most likely evolve from a step wise expansion. [27]

The approach that was used to assess interoperability for the projects in the north of France where multiple HVDC links (both LCC and VSC), as well as a static VAR compensator (SVC) and a wind farm are located in close vicinity is presented in [28]. The method is a screening method on empirical basis and also relies on time domain simulations in combination with the application of the Multi Infeed Interaction Factor (MIIF) introduced in [29].

In the Norwegian Johan Sverdrup project a MMC HVDC link was added in parallel to an existing 2-level VSC HVDC link of a different vendor. The connections supply an offshore oil field with power from the main land. The second connector was commissioned in 2022 and the two links are in parallel operation since December 2022. It is the first multi-vendor HVDC system in grid-forming operation. Comparably to the approach recommended in the BestPaths project, the interaction studies were carried out on the black-boxed models by a third party. The necessary modifications were then implemented by the vendors and an updated black-boxed model was handed to the independent party. In an iterative process the interoperability issues were removed. [30]

In [31] work streams to develop multi-vendor multi-terminal HVDC grids are presented. Next to the demand for standardization, the proposed interoperability studies consider generic models for time domain simulations as well as Software in the Loop (SiL) and Hardware in the Loop (HiL) testing. Modal analyses are not mentioned in the report, presumably due to the lack of suitable models.

To sum up, usually all kinds of stability analysis methods are applied when a HVDC connection is planned. In European multi-vendor setups only black-boxed models have been exchanged and this will supposedly remain the case in the future. This

means that in multi-vendor setups the classical small-signal methods are only applicable to the converters of each vendor separately. So far the number of interacting converters in commercial multi-vendor settings has been small. Currently all published reports on practically applied methods to assess interactions between the converters of multiple vendors are based on time-domain simulations, SiL and HiL investigations. To carry out small-signal analyses based on black-boxed models identification methods like the ones presented in this thesis would have to be applied to gain suitable models. To date there is no evidence that this has been tried in a commercial multi-vendor projects.

1.2.3. Scientific Publications

There are many scientific publications that deal with stability analyses of power electronics dominated systems. The main approaches to assess small-signal stability are impedance based and state-space methods. [32] Besides the classical eigenvalue based-analysis, methods to assess stability in complex power systems based on state-space models are the calculation of participation factors [33, 34] and parametric sensitivity [34]. In [34] a modular approach is presented where the state-space model of a complex system is assembled from the state-space models of the individual components. On an example system of a multi-terminal HVDC it is discussed how interactions between converters can be assessed by this method.

If only black-boxed models are available, model identification techniques as applied in this thesis are necessary to obtain suitable models. Numerous different approaches and algorithms for model identification exist. One of them is vector fitting, which is used in this thesis. In [35] a new stability analysis method was introduced where vector fitting is applied to identify the eigenvalues of a black-boxed system. In [36] it is for example shown how vector fitting can be used to identify the state-space model of single-input single-output converter systems.

1.3. Contribution of this Work

The project Ocean Grid, which this work is a part of, deals with the development of offshore grids to enable a cost efficient integration of offshore wind in Norway. One of the five sub-projects is a knowledge building-project that investigates among others the interoperability of converters in a power-electronic dominated offshore system. [37] The thesis work contributes to the research activities in model identification of black-boxed power converters. The findings could support the establishment of

new techniques to obtain state-space models for interoperability studies and enable analyses based on state-space models also in multi-vendor setups.

The thesis focuses on two topics in particular. In the first part the aim is to create an input-dependent linearized state-space model of a virtual synchronous machine by fitting input sweeps. In the second part laboratory experiments are carried out and two different identification techniques are applied to identify the system based on the measurements.

In the first part the focus is put on the input dependency. To identify a black-boxed simulation model it is in practice necessary to create input data by simulation which is associated with a significant amount of computational effort. Moreover, the number of relevant operating points might be high. Therefore, it would be favourable to reduce the number of operating points for which such an identification is necessary. Hence, it is investigated if the model behaviour can be extrapolated by suitable fit functions.

To fit the operating point dependent linearized model it is necessary to generate the state-space model for a certain number of operating points first. Then the matrix elements of the state space models are fitted with a mathematical function in dependency of the input. The principle of the fitting done for the operating point dependent model is displayed in Figure 1.1

Since the number of input combinations grows exponentially with the number of inputs, a method is developed and tested where at first each input variable is swept over its input range and input combinations are approximated by linear superposition of the individual input dependency. Thus, at first a dependency is fitted for each input separately. Then the combination of all inputs is approximated by a superposition of the individual functions of each input. The basic principle of the superposition is pictured in Figure 1.2 for an example with two input variables. The resulting models are analysed in frequency and time domain and discussed in detail.

The analyses are carried out on the linearized state-space model based on analytical equations of a virtual synchronous machine documented in [34]. It is also tested how the application of identification techniques influences the method, by applying identification methods to the transfer function data calculated from the linearized analytical model. An overview of the different models used in the thesis can be found in Figure 1.3. For the fitting in part 1 of the thesis the models marked by the blue boxes are used. The identification from the transfer function is the ideal case and not realistic for black-boxed systems. Still, it is reasonable to test the new method first with this simpler version. The application of the fitting method to

models from simulation data or measurement marked in grey and green in Figure 1.3 was not tested during this thesis.

In the second part the identification of a system based on laboratory measurements is discussed. This process is represented by the rightmost part of Figure 1.3. While in reality it is more effort and usually more expensive to create a laboratory setup compared to carrying out simulations, identification from measurements might be relevant if no simulation model is available. Also, for interaction studies for multi-vendor HVDC the current recommendation is to include Software- and Hardware-in the Loop simulations to detect interoperability issues [31], so that the laboratory setup might be needed anyways. Since for the generation of the needed input data for an identification the real-time simulations are from experience faster than simulations, it might be favourable in those cases to generate the input data for identification from the real-time setup instead.

In this thesis the input data for the identification was obtained from measurements on a low voltage setup with a two-level VSC controlled as a virtual synchronous machine. The process applied to measure transfer function data is presented. The limitations of laboratory investigations compared to simulation. Due to the physical constraints of the measurement process in the laboratory, the measurements result in incomplete data sets. The impact on the identification and the necessity to generate substitute values for this missing data is discussed. The system is identified using two different methods and the results are validated by comparing the step response of the identified models with step responses measured in the lab.

The thesis work explores the basic challenges and limits of the two approaches. It serves as a foundation for further related research on state-space model identification for power converters.

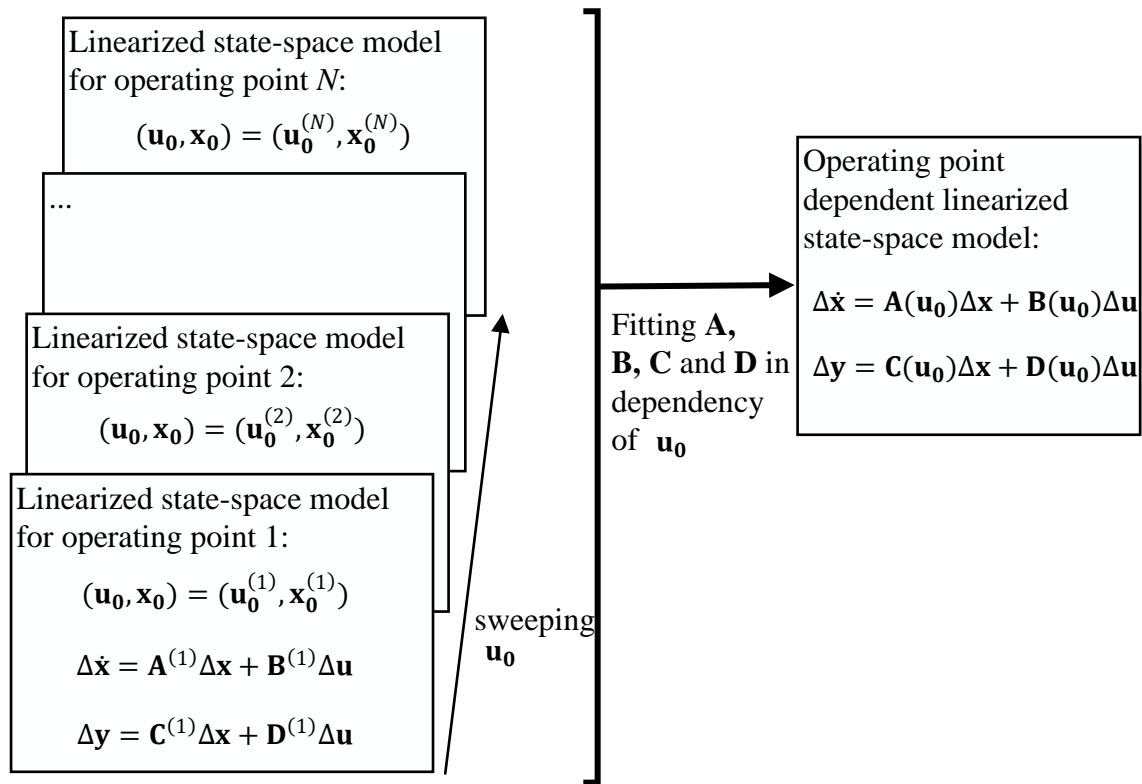


Figure 1.1: Overview of the fitting approach

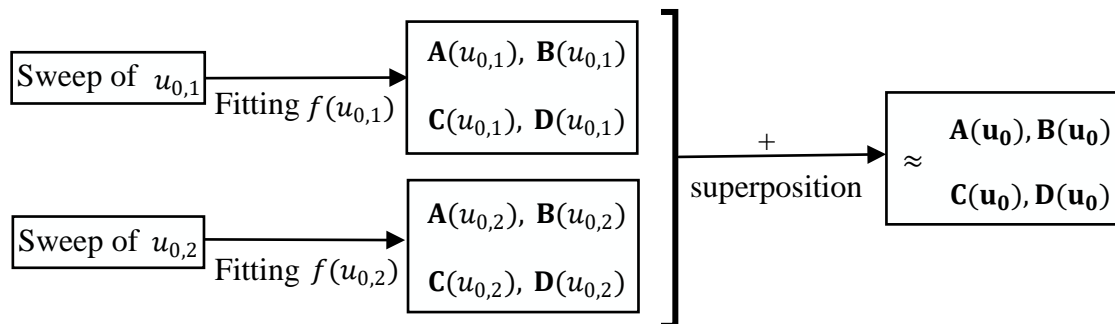


Figure 1.2: Principle of the superposition method for a sample system with two input variables

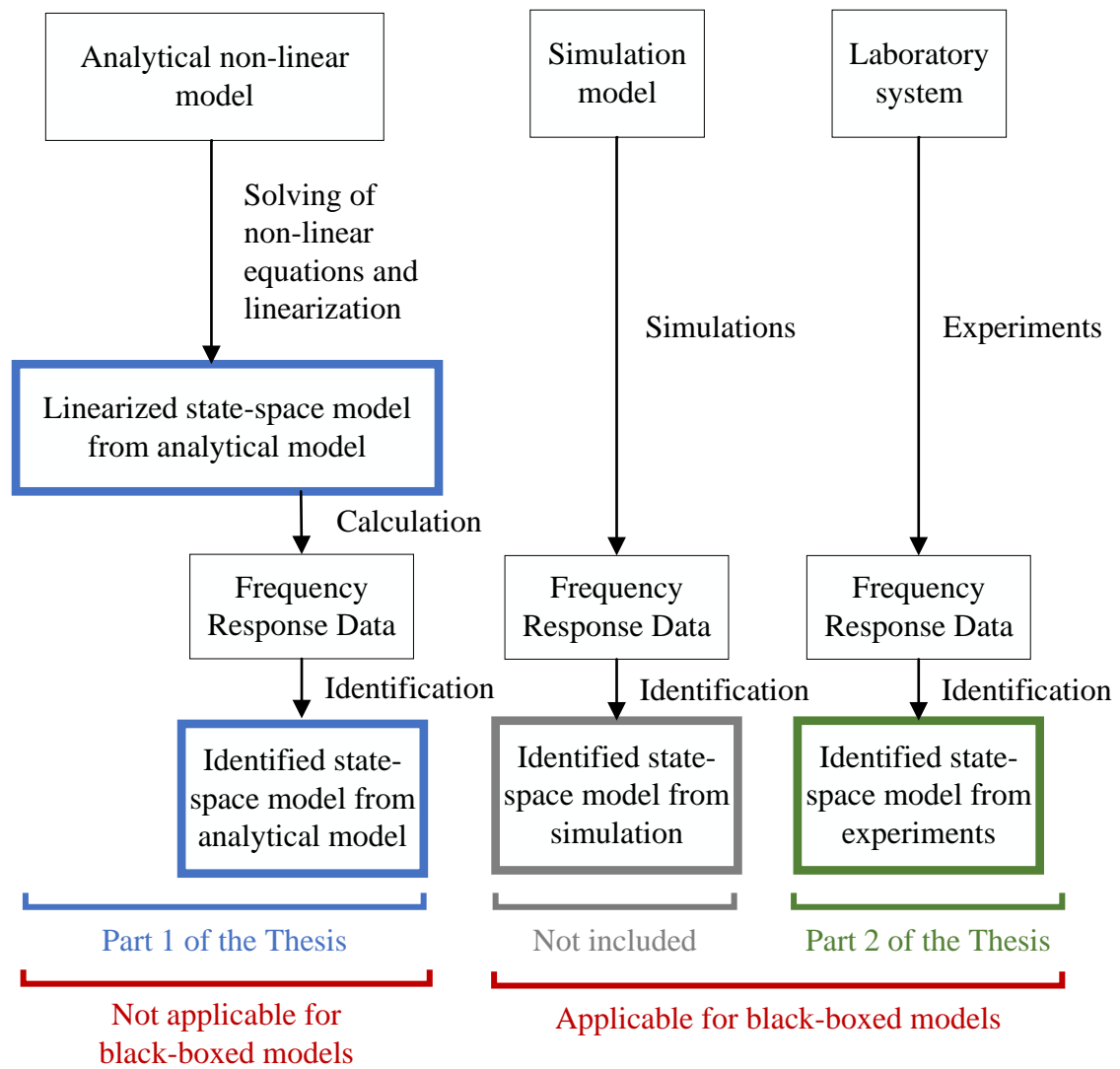


Figure 1.3: Overview of models used in this thesis

1.4. Outline of the Thesis

In section 2 the fundamentals of power system stability and power converter control are outlined. The relevant basics of system representation and control theory methods are presented there as well. In section 3 information is given about the identification techniques applied in this thesis. In section 4 the approach for the generation of the operating point dependent model that was developed during this thesis is explained. The methodology for the laboratory experiments and the corresponding identification are explained in section 5. In section 6 and section 7 the results are presented and discussed. A summary of the results is given in section 8. In section 9 a critical consideration of the thesis work is done. A final conclusion is drawn in section 10 and at last an outlook for the use of this work for further research activities can be found in section 11.

2. Theoretical fundamentals

2.1. Definition of Power System Stability

2.1.1. Classical Definition

Power system stability phenomena have been discussed and analyzed since more than 100 years [38, 39]. In [38] the definition of power system stability is formulated as follows:

"Power system stability may be broadly defined as that property of a power system that enables it to remain in a state of operating equilibrium under normal operating conditions and to regain an acceptable state of equilibrium after being subjected to a disturbance." [38]

In 2004 a formal definition of power system stability was proposed by the dedicated IEEE/CIGRE Joint Task Force in [40], that was commonly adapted and is still valid. The definition reads as follows:

"Power system stability is the ability of an electric power system, for a given initial operating condition, to regain a state of operating equilibrium after being subjected to a physical disturbance, with most system variables bounded so that practically the entire system remains intact." [40]

The key element of power system stability is that, even after a given disturbance, the system state is in or goes back to a steady state, which is viable from an operational point of view and does not deteriorate. Power system stability is not an absolute property, but refers to a given initial condition and a given disturbance.

Stability phenomena appear in different forms and can be classified based on their physical nature. The following three main categories are defined in [40]:

1. **Frequency stability:** is a phenomenon of active power balancing and refers to the ability of the system to maintain or regain a steady frequency
2. **Rotor angle stability:** is a phenomenon of synchronous machines and refers to the ability of the synchronous machines to stay in or regain synchronism
3. **Voltage stability:** is a grid phenomenon and refers to the ability of the system to maintain or regain a steady and acceptable voltage level.

The phenomena can be further differentiated as discussed in [40].

2.1.2. Adapted Definition

The classical power system stability definition was established in a time when the power system behaviour was dominated by the synchronous machines, which is why the definition mainly focuses on phenomena that are significantly slower than the switching effects and the control of power electronics. Moreover, with the increasing share of power converters an increase in interactions between converter controls and a decrease in system inertia and short circuit current comes in. Therefore, an extension of the definitions and classification of power system stability was presented in [15] in 2021. The impact of power electronic systems in the categories of the classical stability phenomenon classification is discussed there in detail. Also, two categories were added to the classification:

- **Resonance Stability:** refers to energy exchange in oscillations that grow due to the lack of energy dissipation.
- **Converter-driven Stability:** refers to oscillations that evolve from interactions between converters and machine dynamics, network elements or other power system components.

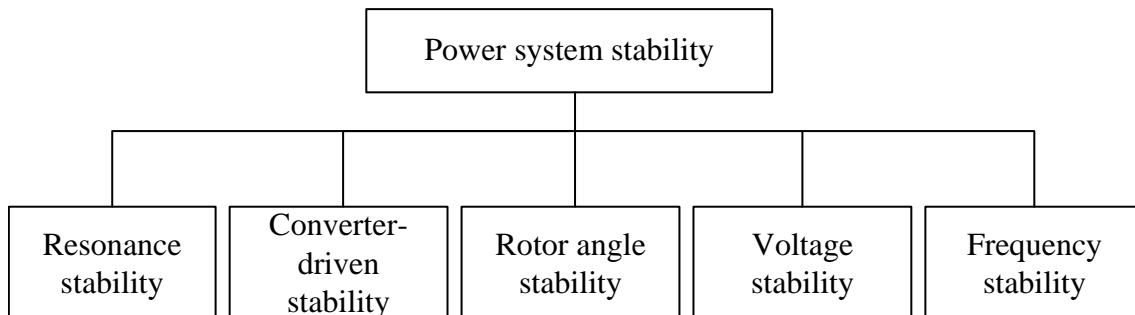


Figure 2.1: Stability phenomena with adapted classification [15]

Figure 2.1 shows the overview of categories of power system stability. The subcategories are not displayed in Figure 2.1 but are discussed in detail in the corresponding literature [40, 15]. The different system-theoretic approaches to assess power system stability are presented in [40]. This thesis focuses on small-signal stability and the corresponding basic theory can be found in subsection 2.5.

2.2. Interactions and Interoperability Issues

There are several different definitions and classifications of interactions and interoperability issues between power electronic devices [22, 20, 28, 19]. Generally speaking, interoperability is the ability of systems to function together. An interaction on the

other hand is "a reciprocal action exerted by a system on one or several other systems" [28]. While not every interaction has a negative effect, all interoperability issues are caused by interactions.

In [20] eight different categories of interoperability issues are given. This thesis could be relevant for the detection of two of them:

- oscillatory behaviour,
- difference in converter dynamics.

Sub-synchronous oscillations (SSO) and harmonic oscillations are major concerns. SSO appear in the frequency range smaller than 50 Hz and are electro-mechanical interactions between the mechanical masses of the generators and passive or active system elements, including amongst others HVDCs. Harmonic interactions on the other hand are eletro-magnetic interactions between power electronic devices and other power electronic devices or passive system elements. They can reach frequency ranges up to several kilohertz. [41, 18]

For example, in [17] a case is discussed where sub-synchronous oscillations appeared in a MMC-HVDC connected wind farm in China. There the offshore converter has a resonance at approximately 21 Hz that in combination with a load-dependent impedance of the wind generation made the system oscillate for operational cases with high load. In the paper the mathematical model is derived and the simulated phenomena are compared to the in the real world on-site measured oscillations.

2.3. Power Converter Control

2.3.1. Classification

The control techniques used for power converter control can be mainly divided into three categories:

- grid-following (also known as grid-feeding),
- grid-supporting,
- grid-forming. [42]

Grid-following control is the conventional control strategy for power converters of distributed generation units. The main goal of this type of control is to provide active and reactive power to the grid. In this case the converter operation is dependant on an energized grid with at least one synchronous generator or a grid-forming converter setting the grid voltage and frequency.

With the increasing share of power electronic converter connected generation the need for the converters having the ability to provide ancillary services arose. The grid-supporting control adapts the active and reactive power flow depending on the grid voltage and frequency making a contribution to the system's voltage and frequency regulation. The simplest example for the implementation is adding active power frequency droop and reactive power voltage droop controllers to a conventional control scheme.

The main goal of grid-forming converters on the other hand is to provide the grid voltage and frequency. Grid-forming converters do not depend on an energized grid and are able to operate in island mode. [42]

A more detailed discussion of the control techniques used for power converters, their classification and their operational features can be found for example in [42] and [43].

2.3.2. Virtual Synchronous Machines

One approach to deal with the decreasing number of synchronous machines and the reduced inertia in the power system is to control power converters in a way that imitates the behaviour of a synchronous machine [44]. The key aspects of this virtual synchronous machine (VSM) approach are the emulation of inertia and the damping of oscillations. Both are represented in the swing equation of the form of (2.1), where J is the inertia, ω the rotating speed, M_m the mechanical torque, M_{el} the electrical torque, D the damping coefficient and ω_g the rotating speed of the grid.

$$J \frac{d\omega}{dt} = M_m - M_{el} - D(\omega - \omega_g) \quad (2.1)$$

With the time constant T_a defined according to (2.2) and the equations in Appendix B, the swing equation can be formulated as (2.3), where p_m is the virtual mechanical power, p the electrical power and k_d the damping constant. The swing equation provides rotational frequency and angle for the following control blocks.

$$T_a = \frac{J\omega_0}{M_r} = \frac{J\omega_0^2}{S_b} \quad (2.2)$$

$$\frac{d\omega_{pu}}{dt} = \frac{p_m}{T_a} - \frac{p}{T_a} - \frac{k_d(\omega_{pu} - \omega_{g,pu})}{T_a} \quad (2.3)$$

Moreover, the quasi-stationary electrical behaviour can be emulated to make the VSM less sensitive to small disturbances. A so called virtual impedance can be introduced. [44].

2.4. System Representation and Modelling

2.4.1. Overview

Power electronic converters behave in general non-linearly. In this thesis linearized models are used. This means that the behaviour for small deviations around the equilibrium point is assumed to be linear. A Taylor series representation around the equilibrium point provides a linear model that is valid for small deviations around this equilibrium point. This is a common approach in the analysis of power system stability [40].

To achieve a time-invariant representation of the time-periodic AC quantities the transformation into a synchronously rotating reference frame is applied resulting in DC-like quantities, called d- and q-component. It is possible to describe three phase quantities with two dq-components if the system conditions are balanced. This means that all three currents or voltages in a three phase quantity have the same amplitude and are shifted by -120° . [45] The equations for the transformation can be found in Appendix A.

Due to the linearization, the transformation to the synchronous reference frame and the fact that all currents and voltages should settle to be constant in steady state, the model can be treated as a linear time-invariant (LTI)-system for operating points close to the equilibrium point. Then the methods of linear control theory can be applied. Due to the linearization, the models used in this thesis are only suitable for small-signal analysis. [45] The mathematical system representations that are relevant for this thesis are the state-space model and the frequency domain representation, which are discussed in the following. As this work deals with multiple-input multiple-output (MIMO) systems, all following remarks refer to MIMO-LTI-systems.

2.4.2. State-Space Model

In the state-space model, the system behaviour is described by equations of the form of (2.4) that relate the derivative of the states and the output vector \mathbf{y} to the state vector \mathbf{x} and the input vector \mathbf{u} .

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= g(\mathbf{x}, \mathbf{u})\end{aligned}\tag{2.4}$$

The transition to a linearized state-space model is documented in [45]. The linearized model has the form of (2.5), where \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} contain the operating point dependent coefficients that describe the model behaviour. \mathbf{x}_0 and \mathbf{u}_0 are the state and input vector of the equilibrium point for which the linearization is done and Δ marks the deviation from the corresponding values in the equilibrium point.

$$\begin{aligned}\Delta\dot{\mathbf{x}} &= \mathbf{A}(\mathbf{x}_0, \mathbf{u}_0) \cdot \Delta\mathbf{x} + \mathbf{B}(\mathbf{x}_0, \mathbf{u}_0) \cdot \Delta\mathbf{u} \\ \Delta\mathbf{y} &= \mathbf{C}(\mathbf{x}_0, \mathbf{u}_0) \cdot \Delta\mathbf{x} + \mathbf{D}(\mathbf{x}_0, \mathbf{u}_0) \cdot \Delta\mathbf{u}\end{aligned}\tag{2.5}$$

For a system with n_i inputs, n_o outputs and n states, \mathbf{A} is a $n \times n$ -matrix, \mathbf{B} a $n \times n_i$ -matrix, \mathbf{C} a $n_o \times n$ -matrix and \mathbf{D} a $n_o \times n_i$ -matrix.

A state is typically associated with an integrator in the control system or with a physical quantity in an energy storage element like the voltage of a capacitor or the current in an inductor. The set of states is not unique so that various equivalent state space representations exist. These representations are related by so called similarity transformations, which are discussed in Appendix C. Typically, when setting up a state space model from a given physical or control system, the states would be chosen to have a physical interpretation, but this is usually not the case for other representations such as the diagonal form used in later sections.

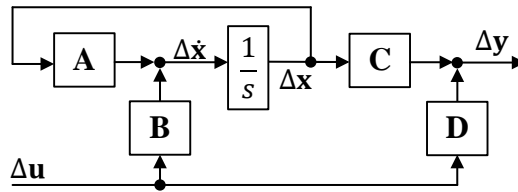


Figure 2.2: Block diagram for a generic state-space model

The control scheme of a state-space model is displayed in Figure 2.2. The advantage of a state-space model is that the system behaviour is described by a matrix representation so that simulations are usually faster than with a full electrical model. The state-space model is in general harder to obtain but gives more information about the overall system behaviour than impedance based analyses that assess stability only locally for the node that they are applied to. [36] With a state-space model available, eigenvalue based analyses are directly applicable and participation factor and sensitivity analyses can be applied which is useful to analyse the possibility of interaction phenomena. [33, 34, 36]

The analyses and the nomenclature in this thesis are based on the linearized small-signal state-space model documented in [45]. To obtain an explicit state-space model for a specific operating point, it is necessary to solve the nonlinear state equations to receive the steady-state values. The steady-state values are then inserted into the linearized model to calculate numerical values for the elements of \mathbf{A} , \mathbf{B} and \mathbf{C} .

2.4.3. Frequency Domain and Transfer Function Matrix

Another possibility for the system representation of LTI-systems is using frequency domain transfer functions. They relate the Laplace-transformed output and input signals $\mathbf{Y}(s)$ and $\mathbf{U}(s)$ according to (2.6), where s is the Laplace variable and $\mathbf{H}(s)$ is the transfer function matrix. [46, 47]

$$\mathbf{Y}(s) = \mathbf{H}(s) \cdot \mathbf{U}(s) \quad (2.6)$$

$\mathbf{H}(s)$ consists of one transfer function for each input-output pair. The matrix element in the k th row and l th column is a transfer function $H_{k,l}$ relating the k th output signal with the l th input signal according to (2.7), when all initial conditions and all input signals except from the l th input are set to zero. [46]

$$H_{k,l} = \frac{Y_k(s)}{U_l(s)} \quad (2.7)$$

The transfer function matrix can also be obtained from the state space model through (2.8) [46, 47], where \mathbf{I} is the identity matrix.

$$\mathbf{H}(s) = \mathbf{C} \cdot (s\mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B} + \mathbf{D} \quad (2.8)$$

Unless there is a pair of pole and zero in the transfer function that cancel each other out, the poles are equivalent to the eigenvalues of the \mathbf{A} -matrix. In real systems an exact cancellation of such a pole-zero pair is rather seldom [48]. Therefore, the terms poles and eigenvalues are often used as synonyms.

2.5. Small-Signal Analysis

Stability analyses can be divided into transient and small-signal analyses. The term small-signal analysis refers to investigations that are limited to small deviations from

the examined operating point. For small-signal analyses it is common to use linearization around an equilibrium operating point to approximate non-linear system behaviour.

To create a model for power system analyses it is needed to make suitable assumptions on unknown variables or external influences. The model needs to be suitable for the investigated problem in terms of detail and time scale. [40] The equations used to describe the system behaviour are usually non-linear, so that as mentioned above it is common to make a linearization for the operating point of interest. [34] The two most important methods are state-space and impedance-based modelling. [34, 32].

The poles of the transfer function and eigenvalues of the A-matrix of the state-space model characterize the dynamic behaviour and the stability of the LTI-system. [34]

The system is:

- stable in the sense of Lyapunov, if all eigenvalues have non-negative real parts and all imaginary eigenvalues are non-repeated
- asymptotically stable, if all eigenvalues have a positive real part
- bounded-input bounded-output (BIBO)-stable, if all poles have a positive real part. [48]

While BIBO-stability is concerned with input-output stability assuming zero initial state, asymptotically stable and Lyapunov stable refer to the behaviour at zero input to the system. If there is no cancellation of pole-zero pairs in the transfer function, the poles are identical to the eigenvalues of the A-matrix and the properties asymptotically stable and BIBO-stable are equivalent. As stated before it is not to be expected that in real systems there is an exact pole-zero-cancellation, so that BIBO-stable and asymptotically stable are the same. The formulation of the three stability definitions Lyapunov, asymptotically and BIBO stable and the criteria for LTI-systems stated above can be found in [48].

In [34] the relation between the eigenvalues and the system stability is explained using a more figurative approach. It is shown that each eigenvalue λ_i of \mathbf{A} is associated with a decoupled contribution to the time response, which can be written for each of the n states as (2.9) if the inputs are zero. The factors k_i are constants.

$$\Delta \dot{x}_i = \sum_{l=1}^n k_l e^{\lambda_l t} = \sum_{l=1}^n k_l (e^{\Re(\lambda_l)t} \cdot e^{j\Im(\lambda_l)t}) = \sum_{l=1}^n k_l (e^{\frac{-t}{\tau}} e^{j\omega t}) \quad (2.9)$$

It can be seen that the imaginary part of an eigenvalue gives the frequency of the oscillation. The real part is the time constant and thus responsible for the damping. If the real part of an eigenvalue is negative, the corresponding contribution to the time response will abate. Therefore, if the real parts of all eigenvalues are negative, the system behavior is considered stable.

If the system is not very large the eigenvalues can be calculated directly from the A-matrix. In [38] it is shown how the formulation and linearization of a state-space model for small-signal analysis can be done for a single-machine and multi-machine systems. In [34] it is described how linearized models can be obtained for DC grids. For very large systems the eigenvalues cannot be calculated from the state-space matrix but special algorithms have to be applied. [38]

3. Model Identification Techniques

3.1. General Remarks

The aim of model identification techniques is to find a mathematical model that describes the system behaviour. The identification is based on measurements. Numerous different approaches for model identification exist. The methods presented in the following are examples based on input output measurements and return a LTI state-space model. All of the methods below are applicable with MIMO-systems.

3.2. Vector Fitting

Vector fitting [49, 50, 51] is a method to identify a state-space model from frequency response data. It was originally applied to assess transients in transmission lines and cables [52] and became popular also for other fields of engineering [53]. In [35] it was applied in a power system stability analysis method. In [36] for example, it was applied on power electronic converters.

The basic principle of the vector fitting method according to [49] is to approximate the frequency response data by a rational function which is in the end converted into a state-space representation. The function to fit the transfer function $f(s)$ has the form of (3.1) [54].

$$f(s) \approx \sum_{n=1}^N \frac{c_n}{s - a_n} + d + sh \quad (3.1)$$

a_n are the poles, c_n the residues, s the complex frequency and d and h are constants. This problem is non-linear as the poles in the denominator are unknown. A set of starting poles is defined and the poles and residues are calculated in a two-step process.

First, an augmented function is formulated. The starting poles \bar{a}_n are inserted into (3.1) and A function $\sigma(s)$ is defined that gives the ratio between the input value $f(s)$ to the value of (3.1) with the starting poles. The numeric value of $\sigma(s)$ is unknown, but (3.2) is a linear problem which can be solved by a least squares approach if enough data points for $f(s)$ are available.

$$\sum_{n=1}^N \frac{c_n}{s - \bar{a}_n} + d + sh \approx \underbrace{\left(\sum_{n=1}^N \frac{\tilde{c}_n}{s - \bar{a}_n} + \tilde{d} \right)}_{\sigma(s)} f(s) \quad (3.2)$$

It is shown in [49] that the zeros of $\sigma(s)$ can be calculated and are the poles in the original problem. The poles are inserted into (3.1) and the process can be repeated to relocate the poles iteratively. Then the least squares problem is solved again to obtain the residues. A recommendation for the selection of the initial poles is made in [49].

Several adaptations of this basic approach exist. In the original version, \tilde{d} is constant at one. In the relaxed vector fitting [54] it is chosen to be variable. The fast vector fitting [51] makes use of a QR-decomposition for solving the least squares problem.

In the least squares problem it is possible to weight each data point in the transfer function individually. In [54] a weighting with the inverse of the absolute value for each value in the transfer function is recommended. Values with small magnitudes are treated then equally important as values with high magnitudes and inverse weighing therefore leads to better accuracy at smaller amplitudes. [54] Depending on the goal of the identification in might be better to use other weighting methods.

The vector fitting method can easily be applied to multi-output systems by simply using a vector for the transfer function and the same set of poles for all transfer functions. [49] For MIMO-systems there are multiple options. In matrix fitting [55] all columns of the matrix are stacked and passed to the vector fitting as a single column so that all elements are fitted at once with a common set of poles. A new approach that is currently developed and soon to be published [56] is to fit each column separately and find a common set of poles. The new method automatically determines the relevant system order.

3.3. N4SID

The acronym N4SID is short for "numerical algorithm for state space subspace space system identification" and is actually an umbrella term for a large number of algorithms. [57] N4SID belongs to the family of subspace algorithms. The model that has to be identified is formulated as a state-space model that contains a deterministic and a stochastic part. The stochastic part is the noise. [58, 59] The algorithms then start with the calculation of a sub-space and an extended observability matrix from the input-output measurements. In a second step the state-space matrices are calculated. [58] A comparison of subspace algorithms is given in [58]. In [59] another overview of subspace methods and a generalized procedure is given.

When in the following N4SID is mentioned, the term refers to the in MATLAB function `n4sid` [60].

3.4. PEM

Prediction error minimization (PEM) is a method that uses optimization to reduce the error between the output of the model and the measurement data. For the algorithm the mathematical structure of the model that has to be known so that the parameters such as the matrix elements in the state-space model for example, can be optimized. The problem is defined in the form of a cost function that includes the error between the model and the predicted output. The theory can be found in [61]. In practice it is common to apply PEM in combination with another identification technique to first identify a model and then refine it with PEM. When PEM is mentioned in the following, it refers to the MATLAB function `pem` [62].

3.5. Application of Identification Methods in this Thesis

The method referred to as vector fitting in this thesis is the new approach that fits the columns of the transfer function separately and then finds a common set of poles. It is soon to be published in [56] and discussed there in detail. All functions applied for the vector fitting in this thesis were provided by SINTEF Energi and are partially available online [53]. As a weighting scheme inverse square root weighting was used.

As a second technique the Matlab function `n4sid` is applied. In the swipes with identification of the analytical model it was sufficient to apply `n4sid` only. In the laboratory context this did especially at first not deliver satisfactory results, why it was combined with `pem`. With better quality of input data also the results of `n4sid` became more acceptable but the combination with `pem` was maintained. Obviously, it would not be a fair comparison between vector fitting and N4SID if PEM is only applied on N4SID but since the extent of this thesis is too limited to make a thorough statement on which method is preferable for the application with laboratory data anyways, it is still decided to do it like that.

The function `n4sid` needs the system order as an input. In this thesis the number of states was set to the number of states that have influence on the input output behaviour in the analytical model, which is 18. In reality it can be expected that the system order for a black-boxed system is not known. How the system order can be determined using Hankel singular values and how system order reduction can be applied is discussed in literature and the Matlab documentation for the function `balred`. In this thesis, the focus is not put on system order estimation and it is assumed that a suitable system order could have been determined by one of the established methods.

Unstable poles were allowed for the sweeps to create the input data for the fitting. For the identification of the laboratory system stability was enforced. More details of the application of the identification techniques relevant to the identification of the laboratory system can be found in subsection 5.8.

4. Fitting of an Operating Point Dependent Model

4.1. Overview

In the first part of the work the focus is put on the operating point dependency of the linearized state-space model. The general idea of the fitting has been roughly outlined in subsection 1.3 and is illustrated in Figure 1.1. Input sweeps are performed and the matrix elements of the linearized state-space model are fitted in dependency of the input values. Before a fitting of the matrix elements is possible, the state space model first has to be brought to a consistent form, which is arguably the most important step and discussed in subsection 4.4. The outcome of the fitting is a linear small-signal state space model with matrix elements that are functions of the input.

The generalized process to get the input dependent state-space model is the following:

1. Define a reasonable set of investigated operating points.
2. Get the state-space model for all investigated operating points.
3. Bring all state-space models in a uniform system representation.
4. Investigate the dependencies of the matrix elements from the operating point and fit each matrix element with a suitable function of the inputs.
5. Validate the fit qualities in the frequency and/or time domain.

Each part of the process is discussed in detail in later subsections. The process of formatting and fitting the state-space models is displayed in the flowchart in Figure 4.3.

At first, only one input is varied at once. Once the fits for the variation of each input individually are created, the inputs are combined and varied pairwise to discuss interdependencies between the inputs. As it is not reasonable to calculate all possible permutations of input values and fit them, an attempt is made to approximate the functions by the superposition of the individual input dependencies gained in the previous step. The resulting model is validated for different operating points in the frequency and time domain.

The code for the formatting and fitting process as well as functions for calculating numerical state-space systems from the fits and for the superposition of fitted models are provided in Appendix J.

The investigations are carried out on the linearized analytical state-space model documented in [34]. The state-space matrices are obtained by solving the nonlinear

steady-state equations and then inserting the steady-state solution into the analytical equations of the model as described in [63] for each investigated operating point. To test the use of the method with identification results, the identification methods are applied to frequency response data calculated from the linearized analytical model. In future work the method should also be tested with identification results from simulation or measurements.

4.2. System Definition and Delimitation

The scheme of the system setup can be found in Figure 4.1. It consists of a converter that is connected to the grid over a filter inductance and capacitance. The control system receives reference values for active and reactive power, grid voltage amplitude and grid frequency as well as measurements of converter current, output voltage and grid frequency. It generates the reference phase voltages for the converter.

The converter is controlled as a VSM. The control scheme used in this thesis is a current controlled VSM (CCVSM) which is in general similar to the structure presented in [64]. In the variant used in this thesis the damping in the swing equation is implemented with a PLL as presented in [44]. The scheme includes active and reactive power droop controllers and a block that accounts for active damping.

An averaged converter model is used so that modulation and switching effects are not covered. The converter acts as an ideal controllable voltage source and the DC side behaviour is neglected. The converter output voltage equals the reference converter voltage.

The analytical model used in this thesis has the reference values for the control system, the grid frequency and grid voltage in dq-components as inputs. Its outputs are the converter current, output current and output voltage in dq-components. A black boxed version of this system could look like Figure 4.2. In general it depends on the specific use case which signals and measurements are black-boxed and which are available to the outside. While in some cases only the electrical measurements might be available, in other cases a list of shared internal signals could be agreed with the vendor to deliver additional information.

To summarize, the inputs of the system and respectively the state-space model are:

- the active power reference p^*
- the reactive power reference q^*
- the grid voltage in dq-components $v_{g,d}$, $v_{g,q}$
- the voltage reference v^* for the voltage at the filter capacitor

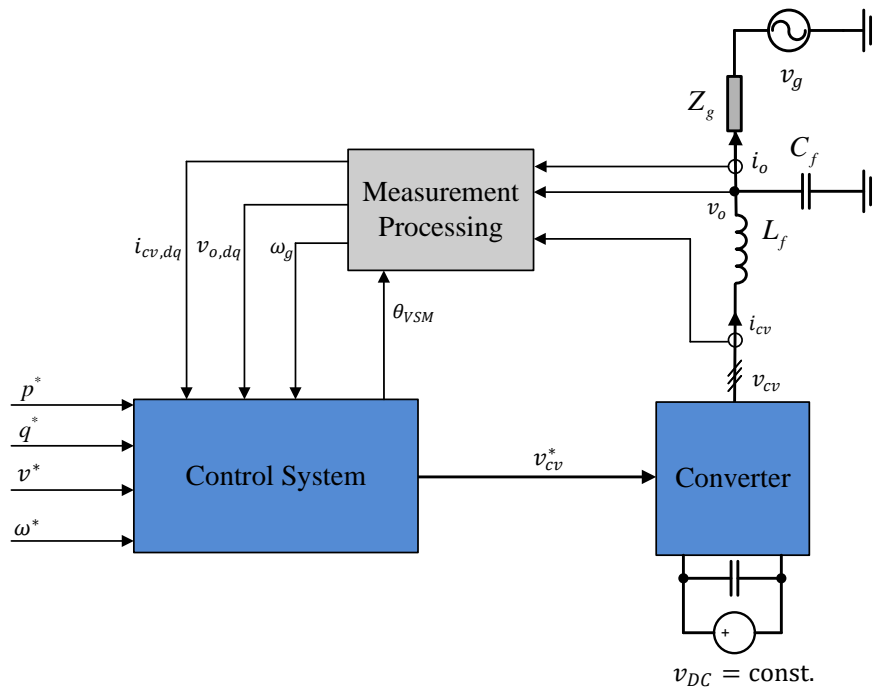


Figure 4.1: System setup

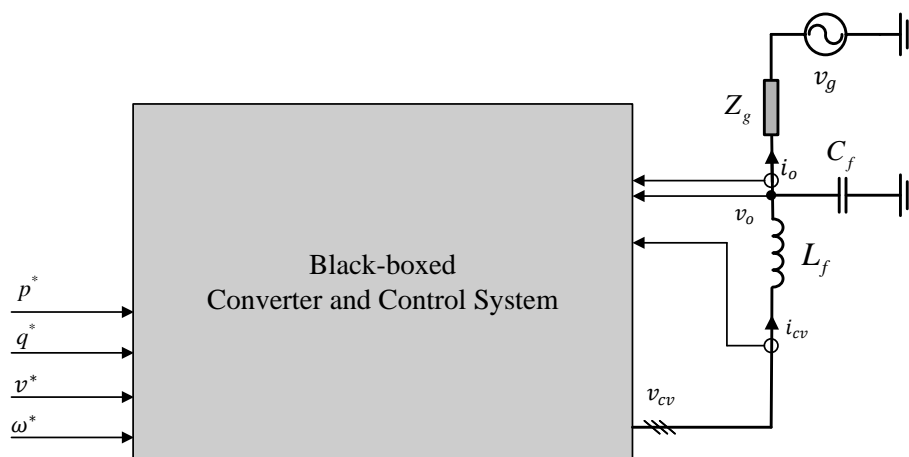


Figure 4.2: System setup black-boxed

- the grid frequency ω_g
- the frequency reference ω^*

The outputs of the system are:

- the output current in dq-components $i_{o,d}, i_{o,q}$
- the converter current in dq-components $i_{cv,d}, i_{cv,q}$
- the output voltage in dq-components $v_{o,d}, v_{o,q}$

This results in the input vector \mathbf{u} and output vectors \mathbf{y} according to (4.1).

$$\begin{aligned}\mathbf{u} &= [p^*, q^*, v_{g,d}, v_{g,q}, v^*, \omega_g, \omega^*]^T \\ \mathbf{y} &= [i_{o,d}, i_{o,q}, i_{cv,d}, i_{cv,q}, v_{o,d}, v_{o,q}]^T\end{aligned}\tag{4.1}$$

4.3. Selection of Operating Points

In the steady-state the operating point is defined by the system input. The states are a result of the input, so that for the investigation of the output behaviour depending on different operating points only the inputs will be manipulated. In fact, in real black-boxed models the states are in general not accessible anyways.

At first, the influence of each input will be investigated separately. For each signal a range of values will be defined and the vector fitting will be done for k_1 equidistant values inside this range. Table 4.1 gives the minima and maxima of the considered values for each input. During the swiping, all other inputs are kept constant at the default value that can be found in Table 4.1 as well.

Table 4.1: Considered input value ranges

input	default	min	max	comment
p^*	0.5	0	1	
q^*	0	-1	1	
$v_{g,d}$	1	0.9	1.1	equals $\pm 10\%$, see IEC 60038 [65]
$v_{g,q}$	0	-0.1	0.1	
v^*	1	0.9	1.1	
ω^*	1	0.95	1.04	equals 47.5 Hz to 52.0 Hz
ω_g	1	0.95	1.04	equals 47.5 Hz to 52.0 Hz, see [66]

As in reality multiple inputs can change at once, the combination of input changes has to be considered as well. A full enumeration with k set points for each of the n_i inputs would lead to k^{n_i} combinations and a high computational effort due to the long simulation times. Moreover the dependencies will be difficult to identify

with this high number of variables. The variation of many inputs at once is thus not reasonable for these investigations. Therefore, to identify inter-dependencies the inputs are varied pairwise. For the pairwise combined fitting and validation k_2 values of one input in the pair are combined with all k_2 values of the other input. Only for the validation all inputs are combined. All possible combinations of k_{all} values per input are examined there. Depending on the use case the selection of the operating points used for fitting and validation should be adapted.

4.4. System Representation

4.4.1. Selection of System Representation

As described in subsection 2.4.2, the state space representation is not unique, but to fit the matrix elements of the state space model a consistent representation for all results is needed. This is for sure the most important step in this first part of the Thesis.

There are various canonical forms known for state space models. Here a diagonal form, also called modal representation [67], was chosen. This representation has the benefit that the states are decoupled and the A-matrix has a simple structure with few elements. Also, the eigenvalues can be directly obtained from the **A**-matrix. However, the modal representation is not canonical, as the state order is not fixed and there is one degree of freedom per state with that the corresponding rows of B and columns of C are scaled (see also subsection D.3). Therefore additional conventions have to be introduced which are discussed in the following. As the examined systems have complex poles, the diagonal form will be complex. For the fitting a representation with A as a real block diagonal matrix is chosen. Also, the Simulink state-space model block only accepts real-valued matrices. The steps needed to obtain a consistent representation throughout a input sweep are discussed in the following.

4.4.2. Transformation into Complex Diagonal Form

At first, the state space model of every step is transformed into a complex diagonal form. The transformation into diagonal and block diagonal form as well as the degrees of freedom are discussed in the appendix.

In this complex diagonal form, the states are in no specific order and the factors of B and C are not consistent. However, the states are decoupled and can each be associated with one diagonal element of A, which is the corresponding eigenvalue,

one row of B and one column of C . This is convenient for the sorting and filtering done in the following.

4.4.3. Sorting and Clustering of Eigenvalues

To make sure that the diagonal representation becomes consistent, it is crucial to sort the states. Each state is characterized by the corresponding eigenvalue, therefore one option to sort the states is to sort them based on their eigenvalues.

Under the use of identification techniques it is not given that the number of correctly identified poles is consistent throughout a sweep. Depending on the method used it is not even given that the identified system always has the same order. Also, there might be outliers that are only part of an identified model in one or few operating points. Therefore it is necessary to match the identified models of all operating points to a common set of states. An easy clustering method is introduced, where in each step of the sweep the eigenvalues are added to clusters of the eigenvalues of previous sweeps depending on their proximity to the existing clusters. If a eigenvalue does not fit to any of the existing clusters, it forms a new cluster. This leads to exactly one eigenvalue per operating point per cluster.

The clustering algorithm most likely needs to be adapted depending on the identification technique and the quality of the input data. Various different clustering algorithms can be found in literature. The selection of the method and the tuning of possible parameters should take the properties of the input data such as the scattering of the data, as well as peculiarities of the identification method such as the possibility for repeated poles or different system orders per operating point into account.

Additionally, as the eigenvalues move, their trajectories might cross during the sweep, so that a simple clustering might not be sufficient but also the direction of their movement should be considered. As this does not occur for the investigated model and the sweeps in this thesis, this is not implemented in this work.

Finally the states are given an order. Here, they are sorted by the real part of their corresponding eigenvalue at the default input value. It is important to do this at the default operating point as this is an easy way to ensure compatibility between models from different input sweeps.

For the analytical model, the eigenvalues could already be fitted after this step. The \mathbf{A} -matrix could be formulated from the eigenvalue fits as they are the diagonal elements of \mathbf{A} in diagonal form. For identified models an additional step is necessary to filter out outliers and bad data as described in the next paragraph.

4.4.4. Filtering of States

As mentioned before, the system identification techniques can produce not correctly identified states and outliers. Clusters with less than a minimum number of eigenvalues in them are therefore excluded. The minimum number should be determined depending on the quality of the identification methods. In this thesis it was chosen to be 80% of the number of operating points. The other states are the ones represented in the final state space model.

The steps of the sweep, where not all of these states are identified were of low number and are skipped in the following steps. By doing so, the need to create substitute values or compare systems of different order is avoided. The result is a consistent set of states that are correctly identified in every considered step. After this procedure, the eigenvalues and \mathbf{A} can be fitted also for identified models.

This subsection is not relevant for the analytical model, as there is always the same number of states and the results are exact.

4.4.5. Rearranging the State-Space Model

For every operating point of the sweep the states in the corresponding state-space model are rearranged according to the order of the related eigenvalue in the clustering. This is done by changing the order of the diagonal elements of \mathbf{A} , the rows of \mathbf{B} and the columns of \mathbf{C} accordingly. The states that are associated to excluded clusters are truncated. After this, the systems are transformed into block diagonal to obtain a real system.

4.4.6. Finding the Correct Factors for the B- and C-matrix

At this point, the state-space models are already in real block diagonal form with the correct order of states. However there is one degree of freedom per state which shows up in the corresponding rows of \mathbf{B} and columns of \mathbf{C} . In Appendix D in subsection D.3 this is thoroughly analysed. The rules to get a consistent representation for the B- and C-matrix can be chosen arbitrarily, as long as the result is an adequate trajectory for every matrix element. Here, the following approach was chosen:

- The norm of each column of \mathbf{C} is 1 and
- the first non-zero element of each column has a positive sign.

If \mathbf{c}_i is the i -th column of the C-matrix and c_{1i} is the first non-zero element in this column, the mathematical formulation of the rules for the calculation of the factors k_i is according to (4.2)

$$k_i = \frac{1}{|\mathbf{c}_i|} \cdot \frac{|c_{1i}|}{c_{1i}} \quad (4.2)$$

It is possible that in the desired output the trajectory of the first element of a column would cross the zero line in the course of the sweep. As the second rule does not allow this, discontinuities appear. Therefore, after the application of the rules above a comparison with the previous C matrix is done in each step. As zero-crossings were not observed for many elements, the following rule was added:

- If the majority of non-zero elements in column i changes their sign compared to the previous step, the column is multiplied by -1 and (4.3) applies instead.

$$k_i = -\frac{1}{|\mathbf{c}_i|} \cdot \frac{|c_{1i}|}{c_{1i}} \quad (4.3)$$

With this additional rule, the crossing of the first non-zero element is tolerated and discontinuities in the other elements avoided.

The final state space representation is then obtained by a similarity transformation with the transformation matrix \mathbf{K} in the form of equation (4.4).

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & k_n \end{bmatrix} \quad (4.4)$$

4.5. Fitting

4.5.1. Fitting of the Individual Input Sweeps

By inspecting plots of the matrix elements over the sweeping range it became apparent that for this VSM model a quadratic function might be a suitable fit function. Therefore a quadratic polynomial function of the form of (4.5) was selected. When u_i is the value of the swept input and $u_{i,def}$ its default value, the independent fit variable x is chosen according to (4.6), resulting in (4.7). The fitting was done by the inbuilt MATLAB functions.

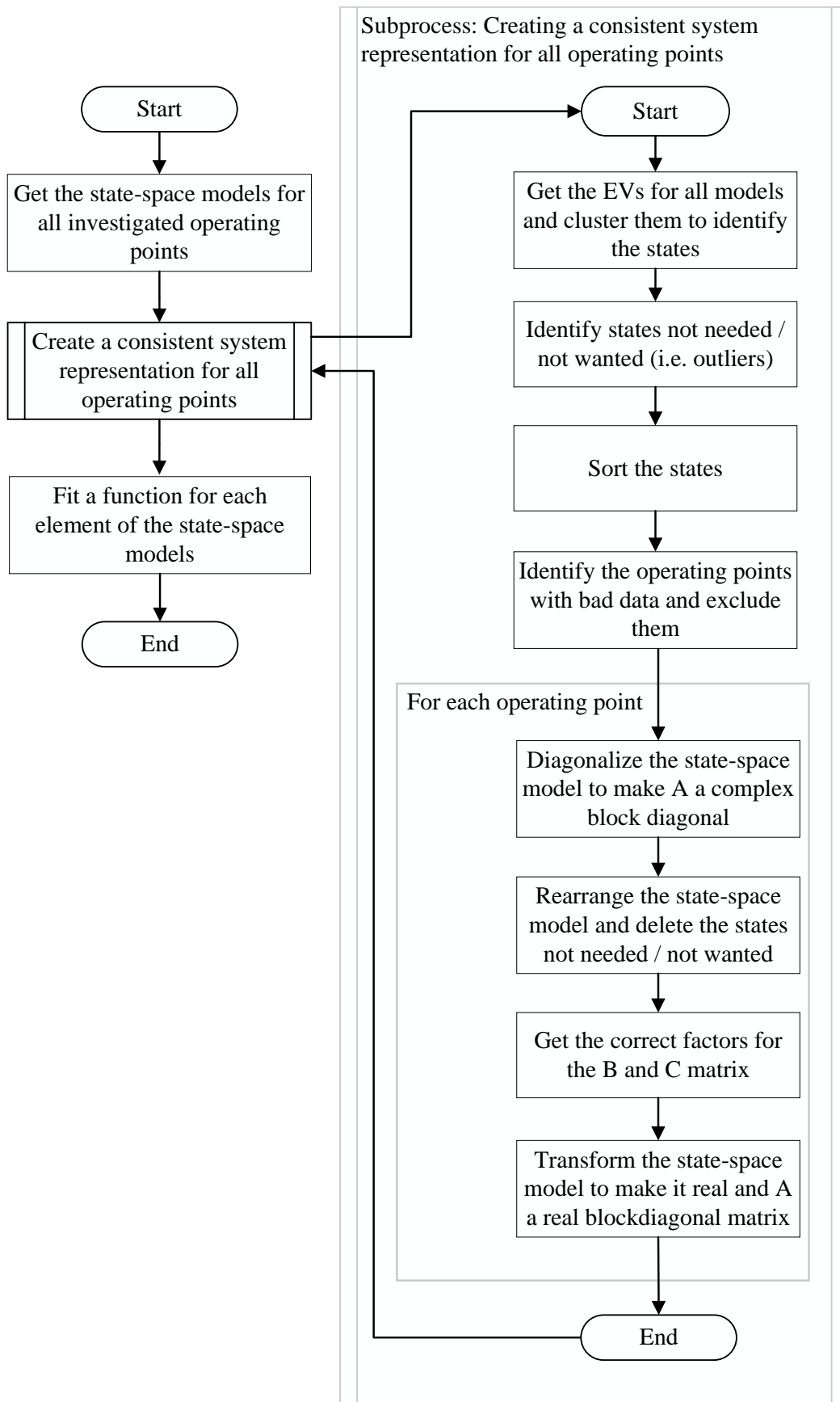


Figure 4.3: Flowchart for input data preparation, creation of a consistent system representation and fitting

$$f_x(x) = p_0 + p_1x + p_2x^2 \quad (4.5)$$

$$x = u_i - u_{i,def} \quad (4.6)$$

$$f_{u_i}(u_i) = p_0 + p_1(u_i - u_{i,def}) + p_2(u_i - u_{i,def})^2 \quad (4.7)$$

(4.7) has the form of a Taylor series expansion of degree 2. Even though this function seems suitable for the model analysed in this thesis, other configurations might need a fit function that also include polynomial terms of higher order, root or rational terms or a combination of all. The fit function would have to be adjusted accordingly.

4.5.2. Fitting of Pairwise Combined Input Sweeps

The fit function for pairwise combined input sweeps should include all terms that can result from a multiplication of two single input sweeps. Therefore the fit function in (4.8) was chosen for the given model. The independent fit variable y is chosen analogously to (4.6) with the second swept input instead resulting in (4.9).

$$f_{xy}(x, y) = \sum_{k=0}^2 \sum_{l=0}^2 p_{kl}x^k y^l = p_{00} + p_{01}y + p_{02}y^2 + p_{10}x + p_{11}xy + p_{12}xy^2 \quad (4.8)$$

$$+ p_{20}x^2 + p_{21}x^2y + p_{22}x^2y^2$$

$$f_{u_i u_j}(u_i, u_j) = \sum_{k=0}^2 \sum_{l=0}^2 p_{kl}(u_i - u_{i,def})^k (u_j - u_{j,def})^l \quad (4.9)$$

In comparison to a Taylor series expansion of degree two for functions with two independent variables, the fit function (4.8) contains the additional mixed terms xy^2 , x^2y and x^2y^2 .

4.5.3. Approximation of Pairwise Combination by Superposition

The combination of two varied inputs u_i and u_j can potentially be approximated by a superposition of the fit functions resulting from the two corresponding single input sweeps. If the single sweep of the input variable u_i was done at $u_j = u_{j,\text{def}}$ and the single sweep of u_j was done with $u_i = u_{i,\text{def}}$ and all other inputs were the same, then the superposition $\tilde{f}_{u_i u_j}(u_i, u_j)$ is calculated by using (4.10).

$$\tilde{f}_{u_i u_j}(u_i, u_j) = f_{u_i}(u_i) + f_{u_j}(u_j) - f_{\text{offset}} \quad (4.10)$$

f_{offset} is the function value at the default input values. In this point the single fits should intersect with $f_{u_i}(u_{i,\text{def}}) = f_{u_j}(u_{j,\text{def}}) = f_{\text{offset}}$.

The single fit functions are chosen to deliver the absolute value for an element and not only the deviation from the default value, which is why in a superposition the default value has to be subtracted from each fit except one to get a meaningful result. With (4.11), (4.10) can be rewritten as (4.12).

$$\Delta f_{u_i}(u_i) = f_{u_i} - f_{\text{offset}} \quad (4.11)$$

$$\tilde{f}_{u_i u_j}(u_i, u_j) = f_{\text{offset}} + \Delta f_{u_i}(u_i) + \Delta f_{u_j}(u_j) \quad (4.12)$$

Δf_{u_i} contains linear and quadratic terms of $u_i - u_{i,\text{def}}$ and Δf_{u_j} contains linear and quadratic terms of $u_j - u_{j,\text{def}}$. In comparison to a Taylor series expansion of degree two for functions with two variables, the mixed term $(u_i - u_{i,\text{def}})(u_j - u_{j,\text{def}})$ is missing. In comparison to the 2D-fit function (4.9) that was applied here to analyse the behaviour, all mixed terms are missing in the superposition.

4.5.4. Approximation of Generalized Model by Superposition

The superposition of all single input sweeps to a model that is dependent of all inputs is analogous to the pairwise combination. The formulation is displayed in (4.13)

$$\tilde{f}(\mathbf{u}) = f_{\text{offset}} + \sum_{i=1}^n \Delta f_{u_i}(u_i) \quad (4.13)$$

4.6. Review of Fit Quality

4.6.1. Overview

In a first step, the goodness of the fit for the matrix elements is analyzed based on the Normalized root mean square error (NRMSE) between the input data for the fit and the values resulting from an evaluation of the fit for each matrix element. This is reasonable as it is the easiest way to check if a fit function is generally suitable to depict the behaviour.

As the NRMSE errors based on the matrix elements are very generic and hard to relate to their impact on the system behaviour it is necessary to analyze the fit quality in the time- and/ or the frequency domain to assess which influence the approximation errors of the single elements in the state-space model have on the system behaviour. Potential difficulties are listed below.

- Which operating points should be investigated? Depending on the examined case the number of relevant operating points might be extremely large. The number of operating point combinations grows exponentially with the number of varied inputs.
- Which metric is used to measure the fit quality in the time or frequency domain? For the examined model there are $6 \cdot 7$ input output combinations for each operating point. In combination with the high number of operating points the generated data needs to be assessed automatized. Therefore a suitable metric has to be found to identify critical cases.
- Which deviations between the approximation and the reference are acceptable and which not? This question needs distinct consideration as this will highly depend on the use case. This cannot be answered here.
- Due to the high computational effort the fitted models are only compared to the analytical model and not to the electrical simulation. However, the analytical model matches the electrical simulation well for small-signals that are relevant here.

Time and frequency domain metrics are defined to identify the most critical cases according to the chosen metric. The selected cases are then discussed individually.

4.6.2. Review of Fit Quality per Matrix Element of the State-Space Model

There are numerous methods to judge the goodness of a fit function for a given set of data. In this process, a normalized root mean square error was chosen. While the

RMSE depends on the absolute value of the fitted data, the normalization makes it possible to judge a great number of fits in different orders of magnitude at once. This is needed as there are many matrix elements to fit, which are located in very different value ranges. During the investigations it proved most convenient to do the normalization with the width of the value range of the reference values.

$$\text{NRMSE}(f, \mathbf{x}, \mathbf{y}) = \frac{\sqrt{\frac{\sum_{i=1}^n (y_i - f(x_i))^2}{n}}}{\left\| \max_{i \in (1,n)} (y_i) - \min_{i \in (1,n)} (y_i) \right\|} \quad (4.14)$$

During the application the following restriction has turned out to be useful: If the denominator, respectively the covered value range, is smaller than 10^{-10} , the data can be declared to be constant. The NRMSE would be very large due to the small denominator even though the fit as a constant value might be suitable. Therefore there is no NRMSE calculated or discussed for these values.

4.6.3. Review of Fit Quality Based on the Frequency Response

To assess the fit quality in the frequency domain over a high number of operating points, the calculation of the metric presented in the following is repeated for equidistant values over the whole considered input range. Ideally, these operating points differ from the ones that were used to generate the input data for the fit and are more fine-meshed.

The frequency response is calculated for each operating point for a set of logarithmically spaced frequencies for the analytical and the fitted model. At first, only a frequency range of 0.1 to 200 Hz with 200 examined frequencies is used. This frequency range covers control loop interactions. In a second step the frequency array is extended to 400 frequencies in the range of 0.1 to 2000 Hz. The splitting into two frequency analyses makes it easier to determine, which phenomena appear only in the lower frequency range and which ones are located in higher frequencies as they appear only in the extended frequency range.

The metric used is the maximum magnitude error of all input output combinations for this operating point. The state-of-the-art of other frequency analysis metrics that could have been applied can be found in [68]. The maximum magnitude errors was chosen, as they are simple to calculate and interpret. It is arguable though that other metrics might contain more or other informative value about the general quality of the frequency response, as the maximum errors only consider the worst case value.

The advantage of the comparison in the frequency domain is that frequencies over the whole frequency range are assessed and equally weighted. The disadvantage of the chosen metric is that the maximum errors might appear at low magnitudes and therefore be less severe than smaller errors at higher magnitudes. Thus, more severe errors might be undetected. As the frequencies are discretized, errors that occur in between stay undetected as well. The bode plots of the operating points and input output combinations that cause the maximum errors should be looked at individually to check the real severity of the inaccuracy.

During the investigations it became apparent that high phase errors are correlated to high magnitude errors. Therefore the maximum phase errors are not discussed separately. Moreover, it turned out to be reasonable to consider only magnitudes and phase errors for values where the magnitude is larger than -60 dB, which equals a factor of 10^{-3} . At this value a change of 10 % in the input, which would already be extremely large for a small signal disturbance, would cause a change of only 0.1 % or 0.0001 pu at the output. Therefore all magnitudes smaller than the -60 dB are declared to be negligible.

4.6.4. Review of fit Quality Based on the Step Response

To assess the fit quality in the frequency domain the step response of the fitted models is evaluated. The amplitudes of the input steps can be found in Table 4.2.

Table 4.2: Step amplitudes for time domain analysis

input	step amplitude
p^*	0.01
q^*	0.01
$v_{g,d}$	0.01
v^*	0.01
ω_g	0.0001
ω^*	0.001

The step response is calculated for 10s in a resolution of $10\ \mu\text{s}$ leading to $N = 1000001$ data points per step response. The time domain error TDE that is used as a metric is calculated for each input output combination and all operating points according to (4.15), where N is the number of data points in the step response, y_{ref} the step response data from the analytical linearized model and y_{fit} from the fit that is analyzed.

$$\text{TDE} = \frac{1}{N} \sum_{i=1}^N (|y_{\text{ref},i} - y_{\text{fit},i}|) \quad (4.15)$$

The TDE can be interpreted as the mean deviation between fit and reference. Again, it is desirable to analyze a high number of operating points to cover the whole operational range.

4.7. Fit Function Analysis

To discuss the nature of the input dependencies of the state space model, the terms of the fit function for each matrix element are analyzed. To do so, the absolute value of each non-constant term is integrated separately over the whole input range. This value I_i is then normalized by the sum of all separate integrals I as displayed in (4.16). The result r_i is a value between 0 and 1 that gives the contribution of the term to the function. If the value is zero, the term has no influence at all. If the value is one, this term is the only one that influences the function value.

$$r_i = \frac{I_i}{I} \quad (4.16)$$

The constant term is neglected as it only represents the offset. If I is smaller than 10^{-10} the function is labeled constant. In this case, the calculation of r_i is omitted.

To illustrate the process, the following example shows how this would look like for an easy quadratic function of the form of (4.17).

$$f(x) = x^2 + 0.5x + 1, x \in [0, 1] \quad (4.17)$$

$$I_1 = \int_0^1 |0.5x| dx = \frac{1}{4}, I_2 = \int_0^1 |x^2| dx = \frac{1}{3}, I = I_1 + I_2 = \frac{7}{12} \quad (4.18)$$

$$r_1 = 0.43, r_2 = 0.57 \quad (4.19)$$

In Figure 4.4 it becomes obvious that the influence of the terms depend on the input range, which is why the integration over the considered input range is chosen.

\mathbf{x}_0 to the current state vector \mathbf{x} . The states that correspond to the initial operating point can still be found in $\Delta\mathbf{x}$. With $\Delta\mathbf{x}' = 0$ follows for the auxiliary output vector $\Delta\mathbf{y}' = 0$. The auxiliary output vector $\Delta\mathbf{y}'$ refers to the current equilibrium point, so to obtain $\Delta\mathbf{y}$ that refers to the initial steady state value, the output of the previous model at the time of transition has to be added. To calculate the output vector \mathbf{y} , the corresponding steady state values for the initial operating point \mathbf{y}_0 have to be added.

It becomes apparent, why the trigger signal is needed: if the operating point, or respectively \mathbf{u}_0 would be updated continuously with $\mathbf{u} = \mathbf{u}_0$, it would follow that $\Delta\mathbf{u} = 0$, $\Delta\mathbf{x} = 0$ and $\Delta\mathbf{y} = 0$ at all times, which is not useful.

This approach has no real practical use case. However, it is used to show the improvement of the model behaviour by adapting the state space model for operation further away from the original operating point.

5. Laboratory Experiments and Identification

5.1. Overview

In the second part, it is shown how a state-space model can be created by applying the model identification techniques N4SID and a new, soon to be published vector fitting method to an estimated transfer function. The transfer function is obtained from laboratory investigations in the National Smart Grid Laboratory. The experiments are carried out in a Hardware-in-the loop setup. A two-level VSC, a grid emulator and a real-time system are used. The transfer functions are determined by applying input perturbations of a set of frequencies and measuring the output perturbations. Problems and limitations of identification based on laboratory data are pointed out. The necessity to generate substitute values to make up for missing data and complement incomplete data sets from laboratory investigations is discussed. The obtained model is validated against additional laboratory measurements.

5.2. Frequency Response Estimation

If the frequency response is not known as it is the case for black-boxed systems, it can be estimated from adequate input output measurements. This can be done with data obtained from simulation or with measurements from laboratory investigations. In both cases, the same basic principle is applied which is discussed in this subsection.

A sequence of sine perturbations is applied to each input separately for a set of 100 logarithmically spaced frequencies between 0.1 Hz and 5 kHz. Each perturbation is ramped up for three seconds and then applied at full magnitude for three seconds before the measurement logging starts. In the simulation the time between the ramp up and the logging was only one second. The ramp and waiting time should be reduced to the necessary minimum for the simulation to decrease the computational effort. In the laboratory where the investigations are carried out in real time, a matter of seconds is not very significant and it is better to wait longer since there might be slower transients present due to the transformer for example. The measurement acquisition lasts three periods or minimum one second with a sampling time of 100 μ s. In the lab, before the application of a perturbation to the next output, the measurement is delayed until the system is detected to be back in steady state before the next output block starts. In the simulation, the system is simply reset and restarted from steady state.

In the simulation the MATLAB function `frestimate` is used which provides the frequency response estimation for Simulink models. The input signals passed to

the function are configured as described above. For the laboratory measurements the Fourier transformation of the measurement for input and output during the perturbation is calculated by using the MATLAB function `fft`. The frequency component in the Fourier spectrum with the highest amplitude is selected and compared to the perturbation frequency as a plausibility check. If the frequency from the Fourier spectrum matches the perturbation frequency with a tolerance of $\pm 20\%$. This makes sure that frequency components in the noise are not falsely chosen as the output perturbation. However, this also means that very small perturbations with amplitudes smaller than the noise amplitude will not be detected even though they might be visible in the measurement curves. In further investigations this calculation of the Fourier transform of the output perturbation should be improved. The frequency response is calculated as ratio of the Fourier transformation of output and input.

This Thesis focuses on the laboratory tests while other work carried out in the Ocean Grid project in parallel to the Thesis work deals with data acquisition in simulation. In the discussion part a comparison of the two approaches is made. The laboratory test are described in detail in the next subsection.

The code for the frequency estimation from the Simulink model and frequency response data for the simulated laboratory setup to accompany the laboratory work were provided by Andrew Smith from SINTEF Energi. The simulation results are not discussed in detail in this work.

5.3. Laboratory Setup

With the laboratory validation it is shown how the system identification can be done based on real measurements. The experiments are carried out in the National Smart Grid Laboratory at NTNU. Photographs of the lab are included in the appendix. During the thesis a grid emulator, a 2-level VSC and a real-time system come into use. The setup is intended for Power-Hardware-in-the-Loop (PHIL) experiments. The real-time system runs at 10 kHz. In each cycle it runs the control algorithm and actuates the converter accordingly. It also controls the grid emulator. The grid emulator has six outputs that can be controlled independently as AC or DC voltage sources, providing both the DC and the three phase AC voltage for the experiments. Between the grid emulator and the converter is a delta star transformer. An overview over the setup can be found in Figure 5.1. A table with all parameters for the laboratory validation can be found in Appendix G.

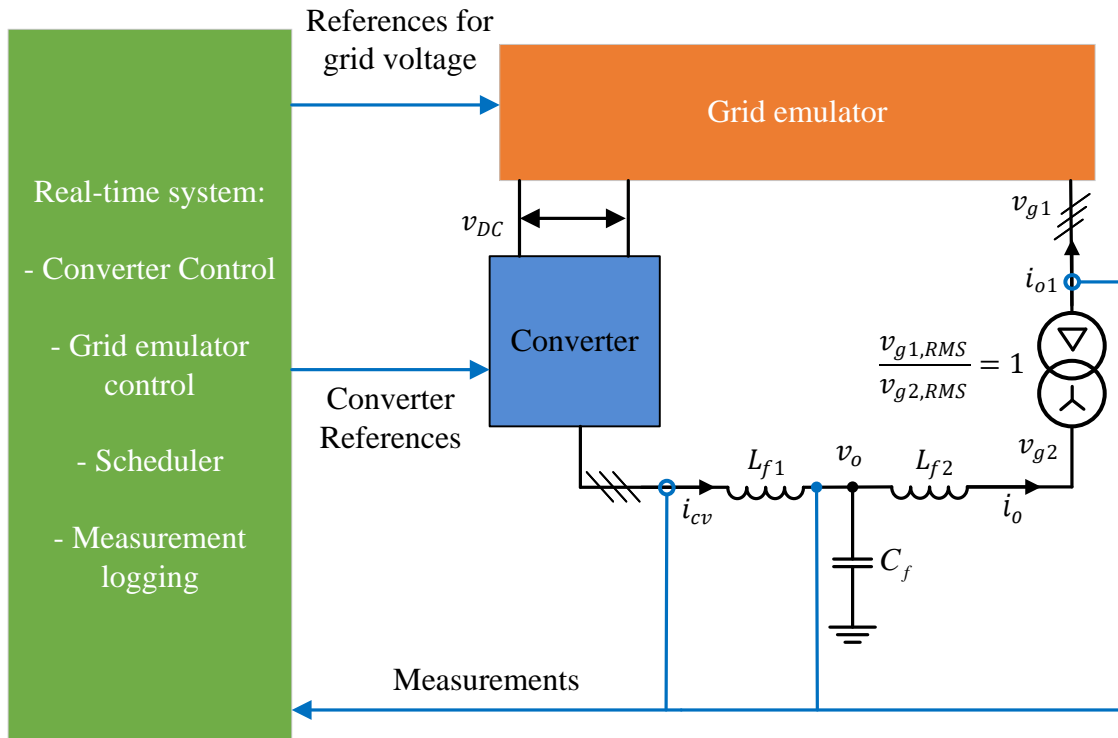


Figure 5.1: Overview laboratory setup

5.4. Data Acquisition for Transfer Function Estimation

The whole measurement procedure for the frequency response estimation as described in the previous subsection is fully automatized. It is controlled by a scheduler that runs on the real-time system which calculates the perturbations and adapts the corresponding signals for the converter and grid emulator references.

The amplitudes for the perturbations were determined by adapting the perturbation according to measurement results. The process was supported by analyses based on the simulation model. The amplitudes for the final measurements presented later on can be found in Table 5.1.

Because of the time limit in the end only one operating point could be examined. The corresponding input values can be found in Table 5.1.

The measurements for perturbation frequencies close to 50 Hz, more specifically between 45 Hz and 55 Hz were omitted. The reason is that the perturbation of 50 Hz in the rotating reference frame creates a DC unbalance between the three phases in the stationary frame. This causes problems due to the delta-winding of the transformer.

Table 5.1: Set points and perturbation amplitudes for laboratory tests

Input	Set point		Perturbation Amplitude	
	SI	p.u.	SI	p.u.
p^*	10 kW	0.2	1 kW	0.02
q^*	0 kvar	0	6 kvar	0.1
$v_{g,d}$	310 V	0.9492	10 V	0.0306
$v_{g,q}$	0 V	0	10 V	0.0306
v^*	310 V	0.9492	8 V	0.0245
ω_g	50 Hz	1	20 mHz	$4 \cdot 10^{-4}$
ω^*	50 Hz	1	200 mHz	$4 \cdot 10^{-3}$

5.5. Adaption of Perturbation Amplitudes

In first tests with constant input perturbation amplitudes for all frequencies it became apparent that the transfer functions for the inputs p^* , ω_g , and ω^* caused observable perturbation at the outputs only for frequencies lower than approximately 9 to 11 Hz. At higher frequencies no change in any of the outputs was visible. The explanation that can be verified by looking at the analytical transfer functions and the measurements that could be obtained in these first test is, that the transfer function magnitude for these inputs decreases continuously at frequencies higher than the mentioned ones. The integrators in the VSM swing equation function as low pass filters. In the averaged simulation model this was not a problem because also very small output perturbations could be measured.

In the laboratory and also in a simulation model with a switching converter model there is noise in the output signal, so that very small signal changes cannot be captured as they cannot be distinguished from noise. In other words, the signal-to-noise-ratio is too low. It could be argued that if the output change is too small, the behaviour is not relevant in this frequency range. However, the transfer function can also at low magnitudes give information about system states, which is why it is desirable to gain information over a larger frequency range even at smaller magnitudes.

In simulations with the switching model it proved useful to simply scale the amplitudes larger than 1 Hz depending on the frequency. For ω_g the amplitude was scaled with the perturbation frequency in Hz and for p^* and ω^* the amplitude was scaled with the perturbation frequency in Hz squared. At high frequencies the input perturbations will become extremely large but as the output perturbation is still small it could nevertheless be argued that this can be treated as small-signal behaviour.

The outputs are usually closer related to the states than the inputs and the states will most likely also only change little if the output perturbation is small.

In the laboratory the scaling of the amplitudes should be done carefully as the system is constrained by physical limits and protection mechanisms. From the analytical model and the simulation it is known that there are no resonance peaks expected in the respective outputs, therefore the scaling with the frequency could be done in a similar relatively simple way. However, the scaling as done in the simulation was too extreme and the measurements had to be stopped as the perturbation amplitude in the current was too high. The scaling was adapted according to Table 5.2 for the final tests presented later on.

Table 5.2: Frequency dependent factors for the scaling of the perturbation amplitude

Input	factor for scaling	limit	frequency range
p^*	$0.5 \cdot (f - 5 \text{ Hz})^2 + 1$	1000	$f > 5 \text{ Hz}$
$v_{g,q}$	2		$f < 0.7 \text{ Hz}$
	0.5		$2 \text{ Hz} < f < 50 \text{ Hz}$
ω_g	$0.5 \cdot (f - 5 \text{ Hz})^{0.75} + 1$	10	$f > 7 \text{ Hz}$
ω^*	$(f - 5 \text{ Hz})^{1.5} + 1$	100	$f > 5 \text{ Hz}$

In these investigations it was possible to determine the adapted amplitudes in a combination of analysis of the simulation from the transfer function and trial and error in the lab. A more sophisticated approach could have been to make the scheduler incrementally increase the input perturbation dynamically depending on the amplitude in the output perturbation. This could be useful in cases where the transfer function is more complex. If the simulation model is available, the input perturbation could also be scaled with the inverse of the transfer function obtained from the simulation to keep a prioritized output constant. Prioritizing different outputs for different frequency ranges would also be possible. However, with this approach it still has to be kept in mind that the simulation will differ in some extent from the real system. The discussed options were not tested since for this test the simple scaling approach was sufficient.

5.6. Validation by Step Response

To validate the identified models step responses are recorded at the investigated operating point. The inputs are set and the measurements are started manually. The step amplitude for each input can be found in Table 5.3

Table 5.3: Step amplitudes for laboratory test validation

Input	Step amplitude	
	SI	p.u.
p^*	2000 W	0.04
q^*	10 kvar	0.2
$v_{g,d}$	5 V	0.0153
$v_{g,q}$	5 V	0.0153
v^*	5 V	0.0153
ω_g	20 mHz	$4 \cdot 10^{-4}$
ω^*	200 mHz	$4 \cdot 10^{-3}$

5.7. Treatment of Transformer Phase Shift

As the converters in the grid emulator are all connected to the same DC bus, the transformer is needed for a galvanic separation to avoid undesirable effects due to possible loops that could be created. Moreover, the transformer is useful to filter out zero-components. However, it introduces a phase shift of -30° between the grid emulator side and the converter side. In other words, the voltages and currents at the converter side lag the ones on the grid emulator side by 30° . This means, that for comparability with the analytical model, the measurements for the output current i_o taken at the grid emulator side as i_{o1} need to be shifted by additional 30° when transformed into the rotating reference frame of the VSM.

Instead of first shifting the three phase quantities and transforming them to the VSM-dq-frame after, a dq-transformation that includes the extra phase shift can be applied directly. Rotating a quantity by an angle, in this case -30° and transforming it to a reference frame is equivalent to transforming it to the reference frame rotated with the negative angle, in this case 30° . This is displayed in Figure 5.2, where x_{g1} represents the quantities on the grid emulator side and x_{g2} the quantities on the converter side of the transformer.

The input perturbations on the grid are not affected by the phase shift. In the analytical model, the perturbations refer to the rotating reference frame that is aligned with the grid voltage and not to the rotating reference frame of the VSM. In the laboratory, they are applied in the rotating reference frame of the grid emulator side that is oriented on the voltage of the grid emulator side. This has the same effect as applying the same perturbation on a reference frame that is oriented on the grid voltage on the converter side. Both the voltage and the reference frame are shifted in the same manner, namely by 30° . Figure 5.3 displays the alignment of the reference frames. This means that if there is for example a perturbation in

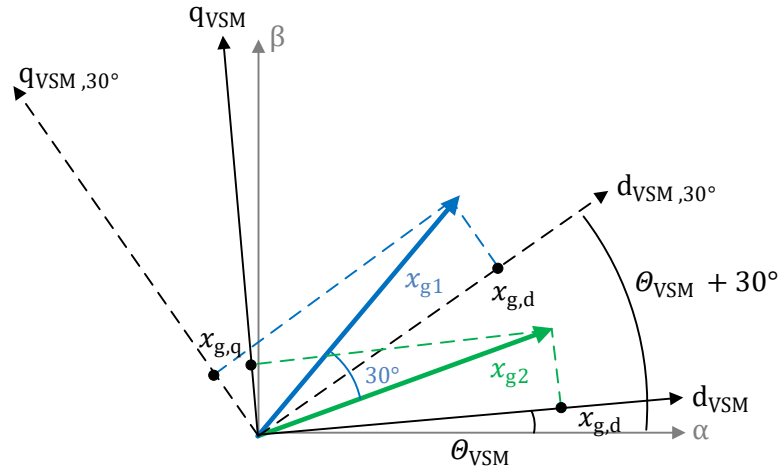


Figure 5.2: Phase shift and transformation into a rotating reference frame vs. direct transformation into a rotating reference frame with adapted angle

the q-axis of the grid emulator system it will also appear in only the q-axis of the converter side reference frame. Applying a perturbation in the dq-system on the grid emulator side is therefore equivalent to applying a perturbation in the dq-system on the converter side of the transformer.

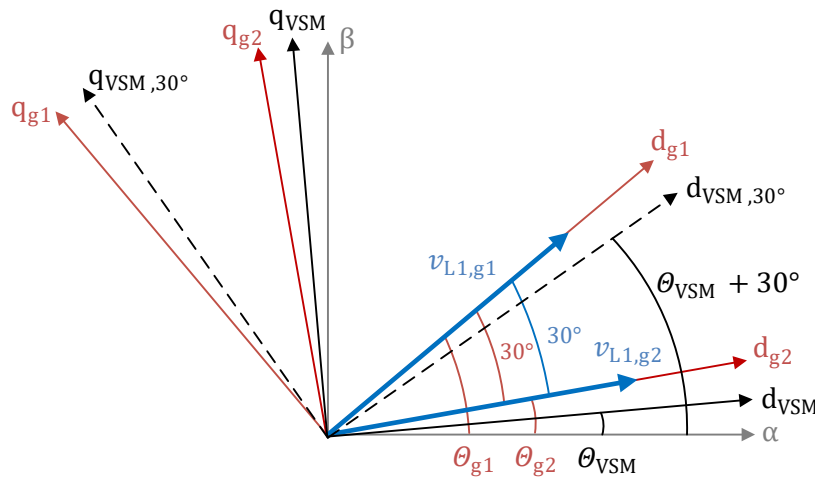


Figure 5.3: Rotating reference frames for the laboratory: grid reference frame on grid emulator and converter side and VSM reference frame

5.8. Identification and Treatment of Missing Data

In subsection 7.2 the data set for the transfer function data measured in the laboratory is discussed in detail. It is pointed out that in real physical systems it is inevitable that the data set is incomplete. There can be missing measurements for certain frequency ranges of an input output combination or missing measurements for a whole input output combination.

For the identification methods it is generally assumed that the data is obtained for a consistent set of frequencies for all input output combinations. In the vector fitting it is possible to simply assign a weight of zero to missing data, either for solitary frequencies of a single input and output combination or for a whole input output combination. It is therefore not necessary to generate substitute values.

The Matlab function `n4sid` provides the option to set a parameter called `weightingFilter` to weight the input data. Amongst others, a weight can be assigned to a certain frequency or a pass band of frequencies. However, as far as the author of this work could tell, the weight is applied to all input output combinations at this frequency. A weighting for individual elements seems not possible in the Matlab function. Therefore it is discussed here how the missing data points can be treated instead.

Setting the missing data to `nan` causes an error in `n4sid` as all response data has to be numeric as far as the author can tell. Setting the missing values to zero might be a suitable option for input output combinations where the output is zero for all frequencies, but for input output combinations where there is only some missing data in between frequencies with sufficient high magnitude, a zero at the missing frequency would be physically not meaningful and could make the identification result significantly worse. A simple interpolation of the transfer function of each input output combination separately with `interp1` for example is an option but a problem is that the interpolated values still have to be meaningful in terms of the system behaviour.

During the course of the investigations it proved most convenient to apply the Matlab function `tfest` to each input output combination where enough data is measured separately and take the frequency response of the transfer function of the identified SISO-model at the desired frequencies as an input for the identification of the complete model. The input output combinations, where no output perturbations could be measured at all are set to zero at all frequencies.

6. Results and Analysis Operating Point Dependently Fitted Model

6.1. Individual Input Sweeps

6.1.1. Eigenvalue Trajectories

In the individual input sweeps, one input is varied while all the others were kept constant at their default value. The trajectories of all individual input sweeps for 31 equidistant values per input are shown in Figure 6.1. Figure 6.2f shows a zoom on the slowest modes. The eigenvalue pair with the imaginary part of about ± 2780 , which is identified as the most important according to the Hankel singular values (see Appendix F) belongs to the eigenvalues that perform the biggest movement throughout the sweeps. This pair is called eigenvalue 3 and 4 in the following.

If during a sweep of $v_{g,q}$ the grid voltage amplitude \hat{v}_g is kept constant by adjusting $v_{g,d}$ accordingly, no significant change in the matrix elements can be observed for this model. The maximum absolute change of the matrix elements is smaller than 10^{-11} for all matrix elements without any regularity. The observed variations are therefore most likely caused by inaccuracies during the calculation and system transformation and seem not related to the sweep of $v_{g,q}$. The impact of $v_{g,q}$ is therefore not further discussed in the following.

Of all inputs ω_g has the biggest influence on the eigenvalues. This can be seen in the eigenvalue trajectories in Figure 6.2e and Figure 6.3a. Mind the x-scale in Figure 6.2e. Also, the only sweep where a real part of an eigenvalue is positive and the system is unstable is the sweep of ω_g . There the real parts of eigenvalue 3 and 4 are positive for small values of ω_g . The stability limit is located between $\omega_g = 0.9968$ and $\omega_g = 0.9977$.

In this model p^* and ω^* as well as v^* and q^* share a trajectory, which is also visible in Figure 6.3b. The explanation is that p^* and ω^* have similar impact on the power-frequency droop controller and q^* and v^* have similar impact on the reactive power voltage droop controller. Those relations cannot be generalized for all control schemes and therefore this should not be used beneficially in the further course of the investigations. However, in practice relations of this kind could possibly be taken into account.

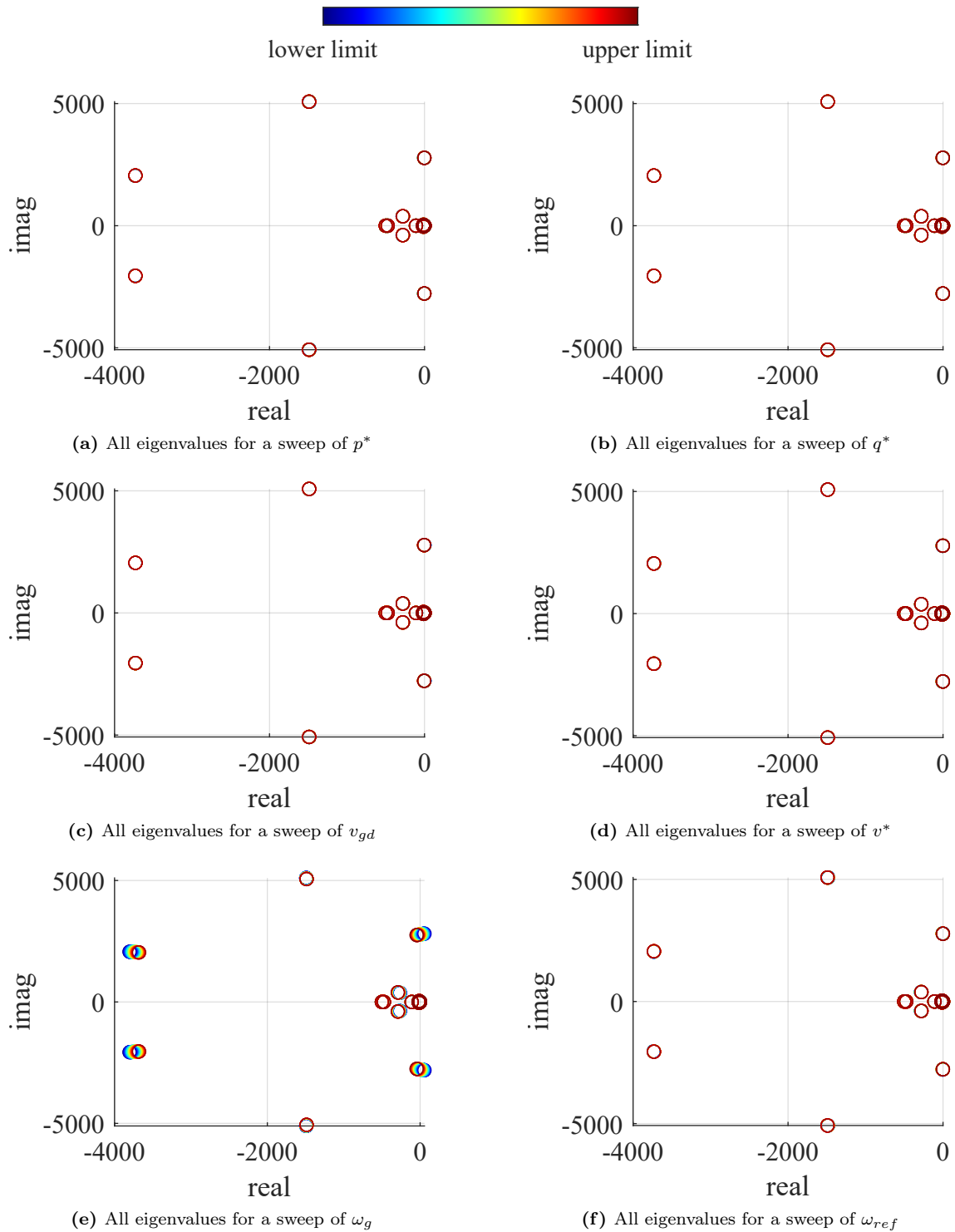


Figure 6.1: Trajectories of all eigenvalues for all individual input sweeps

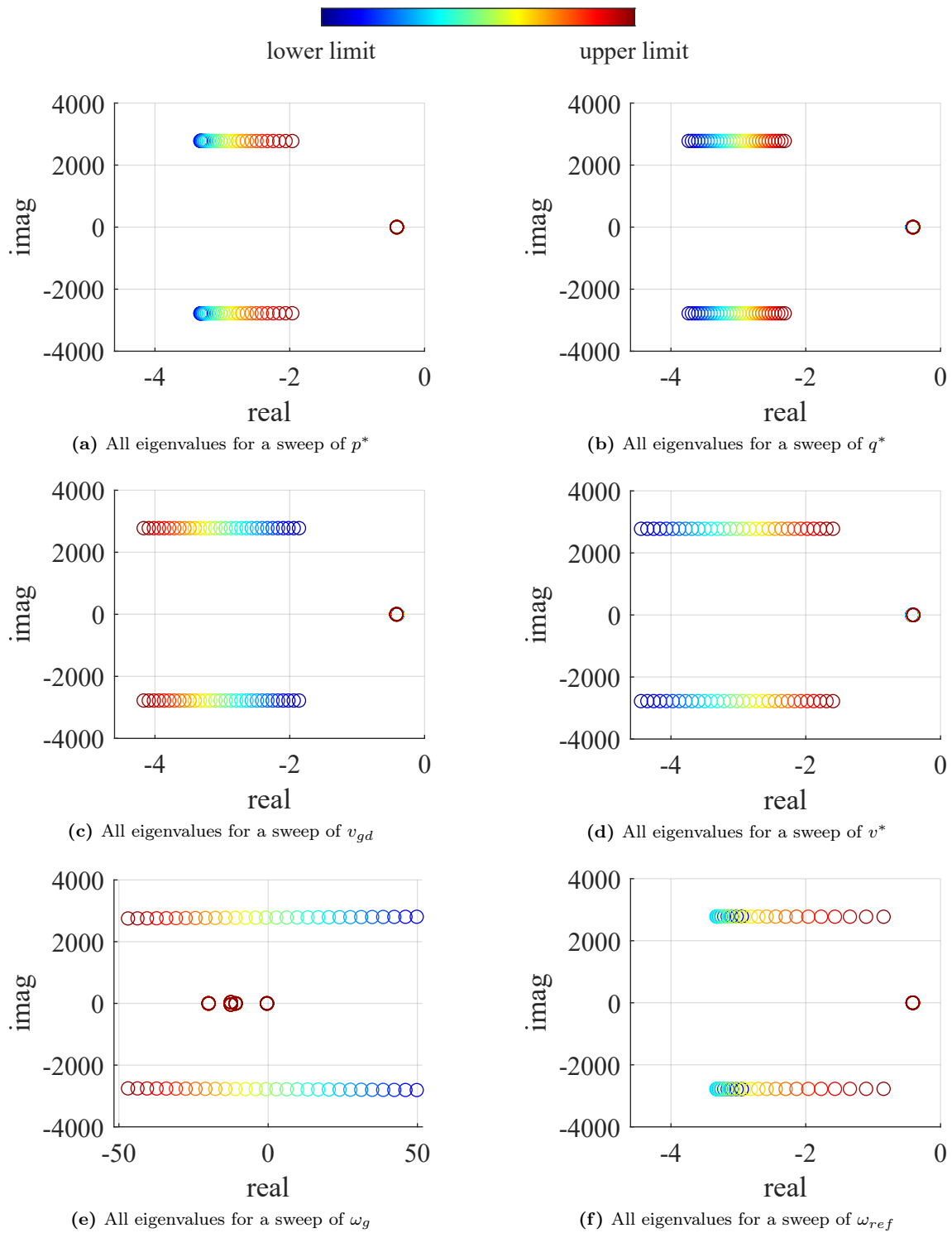


Figure 6.2: Zoom of trajectories of eigenvalues with small imaginary part for all individual input sweeps

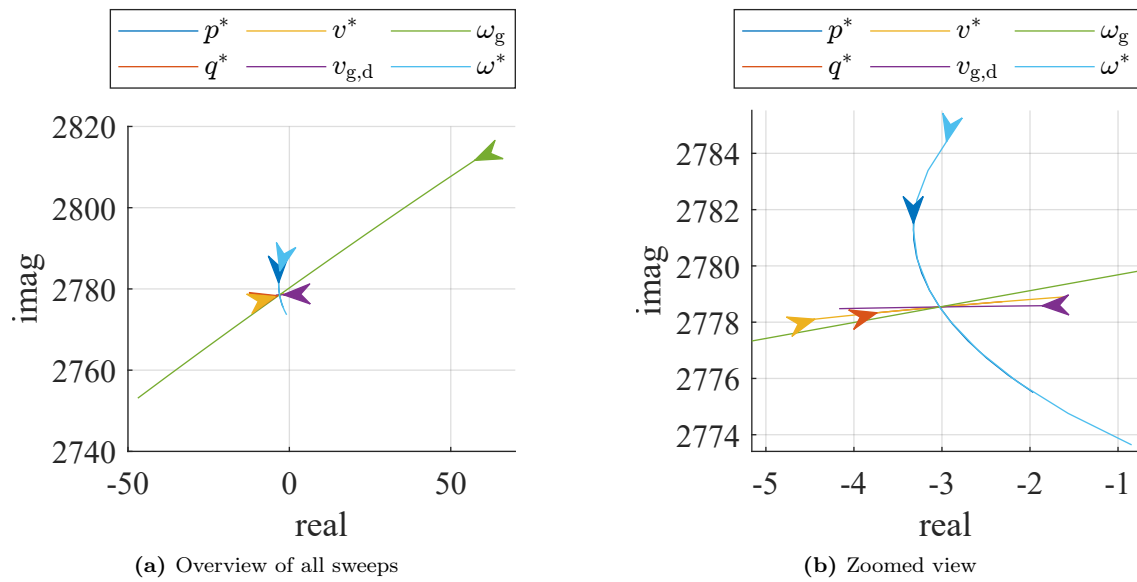


Figure 6.3: Trajectories of eigenvalue 4 for all individual input sweeps

6.1.2. Fitting

Figure 6.4 shows the real and imaginary part of eigenvalue 4 as well as the corresponding elements of B and C for a sweep of p^* as an example. Displayed are the calculated values in the uniform system representation in red and the fit in blue. It can be seen that the fits generally match the calculated values. In this depiction it looks like the elements of C and most elements in B are constant but this is due to the scale on the y-axis. When zooming in on the y-axis it becomes apparent that most elements have a quadratic characteristic. This is discussed in detail in subsection 6.1.5.

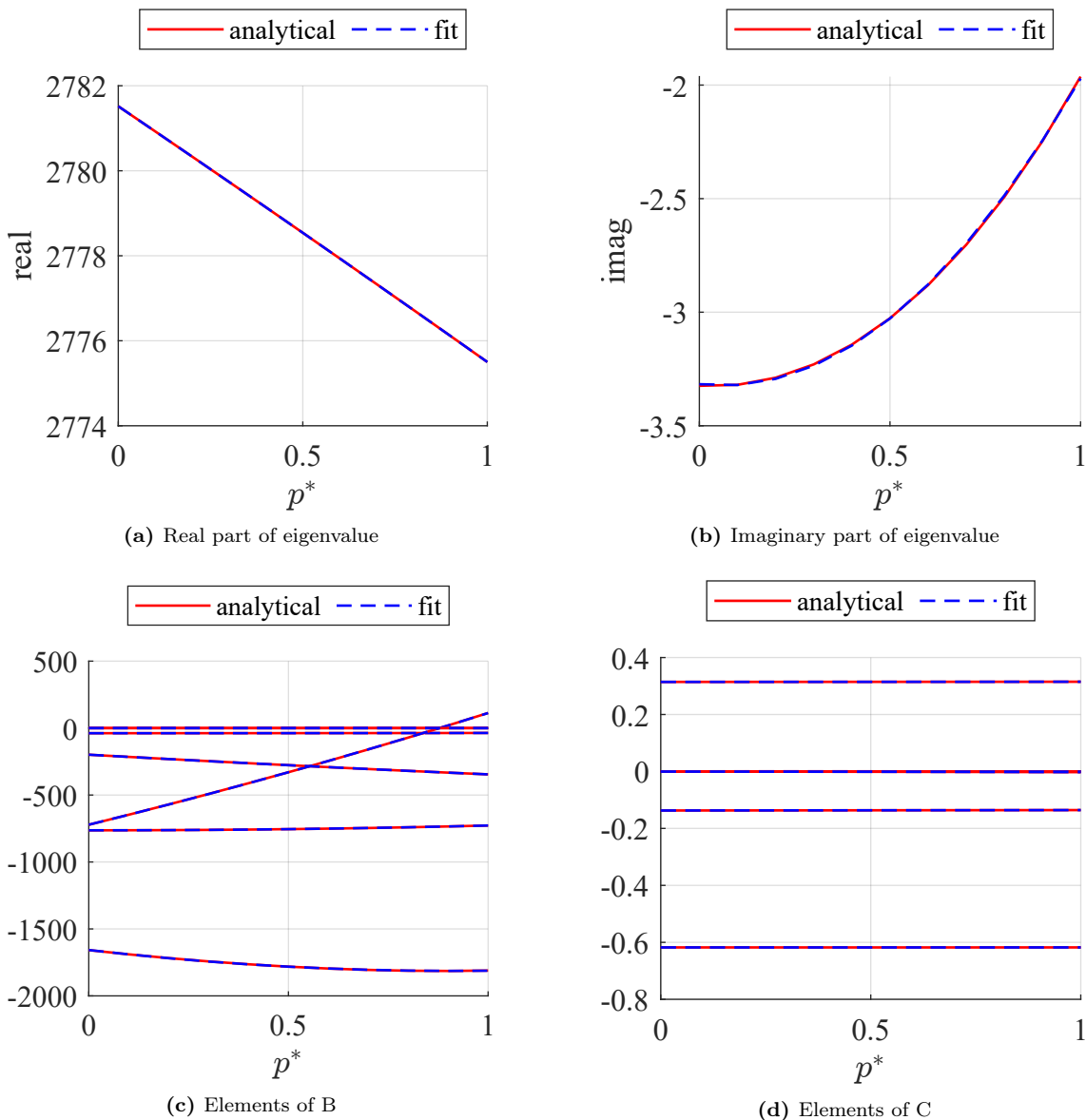


Figure 6.4: Calculated values and fits for real and imaginary part of the eigenvalue and the corresponding elements of B and C associated with state 4 for a sweep of p^*

Table 6.1 shows the maximum and the mean value over all matrix elements for the calculated NRMSEs of the fits for 11 calculated runs per sweep for the individual input sweeps with a quadratic fit function. The results are grouped by matrix.

Table 6.1: Individual input fit accuracy

swiped	NRMSE of fit in %					
	A		B		C	
	mean	max	mean	max	mean	max
p^*	0.24	1.54	0.21	1.47	0.26	1.35
q^*	0.01	0.03	0.02	0.23	0.02	0.23
v_{gd}	0.02	0.06	0.10	5.34	0.02	0.16
v^*	0.04	0.14	0.09	0.81	0.08	0.75
ω^*	0.38	3.75	0.45	5.06	0.43	4.57
ω_g	0.61	3.22	0.58	4.14	0.67	2.99

The values in Table 6.1 show that all elements in all input sweeps can be fitted with a quadratic function with relatively low errors smaller than 5.34%. The maximum NRMSE appears for the sweep of v_{gd} . The mean NRMSE errors for all elements is lower than 0.67%. In general, the maximum values are caused by solitary cases where only one matrix element has a significantly higher NRMSE than all others or are a result of the fact that some fits become more inaccurate close to the limits of the input range. It can be said that the highest NRMSEs usually appear in the B matrix.

In general the quadratic fit function seems suitable for fitting the majority of elements of the given model with small deviations. The higher deviations that occur in solitary cases can possibly be accepted. It has to be checked if the approximation is accurate enough by performing time and/ or frequency domain simulations.

6.1.3. Validation in Frequency Domain

The frequency domain analysis was carried out for 1001 equidistant input values over the full input range of each sweep. At first, 200 logarithmically spaced frequencies in the range from 0.1 Hz to 200 Hz were assessed. In a second step the frequency range was extended to 2000 Hz with 400 logarithmically spaced frequencies.

In Table 6.2 the maximum and the average over all operating points and input output combinations for the maximum magnitude errors are displayed for both frequency ranges. The difference is between the frequency ranges is small. The maximum errors are all located in the lower frequency range.

Table 6.2: Individual fit errors in frequency domain

swept	max. magnitude error in dB			
	200 from 0.1 to 200 Hz		400 from 0.1 to 2000 Hz	
	mean absolute	max	mean absolute	max
p^*	0.04	-9.55	0.05	10.87
q^*	0.00	0.00	0.00	0.00
$v_{g,d}$	0.00	-0.03	0.00	-0.03
v^*	0.00	0.02	0.00	0.02
ω_g	0.31	-31.86	0.33	-32.6
ω^*	0.26	21.77	0.28	-24.78

The mean values for the fit are all lower than 0.33 dB, which indicates that the chosen fit function is suitable for the majority of operating points and input output combinations. It can be seen, that for the fits for q^* , $v_{g,d}$ and v^* also the maximum errors are very small. This means that even in the worst case, the fits resemble the values of the analytical model almost perfectly. In the bode diagram a difference between the two models is not recognizable. As an example, Figure 6.5 shows the bode plot for the fit of $v_{g,d}$ for the operating point and the input-output combination where the maximum errors for the extended frequency range occurred.

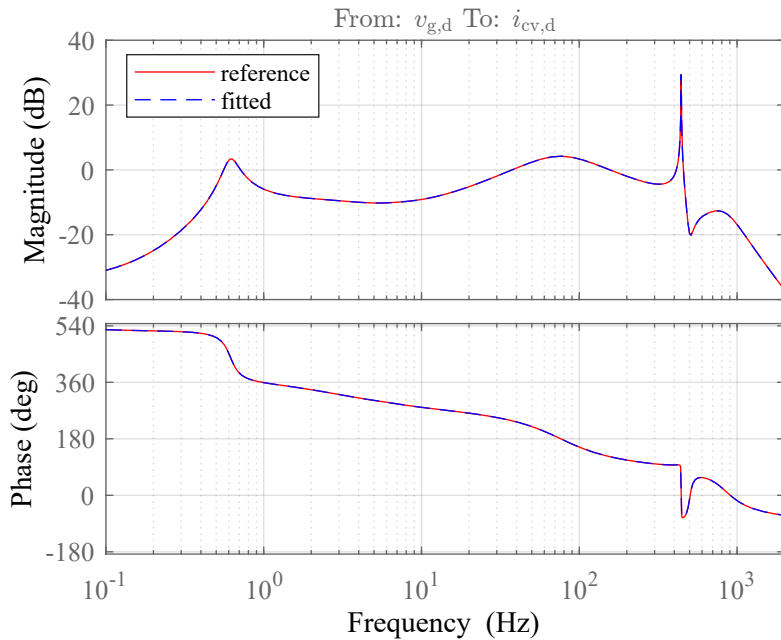


Figure 6.5: Bode plot for $v_{g,d} = 1.1$

For the other inputs p^* , ω_g and ω^* deviations between the fit and the analytical model are clearly visible at the operating points where the maximum errors occurred. Figure 6.6 shows the distribution of the errors for all analyzed input values and all input output combinations for both frequency ranges. The overall picture

of the maximum magnitude errors has not changed with the extension of the frequency range. Solely some smaller errors appear in the extended range that were not detected in the smaller frequency range.

The maximum errors in Table 6.1 all appear in the transfer function between $v_{g,d}$ and $i_{o,d}$ or $i_{cv,d}$. In Figure 6.6 it can be seen that for ω^* high errors occur also in the transfer function from v^* to $v_{o,q}$. In general, the high errors appear only in few input output combinations and only for few input values, which explains why the mean of all maximum magnitude errors is still low in spite of the high maximum.

Moreover, in Figure 6.6 it can be seen that whenever a large positive error occurs, it is followed by a large negative error. This can be explained by examining the magnitude part of the bode plots at the positive and the negative error peaks.

From Figure 6.7 it becomes apparent that the error comes from the peak at 0.6 Hz. This peak is a local phenomenon that in this shape appears only for a narrow value range around $\omega_g \approx 1.016$. At the negative error peak the magnitude dip appears in the fit but not in the reference and vice versa for the positive error peak. This means that the magnitude dip shows up in the fitted model, just slightly shifted in terms of ω_g . The same explanation can be made for the error peaks in p^* and ω^* where similar magnitude dips appear. The frequency of approximately 0.6 Hz equals 3.8 rad/s and the phenomenon is therefore most likely associated with the eigenvalue pair 1 and 2.

For the fit of ω^* the error peaks in the transfer function of v^* to $v_{o,q}$ are caused by a comparable local phenomenon. The corresponding plots are shown in Figure 6.8. The transfer function changes by several dB during a very narrow range of ω^* . However, it can be seen that the fit in Figure 6.8a matches the reference in Figure 6.8b. Again, the errors appear at magnitudes lower than -30 dB and are therefore practically negligible.

Using a polynomial fit function of degree 3 instead of the quadratic fit function reduces the described effects. However, the maximum magnitude error for ω_g will still be at 13.87 dB. For the quadratic fit the large magnitude errors appear all at magnitudes smaller than -20 dB, frequencies smaller than 1 Hz and limited ranges of input values. If this behaviour is tolerable has to be decided for the specific use case but at this point the fits seem to approximate the model behaviour well. The impact on the time domain behaviour is discussed in a later section.

To sum up, the deviations between the fits and the reference are small for the majority of analysed operating points. The high errors that are observed are limited to small magnitudes at low frequencies and appear only for solitary input output

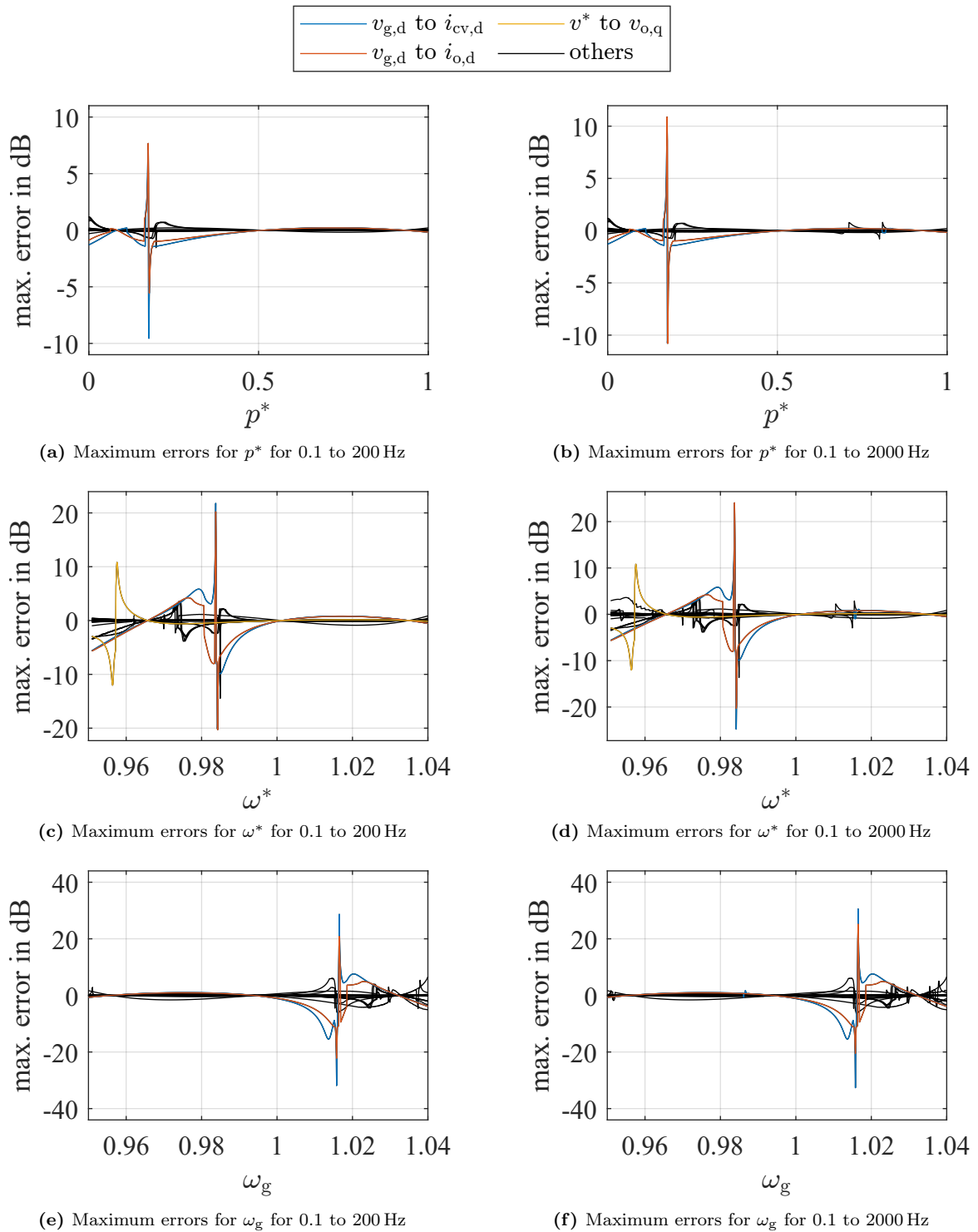


Figure 6.6: Distribution of maximum magnitude errors for the individual sweeps of p^* , ω_g and ω^* for all input output combinations

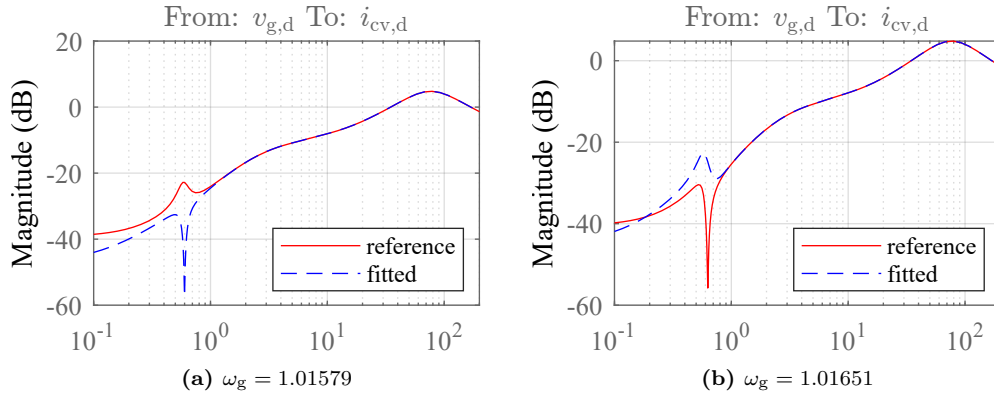


Figure 6.7: Transfer function magnitude for for $\omega_g \approx 1.016$

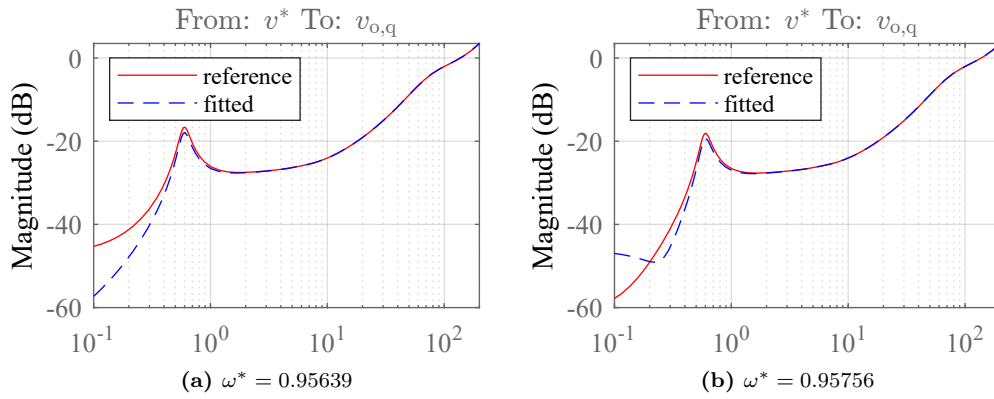


Figure 6.8: Transfer function magnitude for $\omega^* \approx 0.957$

combinations in a limited range of input values. The fit results are satisfactory and the chosen fit function is proclaimed suitable to proceed with. However, one has to be aware of the weaknesses of the individual input fits, especially that locally limited phenomena might not be displayed correctly. It has to be kept in mind that the errors will propagate when the fits are used as a base for further investigations.

6.1.4. Validation in Time Domain

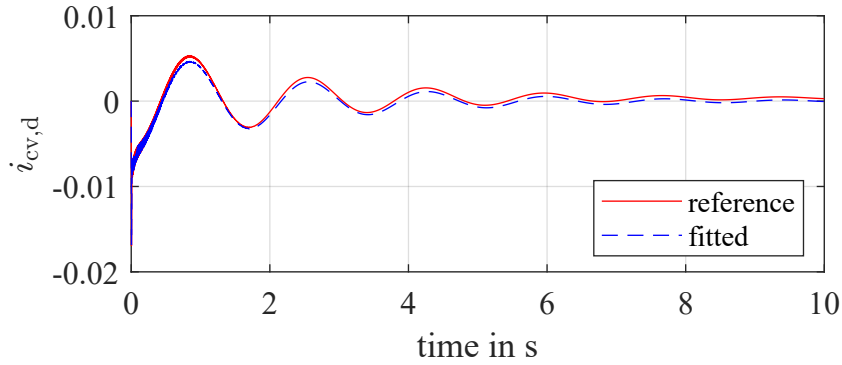
The validation in the time domain was carried out as described in subsection 4.6.4. Because of the higher computational effort, the time domain analysis was carried out for less operating points than the frequency domain analysis. 101 equidistant values per input distributed over the whole input range were examined in this analysis. The mean and maximum TDEs were calculated with the help of (4.15) and are displayed in Table 6.3.

The errors are very small and for p^* , q^* , $v_{g,d}$ and v^* no deviation between fit and reference is visible in the step plot. The plot of the step with the highest TDE is displayed in . This highest error is observed for $\omega^* = 0.9509$, which is the lower limit of the evaluated input range. As in the frequency domain, the highest errors appear in general in $i_{cv,d}$. The input for the highest overall error is for a step in $v_{g,d}$. In it

Table 6.3: Individual fit errors in time domain

swept	TDE	
	mean	max
p^*	$4.1 \cdot 10^{-6}$	$3.2 \cdot 10^{-5}$
q^*	$5.5 \cdot 10^{-8}$	$5.4 \cdot 10^{-7}$
$v_{g,d}$	$2.1 \cdot 10^{-7}$	$1.6 \cdot 10^{-6}$
v^*	$4.5 \cdot 10^{-7}$	$4.5 \cdot 10^{-6}$
ω_g	$2.3 \cdot 10^{-5}$	$1.9 \cdot 10^{-4}$
ω^*	$2.5 \cdot 10^{-5}$	$3.5 \cdot 10^{-4}$

can be seen that the curves do not match but that oscillation frequency and damping are approximated appropriately but that the amplitude is slightly off. The TDE is $3.5 \cdot 10^{-4}$ in this worst case and seems negligible. However, if the approximation is good enough has to be decided depending on the specific use case.

**Figure 6.9:** Step response of $i_{cv,d}$ for a step in $v_{g,d}$ at $\omega^* = 0.9509$

The peaks that appear in the frequency domain errors cannot be observed in the time domain analysis for any of the sweeps, which leads to the conclusion that they have no negative impact on the step response.

To sum up, the errors that are observed in the time domain are small for all sweeps. The problematic operating points that were identified in the frequency domain do not stand out in the step response. The fitted models seem to resemble the system behaviour well in this test. However, in this analysis only the step response was analyzed. Other effects that can appear in the time domain are not covered by this discussion.

6.1.5. Discussion of Fit Functions

To describe the nature of the input dependency, the fit functions for all elements are analyzed as described in subsection 4.7. The results are displayed in Figure 6.10,

where there is one data point per term per matrix element. The color is related to the value on the y-axis. A red dot is associated with a matrix element with high contribution for the term on the x-axis, while a blue dot implies low contribution of the term on the x-axis.

It can be seen that for every input there are elements which show quadratic dependencies. The linear term is only for q^* predominant in all elements and the influence of the quadratic term is smaller than 10 % for the majority of matrix elements. Also, for ω_g and ω^* there are many terms with a high contribution of the quadratic term.

However, the values discussed here are relative to the matrix elements' change throughout the input sweep, which means that a matrix element's behaviour could follow a quadratic dependency but with small absolute change. It has to be taken into account that the interpretation of the matrix elements as it is done here is not related to any physical behaviour, which is why the meaningfulness of this figures is very limited. Moreover, the characteristics of the B and C matrix elements strongly depend on the choice of the canonical representation and the k -factors discussed in subsection 4.4.6.

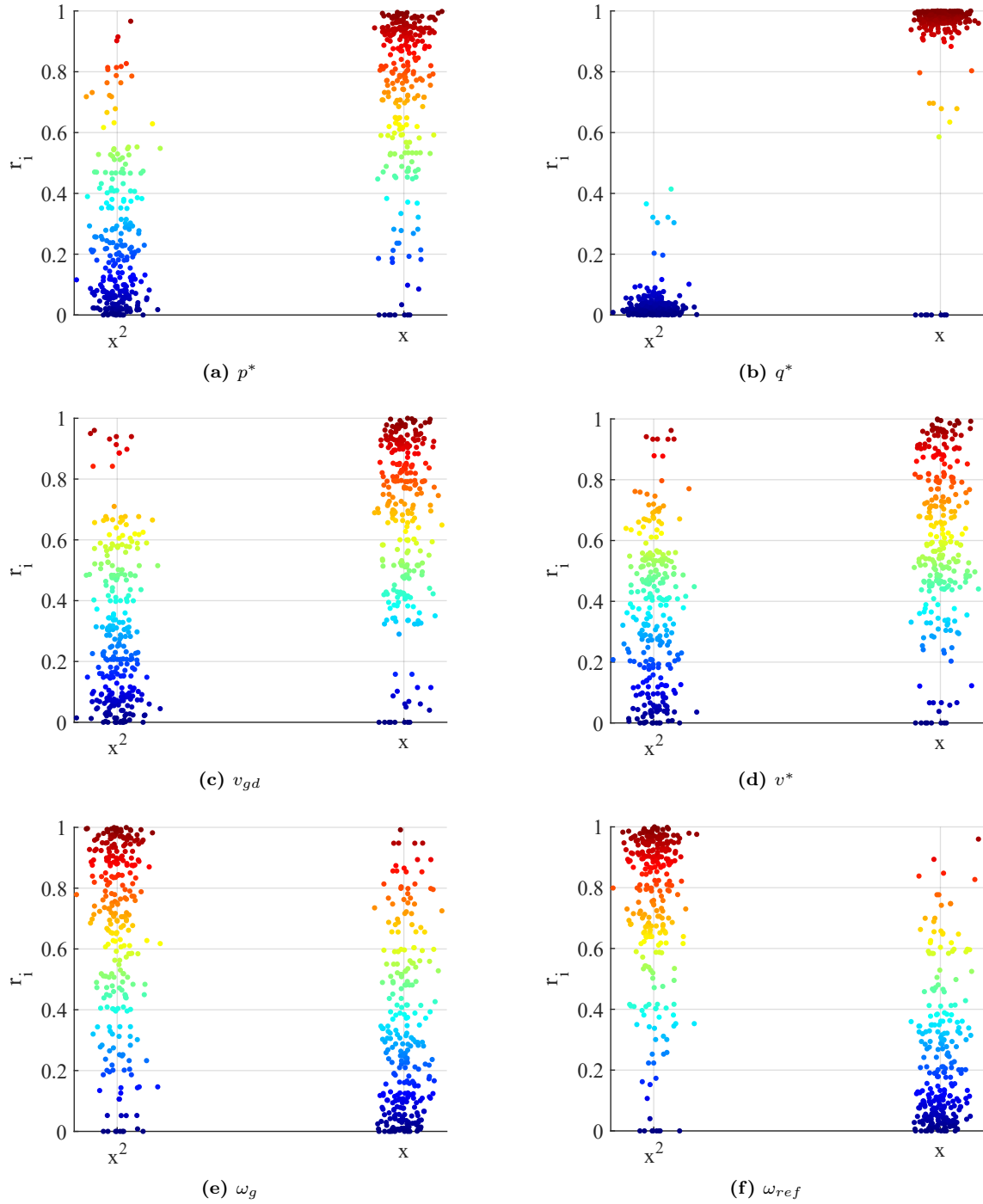


Figure 6.10: Contribution of terms for each fit function for the individual input sweeps

6.2. Pairwise Combination of Input Sweeps

6.2.1. Fit of Matrix Elements

For simplification there is no separate discussion for the elements of the A, B and C matrix in the following but all matrix elements are presented together. Table 6.4 shows the NRMSEs for the pairwise combination of input for the two-dimensional fit and the superposition of the individual fits.

Table 6.4: Pairwise combined fit accuracy

swiped	NRMSE in %			
	fit		superpos. of individual fits	
	mean	max	mean	max
p^*, q^*	0.17	1.32	0.55	4.65
p^*, v_{gd}	0.13	1.05	0.52	7.49
p^*, v^*	0.15	1.15	0.94	9.01
p^*, ω_{g}	0.45	3.11	2.67	44.67
p^*, ω^*	0.63	6.66	3.60	31.06
q^*, v_{gd}	0.03	0.29	0.21	11.85
q^*, v^*	0.07	1.12	0.65	21.2
q^*, ω_{g}	0.39	4.50	0.62	8.84
q^*, ω^*	0.52	3.72	0.70	5.73
v_{gd}, v^*	0.05	0.44	0.30	16.28
$v_{\text{gd}}, \omega_{\text{g}}$	0.36	4.35	0.74	11.80
v_{gd}, ω^*	0.43	2.62	0.71	7.88
v^*, ω_{g}	0.33	3.94	0.89	11.85
v^*, ω^*	0.43	2.99	0.97	6.34
$\omega^*, \omega_{\text{g}}$	0.59	7.77	3.46	41.55

For some input combinations the solver of the nonlinear steady-state equations does not converge. These input values are left out.

6.2.2. Discussion of Fit Functions

To gain information about the pairwise dependencies of inputs, the terms of the fit functions for the pairwise combined sweeps are analyzed as described in subsection 4.7. It is expected that combinations where the terms that depend on both variables only have small contributions to the fit functions can be approximated by a linear combination of the individual sweep fit functions as displayed in (4.10).

For the combination of p^* and q^* the contribution of all mixed terms is smaller than 5 % for all fits and it is expected that the linear combination will be a good approximation for this input pair. For example the combination of ω_g and ω^* on the contrary, the term x^2y^2 is dominant, which indicates a strong non-linearity between these inputs and a high error is expected when the relation is approximated by the linear combination. These suppositions match with the values in Table 6.4, where the maximum error for the superposition for p^* and q^* is very small at 4.6 % and the one for the combination of ω_g and ω^* is high with 41.6 %. The combination of p^* and ω_g produces the highest error, which matches with the observation that according to Figure 6.11 this combination has many elements where the xy^2 or x^2y^2 term has noticeable influence. However, it has to be taken into account, that the mixed terms are less critical for inputs with input ranges close to one (i.e. v_{gd} , v^* , ω_g , ω^*) than for inputs with broader input ranges (p^* , q^*).

Figure 6.11 shows that the fit functions for all three pairwise combinations of q^* , $v_{g,d}$ and v^* contain some mixed terms but the later discussed frequency domain analyses show that they can be neglected. This might be the case because the mixed terms appear in elements that do not have a strong influence on the input output behaviour, because the absolute change of the affected matrix elements is small or because they can be well approximated by a linear term as the input ranges of both v_{gd} and v^* are close to 1.

Again it has to be emphasized that this evaluation of the matrix elements only gives limited information about the real system behaviour and the above considerations have to be seen critically. However, the representation they form a base for an attempt to explain the nature of the input dependencies.

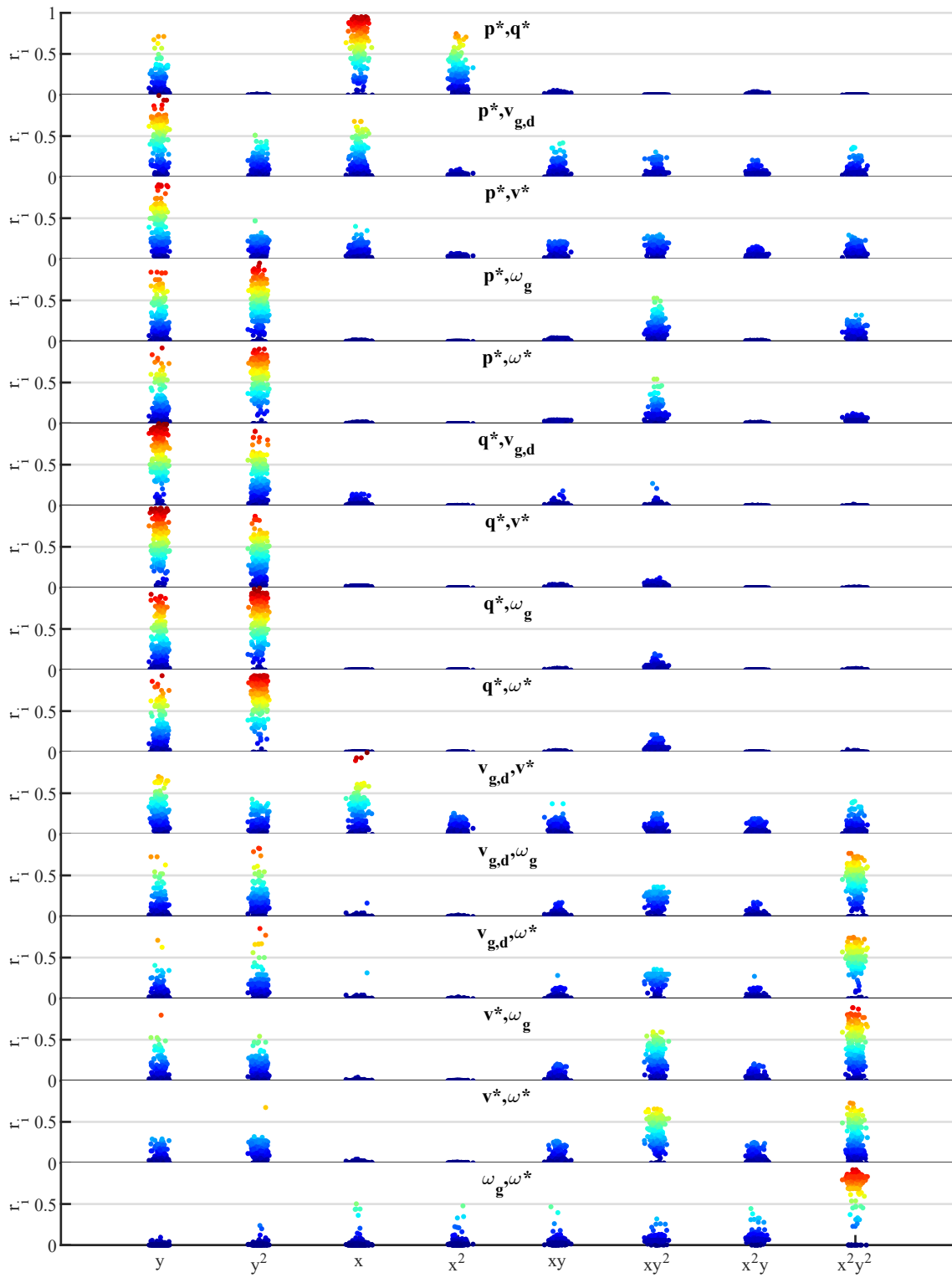


Figure 6.11: Contribution of terms to each fit function for pairwise combined sweeps

6.2.3. Validation in Frequency Domain

Table 6.5: Pairwise combined sweep errors in frequency domain

swept	max. magnitude errors in dB							
	200 from 0.1 to 200 Hz				400 from 0.1 to 2000 Hz			
	fit		superpos.		fit		superpos.	
	mean	max	mean	max	mean	max	mean	max
p^*, q^*	0.1	-5.6	0.3	-28.7	0.0	-5.6	0.3	-28.6
$p^*, v_{g,d}$	0.1	12.4	0.5	27.2	0.1	13.5	0.6	28.2
p^*, v^*	0.1	14.6	0.5	-33.7	0.1	17.23	0.6	35.6
p^*, ω_g	0.3	33.1	1.2	31.5	0.4	-24.2	1.3	-33.1
p^*, ω^*	0.2	-27.1	1.0	28.2	0.2	-22.6	1.2	-29.1
$q^*, v_{g,d}$	0.0	-0.0	0.0	-1.0	0.0	-0.0	0.0	-0.9
q^*, v^*	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.8
q^*, ω_g	0.3	-25.7	0.6	38.1	0.3	-25.2	0.6	36.6
q^*, ω^*	0.2	-20.6	0.6	-27.1	0.3	-21.2	0.7	-27.1
$v_{g,d}, v^*$	0.0	-0.1	0.0	-2.1	0.0	-0.1	0.0	-2.1
$v_{g,d}, \omega_g$	0.3	-24.3	0.9	43.9	0.3	-36.6	1.0	53.0
$v_{g,d}, \omega^*$	0.3	23.1	0.8	-36.4	0.3	24.0	0.9	-36.6
v^*, ω_g	0.3	-31.5	0.9	-48.7	0.3	-31.3	1.0	-46.9
v^*, ω^*	0.2	-18.4	0.9	32.8	0.2	-19.7	1.1	32.8
ω^*, ω_g	0.4	-28.9	1.7	-36.7	0.4	-28.1	2.0	-31.1

The frequency domain analysis was carried out on 961 operating points (31 values per input combined with all 31 values of the second input). This leads to the fact that not every operating point examined in the previous frequency analysis is included in this analysis and explains possible inconsistencies regarding the exact numbers. However, the 961 operating points should be sufficient to judge the model behaviour.

Table 6.5 shows the mean and maximum for the maximum magnitude errors for the pairwise combined input sweeps for the fit as well as for the superposition of the two corresponding one-dimensional fits. The mean values for the fit are all lower than 0.41 dB, which is an indicator that the chosen fit function is suitable for the majority of input output combinations. The maxima of the magnitude errors for the fits are in the same range as for the one-dimensional fits.

For the superposition, the errors are in general higher than for the fit, which was expected. It was also expected that the combinations that contain input variables that had high errors in the individual sweep setup have high errors in the combined

setup as well. The mean absolute errors for the superposition are all lower than 2 dB, which means that the superposition is in general a good approximation for the majority of operating points and input output combinations.

What can be seen Table 6.5 is that the mutual combinations of q^* , v_{gd} and v^* show very low errors of maximum 2.11 dB for the worst case for the superposition. This is a pleasant result as it means that the system behaviour's dependency of those inputs in the frequency domain can be approximated by the superposition for all the operating points that were investigated so far. The operating point that causes the maximum TDE for the pairwise combinations of q^* , v_{gd} and v^* appears at $v_{gd} = 0.9$ and $v^* = 1.1$. Both voltages are at their input range limits. The corresponding frequency response is shown in Figure 6.12. The error appears only at low frequencies and is as aforementioned very small. The superposition approximates the frequency response well even in the worst case for this combination.

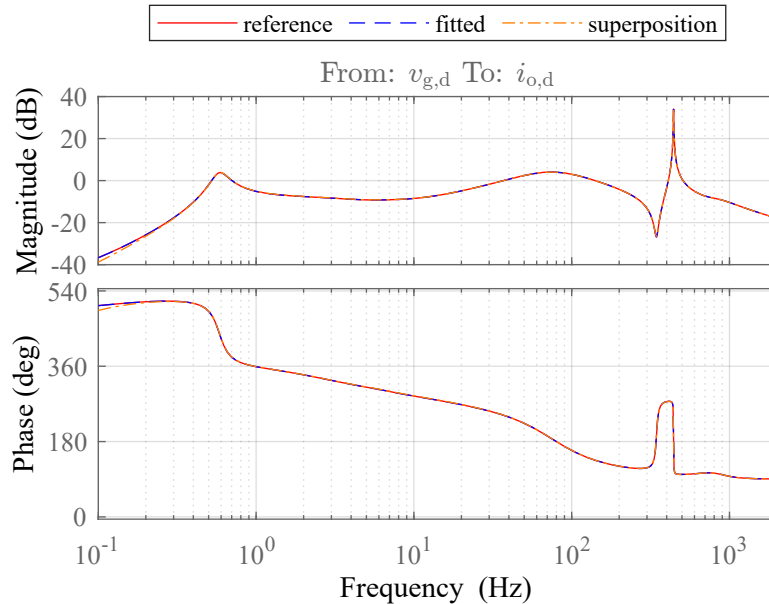


Figure 6.12: Bode plot for $v_{gd} = 0.9$ and $v^* = 1.1$

For all the other combinations the superposition produces maximum magnitude errors with absolute values higher than 27 dB. For some combinations also the 2D-fit has high maximum errors, especially when ω_g or ω^* are part of the combination. This was expected, as they showed these high errors also in the one dimensional fit.

On the other hand, the combination of p^* and q^* for example has a maximum error of only -5.6 dB for the 2D fit but -28.7 dB for the superposition. This indicates that the high error is probably caused by the omission of the mixed terms.

Figure 6.13 shows the maximum magnitude errors for the superposition for all investigated operating points for the combination of p^* and q^* . It can be seen that significant errors appear for operating points with small values of p^* and around

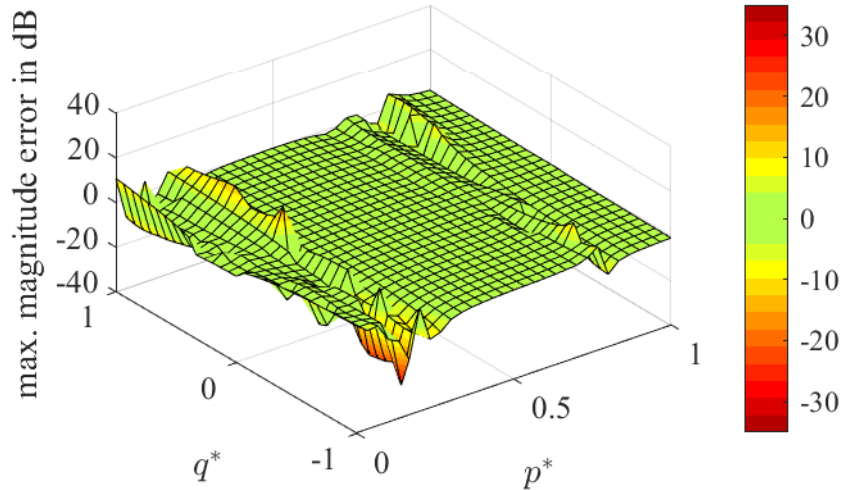


Figure 6.13: Maximum magnitude errors for the superposition for q^* and p^* for all analyzed operating points

$p^* = 0.8$. Errors with absolute values higher than 10 dB are limited to few input output combinations and solitary operating points. The same pattern shows up for the combination of p^* with v_{gd} and v^* . This means that the superposition of all four input dependencies p^* , q^* , v_{gd} and v^* is supposedly a very good approximation of the model behaviour for all input combinations as long as p^* is between approximately 0.4 and 0.7 and above 0.9. The other operating points need a more detailed analysis.

Figure 6.14 shows the bode plot for the operating point combination for p^* and q^* with the highest maximum magnitude error at $p^* = 0.1667$ and $q^* = -0.9333$. It can be seen that the problem appears mainly in the lower frequency range. The superposition underestimates the static gain and produces a negative peak that cannot be found in the reference. Frequencies larger than 1 Hz are approximated well. Furthermore, the plot shows that the 2D-fit that includes the mixed terms approximates the behaviour well in this operating point. This confirms the suspicion that the deviation comes most likely from the omission of the mixed terms.

The errors around $p^* \approx 0.8$ are located in the upper frequency range. Figure 6.15 shows the bode plot for the operating point $p^* = 0.7667$ and $q^* = -0.8$ that causes the largest error of 10.7 dB in this area. The error appears in the negative peak at 510 Hz and can almost not be seen in the graphic. The magnitude is with -50 dB to -60 dB low enough to say with high confidence that this kind of errors are not relevant. For further investigations it might be worth thinking about some kind of weighting to assign errors at low magnitudes less importance. With the assumption that the errors at $p^* = 0.8$ are not relevant due to the small magnitude, the range in which the superposition for the four inputs is a good approximation increases to all operating points with p^* larger than approximately 0.4. However, it is possible that similar errors appear also in the positive peak which would have a more severe

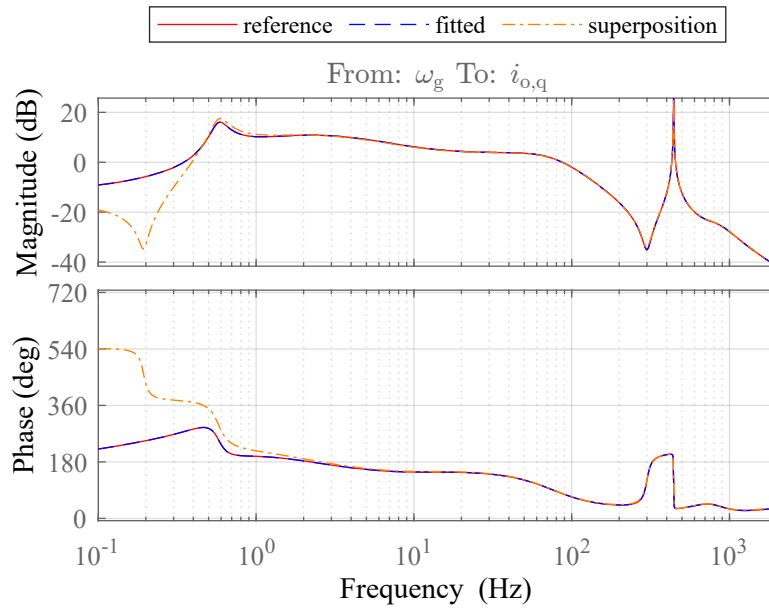


Figure 6.14: Bode plot for $p^* = 0.1667$ and $q^* = -0.9333$

impact in time domain simulations. On the other hand does the approximation locate the pole pair at the right frequency which might be more relevant in modal analyses than the exact amplitude of the transfer function at that point.

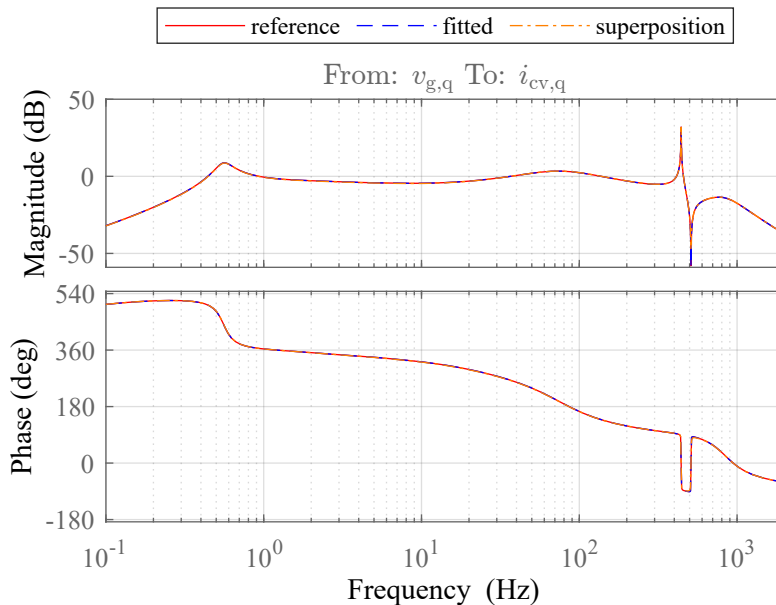


Figure 6.15: Bode plot for $p^* = 0.7667$ and $q^* = -0.8$

The combination of ω_g and ω^* produces the highest mean error, which was expected due to their high errors when analyzed individually. Figure 6.16 shows the distribution of the maximum errors over the input range. It can be seen that there is a contiguous area where the frequency response can be approximated by the superposition with low errors for all input output combinations. Outside that area higher errors appear. The errors are distributed in fronts along diagonal lines in the ω_g - ω^* -area.

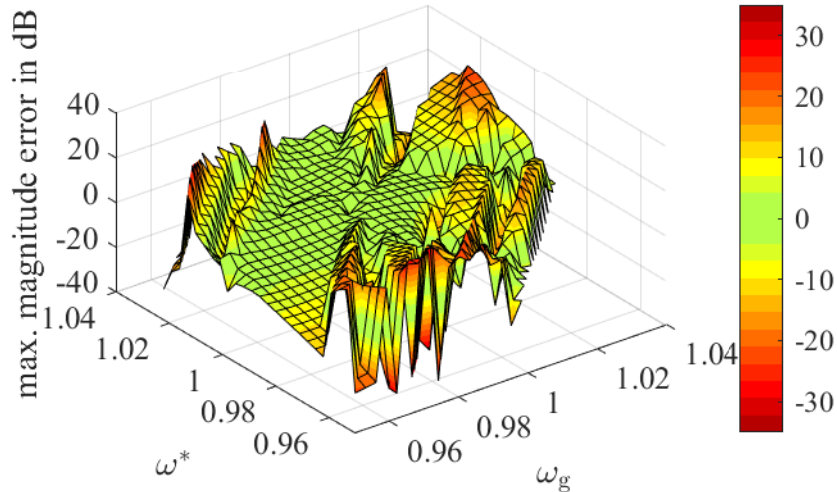


Figure 6.16: Maximum magnitude errors for the superposition for ω_g and ω^* for all analyzed operating points

As in the individual fits, high positive errors are followed by high negative errors. The frequency responses that belong to the highest maximum error of the fit and the superposition are displayed in Figure 6.17a and Figure 6.17b. Again, the problem seems to be the dip around 0.6 Hz that was already discussed in the individual fit analysis.

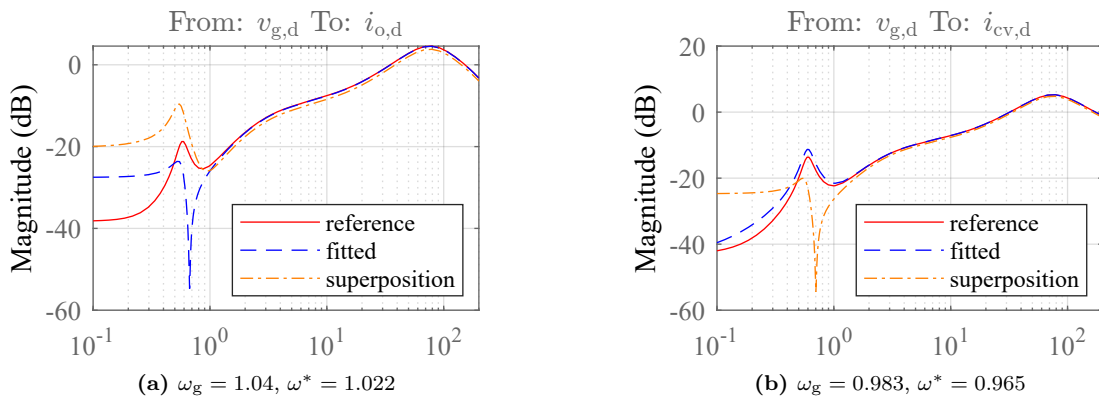


Figure 6.17: Transfer functions that cause the maximum magnitude errors for (a) the fit and (b) the superposition for the combination of ω_g and ω^*

The highest overall error can be found in the combination of v^* and ω_g . The corresponding transfer function shows high similarity to the one with the highest error in p^* and q^* in Figure 6.14. For this combination there is also an area around the default operating point where the superposition is a good approximation with low maximum errors for all input output combinations. This can be seen in Figure 6.18.

For all input combinations, such a coherent area with low maximum magnitude errors for the superposition can be found. In this areas the superposition is with high likelihood a good approximation of the frequency behaviour. However, there might be phenomena that appear between the analysed operating points that are

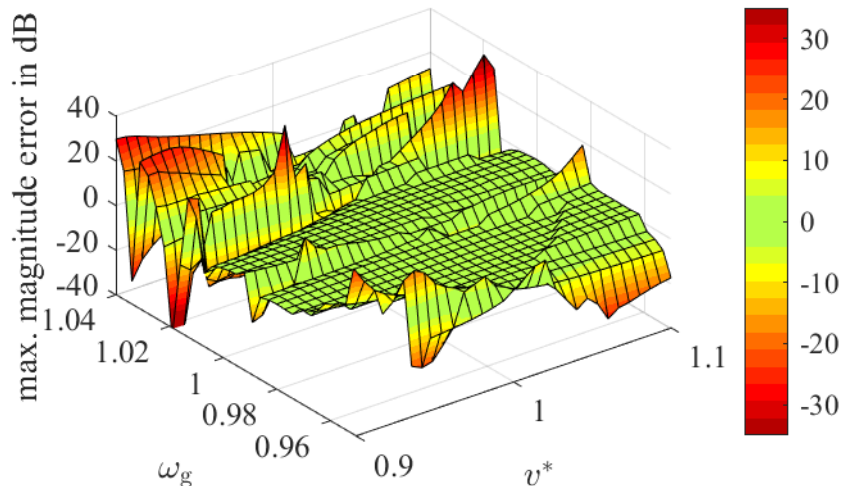


Figure 6.18: Maximum magnitude errors for the superposition for v^* and ω_g for all analyzed operating points

not covered here. Moreover, there might be phenomena that are not covered by this discussion since they were not detected by simply looking at the selected maximum magnitude errors.

Outside of these areas with the low magnitude errors, the cause and impact of the errors need to be assessed in detail, depending on the use case. However, the general impression that evolves from this investigation is that especially in close proximity of the default operating point the superposition is a good approximation for all input combinations. For the combination of p^* , q^* , v_{gd} and v^* the superposition seems to be a good approximation over the whole input range with $p^* > 0.4$. Below that the approximation could be applied depending on the use case.

The two-dimensional fit functions that also include the mixed terms show in most cases significantly lower errors than the superposition. In case the behaviour of the model obtained by superposition is not sufficient, including the mixed terms of the 2D fit could be a possibility. This is not further discussed in this thesis. In the case that for the discussed errors in the individual input sweeps the fit quality has to be improved, modifications of the used fit functions might be necessary as there are likely terms of different order that could improve the fit quality.

To sum up, the frequency domain behaviour for the variation of two inputs can be extrapolated by using a superposition. For the majority of operating points and input output combinations the errors are low. The mean magnitude error is lower than 2 dB. For q^* , v_{gd} and v^* the approximation produces small errors over the whole operating range. In combination with p^* the superposition has only small errors for $p^* > 0.4$. For the other combinations small maximum errors can be found in proximity to the default operating point. Outside of this areas high magnitude

errors can appear in some input output combinations. The impact of the phenomena that occur has to be evaluated depending on the use case. One option to limit errors could be to do sweeps for multiple default operating points instead of covering the whole operating range with fits from one default operating point.

6.2.4. Validation in Time Domain

The time domain analysis was carried out on 21 values per input that were combined with each of the 21 values of the second input for all 15 input combinations. The results are displayed in Table 6.6.

Table 6.6: Pairwise combined fit errors in time domain

swept	TDE			
	fit		superposition	
	mean	max	mean	max
p^*, q^*	$4.4 \cdot 10^{-6}$	$4.2 \cdot 10^{-5}$	$2.3 \cdot 10^{-5}$	$4.8 \cdot 10^{-4}$
$p^*, v_{g,d}$	$4.6 \cdot 10^{-6}$	$5.4 \cdot 10^{-5}$	$3.6 \cdot 10^{-5}$	$9.8 \cdot 10^{-4}$
p^*, v^*	$4.6 \cdot 10^{-6}$	$6.4 \cdot 10^{-5}$	$4.7 \cdot 10^{-5}$	$1.4 \cdot 10^{-3}$
p^*, ω_g	$3.3 \cdot 10^{-5}$	$5.3 \cdot 10^{-5}$	$9.6 \cdot 10^{-5}$	$1.9 \cdot 10^{-3}$
p^*, ω^*	$2.6 \cdot 10^{-5}$	$3.5 \cdot 10^{-3}$	$1.6 \cdot 10^{-4}$	$1.2 \cdot 10^{-2}$
$q^*, v_{g,d}$	$2.4 \cdot 10^{-7}$	$2.2 \cdot 10^{-6}$	$3.1 \cdot 10^{-6}$	$4.7 \cdot 10^{-5}$
q^*, v^*	$4.8 \cdot 10^{-7}$	$6.8 \cdot 10^{-6}$	$9.1 \cdot 10^{-6}$	$1.8 \cdot 10^{-4}$
q^*, ω_g	$2.4 \cdot 10^{-5}$	$2.4 \cdot 10^{-4}$	$3.1 \cdot 10^{-5}$	$8.6 \cdot 10^{-4}$
q^*, ω^*	$2.5 \cdot 10^{-5}$	$4.3 \cdot 10^{-4}$	$5.5 \cdot 10^{-5}$	$5.4 \cdot 10^{-3}$
$v_{g,d}, v^*$	$5.4 \cdot 10^{-7}$	$8.9 \cdot 10^{-6}$	$6.2 \cdot 10^{-6}$	$1.0 \cdot 10^{-4}$
$v_{g,d}, \omega_g$	$2.5 \cdot 10^{-5}$	$3.1 \cdot 10^{-4}$	$4.3 \cdot 10^{-5}$	$8.5 \cdot 10^{-4}$
$v_{g,d}, \omega^*$	$2.6 \cdot 10^{-5}$	$1.9 \cdot 10^{-3}$	$7.4 \cdot 10^{-5}$	$6.0 \cdot 10^{-3}$
v^*, ω_g	$2.1 \cdot 10^{-5}$	$1.9 \cdot 10^{-4}$	$5.2 \cdot 10^{-5}$	$1.6 \cdot 10^{-3}$
v^*, ω^*	$2.3 \cdot 10^{-5}$	$1.8 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$	$8.5 \cdot 10^{-2}$
ω^*, ω_g	$3.7 \cdot 10^{-5}$	$9.2 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	$2.7 \cdot 10^{-3}$

Again, it can be seen that the errors are smallest for the mutual combination of q^* , $v_{g,d}$ and v^* . The step response with the highest TDE error observed for these pairwise combinations is $1.8 \cdot 10^{-4}$ and appears for the combination of q^* and v^* for $q^* = -1$ and $v^* = 0.9$. It is displayed in Figure 6.19. The deviation between reference, fit and superposition is barely visible.

When p^* is combined with the three previously mentioned inputs the maximum TDE grows about a factor 8 to the maximum TDE of $1.4 \cdot 10^{-3}$ in the combination of p^*

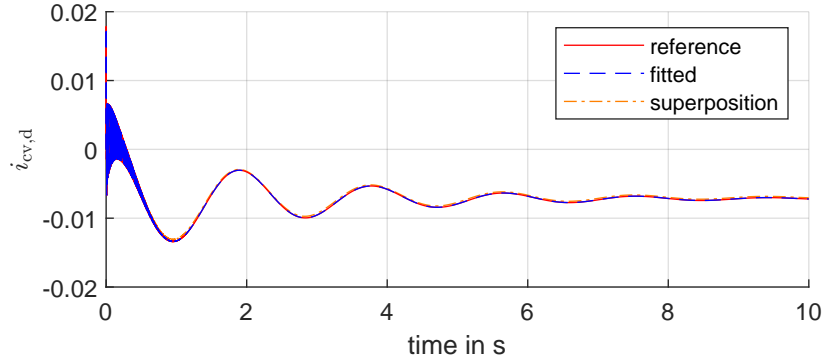


Figure 6.19: Step response of $i_{cv,d}$ for a step in v^* at $q^* = -1$ and $v^* - 0.9$

and v^* . It appears at the upper input range limits $p^* = 1$ and $v^* = 1.1$ in $v_{o,q}$ for a step in v^* . This step response is displayed in Figure 6.20. The main problem there is the damping of the oscillation which is too small in the superposition. The error is no exception. As displayed in Figure 6.21, the errors grow with the distance to the sweep axes. This and the fact that the errors for the 2D fit for this combination are significantly lower underlines that the error evolves from the omission of the mixed terms. This behaviour can be observed in all input output combinations even though for some it is more distinct than for others.

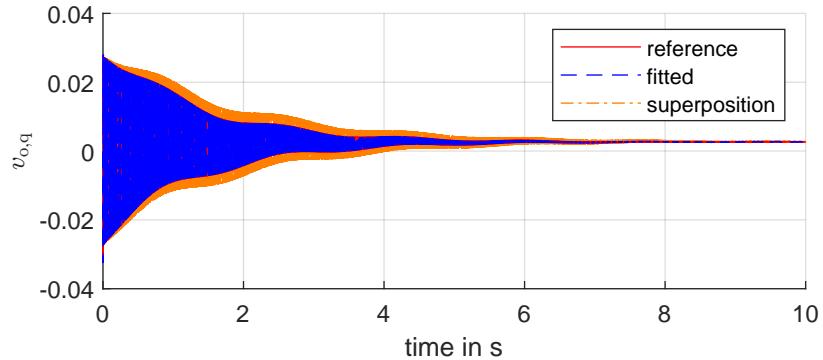


Figure 6.20: Step response of $v_{o,q}$ for a step in v^* at $p^* = 1$ and $v^* = 1.1$

The combinations of p^* , v^* and ω_g with ω^* have the highest average errors for the superposition with $1.4 \cdot 10^{-4}$ to $1.6 \cdot 10^{-4}$. The combination of v^* and ω^* produces the highest overall TDE. The reason is that for operating points close to the stability limit the reference model is still stable while the superposition becomes unstable a bit earlier. The maximum TDE occurs at $v^* = 1.08$ and $\omega^* = 1.04$. This case is displayed in Figure 6.22. The amplitude of the model from the superposition increases, while the reference decreases, leading to the high TDE.

Figure 6.23 shows that the high TDEs for the combination of v^* and ω^* only appear locally as outliers and only in singular input output combinations. The majority of errors is significantly smaller. Similar outliers can be observed for the combination of p^* and ω^* . In that case the damping is significantly higher in the superposition than in the reference. In both cases the oscillation frequency is met well by the

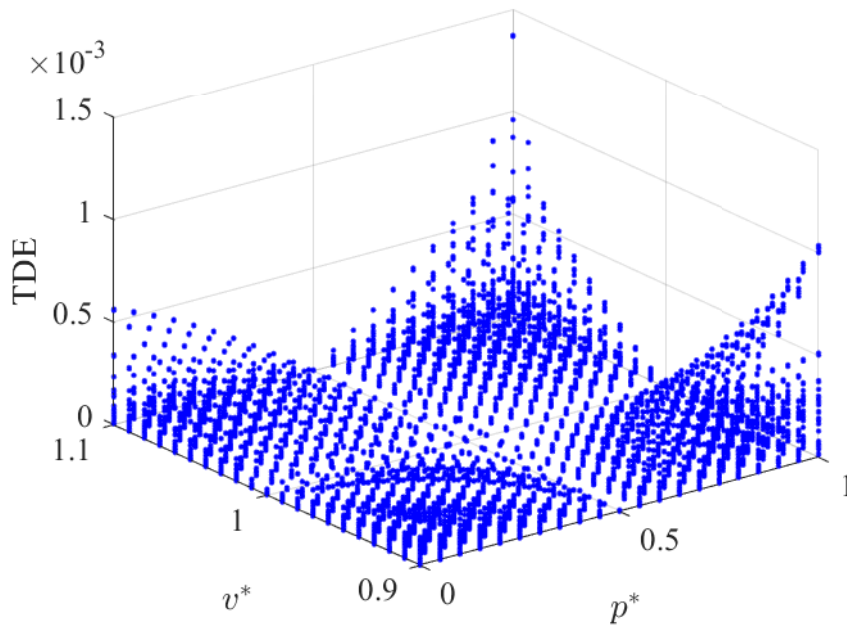


Figure 6.21: TDEs for the superposition for p^* and v^* for all analyzed operating points

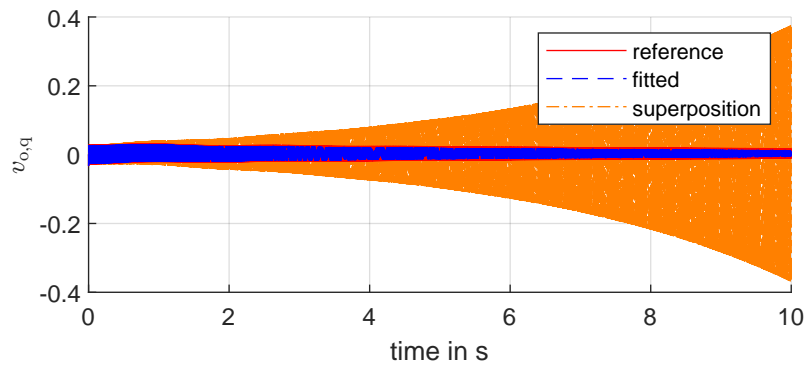


Figure 6.22: Step response of $i_{cv,q}$ for a step in v^* at $v^* = 1.08$ and $\omega^* = 1.04$

superposition. It can be said that in the two discussed cases the deviation of the damping cases the errors.

For the combination of ω_g with ω^* on the contrary, the high errors are no outliers, as displayed in Figure 6.24. The errors are clearly associated to the difference to the default operating point. The approximation causes the highest errors when both ω_g and ω^* are at the upper limits of their input range. The lower half of the input range of ω_g is not displayed because of instability in this area.

Figure 6.25 displays the step response that causes the maximum error for the combination of ω_g and ω^* . It can be seen that the main cause of the error is the steady state deviation. The fit on the other hand approximates the step response well. This indicates that the mixed terms that are omitted by the superposition have in this case a strong influence on the static gain. This matches the observations in

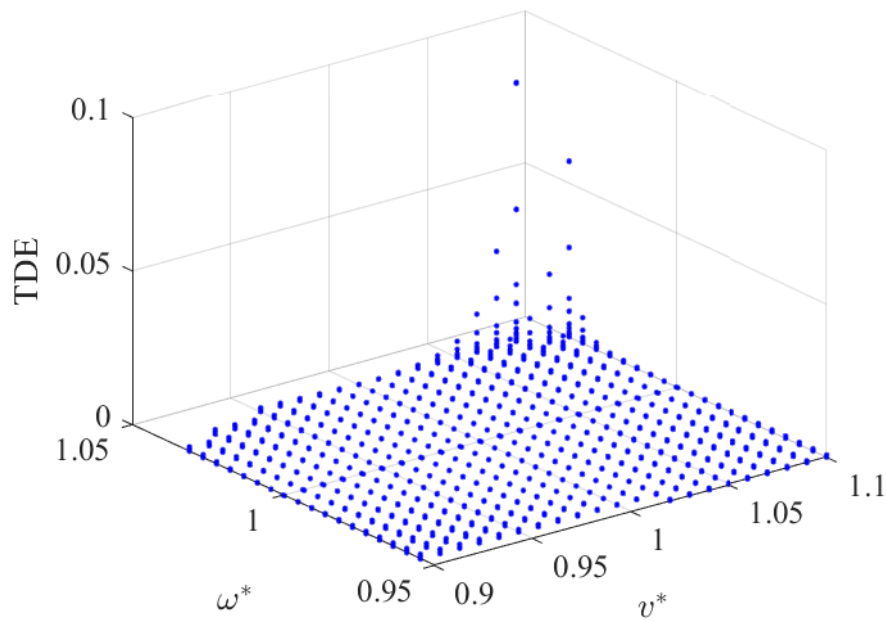


Figure 6.23: TDEs for the superposition for v^* and ω^* for all analyzed operating points

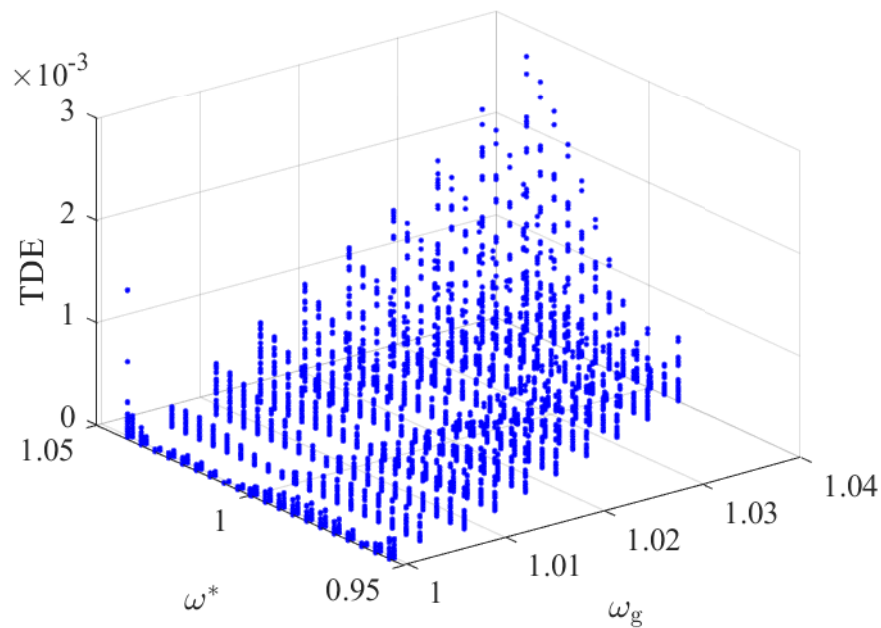


Figure 6.24: TDEs for the superposition for ω_g and ω^* for all analyzed operating points

the frequency domain analyses, that deviations appear mainly in the low frequency range.

In contrast to the frequency domain analysis there are no jumps in the errors but the errors grow monotonously with the distance to the sweep axes. Therefore it is likely that the statements previously made in this time domain analysis are also true for the other operating points between the ones chosen for the analysis that were not explicitly calculated.

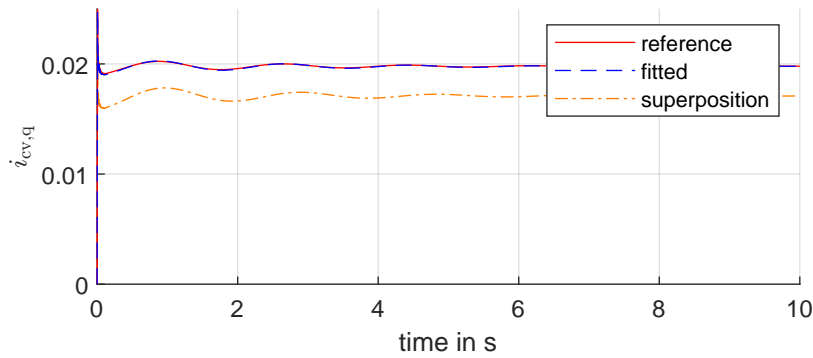


Figure 6.25: Step response of $i_{cv,q}$ for a step in $v_{g,d}$ at $\omega_g = 1.04$ and $\omega^* = 1.04$

6.3. Combination of all Inputs by Superposition

The maximum magnitude error of the transfer function and the TDE were calculated for 15625 operating points. They are the result of a combination of 5 values for each input. As the frequency domain analysis and the application of the maximum magnitude error needs additional careful interpretation of the bode plots when large magnitude errors are detected, this is not feasible for the validation of a combination of all inputs due to the high number of operating points. Therefore it was decided to discuss the frequency domain analysis only for the combination of q^* , v^* and $v_{g,d}$. The time domain analysis is discussed in detail for all combinations.

The frequency domain analysis for the combination for q^* , v^* and $v_{g,d}$ confirmed the hypothesis that the superposition is a good approximation for all input combinations of q^* , v^* and $v_{g,d}$. The maximum magnitude error in the worst case that was detected in this analysis for this combination is 2.3 dB for $q^* = 0.5$, $v^* = 1.1$ and $v_{g,d} = 0.9$. The corresponding bode plot for v^* to $v_{o,q}$ is shown in Figure 6.26. The error is so small that it is not visible in the plot but it is located in the positive at 442 Hz.

Moving on to the time domain evaluation, to display the dependency of the errors on the input value combinations a parallel coordinate plot was chosen. There is one line for every operating point. It crosses the axis that represents an input at the value that this input has for the corresponding operating point and ends on the most right axis at the respective TDE. The color helps to follow the course of the lines. Higher errors are red, the lowest errors dark blue.

At first, only the combination of q^* , $v_{g,d}$ and v^* is discussed based on Figure 6.27. The majority of TDEs is smaller than 10^{-4} . The smallest errors appear when only q^* is varied and $v_{g,d}$ and v^* kept at their default values, which is expected behaviour.

Also the highest TDE occurs for $q^* = 0.5$, $v_{g,d} = 0.9$ and $v^* = 1.1$. This is essentially the worst case with all inputs at their limits, since the same voltage values with $q^* = 1$ lead to an unstable operating point and is thus not evaluated with a TDE.

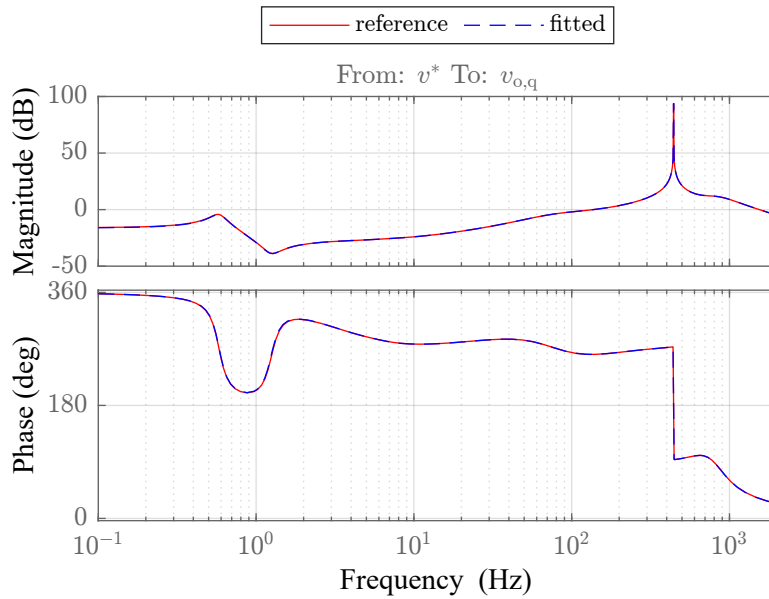


Figure 6.26: Bode plot for $q^* = 0.5$, $v^* = 1.1$ and $v_{g,d} = 0.9$

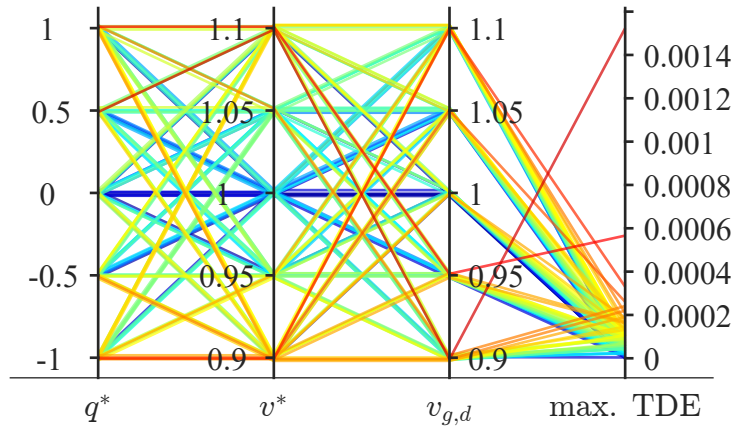


Figure 6.27: Parallel coordinates plot of TDEs for a combination of q^* , $v_{g,d}$ and v^*

The error is only $1.5 \cdot 10^{-3}$ p.u. even in this worst case for this combination. The respective step response to the step in v^* is displayed in Figure 6.28.

The output oscillates with approximately 440 Hz. The oscillation frequency of the approximated system is slightly smaller which only becomes even noticeable after some seconds and should therefore be negligible. The second highest TDE occurs for $q^* = 1$, $v_{g,d} = 0.95$ and $v^* = 1.1$ and is associated with similar oscillatory behaviour. The third highest TDE occurs for the opposite case, namely that q^* and $v_{g,d}$ are low while v^* is high. In that case there is no high frequency oscillation but a small steady state error. To sum up, based on the comparison of the step responses there were no severe problems identified in the combination of q^* , $v_{g,d}$ and v^* and an approximation by superposition seems highly viable.

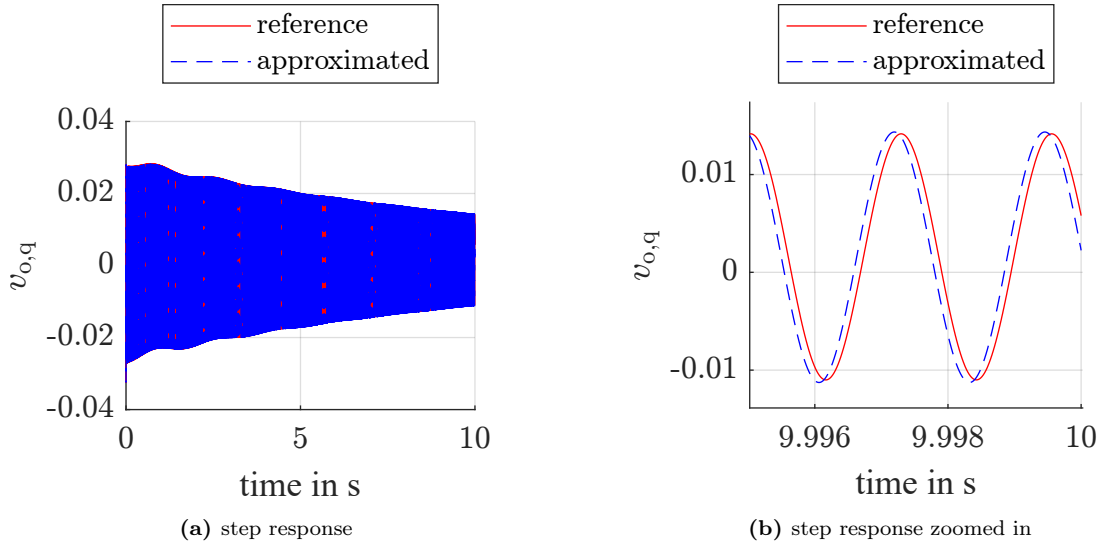


Figure 6.28: Step response of $v_{o,q}$ for a step in v^* for $q^* = 0.5$, $v^* = 1.1$ and $v_{g,d} = 0.9$

Next, the combination of q^* , $v_{g,d}$, v^* and p^* is discussed Figure 6.29. The maximum error is now about factor 10 larger than without the variation of p^* . High errors appear predominantly for high values of p^* .

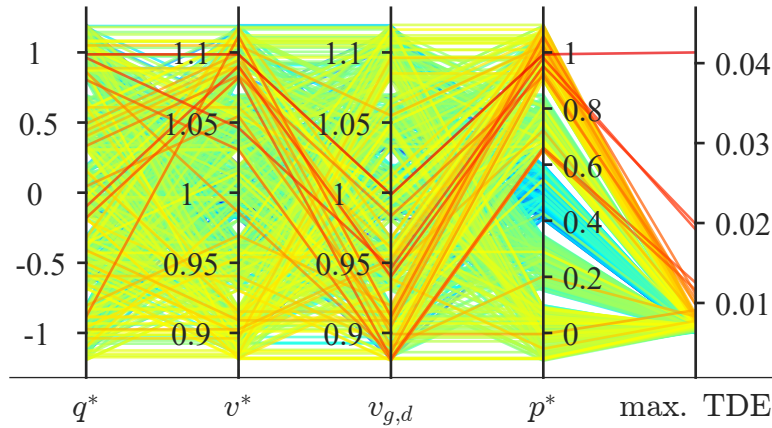


Figure 6.29: Parallel coordinate plot of TDEs for a combination of p^* , q^* , $v_{g,d}$ and v^*

The maximum TDE is detected for $p^* = 1$, $q^* = 1$, $v_{g,d} = 1$ and $v^* = 1.1$. In that operating point, the reference is stable, while the amplitude of the approximated model increases, leading to the high TDE. The corresponding step response is shown in Figure 6.30.

For the 5 highest TDEs it is the case that the errors arise from the fact that the damping of such a high 430 to 440 Hz oscillation is underestimated by the approximation while the frequency is about right. The eigenvalue pair that corresponds to frequency 440 Hz=2756 rad/s is eigenvalue 3 and 4.

There are only few investigated operating points where the errors are this severe and the result with the sixth highest TDE looks already quite acceptable for this simple

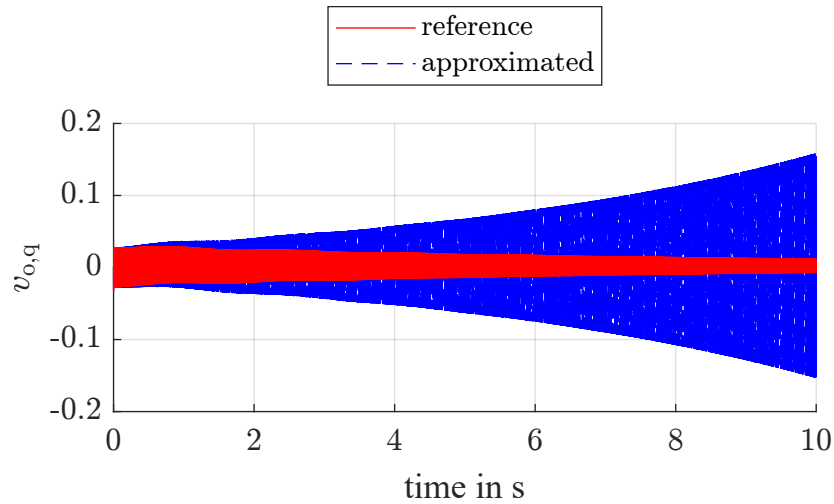


Figure 6.30: Step responses in $v_{o,q}$ for a step in v^* for $p^* = 1$, $q^* = 1$, $v^* = 1.1$ and $v_{g,d} = 1$

approximation. The damping is slightly overestimated there. The corresponding step response is shown in Figure 6.31.

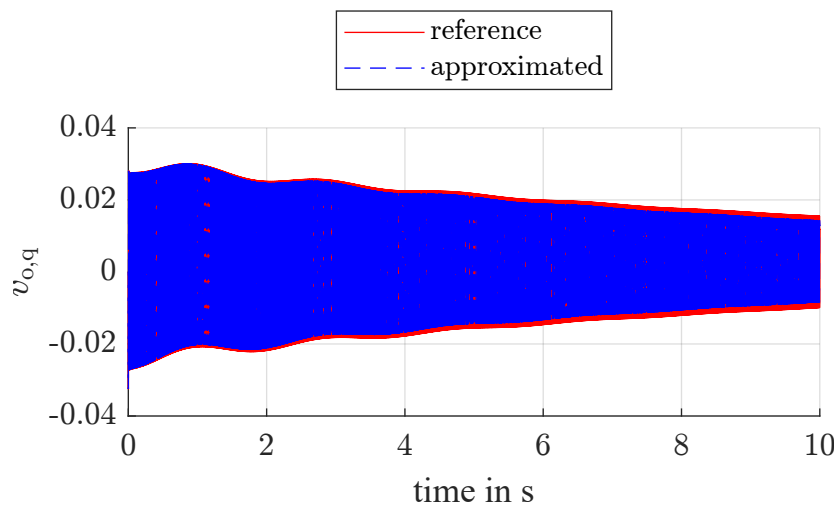


Figure 6.31: Step response of $v_{o,q}$ for a step in v^* for $p^* = 1$, $q^* = 1$, $v^* = 1.1$ and $v_{g,d} = 1$

As all other operating points produce smaller errors, this means that only 5 out of 610 valid investigated operating points show severe approximation errors for the combination of q^* , $v_{g,d}$, v^* and p^* . The conclusion is that based on the examined operating conditions the step response can probably be approximated by the superposition with negligible errors for a majority of operating points. To decrease the probability of approximation errors the input range could be limited to values close to the default operating point.

The combination of all inputs including ω_g and ω^* is displayed in Figure 6.32. The highest errors are extremely large up to over 200 p.u. of average deviation. The axes in the following parallel coordinates plots had to be changed to a logarithmic axis. Figure 6.32 shows that the highest errors are again associated to operating points where multiple inputs are at their limits. The large errors are caused by the

fact that the fitted model is unstable and the amplitude grows extremely fast while the reference is stable and the amplitude is decreasing.

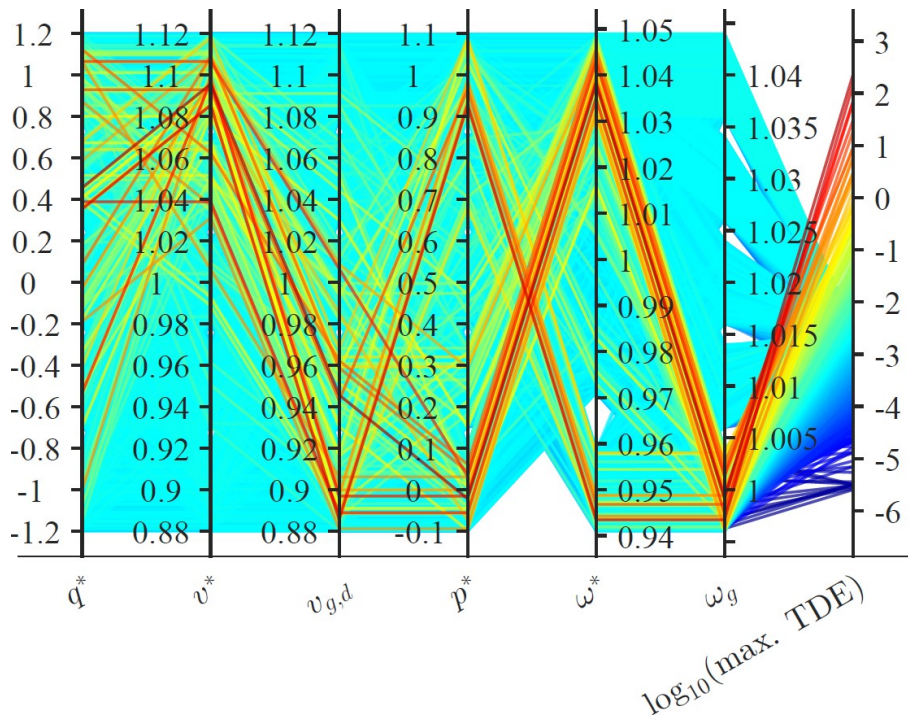


Figure 6.32: Parallel coordinates plot of TDEs for the combination of all inputs

Restricting the considered operating ranges for the evaluation reduces the observed errors significantly. When p^* is limited to 0.2 to 0.8, ω^* to 0.98 to 1.02 and the voltages to 0.95 to 1.05, the maximum error is only 0.004. The corresponding step is shown in Figure 6.33.

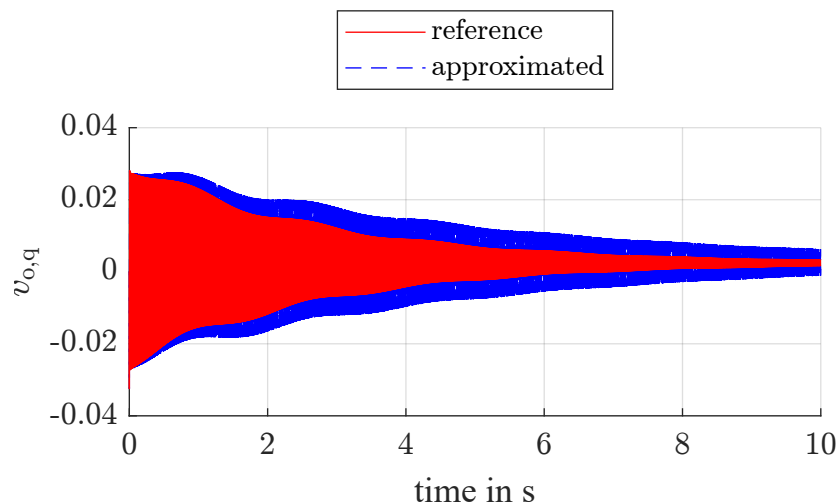


Figure 6.33: step

The analyses here cover 5 values per input and uses only the step response. Therefore the discussion might miss out on some phenomena. It is hard to tackle the exponential number of operating points, so this analysis provides only a rough screening of the input range. The time domain analysis for the combined sweeps showed that

the errors behave most likely continuous throughout the input range so that this is probably also true for the full combination of all inputs. The rough screening of all combinations confirmed that the highest errors in the time domain appear in general at the input or stability limits. Those cases are not the most relevant in practice as they are relatively unlikely and not desirable for steady-state in normal operation, but for a complete mapping of the system behaviour for all allowable operating points they need to be included into the considerations. The analysis also confirmed that for q^* , v^* and $v_{g,d}$ the superposition seems to be a reliable approximation for the whole input range. The most effective way to reduce the risk of high errors to limit the input range in which the model is applied to operating points in proximity to the default operating point. Which errors are still acceptable has to be decided depending on the specific use case and the range in which the model can be applied should be determined accordingly.

6.4. Summary Fitted Model and Approximation by Superposition

In the previous subsections the fitting was discussed in detail. A quadratic function was applied to fit each input dependency individually. Then the inputs were combined and the approximation by superposing the individual input fits were discussed. Even though the specific use case decides if the approximation is good enough, the results can be summed up as follows:

- The individual inputs can be fitted with a quadratic function. The errors between the analytical model and the fitted model in time and frequency domain seem negligible.
- $v_{g,q}$ does not have any effect on the system at all. It is not necessary to create a fit there.
- The combination of q^* , $v_{g,d}$ and v^* can be approximated well by the superposition. The errors in time and frequency domain are insignificant.
- The pairwise combinations of p^* , q^* , $v_{g,d}$ and v^* produce errors which could still be tolerable depending on the use case. The closer the input values to the default operating point the smaller the approximation errors in the time domain. Areas around the default operating point can be found where the errors in the frequency domain are small.
- The combination of p^* , q^* , $v_{g,d}$ and v^* produces significant errors for input values close to the input range limits. However, the step response can be approximated well at the majority of operating points.

- The combinations with ω_g and ω^* are most critical. Even smaller deviations can cause high errors when ω_g and ω^* are changed at the same time.
- The main issues for the approximation are the damping of the 440 Hz oscillation caused by the underdamped pole pair 3 and 4 or the static gain.
- When the input range in which the approximation is applied is limited to operating points in close proximity to the default operating point, the risk of significant errors is low.

The results of the investigations is an input dependent model that inter- and extrapolates the behaviour of the analysed VSM based on individual input sweeps. It approximates the system behaviour relatively reliable in close proximity to the default operating point.

6.5. Adaptive Simulation Model

A test with the adaptive model was run for a step wise transition from $p^* = 0.5$ to $p^* = 0.65$. The transition was taken in 15 steps with a height of 0.1 as shown in Figure 6.34. The time between two steps was 20 seconds which allowed the system to go back to steady state between two steps. The trigger for the switch to the next model was enabled 1 second before the next step. Figure 6.35a and Figure 6.35b show the resulting output curves for $i_{cv,d}$ and $i_{cv,q}$ as an example for the output behaviour.

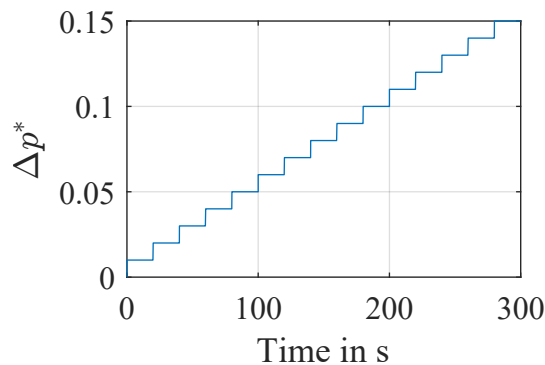


Figure 6.34: Step wise increase in p^* for the test of the adaptive model

While the output curves of the d-components of the currents and the q-component of the voltage stay close together for all models, it can be seen that the q-components of the currents and the d-component of the voltage of the analytical model without adaption to the operating point drift apart from the reference. The adaptive model matches the reference significantly better. However, there is also a deviation in the adaptive model in the form of an offset that grows with every step. This can be explained by the fact that the linearized state space model has a steady state error that adds up with every step.

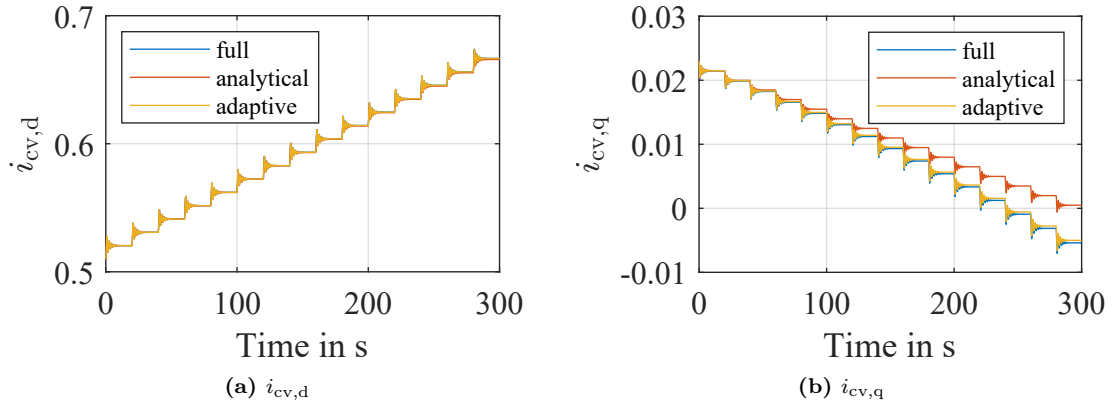


Figure 6.35: Converter current for the step wise increase in p^* for the test of the adaptive model

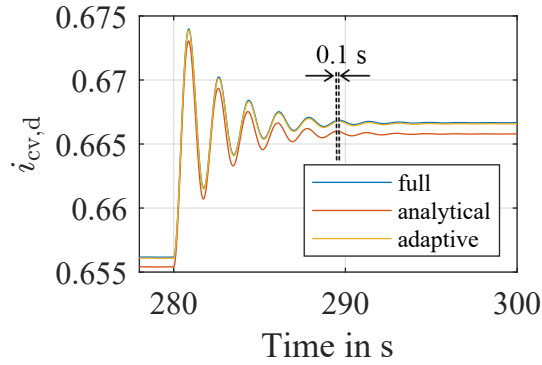


Figure 6.36: Zoomed view of $i_{cv,d}$ for the last step in p^* for the test of the adaptive model

When zooming in on the last step as done in figure Figure 6.36 for $i_{cv,d}$, it can be seen that even for the inputs where the curves match well, there is indeed an offset and a deviation in the oscillatory frequency for the analytical model. This becomes apparent when comparing the locations of the later peaks of the oscillation, even though the difference is very small. The shift of the fifth maximum of the static analytical model compared to the one of the full model is approximately 0.1 seconds. The maxima are marked in Figure 6.36 by the dotted lines. The adaptive model shows a small offset as well but significantly less than the static model and it matches the oscillation frequency better. For $v_{o,q}$ the static analytical model performs best. The offset is nearly zero for all steps but the difference in the oscillation frequency can still be observed at the end of the step wise transition.

To sum up, the application of the adaptive model was successful and the errors between the state-space model and the electrical simulation could be decreased by adapting it to the new operating points.

6.6. Fits from Model Identification

In this section the results are shown for the fitting when the state-space models used as input for the fitting come from the results of N4SID and VF instead of the analytical equations. The input used for the identification is the transfer function of the analytical model.

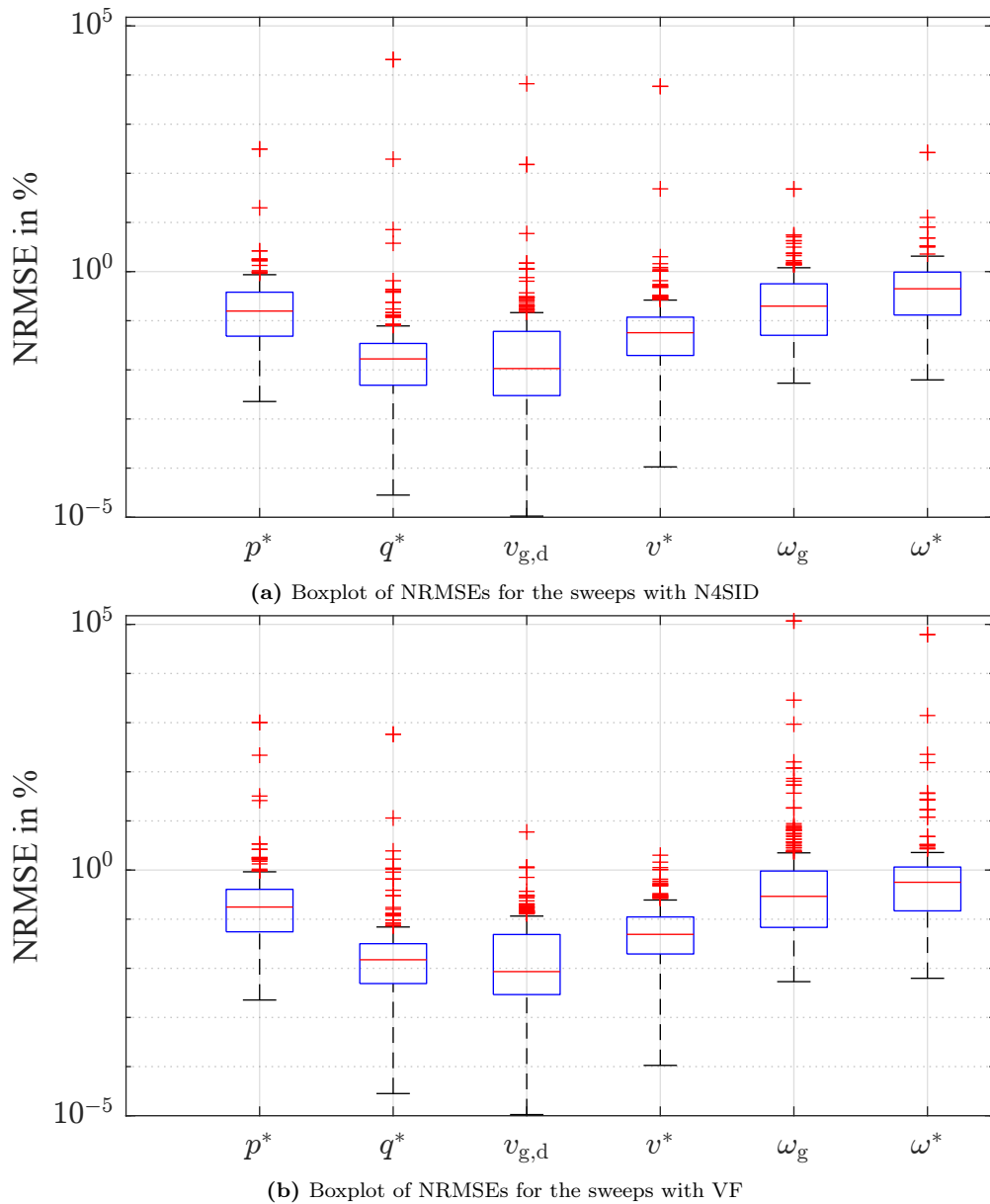
Table 6.7 shows the mean and maximum NRMSEs for the matrix element for the fit from the analytical model and the fit from the N4SID and VF results for a sweep over 11 operating points per input. The numbers are only displayed for completeness and the interpretation of the fit quality needs a more detailed analysis.

Table 6.7: Individual fit accuracy with identification

swiped	NRMSE of fit in %					
	fit from analytical tf		fit from n4sid		fit from vf	
	mean	max	mean	max	mean	max
p^*	0.28	2.63	2.75	311.44	9.12	1007.44
q^*	0.03	0.65	159.43	20619.30	4.58	580.77
v_{gd}	0.08	5.95	26.83	6647.54	0.08	5.95
v^*	0.11	2.00	45.53	5879.74	0.11	2.00
ω_g	0.47	5.54	0.85	47.96	920.74	11732.25
ω^*	0.69	8.07	2.76	264.08	486.29	62125.30

The numbers in Table 6.7 indicate that the fit quality is significantly worse when applying an identification method. Figure 6.37a and Figure 6.37b on the other hand show that the extremely high mean and maximum NRMSEs are caused by outliers and that in reality the majority of elements has very small NRMSEs. The outlying elements in the boxplot belong to elements that cannot be identified exactly in every step but produce some noise around the actual value or have solitary values that significantly deviate from the actual value. These elements are all associated with the state with the eigenvalue -475, which is relatively unimportant for the input-output behaviour according to the HSV. From experiences gained during the investigations it can be said that if an element is not identified reliably, it has very low impact on the input output behaviour of the system. This makes sense, as if it would have a strong impact, it would most likely be identified more reliably by the algorithm.

The algorithms reliably detect all 18 states for all analyzed operating points and the majority of elements is identified very accurately. Therefore the quality of the fitted model as a whole is not as bad as Table 6.7 implies.



This is confirmed when examining the frequency domain analysis. There it can be observed, that there is no noticeable difference between the fits from the analytical model and the results from the identification. In the frequency domain analysis performed on the individual sweeps, it turned out that the difference between the maximum magnitude errors of the fit from the analytical equations and the ones from the identification methods is smaller than $3 \cdot 10^{-3}$ dB. Due to the extremely high resemblance of the models in the frequency domain, a time domain analysis was omitted.

Moreover, some of the large NRMSEs could potentially be avoided by adding methods that detect outlying values throughout the sweep before fitting a function. Such an additional preprocessing of the sweep results was not implemented in this thesis but is most likely beneficial for further use cases with identified data.

Figure 6.37a and Figure 6.37b display that VF and N4SID produce comparable results and the major difference in the NRMSEs are the outliers. In general, the NRMSEs are not a good basis to judge the quality of the identification and therefore a comparison of VF and N4SID is not done at this point.

It has to be underlined again that the averaged and maximum NRMSEs alone are not adequate to judge the quality of the model as their informational value regarding the actual system behaviour is very limited.

Even though in the discussed case the identification method delivered basically perfect results, one has to be aware that here the input for the identification was the transfer function from the analytical averaged model, which is the ideal case. For identification based on simulation results or laboratory investigations the results are with high probability more distorted. Even in the perfect case some states with low contribution to the input output behaviour have not a smooth trajectory for the elements in the B and C matrix. If the identification is applied to simulation data it is likely that the input data for the fits gets more noisy and additional methods need to be included to improve the fit quality.

To sum up, no noticeable deviation between the behaviour of the fits resulting from the identification methods and the fits from analytical model can be observed, even though the NRMSEs based on the matrix elements seem extremely high. The application of identification techniques on the theoretical transfer function data instead of directly using the theoretical values has no negative impact on the performance of the fitted models. If the approach still works if the identification is carried out on simulation data instead of the transfer function of the analytical model could not be investigated during the time of this thesis.

7. Results and Analysis Laboratory Experiments and Identification

7.1. General Remarks

In the previous subsections the state-space models came either directly from the analytical model or from identification applied to the transfer function calculated from the analytical model with a known number of poles. The identification methods delivered in general very good results and the transfer functions and step responses did not show significant errors. However, this is the ideal case and only applicable if the analytical model is known. In this section it is discussed how the identification can be applied on data from laboratory measurements. The transfer functions are obtained by the procedure described in section 5. The measurement results, limitations and possible solutions are discussed in the following subsections. The identified models are verified by a comparison of the step responses with measured data.

7.2. Measurement Results

The input perturbations were applied for all seven inputs with 100 different perturbation frequencies leading to 700 test sequences. One of them is shown as an example in Figure 7.1 for a perturbation of p^* at 0.1 Hz.

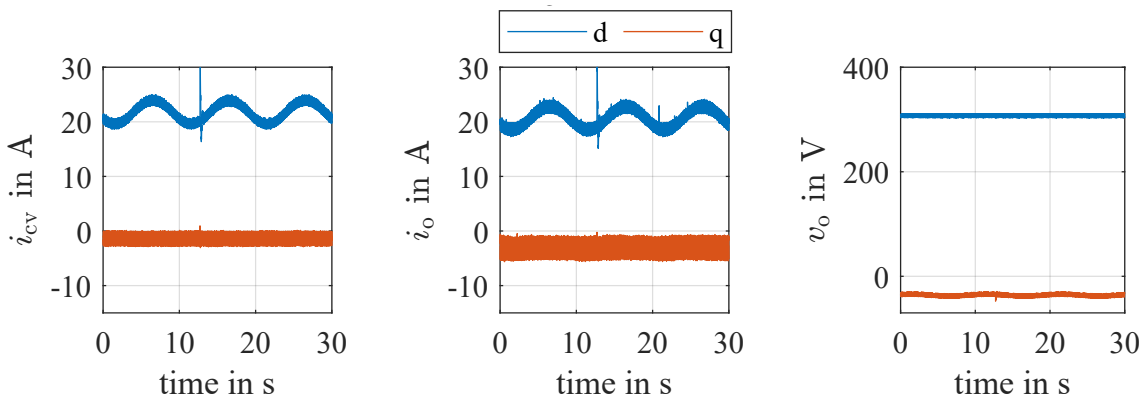


Figure 7.1: Output measurements for a perturbation on p^* with 0.1 Hz

It can be seen that the measurements contain noise and some measurement errors. For this input the output perturbation is clearly visible in $i_{cv,d}$ and $i_{o,d}$. A perturbation can also be observed in $v_{o,q}$. For the other inputs the perturbation is barely visible or not visible at all. In those cases the perturbation could not be detected by the algorithm used in this thesis to select the correct frequency component from

the Fourier spectrum. A different algorithm with a more differentiated plausibility check will possibly enable to also capture smaller perturbations but also this will only work until a certain extent which is limited by the precision of the sensors. In the analytical model it can be seen that the output perturbation in the other three outputs is 30 to 50 dB smaller than in the three visible ones. 30 dB equal a factor of 32. This leads to a main difference of the laboratory tests compared to the simulation. To evoke a clear perturbation also in the remaining outputs, the input perturbation would have to be increased significantly. Then the perturbation in the current would unavoidably get extremely large and the converter would eventually trip. Therefore, it has to be accepted that for some input output combinations no transfer function data can be calculated.

Figure 7.2 shows another example of a test sequence. A perturbation in $v_{g,d}$ at 10 Hz is displayed. The $v_{g,d}$ is the only input which causes a perturbation in $v_{o,d}$. It can be seen that this time all outputs are visibly affected by the perturbation except from $v_{o,q}$. Furthermore, it can be seen that a major share of the noise is caused by oscillations. In the Fourier transform they show up at 300 Hz in the dq-frame which equals a fifth or seventh order harmonics in the stationary reference frame.

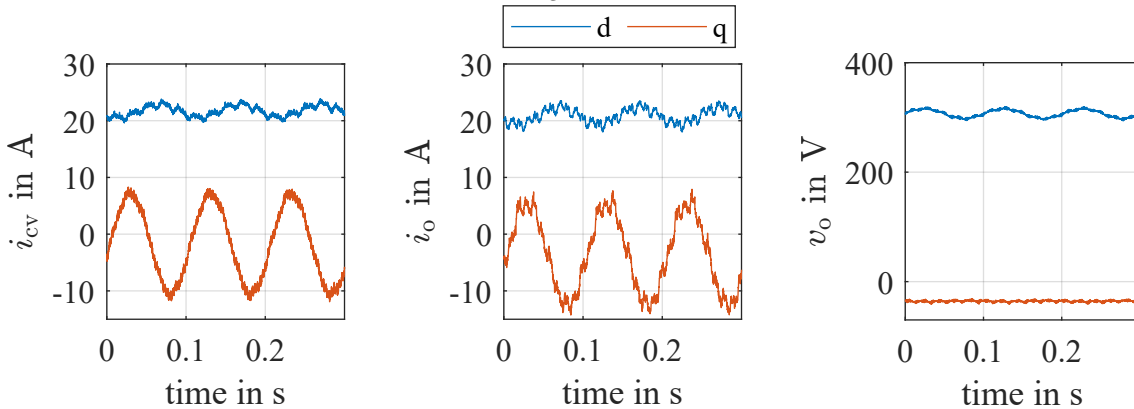


Figure 7.2: Output measurements for a perturbation on $v_{g,d}$ with 10 Hz

Figure 7.3 shows a perturbation in v^* at 210 Hz and Figure 7.4 shows a perturbation in v^* at 1.3 kHz. It can be seen that the perturbations in the current become more irregular at higher frequencies and the sampling at 10 kHz makes them look more edged. Still, it is possible to calculate the Fourier transform and determine the amplitude of the fundamental wave. Since the input perturbation is logged with the same frequency, the signal is distorted in the same way as the output and the ratio should in theory still be accurate. Moreover, in the figures only three periods are displayed while a whole second is logged and processed so that the distortions should average out across the measurement period. What can be nicely seen in Figure 7.4 is the effect of the filter on i_{cv} . While the 1.3 kHz perturbation is visible in i_o , it is filtered out in i_{cv} . In this case there is actually a very small

perturbation also in $v_{t\text{exto},d}$ but as the amplitude is smaller than the one of the noise the selection of the frequency component in the Fourier spectrum that was applied in this thesis does not capture it.

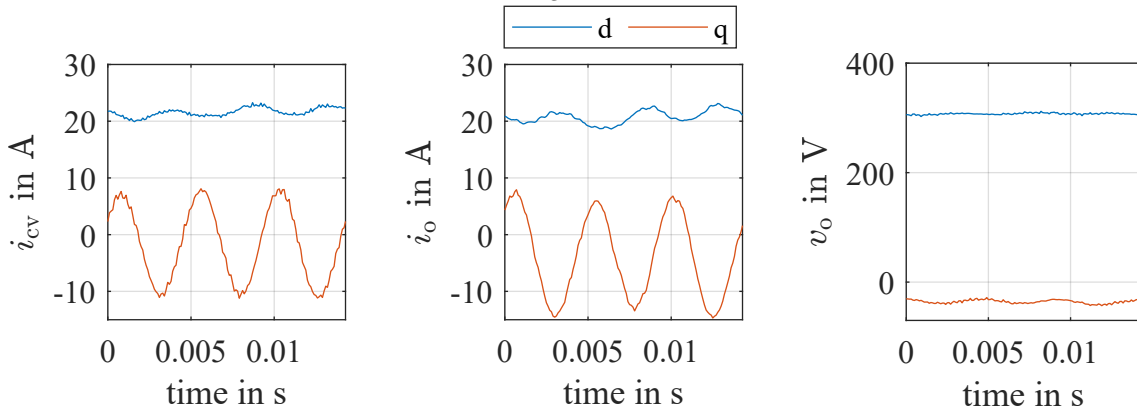


Figure 7.3: Output measurements for a perturbation on v^* with 210 Hz

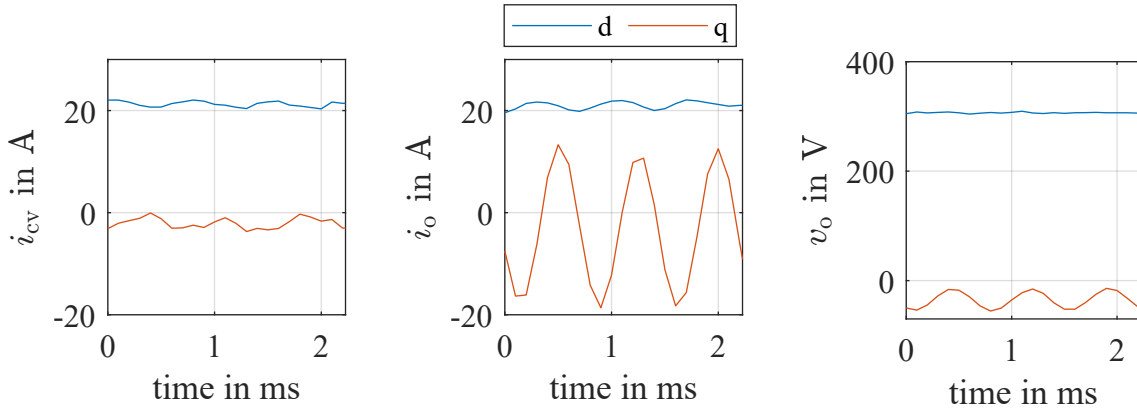


Figure 7.4: Output measurements for a perturbation on v^* with 1.3 kHz

For p^* data could be acquired down to -70 dB at 120 Hz with the frequency dependent adaption of the perturbation amplitudes described in section 5. For ω_g data is available up to 400 Hz and and for ω^* up to 57 Hz. For q^* and v^* even without an adaption of the amplitude, perturbations could be measured up to 2.9 kHz. For $v_{g,d}$ and $v_{g,d}$ data could be acquired up to 1.5 kHz. Three examples of calculated frequency response data in comparison to the analytical model can be found in Figure 7.5. The full data set is presented in Figure I.1 and Figure I.2.

It can be seen that in the three examples in Figure 7.5 the measured values match the data from the analytical model very well for small frequencies, both in phase in amplitude. For higher frequencies there is a mismatch. The characteristics in the magnitudes like the filtering effect in the converter current are recognisable, but the phase deviates significantly at higher frequencies. Decreasing phase at constant magnitude is typical for a delay. In the analytical model no delays are modeled, which is why it was expected that it is not a good approximation at high frequencies. In the lab, the origin of the delay is most the communication between the real time

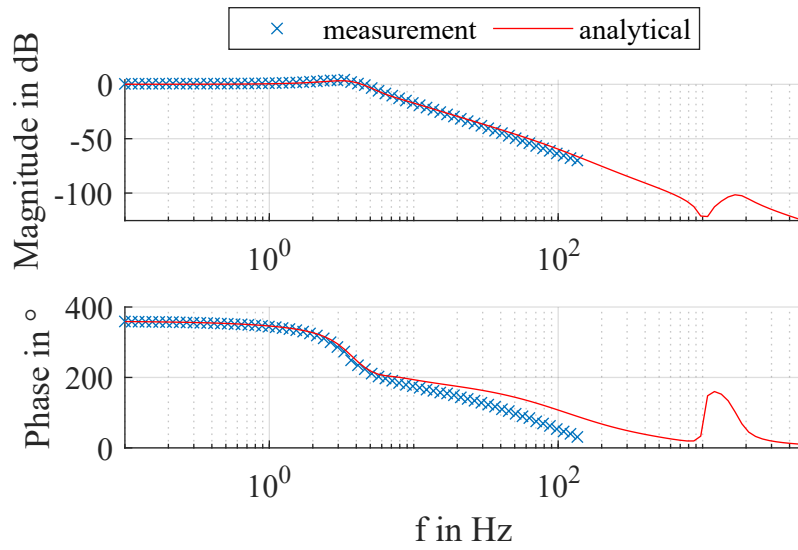
system and the converter and vice versa. An attempt was made to include delays in the simulation model but the try to match the laboratory system was not successful. Since the matching of the model to the laboratory setup did not have priority no further attempts were made to include the delays. For future work the algorithm to select the frequency component from the Fourier spectrum should be improved.

In Figure 7.5 the mismatch can be explained by the missing delays in the analytical model. However, there are input output combinations where there is a strong mismatch between the measurements and the analytical model also at low frequencies. Possible reasons are that not all laboratory parameters are exactly known such as the exact impedance of the connections between the components, the exact parameters of filter and transformer or the parameters of the inner control loop in the converter. The errors appear on the one hand in the combination of $v_{g,d}$ and the d-components of the current while the q-component of the current matches well. This is displayed in Figure 7.6. It can be seen that the mismatch appears in the magnitude only. On the other hand there is a mismatch of the relation of $v_{g,q}$ and the q-component while the d-component matches well. The data looks similar to the one displayed in Figure 7.6. This indicates that there might be some resistance included that is not modeled in the analytical model even though the suspicion could not be verified. The attempts to identify the unknown parameters based on the measurements and adapting the analytical model accordingly as well as swiping certain parameters to get a better match with the measurements were not successful. A comparison between the analytical model or simulations in the following is therefore only to a certain extent meaningful.

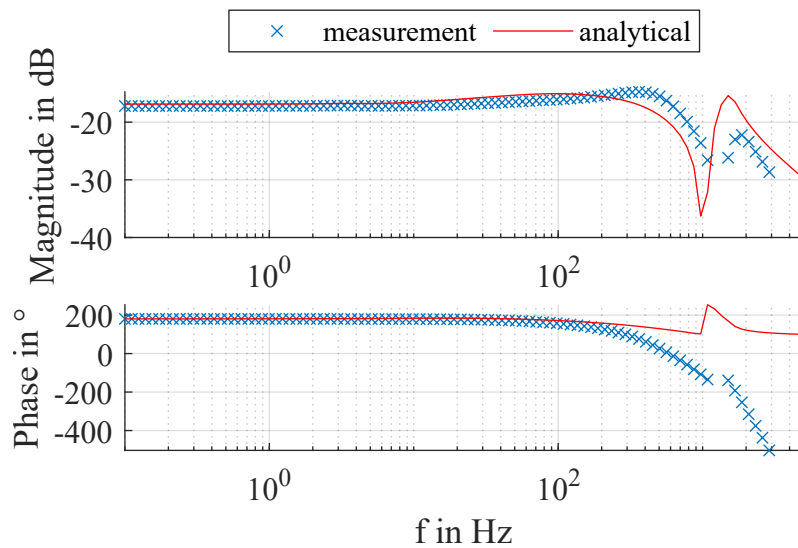
7.3. Identification Results

In this section the identification for four different methods or variants are discussed: vector fitting with and without substitute values, N4SID and N4SID in combination with PEM. The identification results are discussed on the examples displayed in Figure 7.7. The bode plots of all input output combinations can be found in the appendix.

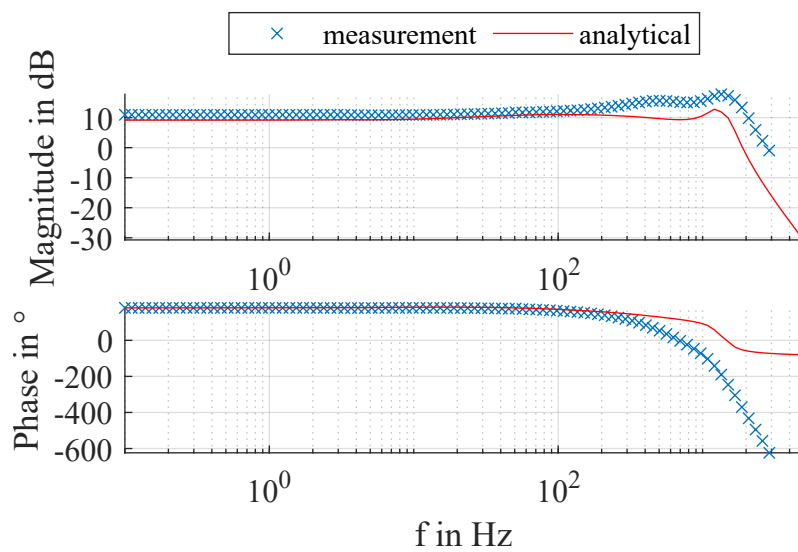
It can be seen that none of the identified models matches the measurements or the substitute values in all discussed input output combinations optimally. However, in many cases the identification methods approximate the measurements well. In Figure 7.7a it can be seen that in the frequency range up to 10 Hz all models except the one from N4SID without PEM fit the measurements comparably well. The variant with vector fitting and substitute values seems to fit the measurements best. Above 20 Hz the amplitudes get very small, but it can be seen that the VF with



(a) Transfer function data calculated from measurement and analytical model for p^* to $i_{cv,d}$



(b) Transfer function data calculated from measurement and analytical model for q^* to $i_{cv,q}$



(c) Transfer function data calculated from measurement and analytical model for v^* to $i_{o,q}$

Figure 7.5: Transfer function data calculated from measurement and analytical model for three example transfer functions

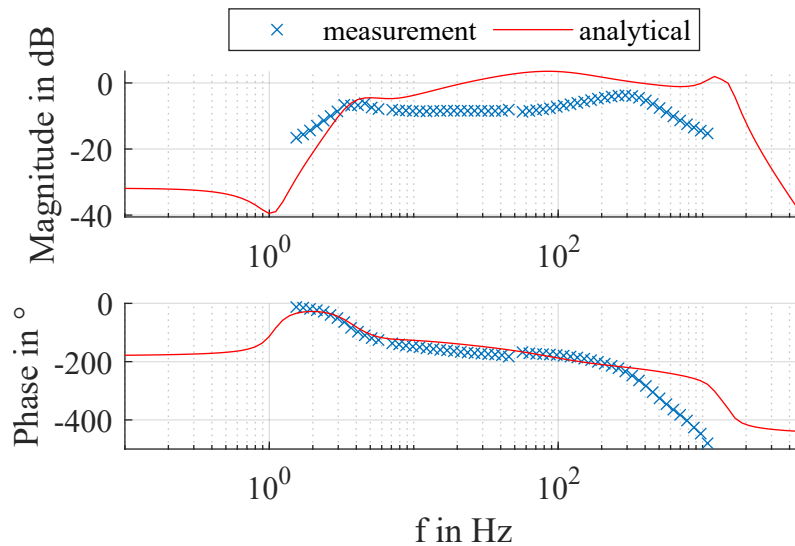


Figure 7.6: Transfer function data calculated from measurement and analytical model for $v_{g,d}$ to $i_{cv,d}$

substitute values has a higher relative accuracy than the other methods between 10 Hz and 100 Hz. It seems that the use of the extrapolated data is beneficial for the VF. Even though N4SID and PEM have the same extrapolated data as input, the accuracy at these small magnitudes is lower. After 100 Hz where no data is measured N4SID and PEM displays the characteristics of magnitude and phase of the extrapolated better than the VF with substitute values, even though the magnitude is not correct. The VF without substitute values did not get any input data there so it is not meaningful to compare it in that range.

For Figure 7.7c both N4SID and N4SID with PEM fit the behaviour best. VF both with and without substitute values does not match the system behaviour for this input output combination for frequencies higher than 300 Hz.

For Figure 7.7b N4SID has a significant error at low frequencies. However, the peak at 2 kHz is visible, even if it is at the wrong magnitude. The characteristics of the filter are modelled best by the variant with PEM even though the phase does not match which is most likely because of the absence of the delay. Up to 1 kHz the both VF variants match the measurements best, but the peak at 2 kHz is not detected. It seems favourable to apply N4SID in combination with PEM, since in this case the results could be improved significantly.

In Figure 7.8 N4SID and N4SID with PEM do not fit the transfer function data well. The VF results on the other hand fit the measurements almost perfectly between 2 and 400 Hz and the VF with substitute values matches the substitute values very well up to 400 Hz. The fit of the vector fitting to the measurements between 1.1 Hz and 3 Hz could be improved compared to the version without substitute values.

The deviations in the VF variants appear mostly at higher frequencies and most often when measurements at high frequencies are available. This could be an indicator that the VF does not handle the phase shift due to the delay well. It is necessary to explore this further to make a statement about it though. Depending on the quality of the substitute values the use of substitute data from input output wise identification can increase the quality of VF. In the final results presented here, the substitute values only have a small impact but during the course of the work it became apparent that their application can cause a much better fit. The final substitute values were chosen to make N4SID work in the best way and due to the time limitation of the thesis work the impact of the substitute values on VF could not be analysed more in depth. However, this seems promising and should in the opinion of the author be explored in further work.

An interesting point that can be seen in the plot of all input output combinations is that N4SID produces transfer functions also for the input output combinations where the input data was set to zero. Vector fitting on the other hand sets the corresponding elements in the C-matrix to zero. The N4SID variants produce changes in the outputs, where this should not be the case. In the step responses it becomes visible that the amplitude of these outputs can be relatively high. This could be an disadvantage of N4SID. On the other hand it could also become an advantage. This missing input output combination is missing because of the missing measurement due to the low gain. However, it is not decoupled from the rest of the system, so it is physically meaningful that it is not exactly zero. This could not be further examined in this thesis but seems interesting for subsequent considerations.

7.4. Validation by Step Response

All step responses were measured and visually inspected in comparison to the identified models. Figure Figure 7.9 shows the measurement and the identified models for a step in p^* of 2000 W as an example. The identification models created with VF match the measurement well. There is only a small difference between the VF methods and the N4SID in most cases, however in some cases the deviation between the measurement and the N4SID is very large such as in Figure 7.10. There $i_{extcv,d}$ is displayed for a step of 5 V in $v_{g,d}$. This is the input output combination as in Figure 7.8. In this case there is also a difference visible between the VF and the VF with substitute values. It seems like in this case the version without substitute data is actually better. In the corresponding bode plot the VF matched some measurement points better when substitute values provided. Also, the VF matched

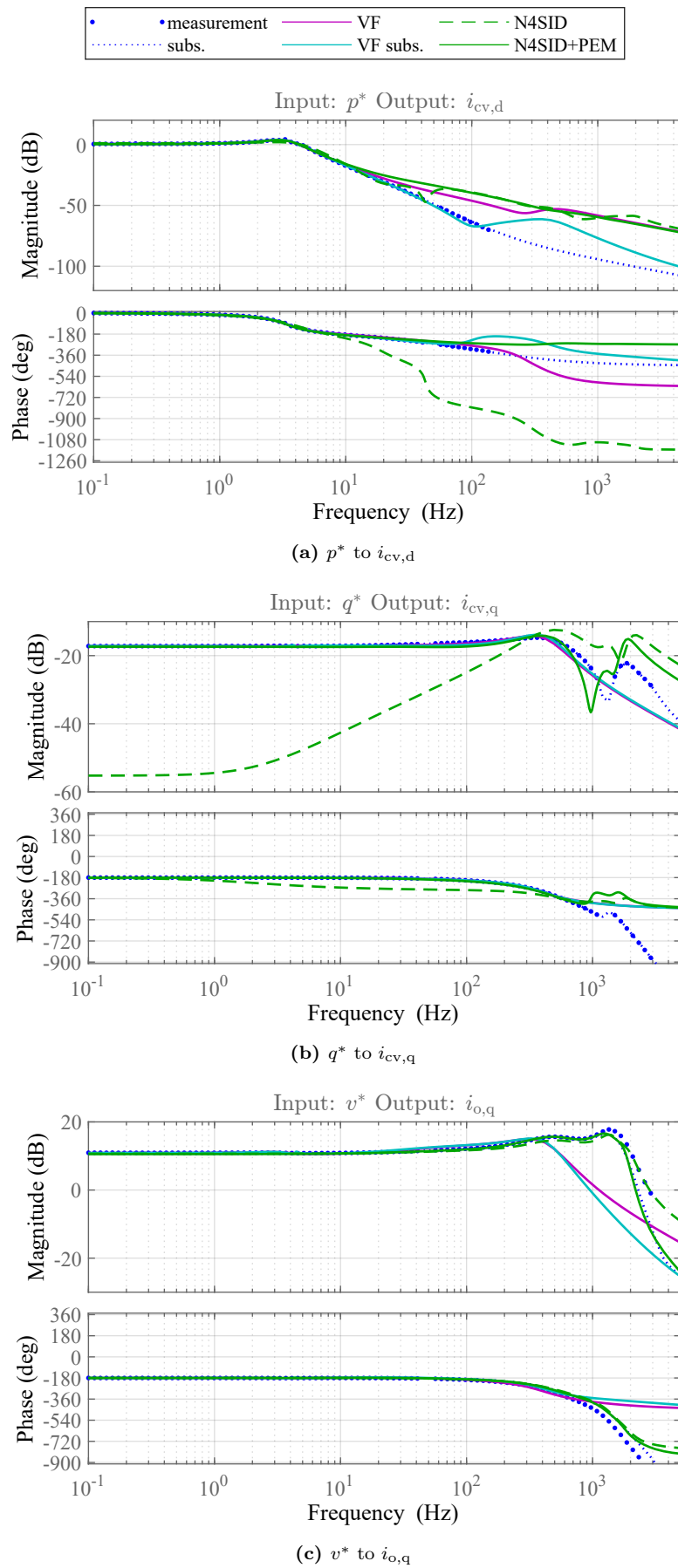


Figure 7.7: Bode plots of three example transfer functions for the measurement and the identification results with VF, VF on substitute values, N4SID and N4SID with PEM

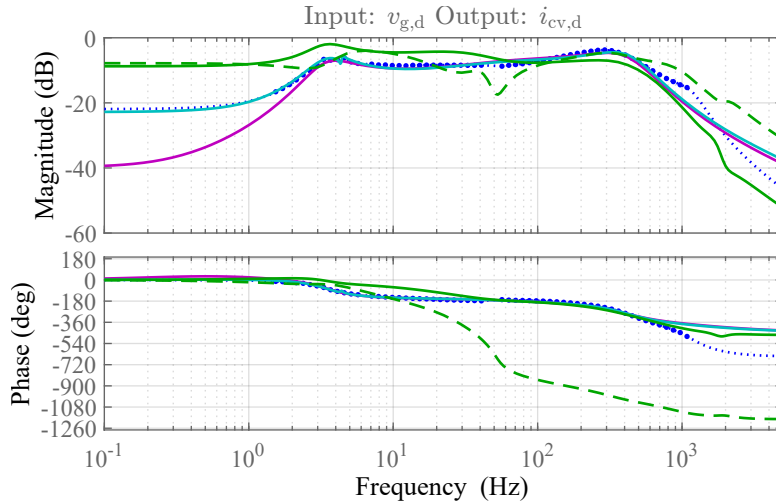


Figure 7.8: Bode plot of an example transfer function v^* to $i_{l,d}$ for the measurement and the identification results with VF, VF on substitute values, N4SID and N4SID with PEM

the provided substitute values in the low frequency range well, but it is always possible that the substitute values do not match the real behaviour on points where no data was measured. Obviously, in case substitute values are used, the quality of the substitute values is crucial for a good identification result.

7.5. Summary

To sum up, it was possible to identify models based on the laboratory measurements using two different identification techniques. In the frequency domain N4SID with PEM seems to perform better at high frequencies than VF. However, there are input output combinations where N4SID even in combination with PEM performs poorly. An explanation for this behaviour cannot be given at this point. VF seems more reliable for frequencies up to 300 Hz and in the step response VF matches the measurements always equally well or significantly better than N4SID with PEM. Even though a complete data set it is not strictly necessary for the algorithm to work, the VF could potentially be improved by using substitute values. However, the choice of substitute values has to be done carefully since an extrapolation of measurement data to points where no data could be measured might inject mistakes into the input data. The generation of substitute values might be an interesting point for further work. A topic that should be explored further is the effect of time delays on the identification.

It has to be underlined, that the statements comparing VF and N4SID are not generalized since only one model and one operating point was considered. A tuning of parameters such as the pole tolerance in the VF, another weighting scheme for VF, other substitute data or a weighting filter in the N4SID could lead to significantly

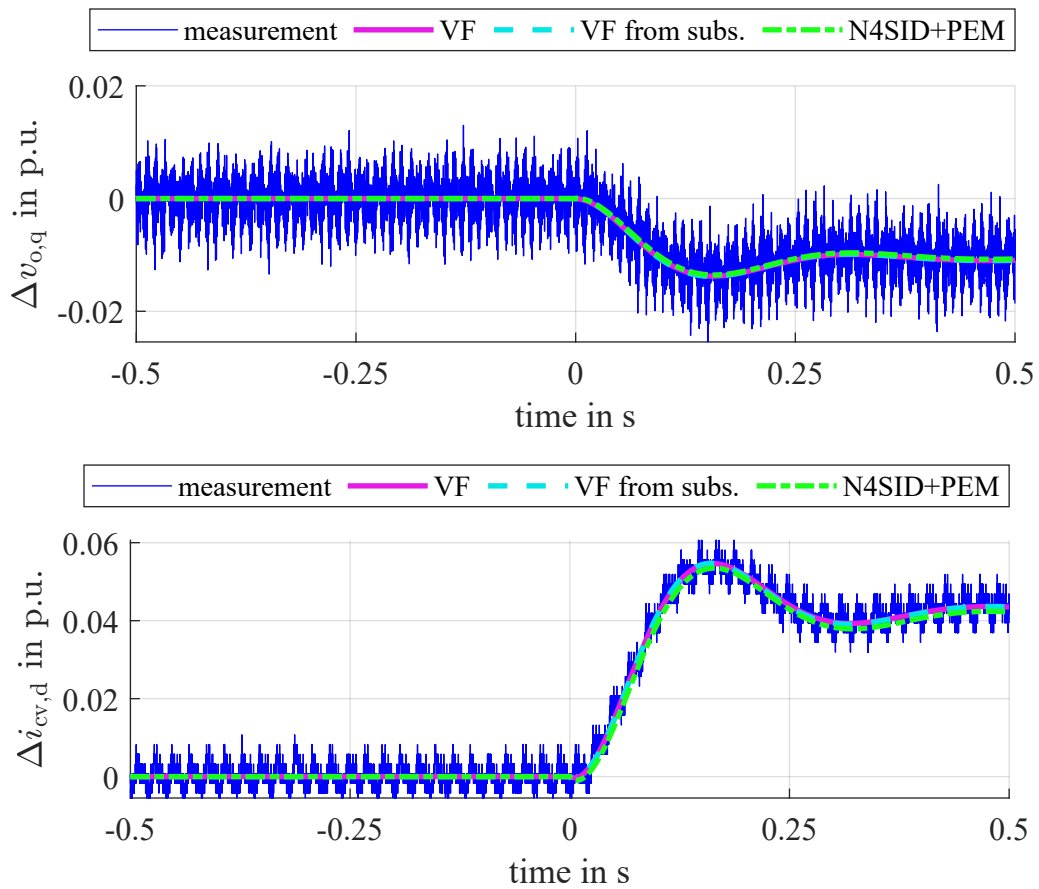


Figure 7.9: Step response to a step of 2000 W in p^* measured, calculated and from identified model

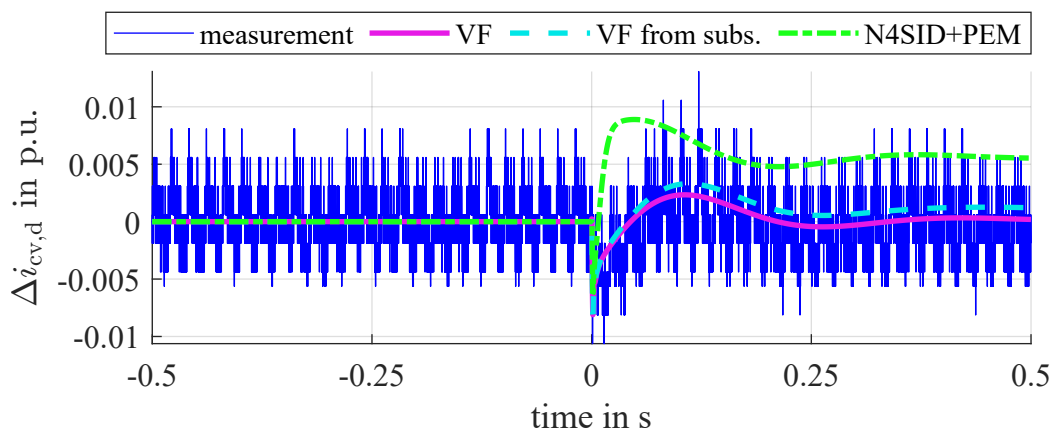


Figure 7.10: Step response to a step of 5 V in $v_{g,d}$ measured, calculated and from identified model

different conclusion. Also, the system order provided to the N4SID will most likely have an important impact. Different variants of input data, such as up-sampled substitute values also affects the quality of the identification. All these options could not be explored in this thesis, which is why the author does not want to make a statement on which method is most favourable for the application with measurement data. However, the general impression gained from the examined example is that the VF method is more beneficial than N4SID.

8. Summary of Results

The thesis work explored two different aspects in particular. In the first part a method was developed to create an input dependent state-space model from input swipes on a linearized analytical state-space model. The process of fitting a model from input sweeps was discussed in detail. The method to create a uniform system representation, which is a fundamental precondition to make a fitting possible, was presented. The code to generate fits from a set of state-space models for different operating points is provided for further use. First, the swipes were carried out for each input separately. Dependencies of pairwise combined input variations were discussed and the possibility to approximate them by a superposition of the fits for each individual input was analysed. The fits and the approximation were evaluated in time and frequency domain analyses and discussed in detail. Finally, by combining all individual input dependencies by superposition an approximation for a dependency on all input variations was created. The model was evaluated in the time domain by assessing the step responses.

The conclusion that was drawn for the examined VSM was that all individual input dependencies can be fitted by a quadratic function for each matrix element. For some inputs the fitting created errors in the frequency domain that seemed of minor importance but the impact has to be assessed in regard to the specific use case. For some combination of input variables, the model behaviour can be approximated well by superposition of the individual input dependencies. For other combinations the quality of the approximation decreases significantly with the difference to the default operating point for the swipes. To increase the reliability of the approximation it is recommended to perform the input swipes for multiple default operating points and limit the operating range in which the model is used. The range should be chosen depending on the use case and the necessary accuracy of the approximation.

In the second part laboratory experiments were carried out to obtain data for an identification of the laboratory system. It was presented how the measurements were carried out. An important difference between simulation and laboratory experiments was pointed out, namely that due to physical constraints in the lab certain input output measurements or frequency ranges can possibly not be measured. It was discussed in detail which impact an incomplete data set has on two different identification methods. It was possible to identify models and validate them by comparison of the step response with the step response measured in the lab.

In the frequency domain at frequencies lower than 300 Hz and in the step response the vector fitting approach delivered very satisfying results. At higher frequencies the fit in the bode plot seemed improvable and a possible correlation with time delays

that become most effective at high frequencies was pointed out. The approach with N4SID and PEM was in general less accurate and more dependent on the quality of substitute values than the VF but seemed to be able to match high frequency behaviour better in some case. Moreover, the N4SID and PEM method predicted outputs where there should not have been one since transfer function data of zero was passed to the algorithm for that combination at all frequencies.

Due to the fact that only one operating point for one sample model was considered and a thorough analysis of the impact of different parameter settings on the identification methods could not be provided, the author does not want to make a generalized statement on which identification method is better for this use case. However, the impression that came up during the work was that the vector fitting approach was more beneficial in regards to accuracy, sensitivity to the input data quality and the need of substitute values for missing data. Two aspects that seemed worth of further investigation were the possibility to improve the fitting by the application of substitute values or up-sampled data and the impact of time delays on the identification result.

9. Discussion and critical consideration

9.1. Importance of efficient state-space identification methods

Eigenvalue based modal analysis is the small-signal analysis method that delivers the most informational value on stability and harmonic oscillations. In contrast to impedance-based methods, the system behaviour is assessed globally for the whole system. A big disadvantage on the other hand is that detailed models are necessary. For large systems and black-boxed models, state-space models are hard to derive due to the lack of detailed information [41, 16, 18]. Therefore, it might seem questionable if the application in multi-vendor systems is feasible at all. The methods that are currently applied for interaction studies between the systems of different manufacturers rely widely on time domain analysis seem sufficient for the projects that have been realized so far. However, the number of involved converters has been small. The number of operational cases that need to be considered grows exponentially with the number of converters in close proximity to each other, which can be high taking the plans to build complex HVDC grids into account. Hence, the consideration of state-space methods seems unavoidable. Another advantage of the state-space representation is that complex systems can be assembled relatively easily from the state-space models of the single components.

Numerous methods for the identification of linear state-space model exist, but the generation of input data from simulations is computationally expensive as it requires extensive time domain simulations. Moreover, they are only valid for small deviations around the specific operating point they were created for. The identification needs to be carried out for each operating point. Therefore, the benefit of the identified models for the stability analyses does not outweigh the effort to create them. Methods are needed to make the model identification process more efficient and state-space analyses more feasible.

9.2. Operating point dependently fitted model

In small-signal analyses of AC systems, a linearization is done for each considered operating point, so that the question arises why this is not done for power electronic systems as well. For systems with mainly large synchronous generators the analytical equations to describe the relevant system behaviour are widely known, why a linearization is comparably easy. To carry out an identification for each operating point is as mentioned before associated to a high computational effort, mainly due to

the simulations necessary to generate the input data for the identification methods. Therefore, the possibility to inter- and extrapolate system behaviour from few operating points to a wider operating range would be highly valuable. An approach to do this by fitting a state-space model from input sweeps was developed and explored in this thesis for the example of a VSM.

It was possible to fit the dependency of the state-space model on each individual input by a quadratic function. It was also possible to fit the model by superposition with very good approximation for some input output combinations over the full area of their input ranges and for others in proximity to the default operating point. However, the results have to be considered critically, as discussed in the following.

The operating range for the full input dependent model could be only roughly scanned because of high number of possible input combinations. Moreover, it is possible, that the frequency domain analyses and discussions in the form they were done did not uncover all relevant phenomena. Another metric that weights the magnitude errors and does not evaluate punctual but maybe also with a NRMSE would probably have been better for a more elaborate discussion and detection of deviations and problems in the fits.

Moreover, the investigations were carried out on an analytical averaged model, so that the input data was highly ideal. Likewise, the version with models from identification was based on transfer function data from the analytical, so that also there the input data still had very good quality. In the used models, no switching phenomena and no time delays are included. Moreover, the investigations limited to AC side behaviour. MMC specific problems are not considered at all. It is therefore highly likely that the results would in reality be a lot more defective. If the approach is still viable if the named points are included is questionable.

Furthermore, the validation of the fitted system is a major problem. In this thesis a lot of additional operating points were assessed in the frequency and time domain. In reality the frequency domain analysis is not feasible as the model is still black-boxed and a identification of additional operating points just for validation would erase the added value of the fitting. Validation of the fits is only possible based on the goodness of fit for each individual fitted element, by comparison of frequency responses with the input data for the fits or in additional time domain simulations. When validating the fit quality only based on the input data, phenomena between the input operating points might get undetected. If additional time domain simulations are carried out for validation the added value of the fitted model decreases.

One drawback in the approach to fit each matrix element separately was that the fit quality was hard to assess by looking at the goodness of fit for each element.

The goodness of fits of the matrix elements is not easy to interpret in regards to the system behaviour. Even small deviations in the fit can cause large deviations in frequency and time domain. Especially when the fitting is done based on identified models, it would be useful to develop a metric that evaluates the fit quality in a better way that takes the importance of a certain element and its effect on the system behaviour into account. One idea would be to weight the NRMSE with the HSV of the corresponding state. An alternative to fitting all matrix elements in the state-space model could be to calculate the transfer function and fit the poles and residues. The fitting of poles is similar to the fitting of the eigenvalues but the fit results for the residues could possibly be easier to interpret than the combination of elements in the B and C matrix. Also there it should be considered to use a metric that weights the goodness of fits.

Another weakness of the approach is that in complex systems, phenomena might appear between two operating points that are used for the fitting and might remain undetected. Moreover, depending on the needed accuracy swipes a high density of input values in the swipes and fitting for many default operating points might be necessary to obtain the desired results. It is questionable if the effort is then still worth it since the added value of the approach decreases with the number of identifications that need to be carried out.

Despite of the discussed obstacles, a starting point for further research activities into this direction was laid out and especially the explanations for the generation of a uniform system representation which is crucial for an interpolation of any kind seem valuable.

9.3. Laboratory Experiments and Identification

In practice it is usually effortful and costly to carry out real-world measurements compared to simulations. It is therefore often favourable to carry out most investigations in simulation. Still, there might be some cases where either no simulation model is available or a test environment with a real time system is set up either way. Currently the recommendation for interoperability studies is to perform SiL and HiL investigations, so that the necessary setup for the measurements performed in this thesis is there anyways. In those cases it might be an favourable option to perform the identification on the real-time measurements instead of simulation data, since the simulations are computationally expensive and time intensive, especially for the measurements at low frequencies. Identification from laboratory measurements could thus bring a time saving compared to simulations since the measurements are

carried out in real-time. Therefore, the identification of real-world measurement data could be relevant.

It was discussed that in the lab, measurement of frequency response data is limited by the physical constraints. Some more data than the measurements used in this thesis could be made available by improving the calculation of the perturbation amplitudes from the measurements. In this thesis the selection of the frequency component in the Fourier spectrum with the highest amplitude compared with the check on right frequency excluded small output perturbations that theoretically could have been used. Moreover, a more careful adaption depending of the input perturbation amplitude on the output perturbation will also increase the range of measured data. Still, there will likely be input output combinations with a transfer function with low gain and output perturbation amplitudes smaller than the tolerance of the sensors.

During the course of this thesis the parameters of the used identification methods could not be tuned and a thorough explanation for the observed errors could not be given. Furthermore, a fixed model order obtained from the analytical model was provided to N4SID, when it would have been methodically more appropriate to perform a system order estimation from the measurements and use the result instead. A generalized statement about the comparison of the two applied algorithms could not be provided due to the limited extend of the investigations. Therefore, it has to be underlined that all analyses only refer to the examined model and operating point and the results could change significantly with the adaption of setup, parameterization or even operating point.

Nevertheless, the experiments gave useful insights on the peculiarities of the frequency response estimation and identification from laboratory data. Especially the model identified by vector fitting performed very satisfactory in the validation by time domain analyses, which can be seen as a success.

10. Final Conclusion

To sum up, the thesis provides many new insights relevant for the establishment of methods that enable feasible state-space analyses for black-boxed power electronic systems. A new approach of fitting a state-space model from input sweeps and creating a full model by superposition of the individual dependencies was developed, tested and discussed in detail. Also, laboratory measurements were carried out and state-space model identification based on experimental data was presented. The model identified by a new vector fitting method delivered overall satisfactory results with potential for improvement. New findings about constraints when performing identification on disturbed and incomplete data sets were pointed out.

All in all, methods were presented that can be useful for further research activities in model identification of power electronic systems, even though they have to be elaborated more before they are applicable in practice. The work is part of research of international importance and helps to enable the successful realization of power converter dominated systems and multi-terminal multi-vendor grids.

11. Outlook

Further research can be built on the findings of this thesis. The developed methods and the code for the fitting were made available and can be used and adapted. Steps that could follow up on the fitting related thesis work are further validation of the fitted non-linear models. The fitting method should be tested with identification from simulation data and in a next step used with more complex models where for example DC side control, switching phenomena, delays or MMC related phenomena are taken into account. Moreover, the fitted models could be used in a sample interaction study to test if the quality of the approximation is sufficient to detect interaction phenomena. Furthermore, it should be explored how the method can be adapted, for example by performing sweeps for multiple default operating points for example.

Research that could follow up on the laboratory tests could deal with methods to improve the input data of the identification, investigate the use of substitute values and finally test if a combination with the fitting method is possible.

Altogether, state-space small-signal analyses of black-boxed models could become more feasible in the future and could pave the way for interoperable converter dominated systems and multi-terminal multi-vendor HVDC grids.

References

- [1] International Renewable Energy Agency, “Global renewables outlook: Energy transformation 2050”, Abu Dhabi, 2020. Available online https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2020/Apr/IRENA_Global_Renewables_Outlook_2020.pdf
- [2] International Renewable Energy Agency, “Future of wind: Deployment, investment, technology, grid integration and socio-economic aspects (A Global Energy Transformation paper)”, Abu Dhabi, 2019. Available online <https://www.irena.org/publications/2019/Oct/Future-of-wind>
- [3] Website of the Federal Government | Home page, “European cooperation for offshore wind power | Federal Government”, 16.09.2022. Available online <https://www.bundesregierung.de/breg-en/news/wind-power-cooperation-2040980> (Last access 16.09.2022).
- [4] Department for Business, Energy & Industrial Strategy, “Ten Point Plan for a Green Industrial Revolution”, 2020. Available online <https://www.gov.uk/government/publications/the-ten-point-plan-for-a-green-industrial-revolution>
- [5] Minister, Office of the Prime, “Ambitious offshore wind initiative”, 14.05.2022. Available online <https://www.regjeringen.no/en/aktuelt/ambitious-offshore-wind-power-initiative/id2912297/> (Last access 16.09.2022).
- [6] WindEurope, “Wind Energy in Europe - 2021 Statistics and the outlook for 2022-2026”, Brussels, 2022. Available online <https://windeurope.org/intelligence-platform/product/wind-energy-in-europe-2021-statistics-and-the-outlook-for-2022-2026/>
- [7] “First power from Hywind Tampen”, 2022. Available online <https://www.equinor.com/news/20221114-first-power-from-hywind-tampen> (Last access 09.03.2023).
- [8] A. Durakovic, “BREAKING: Norway Opens First Offshore Wind Tenders for Applications”, *Offshore Wind*, 29.03.2023. Available online <https://www.offshorewind.biz/2023/03/29/breaking-norway-opens-first-offshore-wind-tenders-for-applications/> (Last access 30.03.2023).
- [9] “The gotland hvdc link | Hitachi Energy”, 21.11.2022. Available online <https://www.hitachienergy.com/de/de/about-us/case-studies/the-gotland-hvdc-link> (Last access 21.11.2022).

-
- [10] ENTSO-E, “Voltage Source Converters”, 21.11.2022. Available online <https://www.entsoe.eu/Technopedia/techsheets/voltage-source-converters> (Last access 21.11.2022).
- [11] CIGRE Working Group B4.52, “HVDC Grid Feasibility Study”, 2013. Available online <https://e-cigre.org/publication/533-hvdc-grid-feasibility-study>
- [12] Bundesministerium für Wirtschaft und Klimaschutz, “BMWK und ÜNB veröffentlichen Pläne zur Vernetzung von Offshore-Windparks in der Nordsee”, 09.03.2023. Available online <https://www.bmwk.de/Redaktion/DE/Pressemitteilungen/2023/02/20230227-bmwk-und-uenb-veroeffentlichen-plaene-zur-vernetzung-von-offshore-windparks-in-der-nordsee.html> (Last access 09.03.2023).
- [13] eurostat, “EU imports of energy products - recent developments”, Luxembourg, 20.12.2022. Available online https://ec.europa.eu/eurostat/statistics-explained/index.php?title=EU_imports_of_energy_products_recent_developments&oldid=554503#Main_suppliers_of_natural_gas_and_petroleum_oils_to_the_EU (Last access 06.02.2023).
- [14] “ENTSO-E Position on Offshore Development”, 2020. Available online <https://www.entsoe.eu/outlooks/offshore-development/>
- [15] N. Hatziaargyriou, J. Milanovic, C. Rahmann, V. Ajjarapu, C. Canizares, I. Erlich, D. Hill, I. Hiskens, I. Kamwa, B. Pal, P. Pourbeik, J. Sanchez-Gasca, A. Stankovic, T. van Cutsem, V. Vittal, C. Vournas, “Definition and Classification of Power System Stability – Revisited & Extended”, *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3271–3281, 2021.
- [16] C. Buchhagen, C. Rauscher, A. Menze, J. Jung, “BorWin1 - First Experiences with harmonic interactions in converter dominated grids”, in *International ETG Congress 2015; Die Energiewende - Blueprints for the new energy age*, 2015, pp. 1–7.
- [17] J. Lyu, X. Cai, M. Amin, M. Molinas, “Sub-synchronous oscillation mechanism and its suppression in MMC-based HVDC connected wind farms”, *IET Generation, Transmission & Distribution*, vol. 12, no. 4, pp. 1021–1029, 2018.
- [18] C. Yin, X. Xie, S. Xu, C. Zou, “Review of oscillations in VSC-HVDC systems caused by control interactions”, *The Journal of Engineering*, vol. 2019, no. 16, pp. 1204–1207, 2019.
- [19] “ENTSO-E Position on Offshore Development: Interoperability”, 2021. Available online <https://www.entsoe.eu/2021/01/25/third-entso-e-position-paper-on-offshore-development-focuses-on-interoperability/>

-
- [20] P. R. O. Despouys, “First recommendations to enhance interoperability in HVDC-VSC multivendor schemes: Deliverable n^o: 4.3”, Paris, 2016. Available online <http://www.bestpaths-project.eu/en/publications>
- [21] Tennet TSO GmbH, “MIGRATE - The Massive InteGRATION of power Electronic devices: Enabling the energy transition by providing solutions for the technological challenges”, Bayreuth, 2019. Available online <https://www.h2020-migrate.eu/downloads.html>
- [22] P. Rault O. Despouys, “D9.3: BEST PATHS DEMO#2: Final Recommendations For Interoperability Of Multivendor HVDC Systems”, Paris, 2018. Available online <http://www.bestpaths-project.eu/en/publications>
- [23] Community Research and Development Information Service, “HVDC-based grid architectures for reliable and resilient WIdeSprEad hybrid AC/DC transmission systems”, Luxembourg, 15.09.2022. Available online <https://cordis.europa.eu/project/id/101075424> (Last access 01.01.2023).
- [24] READY4DC, “Home - READY4DC”, Aachen, 2022. Available online <https://www.ready4dc.eu/> (Last access 01.01.2023).
- [25] European Climate, Infrastructure and Environment Executive Agency, “Successful kick-off of InterOPERA Horizon Europe offshore electricity grids project”, 20.01.2023. Available online https://cinea.ec.europa.eu/news-events/news/successful-kick-interopera-horizon-europe-offshore-electricity-grids-project-2023-01-20_en (Last access 30.03.2023).
- [26] A. Tang, “EU and Industry launch new project charting the way for interconnected offshore wind farms and energy islands”, *WindEurope asbl/vzw*, 19.01.2023. Available online <https://windeurope.org/newsroom/news/eu-and-industry-launching-new-project-charting-the-way-for-interconnected-offshore-wind-farms-and-energy-islands/> (Last access 30.03.2023).
- [27] S. D’Arco, A. G. Endegnanew, G. Guidi, J. A. Suul, “Interoperability of Modular Multilevel Converters and 2-level Voltage Source Converters in a Laboratory-Scale Multi-Terminal DC Grid”, in *2018 International Power Electronics Conference*, 2018, pp. 2003–2010.
- [28] H. Saad, S. Denetière, B. Clerc, “Interactions Investigations between Power Electronics Devices Embedded in HVAC Network”, in *13th IET International Conference on AC and DC Power Transmission (ACDC 2017)*, ser. IET conference publications. Red Hook, NY: Curran Associates Inc, 2017, pp. 55 (7 .)–55 (7 .).

-
- [29] CIGRE Working Group B4.41, “SYSTEMS WITH MULTIPLE DC INFEED”, December 2008. Available online http://e-cigre.org/publication/ELT_241_6-systems-with-multiple-dc-infeed
- [30] RTE international -, “Webinar sessions on HVDC interaction studies with EMT simulation tools - RTE international”, 2020. Available online <https://www.rte-international.com/webinar-sessions-on-hvdc-interaction-studies-with-emt-simulation-tools/?lang=en> (Last access 09.03.2023).
- [31] ENTSO-E, “Workstream for the development of multi-vendor HVDC systems and other power electronics interfaced devices”, March 2021. Available online <https://www.entsoe.eu/news/2021/06/21/entso-e-t-d-europe-and-windeurope-publish-a-joint-report-on-the-development-of-multi-vendor-hvdc-systems-and-other-power-electronics-interfaced-devices/>
- [32] M. Amin M. Molinas, “Small-Signal Stability Assessment of Power Electronics Based Power Systems: A Discussion of Impedance- and Eigenvalue-Based Methods”, *IEEE Transactions on Industry Applications*, vol. 53, no. 5, pp. 5014–5030, 2017.
- [33] J. Beerten, S. D’Arco, J. A. Suul, “Identification and Small-Signal Analysis of Interaction Modes in VSC MTDC Systems”, *IEEE Transactions on Power Delivery*, vol. 31, no. 2, pp. 888–897, 2016.
- [34] A. Garcés, Hrsg., *Modeling, operation, and analysis of DC grids: From high power DC transmission to DC microgrids*. London, United Kingdom and Sand Diego, CA and Cambridge, MA and Oxford, United Kingdom: Elsevier AP Academic Press, 2021.
- [35] A. Rygg, M. Amin, M. Molinas, B. Gustavsen, “Apparent impedance analysis: A new method for power system stability analysis”, in *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL)*. Piscataway, NJ: IEEE, 2016, pp. 1–7.
- [36] M. K. Bakhshizadeh, C. Yoon, J. Hjerrild, C. L. Bak, L. H. Kocewiak, F. Blaabjerg, B. Hesselbaek, “The Application of Vector Fitting to Eigenvalue-Based Harmonic Stability Analysis”, *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, no. 4, pp. 1487–1498, 2017.
- [37] “oceangridproject.no: New grid solutions for profitable offshore wind development”, 02.11.2022. Available online <https://oceangridproject.no/> (Last access 03.11.2022).

-
- [38] P. S. Kundur, Hrsg., *Power system stability and control*, ser. The EPRI power system engineering series. New York: Mc Graw Hill, 1994. Available online <http://www.loc.gov/catdir/enhancements/fy1012/93021456-b.html>
- [39] Charles Steinmetz, “Power control and stability of electric generating stations”, White Sulphur Springs, 01.07.1920.
- [40] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. van Cutsem, V. Vittal, “Definition and Classification of Power System Stability IEEE/CIGRE Joint Task Force on Stability Terms and Definitions”, *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, 2004.
- [41] “Interactions between HVDC systems and other connections: ENTSO-E guidance document for national implementation for network codes on grid connection”, Brussels, 05.03.2018. Available online https://docstore.entsoe.eu/Documents/Network%20codes%20documents/NC%20RfG/IGD-Interactions_between_HVDC_Controllers_final.pdf
- [42] J. Rocabert, A. Luna, F. Blaabjerg, P. Rodríguez, “Control of Power Converters in AC Microgrids”, vol. 27, pp. 4734–4749.
- [43] Kersic, Manuel and Müller, T and Lewis, E and Schaupp, Thomas and Denninger, R and Ernst, P and Reichert, S and Rogalla, S and Singer, R and Roscoe, A and Jalili, K and Dyško, A and Alvarez, A and Hong, Qiteng and Lens, Hendrik and Schöll, Christian, “Testing Characteristics of Grid Forming Converters Part I: Specification and Definition of Behaviour”, 2020.
- [44] S. D’Arco J. A. Suul, “Virtual synchronous machines — Classification of implementations and analysis of equivalence to droop controllers for microgrids”, in *2013 IEEE Grenoble Conference*. IEEE, 2013, pp. 1–7.
- [45] S. D’Arco, J. Beerten, J. A. Suul, “Eigenvalue-based analysis of small-signal dynamics and stability in DC grids”, in *Modeling, operation, and analysis of DC grids*, A. Garcés, Hrsg. London, United Kingdom and Sand Diego, CA and Cambridge, MA and Oxford, United Kingdom: Elsevier AP Academic Press, 2021, pp. 69–128.
- [46] H. J. Bessai, *MIMO signals and systems*, ser. Information technology. New York: Springer, 2005.
- [47] Y. Bavafa-Toosi, *Introduction to linear control systems*. London: Academic Press, 2017.

-
- [48] E. Hendricks, O. Jannerup, P. Haase Sørensen, *Linear systems control: Deterministic and stochastic methods*. Berlin and Heidelberg: Springer, 2008.
- [49] B. Gustavsen A. Semlyen, “Rational approximation of frequency domain responses by vector fitting”, *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052–1061, 1999.
- [50] B. Gustavsen, “Improving the Pole Relocating Properties of Vector Fitting”, *IEEE Transactions on Power Delivery*, vol. 21, no. 3, pp. 1587–1592, 2006.
- [51] D. Deschrijver, M. Mrozowski, T. Dhaene, D. de Zutter, “Macromodeling of Multiport Systems Using a Fast Implementation of the Vector Fitting Method”, *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 6, pp. 383–385, 2008.
- [52] B. Gustavsen A. Semlyen, “Simulation of transmission line transients using vector fitting and modal decomposition”, *IEEE Transactions on Power Delivery*, vol. 13, no. 2, pp. 605–614, 1998.
- [53] SINTEF Energi, “The Vector Fitting Website”, 2023. Available online <https://www.sintef.no/projectweb/vectorfitting/> (Last access 02.04.2023).
- [54] B. Gustavsen, “Relaxed Vector Fitting Algorithm for Rational Approximation of Frequency Domain Responses”, in *Proceedings / 10th IEEE Workshop on Signal Propagation on Interconnects*. Piscataway, NJ: IEEE Service Center, 2006, pp. 97–100.
- [55] B. Gustavsen, “Computer code for rational approximation of frequency dependent admittance matrices”, *IEEE Transactions on Power Delivery*, vol. 17, no. 4, pp. 1093–1098, 2002.
- [56] A. Smith, S. D’Arco, J. A. Suul, B. Gustavsen, “Improved Pole Location, Order Selection, and Compaction for Vector Fitting of MIMO Systems.”
- [57] P. van Overschee B. de Moor, “N4SID: Numerical Algorithms for State Space Subspace System Identification”, *IFAC Proceedings Volumes*, vol. 26, no. 2, pp. 55–58, 1993.
- [58] P. van Overschee B. de Moor, “A unifying theorem for three subspace system identification algorithms”, *Automatica*, vol. 31, no. 12, pp. 1853–1864, 1995.
- [59] S. J. Qin, “An overview of subspace identification”, *Computers & Chemical Engineering*, vol. 30, no. 10-12, pp. 1502–1513, 2006.

-
- [60] “Estimate state-space model using subspace method with time-domain or frequency-domain data - MATLAB n4sid - MathWorks Deutschland”, 28.03.2023. Available online <https://de.mathworks.com/help/ident/ref/n4sid.html> (Last access 28.03.2023).
- [61] L. Ljung, *System identification: Theory for the user*, 2nd ed., ser. Prentice-Hall information and system sciences series. Upper Saddle River, NJ: Prentice Hall, 2012.
- [62] “Prediction error minimization for refining linear and nonlinear models - MATLAB pem - MathWorks Deutschland”, 29.03.2023. Available online https://de.mathworks.com/help/ident/ref/pem.html#bup0qu_-5 (Last access 29.03.2023).
- [63] S. D’Arco, J. A. Suul, O. B. Fosso, “A Virtual Synchronous Machine implementation for distributed control of power converters in SmartGrids”, *Electric Power Systems Research*, vol. 122, pp. 180–197, 2015. Available online <https://www.sciencedirect.com/science/article/pii/S0378779615000024>
- [64] O. Mo, S. D’Arco, J. A. Suul, “Evaluation of Virtual Synchronous Machines With Dynamic or Quasi-Stationary Machine Models”, *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5952–5962, 2017.
- [65] “CENELEC standard voltages: SN EN 60038*IEC 60038:2009, mod. = CENELEC-Normspannungen = Tensions normales du CENELEC.”
- [66] Frequency ranges, “ENTSO-E guidance document for national implementation for network codes on grid connection”, Brussels. Available online https://www.entsoe.eu/Documents/Network%20codes%20documents/NC%20RfG/210412_IGD_Frequency_ranges.pdf (Last access 01.01.2023).
- [67] Julius O. Smith III, “Introduction to Modal Representations”, Stanford, California, 05.02.2019. Available online https://ccrma.stanford.edu/~jos/ModalRep/ModalRep_4up.pdf
- [68] M. H. Samimi S. Tenbohlen, “FRA interpretation using numerical indices: State-of-the-art”, *International Journal of Electrical Power & Energy Systems*, vol. 89, pp. 115–125, 2017. Available online <https://www.sciencedirect.com/science/article/pii/S0142061516316994>
- [69] *Elektrische Maschinen und Antriebe: Grundlagen, Betriebsverhalten*, 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. Available online <http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1506482>

-
- [70] Y. Levron, J. Belikov, D. Baimel, “A Tutorial on Dynamics and Control of Power Systems with Distributed and Renewable Energy Sources Based on the DQ0 Transformation”, *Applied Sciences*, vol. 8, no. 9, p. 1661, 2018.
- [71] C. Sun J. Hahn, “Parameter reduction for stable dynamical systems based on Hankel singular values and sensitivity analysis”, *Chemical Engineering Science*, vol. 61, no. 16, pp. 5393–5403, 2006.

List of Figures

1.1.	Overview of the fitting approach	10
1.2.	Principle of the superposition method for a sample system with two input variables	10
1.3.	Overview of models used in this thesis	11
2.1.	Stability phenomena with adapted classification [15]	14
2.2.	Block diagram for a generic state-space model	18
4.1.	System setup	28
4.2.	System setup black-boxed	28
4.3.	Flowchart for input data preparation, creation of a consistent system representation and fitting	34
4.4.	Example for fit classification	41
4.5.	Scheme of the adaptive model for a step wise transition between op- erating points	41
5.1.	Overview laboratory setup	45
5.2.	Phase shift and transformation into a rotating reference frame vs. direct transformation into a rotating reference frame with adapted angle	49
5.3.	Rotating reference frames for the laboratory: grid reference frame on grid emulator and converter side and VSM reference frame	49
6.1.	Trajectories of all eigenvalues for all individual input sweeps	52
6.2.	Zoom of trajectories of eigenvalues with small imaginary part for all individual input sweeps	53
6.3.	Trajectories of eigenvalue 4 for all individual input sweeps	54
6.4.	Calculated values and fits for real and imaginary part of the eigenvalue and the corresponding elements of B and C associated with state 4 for a sweep of p^*	55
6.5.	Bode plot for $v_{g,d} = 1.1$	57
6.6.	Distribution of maximum magnitude errors for the individual sweeps of p^* , ω_g and ω^* for all input output combinations	59
6.7.	Transfer function magnitude for for $\omega_g \approx 1.016$	60
6.8.	Transfer function magnitude for $\omega^* \approx 0.957$	60
6.9.	Step response of $i_{cv,d}$ for a step in $v_{g,d}$ at $\omega^* = 0.9509$	61
6.10.	Contribution of terms for each fit function for the individual input sweeps	63
6.11.	Contribution of terms to each fit function for pairwise combined sweeps	66
6.12.	Bode plot for $v_{gd} = 0.9$ and $v^* = 1.1$	68

6.13. Maximum magnitude errors for the superposition for q^* and p^* for all analyzed operating points	69
6.14. Bode plot for $p^* = 0.1667$ and $q^* = -0.9333$	70
6.15. Bode plot for $p^* = 0.7667$ and $q^* = -0.8$	70
6.16. Maximum magnitude errors for the superposition for ω_g and ω^* for all analyzed operating points	71
6.17. Transfer functions that cause the maximum magnitude errors for (a) the fit and (b) the superposition for the combination of ω_g and ω^*	71
6.18. Maximum magnitude errors for the superposition for v^* and ω_g for all analyzed operating points	72
6.19. Step response of $i_{cv,d}$ for a step in v^* at $q^* = -1$ and $v^* = 0.9$	74
6.20. Step response of $v_{o,q}$ for a step in v^* at $p^* = 1$ and $v^* = 1.1$	74
6.21. TDEs for the superposition for p^* and v^* for all analyzed operating points	75
6.22. Step response of $i_{cv,q}$ for a step in v^* at $v^* = 1.08$ and $\omega^* = 1.04$	75
6.23. TDEs for the superposition for v^* and ω^* for all analyzed operating points	76
6.24. TDEs for the superposition for ω_g and ω^* for all analyzed operating points	76
6.25. Step response of $i_{cv,q}$ for a step in $v_{g,d}$ at $\omega_g = 1.04$ and $\omega^* = 1.04$	77
6.26. Bode plot for $q^* = 0.5$, $v^* = 1.1$ and $v_{g,d} = 0.9$	78
6.27. Parallel coordinates plot of TDEs for a combination of q^* , $v_{g,d}$ and v^*	78
6.28. Step response of $v_{o,q}$ for a step in v^* for $q^* = 0.5$, $v^* = 1.1$ and $v_{g,d} = 0.9$	79
6.29. Parallel coordinate plot of TDEs for a combination of p^* , q^* , $v_{g,d}$ and v^*	79
6.30. Step responses in $v_{o,q}$ for a step in v^* for $p^* = 1$, $q^* = 1$, $v^* = 1.1$ and $v_{g,d} = 1$	80
6.31. Step response of $v_{o,q}$ for a step in v^* for $p^* = 1$, $q^* = 1$, $v^* = 1.1$ and $v_{g,d} = 1$	80
6.32. Parallel coordinates plot of TDEs for the combination of all inputs	81
6.33. step	81
6.34. Step wise increase in p^* for the test of the adaptive model	83
6.35. Converter current for the step wise increase in p^* for the test of the adaptive model	84
6.36. Zoomed view of $i_{cv,d}$ for the last step in p^* for the test of the adaptive model	84
7.1. Output measurements for a perturbation on p^* with 0.1 Hz	88
7.2. Output measurements for a perturbation on $v_{g,d}$ with 10 Hz	89
7.3. Output measurements for a perturbation on v^* with 210 Hz	90

7.4.	Output measurements for a perturbation on v^* with 1.3 kHz	90
7.5.	Transfer function data calculated from measurement and analytical model for three example transfer functions	92
7.6.	Transfer function data calculated from measurement and analytical model for $v_{g,d}$ to $i_{cv,d}$	93
7.7.	Bode plots of three example transfer functions for the measurement and the identification results with VF, VF on substitute values, N4SID and N4SID with PEM	95
7.8.	Bode plot of an example transfer function v^* to $i_{l,d}$ for the measurement and the identification results with VF, VF on substitute values, N4SID and N4SID with PEM	96
7.9.	Step response to a step of 2000 W in p^* measured, calculated and from identified model	97
7.10.	Step response to a step of 5 V in $v_{g,d}$ measured, calculated and from identified model	97
H.1.	VSC converters	XXV
H.2.	Grid emulator	XXV
H.3.	Real-time simulator	XXVI
H.4.	Workspace	XXVI
I.1.	Transfer functions from lab measurements and analytical model for all input output combinations (part 1)	XXVII
I.2.	Transfer functions from lab measurements and analytical model for all input output combinations (part 2)	XXVIII
I.3.	Transfer functions from lab measurements and identified models for all input output combinations (part 1)	XXIX
I.4.	Transfer functions from lab measurements and identified models for all input output combinations (part 2)	XXX

List of Tables

4.1. Considered input value ranges	29
4.2. Step amplitudes for time domain analysis	39
5.1. Set points and perturbation amplitudes for laboratory tests	46
5.2. Frequency dependent factors for the scaling of the perturbation amplitude	47
5.3. Step amplitudes for laboratory test validation	48
6.1. Individual input fit accuracy	56
6.2. Individual fit errors in frequency domain	57
6.3. Individual fit errors in time domain	61
6.4. Pairwise combined fit accuracy	64
6.5. Pairwise combined sweep errors in frequency domain	67
6.6. Pairwise combined fit errors in time domain	73
6.7. Individual fit accuracy with identification	85
E.1. Parameterization of the analytical VSM model	XXII
F.1. Eigenvalues and corresponding HSVs for the analytical model at default input	XXIII
G.1. Parameterization for the laboratory investigations	XXIV

Listings

1. Function for formatting and fitting of state-space models XXXI
2. Subfunction for clustring of eigenvalues XXXI
3. Subfunction for cluster selection XXXII
4. Subfunction for formatting of state-space models XXXIV
5. Calculation of factors for B and C matrix XXXVI
6. Subfunction for fitting of state-space models XXXVII
7. Function for superposition of fitted state-space models XXXIX
8. Function for combination superposition of fitted state-space models . XL
9. Function that calculates the numerical state space-models from fits . XLI
10. Function to bring the fits into a form that can be used with the
adaptive Simulink model XLIII
11. Code for Simulink block that calculates numerical values of state-
space model XLV

List of Abbreviations

HVDC	high voltage direct current
HVAC	high voltage alternating current
VSC	voltage source converter
LCC	line-commutated converter
DC	direct current
EU	European Union
IRENA	International Renewable Energy Agency
NRMSE	Normalized root mean square error
TSO	transmission system operators
HSVs	Hankel singular values
HiL	Hardware in the loop
SSO	sub-synchronous oscillations
LTI	linear time-invariant
MIMO	multiple-input multiple-output
BIBO	bounded-input bounded-output
PHiL	Power-Hardware-in-the-Loop

Nomenclature

A, B, C, D	matrices of the linearized state space model
D	damping coefficient
H	transfer function matrix
I	sum of integrals of non-constant terms of the fit function over the input value range
I	identity matrix
I_i	integral of term i of the fit function over the input value range
J	inertia
K	diagonal transformation matrix for scaling of B and C
M_{el}	electrical torque
M_{m}	mechanical torque
M_{r}	nominal torque
NRMSE	normalized root mean square error
T	transformation matrix
T_{a}	start-up time constant
T _{dq}	dq-transformation matrix
T_{dq}	operator for dq-transformation
U	Laplace-transformed input signal vector
T	matrix of eigenvectors
W _C	controllability gramian
W _O	observability gramian
Y	Laplace-transformed output signal vector
f_x	fit function with dependency on input x
f_{xy}	fit function with dependency on input x and y
f_{offset}	function value for the default input values
\tilde{f}	approximated function value
H	transfer function
\mathbf{i}_{cv}	output current in p.u.
\mathbf{i}_{o}	output current in p.u.
k_{d}	damping constant
m	modulation index

n	system order/ number of states
n_i	number of inputs
n_o	number of outputs
p	active power in p.u.
p_i, p_{kl}	fit function coefficients
p_m	virtual mechanical power in p.u.
q	reactive power in p.u.
r_i	contribution of term i to the fit function
\mathbf{u}	vector of input signals
$u_{i,\text{def}}$	default value for input i
\mathbf{v}_o	output voltage in p.u.
\mathbf{x}	vector of state variables
x, y	independent variables in fit functions
\mathbf{y}	output vector
ω	rotating speed
ω_g	rotating speed of the grid
$\omega_{g,\text{pu}}$	rotating speed of the grid in p.u.
ω_{pu}	rotating speed in p.u.
ω_0	nominal rotational speed
D as index	marks modal representation of state-space model
d as index	marks d-component in the synchronous reference frame
i as index	counter variable
j as index	counter variable
k as index	counter variable
l as index	counter variable
q as index	marks q-component in the synchronous reference frame
Δ	marks small deviations from the equilibrium point
*	marks a reference value
'	marks auxiliary values in the adaptive state-space model
\sim	marks values in state-space model after linear transformation

A. dq-Transformation

Symmetric three-phase quantities can be expressed using dq-components. One possible transformation into the dq-system is (A.1) [69, p.1039].

$$\underbrace{\begin{bmatrix} x_d \\ x_q \end{bmatrix}}_{=\mathbf{x}_{dq}} = \frac{2}{3} \underbrace{\begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta - \frac{4\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{4\pi}{3}) \end{bmatrix}}_{=\mathbf{T}_{dq}} \cdot \underbrace{\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix}}_{=\mathbf{x}_{abc}} \quad (\text{A.1})$$

The corresponding inverse Transformation is (A.2).

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) \\ \cos(\theta - \frac{4\pi}{3}) & -\sin(\theta - \frac{4\pi}{3}) \end{bmatrix} \cdot \begin{bmatrix} x_d \\ x_q \end{bmatrix} \quad (\text{A.2})$$

It can be shown that with $\theta = \omega t$ a differentiation of three phase quantities is transformed with (A.3), where the operator $\mathcal{T}_{dq}(\mathbf{x}_{abc})$ marks the transformation of a vector \mathbf{x}_{abc} into the dq-space. [70, 63].

$$\mathcal{T}_{dq} \left(\frac{d}{dt} x_{abc} \right) = j\omega x_{dq} + \frac{d}{dt} x_{dq} \quad (\text{A.3})$$

B. Additional Equations VSM

The relation between power and torque is (B.1).

$$P = M\omega \quad (\text{B.1})$$

As the rotational speed is usually approximately the nominal speed, torque and power are identical in p.u. values. (B.2) and (B.3) apply.

$$\omega = \omega_0 \quad (\text{B.2})$$

$$p = m \quad (\text{B.3})$$

C. Similarity Transformations

The state-space model and the choice of states is not unique. The system representations are related via so called similarity transformations. If \mathbf{x} is a state vector and \mathbf{T} a non-singular constant matrix, an equivalent system representation with the states $\tilde{\mathbf{x}}$ can be obtained according to (C.1) (C.2). [48]

$$\tilde{\mathbf{x}} = \mathbf{T}\mathbf{x} \quad (\text{C.1})$$

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \tilde{\mathbf{A}} \cdot \tilde{\mathbf{x}} + \tilde{\mathbf{B}} \cdot \mathbf{u} \\ \mathbf{y} &= \tilde{\mathbf{C}} \cdot \tilde{\mathbf{x}} + \mathbf{D} \cdot \mathbf{u} \end{aligned} \quad (\text{C.2})$$

The matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$, are calculated from the original matrices \mathbf{A} , \mathbf{B} , \mathbf{C} according to (C.3). The D-matrix remains the same. [48]

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1} \\ \tilde{\mathbf{B}} &= \mathbf{T}\mathbf{B} \\ \tilde{\mathbf{C}} &= \mathbf{C}\mathbf{T}^{-1} \end{aligned} \quad (\text{C.3})$$

D. Conversion to Diagonal Form

D.1. Complex Diagonal Form

A transformation, where the transformation matrix is the inverse of the eigenvector matrix \mathbf{V} , results in a diagonal A-matrix. The eigenvector matrix \mathbf{V} contains the eigenvectors \mathbf{v}_i of \mathbf{A} as displayed in (D.1). It equals \mathbf{T}^{-1} in (C.3). The transformation is written in (D.2). The diagonal elements of \mathbf{A}_D are the eigenvalues λ_i in the order of the corresponding eigenvectors \mathbf{v}_i in the transformation matrix as presented in (D.3). In case \mathbf{A} has complex eigenvalues, \mathbf{A}_D is a complex matrix.

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix} \quad (\text{D.1})$$

$$\mathbf{x}_D = \mathbf{V}^{-1}\mathbf{x} \quad (\text{D.2})$$

$$\mathbf{A}_D = \mathbf{V}^{-1} \mathbf{A} \mathbf{V} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \lambda_n \end{bmatrix} \quad (\text{D.3})$$

D.2. Real Block Diagonal Form

To obtain a real block diagonal form instead of a complex diagonal form, the columns in the transformation matrix that contain a complex conjugated pair of eigenvectors $[\mathbf{v}_i \ \mathbf{v}_{i+1}] = [\mathbf{v}_i \ \mathbf{v}_i^*]$ are replaced with $[\Re(\mathbf{v}_i) \ \Im(\mathbf{v}_i)]$. The corresponding block in the block diagonal matrix will then have the form of (D.4).

$$\begin{bmatrix} a_{ii} & a_{i,i+1} \\ a_{i+1,i} & a_{i+1,i+1} \end{bmatrix} = \begin{bmatrix} \Re(\lambda_i) & \Im(\lambda_i) \\ -\Im(\lambda_i) & \Re(\lambda_i) \end{bmatrix} \quad (\text{D.4})$$

D.3. Degrees of Freedom

The diagonal and the block diagonal representation are not canonical. Each of the eigenvectors \mathbf{v}_i can be scaled by a factor k_i as shown in (D.5), where \mathbf{V} is the eigenvector matrix and $\tilde{\mathbf{V}}$ the alternative eigenvector matrix with the scaled eigenvectors.

$$\begin{aligned} \tilde{\mathbf{V}} &= \begin{bmatrix} k_1 \mathbf{v}_1 & k_2 \mathbf{v}_2 & \cdots & k_n \mathbf{v}_n \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}}_{=\mathbf{V}} \underbrace{\begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & k_n \end{bmatrix}}_{=\mathbf{K}} \end{aligned} \quad (\text{D.5})$$

If the factors are written as a diagonal matrix \mathbf{K} like in (D.5), then \mathbf{K}^{-1} is the transformation matrix for the conversion between the different diagonal forms. With (D.6), (D.7) shows that each factor scales exactly one state.

$$\tilde{\mathbf{V}}^{-1} = (\mathbf{V}\mathbf{K})^{-1} = \mathbf{K}^{-1}\mathbf{V}^{-1} \quad (\text{D.6})$$

$$\begin{aligned} \tilde{\mathbf{x}}_D &= \tilde{\mathbf{V}}^{-1} \mathbf{x} = \mathbf{K}^{-1} \mathbf{V}^{-1} \mathbf{x} = \mathbf{K}^{-1} \mathbf{x}_D \\ &= \begin{bmatrix} \frac{1}{k_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{k_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \frac{1}{k_n} \end{bmatrix} \mathbf{x}_D = \begin{bmatrix} \frac{x_{D,1}}{k_1} \\ \frac{x_{D,2}}{k_2} \\ \vdots \\ \frac{x_{D,n}}{k_n} \end{bmatrix} \end{aligned} \quad (\text{D.7})$$

A multiplication with a diagonal matrix from right scales the columns, a multiplication from left scales the rows. So, while in the diagonalized system the A-matrix is not affected, the factor k_i belonging to the i -th eigenvector scales column i of C and its reciprocal scales the i -th row in B. (D.8), (D.9) and (D.10) apply.

$$\begin{aligned} \tilde{\mathbf{A}}_D &= \mathbf{K}^{-1} \mathbf{A}_D \mathbf{K} = \begin{bmatrix} \frac{1}{k_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{k_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \frac{1}{k_n} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & k_n \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \lambda_n \end{bmatrix} = \mathbf{A}_D \end{aligned} \quad (\text{D.8})$$

$$\begin{aligned} \tilde{\mathbf{B}}_D &= \mathbf{K}^{-1} \mathbf{B}_D = \begin{bmatrix} \frac{1}{k_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{k_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \frac{1}{k_n} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n_i} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n_i} \\ \vdots & \ddots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n_i} \end{bmatrix} \\ &= \begin{bmatrix} \frac{b_{1,1}}{k_1} & \frac{b_{1,2}}{k_1} & \cdots & \frac{b_{1,n_i}}{k_1} \\ \frac{b_{2,1}}{k_2} & \frac{b_{2,2}}{k_2} & \cdots & \frac{b_{2,n_i}}{k_2} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{b_{n,1}}{k_n} & \frac{b_{n,2}}{k_n} & \cdots & \frac{b_{n,n_i}}{k_n} \end{bmatrix} \end{aligned} \quad (\text{D.9})$$

$$\begin{aligned}
\tilde{\mathbf{C}}_D = \mathbf{C}_D \mathbf{K} &= \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n_i} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n_i} \\ \vdots & \ddots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n_i} \end{bmatrix} \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & k_n \end{bmatrix} \\
&= \begin{bmatrix} k_1 c_{1,1} & k_2 c_{1,2} & \cdots & k_n c_{1,n_i} \\ k_1 c_{2,1} & k_2 c_{2,2} & \cdots & k_n c_{2,n_i} \\ \vdots & \ddots & \ddots & \vdots \\ k_1 c_{n,1} & k_2 c_{n,2} & \cdots & k_n c_{n,n_i} \end{bmatrix}
\end{aligned} \tag{D.10}$$

E. Parameters Analytical CCVSM model

Table E.1: Parameterization of the analytical VSM model

parameter	value
Filter inductance l_f	0.08 p.u.
Filter resistance r_f	0.00285 p.u.
Filter capacitance c_f	0.074 p.u.
Grid inductance l_g	0.2 p.u.
Grid resistance r_g	0.005 p.u.
Current controller integral gain	14.24
Current controller proportional gain	1.2732
Enable voltage feed-forward k_{ffv}	1
PLL integral gain $k_{p,PLL}$	0.084
PLL integral gain $k_{i,PLL}$	4.69
PLL filter $\omega_{LP,PLL}$	500 rad/s
Active damping filter ω_{ad}	20 rad/s
Active damping gain k_{ad}	1
Virtual resistance r_v	0.04 p.u.
Virtual inductance l_v	0.25 p.u.
Voltage controller filter ω_{vo}	500 rad/s
VSM inertia time constant T_a	50 s
VSM damping coefficient k_d	40
Frequency droop gain k_ω	20
Reactive power droop gain k_q	0.05
Reactive power filter ω_q	100 rad/s

F. Eigenvalues and HSVs of the Analytical Model

Table F.1 contains the eigenvalues and the corresponding Hankel singular values (HSVs) for the analytical model sorted in by their real part in descending order.

The HSVs are the square root of the eigenvalues of the product of the controllability gramian \mathbf{W}_C and observability gramian \mathbf{W}_O . The HSVs quantify the contribution of a state to the input-output behaviour and can therefore serve as a metric for the importance of a state. [71] The higher the HSV, the higher the contribution of a state to the input output behaviour of the system. The HSVs can be useful when doing a model reduction.

To assign the HSVs to the eigenvalues, each eigenvalue is analyzed separately. The system is reduced so that it contains only the considered eigenvalue or complex conjugated eigenvalue pair. The HSVs are calculated for the reduced system. Those HSVs match approximately with the HSVs of the full system.

From the values in Table F.1 it can be said that the input output behaviour is dominated by the eigenvalue pairs 1, 2 and 3, 4 as their HSVs are significantly higher than the others. Eigenvalue 15 has no contribution and state 14 is also of relatively low importance.

Table F.1: Eigenvalues and corresponding HSVs for the analytical model at default input

No.	Eigenvalues	Hankel singular values	
1, 2	$-0.4119 \pm j3.6340$	668.77	651.8
3, 4	$-3.0269 \pm j2778.54$	1064.98	1065.09
5, 6	$-10.0151 \pm j1.1469$	5.91	0.32
7, 8	$-12.5167 \pm j37.3335$	5.09	5.18
9, 10	$-19.9469 \pm j3.6794$	7.24	0.77
11	-110.18	1.56	-
12, 13	$280.43 \pm j3.8528$	3.85	3.67
14	-475.32	0.02	-
15	-500	0	-
16, 17	$-1488.59 \pm j5.0762$	2.86	2.15
18, 19	$-3731.06 \pm j2.0513$	2.52	0.66

G. Parameters Laboratory Investigations

Table G.1: Parameterization for the laboratory investigations

Parameter	Value
Rated AC voltage converter	400 V
Rated DC voltage converter	650 V
Rated power Converter	60 kVA
Voltage base value	326 V
Power base value	50 kVA
Current base value	102.0 A
Impedance base value	3.2 Ohm
Inductance base value	10.2 mH
Capacitance base value	0.995 mF
Filter inductance converter side L_{f1}	0.5 mH
Filter capacitance C_f	50 uF
Filter inductance grid side L_{f2}	0.2 mH
Transformer rated power $S_{r,T}$	70 kVA
Transformer short-circuit resistance e_r	2.16 %
Transformer short-circuit impedance e_z	4.34 %
Transformer impedance Z_T	0.0992 Ohm
Transformer inductance L_T	0.2738 mH
Transformer resistance R_T	0.0494 Ohm
Equivalent grid resistance r_g	0.0494 Ohm
Equivalent grid inductance l_g	0.4738 mH
PLL integral gain $k_{p,PLL}$	26.53
PLL integral gain $k_{i,PLL}$	1474
PLL filter time constant τ_{PLL}	0.002 s
Virtual resistance r_v	0.02 p.u.
Virtual inductance l_v	0.25 p.u.
Voltage controller filter time constant τ_{vo}	0.005 s
VSM inertia time constant t_a	2 s
VSM damping coefficient	40
Frequency droop gain k_ω	5
Reactive power droop gain k_q	0.05
Sample time T_s	100 μ s

H. Pictures Laboratory Setup



Figure H.1: VSC converters



Figure H.2: Grid emulator

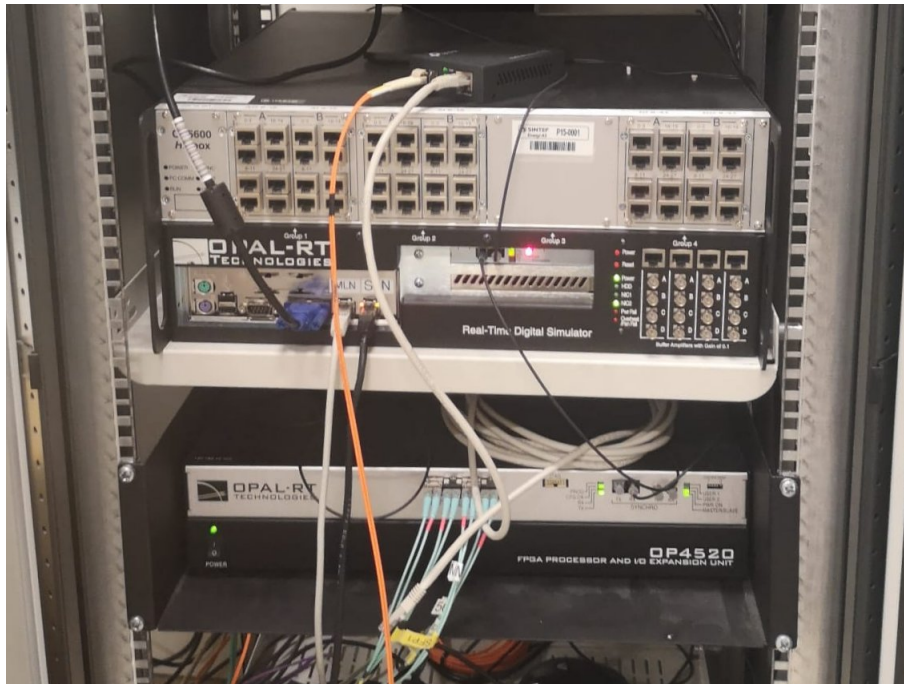


Figure H.3: Reat-time simulator

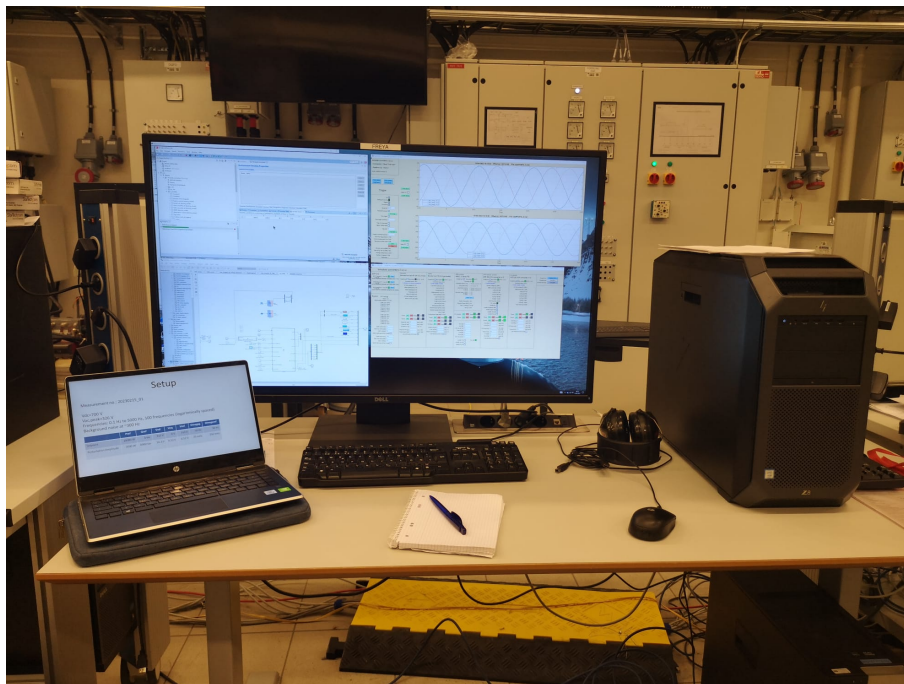


Figure H.4: Workspace

I. Figures Laboratory Measurements and Identification

The bode plots of the frequency response measurements in comparison to the analytical model as well as the comparison of the measurements and the identified models for all input combinations are displayed on the following pages.

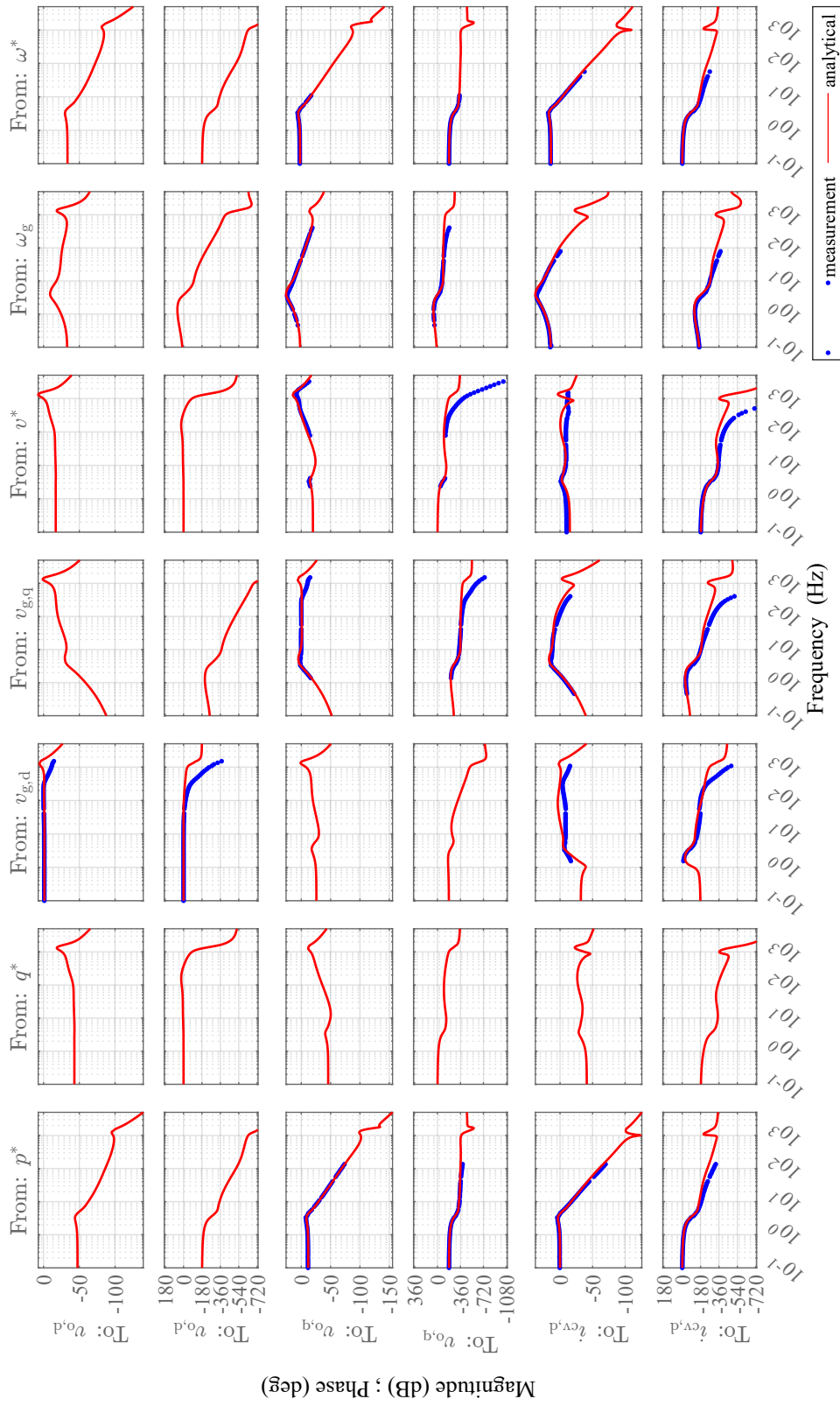


Figure I.1: Transfer functions from lab measurements and analytical model for all input output combinations (part 1)

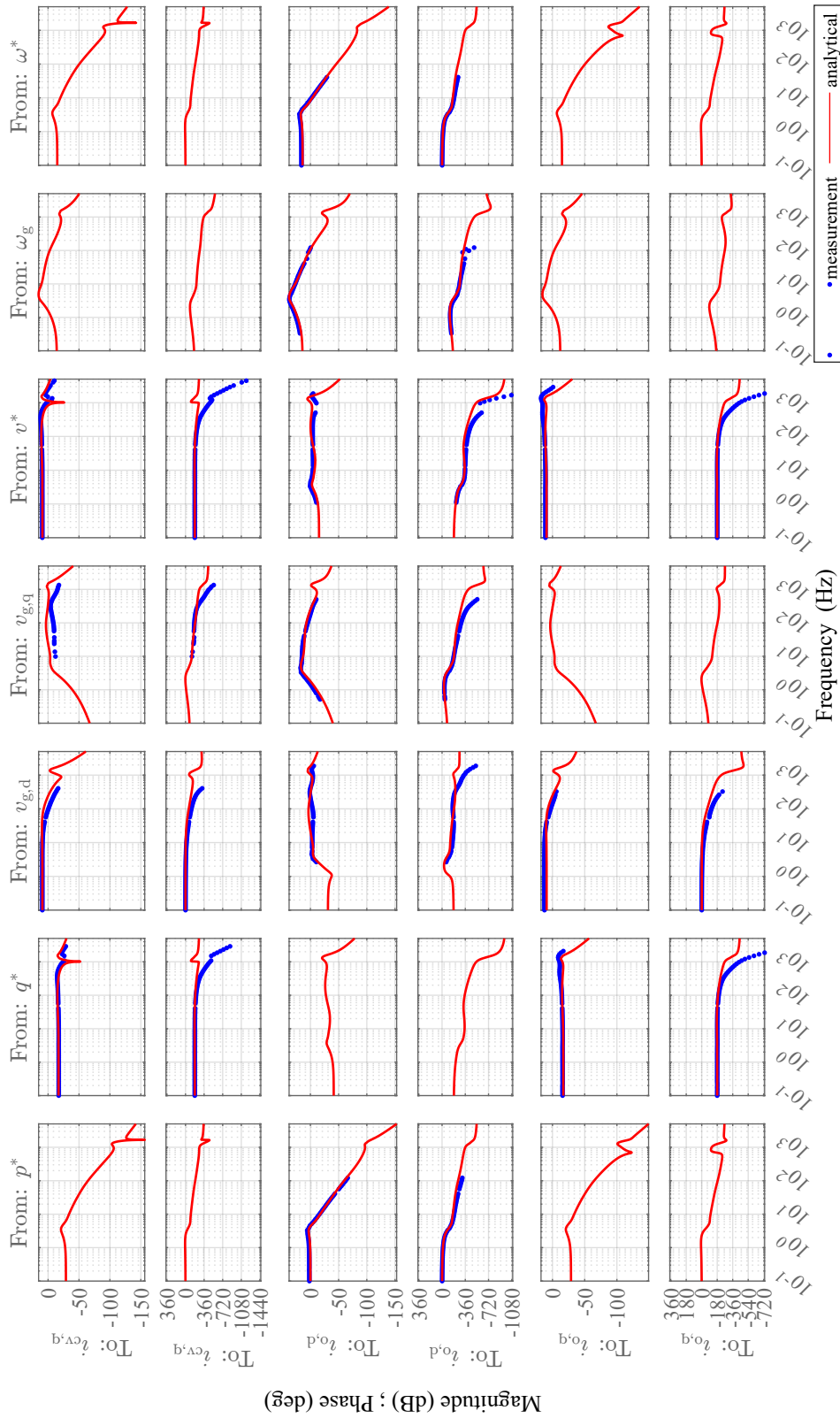


Figure I.2: Transfer functions from lab measurements and analytical model for all input output combinations (part 2)

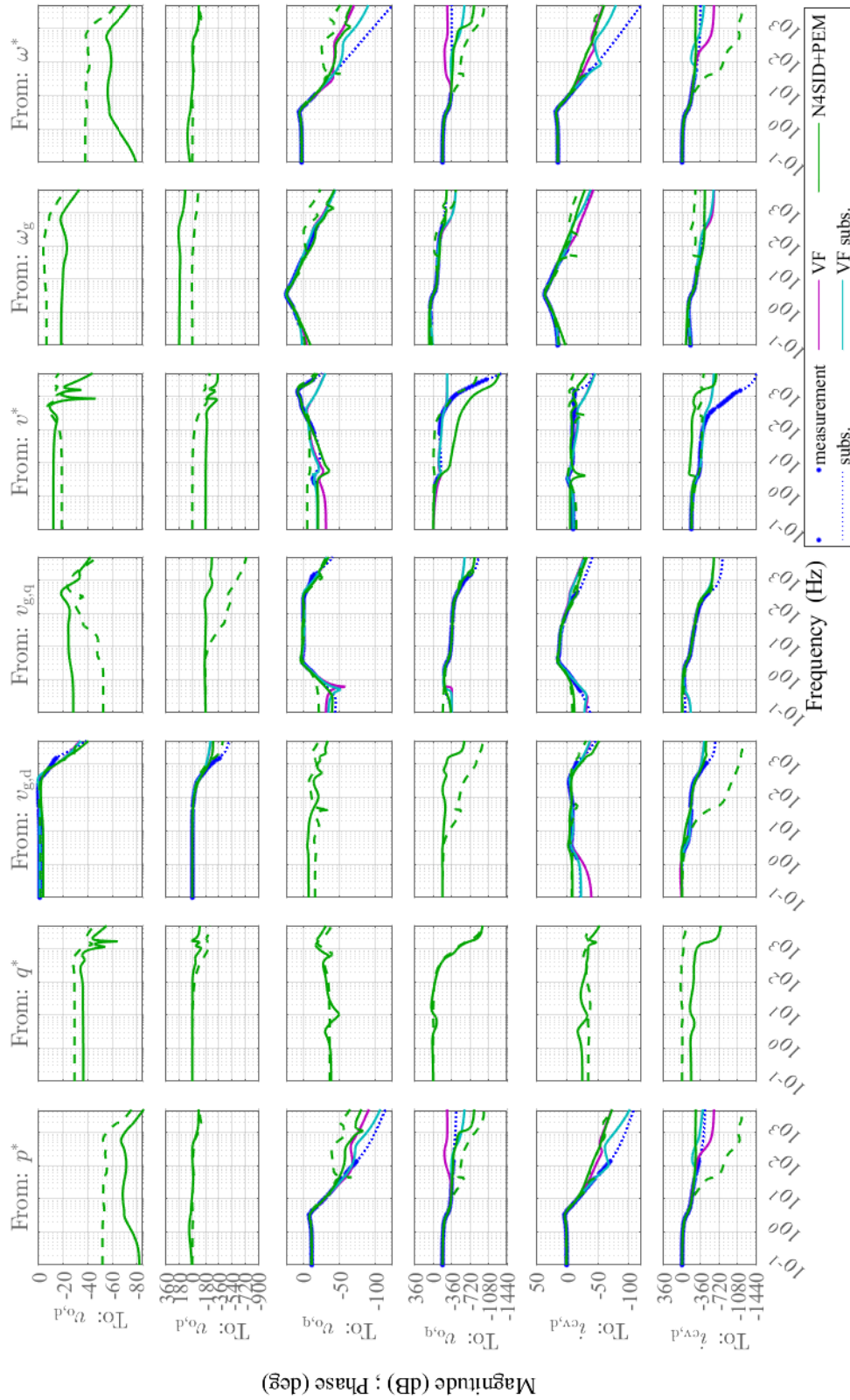


Figure I.3: Transfer functions from lab measurements and identified models for all input output combinations (part 1)

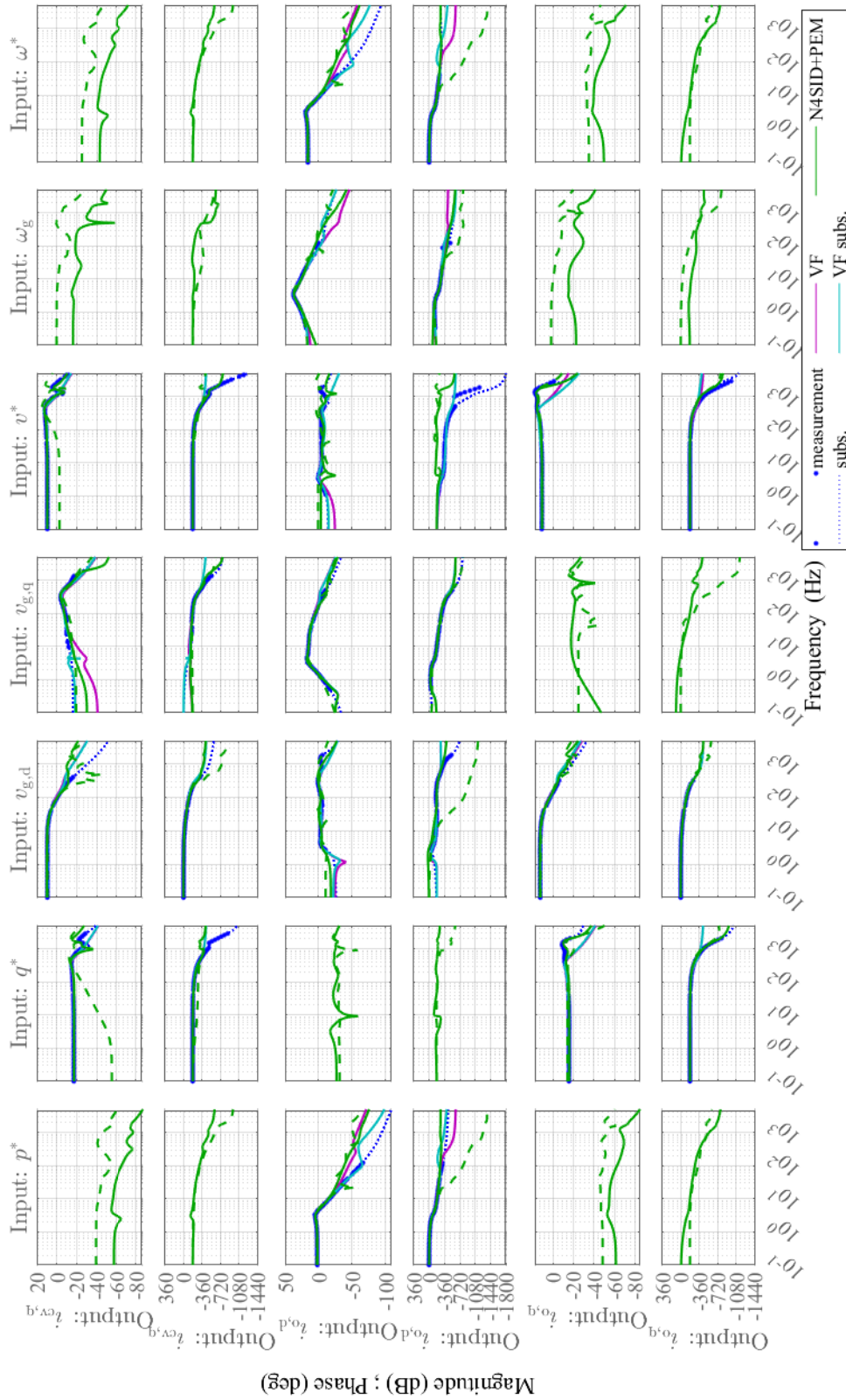


Figure I.4: Transfer functions from lab measurements and identified models for all input output combinations (part 2)

J. Code for Fitting of Input Sweeps

Listing 1: Function for formatting and fitting of state-space models

```

1 function [sys_fitted,struct_array_of_all_ops]=format_and_fit_ss_models(
    struct_array_of_all_ops, inputs_default, fit_function)
2
3 % This function formats all state-space models in a uniform way and creates fits for
4 % each matrix element in the state-space matrices. The start point is a struct array
5 % called struct_array_of_all_ops containing the input values and the unformatted
6 % state-space models for each operating point in the sweep. The struct array
7 % has one field per input and one for the state-space model.
8
9 %% Cluster the eigenvalues of all operating points to identify states
10 struct_array_of_all_ops=cluster_EVs(struct_array_of_all_ops,inputs_default);
11
12 %% Format the state-space models
13 struct_array_of_all_ops=format_ss_models(struct_array_of_all_ops);
14
15 %% Fit the state-space models
16 [sys_fitted, struct_array_of_all_ops]=fit_ss_models(struct_array_of_all_ops,
17 inputs_default, fit_function);
18 end

```

Listing 2: Subfunction for clustring of eigenvalues

```

1 function struct_array_of_all_ss_models_unformatted=cluster_EVs(
    struct_array_of_all_ss_models_unformatted,inputs_default)
2
3 % This function calculates the eigenvalues for each operating point in
4 % the sweep and puts them into clusters.
5 % Clusters with few elements are removed.
6 % The output is a struct array with the input values, the
7 % corresponding, still unformatted state-space models and an array of
8 % filtered and sorted eigenvalues for each operating point.
9
10 number_of_operating_points=size(struct_array_of_all_ss_models_unformatted,1);
11
12 % initialize the array to save the clustered eigenvalues in.
13 % There is one column per operating point, and one row per cluster
14 % of eigenvalues
15 matrix_of_clustered_EVs=nan(1,number_of_operating_points);
16
17 % loop through the operating points
18 for n=1:number_of_operating_points
19
20     % get the EVs
21     vector_of_EVs_raw=eig(struct_array_of_all_ss_models_unformatted(n).sys.A);
22
23     % cluster the EVs
24     if all(isnan(matrix_of_clustered_EVs))
25
26         % no reference data available, so just take the sorted EVs as
27         % initialization for the clusters
28         matrix_of_clustered_EVs(1:numel(vector_of_EVs_raw),n)=vector_of_EVs_raw;
29
30         % for the added columns in the matrix reset the value to nan
31         matrix_of_clustered_EVs(:,1:end~n)=nan;
32
33     else
34         for i=1:numel(vector_of_EVs_raw)

```

```

34
35         % now check for each EV seperately which cluster fits best
36         this_EV=vector_of_EVs_raw(i);
37         matrix_of_clustered_EVs=find_cluster(this_EV, matrix_of_clustered_EVs, n
);
38
39     end
40 end
41
42 end
43
44 number_of_clusters=size(matrix_of_clustered_EVs,1);
45
46 % a state must be identified in minimum 80 % of runs
47 min_number_of_values_per_cluster=round(0.8*number_of_operating_points);
48
49 remove_cluster=zeros(number_of_clusters,1);
50 % find the clusters with only few elements and set a marker
51 for i=1:number_of_clusters
52     if sum(~isnan(matrix_of_clustered_EVs(i,:))<min_number_of_values_per_cluster
53         remove_cluster(i)=1;
54     end
55 end
56
57 % remove the marked clusters
58 matrix_of_clustered_EVs=matrix_of_clustered_EVs(~remove_cluster,:);
59 number_of_clusters=size(matrix_of_clustered_EVs,1);
60
61 % sort the clusters by their value in the default point to get a
62 % consistent representation throughout all swipes
63 index_of_default=(arrayfun(@(x) isequal(struct_array_of_all_ss_models_unformatted(x) .
64     inputs,inputs_default),1:size(struct_array_of_all_ss_models_unformatted,1)));
65 if ~sum(index_of_default)
66     index_of_default=ceil(size(struct_array_of_all_ss_models_unformatted,1)/2);
67 end
68 [~,pos]=sortrows([real(matrix_of_clustered_EVs(:,index_of_default)),imag(
69     matrix_of_clustered_EVs(:,index_of_default))],'descend');
70 matrix_of_clustered_EVs=matrix_of_clustered_EVs(pos,:);
71
72 % save the clustered eigenvalues to the struct array
73 temp=mat2cell(matrix_of_clustered_EVs,number_of_clusters,ones(
74     number_of_operating_points,1)');
75 [struct_array_of_all_ss_models_unformatted.eigenvalues_filtered_and_sorted]=temp{:};
76 end

```

Listing 3: Subfunction for cluster selection

```

1 function matrix_of_clustered_EVs=find_cluster(this_EV, matrix_of_clustered_EVs,
2     index_of_this_op)
3 % This function takes one eigenvalue and puts it into a cluster. If no
4 % suitable cluster is found, a new cluster is created. The function
5 % returns the updated matrix of clustered eigenvalues.
6
7 matrix_of_previous_EVs=matrix_of_clustered_EVs(:,1:index_of_this_op-1);
8
9 % calculated the distance to all the previous EVs
10 matrix_of_dist_to_previous_EVs=abs(matrix_of_previous_EVs-this_EV);
11
12 % normalize this distance

```

```

12 matrix_of_rel_dist_to_previous_EVs=matrix_of_dist_to_previous_EVs./abs(
    matrix_of_previous_EVs);
13
14 % check which distances are bigger than the max distance
15 % NOTE: this value might is a parameter that might need tuning
16 % depending on the input data
17 max_rel_dist=0.5;
18
19 index_suitable_clusters=find(any(matrix_of_rel_dist_to_previous_EVs<max_rel_dist,2))
    ;
20 [~,col_index_shortest_rel_dist_per_cluster]=min(matrix_of_rel_dist_to_previous_EVs
    , [],2);
21
22 % sort the suitable clusters by their relative distance
23 [~,pos]=sort(matrix_of_rel_dist_to_previous_EVs(index_suitable_clusters,
    col_index_shortest_rel_dist_per_cluster(index_suitable_clusters)), 'ascend');
24 index_suitable_clusters=index_suitable_clusters(pos);
25
26 % index suitable_clusters gives now the row numbers in the
27 % clustered_EVs matrix in that the EV would fit, sorted from the
28 % shortest distance to the highest
29
30 if numel(index_suitable_clusters)==0
31     % data does not fit in any of the existing clusters
32     % open a new cluster
33     matrix_of_clustered_EVs(end+1,index_of_this_op)=this_EV;
34     % nan the other added cells
35     matrix_of_clustered_EVs(end,1:end~=index_of_this_op)=nan;
36
37 else
38     % get the nearest suitable cluster
39     pos=index_suitable_clusters(1);
40
41     if isnan(matrix_of_clustered_EVs(pos,index_of_this_op))
42         % if the position is not occupied yet, the EV can be
43         % inserted directly
44         matrix_of_clustered_EVs(pos,index_of_this_op)=this_EV;
45     else
46
47         % if the position is occupied, compare which value fits
48         % better into the cluster
49
50         if min(abs(matrix_of_clustered_EVs(pos,index_of_this_op)-
matrix_of_previous_EVs(pos,:))>min(abs(this_EV-matrix_of_previous_EVs(pos,:)))
51             % this one fits better --> replace the old one and
52             % move to new cluster
53             to_new_cluster=matrix_of_clustered_EVs(pos,index_of_this_op);
54             matrix_of_clustered_EVs(pos,index_of_this_op)=this_EV;
55         else
56             to_new_cluster=this_EV;
57         end
58
59         % insert the value that does not fit better to a new cluster
60         % NOTE: it would be better to check here if the value fits into
61         % another cluster instead
62
63         matrix_of_clustered_EVs(end+1,index_of_this_op)=to_new_cluster;
64
65         % nan the added cells
66         matrix_of_clustered_EVs(end,1:end~=index_of_this_op)=nan;

```



```

67     end
68 end
69 end

```

Listing 4: Subfunction for formatting of state-space models

```

1 function struct_array_of_all_ops=format_ss_models(struct_array_of_all_ops)
2     % This function takes the struct array that includes the unformatted
3     % state-space models and the filtered and sorted eigenvalues for each
4     % operating point of the sweep and formats the state-space models
5     % accordingly. It returns the same struct enhanced with the state-space
6     % model in the uniform system representation for each operating point.
7
8     number_of_operating_points=size(struct_array_of_all_ops,1);
9     table_of_inputs=struct2table(cell2mat(extractfield(struct_array_of_all_ops,'inputs'
10    )));
11     cell_array_of_new_sys=cell(number_of_operating_points,1);
12
13     % loop over the operating points
14     for n=1:number_of_operating_points
15
16         EVs_filtered_and_sorted=struct_array_of_all_ops(n).
17         eigenvalues_filtered_and_sorted;
18
19         % skip operating points where not all states were identified or
20         % the operating points flagged as not valid. For the others do the
21         % transformations to get the uniform system representation.
22
23         if ~any(isnan(struct_array_of_all_ops(n).eigenvalues_filtered_and_sorted))||~
24         struct_array_of_all_ops(n).operating_point_valid
25             sys=struct_array_of_all_ops(n).sys;
26
27             % get the eigenvectors and diagonalize the state space system
28             [EVecs,~]=eig(sys.A);
29             new_sys=ss2ss(sys,inv(EVecs));
30
31             % get the eigenvalues
32             EVs=diag(new_sys.A);
33
34             % caution:
35             % the non-diagonal elements of A are not zero but very small!
36
37             % compare the eigenvalues to the values in the vector with the
38             % filtered and sorted eigenvalues
39
40             ind=zeros(size(EVs_filtered_and_sorted,1),1);
41             for i=1:size(EVs_filtered_and_sorted)
42                 % get the index of the sorted eigenvalues in the vf results
43                 this_EV=EVs_filtered_and_sorted(i);
44                 ind(i)=find(abs(this_EV-EVs)==min(abs(this_EV-EVs)));
45             end
46
47             % get the number of states to exclude
48             num_del=numel(EVs)-numel(EVs_filtered_and_sorted);
49
50             % decide which states to exclude
51             cut_off=zeros(size(ind));
52             if num_del>0
53                 cut_off=~ismember(round(EVs,7,'significant'),round(
54                 EVs_filtered_and_sorted,7,'significant'));

```

```

51         % move states that are not needed to the end
52         ind(end+1:end+num_del)=find(cut_off);
53     end
54
55     % rearrange the system according to the results of the clustering
56     new_sys=xperm(new_sys,ind);
57     EVs=EVs(ind);
58
59     % cut the not wanted states off if needed
60     if num_del>0
61         remove=numel(EVs_filtered_and_sorted)+(1:num_del);
62         new_sys=modred(new_sys,remove,'Truncate');
63     end
64
65     % eliminate very small elements to make the next steps
66     % easier
67     new_sys.A(abs(new_sys.A)<10^-10)=0;
68     new_sys.B(abs(new_sys.B)<10^-10)=0;
69     new_sys.C(abs(new_sys.C)<10^-10)=0;
70
71     % make sure that B and C are in the right form (=mind the
72     % factor of the EVecs during the diagonalization). Do this in
73     % the complex diagonal system as this is simpler.
74
75     C_prev=[];
76
77     % find the closest operating point that this operating point
78     % can be compared with
79     % NOTE: the chosen metric is not optimal, because of different
80     % value ranges for every input
81
82     if n>1
83         [~,ind_of_closest_operating_points]=sort(vecnorm(table_of_inputs(n,:)-
table_of_inputs(1:n-1,:),2,2),'ascend');
84         for i=1:numel(ind_of_closest_operating_points)
85             ind=ind_of_closest_operating_points(i);
86             if ~isempty(cell_array_of_new_sys{ind})
87                 C_prev=cell_array_of_new_sys{ind}.C;
88                 break;
89             end
90         end
91     end
92
93     % compare the values of C in this step with the values of C in
94     % the previous step of the sweep
95     T=getCorrectFactors(new_sys.C,C_prev);
96
97     % transform the system to apply the correct factors
98     new_sys=ss2ss(new_sys,inv(T));
99
100    % add the new system to a cell array
101    cell_array_of_new_sys{n}=new_sys;
102 else
103     % set a flag that this operating point should be excluded
104     struct_array_of_all_ops(n).operating_point_valid=false;
105 end
106 end
107
108 % add the cell array of formatted systems to the struct array
109 [struct_array_of_all_ops.new_sys]=cell_array_of_new_sys{:};

```

110 end

Listing 5: Calculation of factors for B and C matrix

```

1 function T=getCorrectFactors(thisC,C_prev)
2     % This functions compares the matrix thisC with the matrix C_prev and
3     % calculates the transformation matrix T, which includes the factors
4     % that scale the columns of thisC to get it into a similar
5     % representation as C_prev.
6     % This is necessary as the diagonalization of the state space system
7     % has one degree of freedom per state.
8
9     T=zeros(size(thisC,2));
10
11     % loop over the columns
12     for n=1:size(thisC,2)
13
14         thisColumnofC=thisC(:,n);
15
16         % check if there are any non-zero elements in the columns
17         if sum(abs(thisColumnofC))>0
18
19             % check if the reference C_prev exists
20             if ~isempty(C_prev)
21                 thisColumnofC_prev=C_prev(:,n);
22             else
23                 thisColumnofC_prev=[];
24             end
25
26             % find the first non-zero element in a column
27             first_nonzero=min(find(abs(thisColumnofC)>10^-10,1,"first"));
28
29             % sometimes elements have very small imaginary parts that
30             % change sign and lead to a complex conjugated column. The very
31             % small imaginary parts are therefore simply cut off.
32             if abs(imag(thisColumnofC(first_nonzero)))<10^-10
33                 thisColumnofC(first_nonzero)=real(thisColumnofC(first_nonzero));
34             end
35
36             % norm all columns so that the first nonzero element in each
37             % column is positive real with the same absolute value as before
38             factor=1/(thisColumnofC(first_nonzero)*abs(thisColumnofC(first_nonzero)));
39
40             % normalization with absolute value because otherwise the
41             % transformation matrix be badly scaled
42             factor=factor/(norm(abs(thisColumnofC)));
43
44             thisColumnofC_new=thisColumnofC*factor;
45
46             % check for the right sign (it might be that the first non-zero
47             % has a zero crossing but the other elements do not)
48             if ~isempty(thisColumnofC_prev)
49                 if sum(sign(real(thisColumnofC_new(abs(thisColumnofC_new)>10^-10)))~=
50 sign(real(thisColumnofC_prev(abs(thisColumnofC_new)>10^-10))))>0.5* numel(
51 thisColumnofC)
52                 % more than 50 % of the values that are not noise have
53                 % changed their sign --> probably wrong sign
54                 factor=-factor;
55             end
56         end
57     end
58 end

```

```

55         end
56     else
57         factor=1;
58     end
59
60     % write the factor for this column to the corresponding element in
61     % the T-matrix
62     T(n,n)=factor;
63 end
64 end

```

Listing 6: Subfunction for fitting of state-space models

```

1 function [sys_fitted, struct_array_of_all_ops]=fit_ss_models(struct_array_of_all_ops,
    inputs_default, fittype)
2 % This function creates the fit functions with the type fittype from
3 % the uniformly formatted state space-models in the struct array
4 % struct_array_of_all_ops in dependency of the swept input. The
5 % variable inputs_default constrains the input values at the default
6 % operating point of the fit. The output is the structure sys_fitted
7 % containing the fit functions for each matrix
8 % element and some additional information needed to evaluate the fit.
9
10 % initialize some variables
11 table_of_inputs=struct2table(cell2mat(extractfield(struct_array_of_all_ops,'inputs')
    ));
12
13 is_valid_op=cell2mat(extractfield(struct_array_of_all_ops,'operating_point_valid'));
14 first_valid_op=struct_array_of_all_ops(find(is_valid_op, 1,'first'));
15
16 number_of_operating_points=numel(struct_array_of_all_ops);
17 number_of_states=sqrt(numel(first_valid_op.new_sys.A));
18 number_of_inputs=numel(first_valid_op.new_sys.B)/number_of_states;
19 number_of_outputs=numel(first_valid_op.new_sys.C)/number_of_states;
20
21 A_all=nan(numel(first_valid_op.new_sys.A),number_of_operating_points);
22 B_all=nan(numel(first_valid_op.new_sys.B),number_of_operating_points);
23 C_all=nan(numel(first_valid_op.new_sys.C),number_of_operating_points);
24
25 % loop over the operating points
26 for n=1:number_of_operating_points
27
28     if struct_array_of_all_ops(n).operating_point_valid
29
30         sys=struct_array_of_all_ops(n).new_sys;
31
32         % for fitting transform into a real block-diagonal form
33         sys=c2rss(sys);
34
35         % write all state-space elements into a matrix
36         % there is one column per step in the sweep and one row per
37         % matrix element
38         A_all(:,n)=sys.A(:);
39         B_all(:,n)=sys.B(:);
40         C_all(:,n)=sys.C(:);
41
42     end
43 end
44
45 % reduce the independent variables to the inputs that are varied

```

```

46 table_of_inputs_reduced=table_of_inputs(:,~all(table_of_inputs{:, :})==table_of_inputs
    {1, :},1));
47 names_of_varied_inputs=table_of_inputs_reduced.Properties.VariableNames;
48
49 % get the offset for independent variables
50 for i=1:numel(names_of_varied_inputs)
51     u_offset(i)=inputs_default.(names_of_varied_inputs{i});
52 end
53
54 % get the indepentend variables for all valid operating points
55 u=table_of_inputs_reduced{is_valid_op,:};
56
57 % provide start values for fit if it is a custom fit function
58 if isa(fittype, 'fittype')
59     startPoints=zeros(numel(coeffnames(fittype)),1);
60     custom_fit=1;
61 else
62     startPoints=[];
63     custom_fit=0;
64 end
65
66 % fit elements of A
67 A_fitted=cell(size(A_all,1),1);
68 for n=(1:size(A_all,1))
69     array_of_a_ij=A_all(n,is_valid_op)';
70     x=u-u_offset;
71     y=array_of_a_ij;
72
73     if max(y)-min(y)~=0
74         if custom_fit
75             A_fitted{n}=fit(x, y, fittype, 'Start', startPoints);
76         else
77             A_fitted{n}=fit(x, y, fittype);
78         end
79     else
80         A_fitted{n}=mean(y);
81     end
82 end
83
84 A_fitted=reshape(A_fitted,number_of_states, []);
85
86 % fit elements of B
87 B_fitted=cell(size(B_all,1),1);
88 for n=(1:size(B_all,1))
89     array_of_b_ij=B_all(n,is_valid_op)';
90     x=u-u_offset;
91     y=array_of_b_ij;
92     if max(y)-min(y)~=0
93         if custom_fit
94             B_fitted{n}=fit(x, y, fittype, 'Start', startPoints);
95         else
96             B_fitted{n}=fit(x, y, fittype);
97         end
98     else
99         B_fitted{n}=mean(y);
100    end
101 end
102
103 B_fitted=reshape(B_fitted,number_of_states, []);
104

```

```

105 % fit elements of C
106 C_fitted=cell(size(C_all,1),1);
107 for n=(1:size(C_all,1))
108     array_of_c_ij=C_all(n,is_valid_op)';
109     x=u-u_offset;
110     y=array_of_c_ij;
111     if max(y)-min(y)~=0
112         if custom_fit
113             C_fitted{n}=fit(x, y, fittype, 'Start', startPoints);
114         else
115             C_fitted{n}=fit(x, y, fittype);
116         end
117     else
118         C_fitted{n}=mean(y);
119     end
120 end
121
122 C_fitted=reshape(C_fitted, [], number_of_states);
123
124 % copy the structures of fits to a common structure and add further
125 % information
126 sys_fitted.A=A_fitted;
127 sys_fitted.B=B_fitted;
128 sys_fitted.C=C_fitted;
129 sys_fitted.D=sys.D;
130
131 % also save the default of this swipe as it is needed as offset when
132 % evaluating the function
133 sys_fitted.inputs_default=inputs_default;
134
135 % also save which one is the independent variable of the fit
136 sys_fitted.names_of_variables_swiped=names_of_varied_inputs;
137
138 % copy the input and output names
139 sys_fitted.InputName=sys.InputName;
140 sys_fitted.OutputName=sys.OutputName;
141
142 % for later use store also the matrix of elements from the swipe that
143 % served as input for the fit
144 temp=num2cell(A_all,1);
145 [struct_array_of_all_ops.A_all]=temp{:};
146 temp=num2cell(B_all,1);
147 [struct_array_of_all_ops.B_all]=temp{:};
148 temp=num2cell(C_all,1);
149 [struct_array_of_all_ops.C_all]=temp{:};
150 end

```

Listing 7: Function for superposition of fitted state-space models

```

1 function sys_new = combine_fitted_ss(varargin)
2     % This function combines a variable number of uniformly formatted,
3     % fitted-state space models to a single fitted state-space model.
4
5     % initialize with the first input
6     sys_new=varargin{1};
7
8     % loop over other systems and add them to the new fitted system
9     for i=2:numel(varargin)
10         for j=1:numel(sys_new.A)
11             if ~isnumeric(sys_new.A{j}) || ~isnumeric(varargin{i}.A{j})

```

```

12         sys_new.A{j}=combine_fit(sys_new.A{j},varargin{i}.A{j},i);
13     end
14 end
15
16 for j=1:numel(sys_new.B)
17     if ~isnumeric(sys_new.B{j})||~isnumeric(varargin{i}.B{j})
18         sys_new.B{j}=combine_fit(sys_new.B{j},varargin{i}.B{j},i);
19     end
20 end
21
22 for j=1:numel(sys_new.C)
23     if ~isnumeric(sys_new.C{j})||~isnumeric(varargin{i}.C{j})
24         sys_new.C{j}=combine_fit(sys_new.C{j},varargin{i}.C{j},i);
25     end
26 end
27 sys_new.names_of_variables_swiped=[sys_new.names_of_variables_swiped,varargin{i}
    ].names_of_variables_swiped];
28 end
29 end

```

Listing 8: Function for combination superposition of fitted state-space models

```

1 function new_fit=combine_fit(fit1,fit2,index)
2     % This function combines two fit functions by adding the non-constant
3     % terms of fit2 to fit1. The new fit function, coefficient names and
4     % values and variable names are stored in a struct named new_fit.
5     % fit1 and fit2 can either be of the type of the Matlab fit classes
6     % cfit and sfit or a struct with the same fields as new_fit. This is
7     % necessary in case this function is used repeatedly.
8     % The coefficient and variable names in new_fit are the same as in the
9     % original fits but enhanced with _1 if the term originates from fit1
10    % or _2 if the term originates from fit2.
11
12    if ~isnumeric(fit1)
13        % get the formula, the variable names, coefficient names and
14        % coefficient values of the first fit function
15        if contains(class(fit1),{'cfit','sfit'})
16            formulal=formula(fit1);
17            varnames1=indepnames(fit1)';
18            coeffnames1=coeffnames(fit1)';
19            coeffvalues1=coeffvalues(fit1);
20        else
21            formulal=fit1.formula;
22            varnames1=fit1.varnames;
23            coeffnames1=fit1.coeffnames;
24            coeffvalues1=fit1.coeffvalues;
25        end
26
27        % enhance the variable and coefficient names of fit1 with _1 and
28        % put them into formulal
29        for i=1:numel(varnames1)
30            temp=varnames1{i};
31            varnames1_new{i}=strcat(temp,'_1');
32            formulal=strrep(formulal,varnames1{i},varnames1_new{i});
33        end
34        for i=1:numel(coeffnames1)
35            temp=coeffnames1{i};
36            coeffnames1_new{i}=strcat(temp,'_1');
37            formulal=strrep(formulal,coeffnames1{i},coeffnames1_new{i});
38        end

```

```

39     else
40         % if fit1 is a constant numeric value, the numeric value is
41         % converted to a string used as formulal.
42         formulal=num2str(fit1);
43         varnames1_new=[];
44         coeffnames1_new=[];
45         coeffvalues1=[];
46     end
47
48     if ~isnumeric(fit2)
49         % get the formula, the variable names, coefficient names and
50         % coefficient values of the second fit function
51         if contains(class(fit2),{'cfit','sfit'})
52             formula2=formula(fit2);
53             varnames2=indepnames(fit2);
54             coeffnames2=coeffnames(fit2);
55             coeffvalues2=coeffvalues(fit2);
56         else
57             formula2=fit2.formula;
58             varnames2=fit2.varnames;
59             coeffnames2=fit2.coeffnames;
60             coeffvalues2=fit2.coeffvalues;
61         end
62
63         % enhance the variable and coefficient names of fit1 with _1 and
64         % put them into formula2
65         for i=1: numel(varnames2)
66             temp=varnames2{i};
67             varnames2_new{i}=strcat(temp,'_',num2str(i));
68             formula2=strep(formula2,varnames2{i},varnames2_new{i});
69         end
70         for i=1: numel(coeffnames2)
71             temp=coeffnames2{i};
72             coeffnames2_new{i}=strcat(temp,'_',num2str(i));
73             formula2=strep(formula2,coeffnames2{i},coeffnames2_new{i});
74         end
75
76         % remove the offset from formula 2
77         terms=split(formula2,'+');
78         ind=contains(terms,varnames2_new);
79         formula2=join(terms(ind),'+');
80         coeffnames2_new=coeffnames2_new(ind);
81         coeffvalues2_new=coeffvalues2(ind);
82     else
83         % if fit2 is a constant it is removed
84         formula2='0';
85         varnames2_new=[];
86         coeffnames2_new=[];
87         coeffvalues2_new=[];
88     end
89
90     % finally, combine the functions into one
91     new_fit.formula=strcat(formulal,'+',formula2);
92     new_fit.varnames=[varnames1_new,varnames2_new];
93     new_fit.coeffnames=[coeffnames1_new,coeffnames2_new];
94     new_fit.coeffvalues=[coeffvalues1,coeffvalues2_new];
95 end

```

Listing 9: Function that calculates the numerical state space-models from fits


```

1 function sys=calc_numerical_values_for_fitted_sys(inputs,sys_fitted)
2     % This function takes the structure sys_fitted and calculates a numeric
3     % state-space model for the values in the struct inputs.
4     % The output sys has the Matlab type ss.
5
6     % Get the independent values of the fitted system
7     names_of_varied_inputs=sys_fitted.names_of_variables_swiped;
8     x=nan(1,numel(names_of_varied_inputs));
9     for i=1:numel(names_of_varied_inputs)
10        x(i)=inputs.(names_of_varied_inputs{i})-sys_fitted.inputs_default.(
11        names_of_varied_inputs{i});
12    end
13
14    % initialize the state space matrices
15    A=zeros(size(sys_fitted.A));
16    B=zeros(size(sys_fitted.B));
17    C=zeros(size(sys_fitted.C));
18    D=sys_fitted.D;
19
20    % loop over the elements of A and calculate the numeric values
21    for i=1:numel(A)
22        if ~isnumeric(sys_fitted.A{i})
23            % for fits of type cfit or sfit
24            if contains(class(sys_fitted.A{i}),{'cfit','sfit'})
25                A(i)=sys_fitted.A{i}(x);
26            % for fits of the custom struct type
27            else
28                formula=sys_fitted.A{i}.formula;
29                coeff_names=sys_fitted.A{i}.coeffnames;
30                coeff_values=sys_fitted.A{i}.coeffvalues;
31                for k=1:numel(coeff_names)
32                    formula=strrep(formula,coeff_names(k),['(',num2str(coeff_values(k)),
33                    ')']);
34                end
35                var_names=sys_fitted.A{i}.varnames;
36                for k=1:numel(var_names)
37                    formula=strrep(formula,var_names(k),['(',num2str(x(k)),')']);
38                end
39                A(i)=eval(formula{1});
40            end
41        end
42    end
43
44    % loop over the elements of B and calculate the numeric values
45    for i=1:numel(B)
46        if ~isnumeric(sys_fitted.B{i})
47            if contains(class(sys_fitted.B{i}),{'cfit','sfit'})
48                B(i)=sys_fitted.B{i}(x);
49            else
50                formula=sys_fitted.B{i}.formula;
51                coeff_names=sys_fitted.B{i}.coeffnames;
52                coeff_values=sys_fitted.B{i}.coeffvalues;
53                for k=1:numel(coeff_names)
54                    formula=strrep(formula,coeff_names(k),['(',num2str(coeff_values(k)),
55                    ')']);
56                end
57                var_names=sys_fitted.B{i}.varnames;
58                for k=1:numel(var_names)
59                    formula=strrep(formula,var_names(k),['(',num2str(x(k)),')']);
60                end
61            end
62        end
63    end

```

```

58         B(i)=eval(formula{1});
59     end
60 end
61 end
62
63 % loop over the elements of C and calculate the numeric values
64 for i=1:numel(C)
65     if ~isnumeric(sys_fitted.C{i})
66         if contains(class(sys_fitted.C{i}),{'cfit','sfit'})
67             C(i)=sys_fitted.C{i}(x);
68         else
69             formula=sys_fitted.C{i}.formula;
70             coeff_names=sys_fitted.C{i}.coeffnames;
71             coeff_values=sys_fitted.C{i}.coeffvalues;
72             for k=1:numel(coeff_names)
73                 formula=strrep(formula,coeff_names(k),['(',num2str(coeff_values(k)),
74                 ')']);
75             end
76             var_names=sys_fitted.C{i}.varnames;
77             for k=1:numel(var_names)
78                 formula=strrep(formula,var_names(k),['(',num2str(x(k)),')']);
79             end
80             C(i)=eval(formula{1});
81         end
82     end
83
84 % create the state-space system
85 sys=ss(A,B,C,D);
86 sys.OutputName=sys_fitted.OutputName;
87 sys.InputName=sys_fitted.InputName;
88
89 end

```

Listing 10: Function to bring the fits into a form that can be used with the adaptive Simulink model

```

1 function [coeffs_A,coeffs_B,coeffs_C,input_index_selector]=
2     reshape_fits_for_use_with_varying_ss_block_simulink(sys_fitted)
3 % This function is needed for the use of the fitted systems with the
4 % dynamic ss model.
5 %
6 % It takes the coefficients of the fit functions and puts them into matrices
7 % so they can be passed on to the Simulink block that calculates the
8 % dynamic values of the adaptive ss model.
9 %
10 % Additionally, the output in the command window has to be copied BY HAND to
11 % the respective eval_fitted_sys_for_varying_ss_block function in
12 % Simulink.
13
14 % get the coefficient names
15 cell_of_coefficient_names=coeffnames(sys_fitted.A{1});
16 cell_of_ind_vars_names=indepnames(sys_fitted.A{1});
17
18 % initialize the coefficient matrices
19 coeffs_A=zeros(numel(sys_fitted.A),numel(cell_of_coefficient_names)); % one column
20 % per coeff, one line per element
21 coeffs_B=zeros(numel(sys_fitted.B),numel(cell_of_coefficient_names)); % one column
22 % per coeff, one line per element
23 coeffs_C=zeros(numel(sys_fitted.C),numel(cell_of_coefficient_names)); % one column
24 % per coeff, one line per element

```

```

21
22 % for each matrix of the ss model write the coefficients to the
23 % coefficient matrix.
24 for n=1:numel(sys_fitted.A)
25     if ~isnumeric(sys_fitted.A{n})
26         for k=1:numel(cell_of_coefficient_names)
27             coeffs_A(n,k)=sys_fitted.A{n}.(cell_of_coefficient_names{k});
28         end
29     else
30         coeffs_A(n,1:numel(cell_of_coefficient_names))=sys_fitted.A{n};
31     end
32 end
33
34 for n=1:numel(sys_fitted.B)
35     if ~isnumeric(sys_fitted.B{n})
36         for k=1:numel(cell_of_coefficient_names)
37             coeffs_B(n,k)=sys_fitted.B{n}.(cell_of_coefficient_names{k});
38         end
39     else
40         coeffs_B(n,1:numel(cell_of_coefficient_names))=sys_fitted.B{n};
41     end
42 end
43
44 for n=1:numel(sys_fitted.C)
45     if ~isnumeric(sys_fitted.C{n})
46         for k=1:numel(cell_of_coefficient_names)
47             coeffs_C(n,k)=sys_fitted.C{n}.(cell_of_coefficient_names{k});
48         end
49     else
50         coeffs_C(n,1:numel(cell_of_coefficient_names))=sys_fitted.C{n};
51     end
52 end
53
54 % check which inputs to select
55 input_index_selector=[];
56 for i=1:numel(sys_fitted.names_of_variables_swiped)
57     input_name=sys_fitted.names_of_variables_swiped{i};
58     input_index_selector=[input_index_selector,find(contains({'pref','qref','vnd','
59     vnq','vref','Omegag','Omegaref'},input_name))];
60 end
61
62 fit_function_formula=formula(sys_fitted.A{1});
63
64 for i=1:numel(cell_of_coefficient_names)
65     fit_function_formula=strrep(fit_function_formula,cell_of_coefficient_names{i},[
66     coeffs(' ,num2str(i),'')]);
67 end
68 for i=1:numel(cell_of_ind_vars_names)
69     fit_function_formula=strrep(fit_function_formula,cell_of_ind_vars_names{i},[
70     'x(' ,num2str(i),'')]);
71 end
72
73 disp('*** Copy this to the MATLAB function block
74     eval_fitted_sys_for_varying_ss_block in Simulink ***')
75 disp(['element = ',fit_function_formula,'];')
76 end

```

Listing 11: Code for Simulink block that calculates numerical values of state-space model

```

1 function [A,B,C,D]=eval_fitted_sys_for_varying_ss_block(u, u_default, coeffs_A, coeffs_B
, coeffs_C)
2
3     x=u-u_default;
4
5     number_of_states=sqrt(size(coeffs_A,1));
6     number_of_inputs=size(coeffs_B,1)/number_of_states;
7     number_of_outputs=size(coeffs_C,1)/number_of_states;
8
9     A=zeros(number_of_states);
10    B=zeros(number_of_states,number_of_inputs);
11    C=zeros(number_of_outputs,number_of_states);
12    D=zeros(number_of_outputs,number_of_inputs);
13
14    element=0;
15    coeffs=0;
16    for n=1:(numel(A)+numel(B)+numel(C))
17        switch true
18            case n<=numel(A)
19                coeffs=coeffs_A(n,:);
20            case n>numel(A) && n<=numel(A)+numel(B)
21                coeffs=coeffs_B(n-numel(A),:);
22            case n>numel(A)+numel(B)
23                coeffs=coeffs_C(n-numel(A)-numel(B),:);
24        end
25
26        % insert the command window output from the MATLAB function
27        % reshape_fits_for_use_with_varying_ss_block_simulink.m here!
28        %-----
29        element = coeffs(1)*x(1)^2 + coeffs(2)*x(1) + coeffs(3);
30        %-----
31
32        switch true
33            case n<=numel(A)
34                A(n)=element;
35            case n>numel(A) && n<=numel(A)+numel(B)
36                B(n-numel(A))=element;
37            case n>numel(A)+numel(B)
38                C(n-numel(A)-numel(B))=element;
39        end
40    end
41 end

```