

INAUGURAL – DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich–Mathematischen Gesamtfakultät
der
RUPRECHT–KARLS–UNIVERSITÄT
HEIDELBERG

vorgelegt von
M.Sc. Robert Scholz
aus Langenhagen

Tag der mündlichen Prüfung

...

MODEL - BASED OPTIMAL FEEDBACK
CONTROL FOR MICROGRIDS

Advisor

PROF. DR. EKATERINA KOSTINA
PROF. DR. DR. H.C. HANS GEORG BOCK

Zusammenfassung

Modellprädiktive Regelung (MPC) ist eine etablierte Technik zur Prozesssteuerung. Sie basiert auf einem dynamischen Modell, das zur Vorhersage und Optimierung eines dynamischen Prozesses genutzt wird. In festen Abtastintervallen werden Messungen des geregelten Prozesses durchgeführt und in ein parametrisches Optimalsteuerungsproblem (OCP) eingebettet. Die Lösung dieses OCPs wird verwendet, um eine optimale Regelantwort zu berechnen. Diese Methodik ermöglicht es der Regelung auf Störungen optimal zu reagieren. In dieser Arbeit konzentrieren wir uns auf MPC Schemata, die auf gewöhnliche Differentialgleichungen (ODE) oder Differential-algebraische Gleichungen (DAE) basieren und die eine Mehrzielmethode zur Diskretisierung verwenden.

Eine der größten Herausforderungen von MPC ist die Rechenkomplexität der auftretenden OCPs. Die Echtzeit-Iteration (RTI) ist ein Ansatz zur Verringerung des Rechenaufwands durch die Ausnutzung von strukturellen Ähnlichkeiten zwischen aufeinander folgenden OCPs. Die Multi-Level-Iteration (MLI) ist eine Erweiterung des RTI-Schemas, die auf eine weitere Verringerung des Rechenaufwands durch Wiederverwendung von Simulationsdaten abzielt. Zu diesem Zweck wird eine Hierarchie von Aktualisierungsformeln mit zunehmender Komplexität, aber auch mit stärkeren Konvergenzeigenschaften definiert. Diese Aktualisierungsformeln stellen individuelle MPC-Schemata dar, die aber auch parallel ausgeführt werden können. In dieser Arbeit geben wir einen Überblick über die aktuellen Methoden und beschreiben, wie die einzelnen Ebenen zu ganzheitlichen Schemata kombiniert werden können. Wir schlagen einen neuartigen Scheduling-Algorithmus vor, der speziell auf Anwendungen mit hohen Abtastfrequenzen zugeschnitten ist.

Eine genaue Zustandsschätzung ist eine wichtige Voraussetzung für schnelle Regelungsverfahren wie MPC. Für eine effiziente Prozesssteuerung ist es von großer Bedeutung, dass die Zustandsschätzung so schnell wie möglich durchgeführt wird, um den Rückkopplungsmechanismus mit präzisen Informationen zu versorgen und schnelle Reaktionen im Falle von Störungen zu ermöglichen. Die Zustandsschätzung auf bewegten Horizonten (MHE) ist eine modellbasierte Methode zur online-Zustandsschätzung, die auf dynamischer Optimierung aufbaut. Das Modell wird an eine begrenzte Anzahl von vergangenen Messungen angepasst, um den aktuellen Zustand vorherzusagen. Die Methode weist strukturelle Ähnlichkeiten zu MPC auf

und daher kann die RTI auch zur Verringerung des Rechenaufwands von MHE-Schemata eingesetzt werden. Wir präsentieren eine neue Methode, wie die MLI-Aktualisierungsformeln auf die RTI für MHE angewendet werden können, um die Rückkopplungsraten zu erhöhen. Dabei schlagen wir eine Reformulierung des MHE-Problems vor, die nicht nur die Anwendung von MHE-Aktualisierungsformeln, sondern auch deren parallele Auswertung ermöglicht.

Die algorithmischen Entwicklungen in dieser Arbeit sind durch den Bedarf an neuen Regelungskonzepten für Microgrids (MGs) motiviert. Die Energiewende führt zu einer zunehmenden Anzahl von Erzeugern erneuerbarer Energien im Stromnetz, die sich durch eine hohe Volatilität auszeichnen und deren effiziente Integration eine wachsende Herausforderung darstellt. MGs gelten als Schlüsseltechnologie für die Einbindung von erneuerbaren Energien in das Versorgungsnetz. Sie ermöglichen es, lokale Erzeuger und Verbraucher von elektrischer Energie als eine einzige steuerbare Einheit zusammen zu fassen. Jedoch ist auch die Steuerung von MGs eine Herausforderung. Aktuelle Steuerungsansätze stoßen aufgrund des zunehmenden Anteils erneuerbarer Energien und deren Volatilität an ihre Grenzen. In dieser Arbeit fassen wir die Regelungsstruktur von MGs zusammen und führen vollständige transiente Modelle für die Netzwerkdynamik sowie für die wichtigsten Komponenten ein. Das Ziel dieser Arbeit ist es, die Anwendbarkeit von MPC auf die Steuerung von MGs zu untersuchen. Um die Wirksamkeit der vorgeschlagenen mathematischen Methoden zu zeigen, demonstrieren wir ihre Fähigkeiten in anspruchsvollen Lastszenarien für realistisch dimensionierte MGs in numerischen Experimenten.

Abstract

Model Predictive Control (MPC) is a well-established technique for process control. It is based on a dynamical model that is used to predict and optimize the behavior of a dynamical process. In fixed sampling intervals, measurements of the controlled process are carried out and they are embedded into a parametric Optimal Control Problem (OCP). The solution of this OCP is used to generate an optimal feedback control answer, which is applied to the process. This methodology enables the controller to react to disturbances in an optimal manner. In this thesis, we focus on MPC schemes, which consider Ordinary Differential Equation (ODE) or Differential Algebraic Equation (DAE) models, and employ the multiple shooting discretization for the arising OCPs.

One of the main challenges of MPC is the computational complexity of the arising OCPs. The Real-Time Iteration (RTI) is a well-established approach to reduce the computational demand by exploiting the similarities of subsequent OCPs. The Multi-Level Iteration (MLI) is an extension of the RTI scheme that aims at a further reduction of the computational cost by reusing simulation data. For this, it defines a hierarchy of update formulas with an increasing computational complexity, but also with stronger convergence properties. These update formulas state individual MPC schemes, but they can also be executed in parallel. In this thesis, we review the current methodologies and describe how the individual levels can be combined to holistic schemes. We propose a novel scheduling algorithm that is especially tailored to applications with high sampling frequencies.

Accurate state estimation is a vital prerequisite for fast feedback control methods such as MPC. For efficient process control, it is of great importance that the estimation process is carried out as fast as possible to provide the feedback mechanism with precise information and enable fast reactions in case of disturbances. Moving Horizon Estimation (MHE) is a model-based methodology for online state estimation, which builds upon dynamic optimization. The model is fitted to a limited number of past measurements in order to predict to current state. The methodology has structural similarities to MPC and thus, the RTI can be applied to reduce the computational effort of MHE schemes, too. We present a new method to apply the MLI update formulas to the RTI for MHE (RTI-MHE) in order to increase feedback rates.

Therefore, we propose a reformulation of the MHE problems that does not only allow the application of MHE update formulas, but also their parallel evaluation.

The algorithmic developments in this thesis are driven by the need for new control concepts for Microgrids (MGs). The energy transition leads to an increasing number of Renewable Energy Resources (RES), which are characterized by a high volatility, and their efficient integration poses a rising challenge. MGs are considered a key-technology to incorporate RES into the utility grid, because they allow to cluster provider and consumer of electrical energy locally as a single controllable unit. However, the control of MGs is challenging itself and current control approaches reach their limits due to the rising penetration of RES and the fast electrical system dynamics. In this thesis, we summarize the control structure of MGs and introduce full transient models for the networks dynamics as well as for the most important components. The goal of this thesis is to investigate the applicability of MPC to MG control. To prove the efficacy of the proposed mathematical methods, we demonstrate their capabilities in challenging load scenarios for realistically sized MGs in numerical experiments.

Acknowledgements

My deep gratitude goes to all who have inspired and supported me in this work. I gratefully acknowledge the funding and support of the German Federal Ministry of Education and Research (BMBF) in the research project *Modellierung, Optimierung und Regelung von Netzwerken heterogener Energiesysteme mit volatiler erneuerbarer Energieerzeugung* (MORENet), the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp) and Graduate Academy of the Heidelberg University.

I would like to sincerely thank my supervisors and mentors Ekaterina Kostina and Hans Georg Bock for their constant support, their advice, and their ongoing trust in my capabilities.

Special thanks also go to my cooperation partners in the MORENet research project. In particular, I would like to thank Armin Nurkanović and Amer Mešanović from Siemens and Jürgen Gutekunst from the IWR for the gainful collaboration. Furthermore, I thank Andreas Potschka for his support and mentorship. I thank Matthias Schlöder, Ihno Schrot, Conrad Leidereiter, Jürgen Gutekunst and Jing Xu for reading and commenting on parts of this thesis and I thank all members of my workgroup for many productive discussions and helpful hints.

I thank Herta Fitzer, Jeanette Walsch and Ramona Ludwig for the support in administrative matters personally and as a representative for the administration of the Heidelberg University. For the general conditions of the doctorate, I thank the Faculty of Mathematics and Computer Science at the Ruprecht Karl University of Heidelberg. Furthermore, I would like to thank all those who have supported me in my work but are not explicitly mentioned.

Contents

1	Introduction	1
1.1	Model-based Feedback Control	1
1.2	The Challenge of Microgrid Control	4
1.3	Contributions of this Thesis	6
1.4	Structure of the Thesis	10
2	Control of Microgrids	13
2.1	Microgrid Concept	16
2.2	Hierarchical Control Structure of Microgrids	16
3	Mathematical Modeling of Power Systems	21
3.1	Steady State Modeling	21
3.2	Dynamic Modeling	34
3.3	Microgrid Components	40
4	Nonlinear Model Predictive Control	61
4.1	Principle of Nonlinear Model Predictive Control	62
4.2	Process Model	64
4.3	Parametric Optimal Control Problem	65
4.4	Classical Nonlinear Model Predictive Control	68
4.5	Sequential Quadratic Programming	69
4.6	Real-Time Iteration	71
5	Multi-Level Iteration	77
5.1	MLI Update Formulas	78
5.2	Combined MLI Schemes	84
5.3	Adaptive Level Choice by Computation Time	90
6	Multi-Level Iteration for Secondary Microgrid Control	95
6.1	Case Study with One Generator	95
6.2	Grid Control with Adaptive Level Choice by Computation Time	102

6.3 Numerical Results	105
7 Moving Horizon Estimation	109
7.1 The Principle of MHE	110
7.2 A Generalized Gauß-Newton Method Solution Framework	116
7.3 Modified RTI-MHE Formulation	119
7.4 Multi-Level Iteration for Moving Horizon Estimation	127
7.5 Arrival Cost Update Procedure	143
8 Moving Horizon Estimation for Secondary Microgrid Control	151
8.1 Microgrid Model	151
8.2 Numerical Results	153
9 Conclusion	159
Bibliography	163
Figures, Tables & Acronyms	174

Chapter 1

Introduction

Feedback control mechanisms are ubiquitous in industry, society and nature. Whether developed by man or created by nature, they are a vital prerequisite for the stability of many processes that surround us. Process control is one of the key research areas that enables the stable and safe operation of increasingly complex industrial plants. At the same time, the development of ever more sophisticated techniques constantly pushes the efficiency of many plants forward. *Model-based feedback control* is one of the latest advancements in the field of process control, which is based on mathematical optimization and its wide theoretical background. It is characterized by a great flexibility and has proven a high control performance in many applications from a wide variety of areas.

1.1 Model-based Feedback Control

The concept of a *dynamic process* is of fundamental importance in the field of applied mathematics, because it has a very wide applicability to procedures from the real world. It refers to phenomena that are evolving over time. In most relevant application cases, the evolution of a dynamic process is observable or measurable and it is influenced externally, e.g., by disturbances or control actions. Typical examples are chemical reactions in a plant, a car moving on a track or the movements of a robot. However, in many situations it is difficult, time-consuming or expensive to measure the behavior of a dynamic process and therefore, there is a great interest in mathematical tools to predict its behavior under certain conditions. An accurate simulation of a dynamic process may save a lot of time and resources.

In order to describe the behavior of a dynamical process accurately, a predictive mathematical model of it is needed. A model allows to predict the behavior of a dynamic process given an initial condition and external influences, like control actions or disturbances. Depending on the phenomenon of interest, there are many modeling techniques available tailored to the considered processes and time scales [56]. The development of a suitable model is often an intensive research task and requires a deep understanding of the underlying physics. In addition, most models are de-

pendent on parameters, which need to be tuned to the considered situation. However, a calibrated and validated model of a dynamic process allows accurate simulations of its behavior and thus may save many expensive and time-consuming experiments. In this thesis, we focus on processes modeled by Ordinary Differential Equations (ODEs) or Differential Algebraic Equations (DAEs), which can be simulated using a wide variety of integration methods.

Many processes can be controlled by external control inputs over time. The effect of these control actions are often not directly visible and therefore it is an important question, how to control the process optimally in a specific, predefined way. This leads to the problem class of *optimal control* [9]. The desired process behavior is defined in terms of an objective function and constraints. This allows to formulate an Optimal Control Problem (OCP). Its solution trajectory refers to the optimal process behavior in the sense of the objective function. Thereby, the DAE model of the process emerges as constraints of the OCP in order to ensure that the solution trajectory satisfies the model dynamics. The resulting optimal control trajectory is applicable to the considered process in practice. Since the process and therefore the control input is dependent on time, an OCP is an infinite-dimensional optimization problem, which is challenging to solve. Solution approaches for OCPs can be divided into direct and indirect methods. Indirect solution methods apply a *first-optimize-then-discrete* approach. A common methodology is to use Pontryagin's maximum principle [97, 89] in order to define optimality conditions in an infinite-dimensional function space. The resulting equation system is then discretized to compute a numerical solution. In this thesis, we focus on direct approaches, where the OCP is first discretized [14, 11] and then optimized by a suitable Nonlinear Programming Problem (NLP) method [80].

Mathematical models are unable to describe processes exactly in every detail. Instead, mathematical models only consider the dynamics and time scale which are suitable for the intended purpose. Therefore, the process is affected by phenomena, which are not considered by the model and that emerge as disturbances and other unforeseen events. Differences between the actual process behavior and the model predictions are called *plant-model mismatches*. Depending on the process, these plant-model mismatches are accumulating over time and lead to a growing deviation between the predictions and the actual process behavior. A precomputed optimal control trajectory is unable to react to this deviation and may result in a sub-optimal control performance. Larger disturbances may even cause a violation of the operational constraints. Model Predictive Control (MPC) [94, 44, 19, 20] is a technique to take the plant-model mismatches in consideration. The main idea is to not solve an individual OCP, but instead a sequence of similar optimization problems

parameterized by the current system state. In fixed sampling intervals, the system state is measured or estimated. The control answer is generated by the solution of an OCP, which represents the optimal process behavior given the current system state. This allows to adjust the optimal feedback control to current system state constantly and to compensate for plant-model mismatches.

One major disadvantage of classical MPC algorithms is the computational complexity. In every iteration a OCP needs to be solved, which may require many iterations. This limits the sampling rate of MPC algorithms for nonlinear systems with stiff dynamics. The Real-Time Iteration (RTI) [24] is a technique to reduce the computation time significantly. It exploits the similarities between subsequent optimization problems in a closed loop feedback control setup. Instead of solving an NLP until convergence in every sampling interval, only one iteration is performed. Because of a careful initialization of the optimization problem, a single iteration yields an approximation of the real solution of the problem. Furthermore, the RTI reduces the delay between the arrival of new measurements and the feedback control action significantly, by a suitable preparation procedure.

Even though the RTI allows to reduce the computational effort of an MPC algorithm greatly, a simulation and the corresponding sensitivity generation of the underlying DAE system is still necessary in every iteration. This is a computational expensive task for nonlinear systems. The Multi-Level Iteration (MLI) [113] is an extension of the RTI to address this issue. It avoids the full integration and sensitivity generation by performing only partial updates. Thereby the computation times of the individual iterations are reduced even further and very high sampling rates are enabled. The MLI defines four different sets of update formulas called *levels* organized in a hierarchical order. The levels have an increasing computational complexity, but also stronger convergence properties. Every level states an individual MPC algorithm, but it is also possible to operate multiple levels in parallel. This allows to utilize the better performance of the higher levels and the fast feedback times of the lower levels at the same time.

One vital prerequisite for every feedback control strategy is an accurate knowledge of the current state of the process. Therefore, it is necessary to carry out measurements in every sampling interval. However, these measurements may be affected by measurement errors and not all quantities may be directly measurable. State or parameter estimation is a class of algorithms to determine the actual system state from a series of measurements. One of the most popular techniques is the so-called Kalman filter [39] for linear systems, where in every sampling interval prior knowledge and current measurements are used to estimate the current state.

Even though the Kalman filter has extensions to nonlinear systems [22], it is rather inflexible. Moving Horizon Estimation (MHE) [92, 93] is a more general approach to state or parameter estimation, which implements similar ideas as MPC. A model of the observed system is used to set up a dynamic optimization problem, which considers a limited number of past measurements. A least-squares objective function is used to minimize the distance between the model predictions and the past measurements. Thereby it is possible to consider the different error ranges of the individual measurements and operational as well as physical constraints. The state estimate is generated by a simulation of the model, which is fitted to the measurements. MHE can be seen as a dual technique to MPC, since in both cases a model of the process is embedded in a dynamic optimization problem. While in MPC, this model is used to optimize the future behavior of the process, in MHE it is used to reconstruct the past behavior.

MHE shares the main algorithmic ideas with MPC and therefore it inherits not only its flexibility, but also its computational complexity. In every sampling interval a NLP is solved, which may be a demanding task and limits the applicability to processes with low sampling frequencies. However, it is possible to apply similar techniques as the RTI and MLI to MHE in order to increase the sampling rates. The RTI for MHE (RTI-MHE) [59, 28] avoids to solve an NLP until convergence in every iteration. Instead, the similarities between subsequent sampling points are exploited to construct an accurate initial guess of the solution. On every sampling point, only one iteration is performed to correct this initial guess towards the actual solution. This allows to reduce the computational effort greatly, while maintaining accurate state and parameter estimates. In this thesis we introduce the MLI for MHE (MLI-MHE) [47], which avoids the numerical integration of the dynamical system and the evaluation of the corresponding sensitivities. Instead already evaluated system linearizations are reused and adjusted to the recent measurements. The MLI-MHE defines a hierarchy of update formulas, which represent individual MHE schemes, but can be operated in parallel too.

1.2 The Challenge of Microgrid Control

The energy transition describes the ongoing process of replacing fossil energy sources in all sectors of industry and society. The main carrier of energy will be electricity generated by heterogeneous Renewable Energy Resources (RES) and therefore the electrical grid will play a central role for the supply of energy. The most common sources of renewable electrical energy are wind turbines and Photovoltaic (PV) systems and their share will increase significantly. However, RES have exceedingly dif-

ferent properties than traditional power plants and therefore the energy transition will change the structure of the utility grid fundamentally.

- Traditional power plants are fueled externally and this leads to a very predictable energy output. In contrast to that, PV systems and wind turbines are dependent on the current weather conditions and since the weather is volatile, the energy output of RES is volatile too. In addition, extensive power storage is necessary to compensate for temporal shortages of renewable energy.
- The scale of RES is usually much smaller than the scale of traditional power plants and therefore a much higher number of RES units is necessary to provide the same amount of electrical energy. This will lead to much higher number of smaller power plants.
- The frequency in the utility grid is physically coupled with the rotational speed of the generators in traditional power plants. Since these generators have a high mass and store a lot of rotational energy, the frequency has a lot of inertia and is resilient against disturbances. In contrast to that, RES are usually connected to the utility grid by power inverters, which have no internal inertia. Therefore, a high share of RES increases the susceptibility of the utility grid.
- In contrast to traditional power plants, small scale RES are connected mainly to low or medium voltage networks and in order to use them efficiently, electrical energy needs to be transferred between distribution networks. Therefore, the introduction of RES leads to a multidirectional power flow.

In summary, RES are harder to control, more RES are necessary and they behave highly unpredictable. Therefore, the energy transition changes not only the structure of the utility grid, but poses also a huge challenge to current control concepts for electrical networks.

Microgrids (MGs) are considered a key-technology in the energy transition, since they address the challenge of an increasingly complex utility grid. MGs are small, local, electrical networks comprising heterogeneous components, such as generators, storage systems, and loads. Usually, they are connected to the utility grid and thus offer a flexible way to integrate RES. Their main advantage is the ability to handle disturbances and the intermittent behavior of RES locally and hence they can be seen as single controllable entities in a larger electrical network [17]. Thereby, not every RES needs to be considered as an individual plant, but they are clustered into bigger, more controllable units. In addition to that, they can also be operated in an islanded mode after a disturbance, leading to a more fail-safe power supply. MGs are

also used on smaller islands or in remote geographical areas, where no connection to the utility grid is possible.

However, the control of a MG poses a challenge itself. Because of the small scale, the individual RES units are exposed to changing weather conditions more directly and their energy input is harder to predict. The proportion of RES in the network is higher and since they are inverter-interfaced, the inertia in the network is low and disturbances have a more direct impact. Furthermore, a MG may also include electrical consumers, which behave unpredictable on a short time scale too. A further challenge of MG control are the different time scales. While the electrical dynamics take place on a particular short time scale, load prediction and weather forecast consider a time horizon of at least 24 hours.

MGs are typically organized in a hierarchical structure, where the different control layers address different time scales. The higher layers consider longer time scales with low sampling time and communicate target values to the lower control layers. The lower layers implement the provided values on a short time scale, which enables them to react to short-time disturbances. However, the high volatility of RES pushes state-of-the-art control approaches to its boundaries and there is a need for more flexible and efficient control methods. MPC offers a lot of desirable features for the control of MGs, but the established approaches are not tailored for dynamical systems with high sampling frequencies. The ultimate goal of this thesis is to develop new mathematical methods to render MPC feasible for the control of MGs.

1.3 Contributions of this Thesis

The goal of this thesis is to develop a mathematical framework that is able to treat the stiff dynamics of MGs and the high volatility of RES in real-time. Thereby, we focus on a centralized MG controller, which is acting on a short time scale. Existing methods for MPC are neither able to stabilize MGs of moderate size after disturbances nor estimate the system state reliable. The computational effort for the numerical integration of the stiff dynamical system is too high to meet the real-time requirement. To address the challenges of MG control, we develop novel mathematical methods tailored to systems with high sampling frequencies. Thereby, we present new approaches for the controller as well as for the state estimator in order to achieve a full closed loop feedback controller. The mathematical advancements presented in this thesis are motivated by the challenges of MG control, but they state general control schemes that pushes the applicability of MPC and MHE forward.

Not only classical approaches for MPC, but also tailored schemes like the RTI are not able to provide sufficiently accurate control feedback within the required sam-

pling time of MGs. The high feedback rates, the stiffness of the MG models, and the high intermittency of RES render state-of-the-art methods too slow for the control of medium sized MGs. The MLI offers a flexible methodology to generate control feedback with high sampling frequencies. However, the lower MLI levels allow to give feedback with a sufficiently high feedback rate, but since they give suboptimal feedback they are not able to stabilize the system after disturbances. In order to meet the real-time requirement and to stabilize the system at the same time, we develop a novel MPC scheme, which is based on the parallel operation of multiple MLI levels. A new scheduling algorithm, which is especially tailored to applications with high sampling frequencies, utilizes the high feedback rates of the lower levels and the accuracy of the higher levels at the same time. The presented MPC scheme guarantees to provide feedback in real-time and shows a superior control performance compared to state-of-the-art control methods.

An appropriate state estimation methodology is a vital prerequisite for the application of MPC to the control of MGs in a closed loop. MHE offers a lot of desired properties, but the fast dynamics of MGs render even tailored MHE schemes as the RTI-MHE too slow. In this thesis, we develop a new flexible framework for MHE. It defines a general class of MHE algorithms, which are customizable to the requirements of a wide range of applications. The main idea is to transfer the algorithmic ideas of the MLI in the context of online state estimation. However, the standard formulation of MHE is not treatable by MLI update formulas. We present a new MHE problem formulation, which is more flexible. With the Control Trajectory Embedding (CTE), we propose a concept to decouple linearizations from constantly changing online data. This allows not only to apply MLI update formulas to MHE, but also enables the parallelization of multiple levels. We investigate the convergence properties of these schemes by applying the convergence theory of Newton-type methods. We use this framework to define MHE schemes tailored to the online state estimation of MGs. Thereby, we achieve high accuracy state estimates in real time even after major disturbances in the network.

We demonstrate the capabilities of the newly introduced mathematical approaches by comprehensive numerical experiments. Therefore, we consider DAE models of realistic sized test MGs with PV systems, batteries and Synchronous Generators (SGs). We use these models to develop a new MPC scheme with state estimation that is able to respect the operational bounds and the high sampling frequencies of MGs. It is especially tailored to the hierarchical control structure of MGs. We challenge the new methods for MPC as well as MHE in challenging scenarios and compare the results to state-of-the-art control approaches. To the best of our knowledge, it is the

first time that a methodology is presented, which is able to stabilize MGs with full dynamic models in a full closed loop control setup with state estimation.

The main methodological approaches of this thesis are based on the following publications, of which we are the main contributor:

- R. Scholz, A. Nurkanovic, A. Mesanovic, J. Gutekunst, A. Potschka, H. G. Bock, and E. Kostina. Model-Based Optimal Feedback Control for Microgrids with Multi-Level Iterations. In *Operations Research Proceedings 2019. Operations Research Proceedings (GOR (Gesellschaft für Operations Research e.V.))*, 2020. doi: 10.1007/978-3-030-48439-2_9
- R. Scholz, A. Nurkanović, A. Mešanović, J. Gutekunst, A. Potschka, H. G. Bock, and E. Kostina. Multi-level iterations for microgrid control with automatic level choice. In *Scientific Computing in Electrical Engineering*. Springer International Publishing, 2020. In Press
- J. Gutekunst, R. Scholz, A. Nurkanović, A. Mešanović, H. G. Bock, and E. Kostina. Fast moving horizon estimation using multi-level iterations for microgrid control. *at - Automatisierungstechnik*, 68(12):1059–1076, 2020. doi: 10.1515/auto-2020-0081

This thesis associates these articles in the context of MPC for MG control and gives a more comprehensive description of the individual contributions. In particular, this thesis contributes to the field of applied mathematics by the following aspects.

Mathematical Modeling of MGs

We give a comprehensive introduction in the structure and control concepts of MG. Thereby we review common modeling techniques for electrical networks in steady-state as well as full transient models. We introduce Optimal Power Flow (OPF) as the basis for optimization based control techniques for electrical networks and we present full transient models for most important components in MGs. These models emerge as constraints in the OCPs of this thesis and [103, 102, 47].

Adaptive Level Choice by Computation Time

We present a new scheduling algorithm for MLI update formulas, which is tailored to applications with high sampling frequencies and stiff dynamics. Performing a parallel evaluation of multiple update formulas, this algorithm enables a high control performance, while guaranteeing to stay real-time feasible. This approach is published in [103].

Development of Tailored MPC Scheme for MG Control

Based on the new methodological approaches of this thesis, we propose a novel MPC scheme that is tailored to MG control. We model a suitable parametric OCP to be used in the context of MPC. The design of the resulting MPC scheme is geared to the hierarchical control structure of MGs and is compatible with control methods for the primary and tertiary control level. Compared to state-of-the-art control approaches for MGs, the MPC shows not only a superior performance, but it is also able to respect operational bounds. We show the capabilities of new numerical methods in comprehensive numerical studies in challenging load scenarios.

Extensive Numerical Experiments for Control of MGs

We present comprehensive experimental data for the control of MGs using the presented numerical approaches. Thereby, we consider realistic-sized MGs in challenging load scenarios. We compare our new proposed approach with a classical MPC approach and a state-of-the-art control strategy based on Proportional-Integral (PI) controller. Our methodology shows a superior control performance compared the other approach and stays real-time feasible. In this thesis, we present the numerical results of [103, 102] in an extended way.

Control Trajectory Embedding

We present a reformulation of the classical MHE problem, which decouples system linearizations from constantly changing online data. This allows to reuse linearizations over multiple sampling intervals, which is necessary for the application of MLI update formulas. Initially, we introduced this reformulation in [47].

Multi-Level Iteration Update Formulas for Moving Horizon Estimation

We introduce a novel set of hierarchical update formulas for MHE based on the MLI. In order to enable them to be applied in parallel, we present shifting strategies for the optimization variables as well as system linearizations. In this thesis, we give a more detailed description of the originally proposed formulas in [47].

Arrival Cost Term for MHE for Parallel MLI levels

In MHE the arrival cost term allows to incorporate measurements prior to the current estimation horizon. This term is updated in every iteration, which includes a full system reevaluation. We embedded the arrival cost update in the framework of

MLI update formulas in order to avoid system linearizations in every iteration. We give detailed description of this update procedure, which we originally proposed in [47].

Extensions of Software Package

We implement the proposed algorithmic extensions into the existing software framework *MLI* [113] in order to apply MLI update formulas to MPC and MHE. These extensions include the ability to treat DAE systems, the new scheduling algorithm, the CTE formulation for MHE and the MLI-MHE update formulas. The framework allows to define holistic MPC schemes with parameter estimation in a general way and is customizable to a wide range of applications.

Closed Loop Simulations for MHE

We show the capabilities of the newly developed methods for MHE in comprehensive numerical experiments, which consider closed loop as well as open loop scenarios. Thereby, we use full transient models of realistic sized MGs. The experimental data show that the proposed MHE framework is able to provide state estimates in real-time.

1.4 Structure of the Thesis

This thesis is structured into three major parts, which consider the modeling of MGs, methods for MPC and methods for state estimation. The structure of the individual chapter and their mutual dependencies are depicted in Figure 1.1.

The first part of the thesis is dedicated to the application case of MGs. In Chapter 2, we describe the role of MGs in the energy transition from a high level perspective. We describe the structure and the challenges of MG control. In Chapter 3, we present general modeling techniques for Alternating Current (AC) electrical networks and we present dynamical models of the most important components.

The second part considers Nonlinear Model Predictive Control (NMPC). In Chapter 4, we review the principle of NMPC and present the RTI as a technique to enable higher sampling rates. Chapter 5 considers the MLI and we present new methodological schemes which are tailored to the control of MGs. We present the capabilities of the MLI for MG control in numerical experiments in Chapter 6. Thereby, we consider a challenging load scenarios and compare the results with a state-of-the-art control approach.

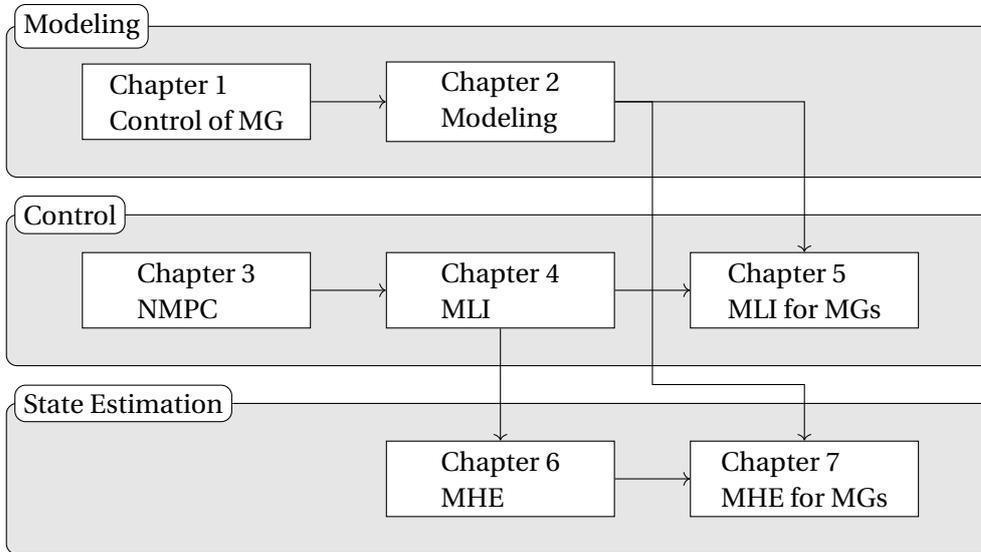


Figure 1.1: Structure of the individual chapters and their mutual dependencies.

In the third part of the thesis, we consider state estimation and MHE. In Chapter 7, we present the RTI for MHE and we introduce CTE as a reformulation of the classical MHE problem in order to make MLI update formulas applicable. Furthermore, we propose MLI update formulas for MHE and we propose a tailored update procedure for the arrival cost term. In Chapter 8, we test the performance of the presented methods in open and closed loop simulations.

In Chapter 9, we give a summary of the most important results of the previous chapters.

Chapter 2

Control of Microgrids

Climate change and limited resources make a transition from fossil-fueled energy production to Renewable Energy Resources (RES) inevitable [70]. In the context of the energy transition, heating and mobility will be driven primarily by electrical energy and therefore electricity will be the most important way to transport energy [107, 90]. In order to prevent greenhouse emissions fossil-fueled power plants need to be replaced by RES as the main source of electrical energy. The most viable RES are Photovoltaic (PV) systems and wind turbines and they have fairly different properties than traditional power plants. This will change the structure of the utility grid fundamentally and thereby new challenges will arise.

The traditional power grid is characterized by large power plants and is organized in a top-down fashion by interconnected subnetworks with different voltage levels. On the top layer the *transmission grid* has the purpose to interconnect different subnetworks and to transfer electrical energy. It uses a high voltage to reduce transmission losses and to enable the efficient transport of electrical energy over long distances. Most of the traditional power plants with a high nominal capacity are directly connected to the transmission grid. On the bottom layer *distribution grids* with low or medium voltage levels are used to distribute electrical energy to the customers. Such a distribution grid may cover cities or industrial plants and may be further divided into smaller networks. The power flow in the traditional power grid is mostly unidirectional. The majority of the electrical energy is produced by large power plants and is directly injected in the transmission grid. It is delivered to the consumer via distribution networks. The customers are connected to the distribution grid on different voltage levels, depending on their size.

The frequency plays an important role for the control of Alternating Current (AC) power grids because it is directly dependent on the rotational speed of the generators in the power plants. Mostly Synchronous Generators (SGs) are used and their rotational speed is in synchronization with the grid frequency. A mismatch between the produced and the consumed electrical energy in the network leads to an acceleration or a deceleration of the frequency and thus of the rotational speed of the generators. Therefore, the rotational energy of the rotating parts of the generators com-

compensates for the power imbalances and stabilizes the frequency. This physical coupling between the frequency and the rotational speed of the generators is an intrinsic feedback control mechanism, which counteracts disturbances in the network. The impact of power imbalances is delayed by this internal inertia of the utility grid and therefore the controllers have a certain time frame to react. A high physical mass of generators leads to a high inertia and therefore the more generators are connected, the more stable a network is. In addition, the frequency is a convenient feedback signal to detect power imbalances, since the frequency is almost the same at all buses of the electrical network. A deviation of the frequency from the nominal value can be counteracted by a power adjustment of the plants. Since traditional power plants are fueled mostly externally by coal, oil or gas, their power output is very predictable and can be adapted to frequency variations.

A detailed description of the physical properties of power systems and control concepts can be found in [60, 40] and a survey of modern control approaches is given in [79]. Simulation based approaches model the power flow in the utility grid in a steady state by the nonlinear power flow equations [36, 53] or approximations of them [21]. The optimal dispatch is computed by an Optimal Power Flow (OPF) problem, which will be introduced in Chapter 3. The OPF problem is a basic formulation and many solution approaches were proposed tailored to specific situations. Solution approaches for classical formulation are collected in [37]. OPF in the context of Model Predictive Control (MPC) are proposed in [72, 71, 55, 32] and stochastic and distributed OPF is described in [33, 78].

The energy transition and the accompanying integration of RES is not only a replacement of physical components, but it will change the structure of the utility grid fundamentally [17]. Several challenges for the control will arise.

Uncertainty PV systems and wind turbines are the most important sources of renewable energy and they have an intrinsic intermittency because of their dependence on the weather. The weather is changing on different time scales. The intensity of solar radiation and wind is roughly predictable over the course of one or several days. However, they are also affected by changing weather conditions on short time scale, which are much harder to predict [54, 55]. Ensuring a reliable supply of electrical energy is more challenging, because these uncertainties need to be considered. A suitable control approach needs to consider a time horizon of at least a day, while being able to react to sudden disturbances almost instantaneously.

Scale RES are typically much smaller than traditional power plants and they are much more spatially distributed. Therefore, a much higher number of indi-

vidual RES units is needed to provide the same nominal power as a traditional power plant. In addition, it is to be expected that the demand for electrical energy will rise because of the electrification of mobility, heating and other sectors [91]. Furthermore, RES have quite different control properties, depending on their primary energy source like wind or solar radiation. Thus, the future utility grid will comprise a much higher number of active energy providers with heterogeneous control characteristics.

Inertia The traditional power grid is resilient against disturbances because of its high inertia. In contrast to that, RES are usually connected to the grid by power inverter, which convert Direct Current (DC) to AC. Power inverter are controlled by power electronics, which allow an independent control of the voltage level at the terminals. They have no internal inertia and their energy flow needs to be controlled in order to meet the demands and to stabilize the frequency. The rising number of RES units reduces the system inertia and concepts are needed to synchronize the inverters.

Bidirectional Power Flow Traditional power plants are mostly connected to the transmission grid and therefore their energy flow is always directed from the high voltage to the low voltage grids. The structure of the energy grid is shaped by its historical development and is adjusted to this unidirectional power flow. However, RES are usually much smaller than traditional power plants and much more spatially distributed. Therefore they are connected to the distribution networks on low or medium voltage levels. In order to transfer electrical energy produced by RES over long distances it needs to be injected into the the transmission grid and this introduces a power flow from the distribution grid to the transmission grid. This poses an additional control challenge, because the low voltage distribution networks emerged predominantly as passive power consumers in the traditional power grid. In addition, more monitoring hardware is needed.

Storage The performance of RES is dependent on current weather conditions and on the time of day. E.g., during the night PV systems produce almost no electrical energy. In order to compensate for this unstable nature of RES and to avoid generation shedding, storage units are necessary. A wide range of energy storage technologies are available, which differ in operating cost, storage capacity and response time. An appropriate control and scheduling of the storage units poses an additional challenge [17].

2.1 Microgrid Concept

The Microgrid (MG) is a concept to address the challenge of rising complexity of the utility grid. A MG is a local cluster of loads, energy providers and storage, which is operated as a single controllable unit. A MG is connected to the utility grid by a Point of Common Coupling (PCC), but has also the possibility to be operated in an islanded mode, e.g., after a disturbance [62, 63, 51]. The clustering of local energy consumer and providers has several practical advantages. From the view of the utility grid, the MG acts as a single controllable unit even though it may include a high number of RES. Because of its heterogeneous components, the MG is able to compensate smaller power imbalances locally. Thus, the MG clusters multiple volatile entities into a single more predictable one. Furthermore, the MG allows to keep the energy production close to loads, which reduces transmission losses. And since the MG can be operated as an individual unit in islanded mode, it is more resilient against disturbances in the utility grid. The MG concept is very flexible and has a wide range of applications. E.g., islanded MGs can be deployed in geographical areas, where a connection to the utility grid is not possible [49] or multiple MGs can be interconnected to form bigger grids [1, 49].

2.2 Hierarchical Control Structure of Microgrids

The challenge of MG control is characterized by phenomena on a wide range of different time scales. On the one hand, the physical behavior of electrical quantities is extremely fast and disturbances have an effect on the voltage and phase angle almost immediately. The nominal frequency in most electrical networks in Europe is 50 Hz and in order to keep it stable, the control hardware needs to react in the same time scale. On the other hand, the weather and load forecast is dependent on the daytime and in order to control a MG efficiently a time horizon of at least a day needs to be considered. To address these different time scales, MG control is usually organized in a hierarchical fashion. The lower control layers react to the fast electrical dynamics and the higher layers consider the long term planing. Every layer uses different models and control techniques, which are suitable for the considered phenomenon. Control signals are sent from the higher levels to lower levels. Measurements from the physical components of the MG are used as a feedback signal in order to enable the control layers to react to disturbances.

The most common control approach considers three hierarchical control layers [83, 17, 10, 54, 23]. A review of common control techniques can be found in [51, 86]. The different levels and their interaction are depicted in Figure 2.1.

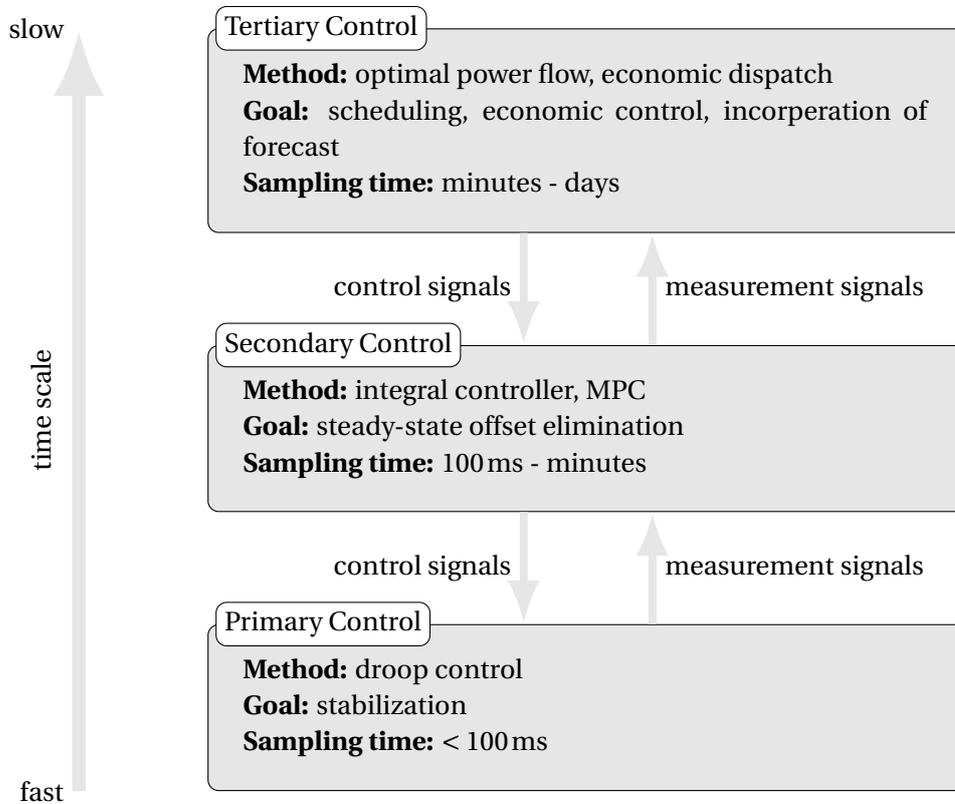


Figure 2.1: Hierarchical control structure of MG.

2.2.1 Primary Control

The goal of the primary control layer is to stabilize the system after disturbances. It is responsible for instantaneous compensation of mismatches between the scheduled and the demanded power. Most of the common active components in MGs are equipped with a primary controller. They rely on local measurements and use droop control laws for instantaneous reaction to load changes. Their response rate is below 100 ms in MGs, but reach up to several seconds in bigger power plants. The performance of the primary control layer depends on a careful tuning of the parameters of the control laws. Since it relies only on local measurements, it is not possible to eliminate state-state offsets for the voltage level [83, 74, 73].

2.2.2 Secondary Control

The purpose of the secondary control layer is to steer the system back to the desired steady-state operating point after a disturbance [45, 17] and to implement control signals from higher control levels. Typical sampling times of the secondary control layer range between 100 ms up to several minutes. It uses the reference values of the primary control layer as controls to interact with the physical hardware of the MG.

Traditionally, secondary control is implemented by local integral controllers, which are tuned to the specific configuration of the MG. This is a flexible approach, since integral controllers consider only local measurements and therefore the MG can be extended without a reconfiguration of the complete grid. However, since the voltage is a local quantity, voltage control by fully decentralized controllers is challenging [105] and requires a stable operation. The rising penetration of RES pushes this approach to its boundaries [54] because of its inherent volatile behavior. There is a high research interest in new control methods [114, 17].

MPC is a control strategy, which gained a lot of attention from the power engineering community lately because of its flexibility and control performance [17, 55, 110]. All MPC approaches are based on models, which are used to predict and optimize the performance of the controlled components. On the secondary control layer it is common to use Ordinary Differential Equations (ODEs) or Differential Algebraic Equations (DAEs) to model the transient behavior of the individual components [100, 6]. The interconnecting network is represented in a rotating reference frame [67] by algebraic equations. Usually, these models include control laws of the primary control layer.

MPC can be applied in a centralized or a distributed fashion. In a distributed MPC approach, the individual components are optimized locally [75, 76, 111, 106] and usually there is also a central control unit, which submits control targets to the individual units. The main advantage of a distributed control approach is that no model of the complete MG is needed and therefore it is easy to include additional components. This is a desired property for bigger electrical networks, where not all the information is available and the structure of the network may change often.

In contrast to that, a centralized MPC control approach uses a model of the complete MG with all controllable units included. This enables the controller to optimize the performance globally taking all measurements at all buses in consideration. Centralized MPC control is especially suited for small MGs, where information about all the components is available and a communication infrastructure is implemented. One of the main challenges of centralized secondary control is the computational complexity of the arising optimization problems. Since the model of the MG includes all

components, the simulation requires a higher computational effort. Several tailored MPC schemes are proposed to address this issue [102, 103, 83, 84], which reduce the computation time significantly.

Scope of this Thesis

In this thesis, we develop mathematical methods for a centralized controller on the secondary level. Thereby, we consider full transient models of the MG and the connected components given by DAEs. They allow accurate predictions of the process behavior in steady-state and after disturbances. However, current schemes for MPC as well as state estimation are not able to meet the real-time requirement for MGs. We propose a new mathematical approach that considers a full feedback loop with controller and state estimator. We discuss the mathematical methodology in Chapter 5 for the controller and in Chapter 7 for the estimator. Compared to state-of-the-art methods, the proposed MPC scheme shows a significantly improved performance. In addition, it is able to respect operational bounds and to control the voltage.

2.2.3 Tertiary Control

The tertiary control layer considers the economic aspects of MG control. It is responsible for the long term scheduling of the individual components and incorporates weather and load forecast. A typical load profile has a high demand during the day and a low demand at night. In addition, the solar radiation is very low in the night. The intensity of wind depends on the season and may change significantly during the day. In order to take the load profile into account, the tertiary control level needs a prediction horizon of at least 24 hours.

Typically, the tertiary control layer uses steady-state models of the MG in order to treat the long prediction horizon [49]. The system state is presented in terms of active and reactive power balances, which satisfy the so-called power flow equations. The power flow equations are a set of nonlinear equations, which characterize physical operating points of the grid. In order to predict the system behavior, the so-called multi-stage power flow equations are solved, which consists of a sequence of power flow equations coupled in time by the state of charge of the storage units [32, 72, 71]. We will discuss the power flow equations and the corresponding optimization problem in Chapter 3. The sample times range from several minutes in smaller MGs up to hours and days for bigger units.

The fundamental assumption of the tertiary control layer is that the secondary control layer is able to keep the MG stable at the desired operating point. The secondary

control layer is responsible to share the additional load between the active components.

On the tertiary control layer, optimal control is a common technique to optimize the operation [77]. From a mathematical point of view, tertiary control is challenging because it may include integer decision variables. Active components, like e.g. SGs, can be switched on or off and may have significant startup costs. A common approach is to relax the nonlinear power flow equations and model the problem as a mixed integer linear program [88] or a mixed integer quadratically constrained program [21]. Depending on the size of the considered MG it may also be beneficial to solve the arising optimization problems in a distributed manner using the ADMM [49, 18] or ALADIN [30] algorithm. A further challenge of the tertiary control layer is to consider uncertainties of the incorporated load and weather forecast. Several approaches were proposed by modeling the uncertainties as random variables [33, 48] or as a minimax MPC scheme [118].

Chapter 3

Mathematical Modeling of Power Systems

In this chapter we consider mathematical modeling of electrical networks and microgrids. First, we will describe the fundamental concepts and terms of electrical networks. We will introduce the phasor as a steady-state modeling technique and we will extend it with the $dq0$ -transformation to full dynamic models. The power flow equations will be used to represent the behavior of a complete electrical network. Afterwards, we will derive full dynamic models for the most important components, which are typical for Microgrids (MGs). Thereby, we focus on Synchronous Generators (SGs) and inverters as the main controllable components. The full dynamical behavior of the grid will be modeled by Differential Algebraic Equations (DAEs).

The models of the network and the components presented in this chapter are used to assemble full transient models of complete MGs. We introduce the models in a general way such that they can be flexible assembled to represent MGs with arbitrary structure. These models are the basis for the development of a novel mathematical control approach for MGs. The corresponding DAE will emerge as constraints in Optimal Control Problems (OCPs) in the later chapters of this thesis. We will use different objective functions in these OCPs, which will be discussed in Chapter 6 and in Chapter 6.

This chapter follows roughly [36] and [67]. A more comprehensive introduction to the modeling of power grids can be found in [40] and a modeling approach tailored to MGs is presented in [17]. The dynamical models for the SG are derived in great detail in [60] and for the inverter in [96].

3.1 Steady State Modeling

There are two fundamental ways for the transfer of electrical energy: Alternating Current (AC) and Direct Current (DC). Whereas DC refers to constant voltage and current signals, AC is an electric current that changes its direction periodically in sinusoidal fashion. Most of the electrical energy is generated in AC, which is a direct consequence of the generator design. Traditional generators use a rotating magnetic field to induce sinusoidal currents at the terminals. One advantageous property of

AC is that the voltage level can be adjusted easily with coil based transformers. Since the power loss on a transmission line is lower for higher voltages, AC enables a more efficient energy transfer. The voltage level of electrical energy is stepped up before transmission. Close to loads, the voltage is stepped down to the required level of the equipment. Therefore, AC is the common way to distribute electrical energy and we will focus on AC throughout this thesis.

In AC power systems current and voltage are given as time-dependent sinusoidal signals. The direct representation of these signals as functions is called *time-domain* representation. However, power analysis using the time-domain may be very laborious. Therefore, it is common to transform these signals into different representations in order to simplify the calculations [40]. In analysis of AC power systems, there is the fundamental distinction between quasi-static and transient models. Quasi-static models assume an operation at fixed frequency and voltage magnitude. Transient models describe the dynamic behavior of the system with time-varying frequencies and magnitudes. Which type of model is suitable for the specific situation depends on the time scale and the considered phenomenon. In this section, we introduce the phasor representation as a quasi-static model of power systems. The $dq0$ transformation will be introduced as a transient generalization of the phasor in a later section.

3.1.1 Phasor Representation

In the following, we assume a quasi-static operation, i.e. a fixed frequency ω and magnitude. Current $i(t)$ and voltage $v(t)$ are given by the time dependent functions

$$\begin{aligned}v(t) &= V_{\text{pk}} \cos(\phi_V + t\omega), \\i(t) &= I_{\text{pk}} \cos(\phi_I + t\omega)\end{aligned}\tag{3.1}$$

with the initial angles ϕ_V and ϕ_I . The peak values are given by V_{pk} and I_{pk} . The functions $v(t)$ and $i(t)$ are the *time-domain* representation of voltage and current. In the *frequency-domain* voltage and current are represented by complex numbers, so-called *phasors*. The phasor of the voltage and current signals is given by

$$\begin{aligned}V &= \tilde{V} e^{j\phi_V}, \\I &= \tilde{I} e^{j\phi_I}.\end{aligned}\tag{3.2}$$

Here we denote the imaginary unit with $j = \sqrt{-1}$. The absolute values of the phasors are given by the Root Mean Square (RMS) values of the time dependent quantities

$\tilde{V} = \frac{V_{pk}}{\sqrt{2}}$ and $\tilde{I} = \frac{I_{pk}}{\sqrt{2}}$. The time-domain representation can be recovered by the real part of the phasor multiplied by $\sqrt{2}e^{j\omega t}$

$$\begin{aligned} v(t) &= \Re(V\sqrt{2}e^{j\omega t}) = \sqrt{2}\tilde{V}\Re(\cos(\phi_V + \omega t) + j\sin(\phi_V + \omega t)) \\ &= V_{pk}\cos(\phi_V + \omega t). \end{aligned} \quad (3.3)$$

In literature [40], it is also common to denote a phasor by the notation $\tilde{V} \angle \phi_V$.

Remark 3.1

Throughout this chapter, we will use the following notations. Signals in the time-domain are time dependent functions, like $v(t)$. Magnitudes of sinusoidal signals are indicated by a subscript pk and the RMS values by a tilde, like V_{pk} or \tilde{V} . Phase angles are denoted by Greek letters as variable names, like ϕ . Variables without tilde and argument refer to signals in the frequency-domain.

3.1.2 Resistors, Inductors and Capacitors

The most fundamental components in AC networks are resistors, inductors and capacitors. They are not only physical components, but also a technique to model other components. For example, transmission lines and loads can be represented as an equivalent circuit of them.

The resistor is a passive component, which represents physical resistance to the current. Its physical properties are specified by the resistance $R \in \mathbb{R}$. Maintaining a flow of current through a resistor consumes electrical power and causes a voltage drop between its terminals. All transmission lines have an internal resistance depending on their length, material and thickness. Ohm's Law describes the relation between voltage and current on a resistor

$$v(t) = Ri(t) \quad (3.4)$$

and this law is applicable to the phasor

$$V = IR \quad (3.5)$$

as well. For time-varying quantities, there are two additional important components: inductors and capacitors. Inductors are governed by the differential equations

$$v(t) = L \frac{di(t)}{dt} \quad (3.6)$$

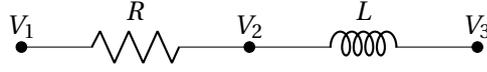


Figure 3.1: Resistor and inductor in series connection.

with the inductance $L \in \mathbb{R}$. In quasi-static operation, this differential equation can be solved directly

$$\begin{aligned} v(t) &= L \frac{di(t)}{dt} = L \frac{d}{dt} (I_{\text{pk}} \cos(\phi_I + t\omega)) = -\omega L I_{\text{pk}} \sin(\phi_I + t\omega) \\ &= -\omega L I_{\text{pk}} \cos\left(\phi_I - \frac{1}{2}\pi + t\omega\right). \end{aligned} \quad (3.7)$$

A comparison of the arguments shows that $\phi_I - \phi_V = \frac{\pi}{2}$ and that an ideal inductor causes a shift between voltage and current by $\frac{\pi}{2}$. In the frequency-domain this can be represented by a multiplication with $-j$

$$V = \omega L j I. \quad (3.8)$$

The scaling factor is called reactance and is usually denoted by $X = \omega L \in \mathbb{R}$. The conductor is governed by the differential equation

$$i(t) = C \frac{dv(t)}{dt} \quad (3.9)$$

with the capacitance $C \in \mathbb{R}$. This equation can be solved in similar way for AC quantities in quasi-static operation

$$i(t) = -C\omega V_{\text{pk}} \cos\left(\phi_V - \frac{1}{2}\pi + t\omega\right). \quad (3.10)$$

In the frequency-domain, the phase shift by $-\frac{\pi}{2}$ can be achieved by a multiplication with $-j$. Solving the equation for V leads to

$$V = X j I \quad (3.11)$$

with the reactance $X = -\frac{1}{\omega C}$. An ideal conductor causes a shift between voltage and current by $\frac{\pi}{2}$ but in the opposite direction as the inductor. Physical components are never purely inductive or capacitive, but they are also resistive. This can be modeled

by a series connection of an inductor or capacitor and a resistor like in Figure 3.1

$$V = V_1 - V_3 = RI + V_2 - V_3 = RI - L\omega jI = (R - L\omega j)I. \quad (3.12)$$

The combined effect of the resistor and inductor is described by the impedance $Z = R + Xj$. In fact every combination of resistors, capacitors and inductors can be represented by a single impedance $Z \in \mathbb{C}$. Therefore the physical properties of a connection in a AC network is described by the impedance $Z \in \mathbb{C}$. The real part $R \in \mathbb{R}$ of the impedance refers to the resistance and the imaginary part $X \in \mathbb{R}$ is called reactance. This allows us to describe the relation between voltage and current in an AC network similar to Ohm's law

$$V = ZI. \quad (3.13)$$

It will become handy to use the admittance $Y = \frac{1}{Z} \in \mathbb{C}$ instead of the impedance in Ohm's law

$$I = YV. \quad (3.14)$$

The real part of the admittance $G = \Re(Y)$ is called conductance and the imaginary part $B = \Im(Y)$ susceptance.

3.1.3 Complex Power

Momentary power $p(t)$ consumed or produced in an electrical component is given by

$$p(t) = v(t)i(t). \quad (3.15)$$

In DC networks, the power is constant and is computed by a multiplication of the peak values. However in AC networks, current and voltage are following a sinusoidal curve and a multiplication of the peak values would lead to wrong results. The momentary power in time-domain is given by

$$\begin{aligned} p(t) &= v(t)i(t) = V_{\text{pk}} \cos(\phi_v + t\omega) I_{\text{pk}} \cos(\phi_I + t\omega) \\ &= \frac{1}{2} V_{\text{pk}} I_{\text{pk}} (\cos(\phi_V - \phi_I) + \cos(2t\omega + \phi_V + \phi_I)). \end{aligned} \quad (3.16)$$

The power $p(t)$ has a constant part, which is not time depend and a part, which is oscillating at double angular velocity. The constant part

$$P = \frac{1}{2} V_{\text{pk}} I_{\text{pk}} \cos(\phi_V - \phi_I) \quad (3.17)$$

is called active power and the term $\cos(\phi_V - \phi_I)$ is referred to as the power factor. In the frequency-domain, we define the apparent power by

$$S = VI^* \quad (3.18)$$

where I^* refers to the complex conjugate of I . The active power is the real part of the apparent power

$$\begin{aligned} P &= \Re(S) = \Re(VI^*) = \Re\left(\tilde{V}e^{j\phi_V} \tilde{I}e^{-j\phi_I}\right) = \frac{1}{2} V_{\text{pk}} I_{\text{pk}} \Re\left(e^{j(\phi_V - \phi_I)}\right) \\ &= \frac{1}{2} V_{\text{pk}} I_{\text{pk}} \cos(\phi_V - \phi_I). \end{aligned} \quad (3.19)$$

The imaginary part of the apparent power is called reactive power and is usually denoted by Q

$$S = P + jQ. \quad (3.20)$$

The reactive power measures the amount of power which is circulating constantly due to the angle displacement. In AC networks, reactive power is produced and consumed by nodes and transmission lines like the active power.

In summary, the phasor representation allows us to identify AC quantities as constant complex numbers. The effect of electrical components and the power can be calculated by a set of rules similar to DC quantities. In a quasi-static system operation, we can perform most calculations without taking the sinusoidal behavior directly in consideration.

3.1.4 Three phase power

In electrical networks usually three-phase AC is used. Every transmission line consists of three individual cables and in every cable the signal is shifted. The transmission of electrical energy in three-phase AC originates from its generation in rotating electrical machines and it has several practical advantages. The individual phases are denoted by a, b, c .

Assumption 3.1 (Balanced 3-phase Operation)

Throughout this chapter, we assume a *balanced* operation of the system. This term refers to two properties:

- All signals are three-phase sinusoidal signals, with a phase shift of $\pm\frac{2}{3}\pi$ and equal magnitude. The sum of the three phases is zero.
- The network is balanced. Every cable of a three-phase connection has the same physical properties and all components load the network equally on each phase.

In balanced quasi-static operation, the voltage (and respectively the current) on the three lines are given by

$$\begin{aligned} v_a(t) &= V_{\text{pk}} \cos(\omega t + \phi_v) \\ v_b(t) &= V_{\text{pk}} \cos\left(\omega t + \phi_v - \frac{2}{3}\pi\right) \\ v_c(t) &= V_{\text{pk}} \cos\left(\omega t + \phi_v + \frac{2}{3}\pi\right). \end{aligned} \quad (3.21)$$

where V_{pk} denotes the voltage magnitude and ϕ_v the offset. In the following, we will refer to three-phase signals by the subscript *abc*, e.g. the three-phase voltage in the time-domain is denoted by

$$v_{abc}(t) = \begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} \in \mathbb{R}^3. \quad (3.22)$$

The momentary power of a balanced three-phase component is the sum of the individual phases. The trigonometric identity $\cos(x)\cos(y) = \frac{1}{2}(\cos(x-y) - \cos(x+y))$ leads to

$$\begin{aligned} p(t) &= p_a(t) + p_b(t) + p_c(t) = v_a(t)i_a(t) + v_b(t)i_b(t) + v_c(t)i_c(t) \\ &= \frac{1}{2} V_{\text{pk}} I_{\text{pk}} \left(3 \cos(\phi_v - \phi_i) + \right. \\ &\quad \left. \left(\cos(2\theta + \phi_v + \phi_i) + \cos(2\theta + \phi_v + \phi_i - \frac{2\pi}{3}) + \cos(2\theta + \phi_v + \phi_i + \frac{2\pi}{3}) \right) \right) \\ &= \frac{3}{2} V_{\text{pk}} I_{\text{pk}} \cos(\phi_v - \phi_i). \end{aligned} \quad (3.23)$$

This shows that in quasi-static operation the momentary power is constant. For many applications this is a desired property and one of the main advantages of three-phase power.

In three-phase networks, it is common to use the line-to-line voltage instead of the line-to-ground voltage. The line-to-line voltage between phase a and b is given by

$$\begin{aligned} v_{ab}(t) &= v_a(t) - v_b(t) = V_{\text{pk}} \cos(\omega t + \phi_v) - V_{\text{pk}} \cos(\omega t + \phi_v - \frac{2}{3}\pi) \\ &= -2V_{\text{pk}} \sin(\omega t + \phi_v) \sin(-\frac{1}{3}\pi) \\ &= \sqrt{3}V_{\text{pk}} \sin(\omega t + \phi_v) \end{aligned} \quad (3.24)$$

and therefore the peak is given by $\sqrt{3}V_{\text{pk}}$. The phasor representation of voltage V_{abc} and current I_{abc} stays valid for three-phase networks and Ohm's law can be applied. However, since the voltage magnitude refers to the line-to-line voltage, the apparent power is computed by

$$S = \sqrt{3}V_{abc}I_{abc} \quad (3.25)$$

in order to stay consistent with equation (3.23).

3.1.5 Per-Unit System

In power system analysis it is common, not to use the International System of Units (SI) directly but instead to express the electrical quantities as a ratio of a base value. This system is called per-unit system. For a quantity x in SI units, the corresponding per-unit quantity is given by

$$x_{\text{pu}} = \frac{x}{x_{\text{base}}} \quad (3.26)$$

where x_{base} is the base value. Since the per-unit value is a factor of the base value, it has no SI unit. The base values for voltage and power are usually the nominal operating points for the electrical network. Base values for complex quantities are always real. Therefore, the per-unit value has always the same angle as the SI value. The other base values are deduced in order to maintain the electrical laws. For example, it is desirable that the apparent power is computed in the per-unit system similar to the SI system for single-phase systems

$$S_{\text{pu}} = I_{\text{pu}} V_{\text{pu}}. \quad (3.27)$$

Therefore, the current base value needs to be chosen as

$$I_{\text{base}} = \frac{S_{\text{base}}}{V_{\text{base}}} \quad (3.28)$$

in order to maintain the definition of apparent power.

Using a per-unit system has several practical advantages. Independently of the size of the grid, the quantities are in the same range. This makes it easier to detect errors and improves the numerical stability. For example, the net frequency and the nominal voltage are usually one in the per-unit system. Especially, the per-unit system makes the distinction between the single and three-phase system obsolete. With a suitable definition of the base values, all computation rules can be applied to both systems in the per-unit system. In the following, we refer exclusively to quantities in the per-unit system and all statements are valid for single and three-phase systems. A comprehensive description of the per-unit system can be found in [40].

3.1.6 Electrical Networks

In the previous sections, we investigated the relation of voltage between two terminals of an electrical component and the current flowing through it. However, in an electrical network the individual buses are interconnected and in order to find a valid point of operation their mutual interactions need to be considered. In this section, we define an electrical network formally.

An electrical network is given as a set of buses $\mathcal{N} = \{1, \dots, n\}$ with $n \in \mathbb{N}$ and a set of branches $\mathcal{Y} \subseteq \mathcal{N} \times \mathcal{N}$. The physical properties of a branch $(i, k) \in \mathcal{Y}$ is described by the admittance Y_{ik} . For branches $(i, k) \notin \mathcal{Y}$, we define $Y_{ik} = 0$. Every bus $i \in \mathcal{N}$ is associated with a voltage $V_i = \tilde{V}_i e^{j\phi_i} \in \mathbb{C}$, which is measured with respect to a reference point, typically the ground. The voltage between two buses of the network $i, k \in \mathcal{N}$ is denoted by V_{ik} . The current flowing through a branch $(i, k) \in \mathcal{Y}$ is given by I_{ik} .

3.1.7 Kirchhoff's Law

The physical behavior of an electrical network is described by Kirchhoff's Laws. In order to get consistent results, it is important to take the direction of voltage and current in consideration. We denote the source bus by the first index and the sink bus by the second. Kirchhoff's Voltage Law (KVL) states that the sum of voltages around a closed loop circle is zero. For every ordered set of indices $i_1, \dots, i_m \in \mathcal{N}$ it

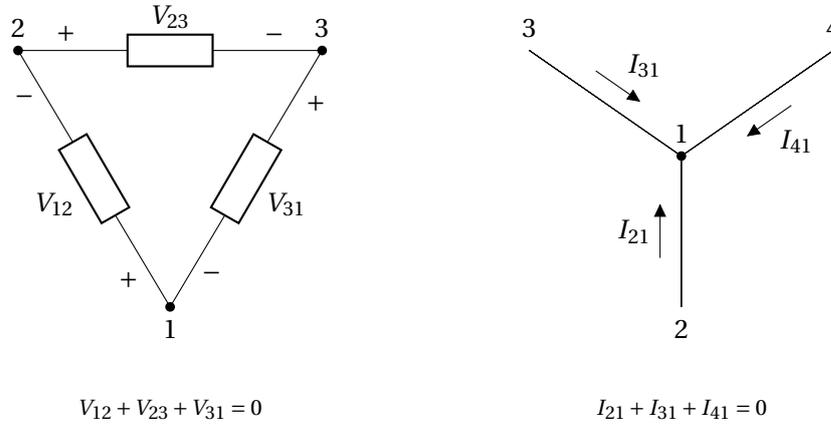


Figure 3.2: Example of KVL and KCL.

holds

$$\sum_{k=1}^{m-1} V_{i_k, i_{k+1}} + V_{i_m, i_1} = 0. \quad (3.29)$$

Using the ground, we can use KVL to compute the voltage between two arbitrary points of the electrical network

$$V_{ik} = V_k - V_i. \quad (3.30)$$

Kirchhoff's Current Law (KCL) states that all currents at a specific bus sum up to zero. For every $i \in \mathcal{N}$ it holds

$$\sum_{k \in \mathcal{N}, (i,k) \in \mathcal{Y}} I_{ik} = 0. \quad (3.31)$$

Both of Kirchhoff's Laws must be satisfied in order to get a physical valid power flow of the electrical network. Figure 3.2 visualizes KVL and KCL.

3.1.8 Power Flow Equations

In order to find valid operation points of electrical networks, we introduce the non-linear AC power flow equations in this section. Since we are mainly interested in the power balances, we look at the active power and reactive power injected or consumed at the buses. Loads act as power consumer whereas generators and other

power sources can be interpreted as positive power injections. At every node there can be multiple power sources and loads connected with different dynamics, but in this section we are only considering net energy exchange at the buses.

In this section $I \in \mathbb{C}^n$ refers to the nodal current injections from outside the network. The vector $V \in \mathbb{C}^n$ collects the nodal voltages with respect to the reference point. The net apparent power injected or consumed at the nodes is given by $S = P + jQ \in \mathbb{C}^n$, with $P, Q \in \mathbb{R}^n$. For every node $i \in \mathcal{N}$, we can apply KCL and KVL to get

$$\begin{aligned} I_i &= - \sum_{k \in \mathcal{N}, (i,k) \in \mathcal{Y}} I_{ki} = - \sum_{k=1, k \neq i}^n Y_{ki} V_{ki} = - \sum_{k=1, k \neq i}^n Y_{ki} (V_i - V_k) \\ &= - \sum_{k=1, k \neq i}^n Y_{ki} V_i + \sum_{k=1, k \neq i}^n Y_{ki} V_k. \end{aligned} \quad (3.32)$$

To simplify this expression, we define the admittance matrix $Y \in \mathbb{C}^{n \times n}$ by

$$(Y)_{ik} = \begin{cases} -\sum_{k=1, k \neq i} Y_{ki} & \text{for } i = k, \\ Y_{ki} & \text{for } i \neq k. \end{cases} \quad (3.33)$$

The real part $G = \Re(Y) \in \mathbb{R}^{n \times n}$ of the admittance matrix is called conductance matrix and the imaginary part $B = \Im(Y) \in \mathbb{R}^{n \times n}$ susceptance matrix. This allows us to write equation (3.32) in the compact form

$$I = YV. \quad (3.34)$$

This expression gives the relation between the nodal voltages and the external current injections. We can use this to eliminate the currents from the power balances. The external apparent power injections at the nodes are given by

$$S = V \cdot I^* = V \cdot (YV)^* \quad (3.35)$$

where \cdot denotes the elementwise multiplication. Row i of this equation is expanded to

$$P_i + jQ_i = S_i = V_i \sum_{k=1}^n Y_{ik}^* V_k^* \quad (3.36)$$

$$= \tilde{V}_i \sum_{k=1}^n \tilde{V}_k e^{j(\phi_i - \phi_k)} (G_{ik} - jB_{ik}) \quad (3.37)$$

$$= \tilde{V}_i \sum_{k=1}^n \tilde{V}_k (\cos(\phi_i - \phi_k) + j \sin(\phi_i - \phi_k)) (G_{ik} - jB_{ik}). \quad (3.38)$$

We can get a closed expressions for the active and reactive power injections by splitting this equation in the real and imaginary part

$$P_i = \tilde{V}_i \underbrace{\sum_{k=1}^N \tilde{V}_k (G_{ik} \cos(\phi_i - \phi_k) + B_{ik} \sin(\phi_i - \phi_k))}_{=:P_i(V,\phi)} \quad (3.39a)$$

$$Q_i = \tilde{V}_i \underbrace{\sum_{k=1}^N \tilde{V}_k (G_{ik} \sin(\phi_i - \phi_k) - B_{ik} \cos(\phi_i - \phi_k))}_{=:Q_i(V,\phi)}. \quad (3.39b)$$

These equations are referred to as nonlinear AC Power Flow (PF) equations. They give a complete algebraic relation of the external power injections, the voltage magnitudes and phase angles. Every steady-state operating point of an electrical network needs to satisfy these equations. The PF equations are the basis for several classical tasks in power engineering and there are a lot of extensions [33].

3.1.9 Conventional Power Flow

The conventional PF task seeks a deterministic solution to PF equations (3.39). It is a feasibility problem for a given set of fixed variables. In this model, every bus i is associated with the four variables \tilde{V}_i, ϕ_i, P_i and Q_i . To ensure that the equations are uniquely solvable, two of the four variables need to be fixed. All nodes are classified into one of the following categories:

Slack Bus There is always exactly one slack bus in every network. It serves as the reference point for the voltage angle, i.e. $\phi = 0$. In addition the voltage is fixed to the nominal level, in the per unit system typically $V = 1 p.u.$. Since the phase angle and the voltage is fixed, the active and reactive power is free. In fact it is the only bus with free active power and is responsible to ensure the power balance in the network.

Load Bus At a load bus P and Q are fixed whereas voltage V and phase angle ϕ are free.

Voltage-Controlled Bus At a voltage-controlled bus P and V are fixed, whereas the reactive power Q and phase angle ϕ are free.

Given this categorization, it is guaranteed that the PF equations have a physical solution. There are additional mathematical solutions to this equations, but they do not

represent physical values. For example, negative voltage magnitudes may appear. A detailed description of the conventional PF task with the mathematical background and solution methods can be found in [36].

3.1.10 Optimal Power Flow

In this section we introduce the classical Optimal Power Flow (OPF) formulation. It uses nonlinear power flow equation to find an optimal generator dispatch with respect to a given objective function. We will refer to this standard formulation in the later section and expand it to our needs. A detailed introduction to OPF from mathematical point of view can be found in [36].

In the previous section we only looked at the net active and reactive power balance on every bus. For OPF we distinguish between the injected power input S^g and the passive load demand S^d . The injected power sums power generated by classical generators or storage systems. The injected power is considered as controllable, but comes with costs. The cost is given by a (typically quadratic) cost function $C(P^g)$. The load includes not only the classical power consumer but also the energy input of Renewable Energy Resources (RES). Input from for example a Photovoltaic (PV)-System appears as a negative load. The demand emerge as fixed parameters in the optimization problem. In total the net power balance at node $i \in \mathcal{N}$ is given by

$$\begin{aligned} P_i &= P_i^d - P_i^g, \\ Q_i &= Q_i^d - Q_i^g. \end{aligned}$$

The load parameters P^d and Q^d are fixed parameters whereas P^g and Q^g are controllable. Similar to the traditional PF problem, there is a slack bus (w.l.o.g. $i = 1$), where the phase angle is fixed at $\phi_1 = 0$ to avoid ambiguity. On the other nodes, voltage and phase angle are free. The complete optimal power flow problem is defined by

$$\begin{aligned} \min_{P^g, Q^g, V, \phi} \quad & C(P^g) \\ \text{s.t.} \quad & P^d - P^g = P(V, \phi), \quad Q^d - Q^g = Q(V, \phi), \\ & \underline{P}^g \leq P^g \leq \overline{P}^g, \quad \underline{Q}^g \leq Q^g \leq \overline{Q}^g, \\ & \underline{V} \leq V \leq \overline{V}, \quad \underline{\phi} \leq \phi \leq \overline{\phi}, \end{aligned}$$

where the operational bounds on the variables are denoted by over- and underbars. Variables without subscript indices refer to vectors over the complete set of buses. The OPF problem formulation is the basis for a lot of more specialized tasks in power analysis. For example multi-stage OPF allows to represent a time dependent behavior of an electrical network [32], distributed OPF allows to solve the optimization on multiple machines with limited shared information, and stochastic OPF allows to consider uncertainties in the parameters [33]. In order to efficiently solve OPF problems with integer decision variables, e.g. for switch events of generators, the PF equations can be convexified with a quadratic convex relaxation [21].

3.2 Dynamic Modeling

The phasor representation of voltage and current introduced in the previous sections is a quasi-static model. Most of the electrical dynamics act on a fast time scale and are stabilized after a short transient phase. Therefore, it is reasonable to neglect these dynamics for long-term planing. However, we are also interested in the fast transient behavior of the electrical network. In this section, we introduce models for the components as well as the network that consider transient dynamics.

Since we do not assume a steady-state operation, we have to deal with dynamically changing phase angles. We will investigate the system with respect to a reference frequency f respectively a reference angular velocity $\omega = 2\pi f$. The reference frequency is usually the nominal frequency of the network. The European utility grid uses a nominal frequency of $f = 50$ Hz and since microgrids are often coupled with the utility grid, this frequency will also be used in our models. The reference angle is denoted by $\theta(t) = \omega t$. Instead of using the phase angle of an AC signal directly, we will indicate it by the offset $\phi(t)$ to the reference angle $\theta(t)$. For notional simplicity, we will omit the time dependence on the angles if they are not needed. This section follows the lines of [67].

For dynamical modeling, we will consider balanced three-phase networks. In balanced operation, the voltages (and respectively the current) on the three lines are given by

$$\begin{aligned} v_a(t) &= V_{pk}(t) \cos(\theta + \phi_v(t)) \\ v_b(t) &= V_{pk}(t) \cos\left(\theta + \phi_v(t) - \frac{2}{3}\pi\right) \\ v_c(t) &= V_{pk}(t) \cos\left(\theta + \phi_v(t) + \frac{2}{3}\pi\right). \end{aligned} \tag{3.40}$$

where $\phi_v(t)$ is the offset to the reference angle and $V_{pk}(t)$ denotes the voltage magnitude. Usually, the phases follow a sinusoidal curve and the magnitude $V_{pk}(t)$ as well as the phase offset $\phi(t)$ are changing slowly compared to the nominal frequency. We collect the three phases in a vector $v_{abc}(t) \in \mathbb{R}^3$.

3.2.1 DQ0-transformation

The phasor representation is a convenient technique to describe a quasi static operation point in an electrical AC network. However, this formulation is not capable to describe transients, because the definition of active and reactive power presumes a quasi static operation. In this section, we introduce the $dq0$ -transformation as a technique to represent 3-phase AC signals. It allows to fully describe the transient behavior of an AC network and to compute the power balances in steady state in a convenient way. Therefore, the $dq0$ -transformation can be interpreted as an extension of the phasor representation. The fundamental idea of the $dq0$ -transformation is to describe the electrical quantities in relation to a reference frame, which is rotating with the nominal angular velocity ω . The transformation is dependent on the current reference angle θ and is given by the matrix

$$T_\theta = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2}{3}\pi) & \cos(\theta - \frac{4}{3}\pi) \\ -\sin(\theta) & -\sin(\theta - \frac{2}{3}\pi) & -\sin(\theta - \frac{2}{3}\pi) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \quad (3.41)$$

The inverse matrix is

$$T_\theta^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos(\theta - \frac{2}{3}\pi) & -\sin(\theta - \frac{2}{3}\pi) & 1 \\ \cos(\theta - \frac{4}{3}\pi) & -\sin(\theta - \frac{2}{3}\pi) & 1 \end{bmatrix}. \quad (3.42)$$

Since the reference angle θ is changing over time, the time derivative of the transformation is not zero. The $dq0$ -transformation is scaled in order to preserve the regular power balance equations. This transformation is deduced in two separate steps. First, the 3-phase signals are mapped into an orthogonal coordinate system by the abc transformation and afterwards a rotation matrix with the reference angle is applied. A detailed derivation can be found in [87].

A fundamental property of the $dq0$ -transformation is that it maps balanced quasi-static AC signals to constants. The $dq0$ -transformed 3-phase voltage signal (3.40) is

given by

$$\begin{aligned}
 v_{dq0} &= T_\theta v_{abc} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2}{3}\pi) & \cos(\theta - \frac{4}{3}\pi) \\ -\sin(\theta) & -\sin(\theta - \frac{2}{3}\pi) & -\sin(\theta - \frac{4}{3}\pi) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} v_{abc} \\
 &= \frac{2}{3} V_{pk}(t) \begin{bmatrix} \frac{3}{2} \cos(\phi) + \frac{1}{2} (\cos(2\theta + \phi) + \cos(2\theta + \phi - \frac{4}{3}\pi) + \cos(2\theta + \phi + \frac{4}{3}\pi)) \\ \frac{3}{2} \sin(\phi) + \frac{1}{2} (\sin(2\theta + \phi) + \sin(2\theta + \phi - \frac{4}{3}\pi) + \sin(2\theta + \phi + \frac{4}{3}\pi)) \\ 0 \end{bmatrix} \\
 &= \begin{bmatrix} V_{pk}(t) \cos \phi \\ V_{pk}(t) \sin \phi \\ 0 \end{bmatrix},
 \end{aligned} \tag{3.43}$$

which is constant for a constant magnitude V_{pk} and a constant angle ϕ . The individual entries are referred to as the *direct*, *quadrature* and *zero* component. Since we assume a balanced operation of the network, the zero component is always zero.

The electrical power in $dq0$ -quantities is given by

$$\begin{aligned}
 P &= i_a v_a + i_b v_b + i_c v_c = i_{abc}^\top v_{abc} \\
 &= i_{dq0}^\top T_\theta^{-\top} T_\theta^{-1} v_{dq0} \\
 &= \frac{3}{2} i_{dq0}^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} v_{dq0} \\
 &= \frac{3}{2} (i_d v_d + i_d v_d + 2i_0 v_0) \\
 &= \frac{3}{2} (i_d v_d + i_d v_d).
 \end{aligned} \tag{3.44}$$

3.2.2 Resistors, Inductors and Capacitors in $dq0$ -Representation

In order to represent a complete network in $dq0$ -quantities, we transform the models of the electrical components. A resistor in a balanced three-phase network is modeled by

$$v_{abc} = R i_{abc}. \tag{3.45}$$

Since this is a linear equation, it is directly transferable to $dq0$ -signals

$$v_{dq0} = R i_{dq0}. \quad (3.46)$$

The model for an inductor is

$$v_{abc} = L \frac{d}{dt} i_{abc} \quad (3.47)$$

with the inductance $L \in \mathbb{R}$. We use $i_{abc} = T_\theta^{-1} i_{dq0}$ to get

$$v_{abc} = L \frac{d}{dt} (T_\theta^{-1} i_{dq0}). \quad (3.48)$$

The matrix T_θ^{-1} depends on the time and therefore the product rule applies

$$v_{abc} = L \left(\frac{d}{dt} T_\theta^{-1} \right) i_{dq0} + L T_\theta^{-1} \frac{d}{dt} i_{dq0}. \quad (3.49)$$

The direct computation of the derivative leads to

$$\frac{d}{dt} T_\theta^{-1} = -T_\theta^{-1} \mathcal{W} \quad (3.50)$$

with

$$\mathcal{W} = \begin{bmatrix} 0 & \frac{d}{dt} \theta & 0 \\ -\frac{d}{dt} \theta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.51)$$

By substitution, we obtain

$$v_{abc} = -L T_\theta^{-1} \mathcal{W} i_{dq0} + L T_\theta^{-1} \frac{d}{dt} i_{dq0}. \quad (3.52)$$

A multiplication by T_θ from the left gives the model for an inductor in $dq0$ signals

$$\frac{d}{dt} i_{dq0} = \mathcal{W} i_{dq0} + \frac{1}{L} v_{dq0}. \quad (3.53)$$

Since we assume a balanced operation, this model can be reduced further to

$$\begin{aligned}\frac{d}{dt}i_d &= \omega i_q + \frac{1}{L}v_d, \\ \frac{d}{dt}i_q &= -\omega i_d + \frac{1}{L}v_q.\end{aligned}\tag{3.54}$$

In a similar way, the model for a capacitor can be deduced

$$\frac{d}{dt}v_{dq0} = \mathcal{W}v_{dq0} + \frac{1}{C}i_{dq0}\tag{3.55}$$

with the explicit expression for balanced networks

$$\begin{aligned}\frac{d}{dt}v_d &= \omega v_q + \frac{1}{C}i_d, \\ \frac{d}{dt}v_q &= -\omega v_d + \frac{1}{C}i_q.\end{aligned}\tag{3.56}$$

3.2.3 Comparison of Phasor and $dq0$ -representation

In this section we investigate the relation between the the phasor and the $dq0$ -representation. The active and reactive power can be identified with direct and quadrature component in quasi-static operation. For a constant phase angle and magnitude, the $dq0$ -signal is constant

$$v_{dq0} = \begin{bmatrix} V_{pk} \cos \phi \\ V_{pk} \sin \phi \\ 0 \end{bmatrix}.\tag{3.57}$$

This allows to represent the phasor by the direct and quadrature component of the $dq0$ -signal

$$V = \frac{V_{pk}}{\sqrt{2}}e^{j\phi(t)} = \frac{V_{pk}}{\sqrt{2}}(\cos(\phi) + j \sin(\phi)) = \frac{1}{\sqrt{2}}(v_d + jv_q).\tag{3.58}$$

In steady-state the direct and quadrature component refers to real and imaginary part of the phasor

$$\begin{aligned}v_d &= \sqrt{2}\Re(V), \\ v_q &= \sqrt{2}\Im(V).\end{aligned}\tag{3.59}$$

Hence, we can use the $dq0$ -quantities to express the active and reactive power of phasors by

$$\begin{aligned} P &= \Re(VI^*) = \frac{1}{2}(v_d i_d + v_q i_q), \\ Q &= \Im(VI^*) = \frac{1}{2}(v_q i_d - v_d i_q). \end{aligned} \quad (3.60)$$

The absolute value and the angle can be expressed by

$$|V|^2 = \Re(V)^2 + \Im(V)^2 = \frac{1}{2}(v_d^2 + v_q^2), \quad (3.61)$$

$$\phi = \text{atan2}(v_q, v_d). \quad (3.62)$$

This shows that we can maintain the definition of active and reactive power and they match the active and reactive power of the phasor in steady state. This can be seen as an extension of the phasor to non steady state signals. In a similar way, the models for electrical components coincide in steady state operation. The phasor model of an inductor is

$$V = \omega L j I.$$

Since we can identify the real and imaginary part of the phasors with the direct and quadrature components of the $dq0$ -signals, this leads to

$$v_d + j v_q = \omega L j (i_d + j i_q). \quad (3.63)$$

The real and imaginary parts of this equation are

$$\begin{aligned} v_d &= -\omega L i_q, \\ v_q &= \omega L i_d. \end{aligned} \quad (3.64)$$

In steady-state operation, i.e. $\frac{di_d}{dt} = \frac{di_q}{dt} = 0$, this expression matches the $dq0$ -model of an inductor

$$\begin{aligned} \frac{d}{dt} i_d &= \omega i_q + \frac{1}{L} v_d, \\ \frac{d}{dt} i_q &= -\omega i_d + \frac{1}{L} v_q. \end{aligned} \quad (3.65)$$

In summary, the $dq0$ -representation is capable to fully describe the transient behavior of an electrical network and in quasi-static operation, all phasors can be read out directly. In addition, it maps 3-phase sinusoidal signals to constants at the nominal frequency, which allows to compute the power balances conveniently.

3.3 Microgrid Components

Microgrids (MGs) are characterized by a wide variety of connected components. However, we will focus on microgrids with included RES and restrict ourselves to their most fundamental components. We consider MGs with the following components:

Generators are part of most MGs, because they are able to provide electrical energy reliable. They are independent of the current weather conditions and are able to react to load changes very fast. Because of their controllability, they can be used to stabilize the grid after disturbances. However, they are powered by fuel, which creates costs and greenhouse gases. Therefore, it is desirable to use the generators as little as possible. Here we focus on synchronous generators governed by diesel engines.

Storage Systems are used to store surplus energy generated for example by RES. In MGs the energy is typically stored in batteries, because they allow a flexible operation. However, there are many storage technologies available [109].

Loads are passive network components and represent power consumers. They are not controllable and affected by sudden changes. PV system and other RES are usually represented as negative loads.

3.3.1 Diesel Generators

Diesel generators are composed of a diesel engine and a SG connected by a shaft. The diesel engine generates mechanical power by a torque which drives the shaft. The purpose of the SG is to convert the mechanical power to electrical power. In this section, we focus on the modeling of the SG. The essentials of modeling of electrical machines can be found in [60] in great detail. However, we present only the basic ideas and follow [67].

The general structure of a SG is depicted in Figure 3.3. The main mechanical components are the stator and the rotor. The rotor is rotating inside the stator, driven by the shaft, and is equipped with the so-called excitation winding. The excitation winding

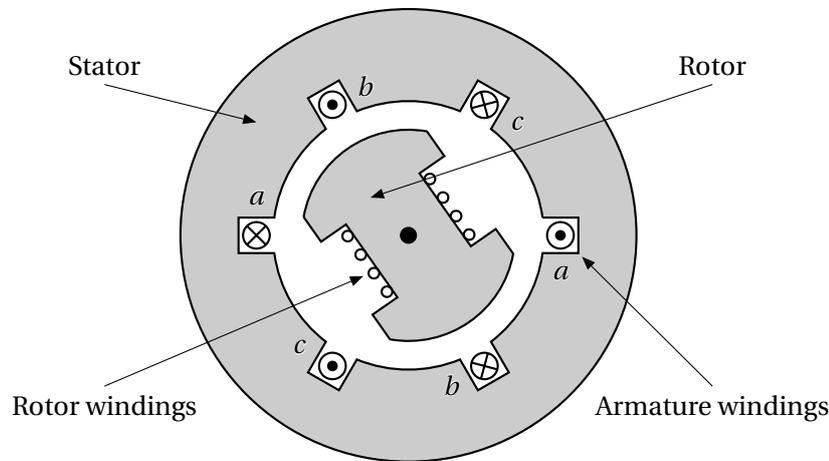


Figure 3.3: Sketch of synchronous generator

is usually connected to a DC power source and currents in the winding create a magnetic field which is rotating with the rotor. Depending on the size of the generator, the magnetic field may have multiple pole pairs. We focus on generators with just a single pole pair, but the extension to multiple pole pairs is straightforward.

The stator holds three windings, the armature windings, which are connected to the terminals of the generator. The rotating magnetic field induces currents in the stator windings. This type of generator is called synchronous because the currents at the terminals of the generator are rotating always with the same frequency as the rotor.

The SG can be controlled by the excitation current i_f and the mechanical torque T_m . The resulting frequency ω and voltage magnitude V depend nonlinearly on these parameters. However, in normal operation, the voltage magnitude is mainly related to the excitation current and the electrical energy to the mechanical torque. Therefore, they are often controlled in two separate control loops. The Automatic Voltage Regulator (AVR) controls excitation current and uses the terminal voltage magnitude as feedback signal. The mechanical torque is controlled by the governor for the diesel engine and uses the frequency as feedback. Both controllers aim at tracking reference values for frequency ω^{ref} and voltage V^{ref} . The signal flow of the SG and its controllers is depicted in Figure 3.3.

We will first describe the mechanical and afterwards the electrical equations of the generator. Since the electromagnetic field is always synchronous with the rotor, we

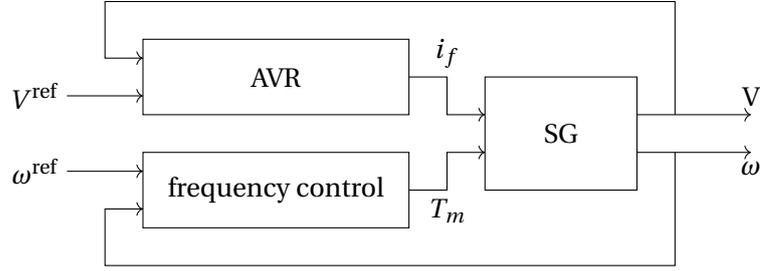


Figure 3.4: Signal flow of SG model.

will transform the electrical equations in rotating $dq0$ -reference frame, which is rotating with the angular velocity of the rotor ω .

Mechanical Equations

The fundamental mechanical dynamics of a SG is described by the swing equation

$$\frac{d}{dt}\omega = \frac{1}{J}(T_m - T_e) \quad (3.66)$$

with the mechanical torque T_m , electrical torque T_e and the angular velocity ω . The physical mass defines the total moment of inertia J of the generator. According to the swing equation (3.66), a difference between the mechanical and electrical torque leads to an acceleration or deceleration of the rotor. A high moment of inertia lowers the effect of a given mismatch between mechanical and electrical torque. Therefore bigger generators with a higher physical mass are more resilient against disturbances in the network and provide a more stable frequency.

The relations between torque and power are given by $P_m = \omega T_m$ and $P_e = \omega T_e$. With $K = \frac{1}{J\omega_s}$, this leads to

$$\frac{d}{dt}\omega = K \frac{\omega_s}{\omega} (P_m - P_e). \quad (3.67)$$

where ω_s is the nominal grid frequency.

The mechanical torque T_m is created by the diesel engine. Usually, it is equipped with a primary control mechanism in order to stabilize the generator frequency at the network frequency. In numerical experiments in Section 6, the IEEE DEGOV1 [83] is used to represent the diesel engine as well as the frequency governor. However, in many situations it is sufficient, to represent the engine by a droop control

law

$$T_m = \frac{1}{\omega_s} \left(3P^{\text{ref}} - \frac{1}{D} (\omega - \omega_s) \right) \quad (3.68)$$

in order to steer the frequency to the nominal operating point ω_s . The nominal load of the generator is given by P^{ref} and D is a damping parameter. This leads to the simplified equation for the frequency

$$\frac{d}{dt} \omega = K \left(3P^{\text{ref}} - \omega_s T_e - \frac{1}{D} (\omega - \omega_s) \right). \quad (3.69)$$

Here P^{ref} and ω_s are external parameters, while ω and T_e are dynamic states of the system.

Electrical Equations

The main mechanism to transform mechanical to electrical energy is the inductive coupling between the rotor and stator windings. A DC power source is connected to the excitation winding and creates a magnetic field, which is carried by the rotor. The magnetic field is flowing through the iron parts of the generator and creates a flux linkage between the stator and the rotor windings. Due to the rotational movement of the rotor, the flux linkage changes periodically and a voltage at the terminals of stator windings is induced. The windings in the stator are designed such that the resulting voltages approximate a sine curve with a period proportional to the rotational speed. The electrical energy provided by the generator causes an electrical torque T_e , which decelerates the rotor.

AC in the armature windings is also causing constantly changing electromagnetic fields and currents are induced mutually. The electromagnetic flux between the stator and rotor windings depends on the physical properties of the stator and rotor. Usually, the mechanical parts are composed of iron layers, in order to carry the electromagnetic field efficiently and to prevent eddy currents. Mutual inductances between the rotor and stator windings are nonlinear dependent on the current rotor angle θ , because the stator is not symmetrical and often salient rotors are used. To consider the geometry of the SG and the electromagnetic dynamics exactly, comprehensive simulations based on partial differential equations are required. In order to get simplified model equations, we make the following assumptions.

Assumption 3.2

1. The SG is a magneto-quasi-static device and the temporal dynamics of the magnetic field are neglected. This assumption is justified since the rotational speed is significantly slower than the dynamics of the magnetic field.
2. We do not consider magnetic saturation, because in normal operating conditions the material is not saturated and the permeability is almost constant.
3. We assume that the inductances are composed of a constant term and cosine term varying with double angular velocity, i.e. they depend on 2θ . This assumption is motivated by the symmetrical form of salient rotors in many electrical machines.

The three stator windings are symmetrically shifted by $\frac{2}{3}\pi$ in order to create a balanced terminal voltage and this carries over to the inductances. We denote the armature windings by a, b, c and the excitation winding by f . This allows us to represent the mutual inductances of the SG in the form

$$l = \begin{bmatrix} l_{aa} & l_{ab} & l_{ac} & l_{af} \\ l_{ab} & l_{bb} & l_{bc} & l_{bf} \\ l_{ac} & l_{bc} & l_{cc} & l_{cf} \\ l_{af} & l_{bf} & l_{cf} & l_{ff} \end{bmatrix} \quad (3.70)$$

where the inductances between the stator winding are defined by

$$\begin{aligned} l_{aa} &= L_{aa} + L_{g2} \cos(2\theta), & l_{ab} &= L_{ab} + L_{g2} \cos\left(2\theta - \frac{3}{2}\pi\right), \\ l_{bb} &= L_{aa} + L_{g2} \cos\left(2\theta + \frac{3}{2}\pi\right), & l_{bc} &= L_{ab} + L_{g2} \cos(2\theta), \\ l_{cc} &= L_{aa} + L_{g2} \cos\left(2\theta - \frac{3}{2}\pi\right), & l_{ac} &= L_{ab} + L_{g2} \cos\left(2\theta + \frac{3}{2}\pi\right) \end{aligned} \quad (3.71)$$

and the mutual inductances between the rotor and stator by

$$\begin{aligned} l_{af} &= L_{af} \cos(2\theta), & l_{bf} &= L_{af} \cos\left(2\theta - \frac{3}{2}\pi\right), \\ l_{cf} &= L_{af} \cos\left(2\theta + \frac{3}{2}\pi\right), & l_{ff} &= L_{ff}. \end{aligned} \quad (3.72)$$

The parameters $L_{aa}, L_{ab}, L_{g2}, L_{af}, L_{ff}$ are constant and defined by the material and geometry of the SG.

Using these inductances, the relationship between the flux linkages of the windings λ_{abc}, λ_f and the currents i_{abc}, i_f is given by

$$\begin{bmatrix} \lambda_{abc} \\ \lambda_f \end{bmatrix} = l \begin{bmatrix} -i_{abc} \\ i_f \end{bmatrix}. \quad (3.73)$$

The negative stator currents are following the convention, to define currents flowing out of the generator positive. Following Faraday's law of induction, the voltages are given by the derivative of the flux linkage. Considering the internal resistance of the armature windings R and excitation winding R_f , the terminal voltages are obtained by

$$\begin{aligned} v_a &= -Ri_a + \frac{d}{dt}\lambda_a, & v_b &= -Ri_b + \frac{d}{dt}\lambda_b, \\ v_c &= -Ri_c + \frac{d}{dt}\lambda_c, & v_f &= R_f i_f + \frac{d}{dt}\lambda_f. \end{aligned} \quad (3.74)$$

Since the inductances are defined in a static coordination system, they are varying in a sinusoidal fashion with the double angular velocity and they are dependent on 2θ . However, in a reference frame, which is rotating with the rotor, they are constant. Therefore, we transform equation (3.73) in $dq0$ coordinates

$$\begin{bmatrix} \lambda_{dq0} \\ \lambda_f \end{bmatrix} = \begin{bmatrix} T_\theta & \\ & 1 \end{bmatrix} l \begin{bmatrix} T_\theta^{-1} & \\ & 1 \end{bmatrix} \begin{bmatrix} -i_{dq0} \\ i_f \end{bmatrix}. \quad (3.75)$$

The transformed inductance matrix has the form

$$\begin{bmatrix} T_\theta & \\ & 1 \end{bmatrix} l \begin{bmatrix} T_\theta^{-1} & \\ & 1 \end{bmatrix} = \begin{bmatrix} L_d & & L_{af} \\ & L_q & \\ \frac{3}{2}L_{af} & & L_{ff} \end{bmatrix} \quad (3.76)$$

with constant $dq0$ inductances

$$\begin{aligned} L_d &= L_{aa} - L_{ab} + \frac{3}{2}L_{g2}, \\ L_q &= L_{aa} - L_{ab} - \frac{3}{2}L_{g2}, \\ L_0 &= L_{aa} + 2L_{ab}. \end{aligned} \quad (3.77)$$

In the $dq0$ reference frame, the flux depends linearly on the currents and is independent of the current rotor angle

$$\begin{aligned}\lambda_d &= -L_d i_d + L_{af} i_f, & \lambda_q &= -L_q i_q, \\ \lambda_0 &= -L_0 i_0, & \lambda_f &= -\frac{3}{2} L_{af} i_d + L_{ff} i_f.\end{aligned}\quad (3.78)$$

This allows us to transform equation (3.74) in $dq0$ quantities

$$\begin{aligned}v_d &= -R i_d + \frac{d}{dt} \lambda_d - \omega \lambda_q, & v_q &= -R i_q + \frac{d}{dt} \lambda_q + \omega \lambda_d, \\ v_0 &= -R i_0 + \frac{d}{dt} \lambda_0, & v_f &= R_f i_f + \frac{d}{dt} \lambda_f.\end{aligned}\quad (3.79)$$

The electrical torque is obtained by the formula for the electromagnetic force for the armature windings. Transformed to $dq0$ -quantities, this leads to

$$T_e = \frac{3}{2} (\lambda_d i_q - \lambda_q i_d). \quad (3.80)$$

The transferred electrical power can be derived by multiplying the torque with the angular velocity

$$P_e = \frac{3}{2} \omega (\lambda_d i_q - \lambda_q i_d). \quad (3.81)$$

Equations (3.69), (3.80) and (3.79) describe the complete dynamics of a SG as a DAE. The set of differential equations is given by

$$\begin{aligned}\frac{d}{dt} \theta &= \omega, \\ \frac{d}{dt} \omega &= K \left(3P^{\text{ref}} - \omega_s T_e - \frac{1}{D} (\omega - \omega_s) \right), \\ \frac{d}{dt} \lambda_d &= v_d + R i_d + \omega \lambda_q, \\ \frac{d}{dt} \lambda_q &= v_q + R i_q - \omega \lambda_d, \\ \frac{d}{dt} \lambda_0 &= v_0 + R i_0, \\ \frac{d}{dt} \lambda_f &= v_f - R_f i_f\end{aligned}\quad (3.82)$$

and the algebraic equations are

$$\begin{aligned}
 0 &= \frac{3}{2}(\lambda_d i_q - \lambda_q i_d) - T_e, \\
 0 &= -L_d i_d + L_{af} i_f - \lambda_d, \\
 0 &= -L_q i_q - \lambda_q, \\
 0 &= -L_0 i_0 - \lambda_0, \\
 0 &= -\frac{3}{2}L_{af} i_d + L_{ff} i_f - \lambda_f.
 \end{aligned} \tag{3.83}$$

The differential states are $\theta, \omega, \lambda_{dq0}, \lambda_f$ and the algebraic states are T_e, i_{dq0}, i_f . The voltages v_{dq0} and v_f are external inputs and depend on the connected network. The nominal power is controlled by P^{ref} .

The differential equation system is specified in a rotating $dq0$ -reference frame, which is rotating with the angular velocity of the rotor. This frame has not necessarily the same angular velocity as the $dq0$ -reference frame of the connected grid. In order to make them compatible, the reference frame of the generator needs to be converted.

Remark 3.2

The DAE model (3.82) and (3.83) will represent a component connected to a bus of a MG. The equations will be part of a holistic dynamical equation system, which represents the network as well as the connected components. Thereby, the external voltages v_{dq0} will be states of the network model and will be determined by the network dynamics.

Extensions and Simplifications

The SG model presented above is a dynamic model that is suitable to describe fast transients and high-frequency effects. However, depending on the use case, there are many extensions and simplifications available [60, 67].

- After a sudden change in load, the power angle of the generator is adjusted dynamically. However, the transition is not smooth but instead the rotor is swinging around the new power angle. To prevent these oscillations, it is common to include so-called damper windings in the rotor. Similar to the principle of induction machines, an current gets induced if the frequency of the rotor differs from the net frequency. This leads to a torque that damps the oscillations. In addition, these windings help to start the generator. In the model, these winding are represented by two additional differential states, usually denoted by λ_{1d} and λ_{1q} .

- In SGs with many pole pairs, the rotor is often almost perfectly round. In this case, it is a reasonable assumption that the inductances are constant.
- In balanced operation, the zero components are i_0, ν_0 are zero. Therefore the corresponding differential equations can be omitted.
- With the additional assumption that the excitation current is constant, the SG can be represented in steady state by an equivalent circuit.
- The model assumes a droop control for the mechanical torque and the excitation voltage is an external input. However, there are different models available to describe the behavior of the control mechanism and they can be incorporated in the model equations.
- In normal operation, the derivative of the flux linkage $\frac{d}{dt}\lambda_d, \frac{d}{dt}\lambda_q$ is small and can be neglected. With this assumption λ_d, λ_q turn into algebraic states.

3.3.2 Inverters

Many MG components provide electrical energy either as DC or as AC with a different frequency. In order to connect these components, the electrical energy needs to be transformed to AC at grid frequency. An inverter is a power electronic device that changes DC to AC and is used as an interface to the connected grid on most components. In this section, we will focus on the inverter design of batteries. They play a crucial role in MG control because they not only have the possibility to store energy, but they can also be controlled in a very flexible way. On a physical level they provide DC and for the connection to the grid a three-phase DC to AC inverter is needed. In addition, the inverter is used to control the energy flow of the battery. PV systems also provide DC on a physical level and are interfaced by an inverter, but they are always operated at maximal load, because the electrical energy comes without costs. Therefore, they can be represented as passive negative load and we will consider them in the next section. A detailed description of control methods for inverters in AC power grids can be found in [96]. However, we will focus on the fundamental concepts and follow the lines of [67].

In Figure 3.5 the basic structure of a three phase DC to AC inverter is depicted. Depending on the connected device on the DC side, the voltage may be varying. Therefore, a DC-link capacitor is used to provide a more stable DC voltage. However, we assume that the DC voltage is constant. The main part is the inverter stage, which is a circuit of transistors. Figure 3.6 shows a sketch of three-phase H-bridge inverter. The layout of the circuit allows to forward the input voltage V_{dc} to any pair of output

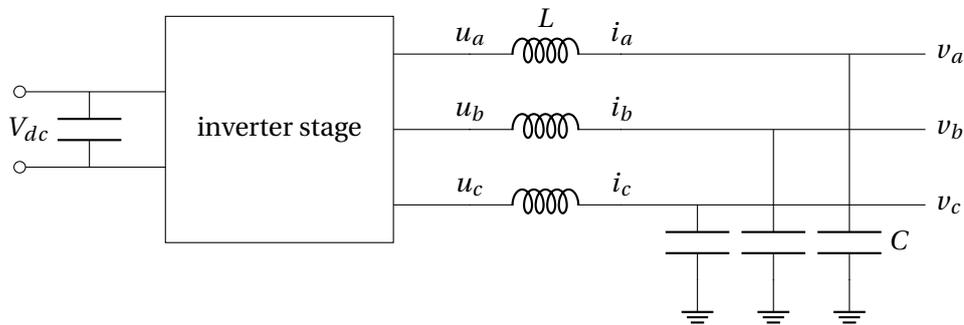


Figure 3.5: Sketch of a three phase inverter with an LC filter.

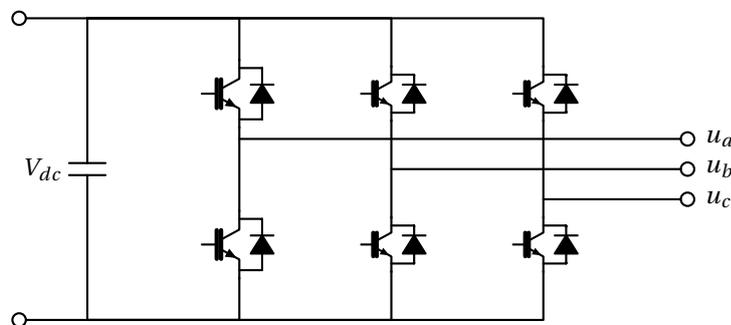


Figure 3.6: Schematic of a three phase DC to AC H-bridge inverter.

wires and allows to invert its polarity. In addition, the voltage can be switched on or off. This leads to a square voltage u_{abc} at the terminals with the magnitude V_{dc} .

The transistors are controlled by Pulse Width Modulation (PWM) at a high frequency, usually in the range of kHz. The PWM aims at creating a periodic square wave with a low harmonic distortion. There are a variety PWM methods, circuit topologies and filter configurations available suitable for different use cases. Since the switching operates on significantly higher frequency than the net frequency, it is a common assumption that the PWM unit is able to provide the setpoints for voltage immediately and that the inverter acts as an ideal voltage source.

The output of the inverter stage is connected to a low pass filter. In Figure 3.5 a typical LC filter configuration is depicted, but other filters are available. The low-pass filter is responsible to eliminate the high order harmonics of u_{abc} to get a sinusoidal output voltage v_{abc} .

PWM is switching the transistors at a high frequency. The duty cycle $d_{abc} \in [-1, 1]^3$ is the proportion of time when the transistors are open and it is the direct control input of the inverter stage. Since the PWM frequency is significantly higher than the net frequency, it is a common assumption that the voltage after the inverter stage is directly controllable. With the duty cycle as control input, the voltage is

$$u_{abc} = V_{dc}d_{abc}. \quad (3.84)$$

In contrast to traditional generators, inverters have no inertia and therefore they have no physical feedback mechanism. They are able to provide voltage independently of the connected grid and in order to be compatible, they need to be controlled accordingly. Therefore, inverters are usually equipped with additional control loops. The voltage and the current at the terminals are measured constantly and used as feedback signals. The control laws for inverter control are formulated in $dq0$ quantities and therefore, the measurements i_{abc}, u_{abc} need to be converted before they can be used in the controller logic. The phase angle θ is provided externally or computed from measurements. There are three basic control strategies: grid-forming, grid-following and grid-supporting inverters.

Grid-forming inverters are controlled to deliver AC at a fixed frequency and with a fixed voltage magnitude. Therefore, they are also called Voltage Source Inverter (VSI). The active and reactive power are defined by the interaction with the network. Typical applications of a grid forming inverters are small microgrids, where they define the net frequency. In PF calculations, they are usually represented as the slack bus. However, they are unable to operate in parallel with other grid forming inverters or generators. Because there is no feedback mechanism, the frequency of multiple inverters will go out of synchronization and this will lead to a high power transfer between the different inverters.

Grid-following inverters are controlled to provide active and reactive power. Frequency and voltage are measured constantly and the output is adjusted accordingly. They can be represented as P-Q bus in PF calculations. Since they adapt to the current frequency and voltage, they can not be operated without a grid forming inverter or generator, which is defining the net frequency. In an energized grid, multiple grid-following inverters can be operated in parallel.

Grid-supporting inverters are grid-forming inverters with an additional control loop. The controller design mimics the behavior of a SG in order to promote a fair load sharing and to stabilize frequency and voltage. This is achieved by a negative linear relationship between active power and frequency respectively volt-

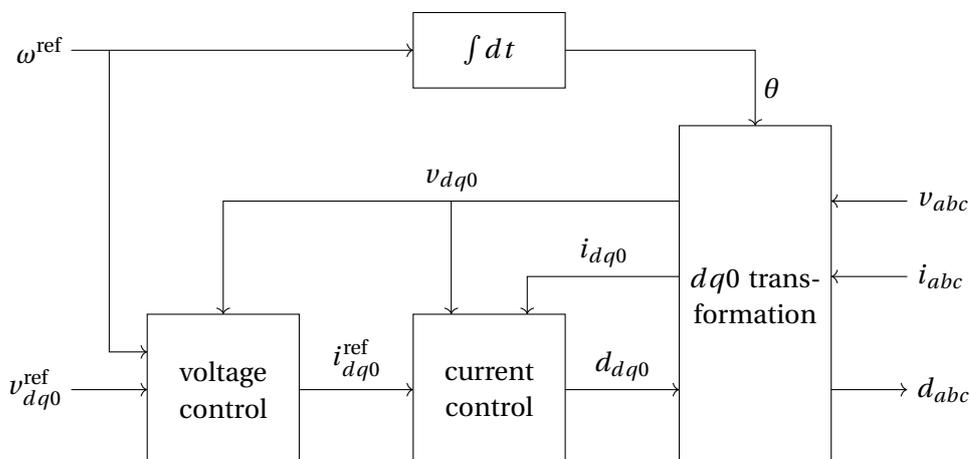


Figure 3.7: Signal flow of a grid-forming inverter.

age magnitude and reactive power. Grid supporting inverters can be operated alone as well as in parallel to other generators or inverters. In PF studies, they have no direct representation.

3.3.3 Grid-Forming Inverter

The grid-forming inverter aims at providing a terminal voltage v_{abc}^{ref} with constant RMS value E^{ref} at a fixed frequency ω^{ref} . Using the $dq0$ -transformation with the current phase angle $\theta(t) = \omega^{\text{ref}}t$, the control law

$$d_d = V_{dc}^{-1} u_d^{\text{ref}}, \quad d_q = V_{dc}^{-1} u_q^{\text{ref}}, \quad d_0 = 0 \quad (3.85)$$

allows to control the internal voltage directly. But to control the voltage v_{abc} at the terminal of the inverter, the dynamics of the filter and the conditions at the connected grid need to be considered. Figure 3.7 shows the signal flow of a practical inverter design with two control loops. An inner feedback loop controls the current and an outer control loop controls the terminal voltages.

The purpose of current control is to steer the terminal currents to the reference value i_{dq0}^{ref} . It receives as feedback signal the output currents i_{dq0} and voltage v_{dq0} and it is controlled by reference values for current i_{dq0}^{ref} and frequency ω^{ref} . The terminal

currents i_{dq0} are governed by the inductor dynamics

$$\begin{aligned}\frac{d}{dt}i_d &= \omega^{\text{ref}}i_q + \frac{1}{L}(u_d - v_d), \\ \frac{d}{dt}i_q &= -\omega^{\text{ref}}i_d + \frac{1}{L}(u_q - v_q).\end{aligned}\tag{3.86}$$

The current control law aims at eliminating the cross terms in equation (3.86) and implementing a Proportional-Integral (PI) control law to steer the current to the desired setpoints

$$\begin{aligned}d_d(t) &= V_{dc}^{-1} \left(v_d(t) - \omega^{\text{ref}}Li_q(t) + k_p(i_d^{\text{ref}}(t) - i_d(t)) + k_i \int_0^t (i_d^{\text{ref}}(\tau) - i_d(\tau))d\tau \right), \\ d_q(t) &= V_{dc}^{-1} \left(v_q(t) - \omega^{\text{ref}}Li_d(t) + k_p(i_q^{\text{ref}}(t) - i_q(t)) + k_i \int_0^t (i_q^{\text{ref}}(\tau) - i_q(\tau))d\tau \right).\end{aligned}\tag{3.87}$$

Replacing of u_d and u_q in Equation (3.86) by Equation (3.87) leads to the closed loop dynamics

$$\begin{aligned}L\frac{d}{dt}i_d &= k_p(i_d^{\text{ref}}(t) - i_d(t)) + k_i \int_0^t (i_d^{\text{ref}}(\tau) - i_d(\tau))d\tau, \\ L\frac{d}{dt}i_q &= k_p(i_q^{\text{ref}}(t) - i_q(t)) + k_i \int_0^t (i_q^{\text{ref}}(\tau) - i_q(\tau))d\tau.\end{aligned}\tag{3.88}$$

Since the cross-terms are eliminated, the system consists of two decoupled PI control laws which steer the current to the desired reference value. The proportional part of the control law (3.88) is tuned by the parameter $k_p \in \mathbb{R}$ and the integral part by $k_i \in \mathbb{R}$.

The voltage is controlled in an outer feedback loop and receives the terminal voltage v_{dq0} as an input signal. The reference value for the voltage v_{dq0}^{ref} is given as an external input. A common choice is to set $v_d^{\text{ref}} = \sqrt{2}E^{\text{ref}}$ and $v_q^{\text{ref}} = v_0^{\text{ref}} = 0$ since the control logic aims at a constant voltage. Here, E^{ref} refers to the voltage reference given as RMS value. The basic principle of the outer control loop is similar to the inner control loop. The dynamics of the capacitor in $dq0$ quantities are

$$\begin{aligned}\frac{d}{dt}v_d &= \omega^{\text{ref}}v_q + \frac{1}{C} \left(i_d^{\text{ref}} - \frac{v_d}{R} \right), \\ \frac{d}{dt}v_q &= -\omega^{\text{ref}}v_d + \frac{1}{C} \left(i_q^{\text{ref}} - \frac{v_q}{R} \right)\end{aligned}\tag{3.89}$$

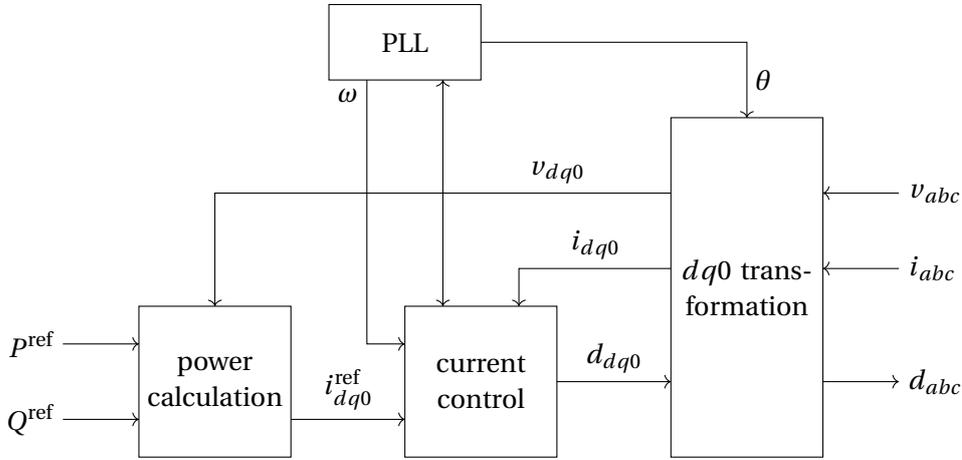


Figure 3.8: Signal flow of a grid-following inverter.

for a resistive load R . The control law

$$\begin{aligned} i_d^{\text{ref}} &= \frac{v_d(t)}{R} - \omega^{\text{ref}} C v_q(t) + k_{p,v} (v_d^{\text{ref}}(t) - v_d(t)) + k_{i,v} \int_0^t (v_d^{\text{ref}}(\tau) - v_d(\tau)) d\tau, \\ i_q^{\text{ref}} &= \frac{v_q(t)}{R} + \omega^{\text{ref}} C v_d(t) + k_{p,v} (v_q^{\text{ref}}(t) - v_q(t)) + k_{i,v} \int_0^t (v_q^{\text{ref}}(\tau) - v_q(\tau)) d\tau \end{aligned} \quad (3.90)$$

removes cross-coupling of the dynamics and allows a PI control of the voltage. Since a grid-forming inverter with this control scheme is acting like a voltage source it is also called VSI.

3.3.4 Grid-Following Inverter

Grid-following inverters are controlled as power sources. They provide active and reactive power to an already energized grid. In contrast to the grid-forming inverter, the frequency and phase angle is not directly controlled. Instead the phase angle needs to be computed from measurements. A Phase Locked Loop (PLL) computes the frequency by zeroing the quadrature-axis component of the voltage signal v_{dq0} .

This is achieved by a PI control law for the frequency

$$\begin{aligned}\frac{d}{dt}\omega(t) &= k_p v_q + k_i \int_0^t v_q(\tau) d\tau, \\ \frac{d}{dt}\theta(t) &= \omega(t).\end{aligned}\tag{3.91}$$

Similar to the grid forming inverter, the grid-following inverter is organized in two control loops. The inner control uses the same logic to control the currents as the grid forming inverter.

The outer control loop is responsible to perform the power calculations. It is controlled by reference values for active and reactive power $P^{\text{ref}}, Q^{\text{ref}}$ and receives the voltage signal v_{dq0} as feedback. The output signal is the current reference value i_{dq0}^{ref} , which is used in the inner control loop. The relation between power and current on a single phase is given by

$$\begin{aligned}P^{\text{ref}} &= \frac{1}{2} (v_d i_d^{\text{ref}} + v_q i_q^{\text{ref}}), \\ Q^{\text{ref}} &= \frac{1}{2} (v_q i_d^{\text{ref}} - v_d i_q^{\text{ref}}).\end{aligned}\tag{3.92}$$

Solving this equation for the target current i_{dq0}^{ref} leads to

$$\begin{aligned}i_d^{\text{ref}} &= \frac{2}{v_d^2 + v_q^2} (P^{\text{ref}} v_d + Q^{\text{ref}} v_q), \\ i_q^{\text{ref}} &= \frac{2}{v_d^2 + v_q^2} (P^{\text{ref}} v_q - Q^{\text{ref}} v_d).\end{aligned}\tag{3.93}$$

In steady state it is $i_{dq0}^{\text{ref}} = i_{dq0}$ and the actual provided power matches the reference values.

A typical application of a grid-following inverter is the interface of a PV-system. Here the reference values $P^{\text{ref}}, Q^{\text{ref}}$ are determined by the *Maximal Power Point Tracking* algorithm of the PV controller, which aims at the maximal utilization of the PV panels [31, 98]. In this case, the power output is not controlled externally and the inverter behaves like a negative passive P-Q load.

3.3.5 Grid-Supporting Inverters

Grid-supporting inverters are a combination of grid-forming and grid-following inverters [95]. They are build on either a grid-forming or a grid-following inverter

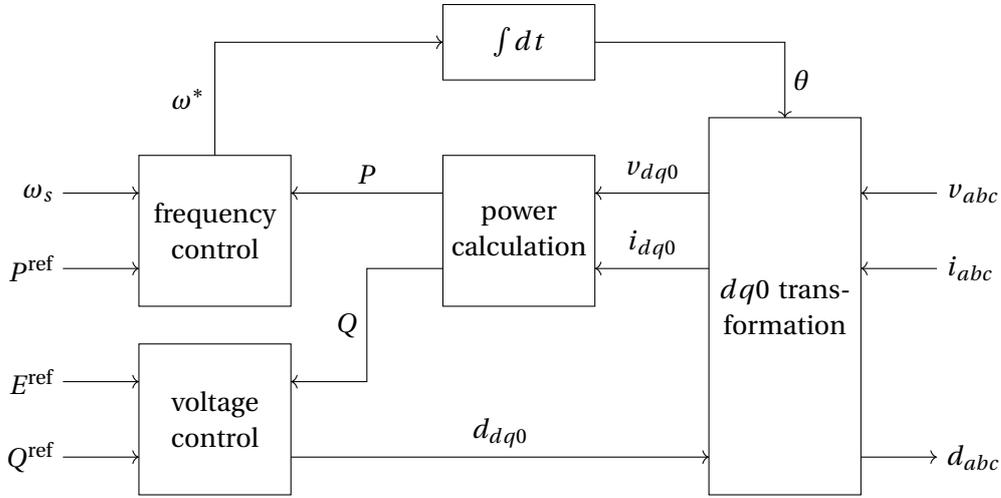


Figure 3.9: Signal flow of a grid-supporting inverter.

and proportional control laws are used to mimic the stabilizing behavior of a SG. In nominal operation, their purpose is to follow reference values for active and reactive power. During a disturbance, they deviate from the reference values in order to stabilize the grid. Multiple grid-supporting inverters can be operated in parallel and it is possible to use them to energize the grid.

Although there are multiple designs available, we consider a grid supporting inverter, which is as a grid forming inverter with an additional control mechanism

$$\omega^* = \omega_s + D(P^{\text{ref}} - P), \quad (3.94a)$$

$$u_d^* = \sqrt{2}(E^{\text{ref}} + k_q(Q^{\text{ref}} - Q)) \quad (3.94b)$$

to control the frequency and the voltage magnitude. Here, ω_s denotes the nominal frequency of the grid and E^{ref} denotes the RMS reference value for the voltage. The first control law (3.94a) causes an increased power angle, if the grid frequency ω_g is below the nominal frequency. This leads to an increased active power injection in order to steer the frequency back to the nominal value. The second control law (3.94b) provides reactive power to maintain the desired voltage magnitude at the terminal of the inverter.

The control laws (3.94) lead to a behavior that is combination of the grid-forming and grid-following inverters. For $D = 0$ the frequency of the inverter is constant and it acts like a grid-forming inverter. If $D \rightarrow \infty$, the active power matches always

the reference value, which mimics a grid-following inverter. The voltage control law acts similarly. If $k_q = 0$, the resulting voltage is constant like a grid forming-inverter and like grid-following inverters, the reactive power matches the reference value for $k_q \rightarrow \infty$. The tuning of the parameters k_q and D defines the exact behavior.

Derivation of Control Laws for Grid-Supporting Inverters

To give a deeper insight into the mechanics of grid-supporting inverters and the impact of the control laws (3.94), we will derive these laws from a simplified model. Furthermore, we will present a full state-space model including the control laws and show the behavior described above numerical experiments. Thereby, we will show, how a grid-supporting inverter is reacting to changes in the reference values as well as changes of the grid frequency.

In order to derive the control laws, we consider an inverter connected to an infinite bus in quasi-static operation as depicted in Figure 3.10. The physical properties of the connection and the filter are given by R and L . The voltage phasor at the connection to the grid is fixed to $V = 1$. The magnitude \tilde{U} and the angle offset ϕ of the inverter voltage

$$U = \tilde{U}e^{j\phi} \quad (3.95)$$

can be controlled directly. The current injected to the grid is given by

$$\begin{aligned} I &= \frac{1}{R + jX}(U - V) = \frac{1}{R + jX}(\tilde{U} \cos(\phi) + j\tilde{U} \sin(\phi) - 1) \\ &= \frac{R}{R^2 + X^2}(\tilde{U} \cos(\phi) - 1) + \frac{X}{R^2 + X^2}\tilde{U} \sin(\phi) \\ &\quad + j\left(\frac{R}{R^2 + X^2}\tilde{U} \sin(\phi) - \frac{X}{R^2 + X^2}(\tilde{U} \cos(\phi) - 1)\right) \end{aligned} \quad (3.96)$$

with $X = \omega L$. This leads to the active power injection to the grid

$$P = \Re(S) = \Re(VI^*) = \frac{R}{R^2 + X^2}(\tilde{U} \cos(\phi) - 1) + \frac{X}{R^2 + X^2}\tilde{U} \sin(\phi). \quad (3.97)$$

With the assumption that the resistive part of the transmission line is low and that the inverter is operated with ϕ close to zero, the active power can be approximated by

$$P \approx \frac{\tilde{U}}{X}\phi. \quad (3.98)$$

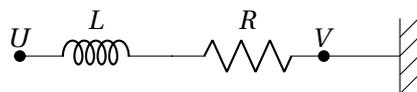


Figure 3.10: Steady-state model of an inverter connected to an infinite bus.

This shows a direct relation between the angle offset ϕ and the emitted active power P and motivates the droop control law (3.94a). In a similar way the reactive power can be approximated by

$$Q \approx \frac{1}{X}(\tilde{U} - 1). \quad (3.99)$$

This shows that the reactive power can be controlled by the magnitude of the inverter voltage. The droop control law (3.94b) is using this relation to steer the active power to the nominal operating point.

In a fully dynamic scenario, the grid frequency is not constant but instead changes over time. We represent the grid frequency by the time depend function $\omega_g(t)$. In addition, the reference values for active and reactive power P^{ref} and Q^{ref} are not constant, but may be changed by a controller on a higher control level. The full dynamic model of a grid-supporting inverter in the $dq0$ -reference frame of the connected grid is given by

$$\begin{aligned} \frac{d}{dt}\theta &= \omega^*, \\ \frac{d}{dt}i_d &= \omega_g(t)i_q + \frac{1}{L}(u_d - Ri_d - v_d), \\ \frac{d}{dt}i_q &= -\omega_g(t)i_d + \frac{1}{L}(u_q - Ri_q - v_q) \end{aligned} \quad (3.100)$$

together with the droop control laws

$$\begin{aligned} \omega^* &= \omega_s + D(P^{\text{ref}} - P), \\ u_d^* &= \sqrt{2}\left(E^{\text{ref}} + k_q(Q^{\text{ref}} - Q)\right), \end{aligned} \quad (3.101)$$

where u_d and u_q is the internal inverter voltage in the grid $dq0$ -reference frame. It is computed by

$$\begin{aligned} u_d &= u_d^* \cos(\phi), \\ u_q &= u_d^* \sin(\phi) \end{aligned} \quad (3.102)$$

with the offset $\phi(t) = \theta(t) - \int_0^t \omega_g(\tau) d\tau$. The active and reactive power is calculated from measurements on the grid side

$$\begin{aligned} P &= \frac{2}{3} (i_d v_d + i_q v_q), \\ Q &= \frac{2}{3} (i_d v_q - i_q v_d). \end{aligned} \quad (3.103)$$

We demonstrate the effect of the droop control laws by numerical simulations. We consider a scenario, where the power reference P^{ref} is changed and the grid frequency ω_g is disturbed in order to show the reaction of the inverter. We assume that the reactive power is not controlled and the voltage magnitude is fixed to $u_d^* = 1$. At the beginning, the grid is operated at the nominal frequency ω_s , the active power reference P^{ref} is zero and the inverter is in steady-state. After 1 s the power reference is set to $P^{\text{ref}} = 0.1$ from a higher control level and at $t = 5$ s, the grid frequency drops by 1% from 50 Hz to 49.5 Hz. A drop in frequency usually indicates a mismatch between the consumed and the provided active power in the grid.

In Figure 3.11 simulation results for a grid-supporting inverter connected to an infinite bus are shown. In the beginning, the complete system is in steady-state and no power is injected to the network. The change of the power reference at $t = 1$ s leads to an instantaneous jump of the internal inverter frequency and the offset ϕ increases. As a result, more active power is injected until the reference value is reached and the inverter frequency matches the grid frequency. At $t = 5$ s, the grid frequency drops and the internal frequency is higher than the grid frequency, i.e. $\omega^* > \omega_g$. Therefore, the offset ϕ and the active power injection increases above the given reference value. The increased power input counteracts the frequency deficiency and helps to steer back the frequency to nominal value. The amount of additional active power is defined by the droop parameter D and needs to be adjusted to the size of the inverter.

3.3.6 Loads

Consumers of electrical energy in a network are represented as passive loads of active and reactive power. They are not controllable and they behave unpredictable on

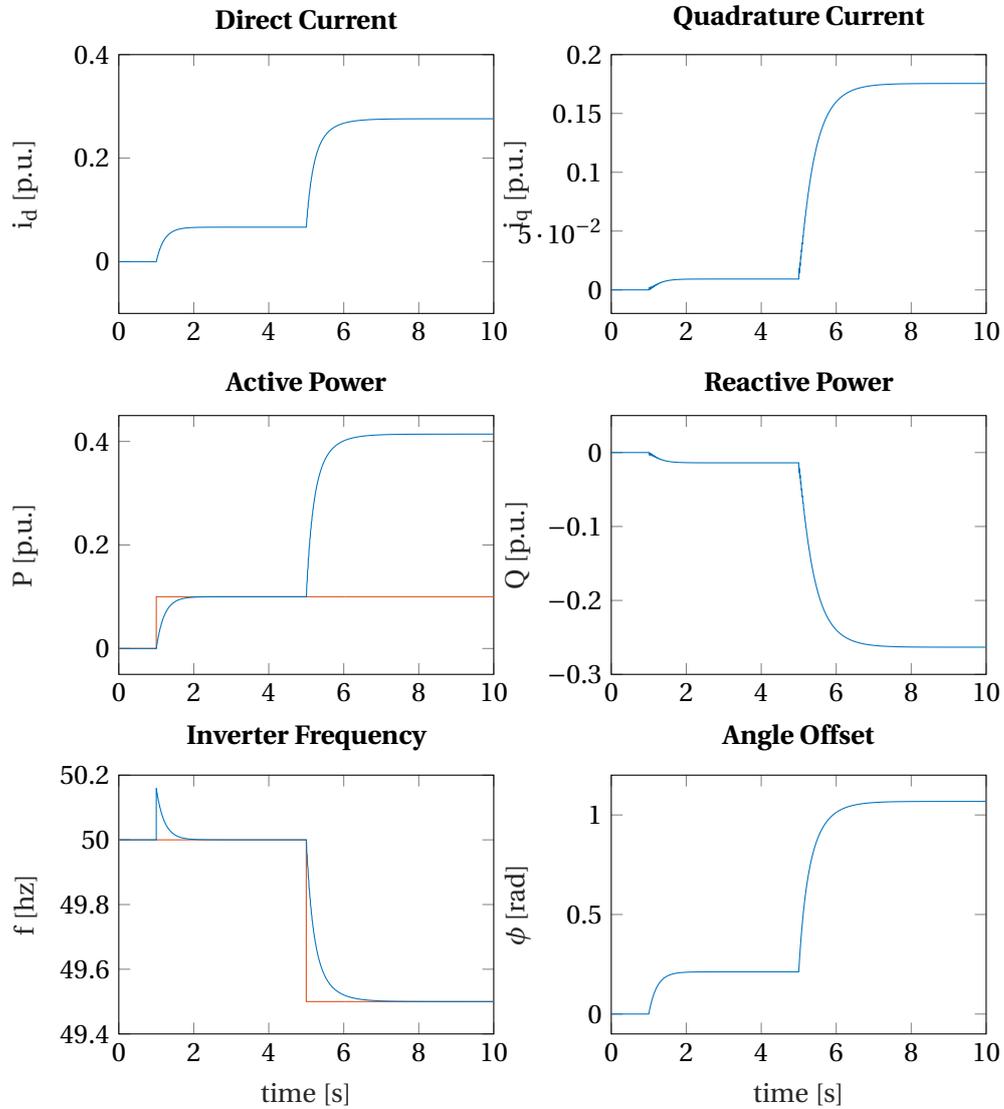


Figure 3.11: Simulation results of a grid-supporting inverter connected to an infinite bus. The blue trajectories show the states of the inverter model and the red plot indicates the active power reference respectively the grid frequency. At the nominal frequency, the inverter follows the provided reference values. At a reduced frequency, it provides additional power to stabilize the grid. The parameters are $R = 0.1$, $L = 0.01$ and $D = 10$.

a short time scale. Usually, they are modeled as a time series of passive parameters P and Q . However, a load prediction may be available on longer time scales. PV systems and other RES are typically interfaced by grid following inverters. Since they provide active and reactive power without costs, they are operated at their maximal capacity most of the time. Therefore, they act as a passive power consumer with negative load. The power input is dependent on the current weather conditions and is therefore volatile. On longer time scales, the performance can be estimated by a weather forecast. In summary, the PV systems are represented in the same way as loads.

Chapter 4

Nonlinear Model Predictive Control

The control of processes is a fundamental challenge that occurs in many areas and control engineering is a wide field of research. Typically, a process is understood as a system with control inputs and output signals which represent measurements. While slow and predictable processes can be controlled by predefined control input, most processes in the real world are subject to disturbances and their behavior is not exactly predictable. Therefore, it is necessary to react to unforeseen disturbances by a suitable feedback mechanism. Control concepts which include a closed feedback loop are referred to as *online control*.

One of the most widely used approaches in process control are proportional-integral-derivative (PID) controllers. The basic idea is to constantly measure a process variable $\xi(t) \in \mathbb{R}$ and to calculate the distance to a predefined setpoint $\bar{\xi} \in \mathbb{R}$. Feedback control $q(t) = \bar{q} + \Delta q(t) \in \mathbb{R}$ is then generated according to the control law

$$\Delta q(t) = K_p (\xi(t) - \bar{\xi}) + K_i \int_0^t (\xi(\tau) - \bar{\xi}) d\tau + K_d \frac{d}{dt} (\xi(t) - \bar{\xi}) \quad (4.1)$$

with the tuning parameters $K_p, K_i, K_d \in \mathbb{R}$. However, the controller is only able to react to instantaneous measurements and is agnostic towards the future behavior of the process. For processes with nonlinear dynamics, multiple control inputs, and process states, this approach reaches its boundaries. In order to achieve the desired process behavior, it is often necessary to build up complex control structures and this approach is very sensitive to the tuning of parameters. In addition, it is not possible to incorporate operational limits on the process variables. A comprehensive introduction to PID control can be found in [108].

Model Predictive Control (MPC) is an approach to overcome the limitations of PID control by using a dynamic model of the process. This allows to predict the reaction of the process to control actions for a certain time horizon. Feedback is generated by repeatedly solving an optimization problem. MPC is not limited to the tracking of setpoints, but the formulation of an objective function allows more general control targets. In addition, path constraints and bounds on the states as well as the controls can be incorporated. Overall, MPC is more powerful than PID control. However, the

downside of MPC is that a model of the process is necessary in order to predict its future behavior. Depending on the process, intensive research may be necessary to understand the dynamics, build a suitable model, and calibrate the parameters. Furthermore, the computational effort of MPC is significantly higher than for PID control.

MPC is not a single algorithm, but a general class of control algorithms. A wide range of models, objective functions, and operational constraints can be treated by MPC. In addition, many specialized solution approaches are proposed. However, we focus on MPC with Differential Algebraic Equation (DAE) models. The goal of this chapter is to introduce the fundamentals of optimal control and MPC. Therefore, we define a general class of parametric Optimal Control Problems (OCPs) and we describe multiple shooting as a discretization method. We summarize the Real-Time Iteration (RTI) as a modern MPC scheme, which aims at high sampling frequencies and low feedback delays. Even though, the RTI is significantly faster than classical MPC approaches, it is not able to meet the real-time requirement of secondary Microgrid (MG) control. We use the RTI as a starting point for the development of a novel MPC scheme in Chapter 5. The new scheme is based on the Multi-Level Iteration (MLI) and is able to give feedback with significantly higher feedback rates. In contrast to PID controller, it is able to respect operational bounds.

A comprehensive overview of MPC and its theoretical background can be found in [94] and a review about the historical development in [64]. An introduction from the perspective of linear MPC is given in [42]. In this chapter, we focus on the RTI, which was originally introduced in [24]. It is an established technique with many successful applications, for example in process control [26, 35]. In addition, many extensions were proposed to more specialized use cases, like the control of processes with integer decision variables [57] or uncertainties [65].

4.1 Principle of Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) is an established approach to control dynamic processes online. The main idea of NMPC is to generate feedback by repeatedly solving an OCP. The process time is divided by an equidistant sampling grid t_0, t_1, \dots with a fixed *sampling time* Δt . On every sampling point t_k , the current process state $\xi_k \in \mathbb{R}^{n_x}$ is estimated or measured. Based on the current state, a parametric OCP on a finite time horizon $[t_k, t_k + T]$ with a length $T \geq \Delta t$ is solved in order to compute a control answer $u_{\xi_k} \in \mathbb{R}^{n_u}$, which is applied in the interval $[t_k, t_{k+1}]$. In simulations, we have two components which are interacting with each other. The *process* represents the reality and constantly provides measurements. The *controller*

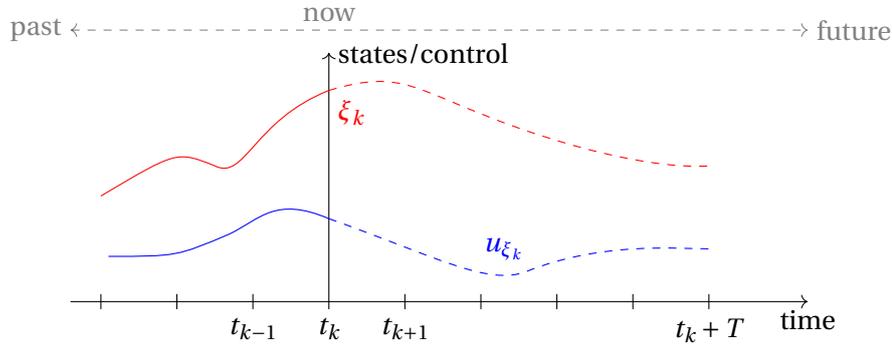


Figure 4.1: Idealized MPC process. The state is depicted in red, the control in blue. At time t_k , the controller receives the current state ξ_k and computes an optimal feedback control u_{ξ_k} on the prediction horizon $[t_k, t_k + T]$.

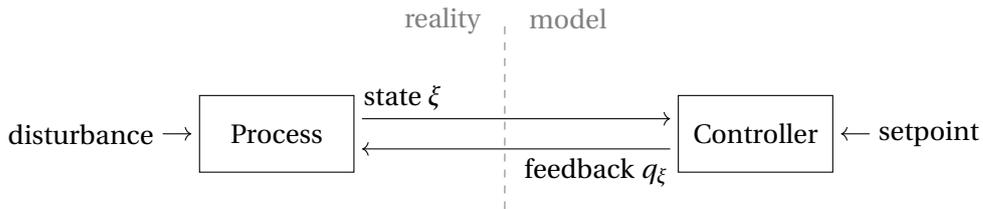


Figure 4.2: Signal flow of MPC algorithm.

uses these measurements to compute a feedback signal in order to react to disturbances. The principle of NMPC is shown in Figure 4.1 and Figure 4.2 depicts the basic signal flow of a closed loop feedback control setup. The core components of every MPC algorithm are the process model and the parametric OCP, which we will discuss in the following sections.

In many real world applications, the system state is not directly observable. Only a subset of the states may be directly or indirectly measurable, and measurement errors may occur. Therefore, a state estimator is often necessary as a third component. However, in this section, we assume that the state is completely available with no measurement errors. Algorithms for state estimation will be discussed in Section 7.

4.2 Process Model

The key component of NMPC is a model that predicts the behavior of the process in the future depending on its current state. In this thesis, we focus on DAE models on a time interval $[t_0, t_f]$. The state of the process is described by differential states $x : [t_0, t_f] \rightarrow \mathbb{R}^{n_x}$ and algebraic states $z : [t_0, t_f] \rightarrow \mathbb{R}^{n_z}$. The process is controlled by an external control function $u : [t_0, t_f] \rightarrow \mathbb{R}^{n_u}$. The model is given by

$$\frac{dx}{dt}(t) = f(x(t), z(t), u(t)), \quad (4.2a)$$

$$0 = g(x(t), z(t), u(t)), \quad (4.2b)$$

$$x(t_0) = x_0 \quad (4.2c)$$

with an initial value $x_0 \in \mathbb{R}^{n_x}$ as well as model functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_z}$.

Assumption 4.1

We assume that the model functions f and g are sufficiently smooth and that for a given control function u and initial value x_0 , the solution of the system exists and is unique. In addition, we assume that the DAE system is of index 1, i.e. the Jacobian $\frac{d}{dz}g(x(t), z(t), u(t))$ is nonsingular.

Remark 4.1

The models of the MG components presented in Chapter 3 are given as DAEs. The right hand side functions f and g will represent the network as well as the connected components in numerical experiments.

4.2.1 Solution of DAE System

Throughout the thesis, we use the solution of DAE systems very often. Therefore, we introduce the notation

$$x(t; t_0, x_0, z_0, q) : [t_0, t_f] \rightarrow \mathbb{R}^{n_x}, \quad (4.3a)$$

$$z(t; t_0, x_0, z_0, q) : [t_0, t_f] \rightarrow \mathbb{R}^{n_z} \quad (4.3b)$$

for the solution trajectory of the following modified DAE system

$$\frac{dx}{dt}(t) = f(x(t), z(t), q), \quad (4.4a)$$

$$0 = g(x(t), z(t), q) - e^{-\alpha \frac{t-t_0}{t_f-t_0}} g(x_0, z_0, q), \quad (4.4b)$$

$$x(t_0) = x_0, \quad (4.4c)$$

$$z(t_0) = z_0. \quad (4.4d)$$

with a constant control parameter $q \in \mathbb{R}^{n_u}$, an initial algebraic state z_0 and a damping factor $\alpha \in \mathbb{R}_{>0}$.

In comparison to the original system (4.2), this system shows two modifications. First, the algebraic equations (4.2b) are replaced by a relaxed algebraic equation

$$0 = g(x(t), z(t), q) - e^{-\alpha \frac{t-t_0}{t_f-t_0}} g(x_0, z_0, q). \quad (4.5)$$

By this relaxation, the initial algebraic states are treated as independent variables. This allows to employ bounds directly on the algebraic states in DAE constrained optimization problems. Details on this relaxation technique can be found, e.g., in [66, 104].

Second, the control function u needs to have a finite-dimensional parametrization in order to be tractable by numerical integrators. We assume that

$$u(t) = \varphi(t, q) \in \mathbb{R}^{n_u}, \quad (4.6)$$

where $\varphi(\cdot, q) : [t_0, t_f] \rightarrow \mathbb{R}^{n_u}$ is a function with local support parametrized by the vector $q \in \mathbb{R}^{n_\varphi}$. Typically, $\varphi(\cdot, q)$ is a polynomial and q refers to its coefficients. However, any DAE system with a finite-dimensional control parametrization can be reformulated as a DAE system with constant control parametrization. To simplify the notation, we assume w.l.o.g. that the control function is constant. In the following, the control function is given by

$$u(t) = q \in \mathbb{R}^{n_u} \quad (4.7)$$

for $t \in [t_0, t_f]$ and we treat the control as parameters of the DAE system (4.4).

4.3 Parametric Optimal Control Problem

At every sampling point t_k the current system state ξ_k is used to set up a parametric OCP with a horizon length T

$$\min_{x(\cdot), z(\cdot), u(\cdot)} \Phi(x(\cdot), z(\cdot), u(\cdot)) \quad (4.8a)$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), z(t), u(t)), \quad 0 = g(x(t), z(t), u(t)), \quad (4.8b)$$

$$x(t_k) = \xi_k, \quad (4.8c)$$

$$x^{\text{lo}} \leq x(t) \leq x^{\text{up}}, z^{\text{lo}} \leq z(t) \leq z^{\text{up}}, u^{\text{lo}} \leq u(t) \leq u^{\text{up}}, \quad (4.8d)$$

$$t \in [t_k, t_k + T]. \quad (4.8e)$$

The control function $u : \mathbb{R} \rightarrow \mathbb{R}^{n_u}$ is a measurable independent function. The definition of the DAE system (4.8b) – (4.8c) and the control function u refers to the process model described in Section 4.2. The NMPC feedback signal applied in the interval $[t_k, t_{k+1}]$ is the first part of the solution $u_{\xi_k}^*(t) = u^*(t)$.

Remark 4.2

In order to improve the readability, we restrict ourselves to problems with box constraints. However, the presented OCP can be extended, e.g., by path or interior point constraints. A comprehensive description of a more general OCP can be found, e.g., in [113].

4.3.1 Objective Function

Typically, there is a distinction between *tracking* and *economic* MPC. In tracking MPC the objective functions aims at steering the system state to a predefined reference trajectory. This is the direct approach to replace the PID control law (4.1). Throughout this thesis, we will primarily deal with tracking MPC even though the concepts are applicable to a wider problem class. In economic MPC, a more general objective function is used to control the process optimally in an economic way. The generic objective functional usually consists of an integral contribution, the Lagrange type objective with integrand ϕ^l , and an end-point contribution, the Mayer objective ϕ^m :

$$\Phi(x(\cdot), z(\cdot), u(\cdot)) = \int_{t_k}^{t_k+T} \phi^l(x(\tau), z(\tau), u(\tau)) d\tau + \phi^m(x(t_k+T), z(t_k+T)). \quad (4.9)$$

In the special case of tracking MPC, the objective function aims at minimizing the distance to a predefined reference trajectory given by error functions e^l and e^m :

$$\Phi(x(\cdot), z(\cdot), u(\cdot)) = \int_{t_k}^{t_k+T} \|e^l(x(\tau), z(\tau), u(\tau))\|^2 d\tau + \|e^m(x(t_k+T), z(t_k+T))\|^2. \quad (4.10)$$

4.3.2 Discretization

OCP (4.8) is formulated as a continuous optimization problem. Even though, the solution of this problem gives theoretically optimal feedback, in practice it is not di-

rectly applicable. In order to solve it by numerical methods, a discretization is necessary. In addition, the controlled process may have a minimal sampling rate and can only accept constant controls. Therefore, the process may have an intrinsic discrete character and a discretization which reflects this properly is suitable. We use the direct multiple shooting method [14] to transform the infinite-dimensional OCP into a finite-dimensional, structured Nonlinear Programming Problem (NLP). Besides multiple shooting, single shooting [99] and collocation [12] are the most common discretization methods.

The main idea of multiple shooting is to divide the prediction horizon $[t_k, t_k + T]$ into $N \in \mathbb{N}$ shooting intervals

$$t_k = \tau_0 < \tau_1 < \dots < \tau_N = t_k + T. \quad (4.11)$$

On every shooting interval, the control is w.l.o.g. assumed to be constant, i.e. $u(t) = q_i$ for $t \in [\tau_i, \tau_{i+1}]$. Intermediate variables for the differential and algebraic states s_i^x, s_i^z are introduced for every shooting node $i = 0, \dots, N$. The intermediate states and controls are collected in $s^x = (s_0^x, \dots, s_N^x) \in \mathbb{R}^{(N+1)n_x}$, $s^z = (s_0^z, \dots, s_N^z) \in \mathbb{R}^{Nn_z}$ and $q = (q_0, \dots, q_{N-1}) \in \mathbb{R}^{Nn_u}$. The DAE system is separately solved on every shooting interval and matching conditions are introduced to ensure a continuous solution trajectory. The discretized optimization problem reads as

$$\min_{s^x, s^z, q} \sum_{i=0}^N l_i(s_i^x, s_i^z, q_i) \quad (4.12a)$$

$$\mathbf{s.t.} \quad 0 = x(\tau_{i+1}; \tau_i, s_i^x, s_i^z, q_i) - s_{i+1}^x, \quad i = 0, \dots, N-1 \quad (4.12b)$$

$$0 = g(s_i^x, s_i^z, q_i), \quad i = 0, \dots, N \quad (4.12c)$$

$$s_0^x = \xi_k, \quad (4.12d)$$

$$x^{\text{lo}} \leq s_i^x \leq x^{\text{up}}, z^{\text{lo}} \leq s_i^z \leq z^{\text{up}}, \quad i = 0, \dots, N \quad (4.12e)$$

$$u^{\text{lo}} \leq q_i \leq u^{\text{up}}, \quad i = 0, \dots, N-1. \quad (4.12f)$$

Here, the relaxed DAE formulation (4.4) is used. The consistency conditions (4.12c) ensure that every feasible point refers to the solution of the original DAE system. The individual summands of the objective function are given by

$$l_i(x, z, q) = \begin{cases} \int_{\tau_i}^{\tau_{i+1}} \phi^l(x(\tau; \tau_i, x, z, q), z(\tau; \tau_i, x, z, q), q) d\tau & \text{if } i = 0, \dots, N-1 \\ \phi^m(x, z) & \text{if } i = N \end{cases} \quad (4.13)$$

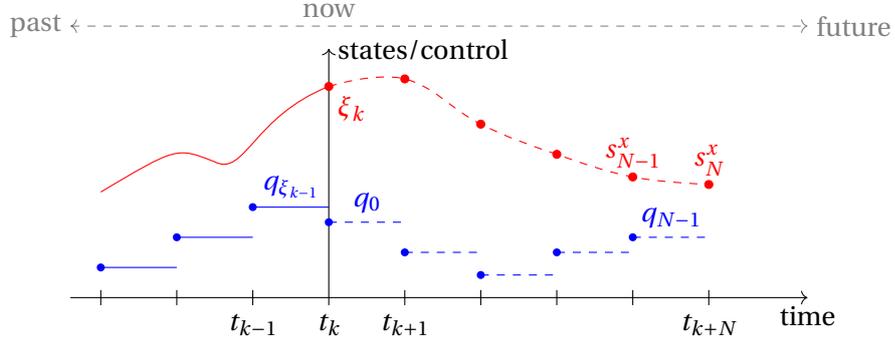


Figure 4.3: Discretized MPC process. The controls are depicted in blue and the states in red. Because of the discretization, the controls are constant in the sampling intervals. The solid lines represent the past states and controls, while the dashed lines indicate the optimized model predictions of the controller.

with $x \in \mathbb{R}^{n_x}$, $z \in \mathbb{R}^{n_z}$ and $q \in \mathbb{R}^{n_q}$. Figure 4.3 shows an exemplary MPC process using a discretized OCP with constant controls.

In order to have a more compact notation, we summarize all optimization variables in a vector $v = (s^x, s^z, q) \in \mathbb{R}^{n_v}$. The equality constraints (4.12b)–(4.12d) are collected in the function $b(v) \in \mathbb{R}^{n_b}$ in combination with a constant embedding matrix E . This allows us to write the discretized OCP in the compact form

$$\text{NLP}(\xi_k) = \min_{v \in \mathbb{R}^{n_v}} l(v) \quad (4.14a)$$

$$\text{s.t.} \quad b(v) + E\xi_k = 0, \quad (4.14b)$$

$$v^{\text{lo}} \leq v \leq v^{\text{up}}. \quad (4.14c)$$

Here, l is the discretized objective function (4.8a), the function b together with the constant matrix E represent the discretized DAE system (4.8b) with the initial value embedding constraint (4.8c), and v^{lo} and v^{up} are the lower and upper bounds on states and controls.

4.4 Classical Nonlinear Model Predictive Control

Problem (4.14) is an NLP and can be treated by any suitable NLP solver. In the classical NMPC approach, a generic NLP method is used to solve Problem (4.14) in every sampling interval and the solution is used to generate feedback. However, the down-

side of this approach is the high computational effort. Almost all NLP solvers implement iterative methods, and many iterations may be necessary to achieve convergence. In addition, every iteration includes the evaluation of the constraints (5.1b) and thus the integration of the DAE system. For derivative-based algorithms, even the corresponding sensitivities must be generated. Depending on the system dynamics, this is a computationally expensive task as the DAE integrator may need a lot of steps. Therefore, this approach is only feasible for applications with low sampling rates. The classical approach is already described in [19, 20] and is the subject of several textbooks [94, 44].

Furthermore, the OCP depends on the current measurement ξ_k and the calculation can only begin when ξ_k is available. This leads to a feedback delay between the measurement and the control answer, the so-called *feedback time*. In the meantime, the process continues and the computed feedback control is the reaction to an outdated system state. The classical NMPC approach includes the solution NLP (4.14) until convergence at every sampling point and has therefore a long feedback time. This renders the classical NMPC only feasible for slow processes.

Conventional NMPC uses a generic NLP solver and starts with a new initial guess in every iteration. Similarities between subsequent NLPs are not considered. The fundamental idea of the RTI is to exploit such similarities in order to speed up the computation process. Since the RTI is built on Sequential Quadratic Programming (SQP), we introduce it in the next section.

4.5 Sequential Quadratic Programming

SQP is a class of iterative methods to solve general inequality constrained NLPs. It is the basis of the RTI, which will be discussed in the next section. A detailed description of SQP and its convergence properties can be found in [80], but we restrict ourselves to a compact overview. The main idea is to find critical points of the Lagrange function of Problem (4.14),

$$\mathcal{L}(v, \lambda, \mu) = l(v) - \lambda^\top (b(v) + E\xi_k) - (\mu^{\text{lo}})^\top (v - v^{\text{lo}}) - (\mu^{\text{up}})^\top (v^{\text{up}} - v), \quad (4.15)$$

where $\lambda \in \mathbb{R}^{n_b}$ is the dual multiplier for the equality constraints (5.1b) and $\mu = [\mu^{\text{lo}}, \mu^{\text{up}}]$ with $\mu^{\text{lo}}, \mu^{\text{up}} \in \mathbb{R}_{\geq 0}^{n_v}$ for the bounds (5.1c). Critical points of the Lagrange function refer to primal-dual points (v, λ, μ) , which satisfy the well-known Karush-Kuhn-Tucker (KKT) conditions. The application of Newton's method to the KKT conditions, leads to an iterative solution process. Starting from an initialization point, a sequence of primal-dual iterates $(v_j, \lambda_j, \mu_j)_{j \in \mathbb{N}}$ is generated, which is converging

to a critical point of the NLP under suitable assumptions. Since the KKT conditions state an inequality system, in every iteration the Quadratic Programm (QP)

$$QP(v_j, \lambda_j, \xi_k) = \min_{\Delta v} \frac{1}{2} \Delta v^\top A(v_j, \lambda_j) \Delta v + a(v_j)^\top \Delta v \quad (4.16a)$$

$$\text{s.t. } B(v_j) \Delta v + b(v_j) + E \xi_k = 0, \quad (4.16b)$$

$$v^{\text{lo}} \leq \Delta v + v_j \leq v^{\text{up}}. \quad (4.16c)$$

is solved. The matrix $A(v_j, \lambda_j)$ is either the Hessian of the Lagrange function with respect to v or an approximation of it. We will discuss Hessian approximations in Section 4.5.1. Note that the dual multipliers for the bounds μ only enter the Lagrange function linearly and therefore do not appear in the Hessian. The linear objective term is defined by the objective gradient $a(v_j) = \nabla_v l(v_j)$ and the constraints (4.16b) are linearizations of (5.1b) based on $b(v_j)$ and its Jacobian $B(v_j) = \nabla b(v_j)^\top$. The solution $(\Delta v^{QP}, \lambda^{QP}, \mu^{QP})$ of QP (4.16) is used to update the primal-dual variables

$$v_{j+1} = v_j + \Delta v^{QP}, \quad \lambda_{j+1} = \lambda^{QP}, \quad \mu_{j+1} = \mu^{QP}. \quad (4.17)$$

SQP is an established method with strong convergence properties. Besides interior-point methods, SQP is one of the most widely used methods to solve NLPs. In state-of-the-art implementations, there are a lot of additional techniques to improve convergence properties and computation time. A comprehensive overview of SQP and additional techniques can be found, e.g., in [80].

4.5.1 Hessian Approximations

The original SQP method uses the Hessian of the Lagrangian $\mathcal{L}(v, \lambda, \mu)$ with respect to v as the matrix $A(v, \lambda)$. However, using the exact Hessian in practical applications may have significant disadvantages. First, the exact Hessian includes second-order derivatives. Since the evaluation of the constraints includes the integration of a DAE system, the computational effort for second-order sensitivities is significantly higher than for first-order derivatives. Second, in general the exact Hessian is not positive definite on the null space of the constraints. Since this leads to non convex QPs with possibly non-unique solutions, the SQP method might not converge.

In many applications, it has turned out to be advantageous to use an approximation of the Hessian instead of the exact one. This leads to a family of so-called Quasi-Newton methods to solve NLPs. It can be shown that under certain conditions Quasi-Newton methods still converge with a super-linear convergence rate. Among

the most common methods Hessian approximations in the research field of MPC are:

BFGS-Updates Broyden–Fletcher–Goldfarb–Shanno (BFGS) update formulas are used to update the Hessian approximation in every iteration. Starting with an initial positive definite matrix, gradient information is used to perform matrix updates which ensure that the matrix remains positive definite. A detailed description can be found, e.g., in [80, Chapter 8].

Constant Hessian In several applications it has turned out that a constant positive definite matrix, like e.g. the unity matrix, leads to a sufficient contraction.

Gauß-Newton The Gauß-Newton approximation is a special approximation for least-squares objective functions of the form

$$l(v) = \frac{1}{2} \|e(v)\|^2 \quad (4.18)$$

with a smooth error function e . The Hessian is approximated by

$$A(v, \lambda) = \nabla e(v)^\top \nabla e(v) \quad (4.19)$$

which refers to a linearization inside the norm. This method originates from least-squares problems. In the context of NLP (4.14), such an objective function may arise, when only a Mayer term is used or when the Lagrange term is a continuous least-squares objective function. Besides the reduced computational effort, this approximation enforces convergence to only statistically stable points, which is a desirable property for parameter estimation problems [80, Chapter 10].

4.6 Real-Time Iteration

The RTI is an approach to reduce the computation time of the classical NMPC algorithm drastically. It is based on SQP, but the complete iterative solution process of a generic NLP method is avoided by utilizing the similarities between subsequent sampling intervals. The RTI exploits that under suitable assumptions local quadratic convergence of SQP is guaranteed. As a consequence, once the iterates are sufficiently close to the true solution, only one iteration per sampling time is sufficient to obtain excellent solution approximations for the NLPs in subsequent sampling intervals [24]. Therefore, the RTI does not solve NLP(ξ_k) at every sampling point t_k until

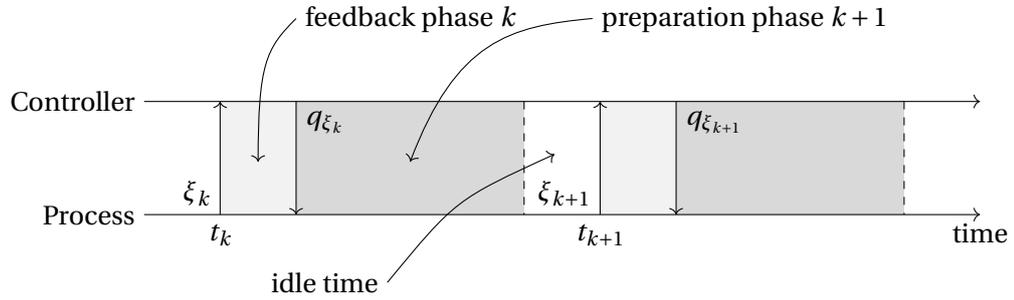


Figure 4.4: Temporal communication between the controller and the process. The transition phase is not depicted because their computation time is negligible.

convergence. Instead, one set of primal-dual iteration variables (v_k, λ_k, μ_k) is used over the complete simulation time and in every sampling interval, only $\text{QP}(v_k, \lambda_k, \xi_k)$ is solved. This approach reduces the computational effort from solving one complete NLP to just one QP in every sampling interval.

In addition, the structure of NLP (4.14) allows to reduce the feedback time significantly. In order to solve QP (4.16), the objective function and the constraints need to be evaluated on the current linearization point first. Since the constraint function b includes the integration of the dynamical system, this is the computational most demanding step. Constraint (4.12d) decouples the current system state ξ_k from the solution of the DAE system and thus it allows to evaluate b without the knowledge of the actual system state. We will refer to this constraint (4.12d) as Initial Value Embedding (IVE). Since it is linear, it is always satisfied after one SQP iteration. IVE allows to evaluate all QP data in advance and as soon as the system state is available, only the QP needs to be solved. Since the computational effort of solving the QP is low in many applications, the feedback time can be reduced drastically.

We split the RTI into a *preparation phase*, a *feedback phase* and a *transition phase*.

Preparation Phase In the preparation phase, all the necessary evaluations of QP (4.16) are done. In particular, the Hessian $A(v_k, \lambda_k)$, the gradient $a(v_k)$, the Jacobian of the constraints $B(v_k)$ and its evaluation $b(v_k)$ are computed. Since the constraints include the evaluation of a dynamical system, this is a computational demanding task. The resulting matrices are sparse and have a diagonal block structure, which originates from the discretization. This structure can be exploited by a preprocessing step in the preparation phase in order to reduce the QP solution time even further.

Condensing [101] is a classical approach to exploit the structure and can be done in the preparation phase.

Feedback Phase In the feedback phase, all QP data is already available. The feedback phase starts as soon as a new system state ξ_k is provided. It completes (4.16) such that the QP can be solved. After the solution is available, the control feedback q_{ξ_k} is submitted to the process. Since the arising QPs originate from subsequent iterates, the consecutive QPs are also related. These similarities can be exploited to efficiently solve the QPs. The so-called online active set strategy is an established method to solve a series of closely related QPs and is implemented in the software package qpOASES [34].

Transition Phase In the transition phase, the iterations variables for the next QP are generated by the solution $(\Delta v, \lambda^{QP}, \mu^{QP})$ of $QP(v_k, \lambda_k, \xi_k)$. In order to achieve good performance, it is desirable to initialize $(v_{k+1}, \lambda_{k+1}, \xi_{k+1})$ as close as possible to the solution of $NLP(\xi_{k+1})$. The two main strategies are shifting and warm start. The shifting strategy builds upon the assumption that the trajectory of the process will be close to the predicted one. The basic idea is to shift all primal variables by one time interval. Details can be found for example in [57, Chapter 4]. However, we will use primarily the warm start strategy because it shows superior performance for problems with short prediction horizons. The primal-dual solution $(\Delta v^{QP}, \lambda^{QP}, \mu^{QP})$ is used to update the iteration variables according to

$$v_{k+1} = v_k + \Delta v^{QP}, \quad \lambda_{k+1} = \lambda^{QP}, \quad \mu_{k+1} = \mu^{QP}. \quad (4.20)$$

The temporal sequence of the phases and the communication between process and controller is depicted in Figure 4.4. After the computations of the preparation phase are finished, the controller is idle and waits for the new system state. We refer to this time as *idle time*. The RTI main loop is summarized in Algorithm 4.1.

Extensions of Model Predictive Control

MPC is a general framework for a wide class of control algorithms. The presented RTI scheme is one extension of the basic MPC algorithm, which aims at high sampling rates. However, several extensions of the RTI and basic MLI algorithm were proposed tailored to special situations. For example, in [57] a RTI scheme is presented, which is able to treat integer decision variables. In [85, 82] the idea of advanced step NMPC [117] is incorporated into the RTI. In [38], an RTI scheme is proposed, where only a

Algorithm 4.1: RTI main loop**loop***preparation phase:*evaluate $A(v_k, \lambda_k), a(v_k), B(v_k), b(v_k)$ wait for system state ξ_k *feedback phase:*

solve QP (4.16)

submit $q_{\xi_k} = q_0 + \Delta q_0$ to controller*transition phase:* $v_{k+1} \leftarrow v_k + \Delta v^{QP}, \lambda_{k+1} \leftarrow \lambda^{QP}, \mu_{k+1} \leftarrow \mu^{QP}$ $k \leftarrow k + 1$ **end loop**

fraction of the linearizations are reevaluated. An MPC scheme for dynamical systems with periodic solutions is presented in [46]. In Chapter 5, we will consider the MLI, which is an extension of the RTI to reduce the computation time.

4.6.1 Stability of MPC and RTI

The theoretical analysis of MPC schemes defines properties and conditions, which reflect a desirable behavior of the controlled process. There is a wide range of research results available that distinct between tracking and economic MPC and set different conditions on the objective function and constraints. In the scope of this thesis, we restrict ourselves to a brief overview over the most important terms and we follow the lines of [43].

Theoretical analysis usually considers nominal MPC schemes, i.e. the current system state is instantly known and not affected by measurement errors

$$\xi_k = x_k \quad (4.21)$$

for $k \in \mathbb{N}$. The MPC feedback law $\mu(x_k)$ is defined by the solution $v^*(x_k)$ of the discretized open loop OCP $\text{NLP}(x_k)$. In order to investigate the theoretical properties of an MPC scheme, the closed loop behavior of the time-discrete system

$$x_{k+1} = x(t_{k+1}; t_k, x_k, z_k, \mu(x_k)) \quad (4.22)$$

is analyzed with an initial state $x_0 \in \mathbb{R}^{n_x}$. The algebraic state $z_k \in \mathbb{R}^{n_z}$ is implicitly defined by the algebraic consistency constraint $0 = g(x_k, z_k, \mu(x_k))$. There are two important properties of an MPC scheme: the *recursive feasibility* and the *stability*. One of the major advantages of MPC is that it is able to respect operational constraints. Therefore, the state and control trajectories of the closed loop system need to satisfy the operational constraints $x_k \in \mathcal{X}_\mu$ and $\mu(x_k) \in \mathcal{U}$, where

$$\begin{aligned} \mathcal{X}_\mu &= \left\{ x \in \mathbb{R}^{n_x} \mid x^{\text{lo}} \leq x \leq x^{\text{up}} \wedge \exists z \in \mathbb{R}^{n_z} \text{ with } z^{\text{lo}} \leq z \leq z^{\text{up}}, 0 = g(x, z, \mu(x)) \right\}, \\ \mathcal{U} &= \left\{ u \in \mathbb{R}^{n_u} \mid u^{\text{lo}} \leq u \leq u^{\text{up}} \right\}. \end{aligned} \quad (4.23)$$

Moreover, the stronger *feasibility* property

$$x_k \in \mathcal{X}_N = \{x \in \mathbb{R}^{n_x} \mid \text{NLP}(x) \text{ is feasible}\} \quad (4.24)$$

needs to be satisfied in order to ensure that the scheme is well-defined. The usual way to address the feasibility is by showing that the scheme recursive feasible. This property requires the existence of a set $A \subseteq \mathcal{X}_\mu$ such that

- for all $x_0 \in A$ the problem $\text{NLP}(x_0)$ is feasible and
- for all $x_0 \in A$ the next iterate is $x(t_1; t_0, x_0, z_0, \mu(x_0)) \in A$.

The recursive feasibility ensures that the MPC scheme is always able to keep the system inside the operational bounds. It is a minimal requirement for the safe application of MPC schemes in practice.

A MPC scheme is referred to as stable, if the closed loop dynamics (4.22)

- are asymptotically stable and
- converge to a predefined equilibrium $x^* \in \mathcal{X}_\mu$

for all initial values x_0 in some set \mathcal{S} . The equilibrium $x^* \in \mathcal{X}_\mu$ is usually the desired operating point of the system. Stability implies that a trajectory x_k , which started close to the equilibrium x^* , stays close to x^* for all $k \in \mathbb{N}$. In practical applications, this means that the MPC controller is able to steer the system back to the desired operating point x^* after a disturbance. This concept of stability requires a constant desired operating point x^* , however it can be extended to time varying reference trajectories [68].

In order to prove the recursive feasibility and stability of a given MPC scheme, it is necessary that $\text{NLP}(x)$ includes a terminal cost term in the objective function and

has a terminal constraint. This allows the definition of an optimal value function, which refers to the cost of an OCP with infinite prediction horizon. Using this optimal value function, the existence of a Lyapunov function can be shown, which is a sufficient condition for asymptotic stability. A comprehensive discussion of the stability of nominal MPC can be found in [94, 44].

However, the presented RTI is not a nominal MPC scheme. It solves the underlying OCPs not until convergence and therefore the optimizer is not able to provide optimal feedback. In order to extend the stability proof from nominal MPC to the RTI, the convergence theory of Newton-type methods needs to be considered. In [27, 25], the Local Contraction Theorem [13] is used to show the stability of an MPC scheme with equality constrained OCPs with a terminal constraint. In [116], the proof is extended to a wider class of OCPs and to arbitrary optimization methods with a Q-linear convergence rate.

Chapter 5

Multi-Level Iteration

The Real-Time Iteration (RTI) presented in Chapter 4 is an established strategy to increase the sampling rate and to decrease the feedback delay of a conventional Nonlinear Model Predictive Control (NMPC) algorithm. However, the RTI still requires the integration of a Differential Algebraic Equation (DAE) system and the corresponding sensitivity generation in every iteration. The standard version of RTI algorithm utilizes the Hessian of the Lagrange function of the underlying optimization problem and therefore even second-order sensitivities are required. These are computationally expensive operations for nonlinear differential equations and they may render the RTI too slow for processes with stiff dynamics.

The Multi-Level Iteration (MLI) is an extension of the RTI, which uses only partial updates instead of full evaluations in every iteration. It is based on the fact that Newton-type methods, such as the Sequential Quadratic Programming (SQP) method described in Chapter 4, does not require the exact computation of derivatives to achieve local convergence. This can be exploited to avoid the expensive evaluation of the Hessian and the Jacobian in every iteration. The core of the MLI is a hierarchy of update formulas, which can be operated individually as separate NMPC algorithms. However, it is also possible to assemble multiple update formulas to holistic schemes. We will refer to the individual update formulas as *levels*.

The goal of this chapter is to develop a novel Model Predictive Control (MPC) scheme that is able to meet the requirements of Microgrid (MG) control. This scheme is based on the MLI update formulas, which are introduced in this chapter. In order to utilize the fast feedback rates of the lower levels and the high accuracy model predictions of the higher levels at the same time, it is necessary to operate multiple levels in parallel. We present methods, how the individual levels are combined to holistic schemes and how they communicate among each other. In order to utilize parallelization on multi-core CPUs, we propose a novel scheduling approach. The resulting new holistic MPC scheme adjusts the sequence of applied levels automatically to the available computational resources. In contrast to existing MLI schemes, it is able to stabilize the MG after major disturbances in real-time. In Chapter 6, we show by comprehensive numerical experiments that this new approach outperforms ex-

isting control methods significantly while staying real-time feasible. Furthermore, it offers the possibility to directly take operational bounds into consideration, which is an essential advantage of MPC over existing methods for MG control.

The MLI was initially introduced in [15]. A detailed description of the update formulas can be found in [113, 81] and their convergence properties are analyzed in [15]. A performance study is presented in [52] and a scheme comprising multiple levels in parallel is presented in [69].

5.1 MLI Update Formulas

The RTI is a methodology to solve a sequence of parametric Nonlinear Programming Problems (NLPs)

$$NLP(\xi) = \min_{v \in \mathbb{R}^{n_v}} l(v) \quad (5.1a)$$

$$\text{s.t. } b(v) + E\xi = 0, \quad (5.1b)$$

$$v^{\text{lo}} \leq v \leq v^{\text{up}} \quad (5.1c)$$

in the context of NMPC. The core of the RTI is a parametric Quadratic Program (QP), which is solved in every iteration

$$QP(v, \lambda, \xi) = \min_{\Delta v} \frac{1}{2} \Delta v^\top A(v, \lambda) \Delta v + a(v)^\top \Delta v \quad (5.2a)$$

$$\text{s.t. } B(v) \Delta v + b(v) + E\xi = 0, \quad (5.2b)$$

$$v^{\text{lo}} \leq v + \Delta v \leq v^{\text{up}}. \quad (5.2c)$$

It is a quadratic model of $NLP(\xi)$ as described in Section 4.5. Here $A(v, \lambda)$ is the Hessian of the Lagrange function of the NLP (5.1) or an approximation of it, $a(v)$ is the gradient of the objective function, and the matrix $B(v)$ is the Jacobian of the constraint function $b(v)$. As described in Section 4, this QP is solved in two phases: the preparation phase and the feedback phase. In the preparation phase, the QP data is evaluated and in the feedback phase the actual QP is solved. In many applications, the preparation phase causes the predominant computational load. The MLI is an approach to reduce its computation time by applying only partial updates of the QP data instead of full evaluations. It defines four levels with descending computational complexity, called Level D, C, B, A. Every level represents its own NMPC scheme and can be operated independently of the others. In this section, we describe the indi-

vidual levels and their convergence properties. In Section 5.2, we present multiple approaches to combine the levels and to operate them in parallel.

In order to have a more compact notation, we denote the primal-dual optimization variables of NLP (5.1) by

$$\vartheta = (v, \lambda, \mu) \in \mathbb{R}^{n_v} \times \mathbb{R}^{n_b} \times \mathbb{R}^{2n_v} \quad (5.3)$$

and we collect the data, which assembles QP (5.2), in the tuple

$$\Theta(\vartheta) = (A(v, \lambda), a(v), B(v), b(v)) \in \mathbb{R}^{n_v \times n_v} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_b \times n_v} \times \mathbb{R}^{n_b}. \quad (5.4)$$

5.1.1 The MLI Methodology

The fundamental idea of the MLI is to avoid the evaluation of the QP data $\Theta(\vartheta)$ in every iteration. Instead, every Level $X \in \{D, C, B, A\}$ uses approximations of the QP data given by an update formula $\Theta^X(\vartheta)$. These update formulas use reference values for the primal-dual iteration variables

$$\bar{\vartheta}^X = (\bar{v}^X, \bar{\lambda}^X, \bar{\mu}^X) \quad (5.5)$$

and for the QP data

$$\bar{\Theta}^X = (\bar{A}^X, \bar{a}^X, \bar{B}^X, \bar{b}^X) \quad (5.6)$$

to generate the approximate QP data $\Theta^X(\vartheta)$. These reference values are determined in advance and they stay fixed during the complete simulation. In every iteration, the Feedback-generating Quadratic Programm (FQP)

$$\text{QP}^X(\vartheta, \xi; \Theta^X(\vartheta)) = \min_{\Delta v} \frac{1}{2} \Delta v^\top A^X(\vartheta) \Delta v + (a^X(\vartheta))^\top \Delta v \quad (5.7a)$$

$$\text{s.t.} \quad B^X(\vartheta) \Delta v + b^X(\vartheta) + E\xi = 0, \quad (5.7b)$$

$$v^{\text{lo}} \leq v + \Delta v \leq v^{\text{up}} \quad (5.7c)$$

is solved, where the approximate QP data

$$\Theta^X(\vartheta) = (A^X(\vartheta), a^X(\vartheta), B^X(\vartheta), b^X(\vartheta)) \quad (5.8)$$

is given by level specific update formulas. We will define these update formulas in the following sections. The pseudocode of the MLI main loop for a fixed Level X is

depicted in Algorithm 5.1. For ease of notation, we will omit the superscript of the reference values if the corresponding level is clear from the context.

Algorithm 5.1: Main loop for Level X

Input: reference values $\bar{\vartheta}$ and $\bar{\Theta}$

$\vartheta_k \leftarrow \bar{\vartheta}$

loop

preparation phase:

evaluate $\Theta^X(\vartheta_k; \bar{\Theta}^X, \bar{\vartheta}^X)$

wait for system state ξ_k

feedback phase:

solve $QP^X(\vartheta_k, \xi_k; \Theta^X(\vartheta_k))$ with primal-dual solution $\Delta\vartheta$

submit $q_{\xi_k} = q_0 + \Delta q_0$ to process

transition phase:

update iteration variables ϑ_k with $\Delta\vartheta$

$k \leftarrow k + 1$

end loop

The reference values $\bar{\Theta}^X$ and $\bar{\vartheta}^X$ are fixed and the approximation quality of the levels depends on a suitable choice. In general, it is desirable to choose the reference values $\bar{\vartheta}^X$ close to the solution $\vartheta^*(\xi)$ of $NLP(\xi)$ and $\bar{\Theta}^X$ close to the corresponding QP data $\Theta(\vartheta^*(\xi))$. Nevertheless, ξ is not known in advance and therefore choosing suitable reference values is a challenging task. In order to find suitable reference values, multiple levels can be operated in parallel and the solution of the higher levels can be used as reference values for the lower levels. We will discuss this in Section 5.2. In case of tracking MPC, the goal of the controller is to steer the process to a reference state $\bar{\xi}$. In this situation, the QP data $\Theta(\nu^*(\bar{\xi}))$ at the solution $\nu^*(\bar{\xi})$ of $NLP(\bar{\xi})$ is often a suitable initialization.

In the following sections, we will describe the individual update formulas and their convergence behavior. The convergence is analyzed for a fixed system state $\xi_k = \xi_0$ for all $k \in \mathbb{N}$. Table 7.1 summarizes which data is computed in each iteration and how the QP data is updated.

5.1.2 Level D

Level D is the highest level and provides the most accurate, but also most expensive QP data. The update formulas $\Theta^D(\vartheta)$ of Level D corresponds to the RTI

$$A^D(\vartheta) = A(v, \lambda), \quad a^D(\vartheta) = a(v), \quad (5.9a)$$

$$B^D(\vartheta) = B(v), \quad b^D(\vartheta) = b(v). \quad (5.9b)$$

The iteration variables are updated similar to the RTI as well

$$v_{k+1} = v_k + \Delta v^{QP}, \quad \lambda_{k+1} = \lambda^{QP}, \quad \mu_{k+1}^D = \mu^{QP} \quad (5.10)$$

where $(\Delta v^{QP}, \lambda^{QP}, \mu^{QP})$ is the primal-dual solution of $QP^D(\vartheta_k, \xi; \Theta^D(\vartheta_k))$. Level D corresponds to a full SQP step and therefore it inherits its local quadratic convergence. The convergence properties of the RTI are well investigated and are already discussed in Section 4.6.1. In particular, the QP data $\Theta^D(\vartheta)$ corresponds to the exact linearization at the iteration variables ϑ and is therefore a suitable candidate for the reference values of the lower levels.

5.1.3 Level C - Optimality Iterations

The preparation phase for Level D includes the full Jacobian evaluation of the constraints. Depending on the definition of the approximate Hessian A , there might be even second order sensitivity generation included. This is by far the most computational expensive part in the preparation phase. *Level C* avoids the full Jacobian evaluation and uses the provided reference matrices instead. In order to achieve convergence to an optimal point, an update formula for the gradient is used

$$a^C(\vartheta) = a(v) + \left(\bar{B}^\top - B(v)^\top \right) \lambda. \quad (5.11)$$

In this formula a directional derivatives of the constraints $B(v)^\top \lambda$ is involved. By using Internal Numerical Differentiation (IND), this can be computed significantly more efficient than the full Jacobian evaluation. A detailed description of IND can be found in [2]. The definition of all update formulas $\Theta^C(\vartheta)$ are

$$A^C(\vartheta) = \bar{A}, \quad a^C(\vartheta) = a(v) + \left(\bar{B}^\top - B(v)^\top \right) \lambda, \quad (5.12a)$$

$$B^C(\vartheta) = \bar{B}, \quad b^C(\vartheta) = b(v). \quad (5.12b)$$

The iteration variables are updated similar to the RTI

$$v_{k+1} = v_k + \Delta v^{QP}, \quad \lambda_{k+1} = \lambda^{QP}, \quad \mu_{k+1} = \mu^{QP}. \quad (5.13)$$

The following theorem motivates why Level C iterations are also referred to as *Optimality Iterations*.

Theorem 5.1 (Optimality Iterations)

If, for a fixed system state ξ_0 , the Level C iterates (v_k, λ_k, μ_k) converge towards a limit (v^*, λ^*, μ^*) , then this limit is a Karush-Kuhn-Tucker (KKT) point of the original NLP(ξ_0).

Proof The proof can be found for example in [15]. □

5.1.4 Level B - Feasibility Iterations

Level B abandons the sensitivity generation completely. In every iteration, only the constraints $b(v)$ and the gradient $a(v)$ are reevaluated. Note that the evaluation of $b(v)$ still includes the integration of the dynamical system and might be computationally expensive. In order to achieve convergence to a feasible point, the gradient is updated according to

$$a^B(\vartheta) = a(v) + \bar{A}(v - \bar{v})\lambda. \quad (5.14)$$

The complete update formulas $\Theta^B(\vartheta)$ are given by

$$A^B(\vartheta) = \bar{A}, \quad a^B(\vartheta) = a(v) + \bar{A}(v - \bar{v})\lambda, \quad (5.15a)$$

$$B^B(\vartheta) = \bar{B}, \quad b^B(\vartheta) = b(v). \quad (5.15b)$$

The following theorem motivates why Level B iterations are also referred to as *Feasibility Iterations*.

Theorem 5.2 (Feasibility Iterations)

If, for a fixed system state ξ_0 , the level B iterates (v_k, λ_k, μ_k) converge towards a limit (v^*, λ^*, μ^*) , then this limit is a KKT point of the disturbed problem

$$\min_v \frac{1}{2}(v - \bar{v})^\top A(v - \bar{v}) + (a + e)^\top v \quad (5.16a)$$

$$\text{s.t. } b(v) + E\xi_0 = 0, \quad (5.16b)$$

$$v^{\text{lo}} \leq v \leq v^{\text{up}}. \quad (5.16c)$$

with $e = (\nabla B(v^*) - B^\top)\lambda^*$.

Proof The proof can be found for example in [15]. □

Theorem 5.2 ensures especially that level B is converging to a feasible point of the original NLP. However, it is not converging to the solution of the original NLP, but to the solution of the disturbed problem (5.16). The distance to the original solution depends on the difference of the Jacobian of the constraints and its approximation.

5.1.5 Level A

In *Level A* no updates of the QP data Θ are performed at all. Instead, the data is fixed to the reference values. Hence, Level A refers to linear MPC with a fixed linear model of the DAE system. It provides feedback with the lowest computational effort, but for nonlinear models convergence can not be guaranteed. The QP data $\Theta^A(\vartheta)$ is fixed to

$$A^A(\vartheta) = \bar{A}, \quad a^A(\vartheta) = \bar{a}, \quad (5.17a)$$

$$B^A(\vartheta) = \bar{B}, \quad b^A(\vartheta) = \bar{b}. \quad (5.17b)$$

In contrast to the other levels, even the constraint function b is not reevaluated. Therefore, the iteration variables are fixed too and only the solution of the QP is used as an control update

$$q_{\xi_k} = \bar{q}_0 + \Delta q_0^{QP}, \quad (5.18)$$

where Δq_0^{QP} is extracted from Δv^{QP} . Since all QP data is fixed for level A, the solution process reduces to solving a linear equation system, if the active set does not change. In this case, only the right hand side is changing because of new measurements. This allows to reuse factorizations and to increase the feedback rate even further [113].

5.1.6 Convergence Properties

Level D and C can be interpreted as Newton-type methods for Problem (4.14) with a fixed initial value $\xi_k = \xi_0$. Level D refers to classical SQP with local quadratic convergence. Level C uses approximations for the matrices and can be analyzed with local convergence theory for Newton-type methods. Under mild conditions, it is converging locally with a q-superlinear convergence rate. The proof is carried out in [113, 29], using the local contraction theorem [13]. In Section 5.1.4, we showed that if

Level	necessary computations			
	$b(v)$	$a(v)$	$B(v)$	$A(v, \lambda)$
D	✓	✓	✓	✓
C	✓	✓	(✓) ¹	✗
B	✓	✗	✗	✗
A	✗	✗	✗	✗

Level	update formula for QP data			
	b	a	B	A
D	$b(v)$	$a(v)$	$B(v)$	$A(v, \lambda)$
C	$b(v)$	$a(v) + (\bar{B}^\top - B(v)^\top)\lambda$	\bar{B}	\bar{A}
B	$b(v)$	$\bar{a} + \bar{A}(v - \bar{v})$	\bar{B}	\bar{A}
A	\bar{b}	\bar{a}	\bar{B}	\bar{A}

¹ Only the vector-matrix product $\lambda^\top B$ needs to be computed in an adjoint fashion.

Table 5.1: Computations and update formulas for the QP data of the different levels.

Level B converges, it converges to an optimal value of the perturbed problem (5.16) and to a feasible point of the original problem. Level A is using a fixed linear model and is therefore in general not converging.

The stability analysis for constantly changing initial values ξ_k is much more challenging, because the combined system-optimizer dynamics need to be considered. A pure Level D scheme refers to the RTI introduced in Chapter 4. Its stability was proven with a zero end point constraint in [27] and with inequalities in [25]. These proofs are formulated in terms of Newton methods and are therefore also valid for pure Level C iterations [113]. The stability of Level B iterations with a quadratic objective function and with a zero terminal constraint is shown in [115].

5.2 Combined MLI Schemes

The MLI comprises a hierarchy of update formulas with descending computational complexity, but also less strong convergence properties. In applications with nonlinear dynamical systems, the reference values $\bar{\Theta}$ deviate from the actual linearizations and the lower levels give suboptimal feedback. This may lead to an instable behavior of the process. Therefore, it is desirable not to operate one level independently, but multiple in parallel. This allows us to utilize the fast feedback rates of the lower levels and the strong convergence properties of the higher levels at the same time. In order

to assemble combined MLI schemes from the presented levels, we have to consider two mayor aspects:

- The individual levels are operated in parallel as independent units. However, they need to communicate with each other in order to update the corresponding reference values. Thereby the iteration variables, the reference values $\bar{\vartheta}$ and the reference QP data $\bar{\Theta}$ can be transferred individually.
- In every sampling interval, feedback can be generated by only one level. In order to define which level is applied a scheduling algorithm is needed.

5.2.1 Communication Architecture

When we want to combine multiple levels in one holistic scheme, we need to utilize parallelization and we sheer off from the strict sequential character of the RTI. Instead, we are dealing with multiple components, which are executed on dedicated CPU cores without affecting the computation time of the other components. The involved components are the process, the Feedback-generating Quadratic Programm (FQP) and the individual levels. The signal flow between these components is depicted in Figure 5.1. The reality is represented by the process. In real world applications, this is the interface to the controlled plant. However, in numerical experiments the process is also represented by a model. In many cases, the same model as in Problem (4.8) is used, but it is also possible to use a more detailed model or to incorporate unforeseen disturbances. The central component, which is responsible for the communication with the process is the FQP. It receives the current system state ξ from the process and then solves QP (5.7). Afterwards, the control feedback q_ξ is submitted to the process. In addition, the FQP communicates with every level, which is involved in the scheme. The involved levels are solely responsible for the evaluation of the corresponding update formulas as described in Sections 5.1. As soon as their computations are finished, they submit the QP data Θ^X to the FQP. Depending on the scheduling scheme, the FQP chooses one of the available levels to set up the next QP. Furthermore, the individual levels are communicating with each other in order to update their reference values and iteration variables.

5.2.2 Communication Schemes

In a combined MLI scheme, the data is reevaluated constantly on every level in parallel with respect to their update formulas. In order to achieve better convergence behavior, it is beneficial to use the recently updated QP data from the higher level

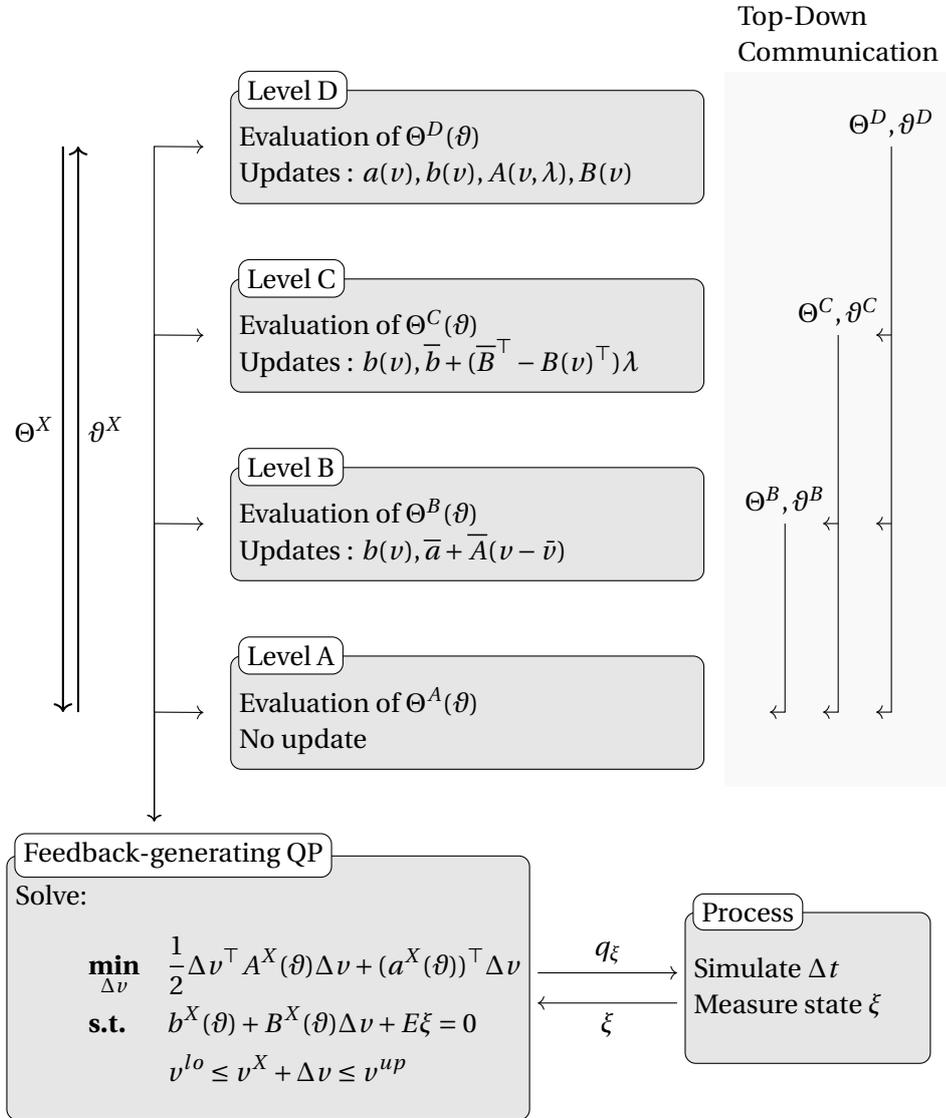


Figure 5.1: Signal flow of MLI scheme with combined levels. The gray boxes represent independent components, which are executed on dedicated CPU cores. The arrows represent the communication signals between the components. In this sketch, a top-down communication for the reference data is used as depicted on the right hand side.

as reference values for the lower levels. This allows to adjust the reference values of the lower levels to the current system state without additional computational effort. The different levels hold two pieces of data, which are communicated separately: the reference QP data $\bar{\Theta}^X$ and the primal-dual iteration variables ϑ^X . While the QP data is primarily communicated from the higher to the lower levels, there are multiple approaches for the iterates. A comprehensive overview of communication schemes are shown in [113].

We will primarily use a top-down communication for QP data as well as the iterates. The QP data Θ^X is always communicated together with the iterates ϑ^X and the communication always triggers a restart of the receiving level. In this approach, the highest level can be interpreted as the main iteration scheme. If the highest level is either Level C or D, it is a fully nonlinear controller, which resets possible suboptimal iteration behavior of Level A or B. However, if there are many intermediate steps between two high level updates, the iterates might be outdated in comparison to the iterates of the lower levels. The approach is summarized in Algorithm 5.2.

Algorithm 5.2: Top-down communication algorithm

switch X :

case D :

$$\begin{aligned} \bar{\Theta}^C &\leftarrow \Theta_k^D, & \bar{\Theta}^B &\leftarrow \Theta_k^D, & \bar{\Theta}^A &\leftarrow \Theta_k^D \\ \bar{\vartheta}^C &\leftarrow \vartheta_k^D, & \bar{\vartheta}^B &\leftarrow \vartheta_k^D, & \bar{\vartheta}^A &\leftarrow \vartheta_k^D \end{aligned}$$

restart level C, B, A

case C :

$$\begin{aligned} \bar{\Theta}^B &\leftarrow \Theta_k^C, & \bar{\Theta}^A &\leftarrow \Theta_k^C \\ \bar{\vartheta}^B &\leftarrow \vartheta_k^C, & \bar{\vartheta}^A &\leftarrow \vartheta_k^C \end{aligned}$$

restart level B, A

case B :

$$\begin{aligned} \bar{\Theta}^A &\leftarrow \Theta_k^B \\ \bar{\vartheta}^A &\leftarrow \vartheta_k^B \end{aligned}$$

restart level A

case A :

no communication

5.2.3 Scheduling Scheme

When operating multiple levels in parallel, a scheduling scheme is needed in order to decide which level is applied next. Multiple approaches have been proposed with different goals. In this section, we will discuss fixed level schemes and Adaptive Level Choice by Contractivity (ALC-Con) and in Section 5.3 we will propose a new approach, which chooses the levels based on computation time.

Remark 5.1

The motivation for combining MLI schemes is to utilize multiple processing units in parallel. We assume that every component is executed on its own dedicated CPU core. The computational load does not affect the computation time of the other components.

Fixed Level Scheme

The most straight forward and also the most common technique for the scheduling of levels is to define an fixed sequence of levels in advance. For every level a pattern is defined, which is repeated continuously. A common notation is for a fixed level scheme is $A^{n_A} B^{n_B} C^{n_C} D^{n_D}$, where the superscript letters refer to the number of sampling intervals after that the corresponding level is scheduled. If two or more levels are scheduled at the same sampling point, the highest available is applied. Levels that does not occur in the scheme are omitted. For example, AD^2 refers to a scheme with a level D iteration every second sampling point and one intermediate level A iteration. The regular RTI is denoted by D^1 and a linear MPC algorithm by A^1 . Since the levels are fixed in advance, it is not clear if the computation time of a level is shorter than its sampling time. In simulations, it is possible to apply level sequences, with a higher computation time than the real-time. We refer to level sequences, where the computation time is shorter than the corresponding sampling time for all participating levels as *real-time feasible*.

In Figure 5.2 is the temporal communication of an exemplary BD^2 scheme depicted. The horizontal lines represent the individual components and the vertical lines represent the signals between them. On every second sample point, a Level D iteration is scheduled followed by an intermediate Level B iteration. In the figure, Level D iterations are scheduled at t_k and t_{k+2} , while Level B provides intermediate feedback at t_{k+1} and t_{k+3} . In addition, a top-down communication for the iteration variables is used. The level D controller always stays on its own set of variables and submits the latest updates to the level B. Level B uses them to compute one intermediate update and receives new iteration variables in the next sampling interval.

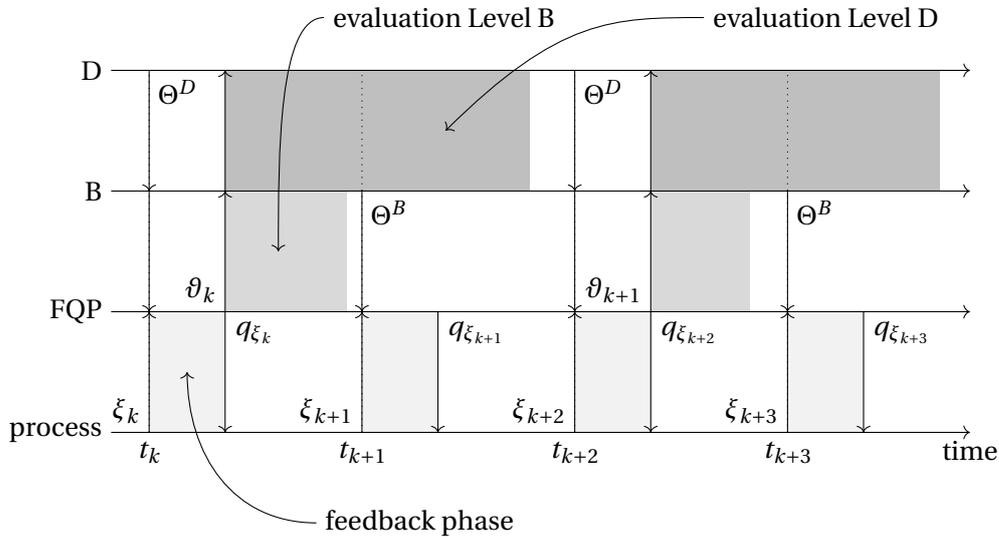


Figure 5.2: Temporal communication of a BD^2 scheme with a Level D iteration every second sampling point and one intermediate Level B iteration. The gray areas indicate when the corresponding level is busy.

An example of a fixed level scheme with AD^2-AD^6 can be found in [4]. Another example of a mixed scheme with level B and D is presented in [69]. And a comprehensive study with different schemes is shown in [52].

Adaptive Level Choice by Contractivity

Fixed level schemes are easy to implement and they lead to good results in many applications, but they are also rather inflexible. Neither the current convergence properties nor the computational load are considered. Adaptive Level Choice by Contractivity (ALC-Con) is an approach to choose the level updates adaptively in order to be computationally as efficient as possible. The main idea is to track the convergence properties in every iteration in order to decide which level is scheduled next. For a fixed system state ξ , Level C can be interpreted as a Quasi-Newton method for the original problem $NLP(\xi)$ and Level B for the modified Problem (5.16). Thus their convergence properties and contraction rate can be analyzed in the sense of the Local Contraction Theorem [13]. The contraction rate δ gives an upper bound of the

contraction between two consecutive iteration steps

$$\|v_k - v_{k-1}\| \leq \delta \|v_{k+1} - v_k\| \quad (5.19)$$

and $\delta < 1$ implies the convergence of the scheme.

The goal of ALC-Con is to apply Level B or C as often as possible. In every step, the contraction rate δ is estimated and only if the estimate is above a predefined bound δ^B or δ^C , a level D iteration is scheduled. Since a Level B iteration requires less computational effort than a Level C iteration, it is the preferred level update. However, it does not converge to the solution of the original problem, but instead to a solution of the modified problem (5.16). Therefore, a low contraction rate does not necessarily imply proximity to the real solution. To counteract this, a Level C iteration is scheduled if the iteration variables are too close to the solution of the modified problem. The norm of the Lagrange gradient of the modified problem is a possible criterion to decide if a Level C iteration is scheduled.

ALC-Con was initially proposed in [113], which includes a detailed description on how to estimate the contraction rate and how the levels are scheduled. In addition, the performance of ALC-Con is demonstrated in a comprehensive numerical study.

5.3 Adaptive Level Choice by Computation Time

ALC-Con, presented in the last section, aims at minimizing the computational load, while guaranteeing a certain feedback quality. However, for many applications the load of the CPU is not the bottleneck. Therefore, Adaptive Level Choice by Computation Time (ALC-Time) aims at giving the best possible feedback for the available computing power. It is especially suitable in situations, where the computational effort of the individual level evaluations is varying strongly between subsequent sampling points and a fixed level scheme is not flexible enough. We proposed this approach initially in [103].

5.3.1 Fixed-step versus Adaptive Integration

The majority of the computational load is caused by the integration of the dynamical system and the sensitivity generation. Most integration methods compute the solution stepwise in time by an integration formula. Independent of the integration formula, there are two approaches for choosing the stepsize: fixed-step and adaptive. *Fixed-step integrators* define a time grid in advance and apply their integration formula for every step. There is no mechanism to ensure a certain accuracy of the solution. In contrast to that, *adaptive integrators* use an integration tolerance. On

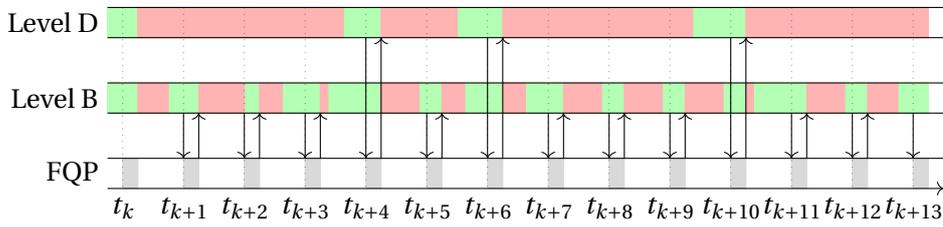


Figure 5.3: Exemplary scheduling of levels with ALC-Time with varying computation times. The colors indicate the state of the levels. Red shows that the corresponding level is busy and green that it is ready. At the sampling points, the highest available level is used to generate feedback. The number of intermediate Level B iterations depend on the computation time of Level D. The grey areas indicate the feedback phase of the FQP.

every step the integration error is estimated and if the error is above the tolerance the grid is refined.

In the context of MLI, the computation time and the predictability of the computation time play a crucial role for the scheduling of the individual levels. Since fixed-step integrators use a predefined time grid, the number of system evaluations is constant for a given time interval. Because one evaluation of the model equations f and g of the system (4.2) requires a constant computational effort, the overall integration time is very predictable and can be determined in numerical experiments. This is a desired property for a fixed-level scheme, because it allows to schedule the updates tightly without compromising the real-time feasibility. However, the disadvantage is that the integrator does not consider the integration quality at all. In order to ensure a certain approximation quality the time grid needs to be sufficiently fine for the complete simulation horizon.

In many applications, the number of required integration steps is varying strongly. In a typical scenario, MPC aims at stabilizing the system at a desired steady state, but the system is in a transient phase after a disturbance. A high number of integration steps is required to achieve a given accuracy. The controller steers the system back to a steady state. While approaching the steady-state, the required number of integration steps to achieve the same accuracy is decreasing significantly. Therefore, a fixed-step integrator may use an unnecessary fine grid in steady-state because the grid needs to be chosen fine enough to also integrate the transient phase accurately. This leads to a conservative scheduling of updates in a fixed level scheme, since the worst-case computation time needs to be treated in order to be real-time feasible.

This problem does not occur for adaptive integration. Because of its error checking, it performs only the necessary number of integration steps to achieve a given accuracy. However, this approach has two major disadvantages for MLI. First, the error checking algorithm requires additional computational effort. In an application, where the sampling time is low, this additional effort might affect the control performance. Furthermore, it is not straight forward to estimate the computation time. E.g., a Level D update in a transient phase may require a multiple of the calculation time of Level D update in steady state. Therefore, it is difficult to define a MLI scheme in advance and conservative scheduling may be necessary.

5.3.2 Time Based Scheduling of Levels

ALC-Time is a methodology to deal with the varying computational effort, while staying efficient. The fundamental idea is to evaluate all levels in parallel and apply at every sampling point the highest level, which finished its evaluations. For this purpose, the levels are equipped with a state $ready \in \{TRUE, FALSE\}$, which represents if the level is ready or busy. The algorithm is initialized with an initial guess ϑ_0 for the primal-dual solution of $NLP(\xi_0)$ and the corresponding QP data $\bar{\Theta}$. In the beginning of the simulation, the evaluations of all levels are triggered with the initial guess. As soon as an evaluation is finished, the corresponding level is marked as ready. When the simulation of the controlled process reaches the next sampling point, the data of the highest level, which is marked as ready, is applied to the QP (5.7). The lower levels are reinitialized with the updated QP data $\bar{\Theta}, \bar{\vartheta}$ as reference values and a new evaluation starts. The pseudocode for the main loop is depicted in Algorithm 5.3 and for the individual levels in Algorithm 5.4.

In the described algorithm, a top-down communication of the iteration variables is used. Here, level A has special purpose. Since it includes no updates of the QP data, its computation time is very low and it can always be used as a fallback. Therefore, it is ensured that always at least one level is ready.

In Figure 5.3 an exemplary scheduling of Level D and B is depicted. The horizontal arrows describe the state of the levels and the vertical arrows represent the communication between them. In the red area the corresponding level is busy and in the green area it is ready. The FQP always uses the highest available level, which is ready. The number of intermediate Level B depends on the computation time of Level D.

ALC-Time allows to use always the most accurate available linearizations whilst having the guarantee of remaining real-time feasible. It is especially tailored to modern multi core CPUs. In comparison to fixed level sequences and ALC-Con, ALC-Time is especially suitable for control applications with high sampling frequencies and pro-

Algorithm 5.3: MLI main loop with ALC-Time

Input: ϑ_0 and $\bar{\Theta}$
 $k \leftarrow 0$
 $\vartheta_k \leftarrow \vartheta_0$
start Algorithm 5.4 for all participating levels with ϑ_0 and $\bar{\Theta}$
loop
 preparation phase:
 wait for system state ξ_k
 choose highest level X with $ready = TRUE$
 feedback phase:
 solve $QP^X(\vartheta_k, \xi_k)$ with solution $(\Delta w, \lambda^{QP}, \mu^{QP})$
 submit $q_{\xi_k} = q_0 + \Delta q_0$ to process
 transition phase:
 submit $(\Delta w, \lambda^{QP}, \mu^{QP})$ to level X
 $k \leftarrow k + 1$
end loop

Algorithm 5.4: Evaluation for Level X

Input: $\bar{\vartheta}$ and $\bar{\Theta}$
 $\vartheta^X \leftarrow \bar{\vartheta}$
 $ready \leftarrow TRUE$
loop
 wait for $(\Delta v, \lambda^{QP}, \mu^{QP})$ from FQP
 $ready \leftarrow FALSE$
 $v^X \leftarrow v^X + \Delta v, \lambda^X \leftarrow \lambda^{QP}, \mu^X \leftarrow \mu^{QP}$
 evaluate Θ^X
 submit data Θ^X and ϑ^X to FQP
 restart all lower levels with ϑ^X and Θ^X as reference values
 $ready \leftarrow TRUE$
end loop

cesses with fast dynamics since it aims at giving the best control performance for the available computation capacity.

ALC-Time is especially beneficial for the control of MGs as described in Chapter 6. The dynamics of a MG are highly nonlinear and stiff and therefore a lot of integration steps are necessary in a transient phase. In contrast to that, the computational effort is low in steady-state and therefore the number of required integration steps changes greatly. In addition, the sampling frequencies are extremely low and the control performance depends on a low feedback delay. ALC-Time allows to react to instantaneous disturbances accurately, while maintaining a high feedback. This will be demonstrated by numerical experiments in Chapter 6.

Chapter 6

Multi-Level Iteration for Secondary Microgrid Control

In this chapter, we present numerical results for the methods presented in Chapter 4 and Chapter 5. First, we will apply the levels of Multi-Level Iteration (MLI) individually to a single generator with a load. Afterwards, we will apply our newly developed Model Predictive Control (MPC) scheme to a more complex electrical network.

6.1 Case Study with One Generator

In this section, we present numerical results for a simple Microgrid (MG). The goal is to investigate the performance of each level as an independent scheme. In particular, we are interested in the required computation time and the possible sampling time. We will compare the performance with a state-of-the-art Proportional-Integral (PI) controller and with the results of an ideal Nonlinear Model Predictive Control (NMPC) controller. The proposed controller is acting on the secondary control level. Therefore, the models include primary controllers and the secondary controller aims at tracking reference values. The results presented in this section are published in [102].

6.1.1 Scenario and Model Description

The considered MG is composed of the different components presented in Chapter 3 and the interconnecting power flow equations. It comprises a Diesel Generator (DG), a Photovoltaic (PV) system and a passive PQ-load. The DG consists of a

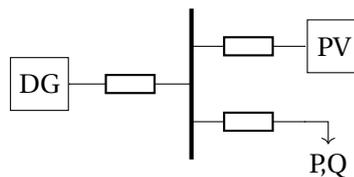


Figure 6.1: Sketch of the test grid with diesel generator, PV and load.

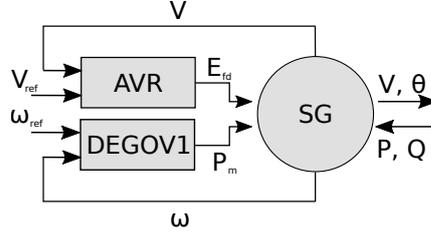


Figure 6.2: Diesel generator with primary controllers.

Synchronous Generator (SG), actuated by a diesel engine. The apparent power base value is $S_{\text{base}} = 100 \text{ MVA}$ and the nominal voltage is $V_{\text{base}} = 400 \text{ V}$. In total, the model has 21 differential and 20 algebraic states and 2 controls. The algebraic states in this model originate from the algebraic power flow equations, as well as from algebraic states in the SG model. The topology of the test grid is depicted in Figure 6.1.

The speed of the DG is controlled by a proportional controller. Both the diesel engine and the speed controller are modeled with the standard IEEE DEGOV1 model. For voltage control, an Automatic Voltage Regulator (AVR) is included, which follows a proportional feedback law. It is modeled with the standard IEEE AC5A model. The setpoints of the primary controllers for frequency ω_{ref} and voltage V_{ref} serve as control variables of the NMPC controller. Details on the models can be found in [84, 74]. Figure 6.2 shows the signal flow of the DG with the primary controllers.

The goal of MG control is to steer the frequency $\omega(t)$ and the voltage $V(t)$ to the nominal value 1 p.u. at the load and prevent peaks that violate the operational limits of $\pm 10\%$ voltage and frequency deviation from the nominal value 1 p.u. or 50 Hz. To achieve this, we use a Lagrange type objective function

$$\Phi(x(\cdot), z(\cdot), u(\cdot)) = \int_{t_k}^{t_k+T} \|\omega(\tau) - 1\|^2 + \|V(\tau) - 1\|^2 d\tau \quad (6.1)$$

with the prediction horizon $T \in \mathbb{R}$. To simulate the intermittent behavior of the PV, we consider a sudden decrease of the PV production from 100% to 5%, lasting 20 s, which corresponds to a cloud passing over the PV plant. During this period, the generator needs to compensate the active power shortage. The PV does not contribute to reactive load sharing. The load profile is depicted in Figure 6.3. Overall, the simulation has a length of 30 s.

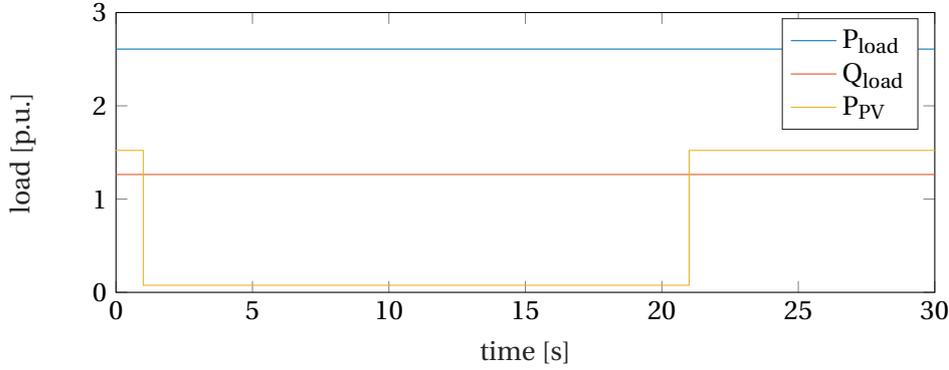


Figure 6.3: Active P_{load} and reactive power Q_{load} demand at the load and active power input P_{PV} of PV system.

6.1.2 Numerical Results

The resulting Optimal Control Problem (OCP) is discretized with $N = 2$ multiple shooting intervals and the prediction horizon is fixed at $T = 2$ s. The length of the first shooting interval corresponds to the sampling time and the second to the rest of the prediction horizon. The numerical simulations are carried out with the NMPC framework MLI [113], written in MATLAB. For integration and sensitivity generation, the SolvIND integrator [2] suite is used, which implements an adaptive BDF method. The Quadratic Programs (QPs) are solved by qpOASES [34], which uses an online active set strategy in order to utilize the similarities between subsequent QPs. In this section, we use the exact Hessian of the Lagrange function.

We compare our proposed MLI-controller with a typical state-of-the-art control setup for small MGs: The DG is equipped with an integral controller for steady-state error elimination of the frequency. It follows an integral control law

$$\omega^{\text{ref}}(t) = K \int_0^t \omega(\tau) - 1 \, d\tau$$

with a parameter $K \in \mathbb{R}$, which is tuned to have a settling time of approximately 20 s. The frequency setpoint is updated every 500 ms. The voltage setpoint V^{ref} for the AVR is kept constant for the complete simulation time. In Figure 6.4 and 6.5 the trajectory of the integral controller is depicted in blue.

In Figure 6.4 the performance of the integral controller is compared to an ideal MPC controller and a Real-Time Iteration (RTI) controller. The ideal MPC controller solves in every iteration the corresponding OCP until convergence using the popular solver

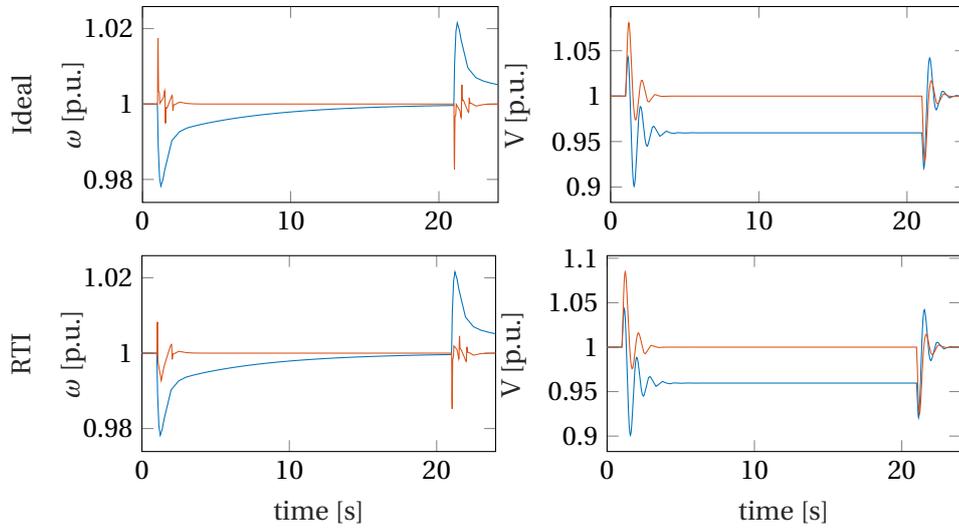


Figure 6.4: Control performance of RTI and ideal MPC.

IPOPT [112]. It allows us to compute reference values on a sampling grid with 500 ms, which leads to a significantly lower initial peak and a lower settling time compared to the traditional control setup. In addition, the voltage is stabilized at the desired operating point with a lower initial peak. Although the control performance is significantly better, this scheme is far from real-time feasibility since its maximal computation time is 587 s.

The RTI controller is over 80 times faster, since its maximal computation time is 7 s and the performance is still significantly better than the integral controller and very similar to the ideal MPC controller. However, with an accurate integration, this scheme is still not real-time feasible. To reduce the computation time of Level D below 0.5 s, a fixed step-size integrator on a coarse grid is necessary. But this degrades the performance of the controller to such an extent that the advantage of MPC vanishes almost completely.

To overcome this downside, we propose to use Level C, B or A instead. They are real-time feasible, even with sampling times below 500 ms. Figure 6.5 shows the performance of Level C, B and A in comparison to the traditional control approach. *Level C* uses a sampling time of 200 ms and is able to steer the frequency and voltage to the nominal value without an offset. Since no updates on the sensitivities are used in *Level B*, it is possible to operate the Level B controller with a sampling time of 100 ms. The system settles significantly faster with a lower initial peak, but with a

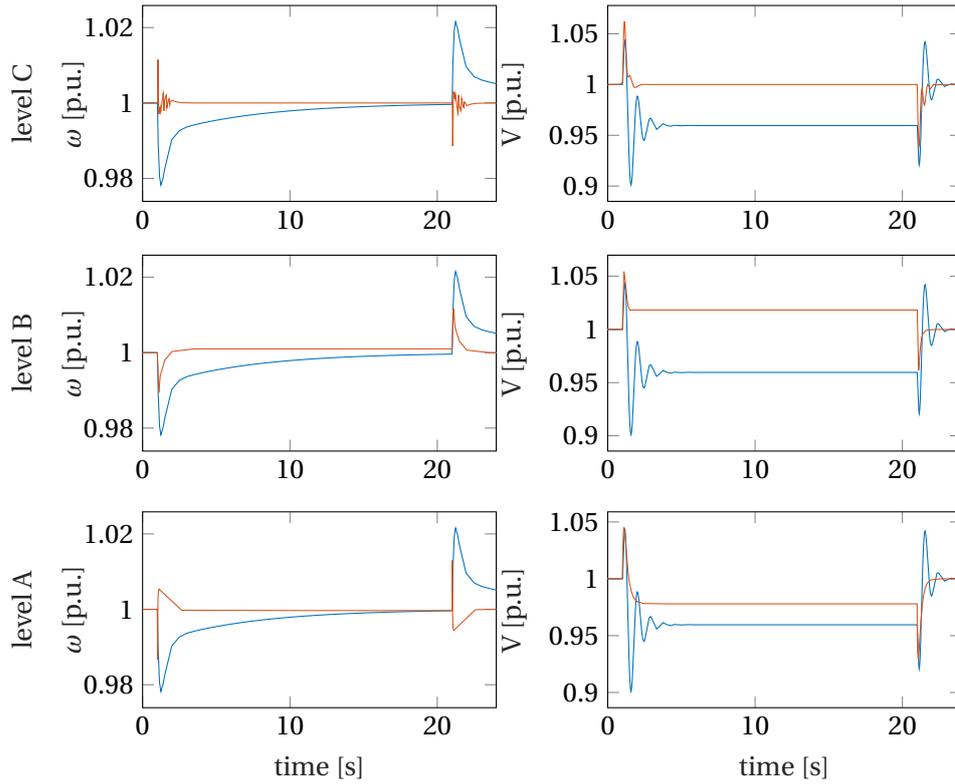


Figure 6.5: Frequency (left) and voltage at the load (right) controlled with Level C, B and A controller. The trajectory of the controllers are depicted in red, the PI-controller for comparison is shown in blue. All schemes are real-time feasible with a sampling rate below 200 ms.

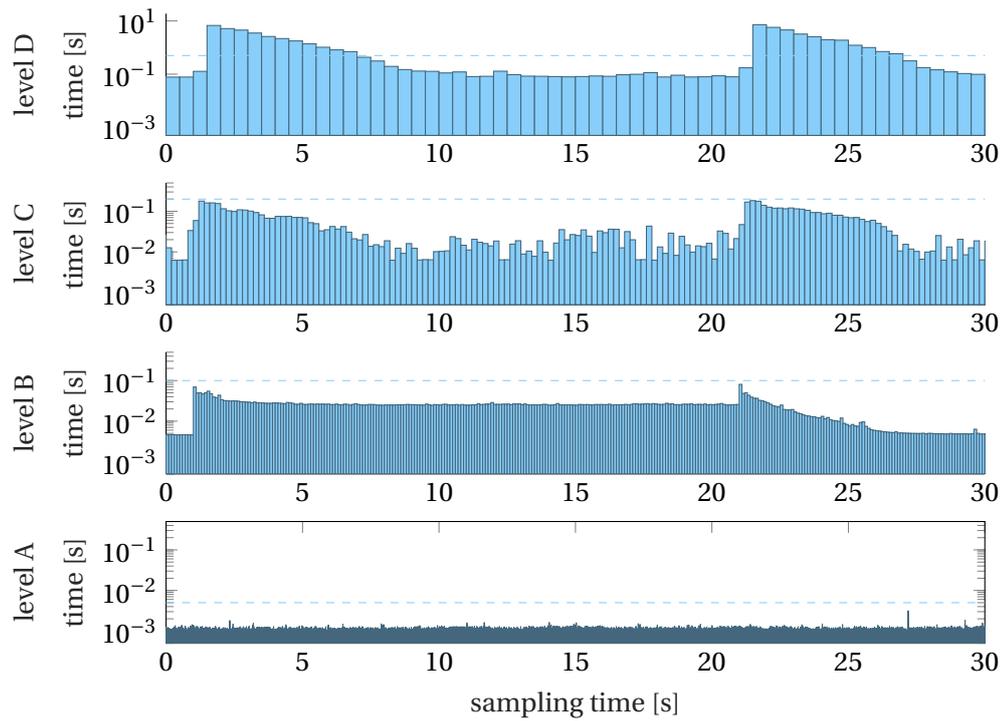


Figure 6.6: Computation time for every iteration for the different levels. The sampling time is represented by the dashed line. For Level A it is 5 ms, for Level B 100 ms and for Level C 200 ms. Level C, B and A are real-time feasible. Level D uses a sampling time of 500 ms but is not real-time feasible.

	sampling time	max loop time
ideal NMPC	500 ms	587 s $\cancel{!}$
RTI	500 ms	7 s $\cancel{!}$
Level C	200 ms	185 ms
Level B	100 ms	80 ms
Level A	5 ms	3.2 ms

Table 6.1: Sampling time and maximal loop time for different schemes.

voltage offset to the nominal value. However, since Level B is guaranteed to converge to a feasible point, the operational limits are satisfied. In *Level A* no integration of the dynamical system is involved and therefore it is possible to reduce the sampling time to 5 ms. These short sampling times allow for a control feedback with the lowest initial peak and the shortest settling time, even though the system is in a slightly suboptimal state during the power shortage. From a theoretical point of view, it is not possible to ensure that the bounds are satisfied, but in this case, the offset is significantly lower than the traditional control approach. The computation time for Level C, B and A is shorter than the sampling time and therefore these methods are real-time feasible. The maximal iteration time for Level A is 3.2 ms, for Level B 80 ms and for Level C 185 ms. The computation times for all schemes are summarized in Table 6.1.

In Figure 6.6, the computation time of the RTI as well as Level C, B and A are depicted over the complete simulation horizon. The top plot shows that the computation times of the RTI are comparatively low in steady state, but increase rapidly as soon as the disturbance occurs. After the disturbance, the system is in a transient phase and the integrator needs to do more integration steps to resolve the dynamics and achieve the given accuracy. During the shortage, the system reaches a new steady state and the computation time decreases. A similar behavior can be observed from Level C and B. Since the integrator needs to do more steps in the transient phase, the computation time increases after the disturbance, but goes down when the system is stabilized. Level A includes no integration of the dynamical system and therefore the computation time does not depend on the current state of the system.

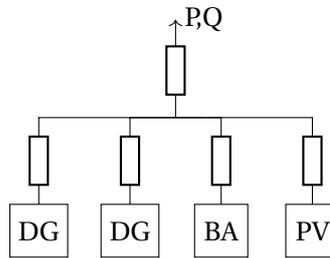


Figure 6.7: Topology of the test MG.

6.2 Grid Control with Adaptive Level Choice by Computation Time

In the previous section, we applied the MLI update formulas to a simple test grid as separate schemes. In this section, we apply MLI to a bigger MG with a realistic topology, which is typical for islanded MGs. The model of this MG is significantly larger and stiffer. Therefore, the individual level controllers are either too computational expensive to be real-time feasible or the levels updates are not accurate enough to stabilize the system. In order to be real-time feasible and to be able to stabilize the system at the desired operating point at the same time, we apply a mixed level scheme with the Adaptive Level Choice by Computation Time (ALC-Time) algorithm, which is presented in Section 5.3.

Contrary to the last section, the MG comprises multiple controllable parts and therefore the load has to be shared between the active components. In particular, this grid includes a battery, which is able to store electrical energy. Since the scheduling of charging and discharging of the batteries requires a long term planning with a prediction horizon of at least 24 hours and the incorporation of a load prediction, this is usually done by the tertiary control level. The target load sharing is then communicated to the secondary control level by reference values for the active and reactive load of the controllable parts. The secondary control level is responsible for tracking these reference values, while maintaining frequency and voltage in the grid. The results presented in this section are published in [103].

6.2.1 Scenario and Model Description

The MG under consideration comprises two identical DGs, a Battery (BA), a PV plant, and a passive PQ-load. Similar to Section 6.1, the MG model is composed by the models for the involved components, which are interconnected by the Power Flow (PF) equations. The DGs are represented by the same model as in Section 6.1.

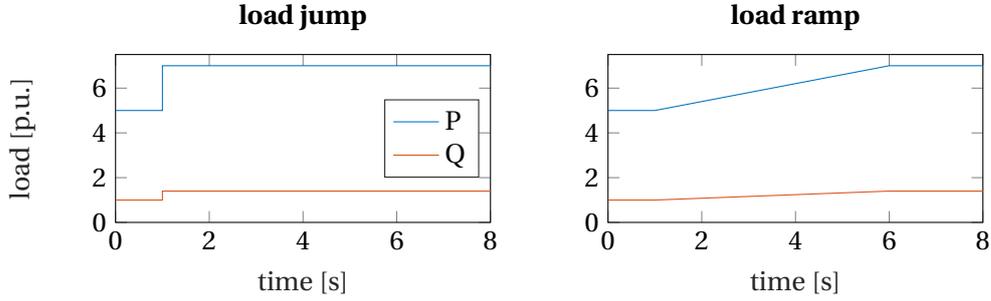


Figure 6.8: Active and reactive power demand P and Q of the load for different load profiles.

These consist of a SG that is actuated by a diesel engine with a governor for frequency stabilization (IEEE DEGOV1) and is equipped with an AVR (IEEE AC5A) for voltage control. The setpoints for frequency ω_{DG}^{ref} and voltage V_{DG}^{ref} serve as control variables of the MLI controller. The battery is modeled as a constant DC voltage source connected to a grid-supporting inverter, like described in Chapter 3. It is controlled by the setpoints for frequency ω_{BA}^{ref} and voltage V_{BA}^{ref} . The base power of the MG is $S_{\text{grid}} = 100\text{kVA}$, the nominal power of the generator is $S_{DG} = 325\text{kVA}$, and the nominal power of the battery is $S_{BA} = 150\text{kVA}$. The complete MG is given as a Differential Algebraic Equation (DAE) system of index 1 with 37 differential and 42 algebraic states and 6 control inputs. Further details on the models and their primary controllers can be found in [84]. Figure 8.1 shows the structure of the MG system.

To demonstrate the capabilities of the proposed controller, we consider two challenging load scenarios: a sudden unscheduled load step and a linear load increase. In both scenarios, the system is in a steady state at the beginning and the reference values for the battery are set to $P_{BA}^{\text{ref}} = Q_{BA}^{\text{ref}} = 0$ p.u.. The generators share the load of $P_{\text{load}} = 5$ p.u. and $Q_{\text{load}} = 1$ p.u. equally. In the first scenario, a sudden unscheduled load step of 40% in active and reactive power takes place after 1 s, which exceeds the capacity of the generators. In the second scenario, the load increases linearly over 5 s. To ensure that the operational limits are satisfied, the battery needs to leave the provided reference values and serve the missing load. The simulation has an overall length of 8 s and the load profiles are depicted in Figure 6.8.

The goal of MG control is primarily to maintain frequency and voltage, i.e. track the nominal values. In addition, the controller aims to track reference values from the tertiary control level in order to share the load between the active components of the

grid. This is modeled by a continuous least squares objective functional

$$\Phi(x(\cdot), z(\cdot), u(\cdot)) = \int_{t_k}^{t_k+T} \|r(x(\tau), z(\tau), u(\tau))\|_{\Sigma}^2 d\tau \quad (6.2)$$

of OCP (4.8) with a weighted norm and a residual function $r(x, z, u)$. The matrix Σ is positive definite and defines the norm by $\|x\|_{\Sigma}^2 = x^{\top} \Sigma x$. The most important goal is to steer the frequency $\omega(t)$ and voltage at the load $V_{\text{load}}(t)$ to the nominal value 1 p.u. after a disturbance. This is achieved by tracking terms

$$r_1(x, z, u) = \omega - 1, \quad r_2(x, z, u) = V_{\text{load}} - 1.$$

During transients, we want to utilize the battery to stabilize frequency and voltage. In steady state, the performance of the battery should follow setpoints $P_{\text{BA}}^{\text{ref}}, Q_{\text{BA}}^{\text{ref}}$ from a higher control level, in order to charge or discharge the battery. The reference values are incorporated in the objective function by tracking terms

$$r_3(x, z, u) = P_{\text{BA}} - P_{\text{BA}}^{\text{ref}}, \quad r_4(x, z, u) = Q_{\text{BA}} - Q_{\text{BA}}^{\text{ref}}.$$

The remaining load is supposed to be shared between the generators equally. We use the terms

$$r_5(x, z, u) = P_1 - P_2, \quad r_6(x, z, u) = Q_1 - Q_2$$

to minimize their difference. Since we have two equally sized generators, this is a convenient way to ensure proper load sharing with only two terms. However, in more complex situations this objective can also be represented by tracking terms for active and reactive power of both generators. The priorities of the different terms are defined by the diagonal elements of the matrix Σ .

The exact Hessian is not only expensive to compute, but it also has disadvantageous numerical properties since it can be indefinite. Therefore, we use the Gauß-Newton approximation of the Hessian as described in Section 4.5.1. Besides its favorable numerical properties, its main advantage is that it relies only on first-order derivatives and we do not have to compute second-order derivatives, which is the most costly task when evaluating QP (4.16). In order to have a standard least squares form of the objective function, we approximate the integral in the objective by a Newton-Cotes formula for numerical integration. Every shooting interval $[\tau_i, \tau_{i+1}]$ is divided into

N sub intervals of equal length

$$h = \frac{\tau_{i+1} - \tau_i}{N} \quad (6.3)$$

and the function is evaluated on the intermediate points $p_j = \tau_i + jh$ for $j = 0, \dots, N$. The integration formula is given by

$$\int_{\tau_i}^{\tau_{i+1}} \|r(x(\tau), z(\tau), u(\tau))\|_{\Sigma}^2 d\tau = (\tau_{i+1} - \tau_i) \sum_{j=0}^N \omega_j \|r(x(p_j), z(p_j), u(p_j))\|_{\Sigma}^2 \quad (6.4)$$

with some weights ω_j . Popular Newton-Cotes formulas are the Trapezoidal rule or the Simpsons's rule [61, Chapter 2].

6.3 Numerical Results

We discretize OCP (4.8) with two multiple shooting intervals and the length of the prediction horizon is fixed to $T = 1$ s. The length of the first shooting interval corresponds to the sampling time of 100 ms and the second to 900 ms. We use the same software stack as in Section 6.1: The numerical simulations are carried out with the NMPC framework MLI [113]. For integration and sensitivity generation, the SolvIND integrator suite is used and the QPs are solved by qpOASES [34].

Similar to Section 6.1, we compare our proposed MLI-controller with a typical state-of-the-art control setup for small MGs: The generators are equipped with an integral controller for steady-state error elimination of the frequency with a settling time of approximately 20 s and a sampling time of 100 ms. The voltage setpoint V^{ref} is kept constant during the full simulation time.

The sampling time is set to 100 ms, since it is the minimal realistic update time. We focus on mixed level schemes with Level D as the main iteration scheme and Level B to generate intermediate feedback. In all experiments, the preparation time of Level B was below 100 ms and it was never necessary to use Level A as a fallback. Since the Gauß-Newton approximation of the Hessian prevents that second-order derivatives have to be evaluated in the preparation phase of Level D, the difference in computation time between Level D and C is low. Therefore, Level C is scheduled very rarely and its impact is low.

Figure 6.9a shows the performance of the proposed MLI controller for the load jump scenario. For comparison, the traditional control approach is shown in blue. The MLI controller steers back the frequency faster with a lower initial drop after the unforeseen disturbance. The voltage gets stabilized faster and the steady state off-

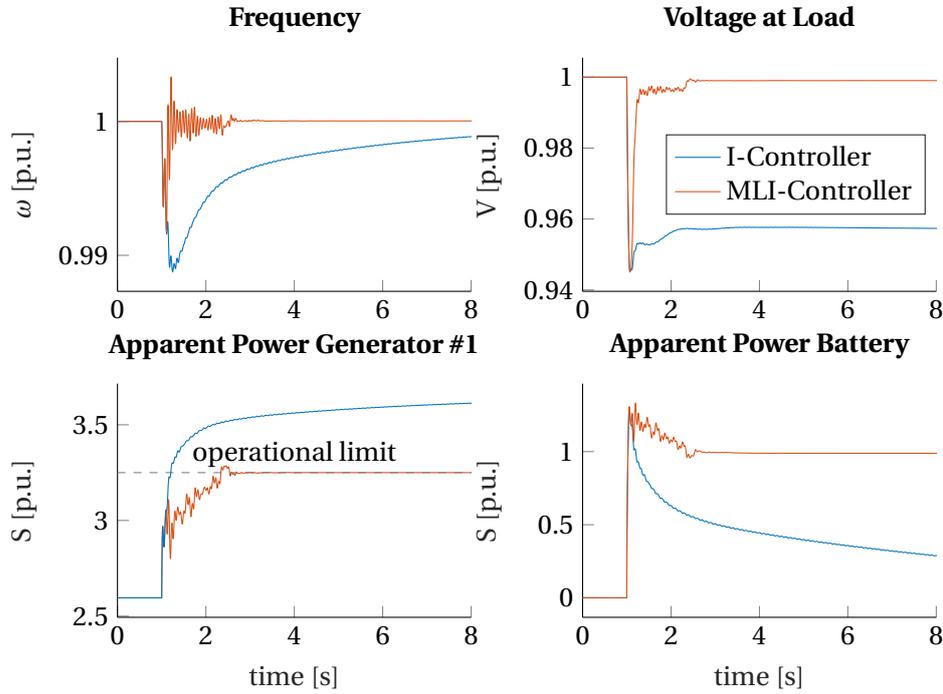
set is eliminated. In the beginning, the battery follows the setpoints and does not contribute to load sharing. After the load drop, the fast reacting dynamics of the battery are used to stabilize the system. Since the overall load exceeds the operational bounds of the generators, the battery temporarily deviates from its reference value and instead serves the necessary additional load. In contrast to this, the integral-controller is not able to obey the operational limits of the generators. If there are no safety measures installed, the generators are overloaded, which may cause physical damage. In Figure 6.9b, the computation time and the scheduling of the different MLI levels are shown. In the beginning, the system is in a steady state and the computation time is short. After the load jump at $t = 1$ s, the system is in a transient phase and the computation time rises sharply, which leads to less Level D evaluations. Afterwards, the system gets steered back to a steady state and the computation time decreases. As the evaluation time for Level B is always below the sampling time, no Level A occurs.

In the load ramp scenario, shown in Figure 6.10, the load gradually increases. Therefore, the initial drop in frequency and voltage is lower. Nevertheless, the MLI controller is able to operate the system closer to the nominal values of frequency and voltage during the transient phase. Since the linear ramp is unscheduled, the control is adjusted on every sampling interval and oscillations are created. After $t = 6$ s, the load is constant again and frequency and voltage are stabilized immediately. In contrast, the I-controller takes more time to steer the frequency back to 1 p.u. and a voltage offset remains. Similar to the load jump scenario, the I-controller is not able to respect the operational bounds.

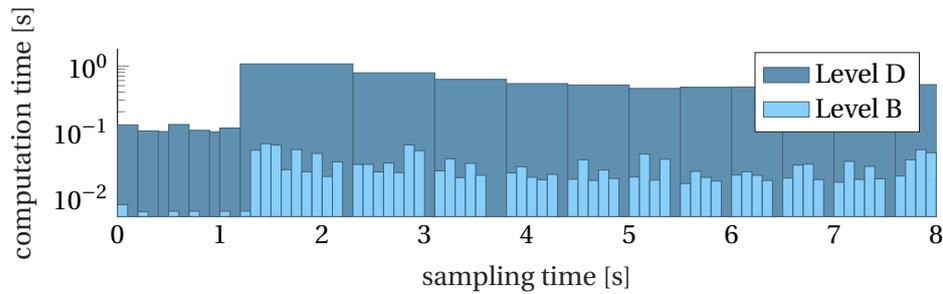
The computation times, depicted in Figure 6.10b, increase after the start of the load ramp $t = 1$ s. In contrast to the load jump scenario, they stay high for longer time, since the system is constantly in a steady state.

Summary

In this chapter, we investigate the capabilities of the MLI on the basis of two exemplary MGs. The simulations on the first example showed that the higher MLI levels are able to achieve a good control performance, but are computationally very expensive. The lower levels enable very low sampling times, but may provide sub-optimal feedback. In the second example, we showed that it is possible to improve the control performance significantly by a mixed level scheme in real-time. With the ALC-Time algorithm, MGs of realistic size can be controlled reliably even in challenging load profiles. Since not all levels are involved in this scheme, it can be assumed that the approach is transferable to even bigger models.

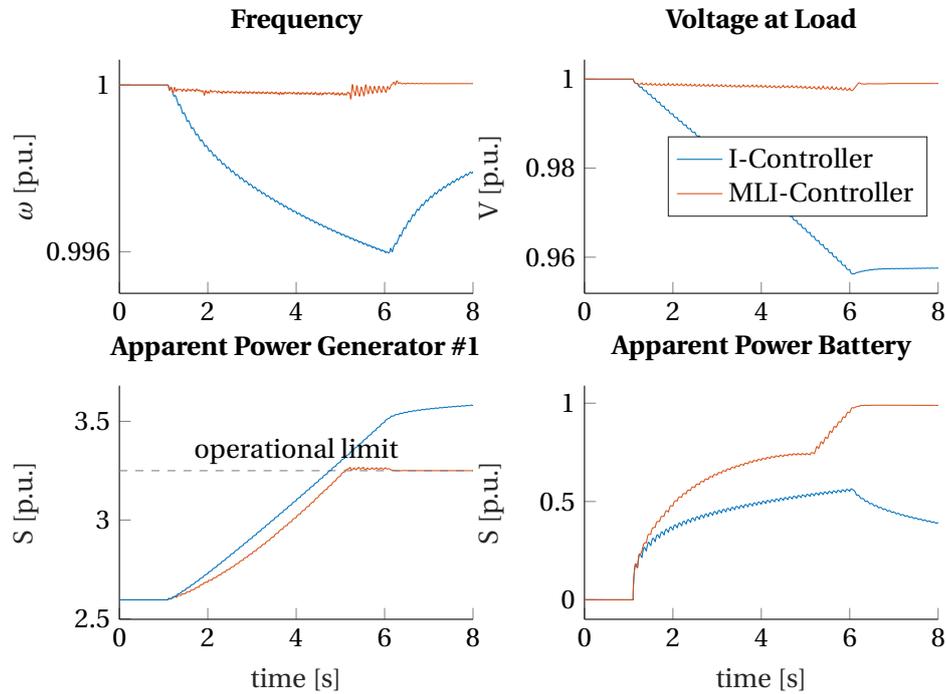


a) Control performance of the MLI-controller in comparison to a traditional control approach. In the top row, the frequency and the voltage at the load is depicted. In the bottom row, the apparent power of the generators and the battery is shown.

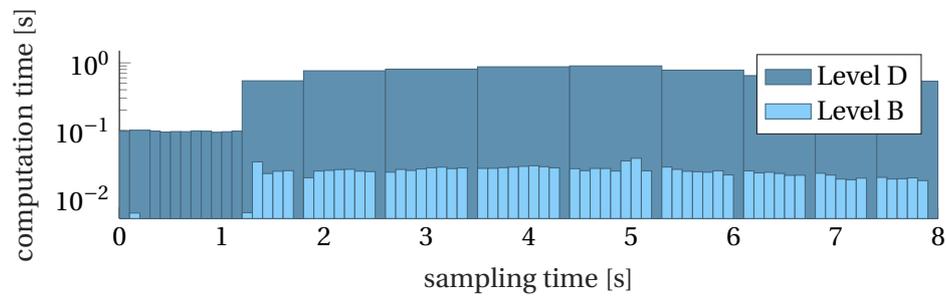


b) Computation times and scheduling of Level B and D. The elapsed computation time is depicted by the height of the bars while the width shows in which sampling intervals the computations were performed.

Figure 6.9: Simulation results for load jump scenario. The MLI-controller is able to steer frequency and voltage back to the nominal value faster and with a lower initial drop, while respecting the operational limits.



a) Control performance of the MLI-controller in comparison to a traditional control approach. In the top row, the frequency and the voltage at the load is depicted. In the bottom row, the apparent power of the generators and the battery is shown.



b) Computation times and scheduling of Level B and D. The elapsed computation time is depicted by the height of the bars while the width shows in which sampling intervals the computations were performed.

Figure 6.10: Simulation results for linear load ramp scenario. The MLI-controller is able to steer frequency and voltage back to the nominal value faster and with a lower initial drop, while respecting the operational limits.

Chapter 7

Moving Horizon Estimation

Many industry applications require fast and accurate state and parameter estimation based on a stream of ongoing measurements. In particular, the performance of feedback control strategies such as Nonlinear Model Predictive Control (NMPC) significantly depends on the quality of the current state estimation. While for unconstrained systems with linear dynamics Kalman Filters [39] are an adequate tool, there exists a variety of methods for nonlinear state estimation. An overview of classical methods can be found in [22]. Many of them are based on particle filters [41] or are extensions of the linear Kalman Filters [8].

Moving Horizon Estimation (MHE), as presented in [92, 93], is an approach that stands out due to its flexibility, as it is capable of handling nonlinear systems even with constraints. In MHE, the estimation process is formulated as an optimization problem defined on a moving time horizon, which includes a limited number of past measurements. This approach is closely related to NMPC, as both methods are based on a flexible optimization framework and solve problems in a moving horizon fashion. The main difference is that NMPC uses a dynamic model to predict the future behavior of the process, whereas MHE uses a similar model to reconstruct the past behavior. Therefore, the approaches can be seen as dual counterparts to each other. NMPC and MHE are usually operated in combination in a closed feedback loop. They are coupled by the current estimation of the system state, which is communicated from the estimator to the NMPC solver.

Even though MHE offers a flexible estimation framework and can outperform Kalman Filter based approaches [50], its main drawback is the computational complexity which arises from solving the underlying optimization problems. However, research has led to major algorithmic improvements, which render these methods employable for an increasing number of applications. In particular, the RTI for MHE (RTI-MHE) as introduced in [28, 59] allows a reduction of the computational complexity from solving a complete optimization problem to performing a single Gauß-Newton step. It uses the algorithmic ideas of the Real-Time Iteration (RTI) for NMPC as introduced Chapter 4 and transfers them to MHE. Nevertheless, the

numerical experiments in Chapter 8 show that the RTI-MHE is not able to provide sufficiently accurate state estimates within the sampling time of Microgrids (MGs).

The Multi-Level Iteration (MLI) as described in Chapter 5 accelerates the RTI even further by exploiting contractivity properties of Newton-type methods. It reduces the number of required function evaluations without sacrificing convergence properties. This enables the application of NMPC for processes with high sampling frequencies. Linear MHE and zero-order MHE [7] are approaches, which apply similar ideas as the MLI to state estimation. They are using fixed linearizations respectively zero-order dynamics updates instead of full Jacobian evaluations in order to reduce the computational effort in each sampling time considerably.

In this chapter, we develop a novel general framework for online parameter and state estimation based on MHE. The fundamental idea is to apply update formulas to the MHE problem similar to the MLI for Model Predictive Control (MPC). This allows the straightforward definition of MHE schemes tailored to a wide range of applications. However, the classical MHE problem depends on constantly changing online data and therefore the MLI update formulas are not directly applicable. We present a new MHE problem formulation, which allows us to decouple linearizations from dynamically changing online data. This reformulation enables us to reuse the computationally expensive linearizations of the system. We define a new hierarchy of update formulas for RTI-MHE, which are adjusted to the special structure of the MHE problem. Furthermore, we describe how to apply the MLI update formulas specifically to the problem parts which are expensive to evaluate, similar to the mixed-level updates introduced in [38]. We prove that the newly introduced levels converge under standard assumptions using the convergence theory of Newton-type methods. In contrast to the classical RTI-MHE, our new framework is able to provide highly accurate state estimates for realistic sized MGs in real time. In Chapter 8, we show its performance in closed and open loop control setups after a major disturbance in the MG. We published the mathematical developments of this chapter and the numerical results presented in Chapter 8 in [47].

7.1 The Principle of MHE

In a MHE setup, the simulation time is divided into a uniform grid of sampling points $0 = t_0 < t_1 < \dots < t_k < \dots$ with a fixed sampling time $\Delta t = t_{k+1} - t_k$. Three different components are interacting with each other: the *process* represents the simulated reality. At every sampling point t_k , control inputs $q_k \in \mathbb{R}^{n_u}$ are received and measurements $\eta_k \in \mathbb{R}^{n_h}$ of the process are carried out.

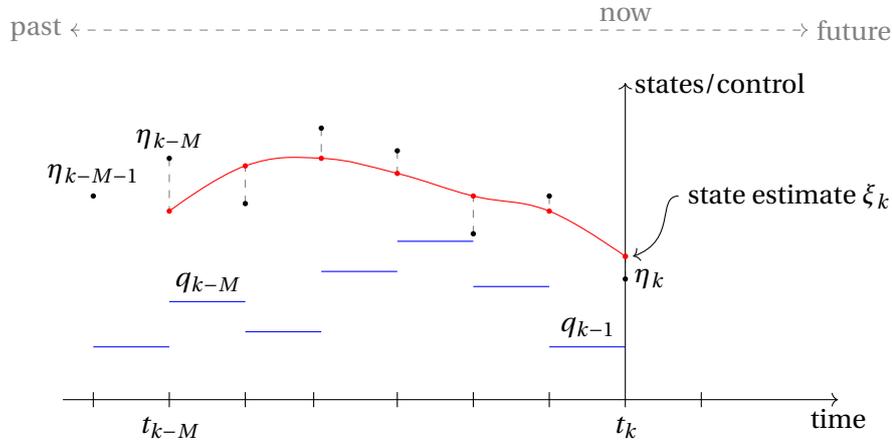


Figure 7.1: Exemplarily MHE process. The measurements are depicted by the black dots, the past controls are shown in blue and the model representation of the process on the horizon $[t_{k-M}, t_k]$ is depicted in red. The state estimate ξ_k is given by the state of the model on the current time point t_k .

The *estimator* uses these measurements to compute an estimate $\xi_k \in \mathbb{R}^{n_x}$ of the current process state. It solves an optimization problem in order to fit a model of the process to the past $M + 1 \in \mathbb{N}$ measurements $\eta_{k-M}, \dots, \eta_k$ and M control inputs q_{k-M}, \dots, q_k . The principle of MHE is depicted in Figure 7.1.

The *controller* is responsible to provide control inputs to the system. There are a lot of different methodologies available to generate these control inputs, like NMPC described in Chapter 4. Typically, there is a distinction between a *closed loop* and an *open loop* setup. In an open loop setup, the controller applies a predefined sequence of controls without consideration of ongoing state measurements. Therefore, the controller is not able to react to unforeseen disturbances. In contrast to that, the *controller* receives a feedback signal in a closed loop setup. It generates controls $q_k \in \mathbb{R}^{n_u}$ based on the current state estimate ξ_k , which is fed back to the process and applied in the interval $[t_k, t_{k+1})$. The signal flow of a closed loop control setup with estimator is depicted in Figure 7.2.

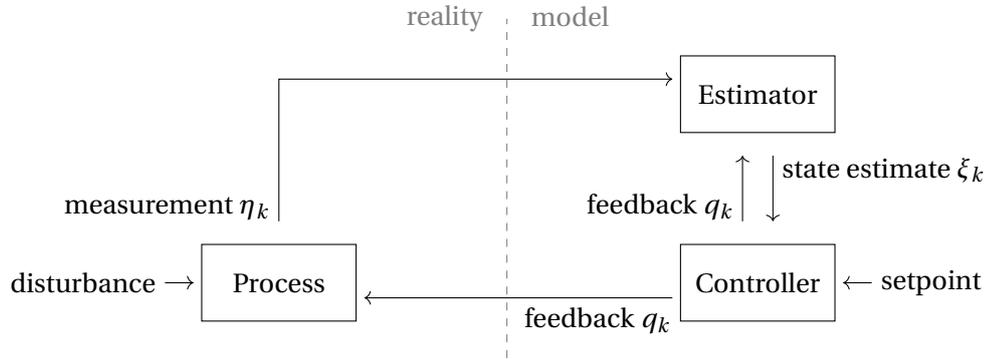


Figure 7.2: Signal flow of closed loop MPC setup with state estimator.

7.1.1 Process Model

Similar to Section 4.2, the model of the process is given as an autonomous Differential Algebraic Equation (DAE) system of differentiation index 1

$$\dot{x}(t) = f(x(t), z(t), u(t)), \quad (7.1a)$$

$$0 = g(x(t), z(t), u(t)), \quad (7.1b)$$

$$x(t_0) = x_0 \quad (7.1c)$$

with differential states $x \in \mathbb{R}^{n_x}$, algebraic states $z \in \mathbb{R}^{n_z}$, and control inputs $u \in \mathbb{R}^{n_u}$. In order to ensure that the solution is well-defined, continuous and all arising derivatives exist, we assume that the right hand side functions $f: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ and $g: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_z}$ are sufficiently smooth. In addition, a smooth measurement function $h: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_h}$ defines the observable quantities of the process, which are measured at every sampling point.

Similar to Chapter 4, we use a relaxed version of the original DAE system as described in Section 4.2.1. We denote the differential and algebraic states of the solution of the relaxed DAE system on the interval $[t_k, t_{k+1}]$ by $x(t; t_k, x_0, z_0, q)$ and $z(t; t_k, x_0, z_0, q)$ with constant controls $q \in \mathbb{R}^{n_u}$ and an initial algebraic state z_0 . In the relaxed DAE system, the algebraic function (7.1b) is replaced by

$$0 = g(x(t), z(t), q) - e^{-\alpha \frac{t-t_k}{t_{k+1}-t_k}} g(x_0, z_0, q) \quad (7.2)$$

with a positive damping factor $\alpha \in \mathbb{R}$. This allows us to treat the initial algebraic states z_0 as independent variables and enables us to use them directly in the objec-

tive function or employ bounds. If we want to evaluate the state of the system at the end of the interval t_{k+1} , we omit the start t_k and end time t_{k+1} in this notation. Since we are dealing with an autonomous DAE on an interval with a fixed length Δt , the solution is independent of t_{k+1} . The assumption that the right hand side functions f and g are smooth, ensures that the evaluation functions x and z are well-defined and smooth too.

7.1.2 MHE Problem Formulation

In the context of MHE, we consider an estimation horizon that includes the last $M + 1$ measurements $\eta_{k-M}, \dots, \eta_k$ taken at times t_{k-M}, \dots, t_k and the last M controls q_{k-M}, \dots, q_{k-1} . This leads to the MHE problem, which we want to solve at the sampling point t_k

$$\min_{v=(s^x, s^z, w)} \frac{1}{2} \|\bar{x}_{k-M} - s_0^x\|_{\Sigma_{\text{arr}, k-M}^{-1}}^2 \quad (7.3a)$$

$$+ \frac{1}{2} \sum_{i=0}^M \|\eta_{k-M+i} - h(s_i^x, s_i^z)\|_{\Sigma_{\text{meas}}^{-1}}^2 \quad (7.3b)$$

$$+ \frac{1}{2} \sum_{i=0}^{M-1} \|w_i\|_{\Sigma_{\text{proc}}^{-1}}^2 \quad (7.3c)$$

s.t.

$$s_{i+1} = x(s_i^x, s_i^z, q_{k-M+i}) + w_i, \quad i = 0, \dots, M-1, \quad (7.3d)$$

$$0 = g(s_i^x, s_i^z, q_{k-M+i}), \quad i = 0, \dots, M. \quad (7.3e)$$

The optimization variables $s_i^x \in \mathbb{R}^{n_x}$ and $s_i^z \in \mathbb{R}^{n_z}$ represent the differential and algebraic variables at the sampling time t_{k-M+i} . The variables $w_i \in \mathbb{R}^{n_x}$ represent *process state noise*. $\|x\|_{\Sigma}^2 = x^\top \Sigma x$ denotes a weighted Euclidean norm with a symmetric, positive definite matrix Σ . The matrices $\Sigma_{\text{arr}, k}$, Σ_{meas} and Σ_{proc} will be discussed in the next section. The estimated state of the process corresponds to the value of the last differential variable $\xi_k = s_M$ at the solution of the Nonlinear Programming Problem (NLP). The variable

$$v = (s^x, s^z, w) \in \mathbb{R}^{(M+1)n_x} \times \mathbb{R}^{(M+1)n_z} \times \mathbb{R}^{Mn_x} \quad (7.4)$$

is a collection of the optimization variables with $s^x = (s_0^x, \dots, s_M^x)$, $s^z = (s_0^z, \dots, s_M^z)$ and $w = (w_0, \dots, w_{M-1})$. The constraints (7.3e) are referred to as *consistency conditions*. They ensure that every feasible point of the NLP corresponds to a solution of the original DAE system (7.1), even though the modified system with the relaxed

algebraic equation (7.2) is used. The *matching conditions* (7.3d) ensure that the process is, up to the process state noise, continuous at the sampling points.

Objective Function

The objective function is composed of multiple maximum-likelihood terms. The fundamental assumption is that measurement errors and plant-model mismatches behave like additive, independent and normally distributed random variables. We derive the maximum-likelihood term in a general way. Given a set of data points $\rho_0, \dots, \rho_n \in \mathbb{R}$, we assume that $\rho_i \sim \mathcal{N}(s_i, \sigma_i^2)$ for a parameter $s_i \in \mathbb{R}$ and a fixed standard deviation $\sigma_i \in \mathbb{R}$ for $i = 1, \dots, n$. We want to find a set of parameters that maximizes the probability of the data points ρ_0, \dots, ρ_n , i.e. we want to maximize the likelihood function. Since we assume the data points as independent, the likelihood function is

$$L(s) = \prod_{i=0}^n p(\rho_i | s_i) = \prod_{i=0}^n \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(\rho_i - s_i)^2}{2\sigma_i^2}\right), \quad (7.5)$$

where $p(\rho_i | s_i)$ is the conditional probability. We exploit the monotonicity of the logarithm to derive the objective function

$$\operatorname{argmax}_s L(s) = \operatorname{argmax}_s \ln(L(s)) \quad (7.6a)$$

$$= \operatorname{argmax}_s - \sum_{i=0}^n \frac{1}{2\sigma_i^2} (\rho_i - s_i)^2 \quad (7.6b)$$

$$= \operatorname{argmin}_s \frac{1}{2} \|\rho - s\|_{\Sigma^{-1}}^2 \quad (7.6c)$$

with a positive definite diagonal matrix $\Sigma = \operatorname{diag}(\sigma_0^2, \dots, \sigma_n^2)$. More details on nonlinear least-squares problems can be found, e.g., in [80].

The three maximum-likelihood terms of the objective function (7.3a) - (7.3c) are motivated by different optimization targets. The term (7.3a) of the objective function is called *arrival cost* and is used to include information from measurements prior to the current estimation horizon. It is assumed that the differential state at the beginning of the estimation horizon s_0^x is normally distributed with expected value \bar{x}_{k-M} and covariance $\Sigma_{\text{arr}, k-M}$. There exist different strategies on how to choose \bar{x}_{k-M} and the corresponding covariance matrix $\Sigma_{\text{arr}, k-M}$. We will cover the update procedure in Section 7.5.

The term (7.3b) aims to minimize the difference between the measurements and the model response. We assume that the measurement errors are normally distributed $\eta_i \sim \mathcal{N}(h(s_i^x, s_i^z), \Sigma_{\text{meas}})$ with a fixed covariance matrix Σ_{meas} .

The process state noise w_i is introduced to take model errors and unforeseen disturbances into consideration. We interpret the process state noise as normally distributed random variables $w_i \sim \mathcal{N}(0, \Sigma_{\text{proc}})$, leading to the corresponding maximum-likelihood term (7.3c).

Remark 7.1

Note that the structure of the MHE problem is very similar to a discretized Optimal Control Problem (OCP) using multiple shooting [14], as presented in Chapter 4. However, length and number of the shooting intervals are in case of MHE not a result of the discretization, but are specified by the experimental setup. The rate of measurements and the constant control inputs are fixed by the primary controller respectively by the controlled hardware.

Remark 7.2

While the covariance matrix for the arrival cost $\Sigma_{\text{arr},k-M}$ is updated every iteration, the covariance matrices for the measurements Σ_{meas} and the process state noise Σ_{proc} needs to be defined in advance. It is part of the modeling process and allows to include prior knowledge about the expected error range of differential states and plant-model mismatches.

Compact Notation of MHE Problem

Overall, the MHE problem solved at time t_k is defined by the online data consisting of the measurements $\boldsymbol{\eta}_k := (\eta_{k-M}, \dots, \eta_k)$, the past control trajectory $\boldsymbol{q}_k := (q_{k-M}, \dots, q_{k-1})$, and the arrival cost state \bar{x}_{k-M} with the corresponding covariance matrix $\Sigma_{\text{arr},k-M}$. We summarize the problem defining data in the tuple

$$y_k := (\boldsymbol{q}_k, \boldsymbol{\eta}_k, \bar{x}_{k-M}, \Sigma_{\text{arr},k-M}). \quad (7.7)$$

This allows us to write NLP (7.3) in the compact form:

$$\text{MHE}(y_k) := \min_{v=(s^x, s^z, w)} \frac{1}{2} \|\ell(v, y_k)\|_{\Sigma_k}^2 \quad (7.8a)$$

$$\text{s.t.} \quad b(v, \boldsymbol{q}_k) = 0. \quad (7.8b)$$

Here, the different residual terms of the objective function (7.3a) – (7.3c) are summarized in the function

$$\ell(v, y_k) = \begin{bmatrix} \bar{x}_{k-M} - s_0^x \\ \eta_{k-M} - h(s_0^x, s_0^z) \\ \vdots \\ \eta_k - h(s_M^x, s_M^z) \\ w \end{bmatrix}. \quad (7.9)$$

The matrix Σ_k is a block diagonal matrix assembled from the covariance matrices $\Sigma_{\text{arr}, k-M}$, Σ_{meas} and Σ_{proc} of the individual least squares terms. The matching conditions (7.3d) together with the algebraic consistency condition (7.3e) are collected in the function b which depends on the optimization variables as well as on the past control trajectory \mathbf{q}_k . We denote the primal solution of MHE(y_k) by $v^*(y_k)$.

Remark 7.3

For notational convenience, the inclusion of simple bounds on the state variables s_i^x, s_i^z is omitted in the MHE formulation. However, the presented framework is extensible to treat simple bounds as well as path constraints. A detailed derivation can be found for example in [113].

7.2 A Generalized Gauß-Newton Method Solution Framework

NLP (7.8) is a constrained nonlinear least-squares problem and can be solved with the Sequential Quadratic Programming (SQP) method described in Section 4.5. However, since it has a least-squares objective function, the generalized Gauß-Newton method is more suitable. In addition to its favorable statistical properties [13], it uses an approximation of the Hessian that contains only first-order derivatives and therefore involves lower computational costs. Similar to the SQP method, a sequence of iterates $(v_j, \lambda_j)_{j \in \mathbb{N}}$ is generated by solving Quadratic Programms (QPs) of the form

$$\text{QP}(v_j, y_k) := \min_{\Delta v} \frac{1}{2} \Delta v^\top A(v_j, y_k) \Delta v + a(v_j, y_k)^\top \Delta v \quad (7.10a)$$

$$\text{s.t. } b(v_j, \mathbf{q}_k) + B(v_j, \mathbf{q}_k) \Delta v = 0. \quad (7.10b)$$

The matrix

$$A(v_j, y_k) = \nabla_v \ell(v_j)^\top \Sigma_k^{-1} \nabla_v \ell(v_j) \quad (7.11)$$

is the Gauß–Newton approximation of the Hessian of the Lagrange function of $\text{MHE}(y_k)$ and $a(v_j, y_k) = \nabla_v \ell(v_j)^\top \Sigma_k^{-1} \ell(v_j, y_k)$ is the gradient of the objective function. Note that the online data y_k enters only linearly in the residual function ℓ and thus $\nabla_v \ell$ is independent of y_k . The Jacobian of the constraints is denoted by $B(v_j, \mathbf{q}_k) = \nabla_v b(v_j, \mathbf{q}_k)^\top$. The iterates are updated by the primal-dual solution $(\Delta v^{\text{QP}}, \lambda^{\text{QP}})$ of $\text{QP}(v_j, y_k)$ by

$$v_{j+1} = v_j + \Delta v^{\text{QP}}, \quad \lambda_{j+1} = \lambda^{\text{QP}}. \quad (7.12)$$

Note that for the regular Gauß-Newton method, the dual variables are not necessary. However, we will need them in Section 7.4, when we introduce MLI updates.

7.2.1 Real-Time Iterations for MHE

The exact primal solution $v^*(y_k)$ of $\text{MHE}(y_k)$ delivers an estimate of the current system state, which is optimal in the sense of the objective function, but the computational demand for solving problem (7.8) might be prohibitively high. Depending on the initialization, many iterations may be necessary to achieve convergence or the method might not converge at all. Therefore, using the exact solution is not feasible for applications with fast dynamics and high sampling frequencies.

The RTI-MHE [58, 28] is an established method to reduce the computational effort drastically. It is an extension of the RTI for NMPC as described in Section 4.6 or [24]. The main idea of RTI-MHE is to interpret $\text{MHE}(y_k)$ as part of a continuously evolving sequence of NLPs, parametrized by the online data y_k . This allows an efficient transition from an approximate solution of $\text{MHE}(y_k)$ to an approximate solution of $\text{MHE}(y_{k+1})$. Given an initial guess v' of the solution of $\text{MHE}(y_{k+1})$ only a single $\text{QP}(v', y_{k+1})$ is solved at time t_{k+1} . The result Δv^{QP} is used as a correction towards the solution of $\text{MHE}(y_{k+1})$ by

$$v_{k+1} := v' + \Delta v^{\text{QP}}. \quad (7.13)$$

The new initial guess for the next iteration is constructed from v_{k+1} by a shifting strategy, which is described below. The quality of the solution and thus state estimate depends on a suitable choice of v' , since the state estimate ξ_k is directly taken from v_{k+1} . Similar to the RTI, the individual iterations are separated in a *preparation*, a *feedback*, and a *transition phase* to reduce the feedback time even further. The main loop of the RTI-MHE is summarized in Algorithm 7.1.

Algorithm 7.1: MHE-RTI main loop**loop***preparation phase:*evaluate $h(s_i^x, s_i^z)$ for $i = 0, \dots, M$ and $\nabla_v \ell(v'), B(v', q_k), b(v', q_k)$ wait for state measurement η_k to complete y_k *feedback phase:*evaluate $A(v', y_k), a(v', y_k)$ solve QP(v', y_k)submit $\xi_k = s_M^x + \Delta s_M^x$ to controller*transition phase:*wait for applied control q_k from controller to complete q_{k+1} $v' = P_{\text{shift}}(v' + \Delta v^{QP}, q_k)$ $k \leftarrow k + 1$ **end loop**

Preparation Phase The essential new data contained in y_{k+1} is the measurement η_{k+1} , which is available at time t_{k+1} . The rest of the online data is already available at time t_k , contained in y_k in a shifted fashion. As η_{k+1} enters only linearly in the function ℓ , most of the QP data can be evaluated in advance. The MHE *preparation phase* comprises the preparation process for QP(v', y_{k+1}), mainly consisting of the evaluation of the constraint function b and its Jacobian. Since it includes the integration of a DAE system, it requires the majority of the computational effort. The measurement function $h(s_i^x, s_i^z)$ and $\nabla_v \ell(v')$ is also evaluated, but since no integration is involved its computational effort is low. The preparation phase is started right after q_k is determined.

Feedback Phase The MHE *feedback phase* starts as soon as the measurement η_{k+1} is available. It includes the evaluation of the objective function residual and the solution of QP(v', y_{k+1}). The result Δv is used to update the primal iteration variables by

$$v_{k+1} = v' + \Delta v \quad (7.14)$$

and the state estimate ξ_{k+1} is directly taken from v_{k+1} . The computational effort of the feedback phase is usually significantly lower than the preparation phase, because no integration of the dynamical system is involved. Thereby, the time delay between

the arrival of new measurements η_{k+1} and the computation of the estimate ξ_{k+1} is greatly reduced.

Transition Phase The linearization point v' needs to be sufficiently close to the solution of $\text{MHE}(y_{k+1})$ in order to get an accurate approximation of $v^*(y_{k+1})$. Therefore, the performance of the RTI-MHE depends on finding a good initialization. This is achieved by a shifting strategy. Since the online data y_{k+1} mainly consists of the online data y_k shifted by one sampling interval and the time horizon of $\text{MHE}(y_{k+1})$ is the shifted horizon from problem $\text{MHE}(y_k)$, it is a canonical choice to initialize the primal variable v' of problem $\text{MHE}(y_{k+1})$ by forward shifting the approximate solution v_k of problem $\text{MHE}(y_k)$. The new differential and algebraic states, which are entering the horizon, are generated by a forward integration using the control feedback q_k . The shift map $v' = P_{\text{shift}}(v_k, q_k)$ is defined by

$$\begin{aligned} s_i'^x &:= \begin{cases} s_{i+1}^x & \text{for } i \leq M-1, \\ x(s_{M-1}'^x, s_{M-1}'^z, q_k) & \text{for } i = M, \end{cases} \\ s_i'^z &:= \begin{cases} s_{i+1}^z & \text{for } i \leq M-1, \\ z(s_{M-1}'^x, s_{M-1}'^z, q_k) & \text{for } i = M, \end{cases} \\ w_i' &:= \begin{cases} w_{i+1} & \text{for } j \leq M-2, \\ 0 & \text{for } i = M-1. \end{cases} \end{aligned} \quad (7.15)$$

The new process state noise variable w_{M-1} , which enters at the end of the horizon, is initialized with 0 since we assume that the model is a proper representation of the controlled plant and therefore that the plant-model mismatch is low. The temporal communication between the state estimator, the controller, and the controlled process is depicted in Figure 7.3.

7.3 Modified RTI-MHE Formulation

Even though the RTI-MHE allows to reduce the computational effort greatly, it has the same drawback as the regular RTI. In every iteration, the complete QP needs to be evaluated, which includes elaborate integration and sensitivity generation. This renders the RTI-MHE too slow for systems with fast dynamics and high sampling frequencies. In Chapter 8 we show that the RTI-MHE is not able to provide sufficiently accurate state estimates within the sampling time of MGs.

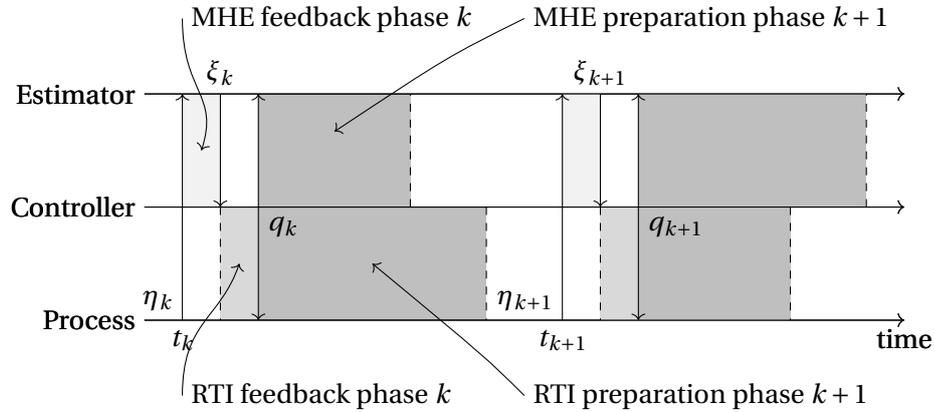


Figure 7.3: Temporal communication of process, estimator, and controller in the RTI-MHE algorithm. The MHE preparation phase starts at the end of the RTI feedback phase, when the applied control is available.

In this chapter, we develop a novel framework for online state and parameter estimation, which is able to provide highly accurate estimates in short sampling intervals. We define a new set of update formulas, which reduce the computational effort of the individual RTI-MHE iterations greatly. As we show in Chapter 8, they are necessary to provide state estimates in real time. The fundamental idea is to reuse linearizations and only apply MLI update formulas in every iteration, similar to MLI for NMPC. The principle of MLI and the individual update formulas are explained in detail in Chapter 5. However, the RTI-MHE algorithm is not flexible enough to allow the application of MLI update formulas directly. This has two major reasons:

- The different levels rely on reference values \bar{B} for the constraint Jacobian $B(v_k, \mathbf{q}_k)$, which is dependent on sequence of past controls \mathbf{q}_k . The controls \mathbf{q}_{k+l} for $l \geq 1$ correspond to past controls \mathbf{q}_k in a shifted fashion together with the newly applied controls q_k, \dots, q_{k+l-1} . Since the distance between \mathbf{q}_{k+l} and \mathbf{q}_k can be large, the reference value \bar{B} may be outdated already after one iteration. In order to make the MLI update formulas compatible, we need to ensure that the linearizations are reusable over multiple sampling intervals.
- When we operate multiple levels in parallel, it is necessary to start the preparation of $\text{QP}(v_{k+l}, y_{k+l})$ already at time t_k . However, y_{k+l} is not determined yet and the RTI-MHE is not able to incorporate future control actions. To apply MLI update formulas in parallel fashion, a methodology is required to prepare

QP(v_{k+l}, y_{k+l}) over multiple intervals based on a prediction of the future control actions.

To address these issues, we propose the following modifications of the RTI-MHE:

- The computationally expensive QP data, i.e. the constraint function b and its Jacobian B , is decoupled from the constantly changing online data by the introduction of auxiliary control variables and a Control Trajectory Embedding (CTE) constraint.
- A generalized shift map is introduced in order to shift optimization variables over multiple intervals.
- Computations are moved from the preparation phase to feedback phase in order to enable the algorithm to react to changes in the online data.

We will discuss these modifications in this section. The resulting MHE problem will be equivalent to the original one, but it will be treatable by MLI update formulas. The application of MLI to the modified RTI-MHE will be discussed in the Section 7.4.

7.3.1 Control Trajectory Embedding (CTE)

The RTI decouples the system dynamics from the constantly changing online data by the introduction of the Initial Value Embedding (IVE), as described in Chapter 4. This allows not only to separate the computations in a preparation and a feedback phase, but also the application of MLI update formulas. To apply MLI update formulas to MHE, we transfer the idea of IVE to the RTI-MHE. In the classical RTI-MHE formulation (7.8), the critical online data are the past controls $\mathbf{q}_{k+1} = (q_{k-M+1}, \dots, q_k)$, since they enter the constraint function (7.8b). The resulting linearizations do not contain any sensitivity information with respect to the controls and therefore it is not possible to adapt to more recent online data. The past measurements $\boldsymbol{\eta}_{k+1}$ as well as the arrival cost \bar{x}_{k-M} and its covariance matrix $\Sigma_{\text{arr}, k-M}$ are changing too, but they enter the objective function algebraically only and can be evaluated in the feedback phase. We will describe how to deal with them in Section 7.3.3.

To decouple the constraint function b from the online data \mathbf{q}_{k+1} , we introduce auxiliary control decision variables $s^q = (s_0^q, \dots, s_{M-1}^q)$ and in the following they will be collected in

$$\boldsymbol{v} = (s^x, s^z, s^q, w) \in \mathbb{R}^{(M+1)n_x} \times \mathbb{R}^{(M+1)n_z} \times \mathbb{R}^{Mn_u} \times \mathbb{R}^{Mn_x} \quad (7.16)$$

too. We modify problem $\text{MHE}(y_k)$ by replacing all occurrences of q_i in (7.3b), (7.3d), and (7.3e) by the decision variables and introduce an additional linear CTE constraint

$$c(v, \mathbf{q}_k) = 0 \quad (7.17)$$

with

$$c_i(v, \mathbf{q}_k) = s_i^q - q_{k-M+i} \quad \text{for } i = 0, \dots, M-1. \quad (7.18)$$

We write the resulting modified NLP in the compact form

$$\widetilde{\text{MHE}}(y_k) := \min_{v=(s^x, s^z, s^q, w)} \frac{1}{2} \|\ell(v, y_k)\|_{\Sigma_k^{-1}}^2 \quad (7.19a)$$

$$\text{s.t.} \quad b(v) = 0, \quad (7.19b)$$

$$c(v, \mathbf{q}_k) = 0 \quad (7.19c)$$

and denote its solution by $\tilde{v}^*(y_k)$. The corresponding QP has the structure

$$\widetilde{\text{QP}}(v, y_k) := \min_{\Delta v} \frac{1}{2} \Delta v^\top A(v, y_k) \Delta v + a(v, y_k)^\top \Delta v \quad (7.20a)$$

$$\text{s.t.} \quad b(v) + B(v) \Delta v = 0, \quad (7.20b)$$

$$c(v, \mathbf{q}_k) + C \Delta v = 0. \quad (7.20c)$$

Note that $c(v, \mathbf{q}_k)$ is linear in v and therefore its Jacobian C is constant. The modified problem $\widetilde{\text{MHE}}(y_k)$ is equivalent to the original problem $\text{MHE}(y_k)$. With this modification, we achieve that constraint (7.19b) is independent of the online data y_k . This allows us to use the evaluations of $\widetilde{\text{MHE}}(y_k)$ not only to prepare for $\widetilde{\text{MHE}}(y_{k+1})$ but for multiple problems $\widetilde{\text{MHE}}(y_{k+l})$ with $l > 1$.

7.3.2 Shift Variable Initialization

In the context of MLI for MHE, the approximate solution of $\widetilde{\text{MHE}}(y_{k+l})$ is prepared based on the approximate solution of $\widetilde{\text{MHE}}(y_k)$ with $l \geq 1$ and therefore we have to modify the primal variable shift accordingly. To deal with a shift over l sampling

intervals, we introduce a generalized shift map $v' = P_{\text{shift}}^l(v, q_k)$ which is defined by

$$\begin{aligned}
s_j'^q &:= \begin{cases} s_{l+j}^q & \text{for } j \leq M-l-1, \\ q_k & \text{for } M-l-1 < j \leq M-1, \end{cases} \\
s_j'^x &:= \begin{cases} s_{l+j}^x & \text{for } j \leq M-l, \\ x(s_{j-1}'^x, s_{j-1}'^z, s_{j-1}'^q) & \text{for } M-l < j \leq M, \end{cases} \\
s_j'^z &:= \begin{cases} s_{l+j}^z & \text{for } j \leq M-l, \\ z(s_{j-1}'^x, s_{j-1}'^z, s_{j-1}'^q) & \text{for } M-l < j \leq M, \end{cases} \\
w_j' &:= \begin{cases} w_{l+j} & \text{for } j \leq M-l-1, \\ 0 & \text{for } M-l-1 < j \leq M-1. \end{cases}
\end{aligned} \tag{7.21}$$

This shift duplicates the last available control q_k to the new part of the horizon and uses piece wise integration to generate corresponding state variables that satisfy the matching condition. This way, the shift has a feasibility preserving property, i.e., feasibility of v_k for problem $\widehat{\text{MHE}}(y_k)$ implies feasibility of $P_{\text{shift}}^l(v_k, q_k)$ for problem $\widehat{\text{MHE}}(y_{k+l})$. Additionally, all but the last l components of the objective residual function $\ell(P_{\text{shift}}^l(v_k, q_k), y_{k+l})$ are a shifted version of the residual at $\ell(v_k, y_k)$. However, there are multiple strategies to initialization the optimization variables on the new part of the horizon. They are discussed in [24, Chapter 4] or [57, Chapter 4].

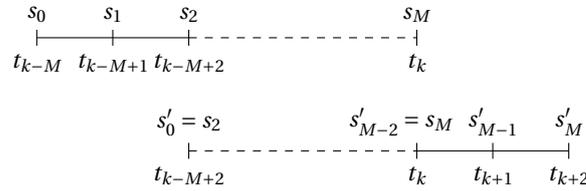


Figure 7.4: Illustration of the estimation horizon shift and the induced variable shift map P_{shift}^l with $l = 2$.

Note that the proposed variable initialization will violate the CTE constraint (7.17) if the predicted controls are not correct. However, as we assume that the shift initializes the primal variable close to the true solution and we therefore always apply full Gauß-Newton steps without any adaptive stepsize globalization strategy, the CTE constraint is satisfied after the first QP step, as it is linear in v .

Remark 7.4

The proposed way to initialize the newly introduced control values at the end of the estimation horizon is only one possibility. A different approach to initialize the new state variables is to duplicate the last known states. This is computationally cheaper, as no integration is needed, but additional infeasibilities at the new nodes are introduced.

7.3.3 Modified Feedback Phase

The CTE constraint allows to decouple the dynamical system from the controls \mathbf{q}_k , but the objective function is still dependent on the measurements $\boldsymbol{\eta}_k$ as well as the arrival cost \bar{x}_{k-M} and its covariance matrix $\Sigma_{\text{arr},k-M}$. In contrast to the standard RTI-MHE, the objective function cannot be evaluated in the preparation phase, since not all the data is available. However, the objective function

$$\phi(v, y_k) = \frac{1}{2} \|\ell(v, y_k)\|_{\Sigma_k^{-1}}^2 \quad (7.22)$$

does not include the integration of the dynamical system and is effortless to evaluate. Therefore, we only evaluate the gradient of the residual function $\nabla_v \ell(v)$ in the preparation phase, since it does not depend on y_k . The Hessian approximation $A(v, y_k) = \nabla_v \ell(v)^\top \Sigma_k^{-1} \nabla_v \ell(v)$ and gradient $a(v, y_k) = \nabla_v \ell(v)^\top \Sigma_k^{-1} \ell(v, y_k)$ are assembled in the feedback phase. This increases the computational effort of the feedback phase, but allows to react to the online data y_k . Since it includes only algebraic operations, the additional effort of the feedback phase is low.

The pseudocode of the modified RTI-MHE with CTE constraint is shown in Algorithm 7.2. Since all the QP data is evaluated in every iteration, the algorithm is equivalent to the standard RTI-MHE.

CTE in Multi-Level Iterations

The reason for the introduction of the CTE constraint and the generalized shift map P_{shift}^l for $l > 1$ is that in the context of MLI the preparation of the QP data may take $l > 1$ sampling intervals and thus it is necessary to start the preparation for problem $\widetilde{\text{MHE}}(y_{k+l})$ already at time t_k . Due to the introduction of auxiliary control variables, $\widetilde{\text{QP}}(v'_{k+l}, y_{k+l})$ contains sensitivities with respect to the not finally determined controls $q_{k+1}, \dots, q_{k+l-1}$. This allows to prepare $\widetilde{\text{QP}}(v'_{k+l}, y_{k+l})$ based on a predicted control trajectory, which is in this case a duplication of the last known control q_k . Deviations between the predicted controls and the actual applied control are taken into account in the QP solution step. This way, CTE can compensate for the lack of

Algorithm 7.2: Modified RTI-MHE main loop**loop***preparation phase:*evaluate $B(v'), b(v'), \nabla_v \ell(v')$ wait for state measurement η_k to complete y_k *feedback phase:*evaluate residual $\ell(v', y_k)$ evaluate $A(v', y_k) = \nabla_v \ell(v')^\top \Sigma_k^{-1} \nabla_v \ell(v')$, $a(v', y_k) = \nabla_v \ell(v')^\top \Sigma_k^{-1} \ell(v', y_k)$ solve $\widetilde{QP}(v', y_k)$ submit $\xi_k = s_M^x + \Delta s_M^x$ to controller*transition phase:*wait for applied control q_k from controller to complete q_{k+1} $v' \leftarrow P_{\text{shift}}^1(v' + \Delta v^{QP}, q_k)$ $k \leftarrow k + 1$ **end loop**

knowledge of the final correct control values in the preparation phase and thus improves the resulting estimation quality. The following academic example illustrates this effect.

Example 7.1 (CTE vs. Predicted Controls)

Consider the dynamic system $\dot{x}(t) = u$ with the estimation horizon $[t_k - 2, t_k]$ discretized with 2 equidistant shooting intervals and nodes $t_k - 2, t_k - 1, t_k$. Let $h(x) = x$ be the trivial measurement function and we assume that the measurements η are normally distributed $\eta - h(x) \sim \mathcal{N}(0, 1)$. For simplicity, we consider the resulting MHE problem without arrival cost term. As both the dynamics and the measurement function are linear, one full Gauß-Newton step corresponds to the solution of $\widetilde{\text{MHE}}(y_k)$. Suppose the preparation of the solution step of $\text{MHE}(y_{k+2})$ is based on data available at time t_k with the predicted controls $\tilde{q}_k, \tilde{q}_{k+1}$. This leads to the optimization problem

$$\min_{s^x \in \mathbb{R}^3} \frac{1}{2} \left\| \begin{pmatrix} s_0^x - \eta_k \\ s_1^x - \eta_{k+1} \\ s_2^x - \eta_{k+2} \end{pmatrix} \right\|^2 \quad \text{s.t.} \quad \begin{pmatrix} s_0^x + \tilde{q}_k - s_1^x \\ s_1^x + \tilde{q}_{k+1} - s_2^x \end{pmatrix} = 0,$$

which, after eliminating s_0^x and s_1^x , gives the estimate

$$s_2^{x*} = \frac{(1, 1, 1)}{\|(1, 1, 1)\|^2} \cdot \begin{pmatrix} \eta_k + \tilde{q}_k + \tilde{q}_{k+1} \\ \eta_{k+1} + \tilde{q}_{k+1} \\ \eta_{k+2} \end{pmatrix}.$$

Here \cdot indicates the scalar product. The solution s_2^{x*} depends on the predicted controls \tilde{q}_k and \tilde{q}_{k+1} and it is obvious that the estimate will be wrong if the predicted controls are not correct. On the other hand, the corresponding MHE problem with CTE constraint $\widehat{\text{MHE}}(y_{k+2})$ reads as

$$\min_{\substack{s^x \in \mathbb{R}^3 \\ s^q \in \mathbb{R}^2}} \frac{1}{2} \left\| \begin{pmatrix} s_0^x - \eta_k \\ s_1^x - \eta_{k+1} \\ s_2^x - \eta_{k+2} \end{pmatrix} \right\|^2 \quad \text{s.t.} \quad \begin{pmatrix} s_0^x + s_0^q - s_1^x \\ s_1^x + s_1^q - s_2^x \\ s_0^q - q_k \\ s_1^q - q_{k+1} \end{pmatrix} = 0$$

which leads to the estimate

$$s_2^{x*} = \frac{(1, 1, 1)}{\|(1, 1, 1)\|^2} \cdot \begin{pmatrix} \eta_k + q_k + q_{k+1} \\ \eta_{k+1} + q_{k+1} \\ \eta_{k+2} \end{pmatrix}$$

with the correct controls q_k . Consequently, although both linearizations are prepared at time t_k , only the CTE version gives the correct estimate as it can incorporate a correction accounting for possibly wrongly predicted controls in the QP solution step.

Remark 7.5

The proposed CTE allows to prepare $\widehat{\text{MHE}}(y_{k+l})$ at time t_k by including sensitivities with respect to the auxiliary control variables s^q . This allows to react to deviations between the predicted and the true, but at preparation time not determined controls $q_{k+1}, \dots, q_{k+l-1}$. Contrary to that, the previous controls q_{k+l-M}, \dots, q_k are already determined and the resulting auxiliary control updates $\Delta s_0^q, \dots, \Delta s_{M-l-1}^q$ will be zero. In case that the necessary number of sampling intervals l is already known at the start of the preparation phase, it is possible to eliminate the corresponding auxiliary variables from the problem in advance. However, the proposed CTE formulation is more flexible and can also be applied when it is unclear at which sampling time the prepared linearizations will be used. Such a situation occurs, e.g., when implementing an MLI scheme for MHE with fully automated level choice as presented in Section 5.3 and used in [103] for NMPC.

7.4 Multi-Level Iteration for Moving Horizon Estimation

In the previous section, we proposed modifications to the standard RTI-MHE algorithm in order to make the underlying QPs compatible with the MLI update formulas. In this section, we describe how the update formulas are transferred from NMPC to MHE. However, since MHE is structural similar to NMPC, the MLI update formulas are similar too. They comprise four hierarchical levels of descending computational complexity, denoted by the letters A, B, C, and D. While on the highest level (Level D) all linearizations are evaluated at the current primal variable, the lower levels only update a subset of the linearization data and instead use reference constraint evaluations and reference constraint Jacobians. Every level can be operated as a separate MHE scheme and is working on its own set of variables. We will first describe the levels individually and afterwards how they are combined in a holistic scheme. In this section, we focus on Level B, C, and D. We do not consider Level A, because it refers to linear MHE and is already extensively studied, see e.g. [5]. The different update levels are inspired by the existing MLI levels for NMPC feedback generation, for which the convergence properties are discussed in [15, 113] and Chapter 5.

The constraint function b of problem (7.8) is a collection of dynamical matching conditions (7.3d) and algebraic consistency conditions (7.3e). Since the dynamical matching conditions include the integration of a dynamical system, they are expensive to evaluate. In contrast to that, the consistency conditions are purely algebraic and therefore the effort to evaluate them and their Jacobians is comparatively low. The MLI update formulas for NMPC are applied to the full constraint function. In contrast to that, we will apply the MLI update formulas only to the dynamical equations. The algebraic equations will be reevaluated in every iteration. This way we only approximate the computational expensive dynamical equations while the algebraic equations are evaluated with full accuracy. In order to distinguish the dynamical from the algebraic constraints, we use the notation

$$b(v) = \begin{bmatrix} b_{\text{dyn}}(v) \\ b_{\text{alg}}(v) \end{bmatrix} \quad (7.23)$$

for the constraint function and

$$B(v) = \begin{bmatrix} B_{\text{dyn}}(v) \\ B_{\text{alg}}(v) \end{bmatrix} \quad (7.24)$$

for the respective Jacobians. Using this notation, QP (7.20) has the structure

$$\widetilde{\text{QP}}(v, \lambda, y) = \min_{\Delta v} \frac{1}{2} \Delta v^\top A(v, y) \Delta v + a(v, y)^\top \Delta v \quad (7.25a)$$

$$\text{s.t.} \quad B_{\text{dyn}}(v) \Delta v + b_{\text{dyn}}(v) = 0, \quad (7.25b)$$

$$B_{\text{alg}}(v) \Delta v + b_{\text{alg}}(v) = 0, \quad (7.25c)$$

$$C \Delta v + c(v, q) = 0. \quad (7.25d)$$

In order to have a compact notation, we collect the primal-dual variables in

$$\vartheta = \left(v, \lambda_{b_{\text{dyn}}}, \lambda_{b_{\text{alg}}}, \lambda_c \right) \in \mathbb{R}^{n_\vartheta}, \quad (7.26)$$

where $\lambda_{b_{\text{dyn}}}$, $\lambda_{b_{\text{alg}}}$ and λ_c are the dual multiplier associated to (7.25b) - (7.25d). Since evaluating (7.25c) and (7.25d) is cheap, we apply the MLI update formulas only to the dynamical constraint function b_{dyn} . The algebraic conditions $b_{\text{alg}}(v)$ are reevaluated in every iteration.

The Principle of the MLI for MHE

Every level $X \in \{D, C, B\}$ holds its own set of primal-dual iterates ϑ^X and a reference value for the constraints Jacobian $\overline{B}_{\text{dyn}}^X$. This reference value plays a crucial role for the MLI update formulas. It is used to adjust the QP data

$$\Theta^X(\vartheta) = \left(A^X(v, \lambda), a^X(v), B_{\text{dyn}}^X(v), b_{\text{dyn}}^X(v) \right) \in \mathbb{R}^{n_\nu \times n_\nu} \times \mathbb{R}^{n_\nu} \times \mathbb{R}^{n_{b_{\text{dyn}}} \times n_\nu} \times \mathbb{R}^{n_{b_{\text{dyn}}}}. \quad (7.27)$$

to the current iteration variables by updates with low computational effort. However, the quality of the resulting Jacobian approximation depends on a suitable choice of $\overline{B}_{\text{dyn}}$. In contrast to MLI for NMPC, we do not need reference values for the objective function, because we reevaluate it in every iteration. The MLI for MHE (MLI-MHE) algorithm works similar as Algorithm 7.2. The only difference is that

$$\widetilde{\text{QP}}^X(\vartheta, y) := \min_{\Delta v} \frac{1}{2} \Delta v^\top A^X(v) \Delta v + (a^X(v))^\top \Delta v \quad (7.28a)$$

$$\text{s.t.} \quad B_{\text{dyn}}^X(v) \Delta v + b_{\text{dyn}}^X(v) = 0, \quad (7.28b)$$

$$B_{\text{alg}}(v) \Delta v + b_{\text{alg}}(v) = 0, \quad (7.28c)$$

$$C \Delta v + c(v, q) = 0. \quad (7.28d)$$

is solved to update the iterates. To simplify the notation, we will omit the superindex, if the corresponding level is clear from the context.

Shifting of Reference Values

One of the central components of the MLI-MHE algorithm is the reference value \bar{B}_{dyn} as an approximation of the real Jacobian $B_{\text{dyn}}(v)$ at a point v . In general, the reference matrix \bar{B}_{dyn} is an arbitrary approximation of the real Jacobian but the performance of the algorithm depends on the approximation quality of it. Usually, the reference value \bar{B}_{dyn} is generated by a full evaluation of the constraint Jacobian $B_{\text{dyn}}(v)$ at a specific point v . Remember that the iteration variables are a collection of the variables

$$v = (s^x, s^z, s^q, w) \quad (7.29)$$

as described in Section 7.3.1. Since b_{dyn} is assembled from the individual matching conditions, its Jacobian is given by

$$B_{\text{dyn}}(v) = \frac{d}{dv} \begin{bmatrix} x(s_0^x, s_0^z, s_0^q) + w_0 - s_1^x \\ \vdots \\ x(s_{M-1}^x, s_{M-1}^z, s_{M-1}^q) + w_{M-1} - s_M^x \end{bmatrix} \quad (7.30)$$

$$= [G^x \quad G^z \quad G^q \quad \mathbb{1}]$$

with

$$G^x = \begin{bmatrix} G_0^x & -\mathbb{1} & & \\ & \ddots & \ddots & \\ & & G_{M-1}^x & -\mathbb{1} \end{bmatrix}, \quad G^z = \begin{bmatrix} G_0^z & & & \\ & \ddots & & \\ & & G_{M-1}^z & 0 \end{bmatrix}, \quad (7.31a)$$

$$G^q = \begin{bmatrix} G_0^q & & & \\ & \ddots & & \\ & & G_{M-1}^q & 0 \end{bmatrix} \quad (7.31b)$$

and

$$G_i^x = \frac{d}{ds^x} x(s_i^x, s_i^z, s_i^q), \quad G_i^z = \frac{d}{ds^z} x(s_i^x, s_i^z, s_i^q), \quad G_i^q = \frac{d}{ds^q} x(s_i^x, s_i^z, s_i^q) \quad (7.32)$$

for $i = 0, \dots, M-1$. In order to exploit the structure of the constraint Jacobian, we propose to shift the individual blocks alongside with the iteration variables. Let the

reference matrix $\bar{B}_{\text{dyn},k}$ be generated by a full Jacobian evaluation $B_{\text{dyn}}(v_k)$ at time t_k . Then, the reference matrix has the same structure as the constraint Jacobian

$$\bar{B}_{\text{dyn},k} = B_{\text{dyn}}(v_k) = \begin{bmatrix} \bar{G}_k^x & \bar{G}_k^z & \bar{G}_k^q & \mathbb{1} \end{bmatrix}, \quad (7.33)$$

with

$$\bar{G}_k^x = \begin{bmatrix} \bar{G}_{0,k}^x & -\mathbb{1} & & \\ & \ddots & \ddots & \\ & & \bar{G}_{M-1,k}^x & -\mathbb{1} \end{bmatrix}, \quad \bar{G}_k^z = \begin{bmatrix} \bar{G}_{0,k}^z & & & \\ & \ddots & & \\ & & \bar{G}_{M-1,k}^z & 0 \end{bmatrix}, \quad (7.34a)$$

$$\bar{G}_k^q = \begin{bmatrix} \bar{G}_{0,k}^q & & & \\ & \ddots & & \\ & & \bar{G}_{M-1,k}^q & 0 \end{bmatrix}. \quad (7.34b)$$

The individual block matrices $\bar{G}_{i,k}^x$, $\bar{G}_{i,k}^z$ and $\bar{G}_{i,k}^q$ are the sensitivity matrices of the matching conditions evaluated at $\bar{s}_{i,k} = (\bar{s}_{i,k}^x, \bar{s}_{i,k}^z, \bar{s}_{i,k}^q)$. In MLI-MHE, the primal variables are shifted with the measurements in every iteration, as described in Section 7.3.2. The fundamental assumption of the RTI-MHE is that at time t_k the primal variables v_k describe the process on the interval $[t_{k-M}, t_k]$ accurately and therefore the update Δv is small. At the transition from iteration k to $k+1$, the primal variables are shifted by the map $v_{k+1} = P_{\text{shift}}^1(v_k)$. Since the update Δv is small, the matrices $\bar{G}_{i,k}^x$, $\bar{G}_{i,k}^z$ and $\bar{G}_{i,k}^q$ are approximations of the Jacobians $G_{i-1,k+1}^x$, $G_{i-1,k+1}^z$ and $G_{i-1,k+1}^q$. In order to get an accurate approximation of the complete constraint Jacobian B_{dyn} , we shift the block rows of $\bar{B}_{\text{dyn},k}$ together with the primal variables

$$\bar{B}_{\text{dyn},k+1} = \begin{bmatrix} \bar{G}_{k+1}^x & \bar{G}_{k+1}^z & \bar{G}_{k+1}^q & \mathbb{1} \end{bmatrix} \quad (7.35)$$

with

$$\bar{G}_{k+1}^x = \begin{bmatrix} \bar{G}_{1,k}^x & -\mathbb{1} & & & \\ & \ddots & \ddots & & \\ & & \bar{G}_{M-1,k}^x & -\mathbb{1} & \\ & & & G_{M-1,k+1}^x & -\mathbb{1} \end{bmatrix}, \quad (7.36a)$$

$$\bar{G}_{k+1}^z = \begin{bmatrix} \bar{G}_{1,k}^z & & & & \\ & \ddots & & & \\ & & \bar{G}_{M-1,k}^z & & \\ & & & G_{M-1,k+1}^z & 0 \\ & & & & 0 \end{bmatrix}, \quad (7.36b)$$

$$\bar{G}_{k+1}^q = \begin{bmatrix} \bar{G}_{1,k}^q & & & & \\ & \ddots & & & \\ & & \bar{G}_{M-1,k}^q & & \\ & & & G_{M-1,k+1}^q & 0 \\ & & & & 0 \end{bmatrix}. \quad (7.36c)$$

Similar to the variable shift, the Jacobians that enter at the last node $G_{M-1,k+1}^x$, $G_{M-1,k+1}^z$ and $G_{M-1,k+1}^q$ are generated by a forward integration. We denote the corresponding shift map by $P_{\text{shift,dyn}}(\bar{B}_{\text{dyn},k}, \nu_k, q_k)$. Even though, this increases the computational effort, the performance can be increased for nonlinear systems since the reference matrix is a better approximation of the exact Jacobian. In level D, the Jacobian is reevaluated in every iteration and therefore the shift is only applied in Level C and B.

This procedure can be viewed as the MHE equivalent of Fractional Level Updates for MLI as introduced in [38]. In fractional level updates the first few shooting intervals are fully reevaluated and only on the last intervals the update formula of level C is used. The pseudocode of the complete MLI algorithm with a fixed level is summarized in Algorithm 7.3.

7.4.1 MLI Update Formulas for MHE

In the following, we present the main ideas behind each level $X \in \{D, C, B\}$ and we introduce the specific update formulas $\Theta^X(\vartheta)$ for the QP data. Table 7.1 gives the exact description which evaluations and derivatives are required and how they are used to assemble the QP data for each level. We will discuss their convergence properties in Section 7.4.2.

Level D – Gauß-Newton (GN) Iterations

Level D provides the QP with new evaluations and Jacobians of all functions at the current primal variable, i.e. the updates $\Theta^D(\vartheta)$ are given by

$$A^D(v) = A(v, y_k), \quad a^D(v) = a(v, y_k), \quad (7.37a)$$

$$B_{\text{dyn}}^D(v) = B_{\text{dyn}}(v), \quad b_{\text{dyn}}^D(v) = b_{\text{dyn}}(v). \quad (7.37b)$$

Algorithm 7.3: Main loop of MLI-MHE algorithm with fixed level $X \in \{D, C, B\}$

loop
preparation phase:

 evaluate $B_{\text{alg}}(v')$, $b_{\text{alg}}(v')$, $\nabla_v \ell(v')$, $B_{\text{dyn}}^X(v')$, $b_{\text{dyn}}^X(v')$

 wait for state measurement η_k to complete y_k
feedback phase:

 evaluate residual $\ell(v', y_k)$ and CTE $c(v', y_k)$

 evaluate $A^X(v', y_k)$ and $a^X(v', y_k)$

 solve $\widetilde{\text{QP}}^X(v', y_k)$

 submit $\xi_k = s_M^x + \Delta s_M^x$ to controller

transition phase:

 wait for applied control q_k from controller to complete q_{k+1}
 $v' \leftarrow P_{\text{shift}}^1(v' + \Delta v^{QP}, q_k)$
 $\overline{B}_{\text{dyn}, k+1} \leftarrow P_{\text{shift}, \text{dyn}}(\overline{B}_{\text{dyn}, k}, v' + \Delta v^{QP}, q_k)$
 $k \leftarrow k + 1$
end loop

Level	necessary evaluations			
	$b_{\text{dyn}}(v)$	$B_{\text{dyn}}(v)$	$a(v, y_k)$	$A(v, y_k)$
D	✓	✓	✓	✓
C	✓	(✓) ¹	✓	✓
B	✓	✗	✓	✓
QP data update formulas				
	b_{dyn}	B_{dyn}	a	A
D	$b_{\text{dyn}}(v)$	$B_{\text{dyn}}(v)$	$a(v, y_k)$	$A(v, y_k)$
C	$b_{\text{dyn}}(v)$	$\overline{B}_{\text{dyn}}$	$a(v, y_k) + (\overline{B}_{\text{dyn}} - B_{\text{dyn}}(v))^{\top} \lambda_{b_{\text{dyn}}}$	$A(v, y_k)$
B	$b_{\text{dyn}}(v)$	$\overline{B}_{\text{dyn}}$	$a(v, y_k)$	$A(v, y_k)$

¹ Only the vector-matrix product $\lambda_{b_{\text{dyn}}}^{\top} B_{\text{dyn}}(v)$ needs to be computed in an adjoint fashion.

Table 7.1: Computations and update formulas for the QP data for the different levels. The linearization points v are the shifted approximate solutions of the previous MHE problem, according to (7.21). The reference evaluations and Jacobians as well as the multipliers in Level C are the shifted versions $\lambda_{b_{\text{dyn}}}$ of the preceding MHE QPs.

The solution of $\widetilde{QP}^D(\vartheta)$ corresponds to a full SQP step with the GN Hessian approximation of the Lagrangian. For constant online data $y_k = y$, these update formulas refer to the GN method for constrained least-square problems as discussed, e.g., in [13]. In the MLI hierarchy, Level D is the most expensive update, but is provide to most accurate system linearizations because the full constraint Jacobian is reevaluated.

Modified Level C – Optimality Iterations

Level C omits the most expensive computation of Level D, namely the full sensitivity generation of the DAE system corresponding to the matrix $B_{\text{dyn}}(v)$. Instead, the matrix-vector product $\lambda_{b_{\text{dyn}}}^T B_{\text{dyn}}(v)$ is computed in an adjoint fashion, which comes at a significantly lower computational cost and is used to modify the gradient a in the objective function. Taking into account the discrepancy of the true Jacobian $B_{\text{dyn}}(v)$ and the reference Jacobian \bar{B}_{dyn} , the adjoint derivative $\lambda^T B_{\text{dyn}}(v)$ can be used to modify the QP gradient a accordingly. The complete formulas for Level C $\Theta^C(\vartheta)$ are

$$A^C(v) = A(v, y_k), \quad a^C(v) = a(v, y_k) + (\bar{B}_{\text{dyn}} - B_{\text{dyn}}(v))^T \lambda_{b_{\text{dyn}}} \quad (7.38a)$$

$$B_{\text{dyn}}^C(v) = \bar{B}_{\text{dyn}}, \quad b_{\text{dyn}}^C(v) = b_{\text{dyn}}(v). \quad (7.38b)$$

For fixed online data $y = y_k$, it can be shown that if Level C iterates converge, they converge to a Karush-Kuhn-Tucker (KKT) point of the original NLP, which is the reason why these iterates are also called *Optimality Iterations*.

Modified Level B – Feasibility Iterations

Level B further avoids any sensitivity computation for the shooting conditions in b_{dyn} . All other functions and derivatives are evaluated. In comparison to MLI for NMPC, we evaluate the objective function fully in every iteration. Therefore a correction of the gradient is not necessary in Level B for MHE. The QP data $\Theta^B(\vartheta)$ is defined by

$$A^B(v) = A(v, y_k), \quad a^B(v) = a(v, y_k), \quad (7.39a)$$

$$B_{\text{dyn}}^B(v) = \bar{B}_{\text{dyn}}, \quad b_{\text{dyn}}^B(v) = b_{\text{dyn}}(v). \quad (7.39b)$$

We note that, except the shifting procedure of the reference matrix, the presented Level B is closely related to zero-order MHE as presented in [7].

Remark 7.6 (Lower Levels)

We emphasize that levels with even fewer evaluations and linearization updates could be defined. The counterpart of Level A from NMPC for MHE is an update formula with no evaluations and no Jacobians at all and subsequent linearizations are just kept constant. Such a level is essentially linear MHE and could be combined with the presented higher levels. For the sake of simplicity we omit a detailed description of such lower levels and refer the reader to [15].

7.4.2 Convergence Properties of MLI for MHE Update Formulas

In order to investigate the convergence properties of the individual levels, we show that they can be interpreted as Newton-type methods for specific nonlinear systems. This enables us to employ the convergence theory of Newton-type methods. Then we show that iterates converge either to the solution of the original problem or to a disturbed problem. We will assume that the online data is fixed, i.e. $y_k = y$ for all $k \in \mathbb{N}$. Consequently, we do not consider the shifting procedures of the variables P_{shift} and the reference data $P_{\text{shift,dyn}}$ throughout this section. This allows us, to only consider the optimization problem and not the interaction with the controlled process. An investigation of the complete closed loop behavior with the process is subject to further research. The theoretical results for NMPC are summarized in Section 5.1.6. The analysis of the local convergence properties of the proposed levels follows the lines of [15] for the NMPC case and is based on the Local Contraction Theorem [13] for Newton-type iterations.

$\widetilde{\text{MHE}}(y)$ is an equality constrained NLP and we assumed that the dynamical system and therefore the constraint function is sufficiently smooth. Therefore a local minimizer ν^* satisfies the KKT conditions for equality constrained NLPs

$$F(\vartheta^*) := \begin{bmatrix} \nabla_v \mathcal{L}_y(\vartheta^*) \\ b(\nu^*) \\ c(\nu^*) \end{bmatrix} = 0 \quad (7.40)$$

with dual variables λ_b^* and λ_c^* and the Lagrange function

$$\mathcal{L}_y(\vartheta) = \frac{1}{2} \|l(\nu, y)\|_{\Sigma^{-1}}^2 - \lambda_b^\top b(\nu) - \lambda_c^\top c(\nu). \quad (7.41)$$

The nonlinear equation system $F(\vartheta) = 0$ can be solved by Newton's method under suitable conditions. Starting from an initial guess ϑ_0 , a sequence of iterates is gener-

ated by $\vartheta_{j+1} = \vartheta_j + \Delta\vartheta$, where $\Delta\vartheta$ is the solution of

$$\frac{d}{d\vartheta}F(\vartheta_j)\Delta\vartheta + F(\vartheta_j) = 0. \quad (7.42)$$

If the starting point ϑ_0 is in the vicinity of the the solution and $\frac{d}{d\vartheta}F(\vartheta)$ is nonsingular, Newton's method converges quadratically. However, $\frac{d}{d\vartheta}F(\vartheta)$ might be singular and expensive to evaluate and therefore Newton's method is not suitable in many applications. In addition, the iterates of Newton's method are attracted to spurious solutions of the KKT system such as maxima or saddle points [13, 16]. Newton-type methods avoid the direct evaluation of $\frac{d}{d\vartheta}F(\vartheta)$ by the usage of an approximation $J(\vartheta)$ instead and the linear equation system

$$J(\vartheta_j)\Delta\vartheta + F(\vartheta_j) = 0. \quad (7.43)$$

is solved in every iteration. To prove that level D, B and C are converging, we show that they can be interpreted as Newton-type methods. To ensure that the iteration rule is well defined, we make the following assumption.

Assumption 7.1 (Regularity of NLP Solution)

$\widetilde{\text{MHE}}(y)$ has a local minimizer $\vartheta^* = (v^*, \lambda_b^*, \lambda_c^*)$ that satisfies the Linear Independence Constraint Qualification (LICQ), i.e. the rows of the constraint Jacobians $B(v^*)$ and C are linearly independent. For $X \in \{D, C, B\}$ we assume that there is a neighborhood U of ϑ^* , such that for all $\vartheta \in U$ the row vectors of the constraint matrix of $\widetilde{\text{QP}}^X(\vartheta, y)$ are linearly independent and the Hessian approximation $A(v, y)$ is positive definite on the null space of the constraint matrix.

Level D

Level D is defined by the update formulas (7.37a) and fully evaluates the dynamical system as well as the sensitives. It is equivalent to a full RTI step for MHE. The following proposition shows that it can be interpreted as a Newton-type method for the KKT system of $\widetilde{\text{MHE}}(y)$.

Proposition 7.2

One Level D iteration defined by the solution of $\widetilde{\text{QP}}^D(\vartheta_j, y)$ is one Newton-type step towards the root of the KKT system $F^D(\vartheta) = 0$ of $\widetilde{\text{MHE}}(y)$ with

$$J^D(\vartheta) = \begin{bmatrix} A(v, y) & -B(v)^\top & -C^\top \\ B(v) \\ C \end{bmatrix} \quad (7.44)$$

as an approximation of the Jacobian $\frac{d}{d\vartheta} F^D(\vartheta)$.

Proof Level D computes the updates of iterates by

$$v_{j+1} = v_j + \Delta v, \quad \lambda_{b,j+1} = \lambda_b^{QP}, \quad \lambda_{c,j+1} = \lambda_c^{QP} \quad (7.45)$$

where $(\Delta v, \lambda_b^{QP}, \lambda_c^{QP})$ is the solution of $\widetilde{QP}^D(\vartheta_j, y)$. Since $\widetilde{QP}^D(\vartheta_j, y)$ is an equality constrained QP, its KKT system is given by

$$J^D(\vartheta_j) \begin{bmatrix} \Delta v \\ \lambda_{b,j+1} \\ \lambda_{c,j+1} \end{bmatrix} + \begin{bmatrix} a(v_j, y) \\ b(v_j) \\ c(v_j, y) \end{bmatrix} = 0 \quad (7.46)$$

with

$$J^D(\vartheta) = \begin{bmatrix} A(v, y) & -B(v)^\top & -C^\top \\ B(v) \\ C \end{bmatrix}. \quad (7.47)$$

Assumption 7.1 ensures that the solution of the KKT system is well defined. This is a well known result of nonlinear programming theory [80, Chapter 16]. Using $\Delta \lambda_b = \lambda_{b,j+1} - \lambda_{b,j}$ and $\Delta \lambda_c = \lambda_{c,j+1} - \lambda_{c,j}$, equation (7.46) is equivalent to

$$J^D(v_j) \begin{bmatrix} \Delta v \\ \Delta \lambda_b \\ \Delta \lambda_c \end{bmatrix} + \begin{bmatrix} a(v_j, y) - B(v_j)^\top \lambda_{b,j} - C^\top \lambda_{c,j} \\ b(v_j) \\ c(v_j, y) \end{bmatrix} = 0. \quad (7.48)$$

Thus, Level D is a Newton-type method for

$$F^D(\vartheta) = \begin{bmatrix} a(v, y) - B(v)^\top \lambda_b - C^\top \lambda_c \\ b(v) \\ c(v, y) \end{bmatrix} \quad (7.49)$$

with $J^D(\vartheta)$ as approximation of $\frac{d}{d\vartheta} F^D(\vartheta)$. The first entry of F^D is the gradient of the Lagrange function $\nabla_v \mathcal{L}_y(\vartheta)$ with respect to v and therefore F^D refers to the KKT conditions (7.40). \square

Remark 7.7

Note that $J^D(\vartheta)$ differs from the exact Jacobian $\frac{d}{d\vartheta} F^D(\vartheta)$ only by $A(v, y)$ as an approximation of the Hessian $\nabla_v^2 \mathcal{L}_y(\vartheta)$. We are using a Gauß-Newton approximation

for $\nabla_v \mathcal{L}_y(\vartheta)$. However, if $A(v, y) = \nabla_v \mathcal{L}_y(\vartheta)$ the Level D iterations correspond to steps of the exact Newton method and has its quadratic convergence rate.

Level C

In Level C, the full sensitivity generation of the constraints are omitted. However, Level C still is a Newton-type method for the KKT system of the original problem $\overline{\text{MHE}}(y)$ as the following proposition shows.

Proposition 7.3

One Level C iteration defined by the solution of $\widetilde{\text{QP}}^C(\vartheta_j, y)$ is one Newton-type step towards the root the KKT system $F(\vartheta)^C = F(\vartheta) = 0$ of $\overline{\text{MHE}}(y)$ with

$$J^C(\vartheta) = \begin{bmatrix} A(v, y) & -\overline{B}_{\text{dyn}}^\top & -B_{\text{alg}}(v)^\top & -C^\top \\ \overline{B}_{\text{dyn}} \\ B_{\text{alg}}(v) \\ C \end{bmatrix} \quad (7.50)$$

as an approximation of the Jacobian $\frac{d}{d\vartheta} F^C(\vartheta)$.

Proof Similar to Level D, Level C computes the updates of iterates by

$$v_{j+1} = v_j + \Delta v, \quad \lambda_{b_{\text{dyn}}, j+1} = \lambda_{b_{\text{dyn}}}^{QP}, \quad \lambda_{b_{\text{alg}}, j+1} = \lambda_{b_{\text{alg}}}^{QP}, \quad \lambda_{c, j+1} = \lambda_c^{QP} \quad (7.51)$$

where $(\Delta v, \lambda_{b_{\text{dyn}}}^{QP}, \lambda_{b_{\text{alg}}}^{QP}, \lambda_c^{QP})$ is the solution of $\widetilde{\text{QP}}^C(\vartheta_j, y)$. The KKT conditions of $\widetilde{\text{QP}}^C(\vartheta_j, y)$ are given by

$$J^C(\vartheta) \begin{bmatrix} \Delta v \\ \lambda_{b_{\text{dyn}}, j+1} \\ \lambda_{b_{\text{alg}}, j+1} \\ \lambda_{c, j+1} \end{bmatrix} + \begin{bmatrix} a(v_j, y) + (\overline{B}_{\text{dyn}} - B_{\text{dyn}}(v_j))^\top \lambda_{\text{dyn}, j} \\ b_{\text{dyn}}(v_j) \\ b_{\text{alg}}(v_j) \\ c(v_j, y) \end{bmatrix} = 0. \quad (7.52)$$

In order to reformulate this equation system, we introduce $\Delta \lambda_{b_{\text{dyn}}} = \lambda_{b_{\text{dyn}}, j+1} - \lambda_{b_{\text{dyn}}, j}$, $\Delta \lambda_{b_{\text{alg}}} = \lambda_{b_{\text{alg}}, j+1} - \lambda_{b_{\text{alg}}, j}$ and $\Delta \lambda_c = \lambda_{c, j+1} - \lambda_{c, j}$. This leads to the system

$$J^C(\vartheta) \begin{bmatrix} \Delta v \\ \Delta \lambda_{b_{\text{dyn}}} \\ \Delta \lambda_{b_{\text{alg}}} \\ \Delta \lambda_c \end{bmatrix} + F^C(\vartheta) = 0 \quad (7.53)$$

with

$$F^C(\vartheta) = \begin{bmatrix} a(v, y) - B_{\text{dyn}}(v)^\top \lambda_{b_{\text{dyn}}} - B_{\text{alg}}(v)^\top \lambda_{b_{\text{alg}}} - C^\top \lambda_c \\ b_{\text{dyn}}(v) \\ b_{\text{alg}}(v) \\ c(v, y) \end{bmatrix}. \quad (7.54)$$

Since $F^C(\vartheta)$ is the KKT system of $\widetilde{\text{MHE}}(y)$, the iterates of Level C correspond to Newton-type iterations with $J^C(\vartheta)$ as an approximation of $\frac{d}{d\vartheta} F^C(\vartheta)$. \square

In contrast to level D, we distinguish between b_{dyn} and b_{alg} , because only B_{dyn} is approximated. Similar to Level D, Level C is a Newton-type method for the KKT conditions (7.40) and they only differ in the Jacobian approximation. The following Theorem shows that the limit of the Level C iterates is a KKT point of the original problem $\widetilde{\text{MHE}}(y)$.

Theorem 7.4 (Limit of Modified Level C Iterations)

Let the online data be fixed, i.e. $y_k = y$ for all $k \in \mathbb{N}$. Let the optimality iterations $\vartheta_j = (v_j, \lambda_{b_{\text{dyn},j}}, \lambda_{b_{\text{alg},j}}, \lambda_{c,j})$ generated by Algorithm 7.3 converge towards a limit $\vartheta^* = (v^*, \lambda_{b_{\text{dyn}}}^*, \lambda_{b_{\text{alg}}}^*, \lambda_c^*)$ and the primal QP solutions Δv_j converge to 0. Then the limit is a KKT point of the problem $\widetilde{\text{MHE}}(y)$.

Proof As the steps Δv_j of the Level C QPs converge to 0, continuity implies that $\Delta v^* = 0$ together with the dual variables $(\lambda_{b_{\text{dyn}}}^*, \lambda_{b_{\text{alg}}}^*, \lambda_c^*)$ is the primal dual solution of $\widetilde{\text{QP}}^C(\vartheta^*, y)$ and the KKT conditions are satisfied. The primal feasibility

$$\overline{B}_{\text{dyn}} \Delta v + b_{\text{dyn}}^C(v^*) = 0, \quad (7.55a)$$

$$B_{\text{alg}}(v^*) \Delta v + b_{\text{alg}}(v^*) = 0, \quad (7.55b)$$

$$C \Delta v + c(v^*, q) = 0 \quad (7.55c)$$

ensures that v^* is feasible for $\widetilde{\text{MHE}}(y)$, because $b_{\text{dyn}}^C(v^*) = b_{\text{dyn}}(v^*)$ and $\Delta v = 0$. The stationarity condition for $\widetilde{\text{QP}}^C(\vartheta^*, y)$ reads as

$$A(v^*, y) + a(v^*, y) + (\overline{B}_{\text{dyn}}^\top - B_{\text{dyn}}(v^*)^\top) \lambda_{b_{\text{dyn}}}^* - \overline{B}_{\text{dyn}}^\top \lambda_{b_{\text{dyn}}}^* - B_{\text{alg}}^\top(v^*) \lambda_{b_{\text{alg}}}^* - C(v^*, y)^\top \lambda_c^* = 0. \quad (7.56)$$

The terms involving \bar{B}_{dyn} cancel out and the stationarity condition of $\widetilde{\text{MHE}}(y)$ remains. Since $\widetilde{\text{MHE}}(y)$ has no inequality constraints, it follows that $(v^*, \lambda_{b_{\text{dyn}}}^*, \lambda_{b_{\text{alg}}}^*, \lambda_c^*)$ is a KKT point of $\widetilde{\text{MHE}}(y)$. \square

7.4.3 Level B

In Level B, we reduce the computational effort even further by omitting the directional derivative in the gradient correction. Therefore, Level B does not correspond to a Newton-type method for KKT system of $\widetilde{\text{MHE}}(y)$. However, the following proposition shows that it is a Newton-type method for a disturbed system, see Remark 7.8.

Proposition 7.5

One Level B iteration defined by the solution of $\widetilde{\text{QP}}^B(\vartheta_j, y)$ is one Newton-type step towards the root of the disturbed system

$$F^B(\vartheta) = \begin{bmatrix} a(v, y) - \bar{B}_{\text{dyn}}^\top \lambda_{b_{\text{dyn}}} - B_{\text{alg}}(v)^\top \lambda_{b_{\text{alg}}} - C^\top \lambda_c \\ b_{\text{dyn}}(v) \\ b_{\text{alg}}(v) \\ c(v, y) \end{bmatrix} \quad (7.57)$$

with the Jacobian approximation

$$J^B(\vartheta) = \begin{bmatrix} A(v, y) & -\bar{B}_{\text{dyn}}^\top & -B_{\text{alg}}(v)^\top & -C^\top \\ \bar{B}_{\text{dyn}} \\ B_{\text{alg}}(v) \\ C \end{bmatrix}. \quad (7.58)$$

Proof The iterates of Level B are computed by

$$v_{j+1} = v_j + \Delta v, \quad \lambda_{b_{\text{dyn}}, j+1} = \lambda_{b_{\text{dyn}}}^{\text{QP}}, \quad \lambda_{b_{\text{alg}}, j+1} = \lambda_{b_{\text{alg}}}^{\text{QP}}, \quad \lambda_{c, j+1} = \lambda_c^{\text{QP}} \quad (7.59)$$

where $(\Delta v, \lambda_{b_{\text{dyn}}}^{\text{QP}}, \lambda_{b_{\text{alg}}}^{\text{QP}}, \lambda_c^{\text{QP}})$ is the solution of $\widetilde{\text{QP}}^B(\vartheta_j, y)$. The KKT conditions of $\widetilde{\text{QP}}^B(\vartheta_j, y)$ are given by

$$J^B(\vartheta) \begin{bmatrix} \Delta v \\ \lambda_{b_{\text{dyn}}, j+1} \\ \lambda_{b_{\text{alg}}, j+1} \\ \lambda_{c, j+1} \end{bmatrix} + \begin{bmatrix} a(v_j, y) \\ b_{\text{dyn}}(v_j) \\ b_{\text{alg}}(v_j) \\ c(v_j, y) \end{bmatrix} = 0. \quad (7.60)$$

In order to reformulate this equation system, we introduce $\Delta\lambda_{b_{\text{dyn}}} = \lambda_{b_{\text{dyn}},j+1} - \lambda_{b_{\text{dyn}},j}$, $\Delta\lambda_{b_{\text{alg}}} = \lambda_{b_{\text{alg}},j+1} - \lambda_{b_{\text{alg}},j}$ and $\Delta\lambda_c = \lambda_{c,j+1} - \lambda_{c,j}$. This leads to the system

$$J^B(\vartheta) \begin{bmatrix} \Delta v \\ \Delta\lambda_{b_{\text{dyn}}} \\ \Delta\lambda_{b_{\text{alg}}} \\ \Delta\lambda_c \end{bmatrix} + F^B(\vartheta) = 0. \quad (7.61)$$

This shows that one Level B iteration refers to one Newton-type step towards the root of $F^B(\vartheta) = 0$ with $J^B(\vartheta)$ as an approximation of $\frac{d}{d\vartheta} F^B(\vartheta)$. \square

In contrast to F^D and F^C , the function F^B does not represent the KKT system of the original problem. However, the following theorem shows that the Level B iterates are still converging to a feasible point.

Theorem 7.6 (Limit of Modified Level B Iterations)

Let the online data be fixed, i.e. $y_k = y$ for all $k \in \mathbb{N}$. Let the feasibility iterations $\vartheta_j = (v_j, \lambda_{b_{\text{dyn}},j}, \lambda_{b_{\text{alg}},j}, \lambda_{c,j})$ generated by Algorithm 7.3 converge towards a limit $\vartheta^* = (v^*, \lambda_{b_{\text{dyn}}}^*, \lambda_{b_{\text{alg}}}^*, \lambda_c^*)$ and the primal QP solutions Δv_j converge to 0. Then the limit is a feasible point of the problem $\widetilde{\text{MHE}}(y)$.

Proof The Level B iterates are updated by

$$v_{j+1} = v_j + \Delta v, \quad \lambda_{b,j+1} = \lambda_b^{QP}, \quad \lambda_{c,j+1} = \lambda_c^{QP} \quad (7.62)$$

where $(\Delta v, \lambda_b^{QP}, \lambda_c^{QP})$ is the solution of $\widetilde{\text{QP}}^B(\vartheta_j, y)$. Therefore, from convergence of v_j towards v^* it follows that Δv_j converges towards 0. Because all arising functions and derivatives are continuous, $(\Delta v^*, \lambda_{b_{\text{dyn}}}^*, \lambda_{b_{\text{alg}}}^*, \lambda_c^*)$ with $\Delta v^* = 0$ is the primal dual solution of $\widetilde{\text{QP}}^B(\vartheta^*, y)$. The KKT conditions of $\widetilde{\text{QP}}^B(\vartheta^*, y)$

$$a(v^*, y) - \bar{B}_{\text{dyn}}^\top \lambda_{b_{\text{dyn}}}^* - B_{\text{alg}}(v^*)^\top \lambda_{b_{\text{alg}}}^* - C^\top \lambda_c^* = 0, \quad (7.63a)$$

$$b_{\text{dyn}}(v^*) = 0, \quad (7.63b)$$

$$b_{\text{alg}}(v^*) = 0, \quad (7.63c)$$

$$c(v^*, q) = 0 \quad (7.63d)$$

are satisfied. The primal feasibility conditions (7.63b) -(7.63d) show that v^* is feasible for the original problem $\widetilde{\text{MHE}}(y)$. This shows that Level B iterations are converging to a feasible point for fixed online data y . \square

Remark 7.8

The equation system (7.63) can be interpreted as the KKT system of a disturbed optimization problem. Let $\vartheta^* = (v^*, \lambda_{b_{\text{dyn}}}^*, \lambda_{b_{\text{alg}}}^*, \lambda_c^*)$ be the limit of the Level B iterations as defined in Theorem 7.6. Then ϑ^* is a KKT point of the following disturbed optimization problem

$$\mathbf{min}_v \frac{1}{2} \|\ell(v, y_k)\|_{\Sigma_k^{-1}}^2 + e^\top v \quad (7.64a)$$

$$\mathbf{s.t.} \quad b_{\text{dyn}}(v) = 0, \quad (7.64b)$$

$$b_{\text{alg}}(v) = 0, \quad (7.64c)$$

$$c(v, \mathbf{q}_k) = 0 \quad (7.64d)$$

with

$$e = \left(\bar{B}_{\text{dyn}}^\top - B_{\text{dyn}}^\top(v^*) \right) \lambda_{\text{dyn}}^*. \quad (7.65)$$

This problem is related to the original problem $\widetilde{\text{MHE}}(y)$, except the error term e . It shows that the distance of ϑ^* to the solution of original problem is dependent on quality of \bar{B}_{dyn} as an approximation of B_{dyn} .

We showed that Level X with $X \in \{D, C, B\}$ is a Newton-type method for a specific nonlinear function F^X with an Jacobian approximation J^X . Therefore, Level X corresponds to solving the linear equation system

$$J^X(\vartheta_j) \Delta \vartheta_j = -F^X(\vartheta_j) \quad (7.66)$$

iteratively with $\vartheta_{j+1} = \vartheta_j + \Delta \vartheta_j$ and a initial guess ϑ_0 . Assumption 7.1 ensures that this equation system has an unique solution. In order to prove the convergence to the solution of the system

$$F^X(\vartheta) = 0 \quad (7.67)$$

we refer to the following result.

Theorem 7.7 (Local convergence of Newton-type Methods)

Let $X \in \{D, C, B\}$ and $U \subseteq \mathbb{R}^{n_\vartheta}$. Let $F^X : U \rightarrow \mathbb{R}^{n_\vartheta}$ be continuous differentiable and let the approximation of the Jacobian $J^X : U \rightarrow \mathbb{R}^{n_\vartheta \times n_\vartheta}$ be continuous with a continuous inverse on U . Furthermore, let us make the following assumptions:

- There is a $\kappa < 1$ such that for all $\vartheta_1, \vartheta_2 \in U$

$$\left\| J^X(\vartheta_1)^{-1} \left(J^X(\vartheta_2) - \frac{dF^X}{d\vartheta}(\vartheta_2) \right) \right\| \leq \kappa. \quad (7.68)$$

- There is a $\omega \in \mathbb{R}$ such that for all $\vartheta_1, \vartheta_2, \vartheta_3 \in U$

$$\left\| J^X(\vartheta_1)^{-1} (J^X(\vartheta_2) - J^X(\vartheta_3)) \right\| \leq \omega \|\vartheta_2 - \vartheta_3\|. \quad (7.69)$$

- The initial step $\Delta\vartheta_0$ is sufficiently small that

$$\delta_0 := \kappa + \frac{\omega}{2} \|\Delta\vartheta_0\| < 1 \quad (7.70a)$$

and that

$$U_0 := \left\{ \vartheta \in \mathbb{R}^{n_\vartheta} \mid \|\vartheta - \vartheta_0\| \leq \frac{\|\Delta\vartheta_0\|}{1 - \delta_0} \right\} \subset U. \quad (7.70b)$$

Under these conditions the sequence of Newton-type iterates defined by Equation (7.66) remains in U_0 and converges to a ϑ^* with $F^X(\vartheta^*) = 0$.

Proof The proof can be found for example in [13] or [24]. □

Theorem 7.7 is a well known result from the convergence theory of Newton-type methods. Assumption (7.68) characterizes the quality of the Jacobian approximation and Assumption (7.69) ensures that the derivative of J^X is bounded. Equation (7.70) makes sure that the initial guess ϑ_0 is close enough to the solution ϑ^* . Under these assumptions, Theorem 7.7 ensures that the Newton-type method converges.

7.4.4 Combined MLI-MHE Schemes

To utilize the accurate linearizations of the higher levels and the fast response times of the lower levels at the same time, the proposed levels are operated simultaneously. Each level is treated as an independent component and is working on its own set of iterates and reference values. This allows to parallelize the evaluation on multi-core CPUs. Every level can be operated on its own CPU core without affecting the computation time of the other levels. There are many ways to combine the different MLI levels to schemes. The most common approach is to apply the levels in a predefined sequence, as described in Section 5.2. Nevertheless, it is also possible to choose online which level is applied based on the contraction rate [113] or on the actual

computation time [103]. These approaches refer to Adaptive Level Choice by Contractivity (ALC-Con) and Adaptive Level Choice by Computation Time (ALC-Time) from MLI for NMPC. In this section, we restrict ourselves to MLI-MHE schemes, where Level D is combined with either Level C or Level B in a repeating sequence as depicted in Figure 7.5. Level D is operated on a lower sampling rate than the actual sampling rate of the process. Between Level D updates, intermediate feedback is generated by the lower level (C or B) with the original sampling rate. After the feedback generation of Level D is finished, the resulting QP data is communicated to the lower level and is used as reference values in the following sampling intervals. This communication method is referred as *top-down communication*. We consider a MLI-MHE scheme as *real-time feasible*, when the preparation phase of every involved level is below its sampling time. Thereby, we assume that every level is running on a dedicated CPU core, which is not influenced by other processes. MLI schemes with a fixed level sequence are described in detail in Chapter 5 and in [113].

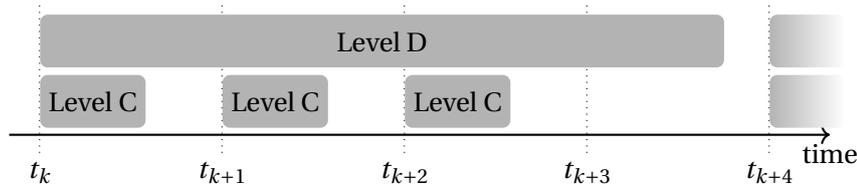


Figure 7.5: Exemplary MLI scheme with one Level D update every fourth sampling point followed by three intermediate Level C updates (CD^4). The gray area denotes, when the corresponding level is busy.

Remark 7.9

The level sequences are commonly indicated by the notation $B^{n_B} C^{n_C} D^{n_D}$. The indices denote after how many sampling intervals the corresponding level is scheduled. The letters for unused levels are omitted. For example, the regular RTI-MHE scheme is denoted by D^1 .

7.5 Arrival Cost Update Procedure

The MHE problem at time t_k considers a limited time horizon $[t_{k-M}, t_k]$ and measurements prior to t_{k-M} are not considered. The purpose of the arrival cost objective

term (7.3a)

$$\frac{1}{2} \|\bar{x}_{k-M} - s_0^x\|_{\Sigma_{\text{arr},k-M}^{-1}}^2 \quad (7.71)$$

is to summarize measurement information originating from the time before the current estimation horizon $[t_{k-M}, t_k]$. The theoretical ideal arrival cost considers all older measurements and leads to the problem

$$\min_{s^x, s^z, w} \frac{1}{2} \sum_{i=-\infty}^{k-M-1} (\|\eta_i - h(s_i^x, s_i^z)\|_{\Sigma_{\text{meas}}^{-1}}^2 + \|w_i\|_{\Sigma_{\text{proc}}^{-1}}^2) \quad (7.72a)$$

s.t.

$$s_{i+1} = x(s_i^x, s_i^z, q_i) + w_i, \quad i = -\infty, \dots, k-M-1, \quad (7.72b)$$

$$0 = g(s_i^x, s_i^z, q_i), \quad i = -\infty, \dots, k-M. \quad (7.72c)$$

Practically, the solution of this problem cannot be computed. In addition, we want to avoid that an infinite number of old measurements contributes to the arrival cost. In the following, we describe a method to update the arrival cost efficiently. First, we will consider the arrival cost for the RTI-MHE. Afterwards, we propose a new methodology, on how to use the MLI reference variables to update the arrival cost term with low additional effort. Originally, we have proposed this method in [47].

Online Data Independent Arrival Cost Update Preparation

As suggested in [28], we want to replace the ideal arrival cost by a quadratic penalty term which is updated in every MHE iteration. In order to derive the update procedure of the arrival cost, we consider the transition from problem $\widetilde{\text{MHE}}(y_k)$ to problem $\widetilde{\text{MHE}}(y_{k+1})$. Ideally, we want to solve the extended problem

$$\min_{s^x, s^z, s^q, w} \frac{1}{2} \|\bar{x}_{k-M} - s_0^x\|_{\Sigma_{\text{arr},-M}^{-1}}^2 \quad (7.73a)$$

$$+ \frac{1}{2} \sum_{i=0}^{M+1} (\|\eta_{k-M+i} - h(s_i^x, s_i^z)\|_{\Sigma_{\text{meas}}^{-1}}^2 + \|w_i\|_{\Sigma_{\text{proc}}^{-1}}^2) \quad (7.73b)$$

s.t.

$$s_{i+1}^x = x(s_i^x, s_i^z, s_i^q) + w_i, \quad i = 0, \dots, M, \quad (7.73c)$$

$$0 = g(s_i^x, s_i^z, s_i^q), \quad i = 0, \dots, M+1, \quad (7.73d)$$

$$s_i^q = q_i, \quad i = 0, \dots, M+1 \quad (7.73e)$$

on the time horizon $[t_{k-M}, t_{k+1}]$ with $s^x \in \mathbb{R}^{(M+2)n_x}$, $s^z \in \mathbb{R}^{(M+2)n_z}$, $s^q \in \mathbb{R}^{(M+2)n_q}$ and $w \in \mathbb{R}^{(M+1)n_x}$. This problem considers all past measurements, but its time horizon is growing with every iteration. In order to prevent a constantly growing time horizon, we want to formulate a problem on the time horizon $[t_{k-M+1}, t_{k+1}]$ that is similar to the extended Problem (7.73). The measurement at t_k and the previous arrival cost should be summarized in an updated arrival cost term.

Remark 7.10

In the following, we denote the subvector $(s_i^x, \dots, s_i^y) \in \mathbb{R}^{(j-i+1)n_x}$ of s^x by $s_{i:j}^x$. A similar notation is applied for s^z, s^q and w .

Principle of Optimality for State Estimation

In order to eliminate the variables s_0^x, s_0^z, s_0^q from the Problem (7.73), we are using a similar idea as the *principle of optimality* for optimal control. The principle of optimality is an approach to solve optimal control problems with a dynamic programming algorithm. It is explained in great detail in [9]. The basic idea is to split up an optimal control problem in sub intervals and optimize the system step-wise backwards in time. We apply a similar idea to MHE by defining the nonlinear function

$$\text{Arr}(s_1^x) = \min_{s_0^x, s_0^z, s_0^q} \frac{1}{2} \|\bar{x}_{k-M} - s_0^x\|_{\Sigma_{arr, k-M}^{-1}}^2 + \frac{1}{2} \|s_1^x - x(s_0^x, s_0^z, s_0^q)\|_{\Sigma_{proc}^{-1}}^2 \quad (7.74a)$$

$$+ \frac{1}{2} \|\eta_{k-M} - h(s_0^x, s_0^z)\|_{\Sigma_{meas}^{-1}}^2 \quad (7.74b)$$

s.t.

$$0 = g(s_0^x, s_0^z, s_0^q), \quad (7.74c)$$

$$0 = s_0^q - q_{k-M}. \quad (7.74d)$$

Using this function, we define the problem

$$\min_{\substack{s_{1:M+1}^x, s_{1:M+1}^z, \\ s_{1:M+1}^q, w_{1:M}}} \text{Arr}(s_1^x) + \frac{1}{2} \sum_{i=1}^{M+1} \left(\|\eta_{k-M+i} - h(s_i^x, s_i^z)\|_{\Sigma_{meas}^{-1}}^2 + \|w_i\|_{\Sigma_{proc}^{-1}}^2 \right) \quad (7.75a)$$

s.t.

$$s_{i+1}^x = x(s_i^x, s_i^z, s_i^q) + w_i, \quad i = 1, \dots, M, \quad (7.75b)$$

$$0 = g(s_i^x, s_i^z, s_i^q), \quad i = 1, \dots, M+1, \quad (7.75c)$$

$$s_i^q = q_{k-M+i}, \quad i = 1, \dots, M+1 \quad (7.75d)$$

which refers to the shifted time horizon $[t_{k-M+1}, t_{k+1}]$. This problem does not explicitly depend on s_0^x, s_0^z, s_0^q anymore. However, the following theorem shows that it is equivalent to the extended Problem (7.73).

Theorem 7.8

Let $(s_{1:M+1}^x, s_{1:M+1}^z, s_{1:M+1}^q, w_{1:M})$ be a solution of (7.75) and let (s_0^x, s_0^z, s_0^q) be a solution of $\text{Arr}(s_1^x)$. Then $v = (s^x, s^z, s^q, w)$ with

$$w_0 = s_1^x - x(s_0^x, s_0^z, s_0^q). \quad (7.76)$$

is a solution of Problem (7.73).

Proof In a first step, we show that v is a feasible point for Problem (7.73). The constraints (7.73d) and (7.73e) are satisfied, because they emerge directly in Problem (7.75) and in $\text{Arr}(s_1^x)$. The matching conditions (7.73c) are satisfied, because they emerge directly in Problem (7.75) for $i = 1, \dots, M$. The matching condition for $i = 0$ is satisfied by the definition of w_0 . Therefore, all constraints of Problem (7.73) are satisfied and v is feasible.

In a second step, we show that v is a solution of (7.73). Lets assume that v is not the solution (7.73), i.e. there is a feasible point $\tilde{v} = (\tilde{s}^x, \tilde{s}^z, \tilde{s}^q, \tilde{w})$ of Problem (7.73) with a lower objective function value. Then the following holds

$$\begin{aligned} \text{Arr}(\tilde{s}_1^x) + \frac{1}{2} \sum_{i=1}^{M+1} \left(\|\eta_{k-M+i} - h(\tilde{s}_i^x, \tilde{s}_i^z)\|_{\Sigma_{\text{meas}}^{-1}}^2 + \|\tilde{w}_i\|_{\Sigma_{\text{proc}}^{-1}}^2 \right) &\leq \\ \frac{1}{2} \|\bar{x}_{k-M} - \tilde{s}_0^x\|_{\Sigma_{\text{arr},-M}^{-1}}^2 + \frac{1}{2} \sum_{i=0}^{M+1} \left(\|\eta_{k-M+i} - h(\tilde{s}_i^x, \tilde{s}_i^z)\|_{\Sigma_{\text{meas}}^{-1}}^2 + \|\tilde{w}_i\|_{\Sigma_{\text{proc}}^{-1}}^2 \right) &< \\ \frac{1}{2} \|\bar{x}_{k-M} - s_0^x\|_{\Sigma_{\text{arr},-M}^{-1}}^2 + \frac{1}{2} \sum_{i=0}^{M+1} \left(\|\eta_{k-M+i} - h(s_i^x, s_i^z)\|_{\Sigma_{\text{meas}}^{-1}}^2 + \|w_i\|_{\Sigma_{\text{proc}}^{-1}}^2 \right) &= \\ \text{Arr}(s_1^x) + \frac{1}{2} \sum_{i=1}^{M+1} \left(\|\eta_{k-M+i} - h(s_i^x, s_i^z)\|_{\Sigma_{\text{meas}}^{-1}}^2 + \|w_i\|_{\Sigma_{\text{proc}}^{-1}}^2 \right). &\quad (7.77) \end{aligned}$$

The last equality is satisfied since (s_0^x, s_0^z, s_0^q) is a solution of $\text{Arr}(s_1^x)$ and by the definition of w_0 . This shows that $(\tilde{s}_{1:M+1}^x, \tilde{s}_{1:M+1}^z, \tilde{s}_{1:M+1}^q, \tilde{w}_{1:M})$ is a feasible point for Problem (7.75) with a lower objective function value and this is a contradiction to the prerequisite that $(s_{1:M+1}^x, s_{1:M+1}^z, s_{1:M+1}^q, w_{1:M})$ is a solution of Problem (7.75). \square

Theorem 7.8 shows that we can equivalently solve Problem (7.75) instead of Problem (7.73). The advantage of Problem (7.75) is that it does not directly depend on

(s_0^x, s_0^z, s_0^q) . However, it still depends implicitly on (s_0^x, s_0^z, s_0^q) by the nonlinear function $\text{Arr}(s_1^x)$. In order to eliminate (s_0^x, s_0^z, s_0^q) completely, we approximate $\text{Arr}(s_1^x)$ by a quadratic penalty term.

Approximation of Ideal Arrival Cost

The nonlinear function $\text{Arr}(s_1^x)$ is defined by an NLP with the optimization variables (s_0^x, s_0^z, s_0^q) . Therefore, the exact evaluation of $\text{Arr}(s_1^x)$ may require a lot of iterations in order to achieve convergence. Especially, the NLP depends on the nonlinear functions $x(\cdot)$, $h(\cdot)$ and $g(\cdot)$, which are expensive to evaluate. In order to save computation time, we approximate $\text{Arr}(s_1^x)$ by an equality constrained QP, which is significantly cheaper to solve. Since we are looking on the transition from $\widehat{\text{MHE}}(y_k)$ at point t_k to the next problem $\widehat{\text{MHE}}(y_{k+1})$ at point t_{k+1} , we already have an approximate solution $\bar{s} = (\bar{s}^x, \bar{s}^z, \bar{s}^q)$ of $\widehat{\text{MHE}}(y_k)$ available. We transform the NLP into an equality constrained QP by linearizing the functions $x(\cdot)$, $h(\cdot)$ and $g(\cdot)$ at $\bar{s}_0 = (\bar{s}_0^x, \bar{s}_0^z, \bar{s}_0^q)$. This leads to the problem

$$\min_{s_0=(s_0^x, s_0^z, s_0^q)} \frac{1}{2} \|\bar{x}_{k-M} - s_0^x\|_{\Sigma_{\text{arr}, k-M}^{-1}}^2 \quad (7.78a)$$

$$+ \frac{1}{2} \left\| x(\bar{s}_0) + \frac{dx(\bar{s}_0)}{ds}(s_0 - \bar{s}_0) - s_1^x \right\|_{\Sigma_{\text{meas}}^{-1}}^2 \quad (7.78b)$$

$$+ \frac{1}{2} \left\| h(\bar{s}_0^x, \bar{s}_0^z) + \frac{dh(\bar{s}_0^x, \bar{s}_0^z)}{d(s^x, s^z)} \begin{pmatrix} s_0^x - \bar{s}_0^x \\ s_0^z - \bar{s}_0^z \end{pmatrix} - \eta_{k-M} \right\|_{\Sigma_{\text{proc}}^{-1}}^2 \quad (7.78c)$$

s.t.

$$0 = g(\bar{s}_0) + \frac{dg(\bar{s}_0)}{ds}(s_0 - \bar{s}_0), \quad (7.78d)$$

$$0 = s_0^q - q_{k-M}. \quad (7.78e)$$

Since the DAE system is of index 1, the derivative $\frac{dg}{ds^z}(\bar{s})$ is nonsingular and constraint (7.78d) can be used to eliminate the algebraic variable s_0^z . The CTE constraint (7.78e) allows to substitute the applied control variable s_0^q . The result is a linear least squares problem in s_0^x dependent on s_1^x as parameter. The minimum of this linearized problem can analytically be expressed as

$$\text{Arr}(s_1^x) \approx \frac{1}{2} \left\| s_1^x - \bar{x}_{k+1-M} \right\|_{\Sigma_{\text{arr}, k+1-M}^{-1}}^2. \quad (7.79)$$

The matrix $\bar{\Sigma}_{\text{arr},k-M+1}$ and the state \bar{x}_{k-M+1} can be calculated explicitly using the above objective and constraint linearizations. For a detailed description of these calculations we refer the reader to [59, 28].

Multi-Level Arrival Cost Update

The transformation of Problem (7.78) into the quadratic penalty term (7.79) can be performed algebraically using a QR transformation. The most expensive part in the course of this arrival cost update is to evaluate the linearization $\frac{dx}{ds}(\tilde{s}_0)$, which requires the sensitivity generation for the solution of the DAE system. In contrast to that, the evaluation of $\frac{dg}{ds}(\tilde{s})$ and $\frac{dh}{ds}(\tilde{s})$ requires a comparatively low computational effort, since it includes only algebraic operations. Since in Level D all the derivatives are reevaluated anyhow, the additional effort of evaluating $\frac{dx}{ds}(\tilde{s})$ is low. In contrast to that, Level C or B require less computations and the evaluation of $\frac{dx}{ds}(\tilde{s})$ may exceed the effort of the other computations. Therefore, a complete evaluation is in general not advisable. We propose to reuse the reference matrix of the corresponding level instead. Since we are shifting also the reference block matrices associated with the primal variables, we already have an approximation of the derivative $\frac{dx}{ds}(\tilde{s})$ available. Lets consider the transition from $\overline{\text{MHE}}(y_k)$ to $\overline{\text{MHE}}(y_{k+1})$ with a Level X update for $X \in \{C, B\}$. We denote the primal solution of $\overline{\text{MHE}}(y_k)$ by $\tilde{v} = (\tilde{s}^x, \tilde{s}^z, \tilde{s}^q, w)$. Level X for $\overline{\text{MHE}}(y_k)$ holds a reference matrix

$$\bar{B}_{\text{dyn},k} = \begin{bmatrix} \bar{G}_k^x & \bar{G}_k^z & \bar{G}_k^q & \mathbb{1} \end{bmatrix}, \quad (7.80)$$

with

$$\bar{G}_k^x = \begin{bmatrix} \bar{G}_{0,k}^x & -\mathbb{1} & & \\ & \ddots & \ddots & \\ & & \bar{G}_{M-1,k}^x & -\mathbb{1} \end{bmatrix}, \quad \bar{G}_k^z = \begin{bmatrix} \bar{G}_{0,k}^z & & & \\ & \ddots & & \\ & & \bar{G}_{M-1,k}^z & 0 \end{bmatrix}, \quad (7.81a)$$

$$\bar{G}_k^q = \begin{bmatrix} \bar{G}_{0,k}^q & & & \\ & \ddots & & \\ & & \bar{G}_{M-1,k}^q & 0 \end{bmatrix}. \quad (7.81b)$$

Thereby, the block matrices $\bar{G}_{i,k}^x, \bar{G}_{i,k}^z, \bar{G}_{i,k}^q$ for $i = 0, \dots, M-1$ are approximations of the partial derivatives

$$\bar{G}_{i,k}^x \approx \frac{d}{ds_i^x} x(\tilde{s}_i^x, \tilde{s}_i^z, \tilde{s}_i^q), \quad \bar{G}_{i,k}^z \approx \frac{d}{ds_i^z} x(\tilde{s}_i^x, \tilde{s}_i^z, \tilde{s}_i^q), \quad \bar{G}_{i,k}^q \approx \frac{d}{ds_i^q} x(\tilde{s}_i^x, \tilde{s}_i^z, \tilde{s}_i^q). \quad (7.82)$$

The structure of the reference matrix $\bar{B}_{\text{dyn},k}$ is explained in detail in Section 7.4. We propose the use the block matrices $\bar{G}_{0,k}^x, \bar{G}_{0,k}^z, \bar{G}_{0,k}^q$ as an approximation of

$$\frac{d}{ds}x(\bar{s}_0^x, \bar{s}_0^z, \bar{s}_0^q) \approx \begin{bmatrix} \bar{G}_{0,k}^x & \bar{G}_{0,k}^z & \bar{G}_{0,k}^q \end{bmatrix} \quad (7.83)$$

instead of an exact evaluation. Since these matrices are already available, there is no additional effort for the sensitivity generation necessary. Note that we work with fixed level sequences. We always know at which sampling time we use the prepared linearizations for updating the arrival cost. However, the CTE constraint (7.78e) eliminates the necessity of knowing the exact sampling time at which the prepared linearizations will be used to update the arrival cost. Thus, the presented update procedure could also be applied to an MLI-MHE scheme with automated level choice, see also Remark 7.5.

Chapter 8

Moving Horizon Estimation for Secondary Microgrid Control

In this section, we demonstrate the capabilities of the presented Moving Horizon Estimation (MHE) algorithm. To that end, we apply a challenging load scenario to the model of a test Microgrid (MG). We apply different Multi-Level Iteration (MLI) schemes in an open and closed loop setting. To evaluate the performance, we compare the estimation error as well as the computation times.

8.1 Microgrid Model

In order to test the presented MHE algorithm, we consider the same MG model as in Section 6.2. It is a 6-bus system with a typical topology for islanded microgrids. The grid comprises two identical Diesel Generators (DGs), a Battery (BA), a Photovoltaic (PV) plant, and a passive PQ-load. In this Chapter, we focus on the model aspects that are relevant for the estimator. A sketch of the topology is depicted in Figure 8.1. For a comprehensive description of the grid with the individual components and the defining numerical parameters we refer the reader to [84]. The individual model components are presented in Chapter 3.

The buses as well as the transmission lines are modeled by the nonlinear algebraic Alternating Current (AC) power flow equations in a rotating dq -reference frame. Every node is described by the active and the reactive power balance P and Q , the volt-

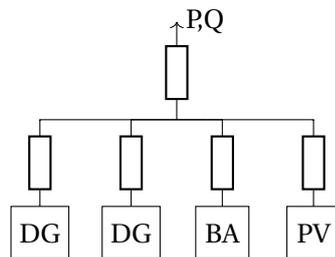


Figure 8.1: Topology of the test MG.

age V , and the phase angle θ . We assume that all these quantities are measurable. The base power of the MG is $S_{\text{grid}} = 100\text{kVA}$.

The DGs consist of a synchronous generator actuated by a diesel engine with a governor for frequency stabilization (IEEE DEGOV1) and are equipped with an Automatic Voltage Regulator (IEEE AC5A). The measurable states for the generator are the terminal voltages u_{dq} and currents i_{dq} , as well as the frequency ω_{DG} , and the excitation current i_{fd} . The control inputs of the generators are the setpoints for frequency ω_{ref} and voltage V_{ref} . The nominal power of the generator is $S_{\text{DG}} = 325\text{kVA}$.

The battery is modeled as a constant DC voltage source connected to an inverter with an internal droop. It is controlled by the setpoints for frequency $\omega_{\text{BA}}^{\text{ref}}$ and voltage $V_{\text{BA}}^{\text{ref}}$. Besides the frequency ω_{BA} , the state of charge is assumed to be measurable. The nominal power of the battery is $S_{\text{BA}} = 150\text{kVA}$.

The PV system and the load are modeled as time-varying active and reactive power infeeds. Disturbances and load changes emerge as jumps in the corresponding states P_{load} , Q_{load} , P_{PV} and Q_{PV} .

The complete MG is given as a DAE system of index 1 with 37 differential and 42 algebraic states and 6 control inputs. The system is highly nonlinear and stiff. In total, 10 of 37 differential and 28 of 42 algebraic states are measurable. The measurement function h is the projection of the vector of all differential and algebraic states to the measurable states.

All process noise and measurement errors are assumed to be time-independent and uncorrelated, which leads to diagonal matrices Σ_{meas} and Σ_{proc} . The active and reactive power at the load are unpredictable time-varying states with process state noise variance 10^{-2} . In addition, we assume that there is no model-plant mismatch and therefore we set the variance of the process state noise of the remaining states to zero. To prevent numerical issues, we eliminate the corresponding variables from the MHE problem. The variance of the measurement errors is set to 10^{-1} for all measurable states.

8.1.1 Scenario and Objectives of Control

We analyze the performance of the estimator in a challenging load scenario. At the beginning, the system is in steady state and the load $P_{\text{load}} = 5\text{ p.u.}$ and $Q_{\text{load}} = 1\text{ p.u.}$ is shared equally by the generators. The battery does not contribute to load sharing. After 1 s, a sudden unscheduled load step of 40% in active and reactive power takes place, which exceeds the capacity of the generators. To ensure that the operational limits are satisfied, the battery needs to serve the missing load. The simulation has an overall length of 4 s.

The main objective of secondary microgrid control is to ensure a stable frequency $\omega = 1$ p.u. and constant voltage at the load $V_{\text{load}} = 1$ p.u.. Since the individual components of the microgrid are equipped with local primary controllers, the system may find a new stable operating point after a disturbance, but an offset to the nominal operation point is introduced. The secondary controller is supposed to restore the desired frequency and voltage by adjusting the reference values of the primary controllers. Furthermore, the secondary controller is responsible for proper load sharing.

8.2 Numerical Results

In this section, we provide numerical results for the presented MHE state estimator. We compare three exemplary MLI-MHE schemes: the MHE-RTI, a CD^4 and a BD^4 scheme. We also include the results from an ideal MHE state estimator that, unlike the other schemes, solves every MHE optimization problem until convergence. We investigate the performance in an open loop as well as in a closed loop setting. In both scenarios the estimator takes the last five measurements into consideration and the estimation horizon is 400ms. The sampling time is 100 ms, which is inherited from the controller in the closed loop setting. The initial variances of the states P_{load} and Q_{load} are set to 10^{-2} and the variance of the remaining states is set to 10^{-5} . The initial covariance matrix Σ_{arr} is a diagonal matrix with the corresponding values as diagonal entries.

All computations were carried out with the Nonlinear Model Predictive Control (NMPC) framework MLI [113] written in MATLAB. For integration and sensitivity generation the SolvIND integrator suite was used which implements an adaptive BDF-method [3]. The Quadratic Programms (QPs) were solved with qpOASES [34], which implements an online active set strategy. All computations were performed on a 64bit Ubuntu 20.04 Linux machine with 16 GB of RAM and Intel i7-9700 CPU @ 3.00 GHz CPU.

We do not consider the computation times of the QP solver, because we use the assumption that feedback is immediately available. The QPs are equality constrained and solving the Karush-Kuhn-Tucker (KKT) system with the backslash operator in MATLAB requires less than 15 ms. Since this method neither exploits the KKT structure nor the shooting structure of the system, we expect that this value can be reduced significantly. In addition, we proposed in Section 7.4 to perform the evaluation and derivative generation of the algebraic consistency conditions in QP 7.20 in the feedback phase. This increases the required computation time for the feedback phase for this specific example by less than 1.5 ms, which is a small value compared

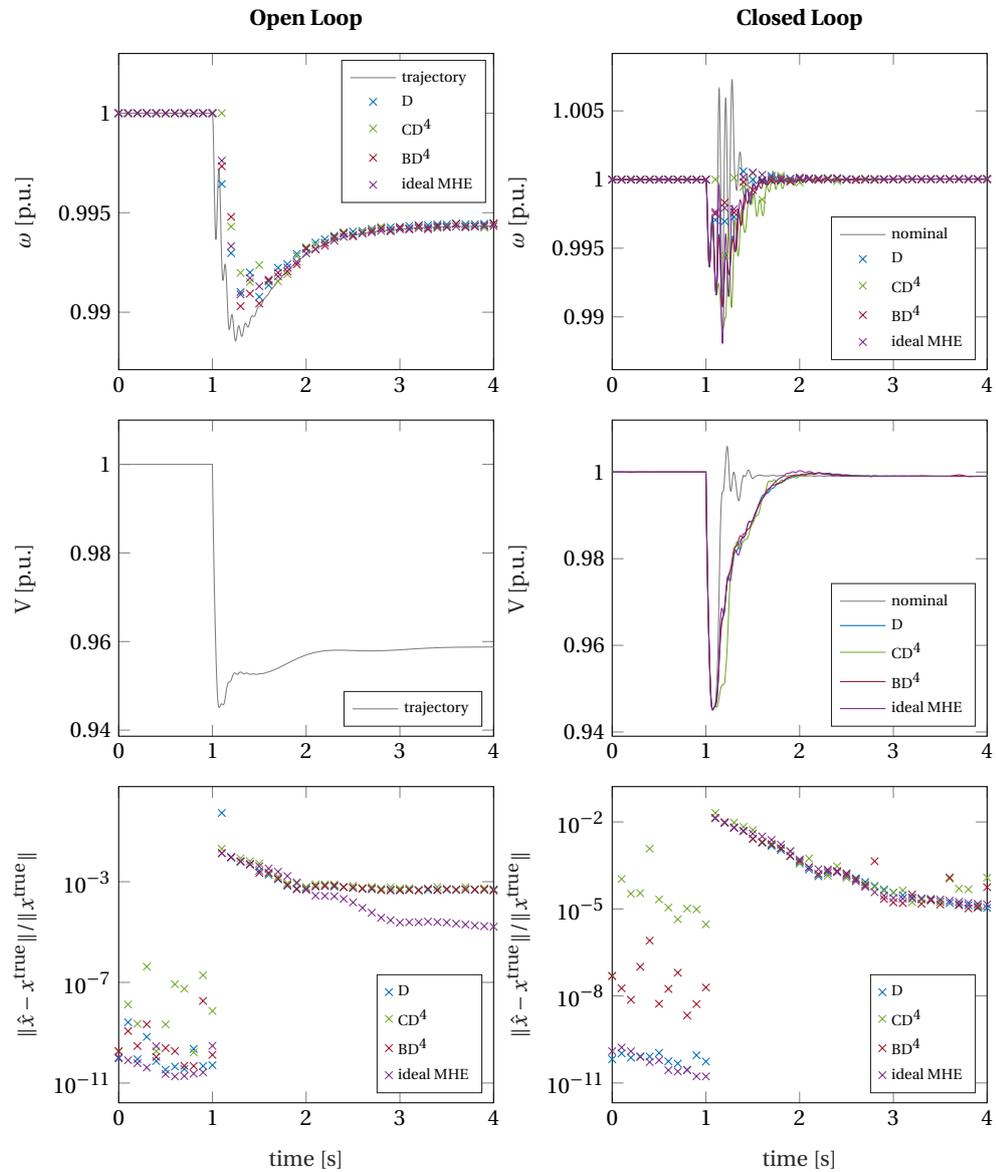


Figure 8.2: Computational results for open and closed loop simulations. In the top row, the trajectories for the frequency are shown and the state estimates of the different MLI-MHE schemes are indicated by the crosses. In the middle, the voltage at the load is depicted. The relative error of the full state estimates for the different schemes is shown in the bottom row.

to the overall sampling time. The assumption of immediate feedback is justified because the overall computational effort of the feedback phase is small compared to the sampling time of the process.

8.2.1 Open Loop Scenario

In the open loop scenario, the control inputs for the primary controllers are kept constant over the whole simulation horizon. At the start of the simulation the reference points for the primary controllers are set to maintain the desired operating point $\omega = 1 \text{ p.u.}$ and $V_{\text{load}} = 1 \text{ p.u.}$ under the given load. After the disturbance at $t = 1 \text{ s}$, the system is in a transient phase, but gets stabilized by the primary controllers. However, the resulting stable frequency and voltage has a significant offset to the nominal value which is a suboptimal operating condition of the microgrid. Since the microgrid is modeled in a dq -reference frame, which is rotating with a constant angular velocity of 1 p.u. , the dynamical system is not decaying to a steady state. On the left hand side of Figure 8.2, the frequency and the voltage at the load is depicted. We show only these two states since controlling the frequency and voltage is the primary objective of microgrid control. As the voltage of the system is modeled by an algebraic state, it is not part of the state estimate and therefore only the simulation results are shown. The bottom left plot shows the relative error of the estimates to the true trajectory. In the beginning, the MHE iteration variables are initialized in the steady state solution, which leads to a very low estimation error. The load jump at $t = 1 \text{ s}$ is a disturbance which is not complied with the dynamical model and instead leads to the introduction of a process state noise contribution. Because the objective function penalizes process state noise an estimation error is introduced for all schemes. After the load jump, the estimation error is decreased by all schemes. However, since the system is not in a steady state at the end of the simulation, an estimation error remains for the presented MLI schemes. In contrast to that, the ideal MHE is iterating until convergence and is therefore able to reduce the estimation error to the range of the integration tolerance.

On the left hand side of Figure 8.3, the computation times of the different schemes are visualized. During steady state operation of the microgrid, the computation times are low. In the transient phase, the computation times increase for all levels because the integrator needs more steps to achieve the desired precision. The maximal computation times in Table 8.1 show that the presented MHE schemes CD^4 and BD^4 are real-time feasible.

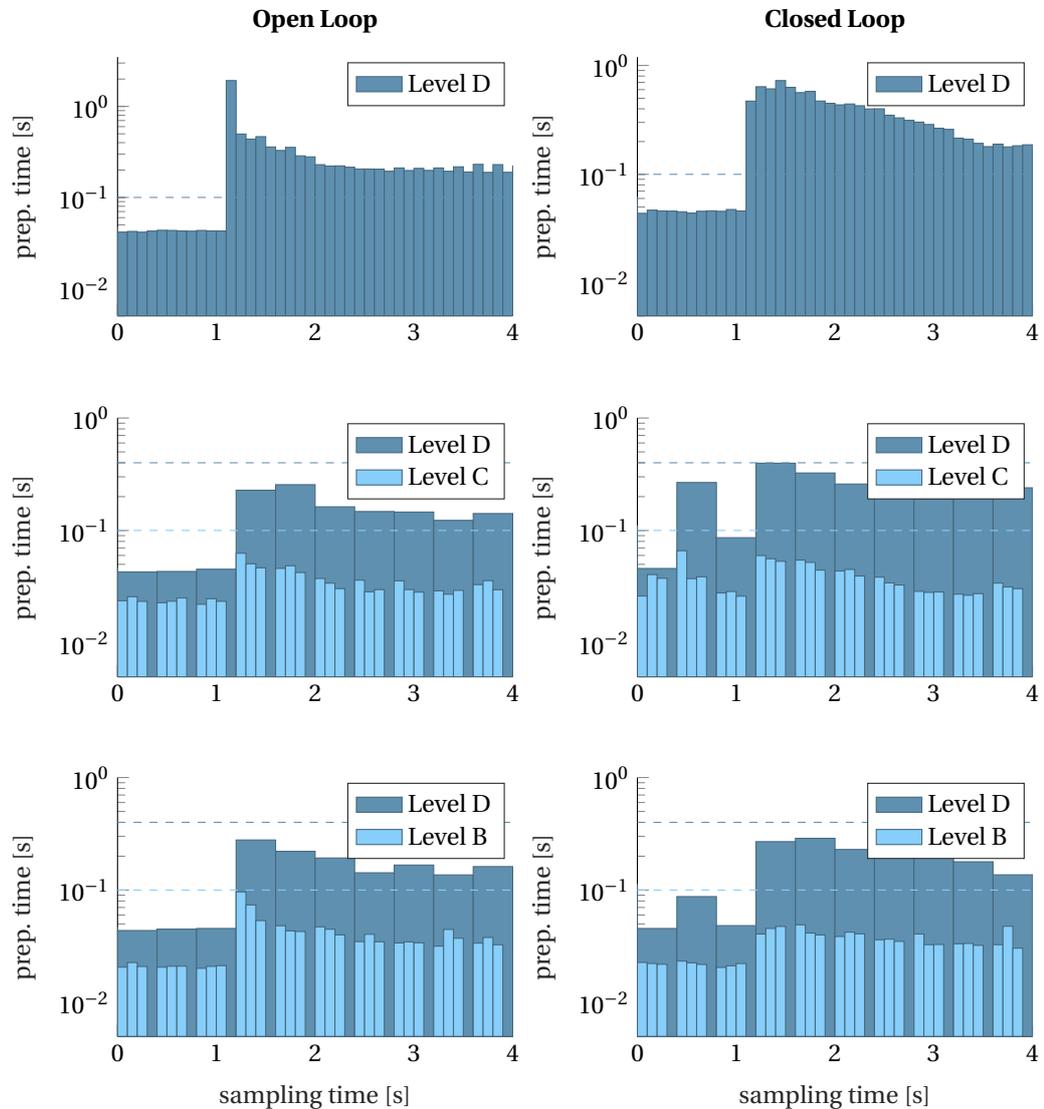


Figure 8.3: Computation times for different MLI-MHE schemes in an open loop and in a closed loop scenario. From top to bottom, computation times for a D^1 estimator, a CD^4 schema estimator, and a BD^4 schema estimator are shown. The elapsed computation times are depicted by the heights of the bars while the width indicates during which sampling intervals the computations were performed. The dashed lines mark the maximal allowed computation time for the different levels to stay real-time feasible.

		open loop [ms]		closed loop [ms]	
		max	median	max	median
MHE-RTI	<i>D</i>	1923	194	726	186
CD^4	<i>D</i>	255	46	396	104
	<i>C</i>	62	29	61	35
BD^4	<i>D</i>	278	47	288	94
	<i>B</i>	96	34	49	33

Table 8.1: Computation times of MHE algorithm for different scenarios and schemes.

8.2.2 Closed Loop Scenario

In the closed loop scenario, the setpoints are repeatedly adjusted by a secondary controller in order to steer the frequency and the voltage back to the nominal operating point. Here, we use an RTI-NMPC tracking controller as described in [103]. The controller does not receive the exact system state, but an estimate provided by the different MHE schemes. The right hand side of Figure 8.2 shows the frequency, the voltage at the load and the relative error to the true trajectory. In addition, the trajectory of a nominal RTI-NMPC controller, which always receives the current system state without an error, is depicted in grey. At the beginning of the simulation all schemes are initialized in the solution of the MHE problem and have therefore a very low estimation error during the steady state operation. An interesting observation is that the relative estimation error of the CD^4 scheme is slightly higher than the error of the BD^4 scheme which may indicate that the dual variable shift-initialization procedure, described in Section 7.4.1, causes a small residuum in the KKT conditions of the MHE problem and leads to a correction in the feedback phase.

The disturbance at $t = 1$ s causes a drop in frequency and voltage and all schemes introduce an estimation error. This is caused by the MHE objective function, which penalizes process state noise and leads to a delayed reaction by the controller. However, the estimates are accurate enough to enable the controller to steer the system back to the nominal operating point. In contrast to the open loop scenario, the system approaches a steady state and the estimation error decreases to the level of the integration tolerance. On the right hand side of Figure 8.3, the computation times of the different schemes are visualized over time. Similar to the open loop scenario, the computation times spike during the transient phase and the CD^4 and BD^4 schemes are real-time feasible.

The results show that the proposed MHE estimator is able to provide the controller with state estimates accurately enough to steer frequency and voltage back to the de-

sired operating point. In comparison to the nominal controller, the estimation error of the MHE schemes lead to less aggressive control actions and it takes to more time to stabilize the system. This is caused by the estimation error of all schemes, since the objective function penalizes jumps in the states for the load. However, the performance of all schemes is significantly better then the performance of a traditional control setup based on PI-controllers, which has usually a settling time of over 10s [103]. The presented MLI updates make the MHE algorithm real-time feasible, without affecting the control performance significantly in comparison to the ideal MHE.

Chapter 9

Conclusion

This thesis is dedicated to the development of new mathematical methods and algorithms for model-based feedback control of dynamical processes. Thereby, we focused on processes modeled by Ordinary Differential Equations (ODEs) or Differential Algebraic Equations (DAEs). Model Predictive Control (MPC) served as the main algorithmic framework for process control. In MPC, the time is divided into fixed sampling intervals. At every sampling point, the control is adjusted to the current system state by the solution of an optimization problem. We followed a direct approach. The arising Optimal Control Problems (OCPs) were discretized by multiple shooting to receive a family of structured Nonlinear Programming Problems (NLPs) parameterized by the current system state. The Real-Time Iteration (RTI) was used to exploit the structure of the NLPs and provide feedback efficiently.

The algorithmic developments of this thesis are motivated by the arising challenges of Microgrid (MG) control. The dynamics of MGs are stiff, extremely fast, and impacted by the high volatility of Renewable Energy Resources (RES). In order to achieve a high control performance, exceedingly low sampling times and highly accurate model predictions are necessary at the same time. Hence, the main challenge of MPC for MG control is to find solutions of the computationally demanding OCPs in a split second. A model of a MG is composed of a model of the grid topology and models for the connected components. We reviewed the Alternating Current (AC) power flow equations as a representation of the electrical grid and we described full transient models for the most important components in MGs. We integrated these models into the context of MPC by the definition of a suitable parametric OCP. The resulting MPC algorithm is tailored to the hierarchical control structure of MGs and is acting on the secondary layer.

A full feedback control loop comprises two main algorithmic components: the controller and the state estimator. These components are structurally related and can be approached by a similar set of mathematical tools. This thesis presented novel mathematical approaches for both components and showed their capabilities in numerical experiments.

For process control, we reviewed the Multi-Level Iteration (MLI) as an extension of the RTI. It is a framework for MPC, which is tailored to enable a high sampling frequency with a low feedback delay. The core of the MLI is a set of update formulas with different convergence properties and computational effort. However, the individual levels are not able to provide sufficiently accurate control feedback for a medium sized MG in real-time. We developed a new methodology to operate them in parallel in order to achieve high sampling rates and accurate feedback at the same time. Therefore, we introduced a new scheduling algorithm that enabled a high performance while staying always real-time feasible. These methodological advancements enabled us to define an MPC that stands out due to its high feedback rates even for stiff dynamical systems. In comprehensive numerical experiments, we showed that our proposed MPC scheme is capable to outperform current state-of-the-art control methods for MGs control significantly. In addition, it is more flexible because it allows the incorporation of operational bounds and global voltage control. In contrast to the RTI scheme, it always stays real-time feasible.

In order to estimate the system state online, we applied the principle of Moving Horizon Estimation (MHE). This approach is based on a moving time horizon, where a fixed number of past measurements is used to determine the state of the underlying process. Similar to MPC, the algorithm solves a sequence of related dynamic optimizations problems. The algorithmic ideas of the RTI were applied to MHE to reduce the computational effort and to enable a lower feedback delay. However, even the RTI for MHE (RTI-MHE) is not able to meet the real-time requirements of MG control. In order to reduce the computational effort even further, we developed a novel framework for online state and parameter estimation based on the RTI-MHE. Thereby, we proposed a reformulation of the arising optimization problems to achieve a decoupling of the system linearizations from the constantly changing measurement data. This enabled the application of MLI update formulas for MHE and thus resulted in a significant reduction of computation times. We presented a new hierarchy of update formulas for MHE and discussed their convergence properties. In order to operate multiple levels in parallel, we established a new methodology for the initialization of the arising problems. We demonstrated the capabilities of the proposed methods on a model of a realistic sized MG in a challenging load scenario. In comprehensive numerical experiments, we showed the capabilities of the proposed MHE scheme. In an open loop control setup, the estimator was able to provide accurate state estimations even after major disturbances within the sampling time of the MG. In closed loop simulations, our proposed MLI for MHE (MLI-MHE) algorithm enabled a high control performance even though the

system state was not completely measurable and the states estimates were affected by errors.

In summary, we showed that the framework of model-based feedback control has the potential to significantly improve the control performance of real world MGs. The proposed MPC scheme based on MLI is able to stabilize MGs faster and more reliable than current control approaches. In addition, it is capable to respect operational bounds and take economic goals into consideration. The newly introduced MHE framework with MLI update formulas enabled fast and accurate state estimates even after major disturbances in the grid. Numerical experiments showed the suitability of MPC and MHE for secondary MG control. Therefore, this thesis is an important step towards its real word implementation.

Bibliography

- [1] M. N. Alam, S. Chakrabarti, and A. Ghosh. Networked Microgrids: State-of-the-Art and Future Perspectives. *IEEE Transactions on Industrial Informatics*, 15(3):1238–1250, 2019. doi: 10.1109/TII.2018.2881540.
- [2] J. Albersmeyer. *Adjoint-based algorithms and numerical methods for sensitivity generation and optimization of large scale dynamic systems*. PhD thesis, Heidelberg University, 2010.
- [3] J. Albersmeyer and H. G. Bock. Sensitivity generation in an adaptive bdf-method. In *Modeling, Simulation and Optimization of Complex Processes*, 2008. doi: 10.1007/978-3-540-79409-7_2.
- [4] J. Albersmeyer, D. Beigel, C. Kirches, L. Wirsching, H. G. Bock, and J. P. Schlöder. Fast nonlinear model predictive control with an application in automotive engineering. In *Nonlinear Model Predictive Control. Lecture Notes in Control and Information Sciences*, volume 384. Springer, 2009. doi: 10.1007/978-3-642-01094-1_37.
- [5] A. Alessandri and M. Gaggero. Fast moving horizon state estimation for discrete-time systems using single and multi iteration descent methods. *IEEE Transactions on Automatic Control*, 62(9):4499–4511, 2017. doi: 10.1109/tac.2017.2660438.
- [6] K. D. Bachovchin and M. Ilić. Automated modeling of power system dynamics using the Lagrangian formulation. *International Transactions on Electrical Energy Systems*, 25(10):2087–2108, 2015. doi: 10.1002/etep.1950.
- [7] K. Baumgärtner, A. Zanelli, and M. Diehl. Zero-Order Moving Horizon Estimation. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4140–4146, 2019. doi: 10.1109/CDC40024.2019.9029525.
- [8] V. Becerra, P. Roberts, and G. Griffiths. Applying the extended kalman filter to systems described by nonlinear differential-algebraic equations. *Control Engineering Practice*, 9(3):267–281, 2001. doi: 10.1016/S0967-0661(00)00110-6.

- [9] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, Belmont, MA, USA, 3rd edition, 2005.
- [10] A. Bidram and A. Davoudi. Hierarchical Structure of Microgrids Control System. *IEEE Transactions on Smart Grid*, 3(4):1963–1976, 2012. doi: 10.1109/TSG.2012.2197425.
- [11] L. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering*, 8:243–248, 1984. doi: 10.1016/0098-1354(84)87012-X.
- [12] L. T. Biegler. *Nonlinear Programming*. Society for Industrial and Applied Mathematics, 2010. doi: 10.1137/1.9780898719383.
- [13] H. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. Bonner Mathematische Schriften. Universität Bonn, 1987.
- [14] H. Bock and K. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984. doi: 10.1016/s1474-6670(17)61205-9.
- [15] H. Bock, M. Diehl, E. Kostina, and J. Schlöder. Constrained optimal feedback control of systems governed by large differential algebraic equations. In *Real-Time PDE-Constrained Optimization*, pages 3–24. 2007. doi: 10.1137/1.9780898718935.ch1.
- [16] H. G. Bock, J. Gutkunst, A. Potschka, and M. E. S. Garcés. A flow perspective on nonlinear least-squares problems. *Vietnam Journal of Mathematics*, 48(4): 987–1003, 2020. doi: 10.1007/s10013-020-00441-z.
- [17] C. Bordons, F. Garcia-Torres, and M. Ridao. *Model Predictive Control of Microgrids*. Springer International Publishing, 2019. doi: 10.1007/978-3-030-24570-2.
- [18] P. Braun, T. Faulwasser, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Hierarchical distributed ADMM for predictive control with applications in power networks. *IFAC Journal of Systems and Control*, 3:10–22, 2018. doi: 10.1016/j.ifacsc.2018.01.001.
- [19] D. Clarke, C. Mohtadi, and P. Tuffs. Generalized predictive control—Part I. The basic algorithm. *Automatica*, 23(2):137–148, 1987. doi: 10.1016/0005-1098(87)90087-2.

- [20] D. Clarke, C. Mohtadi, and P. Tuffs. Generalized Predictive Control—Part II Extensions and interpretations. *Automatica*, 23(2):149–160, 1987. doi: 10.1016/0005-1098(87)90088-4.
- [21] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck. The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow. *IEEE Transactions on Power Systems*, 31(4):3008–3018, 2016. doi: 10.1109/TPWRS.2015.2463111.
- [22] F. Daum. Nonlinear filters: beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69, 2005. doi: 10.1109/maes.2005.1499276.
- [23] K. De Brabandere, K. Vanthournout, J. Driesen, G. Deconinck, and R. Belmans. Control of Microgrids. In *2007 IEEE Power Engineering Society General Meeting*, pages 1–7, 2007. doi: 10.1109/PES.2007.386042.
- [24] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, University of Heidelberg, 2001.
- [25] M. Diehl, R. Findeisen, and F. Allgöwer. A Stabilizing Real-Time Implementation of Nonlinear Model Predictive Control. In *Real-Time PDE-Constrained Optimization*, pages 25–52. doi: 10.1137/1.9780898718935.ch2.
- [26] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4): 577–585, 2002. doi: 10.1016/S0959-1524(01)00023-3.
- [27] M. Diehl, R. Findeisen, H. G. Bock, F. Allgöwer, and J. Schlöder. Nominal stability of real-time iteration scheme for nonlinear model predictive control. *IEE Proceedings - Control Theory and Applications*, 152(3):296–308, 2005. doi: 10.1049/ip-cta:20040008.
- [28] M. Diehl, P. Kühn, H. G. Bock, and J. P. Schlöder. Schnelle Algorithmen für die Zustands- und Parameterschätzung auf bewegten Horizonten (Fast Algorithms for State and Parameter Estimation on Moving Horizons). *at - Automatisierungstechnik*, 54(12):602–613, 2006. doi: 10.1524/auto.2006.54.12.602.
- [29] M. Diehl, A. Walther, H. G. Bock, and E. Kostina. An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization. *Optimization Methods and Software*, 25(4):531–552, 2010. doi: 10.1080/10556780903027500.

- [30] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, and T. Faulwasser. Toward Distributed OPF Using ALADIN. *IEEE Transactions on Power Systems*, 34(1): 584–594, 2019. doi: 10.1109/TPWRS.2018.2867682.
- [31] T. Eram and P. L. Chapman. Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques. *IEEE Transactions on Energy Conversion*, 22(2):439–449, 2007. doi: 10.1109/TEC.2006.874230.
- [32] T. Faulwasser and A. Engelmann. Toward economic NMPC for multistage AC optimal power flow. *Optimal Control Applications and Methods*, 41(1):107–127, 2020. doi: 10.1002/oca.2487.
- [33] T. Faulwasser, A. Engelmann, T. Mühlpfordt, and V. Hagenmeyer. Optimal power flow: an introduction to predictive, distributed and stochastic control challenges. *at - Automatisierungstechnik*, 66(7):573–589, Jul 2018. doi: 10.1515/auto-2018-0040.
- [34] H. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014. doi: 10.1007/s12532-014-0071-1.
- [35] R. Findeisen, M. Diehl, I. Disli-Uslu, S. Schwarzkopf, F. Allgower, H. G. Bock, J. Schloder, and E. Gilles. Computation and performance assessment of nonlinear model predictive control. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 4, pages 4613–4618 vol.4, 2002. doi: 10.1109/CDC.2002.1185105.
- [36] S. Frank and S. Rebennack. An introduction to optimal power flow: Theory, formulation, and examples. *IIE Transactions*, 48(12):1172–1197, 2016. doi: 10.1080/0740817X.2016.1189626.
- [37] S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: a bibliographic survey I. *Energy Systems*, 3(3):221–258, Sep 2012. doi: 10.1007/s12667-012-0056-y.
- [38] J. Frasch, L. Wirsching, S. Sager, and H. G. Bock. Mixed-Level Iteration Schemes for Nonlinear Model Predictive Control. *IFAC Proceedings Volumes*, 45(17):138–144, 2012. doi: 10.3182/20120823-5-nl-3013.00085.
- [39] A. Gelb. *Applied optimal estimation*. MIT press, 1974.

-
- [40] J. D. Glover and M. S. Sarma. *Power System Analysis and Design*. Brooks/Cole Publishing Co., 2001.
- [41] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2):107, 1993. doi: 10.1049/ip-f-2.1993.0015.
- [42] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl. From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control*, 93(1):62–80, 2020. doi: 10.1080/00207179.2016.1222553.
- [43] L. Grüne. Nominal model-predictive control. In *Encyclopedia of Systems and Control*, pages 1–10. Springer London, 2013. doi: 10.1007/978-1-4471-5102-9_1-1.
- [44] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer London, 2017. doi: 10.1007/978-3-319-46024-6.
- [45] J. M. Guerrero, M. Chandorkar, T.-L. Lee, and P. C. Loh. Advanced Control Architectures for Intelligent Microgrids—Part I: Decentralized and Hierarchical Control. *IEEE Transactions on Industrial Electronics*, 60(4):1254–1262, 2013. doi: 10.1109/TIE.2012.2194969.
- [46] J. Gutekunst. *Feedback Control for Average Output Systems*. PhD thesis, Heidelberg University, 2019.
- [47] J. Gutekunst, R. Scholz, A. Nurkanović, A. Mešanović, H. G. Bock, and E. Kostina. Fast moving horizon estimation using multi-level iterations for microgrid control. *at - Automatisierungstechnik*, 68(12):1059–1076, 2020. doi: 10.1515/auto-2020-0081.
- [48] C. A. Hans, P. Sopasakis, A. Bemporad, J. Raisch, and C. Reincke-Collon. Scenario-based model predictive operation control of islanded microgrids. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3272–3277, 2015. doi: 10.1109/CDC.2015.7402711.
- [49] C. A. Hans, P. Braun, J. Raisch, L. Grüne, and C. Reincke-Collon. Hierarchical Distributed Model Predictive Control of Interconnected Microgrids. *IEEE Transactions on Sustainable Energy*, 10(1):407–416, 2019. doi: 10.1109/TSTE.2018.2802922.

- [50] E. Haseltine and J. Rawlings. Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation. *Industrial & Engineering Chemistry Research*, 44(8):2451–2460, 2005. doi: 10.1021/ie034308l.
- [51] N. Hatziaargyriou. *Microgrids: Architectures and Control*. Wiley-IEEE Press, 2014. doi: 10.1002/9781118720677.
- [52] D. Haßkerl, A. Meyer, N. Azadfallah, S. Engell, A. Potschka, L. Wirsching, and H. G. Bock. Study of the performance of the multi-level iteration scheme for dynamic online optimization for a fed-batch reactor example. In *2016 European Control Conference (ECC)*, pages 459–464, 2016. doi: 10.1109/ECC.2016.7810327.
- [53] M. Ilić. Toward a unified modeling and control for sustainable and resilient electric energy systems. *Foundations and Trends® in Electric Energy Systems*, 1(1-2):1–141, 2016. doi: 10.1561/3100000002.
- [54] M. Ilić, R. Jaddivada, and X. Miao. Modeling and analysis methods for assessing stability of microgrids. *IFAC-PapersOnLine*, 50(1):5448–5455, 2017. doi: 10.1016/j.ifacol.2017.08.1081.
- [55] M. Ilić, R. Jaddivada, X. Miao, and N. Popli. Toward multi-layered mpc for complex electric energy systems. In *Handbook of Model Predictive Control*, pages 625–663. Springer International Publishing, 2019. doi: 10.1007/978-3-319-77489-3_26.
- [56] J. Kallrath. *Modeling Languages in Mathematical Optimization*. Springer Publishing Company, Incorporated, 2012. doi: 10.1007/978-1-4613-0215-5.
- [57] C. Kirches. *Fast numerical methods for mixed-integer nonlinear model-predictive control*. Springer, 2010. doi: 10.1007/978-3-8348-8202-8.
- [58] T. Kraus, P. Kühn, L. Wirsching, H. G. Bock, and M. Diehl. A Moving Horizon State Estimation algorithm applied to the Tennessee Eastman Benchmark Process. In *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 2006. doi: 10.1109/mfi.2006.265620.
- [59] P. Kühn, M. Diehl, T. Kraus, J. Schlöder, and H. G. Bock. A real-time algorithm for moving horizon state and parameter estimation. *Computers & Chemical Engineering*, 35(1):71–83, 2011. doi: 10.1016/j.compchemeng.2010.07.012.
- [60] P. Kundur and N. Balu. *Power System Stability and Control*. McGraw-Hill, 1994.

- [61] P. Kythe and M. Schäferkötter. *Handbook of Computational Methods for Integration*. Taylor & Francis, 2004. doi: 10.1201/9780203490303.
- [62] R. Lasseter. Microgrids. In *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.02CH37309)*, volume 1, pages 305–308 vol.1, 2002. doi: 10.1109/PESW.2002.985003.
- [63] R. H. Lasseter, A. A. Akhil, C. Marnay, J. Stephens, J. E. Dagle, R. T. Guttromson, A. S. Meliopoulos, R. J. Yinger, and J. H. Eto. Integration of distributed energy resources: The certs microgrid concept. Technical report, CERTS, 2003.
- [64] J. Lee. Model Predictive Control: Review of the Three Decades of Development. *International Journal of Control, Automation and Systems*, 9:415–424, 2011. doi: 10.1007/s12555-011-0300-6.
- [65] C. Leidereiter, A. Potschka, and H. G. Bock. Dual decomposition for QPs in scenario tree NMPC. In *2015 European Control Conference (ECC)*, pages 1608–1613, 2015. doi: 10.1109/ECC.2015.7330767.
- [66] D. Leineweber, I. Bauer, H. Bock, and J. Schlöder. An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization. Part I: Theoretical Aspects. *Computers & Chemical Engineering*, 27:157–166, 2003. doi: 10.1016/S0098-1354(02)00158-8.
- [67] Y. Levron, J. Belikov, and D. Baimel. A Tutorial on Dynamics and Control of Power Systems with Distributed and Renewable Energy Sources Based on the DQ0 Transformation. *Applied Sciences*, 8(9), 2018. doi: 10.3390/app8091661.
- [68] D. Limon and T. Alamo. Tracking model predictive control. In *Encyclopedia of Systems and Control*, pages 1475–1484. Springer London, 2015. doi: 10.1007/978-1-4471-5058-9_3.
- [69] C. Lindscheid, D. Haßkerl, A. Meyer, A. Potschka, H. G. Bock, and S. Engell. Parallelization of modes of the multi-level iteration scheme for nonlinear model-predictive control of an industrial process. In *2016 IEEE Conference on Control Applications (CCA)*, pages 1506–1512, 2016. doi: 10.1109/CCA.2016.7588014.
- [70] V. Masson-Delmotte, P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Z. (eds.). *Climate change 2021: The physical science basis. contribution of working group i to the sixth*

- assessment report of the intergovernmental panel on climate change. Technical report, IPCC, 2021.
- [71] N. Meyer-Hübner, M. R. Suriyah, T. Leibfried, V. Slednev, V. Bertsch, W. Fichtner, P. Gerstner, M. Schick, and V. Heuveline. Time constrained optimal power flow calculations on the German power grid. In *International ETG Congress 2015; Die Energiewende - Blueprints for the new energy age*, pages 1–7, 2015.
- [72] N. Meyer-Hübner, M. Suriyah, T. Leibfried, V. Slednev, V. Bertsch, W. Fichtner, P. Gerstner, M. Schick, and V. Heuveline. Optimal Storage Operation with Model Predictive Control in the German Transmission Grid. In *Advances in Energy System Optimization*, pages 31–45. Springer International Publishing, 2017. doi: 10.1007/978-3-319-51795-7_3.
- [73] A. Mešanović, D. Unseld, U. Münz, C. Ebenbauer, and R. Findeisen. Parameter Tuning and Optimal Design of Decentralized Structured Controllers for Power Oscillation Damping in Electrical Networks. In *2018 Annual American Control Conference (ACC)*, pages 3828–3833, 2018. doi: 10.23919/ACC.2018.8431420.
- [74] A. Mešanović, U. Münz, A. Szabo, M. Mangold, J. Bamberger, M. Metzger, C. Heyde, R. Krebs, and R. Findeisen. Structured controller parameter tuning for power systems. *Control Engineering Practice*, 101:104490, 2020. doi: 10.1016/j.conengprac.2020.104490.
- [75] X. Miao and M. Ilić. Model predictive excitation control for constrained frequency and voltage stabilization. In *2015 IEEE Power Energy Society General Meeting*, pages 1–5, 2015. doi: 10.1109/PESGM.2015.7285746.
- [76] X. Miao and M. Ilić. Distributed model predictive control of synchronous machines for stabilizing microgrids. In *2017 North American Power Symposium (NAPS)*, pages 1–6, 2017. doi: 10.1109/NAPS.2017.8107347.
- [77] L. I. Minchala-Avila, L. E. Garza-Castañón, A. Vargas-Martínez, and Y. Zhang. A Review of Optimal Control Techniques Applied to the Energy Management and Control of Microgrids. *Procedia Computer Science*, 52:780–787, 2015. doi: 10.1016/j.procs.2015.05.133.
- [78] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei. A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems. *IEEE Transactions on Smart Grid*, 8(6):2941–2962, 2017. doi: 10.1109/TSG.2017.2720471.

- [79] U. Münz, M. Metzger, A. Szabo, M. Reischböck, F. Steinke, P. Wolfrum, R. Sol-lacher, D. Obradovic, M. Buhl, T. Lehmann, M. Duckheim, and S. Langemeyer. Overview of recent control technologies for future power systems: An industry perspective. *at - Automatisierungstechnik*, 63(11):869–882, 2015. doi: 10.1515/auto-2015-0047.
- [80] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.
- [81] A. Nurkanovic, S. Albrecht, and M. Diehl. Multi-Level Iterations for Economic Nonlinear Model Predictive Control. In *Recent Advances in Model Predictive Control. Lecture Notes in Control and Information Sciences*. Springer, 2021. doi: 10.1007/978-3-030-63281-6_4.
- [82] A. Nurkanović, A. Zanelli, S. Albrecht, and M. Diehl. The Advanced Step Real Time Iteration for NMPC. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5298–5305, 2019. doi: 10.1109/CDC40024.2019.9029543.
- [83] A. Nurkanović, A. Mešanović, M. Sperl, S. Albrecht, U. Münz, R. Findeisen, and M. Diehl. Optimization-based primary and secondary control of microgrids. *at - Automatisierungstechnik*, 68(12):1044–1058, 2020. doi: 10.1515/auto-2020-0088.
- [84] A. Nurkanović, A. Mešanović, A. Zanelli, G. Frison, J. Frey, S. Albrecht, and M. Diehl. Real-Time Nonlinear Model Predictive Control for Microgrid Operation. In *2020 American Control Conference (ACC)*, pages 4989–4995, 2020. doi: 10.23919/ACC45564.2020.9147816.
- [85] A. Nurkanović, A. Zanelli, S. Albrecht, G. Frison, and M. Diehl. Contraction Properties of the Advanced Step Real-Time Iteration for NMPC. *IFAC-PapersOnLine*, 53(2):7041–7048, 2020. doi: 10.1016/j.ifacol.2020.12.449.
- [86] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. A. Jiménez-Estévez, and N. D. Hatziargyriou. Trends in Microgrid Control. *IEEE Transactions on Smart Grid*, 5(4):1905–1919, 2014. doi: 10.1109/TSG.2013.2295514.
- [87] C. J. O’Rourke, M. M. Qasim, M. R. Overlin, and J. L. Kirtley. A Geometric Interpretation of Reference Frames and Transformations: dq0, Clarke, and Park. *IEEE Transactions on Energy Conversion*, 34(4):2070–2083, 2019. doi: 10.1109/TEC.2019.2941175.

- [88] A. Parisio, E. Rikos, and L. Glielmo. A Model Predictive Control Approach to Microgrid Operation Optimization. *IEEE Transactions on Control Systems Technology*, 22(5):1813–1827, 2014. doi: 10.1109/TCST.2013.2295737.
- [89] L. Pontryagin, V. Boltyanski, R. Gamkrelidze, and E. Miscenko. The mathematical theory of optimal processes. *Wiley*, Chichester, 1962.
- [90] V. Quaschnig. Sektorkopplung durch die energiewende. Technical report, Hochschule für Technik und Wirtschaft HTW Berlin, 2016.
- [91] r2b energy consulting GmbH, Consentec GmbH, Fraunhofer Institute for Systems and Innovation Research ISI, and TEP Energy GmbH. Monitoring the adequacy of resources in the European electricity markets. Technical report, BMWI, 2021.
- [92] C. Rao, J. Rawlings, and J. Lee. Constrained linear state estimation—a moving horizon approach. *Automatica*, 37(10):1619–1628, 2001. doi: 10.1016/s0005-1098(01)00115-7.
- [93] C. Rao, J. Rawlings, and D. Mayne. Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003. doi: 10.1109/tac.2002.808470.
- [94] J. Rawlings, D. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.
- [95] S. Reichert, G. Griepentrog, and B. Stickan. Comparison between grid-feeding and grid-supporting inverters regarding power quality. In *2017 IEEE 8th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pages 1–4, 2017. doi: 10.1109/PEDG.2017.7972536.
- [96] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodríguez. Control of Power Converters in AC Microgrids. *IEEE Transactions on Power Electronics*, 27(11):4734–4749, 2012. doi: 10.1109/TPEL.2012.2199334.
- [97] I. Ross. *A Primer on Pontryagin’s Principle in Optimal Control: Second Edition*. Collegiate Publishers, 2015.
- [98] V. Salas, E. Olías, A. Barrado, and A. Lázaro. Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems. *Solar Energy Materials and Solar Cells*, 90(11):1555–1578, 2006. doi: 10.1016/j.solmat.2005.10.023.

- [99] R. Sargent and G. Sullivan. The development of an efficient optimal control package. In *Proceedings of the 8th IFIP Conference on Optimization Techniques*, volume 7, pages 158–168, 1978. doi: 10.1007/BFb0006520.
- [100] J. Schiffer, D. Zonetti, R. Ortega, A. M. Stanković, T. Sezi, and J. Raisch. A survey on modeling of microgrids—from fundamental physics to phasors and voltage sources. *Automatica*, 74:135–150, 2016. doi: 10.1016/j.automatica.2016.07.036.
- [101] R. Scholz. Stabiles condensing für optimale steuerung. Master’s thesis, Universität Heidelberg, 2016.
- [102] R. Scholz, A. Nurkanovic, A. Mesanovic, J. Gutekunst, A. Potschka, H. G. Bock, and E. Kostina. Model-Based Optimal Feedback Control for Microgrids with Multi-Level Iterations. In *Operations Research Proceedings 2019. Operations Research Proceedings (GOR (Gesellschaft für Operations Research e.V.))*, 2020. doi: 10.1007/978-3-030-48439-2_9.
- [103] R. Scholz, A. Nurkanović, A. Mešanović, J. Gutekunst, A. Potschka, H. G. Bock, and E. Kostina. Multi-level iterations for microgrid control with automatic level choice. In *Scientific Computing in Electrical Engineering*. Springer International Publishing, 2020. In Press.
- [104] V. Schulz, H. Bock, and M. Steinbach. Exploiting Invariants in the Numerical Solution of Multipoint Boundary Value Problems for DAE. *SIAM Journal on Scientific Computing*, 19(2):440–467, 1998. doi: 10.1137/s1064827594261917.
- [105] J. W. Simpson-Porco, Q. Shafiee, F. Dörfler, J. C. Vasquez, J. M. Guerrero, and F. Bullo. Secondary frequency and voltage control of islanded microgrids via distributed averaging. *IEEE Transactions on Industrial Electronics*, 62(11): 7025–7038, 2015. doi: 10.1109/TIE.2015.2436879.
- [106] O. Stanojev, U. Markovic, P. Aristidou, G. Hug, D. S. Callaway, and E. Vrettos. MPC-Based Fast Frequency Control of Voltage Source Converters in Low-Inertia Power Systems. *IEEE Transactions on Power Systems*, pages 1–1, 2020. doi: 10.1109/TPWRS.2020.2999652.
- [107] P. Sterchele, J. Brandes, J. Heilig, D. Wrede, C. Kost, T. Schlegl, A. Bett, and H.-M. Henning. Paths to a climate-neutral energy system. the german energy transition in its social context. Technical report, Fraunhofer ISE, 2020.

- [108] S. W. Sung, J. Lee, and I.-B. Lee. *Process Identification and PID Control*. John Wiley & Sons, Ltd, 2009. doi: 10.1002/9780470824122.
- [109] X. Tan, Q. Li, and H. Wang. Advances and trends of energy storage technology in microgrid. *International Journal of Electrical Power & Energy Systems*, 44(1): 179–191, 2013. doi: 10.1016/j.ijepes.2012.07.015.
- [110] A. Ulbig, T. Rinke, S. Chatzivasileiadis, and G. Andersson. Predictive control for real-time frequency regulation and rotational inertia provision in power systems. In *52nd IEEE Conference on Decision and Control*, pages 2946–2953, 2013. doi: 10.1109/CDC.2013.6760331.
- [111] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed MPC Strategies With Application to Power System Automatic Generation Control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008. doi: 10.1109/TCST.2008.919414.
- [112] A. Wächter and L. Biegler. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57, 2006. doi: 10.1007/s10107-004-0559-y.
- [113] L. Wirsching. *Multi-Level Iteration Schemes with Adaptive Level Choice for Nonlinear Model Predictive Control*. PhD thesis, Heidelberg University, 2018.
- [114] R. Zamora and A. K. Srivastava. Controls for microgrids with storage: Review, challenges, and research needs. *Renewable and Sustainable Energy Reviews*, 14(7):2009–2018, 2010. doi: 10.1016/j.rser.2010.03.019.
- [115] A. Zanelli, R. Quirynen, and M. Diehl. An Efficient Inexact NMPC Scheme with Stability and Feasibility Guarantees. *IFAC-PapersOnLine*, 49(18):53–58, 2016. doi: 10.1016/j.ifacol.2016.10.139.
- [116] A. Zanelli, Q. Tran-Dinh, and M. Diehl. A Lyapunov function for the combined system-optimizer dynamics in inexact model predictive control. *Automatica*, 134:109901, 2021. doi: 10.1016/j.automatica.2021.109901.
- [117] V. M. Zavala and L. T. Biegler. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1):86–93, 2009. doi: 10.1016/j.automatica.2008.06.011.
- [118] H. Zhou, C. A. Hans, and W. Zhang. Minimax model predictive operation control of grid-connected microgrids. In *2016 IEEE Conference on Control Applications (CCA)*, pages 58–63, 2016. doi: 10.1109/CCA.2016.7587822.

List of Figures

1.1	Structure of the individual chapters and their mutual dependencies. . .	11
2.1	Hierarchical control structure of MG.	17
3.1	Resistor and inductor in series connection.	24
3.2	Example of Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL).	30
3.3	Sketch of synchronous generator	41
3.4	Signal flow of Synchronous Generator (SG) model.	42
3.5	Sketch of a three phase inverter with an LC filter.	49
3.6	Schematic of a three phase Direct Current (DC) to AC H-bridge inverter.	49
3.7	Signal flow of a grid-forming inverter.	51
3.8	Signal flow of a grid-following inverter.	53
3.9	Signal flow of a grid-supporting inverter.	55
3.10	Steady-state model of an inverter connected to an infinite bus.	57
3.11	Simulation results of a grid-supporting inverter connected to an infinite bus. The blue trajectories show the states of the inverter model and the red plot indicates the active power reference respectively the grid frequency. At the nominal frequency, the inverter follows the provided reference values. At a reduced frequency, it provides additional power to stabilize the grid. The parameters are $R = 0.1$, $L = 0.01$ and $D = 10$	59
4.1	Idealized MPC process. The state is depicted in red, the control in blue. At time t_k , the controller receives the current state ξ_k and computes an optimal feedback control u_{ξ_k} on the prediction horizon $[t_k, t_k + T]$	63
4.2	Signal flow of MPC algorithm.	63
4.3	Discretized MPC process. The controls are depicted in blue and the states in red. Because of the discretization, the controls are constant in the sampling intervals. The solid lines represent the past states and controls, while the dashed lines indicate the optimized model predictions of the controller.	68
4.4	Temporal communication between the controller and the process. The transition phase is not depicted because their computation time is negligible.	72

5.1	Signal flow of MLI scheme with combined levels. The gray boxes represent independent components, which are executed on dedicated CPU cores. The arrows represent the communication signals between the components. In this sketch, a top-down communication for the reference data is used as depicted on the right hand side.	86
5.2	Temporal communication of a BD^2 scheme with a Level D iteration every second sampling point and one intermediate Level B iteration. The gray areas indicate when the corresponding level is busy.	89
5.3	Exemplary scheduling of levels with Adaptive Level Choice by Computation Time (ALC-Time) with varying computation times. The colors indicate the state of the levels. Red shows that the corresponding level is busy and green that it is ready. At the sampling points, the highest available level is used to generate feedback. The number of intermediate Level B iterations depend on the computation time of Level D. The gray areas indicate the feedback phase of the Feedback-generating Quadratic Programm (FQP).	91
6.1	Sketch of the test grid with diesel generator, PV and load.	95
6.2	Diesel generator with primary controllers.	96
6.3	Active P_{load} and reactive power Q_{load} demand at the load and active power input P_{PV} of Photovoltaic (PV) system.	97
6.4	Control performance of RTI and ideal MPC.	98
6.5	Frequency (left) and voltage at the load (right) controlled with Level C, B and A controller. The trajectory of the controllers are depicted in red, the PI-controller for comparison is shown in blue. All schemes are real-time feasible with a sampling rate below 200 ms.	99
6.6	Computation time for every iteration for the different levels. The sampling time is represented by the dashed line. For Level A it is 5 ms, for Level B 100 ms and for Level C 200 ms. Level C, B and A are real-time feasible. Level D uses a sampling time of 500 ms but is not real-time feasible.	100
6.7	Topology of the test MG.	102
6.8	Active and reactive power demand P and Q of the load for different load profiles.	103
6.9	Simulation results for load jump scenario. The MLI-controller is able to steer frequency and voltage back to the nominal value faster and with a lower initial drop, while respecting the operational limits.	107
6.10	Simulation results for linear load ramp scenario. The MLI-controller is able to steer frequency and voltage back to the nominal value faster and with a lower initial drop, while respecting the operational limits.	108

7.1	Exemplarily MHE process. The measurements are depicted by the black dots, the past controls are shown in blue and the model representation of the process on the horizon $[t_{k-M}, t_k]$ is depicted in red. The state estimate ξ_k is given by the state of the model on the current time point t_k	111
7.2	Signal flow of closed loop MPC setup with state estimator.	112
7.3	Temporal communication of process, estimator, and controller in the RTI-MHE algorithm. The MHE preparation phase starts at the end of the RTI feedback phase, when the applied control is available.	120
7.4	Illustration of the estimation horizon shift and the induced variable shift map P_{shift}^l with $l = 2$	123
7.5	Exemplary MLI scheme with one Level D update every fourth sampling point followed by three intermediate Level C updates (CD^4). The gray area denotes, when the corresponding level is busy.	143
8.1	Topology of the test MG.	151
8.2	Computational results for open and closed loop simulations. In the top row, the trajectories for the frequency are shown and the state estimates of the different MLI-MHE schemes are indicated by the crosses. In the middle, the voltage at the load is depicted. The relative error of the full state estimates for the different schemes is shown in the bottom row.	154
8.3	Computation times for different MLI-MHE schemes in an open loop and in a closed loop scenario. From top to bottom, computation times for a D^1 estimator, a CD^4 schema estimator, and a BD^4 schema estimator are shown. The elapsed computation times are depicted by the heights of the bars while the width indicates during which sampling intervals the computations were performed. The dashed lines mark the maximal allowed computation time for the different levels to stay real-time feasible.	156

List of Tables

5.1	Computations and update formulas for the QP data of the different levels.	84
6.1	Sampling time and maximal loop time for different schemes.	101
7.1	Computations and update formulas for the QP data for the different levels. The linearization points v are the shifted approximate solutions of the previous MHE problem, according to (7.21). The reference evaluations and Jacobians as well as the multipliers in Level C are the shifted versions $\lambda_{b_{\text{dyn}}}$ of the preceding MHE QPs.	132
8.1	Computation times of MHE algorithm for different scenarios and schemes.	157

List of Acronyms

Acronyms from Power Engineering

AC	Alternating Current
DC	Direct Current
OPF	Optimal Power Flow
PF	Power Flow
KVL	Kirchhoff's Voltage Law
KCL	Kirchhoff's Current Law
RES	Renewable Energy Resources
PV	Photovoltaic
RMS	Root Mean Square
PWM	Pulse Width Modulation
SG	Synchronous Generator
AVR	Automatic Voltage Regulator
MG	Microgrid
PI	Proportional-Integral
PLL	Phase Locked Loop
PID	proportional-integral-derivative
DG	Diesel Generator
BA	Battery
SI	International System of Units
VSI	Voltage Source Inverter
PCC	Point of Common Coupling

Mathematical Acronyms

MHE	Moving Horizon Estimation
RTI	Real-Time Iteration
MLI	Multi-Level Iteration
NMPC	Nonlinear Model Predictive Control
MPC	Model Predictive Control
NLP	Nonlinear Programming Problem
SQP	Sequential Quadratic Programming
DAE	Differential Algebraic Equation
QP	Quadratic Programm
FQP	Feedback-generating Quadratic Programm
IVE	Initial Value Embedding
CTE	Control Trajectory Embedding
BFGS	Broyden–Fletcher–Goldfarb–Shanno
ALC-Con	Adaptive Level Choice by Contractivity
ALC-Time	Adaptive Level Choice by Computation Time
GN	Gauß-Newton
MLI-MHE	MLI for MHE
RTI-MHE	RTI for MHE
IND	Internal Numerical Differentiation
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation