Technische Universität München
Lehrstuhl für Echtzeitsysteme und Robotik

Control and Stability of Power Systems using Reachability Analysis

Ahmed Elguindy

Vollständiger Abdruck der von der Fakultät der Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr. Helmut Seidl
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. Matthias Althoff
	2.	Prof. Dr. Majid Zamani

Die Dissertation wurde am 04.07.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 02.11.2017 angenommen.

Ich versichere, dass ich diese Doktorarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, 02.07.2017

Elguindy, Ahmed

Abstract

This thesis proposes several applications of reachability analysis to control and assess stability of power systems with formal guarantees. Simply put, reachability analysis makes it possible to compute the bounds of all possible trajectories for a range of operating conditions, while simultaneously meeting the practical requirements of realistic systems found in the power industry. Novel methods have been developed in this thesis to exploit the advantages of employing reachability analysis in a wide range of applications.

First, we investigate the assessment of transient stability via compositional techniques to improve the algorithmic efficiency of classical reachability algorithms. A special algorithm was developed, capable of drastically reducing the computational efforts associated with existing techniques. This made it possible to establish transient stability of power systems formalized via a set of differential algebraic equations and consisting of more than 100 state variables.

Second, we propose an algorithmic procedure that extends existing techniques computing reachable sets, in order to estimate the so-called region of attraction, which is known to be of great importance for the stability analysis of nonlinear systems. The developed method is compared with alternative and dominant techniques in this research area.

Third, we present the synthesis and verification of linear-parameter varying controllers in order to robustly establish transient stability of multi-machine power systems with formal guarantees. Both tasks are solved simultaneously in a systematic fashion within the context of a unified framework. Several benchmark examples are considered to showcase the applicability and scalability of the proposed approach.

Finally, we illustrate how reachability analysis can be utilized to verify safety of critical components found in power systems. In particular, we consider a realistic configuration of a boiler system within a combined cycle heat and power plant, in which the loss of the boiler leads to the emergency shut-down of the plant, hence jeopardizing safety of the complete utility grid. The task of verifying safety of the boiler cannot be achieved using numerical time-domain simulations, since only a single trajectory out of infinitely many can be checked at a time.

Zusammenfassung

Diese Arbeit präsentiert mehrere Anwendungen von Erreichbarkeitsanalyse zur Regelung und Stabilitätsbetrachtung von Energiesystemen mit formalen Garantien. Mithilfe von Erreichbarkeitsanalyse kann man die Trajektorien beschränken, die in einem Betriebsbereich auftreten können. Wie in dieser Arbeit gezeigt, ist dies auch für realistische Energiesysteme möglich. Im Rahmen dieser Arbeit wurden neuartige Methoden entwickelt, die es ermöglichen die Vorteile der Erreichbarkeitsanalyse für eine Vielzahl von Anwendungen zu nutzen.

Zuerst wird die Stabilitätsbetrachtung mithilfe kompositioneller Techniken untersucht, mit dem Ziel effizientere Algorithmen zu erhalten, als dies mit klassischen Erreichbarkeitsalgorithmen möglich wäre. Ein spezieller Algorithmus wurde entwickelt, der den Rechenaufwand im Vergleich zu existierenden Ansätzen drastisch reduziert. Dies erlaubt es dynamische Stabilität von Energiesystemen formal zu verifizieren, die mithilfe einer Menge von differential algebraischen Gleichungen und über 100 Zustandsvariablen beschrieben werden können.

Danach wird ein Algorithmus präsentiert, der existierende Techniken zur Berechnung von erreichbaren Mengen erweitert, um das Einzugsgebiet einer Ruhelage zu bestimmen. Dies ist von großer Bedeutung für die Stabilitätsanalyse nichtlinearer Systeme. Die entwickelte Methode wird mit existierenden Techniken in diesem Forschungsgebiet verglichen.

Anschließend wird die Synthese und Verifikation von linearen, parameter-varianten Reglern präsentiert, die die dynamische Stabilität von Mehrmaschinen-Engergiesysteme robust und mit formalen Garantien sicherstellt. Beide Teile werden systematisch und zur gleichen Zeit in einem gemeinsamen Verfahren gelöst. Mehrere repräsentative Beispiele zeigen die Anwendbarkeit und die Skalierbarkeit der vorgestellten Methode.

Zuletzt wird gezeigt wie Erreichbarkeitsanalyse dazu genutzt werden kann um die Sicherheit kritischer Komponenten in Energiesystemen zu verifizieren. Insbesondere wird eine realistische Struktur eines Heizkraftwerks mit Kraft-Wärme-Kopplung betrachtet, bei der der Ausfall des Heizkessels zu einer Notabschaltung des Kraftwerks führen kann und damit die Sicherheit des gesamten Stromnetzes gefährdet. Die Sicherheit des Heizkessels kann nicht allein mithilfe von numerischen Simulationen verifiziert werden, da immer nur eine einzelne Trajektorie aus unendlich vielen zur gleichen Zeit überprüft werden kann.

Acknowledgements

I would like to take the opportunity to mention several people who contributed into making this thesis possible. First, and most sincerely, I would like to thank my advisor Prof. Matthias Althoff, who offered me this interesting PhD opportunity. Furthermore, I am grateful for his clear guidance, valuable advice, and continuous support throughout the course of my PhD.

I would like also to thank Prof. Christine Chen, who hosted me in the summer of the year 2016 at the University of British Columbia. I am grateful for her helpful comments and suggestions with regards to Chapter 3. I am especially delighted that Prof. Majid Zamani from the Department of Electrical Engineering gave me valuable comments with regards to several aspects of my thesis. I wish to express my sincere gratitude to the plant operators at SWM Services GmbH for their willingness to allow me to perform experiments at the power plant München Süd GuD. Additionally, I would like to thank Mr. Julian Niedermeier in particular for his support with supplying measurement data from the power plant, which was very helpful in completing Chapter 6. I would also like to thank Konstantin Schaab and Prof. Olaf Stursberg from the University of Kassel. We worked together on the same DFG project, and their contributions were valuable for my research in Chapter 5.

Many thanks goes to my colleagues at the Cyber-Physical Systems group: Esra, Aaron, Albert, Silvia, Andrea, Sebastian, Stefanie, Markus, Carmella, and Christian. Additionally, special thanks to Dongkun and Bastian, with whom I wrote several papers and shared interesting discussions related to my PhD. Many thanks also go to Ute and Amy for supporting me with the administrative tasks at the Institute of Robotics and Embedded Systems.

My deepest gratitude is dedicated to my uncle Hesham El-Guindy, who we lost a few weeks ago, may God bless his soul. Finally, and most importantly, I wish to thank my parents, who supported me unconditionally throughout my entire life and brought me into the position of writing a PhD thesis.

Ahmed Elguindy

Munich, July 2017

Contents

Chapter 1

Introduction

This introductory chapter provides a general overview of current power systems, in addition to the main challenges which arose following the integration of renewable resources into the transmission network. This chapter also motivates the need for the development of new tools to analyse and control utility grids in a formal fashion. Of course, the consideration of all aspects found in power systems is certainly an ambitious task which cannot be covered in a book, let alone a PhD thesis. First, the scope and goals to be addressed throughout this thesis are formulated. Then, related work is briefly discussed to put this thesis in context. Finally, the main contributions and outline of this work are presented. A literature review is provided at the beginning of each chapter.

1.1 Scope and Goals

Power systems are widely considered to be the most complicated engineering system ever to be built by men in modern society. For example, Fig. 1.1 illustrates the network topology of the 380/220-kV transmission grid owned by 50 Hertz, which is one of the four transmission system operators (TSOs) operating in Germany. In essence, power systems broadly refer to the class of energy conversion systems, whose physical links are interconnected via a complex transmission network supplying electrical power to equipments ranging from industrial machinery to household appliances.

In recent years, the energy sector has undergone a radical transformation due to economical, environmental, and technical reasons; recently there has been an ongoing trend toward more environmentally-driven energy production, such as wind and solar generation. This worldwide trend is in the hope of reducing the carbon dioxide emissions and contributing effectively against global warming. However, the integration of renewable resources introduces notable challenges in the system operation due to their intermittent nature. From an economical perspective, following the energy deregulation and market liberalization, the transmission network handled by each TSO is operated under highly stressed operating conditions in order to reduce the transmission costs and initiate more revenue in current competitive markets. This, however, causes the network to operate close to the stability margins. Finally, owing to the technological advancement of distributed generation (DG) systems, there is a continuous interest in the transition from centralized generation towards a decentralized scheme with a considerable share of DG units.

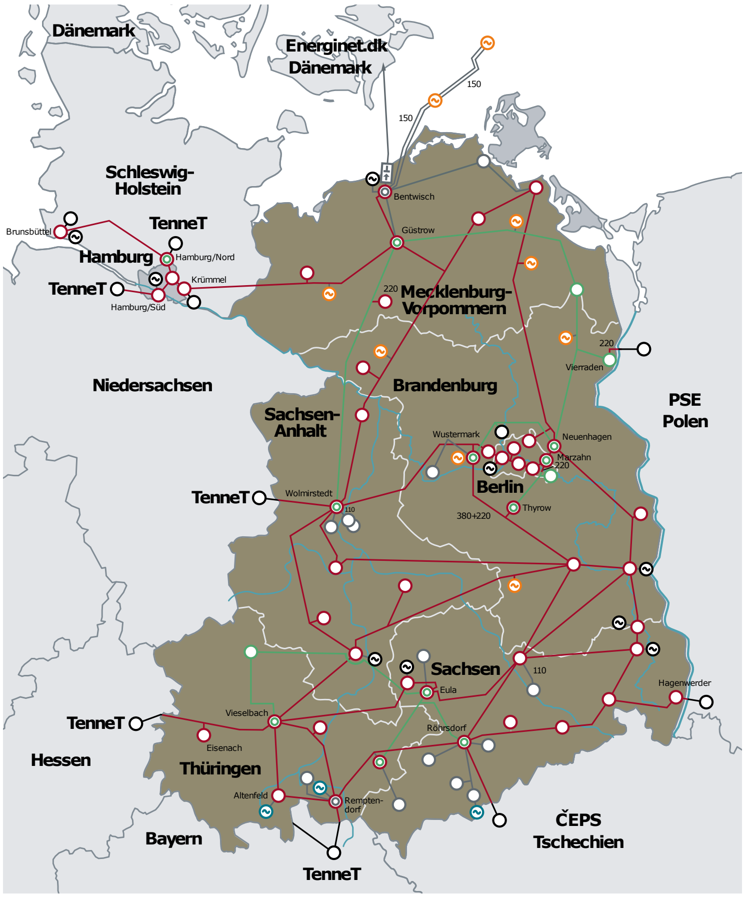


Figure 1.1: Topological overview of the 380/220-kV transmission grid operated by the TSO 50Hertz[1].

As power systems kept evolving through the years, particularly at a higher pace in the last two decades, several forms of system instabilities emerged; this escalated rapidly to a major threat for the system operation, as evidenced by recent blackouts in North America and Europe [15]. Due to the complexity of power systems, reliable operation in practice is achieved via separation of concerns; in other words, to classify the various forms leading to instability. This led to three distinctive stability categories [81, 82] as highlighted in Fig. 1.2; namely, the control area frequency stability, the bus voltage stability, and the rotor angle stability (the so-called *transient stability*), which is the main focus of this thesis.

Frequency stability refers to the ability of TSOs to maintain steady frequency within their control areas following a significant imbalance between generation and demand. Typically, frequency instability results from poor coordination of control between TSOs or due to insufficient generation reserve. In practice

[1] `http://www.50hertz.com/en/Grid-Access/Congestion-management/Static-grid-model`

frequency stability is effectively managed via a centralized control scheme, commonly known as automatic generation control (AGC); it compensates the deviation of the power grid frequency caused by the mismatch between supply and demand of the active power. The control action includes generation units (or loads) that respond in the case of short-term disturbances to the AGC signals referred to as primary and secondary frequency control, or in the case of long-term disturbances to manual operator dispatch commands, known as tertiary control [117].

Voltage stability deals with the ability of the transmission network to maintain the voltage magnitudes at any bus within their rated values. The driving force for voltage instability is usually the mismatch in the generated and consumed reactive power by industrial loads; this leads to a progressive fall of the voltage levels in some buses within the transmission network. In the worst-case scenario, the progressive fall might result eventually in a blackout or abnormally low voltages levels in several parts of the power system due to a series of unavoidable cascading effects. In practice, voltage stability is effectively handled at the generation side using automatic voltage regulators (AVRs) which influence the excitation system of synchronous generators. This in turn restores the voltage of its corresponding bus to its nominal values. On the other hand, within the transmission network, the issue is managed via tap-changing transformers, reactive power injection using synchronous condensers, or special controllers, such as the static synchronous compensator (STATCOM) which is based on power electronics voltage-source converters.

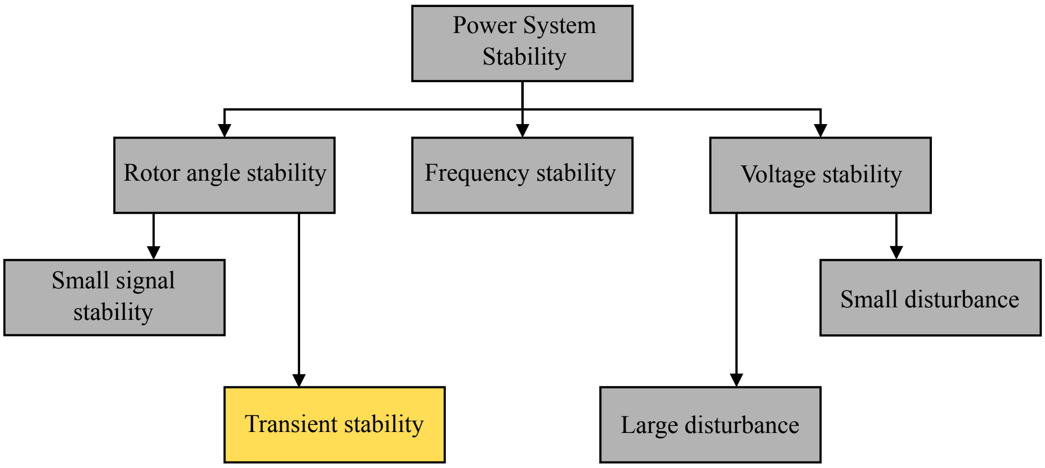


Figure 1.2: Classification of different stability categories in power systems according to [82].

The last category is transient stability, widely recognized technically and historically among theorists and practitioners alike as the most problematic issue when considering the dynamic security assessment of power systems [82]. The problem refers to the ability of the synchronous generators to remain in synchronism with the utility grid following a disturbance in the transmission network. A fundamental factor affecting this problem is the fact that the electrical power output of the generator varies according to the changes of the rotor angle. Instability results if the generator cannot absorb the kinetic energy corresponding to the speed difference of different rotors. In particular, we consider severe disturbances, such as e.g. a short circuit on transmission lines, resulting in large deviation from the initial operating point, such that small-signal analysis can no longer be applied, due to the highly nonlinear nature of the power-angle relationship of the synchronous machine [81, p. 21].

In practice, the TSOs establish transient stability via the so-called $(N-1)$ dynamic security assessment criteria. Obviously, the electric utility handled by the TSO is required to supply power to its connected loads, properly at all times without interruption, when its N components are available. As the naming of the $(N-1)$ criterion suggests, it requires that all system states can be restored back to a stable equilibrium if any single component fails; that is, the $(N-1)$ components still in service are capable of supplying the loads they were carrying before the fault event, in addition to the load supplied by the component subjected to the fault scenario. A generalized method resulting in a more reliable system would be the $(N-k)$ criteria, which assumes the loss of k components from the system. Clearly the criterion guarantees a reliable but more expensive system; thus, one always faces a technical-economical optimization problem.

The $(N-1)$ security assessment is typically examined using deterministic approaches employing time domain simulations. These approaches use numerical methods to integrate a set of nonlinear equations describing the dynamical behaviour of the power system under study. Numerical simulations are generally versatile, easy to implement, and their computational requirements grow moderately with the system dimensions. This has served the industry reasonably well, as it has led to high security levels and minimized the effort to establish transient stability. Numerical simulations, however, provide satisfying results only when there are no parametric or input uncertainties. To begin with, this is not generally the case due to the unavoidable mismatch between actual physical phenomena and derived models [107]. Furthermore, numerical simulation is not a formal technique to establish transient stability; in other words, numerical simulations do not provide any formal (mathematical) guarantees that the post-fault trajectory of the system state variables converge to an equilibrium point. One, however, can prove that transient stability cannot be established if a counter-example is produced using numerical simulations; this task can become computationally expensive since there exists infinitely many possible trajectories starting from an initial set of states. Finally, owing to increasingly varying operating conditions in power systems, associated with parametric and uncertain inputs introduced via the continuous integration of renewable resources, one has to consider a set of initial states to rigorously account for all possible eventualities during the fault scenario. This results in an exponential complexity, with regards to the number of simulations one has to run, in order to fully consider all initial states of the systems.

An alternative class of methods with a growing body of literature is the class of techniques based on Lyapunov stability theory and its various extensions. These methods can offer sufficient conditions for verifying stability of the power system during fault scenarios, using the so-called energy-like Lyapunov functions (LFs) [76]. The main attractive feature about Lyapunov methods is that running exhaustive time-domain simulations is no longer required in order to examine stability of the post-fault trajectory. Instead, one can determine a region in the state-space surrounding an equilibrium point, from which it can be proven that any initial state can be attracted by this equilibrium. However, Lyapunov-based techniques have several drawbacks which limit their applicability in practice. First, the approach relies on the existence of suitable LFs which are extremely difficult to find for nonlinear systems. This is due to the non-constructive nature of the Lyapunov theory; that is, the theorem only ensures the existence of a region of attraction, yet it does not provide a systematic way to find an initial feasible LF. Additionally, Lyapunov methods suffer from conservatism in estimating the stability regions, since the techniques often relax the optimization problem, to maximize the sub-level set of the LF. This is done either by enforcing convexity of the solution with conservative linear matrix inequalities (LMIs), or by employing non-convex bilinear matrix inequalities. Another disadvantage of this class of techniques is that it cannot formally verify if the system constraints are being met, for example if the bus voltage of a power system drops beyond limitations imposed by the grid operator. This is due to the fact that Lyapunov methods only analyze if a steady state of a disturbed system is eventually reached without specifying the exact system trajectory.

Recently, the computation of reachable sets has emerged as an alternative, and promising, technique for the analysis of power systems. The most interesting feature of reachability analysis is that it combines the advantages of numerical simulations and Lyapunov-based methods; that is, reachability analysis is a formal technique capable of establishing transient stability with formal guarantees, and more importantly, it scales moderately with the system dimensions compared to Lyapunov-based methods, which can only handle a maximum of five state variables. Generally speaking, reachability analysis refers to the class of techniques that can determine the set of states that a system can reach over a time-horizon starting from a set of initial states under the influence of a set of uncertain inputs. Thus, instead of simulating single trajectories, specified for a vector of deterministic input variables, one can compute using reachability analysis the set that encloses all possible eventualities (infinitely many). In fact, reachability analysis makes it possible to formally verify whether the algebraic constraints, such as the bus voltage or the line frequency of a power system, leave the permitted ranges specified by the TSO. A simple scenario is illustrated in Fig. 1.3, where one can see that the evolution of the reachable set with respect to time for a generic power system does not intersect with the limits imposed by the grid operator. Hence one can ensure safety of the power system while meeting the grid requirements using reachability analysis.

1.2 Thesis Outline

This thesis is concerned with the analysis and control of power systems via the computation of reachable sets, with a special emphasis on studies involving transient stability. Note that the chapters are not

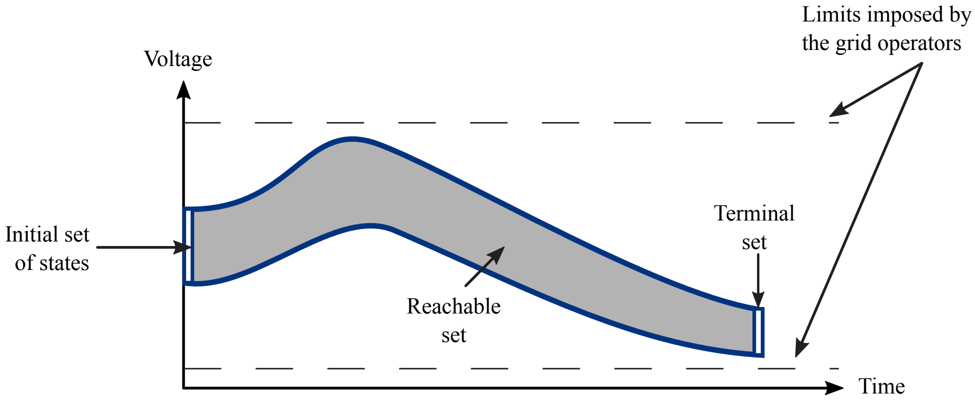


Figure 1.3: Illustration of the reachable set to assess voltage levels in power systems.

directly based on one another; instead, each chapter can be considered as a stand-alone contribution, such that the reader can go through this thesis in any order. In each chapter, we present relevant related work through a detailed literature review, then we formulate the problem and describe the proposed algorithmic procedure. Finally, we demonstrate the applicability of the approach on various benchmark examples, commonly employed in the power system community. In the final chapter, we illustrate the applicability of reachability analysis on a real power plant based in Munich, Germany.

Reachability Analysis of Power Systems

The purpose of this preliminary chapter is to introduce the reader, unfamiliar with reachability analysis, to the computation of reachable sets of nonlinear index-1 differential algebraic equations (DAEs), the standard modelling formalization of power systems. In fact, DAEs are useful in modelling a large variety of dynamic phenomena scattered throughout the engineering disciplines [18, 24]; thus, the presented algorithm can in principle be applied to a wide range of applications, such as e.g. chemical reactors and robotic manipulators. Since this thesis is primarily focused on transient stability, we will present standard models capable of capturing the dynamical behaviour of the electromechanical oscillations leading to instability of power systems. First we formulate the power-flow problem of the distribution network describing active and reactive power under steady-state conditions. Then we introduce the generalized swing equation of synchronous machines and asynchronous generators, employed in conventional power plants and wind turbines, respectively. Finally, we make necessary modelling assumptions of the remaining components in order to assess stability of the power system following the event of a large disturbance within the transmission network.

Compositional Reachability Analysis of Power Systems

The main challenge associated with the analysis of power systems via the computation of reachable sets is improving the algorithmic efficiency to scale towards industrially relevant problem sizes. In this chapter, we present a compositional procedure that drastically reduces the computational effort required to assess the dynamical response of power systems using reachability analysis. The main reason for the algorithmic efficiency is that we reformulate the transmission network into a set of subsystems, each consisting of a generating unit connected to its corresponding generator bus, whose algebraic constraints are unknown-but-bounded within some confidence intervals. This makes it possible to parallelize the computation of reachable sets and, most importantly, preserve the interaction and the correlation between different machines connected to the grid. The applicability of the proposed compositional algorithm is illustrated on several benchmark examples such as the IEEE 14-bus and the IEEE 30-bus, which is comprised of more than 100 state variables. Furthermore, our method is compared to an alternative algorithm in which the reachable set is computed without employing any compositional techniques.

The contributions of this chapter are published in [42]. This chapter is based on a collaboration with the department of Electrical Engineering at the University of British Columbia.

Estimation of the Region of Attraction

As mentioned earlier, Lyapunov direct method is the dominant technique for establishing transient stability of power systems with formal guarantees; however, this technique has several drawbacks limiting its applicability in practice. The technique basically computes the so-called Region of Attraction (ROA); that is, the region surrounding an equilibrium point from which any initial state can be attracted by this equilibrium. The estimation of these stability regions is of fundamental importance in power systems as it can immediately determine stability of the system following a perturbation in the grid. In this chapter, we present for the first time an algorithm to estimate the ROA of an equilibrium point via the computation of forward reachable sets. We describe and implement a scalable and versatile algorithm that can provide accurate, and more importantly, provable estimates of the stability region. By versatile, we refer to the ability of the algorithm to deal with general systems involving non-polynomial models, thus covering a wide range of applications. Furthermore, we compare three different techniques that can provide estimates of the stability region; namely, we compare the proposed forward reachability analysis with the established backward reachability analysis, and the Lyapunov direct method.

The bulk of this chapter is based on the contribution published in [44]. Contributions associated with the Lyapunov method are published in [58, 59].

Formal Linear-parameter Varying Control of Power Systems

This chapter is concerned with establishing transient stability of power systems using specialized controllers. To this end, we propose the design and verification of linear-parameter varying (LPV) controllers to robustly establish transient stability of multi-machine power systems with formal guarantees. First, we transform power systems described by standard DAEs into modular LPV systems, such that the interaction between different machines connected to the grid is preserved. Then, we employ reachability analysis to determine the set of the time-varying parameters required for the LPV controller synthesis. Afterwards, reachability analysis is also used to formally guarantee that the synthesized controller encloses the time-varying parameters within chosen parameter ranges during transients. Both tasks are solved simultaneously in a systematic fashion within the context of a unified framework. Several benchmark examples are considered to showcase the applicability and scalability of the proposed framework.

The contributions of this chapter are published in [47,123]. This chapter is based on a collaboration with the institute of Control and Systems Theory at Universität Kassel .

Formal Analysis and Control of Combined Cycle Power Plants

This chapter illustrates the design, verification, and implementation of a centralized multivariable feedback controller to optimize the dynamical performance of a realistic configuration of a boiler system located within a 450 MW combined cycle power plant in Munich, Germany. Namely we consider the steam-drum unit known to degrade the load-following capabilities of conventional power plants; thus, limiting their flexibility to meet the strict requirements imposed by the corresponding TSO. Furthermore, emergency shutdowns are typically trigged in thermal plants due to poor regulation of the water level inside the drum unit during fast-load changes when employing the industry standards PID-controllers. This causes the water level inside the drum to exceed its safety limits; hence, the system is particularly suited to employ reachability analysis as a means of verifying safety under various loading conditions. Additionally, the system is known to be challenging from a control perspective due to its nonlinear behaviour, coupling of its inputs channels, and most importantly its non-minimum phase response.

This chapter is based on [43, 45, 46]. The contributions of this chapter are based on some results that were published during involvement of the author with Stadtwerke München GmbH, Friedrich-Wilhelm-Bessel-Institut Forschungsgesellschaft mbH, and the institute of Automation at Universität Bremen.

Chapter 2

Reachability Analysis of Power Systems

This chapter introduces standard power system models used throughout this thesis, in addition to the computation of reachable sets for nonlinear differential algebraic equations (DAEs); DAEs are useful to model a large variety of dynamic phenomena scattered throughout the engineering disciplines, such as chemical reactors and power systems. Generally, DAEs are encountered in practical applications when the differential variables are subjected to algebraic constraints. Particularly, DAEs arise in power systems, since the electrical current injections at each grid node within the transmission network are constrained by Kirchhoff's law; thus, the nonlinear equations governing the dynamical behavior of the system are comprised of differential variables and algebraic constraints whose time-derivatives do not appear.

2.1 Modelling of Power Systems

Modern power systems can vary in structural size and components functionality, yet they are always categorized into three subsystems as illustrated in Fig. 2.1; namely, the generation, the transmission, and the distribution. The generation subsystem includes the base-load power sources capable of supplying the minimum level of demand to the utility grid. It basically consists of conventional power stations, such as coal-fired, gas, hydro, and nuclear power plants. These plants are based on the same working principle: a prime mover converts potential energy of dammed water or produced super-heated steam into mechanical energy, which in turn is converged into electricity via the synchronous generator. The transmission network is regarded as the backbone of the power system; it consists of the step-up transformers at the generation side which aim at reducing the losses of the transmitted power via transmission lines over long distances to the load centers. Additionally, protective circuit breakers are installed to clear faults occurring in the network, e.g. 3-phase fault, or line-to-ground fault. Finally, the distribution handles delivery of electricity to the end customers by stepping down the transmitted power to the medium-voltage (MV) level for industrial customers and to the low-voltage (LV) level for residential and commercial customers. Furthermore, it consists of distributed generation (DG) units using renewable resources which started to increasingly play an important role in the utility grid due to their modularity, flexility, and positive impact on the environment.

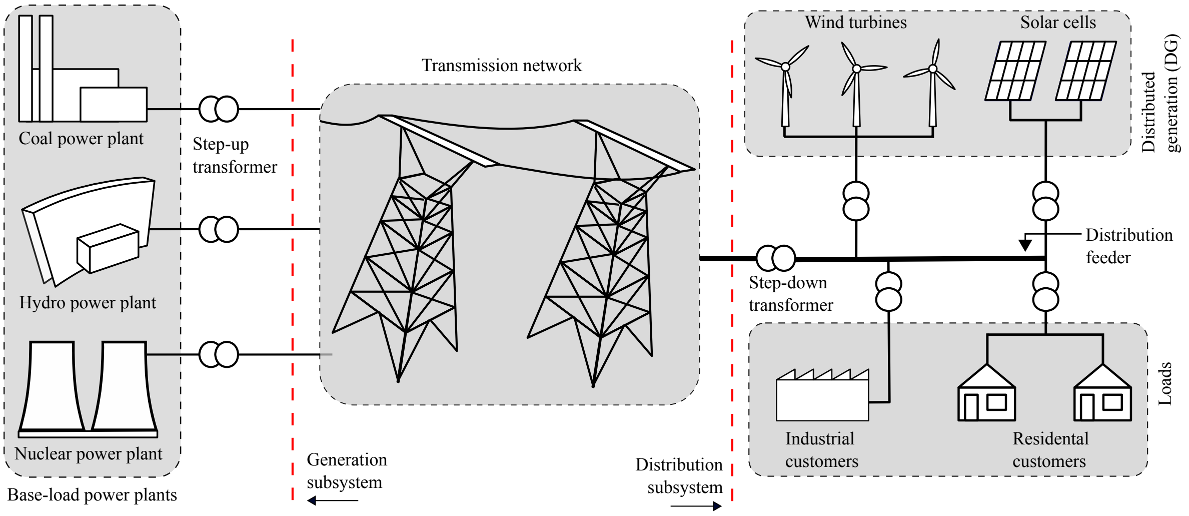


Figure 2.1: Structure of modern power systems.

In the following sections we present standard models used throughout this thesis for the analysis and control of power systems with formal guarantees. Since this thesis focuses on studies involving transient stability, we primarily consider models capable of capturing the dynamical behavior of the electromechanical oscillations leading to instability of power systems. First, we formulate the power-flow problem of the distribution network describing active and reactive power under steady-state conditions. Then, we introduce the generalized swing equation of synchronous machines and asynchronous generators, employed in conventional power plants and wind turbines, correspondingly. Finally, we make necessary modelling assumptions of the remaining components, i.e. transformers and loads, in order to assess stability of the power system following the event of large disturbance within the transmission network.

2.1.1 Power-flow Model

The power-flow (load-flow) analysis is an important tool to assess the static performance of power systems. Generally speaking, the power-flow analysis examines whether a power system, under steady-state conditions, adheres to the following specifications:

- Generation units are capable of supplying demands and losses of the transmission network, while operating within their specified active and reactive power limits.
- The voltage magnitudes at any bus are close to their rated value.
- The transformers and transmission lines are not overloaded.

In order to formalize the power-flow equations, we represent a generic power system via the standard single-line diagram notation as illustrated in Fig. 2.2. Here the distribution network can be regarded as a simple graph formalized using basic concepts from graph theory. The network consists of N buses (graph vertex) represented by the set $\mathcal{N} = \{1, \ldots, N\}$, and T transmission lines (graph edges). The transmission lines are represented by the Π-model with lumped parameters [81, p. 256], in which each line spanned between two buses comprises of a series impedance $\bar{z}_{m,n} \in \mathbb{C}$ and a shunt admittance $\bar{y}^{\text{sh}} \in \mathbb{C}$ expressed by:

$$\begin{aligned} \bar{z}_{m,n} &:= r_{m,n} + jX_{l_{m,n}}, & & \text{series impedance [p.u.]} \\ & & \Longrightarrow & \\ \bar{y}^{\text{sh}}_{m,n} &:= jX_{c_{m,n}}, & & \text{shunt admittance [p.u.]} \end{aligned} \tag{2.1}$$

where r, X_l and X_c corresponds to the line series resistance, series reactance, and shunt susceptance, respectively. Here the $\bar{\bullet}$ denotes complex-valued quantities, with $j = \sqrt{-1}$ being the unit imaginary number. The subscripts m and n corresponds to the edges of the transmission line spanned between the m-th and n-th bus. Note that the line impedance and admittance are expressed according to the per-unit [p.u.] system. Conversion to the per-unit system is a standard procedure in power system which basically expresses system quantities as fractions of a defined base unit quantity; thus, the representation of elements become more uniform regardless of the unit size [81, p. 75].

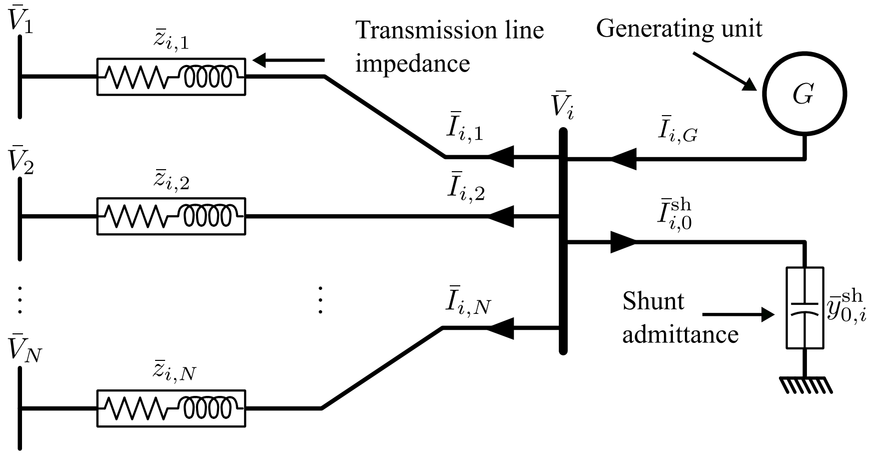


Figure 2.2: Power flow formulation at the i-th bus.

We denote the current injected at the i-th bus by $\bar{I}_{G,i}$ [p.u.] $\in \mathbb{C}$. The current is obtained based on Kirchhoff's current law as follows:

$$0 = \bar{I}_{G,i} - \bar{I}^{\text{sh}}_{i,0} - \sum_{\substack{k \in \mathcal{N} \\ k \neq i}} \bar{I}_{i,k}, \tag{2.2}$$

with $\bar{I}_{i,k}$ and $\bar{I}^{\text{sh}}_{i,0}$ being the currents flowing through the series impedance and the shut admittance, correspondingly. Next we introduce the voltage $\bar{V}_i := V_i e^{j\theta_i}$ [p.u.] $\in \mathbb{C}$, with V_i, $\theta_i \in \mathbb{R}$ as the voltage absolute value and its phase angle, respectively. Recall from Ohm's law that the current through a conductor is directly proportional to the potential difference between its nodes; that is:

$$\bar{I}_{i,k} = \left(\bar{V}_i - \bar{V}_k\right) \bar{y}_{i,k}, \tag{2.3}$$

where $\bar{y}_{i,k} := {}^{1}/_{\bar{z}^{\text{sr}}_{i,k}}$ is the constant of proportionality satisfying Ohm's law. Then by replacing the currents by their expressions (2.2) in (2.3), the injected current at the i-th bus can be rewritten as

$$\begin{aligned}\bar{I}_{G,i} =& \bar{V}_i \left[\bar{y}^{\text{sh}}_{i,0} + (\bar{y}_{i,1} + \ \dots \ + \bar{y}_{i,i-1} + \bar{y}_{i,i+1} + \ \dots \ + \bar{y}_{i,N})\right] - \\ & \bar{V}_1\bar{y}_{i,1} - \dots - \bar{V}_{i-1}\bar{y}_{i,i-1} - \bar{V}_{i+1}\bar{y}_{i,i+1} - \dots + \bar{V}_N\bar{y}_{i,N} \\ =& \underbrace{\bar{V}_i \left(\bar{y}^{\text{sh}}_{i,0} + \sum_{\substack{k\in\mathcal{N}\\k\neq i}} \bar{y}_{i,k}\right)}_{=:\bar{y}_{i,i}} - \sum_{\substack{k\in\mathcal{N}\\k\neq i}} \bar{V}_k\bar{y}_{i,k}.\end{aligned} \tag{2.4}$$

Throughout the same steps, the injected currents at all N buses of the system can be formalized in compact form via

$$\begin{bmatrix} \bar{I}_{G,1} \\ \bar{I}_{G,2} \\ \vdots \\ \bar{I}_{G,N} \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{y}_{1,1} & -\bar{y}_{1,2} & \dots & -\bar{y}_{1,N} \\ -\bar{y}_{2,1} & \bar{y}_{2,2} & \dots & -\bar{y}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ -\bar{y}_{N,1} & -\bar{y}_{N,2} & \dots & \bar{y}_{N,N} \end{bmatrix}}_{=:\bar{Y}} \begin{bmatrix} \bar{V}_1 \\ \bar{V}_2 \\ \vdots \\ \bar{V}_N \end{bmatrix}, \tag{2.5}$$

with $\bar{\boldsymbol{Y}}$ [p.u.] being the so-called nodal admittance matrix, with the entry $\bar{Y}_{ik} := Y_{ik}e^{j\Theta_{ik}}$ where Y_{ik} and Θ_{ik} denote the absolute value and phase angle, respectively. It can be easily seen that the diagonal entries will be non-zero only if a physical link exists between two buses; thus, in practice the matrix $\bar{\boldsymbol{Y}}$ is typically a very sparse matrix with almost more than 99% of its elements are zeros.

Finally, with the derivation of the current injections in terms of the bus voltage, it still remains to formalize the power injections at the i-th bus. Recall that the complex power is

$$\bar{S}_i = \bar{V}_i\bar{I}^*_{G,i}, \tag{2.6}$$

with $\bullet^*$ returning the conjugate of a complex quantity. Noting that $\bar{V}_i = Ve^{j\theta_i}$ and $\bar{Y}_{ik} := Y_{ik}e^{j\Theta_{ik}}$, then inserting (2.5) in (2.6) yields

$$\bar{S}_i = V_i \sum_{k\in\mathcal{N}} V_k Y_{ik} e^{j(\theta_i - \theta_j - \Theta_{ik})} := P_i + jQ_i \tag{2.7}$$

where P_i [p.u.] and Q_i [p.u.] denoting the active and reactive power injections, respectively. Thus, the injected active and reactive power at the i-th bus are formalized by expressing (2.7) via

$$\begin{aligned} P_i &= p_i(\theta_1, \ldots, \theta_N, V_1, \ldots, V_N), \\ &= V_i \sum_{k \in \mathcal{N}} Y_{ik} V_k \cos(\Theta_{ik} + \theta_k - \theta_i), \end{aligned} \tag{2.8}$$

$$\begin{aligned} Q_i &= q_i(\theta_1, \ldots, \theta_N, V_1, \ldots, V_N), \\ &= V_i \sum_{k \in \mathcal{N}} Y_{ik} V_k \sin(\Theta_{ik} + \theta_k - \theta_i). \end{aligned} \tag{2.9}$$

Note that the formulation of (2.7) is based on Euler's formula $e^{j\varphi} = \cos(\varphi) + j\sin(\varphi)$ and symmetry of trigonometric functions, e.g. $\cos(-\varphi) = \cos(\varphi)$.

Remark 2.1. In the power flow formulation, ones has to specify the known and unknown variables of the system, which is done by identifying the type of each bus. First a bus known as the slack bus is assigned to be the reference of the whole system; hence, its voltage level and phase angle are known and the corresponding equations are not included in the power flow formulation. Any bus connected to a generating unit is denoted as a PV-bus, whose active power and voltage level are known. Finally, all remaining buses are labeled as PQ-buses, where the active and reactive are known.

Remark 2.2. The nonlinear power flow equations (2.8), (2.9) generally offer no closed-form solution. They are solved using numerical methods such as Newton-Raphson, Gauss-Seidel, and fast decoupled load-flow, see [81, p. 267].

So far we only formalized the power flow to express the injected active and reactive power at each bus. It still remains to consider dynamics of the generating units to fully describe the behaviour of the generic system illustrated in Fig. 2.2. In this thesis, we only take into account dynamics of conventional power plants employing the standard AC synchronous machine, and wind turbines connected to asynchronous generators. The photovoltaic systems are not taken into account since their time-constant is much faster[1] compared to the former generating units; thus, their influence on the overall dynamical behavior can be neglected.

2.1.2 Synchronous Generator Model

Synchronous generators are the main source of generating electrical energy. The machine schematic diagram considered in this thesis is illustrated in Fig. 2.3. Here the machine consists of three essential elements; namely, the stator housing the three-phase armature windings distributed 120° in space, the rotor coupled with a prime mover via a rotor shaft, and the exciter carrying direct current to induce a magnetic field in the stator windings. Under normal operation, conventional power plants converge the potential energy of dammed water or produced super-heated steam into mechanical energy capable of

[1] Typically within the range of milliseconds resulting from the fast-acting switches of the power converters.

turning the prime mover. This results in the generation of a rotational magnetic field inducing the stator armature windings; thus, producing electricity according to Faraday's law of mutual induction. Notice that the term synchronous refers to the fact that the rotor and the magnetic field rotate with the same speed.

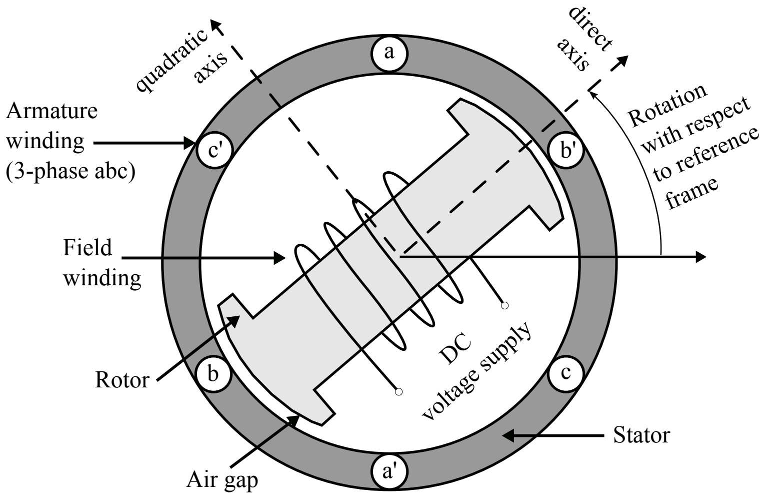


Figure 2.3: Construction of a synchronous generator.

One of the main challenges associated with modelling of the synchronous machine, is the fact that all quantities vary with the angular position of the rotor shaft θ_m [rad/s], which in turn varies in time. This generally leads to considerable complexity in deriving mathematical models suitable for dynamic security assessment. This problem is solved via the standard Park's $dq0$-transformation based on the two Reaction theory [81, p. 67]. Here, it is proposed to project the electrical quantities into three components referred to as direct d-, quadratic q- and homopolar 0- axes, respectively, with the d-axis located along the rotor axis and the q-axis perpendicular to it. This makes it possible to eliminate the effects of the time-varying inductances, thus reducing complexity of the differential equations governing dynamics of the synchronous generator since the transformation maps the three-phase stator and rotor quantities into a single synchronously rotating reference.

Equivalent circuit

There exists a wide variety of models describing the dynamical behavior of the synchronous generator, see [96, Table 15.2]. Each model considers a set of assumptions depending on the specified analysis. In particular we employ the standard d-axis model commonly used for studies involving transient stability. In this modelling framework, the synchronous generator is described by the equivalent circuit as shown in Fig. 2.4. Here the generator is represented by a single phase AC voltage source connected in series with a transient impedance. The machine parameters and their meanings are described in Table 2.1.

Table 2.1: Machine parameters of the synchronous generator

Variable	Description	Unit
ω_s	Base synchronous frequency	[rad/s]
ω_{ref}	Reference angular speed	[p.u.]
H	Inertia coefficient	[MWs/MVA]
D	Damping coefficient	[p.u.]
τ_d	d-axis open-circuit transient time constant	[s]
X_d	d-axis reactance	[p.u.]
X'_d	d-axis transient reactance	[p.u.]
r_a	armature windings resistance	[p.u.]

Differential Equations

The differential equations of the d-axis model of the synchronous generator are expressed by [96, p. 334]:

$$\begin{aligned}\dot{\delta}_j &= \omega_s\left(\omega_j - \omega_{\text{ref}}\right),\\ \dot{\omega}_j &= \frac{1}{2H_j}\left(T_{m,j} - T_{e,j} - D_j(\omega_j - \omega_{\text{ref}})\right),\end{aligned} \tag{2.10}$$

$$\dot{E}'_{q,j} = \frac{1}{\tau'_{d,j}}\left(v_{f,j} - E'_{q,j} - i_{d,j}(X_{d,j} - X'_{d,j})\right), \tag{2.11}$$

where δ [rad] is the rotor angular position, ω [rad/s] is the rotor angular velocity, and E' [p.u.] is the machine transient voltage. The system inputs are the field voltage v_f [p.u.] and the torque T_m [p.u.]. Here the subscript j, m, and e are corresponds to the j-th machine, the mechanical and electrical components, respectively, and d and q denote the d- and q-axes, associated with Park's transformation. The differential equations (2.10) consider the electromechanical oscillations of the system via the so-called swing equation, and (2.11) handles modelling of the AC voltage source.

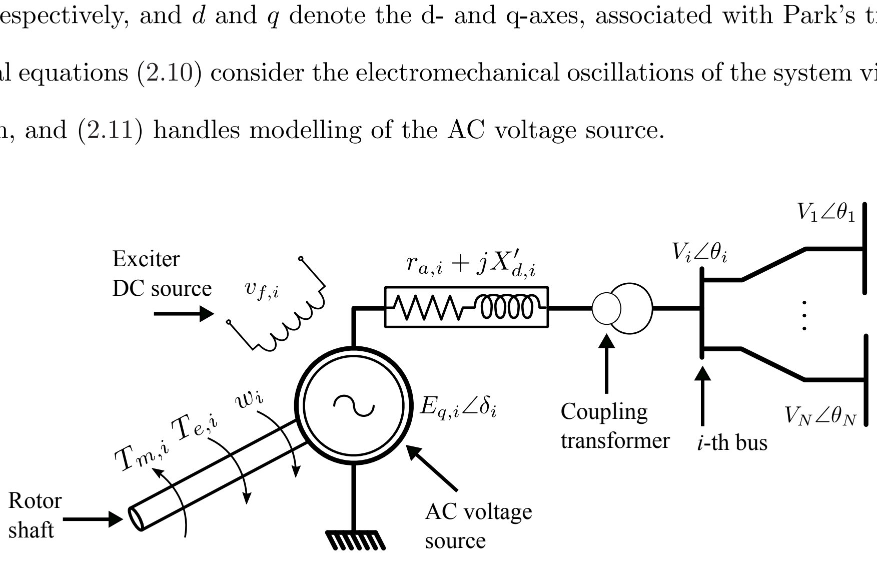


Figure 2.4: Equivalent circuit of the synchronous generator at the i-th bus.

Algebraic Equations

The remaining variables are obtained by solving a set of algebraic equations; first the electrical torque T_e [p.u.] is obtained as follows:

$$0 = T_{e,j} - (v_{d,j} + r_{a,j} i_{d,j}) i_{d,j} - (v_{q,j} + r_{a,j} i_{q,j}) i_{q,j}, \tag{2.12}$$

where the stator voltages v_d [p.u.] and v_q [p.u.] are computed based on the voltage level of their corresponding h-th bus, that is:

$$\begin{aligned} 0 &= v_{d,j} - V_h \sin(\delta_j - \theta_h), \\ 0 &= v_{q,j} - V_h \cos(\delta_j - \theta_h), \end{aligned} \tag{2.13}$$

and the stator currents i_d [p.u.] and i_q [p.u.] are obtained via simple nodal analysis of the synchronous generator equivalent circuit (see Fig. 2.4), yielding the following algebraic equations:

$$\begin{aligned} 0 &= v_{q,j} + r_{a,j} i_{q,j} + X'_{d,j} i_{d,j} - E'_{q,j}, \\ 0 &= v_{d,j} + r_{a,j} i_{d,j} - X_{q,j} i_{q,j}. \end{aligned} \tag{2.14}$$

Finally, with the knowledge of the currents and voltages within the armature windings, the active and reactive power can be computed as

$$\begin{aligned} 0 &= P_{e,j} - v_{d,j} i_{d,j} - v_{q,j} i_{q,j}, \\ 0 &= Q_{e,j} - v_{q,j} i_{d,j} + v_{d,j} i_{q,j}. \end{aligned} \tag{2.15}$$

2.1.3 Wind Turbine Model

As mentioned earlier, there is an ongoing trend towards more environmentally-driven production of electricity, such as wind and solar generation, in the hope of reducing the CO_2 emissions and contributing effectively against global warming. Wind turbines are generally the most common wind energy conversion system (WECS) converting the wind kinetic energy into electricity. In contrast to conventional power plants, wind turbines mostly employ asynchronous (induction) generators where the rotor speed is no longer synchronized with the magnetic field of the stator. This makes the induction generators attractive for wind generating stations since they are capable of producing power at varying rotor speeds compared to synchronous generators. Wind turbines are divided into four different type; the fixed speed Type-1, limited variable speed Type-2, or variable speed with either partial or full power electronic conversion, Type-3 and Type-4, correspondingly.

In this thesis we employ the Type-3 wind turbine using the doubly-fed induction generator as the WECS, which is a generating principle widely used in wind turbines [27, 96, 104, 108]. The operating principle is illustrated in Fig. 2.5. The system consists of three main elements; namely, the drive train, the generator, and the back-to-back converter. The drive train has its the low speed shaft facing the wind, and it is responsible of making the high speed shaft turn approximately between 40 to 50 times faster than the low speed shaft depending on the manufacturer. This in turn rotates the rotor of the induction generator coupled with the high speed shaft. The generating unit is a wound-rotor induction generator, where the stator side is connected directly to the point of common coupling (PCC) prior to the connection with the utility grid, and the rotor side is connected to the PCC via special power converters. The power converter is commonly the back-to-back voltage source converter, which is comprised of two separate bi-directional converters coupled via a DC link. This makes it possible to control the rotor speed where the rotor frequency can freely differ from the frequency of the utility grid. Furthermore, using this topology, one may adjust the rotor currents, which in turn indirectly specifies the active and reactive power fed to the grid from the stator, independently of the rotor turning speed [96, Ch. 20].

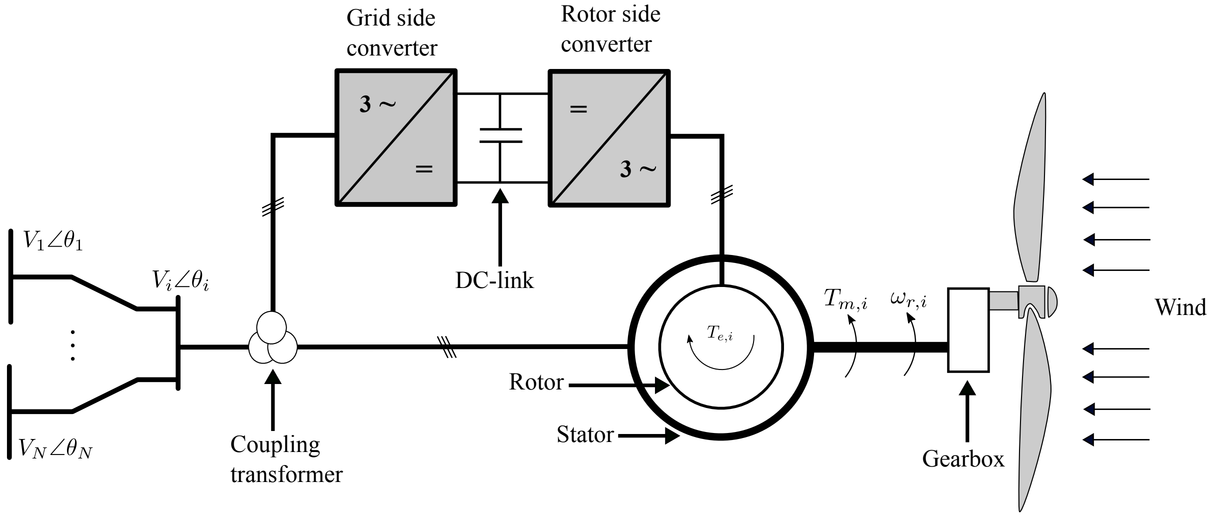


Figure 2.5: Schematic diagram of a DFIG-based wind generation system at the i-th bus.

Equivalent circuit

The WECS based on the doubly-fed induction generator can be described as a third order model including dynamics of the drive train, the asynchronous generator, and the power converter [51]. In this modelling framework, the doubly-fed induction generator is equivalent to the electrical circuit shown in Fig. 2.4. Here, the circuit is supplied via the PCC, and the winding included in the stator and the rotor are aggregated via a constant impedance. The machine parameters and their meanings are described in Table 2.2.

One can notice from the equivalent circuit that the rotor voltage depends on a new variable s associated with the so-called slip ratio defined as

$$s_j = \frac{\omega_{s,j} - \omega_{r,j}}{\omega_{s,j}}, \tag{2.16}$$

Table 2.2: Machine parameters of the doubly-fed induction generator

Variable	Description	Unit
H_w	Sum of turbine and rotor inertia constant	[MWs/MVA]
X_ν	Magnetizing reactance	[p.u.]
X_s	Stator reactance	[p.u.]
X_r	Rotor reactance	[p.u.]
r_s	Stator resistance	[p.u.]
r_r	Rotor resistance	[p.u.]

with ω [p.u.] being the rotational speed. The subscripts s, r, and j corresponds to the stator, the rotor, and the j-th machine, respectively. As stated earlier, the key distinction of asynchronous generators compared to synchronous machines, is that the magnetic field of the rotor is no longer synchronized with that of the stator. In fact, the slip is a very important parameter in the electrical circuit, because it relates how fast the rotor is spinning with the electrical side. Without the slip, the equivalent circuit of the doubly-fed induction generator becomes identical to a transformer circuit, which is a motionless device simply varying the voltage levels from the stator to the rotor.

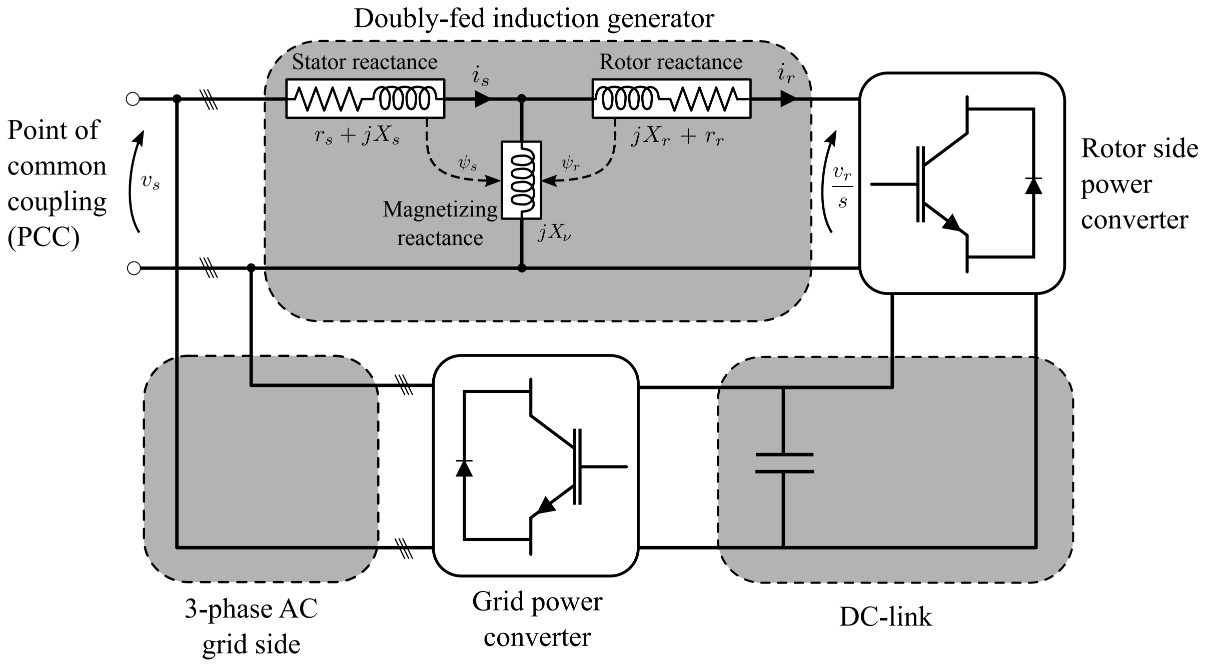


Figure 2.6: Equivalent circuit of the DFIG of the wind energy conversion system.

Differential Equations

Prior to introducing the differential equations governing the mathematical model, some assumptions concerning the doubly-fed induction generator are made. The DC/AC converter on the grid side is assumed to operate loss-less and completely synchronized with the grid, hence the active power flowing in the back-to-back converter is equal and the reactive power of the DC/AC converter is zero. Furthermore, the transient behaviour associated with the stator flux is neglected, i.e. $\dot{\psi}_s \overset{!}{=} 0$, due to the fact that the wind turbine is connected through the stator to the grid, which is modeled by algebraic variables via the power flow equations (2.8).

With these basic assumptions, the WECS based on the doubly-fed induction generator can be described via the following third order model [96, Ch. 20]

$$\dot{\omega}_{r,j} = \frac{1}{2H_{w,j}}(T_{m,j} - T_{e,j}), \tag{2.17}$$

$$\begin{aligned} \dot{\psi}_{r,d,j} &= v_{r,d,j} + r_r i_{r,d,j} + s_j \omega_{s,j} \psi_{r,q,j}, \\ \dot{\psi}_{r,q,j} &= v_{r,q,j} + r_r i_{r,q,j} - s_j \omega_{s,j} \psi_{r,d,j}, \end{aligned} \tag{2.18}$$

where ψ_r [p.u.] is the rotor flux. The system inputs are the rotor voltage v_r [p.u.] and the torque T_m [p.u.]. Here the subscripts j, m, and e are corresponds to the j-th machine, the mechanical and electrical components, respectively, and d and q denote the d- and q-axes, associated with Park's transformation. The differential equation (2.17) considers the electromechanical oscillations of the system via the so-called swing equation and the equations (2.18) handle modelling of the machine magnetic flux on the rotor side.

Algebraic Equations

Similarly to the model of the synchronous generator, the remaining variables are obtained by solving a set of algebraic equations; first the electrical torque T_e is obtained via:

$$0 = T_{e,j} - X_{\nu,j}\left(i_{r,q,j} i_{s,d,j} - i_{r,d,j} i_{s,q,j}\right), \tag{2.19}$$

where i [p.u.] is the current. The stator currents are obtained by applying the nodal analysis on the equivalent circuit (see Fig. 2.6) yielding the following equations

$$\begin{aligned} 0 &= v_{s,d,j} + r_{s,j} i_{s,d,j} + \omega_{s,j} \psi_{s,q,j}, \\ 0 &= v_{s,q,j} + r_{s,j} i_{s,q,j} + \omega_{s,j} \psi_{s,d,j}, \end{aligned} \tag{2.20}$$

with v_s [p.u.] as the stator voltage computed based on the voltages levels at their corresponding h-th bus

$$\begin{aligned} 0 &= v_{s,d,j} - V_h \sin(\theta_h), \\ 0 &= v_{s,q,j} - V_h \cos(\theta_h). \end{aligned} \tag{2.21}$$

It still remains to compute the rotor currents i_r [p.u.] and the stator flux ψ_s [p.u.]. The aforementioned variables are obtained by solving the following set of equations:

$$\begin{aligned}
0 &= \psi_{r,d,j} + \left(X_{r,j} i_{r,d,j} + X_{\nu,j} i_{s,d,j}\right), \\
0 &= \psi_{r,q,j} + \left(X_{r,j} i_{r,q,j} + X_{\nu,j} i_{s,q,j}\right), \\
0 &= \psi_{s,q,j} + \left(X_{s,j} i_{s,q,j} + X_{\nu,j} i_{r,d,j}\right), \\
0 &= \psi_{s,d,j} + \left(X_{s,j} i_{s,q,j} + X_{\nu,j} i_{r,q,j}\right),
\end{aligned} \tag{2.22}$$

and with the knowledge of the currents and voltages within the windings of the stator and the rotor, the active and reactive power can be computed according to:

$$\begin{aligned}
0 &= P_{e,j} - v_{s,d,j} i_{s,d,j} - v_{s,q,j} i_{s,q,j} - v_{r,d,j} i_{r,d,j} - v_{r,q,j} i_{r,q,j}, \\
0 &= Q_{e,j} - v_{s,q,j} i_{d,j} + v_{r,d,j} i_{q,j}.
\end{aligned} \tag{2.23}$$

2.1.4 Transformers and Loads

The modelling of transformers and loads is an important topic in power systems, since both elements are heavily present in any transmission network. However, a detailed model of these elements is only critical with regards to studies involving voltage stability analysis; this due to the fact that transformers and loads directly affect the long-term voltage levels. Suppose a fault occurs in a power system resulting in a voltage drop, this will initially result in a decay of the load, then after a few seconds, a load restoration process will commence which can lead to heavily loaded conditions, and even worse to a voltage collapse if these loading conditions were not addressed using appropriate control decisions.

Since this thesis is primarily concerned with transient stability (short-term disturbances), several assumption can be made -and justified- to simplify the modelling of loads and transformers, see [81, Ch. 12] and [96, Ch. 10]. In particular, we consider the so-called voltage-dependant loads, in which during the standard transient stability analysis, the loads absorbing active and reactive power at the i-th bus are represented by [96, p. 258]

$$\begin{aligned}
P_{i,l} &= P_{i,l_0} \left(\frac{V_i}{V_{i,l_0}}\right)^2, \\
Q_{i,l} &= Q_{i,l_0} \left(\frac{V_i}{V_{i,l_0}}\right)^2.
\end{aligned} \tag{2.24}$$

Here the subscripts l and 0 corresponds to the load and the initial value, respectively. The initial values -associated with the voltage, in addition to the active and reactive power- are obtained by solving the power-flow equations at $t = 0$, see Remark 2.2. With regards to transformers, we consider tap-changing transformers which can be regarded in transient stability as ideal circuits represented via constant impedances, within the time-frame of simulation of transient stability, see [81, p. 859].

2.2 Problem Formulation

The power system models presented in the previous section can be described in compact form as a set of time-invariant nonlinear DAEs:

$$\mathbf{0} = \boldsymbol{F}(\dot{\boldsymbol{\chi}}(t), \boldsymbol{\chi}(t), \boldsymbol{u}(t)), \tag{2.25}$$

where the vector $\boldsymbol{\chi} \in \mathbb{R}^{n_\chi}$ includes the state variables of a power system, e.g. the synchronous generator rotor speed δ and the bus voltage V, the vector $\boldsymbol{u} \in \mathbb{R}^{n_u}$ contains to the system inputs, such as controller set-points and disturbances, and $\mathbf{0}$ is a vector of zeros with proper dimension. Note that the time dependency is often omitted for simplicity of notation. Notice that if the Jacobian matrix corresponding to the time-derivative of the state variables is non-singular, i.e. $\det\left({}^{\partial \boldsymbol{F}}/{}_{\partial \dot{\boldsymbol{\chi}}}\right) \neq 0$, then, the DAE simplifies to an explicit set of ordinary differential equations (ODEs). In other words, DAEs can be interpreted as a set of ODEs subject to a set of algebraic constraints. Loosely speaking, the degree of complexity to transform (2.25) into an explicit ODE system is determined via the so-called DAE-index; the index refers to the number of differentiation steps required to find a description of the time-derivatives for all state variables. Clearly, as the DAE-index gets higher, the more difficult will it become to solve the set of DAEs numerically, since no analytical solution exists for this class of equations. Throughout this thesis we only consider index-1 DAEs; this is a fairly general assumption that holds for many practical problems, especially for the standard power system models presented in Sec. 2.1.

2.2.1 Objective

The objective of the chapter is to introduce the reader -unfamiliar with reachability analysis- to the computation of reachable sets for the class of index-1 DAE systems. Note that we employ reachability analysis throughout this thesis to

- **Chapter 3**: Analyze transient stability of power systems in a compositional manner.
- **Chapter 4**: Estimate the region of attraction of an equilibrium in power systems.
- **Chapter 5**: Synthesize a set of decentralized linear-parameter varying (LPV) controllers to robustly establish transient stability with formal guarantees of multi-machine power systems.
- **Chapter 6**: Verify safety of critical components found in power plants.

As mentioned earlier, reachability analysis basically determines the set enclosing all possible trajectories of differential and algebraic variables over a user-defined time-horizon. A definition of reachable sets is given as follows:

Definition 2.1. Reachable Set: Given an implicit DAE system described as in (2.25), the reachable set of differential and algebraic variables over the time-horizon $t \in [0, t_f]$, where t_f is the final time, starting from the set of consistent initial states $\boldsymbol{\mathcal{R}}(0)$ and the set of uncertain inputs $\boldsymbol{\mathcal{U}}$, is defined as:

$$\mathbf{reach}(\boldsymbol{\mathcal{R}}(0), \boldsymbol{\mathcal{U}}, t_f) := \left\{ \boldsymbol{\chi}(t) \; : \; \boldsymbol{\chi}(t) \text{ satisfies (2.25) within } [0, t_f] \text{ for } [\dot{\boldsymbol{\chi}}(0), \, \boldsymbol{\chi}(0), \, \boldsymbol{u}(t)] \in \boldsymbol{\mathcal{R}}(0) \times \boldsymbol{\mathcal{U}} \right\}.$$

□

It is worth noting that Def. 2.1 corresponds to the exact reachable set. In fact, except for very specific classes of systems, exact computation of reachable sets is difficult or even impossible [111]; thus, existing techniques aim at introducing traceable and efficient numerical procedures to compute an over-, or under-approximation of the reachable set as illustrated in Fig. 2.7. In this thesis we mainly consider computation of over-approximative reachable sets; that is in other words, an outer-approximation enclosing as tightly as possible all behaviours of the nonlinear DAE system such that:

$$\boldsymbol{\mathcal{R}}([0, t_f]) \supseteq \mathbf{reach}(\boldsymbol{\mathcal{R}}(0), \boldsymbol{\mathcal{U}}, t_f), \tag{2.26}$$

with $\boldsymbol{\mathcal{R}}([0, t_f])$ denoting a superset of the exact reachable set.

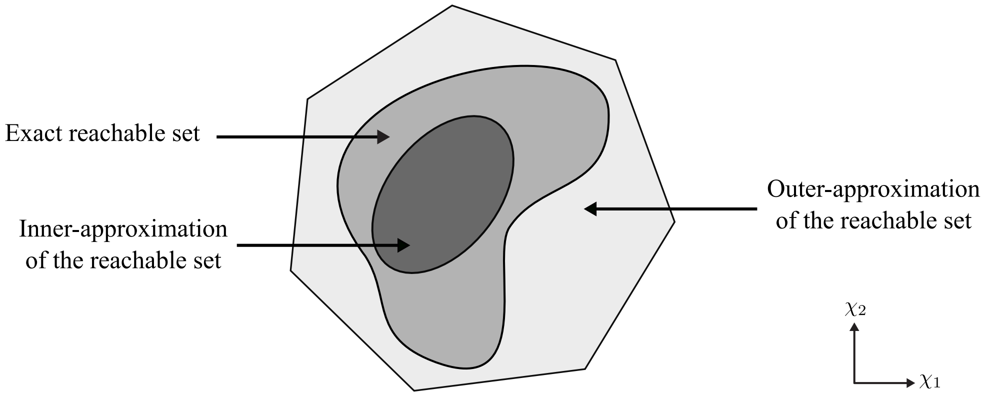

Exact reachable set
Inner-approximation of the reachable set
Outer-approximation of the reachable set
χ_2
χ_1

Figure 2.7: Projection of the inner- and outer-approximation of an exact reachable.

2.2.2 Existing Techniques

The techniques for reachability computation are generally categorized into two classes of methods [98]; that is either Eulerian schemes based on level set methods (LSMs) or Lagrangian techniques that follow the flow of the system's underlying dynamics. Shortly after we only consider Lagrangian reachability computation since the algorithm employed throughout this thesis is based on this class of techniques. Note that Eulerian methods are addressed later in this thesis; specifically in Ch. 4 when we consider the estimation of the ROA via the computation of backward reachable sets.

The Lagrangian techniques compute reachable sets similarly to numerical integration methods; that is in other words, by propagating the set of reachable states instead of only computing the solution for a single point over the specified time-horizon. One consequence of this is that Lagrangian approaches can handle higher-dimensional systems, where the associated memory requirements grow moderately with the system dimension, depending the choice of the reachable set representation discussed shortly after in the following section. A simplified illustration of this concept for a generic system is shown in Fig. 2.8.

There exists a large variety of well-developed methods that considers nonlinear ODE systems, such as abstraction using local linearization [13,56] and Taylor models [29], however, there is little work regarding an efficient algorithmic procedure for the computation of reachable sets for DAE systems that can scale towards industrially-relevant problem sizes. One obvious reason is that an extension of the reachability algorithms based on Lagrangian schemes for ODEs to handle DAEs is necessary; this task, however, is not straightforward since the class of DAE systems differs in both theoretical and numerical properties [37].

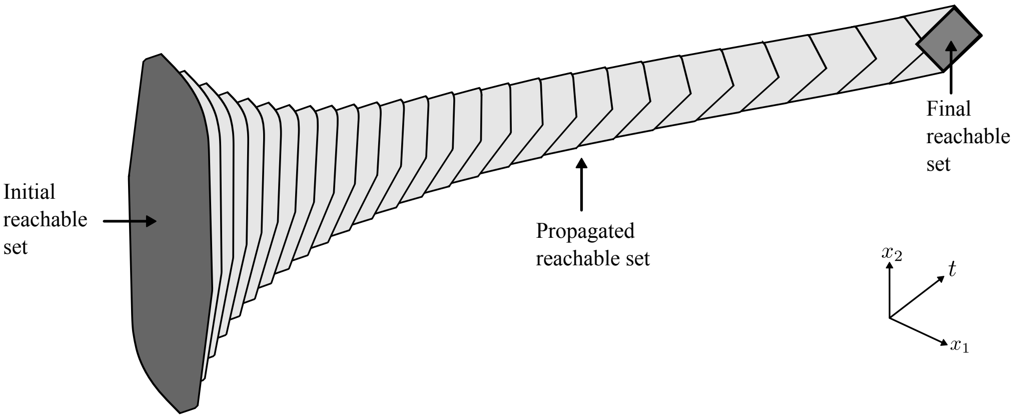


Figure 2.8: Illustration of the propagation of the reachable set spanned through the time-horizon $t \in [0,\ t_f]$.

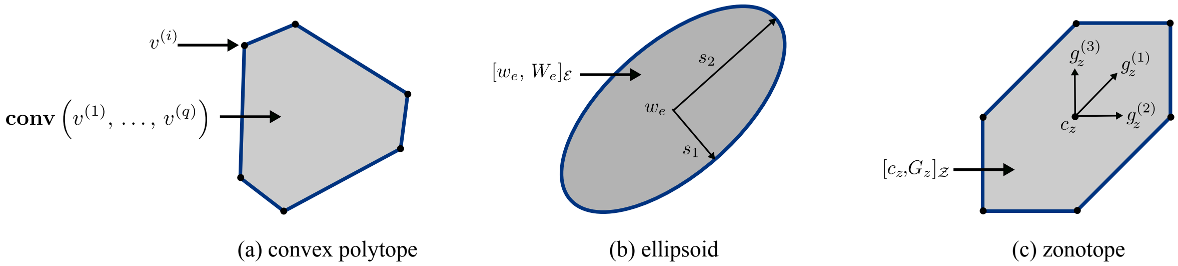
So far all the results reported in the literature tackle DAEs using two approaches in order to exploit the efficient methods developed for the class of ODEs: (1) They perform a local linearization around a stable equilibrium point in order to bring the system to a set of explicit ODEs via its index-1 property. (2) Alternatively, they make several assumptions to eliminate the set of algebraic equations inherently present in the system formulation; hence bringing the system to a set of ODEs. Note that a detailed literature review about existing works that employ reachability computation for the analysis of power system is provided later in Ch. 3.

In this thesis, we employ the first approach performing a local linearization. The reason behind this choice is that the assumptions employed in the literature to eliminate the set of algebraic constraints are often unrealistic and do not meet practical requirements. These assumptions are addressed later in this chapter when we illustrate the applicability of reachability analysis to several benchmark examples.

The remainder of this introductory chapter is organized as follows: First we introduce some basics about set representation in Sec. 2.3, then in Sec. 2.4 we recapitulate from [6, 12] well-know techniques for computing over-approximative reachable sets for the class of DAE systems. Finally in Sec. 2.5 we apply the reachability algorithms discussed throughout this chapter on some examples, with a particular focus on studies involving transient stability of power systems.

2.3 Set Representation and Basic Operations

We will recall typical set representation commonly employed in reachability computations; namely we consider polytopes, ellipsoids, zonotopes and multi-dimensional intervals which are illustrated in Fig. 2.9. Then, we briefly elaborate why zonotopes are generally preferable, in terms of accuracy and algorithmic efficiency for the class of DAEs describing standard power systems.


$v^{(i)}$
conv $\left(v^{(1)}, \ldots, v^{(q)}\right)$
$[w_e, W_e]_{\mathcal{E}}$
s_2
w_e
s_1
$g_z^{(3)}$
$g_z^{(1)}$
c_z
$g_z^{(2)}$
$[c_z,G_z]_{\mathcal{Z}}$

(a) convex polytope (b) ellipsoid (c) zonotope

Figure 2.9: Set representation using polytopes, ellipsoids, and zonotopes according to Def. 2.2, Def. 2.3, and Def. 2.4, respectively.

2.3.1 Convex Sets

First we introduce some basics about convex combinations and convex sets. Consider the vector $\boldsymbol{\vartheta} \in \mathbb{R}^n$, and a set of vectors $\boldsymbol{v}^{(i)} \in \mathbb{R}^n$, $i \in \{1, \ldots, q\}$; then $\boldsymbol{\vartheta}$ is said to be a convex combination of the q given vectors, if $\boldsymbol{\vartheta}$ can be expressed as:

$$\boldsymbol{\vartheta} = \sum_{i=1}^{q} \alpha_i \boldsymbol{v}^{(i)}, \quad \text{s.t.} \begin{cases} \alpha_i \geq 0, \\ \sum_{i=1}^{q} \alpha_i = 1, \end{cases} \tag{2.27}$$

with $\boldsymbol{\alpha}$ denoting the vector of coefficients of the convex combination.

2.3.2 Convex Polytopes

Based on the concept of convex combinations, we can introduce one of the most general set representations; in particular we refer to convex polytopes defined as follows:

Definition 2.2. Convex Polytope: Given a finite set of points (vertices) $\boldsymbol{v}^{(i)} \in \mathbb{R}^n$, $i \in \{1 \ldots q\}$ whose linear combination is expressed as in (2.27), then a convex polytope is defined as the convex hull of the finite set of q points expressed in the *V-representation* as

$$\boldsymbol{\mathcal{P}} = \mathbf{conv}(\boldsymbol{v}^{(1)},\, \ldots,\, \boldsymbol{v}^{(q)}) := \left\{ \boldsymbol{\vartheta} \in \mathbb{R}^n \;:\; \boldsymbol{\vartheta} = \sum_{i=1}^{q} \alpha_i \boldsymbol{v}^{(i)},\, \alpha_i \geq 0,\, \sum_{i=1}^{q} \alpha_i = 1 \right\}. \qquad \square$$

The operator $\mathbf{conv}(\,\cdot\,)$ denotes the convex hull and the q given vectors are also known as vertices of the polytope.

Remark 2.3. Convex polytopes and their generalization to describe system matrices, rather than points, will be intensively used in Ch. 5. There we shall consider the synthesis of LPV controllers with formal guarantees using closed-form expressions of convex combinations, see Sec. 5.3.

2.3.3 Ellipsoids

Now we consider set representation using ellipsoids which are primarily employed in the power system community to describe inner-approximations of the Lyapunov function sub-level sets [16]. This makes it possible to express, analytically, provable stability regions of post-fault scenarios. Further details will be addressed throughout Ch. 4 when we compare various techniques to estimate the so-called ROA of nonlinear systems, see Sec. 4.4.

Recently, ellipsoids have become a popular choice for reachability computations as well; namely for the analysis of the static and the dynamic performance of power systems, see for example [30, 69]. Formally, an ellipsoid is defined as follows:

Definition 2.3. Ellipsoid: Given a positive definite matrix $\boldsymbol{W}_e \in \mathbb{R}^{n\times n}$ and a vector $\boldsymbol{w}_e \in \mathbb{R}^n$ denoting the ellipsoid center, an ellipsoid is the set expressed via

$$\boldsymbol{\mathcal{E}} = [\boldsymbol{w}_e,\, \boldsymbol{W}_e]_{\mathcal{E}} := \left\{ \boldsymbol{x}_e \in \mathbb{R}^n \;:\; (\boldsymbol{x}_e - \boldsymbol{w}_e)^T \boldsymbol{W}_e^{-1} (\boldsymbol{x}_e - \boldsymbol{w}_e) \leq 1,\, \boldsymbol{W}_e > 0 \right\}. \qquad \square$$

Here the matrix $\boldsymbol{W}_e$ is directly associated with the shape of the ellipsoid, where its eigenvectors and eigenvalues specify directions and lengths of the ellipsoid semi-axes $\boldsymbol{s}_e$ according to

$$\forall i \in \{1 \ldots n\} : \quad s_{e,i} = \frac{\mathbf{v}^{(i)}}{\sqrt{\lambda_{e,i}}}, \text{ s.t. } \boldsymbol{W}_e \mathbf{v}^{(i)} = \lambda_{e,i} \mathbf{v}^{(i)},$$

with $\lambda_{e,i}$, and $\mathbf{v}^{(i)}$ corresponding to the i-th eigenvalue and eigenvector of the matrix $\boldsymbol{W}_e$, respectively, and $\boldsymbol{I}$ denoting the identity matrix with proper dimension.

2.3.4 Zonotopes

We now consider representation of sets using zonotopes which are regarded as a special case of convex polytopes; basically a zonotope is a centrally-symmetric polytope defined as follows

Definition 2.4. Zonotope: Given the so-called set of generators $\boldsymbol{g}_z^{(i)}$, $i \in \{1, \ldots, p_{\mathcal{Z}}\}$ and a vector $\boldsymbol{c}_z \in \mathbb{R}^n$ denoting the zonotope center, a zonotope is the set expressed according to the *G-representation*

$$\boldsymbol{\mathcal{Z}} = [\boldsymbol{c}_z, \boldsymbol{G}_z]_{\boldsymbol{\mathcal{Z}}} := \left\{ \boldsymbol{x}_z \in \mathbb{R}^n \, : \, \boldsymbol{x}_z = \boldsymbol{c}_z \oplus \sum_{i=1}^{p_{\mathcal{Z}}} \boldsymbol{g}_z^{(i)} \alpha_i, \, -1 \leq \alpha_i \leq 1 \right\}. \qquad \square$$

Here $\boldsymbol{G}_z := \left(\boldsymbol{g}_z^{(1)}, \ldots, \boldsymbol{g}_z^{(p_{\mathcal{Z}})}\right) \in \mathbb{R}^{n \times p_{\mathcal{Z}}}$ specifies the generator matrix, and the operator $\oplus$ corresponds to the Minkowski sum. It can be seen from Def. 2.4 that zonotopes have a very special structure; they can be interpreted as the geometric sum of a finite number of line segments, each defined by

$$\forall i \in \{1, \ldots, n\} \, : \quad l^{(i)} = \boldsymbol{U} \cdot \boldsymbol{g}_z^{(i)}. \tag{2.28}$$

with $\boldsymbol{U} := [-1; 1]$ as the unit interval. In fact (2.28) defines the step-by-step construction of zonotopes as illustrated in Fig. 2.10.

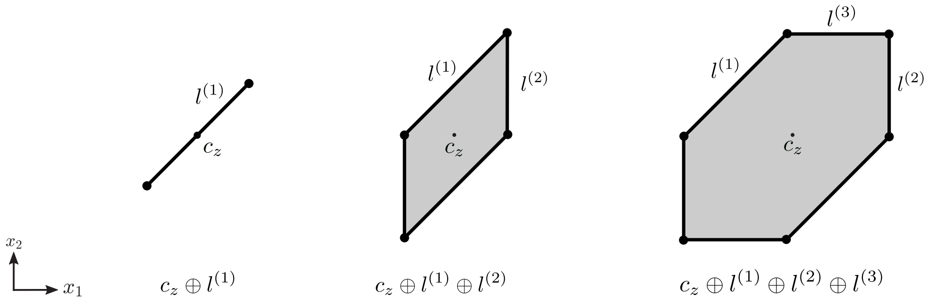


Figure 2.10: Step-by-step construction of zonotopes according to (2.28)

2.3.5 Multi-dimensional Intervals

Finally, we frequently employ multi-dimensional intervals throughout this thesis; in particular, to strictly bound the so-called set of Lagrangian remainders as shown later in Sec. 2.4.3. In the literature a multi-dimensional interval is also referred to as a hyper-rectangle, an n-dimensional orthotope, or simply an interval vector. Notice that some basics about interval arithmetics are addressed shortly in this section. Formally, a multi-dimensional interval is defined according to

Definition 2.5. Multi-dimensional interval: Given two vectors of real numbers denoted by $\overline{\gamma}$, and $\underline{\gamma}$, such that $\forall j \in \{1, \ldots, n\} : \overline{\gamma}_j \geq \underline{\gamma}_j$, a multi-dimensional interval is expressed via

$$\begin{aligned} \boldsymbol{\mathcal{I}} = \left[\underline{\gamma}, \overline{\gamma}\right] &:= \left\{ x \in \mathbb{R}^n \, : \, \underline{\gamma}_j \leq x_j \leq \overline{\gamma}_j \right\}, \\ &= \left[\underline{\gamma}_1, \overline{\gamma}_1\right] \times \left[\underline{\gamma}_2, \overline{\gamma}_2\right] \times \cdots \times \left[\underline{\gamma}_n, \overline{\gamma}_n\right]. \qquad \square \end{aligned}$$

The operator × corresponds to the Cartesian product.

2.3.6 Set-based Operations

The most important factor behind the choice of the set representation for reachability analysis is that the most reoccurring set-based operations can be computed efficiently with respect to the system dimension. Furthermore, the corresponding set-based operations have to be computed exactly or as tightly as possible.

Consider an arbitrary matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ and two convex sets $\boldsymbol{\mathcal{S}}_1$, $\boldsymbol{\mathcal{S}}_2 \subset \mathbb{R}^n$, loosely speaking, the main operations for reachability computations are [56]:

$$\boldsymbol{\mathcal{S}}_1 \oplus \boldsymbol{\mathcal{S}}_2 := \left\{ \boldsymbol{s}_1 + \boldsymbol{s}_2 \,:\, \boldsymbol{s}_1 \in \boldsymbol{\mathcal{S}}_1,\, \boldsymbol{s}_2 \in \boldsymbol{\mathcal{S}}_2 \right\}, \tag{2.29}$$

$$\boldsymbol{M} \cdot \boldsymbol{\mathcal{S}}_1 := \left\{ \boldsymbol{M}\boldsymbol{s} \,:\, \boldsymbol{s} \in \boldsymbol{\mathcal{S}}_1 \right\}, \tag{2.30}$$

$$\mathbf{conv}\,(\boldsymbol{\mathcal{S}}_1, \boldsymbol{\mathcal{S}}_2) := \left\{ \boldsymbol{\alpha}\boldsymbol{s}_1 + (1 - \boldsymbol{\alpha})\boldsymbol{s}_2 \,:\, \boldsymbol{s}_1 \in \boldsymbol{\mathcal{S}}_1,\, \boldsymbol{s}_2 \in \boldsymbol{\mathcal{S}}_2,\, \alpha_i \in [0, 1] \right\}. \tag{2.31}$$

with (2.29)-(2.31) corresponding to the Minkowski Sum, linear mapping and convex hull enclosure, respectively.

Table 2.3 provides an illustrative comparison with regards to the complexity and boundness of the aforementioned operations. It is obvious that polytopes are superior in terms of accuracy since all set-based operations can be computed exactly without requiring any approximations; however, they are only practical for low-dimensional systems as their number of vertices grows exponentially with the system dimensions. On the other hand, multi-dimensional intervals are very efficient to handle high-dimensional system, but the obtained results are rather conservative, often leading to unacceptable over-approximations of the exact solution.

Alternatively, ellipsoids and zonotpes are much more efficient, however, they are not as accurate as polytopes. Hence, they offer a comprise between accuracy and efficiency. In particular, we choose zonotopes over ellipsoids for set representation in this thesis, since the Minkowski sum can be computed exactly, in contrast to ellipsoids which requires an over-approximation in order to represent the resulting set, see [83].

Table 2.3: Comparison of basic operations necessary for reachability computations using typical set representation.

	Polytopes	Zonotopes	Ellipsoids	Intervals
Minkowski Sum	closed	closed	not closed	closed
Linear Mapping	closed	closed	closed	not closed
Convex Hull	closed	not closed	not closed	not closed

2.3.7 Basic Operations on Zonotopes

Now we present the operations performed on zonotopes to compute reachable sets. These operations are the Minkowski sum, linear transformation, convex and interval enclosure, and Cartesian product. First we introduce two zonotopic sets of equal dimensions $\boldsymbol{\mathcal{Z}}_1 := [\boldsymbol{c}_{1,z}, \boldsymbol{G}_{1,z}]_{\boldsymbol{\mathcal{Z}}}$, and $\boldsymbol{\mathcal{Z}}_2 := [\boldsymbol{c}_{2,z}, \boldsymbol{G}_{2,z}]_{\boldsymbol{\mathcal{Z}}}$, and reuse the arbitrary matrix $\boldsymbol{M}$. The Minkowski addition of two zonotopes is defined as [56]

$$\boldsymbol{\mathcal{Z}}_3 = \boldsymbol{\mathcal{Z}}_1 \oplus \boldsymbol{\mathcal{Z}}_2 := \Bigg[\underbrace{\Big(\boldsymbol{c}_{z,1} + \boldsymbol{c}_{z,2}\Big)}_{=:\,\boldsymbol{c}_{z,3}}, \underbrace{\Big(\boldsymbol{g}_{z,1}^{(1)}, \dots, \boldsymbol{g}_{z,1}^{(p_1)}, \boldsymbol{g}_{z,2}^{(1)}, \dots, \boldsymbol{g}_{z,2}^{(p_2)}\Big)}_{=:\,\boldsymbol{G}_{z,3}} \Bigg]_{\boldsymbol{\mathcal{Z}}}, \tag{2.32}$$

and the linear transformation of the resulting zonotope $[\boldsymbol{c}_{z,3}, \boldsymbol{G}_{z,3}]_{\boldsymbol{\mathcal{Z}}}$ by the matrix $\boldsymbol{M}$ is

$$\boldsymbol{M} \cdot \boldsymbol{\mathcal{Z}}_3 := \Big[(\boldsymbol{M} \cdot \boldsymbol{c}_{z,3}), (\boldsymbol{M} \cdot \boldsymbol{G}_{z,3}) \Big]_{\boldsymbol{\mathcal{Z}}}. \tag{2.33}$$

Notice that both operations are closed since the resulting set is a zonotope as well. This property is advantageous for zonoptes since the Minkowski sum and linear transformation are used extensively in reachability computations, see Table 2.3. The following two operations are over-approximative enclosures; first we consider the interval enclosure of zonotopes [6]

$$\boldsymbol{\mathcal{Z}}_3 \subset \textbf{interval}(\boldsymbol{\mathcal{Z}}_3) := \big[\underline{\boldsymbol{\Delta}}, \overline{\boldsymbol{\Delta}}\big], \quad \text{with:} \begin{cases} \underline{\Delta}_j := c_{z,3,j} - \sum_{i=1}^{p_3} |\, g_{z,3,j}^{(i)} \,|, \\ \overline{\Delta}_j := c_{z,3,j} + \sum_{i=1}^{p_3} |\, g_{z,3,j}^{(i)} \,|, \end{cases} \tag{2.34}$$

where the subscript j corresponds to the j-th dimension and the operator $|\cdot|$ returns the absolute value. Second, the convex hull operator required to enclose two zonotopes by another zonotope is [56]

$$\boldsymbol{\mathcal{Z}}_4 \subseteq \textbf{conv}(\boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_3) := \Big[\Big(\boldsymbol{c}_{1,z} + \boldsymbol{c}_{3,z}\Big), \frac{1}{2}\Big(\boldsymbol{G}_{1,z} + \boldsymbol{G}_{3,z}, \boldsymbol{G}_{1,z} - \boldsymbol{G}_{3,z}\Big) \Big]_{\boldsymbol{\mathcal{Z}}}, \tag{2.35}$$

Note that this operation is computed in an over-approximative manner, since the convex enclosure of two zonotopes is generally not a zonotope [56]. Finally, the Cartesian product of two zonotopes is

$$\boldsymbol{\mathcal{Z}}_1 \times \boldsymbol{\mathcal{Z}}_2 := \left[\begin{pmatrix} \boldsymbol{c}_{1,z} \\ \boldsymbol{c}_{2,z} \end{pmatrix}, \begin{pmatrix} \boldsymbol{G}_{1,z} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_{2,z} \end{pmatrix} \right]_{\boldsymbol{\mathcal{Z}}}, \tag{2.36}$$

where $\boldsymbol{0}$ is a matrix of zeros with proper dimension.

Example 2.1. Consider the following zonotopes $\boldsymbol{\mathcal{Z}}_1$, $\boldsymbol{\mathcal{Z}}_2$, and the matrix $\boldsymbol{M}$ such that:

$$\boldsymbol{\mathcal{Z}}_1 = \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \end{pmatrix} \right], \quad \boldsymbol{\mathcal{Z}}_2 = \left[\begin{pmatrix} 8 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \right], \quad \boldsymbol{M} = \begin{pmatrix} 0.5 & 0.2 \\ 0.1 & 0.3 \end{pmatrix}.$$

The operations described in (2.32)-(2.35) are illustrated in Fig. 2.11 on the aforementioned zonotopes.

2.3.8 Interval Arithmetics

Interval arithmetics refers to an analysis tool initially developed for bounding rounding errors associated with mathematical computation. Simply put, intervals techniques can be thought as a generalization of standard operations when the considered variables are not know explicitly but can be described as unknown-but-bounded; that is, their values varies within a specified interval. To gain insight about interval analysis, the reader is referred to [68, 102].

In general interval arithmetics operations of two intervals denoted by $[a, b]$ and $[c, d]$ are defined as follows:

$$[a, b] \circ [c, d] = \{x \circ y : x \in [a, b],\ y \in [c, d]\}\,,$$

with $\circ \in \{\oplus, \ominus, \odot, \oslash\}$ representing the basic arithmetic operations, corresponding to addition, substraction, multiplication and division, respectively. Provided that $0 \notin [c, d]$, the aforementioned operations can be computed according to:

$$\begin{aligned}
&[a, b] \oplus [c, d] = [a + c, b + d],\\
&[a, b] \ominus [c, d] = [a - d, b - c],\\
&[a, b] \odot [c, d] = [\min(a \cdot c, a \cdot d, b \cdot c, b \cdot d), \max(a \cdot c, a \cdot d, b \cdot c, b \cdot d)]\\
&[a, b] \oslash [c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)]
\end{aligned} \tag{2.37}$$

with $\max(\,\cdot\,)$ and $\min(\,\cdot\,)$ returning the maximum and minimum values of a given set.

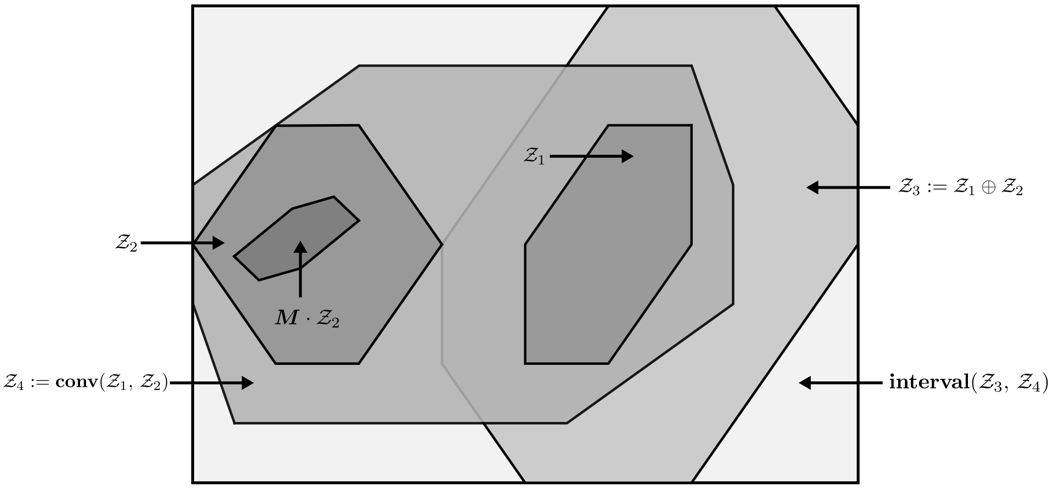

$\mathcal{Z}_1$
$\mathcal{Z}_3 := \mathcal{Z}_1 \oplus \mathcal{Z}_2$
$\mathcal{Z}_2$
$\boldsymbol{M} \cdot \mathcal{Z}_2$
$\mathcal{Z}_4 := \mathbf{conv}(\mathcal{Z}_1,\, \mathcal{Z}_2)$
$\mathbf{interval}(\mathcal{Z}_3,\, \mathcal{Z}_4)$

Figure 2.11: Illustration of the basic operations performed on zonotopes with $\boldsymbol{\mathcal{Z}}_1$ and $\boldsymbol{\mathcal{Z}}_2$ being two zonotopic sets of equal dimensions. The set $\boldsymbol{\mathcal{Z}}_3$ results from the Minkowski sum, and $\boldsymbol{\mathcal{Z}}_4$ from the convex enclosure as in (2.32) and (2.35), respectively. The black solid box represents the interval enclosure of the resulting zonotopes.

Interval arithmetics can be also applied to interval matrices. By an interval matrix we refer to a matrix denoted by $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n \times n}$, whose elements are interval numbers, that is $\mathcal{A}_{ij} := [\underline{a}_{ij}, \overline{a}_{ij}]$, $\overline{a}_{ij} > \underline{a}_{ij} \in \mathbb{R}$.

Remark 2.4. Interval arithmetics are employed in the reachability algorithm presented shortly in the following section in order to bound to the so-called Lagrangian remainders of the Taylor series expansion. Furthermore, interval arithmetics is also used in Ch. 5 in order to obtain the set of the time-varying parameters required to synthesis a set of LPV controllers with formal guarantees, see Sec. 5.4.1.

2.4 Computation of Reachable Sets

In this section we introduce the reachability algorithm, employed throughout this thesis, to compute reachable sets of differential and algebraic variables of the DAE system (2.25) in an over-approximative manner. The presented approach is based on known techniques for evaluating reachable sets of linear differential inclusions (LDIs) [23, p. 51]. LDIs are considered as a natural extension of the concept of ODE and often arise in many problems concerned with control theory, see for example Ch. 5. In fact, LDIs can be interpreted as describing a family of linear time-varying systems. Shorty after, we illustrate how LDIs can be employed to compute reachable sets of DAE systems.

2.4.1 Differential-Algebraic Model

Prior to discussion of the algorithmic procedure, we transform the fully-implicit DAE system (2.25) into the standard semi-explicit index-1 DAE formalization that applies to almost any power system [14, 81, 96]

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), && \text{Set of explicit ODEs} \\ \boldsymbol{0} &= \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), && \text{Set of algebraic equations} \end{aligned} \tag{2.38}$$

with $\boldsymbol{f} : \mathbb{R}^{n_x+n_y+n_u} \mapsto \mathbb{R}^{n_x}$ and $\boldsymbol{g} : \mathbb{R}^{n_x+n_y+n_u} \mapsto \mathbb{R}^{n_y}$. Here, the vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$ includes the differential states variables, the vector $\boldsymbol{y} \in \mathbb{R}^{n_y}$ specifies the algebraic variables, and the vector $\boldsymbol{u} \in \mathbb{R}^{n_u}$ considers the controllable inputs and/or input uncertainties. Note that any fully-implicit DAE can always be transformed into the semi-explicit form, furthermore, the index-1 property holds if and only if the Jacobian matrix of the algebraic equations is invertible (non-singular), that is

$$\det\left(\frac{\partial \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t))}{\partial \boldsymbol{y}}\right) \neq 0,\ t > 0. \tag{2.39}$$

2.4.2 Local Linearization

As stated earlier, our approach is based on abstracting the set of semi-explicit DAEs (2.38) to LDIs for consecutive time intervals; that is to perform a local linearization within the interval $\tau_k := [t_k, t_{k+1}]$, where $k \in \mathbb{N}$ is the time step. We only use constant-size time intervals $t_k = k \cdot t_r$, with $t_r \in \mathbb{R}^+$ being the time increment. An extension, however, covering variable-size time steps can found in [54].

For a concise notation, we introduce the vector $\boldsymbol{z} := (\boldsymbol{x}^T\,\boldsymbol{y}^T\,\boldsymbol{u}^T)^T \in \mathbb{R}^{n_z}$, and the linearization point $\boldsymbol{z}_k := (\boldsymbol{x}_k^T\,\boldsymbol{y}_k^T\,\boldsymbol{u}_k^T)^T$. Notice that the subscript k denotes the time step; that is the linearization point is updated at each time interval.

Remark 2.5. Since the linearization of (2.38) causes additional errors, these errors are determined in an over-approximative manner and considered as additional uncertain inputs. By recomputing the linearization for each τ_k, the over-approximation of the exact reachable set remains small and accurate results are guaranteed.

The local linearization of the DAE system (2.38), at each time interval τ_k is performed by an infinite Taylor series expressed as follows:

$$\begin{aligned}\dot{x}_i(t) &= f_i(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)),\\ &= \underbrace{f_i(\boldsymbol{z}_k) + \sum_{m=1}^{n_z} \frac{\partial f_i(\boldsymbol{z})}{\partial z_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{z}_m}_{\text{1-st order Taylor expansion}} + \underbrace{\frac{1}{2!}\sum_{l=1}^{n_z}\sum_{m=1}^{n_z} \frac{\partial^2 f_i(\boldsymbol{z})}{\partial z_l \partial z_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{z}_l\,\tilde{z}_m + \ldots}_{\text{Infinite number of terms of } f_i(z)}, \qquad (2.40)\end{aligned}$$

$$\begin{aligned}0 &= g_i(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)),\\ &= \underbrace{g_i(\boldsymbol{z}_k) + \sum_{m=1}^{n_z} \frac{\partial g_i(\boldsymbol{z})}{\partial z_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{z}_m}_{\text{1-st order Taylor expansion}} + \underbrace{\frac{1}{2!}\sum_{l=1}^{n_z}\sum_{m=1}^{n_z} \frac{\partial^2 g_i(\boldsymbol{z})}{\partial z_l \partial z_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{z}_l\,\tilde{z}_m + \ldots}_{\text{Infinite number of terms of } g_i(z)}, \qquad (2.41)\end{aligned}$$

with $\tilde{\boldsymbol{z}}(t) := \boldsymbol{z}(t) - \boldsymbol{z}_k$ and the subscript i denoting the i-th coordinate. Here the infinite number of terms of the Taylor expansion can be over-approximated using the following proposition, see [22]:

Proposition 2.1. Lagrangian remainder: For the time interval τ_k, suppose the vector $\boldsymbol{z}(t)$ varies within an arbitrary set $\boldsymbol{\mathcal{R}}^z(\tau_k)$, then the Taylor series of the DAE system (2.38) at the linearization point $\boldsymbol{z}_k$ is expressed via a 1-st order Taylor expansion in addition to the remainder terms bounded within the so-called Lagrangian remainder

$$\boldsymbol{\mathcal{L}}^x(\tau_k) := \left\{ \boldsymbol{L}^x : L_j^x = \frac{1}{2!}\sum_{l=1}^{n_z}\sum_{m=1}^{n_z} \frac{\partial^2 f_j(\boldsymbol{z})}{\partial z_l \partial z_m}\bigg|_{\boldsymbol{z}=\boldsymbol{\mu}} \tilde{z}_l\,\tilde{z}_m,\ \boldsymbol{\mu} \in \boldsymbol{\zeta}(\tau_k) \right\},$$

$$\boldsymbol{\mathcal{L}}^y(\tau_k) := \left\{ \boldsymbol{L}^y : L_j^y = \frac{1}{2!}\sum_{l=1}^{n_z}\sum_{m=1}^{n_z} \frac{\partial^2 g_j(\boldsymbol{z})}{\partial z_l \partial z_m}\bigg|_{\boldsymbol{z}=\boldsymbol{\mu}} \tilde{z}_l\,\tilde{z}_m,\ \boldsymbol{\mu} \in \boldsymbol{\zeta}(\tau_k) \right\},$$

with $\boldsymbol{\mathcal{L}}^x$ and $\boldsymbol{\mathcal{L}}^y$ corresponding to the remainders of differential and algebraic variables, respectively, and the variable $\boldsymbol{\mu}$ takes any value from the set $\boldsymbol{\zeta}(\tau_k)$, which is expressed according to

$$\boldsymbol{\zeta}(\tau_k) := \left\{ \boldsymbol{\alpha}\boldsymbol{z} + (1-\boldsymbol{\alpha})\boldsymbol{z}_k \ :\ 0 \le \alpha_i \le 1,\ \boldsymbol{z} \in \boldsymbol{\mathcal{R}}^z(\tau_k) \right\}. \qquad \square$$

2.4.2.1 Abstraction to Linear Differential Inclusions

Based on Prop. 2.1, we can rewrite the Taylor expansions (2.40) and (2.41) in state-space form

$$\begin{aligned}
\forall t \in \tau_k: \quad \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), & (2.42)\\
&\in \boldsymbol{f}(\boldsymbol{z}_k) + \underbrace{\sum_{m=1}^{n_x} \frac{\partial \boldsymbol{f}(\boldsymbol{z})}{\partial x_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{x}_m}_{=:\,\boldsymbol{A}_k\tilde{\boldsymbol{x}}} + \underbrace{\sum_{m=1}^{n_u} \frac{\partial \boldsymbol{f}(\boldsymbol{z})}{\partial u_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{u}_m}_{=:\,\boldsymbol{B}_k\tilde{\boldsymbol{u}}} + \underbrace{\sum_{m=1}^{n_y} \frac{\partial \boldsymbol{f}(\boldsymbol{z})}{\partial y_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{y}_m}_{=:\,\boldsymbol{C}_k\tilde{\boldsymbol{y}}} \oplus \boldsymbol{\mathcal{L}}^x(\tau_k),\\
\boldsymbol{0} &= \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), & (2.43)\\
&\in \boldsymbol{g}(\boldsymbol{z}_k) + \underbrace{\sum_{m=1}^{n_x} \frac{\partial \boldsymbol{g}(\boldsymbol{z})}{\partial x_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{x}_m}_{=:\,\boldsymbol{D}_k\tilde{\boldsymbol{x}}} + \underbrace{\sum_{m=1}^{n_u} \frac{\partial \boldsymbol{g}(\boldsymbol{z})}{\partial u_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{u}_m}_{=:\,\boldsymbol{E}_k\tilde{\boldsymbol{u}}} + \underbrace{\sum_{m=1}^{n_y} \frac{\partial \boldsymbol{g}(\boldsymbol{z})}{\partial y_m}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} \tilde{y}_m}_{=:\,\boldsymbol{F}_k\tilde{\boldsymbol{y}}} \oplus \boldsymbol{\mathcal{L}}^y(\tau_k),
\end{aligned}$$

with: $\tilde{\boldsymbol{x}}(t) := \boldsymbol{x}(t) - \boldsymbol{x}_k$, $\tilde{\boldsymbol{u}}(t) := \boldsymbol{u}(t) - \boldsymbol{u}_k$, $\tilde{\boldsymbol{y}}(t) := \boldsymbol{y}(t) - \boldsymbol{y}_k$.

Here the matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$, $\boldsymbol{C}_k$ denote the system matrices of the linearized differential equations, and likewise, $\boldsymbol{D}_k$, $\boldsymbol{E}_k$, $\boldsymbol{F}_k$ specify the matrices of the linearized algebraic constraints with proper dimension. Due to the index-1 property of the DAE system, it is guaranteed that the matrix $\boldsymbol{F}_k$ is non-singular, thus

$$\tilde{\boldsymbol{y}} \in -\boldsymbol{F}_k^{-1} \cdot \left(\boldsymbol{g}(\boldsymbol{z}_k) + \boldsymbol{D}_k\tilde{\boldsymbol{x}} + \boldsymbol{E}_k\tilde{\boldsymbol{u}} \oplus \boldsymbol{\mathcal{L}}^y(\tau_k)\right), \tag{2.44}$$

then inserting (2.44) into (2.42) yields

$$\begin{aligned}
\dot{\boldsymbol{x}}(t) &\in \boldsymbol{f}(\boldsymbol{z}_k) + \boldsymbol{A}_k\tilde{\boldsymbol{x}} + \boldsymbol{B}_k\tilde{\boldsymbol{u}} - \boldsymbol{C}_k\left[\boldsymbol{F}_k^{-1}\left(\boldsymbol{g}(\boldsymbol{z}_k) + \boldsymbol{D}_k\tilde{\boldsymbol{x}} + \boldsymbol{E}_k\tilde{\boldsymbol{u}} \oplus \boldsymbol{\mathcal{L}}^y(\tau_k)\right)\right] \oplus \boldsymbol{\mathcal{L}}^x(\tau_k),\\
&\in \underbrace{\boldsymbol{f}(\boldsymbol{z}_k) - \boldsymbol{C}_k\boldsymbol{F}_k^{-1}\boldsymbol{g}(\boldsymbol{z}_k)}_{=:\boldsymbol{w}(\boldsymbol{z}_k)} + \underbrace{\left(\boldsymbol{A}_k - \boldsymbol{C}_k\boldsymbol{F}_k^{-1}\boldsymbol{D}_k\right)}_{=:\tilde{\boldsymbol{A}}_k}\tilde{\boldsymbol{x}},\\
&\qquad + \underbrace{\left(\boldsymbol{B}_k - \boldsymbol{C}_k\boldsymbol{F}_k^{-1}\boldsymbol{E}_k\right)}_{=:\tilde{\boldsymbol{B}}_k}\tilde{\boldsymbol{u}} \oplus \underbrace{\left(\boldsymbol{\mathcal{L}}^x(\tau_k) \oplus (-\boldsymbol{C}_k\boldsymbol{F}_k^{-1}) \cdot \boldsymbol{\mathcal{L}}^y(\tau_k)\right)}_{=:\boldsymbol{\mathcal{L}}(\tau_k)},
\end{aligned} \tag{2.45}$$

where $\boldsymbol{\mathcal{L}}(\tau_k)$ is the set of linearization errors. One can further simplify (2.45) by merging the set of uncertain inputs and the set of linearization errors together, that is

$$\tilde{\boldsymbol{\mathcal{U}}}(\tau_k) := \boldsymbol{w}(\boldsymbol{z}_k) \oplus \tilde{\boldsymbol{B}}_k\left(\boldsymbol{\mathcal{U}} \oplus (-\boldsymbol{u}_k)\right) \oplus \boldsymbol{\mathcal{L}}(\tau_k). \tag{2.46}$$

This allows the abstraction of the differential equations of the DAE system (2.38) by the following LDI

$$\forall t \in \tau_k: \quad \dot{\boldsymbol{x}}(t) \in \tilde{\boldsymbol{A}}_k\tilde{\boldsymbol{x}}(t) \oplus \tilde{\boldsymbol{\mathcal{U}}}(\tau_k). \tag{2.47}$$

Remark 2.6. Note that the DAE system (2.38) is abstracted at each time interval τ_k. Additionally, the inclusion (2.47) encloses all possible trajectories of the nonlinear DAE system, as we consider the linearization errors via the set of Lagrangian remainder included within the set of uncertain inputs $\tilde{\boldsymbol{\mathcal{U}}}$.

2.4.2.2 Reachable Set Computation of Linear Inclusions

With the abstraction of the DAE system by (2.47), we can now employ known techniques to computing reachable sets for linear systems. Analogously to the solution of linear-time invariant (LTI) systems described by state equations, the solution of (2.47) for the next time instant t_{k+1}, based on the superposition principle, is well-known to be [56]

$$\boldsymbol{\mathcal{R}}^x(t_{k+1}) = \underbrace{e^{\tilde{\boldsymbol{A}}_k t_r} \boldsymbol{\mathcal{R}}^x(t_k)}_{=:\boldsymbol{\mathcal{R}}^x_h(t_{k+1})} \oplus \underbrace{\int_0^{t_r} e^{\tilde{\boldsymbol{A}}_k (t_r - t)} \tilde{\boldsymbol{\mathcal{U}}} dt}_{=:\boldsymbol{\mathcal{R}}^x_p([0,\, t_r])} . \tag{2.48}$$

Here the sets $\boldsymbol{\mathcal{R}}^x(t_{k+1})$ and $\boldsymbol{\mathcal{R}}^x(t_k)$ specify the reachable set at different time-steps, whereas $\boldsymbol{\mathcal{R}}_h(t_{k+1})$ and $\boldsymbol{\mathcal{R}}_p([0, t_r])$ denote the homogenous and particular (inhomogeneous) solutions, respectively, and $e^{\tilde{A} t_r}$ is the matrix exponential. Note that the subscript x corresponds to differential state variables.

In this thesis we describe $e^{\tilde{A} t_r}$ via a finite Taylor series, up to an order σ, in addition to the remainder that considers all higher-order terms [101]

$$e^{\tilde{\boldsymbol{A}}_k t_r} = \left(\boldsymbol{I} + \frac{\tilde{\boldsymbol{A}}_k t_r}{1!} + \frac{(\tilde{\boldsymbol{A}}_k t_r)^2}{2!} + \dots + \frac{(\tilde{\boldsymbol{A}}_k t_r)^\sigma}{\sigma!} \right) + \left(\frac{(\tilde{\boldsymbol{A}}_k t_r)^{\sigma+1}}{(\sigma+1)!} + \dots + \frac{(\tilde{\boldsymbol{A}}_k t_r)^{\sigma+n}}{(\sigma+n)!} + \dots \right), \tag{2.49}$$

$$\in \sum_{i=0}^{\sigma} \frac{(\tilde{\boldsymbol{A}}_k t_r)^i}{i!} \oplus \underbrace{[-\boldsymbol{1},\, \boldsymbol{1}] \frac{(|\tilde{\boldsymbol{A}}_k|^{\sigma+1} t_r^{\sigma+1}}{(\sigma+1)!} \frac{1}{1-\epsilon}}_{=:\boldsymbol{\mathcal{M}}([0,t_r])}, \quad \text{s.t.} \quad \epsilon = \frac{|\tilde{\boldsymbol{A}}_k| t_r}{\sigma+2} \overset{!}{<} 1, \tag{2.50}$$

where the set $\boldsymbol{\mathcal{M}}([0, t_r])$ corresponds to the set over-approximating the remainder terms of the Taylor expansion of the matrix exponential.

Note that (2.48) is just the solution of the LDI at the time t_{k+1}. However, we are mainly interested in the solution within the overall time interval τ_k; thus we must enclose the reachable sets at the time instants t_k and t_{k+1} in the least conservative way. The over-approximation of the reachable set within τ_k is obtained as suggested in [38]

$$\tau_k \in [t_{k+1},\, t_k] : \quad \boldsymbol{\mathcal{R}}^x_h(\tau_k) \subseteq \textbf{conv}\left(\boldsymbol{\mathcal{R}}^x(t_k), e^{\tilde{\boldsymbol{A}}_k t_r} \boldsymbol{\mathcal{R}}^x(t_k) \right) \oplus \boldsymbol{\mathcal{C}}([0, t_r]) \boldsymbol{\mathcal{R}}^x(t_k) \tag{2.51}$$

$$\text{with :} \quad \boldsymbol{\mathcal{C}}([0, t_r]) = \sum_{i=1}^{\sigma} \left[\left(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}} \right) t_r^i \right] \frac{\tilde{\boldsymbol{A}}_k^i}{i!} \oplus \boldsymbol{\mathcal{M}}([0, t_r]). \tag{2.52}$$

with the operator **conv**(·) returning the convex hull enclosure of two sets as in (2.29). Here the additional term $\boldsymbol{\mathcal{C}}([0, t_r])$ is an uncertainty factor handling correction of the enclosure set accounting for bloating of the reachable set. The computation of $\boldsymbol{\mathcal{C}}([0, t_r])$ as in (2.52) is based on [6, Prop. 3.1] which employs interval arithmetics.

With the computation of the homogenous solution, it remains to express the effect of the uncertain input; that is to consider the inhomogeneous solution of the LDI (2.48). First we consider the simple case, in which we assume that the system matrix $\tilde{\boldsymbol{A}}_k$ is invertible and the trajectories of the uncertain inputs $\tilde{\boldsymbol{\mathcal{U}}}$ remains constant over the interval τ_k; thus the set enclosing the particular solution may be expressed by

$$\begin{aligned}\boldsymbol{\mathcal{R}}_p([0,t_r]) = \tilde{\boldsymbol{\mathcal{U}}}(\tau_k)\int_0^{t_r} e^{\tilde{\boldsymbol{A}}_k(t_r-t)}dt &= \tilde{\boldsymbol{\mathcal{U}}}(\tau_k)\cdot\tilde{\boldsymbol{A}}_k^{-1}\left(e^{\tilde{\boldsymbol{A}}_k(t_r-t)}\right)\Big|_0^{t_r}\\ &= \tilde{\boldsymbol{\mathcal{U}}}(\tau_k)\cdot\tilde{\boldsymbol{A}}_k^{-1}(e^{\tilde{\boldsymbol{A}}_k t_r}-\boldsymbol{I}).\end{aligned}$$

However, the aforementioned assumptions do not hold in many practical situations. In the event that either the matrix $\tilde{\boldsymbol{A}}_k$ is singular or the set $\tilde{\boldsymbol{\mathcal{U}}}(\tau_k)$ varies over time, one has to over-approximate the particular solution according to the following proposition, see [6, 85]:

Proposition 2.2. Over-approximation of the LDI particular solution: Given the LDI described as in (2.47), then the over-approximation of its particular solution is given by

$$\boldsymbol{\mathcal{R}}_p([0,t_r]) \subset \left(\sum_{i=0}^{\sigma}\frac{|\tilde{\boldsymbol{A}}_k^i|\,{t_r}^{i+1}}{(i+1)!}\cdot\tilde{\boldsymbol{\mathcal{U}}}\right)\oplus\left(\boldsymbol{\mathcal{M}}\cdot t_r\cdot\tilde{\boldsymbol{\mathcal{U}}}\right). \tag{2.53}$$

with $\boldsymbol{\mathcal{M}}$ and σ as the variables appearing in (2.49) to over-approximate the matrix exponential. □

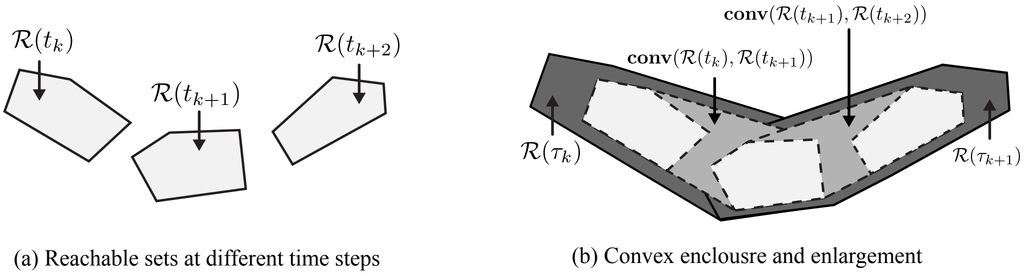
Remark 2.7. Note that in Prop. 2.2, it is assumed that the set $\tilde{\boldsymbol{\mathcal{U}}}(\tau_k)$ contains the origin, which is not always the case and certain correction measures have to be applied to consider this case. Simply put, one needs to split the effect of $\tilde{\boldsymbol{\mathcal{U}}}(\tau_k)$ into a constant $\boldsymbol{u}_c$ corresponding to its center and another set $\tilde{\boldsymbol{\mathcal{U}}}_\Delta$ specifying the deviation from the center $\boldsymbol{u}_c$.

2.4.2.3 Algorithmic Realization

Combining all previous results, the reachable set of the LDI for consecutive time intervals can be computed, as illustrated in Fig. 2.12, according to the following steps:

(a) Compute the homogeneous reachable sets of the LDI at different time instants based on previously computed sets.

(b) Obtain and bloat the convex hull enclosure of the homogenous reachable sets. The enlargement of the convex hull is necessary for two reasons: first to over-approximate the reachable set for each time interval, and second to account for the effect of the inhomogeneous solution of the LDI.


$\mathcal{R}(t_k)$
$\mathcal{R}(t_{k+1})$
$\mathcal{R}(t_{k+2})$
$\mathbf{conv}(\mathcal{R}(t_{k+1}),\mathcal{R}(t_{k+2}))$
$\mathbf{conv}(\mathcal{R}(t_k),\mathcal{R}(t_{k+1}))$
$\mathcal{R}(\tau_k)$
$\mathcal{R}(\tau_{k+1})$

(a) Reachable sets at different time steps

(b) Convex enclousre and enlargement

Figure 2.12: Stepwise computation of reachable sets of linear differential inclusions expressed as in (2.47).

The aforementioned steps can be summarized as in Alg. 1. The algorithm computes the reachable set of a family of LTI systems enclosed by the LDI described as in (2.47), for the next point in time using previously computed reachable sets. First, through lines 1-3, the algorithm computes the remainder of the matrix exponential $\boldsymbol{\mathcal{M}}([0,t_r])$ based on the number of Taylor terms σ which is a user-defined variable. Using the remainder terms, one can over-approximate the matrix exponential $e^{\tilde{\boldsymbol{A}}t_r}$ to obtain the homogenous solution of the LDI, in addition to the correction term $\boldsymbol{\mathcal{C}}([0,t_r])$ accounting for over-approximation of the reachable set within the interval τ_k, as illustrated in lines 4 and 5. Finally, to fully describe the reachable set for the next time instant and current time interval, the algorithm over-approximates the particular solution of the LDI according to Prop. 2.2, as specified in lines 6-8.

Remark 2.8. In order to obtain the reachable for the complete time-horizon $t \in [0,\, t_f]$, with t_f as the terminal time specified by the user, one has to continuously call Alg. 1 to obtain the reachable sets at different time instants until $t > t_f$. Afterwards the reachable set is assembled via the union of the computed sets, that is $\boldsymbol{\mathcal{R}}([0,t_f]) := \bigcup_{k=1}^{tf/t_r} \boldsymbol{\mathcal{R}}(\tau_k)$.

Algorithm 1 ReachNextStep

```
Require: Uncertain inputs $\tilde{\boldsymbol{\mathcal{U}}}(\tau_k)$, the time increment $t_r$, number of Taylor terms $\sigma$ of the matrix exponential, time step $k$, and the system matrix $\tilde{\boldsymbol{A}}_k$
Ensure: $\boldsymbol{\mathcal{R}}^x(t_{k+1})$, $\boldsymbol{\mathcal{R}}^x(\tau_k)$
1: $\boldsymbol{\mathcal{M}}([0,t_r]) \overset{(2.50)}{=} [\mathbf{-1},\, \mathbf{1}]\frac{|\tilde{\boldsymbol{A}}_k|^{\sigma+1} t_r^{\sigma+1}}{(\sigma+1)!}\frac{1}{1-\epsilon}$   ▷ Remainder of matrix exponential
2: $e^{\tilde{\boldsymbol{A}}_k t_r} \overset{(2.49)}{\subset} \sum_{i=0}^{\sigma} \frac{(\tilde{\boldsymbol{A}}_k t_r)^i}{i!} \oplus \boldsymbol{\mathcal{M}}([0,t_r])$   ▷ Over-approximation of matrix exponential
3: $\boldsymbol{\mathcal{C}}([0,t_r]) \overset{(2.52)}{=} \sum_{i=1}^{\sigma} \left[\left(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}}\right) t_r^i\right] \frac{\tilde{\boldsymbol{A}}_k^i}{i!} \oplus \boldsymbol{\mathcal{M}}([0,t_r])$   ▷ Correction interval matrix
4: $\boldsymbol{\mathcal{R}}_h(t_{k+1}) \overset{(2.48)}{=} e^{\tilde{\boldsymbol{A}}_k t_r}\boldsymbol{\mathcal{R}}^x(t_k)$   ▷ LDI homogeneous solution
5: $\boldsymbol{\mathcal{R}}_h^x(\tau_k) \overset{(2.51)}{\subset} \mathbf{conv}(\boldsymbol{\mathcal{R}}^x(t_k), e^{\tilde{\boldsymbol{A}}_k r}\boldsymbol{\mathcal{R}}^x(t_k)) \oplus \boldsymbol{\mathcal{C}}([0,t_r])\boldsymbol{\mathcal{R}}^x(t_k)$   ▷ LDI homogenous solution within $\tau_k$
6: $\boldsymbol{\mathcal{R}}_p([0,t_r]) \overset{(2.53)}{\subset} \left(\sum_{i=0}^{\sigma} \frac{|\tilde{\boldsymbol{A}}_k^i|\, {t_r}^{i+1}}{(i+1)!} \cdot \tilde{\boldsymbol{\mathcal{U}}}\right) \oplus \left(\boldsymbol{\mathcal{M}} \cdot t_r \cdot \tilde{\boldsymbol{\mathcal{U}}}\right)$   ▷ LDI inhomogeneous solution
7: $\boldsymbol{\mathcal{R}}^x(t_{k+1}) = \boldsymbol{\mathcal{R}}_h(t_{k+1}) \oplus \boldsymbol{\mathcal{R}}_p([0,t_r])$   ▷ LDI solution for next point in time
8: $\boldsymbol{\mathcal{R}}^x(\tau_k) = \boldsymbol{\mathcal{R}}_h^x(\tau_k) \oplus \boldsymbol{\mathcal{R}}_p([0,t_r])$   ▷ LDI solution within the interval $\tau_k$
```

2.4.3 Handling the Set of Linearization Errors

So far we presented the algorithmic procedure to compute the reachable set of differential variables $\boldsymbol{\mathcal{R}}^x(\tau_k)$ for the time interval τ_k as summarized in Alg. 1. However, this was shown without illustrating how to obtain the set of linearization errors $\boldsymbol{\mathcal{L}}^{\boldsymbol{x}}$ and $\boldsymbol{\mathcal{L}}^y$ originating in (2.42) and (2.43), respectively. Furthermore, we still have not computed the reachable set of the algebraic constraints $\boldsymbol{\mathcal{R}}^y(\tau_k)$. In the following we detail how to compute the remaining sets to fully describe the reachable set of the DAE system.

Recall from Prop. 2.1 that the Lagrangian remainders enclose all higher order terms of the Taylor expansion when the variable $\boldsymbol{\mu}$ takes any value from the set $\boldsymbol{\zeta} = \{\boldsymbol{\alpha}\tilde{\boldsymbol{z}} + (1-\boldsymbol{\alpha})\boldsymbol{z}_k,\, 0 \leq \alpha_i \leq 1,\, \boldsymbol{z} \in \boldsymbol{\mathcal{R}}^z\}$.

The problem is that $\boldsymbol{\mu}$ depends on the time variation of the vector $\tilde{\boldsymbol{z}}$, which in turn depends on the remainder terms; notice from (2.46) that the set of uncertain inputs depends on the computation of the linearization errors, and consequently the reachable set $\boldsymbol{\mathcal{R}}^x(\tau_k)$ cannot be estimated via Alg. 1. To overcome this mutual dependency, we introduce the constant sets $\boldsymbol{\mathcal{L}}^x_{\max}$ and $\boldsymbol{\mathcal{L}}^y_{\max}$ denoting the maximum allowable linearization errors, which has to be specified by the user. Using $\boldsymbol{\mathcal{L}}_{\max}$ one can make an initial assumption of the Lagrangian remainders, then insert their values into (2.46), and apply the exact procedure summarized in Alg. 1 to compute the reachable set of differential variables $\boldsymbol{\mathcal{R}}^x(\tau_k)$.

The next step is to evaluate the reachable set of algebraic variables $\boldsymbol{\mathcal{R}}^y(\tau_k)$ in order to fully describe the set of all state variables required to evaluate the exact linearization errors. Analogously to (2.44), due to the index-1 property of the DAE system, the set of algebraic constraints based on $\boldsymbol{\mathcal{R}}^x(\tau_k)$ is

$$\boldsymbol{\mathcal{R}}^y(\tau_k) = \boldsymbol{y}_k \oplus (-\boldsymbol{F}_k^{-1}) \cdot \left[\boldsymbol{g}(\boldsymbol{z}_k) \oplus \boldsymbol{D}_k \underbrace{(\boldsymbol{\mathcal{R}}^x(\tau_k) \oplus (-\boldsymbol{x}_k))}_{=:\boldsymbol{\mathcal{R}}^x_\Delta(\tau_k)} + \boldsymbol{E}_k \underbrace{(\boldsymbol{\mathcal{U}}_{\max} \oplus (-\boldsymbol{u}_k))}_{\boldsymbol{\mathcal{U}}_{\Delta,\max}} \oplus \boldsymbol{\mathcal{L}}^y_{\max} \right], \tag{2.54}$$

with $\boldsymbol{\mathcal{U}}_{\max}$ as the maximum allowable uncertain inputs resulting from inserting $\boldsymbol{\mathcal{L}}_{\max}$ in (2.46). Hence, the set of differential and algebraic variables can be written in compact form as

$$\begin{pmatrix} \boldsymbol{\mathcal{R}}^x(\tau_k) \\ \boldsymbol{\mathcal{R}}^y(\tau_k) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{y}_k - \boldsymbol{F}_k^{-1}\boldsymbol{g}(\boldsymbol{z}_k) \end{pmatrix} \oplus \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{F}_k^{-1}\boldsymbol{D}_k & -\boldsymbol{F}_k^{-1}\boldsymbol{E}_k & -\boldsymbol{F}_k^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\mathcal{R}}^x_\Delta(\tau_k) \\ \boldsymbol{\mathcal{U}}_{\Delta,\max} \\ \boldsymbol{\mathcal{L}}^y_{\max} \end{pmatrix}, \tag{2.55}$$

with $\boldsymbol{I}$ being the identity matrix. Suppose the zonotopes $\boldsymbol{\mathcal{R}}^x(\tau_k) := [\boldsymbol{c}^x_z, \boldsymbol{G}^x_z]_{\mathcal{Z}}$, $\boldsymbol{\mathcal{U}}_{\max} := [\boldsymbol{c}^u_z, \boldsymbol{G}^u_z]_{\mathcal{Z}}$, and $\boldsymbol{\mathcal{L}}^y_{\max} := [\boldsymbol{c}^l_z, \boldsymbol{G}^l_z]_{\mathcal{Z}}$ are given, then the reachable set of differential and algebraic variables $\boldsymbol{\mathcal{R}}(\tau_k)$ is expressed via

$$\begin{aligned} \boldsymbol{\mathcal{R}}(\tau_k) &= \left[\begin{pmatrix} \boldsymbol{c}^x_z \\ \boldsymbol{c}^y_z \end{pmatrix}, \begin{pmatrix} \boldsymbol{G}^x_z & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{F}_k^{-1}\boldsymbol{D}_k\boldsymbol{G}^x_z & -\boldsymbol{F}_k^{-1}\boldsymbol{E}_k\boldsymbol{G}^u_z & -\boldsymbol{F}_k^{-1}\boldsymbol{G}^l_z \end{pmatrix} \right]_{\mathcal{Z}}, \\ \text{with :} \quad & \boldsymbol{c}^y_z = \boldsymbol{y}_k - \boldsymbol{F}_k^{-1} \left(\boldsymbol{g}(\boldsymbol{z}_k) + \boldsymbol{D}_k \left(\boldsymbol{c}^x_z - \boldsymbol{x}_k \right) + \boldsymbol{E}_k \left(\boldsymbol{c}^u_z - \boldsymbol{u}_k \right) + \boldsymbol{c}_l \right). \end{aligned} \tag{2.56}$$

Remark 2.9. Another possibility to express the set $\boldsymbol{\mathcal{R}}(\tau_k)$ is to easily compute the Cartesian product of zonotopes as in (2.36), that is $\boldsymbol{\mathcal{R}}(\tau_k) := \boldsymbol{\mathcal{R}}^x(\tau_k) \times \boldsymbol{\mathcal{R}}^y(\tau_k)$; however (2.56) is in fact tighter than the Cartesian product as the latter has as many generators as the number of generators of $\boldsymbol{\mathcal{R}}^x(\tau_k)$. Furthermore, the complexity of representing the zonotope enclosing differential and algebraic variables would grow substantially as the number of generators increases by applying the Cartesian product at each time step.

2.4.4 Bounding the Lagrangian Remainder

With the computation of the reachable set $\boldsymbol{\mathcal{R}}(\tau_k)$ based on $\boldsymbol{\mathcal{L}}_{\max}$, it remains to compute the real set of linearization errors $\boldsymbol{\mathcal{L}}(\tau_k)$ and verify that the following condition holds

$$\forall t \in \tau_k : \ \boldsymbol{\mathcal{L}}(\tau_k) \subseteq \boldsymbol{\mathcal{L}}_{\max}. \tag{2.57}$$

Remark 2.10. If the constraining condition (2.57) does not hold, one can reduce the time increment t_r to directly influence the quality of the over-approximation, as seen in (2.53). Another possibility would be to continuously split the reachable set into smaller sets, thus minimizing expansion of the Lagrangian remainders to fulfill the constraining condition. Splitting the reachable set, however, is not practical for high-dimensional systems as it results in an exponential complexity with respect to the system dimension. In fact, splitting of reachable sets is not employed in this thesis, and it is assumed that the reachable set is not converging, i.e. the system trajectory is unbounded (not stable), if the set of Lagrangian remainder exceeds the maximum allowable linearization errors. Further discussion and illustration of the aforementioned is shown later in Example 2.4.

To simplify the notation for further derivation, we introduce $\boldsymbol{\mathcal{R}}^z(\tau_k) := \boldsymbol{\mathcal{R}}(\tau_k) \times \boldsymbol{\mathcal{U}}$ and rewrite the set of Lagrangian remainders $\boldsymbol{\mathcal{L}}^x$ and $\boldsymbol{\mathcal{L}}^y$ in the following compact form:

$$\boldsymbol{\mathcal{L}} = \left\{ \frac{1}{2} \left(\boldsymbol{L}^x - \boldsymbol{C}_k \boldsymbol{F}_k^{-1} \boldsymbol{L}_y \right) \ : \ L_j^x = \tilde{\boldsymbol{z}}^T \boldsymbol{H}^{x,(j)}(\boldsymbol{\mu}) \tilde{\boldsymbol{z}}, \, L_i^y = \tilde{\boldsymbol{z}}^T \boldsymbol{H}^{y,(i)}(\boldsymbol{\mu}) \tilde{\boldsymbol{z}}, \, \boldsymbol{\mu} \in \boldsymbol{\zeta}, \, \tilde{\boldsymbol{z}} \in \boldsymbol{\mathcal{R}}^z \right\}, \tag{2.58}$$

Here $\boldsymbol{H}^{x,(j)} \in \mathbb{R}^{n_z \times n_z}$, $j \in \{1, \ldots, n_x\}$, and likewise $\boldsymbol{H}^{y,(i)}$, $i \in \{1, \ldots, n_y\}$, corresponds to a discrete set of Hessian matrices. The collection of these matrices creates a Hessian tensor; in other words, this set can be regarded as an array of n_x matrices for the functions $\boldsymbol{f}(\boldsymbol{z})$, and n_y matrices for the function $\boldsymbol{g}(\boldsymbol{z})$. Each matrix is associated with the second-order partial derivatives of the j-th and the i-th coordinate of the vector-field function $\boldsymbol{f}(\boldsymbol{z})$ and $\boldsymbol{g}(\boldsymbol{z})$, respectively. The Hessian matrix in each j-th dimension of the function describing the differential equations is:

$$\boldsymbol{H}^{x,(j)}(\boldsymbol{\mu}) := \begin{pmatrix} \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_1^2}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} & \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_1 z_2}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} & \cdots & \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_1 \partial z_{n_z}}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} \\ \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_2 z_1}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} & \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_2^2}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} & \cdots & \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_2 \partial z_{n_z}}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} \\ \vdots & \vdots & \ddots & \vdots \\ \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_{n_z} \partial z_1}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} & \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_{n_z} \partial z_2}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} & \cdots & \left.\dfrac{\partial^2 f_j(\boldsymbol{z})}{\partial z_{n_z}^2}\right|_{\boldsymbol{z}=\boldsymbol{\mu}} \end{pmatrix}, \tag{2.59}$$

$$\text{with: } \boldsymbol{\mu}(\tau_k) \in \left\{ \boldsymbol{\alpha}\boldsymbol{z} + (1 - \boldsymbol{\alpha})\boldsymbol{z}_k \ : \ 0 \leq \alpha_i \leq 1, \, \boldsymbol{z} \in \boldsymbol{\mathcal{R}}^z(\tau_k) \right\}.$$

and the Hessian $\boldsymbol{H}^{y,(i)}(\boldsymbol{\mu})$ is formulated similarly by replacing $f_j(\boldsymbol{z})$ with $g_i(\boldsymbol{z})$.

To obtain a tight over-approximation of $\boldsymbol{\mathcal{L}}$, one has to compute the discrete set of the Hessian matrices $\boldsymbol{H}^{x,(j)}(\boldsymbol{\mu})$, $\boldsymbol{H}^{y,(i)}(\boldsymbol{\mu})$. This is done first by over-approximating the set of states variables $\boldsymbol{\mathcal{R}}^z(\tau_k)$ via

$$\boldsymbol{\mathcal{R}}^z(\tau_k) \subset \boldsymbol{\mathcal{I}}^z(\tau_k) := \textbf{interval}(\boldsymbol{\mathcal{R}}^z(\tau_k)), \tag{2.60}$$

which is obtained according to (2.34). The set $\boldsymbol{\mathcal{I}}^z$ denotes the multidimensional interval enclosing all possible trajectories of the differential and algebraic state variables, in addition to the uncertain inputs. Then by employing interval arithmetics (see Sec. 2.3.8), it is possible to bound each entry of the Hessian matrices of the differential variables as follows

$$\forall j \in \{1, \ldots, n_x\} : \ \boldsymbol{\mathcal{H}}^{x,(j)} = \left\{\boldsymbol{H}^{x,(j)}(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \boldsymbol{\mathcal{I}}^z\right\} := \left[\underline{\boldsymbol{H}}^{x,(j)}, \overline{\boldsymbol{H}}^{x,(j)}\right]. \tag{2.61}$$

Here $\boldsymbol{\mathcal{H}}^{x,(j)}$ denotes the set enclosing all possible values of each j-th entry of the Hessian tensor of differential variables; in fact this set corresponds to an interval matrix with a lower and an upper bound denoted by $\underline{\boldsymbol{H}}^{(j)}$, and $\overline{\boldsymbol{H}}^{(j)}$, respectively. Similar formulation holds for each i-th entry of the Hessian matrices of the algebraic variables, that is

$$\forall i \in \{1, \ldots, n_y\} : \ \boldsymbol{\mathcal{H}}^{y,(i)} = \left\{\boldsymbol{H}^{y,(i)}(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \boldsymbol{\mathcal{I}}^z\right\} := \left[\underline{\boldsymbol{H}}^{y,(i)}, \overline{\boldsymbol{H}}^{y,(i)}\right]. \tag{2.62}$$

Since the set of Hessian matrices are now bounded within specified intervals, it is possible to compute the set of Lagrangian reminders; simply by inserting (2.61) and (2.62) in (2.58) yields

$$\begin{aligned}\boldsymbol{\mathcal{L}}(\tau_k) = \Bigg\{ \frac{1}{2}\left(\boldsymbol{L}^x - \boldsymbol{C}_k \boldsymbol{F}_k^{-1} \boldsymbol{L}_y\right) \ : \ L^x_j = \tilde{\boldsymbol{z}}^T \boldsymbol{H}^{x,(j)}(\boldsymbol{\mu})\tilde{\boldsymbol{z}},\ L^y_i = \tilde{\boldsymbol{z}}^T \boldsymbol{H}^{y,(i)}(\boldsymbol{\mu})\tilde{\boldsymbol{z}}, \\ \tilde{\boldsymbol{z}} \in \boldsymbol{\mathcal{R}}^z(\tau_k),\ \boldsymbol{H}^{x,(j)}(\boldsymbol{\mu}) \in \boldsymbol{\mathcal{H}}^{x,(j)},\ \boldsymbol{H}^{y,(i)}(\boldsymbol{\mu}) \in \boldsymbol{\mathcal{H}}^{y,(i)} \Bigg\}.\end{aligned} \tag{2.63}$$

The problem with (2.63) is that it heavily relies on interval arithmetics to compute the set of linearization errors. Furthermore, the result of the over-approximation is rather conservative and does not tightly enclose the exact solution. The alternative solution is an interval-free procedure employing the so-called quadratic mapping of zonotopes according to the terminology introduced in [11]:

Proposition 2.3. Quadratic Mapping of Zonotopes: Consider the following set

$$\boldsymbol{\mathcal{Z}}_H := \left\{ \boldsymbol{\delta} \in \mathbb{R}^n \ : \ \delta_i = \sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} z_j z_k,\ \boldsymbol{z} \in \boldsymbol{\mathcal{Z}} := \left[\boldsymbol{c}_z, \boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(p)}\right]_{\boldsymbol{\mathcal{Z}}},\ \boldsymbol{H}^{(i)} \in \mathbb{R}^{n\times n} \right\},$$

involving a quadratic multiplication of the vector $\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}$ with the discrete set of matrices $\boldsymbol{H}^{(i)}$. The non-convex set $\boldsymbol{\mathcal{Z}}_H$ can be over-approximated via the zonotope $\boldsymbol{\mathcal{Z}}_Q$ according to:

$$\boldsymbol{\mathcal{Z}}_H \subseteq \boldsymbol{\mathcal{Z}}_Q := \textbf{quad}\left(\boldsymbol{\mathcal{Z}}, \boldsymbol{H}^{(1)}, \ldots, \boldsymbol{H}^{(n)}\right) := \left[\boldsymbol{c}^q_z, \boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(\sigma)}\right]_{\boldsymbol{\mathcal{Z}}} \tag{2.64}$$

$$\text{with:}\begin{cases} & c^{q}_{z,i} := \displaystyle\sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} c_{z,j} c_{z,k} + \frac{1}{2}\sum_{m=1}^{p}\sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} g^{(m)}_{z,j} g^{(m)}_{z,k}, \\ h \in \{1 \ldots p\} : & q^{(h)}_{i} := \displaystyle\sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} c_{z,j} g^{(h)}_{z,k} + \sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} c_{z,k} g^{(h)}_{z,j}, \\ & q^{(p+h)}_{i} := \displaystyle\sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} g^{(h)}_{z,j} g^{(h)}_{z,k}, \\ l = \displaystyle\sum_{o=1}^{p-1}\sum_{r=o+1}^{p} 1 : & q^{(2p+l)}_{i} := \displaystyle\sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} g^{(o)}_{j} g^{(r)}_{k} + \sum_{j=1}^{n}\sum_{k=1}^{n} H^{(i)}_{jk} g^{(o)}_{k} g^{(r)}_{j}. \end{cases}$$

Here $\boldsymbol{c}^{q}_{z} \in \mathbb{R}^{n}$ and $\boldsymbol{q}^{(i)}_{z} \in \mathbb{R}^{n}$, $i \in \{1, \ldots \sigma\}$ corresponds to the center and generators of the zonotope $\boldsymbol{\mathcal{Z}}_{Q}$, respectively, and σ denotes the number of generators obtained using a binomial coefficient expressed via

$$\frac{(p+2)!}{2!\,p!} - 1,$$

where p is the number of generators corresponding to the zonotope $\boldsymbol{\mathcal{Z}}$. □

Notice how the set $\boldsymbol{\mathcal{Z}}_{H}$ demonstrated for Prop. 2.3 is similar to (2.63); however, the main difference is that the set of discrete matrices is a set of interval matrices. To overcome this limitation, some modifications are necessary to use the operator **quad**. First, we describe $\left(\boldsymbol{\mathcal{H}}^{x,(j)}, \boldsymbol{\mathcal{H}}^{y,(i)}\right)$, $(j,i) \in \{1, \ldots, (n_x, n_y)\}$ in terms of their center and radius, rather than their lower and upper bound, that is:

$$\boldsymbol{\mathcal{H}}^{x,(j)} = \boldsymbol{H}^{x,(i)}_{c} \oplus \left[-\boldsymbol{H}^{x,(i)}_{r},\, \boldsymbol{H}^{x,(i)}_{r}\right] \tag{2.65}$$

$$\boldsymbol{\mathcal{H}}^{y,(i)} = \boldsymbol{H}^{y,(i)}_{c} \oplus \left[-\boldsymbol{H}^{y,(i)}_{r},\, \boldsymbol{H}^{y,(i)}_{r}\right] \tag{2.66}$$

$$\text{with :}\begin{cases} \boldsymbol{H}^{x,(j)}_{c} = \dfrac{\overline{\boldsymbol{H}}^{x,(j)} + \underline{\boldsymbol{H}}^{x,(j)}}{2}, \boldsymbol{H}^{x,(j)}_{r} = \dfrac{\overline{\boldsymbol{H}}^{x,(j)} - \underline{\boldsymbol{H}}^{x,(j)}}{2} \\ \boldsymbol{H}^{y,(i)}_{c} = \dfrac{\overline{\boldsymbol{H}}^{y,(i)} + \underline{\boldsymbol{H}}^{y,(i)}}{2}, \boldsymbol{H}^{y,(i)}_{r} = \dfrac{\overline{\boldsymbol{H}}^{y,(i)} - \underline{\boldsymbol{H}}^{y,(i)}}{2} \end{cases} \tag{2.67}$$

with $\boldsymbol{H}^{x,(j)}_{c}$ and $\boldsymbol{H}^{x,(j)}_{r}$ as the center and radius of the interval matrix $\boldsymbol{\mathcal{H}}^{x,(j)}$, and likewise for $\boldsymbol{\mathcal{H}}^{y,(i)}$. With the reformulation of the interval Hessian matrices, it is now possible to over-approximate the set of Lagrangian remainders $\boldsymbol{\mathcal{L}}^{x}$ and $\boldsymbol{\mathcal{L}}^{y}$, at each time interval τ_k, using the quadratic mapping of zonotopes according to [12, Corollary 1]:

$$\begin{aligned} \boldsymbol{\mathcal{L}}^{x}(\tau_k) &\subseteq \textbf{quad}\left(\boldsymbol{\mathcal{R}}^{z}, \boldsymbol{\mathcal{H}}^{x,(1)}, \ldots, \boldsymbol{\mathcal{H}}^{x,(n_x)}\right) := \textbf{quad}\left(\boldsymbol{\mathcal{R}}^{z}, \boldsymbol{H}^{x,(1)}_{c}, \ldots, \boldsymbol{H}^{x,(n_x)}_{c}\right) \oplus \left[-\boldsymbol{\Delta}^{x},\, \boldsymbol{\Delta}^{x}\right], \\ \boldsymbol{\mathcal{L}}^{y}(\tau_k) &\subseteq \textbf{quad}\left(\boldsymbol{\mathcal{R}}^{z}, \boldsymbol{\mathcal{H}}^{y,(1)}, \ldots, \boldsymbol{\mathcal{H}}^{y,(n_y)}\right) := \textbf{quad}\left(\boldsymbol{\mathcal{R}}^{z}, \boldsymbol{H}^{y,(1)}_{c}, \ldots, \boldsymbol{H}^{y,(n_y)}_{c}\right) \oplus \left[-\boldsymbol{\Delta}^{y},\, \boldsymbol{\Delta}^{y}\right], \\ \text{with :}&\begin{cases} \Delta^{x}_{i} := |\, \boldsymbol{\mathcal{R}}^{z,T}(\tau_k) \,|\, \boldsymbol{H}^{x,(i)}_{r} \,|\, \boldsymbol{\mathcal{R}}^{z}(\tau_k) \,|, \\ \Delta^{y}_{j} := |\, \boldsymbol{\mathcal{R}}^{z,T}(\tau_k) \,|\, \boldsymbol{H}^{y,(j)}_{r} \,|\, \boldsymbol{\mathcal{R}}^{z}(\tau_k) \,|, \end{cases} \end{aligned} \tag{2.68}$$

and $|\cdot|$ returns the element-wise absolute value.

Remark 2.11. The computational complexity of constructing a quadratic zonotope is $\mathcal{O}(n_z^5)$ with n_z corresponding to the number of differential, algebraic and input variables. Hence, the over-approximation of the set of Lagrangian remainders can become computationally expensive for high-dimensional DAE systems. An alternative, but a more conservative approach is to estimate the set of the linearization errors according to [13]

$$
\begin{aligned}
&\boldsymbol{\mathcal{L}}_i^x(\tau_k) \subseteq [-L_i^x(\tau_k), L_i^x(\tau_k)],\ \boldsymbol{\mathcal{L}}_j^y(\tau_k) \subseteq [-L_j^y(\tau_k), L_j^y(\tau_k)],\\
&\text{with} : \begin{cases} L_i^x = |\,\boldsymbol{\mathcal{I}}_\Delta^{z,T}(\tau_k)\,|\,\boldsymbol{\mathcal{H}}^{x,(i)}\,|\,\boldsymbol{\mathcal{I}}_\Delta^z(\tau_k)\,|, \\ L_j^y = |\,\boldsymbol{\mathcal{I}}_\Delta^{z,T}(\tau_k)\,|\,\boldsymbol{\mathcal{H}}^{y,(j)}\,|\,\boldsymbol{\mathcal{I}}_\Delta^z(\tau_k)\,|, \\ \boldsymbol{\mathcal{I}}_\Delta^z = \textbf{interval}\left(\boldsymbol{\mathcal{R}}_\Delta^z(\tau_k)\right),\, \boldsymbol{\mathcal{R}}_\Delta^z(\tau_k) := \boldsymbol{\mathcal{R}}(\tau_k) \oplus (-z_k). \end{cases}
\end{aligned}
\tag{2.69}
$$

which have a computational complexity of $\mathcal{O}(n_z^3)$ with respect to the system dimension.

Remark 2.12. In general, the over-approximation using (2.69) offers a trade-off between accuracy and computational efficiency; however it highly depends on the system dynamics. In fact, this approach leads to unacceptable over-approximation for the class of DAE systems as shown later in the discussion of the numerical examples, see example 2.4 later in this chapter.

2.4.4.1 Algorithmic Realization

Combining all previous results, the basic procedure to obtain the set of linearization errors, for each time intervals τ_k, is summarized in Alg. 2. First, the algorithm combines the set of differential and algebraic variables with the set of uncertain inputs, afterwards, using the interval enclosure as in (2.34), the bounds of all possible trajectories of the vector $\boldsymbol{z} := (\boldsymbol{x}^T, \boldsymbol{y}^T, \boldsymbol{u}^T)^T$ are obtained. This makes it possible to compute the set of Hessian matrices, which in turn specify bounds of the set of the Lagrangian remainder within the time interval τ_k using the quadratic mapping of zonotopes.

Algorithm 2 ErrorSolution

```
Require: Reachable set of differential and algebraic variables $\boldsymbol{\mathcal{R}}(\tau_k)$, set of uncertain inputs $\boldsymbol{\mathcal{U}}(\tau_k)$, and
    the Hessian tensors $\boldsymbol{H}^x$ and $\boldsymbol{H}^y$
Ensure: $\boldsymbol{\mathcal{L}}^x(\tau_k)$, $\boldsymbol{\mathcal{L}}^y(\tau_k)$
  $\boldsymbol{\mathcal{R}}^z(\tau_k) := \boldsymbol{\mathcal{R}}(\tau_k) \times \boldsymbol{\mathcal{U}}(\tau_k)$
  $\boldsymbol{\mathcal{I}}^z(\tau_k) \overset{(2.60)}{\leftarrow} \textbf{interval}\left(\boldsymbol{\mathcal{R}}^z(\tau_k)\right)$        ▷ Over-approximation by an interval hull
  $\left[\boldsymbol{\mathcal{H}}^x(\tau_k), \boldsymbol{\mathcal{H}}^y(\tau_k)\right] \xleftarrow{(2.61),\,(2.62)} \boldsymbol{\mathcal{I}}^z(\tau_k),$        ▷ Intervals of Hessian tensors
  $\left[\boldsymbol{\mathcal{L}}^x(\tau_k), \boldsymbol{\mathcal{L}}^y(\tau_k)\right] \overset{(2.68)}{\subseteq} \textbf{quad}\Big(\left(\boldsymbol{\mathcal{R}}^z(\tau_k),\ \boldsymbol{\mathcal{H}}^x(\tau_k), \boldsymbol{\mathcal{H}}^y(\tau_k)\right)\Big)$        ▷ Over-approximation
```

2.4.5 Overall Algorithm

Using Alg. 1 and Alg. 2, we can now summarize the overall numerical procedure to compute the reachable set of the DAE system (2.38) over the complete time-horizon $t \in [0, t_f]$, where t_f is the terminal time specified by the user. In Alg. 3 throught lines 3 and 4, the algorithm determines the local linearization point at the time-step k. Typically this point is associated with the estimate of the reachable set in the following time-step, that is $\boldsymbol{z}_k := \mathbf{center}(\boldsymbol{\mathcal{R}}(t_{k+1}) \times \boldsymbol{\mathcal{U}})$. Using the linearization point, the algorithm obtains an abstraction of the DAE described via a LDI consisting of a 1-st order Taylor expansion, in addition to the Lagrangian remainders. Since the set of linearization errors is not initially known, the algorithm uses the maximum allowable errors based on the user-defined variable $\boldsymbol{\mathcal{L}}_{\max}$, which in turn specifies the maximum set of uncertain inputs as seen in lines 5 and 6. With the knowledge of the LDI parameters abstracting the DAE system, the reachable set of differential variables $\boldsymbol{\mathcal{R}}^x(\tau_k)$ is computed for the time-step k according to Alg. 1.

Throughout lines $8 - 11$, the set of algebraic constraints $\boldsymbol{\mathcal{R}}^y(\tau_k)$ is constructed based on the knowledge of $\boldsymbol{\mathcal{R}}^x(\tau_k)$, which in turn specifies the zonotopic set enclosing the differential and algebraic variables $\boldsymbol{\mathcal{R}}(\tau_k)$, in addition to the exact Lagrangian remainders $\boldsymbol{\mathcal{L}}(\tau_k)$ determined as summarized in Alg. 2.

Since the computation of $\boldsymbol{\mathcal{L}}(\tau_k)$ is associated with the initial assumption $\boldsymbol{\mathcal{L}}_{\max}$, the algorithm checks whether the computed linearization errors are in fact a subset of the maximum allowable Lagrangian remainders. This is specified via the conditional if described in line 13; if $\boldsymbol{\mathcal{L}}(\tau_k) \nsubseteq \boldsymbol{\mathcal{L}}_{\max}$, it is assumed that the system trajectories starting from the initial reachable set $\boldsymbol{\mathcal{R}}(0)$ are not converging to a stable equilibrium, see Remark 2.10. Otherwise, the algorithm re-computes the set of uncertain inputs based on $\boldsymbol{\mathcal{L}}(\tau_k)$ and re-evaluates the reachable set of differential variables for the following time step $k+1$. The aforementioned procedure is continued until $t > t_f$. Afterwards the reachable set is assembled via the union of the computed sets, that is:

$$\boldsymbol{\mathcal{R}}([0, t_f]) := \bigcup_{k=1}^{tf/t_r} \boldsymbol{\mathcal{R}}(\tau_k) \overset{(2.26)}{\supseteq} \mathbf{reach}(\boldsymbol{\mathcal{R}}(0), \boldsymbol{\mathcal{U}}, t_f).$$

2.5 Numerical Examples

In this section, we demonstrate the presented reachability algorithm on the power system models presented earlier in sec. 2.1. All computations are performed on a standard computer with an Intel Core i7-4810MQ CPU and 16GB of RAM. The algorithms are implemented in MATLAB2016b using the **Co**ntinuous **R**eachability **A**nalyzer (CORA) toolbox[1]. The toolbox is collection of MATLAB classes for the formal verification of cyber-physical systems using reachability analysis. CORA interfaces with the

[1] `http://www6.in.tum.de/Main/SoftwareCORA` [9]

Algorithm 3 NonLinearReachability

```
Require: Initial reachable set of differential variables $\boldsymbol{\mathcal{R}}^x(0)$, set of uncertain inputs $\boldsymbol{\mathcal{U}}$, the time increment $t_r$, number of Taylor terms $\sigma$ of the matrix exponential, the maximum allowable set of Lagrangian remainders for differential and algebraic variables $\boldsymbol{\mathcal{L}}_{\max}$
Ensure: $\boldsymbol{\mathcal{R}}^x([0,t_f]) := \bigcup_{m=1}^{k} \boldsymbol{\mathcal{R}}^x(\tau_m)$, $\boldsymbol{\mathcal{R}}^y([0,t_f]) := \bigcup_{m=1}^{k} \boldsymbol{\mathcal{R}}^y(\tau_m)$
Initialization: $k = 0$, $t_k = 0$, $t_{k+1} = t_r$, $\tau_k = [t_k, t_{k+1}]$, and isConverging := true
repeat
    $\boldsymbol{z}_k \leftarrow \Big(\boldsymbol{\mathcal{R}}^x(t_k),\, \boldsymbol{\mathcal{R}}^y(t_k),\, \boldsymbol{\mathcal{U}}\Big)$    ▷ Local linearization point
    $\Big(\tilde{\boldsymbol{A}}_k, \tilde{\boldsymbol{B}}_k, \boldsymbol{C}_k,\, \boldsymbol{F}_k,\, \boldsymbol{w}(\boldsymbol{z}_k),\, \boldsymbol{H}^x,\, \boldsymbol{H}^y\Big) \overset{(2.42)}{\longleftarrow} \textbf{taylor}(\boldsymbol{f}(\boldsymbol{z}), \boldsymbol{g}(\boldsymbol{z}))$    ▷ Taylor expansion around $\boldsymbol{z}_k$
    $\boldsymbol{\mathcal{L}}_{\max} = \boldsymbol{\mathcal{L}}^x_{\max} \oplus \left(-\boldsymbol{C}_k \boldsymbol{F}_k^{-1} \boldsymbol{\mathcal{L}}^y_{\max}\right)$    ▷ Maximum Lagrangian remainders
    $\boldsymbol{\mathcal{U}}_{\max} \overset{(2.46)}{=} \boldsymbol{w}(\boldsymbol{z}_k) \oplus \tilde{\boldsymbol{B}}_k\left(\boldsymbol{\mathcal{U}} \oplus (-\tilde{\boldsymbol{u}}_k)\right) \oplus \boldsymbol{\mathcal{L}}_{\max}$    ▷ Maximum uncertain inputs
    $\boldsymbol{\mathcal{R}}^x(\tau_k) \overset{\text{Alg.1}}{\longleftarrow} \textbf{ReachNext}\Big(\tilde{\boldsymbol{A}}_k,\, t_r,\, \sigma,\, \boldsymbol{\mathcal{R}}^x(t_k), \boldsymbol{\mathcal{U}}_{\max}\Big)$    ▷ Reachable set of differential variables
    $\boldsymbol{\mathcal{R}}^y(\tau_k) \overset{(2.54)}{\longleftarrow} \Big(\boldsymbol{z}_k,\, \boldsymbol{\mathcal{R}}^x(\tau_k),\, \boldsymbol{\mathcal{L}}^y_{\max}, \boldsymbol{\mathcal{U}}_{\max}\Big)$    ▷ Reachable Set of algebraic variables
    $\boldsymbol{\mathcal{R}}(\tau_k) \overset{\text{remark 2.9}}{\longleftarrow} \boldsymbol{\mathcal{R}}^x(\tau_k) \times \boldsymbol{\mathcal{R}}^y(\tau_k)$    ▷ Reachable set of Differential and algebraic variables
    $\Big[\boldsymbol{\mathcal{L}}^x(\tau_k), \boldsymbol{\mathcal{L}}^y(\tau_k)\Big] \overset{\text{Alg.2}}{\longleftarrow} \textbf{ErrorSolution}\left(\boldsymbol{\mathcal{R}}(\tau_k), \boldsymbol{\mathcal{U}}_{\max},\, \boldsymbol{H}^x,\, \boldsymbol{H}^y\right)$    ▷ Lagrangian remainders
    $\boldsymbol{\mathcal{L}}(\tau_k) \overset{(2.45)}{=} \left[\boldsymbol{\mathcal{L}}^x(\tau_k) \oplus \boldsymbol{C}_k \boldsymbol{F}_k^{-1} \boldsymbol{\mathcal{L}}^y(\tau_k)\right]$
    if $\boldsymbol{\mathcal{L}}(\tau_k) \nsubseteq \boldsymbol{\mathcal{L}}_{\max}$ then
        isConverging := false    ▷ Reachable set not converging to an equilibrium point
    else
        $\tilde{\boldsymbol{\mathcal{U}}} \overset{(2.46)}{=} \boldsymbol{w}(\boldsymbol{z}_k) \oplus \tilde{\boldsymbol{B}}_k\left(\boldsymbol{\mathcal{U}} \oplus (-\tilde{\boldsymbol{u}}_k)\right) \oplus \boldsymbol{\mathcal{L}}(\tau_k)$    ▷ Recompute set of uncertain inputs
        $\boldsymbol{\mathcal{R}}^x(t_{k+1}) \overset{\text{Alg.1}}{\longleftarrow} \textbf{ReachNext}\Big(\tilde{\boldsymbol{A}}_k,\, t_r,\, \sigma,\, \boldsymbol{\mathcal{R}}^x(t_k),\, \tilde{\boldsymbol{\mathcal{U}}}\Big)$    ▷ Set in next point in time
        $t_k := t_k + t_r$, $k := k + 1$
    end if
until $t_k > t_f$
```

Multi-**P**arametric **T**oolbox (MPT) toolbox[1] which is also written in MATLAB. In the previous release, the CORA toolbox used to interface as well with **Int**erval **Lab**oratory (INTLAB) toolbox[2], however CORA no longer uses INTLAB as it is not freely available anymore.

In this chapter we shall consider two benchmark problems commonly employed in the power system community; namely the single-machine infinite bus (SMIB) [81, Ch. 12] and the IEEE 9-bus [14, Ch. 2]. Furthermore, we only use the synchronous machine as the generating unit modelled via the swing equation (2.10); hence the mathematical model governing dynamics of the power system under consideration

[1] `http://people.ee.ethz.ch/~mpt/3/` [64]

[2] `http://www.ti3.tu-harburg.de/rump/intlab/` [119]

simplifies to the differential equations:

$$\begin{aligned} \dot{\delta}_j &= \omega_s \left(\omega_j - \omega_{\text{ref}}\right), \\ \dot{\omega}_j &= \frac{1}{2H_j}\left(P_{m,j} - P_{e,j} - D_j(\omega_j - \omega_{\text{ref}})\right), \\ \text{with:} \quad P_{e,j} &= E'_{q,j} V_j Y^g_j \cos(\Theta^g_j + \delta_j - \theta_j) - V_j^2 Y^g_j \cos(\Theta^g_j), \end{aligned} \tag{2.70}$$

and the power-flow equations:

$$\begin{aligned} P_i &= P_{e,i} - P_{l,i} = V_i \sum_{k\in\mathcal{N}} V_i Y_{ik} \cos(\Theta_{ik} - \theta_k - \theta_i), \\ Q_i &= Q_{e,i} - Q_{l,i}, = -V_i \sum_{k\in\mathcal{N}} V_i Y_{ik} \sin(\Theta_{ik} - \theta_k - \theta_i). \end{aligned} \tag{2.71}$$

Here the subscripts j and i specify the j-th machine and the i-th bus, respectively. It is assumed that the mechanical power P_m and the generator voltage E'_q are kept constant. The constants Y^g and Θ^g denote to the absolute value and the phase angle of the admittance spanned from the generator to its corresponding bus. All remaining variables and parameters were previously introduced as in Table 2.1 and Sec. 2.1.2.

Remark 2.13. Notice that the DAE system described via (2.70) and (2.71) can be simplified to an explicit ODE system if we replace the bus voltage V_j by a constant value which in turn eliminates the need to solve the set of algebraic equations associated with the power-flow equations (2.71). Clearly this assumption is unrealistic and does not hold for many practical situations. Note that this is a basic, and necessary, assumption employed to establish transient stability via Lyapunov direct methods [118]. This assumption is not required to assess transient stability using reachability analysis.

Remark 2.14. We simplified the models in (2.70) just to focus on the demonstration of the reachability algorithm for readers unfamiliar with reachability analysis. It should be stressed that we employ the introduced power system models without any modelling simplification in the following chapters.

2.5.1 Single-Machine Infinite Bus

The first example is the so-called SMIB system. The system illustrated in Fig. 2.13 consists of a synchronous generator connected to an infinite bus; a special bus whose voltage and phase angle are kept constant regardless of the changes occurring in the network. The SMIB power system is compromised of six state variables: two variables corresponding to the generator dynamic states appearing in (2.70) and four algebraic variables associated with the constraints at the generator bus.

2.5.1.1 SMIB Linear Model

Here we shall consider the linear model of the SMIB system.

Example 2.2. The linearization of the SMIB power system at its equilibrium point results in the LDI on the form $\dot{\boldsymbol{x}}(t) \in \boldsymbol{A}\boldsymbol{x}(t) \oplus \boldsymbol{\mathcal{U}}$ such that

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ 14 & -0.5 \end{pmatrix}, \ \boldsymbol{\mathcal{U}} = \begin{pmatrix} [0,\, 0] \\ [-0.03,\, 0.03] \end{pmatrix}, \ \boldsymbol{\mathcal{R}}(\mathbf{0}) = \begin{pmatrix} [0.7,\, 1.1] \\ [-0.2,\, 0.2] \end{pmatrix}.$$

with the state variables $\boldsymbol{x} = (\delta,\, \omega)^T$, and $\boldsymbol{\mathcal{U}}$ as a set of input disturbance affecting the frequency ω.

The reachable set is computed according to Alg. 1 starting from the initial reachable set $\boldsymbol{\mathcal{R}}(0)$, over a time-horizon $t_f = 5\,\mathrm{s}$ under the influence of the set of uncertain inputs $\boldsymbol{\mathcal{U}}$. The chosen time-increment is $t_r = 0.005\,\mathrm{s}$ and the number of Taylor terms specified for the matrix exponential is $\sigma = 6$. The time-domain projection of the reachable set is shown in Fig. 2.14 in addition to randomly generated trajectories ($n = 20$). Notice how the reachable set encloses all possible system trajectories by running the reachability algorithm just once, whereas one needs to run several simulations (infinitely many) in order to approximate the same result.

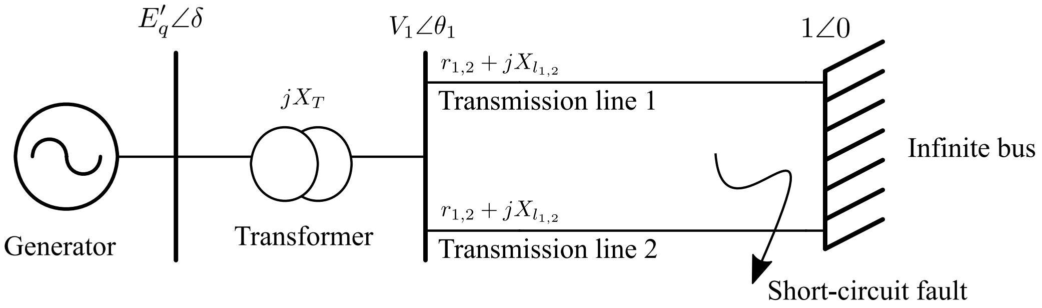


Figure 2.13: The single-machine infinite bus benchmark problem.

2.5.1.2 SMIB DAE Model

Now we consider the SMIB system modelled via the set of nonlinear DAEs described in (2.70) and (2.71).

Example 2.3. Here the fault scenario under consideration is the loss of the second transmission line at $t = 0.01\,\mathrm{s}$, followed by its reconnection to the network after clearance of the fault $t = 0.02$. The reachable set is computed according to Alg. 3 starting from $\boldsymbol{\mathcal{R}}^x(0) = ([0.65075,\, 0.66675]\ [-0.008, 0.008])^T$. We include uncertainty in the initial set of differential variables, since initial states are not exactly known due to increasingly varying operating conditions in current power systems. Here we do not specify the terminal time t_f, instead, the reachability algorithm runs until all states are enclosed by $\boldsymbol{\mathcal{R}}(0)$, which occurs at $t = 0.23$.

The chosen time-increment is $t_r = 0.0006\,s$ and the number of Taylor terms specified for the matrix exponential is $\sigma = 6$. Fig. 2.15 shows projection corresponding to the reachable set of the differential

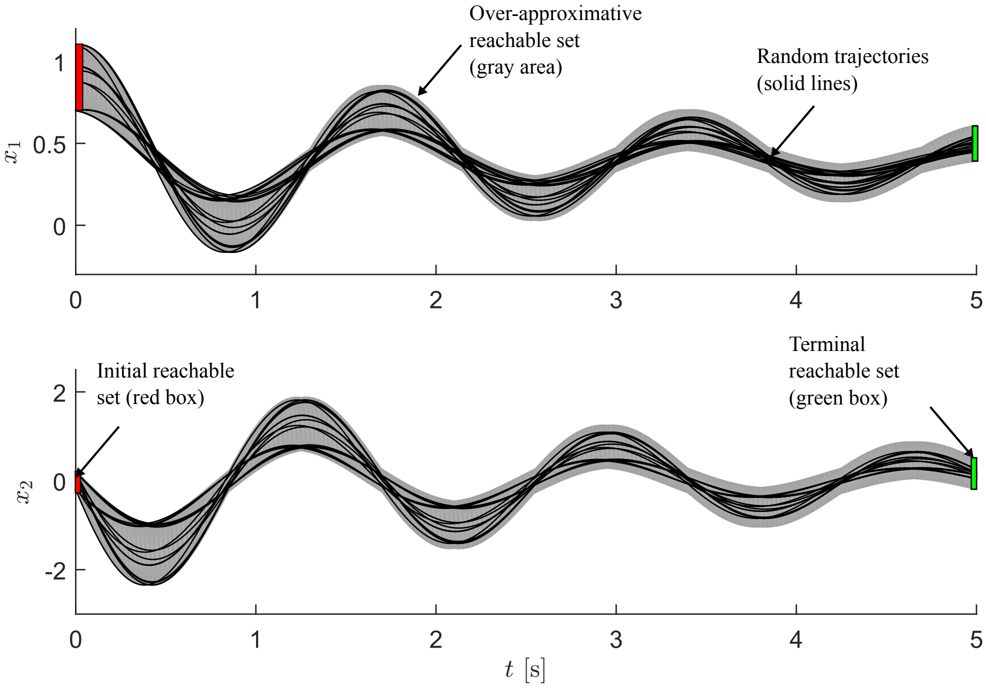


Figure 2.14: Time-domain bounds of the reachable set of Example 2.2.

state variables δ and ω starting from the initial reachable set $\boldsymbol{\mathcal{R}}^x(0)$. Fig. 2.16 illustrates the time-domain bounds of the algebraic constraints $\boldsymbol{y} = (P, Q, V, \theta)^T$ at bus 1. Here the discontinuous jump occurring in the reachable set is associated with the fault scenario. Immediately after losing the transmission line, entries of the admittance matrix (2.5) change thus leading to a discontinuous jump in the algebraic variables to satisfy the power flow equations (2.71).

2.5.1.3 Effect of the Algorithm Parameters

Now we discuss the effect of the parameters associated with the reachability algorithm; namely we consider the time-increment t_r and how the set of the Lagrangian remainders is computed.

Example 2.4. In Fig. 2.17(a), the set of linearization errors is computed according to the conservative method in (2.69), see Remark (2.11), rather than the tight technique using the quadratic mapping of zonotopes suggested in Prop. 2.3. It can be seen that after four time steps, the constraining condition $\boldsymbol{\mathcal{L}} \subseteq \boldsymbol{\mathcal{L}}_{\max}$ was not fulfilled, and Alg. 3 terminates as the reachable set is no longer converging to the initial operating point. This is clearly not true, as we just demonstrated that the power system is clearly stable for the considered fault scenario. Here, the reachability algorithm fails to converge as the computation of the set $\boldsymbol{\mathcal{L}}$ led to unacceptable over-approximation of the reachable set.

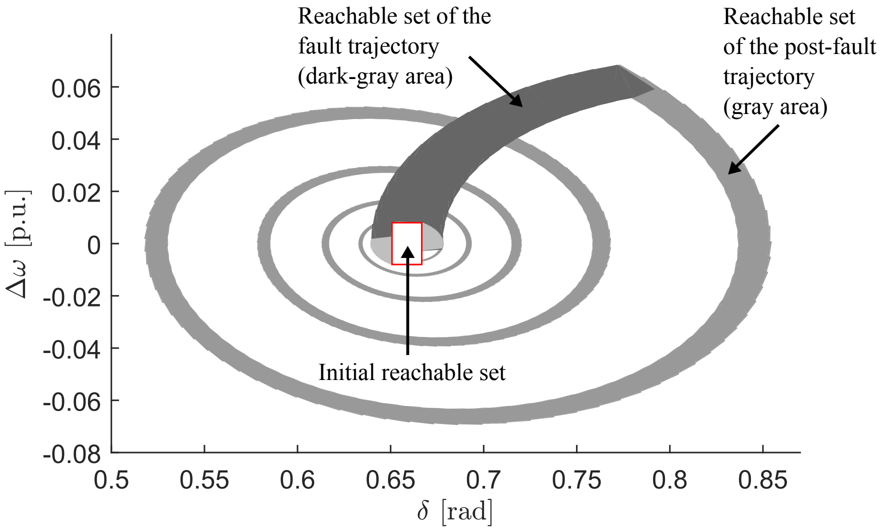


Figure 2.15: Reachable set projection of the differential variables δ and ω of the SMIB power system. The dark-gray area show the reachable set during the fault, and the gray area specifies the reachable set of the post-fault trajectory. The considered fault scenario is the loss of the second transmission line connecting the synchronous generator to the infinite bus. The line is reconnected after the clearance of the fault, and the reachable set is computed until all states are enclosed by the initial set of states $\mathcal{R}(0)$.

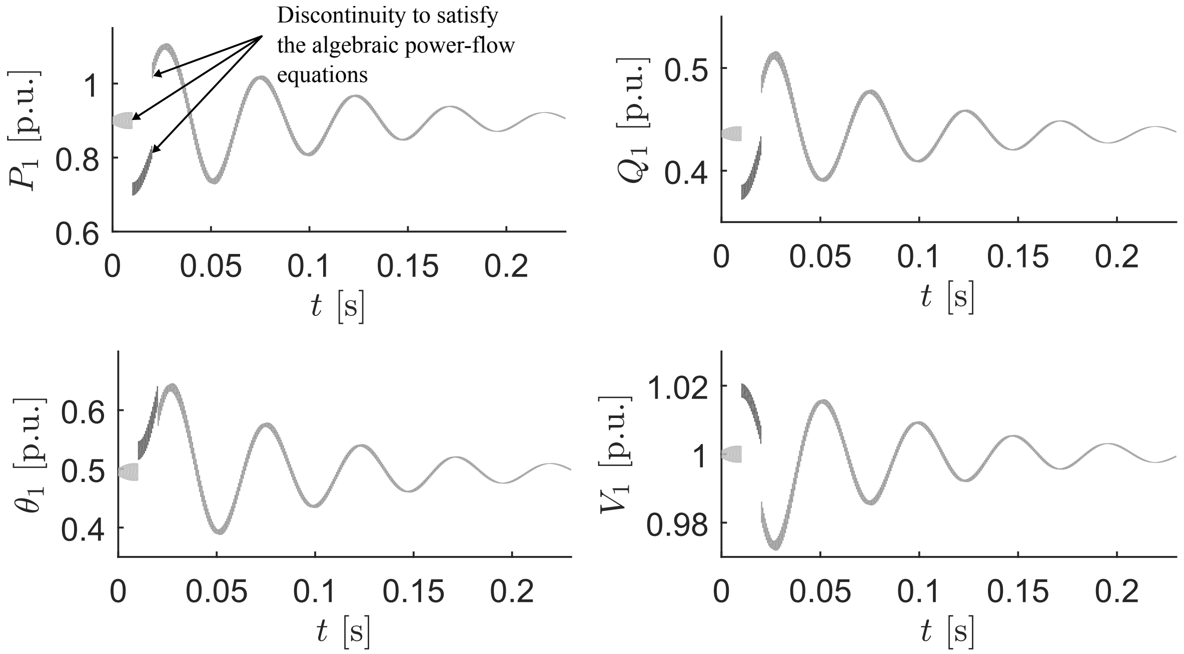
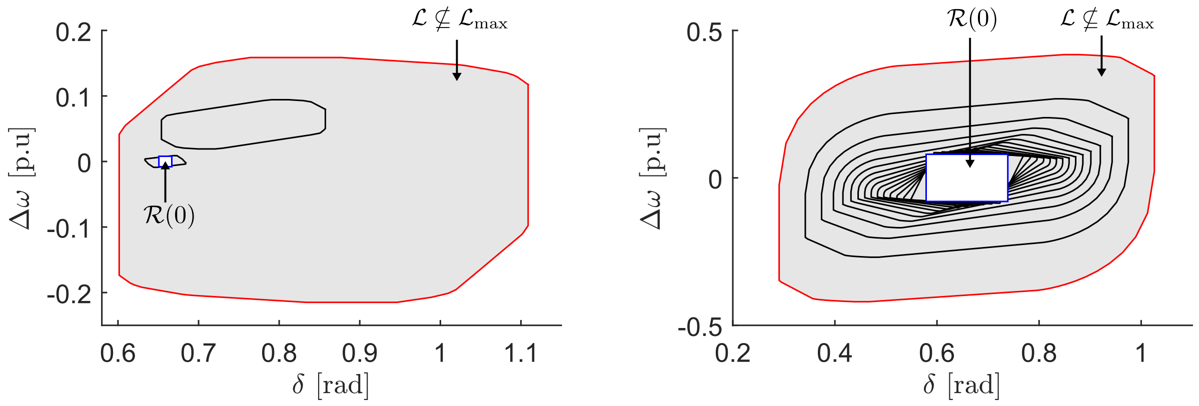


Figure 2.16: Time-domain bounds of the algebraic constraints of the SMIB power system. Here the variables P, Q, V, θ changes instantly at $t = 0.01$ and $t = 0.02$ in order to satisfy the power flow equations (2.71). This leads to the discontinuity of the reachable set.

Example 2.5. In Fig. 2.17(b), we increased $t_r = 0.0006\,s$ to $t_r = 0.006\,s$. Notice how t_r affects the over-approximation of the particular solution, according to Prop. 2.2, associated with the LDI (2.47). Clearly, the larger the time-increment, the more conservative the particular solution becomes and convergence of the reachable of the solution is no longer guaranteed as seen after several time-steps in Fig. 2.17(b).

Remark 2.15. The outcomes of reachability analysis using the CORA toolbox heavily relies on the chosen parameters for the analysis. Improper choice of parameters can result in an unacceptable over-approximation although reasonable results could be achieved by using appropriate parameters as illustrated in the previous examples. Currently a self-tuning algorithm which specifies the optimal parameters for reachability analysis is being investigated as part of future work at the institute.



(a)

(b)

Figure 2.17: Influence of the algorithm parameters on computation of the reachable sets. In (a) the set of linearization errors is computed according to the conservative method proposed in (2.69). In (b) the time-increment t_r is slightly increased resulting in unacceptable over-approximation of the reachable set.

2.5.2 IEEE 9-bus Benchmark Problem

The second system to consider is the IEEE 9-bus test case as illustrated in Fig. 2.18. The system represents a simple approximation of the Western System Coordinating Council to an equivalent system with 9 buses and three generators comprising of six differential variables and 35 algebraic constraints. The system has three PQ-loads, which are modelled during transients as in (2.24).

Example 2.6. Similarly to the SMIB benchmark example, the considered fault scenario is the loss of the transmission line between bus 5 and 7 at $t = 0.01\,s$, followed by its reconnection to the network after clearance of the fault at $t = 0.05\,s$. The reachable set is computed according to Alg. 3 starting from the initial states $\forall i \in \{1, 2, 3\} : \delta_i(0) = \delta_i^0 \oplus [0.005, 0.005]$, $\omega_i(0) \in \omega_i^0 \oplus [0.005, 0.005]$, with the superscript 0 corresponding to the steady-sate solution of (2.70). The reachability algorithm runs until all states are enclosed by $\boldsymbol{\mathcal{R}}(0)$, which occurs at $t = 0.33$.

Fig. 2.19 shows projections of the reachable set for selected state variables. The algorithm requires roughly 20 min of CPU time to compute the reachable set and establish transient stability. It might seem at first that the proposed reachability algorithm requires enormous computational efforts compared to numerical simulations. This holds when considering a single simulation using the `ODE`-15s solver which takes approximately 0.35 s to model the IEEE 9-bus in MATLAB. However, notice that we consider a set of states, associated with the uncertainty arising from renewable resources affecting the initial operating point. If one considers the simulation of all corner cases, this leads to 2^n simulations (exponential problem), where $n = 35$ is the number of states of the 9-bus benchmark problem. This will consume roughly 3.3405E6 hour of computational time. Hence, it is clear that the reachability algorithm can be faster than sufficiently many numerical simulations.

2.6 Summary

In this chapter we have introduced the reader to reachability analysis and its application to power systems with regards to studies involving transient stability analysis. First, we presented the power systems models formalized as a set of nonlinear DAEs, then we briefly described the existing techniques to compute continuous reachable sets, in addition to the typical set representation employed in the analysis of power systems. Afterwards, we highlighted the many advantages of zonotopic set membership to compute reachable sets.

The presented reachability algorithm is based on abstracting the DAE system into LDIs for consecutive time-intervals. This is also of interest to other analysis techniques in power systems, such as small-signal stability, or for analyzing linear models if the linearization can be justified. As seen throughout the numerical examples, the computation of reachable sets makes it possible to rigorously establish transient stability of power systems subject to varying operating conditions. Notice that we considered standard power systems modelled via a set of semi-explicit nonlinear DAEs without requiring some simplifications to bring the set of DAE to an explicit set of ODEs similarly to alternative techniques, such as Lyapunov direct methods, to establish transient stability with formal guarantees.

In the following chapter, we present a compositional technique to improve scalability and drastically reduce the computational efforts associated with the computation of reachable sets for the class of power systems.

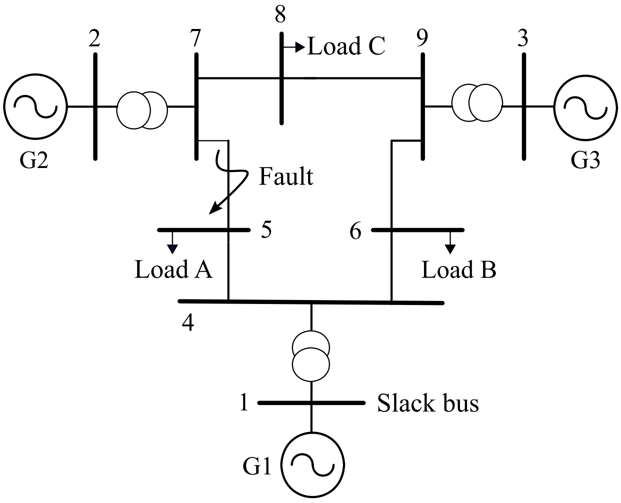


Figure 2.18: Network topology of the IEEE 3-machine 9-bus benchmark [14, Ch. 2]. Here bus 1 is chosen as the slack bus, see remark 2.1.

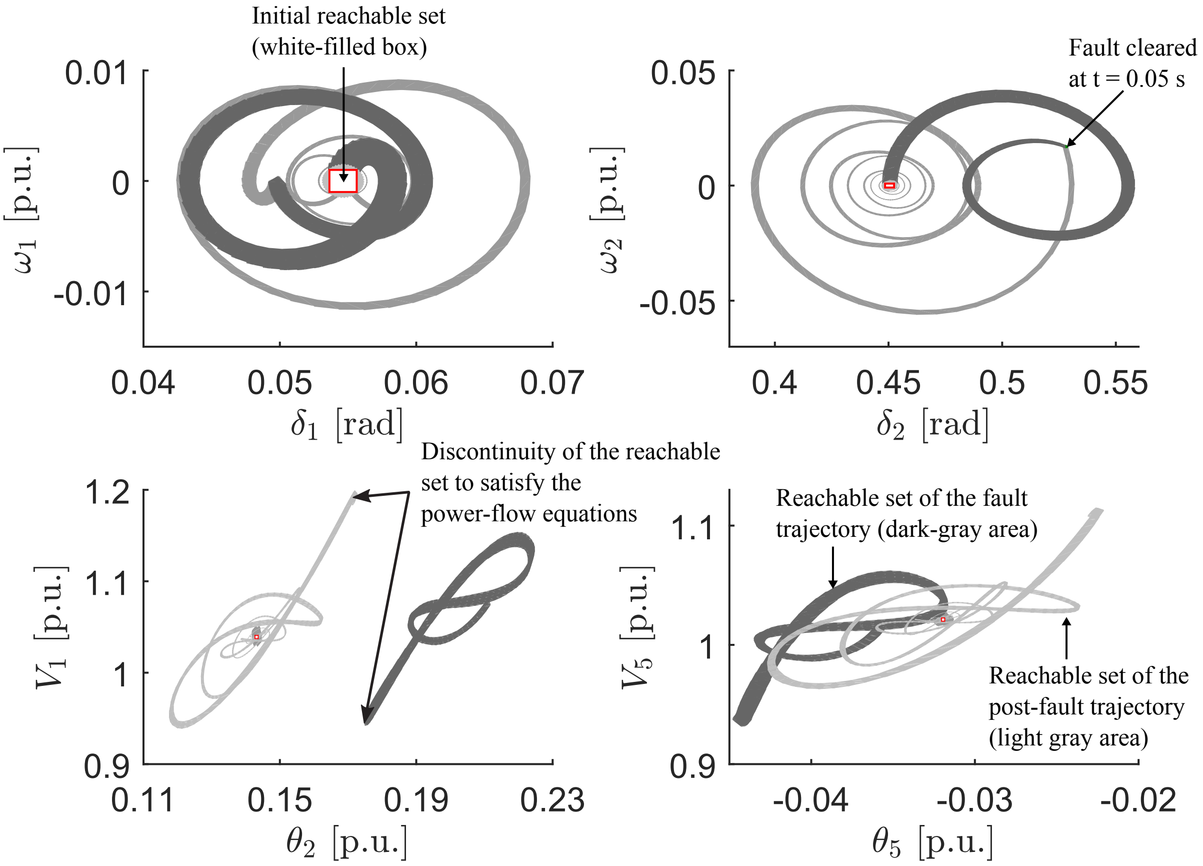


Figure 2.19: Reachable set projection of chosen state variables of the IEEE 9-bus power system. The dark gray area show the reachable set during the fault, whereas the gray area specifies the reachable set of the post-fault. The considered fault scenario is the loss of the one of the transmission lines connecting buses 5 and 7. The line is reconnected after the clearance of the fault, and the reachable set is computed until all states are enclosed by the initial set of states $\boldsymbol{\mathcal{R}}(0)$.

Chapter 3

Compositional Reachability Analysis

The main challenge associated with the analysis of power systems via the computation of reachable sets is improving the algorithmic efficiency to scale towards industrially relevant problem sizes. In this chapter, we present a compositional procedure that can drastically reduce the computational effort required to assess the dynamical response of power systems using reachability analysis. The main reason for the algorithmic efficiency is that we reformulate the transmission network into a set of subsystems, each consisting of a generating unit connected to its corresponding generator bus, whose algebraic constraints are unknown-but-bounded within some confidence intervals. This makes it possible to parallelize the computation of reachable sets for transient stability analysis and, more importantly, preserve the interaction and the correlation between different machines connected to the grid. The applicability of the proposed compositional algorithm is illustrated on several benchmark examples, e.g. the IEEE 6-machines 30-bus benchmark. Furthermore, the method's algorithmic efficiency is compared to the algorithm presented earlier in Ch. 2, in which the reachable set is computed without employing any compositional techniques.

3.1 Introduction

Recently, reachability analysis has emerged as an alternative and promising technique for the analysis of power systems. Basically, reachability analysis makes it possible to compute bounds of all system trajectories, starting from a set of unknown initial states, while simultaneously considering the influence of parametric and input uncertainties. These uncertainties are typically associated with various fault scenarios, loading conditions, and renewable resources that are continuously integrated into the grid. The applicability of reachability analysis in power systems has been reported in a wide range of applications. Shortly after, a review of related literature is presented.

3.1.1 State-of-the-art

The assessment of the static and dynamic performance under load and wind variability is addressed in [30, 39, 69, 136]; specifically [39, 136] employ multi-dimensional intervals to obtain bounds on the solution of the power flow for uncertain loading conditions. As stated earlier, interval arithmetics often result in conservative solutions, hence a more tight set enclosure using ellipsoids and zonotopes is suggested in [30, 69]. In [43, 110] the safety verification of generating units employed in conventional and renewable-based power plants is reported; in particular [43] considers the verification of the steam-drum unit to maximize the load-following capabilities of a combined cycle power plant, whereas [110] focused on the specifications of voltage ride-through of wind turbines in order to meet the grid codes imposed by the corresponding transmission system operator (TSO). The verification of power-converters under varying operating conditions is described in [66, 67, 72, 109]; power converters are becoming heavily present in current power systems due to the continuous integration of renewable-based resources and distributed generation (DG) units. Reachability-based control synthesis is applied in power systems as shown in [47, 74]; namely [74] employs backward reachability computations to synthesize an optimal controller for a high-voltage DC link spanned in a two-area power system, and [47] design a set of linear-parameter varying (LPV) controllers to robustly establish transient stability with formal guarantees for multi-machine power systems. The stability regions obtained using Lyapunov methods can also be estimated using reachability analysis as presented in [44, 70]. In [49, 100, 133] robust methodologies are developed based on computation of backward reachable sets to destabilize a system in the case of a cyber attack in the closed-loop of automatic generation control (AGC). This policy makes it possible to identify potential fake signals which can be caused by an attacker to enforce an undesirable behaviour of the AGC signals. Recall from Ch. 1 that AGC is the standard controller to ensure frequency stability of power systems; hence, the method is potentially beneficial for practical applications.

Transient stability analysis, the main focus of this chapter, was investigated as well using computation of reachable sets in [8, 10, 36, 71, 127, 128]. Transient stability dates back to the 1920s [126] and is widely recognized technically and historically among theorists and practitioners alike as the most problematic issue when considering the dynamic security assessment of power systems [82]. Simply put, transient stability refers to the ability of synchronous generators to remain in synchronism with the frequency of the utility grid following the event of a large disturbance in the transmission network [14, 81]. Early contributions applying reachability analysis for this class of problems in power systems were reported in [36, 70, 71]. These algorithms rely on an Eulerian scheme that employs level set methods (LSMs) to compute backward reachable sets starting from a target set. This is achieved via the formulation of a Hamilton-Jacobi-Isaacs (HJI) partial differential equations (PDEs), where it is proven that the viscosity solution of the time-dependent HJI PDEs provides an implicit surface representation of the continuous backward reachable set [99]. This makes it possible to estimate a region of attraction from which one can identify the initial states of the post-fault scenario that converge back to the equilibrium. Along the same lines, the contributions presented in [127, 128] formulate power systems as a hybrid automaton to compute forward reachable sets using techniques based on level set methods for hybrid systems. The main drawback of this class of techniques, however, is that the computational requirements grow rapidly with the system dimension due to the fact that no analytical solution exists for the set of PDEs based on the HJI formulation. Thus, the state space has to be continuously discretized, resulting in an exponential complexity with respect to the number of state variables. This limited the applicability of LSMs to the single-machine infinite bus (SMIB) and the double-machine infinite bus benchmark problems, in which only a maximum of five state variables have been reported. Another limitation of the level set method is that it only provides an accurate approximation of the reachable set, rather than a rigorous enclosure of it; hence, it does not hold as a formal technique to establish transient stability in practice.

The alternative class of methods for reachability computation is based on Lagrangian techniques, which compute reachable sets similar to numerical integration methods. This is achieved by propagating the set of reachable states instead of only computing the solution for a single point in time, see Fig. 2.8. Although there exists a large variety of well-developed methods that consider nonlinear systems of ordinary differential equations (ODEs), such as abstraction via local linearization [13, 56] or Taylor models [29], there is, however, little work regarding an efficient algorithmic procedure for computation of reachable sets with rigourous bounds for power systems described via the standard formalization using DAEs. An obvious reason is that an extension of reachability algorithms, based on Lagrangian schemes for ODEs, to handle DAEs is necessary. This task, however, is not straightforward since the class of DAE systems differs in both theoretical and numerical properties [37]. So far the majority of results reported in literature perform two workarounds in order to exploit the efficient methods developed for the class of ODEs: they either perform a local linearization around a stable equilibrium point in order to bring the system to a set of explicit ODEs via due to its index-1 property, or they make several unrealistic assumptions, similarly to Lyapunov approaches, to eliminate the algebraic equations resulting from the power-flow formulation, see (2.8). As previously mentioned, it is possible to eliminate the algebraic equations (power-flow) by neglecting the transfer conductance within the transmission network. In other words, by representing the transfer impedance of the transmission line purely via a series reactance without considering the effect of the series line resistance. Furthermore, one has to assume constant field voltage by the exciter and employ the swing equation to model dynamics of the synchronous generator, see Remark 2.13.

The work presented in [12], and recapitulated in Ch. 2, proposed a numerical procedure to compute reachable sets for the class of power systems formulated via a set of DAEs. Although the proposed reachability algorithm has a polynomial complexity $\mathcal{O}(n^5)$, with n corresponding to the number of state variables, the computational requirements were enormous to establish transient stability of the IEEE 30-bus benchmark [12]. In order to improve the scalability and algorithmic efficiency of the proposed procedure, compositional techniques were investigated by the same author in [8]; this is achieved by decomposing the full system into several subsystems via either coherency-based or graph-based decomposition [19, 87, 139]. The CPU time was certainly improved for the same benchmark problem examined in [8] put in comparison against [12]; however, the overall computational requirements were still relatively tremendous. Furthermore, the approach assumes a reasonable partitioning scheme without providing guidelines for choosing specific nodes prior to decomposing the system. Note that this task still remains an open question in the power systems literature. This is based on the fact that the decomposition at certain nodes within a power system must preserve the correlation between different machines connected to the grid; a task known to be extremely challenging and non-trivial.

3.1.2 Contributions and Organization

The main contribution of this chapter is a compositional algorithm to rigorously enclose reachable sets for power systems modelled by standard DAEs, whose computational and associated memory requirements grow moderately with the system dimension in comparison with existing methods reported previously in [8, 12], see Sec. 2.4. The proposed numerical procedure fully considers the set of DAEs governing dynamics of power systems without requiring any modelling simplifications; in other words, without neglecting transfer conductances within the transmission network to eliminate the algebraic variables. Note that this is a *common unrealistic* modelling simplification, often considered in the power system community, to overcome the limitation imposed on techniques based on Lyapunov stability theory [59].

The reason behind the algorithmic efficiency is that we abstract the transmission network via a set of subsystems, each consisting of a generating unit connected to a generator bus, whose algebraic constraints are unknown-but-bounded within some confidence intervals. This makes it possible to parallelize the computation of reachable sets, thus drastically reducing the computational efforts, and most importantly, preserving the interaction and correlation between different machines connected to the grid.

3.2 Problem Formulation and Objective

We consider power systems described by standard models formalized by a set of time-invariant, nonlinear, semi-explicit, index-1 DAEs as in (2.38). Our objective is to assess the stability of power systems during transients by computing the reachable set of the dynamic states variables of (2.38), over a time-horizon $t \in [0, t_f]$, starting from a set of consistent initial states $\boldsymbol{\mathcal{R}}(0)$ and a set of possible inputs $\boldsymbol{\mathcal{U}}$.

Due to the fact that power systems contains hundreds of states variables, the computational efforts associated with existing reachability algorithms for DAE systems are typically enormous, even thought they have a polynomial complexity with respect to the system dimension; hence, we propose a decomposition of the system as illustrated in Fig. 3.1, in order to compute reachable sets compositionally, which in turn can substantially reduce the computational requirements.

First, we assume that the DAE system (2.38) contains n_g buses associated with the buses connected to a generating unit, such as for example a conventional power plant with synchronous generators or a wind

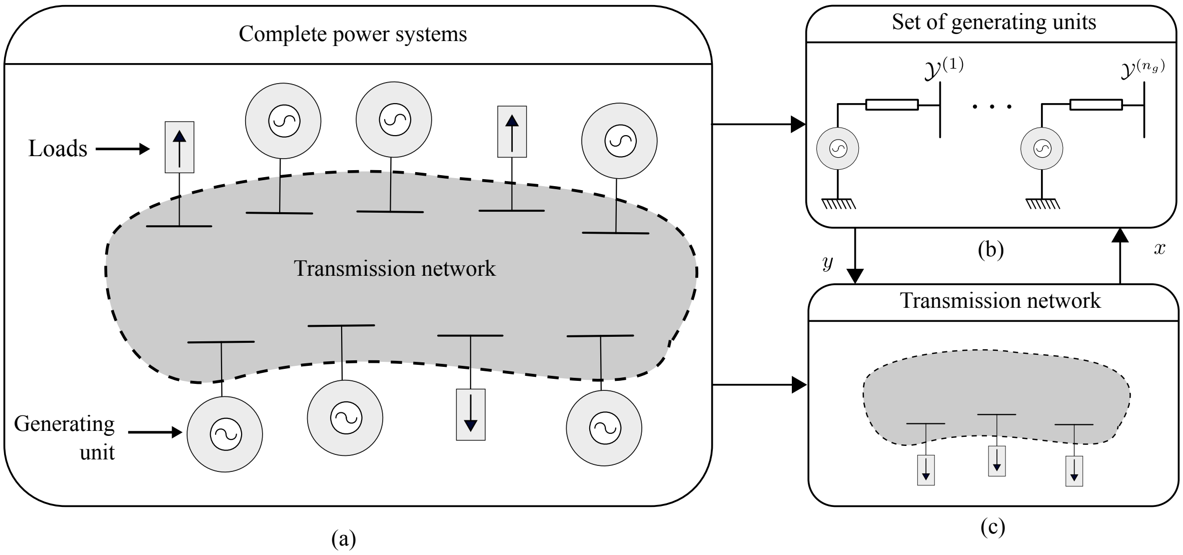


Figure 3.1: Illustration of the compositional approach. In (a) the complete power system is modelled using the set of DAEs as described in (2.38). This system is abstracted by the compositional model (b) as proposed in (3.1), in addition to the transmission network (c), which solves a set of nonlinear algebraic equations associated with the power-flow formulation (2.8). The interaction between different machines is preserved since the algebraic constraints corresponding to each generator bus are not kept constant, but rather are known to vary within some confidence intervals.

farm with induction (asynchronous) motors. Moreover, it is assumed that the algebraic variables at the n_g buses are *unknown-but-bounded*; that is, their values are known to lie within some confidence intervals around some nominal values. These assumptions allow the abstraction of the DAE model (2.38) into

$$
\begin{aligned}
\dot{\boldsymbol{x}}^{(1)}(t) &= \mathcal{G}^{(1)}\left(\boldsymbol{x}^{(1)}(t), \boldsymbol{y}^{(1)}(t), \boldsymbol{u}^{(1)}(t)\right), & \boldsymbol{y}^{(1)}(t) &\in \boldsymbol{\mathcal{Y}}^{(1)}, & \boldsymbol{u}^{(1)}(t) &\in \boldsymbol{\mathcal{U}}^{(1)},\\
&\vdots\\
\dot{\boldsymbol{x}}^{(n_g)}(t) &= \mathcal{G}^{(n_g)}\left(\boldsymbol{x}^{(n_g)}(t), \boldsymbol{y}^{(n_g)}(t), \boldsymbol{u}^{(n_g)}(t)\right), & \boldsymbol{y}^{(n_g)}(t) &\in \boldsymbol{\mathcal{Y}}^{(n_g)}, & \boldsymbol{u}^{(n_g)}(t) &\in \boldsymbol{\mathcal{U}}^{(n_g)},
\end{aligned}
\tag{3.1}
$$

with $\mathcal{G}^{(i)} : \mathbb{R}^{n_x+n_y+n_u} \to \mathbb{R}^{n_x}$ and the vectors $\boldsymbol{x}^{(i)} \in \mathbb{R}^{n_x}$, $\boldsymbol{y}^{(i)} \in \mathbb{R}^{n_y}$, and $\boldsymbol{u}^{(i)} \in \mathbb{R}^{n_u}$ denoting the dynamic state variables, the algebraic variables, and the uncertain inputs, respectively, at the i-th bus. The unknown-but-bounded sets $\boldsymbol{\mathcal{Y}}^{(i)}$ and $\boldsymbol{\mathcal{U}}^{(i)}$ enclose the time variation of the vectors $\boldsymbol{y}^{(i)}$ and $\boldsymbol{u}^{(i)}$, correspondingly.

Remark 3.1. It should be stressed that (3.1) is an abstraction, and it is not considered a simplification of the original DAE system (2.38). While, this simple abstraction may not seem substantial at first, it has several advantages:

1. The reachable set computation of the state variables for each i-th subsystem can be parallelized.
2. The verification of each i-th subsystem is performed separately, which is a much easier task compared to the verification of the complete power system (2.38).

3. Most importantly, the correlation between all machines connected to the grid is still preserved. This is due to the fact that the network interactions are included within the set of uncertainty $\boldsymbol{\mathcal{Y}}^{(i)}$ arising from the algebraic variables associated with the n_g generator buses.

With the abstraction of the semi-explicit DAE system into the compositional model (3.1) described via a set of explicit ODEs, we may apply reachability analysis in a compositional fashion; that is, each i-th subsystem can be handled separately and the reachable set of the complete power system (2.38) is obtained by aggregating the reachable set of the compositional model subsystems to

$$\boldsymbol{\mathcal{R}}([0,t_f]) := \boldsymbol{\mathcal{R}}^{(1)}([0,t_f]) \times \cdots \times \boldsymbol{\mathcal{R}}^{(n_g)}([0,t_f]),$$

with $\boldsymbol{\mathcal{R}}^{(i)}$ being the reachable set of the i-th machine.

The remainder of this chapter is organized as follows: The proposed compositional algorithm is described in Sec. 3.3 and in Sec. 3.4 we illustrate the algorithm's applicability on various benchmark examples. Finally we conclude the chapter in Sec. 3.5.

3.3 Compositional Algorithm

3.3.1 Local Linearization

Similarly to Ch. 2, the reachability algorithm is based on abstracting the differential equations of the compositional model (3.1) into linear differential inclusions (LDIs) for consecutive time interval $\tau_k := [t_k, t_{k+1}]$, with $t_k := k \cdot t_r$ such that $t_r \in \mathbb{R}^+$ corresponds to the time increment and $k \in \mathbb{N}$ being the time step. After introducing the vector $\boldsymbol{z}^{(i)} := \left(\boldsymbol{x}^{(i)^T}\, \boldsymbol{y}^{(i)^T}\, \boldsymbol{u}^{(i)^T}\right)^T \in \mathbb{R}^{n_z}$ and the linearization point $\boldsymbol{z}_k^{(i)} := \left(\boldsymbol{x}_k^{(i)^T}\, \boldsymbol{y}_k^{(i)^T}\, \boldsymbol{u}_k^{(i)^T}\right)^T$, one can express the inclusion of the i-th generator using a first order Taylor expansion with the Lagrangian remainder, see Prop. 2.1:

$$\forall \tau_k \in [t_k,\, t_{k+1}]: \; \dot{\boldsymbol{x}}^{(i)} \in \underbrace{\sum_{j=1}^{n_x} \frac{\partial \mathcal{G}^{(i)}(\boldsymbol{z}^{(i)})}{\partial x_j^{(i)}} \bigg|_{\boldsymbol{z}^{(i)}=\boldsymbol{z}_k^{(i)}} \Delta x_j^{(i)}}_{=: \boldsymbol{A}_k^{(i)} \Delta \boldsymbol{x}^{(i)}} \oplus \tilde{\boldsymbol{\mathcal{U}}}^{(i)}(\tau_k) \tag{3.2}$$

with $\Delta \boldsymbol{x}^{(i)} := \boldsymbol{x}^{(i)} - \boldsymbol{x}_k^{(i)}$. Here $\boldsymbol{A}_k^{(i)} \in \mathbb{R}^{n_x \times n_x}$ is the system matrix of the i-th machine at the time step k, and $\tilde{\boldsymbol{\mathcal{U}}}^{(i)}$ is the set of uncertain inputs expressed by

$$\begin{aligned}\tilde{\boldsymbol{\mathcal{U}}}^{(i)}(\tau_k) := \Bigg\{ \tilde{\boldsymbol{u}}^{(i)} \in \mathbb{R}^{n_x} \,:\, \tilde{u}_p^{(i)} = \mathcal{G}_p^{(i)}(\boldsymbol{z}_k^{(i)}) \oplus \mathcal{L}_p^{(i)}(\tau_k) \oplus \sum_{j=1}^{n_y} \frac{\partial \mathcal{G}_p^{(i)}(\boldsymbol{z})}{\partial y_j^{(i)}} \bigg|_{\boldsymbol{z}=\boldsymbol{z}_k^{(i)}} (y_j^{(i)} - y_{j,k}^{(i)}) \\ \oplus \sum_{j=1}^{n_u} \frac{\partial \mathcal{G}_p^{(i)}(\boldsymbol{z})}{\partial u_j^{(i)}} \bigg|_{\boldsymbol{z}=\boldsymbol{z}_k^{(i)}} (u_j^{(i)} - u_{j,k}^{(i)}),\, \boldsymbol{y}_k^{(i)} \in \boldsymbol{\mathcal{Y}}^{(i)},\; \boldsymbol{u}_k^{(i)} \in \boldsymbol{\mathcal{U}}^{(i)} \Bigg\},\end{aligned} \tag{3.3}$$

where the subscript p corresponds to the p-th coordinate. Here the set of Lagrangian remainders, at the i-th bus, is denoted by $\boldsymbol{\mathcal{L}}^{(i)}(\tau_k)$. This set contains all possible linearization errors within the time interval τ_k, see Prop. 2.1

$$\boldsymbol{\mathcal{L}}^{(i)} := \Bigg\{ \boldsymbol{L}^{(i)} \in \mathbb{R}^{n_x} \, : \, L_p^{(i)} = \frac{1}{2} \sum_{l=1}^{n_z} \sum_{m=1}^{n_z} \frac{\partial^2 \mathcal{G}_p^{(i)}(\boldsymbol{z})}{\partial z_l \partial z_m} \Bigg|_{\boldsymbol{z}=\boldsymbol{\mu}} \Delta z_l^{(i)} \Delta z_m^{(i)}, \\ \boldsymbol{z}^{(i)} \in \boldsymbol{\mathcal{R}}^{(i)}(\tau_k), \, \boldsymbol{\mu} \in \textbf{interval}\left(\boldsymbol{\mathcal{R}}^{(i)}(\tau_k)\right) \Bigg\},$$

with $\Delta \boldsymbol{z}^{(i)} := \boldsymbol{z}^{(i)} - \boldsymbol{z}_k^{(i)}$. Additionally, **interval**$(\,\cdot\,)$ returns the interval hull as in (2.34) and $\boldsymbol{\mathcal{R}}(\tau_k)$ corresponds to the reachable set of the LDI (3.2), which is computed for one time-step as summarized in Alg. 1. For further details, the reader is referred to Sec. 2.4.

3.3.2 Estimating the Set of Algebraic Variables

So far we have presented the computation of the reachable set of the i-th machine under the assumption that the set of uncertain inputs is known in advance. However, this is not the case due to the mutual dependency between the algebraic and differential variable as illustrated in Fig. 3.1.

First, we introduce the vector $\boldsymbol{z} := \left(\boldsymbol{x}^T \, \boldsymbol{y}^T \, \boldsymbol{u}^T\right)^T \in \mathbb{R}^{n_z}$ and the linearization point $\boldsymbol{z}_k := \left(\boldsymbol{x}_k^T \, \boldsymbol{y}_k^T \, \boldsymbol{u}_k^T\right)^T$ for a concise notation, similarly to (3.2). Note that the vectors $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{u} \in \mathbb{R}^o$ corresponds to the variables of the DAE system (2.38). Now we approximate the nonlinear equations of the algebraic variables using a first order Taylor expansion

$$\begin{aligned} \boldsymbol{0} \approx & \underbrace{\sum_{j=1}^{n} \frac{\partial \boldsymbol{g}(\boldsymbol{z})}{\partial x_j} \Bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} x_j}_{=:\, \boldsymbol{J}_k \boldsymbol{x}} + \underbrace{\sum_{j=1}^{m} \frac{\partial \boldsymbol{g}(\boldsymbol{z})}{\partial y_j} \Bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} y_j}_{=:\, \boldsymbol{W}_k \boldsymbol{y}} + \underbrace{\sum_{j=1}^{o} \frac{\partial \boldsymbol{g}(\boldsymbol{z})}{\partial u_j} \Bigg|_{\boldsymbol{z}=\boldsymbol{z}_k} u_j}_{=:\, \boldsymbol{Q}_k \boldsymbol{u}} \\ & + \underbrace{\boldsymbol{g}(\boldsymbol{z}_k) - \boldsymbol{J}_k \boldsymbol{x}_k - \boldsymbol{W}_k \boldsymbol{y}_k - \boldsymbol{Q}_k \boldsymbol{u}_k}_{=:\, \boldsymbol{y}_0}, \end{aligned} \tag{3.4}$$

where $\boldsymbol{J}_k \in \mathbb{R}^{m \times n}$, $\boldsymbol{W}_k \in \mathbb{R}^{m \times m}$, and $\boldsymbol{Q}_k \in \mathbb{R}^{m \times o}$ are the matrices of the nonlinear algebraic equations $\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$ at the linearization point $\boldsymbol{z}_k$.

We shall denote $\tilde{\boldsymbol{y}}_k^*$ as one of the roots of the nonlinear algebraic equations $\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$ at the time step k. The roots can be obtained using a variety of techniques, for example Newton-Raphson's method which is a powerful technique for solving a set of nonlinear equations numerically. Let $\boldsymbol{y}_{k,0}$ specify the initial guess $\tilde{\boldsymbol{y}}_k^*$. From $\boldsymbol{y}_{k,0}$ we can produce an improved estimate $\boldsymbol{y}_{k,1}$, which in turn specify $\boldsymbol{y}_{k,2}$ up to the q-th iteration. This iterative procedure is performed until the estimate is approximately equal to one of the roots; in other words, the iterative procedure stops at the q-th iteration if the following holds

$$\boldsymbol{y}_{k,q} - \boldsymbol{y}_{k,q+1} \leq \boldsymbol{\epsilon} \quad \text{s.t.} \begin{cases} \boldsymbol{y}_{k,q+1} = \boldsymbol{y}_{k,q} - \boldsymbol{\Delta}_y, \\ \quad \boldsymbol{\Delta}_y = \boldsymbol{W}_k^{-1}(\boldsymbol{z}_{k,q}) \boldsymbol{g}(\boldsymbol{z}_{k,q}), \\ \quad \boldsymbol{z}_{k,q} = \left(\boldsymbol{x}_k^T, \, \boldsymbol{y}_{k,q}^T, \, \boldsymbol{u}_k^T\right)^T. \end{cases} \tag{3.5}$$

where $\boldsymbol{\epsilon}$ is chosen to be sufficiently small (e.g. 1E-3) such that convergence to one of the real roots is achieved. In fact, Newton-Raphson's method has a quadratic rate of convergence if the initial estimate $\boldsymbol{y}_{k,0}$ is chosen sufficiently close to one of the real root $\tilde{\boldsymbol{y}}_k^*$. In (3.5) the variable $\boldsymbol{\Delta}_y$ is obtained by solving a system of linear equations which has a one unique solution. Furthermore the solution always exists due the non-singularity of the matrix $\boldsymbol{W}_k$, which is associated with the index-1 property of the power system modelled as in (2.38).

After obtaining a close estimate to one of the real root $\tilde{\boldsymbol{y}}_k^*$, we make an initial guess that bounds the algebraic constraints associated with the i-th subsystem using a multidimensional interval

$$\overline{\boldsymbol{\mathcal{Y}}}^{(i)} = [\underline{\boldsymbol{y}}^{(i)}, \overline{\boldsymbol{y}}^{(i)}], \quad \text{with} \begin{cases} \underline{\boldsymbol{y}}^{(i)} := \boldsymbol{y}_k^{(i),*} - \boldsymbol{\gamma}^{(i)}, \\ \overline{\boldsymbol{y}}^{(i)} := \boldsymbol{y}_k^{(i),*} + \boldsymbol{\gamma}^{(i)}, \end{cases} \tag{3.6}$$

with $\boldsymbol{\gamma} \in \mathbb{R}^m$ as a user-defined enlargement factor and $\boldsymbol{y}_k^*$ is the estimate of the root obtained using (3.5). With the knowledge of the uncertain input set, one can assemble the reachable set as

$$\boldsymbol{\mathcal{R}}(\tau_k) := \boldsymbol{\mathcal{R}}^{(1)}(\tau_k) \times \cdots \times \boldsymbol{\mathcal{R}}^{(n_g)}(\tau_k), \tag{3.7}$$

where the sets $\boldsymbol{\mathcal{R}}^{(i)}(\tau_k)$, $i \in \{1 \ldots n_g\}$ are computed using Alg. 3 according to the numerical procedure presented earlier in Sec. 2.4.

Recall that the matrix $\boldsymbol{W}_k$ is always invertible, thus (3.4) can be rewritten as

$$\boldsymbol{y} \approx -\boldsymbol{W}_k^{-1} \left(\boldsymbol{y}_0 + \boldsymbol{J}_k \boldsymbol{x} + \boldsymbol{Q}_k \boldsymbol{u} \right), \tag{3.8}$$

then by replacing the differential state variables $\boldsymbol{x}$ and the uncertain inputs $\boldsymbol{u}$ by their corresponding set $\boldsymbol{\mathcal{R}}(\tau_k)$ and $\boldsymbol{\mathcal{U}}$ within the time interval τ_k, one can estimate the set of algebraic constraints using

$$\boldsymbol{\mathcal{Y}}(\tau_k) := \Big\{ \boldsymbol{y} \in \mathbb{R}^m \,:\, \boldsymbol{y} = -\boldsymbol{W}_k^{-1} \left(\boldsymbol{y}_0 \oplus \boldsymbol{J}_k \cdot \boldsymbol{\mathcal{R}}(\tau_k) \oplus \boldsymbol{Q}_k \cdot \boldsymbol{\mathcal{U}} \right) \Big\} \tag{3.9}$$

which involves a linear transformation of the set of dynamic state variables and set of uncertain inputs. In the event that $\exists i : \boldsymbol{\mathcal{Y}}^{(i)}(\tau_k) \nsubseteq \overline{\boldsymbol{\mathcal{Y}}}^{(i)}$, one must further enlarge the initial guess $\overline{\boldsymbol{\mathcal{Y}}}^{(i)}$, $\forall i \in \{1, \ldots, n_g\}$ and recompute the reachable set $\boldsymbol{\mathcal{R}}(\tau_k)$.

Remark 3.2. For practical consideration, this iterative procedure stops if

$$\boldsymbol{\mathcal{Y}} \nsubseteq \boldsymbol{\mathcal{Y}}_{\max}, \tag{3.10}$$

with $\boldsymbol{\mathcal{Y}}_{\max}$ being a user-defined set associated with the maximum allowable bounds of the algebraic variables.

3.3.3 Overall Algorithm

The complete procedure to compute reachable sets of power systems modelled by (2.38) using the proposed compositional approach is summarized in Alg. 4. The algorithm consists of two loops: the outer loop computes the reachable set for the specified fault scenario and determines if the reachable set converges back to the equilibrium/stability region. The inner loop parallelizes the computation of the reachable set of the generating units at each time interval τ_k and verifies if the set of algebraic constraints at each of the n_g buses is enclosed by the initial guess.

First in the inner loop, starting from line 5 the local linearization point of differential variables is determined at the time step k. Typically this point is associated with the center of the estimate of the reachable set in the following time step, that is $\boldsymbol{x}_k := \textbf{center}(\boldsymbol{\mathcal{R}}(t_{k+1}))$. Then, using the Newton-Raphson iterative procedure, the root of the algebraic variables $\boldsymbol{y}_k^*$ are obtained, which is specified via the operator $\textbf{newton}(\,\cdot\,)$ as in line 6. With the knowledge of $\boldsymbol{y}_k^*$, one can construct the unknown-but-bounded set for each generator bus via the user-defined variable $\boldsymbol{\gamma}$. Throughout line $8-14$ the computation of the reachable set of each i-th machine is parallelized. The computation of the reachable is similar to the standard procedure in Sec. 2.4, in which the step-wise illustration at each time step is shown in Fig. 2.12.

With the knowledge of the reachable set of differential variables, associated with each machine connected to the grid, one may construct the set of algebraic constraints as in line 15. Afterwards, the inner loop of the algorithm examines in line 17 whether the set $\boldsymbol{\mathcal{Y}}^{(i)}$ at each generator bus is confined within the initial guess, or if there exists a single guess which exceeds the maximum allowable set of algebraic constraints. If the latter case occurs, the algorithm breaks and it is concluded that transient stability of the power system cannot be established, i.e. generators lost synchronism following occurrence of the fault. This condition is examined in line 21 within the outer loop of the algorithm.

Finally, if the estimations are always kept confined within the initial guess, the reachable set is computed until all states converge back to the initial operating condition, i.e. the reachable set becomes a subset of the initial set of differential state variables $\boldsymbol{\mathcal{R}}(0)$. Notice that this is the same termination condition employed in the numerical examples of Sec. 2.5 to establish transient stability using reachability analysis.

3.4 Numerical Examples

This section illustrates the application of the proposed compositional algorithm on several benchmark examples to showcase the applicability and scalability of the algorithm. All computations are performed on a standard computer with an Intel Core i7-4810MQ CPU and 16GB of RAM. The algorithms are realized in MATLAB2016b using the CORA toolbox [9]. We compare the computational time and validate accuracy of the results against the algorithm recapitulated from [12] in the previous chapter throughout Alg. 3.

Algorithm 4 CompositionalReach

```
Require: The initial sets $\boldsymbol{\mathcal{R}}^{(i)}(0)$, $i \in \{1 \dots n_g\}$, $\boldsymbol{\mathcal{Y}}(0)$, the set of uncertain inputs $\boldsymbol{\mathcal{U}}$, the time increment
    $t_r$, number of Taylor terms $\sigma$ of the matrix exponential (see (2.50)), and the enlargement factor $\boldsymbol{\gamma}$
Ensure: $\bigcup_{m=1}^{k} \boldsymbol{\mathcal{R}}(\tau_m)$, isConverging, $t_f$
$k = 0$, $t_k = 0$, $t_{k+1} := t_k + r$, $\tau_k = [t_k, t_{k+1}]$, and isConverging = true    ▷ Initialization
$\boldsymbol{\mathcal{R}}(t_k) \xleftarrow{(3.7)} \boldsymbol{\mathcal{R}}^{(1)}(t_k) \times \cdots \times \boldsymbol{\mathcal{R}}^{(n_g)}(t_k)$    ▷ Aggregation of initial reachable sets
repeat    ▷ Outer loop to compute reachable set
    repeat    ▷ Inner loop to parallelize computation of reachable set
        $\boldsymbol{z}_k \leftarrow \Big(\boldsymbol{\mathcal{R}}(t_{k+1}), \boldsymbol{\mathcal{Y}}(\tau_k), \boldsymbol{\mathcal{U}}(\tau_k)\Big)$    ▷ Local linearization point
        $\boldsymbol{y}_k^* \xleftarrow{(3.5)} \textbf{newton}\,(\boldsymbol{z}_k, \boldsymbol{g}(\boldsymbol{z}_k))$    ▷ Newton-Raphson's method
        $\overline{\boldsymbol{\mathcal{Y}}}(\tau_k) \xleftarrow{(3.6)} \textbf{estimate}(\boldsymbol{y}_k^*, \gamma)$    ▷ Estimation of algebraic variables
        parfor $i \leftarrow 1 \dots n_g$ do    ▷ Parallel computation of reachable sets
            $\boldsymbol{z}_k^{(i)} \leftarrow \Big(\boldsymbol{\mathcal{R}}^{(i)}(t_{k+1}), \overline{\boldsymbol{\mathcal{Y}}}^{(i)}(\tau_k), \boldsymbol{\mathcal{U}}^{(i)}\Big)$    ▷ Local linearization point at the $i$-th subsystem
            $\boldsymbol{A}_k^{(i)} \xleftarrow{(3.2)} \textbf{taylor}\,\Big(\mathcal{G}^{(i)}(\boldsymbol{z}^{(i)}),\, \boldsymbol{z}_k^{(i)}\Big)$    ▷ System matrix of the linear differential inclusion
            $\overline{\widetilde{\boldsymbol{\mathcal{U}}}}^{(i)}(\tau_k) \xleftarrow{(3.3)} \overline{\boldsymbol{\mathcal{Y}}}^{(i)}(\tau_k)$
            $\Big(\boldsymbol{\mathcal{R}}^{(i)}(t_{k+1}), \boldsymbol{\mathcal{R}}^{(i)}(\tau_k)\Big) \xleftarrow{\text{Alg.1}} \textbf{ReachNext}\Big(\boldsymbol{A}_k^{(i)}, t_r, \sigma, \boldsymbol{\mathcal{R}}^{(i)}(t_k), \overline{\widetilde{\boldsymbol{\mathcal{U}}}}^{(i)}\Big)$
            $\boldsymbol{\mathcal{R}}(\tau_k) \leftarrow \boldsymbol{\mathcal{R}}^{(1)}(\tau_k) \times \cdots \times \boldsymbol{\mathcal{R}}^{(n_g)}(\tau_k)$
            $\boldsymbol{\mathcal{R}}(t_{k+1}) \leftarrow \boldsymbol{\mathcal{R}}^{(1)}(t_{k+1}) \times \cdots \times \boldsymbol{\mathcal{R}}^{(n_g)}(t_{k+1})$
        end parfor
        $\boldsymbol{\mathcal{Y}}(\tau_k) \xleftarrow{(3.9)} (\boldsymbol{\mathcal{R}}(\tau_k), \boldsymbol{\mathcal{U}})$    ▷ Linear mapping
        $t_{k+1} := t_k + t_r$, and $k := k + 1$    ▷ Update for next Iteration
    until $\forall i : \boldsymbol{\mathcal{Y}}^{(i)}(\tau_k) \subset \overline{\boldsymbol{\mathcal{Y}}}^{(i)}(\tau_k) \vee \exists i : \boldsymbol{\mathcal{Y}}^{(i)}(\tau_k) \nsubseteq \boldsymbol{\mathcal{Y}}_{\max}^{(i)}$
    if $\exists i : \boldsymbol{\mathcal{Y}}^{(i)}(\tau_k) \nsubseteq \boldsymbol{\mathcal{Y}}_{\max}^{(i)}$ then
        isConverging := false
        break    ▷ Abort reachability computation
    end if
until $\boldsymbol{\mathcal{R}}(t_{k-1}) \subseteq \boldsymbol{\mathcal{R}}(0) \vee$ isConverging = false    ▷ Examine stability
$t_f := t_{k-1}$    ▷ Terminal time
```

Similarly to Ch. 2, we only use the synchronous machine as the generation unit modelled via the swing equation (2.10); hence the mathematical model governing dynamics of the power system under consideration simplifies to the differential equations

$$
\begin{aligned}
\dot{\delta}_j &= \omega_s \left(\omega_j - \omega_{\text{ref}}\right), \\
\dot{\omega}_j &= \frac{1}{2H_j} \left(P_{m,j} - P_{e,j} - D_j(\omega_j - \omega_{\text{ref}})\right), \\
\text{with:} \quad P_{e,j} &= E'_{q,j} V_j Y_j^g \cos(\Theta_j^g + \delta_j - \theta_j) - V_j^2 Y_j^g \cos(\Theta_j^g),
\end{aligned}
$$

and the power-flow equations:

$$
\begin{aligned}
P_i = P_{e,i} - P_{l,i} &= V_i \sum_{k \in \mathcal{N}} V_i Y_{ik} \cos(\Theta_{ik} - \theta_k - \theta_i), \\
Q_i = Q_{e,i} - Q_{l,i}, &= -V_i \sum_{k \in \mathcal{N}} V_i Y_{ik} \sin(\Theta_{ik} - \theta_k - \theta_i).
\end{aligned}
$$

Here the subscripts j and i specify the j-th machine and the i-th bus, respectively. It is assumed that the mechanical power P_m and the generator voltage E_q' are kept constant. The constants Y^g and Θ^g denote to the absolute value and the phase angle of the admittance spanned from the generator to its corresponding bus. All remaining variables and parameters were previously introduced as in Table 2.1 and Sec. 2.1.2.

Shortly after we consider four common benchmark examples: the single-machine infinite bus, the IEEE 3-machine 9-bus, the IEEE 5-machines 14-bus, and the IEEE 6-machines 30-bus.

3.4.1 Single-Machine Infinite Bus

The first system to consider is the SMIB system as illustrated previously in example 2.5.1. Here the fault scenario under consideration is the loss of the second transmission line at $t = 0.01\,s$, followed by its reconnection to the network after clearance of the fault $t = 0.02\,s$. The reachable set is computed according to Alg. 3 starting from the initial set $\boldsymbol{\mathcal{R}}(0) = ([0.65075,\ 0.66675]\ [-0.008, 0.008])^T$. Here we do not specify the terminal time t_f, instead, the reachability algorithm runs until all states are enclosed by $\boldsymbol{\mathcal{R}}(0)$.

Fig. 3.2 illustrates the computed reachable set of the differential variables δ and ω according to the proposed algorithm outlined in Alg. 4. To validate our results, we compare the resulting reachable set to those obtained when considering the full system, i.e., no compositional techniques are applied. The reachability algorithm used in the comparison is described in detail in Alg. 3 and considers as well power systems modelled by the original set of nonlinear DAEs (2.38).

3.4.2 IEEE 3-machine 9-bus

The second system to consider is the IEEE 9-bus test case as illustrated in Fig. 2.18. Similarly to the SMIB benchmark example, the considered fault scenario is the loss of the transmission line between bus 5 and 7 at $t = 0.01\,s$, followed by its reconnection to the network after clearance of the fault $t = 0.05\,s$. The reachable set is computed according to Alg. 3 starting from the initial states $\forall i \in \{1, 2, 3\} : \delta_i(0) = \delta_i^0 \oplus [0.005,\ 0.005]$, $\omega_i(0) \in \omega_i^0 \oplus [0.005,\ 0.005]$, with the superscript 0 corresponding to the steady-state solution of (2.70). The reachability algorithm runs until all states are enclosed by $\boldsymbol{\mathcal{R}}(0)$.

Fig. 3.3 shows projections of the reachable set for the differential variables of each generating unit, in addition to the reachable set of algebraic variables at the buses associated with these units. Here we validate accuracy of the results by simulating random trajectories using the `ODE`-15 solver in MATLAB. It can be seen that all trajectories are enclosed by the reachable set computed via the compositional method.

3.4.3 IEEE 5-machine 14-bus

The third system is the IEEE 14-Bus test case whose topology is shown in Fig. 3.4. The system represents a portion of the electric power system of the midwestern US which comprises of five synchronous machines and 11 PQ-loads. The toolbox MATPOWER[1] [142] is used to obtain the solution of the power-flow equations (2.71), which in turn specifies consistent initial states of the system.

The considered fault scenario is the loss of the transmission line connecting buses 2 and 4. Similarly to the previous examples, the reachability algorithm runs until all states are enclosed by $\boldsymbol{\mathcal{R}}(0)$. Fig. 3.5 shows projections of the reachable set for the differential variables of each of the 5 generating units. We validate accuracy of the results by simulating random trajectories using the `ODE`-15s solver. Once again, it can be seen that all trajectories are enclosed by the reachable set computed via the compositional method.

3.4.4 IEEE 6-machine 30-bus

The final system is the IEEE 6-machine 30-bus whose network topology is illustrated in Fig. 3.6. The system consists of 131 state variables: 12 differential variables (two per machine) and 119 algebraic variables (four per bus with the exception of the slack bus)[2].

Here we considered two faults (one at a time) to showcase the ability of the algorithm to establish transient stability for different fault scenarios. Fig. 3.7 shows projections of the reachable set for the differential variables when the fault occurs at the transmission line connecting buses 4 and 6. Fig. 3.8 considers occurrence of the fault between the buses 29 and 30. The accuracy of the results is validated by simulating random trajectories starting from the initial set of states using the `ODE`-15s solver.

3.4.5 Discussion

It can be seen that our proposed compositional algorithm provides fairly accurate results compared to the reachable set computed for the exact DAE system (2.38), see Fig. 3.2. Furthermore, the nonlinear trajectories of the dynamic and algebraic variables of the simulated DAE system are tightly enclosed by the reachable set. The computational times for the benchmark examples are listed in Table 3.1. It is obvious that the computational resources are drastically reduced when computing the reachable set

[1] `http://www.pserc.cornell.edu/matpower/`

[2] θ_1 is not considered as a state variable as its corresponding bus is chosen as the slack bus, see remark 2.1.

in a compositional way starting from the IEEE 3-machine 9-bus system; however, the CPU time when computing the reachable set of the SMIB system is comparable, due to the simplicity of the system.

Our proposed algorithm, however, introduces some conservatism, which can be considered a tradeoff between accuracy and efficiency. This conservatism results from the uncertainty of the input set associated with the algebraic constraints at the n_g generator buses. This leads to further over-approximation of the reachable set since we consider all possible values taken by the bus voltage V_i and phase angle θ_i, even the unrealistic values. It should be noted that the conservatism does not affect the security assessment during transient response; however, it can degrade the performance of the system, if the computed reachable set intersects with safety limits, e.g., the bus voltage exceeding limits defined by the grid operators. Reducing and even eliminating the resulting conservatism is an interesting issue which can be further investigated in a future work.

Table 3.1: Comparison of the CPU time for different benchmark examples.

Benchmark example	Computational time		State variables	
	Proposed algorithm	Algorithm in [12]	Dynamic	Algebraic
Single-machine infinite bus	9.78 s	10.14 s	2	4
IEEE 3-machine 9-bus	50.72 s	20 min	6	35
IEEE 5-machine 14-bus	170.42 s	-	10	55
IEEE 6-machine 30-bus	381.97 s	-	12	119

3.5 Summary

We presented a new algorithm for compositional transient stability analysis of power systems via the computation of reachable sets. Using our proposed technique, we drastically reduced the computational time required to compute the reachable set compared to the algorithm recapitulated from [12] in Ch. 2. The main reason for the improved algorithmic efficiency is that we abstract the complete power system into a set of subsystems, each consisting of a generating unit connected to a generator bus, whose algebraic constraints are unknown-but-bounded within some confidence intervals. This makes it possible to parallelize the computation of the reachable set for each generating unit, while, most importantly, preserving the interaction between different machines connected to the grid during faults. The applicability of the algorithm has been illustrated on various benchmark examples, in which we demonstrated the capability of the algorithm to handle more than 100 state variables in a reasonable time for the class of power systems, see Table 3.1. Furthermore, the tradeoff between accuracy of the solution (conservatism) and the algorithmic efficiency is demonstrated by validating the results against alternative techniques which do not employ any compositional techniques.

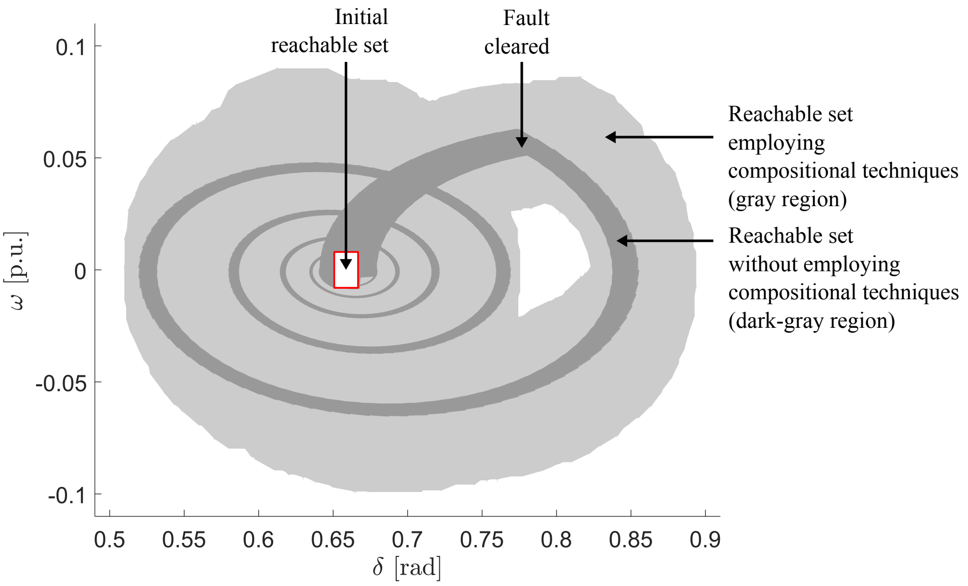


Figure 3.2: Projection of the dynamic state variables of the SMIB system. The projections show a comparison between the reachable set using the proposed compositional technique (gray area), outlined in Alg. 4, and those computed using the algorithm summarized in Alg. 3 (dark gray area). The initial set of the generator dynamic state variables $\mathcal{R}(0)$ is the white box.

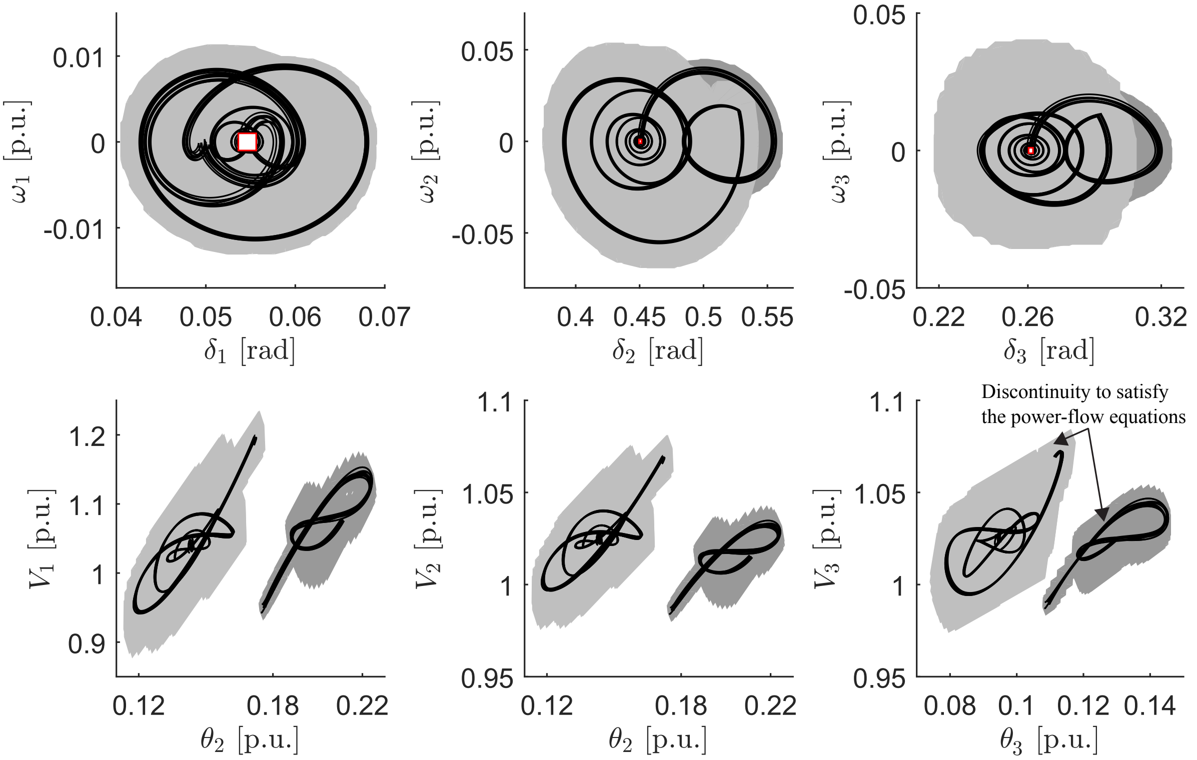


Figure 3.3: Projection of selected differential and algebraic variables for the IEEE 3-machine 9-bus benchmark. The dark gray areas show the reachable set during faults. The considered fault scenario is the loss of the transmission line connecting buses 5 and 7. The line is reconnected after clearance of the fault, and the reachable set is computed until all states are enclosed by the initial reachable set $\mathcal{R}(0)$. The solid lines present random simulations starting from $\mathcal{R}(0)$. The trajectories are obtained using the ODE-15s solver.

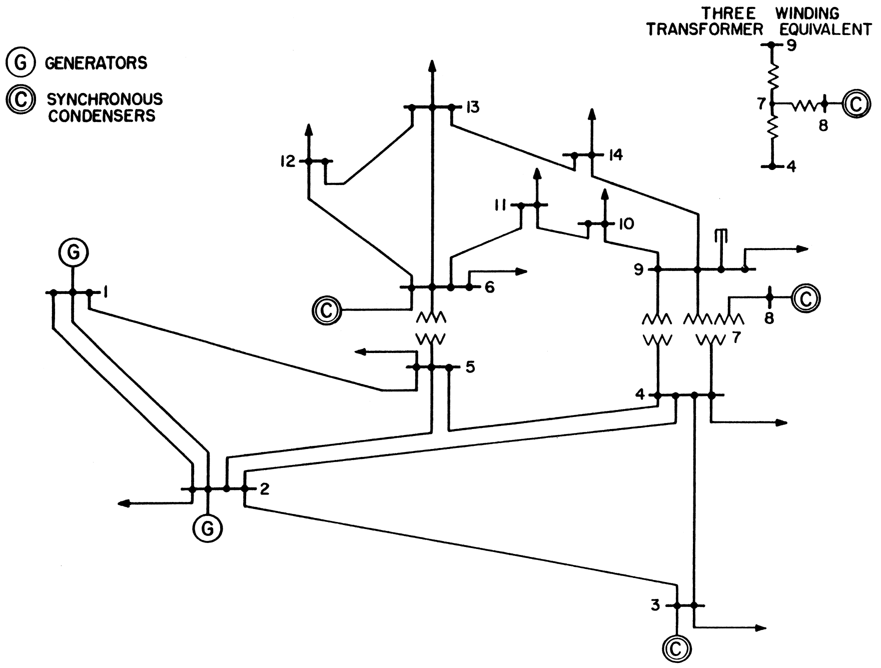
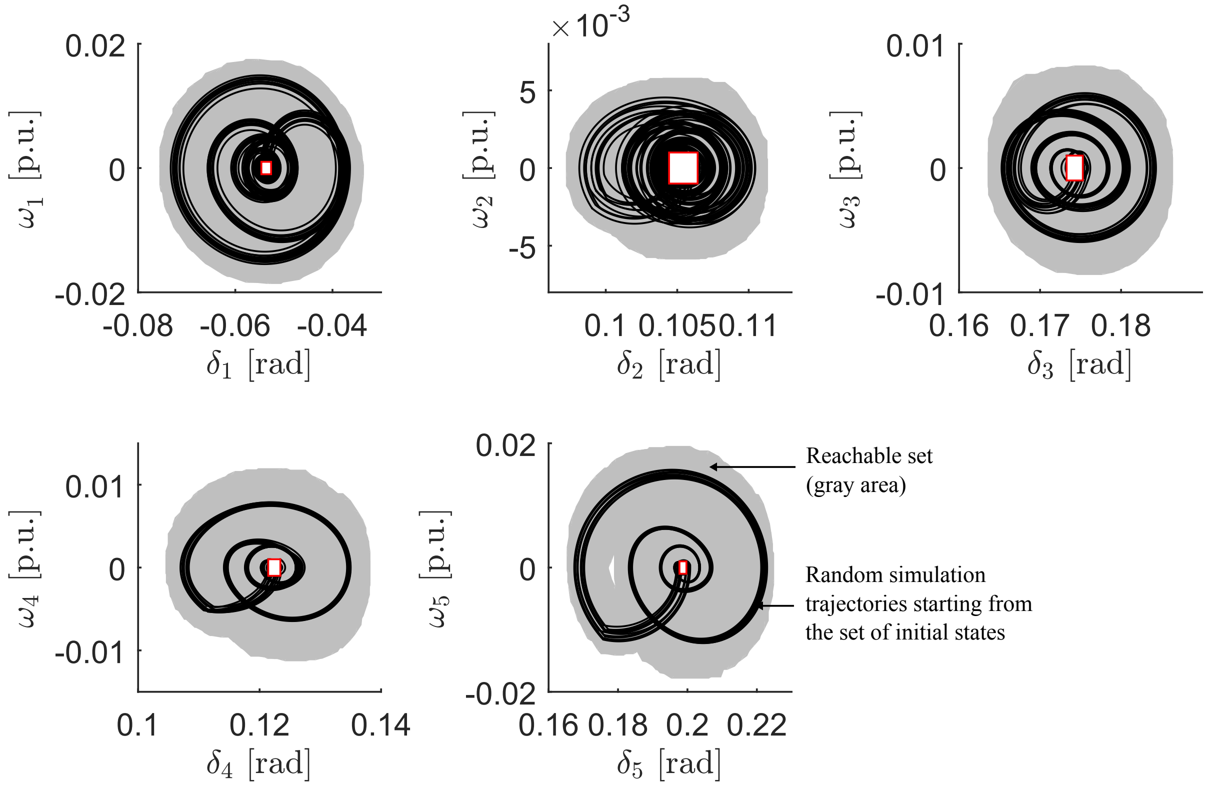


Figure 3.4: Network topology of the IEEE 5-machine 14-bus benchmark problem.



Reachable set (gray area)

Random simulation trajectories starting from the set of initial states

Figure 3.5: Projections of the dynamic state variables of the IEEE 14-bus benchmark. The initial set of the generator dynamic state variables $\mathcal{R}(0)$ is the white box. The considered fault scenario is the loss of the transmission line connecting buses 2 and 4. The line is reconnected after clearance of the fault, and the reachable set is computed until all states are enclosed by $\mathcal{R}(0)$. The solid lines present random simulation results starting from the set the initial states, which are obtained via the `ODE`-15s.

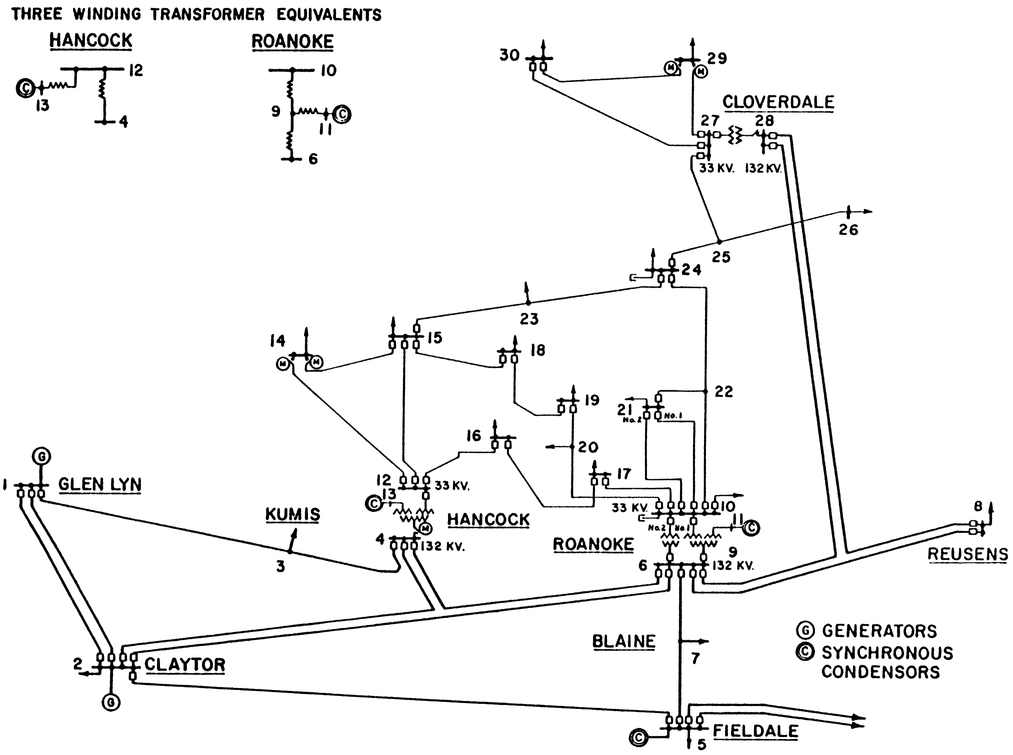


Figure 3.6: Network topology of the IEEE 6-machine 30-bus benchmark problem.

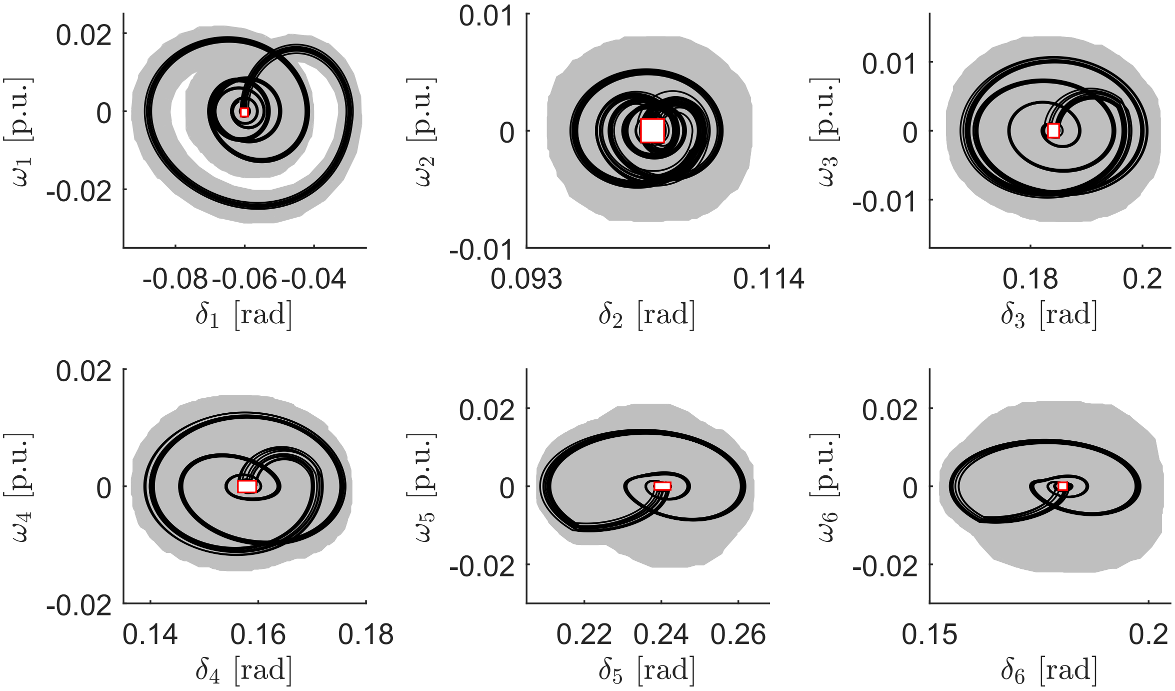


Figure 3.7: Chosen projections of the dynamic state variables of the IEEE 30-bus benchmark. The initial set of the generator dynamic state variables $\mathcal{R}(0)$ is the white box. The considered fault scenario is the loss of the transmission line connecting buses 4 and 6. The line is reconnected after clearance of the fault, and the reachable set is computed until all states are enclosed by $\mathcal{R}(0)$. The solid lines present random simulation results starting from the set of initial states.

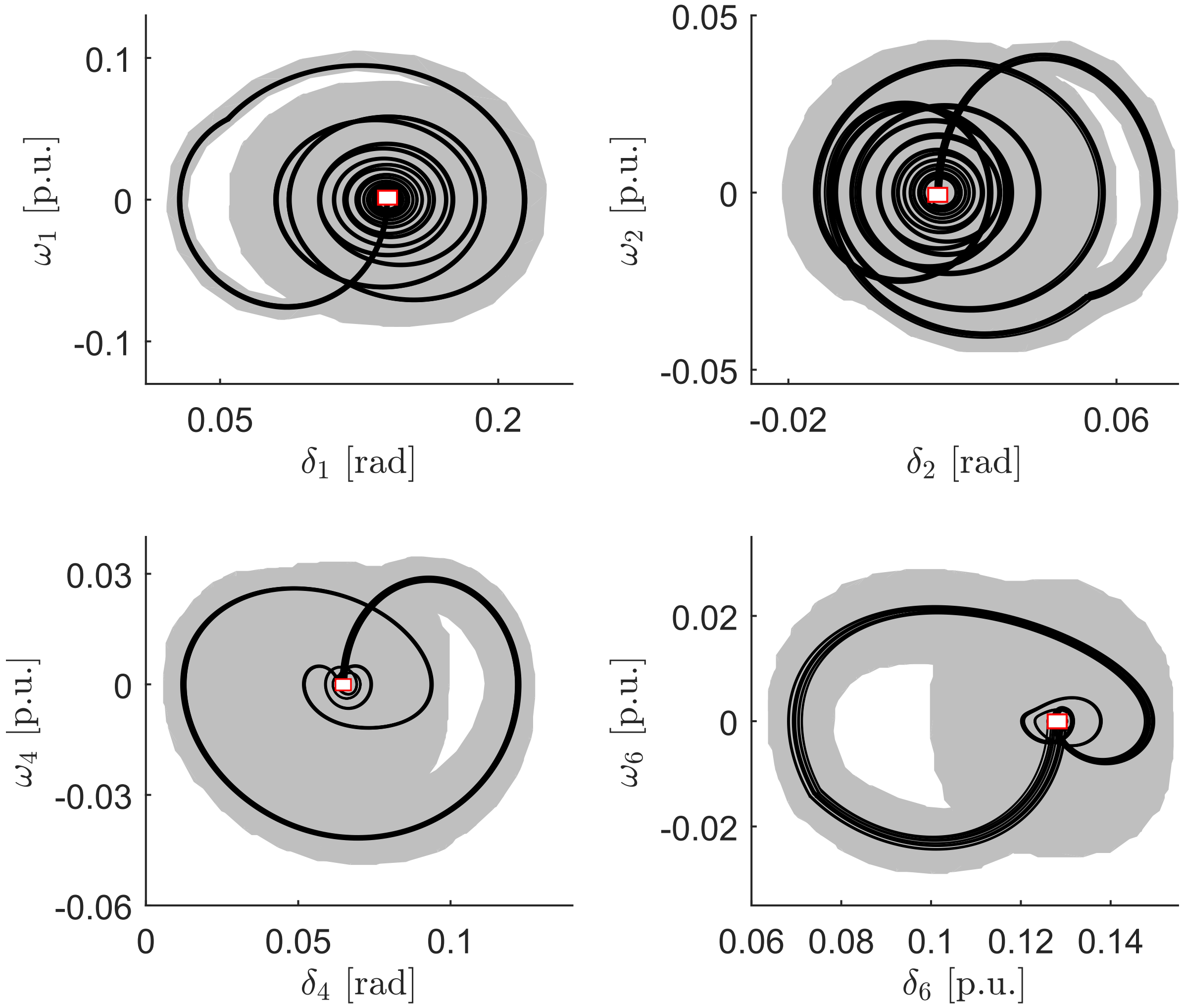


Figure 3.8: Chosen projections of the dynamic state variables of the IEEE 6-machine 30-bus benchmark. The initial set of the generator dynamic state variables $\boldsymbol{\mathcal{R}}(0)$ is the white box. The considered fault scenario is the loss of the transmission line connecting the buses 29 and 30. The line is reconnected after clearance of the fault, and the reachable set (gray area) is computed until all states are enclosed by $\boldsymbol{\mathcal{R}}(0)$. The solid lines present random simulation results starting from the set of initial states. The trajectories are obtained using the `ODE`-15s solver.

Chapter 4

Estimation of the Region of Attraction

This chapter proposes an algorithmic procedure based on reachability analysis to estimate the region of attraction (ROA) of an equilibrium point for nonlinear systems. We compare our results with well established techniques in this area; namely the optimization of the Lyapunov function (LF) sub-level set using sum-of-squares (SOS) decomposition, and the computation of backward reachable sets of a target set using the viscosity solution of a time-dependant Hamilton-Jacobi-Isaacs (HJI) formulation. The proposed method can overcome many limitations imposed on the applicability of Lyapunov-based approaches, such as conservatism in estimating the stability region, and difficulties associated with choosing a suitable LF. This is due to the fact that our reachability algorithm does not require a LF in order to provide an estimate of the ROA. Furthermore, the proposed procedure can estimate the exact ROA quite accurately, and more importantly, scales moderately with the system dimension compared to the approach based on backward reachability analysis. The aforementioned techniques are illustrated on various numerical examples.

4.1 Introduction

The estimation of stability regions of nonlinear systems is of fundamental importance in a wide range of applications, such as autonomous systems [57, 86], control of robotic manipulators [95], and transient stability of power systems [16, 25]. This problem is continuously receiving a lot of attention in the literature as it still remains unsolved. Simply put, instead of examining stability of an equilibrium point, one is often more interested in determining the region from which an initial state can be attracted by this equilibrium; this is particularly beneficial for studies involving transient stability, since one can avoid exhaustive time-domain simulations by simply inspecting if the system states of the post-fault scenario are confined within the ROA.

Finding the stability region, however, is difficult due to the fact that it is often a complicated set which is hard to be expressed analytically [76], furthermore, developing efficient algorithms to tackle this problem is known to be an extremely challenging task. These challenges consists of, but are not limited to, the conservatism in estimating the ROA, the overall computational costs, and the associated memory requirements, which grow rapidly with the system dimension. In this chapter, we present an algorithmic procedure based on forward reachability computations to provide accurate estimates of the stability region. Since this thesis is focused on power systems, the following literature review is primarily concerned with this class of systems, where the existing techniques are broadly classified as in Fig. 4.1.

4.1.1 State-of-the-art

The dominant techniques for estimating the ROA in power systems are based on Lyapunov's stability theory and its various extensions. These methods can offer sufficient conditions for verifying stability of ordinary differential equations (ODEs), using the so-called Lyapunov functions. Among its variations, the Zubov's method and the maximal Lyapunov function method [112,131] can obtain the exact stability region; however, both techniques are generally difficult to employ, since they reformulate the problem into solving a set of partial differential equations (PDEs) whose solution is not easily found in most cases.

Another variant is the so-called closest Unstable Equilibrium Point (UEP) in which the main concern is to estimate the relevant local stability boundary to which the fault-on system is heading. The UEP method establishes stability of an equilibrium point on the stability boundary resulting in a very conservative ROA. Furthermore, the method requires a significant computational effort making it infeasible for practical applications, although it can offer mathematical guarantees. Alternatively, the controlling UEP provides less conservative estimates than the closest UEP, but it is generally very difficult to finding the controlling UEP relative to the fault-on trajectory [16]. Both methods were heavily investigated in the literature, although they are immune to the second swing uprising and more importantly, they are only suitable for a specific class of power systems, in which one makes various unrealistic assumptions to eliminate the set of algebraic constraints governing dynamics of the system, see [28,33,88].

An alternative, and popular approach in the literature is based on the inner/under-approximation of the exact ROA via the optimization of the Lyapunov function sub-level set. This approach has proved to be quite effective following the rapid development of semi-algebraic geometry and SOS decomposition, which resulted in increasingly efficient optimization techniques involving linear matrix inequalities (LMIs) and semi-definite programming (SDP), see for example [16,32,59,94,113,129,134].

The Lyapunov-based techniques have several drawbacks which limit their applicability in practice: first, the approach relies on the existence of suitable Lyapunov functions which are extremely difficult to find for nonlinear systems. This is due to the non-constructive nature of the Lyapunov theory; in other words, the theorem only ensures the existence of a domain of attraction, yet it does not provide a systematic way to find an initial feasible Lyapunov function. Although a quadratic Lyapunov function can be easily constructed by solving the Lyapunov equation of the linearized system, it only captures the local behaviour around the equilibrium point. Second, Lyapunov methods suffer from conservatism in estimating the ROA, since the techniques often relax the optimization problem associated with the enlargement of the sub-level set of the Lyapunov function. This is done either by enforcing convexity of the solution using conservative linear matrix inequalities (LMIs) conditions, or by employing non-convex bilinear matrix inequalities [62]. Another disadvantage of this class of techniques is that it cannot formally verify if system constraints are met, e.g. the bus voltage of a power system dropping beyond limits imposed by the corresponding transmission system operator (TSO); this is due to the fact that Lyapunov methods analyze if a steady state of a disturbed system is eventually reached, however, without specifying the exact trajectory of the state variables.

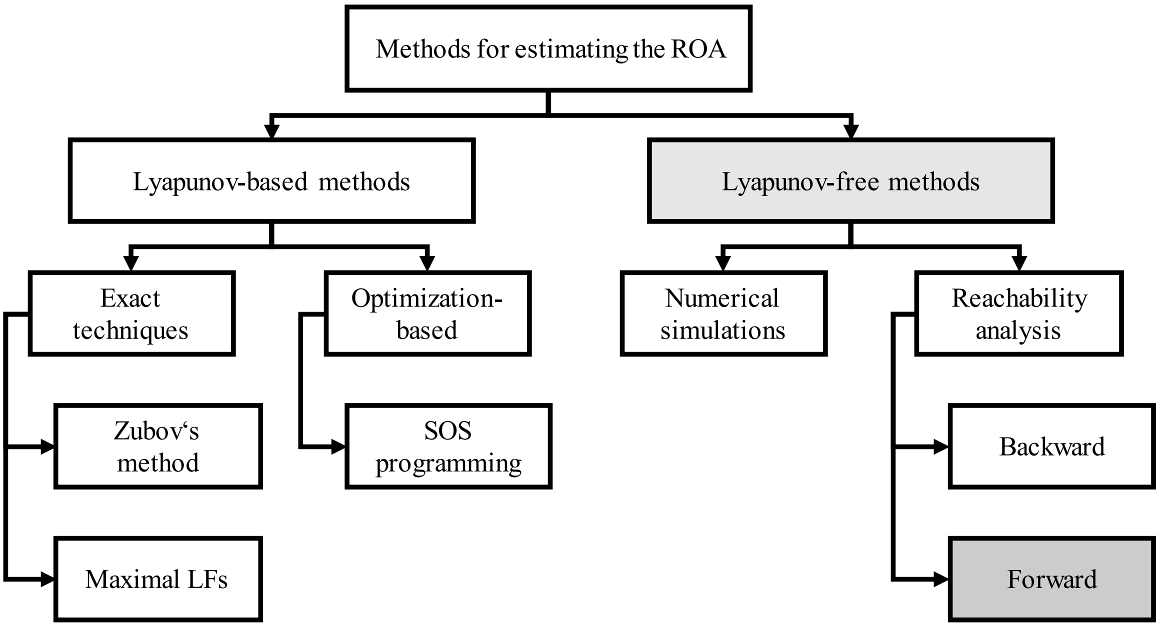


Figure 4.1: Broad classification of various methods computing the region of attraction in power systems.

An alternative direction is a class of methods which is not based on the Lyapunov stability theory. An overview covering the early contributions of this kind of techniques is found in [55]. Very recently, the computation of backward reachable enclosures using level set methods (LSMs), starting from a small neighborhood around the equilibrium point, has turned out to be an effective tool that can provide accurate estimates of the ROA, see e.g. [70]. This is achieved using an Eulerian technique via the formulation of a Hamilton-Jacobi-Isaacs (HJI) partial differential equations (PDEs), where it is proven that the viscosity solution of the time-dependent PDEs provides an implicit surface representation of the continuous backward reachable set [99].

As stated earlier in the previous chapter, the main drawback of this technique is that the computational requirements grow rapidly with the system dimension, due to the fact that no analytical solution exists for the specified set of PDEs; thus, one has to continuously perform a discretization of the state space, resulting in an exponential complexity with respect to the number of continuous state variables. Another limitation of level set methods is that they only provide an accurate approximation of the reachable set, rather than a rigorous enclosure of it; thus it does not qualify as a formal technique for the estimation of the ROA [124].

4.1.2 Contributions

The approach presented in this work is similar in spirit to the class of methods using reachability analysis, however, the key distinction is that our algorithm is based on Lagrangian techniques, which in contrast to Eulerian methods, compute reachable sets similar to numerical integration. This is achieved by propagating the set of reachable states instead of only computing the solution for a single value, see Fig. 2.8. A consequence of this is that we can handle higher-dimensional systems, since the associated memory requirements grow moderately with the system dimension.

The bulk of this chapter is based on the work published in [44]. To the best of our knowledge, this is the first contribution presenting an algorithmic procedure to estimate the ROA of an equilibrium point via the computation of forward reachable sets. We shall describe and implement a scalable and versatile algorithm that can provide accurate, and more importantly, provable estimates of the stability region; by versatile, we refer to the ability of the algorithm to deal with general systems involving non-polynomial models, thus covering a wide range of applications. Furthermore, we compare three different techniques that can provide estimates of the stability region; namely, we compare the proposed Lagrangian approach (*forward reachability using zonotopes*) with the Eulerian method (*backward reachability using level sets*) and the Lyapunov direct method (*optimization of the Lyapunov function sub-level set using SOS decomposition*).

4.2 Preliminaries

In this chapter, we consider general autonomous nonlinear systems formalized as a set of ODEs

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)),\ \forall \boldsymbol{x}(t) \in \boldsymbol{\mathcal{D}} \subseteq \mathbb{R}^n, \tag{4.1}$$

where $\boldsymbol{x}$ is the vector of state variables, $\boldsymbol{\mathcal{D}}$ is the domain and $\boldsymbol{f} : \boldsymbol{\mathcal{D}} \mapsto \mathbb{R}^n$ is locally Lipschitz continuous. The time dependency is often omitted for simplicity of notation.

4.2.1 Problem Formulation and Objective

Suppose the system under investigation has a position of rest denoted by $\boldsymbol{x}_s \in \boldsymbol{\mathcal{D}}$, i.e. $\boldsymbol{f}(\boldsymbol{x}_s) = \boldsymbol{0}$. First we will establish stability of $\boldsymbol{x}_s$ for the system (4.1) in the sense of Lyapunov. In fact, there are different forms of stability such as exponential, uniform or bounded-input bounded-output (BIBO), see [76]; in this chapter, however, we particularly consider the so-called Lyapunov asymptotic stability, since it is directly associated with the estimation of the ROA. Lyapunov asymptotic stability establishes whether the initial state $\boldsymbol{x}(0)$ of (4.1) will always remain confined within a small neighbour surrounding $\boldsymbol{x}_s$, if they are located near the equilibrium point. Formally, asymptotic stability is defined precisely as:

Definition 4.1. Lyapunov asymptotic stability: The equilibrium point $\boldsymbol{x}_s \in \boldsymbol{\mathcal{D}}$ of (4.1) is said to be Lyapunov stable at $t = 0$ if, and only if, for any $\boldsymbol{\epsilon} > \boldsymbol{0}$ there exists a $\boldsymbol{\delta} > \boldsymbol{0}$ such that

$$|\, \boldsymbol{x}(0) - \boldsymbol{x}_s \,| < \boldsymbol{\delta} \quad \Rightarrow \quad |\, \boldsymbol{\chi}(\boldsymbol{x}(0); t) - \, \boldsymbol{x}_s \,| < \boldsymbol{\epsilon}, \, \forall t \geq 0,$$

and asymptotical stability is established if $\boldsymbol{x}_s$ is stable and locally attractive; that is the region $\boldsymbol{\delta}$ have to be chosen such that

$$|\, \boldsymbol{x}(0) - \boldsymbol{x}_s \,| < \boldsymbol{\delta} \quad \Rightarrow \quad \lim_{t \to \infty} \boldsymbol{\chi}(\boldsymbol{x}(0); t) = \boldsymbol{x}_s. \qquad \square$$

Here $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$ correspond to two arbitrary regions in the state-space, and $\boldsymbol{\chi}(\boldsymbol{x}(0); t)$ specifies a system trajectory of (4.1) starting from the initial condition $\boldsymbol{x}(0)$.

Assuming that the equilibrium point $\boldsymbol{x}_s$ of (4.1) is stable according to Def. 4.1, then one can clearly see that $\boldsymbol{\delta}$ defines a region in the domain $\boldsymbol{\mathcal{D}}$ from which any initial state is guaranteed to be attracted by $\boldsymbol{x}_s$. In this chapter we seek to find this stability region, which is formally defined as follows:

Definition 4.2. Region of attraction: Given the nonlinear system (4.1) whose equilibrium point $\boldsymbol{x}_s$ is locally asymptotically stable in the sense of Lyapunov according to Def. 4.1, then the ROA surrounding $\boldsymbol{x}_s$ is expressed via the set

$$\boldsymbol{\mathcal{S}}^e(\boldsymbol{x}_s) = \left\{ \boldsymbol{x}_0 \in \boldsymbol{\mathcal{D}} \, : \, \lim_{t \to \infty} \boldsymbol{\chi}(\boldsymbol{x}(0); t) = \boldsymbol{x}_s \right\}. \qquad \square$$

Here the subscript e specifies the exact stability region which is hard, or even impossible, to express analytically except for a specific class of nonlinear systems; hence, similarly to computation of reachable sets, our objective is to find an estimate of this region such that $\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s) \subseteq \boldsymbol{\mathcal{S}}^e(\boldsymbol{x}_s)$ with $\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s)$ as the set associated with the under-approximation of the exact ROA.

4.2.2 Existing Techniques

As stated earlier and illustrated in Fig. 4.1, the existing methods to estimate the ROA can be broadly categorized into Lyapunov-based and Lyapunov-free techniques. Here we shall briefly illustrate a technique from each category in order to compare them with our proposed contribution later in Sec. 4.4.

4.2.2.1 Level Set Method

The first class of techniques is based on an Eulerian scheme using level set methods. Generally speaking, LSMs are a collection of numerical algorithms solving a specific class of PDEs often encountered in simulation of dynamic implicit surfaces for a variety of applications, such as fluids, image processing, and computer vision [97]. Instead of explicitly representing a surface via its vertices, edges or faces, the LSM describes surfaces implicity via a level set function, often denoted by $\phi(t, \boldsymbol{x}) : \mathbb{R}^n \mapsto \mathbb{R}$. Using LSMs, one can compute the backward reachable set starting from a small neighboured surrounding $\boldsymbol{x}_s$.
Here finding an estimate of the ROA is reformulated into a reachability problem, which in turn is casted into solving the following first-order time-dependent HJI PDE:

$$\frac{\partial \phi(t, \boldsymbol{x})}{\partial t} + \mathscr{H}\left(\boldsymbol{x},\, \frac{\partial \phi(t, \boldsymbol{x})}{\partial x}\right) = 0, \tag{4.2}$$

where $\mathscr{H}$ is a basic Hamiltonian function. For further details about Hamiltonian dynamics, HJI formulation and viscosity solutions, the reader is referred to [21, Ch. 2]. It is proven in [99] that the viscosity solution of (4.2) provides a surface representation $\phi(t, \boldsymbol{x}) \leq 0$ corresponding to the continuous reachable set of differential state variables; that is the backward reachable set is expressed via:

$$\textbf{BackwardReach}(\boldsymbol{\mathcal{R}}(0), t) := \Big\{ \boldsymbol{x} \in \boldsymbol{\mathcal{D}} \,:\, \phi(t, \boldsymbol{x}) \leq 0 \Big\} = \boldsymbol{\mathcal{S}}^e(\boldsymbol{x}_s),$$

with **BackwardReach** denoting the backward reachable set starting from the initial set of states $\boldsymbol{\mathcal{R}}(0)$.

4.2.2.2 Lyapunov method

The second class of techniques is based on Lyapunov direct method. Without loss of generality, we place the equilibrium point $\boldsymbol{x}_s$ at the origin $\mathbf{0}_n$. Let $V(\boldsymbol{x}) : \boldsymbol{\mathcal{D}} \mapsto \mathbb{R}$ denote a scalar positive definite function, that is $V(\boldsymbol{x}) > 0$, and $V(\mathbf{0}_n) = 0$. This function is often refereed to as the so-called Lyapunov function of (4.1), where its partial derivatives are continuous functions within the domain $\boldsymbol{\mathcal{D}}$ surrounding $\boldsymbol{x}_s$. Furthermore, the derivative with respect to time has to be negatively definite in $\boldsymbol{\mathcal{D}}$, that is:

$$\dot{V} = \sum_{i=1}^{n} \frac{\partial V(\boldsymbol{x})}{\partial x_i} f_i(\boldsymbol{x}) < 0,\ \boldsymbol{x} \in \boldsymbol{\mathcal{D}} / \{\mathbf{0}_n\}.$$

Here the problem of providing an estimate of the ROA is reformulated into finding an optimal Lyapunov function, whose sub-level set provides the largest estimate of the ROA. Formally, the aforementioned task is equivalent to solving the following optimization problem [59]:

$$\sup_{c_v} \rho\left(\mathcal{V}(c_v)\right) \quad \text{s.t.} \begin{cases} \boldsymbol{x} \in \boldsymbol{\mathcal{D}}/\{\mathbf{0}_n\} :\ \dot{V}(\boldsymbol{x}) < 0,\, V(\boldsymbol{x}) > 0, \\ \dot{V}(\mathbf{0}_n) = 0,\, V(\mathbf{0}_n) = 0, \\ \mathcal{V}(c_v) := \Big\{ \boldsymbol{x} \in \boldsymbol{\mathcal{D}} \,:\, V(\boldsymbol{x}) \leq c_v \Big\} \subseteq \boldsymbol{\mathcal{S}}^e(\boldsymbol{x}_s), \end{cases} \tag{4.3}$$

with the constant $c_v \in \mathbb{R}^+$ and $\rho(\,\cdot\,)$ specifying a pre-definable measure of the sub-level set $\mathcal{V}(c_v)$.

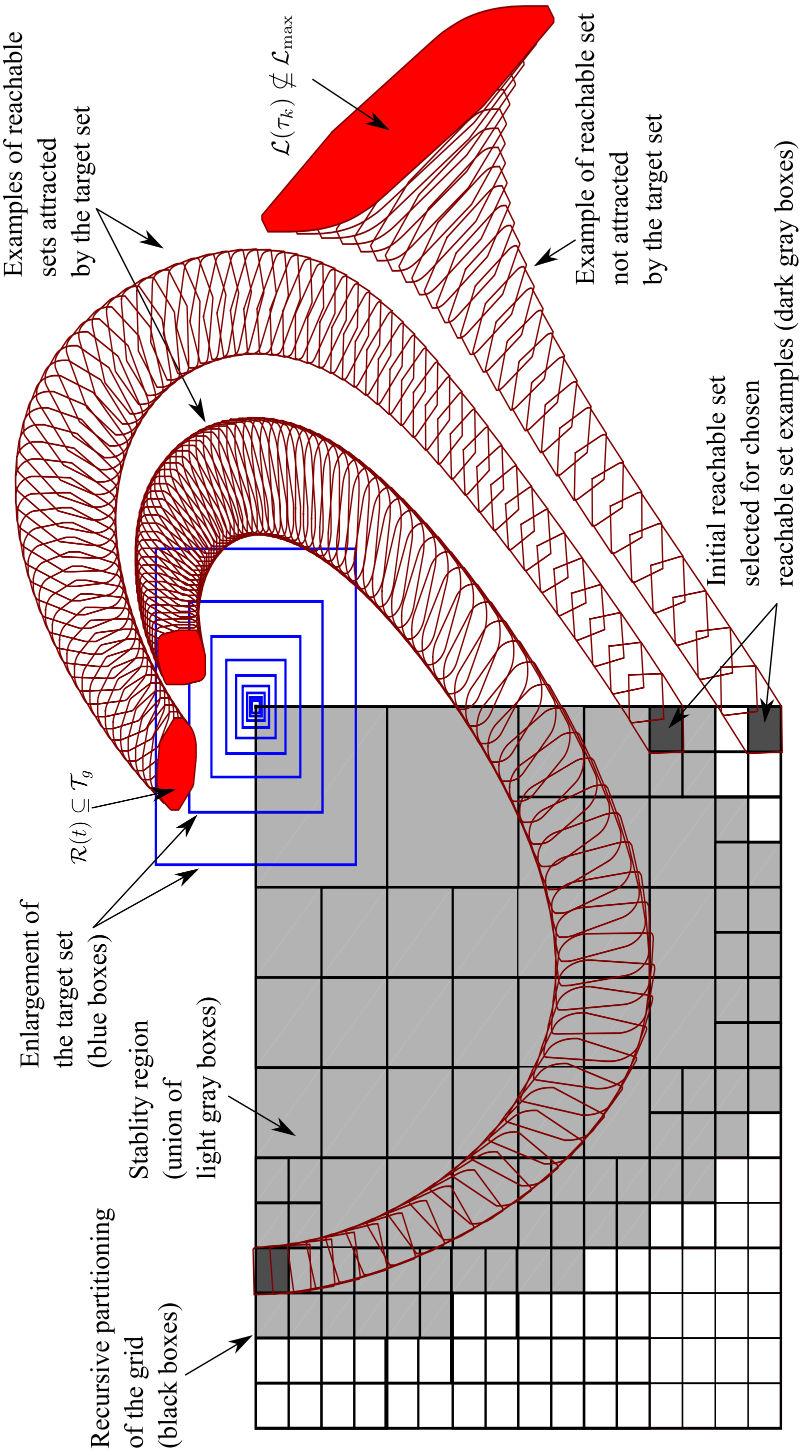

Examples of reachable sets attracted by the target set
$\mathcal{L}(\tau_k) \nsubseteq \mathcal{L}_{\max}$
Example of reachable set not attracted by the target set
Initial reachable set selected for chosen reachable set examples (dark gray boxes)
$\mathcal{R}(t) \subseteq \mathcal{T}_g$
Enlargement of the target set (blue boxes)
Stablity region (union of light gray boxes)
Recursive partitioning of the grid (black boxes)

Figure 4.2: Step-by-step computation of the stability region using forward reachable sets. The blue boxes represent the iterative enlargement of the target set $\boldsymbol{\mathcal{T}}_g$ constructed around an equilibrium point (step 1). The boxes show the recursive partitioning of the grid to provide an accurate estimate of the ROA (step 2), see remark 4.2. The dark gray areas are the cells whose reachable set is attracted by the enlarged target set, whereas the white areas correspond to the cells whose reachable set does not converge. Three random cells were chosen to illustrate the evolution of the reachable set (dark red), where the computation stops (highlighted by the red area) when the reachable set of the i-th cell $\boldsymbol{\mathcal{R}}_i(t)$, $t \geq 0$ is either a subset of the target set $\boldsymbol{\mathcal{R}}_i(t) \subseteq \boldsymbol{\mathcal{T}}_g$ or if the Lagrangian remainder is not a subset of the maximum linearization errors $\boldsymbol{\mathcal{L}} \nsubseteq \boldsymbol{\mathcal{L}}_{\max}$, see (4.8).

4.2.3 Proposed Approach

The problem is approached in this chapter using forward reachable sets based on the algorithms presented earlier in Sec. 2.4. Hereafter, we introduce the target set denoted by $\boldsymbol{\mathcal{T}}_g$. Basically, the target set is a small region surrounding the stable equilibrium point, based on Lyapunov asymptotic stability, see Def. 4.1. With the introduction of the target set, our problem of finding $\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s)$ is reformulated into

$$\begin{aligned}
&\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s) := \boldsymbol{\mathcal{R}}_1(0) \cup \boldsymbol{\mathcal{R}}_2(0) \cup \ldots \cup \boldsymbol{\mathcal{R}}_{N_{\text{cells}}}(0) \subseteq \boldsymbol{\mathcal{S}}^e(\boldsymbol{x}_s) \qquad (4.4)\\
&\text{s.t.} \begin{cases} \forall i \in \{1, \ldots, N_{\text{cells}}\} : \\ \exists t \geq 0 : \boldsymbol{\mathcal{R}}_i(t) \overset{\text{Def.4.1}}{\subseteq} \boldsymbol{\mathcal{T}}_g, \\ \boldsymbol{\mathcal{R}}_i(t) \in \mathbf{reach}(\boldsymbol{\mathcal{R}}_i(0)) := \Big\{\boldsymbol{x}(t) \in \boldsymbol{\mathcal{D}} \,:\, \boldsymbol{x}(t) \text{ satisfies (4.1) within } t \geq 0 \text{ for } \boldsymbol{x}(0) \in \boldsymbol{\mathcal{R}}_i(0)\Big\}. \end{cases}
\end{aligned}$$

In (4.4), the stability region is specified via the union of initial reachable sets $\boldsymbol{\mathcal{R}}_i(0)$, $i \in \{1, \ldots, N_{\text{cells}}\}$, where $N_{\text{cells}} \in \mathbb{N}$ specifies the number of cells resulting from a partitioning of the domain $\boldsymbol{\mathcal{D}}$. The main idea is to discretize the working domain into smaller regions and examine whether each cell belongs to the ROA; that is for each $\boldsymbol{\mathcal{R}}_i$, we check at each time instant whether the reachable set of differential state variables is confined within a target set denoted by $\boldsymbol{\mathcal{T}}_g$, see Fig. 4.2. As stated earlier throughout Ch. 2, it is proven that the exact reachable set for the class of nonlinear systems are not computable [111]; thus an over-approximation, which includes all behaviours of (4.1) is performed as tightly as possible, see Fig. 2.7.

4.2.4 Organization

The remainder of the chapter is organized as follows: In Sec. 4.3 we briefly recall reachability computations for the class of systems described via nonlinear ODEs, then we present the estimation algorithm. In Sec. 4.4 we illustrate the applicability of our algorithm and compare it with existing techniques. The chapter is summarized in Sec. 4.5.

4.3 Estimation Algorithm

In this section, we present our proposed algorithm to estimate the stability region of the equilibrium point $\boldsymbol{x}_s$ using forward reachability computations.

4.3.1 Computation of Forward Reachable Sets

Recall that we previously introduced a reachability algorithm for the class of nonlinear DAE systems; hence, we need to make some modifications to fit it to the class of autonomous nonlinear systems. First, we need to abstract the original differential equations (4.1) into linear differential inclusions (LDIs) for each consecutive time interval $\tau_k := [t_k, t_{k+1}]$, where $t_k = k \cdot t_r$, such that $k \in \mathbb{N}$, and $t_r \in \mathbb{R}^+$ corresponds to the time step and the time increment, respectively.

After introducing the linearization point $\boldsymbol{x}_k \in \boldsymbol{\mathcal{D}}$, and defining $\Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_k$, we abstract (4.1) at each time interval τ_k, by a LDI expressed as a first-order Taylor expansion with the Lagrangian remainder, see Prop. 2.1

$$\forall t \in \tau_k = [t_k, t_{k+1}] : \; \dot{\boldsymbol{x}}(t) \in \underbrace{\sum_{j=1}^{n} \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_j}\bigg|_{\boldsymbol{x}=\boldsymbol{x}_k} \Delta x_j}_{=: \boldsymbol{A}_k \Delta \boldsymbol{x}} \oplus \underbrace{\boldsymbol{f}(\boldsymbol{x}_k) \oplus \boldsymbol{\mathcal{L}}(\tau_k)}_{=: \boldsymbol{\mathcal{U}}(\tau_k)}, \tag{4.5}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{n \times n}$ is the system matrix, $\boldsymbol{\mathcal{U}}$ is the set of uncertain inputs, and $\boldsymbol{\mathcal{L}}$ denotes the set of the Lagrangian remainder containing all possible linearization errors within the time interval τ_k

$$\boldsymbol{\mathcal{L}}(\tau_k) = \left\{ \boldsymbol{L} \in \boldsymbol{\mathcal{D}} \,:\, L_j = \frac{1}{2} \sum_{l=1}^{n} \sum_{m=1}^{n} \frac{\partial^2 f_j(\boldsymbol{x})}{\partial x_l \partial x_m}\bigg|_{\boldsymbol{x}=\boldsymbol{\mu}} \Delta x_l \Delta x_m,\, \boldsymbol{x} \in \boldsymbol{\mathcal{R}}(\tau_k),\, \boldsymbol{\mu} \in \textbf{interval}(\boldsymbol{\mathcal{R}}(\tau_k)) \right\},$$

with the subscript j corresponding to the j-th coordinate. Here $\boldsymbol{\mathcal{R}}(\tau_k)$ denotes the reachable set at τ_k which is computed as summarized in Alg. 1 and the operator **interval** returns the interval enclosure of a zonotope as in (2.34).

4.3.2 Estimation of the ROA

Now we present the proposed algorithmic procedure to estimate the stability region of the equilibrium point $\boldsymbol{x}_s$. The estimation algorithm consists of four steps:

Step ① ***Construction of a target set***: Since we require a termination condition for forward reachable sets, we establish a small region around the equilibrium point. Once a reachable set is enclosed by this region, we can terminate our computations and conclude that all solutions converge to $\boldsymbol{x}_s$. To this end, the target set $\boldsymbol{\mathcal{T}}_g \subset \boldsymbol{\mathcal{D}}$ is expressed via the multidimensional interval

$$\begin{aligned} \boldsymbol{\mathcal{T}}_g &= [\underline{\boldsymbol{x}}_s, \overline{\boldsymbol{x}}_s], \\ \text{with} \quad \underline{\boldsymbol{x}}_s &= \boldsymbol{x}_s - \boldsymbol{\epsilon}, \quad \overline{\boldsymbol{x}}_s = \boldsymbol{x}_s + \boldsymbol{\epsilon}, \end{aligned} \tag{4.6}$$

with $\boldsymbol{\epsilon}$ chosen to be sufficiently small (e.g. 1E-3) such that $\boldsymbol{\mathcal{T}}_g$ only surrounds a small neighborhood around $\boldsymbol{x}_s$, thus ensuring attraction to the equilibrium. This step is based on Lyapunov asymptotic stability as in Def. 4.1, which states that $\boldsymbol{x}_s$ attracts any initial state $\boldsymbol{x}(0)$ located adequately close.

Step ② ***Enlargement of the target set***: This step is illustrated in Fig. 4.2 by the blue boxes. Here we perform an iterative procedure to enlarge $\boldsymbol{\mathcal{T}}_g$ using forward reachability computations. The enlargement of $\boldsymbol{\mathcal{T}}_g$ is executed for two reasons: First, in order to obtain a larger provable region that guarantees convergence to the equilibrium, second, and more importantly, to reduce the overall computational costs associated with reachability computations of initial states far away from $\boldsymbol{\mathcal{T}}_g$, as described shortly. In each iteration, the enlarged target set is selected as the initial reachable set, such that

$$\boldsymbol{\mathcal{R}}(0) = \Lambda \cdot [\underline{\boldsymbol{x}}_s, \overline{\boldsymbol{x}}_s], \tag{4.7}$$

where $\Lambda \in \mathbb{R}^+$ is an enlargement factor set by the user.

Remark 4.1. The enlargement in (4.7) only holds if the origin is included in the target set $\boldsymbol{\mathcal{T}}_g = [\underline{\boldsymbol{x}}_s, \overline{\boldsymbol{x}}_s]$. To overcome this limitation and generalize (4.7), one still needs to add a correction factor which corresponds to the center of $\boldsymbol{\mathcal{T}}_g$, thus

$$\boldsymbol{\mathcal{R}}(0) = \Lambda \cdot [\underline{\boldsymbol{x}}_s, \overline{\boldsymbol{x}}_s] \oplus \left(-\frac{\overline{\boldsymbol{x}}_s + \underline{\boldsymbol{x}}_s}{2} \right). \qquad \square$$

The iterative procedure stops when $\boldsymbol{\mathcal{R}}(0)$ leads to a reachable set which is no longer attracted by the equilibrium. To examine this condition, we check if

$$\boldsymbol{\mathcal{L}} \nsubseteq \boldsymbol{\mathcal{L}}_{\max}, \tag{4.8}$$

where the set of Lagrangian remainders $\boldsymbol{\mathcal{L}}$ is computed similarly to the procedure described in Sec. 2.4.3 and the set $\boldsymbol{\mathcal{L}}_{\max}$ is the maximum linearization errors chosen by the user. The aforementioned condition is based on the fact that larger initial sets lead to large over-approximations of the Lagrangian remainder $\boldsymbol{\mathcal{L}}$, and convergence is no longer guaranteed. Therefore, the enlargement procedure of $\boldsymbol{\mathcal{T}}_g$ is limited and would only result in a very conservative stability region using forward reachability computations. Hence, in the following step, we investigate the domain surrounding $\boldsymbol{x}_s$ using a complementary procedure in order to estimate the ROA more accurately.

Step ③ ***Partitioning of the domain***: This step is illustrated in Fig. 4.2, where the boxes present a recursive discretization of the grid. Here, we perform a partitioning of the state space to overcome the limitations imposed on the enlargement of $\boldsymbol{\mathcal{T}}_g$. Clearly, it is neither practical nor computationally feasible to discretize the whole state space $\mathbb{R}^n$ when attempting to estimate the stability region. This is based on the fact that one is particularly interested in a specific domain $\boldsymbol{\mathcal{D}}$ associated with the practical constraints of the system. We assume this domain is expressed via a multi-dimensional interval; that is $\boldsymbol{\mathcal{D}} := [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$, with $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ denoting its upper and lower bound.

In this chapter, the domain $\boldsymbol{\mathcal{D}}$ is partitioned into a number of segments, each represented as well via a multi-dimensional interval such that

$$\mathbf{partition}(\boldsymbol{\mathcal{D}}, \boldsymbol{\xi}) = \bigcup_{i=1}^{N_{\text{cells}}} \boldsymbol{\mathcal{B}}_i, \tag{4.9}$$

with the vector $\boldsymbol{\xi} \in \mathbb{N}^n$ specifying the desired number of segments in each j-th dimension, $N_{\text{cells}} \in \mathbb{N}$ denoting the total number of segments

$$N_{\text{cells}} = \prod_{j=1}^{n} \xi_j, \tag{4.10}$$

resulting from the discretization of the domain $\boldsymbol{\mathcal{D}}$, and $\boldsymbol{\mathcal{B}}_i$, $i \in \{1 \ldots N_{\text{cells}}\}$ being the resulting cells, where each i-th cell is described according to

$$\forall i \in \{1, \ldots, N_{\text{cells}}\} : \boldsymbol{\mathcal{B}}_i := [\underline{\boldsymbol{B}}_i, \overline{\boldsymbol{B}}_i] \quad \text{s.t.} \begin{cases} \underline{\boldsymbol{B}}_i = \underline{\boldsymbol{x}} + \left(\boldsymbol{\Gamma}^{(i)} \bullet \boldsymbol{S}_l\right) - \mathbf{1}, \\ \overline{\boldsymbol{B}}_i = \underline{\boldsymbol{x}} + \left(\boldsymbol{\Gamma}^{(i)} \bullet \boldsymbol{S}_l\right), \end{cases} \tag{4.11}$$

where $\underline{\boldsymbol{B}}_i$ and $\overline{\boldsymbol{B}}_i$ denotes to the upper and lower bound of the i-th cell, respectively, $\boldsymbol{S}_l \in \mathbb{R}^n$ is the vector specifying the segment length in each dimension such that

$$S_{l,j} = \frac{\overline{x}_j - \underline{x}_j}{\xi_j}, \tag{4.12}$$

and the operand $\bullet$ returns the Hadamard product (element-wise multiplication of matrices). Here the matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times N_{\text{cells}}}$ appearing in (4.11) is obtained according to

$$\begin{aligned} \Gamma &:= \mathbf{comVec}\left(\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \ldots, \boldsymbol{a}^{(n)}\right), \ \boldsymbol{a}^{(i)} \in \mathbb{N}^{\xi_i} \\ \text{s.t.} \quad & \forall i \in \{1, \ldots, n\} : \ \boldsymbol{a}^{(i)} = \begin{pmatrix} 1 & 2 & \ldots & \xi_i \end{pmatrix} \end{aligned} \tag{4.13}$$

with $\boldsymbol{a}^{(i)}, i \in \{1, \ldots, n\}$ specifying a set of column vectors with different dimensions and the operator $\mathbf{comVec}(\,\cdot\,)$ returns a matrix of N_{cells} column vectors such that the columns consist of all possibilities of the vector $\boldsymbol{a}^{(n)}$ appended to $\boldsymbol{a}^{(n-1)}$, up to the vector $\boldsymbol{a}^{(1)}$.

Following the discretization of $\boldsymbol{\mathcal{D}}$, each i-th grid cell is selected as the initial set for the reachability algorithm. The cell is formally proven to belong to the stability region of the equilibrium point if the resulting reachable set of states is a subset of the target region, that is:

$$\exists t : \mathbf{reach}(\boldsymbol{\mathcal{B}}_i, t) \subseteq \boldsymbol{\mathcal{T}}_g, \tag{4.14}$$

with **reach** returning the reachable set of (4.5) as described shortly after.

Remark 4.2. During the implementation of our algorithm, we have found that a recursive partitioning of the grid, starting with a large cell size is more efficient in terms of computational time than the discretization of the domain of investigation directly with a fixed $\boldsymbol{\xi}$. This is due to the fact that a recursive partitioning with different sizes allows one to rapidly explore large areas around $\boldsymbol{\mathcal{T}}_g$, and if these areas converge to the equilibrium, one does not need to re-examine them.

Step ④ ***Aggregation of results***: Following the examination of all cells, the stability region is assembled via the union of cells formally proven to converge to $\boldsymbol{\mathcal{T}}_g$, that is

$$\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s) := \bigcup_{i=1}^{l} \boldsymbol{\mathcal{B}}_i \subset \boldsymbol{\mathcal{S}}^e(\boldsymbol{x}_s), \quad \text{s.t} : (4.14) \text{ holds} \tag{4.15}$$

4.3.3 Algorithmic Realization

The overall procedure to obtain the stability region is summarized in Alg. 5 and Alg. 6, which outline the computation of forward reachable sets and the estimation of the ROA, respectively. Six parameters are passed to the algorithm by the user: the stable equilibrium point $\boldsymbol{x}_s$, the domain of investigation $\boldsymbol{\mathcal{D}}$, the target set enlargement factor Λ, the partitioning size of the grid $\boldsymbol{\xi}$, the time increment t_r, and the maximum linearization errors $\boldsymbol{\mathcal{L}}_{\max}$.

4.3.3.1 Forward Reachability Algorithm

Alg. 5 is a modified procedure of the reachability analysis algorithm introduced earlier in Sec. 2.4 to compute reachable sets of nonlinear DAE systems. Clearly, the present algorithm is simpler than Alg. 1 since we are dealing with autonomous systems described via a set of ODEs rather than DAEs which differ in both theoretical and numerical properties.

First the algorithm obtains the local linearization point then abstracts the nonlinear system (4.1) into the LDI described via (4.5). This step is described via the operation **taylor**$(\,\cdot\,)$. Throughout lines $6-11$, reachability computations are performed for consecutive time intervals $\tau_k := [t_k, t_{k+1}]$; the procedure stops if the reachable set is enclosed by the stability region, passed from Alg. 6 and discussed shortly after, or if the set of Lagrangian remainder $\boldsymbol{\mathcal{L}}$ exceeds the maximum allowable error $\boldsymbol{\mathcal{L}}_{\max}$ specified by the user, see remark 2.10. Finally, the output of the forward reachability algorithm is then passed to Alg. 6 in order to construct the stability region.

4.3.3.2 ROA Algorithm

Now we summarize the algorithmic procedure to estimate the stability region based on computation of forward reachable sets as shown in Alg. 6. First, throughout lines $2-9$, we construct and enlarge the region surrounding the equilibrium point $\boldsymbol{x}_s$ based on the definition of asymptotic stability in the sense of Lyapunov, see Def. 4.1. These steps are performed via the operations **Lyap** and **enlarge**, correspondingly.

As stated earlier, the enlargement procedure is limited to a certain extent, hence throughout lines $10-20$, we discretize the domain $\boldsymbol{\mathcal{D}}$ into smaller segments and examine each one using Alg. 5. Finally, the ROA is specified via the union of the segments whose reachable set is formally proven to converge to $\boldsymbol{\mathcal{T}}_g$.

Algorithm 5 ForwardReach

```
Require: The initial set $\boldsymbol{\mathcal{R}}(0)$, the stability region $\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s)$, and the maximum linearization errors $\boldsymbol{\mathcal{L}}_{\max}$
Ensure: isConverging
Initialization: $k = 0$, $t_k = 0$, $\tau_k = [t_k, t_{k+1}]$, and isConverging = true
repeat
    $\tilde{\boldsymbol{x}}_k \leftarrow$ center($\boldsymbol{\mathcal{R}}(t_{k+1})$)    ▷ Local linearization point
    $\boldsymbol{A}(t_k)$, $\boldsymbol{f}(\tilde{\boldsymbol{x}}_k) \overset{(4.5)}{\longleftarrow}$ taylor($\boldsymbol{f}(\boldsymbol{x})$)    ▷ Abstraction to LDIs
    Compute $\boldsymbol{\mathcal{R}}(t_{k+1})$ and $\boldsymbol{\mathcal{R}}(\tau_k)$ similarly to Alg. 1
    Compute $\boldsymbol{\mathcal{L}}(\tau_k)$ based on $\boldsymbol{\mathcal{R}}(\tau_k)$ similarly to Alg. 2
    if $\boldsymbol{\mathcal{L}}(\tau_k) \overset{(4.8)}{\nsubseteq} \boldsymbol{\mathcal{L}}_{\max} \vee \boldsymbol{\mathcal{R}}(t_{k+1}) \subseteq \boldsymbol{\mathcal{S}}(\boldsymbol{x}_s)$ then    ▷ Termination conditions
        isConverging = false
    end if
    Set $t_{k+1} := t_k + t_r$, $k := k + 1$    ▷ Update parameters for next iteration
until $\neg$ isConverging
if $\boldsymbol{\mathcal{R}}(t_{k-1}) \subseteq \boldsymbol{\mathcal{S}}(\boldsymbol{x}_s)$ then    ▷ Check if recently computed set belongs to the stability region
    isConverging = true
end if
```

Algorithm 6 EstimateROA

```
Require: The stable equilibrium point $\boldsymbol{x}_s$, the number of segments in each dimension $\boldsymbol{\xi}$, the domain $\boldsymbol{\mathcal{D}}$,
    and the enlargement factors $\Lambda$ and $\boldsymbol{\epsilon}$.
Ensure: $\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s)$
Initialization: $l = 1$, $\boldsymbol{\mathcal{G}}_l = \emptyset$, $\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s) = \emptyset$ and $N_{\text{cells}} = \prod_{i=1}^{n} \xi_i$
$\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s) \overset{\text{Def. 4.1}}{\longleftarrow}$ Lyap ($\boldsymbol{x}_s$, $\boldsymbol{\epsilon}$)    ▷ Construction of the ROA based on a target set step ①
repeat
    $\boldsymbol{\mathcal{R}}(0) \overset{(4.7)}{\longleftarrow}$ enlarge($\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s)$, $\Lambda$)    ▷ Enlargement of the ROA step ②
    isConverging $\overset{\text{Alg. 5}}{\longleftarrow}$ ForwardReach($\boldsymbol{\mathcal{R}}(0), \boldsymbol{\mathcal{S}}(\boldsymbol{x}_s), \dots$)
    if isConverging then
        $\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s) \leftarrow \boldsymbol{\mathcal{R}}(0)$    ▷ Update the stability region
    end if
until $\neg$ isConverging    ▷ Abort enlargement of the ROA when $\boldsymbol{\mathcal{R}}(0)$ is no longer converging to $\boldsymbol{x}_s$
$\bigcup_{i=1}^{N_{\text{cells}}} \boldsymbol{\mathcal{R}}_i(0) \longleftarrow$ partition($\boldsymbol{\mathcal{D}}, \boldsymbol{\xi}$)    ▷ Partitioning of the domain according to step ③
for $i = 1 \dots N_{\text{cells}}$ do
    if $\neg(\boldsymbol{\mathcal{R}}_i(0) \subseteq \boldsymbol{\mathcal{S}}(\boldsymbol{x}_s))$ then
        isConverging $\overset{\text{Alg. 5}}{\longleftarrow}$ ForwardReach($\boldsymbol{\mathcal{R}}_i(0), \boldsymbol{\mathcal{S}}(\boldsymbol{x}_s), \dots$)
        if isConverging then
            $\boldsymbol{\mathcal{G}}_l \leftarrow \boldsymbol{\mathcal{R}}_i(0)$, $l := l + 1$
        end if
    end if
end for
$\boldsymbol{\mathcal{S}}(\boldsymbol{x}_s) \overset{\text{step ④}}{\longleftarrow} \bigcup_{k=1}^{l} \boldsymbol{\mathcal{G}}_k$,    ▷ Estimate of the stability region
```

4.4 Results

We demonstrate the applicability of our proposed algorithm on various benchmark examples. All computations are performed in MATLAB2016b on a standard computer with an Intel Core i7-4810MQ CPU and 16GB of RAM. Our algorithm computes forward reachable sets using the CORA toolbox [9]. Our results are compared with other well-established tools: The level set toolbox (LST) toolbox[1] [97] and the SMRSOFT toolbox[2] [31], which compute the ROA using alternatives techniques. In particular, LST computes the backward reachable set of a target set enclosing an equilibrium point using the viscosity solution of a time-dependant HJI PDE as in (4.2), while SMRSOFT estimates the stability region by maximizing the Lyapunov function sub-level set using SOS programming, see (4.3). Note that the toolboxes offer various functions other than estimating the ROA for nonlinear systems.

In this chapter, the Lyapunov function $V(\boldsymbol{x})$ is usually chosen to be quadratic[3]

$$V(\boldsymbol{x}) = \sum_{l=1}^{n} \sum_{m=1}^{n} M_{lm} x_l x_m,$$

with $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ being a real symmetric matrix obtained by solving the continuous Lyapunov equation $\boldsymbol{A}^T \boldsymbol{M} + \boldsymbol{M}\boldsymbol{A} + \boldsymbol{Q} = \boldsymbol{0}$, with $\boldsymbol{Q}$ chosen to be the identity matrix, and $\boldsymbol{A}$ is the system matrix of the linearized system at $\boldsymbol{x}_s$.

4.4.1 Two-Dimensional Systems

First, we consider the backward Van-der-Pol oscillator

$$\begin{aligned} \dot{x}_1 &= -x_2, \\ \dot{x}_2 &= -x_2(1 - x_1^2) + x_1. \end{aligned} \tag{4.16}$$

The system has a stable equilibrium point at the origin, the domain of investigation is defined by the multi-dimensional interval $\boldsymbol{\mathcal{D}} := [-2.5, 2.5] \times [-2, 2]$, where the operator $\times$ denotes the Cartesian product, and the chosen Lyapunov function is $V(\boldsymbol{x}) = 1.5x_1^2 - x_1x_2 + x_2^2$ (*only needed in SMRSOFT*). The estimated ROA is shown in Fig. 4.3. The regions are obtained using our proposed algorithm, the level set toolbox, and SMRSOFT. The gray areas show the cells attracted by the equilibrium, following the recursive partitioning of the grid. Notice that additional partitioning takes place when examining areas far away from the equilibrium point (target set). It can be seen that the estimate of the stability region using the Lyapunov-based approach is relatively conservative compared to both regions obtained by forward/backward reachable sets. Furthermore, the ROA is slightly more accurate using the level set toolbox.

[1] `http://www.cs.ubc.ca/~mitchell/ToolboxLS/`

[2] `https://www.eee.hku.hk/~chesi/y_smrsoft.htm`

[3] Note that a Lyapunov function $V(\boldsymbol{x})$ is not required in our algorithm

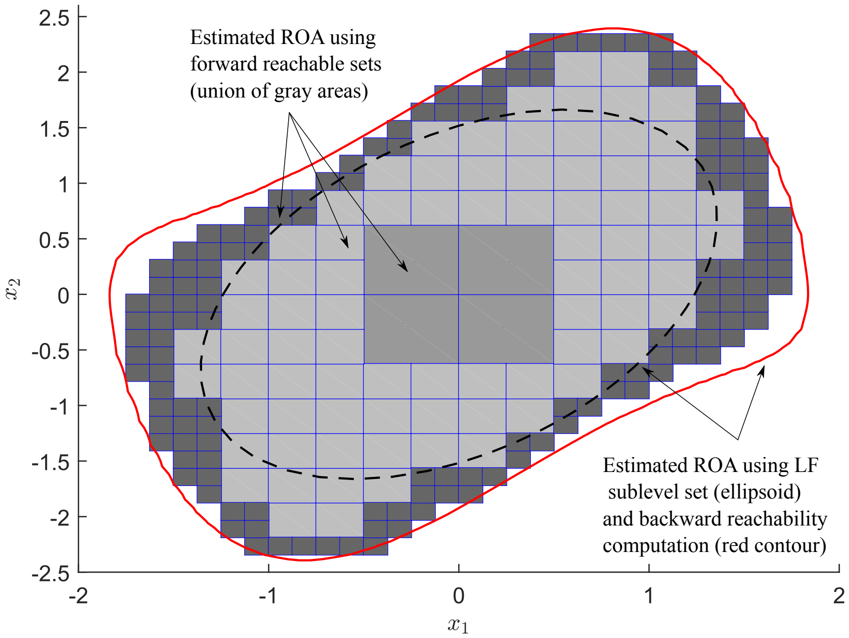


Figure 4.3: Estimation of the stability region of (4.16) using our proposed algorithm (gray cells), level set toolbox (red-solid contour), and SMRSOFT (black-dotted ellipsoid). The equilibrium point is located at the origin. The gray areas of the ROA are associated with the recursive partitioning of the domain, see remark 4.2.

Next, we examine the following example from [135]:

$$\begin{aligned} \dot{x} &= -2x + y + x^3 + y^5, \\ \dot{y} &= -x - y + x^2 y^5. \end{aligned} \tag{4.17}$$

The system has a stable equilibrium point at the origin. Here, the domain of investigation is $\boldsymbol{\mathcal{D}} := [-2, 2] \times [-2, 2]$. The stability region is provided in Fig. 4.4, where it can be seen that both forward/backward reachability algorithms still provide larger estimates of the ROA compared to the ellipsoid associated with the sub-level set of the chosen Lyapunov function $V(\boldsymbol{x}) = 5/18x_1^2 - 1/9x_1x_2 + 4/9x_2^2$.

Finally, we examine the single-machine infinite bus system, see Sec. 2.5.1. The system is simplified to an ODE system based on remark 2.13, hence the nonlinear equations governing the model are

$$\begin{aligned} \dot{\delta} &= \omega_s \left(\omega - \omega_{\text{ref}}\right), \\ \dot{\omega} &= \frac{1}{H}(P_m - P_e \sin(\delta) - D\left(\omega - \omega_{\text{ref}}\right)), \end{aligned} \tag{4.18}$$

where the state variables δ and ω correspond to the generator angle and its rotational speed. The meaning and values of the constant values H, P_e, P_m, and D are discussed previously in Sec. 2.1.2. The equilibrium point is $x_s = (0.657, 0)^T$, the domain of investigation is $\boldsymbol{\mathcal{D}} := [-0.5, 2.5] \times [-3, 3]$, and the chosen Lyapunov function is $V(\boldsymbol{x}) = 3.854x_1^2 + 0.313x_1x_2 + 0.142x_2^2 + 0.019x_2$.

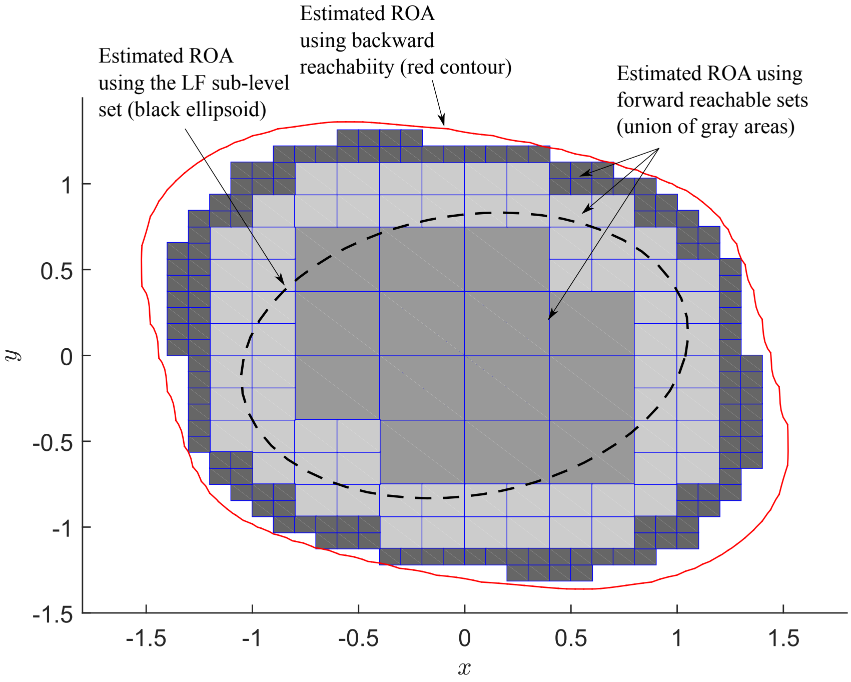


Figure 4.4: Estimation of the stability region of (4.17) using our proposed algorithm (gray cells), level set toolbox (red-solid contour), and SMRSOFT (black-dotted ellipsoid). The equilibrium point is located at the origin. The gray areas of the ROA are associated with the recursive partitioning of the domain, see remark 4.2.

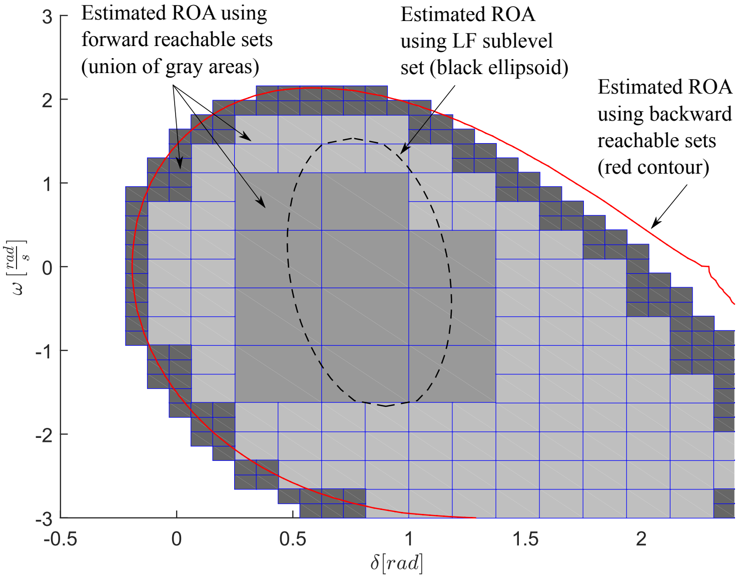


Figure 4.5: Estimation of the stability region of (4.18) using our proposed algorithm (gray cells), level set toolbox (red-solid contour), and SMRSOFT (black-dotted ellipsoid). The equilibrium point is located at $\boldsymbol{x}_s = [0.657, 0]^T$. The gray areas of the ROA are associated with the recursive partitioning of the domain, see remark 4.2.

Remark 4.3. Here the SMIB was simplified to an ODE system for comparison purposes; this is based on the fact that SMRSOFT only computes the ROA for ODEs described by a polynomial vector, whereas the forward/backward reachability algorithms implemented in CORA and the level set toolbox, respectively can compute the ROA for DAE systems as well.

The ROA is illustrated in Fig. 4.5, where both reachability algorithms significantly outperform the Lyapunov-based method in terms of accuracy. Here, additional conservatism can be observed by the estimate of the sub-level set of the Lyapunov function, due the fact that SMRSOFT can only work with systems with a polynomial vector field; thus, important system information was lost during the polynomialization of the nonlinear system. This step is not required using our algorithm as it fully considers the system nonlinearities within the set of Lagrangian remainders.

4.4.2 High-Dimensional Systems

The proposed algorithm also works in higher dimensions. The first example is a 3-dimensional system with a polynomial vector from [116]:

$$\begin{aligned} \dot{x}_1 &= -x_2, \\ \dot{x}_2 &= -x_3, \\ \dot{x}_3 &= -x_1 - 2x_2 - x_3 + x_1^3. \end{aligned} \tag{4.19}$$

The system has a stable equilibrium point at the origin, the chosen domain of investigation is $\boldsymbol{\mathcal{D}} := [-1,1] \times [-2,2] \times [-2,2]$, and the stability region is provided in Fig. 4.6.

In this example, it is clear that the accuracy of the stability region is better using our algorithm compared to backward reachability computations. We could not obtain an estimate of the ROA using the sub-level set of the Lyapunov function since the toolbox SMRSOFT did not return a feasible solution, although numerous Lyapunov functions were examined.

The final example is a 4-dimensional non-polynomial system. Here we consider the inverted pendulum on a moving cart governed by the differential equations

$$\begin{aligned} (m_c + m_p)\ddot{x}_p &= u_p + m_p\ddot{\theta}\cos(\theta) - m_p l\dot{\theta}^2\sin(\theta), \\ l\ddot{\theta} - \ddot{x}_p\cos(\theta) &= g\sin(\theta), \end{aligned} \tag{4.20}$$

where m_c, m_p are the masses of the cart and the pendulum, respectively, l is the length of the pendulum, g is the gravitational acceleration, and u_p is the force applied to the cart. The state variables are the cart position x_p, its velocity $\dot{x}_p$, the pendulum angle θ, and its rotational speed $\dot{\theta}$. The system is controlled using a linear-quadratic regulator (LQR) to stabilize the pendulum in the upright position $\theta = 0$.

In this example, we would like to identify the critical pendulum angle from which the LQR would manage to bring the pendulum to the equilibrium. Based on the practical constraints of the chosen setup, the

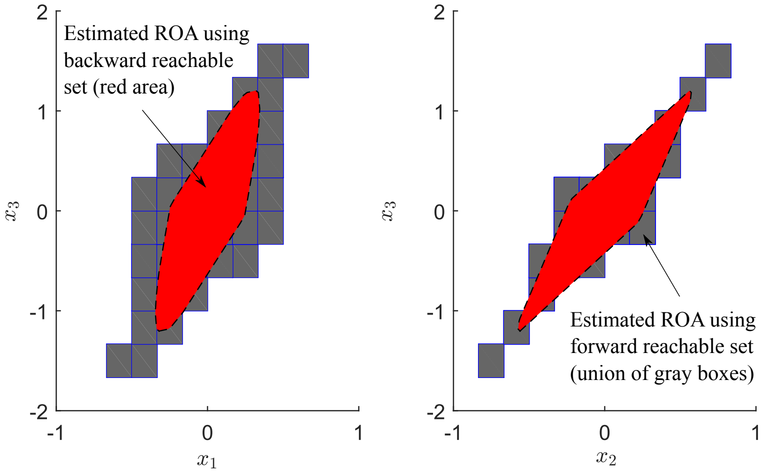


Figure 4.6: Projection of the estimation of the ROA of (4.19) using our proposed algorithm (gray areas), the level set toolbox (red area with a dotted stroke). The equilibrium point is located at the origin. The gray areas of the ROA are associated with the recursive partitioning of the domain, see remark 4.2.

domain of investigation is chosen to be $\boldsymbol{\mathcal{D}} := [-0.5, 0.5] \times [-1, 1] \times [-\pi/2, \pi/2] \times [-2, 2]$. Fig. 4.7 shows a projection of the ROA obtained using our reachability algorithm. We could not obtain the ROA using the level set toolbox, since the algorithm exceeded the maximum associated memory (16GB) of our machine at $t = 157\,\mathrm{s}$; thus, we had to abort the computation in MATLAB. Furthermore, similarly to the 3-D example, we did not manage to obtain an estimate of the ROA using SMRSOFT as the toolbox did not provide a solution, although numerous Lyapunov functions were examined.

4.4.3 Discussion

It is shown in our results that both reachability algorithms (backward and forward) provide fairly accurate and almost identical estimates of the stability region compared to the conservative ROA estimated using the sub-level set of the Lyapunov function. The computational time required to estimate the ROA for each example using our algorithm is listed in Table 4.1. It can be seen that our reachability algorithm clearly scales moderately; however, the associated memory requirements grew rapidly for backward reachable computations as seen in the 4-D example (4.20). This is due the fact that the computation of backward reachable sets requires a continuous partitioning of the grid, in order to find the viscosity solution of the Hamilton-Jacobi-Isaacs PDE formulation (4.2), i.e. exponential complexity with respect to the number of state variables. Although our algorithm suffers from an exponential complexity as well, this complexity is only related to the partitioning of the investigation domain. In other words, only the number of cells to be examined grows exponentially with the system dimensions, i.e. we have a smaller base for the computational complexity. Note that the computation of forward reachable sets using our algorithm has a polynomial complexity $\mathcal{O}(n^5)$ with respect to the number of state variables, or $\mathcal{O}(n^3)$ depending on the method on which the set of Lagrangian remainders is computed, see remark 2.11.

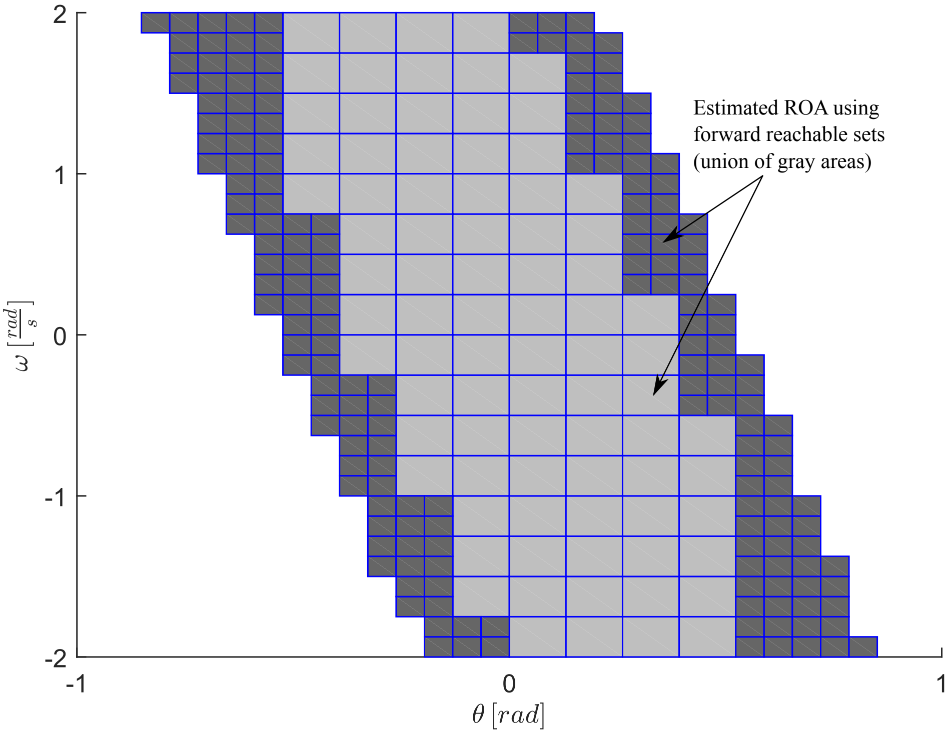


Figure 4.7: Projection of the estimation of the ROA of (4.20) using our proposed algorithm (gray cells). The equilibrium point is located at the origin. The gray areas of the ROA are associated with the recursive partitioning of the domain, see remark 4.2.

4.5 Summary

We presented an algorithm based on reachability analysis to estimate the region of attraction of an equilibrium point for nonlinear systems. Our results are compared to well-established techniques in this area, namely the computation of backward reachable sets starting from a target region using the viscosity solution of a time-dependant HJI PDE (4.2), and the optimization of the Lyapunov function sub-level set using SOS programming (4.3). Note that the intention of the presented comparison is not to replace Lyapunov-based techniques or backward reachability computations, but rather to complement them with a more rigorous stability analysis of nonlinear systems. Each technique has its own share of advantages and disadvantages. For example the Lyapunov approach is superior in terms of CPU time for low-dimensional systems; however, it provides a conservative ROA. On the contrary, backward reachability computations provide accurate estimates of the stability region, but the associated memory requirements grow rapidly with the system dimension. Furthermore, it is not a formal method, i.e. it only provides an accurate estimation rather than a provable stability region.

Our proposed approach provides accurate, provable estimates of the ROA and scales moderately with system dimensions. More importantly, the algorithm is not based on Lyapunov stability theory, i.e. it does

Table 4.1: Comparison of the CPU time.

Model	Computational time		
	CORA [9]	SMRSOFT [31]	Level set toolbox [97]
Van-der-Pol	373.27s	2.63s	476.71s
2-D example	72.84s	5.79s	62.39s
SMIB	382.71s	7.01s	97.58s
3-D example	428.3s	-	687.07s
Pendulum	571.91s	-	-

not use or require the existence of a Lyapunov function in order to provide a provable ROA. Additionally, the approach is capable of dealing with general non-polynomial systems without redefining them as a polynomial vector. Our algorithm, unfortunately, suffers as well form the curse of dimensionality. This is due the fact that it requires a partitioning of an investigation domain, hence the computational requirements associated with the discretization of the state-space will grow exponentially for large-scale systems. However, the base of this exponential problem is controllable, as we can select the size of the desired grid.

In the future, one may investigate the possibilities of unifying the aforementioned techniques under one framework which only exploits their advantages. Furthermore, to improve accuracy of the estimated ROA, one can use the so-called polynomial zonotopes, which were introduced in [7], as set representation for forward reachability computations. Polynomial zonotopes allow one to select larger initial sets, thus reducing the size of the partitioned grid, which in return would substantially reduce the overall computational costs associated with our estimation algorithm. Finally, another direction would be to combine the proposed approach with the compositional reachability algorithm for transient stability analysis of power systems as in Ch. 3. This allows one to compute provable stability margins, which are of great importance to transmission system operators to dynamically assess security of their transmission network.

Chapter 5

Formal LPV Control for Transient Stability of Power Systems

This chapter is concerned with establishing transient stability of power systems using specialized controllers. To this end, we propose the design and verification of linear-parameter varying (LPV) controllers to robustly establish transient stability with formal guarantees for multi-machine power systems. First, we transform power systems described by standard differential algebraic equations (DAEs) into modular LPV systems, such that the interaction and the correlation between different machines connected to the grid is preserved. Then, we employ reachability analysis to determine the set of the time-varying parameters which is required for the LPV controller synthesis. Afterwards, reachability analysis is also used to formally guarantee that the synthesized controller encloses the time-varying parameters within chosen parameter ranges during transients. Both tasks are solved simultaneously in a systematic fashion within the context of a unified framework. The method is demonstrated on several benchmark examples to showcase the applicability and scalability of the approach.

5.1 Introduction

So far we presented the application of reachability analysis and Lyapunov direct method for the analysis of post-fault scenarios in order to establish transient stability. This chapter, however, is primarily concerned with the improvement of the rotor angle stability which is considered by both theorists and practitioners to be one of the most challenging tasks in power systems [82]. In practice, transient stability is effectively managed using fast circuit breakers or via special controllers acting on the excitation system of the synchronous generators. The most relevant regulator employed in practice is a standard controller referred to as the power system stabilizer (PSS). The PSS basically introduces the necessary damping torque to eliminate the electrochemical oscillations during transients, see [81, pp. 761-781].

There exists numerous techniques which have been proposed for the design of the PSS, such as heuristic search algorithms [1,3,5,40], fuzzy logic and neural networks [48,53,138], and robust controllers [2,50,141]. The main drawback of the aforementioned methods, however, is that the PSS is synthesized based on a linearized model of the synchronous machine; thus, limiting effectiveness of the PSS to a small region surrounding the linearization point. This will become even more challenging in the foreseeable future due to increasingly varying operating conditions in power systems.

An alternative approach to handle system nonlinearities, input uncertainties, and parameter variations in power systems is the use of controllers synthesized based on the formulation of LPV systems. In general, LPV plants can be treated as linear time-invariant systems subject to the uncertainty of some time-varying parameters accounting for the operational range of the system. This makes it possible to tailor powerful linear controller synthesis tools for nonlinear systems, e.g. robust $\mathcal{H}_\infty$ and pole placement [17, 34]. This is done via the reformulation of the LPV control synthesis problem into a convex optimization problem involving linear objective functions and positive definiteness constraints, expressed with a set of linear matrix inequalities (LMIs), where the optimization problem can be solved following the rapid development of semi-definite programming (SDP).

Recently, the aforementioned approach started to attract considerable attention to design a set of controllers, which take into account the nonlinearities arising in synchronous generators and asynchronous machines. In [89, 90, 114], the design of a set of decentralized PSSs was proposed via the reformulation of the power system as an LPV system. In these contributions, the LPV system is not derived from the original DAE system governing dynamics of the process, instead it is obtained by linearizing around several operating points and interpolating in between via a gridding of the space of the time-varying parameters; thus, the success of this approach is depending on the underlying gridding. In [60, 61], an exact LPV model was derived for the single-machine infinite bus benchmark problem and by employing this model, the LPV controller is synthesized to establish transient stability following occurrence of a fault in the transmission network. The main disadvantage of this model is that it contains the algebraic equations of the grid; hence the synthesis procedure cannot be generalized to larger power systems due to the inflexibility of the LPV model; in other words, the LPV model lacks modularity to describe multi-machine power systems.

LPV-based techniques may as well address robustness of the resulting controller and have been already used within the context of wind energy conversion systems. However, most of the related literature is focused on minimizing fatigue loads of the blades or improving aerodynamics by damping the mechanical oscillations of the turbine[1] [121]. Furthermore, the existing LPV models describing dynamics of the wind turbine handles the electrical and mechanical parts separately, thus making the controller synthesis

[1] An illustration of the wind turbine construction and principle of operation is illustrated in Fig. 2.5.

procedure more complicated, as two sets of controllers have to be designed simultaneously in order to regulate both the electrical and mechanical components.

During our cooperation with the university of Kassel, an exact LPV model was derived for the synchronous generator and the asynchronous machine found in conventional power plants and wind turbines, respectively. The modularity of the LPV system representation proposed in [121, 122] made it possible to synthesize a set of decentralized LPV controllers that consider physical interaction between different grid nodes, and most importantly, ensure stability of the transmission network in the sense of Lyapunov. However, one aspect which was ignored during the synthesis procedure was the verification of the resulting controller; that is, the formal guarantee that the time-varying parameter will always remain within the specified space using the synthesized controller under all eventualities. Instead, the controller was examined within a simulation environment that does not provide any formal guarantees.

Motivated by the shortcomings of the standard PSS controller and the existing synthesis procedure of LPV controllers, we present a unified approach, based on reachability analysis, in which we combine synthesis and verification of LPV controllers under one framework. As previously mentioned, reachability analysis basically determines the set of states that a system can reach over a time horizon starting from a set of initial states. The proposed framework is particularly beneficial when synthesizing LPV controllers of multi-machine power systems since finding consistent parameter ranges for each generator -simultaneously in the least conservative way- can become a difficult task when not following a systematic approach.

5.2 Problem Formulation

We consider standard power systems formulated via a set of nonlinear DAEs as in (2.38)

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)),\\ \boldsymbol{0} &= \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)).\end{aligned}$$

Here power systems under consideration shall consist of G_n generating units including the synchronous generators and the doubly-fed induction generators whose mathematical models are governed by the nonlinear equations introduced earlier in Sec. 2.1.2 and 2.1.3, respectively.

With a proper choice of the time-varying parameters denoted by $\boldsymbol{\varphi}_j \in \mathbb{R}^{n_{\varphi_j}}$, any j-th subsystem corresponding to a generating unit of the DAE system can be transformed into an LPV plant; a class of nonlinear systems modelled as a parameterized linear system whose parameters changes according to the time variations of the state variables. An LPV model of the j-th generating unit can be expressed in the generalized state-space description as follows:

$$\begin{pmatrix} \dot{\tilde{\boldsymbol{x}}}_j \\ \tilde{\boldsymbol{z}}_{\infty,j} \end{pmatrix} = \left(\begin{array}{c|cc} \tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j) & \tilde{\boldsymbol{B}}_{u,j}(\boldsymbol{\varphi}_j) & \tilde{\boldsymbol{B}}_{\infty,j}(\boldsymbol{\varphi}_j) \\ \hline \tilde{\boldsymbol{C}}_{\infty,j}(\boldsymbol{\varphi}_j) & \tilde{\boldsymbol{D}}_{u,j}(\boldsymbol{\varphi}_j) & \tilde{\boldsymbol{D}}_{\infty,j}(\boldsymbol{\varphi}_j) \end{array}\right) \begin{pmatrix} \tilde{\boldsymbol{x}}_j \\ \tilde{\boldsymbol{u}}_j \\ \tilde{\boldsymbol{w}}_j \end{pmatrix}. \tag{5.1}$$

Here the vectors $\tilde{\boldsymbol{x}}_j \in \mathbb{R}^{n_{x,j}}$, and $\tilde{\boldsymbol{u}}_j \in \mathbb{R}^{n_{u,j}}$ include the state variables and the manipulated inputs of the j-th generating unit, respectively. The vectors $\tilde{\boldsymbol{z}}_j \in \mathbb{R}^{n_{z,j}}$ and $\tilde{\boldsymbol{w}}_j \in \mathbb{R}^{n_{w,j}}$ specify the controllable outputs and the exogenous inputs (e.g. disturbances and set values), correspondingly. Additionally the time-varying matrices $\tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j)$, $\tilde{\boldsymbol{B}}_{u,j}(\boldsymbol{\varphi}_j)$, $\tilde{\boldsymbol{B}}_{\infty,j}(\boldsymbol{\varphi}_j)$, $\tilde{\boldsymbol{C}}_{\infty,j}(\boldsymbol{\varphi}_j)$, and $\tilde{\boldsymbol{D}}_{u_j,j}(\boldsymbol{\varphi}_j)$, $\tilde{\boldsymbol{D}}_{\infty,j}(\boldsymbol{\varphi}_j)$ denote the system, the input, the output and the feed-forward matrices, respectively. Note that we distinguish between the variables of the original DAE system and those of the LPV plant using the tilde. In this chapter we seek to formally design and verify an LPV controller that employs a state-feedback scheme in the closed-loop as illustrated in Fig. 5.1 for the synchronous generator.

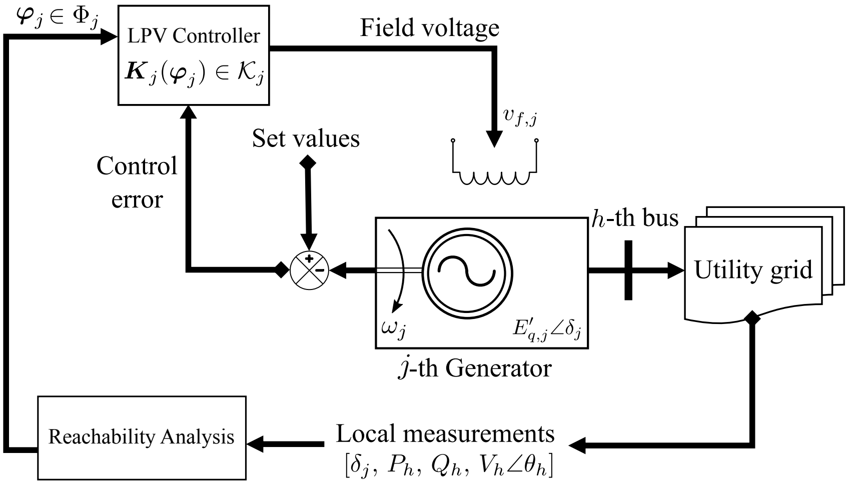


Figure 5.1: Simplified diagram of the proposed LPV controller for the synchronous generator. In this illustration, the j-th synchronous generator, modelled as in (2.10)-(2.15), is controlled via a discrete set of state-feedback controllers denoted by $\boldsymbol{K}_j(\boldsymbol{\varphi}_j) \in \boldsymbol{\mathcal{K}}_j$, as presented shortly in Sec. 5.3. The controller gain generates the control signal of the field voltage v_f, depending on some time-varying parameters $\boldsymbol{\varphi}_j \in \boldsymbol{\Phi}_j$ (see Sec. 5.3.1). These parameters are obtained using reachability analysis as illustrated in Sec. 5.4. The interaction at the h-th bus with other machines connected to the grid is preserved by considering local measurements at each bus.

Hereafter, the LPV controller state-feedback matrix of the j-th generating unit shall be denoted by $\tilde{\boldsymbol{K}}_j(\boldsymbol{\varphi}_j) \in \tilde{\boldsymbol{\mathcal{K}}}_j$. Our objective is to synthesize a set of decentralized LPV controllers capable of robustly establishing transient stability of multi-machine power systems in real-time control, and most importantly, to formally guarantee that the family of controllers meet the control performance specified for pole placement and $\mathcal{H}_\infty$ design under all eventualities; that is, the controllers should adhere to the following specifications:

(S.1) The state-feedback controller employed in the closed-loop of each j-th generating unit, described as an LPV plant, should confine the poles within a pre-defined region of the complex plane.

(S.2) The state-feedback controller should guarantee minimization of the $\mathcal{H}_\infty$ performance; that is to minimize the infinity norm of the closed-loop transfer function from the exogenous inputs $\tilde{\boldsymbol{w}}_j$ to the vector of controllable outputs $\tilde{\boldsymbol{z}}_{\infty,j}$.

In order to verify the controller specifications and guarantee that the time-varying parameters are always confined within the chosen parameters ranges, we propose a new approach, based on reachability analysis, which combines in one framework synthesis and verification of LPV controllers. The proposed framework consists of the following steps:

1 Transform the nonlinear model governing dynamics of the j-th generating unit to an LPV plant (Sec. 5.3.1).

2 Examine transient stability analysis of the uncontrolled DAE system using reachability analysis to obtain an initial guess of the time-varying parameters $\boldsymbol{\Phi}_j$ from computed reachable sets of differential and algebraic variables (Sec. 5.4).

3.a Synthesize a set of state-space controllers that account for trajectories of the time-varying parameters by formulating a set of LMIs specified for pole placement and $\mathcal{H}_\infty$ design (Sec. 5.3.4).

3.b Express the LPV controller analytically based on the synthesized set of state-space controllers using closed-form expressions of convex combinations (Sec. 5.3.5).

4 Re-compute the set of the time-varying parameters $\boldsymbol{\Phi}_j$ based on reachable sets of the DAE system, where the synthesized LPV controller is used in the closed-loop (Sec. 5.4).

Remark 5.1. The final step 4 is important as it reduces conservatism of the resulting LPV controller by obtaining a tighter set of the time-varying parameters $\boldsymbol{\Phi}_j$ required for the synthesis procedure.

5.3 LPV Controller Synthesis

We consider a specific class of LPV systems for our particular control specifications; namely, the so-called *polytopic* LPV plants based on the terminology proposed in [17]. Polytopic LPVs are generally suitable for controller design specified for pole placement and robust $\mathcal{H}_\infty$.

There are several assumptions encompassing many practical situations with regards to this class of plants:

(A.1) The feed-forward matrix $\tilde{\boldsymbol{D}}_{u,j}$ is set to zero; this is not regarded as a hard restriction for our particular problem due to the low-pass characteristic of the synchronous generator and the doubly-fed induction generator.

(A.2) The transformation of the nonlinear system into the LPV description should result in an LPV plant whose matrices $\tilde{\boldsymbol{B}}_{\infty,j}$, $\tilde{\boldsymbol{B}}_{u_j,j}$, $\tilde{\boldsymbol{C}}_{\infty,j}$, and $\tilde{\boldsymbol{D}}_{\infty,j}$ are parameter independent.

(A.3) The time-varying parameters $\boldsymbol{\varphi}_j$ are assumed to vary within a convex polytope denoted by $\boldsymbol{\Phi}_j$ with $v_{\varphi,j}$ vertices, see Def. 2.2. Furthermore, the system matrix $\tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j)$ of the LPV system depend affinely on the time-varying parameters.

(A.4) The final and most important assumption is that the pair $\left(\tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j),\, \tilde{\boldsymbol{B}}_{u,j}\right)$ is controllable (stablizable) over the complete space spanned by the time-varying parameters.

The outcome of assumptions **(A.1)** and **(A.2)** is fairly straightforward resulting in a simplification of the generalized LPV plant (5.1) into the following polytopic LPV system:

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}}_j &= \tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j(t))\tilde{\boldsymbol{x}}_j + \tilde{\boldsymbol{B}}_{u,j}\tilde{\boldsymbol{u}}_j + \tilde{\boldsymbol{B}}_{\infty,j}\tilde{\boldsymbol{w}}_j, \\ \tilde{\boldsymbol{z}}_j &= \tilde{\boldsymbol{C}}_{\infty,j}\tilde{\boldsymbol{x}}_j + \tilde{\boldsymbol{D}}_{\infty,j}\tilde{\boldsymbol{w}}_j, \end{aligned} \tag{5.2}$$

and the assumption **(A.3)** basically means that the time-varying matrix $\tilde{\boldsymbol{A}}_j$, and likewise $\tilde{\boldsymbol{K}}_j$, are bound inside matrix polytopes defined by:

$$\begin{aligned} \tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j) \in \tilde{\boldsymbol{\mathcal{A}}}_j &:= \left\{ \sum_{i=1}^{v_{\varphi,j}} \lambda_{i,j}(\boldsymbol{\varphi}_j)\tilde{\boldsymbol{A}}_j^{(i)} \;:\; \lambda_{i,j}(\boldsymbol{\varphi}_j) \geq 0,\, \sum_{i=1}^{v_{\varphi,j}} \lambda_{i,j}(\boldsymbol{\varphi}_j) = 1 \right\}, \\ \tilde{\boldsymbol{K}}_j(\boldsymbol{\varphi}_j) \in \tilde{\boldsymbol{\mathcal{K}}}_j &:= \left\{ \sum_{i=1}^{v_{\varphi,j}} \lambda_{i,j}(\boldsymbol{\varphi}_j)\tilde{\boldsymbol{K}}_j^{(i)} \;:\; \lambda_{i,j}(\boldsymbol{\varphi}_j) \geq 0,\, \sum_{i=1}^{v_{\varphi,j}} \lambda_{i,j}(\boldsymbol{\varphi}_j) = 1 \right\}. \end{aligned} \tag{5.3}$$

Notice that the definition of the matrix polytope $\tilde{\boldsymbol{\mathcal{A}}}_j$, and likewise $\tilde{\boldsymbol{\mathcal{K}}}_j$ is analogous to the definition of convex polytopes, see Def. 2.2. In (5.3), the vertices of the matrix polytope corresponds to a finite number of matrices $\tilde{\boldsymbol{A}}_j^{(i)}$, and $\tilde{\boldsymbol{K}}_j^{(i)}$ with the same dimensions.

The final assumption **(A.4)** presumes that there exists a state-feedback matrix in the control law

$$\tilde{\boldsymbol{u}}_j = -\tilde{\boldsymbol{K}}_j(\boldsymbol{\varphi}_j)\tilde{\boldsymbol{x}}_j, \tag{5.4}$$

that guarantees stability of the closed-loop LPV system:

$$\dot{\tilde{\boldsymbol{x}}}_j = \underbrace{\tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j) - \tilde{\boldsymbol{B}}_{u,j}\tilde{\boldsymbol{K}}_j(\boldsymbol{\varphi}_j)}_{=:\tilde{\boldsymbol{A}}_{j,cl}(\boldsymbol{\varphi}_j)}\tilde{\boldsymbol{x}}_j + \tilde{\boldsymbol{B}}_{\infty,j}\tilde{\boldsymbol{w}}_j. \tag{5.5}$$

Here $\tilde{\boldsymbol{A}}_{j,cl}(\boldsymbol{\varphi}_j)$ corresponds to the system matrix of the LPV system employing $\tilde{\boldsymbol{K}}_j(\boldsymbol{\varphi}_j)$ in the closed-loop. It is worth mentioning that **(A.4)** is equivalent to the assumption that there exists a positive definite matrix denoted by $\boldsymbol{M}_j > 0$ that satisfies the Lyapunov inequality [23]:

$$\forall i \in \{1,\, \dots,\, v_{\varphi,j}\}: \quad \tilde{\boldsymbol{A}}_{j,cl}^{(i)}\boldsymbol{M}_j + \boldsymbol{M}_j\tilde{\boldsymbol{A}}_{j,cl}^{(i);T} < 0,\; \boldsymbol{M}_j > 0. \tag{5.6}$$

In the following, we will present the modular LPV models which describe dynamics of the synchronous generator and the doubly-fed induction generator. Then we will discuss the LMI-based characterization for pole placement and control performance according to $\mathcal{H}_\infty$ specifications.

5.3.1 Modelling of LPV Power Systems

Now we introduce the LPV models of the synchronous generator and the doubly-fed induction generator according to the exact analytical transformation suggested in [121, 122]. These models are particularly suitable to design a set of decentralized controllers capable of robustly establishing stability of the voltage and the rotor angle. It worth mentioning that the transformation of a nonlinear model to the standard LPV state-space compact form as in (5.2) is generally not trivial and not unique.

5.3.1.1 LPV Model of the Synchronous Generator

Recall that the mathematical model governing dynamics of the j-th synchronous generator is expressed via the differential equations (2.10) and (2.11), in addition to the algebraic equations (2.12)-(2.15). Inserting expressions of the machine voltages v_d and v_q into the formula describing the electrical torque T_e, then using the result in the differential equation describing the rotational speed ω, yields the following LPV description (see [122]):

$$\begin{pmatrix} \dot{\delta}_j \\ \dot{\omega}_j \\ \dot{E}'_{q,j} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \omega_s & 0 \\ \varphi_{1,j} & -D_j & \varphi_{2,j} \\ \varphi_{3,j} & 0 & -\frac{1}{\tau'_{d,j}} \end{pmatrix}}_{=:\tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j)} \tilde{\boldsymbol{x}}_j + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau'_{d,j}} \end{pmatrix}}_{=:\tilde{\boldsymbol{B}}_{u,j}} v_{f,j}, \tag{5.7}$$

$$\text{with: } \begin{cases} \varphi_{1,j} = \dfrac{P_{m,j} - (X_{q,j} - X'_{d,j}) i_{d,j} i_{q,j}}{2H_j \delta_j}, \\ \varphi_{2,j} = -\dfrac{i_{q,j}}{2H_j}, \quad \varphi_{3,j} = \dfrac{i_{d,j}(X'_{d,j} - X_{d,j})}{\tau'_{d,j}\delta_j}. \end{cases} \tag{5.8}$$

Note that the power P_m is kept constant and is included within the parameter $\varphi_{1,j}$; this is easily justified for studies involving transient stability [81, Ch. 13].

Remark 5.2. Here it should be stressed that (5.7) is an exact analytical reformulation and is not a linearization of (2.10)-(2.15). This transformation allows one to set up modular models of the generators to synthesize and verify each machine separately, and more importantly, the transformation preserves the interaction with the grid, which is critical for studies involving transient stability analysis of multi-machine power system models.

Remark 5.3. The main restriction of the synchronous generator's LPV model is that the angular position δ_j should not be equal to zero, otherwise the parameters $\varphi_{1,j}$ and $\varphi_{3,j}$ would be mathematically undefined as they involve a division by zero. While this limitation might pose theoretical limitations on the system, it is not relevant for our practical considerations; this is due to the fact, that the rotor angular position is always greater than zero when being connected to the utility grid, i.e. the machine is rotating and producing electrical power.

5.3.1.2 LPV Model of the Doubly-fed Induction Generator

Now we consider the LPV model of the doubly-fed induction generator based on the model governed by the set of DAEs formalized via the differential equations (2.17) and (2.18), in addition to the algebraic equations (2.19)-(2.23). According to the derivation presented in [121, Sec. 3.2] the LPV-model of the DFIG-based wind energy conversion system is summed up to:

$$\begin{pmatrix} \dot{\omega}_{r,j} \\ \dot{\psi}_{r,d,j} \\ \dot{\psi}_{r,q,j} \end{pmatrix} = \underbrace{\begin{pmatrix} \varphi_{1,j} & \varphi_{2,j} & \varphi_{3,j} \\ \varphi_{4,j} & -\frac{r_{r,j}}{X_{\nu,j}} & 0 \\ \varphi_{5,j} & 0 & -\frac{r_{r,j}}{X_{\nu,j}} \end{pmatrix}}_{=:\tilde{\boldsymbol{A}}_j(\boldsymbol{\varphi}_j)} \tilde{\boldsymbol{x}}_j + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}}_{=:\tilde{\boldsymbol{B}}_{u,j}} \begin{pmatrix} v_{r,d} \\ v_{r,q} \end{pmatrix}, \tag{5.9}$$

$$\text{with: } \begin{cases} \varphi_{1,j} = \dfrac{T_{m,j}}{2H_{w,j}\omega_{r,j}}, \quad \varphi_{2,j} = -\dfrac{X_{\nu,j} i_{s,q,j}}{2X_{r,\nu,j}H_{w,j}}, \quad \varphi_{3,j} = \dfrac{X_{\nu,j} i_{s,d,j}}{2X_{r,\nu,j}H_{w,j}}, \\ \varphi_{4,j} = \dfrac{\psi_{r,q,j}X_{\nu,j}\left(1-\omega_{r,j}\right) - r_{r,j}X_{\nu,j}i_{s,d,j}}{\omega_{r,j}}, \\ \varphi_{5,j} = \dfrac{\psi_{r,d,j}X_{\nu,j}\left(\omega_{r,j}-1\right) - r_{r,j}X_{\nu,j}i_{s,q,j}}{\omega_{r,j}}. \end{cases} \tag{5.10}$$

Remark 5.4. Similarly remark 5.3, the main restriction of the LPV model of the doubly-fed induction generator is that the angular velocity ω_r should not be equal to zero, otherwise the parameters $\varphi_{1,j}$, $\varphi_{4,j}$, and $\varphi_{5,j}$ would be mathematically undefined as they involve a division by zero. It is obvious that this condition does not occur in real-time control as the value 0 implies that the rotor of the wind turbine is in standstill, i.e. not operational. The angular velocity can be close to zero only at start-up of the wind turbine, however, this is handled using a separate controller. Therefore, the applicability of the model is not compromised.

With the introduction of the exact LPV models describing dynamics of the generating units, we can now describe the LMI-based characterization for pole placement and robust $\mathcal{H}_\infty$ design.

5.3.2 LMI Formulation for Pole Placement Design

As specified earlier in **(S.1)**, we seek to find a discrete set of state-feedback controllers that place the poles of the closed-loop system matrix $\tilde{\boldsymbol{A}}_{j,cl}(\boldsymbol{\varphi}_j)$ within a specified region of the complex plane. It is well known that the transient response of a linear system is directly associated with the location of the poles in the complex plane; hence, one for example can enforce sufficient damping on the system oscillations by placing the poles in a pre-defined region.

In this chapter, our region of interest illustrated in Fig. 5.2 is expressed via the set [34]:

$$\mathcal{Q}_j\left(\alpha_j, \vartheta_j, R_j\right) := \left\{ s_j \in \mathbb{C} \,:\, s_j = \sigma_j + j\Omega_j,\, \sigma_j < -\alpha_j < 0,\, \lvert\, \sigma_j + j\Omega_j \,\rvert < R_j,\, \sigma_j \tan\vartheta_j < -\lvert\, \Omega_j \,\rvert \right\}.$$

The region specified by $\mathcal{Q}$ is the intersection of three different shapes; namely a disk with radius R, an arc whose angle is ϑ, and the left-half of the complex starting at a distance α from the imaginary-axis. Placing the poles of the closed-loop within this region guarantees a minimum decay rate α, a minimum damping ratio $\cos(\vartheta)$, and a maximum undamped natural frequency $R\sin(\vartheta)$. This in turn bounds the maximum overshoot, the frequency of the oscillatory modes, the delay, rise, and settling time of the LPV dynamic response, respectively.

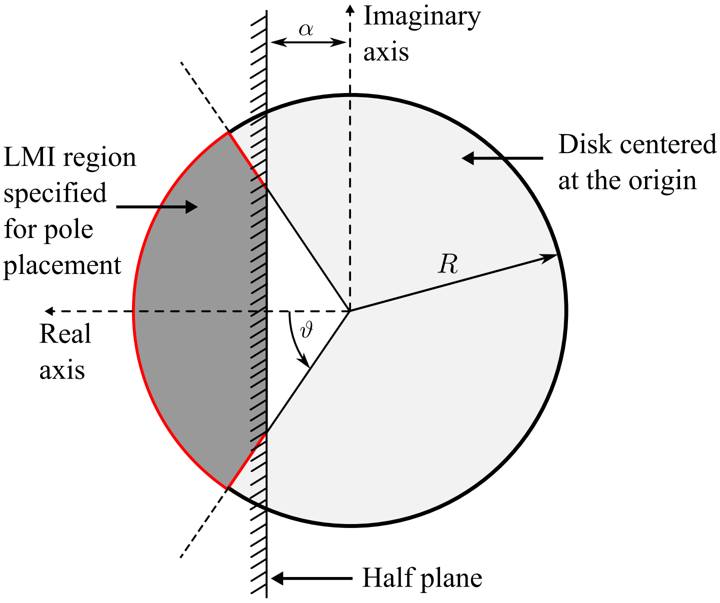

α
Imaginary axis
Disk centered at the origin
LMI region specified for pole placement
R
Real axis
ϑ
Half plane

Figure 5.2: LMI region specified for pole placement design. The region is the intersection the disk whose radius is R, the arc whose angle is ϑ, and the left-half of the complex starting at a distance α from the imaginary-axis.

Now we need to establish stability of the region $\mathcal{Q}$ via the characterization of a Lyapunov inequality as in (5.6), in order to specify a set of linear matrix inequalities for pole placement design.

Theorem 5.1. **$\mathcal{D}$-Stability** [34, Th. 2.2]: The closed-loop of the LPV system (5.5) is said to be $\mathcal{D}$-stable within any arbitrary subregion $\mathcal{D} \subset \mathbb{C}$ of the complex plane, if, and only if there exists a symmetric positive definite matrix $\boldsymbol{M}_j > 0$ satisfying the following LMI:

$$\forall i \in \{1, \ldots, v_{\varphi,j}\} : \quad \boldsymbol{\eta}_j \otimes \boldsymbol{M}_j + \boldsymbol{\mu}_j \otimes \left(\tilde{\boldsymbol{A}}_{j,cl}^{(i)} \boldsymbol{M}_j\right) + \boldsymbol{\mu}_j^T \otimes \left(\tilde{\boldsymbol{A}}_{j,cl}^{(i)} \boldsymbol{M}_j\right)^T < 0,\ \ \boldsymbol{M}_j > 0. \qquad \square$$

Here $\boldsymbol{\eta}_j$ and $\boldsymbol{\mu}_j$ corresponds to symmetric matrices with the same dimensions as $\boldsymbol{M}_j$ and the operator $\otimes$ returns the Kronecker product of matrices, see [26]. Notice that Th. 5.1 simplifies to the general Lyapunov inequality (5.6) by setting $\boldsymbol{\eta}_j$ to zero and $\boldsymbol{\mu}_j$ to the identify matrix. The generalization of Th. 5.1, subject to each vertex $i \in \{1, \ldots, v_{\varphi,j}\}$ of the LPV plant (5.5) yields the following set of LMIs [34]:

$$\textbf{Disc:} \quad \begin{pmatrix} -R\boldsymbol{M}_j & \tilde{\boldsymbol{A}}^{(i)}_{j,cl}\boldsymbol{M}_j \\ \boldsymbol{M}_j\tilde{A}^{(i),T}_{j,cl} & -R\boldsymbol{M}_j \end{pmatrix} < 0, \tag{5.11}$$

$$\textbf{Arc:} \quad \begin{pmatrix} \sin\vartheta\left(\tilde{\boldsymbol{A}}^{(i)}_{j,cl}\boldsymbol{M}_j + \boldsymbol{M}_j\tilde{\boldsymbol{A}}^{(i),T}_{j,cl}\right) & \cos\vartheta\left(\tilde{\boldsymbol{A}}^{(i)}_{j,cl}\boldsymbol{M}_j - \boldsymbol{M}_j\tilde{\boldsymbol{A}}^{(i),T}_{j,cl}\right) \\ \cos\vartheta\left(M_j\tilde{\boldsymbol{A}}^{(i),T}_{j,cl} - \tilde{\boldsymbol{A}}^{(i)}_{j,cl}\boldsymbol{M}_j\right) & \sin\vartheta\left(\tilde{\boldsymbol{A}}^{(i)}_{j,cl}\boldsymbol{M}_j + \boldsymbol{M}_j\tilde{\boldsymbol{A}}^{(i),T}_{j,cl}\right) \end{pmatrix} < 0, \tag{5.12}$$

$$\textbf{Half-plane:} \quad 2\alpha\boldsymbol{M}_j + \tilde{\boldsymbol{A}}^{(i)}_{j,cl}\boldsymbol{M}_j + \boldsymbol{M}_j\tilde{\boldsymbol{A}}^{(i),T}_{j,cl} < 0. \tag{5.13}$$

5.3.3 LMI Formulation for $\mathcal{H}_\infty$ Design

Now we shall consider the second specification **(S.2)** associated with the LPV controller. We seek to guarantee the $\mathcal{H}_\infty$ performance of the transfer function mapping the controllable outputs to the exogenous inputs. More specifically, the controller addresses the problem concerned with the minimization of the infinity norm such that:

$$||\, T_{\tilde{w}_j\tilde{z}_{\infty,j}}(\boldsymbol{s}_j)\, ||_\infty < \gamma_j,$$

with $T_{\tilde{w}_j\tilde{z}_{\infty,j}}(\boldsymbol{s}_j)$ denoting the transfer function from $\tilde{\boldsymbol{w}}_j$ to $\tilde{\boldsymbol{z}}_{\infty,j}$, $||\cdot||_\infty$ returns the infinity norm, and the positive scalar $\gamma_j > 0$ specifies the bound of the $\mathcal{H}_\infty$ performance.

To meet the $\mathcal{H}_\infty$ performance of the closed-loop of the LPV system, the problem is reformulated to finding the positive definite matrix $\boldsymbol{M}_{\infty,j}$ that satisfies the following LMI:

$$\begin{pmatrix} \tilde{\boldsymbol{A}}^{(i)}_{cl,j}\boldsymbol{M}_{\infty,j} + \boldsymbol{M}_{\infty,j}\tilde{\boldsymbol{A}}^{(i)T}_{cl,j} & \tilde{\boldsymbol{B}}^T_{\infty,j} & \boldsymbol{M}_{\infty,j}\tilde{\boldsymbol{C}}^T_{\infty,j} \\ \tilde{\boldsymbol{B}}^T_{\infty,j} & -\gamma_j\boldsymbol{I} & \tilde{\boldsymbol{D}}_{\infty,j} \\ \boldsymbol{M}_{\infty,j}\tilde{\boldsymbol{C}}^T_{\infty,j} & \tilde{\boldsymbol{D}}_{\infty,j} & -\gamma_j\boldsymbol{I} \end{pmatrix} < 0, \tag{5.14}$$

with $\boldsymbol{I}$ being the identify matrix with proper dimensioning. Note that (5.14) is based on the well-known bounded-real Lemma, see [23, p. 142].

5.3.4 Multi-objective Design

So far we only presented the set of LMIs specified separately for pole placement and $\mathcal{H}_\infty$ control design. It still remains to formalize a semi-definite program (SDP) of the optimization problem associated with the multi-objective design of the LPV controller. The sought SDP is formulated according to the following steps:

① ***Computing vertices of the matrix polytope:*** The first step is to identify the convex polytope $\boldsymbol{\Phi}_j$ enclosing the time-varying parameters for each j-th generating unit; this in turn specifies the $v_{\varphi,j}$ vertices of the matrix polytope $\tilde{\mathcal{A}}_j$ which are required for the controller synthesis. Note that in other work, the identification of the set $\boldsymbol{\Phi}_j$ is always assumed to lie within some limits specified by the user, however, we will present later in section 5.4 a systematic procedure based on reachability analysis to identify $\boldsymbol{\Phi}_j$ in the least conservative way.

② ***Enforcing a single positive definite matrix*** Notice that combining the LMIs (5.11)-(5.14) requires the existence of two positive definite matrices; namely $\boldsymbol{M}_j$ for pole placement and $\boldsymbol{M}_{\infty,j}$ for robust control. Generally, the optimization over the variables $\left(\boldsymbol{M}_j,\, \boldsymbol{M}_{\infty,j},\, \tilde{\boldsymbol{A}}_{j,cl}^{(i)}\right)$ for all $i \in \{1,\, \dots,\, v_{\varphi,j}\}$ is not jointly convex. Convexity can be enforced, however, by seeking a common solution, i.e. satisfy the constraints via a single matrix, i.e. $\boldsymbol{M}_j \overset{!}{=} \boldsymbol{M}_{\infty,j}$.

③ ***Relax the constraint conditions*** The multiplication of the controller vertices $\tilde{\boldsymbol{K}}_j^{(i)}$ with the positive definite matrix $\boldsymbol{M}_\infty$, and likewise $\boldsymbol{M}_j$, would result in a non-convex optimization problem over the variables $\left(\boldsymbol{M}_{\infty,j},\, \tilde{\boldsymbol{A}}_{j,cl}^{(i)}\right)$; hence, we introduce the auxiliary variable $\boldsymbol{Y}_j^{(i)} := \tilde{\boldsymbol{K}}_j^{(i)} \boldsymbol{M}_{\infty,j}$ and rewrite (5.11)-(5.14) in order to readily restore convexity of the sought optimization problem, for more details see [23, pp. 101-102].

Hereafter, we introduce another auxiliary variable $\tilde{\boldsymbol{T}}_j^{(i)} := \tilde{\boldsymbol{A}}_j^{(i)} \boldsymbol{M}_{\infty,j} + \tilde{\boldsymbol{B}}_{u,j}^T \tilde{\boldsymbol{Y}}_j^{(i)}$. Combining all previous steps, the synthesis of the LPV controller is equivalent to solving the following semi-definite program:

$$
\begin{gathered}
\underset{\boldsymbol{M}_{\infty,j},\, \tilde{\boldsymbol{Y}}_j^{(i)}}{\text{minimize}}\ \gamma_j, \quad \text{s.t. } \forall i \in \{1, \dots, v_{\varphi,j}\}: \\
\left.\begin{aligned}
\boldsymbol{M}_{\infty,j} &> 0, \\
2\alpha \boldsymbol{M}_{\infty,j} + \tilde{\boldsymbol{T}}_j + \boldsymbol{T}_j^T &< 0,
\end{aligned}\right\} \textbf{Stability} \\
\left.\begin{aligned}
\begin{pmatrix} -R\boldsymbol{M}_{\infty,j}, & \tilde{\boldsymbol{T}}_j \\ \tilde{\boldsymbol{T}}_j^T & -R\boldsymbol{M}_{\infty,j} \end{pmatrix} &< 0, \\
\begin{pmatrix} \sin\vartheta \left(\tilde{\boldsymbol{T}}_j + \tilde{\boldsymbol{T}}_j^T\right) & \cos\vartheta \left(\tilde{\boldsymbol{T}}_j - \tilde{\boldsymbol{T}}_j^T\right) \\ \cos\vartheta \left(\tilde{\boldsymbol{T}}_j^T - \tilde{\boldsymbol{T}}_j\right) & \sin\vartheta \left(\tilde{\boldsymbol{T}}_j + \tilde{\boldsymbol{T}}_j\right) \end{pmatrix} &< 0,
\end{aligned}\right\} \textbf{Damping} \\
\left.\begin{pmatrix} \tilde{\boldsymbol{T}}_j + \tilde{\boldsymbol{T}}_j^T & \tilde{\boldsymbol{B}}_{\infty,j}^T & \boldsymbol{M}_{\infty,j} \tilde{\boldsymbol{C}}_{\infty,j}^T \\ \tilde{\boldsymbol{B}}_{\infty,j}^T & -\gamma_j \boldsymbol{I} & \tilde{\boldsymbol{D}}_{j,\infty} \\ \boldsymbol{M}_{\infty,j} \tilde{\boldsymbol{C}}_{\infty,j}^T & \tilde{\boldsymbol{D}}_{j,\infty} & -\gamma_j \boldsymbol{I} \end{pmatrix} < 0. \quad \right\} \textbf{Robustness}
\end{gathered}
\tag{5.15}
$$

If (5.15) is feasible for all $i \in \{1,\, \dots,\, v_{\varphi,j}\}$, this implies a suboptimal solution is found for the sought minimization problem at each i-th vertex of the matrix polytope $\tilde{\mathcal{A}}_j$; in other words, there exists a state-feedback controller that places the poles of the closed-loop in the LMI region $\mathcal{Q}\left(\alpha_j, \vartheta_j, R_j\right)$ while simultaneously guaranteing robustness with an infinity norm of less than γ_j.

Let $\left(\boldsymbol{M}_{\infty,j}^*,\, \tilde{\boldsymbol{Y}}_j^{(i),*}\right)$ denote the suboptimal solution of (5.15). Notice that the semi-definite program returns the positive definite matrix and the axillary variable $\tilde{\boldsymbol{Y}}_j^{(i),*}$ introduced earlier to enforce convexity of the optimization problem; hence the vertices of the state feedback matrix are given by:

$$\forall i \in \{1, \ldots, v_{\varphi,j}\} : \ \tilde{\boldsymbol{K}}_j^{(i)} = \boldsymbol{Y}_j^{(i),*} \boldsymbol{M}_{\infty,j}^{*,-1}. \tag{5.16}$$

5.3.5 Realization of the LPV Controller

Although the synthesis procedure of the LPV controller is performed off-line, the controller gains are not known explicitly in real-time since the semi-definite program formulated in the previous section only returns the controller vertices, see (5.16). The controller gains, however, may be expressed in terms of the time-varying parameters using convex combinations due to the affine dependency on the parameters, thus

$$\tilde{\boldsymbol{K}}_j(\varphi_j) = \sum_{i=1}^{v_{\varphi,j}} \lambda_{i,j}(\boldsymbol{\varphi}_j) \tilde{\boldsymbol{K}}_j^{(i)}. \tag{5.17}$$

The problem with (5.17) is that a convex decomposition problem has to be solved online at each time step; that is, at every instance, we want to find $\boldsymbol{\lambda}_j(\boldsymbol{\varphi}_j)$ for a given parameter $\boldsymbol{\varphi}_j(t)$. This issue can be tackled online using one of two alternatives: Linear programming or closed-form expression of convex combinations, as we proposed in [123]

The linear program used to obtain the parameters of the convex combination is concerned with finding a solution to the optimization problem specified as follows:

$$\forall t \in [0, \infty[: \quad \min_{\boldsymbol{\lambda}_j} \boldsymbol{c}_j^T \boldsymbol{\lambda}_j\left(\boldsymbol{\varphi}_j(t)\right) \quad \text{s.t.} : \begin{cases} \sum_{i=1}^{v_{\varphi,j}} \lambda_{i,j}(\boldsymbol{\varphi}_j(t)) \boldsymbol{\varphi}_j^{(i)} = \boldsymbol{\varphi}_j(t), \\ \sum_{i=1}^{v_{\varphi,j}} \lambda_{i,j}(\boldsymbol{\varphi}_j(t)) = 1, \\ \lambda_{i,j}(\boldsymbol{\varphi}_j(t)) \geq 0 \end{cases} \tag{5.18}$$

with $n_{\varphi,j} + 1$ equalities, $v_{\varphi,j}$ inequalities, $v_{\varphi,j}$ unknowns, and $\boldsymbol{c}_j$ being the objective function of the optimization problem. Notice that this optimization problem has to be solved at each instance for all the G_n generating units employing the LPV controller. For high-dimensional polytopes, whose number of extreme points grows exponentially with the system dimension, see Table 2.3. This results in a large number of inequalities to be considered, for which finding the solution can become time consuming for real-time applications.

In our previous contribution [123], we proposed an alternative solution to identifying the coefficient of the convex combination $\boldsymbol{\lambda}_j(\boldsymbol{\varphi}_j(t))$, $t > 0$. Basically, we derive an analytical expression that describes the convex combination (5.17) in real-time. In Sec. 5.4.1, we will illustrate using a systematic procedure how to enclose the time-varying parameters using multi-dimensional intervals which are expressed via

$$\begin{aligned} \boldsymbol{\Phi}_j &\overset{\text{Def.2.5}}{=} \mathcal{I}_{1,j} \times \mathcal{I}_{2,j} \times \cdots \times \mathcal{I}_{n_{\varphi,j},j}, \\ \forall k \in \{1 \ldots n_{\varphi,j}\} : \quad \mathcal{I}_{k,j} &= \left[\underline{\varphi}_{k,j}, \overline{\varphi}_{k,j}\right] := \left\{\varphi_{k,j} \in \mathbb{R} : \underline{\varphi}_{k,j} \leq \varphi_{k,j} \leq \overline{\varphi}_{k,j}\right\}. \end{aligned} \tag{5.19}$$

Here $\underline{\varphi}_{k,j}$ and $\overline{\varphi}_{k,j}$ denote the lower and upper bound of time-varying parameters, correspondingly, and $\mathcal{I}_{k,j}$ is the interval of the parameters in the k-th dimension for the j-th generating unit. Note that for multi-dimensional intervals, the number of vertices is uniquely defined by $v_{\varphi,j} := 2^{n_{\varphi,j}}$, thus the dependency of the coefficients $\boldsymbol{\lambda}_j$ on $\boldsymbol{\varphi}_j$ is described by the following closed-form expression [123]:

$$t > 0,\, \forall i \in \{1,\, \dots,\, 2^{n_{\varphi,j}}\}: \quad \lambda_{i,j}(\boldsymbol{\varphi}_j(t)) = \prod_{m=1}^{n_{\varphi,j}} \overline{\nu}_{i,m}(\boldsymbol{\varphi}_j(t)), \tag{5.20}$$

$$\text{with:} \begin{cases} \overline{\nu}_{i,m}(\boldsymbol{\varphi}_j) = \begin{cases} \nu_m(\boldsymbol{\varphi}_j(t)) & \text{if } \hat{\boldsymbol{\varphi}}_{i,j}^{(m)} = \underline{\boldsymbol{\varphi}}_j^{(m)}, \\ 1 - \nu_m(\boldsymbol{\varphi}_j(t)) & \text{if } \hat{\boldsymbol{\varphi}}_{i,j}^{(m)} = \overline{\boldsymbol{\varphi}}_j^{(m)}, \end{cases} \\ \nu_m(\boldsymbol{\varphi}_j(t)) = \dfrac{\boldsymbol{\varphi}_j^{(m)} - \overline{\boldsymbol{\varphi}_j}^{(m)}}{\underline{\boldsymbol{\varphi}}_j^{(m)} - \overline{\boldsymbol{\varphi}}_j^{(m)}}, \end{cases} \tag{5.21}$$

with i and m being the i-th entry and the m-th dimension, respectively, and $\hat{\boldsymbol{\varphi}}_i \in \mathbb{R}^{n_\varphi}$, $i \in \{1, \dots, 2^{n_\varphi}\}$ denoting the vertices of $\boldsymbol{\Phi}_j$ described as in (5.19).

5.4 Reachability Analysis

So far we have presented the synthesis procedure of LPV controllers under the assumption that the set of time-varying parameters $\boldsymbol{\Phi}_j$ is known in advance. In this section, we illustrate a systematic procedure using reachability analysis to obtain parameter ranges based on computations of over-approximative reachable sets of differential and algebraic variables. The reachability algorithm is based on abstracting (2.38) into linear differential inclusions (LDIs) for each consecutive time interval $\tau_k := [t_k, t_{k+1}]$, with $t_k := kt_r$, where $t_r \in \mathbb{R}^+$ refers to the step size, and $k \in \{1,\, \dots,\, h\}$ is the time step. As shown previously in (2.47), the abstraction is expressed at each time interval τ_k via

$$\forall t \in \tau_k: \quad \dot{\boldsymbol{x}}(t) \in \tilde{\boldsymbol{A}}_k \tilde{\boldsymbol{x}}(t) \oplus \tilde{\boldsymbol{\mathcal{U}}}(\tau_k).$$

Using this LDI, the reachable set of differential and algebraic variables can be computed via the exact procedure described in Sec. 2.4 and outlined in Alg. 3.

5.4.1 Reachable Set of Time-Varying Parameters

In order to estimate the admissible space of time-varying parameters $\boldsymbol{\Phi}_j$, we first over-approximate computed sets at each time interval τ_k using

$$\begin{aligned} \boldsymbol{\eta}_k^x &:= \textbf{interval}(\boldsymbol{\mathcal{R}}^x(\tau_k)) \overset{(2.34)}{=} [\underline{\boldsymbol{\eta}}_k^x, \overline{\boldsymbol{\eta}}_k^x], \\ \boldsymbol{\eta}_k^y &:= \textbf{interval}(\boldsymbol{\mathcal{R}}^y(\tau_k)) = [\underline{\boldsymbol{\eta}}_k^y, \overline{\boldsymbol{\eta}}_k^y]. \end{aligned} \tag{5.22}$$

Recall that we use zonotopes as a means of representing reachable sets, see Def. 2.4. The operator **interval** returns the interval hulls, denoted by $\boldsymbol{\eta}^x_k$ and $\boldsymbol{\eta}^y_k$ which corresponds to the over-approximation of computed reachable sets. Here the superscripts x and y are associated with differential and algebraic variables, respectively.

The interval hulls $\boldsymbol{X}$ and $\boldsymbol{Y}$ enclosing the evolution of differential and algebraic variables for the DAE system (2.38) over a time-horizon t_f with h time steps, are

$$\begin{aligned} \boldsymbol{X} &:= \textbf{enclose}\left(\boldsymbol{\eta}^x_k, \dots, \boldsymbol{\eta}^x_h\right) = \left[\underline{\boldsymbol{x}},\, \overline{\boldsymbol{x}}\right], \\ \boldsymbol{Y} &:= \textbf{enclose}\left(\boldsymbol{\eta}^y_k, \dots, \boldsymbol{\eta}^y_h\right) = \left[\underline{\boldsymbol{y}},\, \overline{\boldsymbol{y}}\right], \\ \text{using} \quad \underline{\boldsymbol{x}} &:= \min_{k\in\{1\dots h\}} \underline{\boldsymbol{\eta}}^x_k, \text{ and } \quad \overline{\boldsymbol{x}} := \max_{k\in\{1\dots h\}} \overline{\boldsymbol{\eta}}^y_k, \\ \underline{\boldsymbol{y}} &:= \min_{k\in\{1\dots h\}} \underline{\boldsymbol{\eta}}^y_k, \text{ and } \quad \overline{\boldsymbol{y}} := \max_{k\in\{1\dots h\}} \overline{\boldsymbol{\eta}}^y_k, \end{aligned} \tag{5.23}$$

therefore, the set of admissible parameter values of (5.2) over a time-horizon t_f may be expressed by

$$\boldsymbol{\varphi}_j = \{\boldsymbol{\varphi}_j = \boldsymbol{F}_j(\boldsymbol{x}, \boldsymbol{y}) \,:\, \boldsymbol{x} \in \boldsymbol{X},\, \boldsymbol{y} \in \boldsymbol{Y}\}\,. \tag{5.24}$$

Here, $\boldsymbol{F}_j(\boldsymbol{x}, \boldsymbol{y})$ is a nonlinear function evaluated using interval arithmetic, since the differential and algebraic variables are now presented by interval vectors in (5.23), see Sec. 2.3.8 for more details about interval arithmetics. In fact, the function $\boldsymbol{F}_j(\boldsymbol{x}, \boldsymbol{y})$ describes the time-varying parameters in terms of the differential variables of (2.38), see the LPV models of the synchronous generator (5.7) and the doubly-fed induction generator (5.9) as an example.

Remark 5.5. The main reason behind the approximation of the reachable set by an interval hull is that the bounds of differential and algebraic variables are required to evaluate the nonlinear function $\boldsymbol{F}_j(\boldsymbol{x}, \boldsymbol{y})$. After obtaining these bound and using interval arithmetics, we can easily calculate the multiplication and the division present in $\boldsymbol{F}_j(\boldsymbol{x}, \boldsymbol{y})$, which are associated with the description of the time-varying parameters in terms of the differential and algebraic variables of the original DAE system, see (5.8) and (5.10).

Remark 5.6. As previously mentioned in Ch. 2, the interval hull can be computed efficiently by using zonotopes; this, however, also holds for other set representations such as polytopes and ellipsoids, see Def. 2.2 and Def. 2.3, respectively. Hence, the proposed method can in principle be extended to alternative algorithms which present reachable sets using other representations.

5.5 Simulation Results

This section illustrates the application of our proposed method on two benchmark examples: The single-machine infinite bus [81, Ch. 12] and the IEEE 3-machine 9-bus [14, Ch. 2]. Both benchmark examples were presented earlier in Sec. 2.5. All computations are performed in MATLAB2016b on a standard computer with an Intel Core i7-4810MQ CPU and 16GB of RAM. The set of LMIs of the control synthesis procedure is realized, and solved, using the YALMIP toolbox[1] [91]. The reachability analysis algorithm is implemented using the **Co**ntinuous **R**eachability **A**nalyzer (CORA) toolbox [9].

In contrast to the previous chapters, we apply the proposed framework on the power system models introduced in Sec. 2.1, without any modelling simplifications; in other words, we also take into account the excitation voltage of the synchronous machine, in addition to the magnetic flux of the doubly-fed induction generator found in the wind turbine.

5.5.1 Single-Machine Infinite Bus

The first system to consider is the SMIB system as illustrated previously in example 2.3. The generating unit connected to the infinite bus is the synchronous generator, whose dynamic state variables are modelled according to (2.10)-(2.11), and the algebraic equations are described via (2.12)-(2.15).

First we illustrate the initial step of the proposed method, which is based on reachability analysis as in Sec. 5.4; this step is required in order to obtain the initial guess of the time-varying parameters $\boldsymbol{\Phi}$. Then with the knowledge of the ranges of these parameters, we will design the LPV controller according to the synthesis procedure described in Sec. 5.3. Recall that the synthesis only returns the vertices of the state-feedback controller $\tilde{\boldsymbol{\mathcal{K}}}$; hence, we generate a closed-form expression of the convex combination of the controller vertices, in order to realize the LPV controller in real-time control, see Sec. 5.3.5. Finally, we use the controller in the closed-loop with the generating unit and verify using reachability analysis whether the controller can robustly establish transient stability, and most importantly, can keep the parameters values within the specified ranges of the synthesis procedure.

5.5.1.1 Initial Guess of Time-Varying Parameters

The fault scenario under consideration is the loss of the second transmission line, connecting the generator bus and the infinite bus, at $t = 0.1\,s$, followed by its reconnection to the network after clearance of the fault $t = 0.3\,s$. Fig. 5.3 shows the time-domain bounds of the state variables describing the nonlinear function of the time-varying parameters, i.e. the rotor angle δ, and the stator current i in the d- and q-axis, see (5.8). Notice that we first compute the reachable set of the uncontrolled SMIB system, i.e. the field voltage remains constant in (2.11). As the field voltage remains constant following occurrence of the fault, the system becomes highly undamped resulting in an oscillatory response for the state variables as shown in Fig. 5.3. The reachable set is computed according to Alg. 3 for a specified time-horizon $t_f = 3\,s$. In this example, we do not compute the reachable set until all states are enclosed by the initial reachable set similarly to Ch. 3; this is due to the fact that our goal is to obtain an initial guess concerning the

[1] `https://yalmip.github.io/`

bounds of the time-varying parameters. Clearly, if our objective was to establish transient stability, we must compute the reachable set until the states are enclosed by the initial set.

Remark 5.7. In Ch. 2 and Ch. 3 we showed that the SMIB system converges faster to the equilibrium point for the same fault scenario considered in the aforementioned example. Furthermore, a controller was not present in the closed-loop. This, however, is not the case in this example due to the fact that we previously only considered the swing equation (2.10) and did not take into account the excitation system as an additional state variable, see remark 2.14. Basically, if the excitation of the synchronous generator is considered, one still needs to implement a special controller that regulates the field voltage, which in turn directly influences the excitation system. This regulation introduces the necessary damping torque to eliminate the electromechanical oscillations and stabilize the system in a reasonable time, i.e. a few seconds following clearance of the fault.

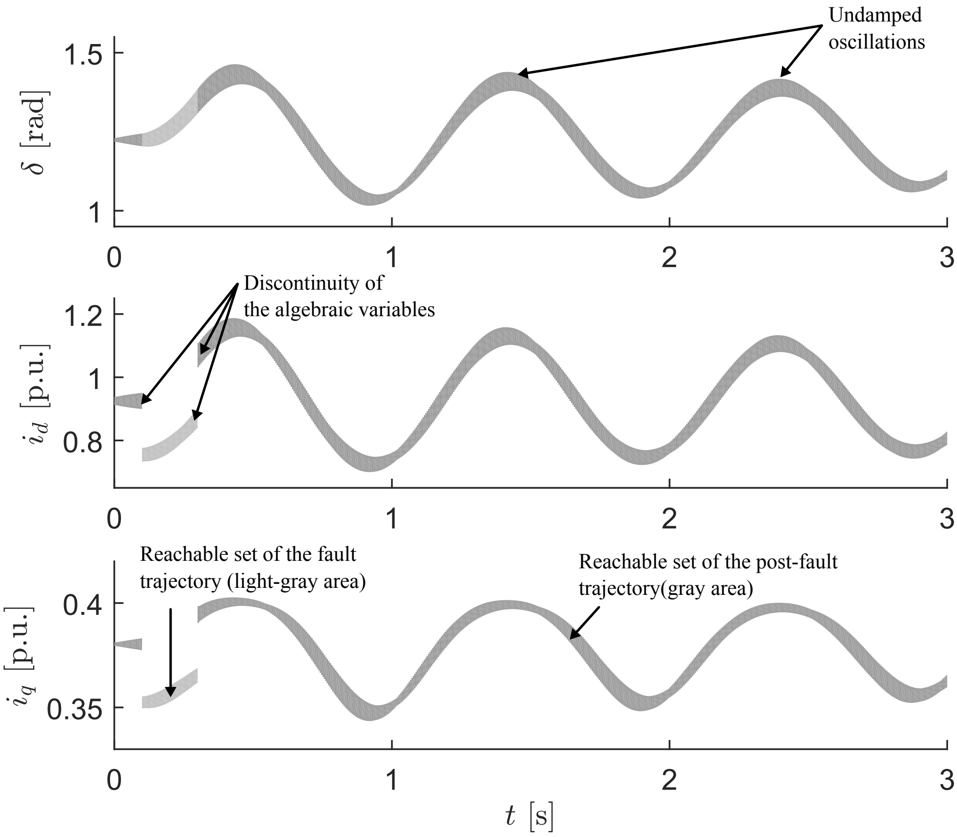


Figure 5.3: Time-domain bounds of chosen state variables corresponding to the uncontrolled SMIB system. The light-gray and gray areas belong to the reachable set during fault and post-fault, respectively. The discontinuity at $t = 0.1$, and $t = 0.2$ is associated with occurrence and clearance of the fault which leads to the discontinuous change in the reachable set. Immediately after losing and reconnecting the transmission line, entries of the admittance matrix $\boldsymbol{Y}$ (2.5) change, which in turn change the algebraic variables in order to satisfy the power flow equations (2.8).

5.5.1.2 Transient Stability using the LPV Controller

After we obtained an initial guess of the time-varying parameters, we can use the LPV controller synthesis using the set of LMIs specified for pole placement and $\mathcal{H}_\infty$ design, see Sec. 5.3.4. The LPV controller is synthesized around an LMI-region consisting of the half-plane $\text{Re}(s) > -1$, the arc with an angle $\alpha = 45°$, and the disk with the radius $R = 10$, see Fig. 5.2. The chosen parameters have to enforce sufficient damping torque, thus eliminating the electromechanical oscillations present in the closed-loop.

Fig. 5.4 illustrates selected projections of the reachable sets for transient stability analysis of the DAE system using the LPV controller in the closed-loop. To ensure stability of the system, we compute the reachable set of the post-fault phase until all continuous state variables return to the initial set $\boldsymbol{\mathcal{R}}^x(0)$. The location of the poles in the complex-plane is shown in Fig. 5.5, where it is clear that the poles are confined within the LMI region specified for pole placement design.

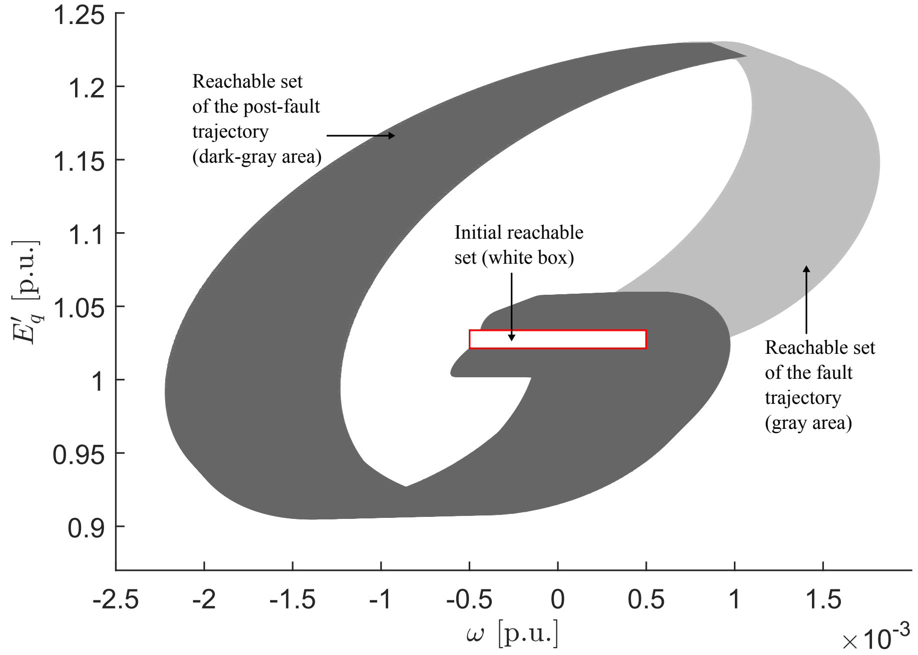


Figure 5.4: Selected projections of the reachable sets corresponding to the SMIB DAE system with the LPV controller in the closed-loop. The dark-gray and the gray areas correspond to the reachable set during fault, and post-fault, respectively. The computation of the reachable set terminates when all state variables converges back to the initial reachable set $\boldsymbol{\mathcal{R}}^x(0)$ (white box).

We include uncertainty in the initial set of differential variables, since initial states are not exactly known due to increasingly varying operating conditions in current power systems. We introduce the unit interval $\boldsymbol{U} := [-1, 1]$, and use the center of the initial set as the steady state solution of (2.38) denoted by the superscript 0. Therefore, the initial phase angle of the generator is $\delta(0) \in \delta^0 \oplus 0.01 \cdot \boldsymbol{U}$ [rad], the initial rotational speed is $\omega(0) \in \omega^0 \oplus 0.005 \cdot \boldsymbol{U}$ [p.u.], and the initial q-axis transient voltage is $E'_q(0) \in E'^0_q \oplus 0.01 \cdot \boldsymbol{U}$ [p.u.].

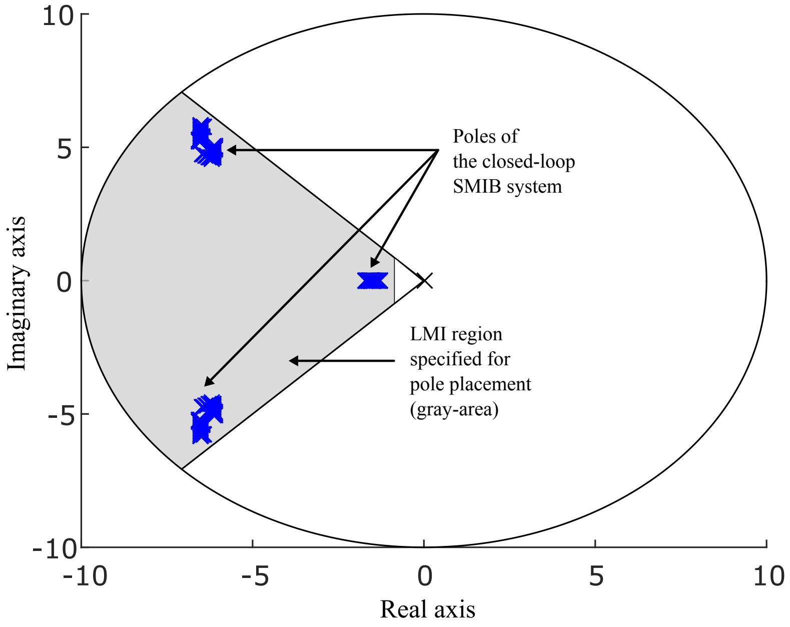


Figure 5.5: Location of the poles corresponding to the closed-loop of the LPV model.

We still need to guarantee that the time-varying parameter will always remain within the specified bounds using the synthesized controller under all eventualities. The bounds are obtained by over-approximating the reachable set of the state variables δ, i_d, and i_q (see Fig. 5.3) via a multi-dimensional interval. Since these variables are described via an upper and lower bound, we can use interval arithmetics and easily calculate the values of the time-varying parameters using (5.8). Fig. 5.6 shows the time-domain bounds of the time-varying parameters corresponding to the controlled SMIB system.

5.5.2 IEEE 3-machine 9-bus

Now we consider the IEEE 9-bus benchmark previously introduced in Ch. 2 and illustrated in Fig. 2.18. In this case study we use two variants of this power system: In the first variant all three generators are modelled as synchronous generators. In the second variant of the benchmark system, the third generator G_3 is replaced by the doubly-fed induction generator, and it is controlled as well via the decentralized LPV-controller. Note that the variants are chosen arbitrarily for demonstration purposes of the method's applicability and scalability.

5.5.2.1 Control of the Synchronous Generator

The considered fault scenario is the loss of the transmission line between bus 5 and 7 at $t = 0.01\,s$, followed by its reconnection after clearance of the fault $t = 0.2\,s$. The reachable set is computed according to Alg. 3. Similarly to the SMIB example, we first compute the reachable set of the uncontrolled system, i.e. the field voltage remains constants for all three generators. After finding the initial guess of the time-varying parameters, the LPV controller is designed and used in the closed-loop to establish transient stability.

Fig. 5.7 shows chosen projections of the reachable set corresponding to the controlled IEEE 9-bus system for the considered fault. Immediately after losing the transmission line connecting buses 5 and 7, entries of the admittance matrix $\boldsymbol{Y}$ change. This disturbance generates a new control action from the unified control structure, see Fig. 5.1, as the local measurements $[\delta_j,\, P_j,\, Q_j,\, V_j\angle\theta_j]$ at each j-th generator bus were affected by the perturbation in the transmission network. Fig. 5.8 illustrates the estimated bounds of the time-varying parameters.

5.5.2.2 Control of the Doubly-fed Induction Generator

The considered fault is the loss of the transmission line between bus 5 and 7 at $t = 0.1\,s$, followed by its reconnection to the network after clearance of the fault $t = 0.3\,s$. The reachable set is computed

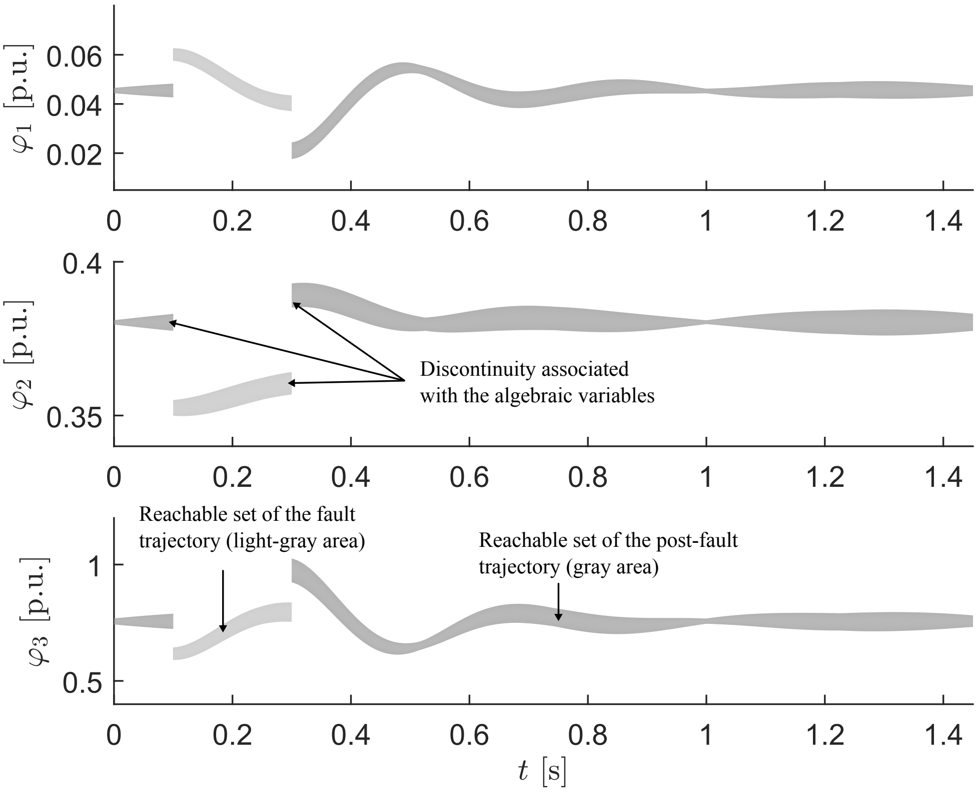


Figure 5.6: Time-domain bounds of the time-varying parameters corresponding to the controlled SMIB system. The light-gray and gray areas belong to the reachable set during fault and post-fault, respectively. The discontinuity at $t = 0.1$, and $t = 0.2$ is associated with the fault which leads to the discontinuous change in the reachable set. Immediately after losing and reconnecting the transmission line, entries of the admittance matrix $\boldsymbol{Y}$ (2.5) change, which in turn change the algebraic variables i_d and i_q in order to satisfy the power flow equations (2.8). Notice that the time-varying parameters depend on i_d and i_q as in (5.8).

according to Alg. 3. In this example the uncertainty of the initial reachable set is directly associated with the intermittent nature of the wind speed, which in turn affects the rotational speed of the doubly-fed induction generator $w_{r,3}$.

Similarly to the previous examples, first the reachable set of the uncontrolled system is computed to obtain an initial guess of the time-varying parameters. In this case, the rotor voltages of the doubly-fed induction generator $v_{r,q}$ and $v_{r,d}$ are kept constant, see (2.18). Afterwards the LPV controller is designed for each generator and used in the closed-loop to examine stability of the multi-machine power system for the considered fault. Fig. 5.9 shows projections of the dynamic state variables corresponding to the DFIG. It is clear that the synthesized LPV controller robustly establishes stability of the closed-loop, since the state variables converged back to the initial reachable set after clearance of the fault from the transmission network. Fig. 5.10 shows the estimated bounds of chosen time-varying parameters associated with the generator of the wind turbine.

5.6 Summary

In this chapter we proposed a unified approach, based on reachability analysis, to combine in one framework synthesis and verification of LPV controllers to robustly establish transient stability of multi-machine power systems with formal guarantees. The proposed framework first reformulates the set of nonlinear DAEs governing dynamics of multi-machine power systems into modular LPV systems, thus allowing one to systematically synthesize and verify decentralized controllers while preserving the correlation between different machines connected to the grid.

Using our method, one can systematically obtain the set of the time-varying parameters, required for the synthesis procedure, in the least conservative way, using reachability analysis. This is particularly beneficial for larger power systems since finding consistent parameter ranges for the synthesis of LPV controllers, simultaneously for each LPV system, can be difficult when not following a systematic procedure. Furthermore, one can check using our approach whether system constraints, e.g. frequency and bus voltage remain within permitted ranges. This due to the fact that the bounds of the time-varying parameters can be estimated from the reachable set of differential and algebraic variables.

We illustrate the application and scalability of this approach for transient stability analysis without any modelling simplifications, as we consider the set of semi-explicit, nonlinear, index-1 DAEs governing dynamics of multi-machine power systems. In our case study, we considered the IEEE 3-machine 9-bus benchmark problem with different variants, i.e. by considering the synchronous generator and the doubly-fed induction generator as the generating units of the system.

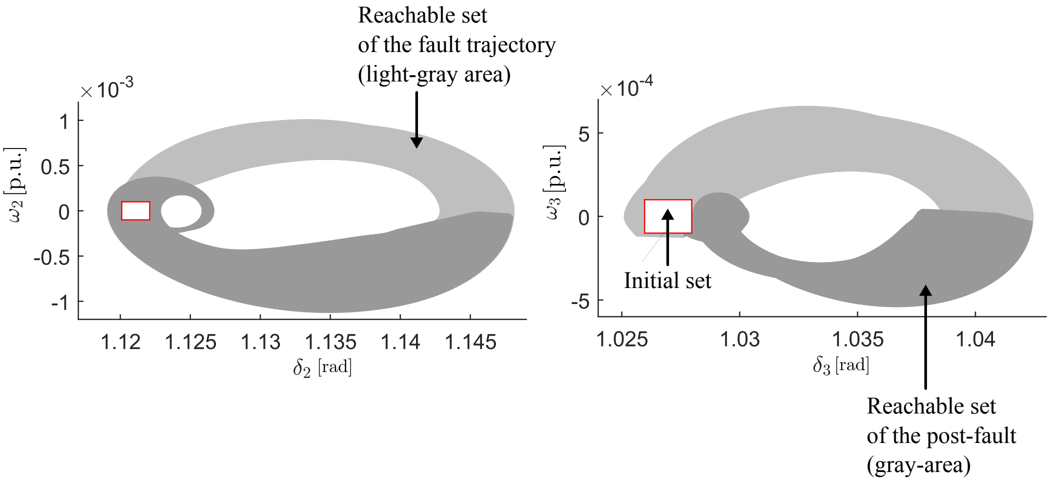


Figure 5.7: Projection of the reachable set of differential variables for generators G_2 and G_3 corresponding to the controlled IEEE 3-machines 9-bus benchmark. The light-gray and gray areas belong to the reachable set during fault and post-fault, respectively. The white box corresponds to the set of initial states $\mathcal{R}^x(0)$. The considered fault is the loss of the transmission line connecting buses 5 and 7. The line is reconnected after clearance of the fault, and the reachable set is computed until all states are enclosed by $\mathcal{R}^x(0)$.

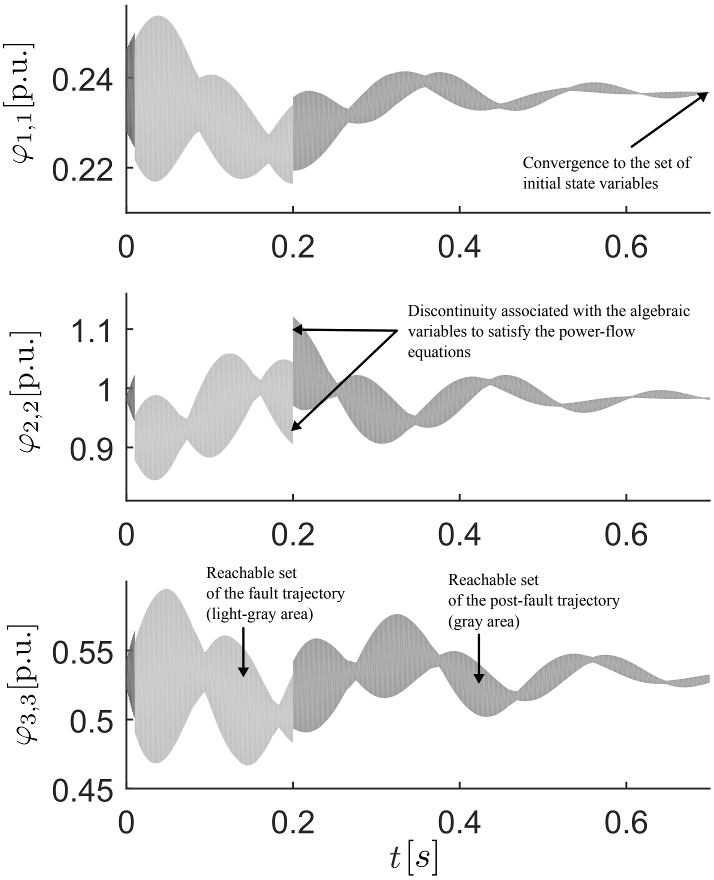


Figure 5.8: Time-domain bounds of chosen time-varying parameters $\varphi_{n,m}$, with n being the n-th coordinate of $\mathbf{\Phi}_m$ and m denoting the m-th generator of the controlled IEEE 9-bus system. The light-gray and gray areas belong to the reachable set during fault and post-fault, respectively. The discontinuity in the projections at $t = 0.01$, and $t = 0.2$ is associated with the fault which leads to the discontinuous change in the algebraic variables to satisfy the power flow equations (2.8).

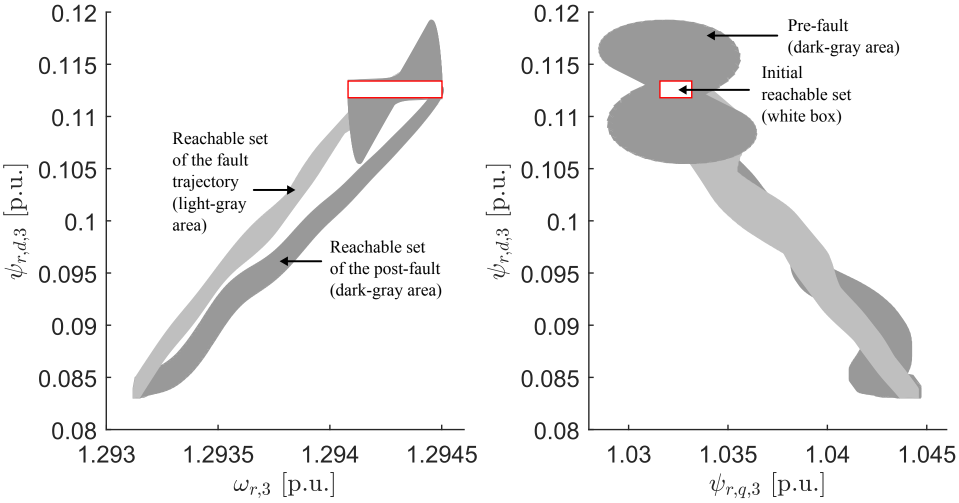


Figure 5.9: Projection of the reachable set of differential variables for the doubly-fed induction generator in the controlled IEEE 9-bus benchmark problem. The light-gray and dark-gray areas belong to the reachable set during fault and post-fault, respectively. The white box corresponds to the set of initial state variables $\boldsymbol{\mathcal{R}}^x(0)$. The considered fault is the loss of the transmission line connecting the buses 5 and 7. The line is reconnected after clearance of the fault, and the reachable set is computed until all states are enclosed by $\boldsymbol{\mathcal{R}}^x(0)$ to formally verify that the LPV controller introduces sufficient damping torque to converge state variables back to a stable equilibrium.

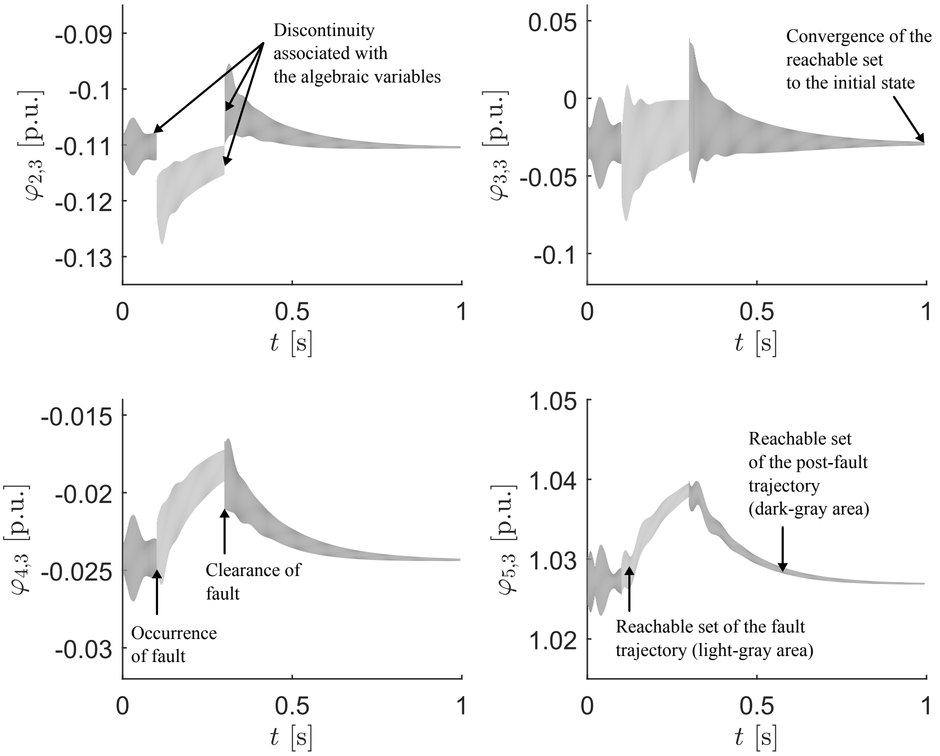


Figure 5.10: Time-domain bounds of the time-varying parameters corresponding to the doubly-fed induction generator of the controlled IEEE 9-bus benchmark problem. The light-gray and dark-gray areas belong to the reachable set during fault and post-fault, respectively. Immediately after losing and reconnecting the transmission line, entries of the admittance matrix $\boldsymbol{Y}$ (2.5) change, which in turn change the algebraic variables $i_{s,d}$ and $i_{s,q}$ in order to satisfy the power flow equations (2.8). Notice that the time-varying parameters depend on $i_{s,d}$ and $i_{s,q}$ as in (5.10).

Chapter 6

Formal Analysis and Control of Power Plants

In the previous chapters we have presented the application of reachability analysis with regards to control and stability analysis of general power systems. In this chapter, however, we are concerned with the application of reachability analysis to verify safety of a realistic configuration of a boiler system located within a 450 MW combined cycle power plant in Munich, Germany. Namely we consider the steam-drum unit known to degrade the load-following capabilities of conventional power-producing units; thus, limiting flexibility of power plants to meet the string requirements imposed by the corresponding transmission system operator (TSO), which in turn can endanger frequency and transient stability in the control area of the TSO. The proposed reachability algorithm makes it possible to compute the bounds of all possible trajectories for a range of operating conditions while simultaneously meeting the practical requirements of a real power plant. In contrast to previous works in this area, we use for the first time an abstract model which considers the modelling errors to ensure that all dynamic behaviours of the process are replicated by the abstraction. The modelling errors are obtained based on measurement data from the boiler system, with a very rich excitation covering the entirety of the operational range of the process. Through the implementation of our abstract model, we formally guarantee that the water level inside the drum always remains within safe limits for load changes equivalent to 40 MW which, as a result, exploits the power plant's adaptability and load-following capabilities.

6.1 Introduction

There is a paradigm shift with regards to the operation of conventional power plants following the deregulation of the energy sector, the introduction of competitive markets, and the transition towards decentralized generation with considerable share of renewable resources [20, 79, 125]. Particularly, the load-following capabilities of conventional power-producing units became increasingly important in recent years as progressively intermittent and variable generators, such as wind turbines and solar photovoltaics, are added to the grid. This radical change forces the power plants to adapt their power outputs more regularly to ensure a reliable operation of the transmission network. Typically, each transmission system operator (TSO) preserves the frequency in its control area using a centralized control scheme compensating the deviation of the power grid frequency. This deviation from the specified set-points results from the mismatch between supply and demand of the active power. The control action includes generation units (or loads) that respond to automatic generation control (AGC) signals in the case of secondary frequency control or to manual operator dispatch commands in the event of tertiary control [117]. In Germany, the generation units subjected to this control scheme are obligated to provide active power in both directions (increased/reduced generation) without interruption, with typical limits ranging between 5 MW and 60 MW within a short time-scale of $5 - 15$ minutes [65]. These stringent requirements escalated rapidly new challenges that have to be met by the existing process-controllers; in other words, the controllers have to be designed in such a manner that can fulfill frequent load-changes requested by the TSOs, while simultaneously bearing in mind safety and life span of the power plant critical elements, such as the steam turbine and the boiler.

Steam-drum units of the boiler system are known to naturally degrade the load-following capabilities of thermal power plants and limit their flexibility to meet the stringent requirements imposed the corresponding TSO; namely during high-load changes (≤ 40 MW) when subjected to the time constraints of secondary frequency control (5 min). A reason for this is that the regulation of the water level inside the drum is a tedious control task due to the process nonlinearities, strong coupling between its input and output channels, in addition to the process non-minimum phase behaviour associated with the shrink and swell physical phenomena [73].

In this chapter, we consider the realistic configuration of the low-pressure drum unit located within the 450 MW München Süd GuD 2 power plant owned by Munich City Utilities[1]. As reported by the plant operators, the boiler unit previously tripped on multiple occasions as the water level inside the drum exceeded the safety limits (±300 mm), see Fig. 6.1. Simply put, when the water level exceeds the upper limit, the water will be carried over to the superheater leading to an outage of the boiler. Surpassing the lower limit will cause overheating of the water wall tube, resulting in serious tube rupture and severe damage. Clearly, the outage of the boiler has serious technical and economical consequences: the power plant is subjected to drastic economical losses, furthermore the TSO loses one of its generating units, thus, jeopardizing transient and frequency stability in its balancing area. In fact, this problem is reoccurring in thermal power plants where emergency shutdowns of conventional power-producing units are commonly trigged due to poor regulation of the water level, see e.g. [80, 105, 106, 140].

To address this problem, we previously presented the design and successful implementation of a centralized multivariable feedback controller whose control action is based on the inner dynamics of the process,

[1] SWM Services GmbH `https://www.swm.de/`

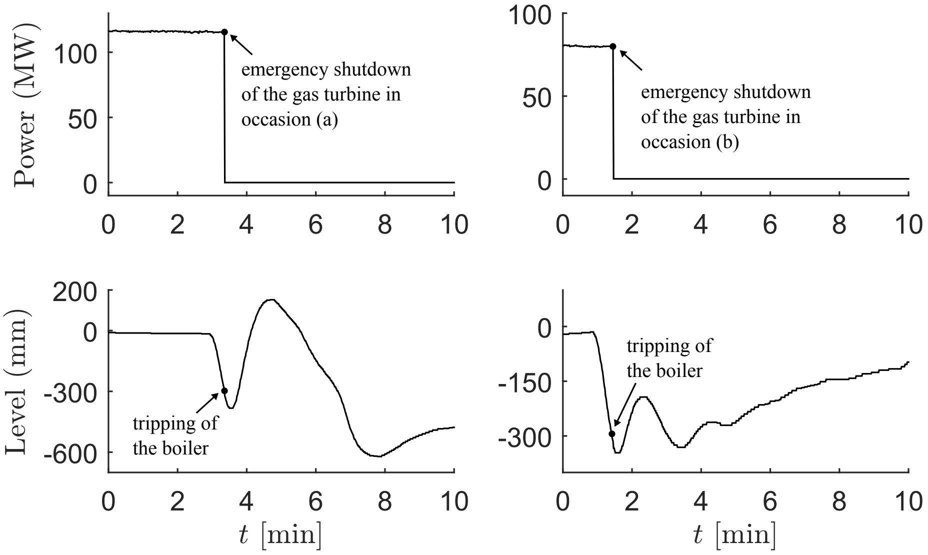


Figure 6.1: Measurement data from the steam-drum unit illustrating tripping of the gas turbine on multiple occasions while employing the conventional PID-controller in the closed loop.

which are captured via a Luenberger observer, see [41, 45, 46]. The proposed controller replaced in November 2014 the industry-standard PID-controller within the distributed control system (DCS)-Mauell ME-4012[1], which is employed in the power plant München Süd GuD 2; however, due to the fact that the control action is based on a mathematical model of the real process (observer-based controller), plant operators voiced skepticism about the safety of the new control scheme following the commissioning phase within their DCS. Hence, we additionally present an algorithmic procedure based on reachability analysis in order to verify the control action in real-time control. This work corresponds to the growing body of literature that considers the intersection of formal methods and control theory. To the best of our knowledge, no work exists in the literature that formally analyzes the correctness of the low-level controllers prior to the commissioning stage within the high-level DCS of the power plant; that is, the formal guarantee that the synthesized controller shall meet the performance specifications under all eventualities. Instead, in other work, the control action is either examined via experimental observation or within a simulation environment that does not provide any formal guarantees, i.e. one cannot certify whether the system specifications will always remain within safe limits. Because reachability analysis establishes in advance whether or not a requested load dispatch by the TSO will trigger the water level safe limit, the plant operator can potentially avoid an unnecessary shutdown of the facility; thus maximally exploiting the power plant adaptability and load-following capabilities.

Formal verification of the steam-drum unit was initially proposed in [4] as a benchmark problem for formal analysis and controller synthesis of embedded systems. The benchmark attracted considerable attention as an interesting theoretical problem and became a classical case study for testing and comparing formal

[1] http://www.mauell.de/index.php?Produktmodul/

methods among computer scientists, see [63,93,132,137]. The main drawback of the benchmark problem, however, is that the modelling is based on elementary assumptions and abstract decisions that do not capture the dynamics of the process associated with the shrink and swell physical phenomena. Thus, we extend the benchmark problem with a well-developed mathematical model, validated against data measurement, which are obtained from a real steam-drum unit, with a very rich excitation covering the entire operational range of the process.

In contrast to any previous work on reachability analysis, we construct an abstract model of the real process taking the modelling errors into account. The modelling errors are obtained in a systemic procedure based on measurement data, and considered as additional uncertain inputs when computing the reachable set. This procedure ensures that all behaviours of the system are included within the abstraction. Furthermore this is the first work to consider formal analysis of a real process in the power industry. The reachability algorithm we propose is computationally feasible and meets the practical requirements of a real power plant when subjected to the time constraints of secondary frequency control (5 min). Our algorithm offers the plant operator an opportunity to potentially avoid unnecessary shutdown of the facility since reachability analysis establishes in advance whether a requested load dispatch by the TSO will trigger the safe limits of the water level when considering all eventualities.

6.2 Problem Formulation

The steam-drum system falls under the class of systems modelled as a set of nonlinear, ordinary differential equations (ODEs)

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t), \end{aligned} \tag{6.1}$$

with $\boldsymbol{f} : \mathbb{R}^{n_x+n_u} \mapsto \mathbb{R}^{n_x}$, $\boldsymbol{x} \in \mathbb{R}^{n_x}$, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ and $\boldsymbol{y} \in \mathbb{R}^{n_y}$ denoting the state, input and output vectors, respectively, and $\boldsymbol{C} \in \mathbb{R}^{n_y \times n_x}$ as the output matrix. It is assumed that the system is controllable and observable. Furthermore, the function $\boldsymbol{f}(\,\cdot\,)$ is locally Lipschitz continuous thus differentiable in $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$. This is a fairly general assumption that holds for many practical problems.

In our previous work [46], we proposed to regulate the pressure and water-level inside the drum using a multivariable feedback controller whose control law is:

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}, \tag{6.2}$$

with $\boldsymbol{K} \in \mathbb{R}^{n_u \times n_x}$ is the controller feedback matrix. Notice that the controller is a special case of the linear-parameter varying controller addressed in Ch. 5. This due to the fact that entries of the matrix $\boldsymbol{K}$ are no longer time-varying, but instead they are kept constant[1]. Generally, the drum state variable are

[1] A gain-scheduling controller was proposed to the plant operators, however, it was rejected in early development due to its complexity and computational difficulties associated with its realization in practice within the DCS of the power plant.

not measurable in real-time control, hence one requires additionally an observer to estimate the system states and employ the proposed control law in the closed-loop. Recall that (6.1) is differentiable in $\boldsymbol{x}$ and $\boldsymbol{u}$; this makes it possible to design the so-called *Luenberger-like* nonlinear observer expressed via (see [115])

$$\dot{\tilde{\boldsymbol{x}}}(t) = A\tilde{\boldsymbol{x}}(t) + \boldsymbol{\Omega}(\tilde{\boldsymbol{x}}(t), \boldsymbol{u}(t)) + \boldsymbol{L}\,\underbrace{\boldsymbol{C}(\boldsymbol{x}(t) - \tilde{\boldsymbol{x}}(t))}_{=:\boldsymbol{e}(t)}.$$

Here $\tilde{\boldsymbol{x}}$ is the vector of estimated state variables, $\boldsymbol{e}$ is the estimation error, $\boldsymbol{L} \in \mathbb{R}^{n_x \times n_y}$ corresponds to the observer correction matrix, and $\boldsymbol{\Omega} : \mathbb{R}^{n_x+n_u} \mapsto \mathbb{R}^{n_x}$ is a Lipschitz nonlinearity.

The objective of this chapter is to verify safety of the low-pressure drum unit which employs the proposed controller (6.2) in the closed-loop. This task is addressed by computing the reachable set of the drum over a user-defined time horizon $t \in [0, t_f]$ starting from a set of initial states $\boldsymbol{\mathcal{R}}(0)$ and a set of possible inputs/disturbances $\boldsymbol{\mathcal{U}}$

$$\boldsymbol{\mathcal{R}}^e([0,t_f]) := \left\{ \boldsymbol{x}(t) \in \mathbb{R}^{n_x} \,:\, \boldsymbol{x}(t) = \int_0^t \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau, \boldsymbol{x}(0) \in \boldsymbol{\mathcal{R}}(0),\, \boldsymbol{u}(t) \in \boldsymbol{\mathcal{U}},\, t \in [0,t_f] \right\}.$$

Recall from Ch. 2 that the exact reachable set $\boldsymbol{\mathcal{R}}^e([0,t_f])$ can only be computed in special cases [84]; thus, an over-approximation of the reachable set $\boldsymbol{\mathcal{R}}([0,t_f]) \supseteq \boldsymbol{\mathcal{R}}^e([0,t_f])$ is performed as tightly as possible, see Fig. 2.7. Clearly, if the over-approximative reachable set does not intersect with an unsafe set, then the original system is also safe. Naturally in this case study, the unsafe set would be the limits of the water level inside the drum (±300 mm).

With regards to the overall procedure, we propose a generic approach using an abstract model described by a polynomial differential inclusion. The concept of model abstraction is frequently applied in the field of computer science within the context of model checking and software verification. An abstraction basically reduces the complexity associated with a mathematical model, such that the resulting approximated model preserves certain user-defined properties of the original system [35]. The considered polynomial abstraction takes the modelling errors into account, thus ensuring that all behaviours of the system are confined within the following inclusion

$$\dot{\hat{\boldsymbol{x}}}(t) \in \boldsymbol{P}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t)) \oplus (\boldsymbol{L} \cdot \boldsymbol{\mathcal{E}}). \tag{6.3}$$

Here $\boldsymbol{P} : \mathbb{R}^{n_x+n_u} \mapsto \mathbb{R}^{n_x}$ is a polynomial vector-field function, $\boldsymbol{\mathcal{E}} \subset \mathbb{R}^{n_y}$ is the set of the modelling errors, and $\hat{\boldsymbol{x}} \in \mathbb{R}^{n_x}$ is the vector of the abstract model state variables. The abstraction includes set-based addition (Minkowski sum) and linear transformation, as defined previously in (2.29), see Sec. 2.3.6.

Our approach, illustrated in Fig. 6.2, consists of four main steps: (1) modelling from first-principles, (2) polynomial approximation, (3) abstraction, and (4) computation of the over-approximative reachable set. In the following we describe the modelling of the drum unit in Sec. 6.3, followed by the proposed polynomial abstraction and the basic procedure to compute the reachable set in Sec. 6.4. Note that the proposed approach can be applied for different systems in many areas, including robotics and autonomous cars, as long as the system is modelled as in (6.1).

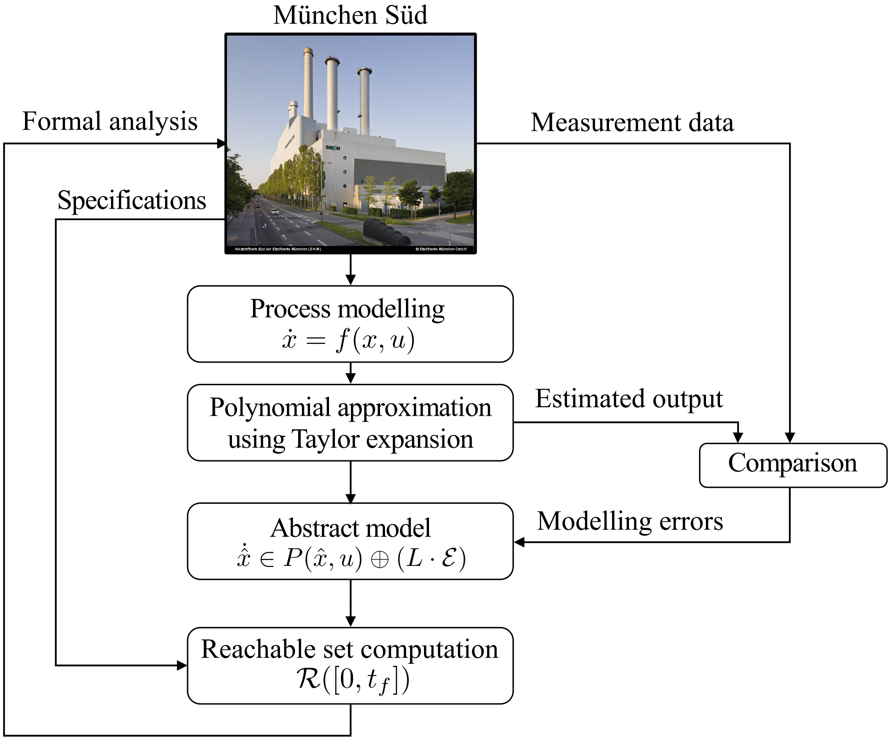


Figure 6.2: Overview of the proposed approach to verify safety of the water level inside the drum in real-time control using reachability analysis.

6.3 Process Modelling

In this section we describe briefly the mathematical model of the complete process; first we illustrate the basic working principle of the power plant, then we focus on the steam generation process within the boiler system using the low-pressure drum unit. Finally we illustrate the state-feedback controller currently in operation within the power plant's DCS to regulate the water level and pressure of the drum.

6.3.1 Basic Working Principle

The plant München Süd GuD 2 is classified as a combined cycle co-generation power plant handling concurrent production of electrical power and heat. These plants utilize a class of sustainable integrated technologies progressively being used in the energy sector. Co-generation plants reduce thermal and mechanical losses, harmful carbon dioxide emissions, and more importantly increases the overall plant efficiency to approximately 81% in comparison to standalone thermal plants which do not exceed an efficiency of 45%.

The combined cycle working principle of the plant München Süd GuD 2 is shown in Fig. 6.3 starting from the gas turbine unit whose process is based on the Brayton thermodynamic cycle [103, p. 525]: first, fresh air is being compressed and mixed with the supplied natural gas, then the mixture is burned inside a combustion chamber at around 1124° C. The hot compressed air expands within the turbine, thus driving its blades which in turn acts as the prime mover of the synchronous generator. The generator converges the mechanical energy, which is supplied by the prime mover via its rotor shaft into electricity, according to the working principle of the synchronous machine, see Sec. 2.1.2. Finally, the exhaust gas leaves the turbine at low pressure and temperature and it is used as the heat source of the heat recovery steam generator.

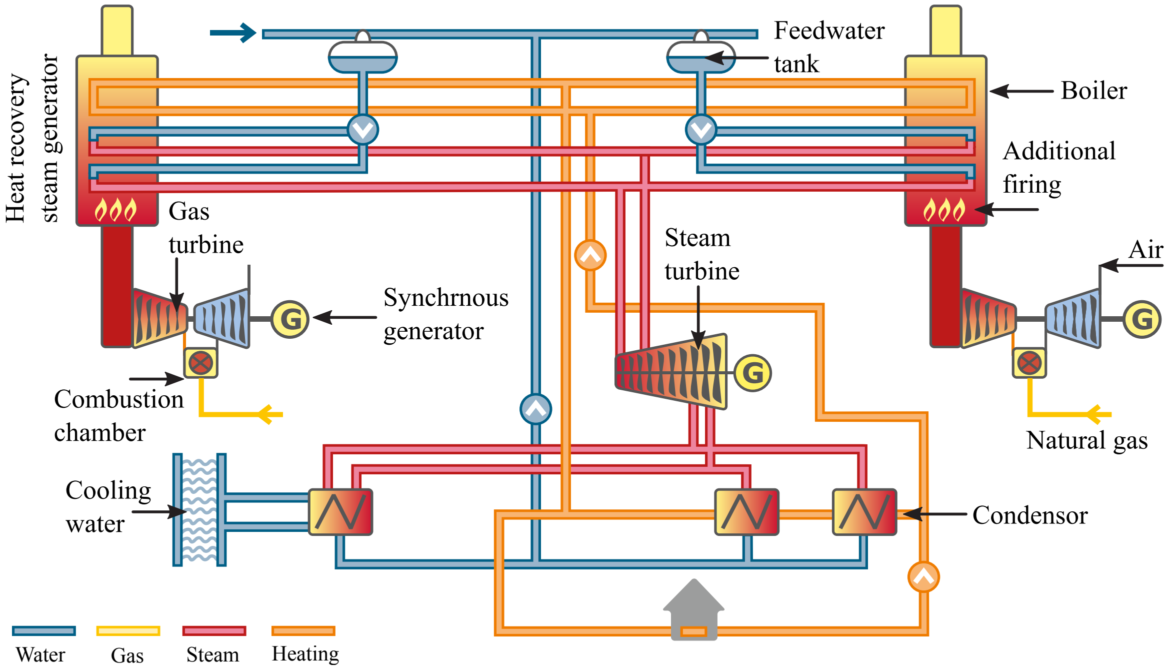


Figure 6.3: Simplified schematic diagram of the combined-cycle working principle of the power plant München Süd GuD[1].

The heat recovery steam generator acts as a heat exchanger between the exhaust heat supplied from the gas turbine and the liquid/vapour mixture circulating into finned tubes through three heat exchangers as highlighted in Fig. 6.3. Notice that additional firing can take place if necessary, particulary during cold weather. Production of high pressure steam is carried out using high- and low-pressure drum units through the following stages:

1. **Economizer stage** At first cold water is fed through a feedwater pump to supply the drum inlet. The water is then preheated through the economizer in order to reduce energy consumption.

[1] https://www.swm.de/dam/jcr:54c2b62e-4173-43a2-8850-a7d1aed94717/heizkraftwerk-sued.pdf

2. **Evaporator stage** The preheated water flows down through a naturally circulated downcomer-riser loop under the gravitational influence. The evaporator produces saturated steam flowing within the riser tubes before being collected and fed back once again into the drum.
3. **Superheater stage** Finally, saturated steam flows through the water level inside the drum until it reaches the drum outlet. Here it is reheated through the superheater before being supplied to the steam turbine.

In contrast to the gas turbine, the working principle of the steam turbine is based on the Rankine thermodynamic cycle [103, p.446]. The superheated steam at high-pressure and temperature flows through the turbine inlet converting the steam thermal energy into mechanical energy supplied to the prime mover, which similarly in turn is converted into additional electricity via another synchronous generator. Upon exiting through the turbine outlet, the steam, which has already lost most of its temperature during the conversion process, is collected and fed into the final stage of the combined cycle through the surface condenser. The condenser basically makes it possible to achieve maximum attainable efficiency from the thermodynamic cycle as it condenses the exhaust steam via cold water; thus carrying off the steam waste heat due to its availability, high specific thermal capacity, and heat transfer properties.

With this brief illustration of the power plant working principle, we can now focus on the steam generation process through the configuration of the drum unit at München Süd GuD 2. The simplified process relevant to our analysis is illustrated in Fig. 6.4. This system consists of four components; namely, the steam-drum unit, the regulating valves, the gas turbine exhaust heat, and the state-feedback controller[1].

6.3.2 Model of the Steam-Drum Unit

As stated earlier, the drum is a nonlinear system with a strong coupling between its input and output channels in addition to a non-minimum phase response associated with the shrink and swell of the steam bubbles under the water level. Over the years, several models were introduced to capture the dynamical behaviour of the process, see for example [52, 77, 78]. In particular, we consider the so-called Åström - Bell model introduced in [73]; the model is the result of various improvements through the course of its development cycle which led to a 4-th order system with three actuating variables and two output channels capable of capturing most of the complicated dynamics occurring within the drum. The model basically considers mass-flow and energy balance at different parts of the drum: for the whole system, within the naturally-circulated downcomer-riser loop, and finally with regards to the condensation inside the drum, see Fig. 6.5. Shortly after, we introduce the governing mathematical equations describing dynamics of the systems. The model constant parameters are listed in Table 6.1.

[1] The low-pressure steam-drum unit was previously regulated via the industry standard 3-element PID-controller. For further details with regards to this control architecture, the reader is referred to our previous work [41, Ch. 2].

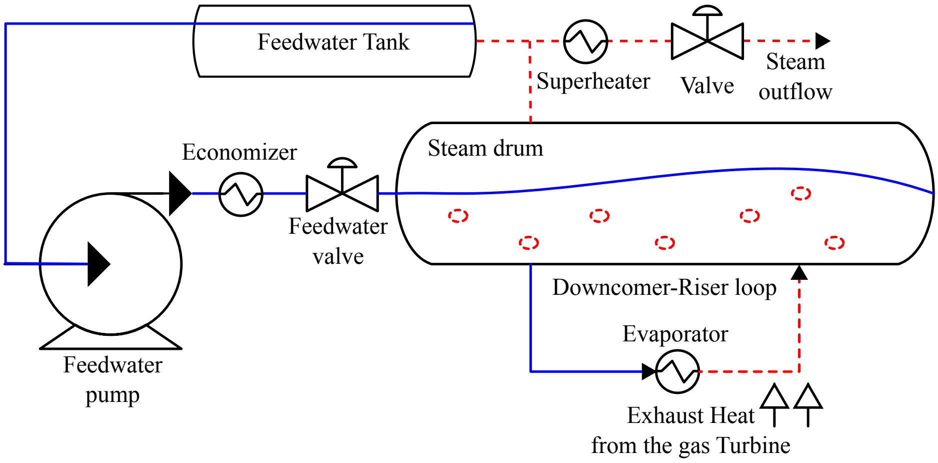


Figure 6.4: Simplified illustration of the steam generation process. The red line (dotted) indicates hot steam and the blue line (solid) indicates cold water. The cold water inside the feedwater tank is pumped and heated at the economizer stage before going through the drum inlet. Due to the gravitational force, feedwater flows through the naturally circulated downcomer riser loop, where it is converted into steam at the evaporator stage. Different riser tubes collect the steam and supply it back into the drum. In the final stage, the saturated steam is taken from the drum outlet to the superheater.

6.3.2.1 Global Mass-flow and Energy Balance

The model is based on the assumption that most of the system parts will be under thermal equilibrium due to their direct contact with the saturated liquid/vapour mixture; in other words, the energy stored in the mixture is either absorbed or released quickly following changes in the drum pressure. This leads various metal parts of the system to adapt their temperatures in the same manner. This basic assumption agrees with experimental observation as it was proven that the temperature difference is almost negligible; thus a detailed representation of the temperature distribution within the metal is not necessary.

With this basic assumption, the mass-flow and energy balance for the overall steam-drum unit are expressed by:

$$q_{fw} - q_s = \frac{\mathrm{d}}{\mathrm{d}t}\Big(\rho_s V_{st} + \rho_w V_{wt}\Big), \tag{6.4}$$

$$Q + q_{fw}h_{fw} - q_s h_s = \frac{\mathrm{d}}{\mathrm{d}t}\Big(\rho_s h_s V_{st} + \rho_w h_w V_{wt} - P V_{wt} + m_t C_p t_{\mathrm{sat}}\Big), \tag{6.5}$$

with (6.4) and (6.5) describing to the balance of the mass-flow and the energy-flow, correspondingly. Here $V_{wt}\,[\mathrm{m}^3]$ is the water total volume, whereas $V_{st}\,[\mathrm{m}^3]$ is the steam total volume, $Q\,[\mathrm{W}]$ is the heat-flow rate associated with the gas turbine exhaust temperature, q_{fw} [kg/s] and q_s [kg/s] denote mass-flow rates of cold feedwater and superheated steam, respectively, and $P\,[\mathrm{Pa}]$ describes the drum absolute pressure. Additionally, the thermal properties $h\,[\mathrm{J/kg}]$ and $\rho\,[\mathrm{m}^3/\mathrm{kg}]$ describe the specific enthalpy and density at the saturation pressure P, correspondingly. Note that the subscripts s, w, t, fw refer to steam, water, total, and feedwater, respectively.

6.3.2.2 Mass-flow and Energy Balance of Downcomer-Riser Loop

Now we consider the mass-flow and energy balance for the naturally circulated loop with the evaporator. Here, the governing differential equations are:

$$q_{dc} - q_r = \frac{\mathrm{d}}{\mathrm{d}t}\left(\rho_s \alpha_v V_r + \rho_w (1-\alpha_v) V_r\right), \tag{6.6}$$

$$Q + q_{dc} h_w - q_r (h_w + \alpha_r h_c) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\rho_s h_s \alpha_v V_r + \rho_w h_w (1-\alpha_v) V_r - P V_r + m_r C_p t_{\mathrm{sat}}\right), \tag{6.7}$$

with $\alpha_r\,[-]$ and $\alpha_v\,[-]$ as the quality and average volume fraction of the steam within the riser tubes, correspondingly, $h_c := h_s - h_w$ [J/kg] is the condensation specific enthalpy, and q_{dc} [kg/s] and q_r [kg/s] corresponds to the mass-flow rate through the downcomer and riser tubes, respectively. The balancing equations (6.6) and (6.7) are based on a lumped model approximating the dynamics of water and steam inside a heated tube, governed by a complicated set of partial differential equations. Using this lumped model, one can express q_{dc} empirically via the following the algebraic equation [73]:

$$\begin{aligned} q_{dc} &= \sqrt{\frac{2\rho_w A_{dc}(\rho_w - \rho_s) g \alpha_v V_r}{F_d}}, \\ \text{with: } \alpha_v &= \frac{\rho_w}{\rho_w - \rho_s}\left(1 - \frac{\rho_s}{(\rho_w - \rho_s)\alpha_r}\ln\left(1 + \frac{\rho_w - \rho_s}{\rho_s}\alpha_r\right)\right). \end{aligned} \tag{6.8}$$

6.3.2.3 Distribution of Steam inside the Drum

The final set of equations considers the mass-flow and energy balance through the water level inside the drum. Generally, it is extremely hard to develop a mathematical model from first principles with a reasonable degree of complexity considering the complication of the physical phenomena occurring inside the drum; thus, an empirical equation resulting from various attempts to fit with the experimental data was proposed in [73]. This empirical equation is expressed via:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(\rho_s V_{sd}\right) = {} & \underbrace{\left(q_{dc} - V_r\left(\alpha_v \frac{\partial \rho_s}{\partial P} + (1-\alpha_v)\frac{\partial \rho_w}{\partial P} + (\rho_w - \rho_s)\frac{\partial \alpha_v}{\partial P}\frac{dP}{dt}\right) + (\rho_w - \rho_s) V_r \frac{\partial \alpha_v}{\partial \alpha_r}\frac{d\alpha_r}{dt}\right)}_{=:q_r} \alpha_r - \\ & \underbrace{q_f \frac{h_f - h_{fw}}{h_c} + \frac{1}{h_c}\left(\rho_s V_{sd}\frac{\partial h_s}{\partial P} + \rho_w V_{wd}\frac{\partial h_w}{\partial P} - (V_{sd} + V_{wd})\frac{\mathrm{d}P}{\mathrm{d}t} + m_d C_p \frac{\partial t_{\mathrm{sat}}}{\partial P}\right)}_{=:q_{cd}} - \\ & \underbrace{\frac{\rho_s}{\tau_d}(V_{sd} - V_{sd}^{\circ}) + \alpha_r q_{dc} + \alpha_r \beta_d (q_{dc} - q_r)}_{=:q_{sd}}. \end{aligned} \tag{6.9}$$

Here the mass balance of the steam bubbles under the water level is defined in terms of the condensation flow q_{cd}, in addition to the steam flow through the liquid surface q_{sd}. This flow is driven by the density difference of the mixture, in addition to the momentum of the flow q_r entering through the riser tubes.

In fact, many of the complex phenomena inside the drum can be captured by (6.16) using proper parameterizations of the constants listed in Table 6.1. Since we accounted for the distribution of the steam bubbles under the water level, we can now describe the level via a linearized behaviour expressed with the knowledge of the drum surface cross sectional area $A_{dl}\,[\mathrm{m}^2]$; that is

$$\begin{aligned} l_d &= \frac{V_{wd} - V_{sd}}{A_{dl}} := l_{wd} + l_{sd}, \\ \text{with: } V_{wd} &= V_{wt} - V_{dc} - (1-\alpha_v)V_r \end{aligned} \tag{6.10}$$

with $l_{wd}\,[\mathrm{m}]$ and $l_{sd}\,[\mathrm{m}]$ denoting variations of the level resulting from changes in the water and steam, respectively.

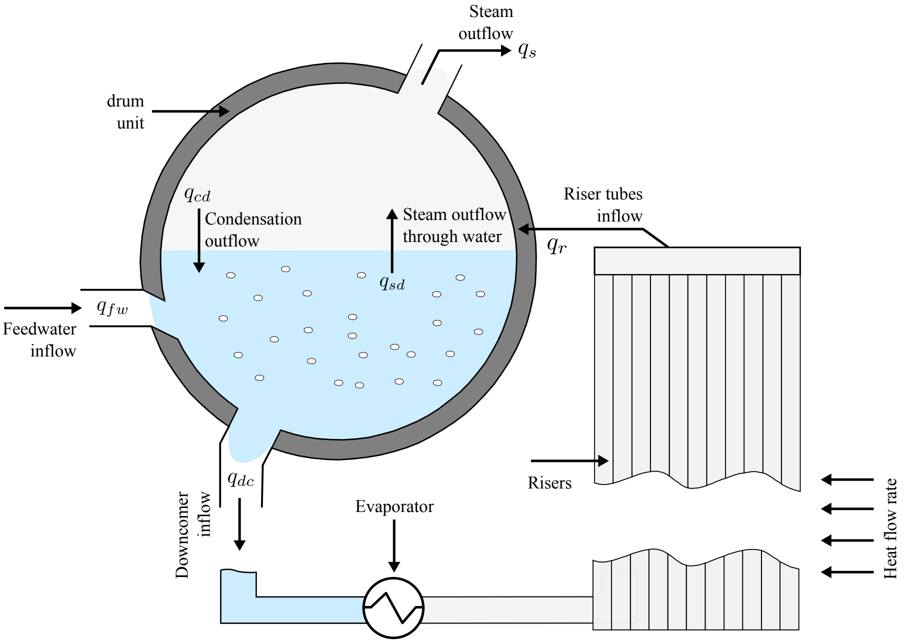


Figure 6.5: Schematic diagram of the downcomer-riser circulation loop.

6.3.2.4 ODE Model

Combining all previous results, we can formulate a nonlinear model expressed via a set of explicit ODEs. First we introduce the vector $\boldsymbol{x}_d := \big(P,\ V_{wt},\, \alpha_r,\, V_{sd}\big)^T$ to describe the state variables of the drum. Here the drum pressure P is obviously chosen being a state since it describes the total energy of the system, the accumulation of water related to the total water volume V_{wt} in the system is selected being a state since it represents the storage of mass, whereas the steam quality α_r in the riser tubes and the volume of the steam bubbles under the liquid level V_{sd} are chosen as state variables to describe distribution of steam under the water, thus making it possible to estimate the level as shown in (6.10). Additionally, we introduce the vectors $\boldsymbol{u}_d := \big(Q,\, q_{fw},\, q_s\big)^T$ and $\boldsymbol{y} := \big(P,\, l_d\big)^T$ being the vectors of the input variables and the measurable outputs.

After defining $V_t := V_{wt} + V_{st}$ and calculating the time derivatives in (6.4) and (6.5) via the chain rule, it is not hard to see that the mass- and energy-flow can be rewritten as

$$\begin{aligned} e_{11}\frac{\mathrm{d}V_{wt}}{dt} + e_{12}\frac{\mathrm{d}P}{\mathrm{d}t} &= q_{fw} - q_s, \\ e_{21}\frac{\mathrm{d}V_{wt}}{dt} + e_{22}\frac{\mathrm{d}P}{\mathrm{d}t} &= Q + q_{fw}h_{fw} - q_s h_s, \end{aligned} \tag{6.11}$$

with the variable coefficients e_{nm} expressed via:

$$\begin{aligned} e_{11} &= V_{wt}\frac{\partial \rho_w}{\partial P} + V_{st}\frac{\partial \rho_s}{\partial P}, \\ e_{12} &= \rho_w - \rho_s, \\ e_{21} &= V_{wt}\left(h_w\frac{\partial \rho_w}{\partial P} + \rho_w\frac{\partial h_w}{\partial P} - 1\right) + V_{st}\left(h_s\frac{\partial \rho_s}{\partial P} + \rho_s\frac{\partial h_s}{\partial P} - 1\right) + m_t C_p\frac{\partial t_{\mathrm{sat}}}{\partial P}, \\ e_{22} &= \rho_w h_w - \rho_s h_s. \end{aligned} \tag{6.12}$$

Here the thermal properties $\frac{\partial \rho_w}{\partial P}$[kg/J], $\frac{\partial \rho_s}{\partial P}$ [kg/J], $\frac{\partial h_w}{\partial P}$ [J/Kg · Pa], $\frac{\partial h_s}{\partial P}$[J/Kg · Pa], and $\frac{\partial t_{\mathrm{sat}}}{\partial P}$[K/Pa] resulted from the derivation of the density and enthalpy with respect to time and pressure. This is based on the fact their corresponding values, obtained from steam tables, change depending on the saturation pressure, which in turn varies over time.

In order to describe (6.6) and (6.7) as an ODE similarly to (6.11), we first eliminate q_r from (6.6) and (6.7) by multiplying (6.7) by the factor $(h_w + \alpha_r h_c)$, then adding the result to (6.6), and finally we evaluate the time-derivative via the chain rule. Thus the distribution of steam within the naturally circulated loop becomes:

$$e_{31}\frac{\mathrm{d}P}{dt} + e_{33}\frac{\mathrm{d}\alpha_r}{dt} = Q - \alpha_r q_{dc} h_c, \tag{6.13}$$

whose variable coefficients e_{nm} are expressed by:

$$\begin{aligned} e_{31} &= \left(\rho_w\frac{\partial h_w}{\partial P} - \alpha_r h_c\frac{\partial \rho_w}{\partial P}\right)(1-\alpha_v)V_r + \left(\rho_s\frac{\partial h_s}{\partial P} + (1-\alpha_r)h_c\frac{\partial \rho_s}{\partial P}\right)\alpha_v V_r + \\ &\quad \left((\rho_s + \alpha_r(\rho_w - \rho_s))h_c\frac{\partial \alpha_v}{\partial P} - 1\right)V_r + m_r C_p\frac{\partial t_{\mathrm{sat}}}{\partial P}, \\ e_{33} &= (\rho_s + \alpha_r(\rho_w - \rho_s))h_c V_r\frac{\partial \alpha_v}{\partial \alpha_r}, \end{aligned} \tag{6.14}$$

where:

$$\begin{aligned} \frac{\partial \alpha_v}{\partial \alpha_r} &:= \frac{\rho_w}{\rho_s \zeta}\left(\frac{\ln(1+\zeta)}{\zeta} - \frac{1}{1+\zeta}\right), \\ \frac{\partial \alpha_v}{\partial P} &:= \frac{1}{(\rho_w - \rho_s)^2}\left(\rho_w\frac{\partial \rho_s}{\partial P} - \rho_s\frac{\partial \rho_w}{\partial P}\right)\left(1 + \frac{\rho_w}{\rho_s(1+\zeta)} - \frac{\rho_s + \rho_w}{\zeta \rho_s}\ln(1+\zeta)\right), \\ \zeta &:= \alpha_r\frac{(\rho_w - \rho_s)}{\rho_s}. \end{aligned} \tag{6.15}$$

After tedious, but straight forward calculations, we can express (6.9) in terms of the drum pressure and the steam quality inside the riser tube according to:

$$e_{41}\frac{\mathrm{d}P}{\mathrm{d}t} + e_{43}\frac{\mathrm{d}\alpha_r}{\mathrm{d}t} + \rho_s\frac{\mathrm{d}V_{sd}}{\mathrm{d}t} = \frac{1}{\tau_d}(V_\circ - V_{sd}) - q_{fw}\frac{h_{fw} - h_w}{h_c}, \tag{6.16}$$

with the variables e_{nm} described by:

$$\begin{aligned}
e_{41} &= V_{sd}\frac{\partial \rho_s}{\partial P} + \alpha_r(1+\beta_d)V_r\left(\alpha_v\frac{\partial \rho_s}{\partial P} + (1-\alpha_v)\frac{\partial \rho_w}{\partial P} + (\rho_s - \rho_w) + \frac{\partial \alpha_v}{\partial P}\right) - \\
&\quad \frac{1}{h_c}\left(\rho_s V_{sd}\frac{\partial h_s}{\partial P} + \rho_w\left(V_{wt} - V_{dc} - (1-\alpha_v)V_r\right)\frac{\partial h_w}{\partial P} - (V_{sd} + V_{wd})\frac{\mathrm{d}P}{\mathrm{d}t} + m_d C_p\frac{\partial t_{\mathrm{sat}}}{\partial P}\right), \\
e_{43} &= \alpha_r(1+\beta_d)(\rho_w + \rho_s)V_r\frac{\partial \alpha_v}{\partial P}.
\end{aligned} \tag{6.17}$$

Finally with simple arrangements of (6.11), (6.13) and (6.16), the drum model can be expressed via the following set of ODEs:

$$\begin{aligned}
\frac{\mathrm{d}P}{\mathrm{d}t} &= \left[\frac{e_{11}Q + q_{fw}(e_{11}h_{fw} - e_{21}) - q_s(e_{21} - e_{11}h_s)}{e_{11}e_{22} - e_{12}e_{21}}\right], \\
\frac{\mathrm{d}V_{wt}}{\mathrm{d}t} &= \left[\frac{Q + q_{fw}h_{fw} - q_s h_s - e_{22}\,{}^{\mathrm{d}P}\!/_{\mathrm{d}t}}{e_{21}}\right], \\
\frac{\mathrm{d}\alpha_r}{\mathrm{d}t} &= \left[\frac{Q - \alpha_r q_{dc}(h_s - h_w) - e_{31}\,{}^{\mathrm{d}P}\!/_{\mathrm{d}t}}{e_{33}}\right], \\
\frac{\mathrm{d}V_{sd}}{\mathrm{d}t} &= \left[\frac{1}{\tau_d}(V_{sd}^{\circ} - V_{sd}) - q_{fw}\frac{h_{fw} - h_w}{\rho_s(h_s - h_w)} - \frac{e_{41}\,{}^{\mathrm{d}P}\!/_{\mathrm{d}t} + e_{43}\,{}^{\mathrm{d}\alpha_r}\!/_{\mathrm{d}t}}{\rho_s}\right].
\end{aligned} \tag{6.18}$$

6.3.3 Controller Model

As mentioned earlier, the design and implementation of the controller to regulate the water level and the pressure inside the drum was part of a previously published work, see [45, 46]; hence we only briefly discuss the overall methodology and governing mathematical model.

Table 6.1: Parameters of the steam-drum model

Variable	Description	Unit
V_d	Drum volume	[m^3]
V_r	Riser volume	[m^3]
V_{dc}	Downcomer volume	[m^3]
V_{sd}°	Hypothetical volume of the steam	[m^3]
τ_d	Residence time of steam inside drum	[s]
β_d	Empirical coefficient in (6.9)	[−]
F_d	Friction coefficient in downcomer-riser loop	[−]
C_p	Metal specific heat capacity	[J/kg K]
t_{sat}	Saturation temperature	°C
A_{dl}	Surface area of drum Water level	[m^2]
A_{dc}	Downcomer cross-sectional area	[m^2]

Fig. 6.6 illustrates the block diagram of the proposed state-feedback controller which is later implemented in the distributed control system (DCS) Mauell ME-4012 employed in the power plant München Süd GuD 2. This controller was commissioned in November 2014 and replaced the industry-standard 3-element PID-controller. Fig. 6.7 illustrates the human machine interface (HMI) of the controller which is comprised of a state observer (bottom, left-side) and a state-feedback controller with an Integral controller (top, left-side).

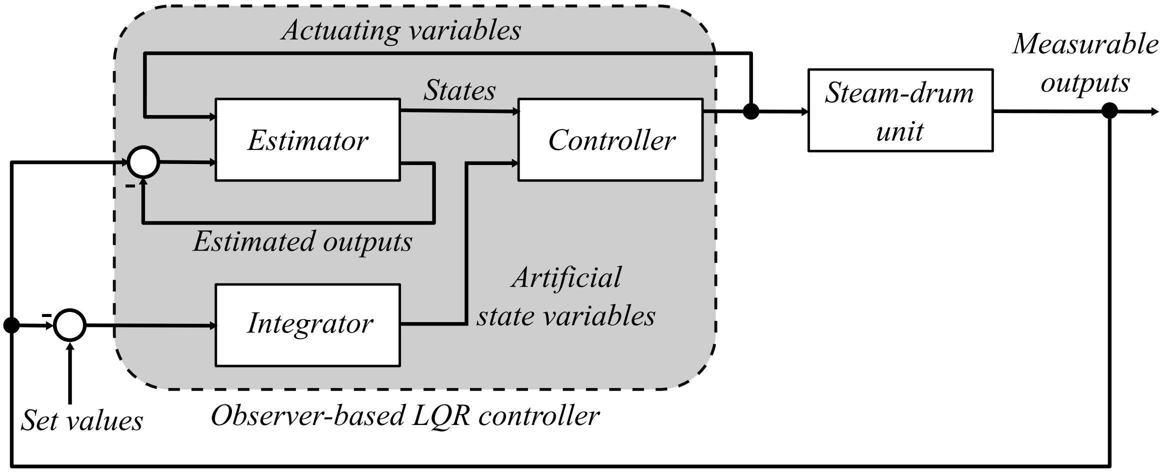


Figure 6.6: Schematic diagram of the LQR controller realized in the power plant München Süd GuD 2 to control the pressure and the water level inside the low-pressure steam-drum unit.

The additional integrators ensure that the drum pressure (6.18) and water level (6.10) track their corresponding reference signals, which we denote via w_p and w_l. Hence, the vector of integrated errors $\boldsymbol{h} := (\eta,\, \psi)^T$ is treated as an artificial state variable such that:

$$\begin{aligned}\dot{\eta} &= w_P - P,\\ \dot{\psi} &= w_l - l_d.\end{aligned} \tag{6.19}$$

Let $\hat{q}_{fw}$ [kg/s] and $\hat{r}_s$[%] denote the control action of the state-feedback controller corresponding to the feedwater flow rate and the steam valve opening percentage, respectively. These signals are generated by the controller to preserve the water level and pressure at the desired reference values according to

$$\begin{pmatrix} \hat{q}_{fw} \\ \hat{r}_s \end{pmatrix} = \boldsymbol{K} \cdot \begin{pmatrix} \boldsymbol{x}_d \\ \boldsymbol{h} \end{pmatrix}, \tag{6.20}$$

where $\boldsymbol{K}$ is the state feedback matrix and $\boldsymbol{x}_d := \big(P,\, V_{wt},\, \alpha_r,\, V_{sd}\big)^T$ is the vector of the dynamic state variables of the drum unit, see (6.18).

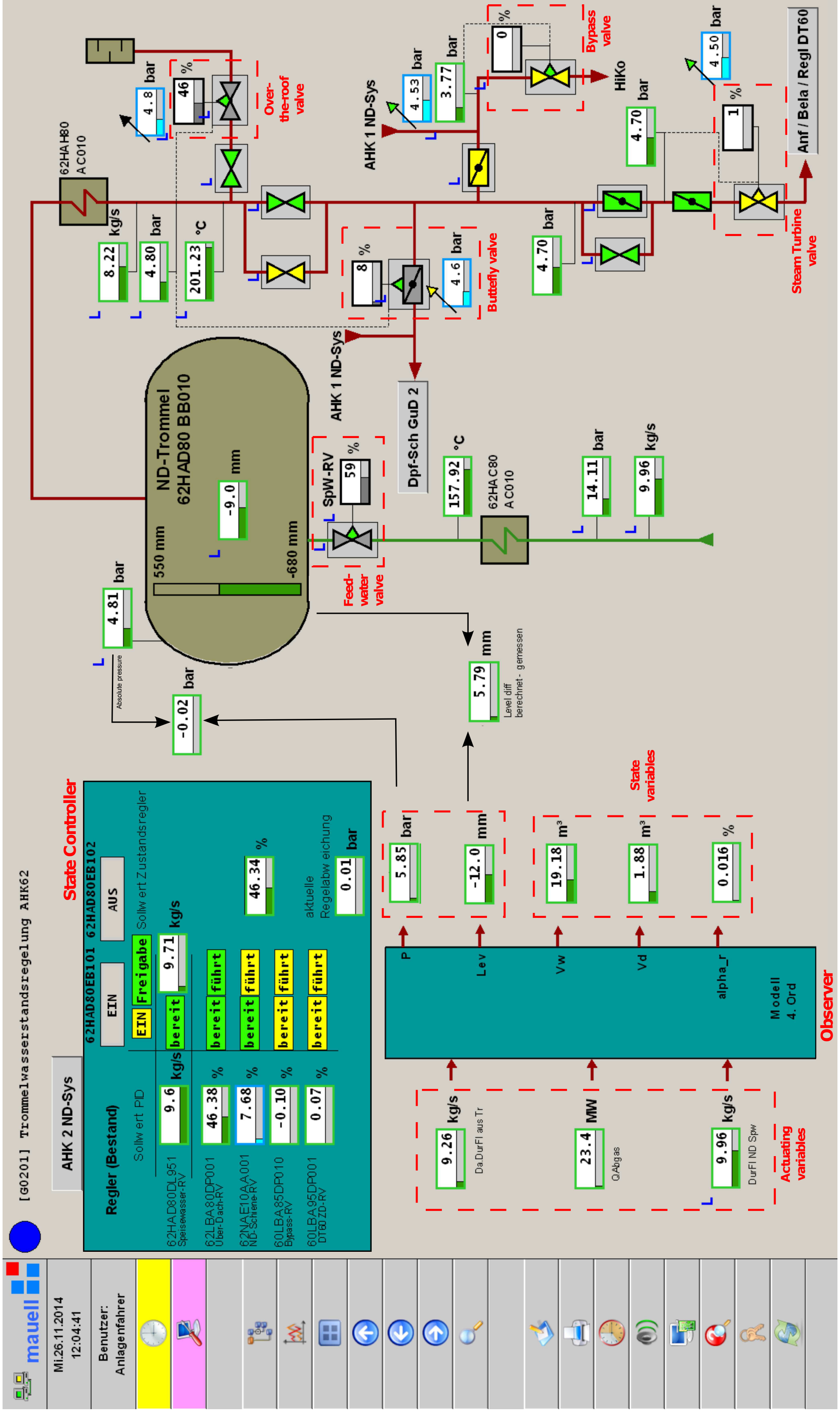


Figure 6.7: Screenshot of the distributed control system (DCS) - Mauell ME-4012 illustrating the human machine interface (HMI) of the drum unit (right) alongside the observer-based LQR controller (left).

The control action of the feedwater flow rate $\hat{q}_{fw}$ is compared to the actual flow rate, and the deviation between both generates the controller output of the feedwater valve opening percentage $\hat{r}_{fw}$ using another feedback loop employing a PI-controller

$$\hat{r}_{fw} = v_{pfw} \left(1 + v_{ifw} \int_0^t (\hat{q}_{fw}(\tau) - q_{fw}(\tau)) d\tau \right), \tag{6.21}$$

where $v_{p,fw}$ and $v_{i,fw}$ are the proportional and integrator scalar gains of the PI-controller.

6.3.4 Model of the Input Variables

So far we only introduced the mathematical model governing dynamics of the drum unit and the controller employed in the closed-loop. In this subsection we extended the model to account for modelling of the regulation of the mass- and heat-flow rates using their corresponding valves. To easily represent the remaining equations, let κ [kg/h] denote the valve sizing coefficient, r [%] being the valve opening percentage, ν and τ specify the gain and time constant of a first-order lag element, and the subscripts s, fw, and D refer to steam, feedwater, and current demand, respectively.

6.3.4.1 Heat-flow Rate

The heat flow rate Q is in fact considered as an additional state and not treated as an input variable due to the combined-cycle nature of the process; that is, the heat is directly associated with the gas turbine exhaust temperature which in turn corresponds to the electrical power demand P_D [MW] established by the TSO. The commanded power is basically the set value assigned by the automatic generator control (AGC) to balance load and demand within the control area of the TSO as explained earlier in the introductory section. It is worth mentioning that Q becomes a control variable in stand-alone thermal plants as it can be regulated directly by adjusting the boiler firing rate.

The gas turbine electrical output in the power plant changes with ramp function whose slope is $\frac{2}{15} \frac{\text{MW}}{\text{s}}$. Experimental observation at München Süd GuD 2 shows that the heat-flow rate can be related linearly to the gas turbine output power a first-order lag element transfer function, that is:

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \frac{\nu_D P_D - Q}{\tau_D}. \tag{6.22}$$

6.3.4.2 Feedwater-flow Rate

The supplied cold water inflow q_{fw} from the feedwater pump is regulated using one fast-acting control valve. This flow can be expressed via a nonlinear relationship often employed by mechanical engineers to size valves and meet mass-flow requirements. The nonlinear equation is expressed via (see [120]):

$$q_{fw} = r_{fw} \frac{\kappa_f \rho_w \sqrt{\Delta P \cdot 1\text{E-}5}}{3600} \tag{6.23}$$

with κ [kg/h] denoting the valve sizing coefficient, ΔP [Pa] is the pressure drop across the feedwater valve, and r_{fw} [%] is the valve opening percentage. The valve is controlled using a stepper motor to drive the valve gates to the desired opening percentage. A common approximation to model dynamics of the motor is:

$$\frac{\mathrm{d}r_{fw}}{\mathrm{d}t} = \frac{\nu_{fw}\hat{r}_{fw} - r_{fw}}{\tau_{fw}}, \tag{6.24}$$

with $\hat{r}_{fw}$ being the opening percentage set-value established by the PI-controller of the cascaded control loop, see (6.21).

6.3.4.3 Steam-flow Rate

Similarly to the regulation of feedwater, we employ a common approximation employed in practice which is expressed via

$$\begin{aligned} q_s &= r_s \cdot \frac{Z_s\,\kappa_s\sqrt{\rho_s(P \cdot 1\text{E-}5)}}{3600}, \\ \frac{\mathrm{d}r_s}{\mathrm{d}t} &= \frac{\nu_s\hat{r}_s - r_s}{\tau_s}, \end{aligned} \tag{6.25}$$

with Z_s being a dimensionless compressibility factor associated with the valve type, e.g. butterfly, fast-acting, or bypass.

Combining all previous results (6.18)-(6.25), one obtains an 11-th order model with the input signals $\boldsymbol{u} = \left(P_D,\, w_p,\, w_l\right)^T$, the state variables $\boldsymbol{x} = \left(P,\, V_{wt},\, \alpha_r,\, V_{sd},\, \eta,\, \psi,\, Q,\, \hat{r}_{fw},\, r_{fw},\, r_s,\, \hat{q}_{fw}\right)^T$, and the output vector $\boldsymbol{y} = \left(P,\, l_d\right)^T$. The model parameters described in Table 6.1 are provided in [41, p. 20].

6.4 Model Abstraction

In this section, we present how to obtain the polynomial model $\boldsymbol{P}(\hat{\boldsymbol{x}}, \boldsymbol{u})$ of the inclusion (6.3) using Taylor expansion. We then illustrate the technique of linear output injection to guarantee asymptotic estimates of the state variables. Note that the estimation of the state variables is necessary as well for the multivariable feedback controller described in the previous section. Finally, we construct the set of modelling errors in an over-approximative manner so that the proposed abstraction includes all dynamic responses of the real system (6.1). It should be noted that our approach is systematic but not formal; however, to the best of our knowledge there exists no formal way to translate reality into a mathematical representation.

6.4.1 Polynomial Approximation using Taylor Expansion

The motivation behind the choice of the polynomial approximation is its ability to represent complex mathematical expressions, present in the model of the drum unit, via a simplified form without sacrificing the model accuracy and its fit to measurement data. This abstraction makes it possible to substantially reduce the computational time required to compute the reachable set in real-time control as shown later in Sec. 6.5 when we illustrate the computational time using different models.

First we combine the state and input variables into one vector $\boldsymbol{z} := \left(\boldsymbol{x}^T,\, \boldsymbol{u}^T\right)^T \in \mathbb{R}^{n_z}$. This allows one to reformulate the nonlinear system (6.1) into

$$\begin{gathered}\dot{\boldsymbol{x}} = \boldsymbol{P}(\boldsymbol{z}) = \underbrace{\sum_{j=1}^{n_z} \frac{\partial \boldsymbol{f}(\boldsymbol{z})}{\partial z_j}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_{l*}} \Delta z_j}_{=:\boldsymbol{A}\Delta\boldsymbol{x}+\boldsymbol{B}\Delta\boldsymbol{u}} + \boldsymbol{\Omega}(\boldsymbol{z}),\\ \text{with } \Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_{l*},\ \Delta\boldsymbol{u} := \boldsymbol{u} - \boldsymbol{u}_{l*},\ \Delta\boldsymbol{z} := \boldsymbol{z} - \boldsymbol{z}_{l*}.\end{gathered} \tag{6.26}$$

where $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$ and $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$ are the time-invariant system and input matrices evaluated at the linearization point $\boldsymbol{z}_{l*} := \left(\boldsymbol{x}_{l*}^T, \boldsymbol{u}_{l*}^T\right)^T$, and the locally Lipschitz continuous function $\boldsymbol{\Omega}(\boldsymbol{z})$ contains all higher order terms of the Taylor expansion, that is

$$\begin{aligned}\Omega_i(\boldsymbol{z}) = f_i(\boldsymbol{z}_{l*}) &+ \frac{1}{2!}\sum_{j=1}^{n_z}\sum_{k=1}^{n_z} \frac{\partial^2 f_i(\boldsymbol{z})}{\partial z_j \partial z_k}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_{l*}} \Delta z_j \Delta z_k + \frac{1}{3!}\sum_{j=1}^{n_z}\sum_{k=1}^{n_z}\sum_{l=1}^{n_z} \frac{\partial^3 f_i(\boldsymbol{z})}{\partial z_j \partial z_k \partial z_l}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_{l*}} \Delta z_j \Delta z_k \Delta z_l \\ &+ \dots + \frac{1}{\sigma!}\sum_{o=1}^{n_z}\cdots\sum_{k=1}^{n_z}\sum_{l=1}^{n_z} \frac{\partial^\sigma f_i(\boldsymbol{z})}{\partial z_o \dots \partial z_k \partial z_l}\bigg|_{\boldsymbol{z}=\boldsymbol{z}_{l*}} \Delta z_o \dots \Delta z_k \Delta z_l + \dots\ ,\end{aligned} \tag{6.27}$$

with the subscript i denoting the i-th coordinate of the function $\boldsymbol{f}(\boldsymbol{z})$.

6.4.2 Linear Output Injection

The vector $\hat{\boldsymbol{z}} := \left(\hat{\boldsymbol{x}}^T, \boldsymbol{u}^T\right)^T$ and the variable $\Delta\hat{\boldsymbol{x}} := \hat{\boldsymbol{x}} - \boldsymbol{x}_{l*}$ are introduced for further derivation. The concept of linear output injection was first proposed in [92] to construct the so-called *Luenberger observer*. The main idea is to use the discrepancy between actual measurement data $\boldsymbol{y}$ and observable output $\hat{\boldsymbol{y}}$, i.e. to use the output error as a correction term applied to a feedback matrix $\boldsymbol{L}$ that decays the modelling errors over time. Hence the evolution of $\dot{\hat{\boldsymbol{x}}}$ may be expressed by a polynomial observer modelled as

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}}(t) &= \hat{\boldsymbol{\phi}}(\hat{\boldsymbol{z}}(t), \boldsymbol{e}(t))\\ &= \boldsymbol{A}\Delta\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\Delta\boldsymbol{u}(t) + \boldsymbol{\Omega}(\hat{\boldsymbol{z}}(t)) + \boldsymbol{L}\underbrace{\boldsymbol{C}(\boldsymbol{\Delta x}(t) - \Delta\hat{\boldsymbol{x}}(t))}_{=:\boldsymbol{e}(t)},\end{aligned} \tag{6.28}$$

with $\hat{\boldsymbol{\phi}} : \mathbb{R}^{n_x+n_u+n_y} \mapsto \mathbb{R}^{n_x}$.

Constructing a stable observer, however, is not obvious due to the nonlinear dynamics of the modelling errors $\dot{\boldsymbol{e}}(t) := \dot{\boldsymbol{x}}(t) - \dot{\hat{\boldsymbol{x}}}(t)$. In fact, placing the poles of the observer far into the left of the complex-plane does not necessarily guarantee stability [115]. Furthermore, the matrix $\boldsymbol{L}$ is not unique, due to extra degrees of freedom arising in multivariable systems [75]. We assume that the matrix $\boldsymbol{L}$ is already chosen and focus on examining the asymptotic stability of the dynamics of the modelling errors. It can be seen from (6.26) and (6.28) that

$$\dot{\boldsymbol{e}}(t) = (\boldsymbol{A} - \boldsymbol{LC})(\Delta\boldsymbol{x}(t) - \Delta\hat{\boldsymbol{x}}(t)) + (\boldsymbol{\Omega}(\boldsymbol{z}(t)) - \boldsymbol{\Omega}(\hat{\boldsymbol{z}}(t)))\,. \tag{6.29}$$

Theorem 6.1. Observer stability [130]: If the feedback correction matrix $\boldsymbol{L}$ is chosen to hold

$$\Psi < \frac{\boldsymbol{\lambda}_{\min}(\boldsymbol{Q})}{2\boldsymbol{\lambda}_{\max}(\boldsymbol{P})}, \tag{6.30}$$

then (6.29) yields asymptotic stable estimates for (6.26). Here $\boldsymbol{Q}$ and $\boldsymbol{P}$ are the symmetric matrices of the continuous Lyapunov equation $(\boldsymbol{A} - \boldsymbol{LC})^T\boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A} - \boldsymbol{LC}) = -\boldsymbol{Q}$, $\boldsymbol{\lambda}(\,\cdot\,)$ returns the eigenvalues of a matrix, and Ψ is the constant of the Lipschitz nonlinearity $\boldsymbol{\Omega}$ that satisfies:

$$||\,\boldsymbol{\Omega}(\boldsymbol{z}) - \boldsymbol{\Omega}(\hat{\boldsymbol{z}})\,|| \leq \Psi\,||\Delta\boldsymbol{x} - \Delta\hat{\boldsymbol{x}}\,||. \tag{6.31}$$

Remark 6.1. The theorem presented above only serves as a means to checking the observer stability. Due to its non-constructive nature, the theorem cannot be considered as a design approach. The design of the observer matrix $\boldsymbol{L}$ was a separate task related to the state-space controller commissioned in the power plant, see [41, p. 45].

6.4.3 Constructing the Set of the Modelling Errors

Now we present a systematic approach to construct the set of the modelling errors $\boldsymbol{\mathcal{E}}$ in an over approximative manner. The proposed over-approximation guarantees that all possible trajectories of the errors $\boldsymbol{e}(t)$, $t > 0$ for n simulations are included within the constructed set. This is achieved by comparing the output $\hat{\boldsymbol{y}}(t)$ within a simulation environment against validation data $\boldsymbol{y}(t)$ with very a rich excitation that covers the entire operational range of the process, thus including all possible values of the errors resulting from n scenarios.

The set of the modelling errors $\boldsymbol{\mathcal{E}}$ is obtained by introducing a closed interval such that

$$\mathcal{E}_i := [-\gamma_i,\, \gamma_i] = \{a \in \mathbb{R} \,:\, -\gamma_i \leq a \leq \gamma_i\}\,, \tag{6.32}$$

$$\text{with}\begin{cases} \gamma_i := \max\limits_{k\,:1\ldots n}\left(\delta_i^k\right), \\ \delta_i^k := \max\limits_{t\in[0,t_f]}\left\{e_i^k(t) \in \mathbb{R}^+ \,:\, e_i^k(t) = \left|y_i^k(t) - \hat{y}_i^k(t)\right|\right\}, \end{cases}$$

where the max(·) operator is applied element-wise to return the maximum value, the subscript i and the superscript k denote the i-th measurable output and the k-th experiment, respectively, δ is the maximum of the absolute value of the error in the k-th experiment, while γ is the maximum error resulting from n simulations.

Clearly, in (6.32) there is a tradeoff between the size of n and the accuracy of the resulting set, which can be measured by computing the volume of $\boldsymbol{\mathcal{E}}$

$$\mathrm{vol}(\boldsymbol{\mathcal{E}}) = 2^{n_y} \cdot \textstyle\prod_{i=1}^{n_y} \gamma_i. \tag{6.33}$$

Clearly, this volume approaches the true value as n is closer to infinity; however, it is not possible to simulate infinitely many possible scenarios. Thus, we heuristically choose n to hold

$$\frac{\mathrm{vol}(\overline{\boldsymbol{\mathcal{E}}}) - \mathrm{vol}(\underline{\boldsymbol{\mathcal{E}}})}{\mathrm{vol}(\overline{\boldsymbol{\mathcal{E}}})} \leq \epsilon, \tag{6.34}$$

where the set $\overline{\boldsymbol{\mathcal{E}}}$ is obtained by setting the number of simulations to be twice as large as the size of n which is required to construct $\underline{\boldsymbol{\mathcal{E}}}$, such that the value of ϵ becomes sufficiently small (e.g. 1E-3). The underlying idea behind (6.34) is that the double effort to conduct further simulations is not justified because the percentage of the improvement is negligible. Therefore, the chosen size of n reliably ensures that

$$\lim_{n\to\infty} \mathrm{vol}(\boldsymbol{\mathcal{E}}_\infty) \approx \mathrm{vol}(\overline{\boldsymbol{\mathcal{E}}}). \tag{6.35}$$

Here $\boldsymbol{\mathcal{E}}_\infty$ is the set of modelling errors constructed by running theoretically infinitely many simulations. After introducing a safety factor $\boldsymbol{\xi}$, one may choose $\boldsymbol{\mathcal{E}} := \boldsymbol{\xi} \cdot \overline{\boldsymbol{\mathcal{E}}}$.

6.4.4 Reachability Analysis

After the construction of the set of modelling errors according to (6.32)-(6.35), we still need to compute the reachable set of the abstract model, as illustrated shortly after. Similarly to the previous chapters, the reachability algorithm is based on abstracting the nonlinear dynamics of the system to linear differential inclusions (LDIs) for consecutive time intervals $\tau_k := [t_k, t_{k+1}]$. By using the model (6.28) for reachability analysis, we guarantee that all dynamic behaviours of the original system (6.1) are enclosed by computed reachable sets; since the polynomial observer takes into account the modelling errors which are obtained according to the procedure described in the previous subsection.

We introduce the vector $\boldsymbol{v} := \left(\hat{\boldsymbol{x}}^T,\, \boldsymbol{u}^T,\, \boldsymbol{e}^T\right)^T \in \mathbb{R}^{n_v}$, the linearization point $\boldsymbol{v}_k^* := \left(\boldsymbol{x}_k^{*,T},\, \boldsymbol{u}_k^{*,T},\, \boldsymbol{e}_k^{*,T}\right)^T$, and the axillary variable $\Delta\hat{\boldsymbol{x}}_k := \hat{\boldsymbol{x}} - \boldsymbol{x}_k^*$. Hence, the nonlinear model (6.28) can be abstracted by a first-order Taylor expansion with the Lagrangian remainder $\boldsymbol{\mathcal{L}}$, see Prop. 2.1

$$\begin{aligned}
&\forall t \in [t_k, t_{k+1}] : \\
&\qquad \dot{\hat{\boldsymbol{x}}} \in \underbrace{\sum_{j=1}^{n_x} \frac{\partial \hat{\boldsymbol{\phi}}(\boldsymbol{v})}{\partial x_j}\bigg|_{\boldsymbol{v}=\boldsymbol{v}_k^*} \Delta \hat{x}_j}_{=:\hat{\boldsymbol{A}}_k \Delta\hat{\boldsymbol{x}}_k} \oplus \hat{\boldsymbol{\phi}}(\boldsymbol{v}_k^*) \oplus \underbrace{\sum_{j=1}^{n_u} \frac{\partial \hat{\boldsymbol{\phi}}(\boldsymbol{v})}{\partial u_j}\bigg|_{\boldsymbol{v}=\boldsymbol{v}_k^*} \Delta u_j}_{=:\hat{\boldsymbol{B}}_k \Delta\boldsymbol{u}_k} \oplus \underbrace{\sum_{j=1}^{n_y} \frac{\partial \hat{\boldsymbol{\phi}}(\boldsymbol{v})}{\partial e_j}\bigg|_{\boldsymbol{v}=\boldsymbol{v}_k^*} \Delta e_j}_{=:\boldsymbol{L}\Delta\boldsymbol{e}_k} \oplus \boldsymbol{\mathcal{L}}, \\
&\qquad\quad \in \hat{\boldsymbol{A}}_k \Delta\hat{\boldsymbol{x}}_k \oplus \underbrace{\hat{\boldsymbol{\phi}}(\boldsymbol{v}_k^*) \oplus \hat{\boldsymbol{B}}_k \cdot \boldsymbol{\mathcal{U}} \oplus \boldsymbol{L} \cdot \boldsymbol{\mathcal{E}} \oplus \boldsymbol{\mathcal{L}}}_{=:\hat{\boldsymbol{\mathcal{U}}}(\tau_k)}
\end{aligned} \tag{6.36}$$

with $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{B}}_k$ being the system and input matrices of the LDI, respectively, and $\hat{\boldsymbol{\mathcal{U}}}$ is the set of uncertain inputs.

Remark 6.2. The linearization of (6.28) is performed at each time interval τ_k, whereas the expansion using Taylor series of (6.1) expressed by (6.26) is performed once to obtain the polynomial model as proposed by our generic approach as in Fig. 6.2.

In the LDI (6.36), the set $\boldsymbol{\mathcal{L}}$ denotes the Lagrangian remainder containing all possible linearization errors within the time interval τ_k; that is

$$\boldsymbol{\mathcal{L}}(\tau_k) = \left\{ \boldsymbol{L} \in \mathbb{R}^{n_x} \,:\, L_j = \frac{1}{2} \sum_{l=1}^{n_v} \sum_{m=1}^{n_v} \frac{\partial^2 \hat{\phi}_j(\boldsymbol{v})}{\partial v_l \partial v_m}\bigg|_{\boldsymbol{v}=\boldsymbol{\mu}} \Delta v_l \Delta v_m,\, \boldsymbol{v} \in \boldsymbol{\mathcal{V}}(\tau_k),\, \boldsymbol{\mu} \in \textbf{interval}(\boldsymbol{\mathcal{V}}(\tau_k)) \right\},$$

with $\Delta\boldsymbol{v} := \boldsymbol{v} - \boldsymbol{v}_k^*$ and the subscript j corresponding to the j-th coordinate of the nonlinear observer $\hat{\boldsymbol{\phi}}(\boldsymbol{v})$. Here the operator **interval** returns the interval enclosure of a set as in (2.34) and $\boldsymbol{\mathcal{V}}(\tau_k)$ denotes the reachable set resulting from the following cartesian product:

$$\boldsymbol{\mathcal{V}}(\tau_k) := \boldsymbol{\mathcal{R}}(\tau_k) \times \boldsymbol{\mathcal{U}} \times \boldsymbol{\mathcal{E}}.$$

with $\boldsymbol{\mathcal{R}}(\tau_k)$ as the reachable set of the LDI (6.36) which is computed for one time-step as summarized in Alg. 1 and illustrated in Fig. 2.12. For further details, the reader is referred to Sec. 2.4.

Remark 6.3. As previously mentioned throughout the previous chapters, the computation of the linearization errors consumes roughly more than 95% of the CPU to obtain the reachable set. In this chapter, these errors are computed via a conservative approach resulting in larger over-approximation compared to the tight procedure described earlier in Sec. 2.4.3. However, this conservative approach makes it possible to compute the reachable set within the time-frame of secondary frequency control, thus meeting the practical requirements of verifying safety of the water level inside the drum unit in real-time control using reachability analysis.

Recall that we use zonotopes as a means of representing reachable sets; thus, based on remark 6.3, the set of Lagrangian remainders is over-approximated according to (see [13, Prop. 1])

$$\begin{aligned} & |\boldsymbol{\mathcal{L}}| \subseteq [\mathbf{0}, \boldsymbol{l}], \\ \text{with} \quad & l_j = \boldsymbol{\lambda}^T \max_{\boldsymbol{v} \in \boldsymbol{\mathcal{V}}(\tau_k)} \left\{ \left| \frac{\partial^2 \hat{\phi}_j(\boldsymbol{\mu})}{\partial \boldsymbol{v}^2} \right| \right\} \boldsymbol{\lambda},\ \boldsymbol{\mu} \in \left\{ \boldsymbol{\alpha v} + (1 - \boldsymbol{\alpha}) \boldsymbol{v}_k^*,\ 0 \leq \alpha_i \leq 1,\ \boldsymbol{v} \in \boldsymbol{\mathcal{V}}(\tau_k) \right\}, \\ \text{and} \quad & \boldsymbol{\lambda} = |\boldsymbol{c}_v - \boldsymbol{v}_k^*| + \sum_{i=1}^{p} |\boldsymbol{g}_v^{(i)}|, \end{aligned} \tag{6.37}$$

where $\boldsymbol{c}_v$ and $\boldsymbol{g}_v^{(i)}$ are the center and generators of the zonotope $\boldsymbol{\mathcal{V}}(\tau_k)$, respectively, see Def. 2.4 with regards to the representation of zonotopes.

Shortly after we illustrate how the proposed procedure can be employed in order to verify safety of the water level inside the drum in real-time control.

6.5 Results

First we will present the validation of the polynomial model (6.26) against measurement data from the power plant to show its ability to capture the dynamic behaviour of the real process. The validation data covers almost the entirety of the gas turbine operational range, i.e. from 70 MW to 120 MW in both directions (increased/decreased generation). The data was collected over a period of one year during the author's involvement with the power plant München Süd GuD 2 as a project engineer. We then illustrate the histogram distribution of the modelling errors when employing the technique of linear output injection, and investigate the safety of the water level against high-load transitions (≤ 40 MW) by computing the over-approximative reachable set of the polynomial observer (6.28). Finally, we outline the special findings using reachability analysis. All computations are performed on a standard computer with an Intel Core i7-4810MQ CPU and 16GB of RAM.

6.5.1 Validation of the Polynomial Model

The model is realized in MATLAB R2014b using the Symbolic toolbox, and is approximated by a polynomial function using the Taylor expansion at $P_D = 95$ MW (center of the gas turbine operational range). The validation procedure is carried out by simulating the models (1-st, 2-nd and 3-rd order polynomial functions) and comparing the simulation results to the experimental data. All simulations are performed using the `ODE-45` solver in MATLAB.

It is shown in Fig. 6.8 and Fig. 6.9 that the polynomial models replicate the dynamic behaviour of the real process. The model inaccuracy is acknowledged and identifiable and is caused by the nature of the proposed polynomial approximation, in addition to the simplification of certain components during the modelling procedure. Practically speaking, this inaccuracy can be deemed acceptable and is adequate for further analysis, i.e. design of the matrix $\boldsymbol{L}$, model-based control design, and computation of the reachable set as illustrated shortly after.

6.5.2 Validation using Linear Output Injection

The observer (6.28) is compared to multiple trajectories with a duration ranging between 2 h and 5 h. The comparison covered all major possible excitation conditions; that is, either slow or fast transition with small, medium and high load change. We only present three conditions: (a) fast transitions with small load changes equivalent to 10 MW, (b) slow transitions with medium load changes equivalent to 20 MW, and (c) fast transitions with high load changes equivalent to 40 MW (worst-case scenario). Fig. 6.10 shows the histogram distribution of the modelling errors e_p and e_l for the aforementioned scenarios using the 3-rd order polynomial model with linear output injection.

Table 6.2: Interval of the modelling errors for different models using linear output injection

Model	$\mathcal{E}_P$ [bar]	$\mathcal{E}_l$ [mm]
Linear model	$[-0.1100, 0.1100]$	$[-21.8374, 21.8374]$
2-nd order polynomial model	$[-0.0532, 0.0532]$	$[-14.3283, 14.3283]$
3-rd order polynomial model	$[-0.0512, 0.0512]$	$[-14.1288, 14.1288]$
Nonlinear model (6.11)-(6.18)	$[-0.0380, 0.0380]$	$[-10.6481, 10.6481]$

The interval of the modelling errors, which is obtained using (6.32) according to the procedure described in Sec. 6.4.3, is shown in Table 6.2. It is clear that the 1-st order (linear model) does not fit the experimental data well. The 4-th and higher order models are not considered since little improvement is achieved when comparing the error resulting from both the 2-nd and 3-rd order approximations; in other words, the complexity of using a higher polynomial approximation is not justified since it only results in a more complicated model which contradicts our goal of simplifying the complex nonlinear expressions found in (6.18), see derivation of the drum model in Sec. 6.3. Although the original model (6.11)-(6.18) has the minimum modelling errors, we illustrate in the following subsection the computational benefit of using a polynomial model. All subsequent results are over-approximated as we fully consider the modelling errors $\boldsymbol{\mathcal{E}}$ from Table 6.2.

6.5.3 Load-following Safety Verification

The reachable set is computed over a time-horizon $t_f = 5\,\text{min}$ with a time increment $t_r = 1\,\text{s}$ using the **Co**ntinuous **R**eachability **A**nalyser (CORA) toolbox [9]. When the plant München Süd GuD 2 is subjected to secondary frequency control, it is notified 5 min in advance by the TSO to meet a load change equivalent to 40 MW.

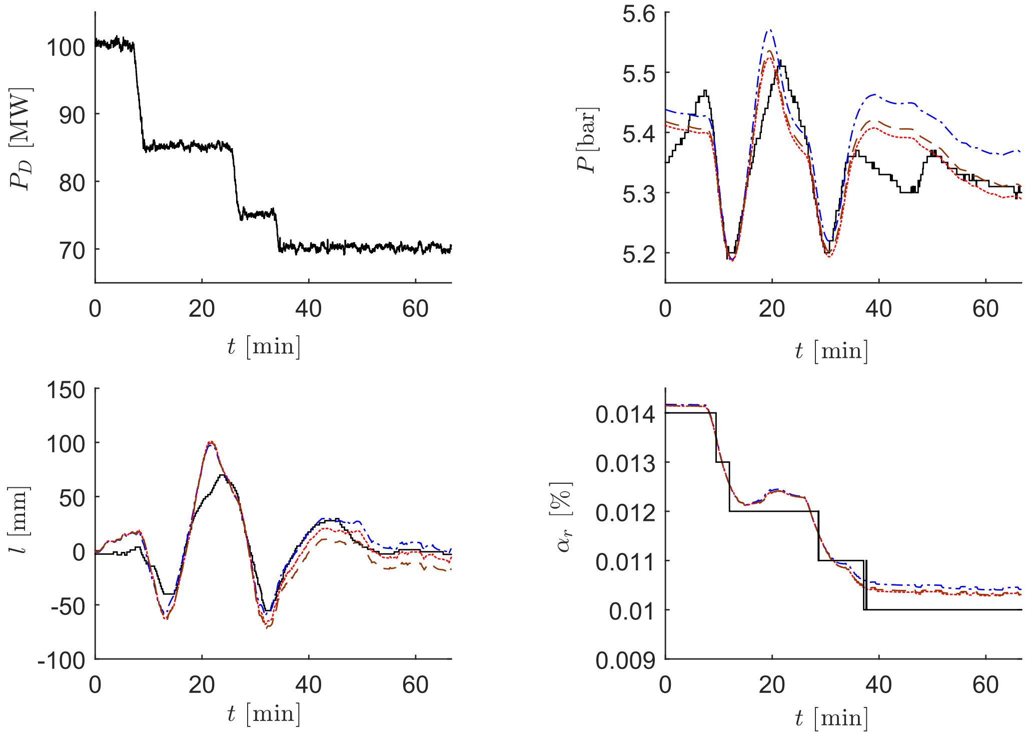


Figure 6.8: Comparison between measurement data (black solid line), 1-st (blue dashed-dotted line), 2-nd (red dotted line) and 3-rd (brown dashed line) order polynomial model for decrease of the gas turbine power from 100 MW to 70 MW.

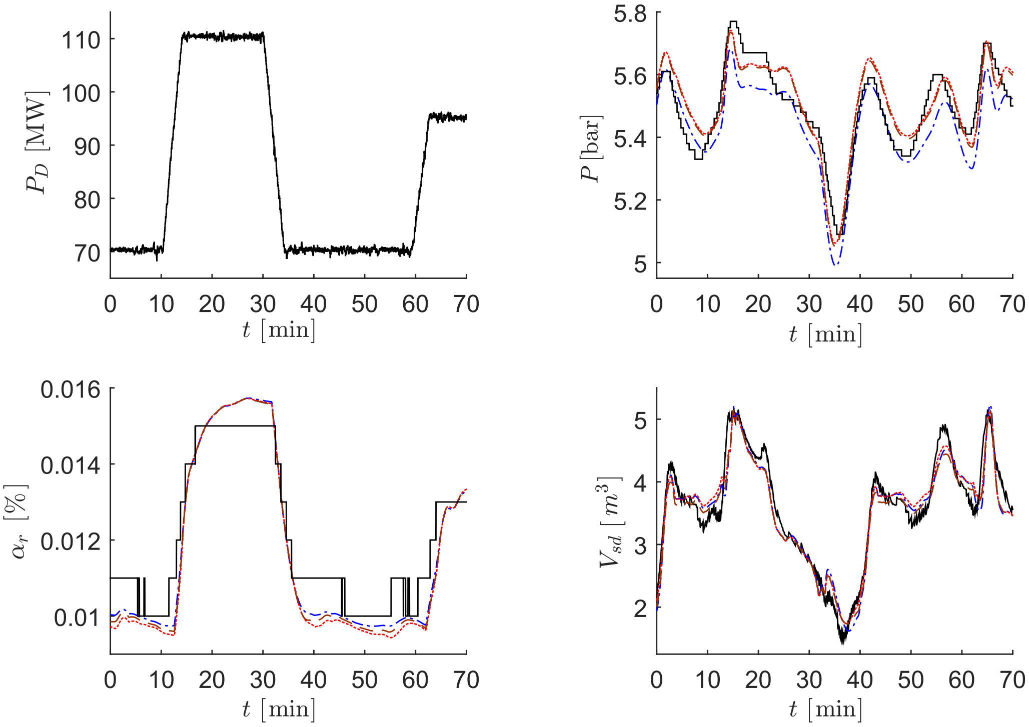
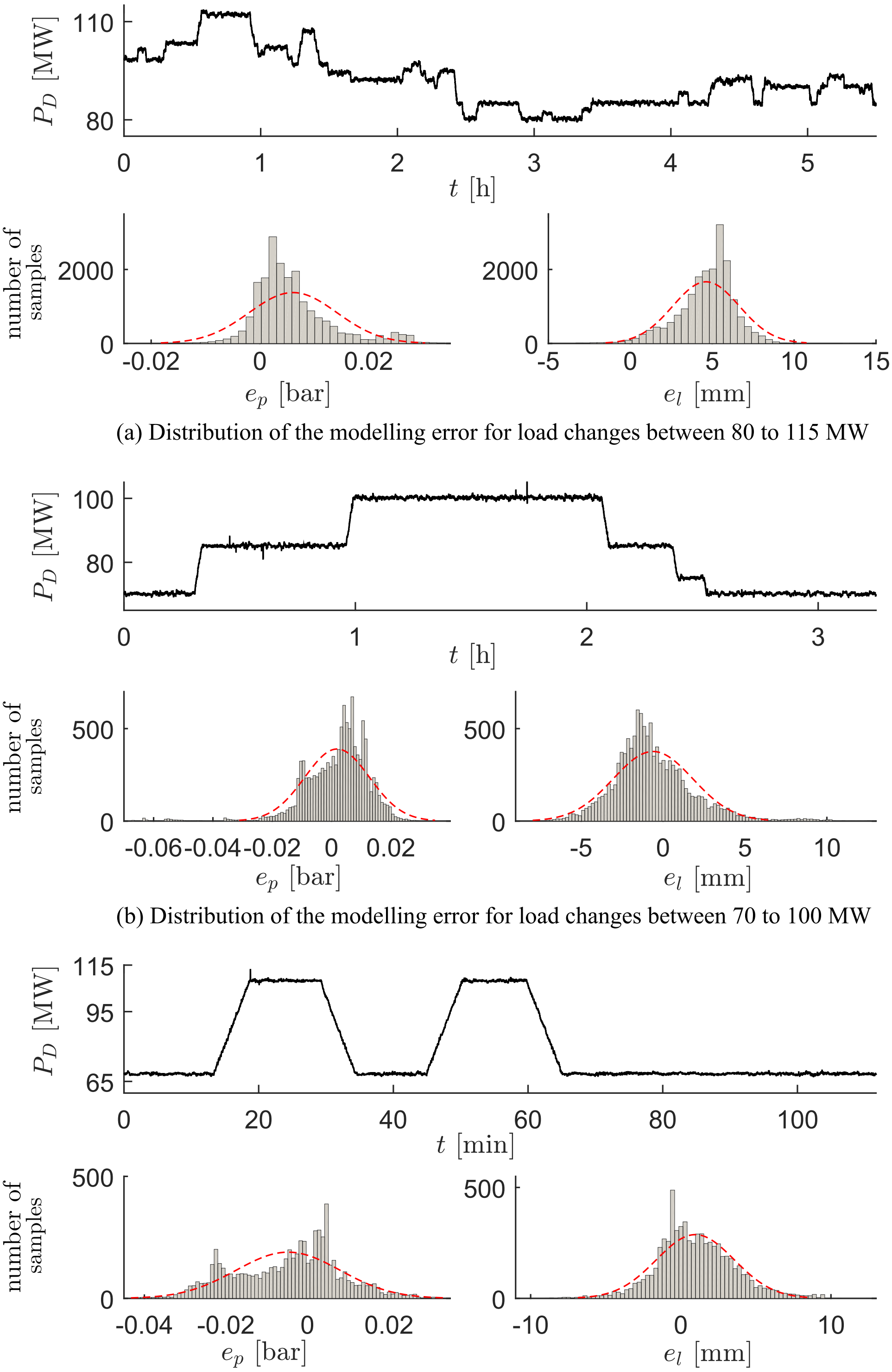


Figure 6.9: Comparison between measurement (black solid line), 1-st (blue dashed-dotted line), 2-nd (red dotted line) and 3-rd (brown dashed line) order polynomial model for perturbations of the gas turbine power.



(a) Distribution of the modelling error for load changes between 80 to 115 MW

(b) Distribution of the modelling error for load changes between 70 to 100 MW

(c) Distribution of the modelling error for load changes between 67 to 107 MW

Figure 6.10: Histogram distribution of the modelling errors e_p and e_l for different loading conditions: (a) rapid transitions for small load-changes, (b) slow transition for medium load-changes, (c) rapid transitions for high-load changes.

The task is to guarantee that the water level l inside the drum does not surpass $\pm 300\,\mathrm{mm}$ during the load transition. If the limits are triggered, the boiler is tripped as a safety precaution to protect critical components, e.g., the superheater and the steam-turbine. Recall that this previously occurred in the power plant on multiple occasions, see Fig. 6.1. During this time, the power plant is no longer operational and is unable to meet load requirements of the TSO.

The power demand P_D has a fixed trajectory for a load change of 40 MW in both directions (increased/decreased generation) which is the maximum load transition that can be requested by the TSO. The input P_D follows the maximum allowable load gradient that can be imposed on the plant ($8\,\frac{\mathrm{MW}}{\mathrm{min}}$). We include uncertainty to the initial set of the drum state variables when computing the reachable set, since initial states are not exactly known due to increasingly varying operating conditions in current power systems. We define the interval $\boldsymbol{U} := [-1, 1]$, and assign the center of the initial set as the steady state solution of (6.28) denoted by the superscript 0. Therefore, the initial drum pressure is $P(0) \in P^0 \oplus 1\mathrm{E}4 \cdot \boldsymbol{U}\,[\mathrm{Pa}]$, the initial volume of water is $V_{wt}(0) \in V_{wt}^0 \oplus 0.5 \cdot \boldsymbol{U}\,[\mathrm{m}^3]$, the initial steam quality is $\alpha_r(0) \in \alpha_r^0 \oplus 0.001 \cdot \boldsymbol{U}\,[-]$, the initial steam volume under the water level is $V_{sd}(0) \in V_{sd}^0 \oplus 0.5 \cdot \boldsymbol{U}\,[\mathrm{m}^3]$, and the initial heat flow rate is $Q(0) \in Q^0 \oplus 1\mathrm{E}5 \cdot \boldsymbol{U}\,[\mathrm{W}]$.

The time-domain bounds of the reachable set of the drum pressure and water level, and the reachable set projections of chosen state variables are illustrated in Fig. 6.11 and Fig. 6.12, respectively. The visualization of the 2-D projection is quite beneficial to the plant operators, since they are interested in the dynamic behaviour of the water level and pressure against the loading condition of the gas turbine. It can be seen that the water level does not reach the safety limit, hence the safety of the steam-drum unit is formally verified under maximum loading condition imposed by the TSO.

Fig. 6.13 illustrates the quality of the over-approximation resulting from the set of the uncertain modelling errors. We always have to account for the maximum error resulting from n simulations, otherwise we cannot formally guarantee that the specifications of our particular problem are always met under all eventualities. Finding simple but less conservative enclosures is certainly an interesting task and still remains an open question. In this example the time increment t_r is chosen to reduce conservatism of the solution, i.e. ensure that the computed reachable set is computed as tightly as possible.

Table 6.3 shows a comparison between the computational time of reachability analysis using different models. Using the 3-rd order polynomial model, it takes 206.37 s to compute the reachable set, where the calculation of the Lagrangian remainder consumes 96 % of the time. The computational time meets practical requirements of the plant as it allows online verification of the process safety for high-load transitions in the requested time ($\mathrm{t} \leq 5\,\mathrm{min}$). Using the nonlinear model (6.11)-(6.18) presented in Sec. 6.3 without a polynomial approximation, it takes 57.2 min to compute the reachable set for a time-horizon of 10 s.

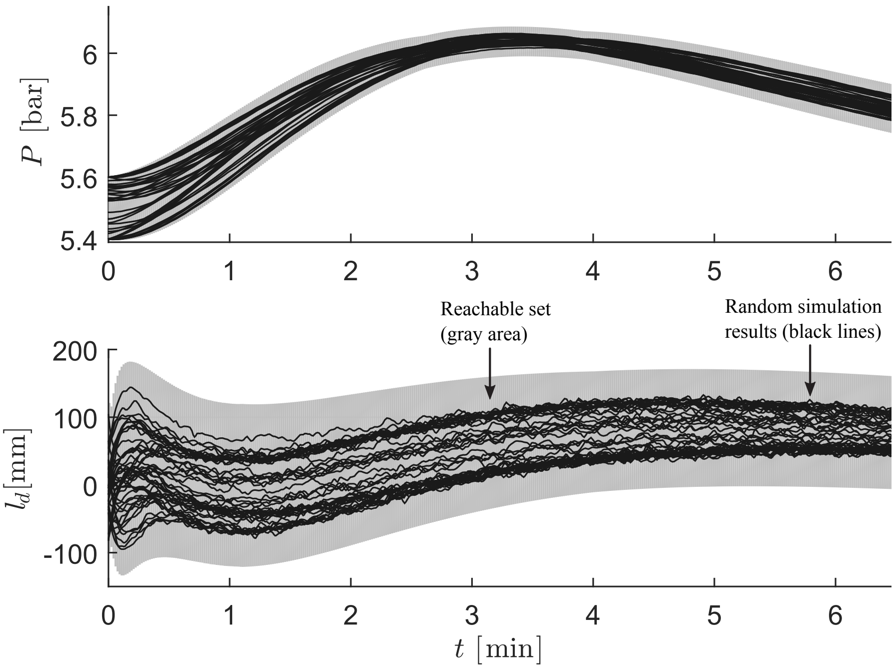


Figure 6.11: Time-domain bounds of the reachable set for a load-change of the gas turbine from 70 MW to 110 MW. Black lines represents random simulation results ($n = 50$), the gray area shows the reachable set.

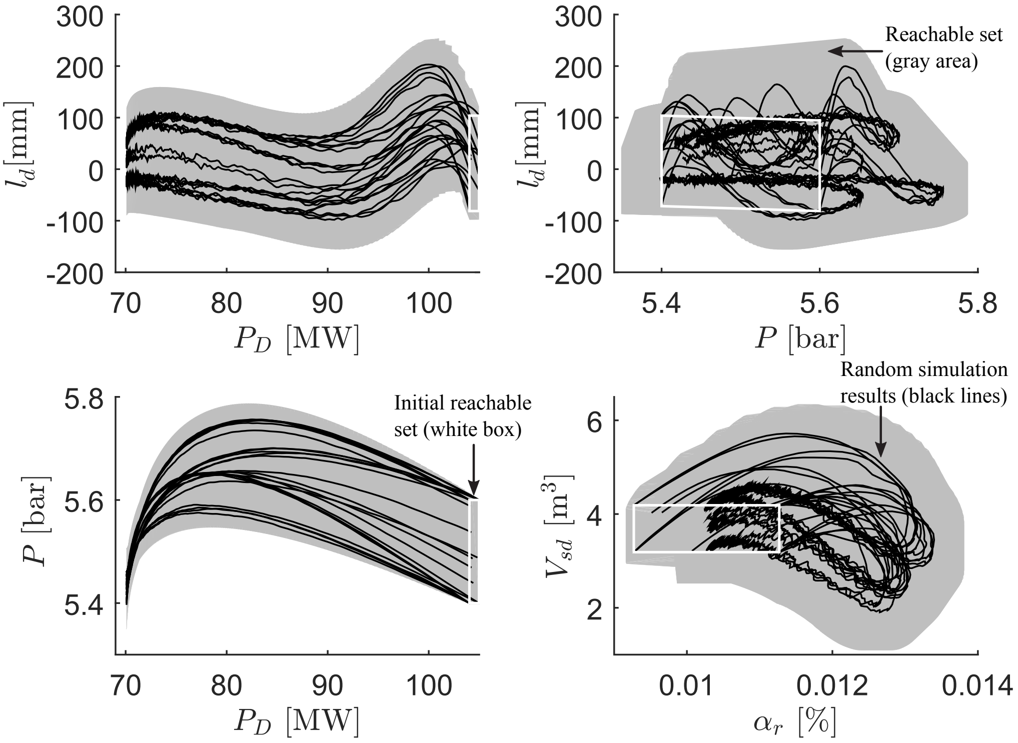


Figure 6.12: Selected projections of the reachable set for a load-change of the gas turbine from 110 MW to 70 MW. Black lines represent random simulation results ($n = 25$), the gray area shows the reachable set, the white box is the initial set of state variables $\boldsymbol{\mathcal{R}}(0)$.

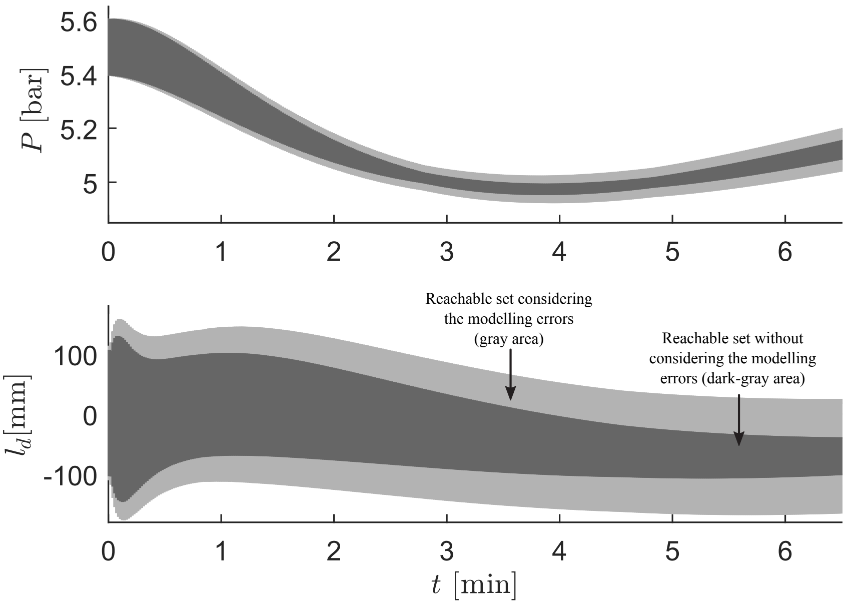


Figure 6.13: Comparison between the time-domain bounds of the reachable set without considering uncertain inputs (dark gray), and considering uncertain inputs (light gray) for a load-change of the gas turbine from 110 MW to 70 MW.

The huge difference in the computational time is due to the complicated expressions of the nonlinear paraments e_{nm}, see Sec. 6.3. These expressions were simplified with a polynomial function, thus in return substantially reduces the computational complexity making the algorithm feasible for practical problems.

6.5.4 Discussion of Results

The special findings using reachability analysis can be summarized as follows: we formally guarantee that the controller meets the performance specifications under all eventualities, because we have taken the modelling errors into account. Meeting performance specifications cannot be achieved using numerical simulations since they do not guarantee any formal properties. Furthermore, it is not possible to simulate infinitely many simulations corresponding to infinitely many possible uncertain input trajectories. Numerical simulations are indeed simple to implement, however, they are only useful to gain an initial idea about the system behaviour, see Fig. 6.1, Fig. 6.8 and Fig. 6.9.

The proposed reachability algorithm can be faster than sufficiently many numerical simulations. One simulation of our model takes approximately 0.49 s using the `ODE`-45 solver; however, the simulation of all corner cases requires 2^n simulations (exponential problem), where n is the number of states, consuming roughly 1012 s of computational time. On the contrary, using our approach, the computational time necessary to compute the reachable set is 41.41 s, see Table 6.3.

The significant difference in the computational time showcases the feasibility of our algorithm compared to deterministic simulations. Our approach meets the practical requirements of the power plants, when subjected to the time constraints of secondary frequency control (5 min) imposed by the TSO. Furthermore, the algorithm considers all eventualities and formally verifies the safety of the boiler in real-time.

Table 6.3: Comparison between the computational time of reachability analysis using different models

Model	Time-horizon t_f	Reachable set $\boldsymbol{\mathcal{R}}([0, t_f])$	Remainders $\boldsymbol{\mathcal{L}}([0, t_f])$
2-nd order polynomial observer	5 min	41.41 s	34.06 s
3-rd order polynomial observer	5 min	206.37 s	198.53 s
Nonlinear model (6.11)-(6.18)	10 s	3420 s	3078 s

6.6 Summary

We present a new approach to rigorously verify the safety of the critical components in power plants when subjected by the TSO to high-load changes which, as a result, exploits the power plant's flexibility and load-following capabilities. Our analysis is based on a real boiler system located at a 450 MW combined cycle plant in Munich, Germany. In order to demonstrate the effectiveness of the proposed technique, we compute the reachable set for the evolution of the state variables of the drum with load changes equivalent to 40 MW (worst-case scenario).

An abstraction to a polynomial differential inclusion based on measurement data is proposed. It is shown that the Åström - Bell model [73], modelled via (6.11)-(6.18), can be approximated by a polynomial function without losing the fit to experimental data. The abstraction is systematically performed and returns the modelling errors, whereby all dynamic behaviours of the original system are captured by the abstraction. The proposed abstraction substantially reduces the computational time required to compute the reachable set in comparison with the original system.

According to our final results, it is computationally feasible to implement the proposed reachability algorithm while meeting the practical requirements of a real power plant. The reachability algorithm can be easily integrated into a distributed control system, in parallel to the existing control structure, and operates automatically without any interaction from the operator. Because reachability analysis establishes in advance whether or not a requested load dispatch by the TSO will trigger the water level safe limit considering all eventualities, the plant operator can potentially avoid an unnecessary shutdown of the facility.

Chapter 7

Concluding Remarks

In this chapter, we present a summary of this PhD thesis, highlighting its main contributions with regards to the application of reachability analysis to control and assess stability of power systems with formal guarantees. This chapter concludes with the author's final thoughts, observations, and future directions.

7.1 Summary and Contributions

Introduction. In this introductory chapter we motivated the need for the development of new tools to analyse and control utility grids in a formal fashion. With respect to the control of power systems, the industry standard controllers used in practice are often based on a linearized model of the process, thus limiting their effectiveness to a nearby region from the linearization point. On the other hand, the existing techniques to analyze stability of power systems, such as e.g. numerical simulations and Lyapunov direct method have served the industry well; however, it might be difficult to expect the same outcome in the foreseeable future given the current challenges associated with current power systems, e.g. the uncertainty on the generation side arising from the intermittent nature of renewable resources. This is evidenced by recent blackouts occurring in North America and Europe, in which the existing techniques did not identify the problem properly in a reasonable time. In this thesis, we proposed using reachability analysis as a means of controlling and assessing stability of power systems.

Reachability Analysis of Power Systems. In this chapter we introduced the reader unfamiliar with reachability analysis to: The basics of computing reachable sets, set representations and set-based operations, and an algorithmic procedure that computes reachable sets of differential algebraic equations (DAEs). Furthermore, we presented the standard power system models used to examine stability of utility grids; we primarily considered models capable of capturing the dynamical behavior of the electromechanical oscillations leading to instability of power systems. This chapter is based on the previous work published by Matthias Althoff in [12], in which an algorithmic procedure was developed based on well-know techniques for computing over-approximative reachable sets for the class of DAE systems. Using the standard power system models we applied the reachability algorithm recapitulated from [12] on two benchmark examples with a particular focus on studies involving transient stability of power systems. In the end we made several important observations with regards to the algorithm tunable parameters, in addition to the computational limitations imposed on the current algorithmic procedure.

Compositional Reachability Analysis of Power Systems. The main drawback with the algorithm presented in the previous chapter is that the computational efforts required to compute the reachable set are enormous. Basically, the objective of this chapter was to improve the algorithmic efficiency, thus scaling reachability analysis towards industrially relevant problem sizes. The main contribution of this chapter is a compositional procedure that can drastically reduce the computational effort required to assess the dynamical response of power systems during transients. The basic idea behind the proposed methodology was to abstract the complete transmission network into a set of subsystems, each consisting of a generating unit connected to its corresponding generator bus, whose algebraic constraints are unknown-but-bounded within some confidence intervals. This new abstraction makes it possible to obtain a set of subsystems that preserve the interaction and the correlation of multi-machine power systems, and most importantly, allows one to parallelize the computation of reachable sets for transient stability analysis. This drastically reduces the CPU time and renders reachability analysis feasible for practical applications. In order to demonstrate the efficiency and applicability of the proposed compositional algorithm, we illustrated the methodology on several benchmark examples often used in the power system community. The largest system we have considered is the IEEE 6-machine 30-bus which consists of more than 100 state variables. Furthermore, we compared the CPU time to alternative techniques which compute the reachable set without employing any compositional techniques.

Estimation of the Region of Attraction. Ever since the introduction of the Lyapunov direct method, it remained the exclusive tool in power systems to establish transient stability with formal guarantees. This method uses the so-called Lyapunov energy functions to find a region of attraction (ROA) within the state-space from which any initial state is guaranteed to be attracted by an equilibrium point. The main drawbacks of this method, however, are

- The approach results in a conservative stability region which can affect the system performance.
- Simplification of the power system dynamics is required in order to eliminate the set of algebraic equations inherently present in the system. These simplifications are generally unrealistic and do not hold for many practical situations.
- The method requires the existence of a feasible Lyapunov function, which is known to be extremely hard to find for the class of nonlinear system.
- The scalability of the Lyapunov direct method is questionable, as it often only handles a maximum of five state variables.

In this chapter we proposed an algorithmic procedure based on reachability analysis to estimate the ROA of an equilibrium point for nonlinear systems. Our proposed method overcame many limitations imposed on the applicability of Lyapunov-based approaches; this is due to the fact that our reachability algorithm does not require a Lyapunov function in order to provide an estimate of the ROA. Instead, our proposed method discretizes the state-space into smaller regions and examines whether each cell belongs to the ROA; that is, for each cell of the partitioned space, we check at each time instant whether its reachable set of differential state variables is confined within a target set. Basically, the target set is a small region surrounding the equilibrium which can establish stability of the cell under examination, according to the definition of asymptotic stability in the sense of Lyapunov. We showed that the proposed procedure estimates the exact ROA quite accurately, and more importantly, scales moderately with the system dimension. Furthermore we compared our results with two dominant techniques in this research area; namely, the optimization of the Lyapunov function sub-level set using sum-of-square decomposition and the computation of backward reachable sets using level set methods.

Formal LPV control of Power Systems. The first two contributions of this thesis were focused on the stability analysis of power systems and general nonlinear systems. This chapter, however, was primarily concerned with the control of multi-machine power systems in order to establish robust stability with formal guarantees. The existing controller used in practice is the so-called power system stabilizer (PSS) which is based on a linearized model of the generating unit. Clearly, the controller can deliver an optimal performance if the system trajectories are confined within a small neighbourhood around the linearization point. This is no longer the case in current power systems for several reasons, e.g. the integration of renewable resources and the introduction of competitive markets.

In this chapter we proposed a unified framework that considers the synthesis and the verification of a set of linear-parameter varying (LPV) controllers employing a state-feedback scheme in the closed-loop. LPV systems are generally useful for handling system nonlinearities, input uncertainties, and parameter variations of power systems. Furthermore, an exact reformulation of nonlinear systems into the standard LPV description makes it possible to apply powerful linear controller synthesis tools for nonlinear systems, e.g. robust $\mathcal{H}_\infty$ design and pole placement. Our proposed framework first transforms power systems described via the standard DAE formulation into a set of modular LPV systems. Afterwards, the set of time-varying parameters, which is required for the synthesis procedure, is identified using reachability analysis. Since the synthesis procedure only returns the vertices of the controller stabilizing the system, the framework generates a closed-form expression that describes the LPV controller in real-time control in terms of its vertices. The final step of the framework is to verify the resulting controller; that is, to provide the formal guarantee that the time-varying parameters will always remain within the specified parameter ranges under all eventualities. The proposed framework was demonstrated on two benchmark examples employing the synchronous generator and the doubly-fed induction generator as the generating units.

Formal Analysis of Power Plants. In the final chapter we illustrated the applicability of reachability analysis on a realistic configuration of a boiler system found in a 450 MW combined cycle power plant in Munich, Germany. In particular, we used reachability analysis to verify safety of the water level inside the steam-drum unit of the boiler system. Typically, the drum unit is known to degrade the load-following capabilities of conventional power plants, thus limiting their flexibility to meet the strict requirements imposed by the corresponding transmission system operator (TSO). In fact, the drum unit often triggers emergency shutdowns in thermal plants due to poor regulation of the water level inside the drum during fast-load changes; hence, the system is particularly suitable for employing reachability analysis as a means of verifying safety under various loading conditions. Our proposed reachability algorithm makes it possible to compute the bounds of all possible trajectories for a range of operating conditions while simultaneously meeting the practical requirements of a real power plant. In contrast to previous works in this area, we used for the first time an abstract model which considers the modelling errors to ensure that all dynamic behaviors of the real process are replicated by the abstraction. These modelling errors are obtained based on measurement data from the boiler system, with a very rich excitation covering the entirety of the operational range of the process. According to our final results, it is computationally feasible to implement the proposed reachability algorithm while meeting the practical requirements of a real power plant. The reachability algorithm can in principle be integrated into the existing distributed control system of the power plant, and more importantly, the algorithm can operate in real-time control without any interaction from the plant operator. Because reachability analysis establishes in advance whether or not a requested load dispatch by the TSO will trigger the water level safe limit considering all eventualities, the plant operator can potentially avoid an unnecessary shutdown of the facility.

7.2 Possible Future Directions

Future directions are separately discussed for each chapter

Compositional Reachability Analysis of Power Systems. The proposed compositional algorithm clearly outperformed existing techniques to compute reachable sets of nonlinear DAE systems and opened many promising directions to apply reachability analysis for larger power systems. However, two issues still remain an open question. The first question is how to guarantee convergence of the set of algebraic constraints at each generator bus in order to compute the reachable set? It is clear from the algorithmic procedure that this set is obtained in an iterative manner based on an initial guess; however, there are no mathematical guarantees that after n iterations the set of algebraic constraints shall converge to the real solution.

Clearly, reducing and even eliminating the resulting conservatism of the compositional algorithm is another interesting open question to be investigated. This conservatism results from the uncertainty of all possible values taken by the bus voltage and phase angle, even the unrealistic ones. One may argue that this is the tradeoff between accuracy and efficiency of the algorithm. It should be noted that the conservatism does not affect the security assessment during transient response; however, it can degrade the performance of the system, if the computed reachable set intersects with safety limits, e.g. bus voltage exceeding limits defined by the grid operators.

Estimating the Region of Attraction. As previously mentioned in this chapter, there are several promising directions for further research. One obvious direction would be to combine the existing techniques together, i.e. forward reachability, Lyapunov direct method and backward reachability. Clearly, each technique has its own share of advantages and disadvantages. For instance the Lyapunov approach is superior in terms of CPU time for low-dimensional systems but only provides a conservative ROA; on the other hand, our proposed algorithm consumes more computational power but provides accurate estimates of the ROA. Hence, in order to improve the overall computational time, one may obtain an initial estimate of the ROA using the Lyapunov-based approach, and use this region as the target set for the proposed forward reachability algorithm.

Another direction would be to apply this technique for larger power systems; one possibility would be to combine the proposed approach with the compositional reachability algorithm suggested earlier in this thesis. Finally, another possibility for future research would be related to the improvement of the algorithmic efficiency using the so-called polynomial zonotopes [7] as set representation for forward reachability computations. Polynomial zonotopes allow one to select larger initial sets, thus reducing the size of the partitioned grid, which in return would substantially reduce the overall computational costs associated with our estimation algorithm.

Formal LPV Control. This chapter has two possible future directions. A systematic modelling procedure to obtain LPV models of power systems is still missing. So far we only presented the proposed framework to synthesize and verify LPV controllers under the assumption that an LPV model is available, which is generally a non-trivial task. Another direction would be to modify the synthesis procedure such that the LPV controller can additionally control the frequency and the voltage at each bus connected to a generating unit.

Formal analysis of power plants. This chapter illustrated the feasibility of applying reachability analysis on realistic systems found in the power industry using abstract models. These models are constructed using measurement data from the real process. In principle, the proposed approach is extendable to variety of systems covering a wide range of applications in power systems. In our example, we particularly considered the safety verification of the water level inside the drum. Other applications relevant to the energy sector would be for example the verification of the grid codes supplied by the transmission system operators, for example verification of the low-voltage ride through of wind turbines. Clearly, meeting the specifications of these codes is of great importance to ensure reliability of the utility grid.

Bibliography

[1] Y. Abdel-Magid, M. Abido, S. Al-Baiyat, and A. Mantawy. Simultaneous stabilization of multimachine power systems via genetic algorithms. *IEEE transactions on Power Systems*, 14(4):1428–1439, 1999.

[2] M. Abido. Robust design of multimachine power system stabilizers using simulated annealing. *IEEE transactions on Energy conversion*, 15(3):297–304, 2000.

[3] M. Abido. Optimal design of power-system stabilizers using particle swarm optimization. *IEEE Transactions on Energy Conversion*, 17(3):406–413, 2002.

[4] J.-R. Abrial. Steam-boiler control specification problem. In *Formal Methods for Industrial Applications*, pages 500–509. Springer, 1996.

[5] A. T. Al-Awami, Y. Abdel-Magid, and M. Abido. A particle-swarm-based approach of power system stability enhancement with unified power flow controller. *International Journal of Electrical Power and Energy Systems*, 29(3):251–259, 2007.

[6] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. PhD thesis, Technische Universität München, 2010.

[7] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Hybrid Systems: Computation and Control*, pages 173–182, 2013.

[8] M. Althoff. Formal and compositional analysis of power systems using reachable sets. *IEEE Transactions on Power Systems*, 29(5):2270–2280, 2014.

[9] M. Althoff. An Introduction to CORA 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[10] M. Althoff, M. Cvetković, and M. Ilić. Transient stability analysis by reachable set computation. In *Proc. of the IEEE PES Innovative Smart Grid Technologies Europe*, pages 1–8, 2012.

[11] M. Althoff and B. H. Krogh. Avoiding geometric intersection operations in reachability analysis of hybrid systems. In *Hybrid Systems: Computation and Control*, pages 45–54, 2012.

[12] M. Althoff and B. H. Krogh. Reachability analysis of nonlinear differential-algebraic systems. *IEEE Transactions on Automatic Control*, 59(2):371–383, 2014.

[13] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proc. of the 47th IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[14] P. M. Anderson and A. A. Fouad. *Power System Control and Stability.* Wiley-IEEE Press, 2002.

[15] G. Andersson, P. Donalek, R. Farmer, N. Hatziargyriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, et al. Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance. *IEEE transactions on Power Systems*, 20(4):1922–1928, 2005.

[16] M. Anghel, F. Milano, and A. Papachristodoulou. Algorithmic construction of Lyapunov functions for power system stability analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(9):2533–2546, 2013.

[17] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: A design example. *Automatica*, 31(9):1251–1261, 1995.

[18] U. M. Ascher and L. R. Petzold. *Computer methods for ordinary-differential equations and differential-algebraic equations.* Siam, 1998.

[19] B. Avramovic, P. V. Kokotovic, J. R. Winkelman, and J. H. Chow. Area decomposition for electromechanical models of power systems. *Automatica*, 16(6):637–648, 1980.

[20] H. Banakar, C. Luo, and B. T. Ooi. Impacts of wind power minute-to-minute variations on power system operation. *IEEE Transactions on Power Systems*, 23(1):150–160, 2008.

[21] M. Bardi. Some applications of viscosity solutions to optimal control and differential games. *Viscosity Solutions and Applications*, pages 44–97, 1997.

[22] M. Berz and G. Hoffstätter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing*, 4(1):83–97, 1998.

[23] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory.* SIAM, 1994.

[24] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. SIAM, 1995.

[25] N. G. Bretas and L. F. Alberto. Lyapunov function for power systems with transfer conductances: Extension of the invariance principle. *IEEE Transactions on Power Systems*, 18(2):769–777, 2003.

[26] J. Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on circuits and systems*, 25(9):772–781, 1978.

[27] E. Camm, M. Behnke, O. Bolado, M. Bollen, M. Bradt, C. Brooks, W. Dilling, M. Edds, W. Hejdak, D. Houseman, et al. Reactive power compensation for wind power plants. In *Proc. of the 2009 IEEE Power and Energy Society General Meeting*, pages 1–7.

[28] H.-D. Chang, C.-C. Chu, and G. Cauley. Direct stability analysis of electric power systems using energy functions: Theory, applications, and perspective. *Proc. of the IEEE*, 83(11):1497–1529, 1995.

[29] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Proc. of the International Conference on Computer Aided Verification*, pages 258–263, 2013.

[30] Y. C. Chen and A. D. Domínguez-García. A method to study the effect of renewable resource variability on power system dynamics. *IEEE Transactions on Power Systems*, 27(4):1978–1989, 2012.

[31] G. Chesi. SMRSOFT: A MATLAB toolbox for optimization over polynomials and dynamical systems study via SOS programming. `http://www.eee.hku.hk/~chesi/smrsoft.htm`.

[32] G. Chesi. *Domain of attraction: Analysis and control via SOS programming*. Springer Science and Business Media, 2011.

[33] H.-D. Chiang and J. S. Thorp. The closest unstable equilibrium point method for power system dynamic security assessment. *IEEE Transactions on Circuits and Systems*, 36(9):1187–1200, 1989.

[34] M. Chilali and P. Gahinet. $\mathcal{H}_\infty$ design with pole placement constraints: An LMI approach. *IEEE Transactions on Automatic Control*, 41(3):358–367, 1996.

[35] E. M. Clarke, O. Grumberg, and D. E. Long. Model checking and abstraction. *ACM transactions on Programming Languages and Systems*, 16(5):1512–1542, 1994.

[36] E. A. Cross and I. M. Mitchell. Level set methods for computing reachable sets of systems with differential algebraic equation dynamics. In *Proc. of the IEEE American Control Conference*, pages 2260–2265, 2008.

[37] T. Dang, A. Donzé, and O. Maler. Verification of analog and mixed-signal circuits using hybrid system techniques. In *Proc. of the International Conference on Formal Methods in Computer-Aided Design*, pages 21–36. Springer, 2004.

[38] T. X. T. Dang. *Vérification et synthese des systemes hybrides*. PhD thesis, Institut polytechnique de Grenoble, 2000.

[39] B. Das. Radial distribution system power flow using interval arithmetic. *International Journal of Electrical Power and Energy Systems*, 24(10):827–836, 2002.

[40] A. L. Do Bomfim, G. N. Taranto, and D. M. Falcao. Simultaneous tuning of power system damping controllers using genetic algorithms. *IEEE Transactions on Power Systems*, 15(1):163–169, 2000.

[41] A. El-Guindy. Drum-boiler control performance optimization using an observer-based state-feedback controller within MATLAB/Simulink environment. Master's thesis, Universität Bremen, 2013. `http://elib.suub.uni-bremen.de/edocs/00104228-1.pdf`.

[42] A. El-Guindy, Y. C. Chen, and M. Althoff. Compositional transient stability analysis of power systems via the computation of reachable sets. In *Proc. of the IEEE Amercian Control Conference*, pages 2536–2543, 2017.

[43] A. El-Guindy, D. Han, and M. Althoff. Formal analysis of drum-boiler units to maximize the load-following capabilities of power plants. *IEEE Transactions on Power Systems*, 31(6):4691–4702, 2016.

[44] A. El-Guindy, D. Han, and M. Althoff. Estimating the region of attraction via forward reachable sets. In *Proc. of the IEEE Amercian Control Conference*, pages 1263–1270, 2017.

[45] A. El-Guindy, F. Nickel, and K. Michels. Centralized multivariable feedback control of steam drums in combined cycle power plants. *VGB PowerTech*, 95(4):73–78, 2015.

[46] A. El-Guindy, S. Runzi, and K. Michels. Optimizing drum-boiler water level control performance: A practical approach. In *Proc. of the IEEE Conference on Control Applications*, pages 1675–1680, 2014.

[47] A. El-Guindy, K. Schaab, B. Schürmann, O. Stursberg, and M. Althoff. Formal LPV control for transient stability of power systems. In *Proc. of the IEEE Power and Energy Society General Meeting*, pages 1–5, 2017.

[48] A. El-Zonkoly, A. Khalil, and N. Ahmied. Optimal tunning of lead-lag and fuzzy logic power system stabilizers using particle swarm optimization. *Expert Systems with Applications*, 36(2):2097–2106, 2009.

[49] P. M. Esfahani, M. Vrakopoulou, K. Margellos, J. Lygeros, and G. Andersson. A robust policy for automatic generation control cyber attack in two area power network. In *Proc. of the 49th IEEE Conference on Decision and Control*, pages 5973–5978, 2010.

[50] L. Fan. Review of robust feedback control applications in power systems. In *Proc. of the IEEE Power Systems Conference and Exposition*, pages 1–7, 2009.

[51] L. M. Fernández, F. Jurado, and J. R. Saenz. Aggregated dynamic model for wind farms with doubly fed induction generator wind turbines. *Renewable energy*, 33(1):129–140, 2008.

[52] M. Flynn and M. O. Malley. A drum boiler model for long term power system dynamic simulation. *IEEE Transaction Power System*, 14(1):209–217, 1999.

[53] J. Fraile-Ardanuy and P. J. Zufiria. Design and comparison of adaptive power system stabilizers based on neural fuzzy networks and genetic algorithms. *Neurocomputing*, 70(16):2902–2912, 2007.

[54] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *Computer Aided Verification*, pages 379–395. Springer, 2011.

[55] R. Genesio, M. Tartaglia, and A. Vicino. On the estimation of asymptotic stability regions: State of the art and new proposals. *IEEE Transactions on Automatic Control*, 30(8):747–755, 1985.

[56] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, pages 291–305. Springer, 2005.

[57] E. Glassman, A. L. Desbiens, M. Tobenkin, M. Cutkosky, and R. Tedrake. Region of attraction estimation for a perching aircraft: A Lyapunov method exploiting barrier certificates. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2235–2242, 2012.

[58] D. Han, A. El-Guindy, and M. Althoff. Estimating the domain of attraction based on the invariance principle. In *Proc. of the 55th IEEE Conference on Decision and Control*, pages 5569–5576, 2016.

[59] D. Han, A. El-Guindy, and M. Althoff. Power systems transient stability analysis via optimal rational Lyapunov functions. In *Proc. of the IEEE Power and Energy Society General Meeting*, pages 1–5, 2016.

[60] R. He, K.-Z. Liu, and S. Mei. LPV modelling and gain-scheduled control approach for the transient stabilization of power systems. *IEEJ Transactions on Electrical and Electronic Engineering*, 5(1):87–95, 2010.

[61] R. He, K.-Z. Liu, S. Mei, and X. Gui. A gain-scheduled state feedback control method for the transient stability control of a single-machine infinite-bus power system. In *Proc. of the IEEE Chinese Control Conference*, pages 2147–2152, 2006.

[62] D. Henrion and M. Korda. Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control*, 59(2):297–312, 2014.

[63] T. A. Henzinger and H. Wong-Toi. *Using HyTech to synthesize control parameters for a steam boiler*. Springer, 1996.

[64] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, 2013. `http://control.ee.ethz.ch/~mpt`.

[65] L. Hirth and I. Ziegenhagen. Control power and variable renewables: A glimpse at German data. In *Proc. of the 10th International Conference on European Energy Market*, 2013.

[66] E. M. Hope, X. Jiang, and A. D. Domínguez-García. A reachability-based method for large-signal behavior verification of DC-DC converters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(12):2944–2955, 2011.

[67] S. Hossain, S. Dhople, and T. T. Johnson. Reachability analysis of closed-loop switching power converters. In *Proc. of the IEEE Power and Energy Conference at Illinois*, pages 130–134, 2013.

[68] L. Jaulin, M. Kieffer, and O. Didrit. *Applied Interval Analysis*. Springer, 2006.

[69] X. Jiang, Y. C. Chen, and A. D. Dominguez-Garcia. A set-theoretic framework to assess the impact of variable generation on the power flow. *IEEE Transactions on Power Systems*, 28(2):855–867, 2013.

[70] L. Jin, R. Kumar, and N. Elia. Reachability analysis based transient stability design in power systems. *International Journal of Electrical Power and Energy Systems*, 32(7):782–787, 2010.

[71] L. Jin, H. Liu, R. Kumar, J. D. Mc Calley, N. Elia, and V. Ajjarapu. Power system transient stability design using reachability based stability-region computation. In *Proc. of the 37th Annual North American Power Symposium*, pages 338–343, 2005.

[72] T. T. Johnson, Z. Hong, and A. Kapoor. Design verification methods for switching power converters. In *Proc. of the Power and Energy Conference at Illinois*, pages 1–6, 2012.

[73] K. J. Åström and R. D. Bell. Drum boiler dynamics. *Automatica*, 36:363–378, 2000.

[74] M. Kamgarpour, C. Beyss, and A. Fuchs. Reachability-based control synthesis for power system stability. *IFAC-PapersOnLine*, 49(27):238–243, 2016.

[75] J. Kautsky, N. K. Nichols, and P. Van Dooren. Robust pole assignment in linear state feedback. *International Journal of Control*, 41(5):1129–1155, 1985.

[76] H. K. Khalil and J. Grizzle. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.

[77] H. Kim and S. Choi. A model on water level dynamics in natural circulation drum-type boilers. *International Communications in Heat and Mass Transfer*, 32:786 – 796, 2005.

[78] T. Kim, D. Lee, and S. Ro. Analysis of thermal stress evolution in the steam drum during start-up of a heat recovery steam generator. *Applied Thermal Engineering*, 20(11):977 – 992, 2000.

[79] D. S. Kirschen. Demand-side view of electricity markets. *IEEE Transactions on Power Systems*, 18(2):520–527, 2003.

[80] M. V. Kothare, B. Mettler, M. Morari, P. Bendotti, and C.-M. Falinower. Level control in the steam generator of a nuclear power plant. *IEEE Transactions on Control Systems Technology*, 8(1):55–69, 2000.

[81] P. Kundur, N. J. Balu, and M. G. Lauby. *Power system stability and control*, volume 7. McGraw-hill New York, 1994.

[82] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, and C. Taylor. Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions. *IEEE Transactions on Power Systems*, 19(3):1387–1401, 2004.

[83] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis: Internal approximation. *Systems and control letters*, 41(3):201–211, 2000.

[84] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *Journal of Symbolic Computation*, 32(3):231 – 253, 2001.

[85] C. Le Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.

[86] T. Lee, M. Leoky, and N. H. McClamroch. Geometric tracking control of a quadrotor UAV on SE (3). In *Proc. of the 49th IEEE Conference on Decision and Control*, pages 5420–5425, 2010.

[87] B. C. Lesieutre, S. Roy, V. Donde, and A. Pinar. Power system extreme event screening using graph partitioning. In *Proc. of the 38th North American Power Symposium*, pages 503–510, 2006.

[88] C.-W. Liu and J. S. Thorp. A novel method to compute the closest unstable equilibrium point for transient stability region estimate in power systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(7):630–635, 1997.

[89] Q. Liu, V. Vittal, and N. Elia. Expansion of system operating range by an interpolated lpv facts controller using multiple lyapunov functions. *IEEE transactions on power systems*, 21(3):1311–1320, 2006.

[90] Q. Liu, V. Vittal, and N. Elia. LPV supplementary damping controller design for a thyristor controlled series capacitor (TCSC) device. *IEEE Transactions on Power Systems*, 21(3):1242–1249, 2006.

[91] J. Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proc. of the 13th IEEE International Symposium on Computer Aided Control System Design*, pages 284–289, 2004.

[92] D. G. Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8:74–80, 1964.

[93] J. Lygeros, C. Tomlin, and S. Sastry. Multiobjective hybrid controller synthesis. In *Hybrid and Real-Time Systems*, pages 109–123. Springer, 1997.

[94] A. Majumdar, A. A. Ahmadi, and R. Tedrake. Control and verification of high-dimensional systems with DSOS and SDSOS programming. In *Proc. of the 53rd IEEE Conference on Decision and Control*, pages 394–401, 2014.

[95] J. L. Meza, V. Santibáñez, and R. Campa. An estimate of the domain of attraction for the PID regulator of manipulators. *International Journal of Robotics and Automation*, 22(3):187–195, 2007.

[96] F. Milano. *Power system modelling and scripting*. Springer Science and Business Media, 2010.

[97] I. Mitchell. A toolbox of level set methods. `http://www.cs.ubc.ca/~mitchell/ToolboxLS/`.

[98] I. M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In *Hybrid systems: computation and control*, pages 428–443. Springer, 2007.

[99] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[100] P. Mohajerin Esfahani, M. Vrakopoulou, K. Margellos, J. Lygeros, and G. Andersson. Cyber attack in a two-area power system: Impact identification using reachability. In *Proc. of the IEEE American Control Conference*, pages 962–967, 2010.

[101] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49, 2003.

[102] R. E. Moore. *Methods and applications of interval analysis.* SIAM, 1979.

[103] M. J. Moran, H. N. Shapiro, D. D. Boettner, and M. B. Bailey. *Fundamentals of engineering thermodynamics.* John Wiley & Sons, 2010.

[104] S. Muller, M. Deicke, and R. W. De Doncker. Doubly fed induction generator systems for wind turbines. *IEEE Industry applications magazine*, 8(3):26–33, 2002.

[105] M. G. Na. Auto-tuned PID controller using a model predictive control method for the steam generator water level. *IEEE Transactions on Nuclear Science*, 48(5):1664–1671, 2001.

[106] M. Nakamoto, K. Shimizu, and H. Fukuda. Multivariable control for a combined cycle power plant. *Control Engineering Practice*, 3(4):465–470, 1995.

[107] W. L. Oberkampf, S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin. Error and uncertainty in modeling and simulation. *Reliability Engineering and System Safety*, 75(3):333–357, 2002.

[108] R. Pena, J. Clare, and G. Asher. Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation. *IEE Proc. of Electric Power Applications*, 143(3):231–241, 1996.

[109] H. N. V. Pico, D. C. Aliprantis, and E. C. Hoff. Reachability analysis of power system frequency dynamics with new high-capacity hvac and hvdc transmission lines. In *Proc. of the IREP Bulk Power System Dynamics and Control Symposium*, pages 1–9, 2013.

[110] V. Pico, N. Hugo, and D. C. Aliprantis. Voltage ride-through capability verification of wind turbines with fully-rated converters using reachability analysis. *IEEE Transactions on Energy Conversion*, 29(2):392–405, 2014.

[111] A. Platzer and E. M. Clarke. The image computation problem in hybrid systems model checking. In *Hybrid Systems: Computation and Control*, pages 473–486. Springer, 2007.

[112] F. Prabhakara, A. El-Abiad, and A. Koivo. Application of generalized zubov's method to power system stability. *International Journal of Control*, 20(2):203–212, 1974.

[113] S. Prajna, A. Papachristodoulou, and F. Wu. Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach. In *Proc. of the 5th Asian Control Conference*, pages 157–165, 2004.

[114] W. Qiu, V. Vittal, and M. Khammash. Decentralized power system stabilizer design using linear parameter varying approach. *IEEE Transactions on Power Systems*, 19(4):1951–1960, 2004.

[115] R. Rajamani. Observers for Lipschitz nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):397–401, 1998.

[116] S. Ratschan and Z. She. Providing a basin of attraction to a target region of polynomial systems by computation of Lyapunov-like functions. *SIAM Journal on Control and Optimization*, 48(7):4377–4394, 2010.

[117] Y. G. Rebours, D. S. Kirschen, M. Trotignon, and S. Rossignol. A survey of frequency and voltage control ancillary services - Part I: Technical features. *IEEE Transactions on Power Systems*, 22(1):350–357, 2007.

[118] M. Ribbens-Pavella and F. Evans. Direct methods for studying dynamics of large-scale electric power systems: A survey. *Automatica*, 21(1):1–21, 1985.

[119] S. Rump. INTLAB - INTerval LABoratory. In *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, 1999.

[120] Samson AG. *Application Notes for Valve Sizing*, 2012.

[121] K. Schaab, J. Hahn, M. Wolkov, and O. Stursberg. Robust control for voltage and transient stability of power grids relying on wind power. *Control Engineering Practice*, 60:7–17, 2017.

[122] K. Schaab and O. Stursberg. Robust decentralized LPV control for transient stability of power systems. *IFAC-PapersOnLine*, 48(30):566–571, 2015.

[123] B. Schürmann, A. El-Guindy, and M. Althoff. Closed-form expressions of convex combinations for controller design. In *Proc. of the IEEE American Control Conference*, pages 2795 – 2801, 2016.

[124] J. K. Scott. *Reachability analysis and deterministic global optimization of differential-algebraic systems*. PhD thesis, Massachusetts Institute of Technology, 2012.

[125] J. C. Smith, M. R. Milligan, E. A. DeMeo, and B. Parsons. Utility wind integration and operating impact state of the art. *IEEE Transactions on Power Systems*, 22(3):900–908, 2007.

[126] C. P. Steinmetz. Power control and stability of electric generating stations. *Transactions of the American Institute of Electrical Engineers*, 39(2):1215–1287, 1920.

[127] Y. Susuki, T. J. Koo, H. Ebina, T. Yamazaki, T. Ochi, T. Uemura, and T. Hikihara. A hybrid system approach to the analysis and design of power grid dynamic performance. *Proc. of the IEEE*, 100(1):225–239, 2012.

[128] Y. Susuki, T. Sakiyama, T. Ochi, T. Uemura, and T. Hikihara. Verifying fault release control of power system via hybrid system reachability. In *Proc. of the 40th North American Power Symposium*, pages 1–6, 2008.

[129] W. Tan, A. Packard, et al. Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. *IEEE Transactions on Automatic Control*, 53(2):565–570, 2008.

[130] F. E. Thau. Observing the state of non-linear dynamic systems. *International Journal of Control*, 17(3):471–479, 1973.

[131] A. Vannelli and M. Vidyasagar. Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, 21(1):69–80, 1985.

[132] J. Vitt and J. Hooman. Assertional specification and verification using PVS of the steam boiler control system. In *Formal Methods for Industrial Applications*, pages 453–472. Springer, 1996.

[133] M. Vrakopoulou, P. M. Esfahani, K. Margellos, J. Lygeros, and G. Andersson. Cyber-attacks in the automatic generation control. In *Cyber Physical Systems Approach to Smart Electric Power Grid*, pages 303–328. Springer, 2015.

[134] T. L. Vu and K. Turitsyn. Lyapunov functions family approach to transient stability assessment. *IEEE Transactions on Power Systems*, 31(2):1269–1277, 2016.

[135] T.-C. Wang, S. Lall, and M. West. Polynomial level-set methods for nonlinear dynamical systems analysis. In *Proc. of Allerton conference on communication, control and computing*, pages 640–649, 2005.

[136] Z. Wang and F. L. Alvarado. Interval arithmetic in power flow analysis. *IEEE Transactions on Power Systems*, 7(3):1341–1349, 1992.

[137] H. Wong-Toi. The synthesis of controllers for linear hybrid automata. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 5, pages 4607–4612, 1997.

[138] R. You, H. J. Eghbali, and M. H. Nehrir. An online adaptive neuro-fuzzy power system stabilizer for multimachine systems. *IEEE Transactions on Power systems*, 18(1):128–135, 2003.

[139] S. Yusof, G. Rogers, and R. Alden. Slow coherency based network partitioning including load buses. *IEEE Transactions on Power Systems*, 8(3):1375–1382, 1993.

[140] F. Zhao, J. Ou, and W. Du. Simulation modeling of nuclear steam generator water level process: A case study. *ISA transactions*, 39(2):143–151, 2000.

[141] C. Zhu, M. Khammash, V. Vittal, and W. Qiu. Robust power system stabilizer design using $\mathcal{H}_\infty$ loop shaping approach. *IEEE Transactions on Power Systems*, 18(2):810–818, 2003.

[142] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, 26(1):12–19, 2011.