THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Data-driven methods for real-time dynamic stability assessment and control

HANNES HAGMAR

CHALMERS
UNIVERSITY OF TECHNOLOGY

Division of Electric Power Engineering
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2022

Data-driven methods for real-time dynamic stability assessment and control
HANNES HAGMAR

ISBN 978-91-7905-707-7
Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5173
ISSN 0346-718X

Division of Electric Power Engineering
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Data-driven methods for real-time dynamic stability assessment and control

HANNES HAGMAR
Division of Electric Power Engineering
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Electric power systems are becoming increasingly complex to operate; a trend driven by an increased demand for electricity, large-scale integration of renewable energy resources, and new system components with power electronic interfaces. In this thesis, a new real-time monitoring and control tool that can support system operators to allow more efficient utilization of the transmission grid has been developed. The developed tool is comprised of four methods aimed to handle the following complementary tasks in power system operation: 1) preventive monitoring, 2) preventive control, 3) emergency monitoring, and 4) emergency control. The methods are based on recent advances in machine learning and deep reinforcement learning to allow real-time assessment and optimized control, while taking into account the dynamic stability of a power system.

The developed method for preventive monitoring is proposed to be used to ensure a secure operation by providing real-time estimates of a power system's dynamic security margins. The method is based on a two-step approach, where neural networks are first used to estimate the security margin, which then is followed by a validation of the estimates using a search algorithm and actual time-domain simulations. The two-step approach is proposed to mitigate any inconsistency issues associated with neural networks under new or unseen operating conditions. The method is shown to reduce the total computation time of the security margin by approximately 70 % for the given test system. Whenever the security margins are below a certain threshold, another developed method, aimed at preventive control, is used to determine the optimal control actions that can restore the security margins to a level above a predefined threshold. This method is based on deep reinforcement learning and uses a hybrid control scheme that is capable of simultaneously adjusting both discrete and continuous action variables. The results show that the developed method quickly learns an effective control policy to ensure a sufficient security margin for a range of different system scenarios.

In case of severe disturbances and when the preventive methods have not been sufficient to guarantee a stable operation, system operators are required to rely on emergency monitoring and control methods. In the thesis, a method for emergency monitoring is developed that can quickly detect the onset of instability and predict whether the present system state is stable or if it will evolve into an alert or an emergency state in the near future. As time progresses and if new events occur in the system, the network can update the assessment continuously. The results from case studies show good performance and the network can accurately, within only a few seconds after a disturbance, predict voltage instability in almost all test

cases. Finally, a method for emergency control is developed, which is based on deep reinforcement learning and is aimed to mitigate long-term voltage instability in real-time. Once trained, the method can continuously assess the system stability and suggest fast and efficient control actions to system operators in case of voltage instability. The control is trained to use load curtailment supplied from demand response and energy storage systems as an efficient and flexible alternative to stabilize the system. The results show that the developed method learns an effective control policy that can stabilize the system quickly while also minimizing the amount of required load curtailment.

Keywords: *deep reinforcement learning*, *dynamic voltage security margin*, *machine learning*, *neural networks*, *optimal control*, *security assessment*, *voltage instability prediction*, *voltage stability.*

Acknowledgements

Finally, my time as a PhD student has come to an end and I have truly enjoyed it. To be able to pursue new research ideas, travel to interesting places, and have flexibility in your work; what is not to like?! So, naturally, I have a lot of people I would like to thank for this opportunity and for the help and support they have given me during this period.

I would like to start by thanking my supervisor Anh Tuan Le and my examiner Ola Carlson for very good discussions and guidance all these years. I also appreciate that you have always encouraged me to explore my own research angles and ideas in this project. A special thanks go to my co-supervisor Robert Eriksson at Svenska kraftnät who has been very supportive throughout the research work, both by providing practical knowledge of how the actual power systems work, and whose experience has been most helpful. I would also like to thank all the members of the reference group that have guided me and the project during these years. The financial support provided by Energimyndigheten, Svenska kraftnät, and Energiforsk is also gratefully acknowledged. I would like to give a shout-out to all my former colleagues at RISE and also a big thanks also to all my colleagues at the Division of Electric Power Engineering. A special recognition goes to all my previous office mates: Ankur, Ali, Ioannis, Kelly, and Rahmat. It has been a privilege to get to know you and I will miss our good times.

While pursuing a PhD during the Covid-19 pandemic could have been a quite lonesome activity for some, I had the privilege of working from home with the best person I know; my partner and soulmate Lina. Thanks for being the best colleague I could possibly have these last years. And finally, I would also like to thank my whole family, you are the best. Love you all.

Hannes Hagmar

List of Acronyms

AC	alternating current
ATC	available transfer capacity
CIG	converter-interfaced generation
CLOD	complex load
CPF	continuation power flow
DR	demand response
DRL	deep reinforcement learning
DSA	dynamic security assessment
DT	decision tree
DVSM	dynamic voltage security margin
ESS	energy storage systems
FACTS	flexible alternating current transmission systems
HVDC	high voltage direct current
IM	induction motor
LIVES	local identification of voltage emergency situations
LSTM	long short-term memory
LTC	load tap changer
LTVS	long-term voltage stability
ML	machine learning
MDP	Markov decision process
MPC	model predictive control
MSE	mean squared error
NN	neural network
NTC	net transfer capacity
OC	operating condition
OEL	over-excitation limiter

PCLL	post-contingency loadability limit
PMU	phasor measurement unit
PPO	proximal policy optimization
QSS	quasi-steady state
RDS-ACT	real-time dynamic stability assessment and control tool
ReLU	rectified linear unit function
RL	reinforcement learning
RNN	recurrent neural network
SCADA	supervisory control and data acquisition
SOL	secure operating limit
SPS	system protection scheme
Tanh	hyperbolic tangent function
TE	Thévenin equivalent
TDT	temporal decision tree
TRM	transmission reliability margin
TTC	total transfer capacity
VID	voltage instability detection
VIP	voltage instability prediction
VSM	voltage security margin

Contents

Chapter 1

Introduction

This chapter presents the background and a problem overview, followed by a summary of previous work and identified research questions. The aim and the contributions are presented together with a list of the publications on which the thesis is based upon.

1.1 Background

The development towards a more sustainable energy system has resulted in a rapidly growing share of non-controllable renewable generation. Wind and solar power are among the fastest-growing sources of electric energy globally; a trend that is likely to continue given that these energy sources have become the lowest-cost source of new power generation in most parts of the world [1, 2]. Sweden has seen a similar development, where the installed wind power capacity increased from 241 MW to 9 976 MW between the years 2000 and 2020 [3]. Simultaneously, electric power generated from conventional fossil-fueled power plants, primarily coal and gas, needs to be decommissioned at a rapid rate to reduce carbon emissions to the atmosphere. In Sweden, fossil fuels have not been used for electric power generation to any significant extent, and nuclear power in combination with hydropower has instead been used for a large share of the base power. However, due to the lack of economic profitability, political decisions, and reactors approaching their technical lifetime, nuclear power is now gradually being decommissioned. By the end of 2019, one reactor of the Swedish nuclear power station Ringhals was shut down, followed by the decommissioning of a second reactor at the end of 2020 [4]. Out of originally twelve reactors in Sweden, only six remain currently in operation, with a total rated capacity of about 6 900 MW. While there is a political debate about whether Sweden should continue to rely on nuclear power and even invest in new reactors, its future is still uncertain and the development of new nuclear reactors would lie many years into the future.

Although a larger share of renewable energy sources in the power system is desirable from a sustainability perspective, the intermittent nature of these may cause significant planning and operational challenges. In particular, maintaining what

is known as the power balance, which requires that the generated power needs to match the load demand at every time instant, will become increasingly challenging in a future with a larger share of renewable generation [4]. While technologies such as energy storage systems or load flexibility could alleviate the problems caused by an increased generation from intermittent energy sources, these technologies are still expensive and/or are not yet well-developed. In Sweden, the power balance is instead generally maintained by regulating the output of hydropower stations. In cases of negative power balance and when the balancing capacity of the hydropower is not sufficient, electric power is typically imported from neighboring countries. Sweden's grid is a part of the synchronous inter-Nordic system, which also includes the transmission grids of Norway, Finland, and eastern Denmark. In addition, a growing number of HVDC connections link the Nordic system to the European continent and the United Kingdom. Market integration and electricity connections linking Sweden to neighboring countries also allow ancillary services to be supported when required. In Europe, a recent example is the development of the joint automatic frequency restoration reserve, known as the European Platform for the International Coordination of the Automatic frequency restoration process and Stable System Operation [5].

To ensure that the balancing capacity of hydropower stations can be utilized to its full extent, or that sufficient power can be imported from neighboring countries, the importance of *transmission capacity* in the power system is expected to increase in the future [6]. An increased transmission capacity allows energy to be generated where it is most economically efficient to do so and it sets the limit on how much power can be transmitted through the transmission lines. Most commonly, it is not the physical capacity of the transmission lines that limits the transmission capacity in a system, but rather power system stability-related phenomena [7]. Electric power systems are most commonly required to be operated according to the N-1 contingency criterion, meaning that the system should be able to withstand the loss of any single system component, such as generation or transmission capacity, and still remain stable. A system that satisfies this criterion is said to be (N-1) *secure*, and a power system's transmission capacity is typically computed to always ensure that this criterion is fulfilled [4]. This thesis deals mainly with the stability phenomenon called voltage stability, which refers to the capability of a system to maintain system voltages following a disturbance. Voltage stability is especially an issue in power systems where generation and loads are geographically separated over large distances; Sweden being a typical example. A voltage collapse is referring to a sequence of events accompanying voltage instability that leads to abnormally low voltages or a blackout in a significant part of a power system [7]. Although voltage collapses occur relatively seldom, they are related to extremely high costs to society and system operators need to continuously operate their systems to minimize the risk of such events.

1.2 Problem overview and research questions

Most power systems of today are operated as deregulated energy markets, where market participants consisting of both buyers (mainly electricity retailers and larger industries) and sellers (generating companies) place bids on an open energy market [8]. The role of the system operator is to both maintain the security and stability of the system and to operate the system efficiently to maximize the use for the other market participants [9]. While an operation close to the security limits enables a cost-effective utilization of the system, it may also make it more vulnerable to disturbances. Consequently, there exists a balance between a power system that is operated efficiently and one that is operated securely.

The state and stability of a power system are generally monitored using supervisory control and data acquisition (SCADA) systems. SCADA systems use loosely synchronized scalar measurements with a refresh rate of around two to four seconds taken at remote terminal units which are then processed by a state estimator to estimate the most probable state of the power system [10]. In recent years, the deployment of phasor measurement units (PMUs) has increased significantly. PMUs are highly accurate measuring devices, allowing time-synchronized real-time phasor measurements of electrical quantities in the power system at a much higher refresh rate than that of SCADA. If the system is fully observable by time-synchronized phasor measurements, the measurements can be filtered through a linear state estimator, allowing significantly more frequent and faster estimates of the system state [11]. With sufficient deployment, PMUs could help in the decision-making process in real-time power system operation and control by visualizing the system condition and stability margins in real-time.

To always ensure that a power system is operated securely, system operators continuously monitor the system state and evaluate the system security. However, even when the system is secure for a given operation point, system operators are also required to know how *far* the system can move from its current operating point and still remain secure. Therefore, system operators continuously compute security margins, which in turn represent the available transmission capacity in the system. One commonly used measure of a power system's security margin is based on the post-contingency loadability limit (PCLL) [12]. The PCLL is evaluated by estimating the loadability limit of a post-contingency operating point, where a solution path is traced by iteratively increasing the system stress until the system's critical point is reached [13, 14]. The characteristics of the iteratively increased system stress in the post-contingency setting are similar to that of the slow load restoration that typically follows in a long-term voltage stability (LTVS) event. Whenever the security margins are deemed insufficient, system operators take preventive actions to ensure that system security can be restored.

In the case of emergency events, such as the occurrence of simultaneous multiple disturbances, system operators have to rely on the robustness of the system and system protection schemes (SPSs) to control the system back to a stable state [12]. If a disturbance is causing fast-developing system instability, the time to manually assess and choose suitable control actions will be too short, and system operators are

typically required to rely on automated local controls instead. For slower-developing system instability, such as long-term voltage instability, system operators may have a chance to assess the situation and choose a suitable set of control actions to steer the system back into stable operation again. Time is an important issue in stability assessment and the earlier monitoring systems can detect the onset of instability, the faster system operators can issue control actions to steer the system back to a stable operation again. A fast response reduces the risk of a system collapse and can in many cases also reduce the cost of stabilizing the system. Thus, in the case of a possible emergency event, the ability of system operators to quickly detect instability and act quickly with the correct control measures is imperative. For voltage stability, the perhaps simplest method available to system operators is to monitor system voltages and use the voltage magnitudes as an indicator for system instability. However, due to the reactive power support of generators and other voltage control devices in the system, the system voltage magnitudes can remain at near-nominal levels for a relatively long time following a disturbance. Fast system degradation and drops in system voltages may then be triggered by, for instance, the activation of field current limiters of synchronous generators. The remaining time it takes before a potential voltage collapse is developed may then be too short for system operators to initiate the required control actions.

The complexity of the power system requires system operators to often rely on simplified static models of how the system behaves during various operating conditions and with respect to disturbances, as more advanced dynamic models are too time-consuming to use in real-time applications. However, electric power systems are becoming increasingly complex, with numerous dynamic components such as non-linear loads, converter-based generators, and other power electronic devices found in, for instance, high voltage direct current (HVDC) and flexible alternating current transmission systems (FACTS) [15, 16]. These components have in common that their dynamic response following a disturbance in the system is significantly faster when compared to more conventional equipment. Depending on control settings and specified grid codes, the faster dynamic response can be beneficial for the system stability, where, for instance, a fast reactive power response of converter-based wind power or HVDC can help to stabilize the system following a disturbance. However, other components, such as power electronic interfaced loads, inhibit characteristics that may cause instability events to develop faster, causing existing monitoring and emergency systems to act too slowly. The dynamic response following a contingency can typically not be captured using static assessments, and studies have indicated that static models of the power system might be insufficient when estimating the actual security margin [13, 17]. Faster dynamics will also result in power system instability developing more quickly, increasing the need for faster methods for instability detection. In comparison to the often used static models, there are more advanced methods available that can better account for the actual dynamic response in a power system. However, even with recent progress in high-performance computing, these methods are not readily available for use in real-time monitoring of a large power system. To overcome this issue, various data-driven methods and machine learning (ML) methods have been proposed in the literature. The main advantage

of ML methods is that time-consuming computations can be performed in an offline setting. Once a ML algorithm is trained, it can almost instantaneously provide estimations and warnings to operators that otherwise would require time-consuming computations. However, despite years of research, examples where ML methods have been practically applied in system operators' monitoring and control systems are, to the author's best knowledge, very few. These methods still suffer from robustness issues and uncertainty when handling operating conditions not included in the training of the algorithms. Blackouts and other major failures are related to extremely high costs, and from a system operator's point of view, a robust method that always works is generally preferred to a generally more efficient method that in some instances does not.

Another challenge in operating a power system is the design of SPSs and suitable control actions. Power systems are very large and interconnected systems that exhibit complex characteristics with a large number of states, nonlinear dynamics, and system uncertainties [18, 19]. To determine what set of actions at a unique operating point might stabilize the system most cost-efficiently, or what actions can ensure that the system is returned to a secure state, is far from trivial. The result is that system operators generally have to rely on relatively simple and robust control schemes which might either take overly drastic actions or choose actions that are sub-optimal from a system perspective. Reinforcement learning (RL), a sub-field in ML aimed for control purposes, has received significant attention in the research lately and has proven effective in solving complex control problems in environments such as games [20, 21], autonomous driving [22], and robotics [23]. Its applications in power systems are still relatively unexplored in the research, but successful implementations of it could allow more efficient control and a safer operation of the power system.

Based on the problem overview and the motivation of the thesis, the following research questions have been identified:

- **Research question 1:** How will the transition towards a higher penetration of converter-based renewable generations and loads in the power system affect current methods for security margin assessment?

- **Research question 2:** How can the current methods for assessment of security margins be improved, taking into account assessment speed, accuracy, and robustness?

- **Research question 3:** What are the requirements for future voltage instability detection methods? How can they be developed to be sufficiently advanced to capture the intricate dynamics during a voltage collapse, while at the same time being fast enough to be used in real-time?

- **Research question 4:** What are the requirements for future control systems used to maintain a stable and secure power system? How can they be developed to handle the highly non-linear and complex dynamical nature of an

electric power system?

- **Research question 5:** In the case of data-driven methods, what are the main practical aspects to consider to mitigate robustness issues and other barriers to allow the implementation of such methods in real power system applications?

1.3 Previous work

In this section, an overview of previous work related to the developed research questions is presented.

To ensure that a power system can handle the dynamic event following a disturbance, system operators often use an approach referred to as dynamic security assessment (DSA). In DSA, time-domain analysis is typically used to test a power system's dynamic response after a disturbance to ensure its ability to reach a stable post-disturbance operating point [12]. The assessment of a power system's dynamic stability with respect to a large set of different possible contingencies is a complex and time-consuming task and is typically impractical for real-time applications [24]. However, progress in high-performance computing has enabled close-to-real-time DSA to be deployed and tested in various systems. In [25], the deployment of an online DSA platform was reported in the PJM control center (a regional transmission organization in the United States), which was able to process over 3,000 different contingencies every 15 minutes. In [26], a computing platform was shown to be capable of simulating 1,000 contingencies for a test system consisting of 20,502 buses and 3,263 generators in under 27 minutes. Another example includes a platform for transient stability which was able to simulate over 4,000 contingencies across six pre-fault operating points on a 25,000 bus system at an average of 200 seconds [27]. In [28], an online DSA platform, adapted by California's independent system operator, was evaluated, showing that it was capable of performing DSA within 11-15 minutes for a base system state and with respect to 60 different contingencies.

Despite these recent advances in high-performance computing, the required time to assess the dynamic security with respect to a large range of different contingencies is still relatively substantial. To overcome this issue, various ML methods have been proposed in the literature. The idea of using ML is to train a classifier, using both simulated and historic data, to be able to interpret the system state and assess its security. Once the classifier is trained, it can provide almost instantaneous assessments of the system's dynamic security. The concept is originating from research undertaken in [29, 30] and a large number of publications in this research area have since been published. A large range of different classifiers have been proposed and are most commonly based on decision trees (DTs) [16, 31–35] or neural networks (NNs) [30, 36, 37].

In DSA, system operators are only provided with information regarding whether the *current* operating point is dynamically secure. An alternative measure of the *margin* to the most stressed point where the system can remain dynamically secure

following a disturbance is the secure operating limit (SOL). The SOL is the margin to the most stressed *pre*-contingency operating point that can withstand a set of credible contingencies [13, 15]. In some applications where the SOL is mainly used in monitoring voltage stability margins, it has also been referred to as the dynamic voltage security margin (DVSM). SOL estimation requires several time-domain simulations to trace the security limit for a set of different contingencies, which is not feasible to perform in the time frame needed by system operators. In [13], an attempt to reduce the computational cost in estimating the SOL was developed based on quasi-steady state (QSS) simulations. The method was further developed in [38], where a combination of QSS and time-domain simulations was proposed to include the impact of short-term effects during the transient state following a disturbance. Although this approach reduces the computational effort compared to a full time-domain simulation, it may still prove too slow for some real-time applications. In [39–42], different ML approaches based on NNs were proposed to allow real-time estimation of the DVSM.

Following more severe events and disturbances, system operators need to identify whether the current system state is stable, or if it is drifting towards instability. There has been research involving different voltage instability detection (VID) methods, where the complexity and the scope of the developed methods vary significantly [43]. A drawback of several of the developed VID methods is that they provide a relatively late indication of instability, which reduces the available time system operators have to control the system back into stable operation and increases the risk of a voltage collapse. An alternative approach is voltage instability prediction (VIP), where the future state of the system is *predicted* using information and measurements gathered in the (short) time left following a voltage instability event. A method for VIP based on ML was first proposed in [31], where a DT was trained on a generated database consisting of the intermediate short-term equilibrium that follows a disturbance. This post-contingency state, where the majority of the electromechanical transients have died out, was referred to as the "*just after disturbance*"-state. Extensions of the method utilizing phasor measurements have later been proposed in [44–46], where the performance of different attributes or input data has been tested. A method based on training a NN to online monitor voltage security was proposed in [47]. An attempt to incorporate some time-related features to improve the performance in VIP was presented in [48], where a temporal decision tree (TDT) approach was proposed. The TDT method, further discussed in [49] and [50], could incorporate some time-related variables, such as the difference between two measurements for a specific value of elapsed time.

If a power system is no longer secure, or is even drifting towards an unstable state, system operators need to take actions to restore security and/or mitigate instability. Optimal control methods such as model predictive control (MPC) have in several studies been proposed to deal with various types of electric power system control problems [18, 51]. However, to be able to compute the optimal control actions in a time frame required by system operators, significant simplifications of the system model are then generally required. Data-based control schemes, such as in [52, 53], are an alternative approach based on an (offline) assessment of different states and

actions. The optimal actions for each state are stored in a database and supervised learning algorithms are trained to map a state to a set of optimized actions. However, the combinatorial complexity in assessing all the state-action pairs, especially if the number of available control actions is high, may result in slow learning, resulting in the method being unsuitable to handle changes (e.g. in topology) in the underlying power system.

In recent years, significant progress has been made in solving complex control problems by using RL. RL is a data-driven approach where a control agent learns an optimal policy through interactions with a real power system or its simulation model [54, 55]. Its combination with deep learning, called deep reinforcement learning (DRL), has proven effective in solving complex control problems in a range of different applications. DRL enables automatic high-dimensional feature extraction, making the control agent capable of handling the large number of states that are involved in electric power system control. Previous implementations of DRL in electric power system control have so far mainly been focused on emergency control, which has the role of controlling the system back into a stable state after a disturbance has already occurred [56]. Implementations include methods adapted for automatic voltage control [57–60], optimal load shedding [19, 61, 62], dynamic breaking [19], and oscillation damping [63]. DRL-based implementations adapted for preventive control found in the literature are few and in [56], the authors argue that the reason may be that these control problems have traditionally been formulated as static optimization problems. One example of an RL-based implementation for preventive control is presented in [64], which aimed to determine the optimal control of active power generation for preventing cascading failures and blackouts.

1.4 Aim of the thesis

The aim of this thesis is to develop a new real-time dynamic stability assessment and control tool (RDS-ACT) that can support system operators and allow more efficient utilization of the transmission grid. This is achieved partly by developing methods that can provide better knowledge of the actual security margins in real-time, and partly by developing better methods to detect and assess system conditions and disturbances that might cause the system to become unstable. Fast detection of voltage instability reduces the risks and the related costs of controlling the system back into stable operation, and it also allows system operators to operate their systems with higher confidence in quickly detecting instability. Furthermore, this thesis develops two different methods that in real-time can suggest optimized actions to either restore the security margins above defined thresholds, or that can mitigate instability if a larger disturbance has taken place in the system. The capability to choose efficient and suitable control actions can both minimize the impact on the end consumers and significantly improve the operational efficiency during stressed operating conditions.

1.5 Main contributions

The main contributions of this thesis are the following:

1. A methodology that allows a fair comparison between the PCLL and the SOL is developed. Furthermore, the impact that various load models may have on these security margin estimation methods is evaluated. An extensive numerical comparison between the SOL and the PCLL is performed for a range of both static and dynamic load model configurations to determine under what circumstances the SOL is preferable to the PCLL.

2. A method for fast and robust computing of the DVSM is proposed and tested. The method is based on ML to support the estimation of the DVSM, which otherwise is typically too time-consuming to perform in real-time. To mitigate inconsistency issues associated with ML methods under new or unseen operating conditions, a method to quickly validate the estimated results is developed.

3. A method for VIP using an recurrent neural network (RNN) with long short-term memory (LSTM) is developed. This specific design of the network can utilize previous measurements and information, such as the trend of bus voltage magnitudes, tap changes, or fault locations, to improve the accuracy for VIP. A methodology for including consecutive contingencies (N-1-1) into the training data for the VIP algorithm is presented. A new training approach of the VIP algorithm is developed to provide system operators with an *online* assessment tool. As time progresses after a voltage instability event, the network is capable of incorporating new observations and continuously updating the assessment.

4. A DRL-based method for *preventive* control is developed. The method can in real-time suggest optimal control actions to maintain a sufficient SOL to ensure a secure operation of a power system. The developed DRL method uses a hybrid control scheme that is capable of simultaneously adjusting both discrete (e.g. switching of a shunt capacitor) and continuous (e.g. the level of active power generation rescheduling and load curtailment) action variables.

5. A DRL-based method for *emergency* control is developed aimed to mitigate long-term voltage instability. The method can monitor the system state, and in case of larger disturbances, it can in real-time provide fast and efficient control actions to stabilize the system. The DRL control is trained to use system services from demand response (DR) and energy storage systems (ESS) as a more efficient and flexible alternative to stabilize the system, compared to, for instance, conventional load shedding.

1.6 List of publications

Below is a list of publications included in the thesis.

Paper I: H. Hagmar, L. A. Tuan, R. Eriksson, "Deep Reinforcement Learning for Long-Term Voltage Stability Control" in *2022 IREP Bulk Power System Dynamics and Control Symposium*, 2022. Banff, Canada, 2022.

Paper II: H. Hagmar, R. Eriksson, L. A. Tuan, "Real-time Security Margin Control Using Deep Reinforcement Learning" submitted to *IET Smart Grids*, 2022.

Paper III: H. Hagmar, L. A. Tuan, R. Eriksson, "Impact of static and dynamic load models on security margin estimation methods" in *Electric Power System Research*, Jan. 2022. DOI: 10.1016/j.epsr.2021.107581.

Paper IV: H. Hagmar, L. Tong, R. Eriksson, L. A. Tuan, "Voltage Instability Prediction Using a Deep Recurrent Neural Network" in *Transactions on Power Systems*, Jan. 2021. DOI: 10.1109/TPWRS.2020.3008801.

Paper V: H. Hagmar, R. Eriksson, L. A. Tuan, "Fast Dynamic Voltage Security Margin Estimation: Concepts and Development" in *IET Smart Grid*, Aug. 2020. DOI: 10.1049/iet-stg.2019.0278

Paper VI: H. Hagmar, L. A. Tuan, O. Carlson, R. Eriksson, "On-line Voltage Instability Prediction using an Artificial Neural Network" in *Proc. 2019 PowerTech Milan*, Milan, Italy, 2019. DOI: 10.1109/PTC.2019.8810808.

Paper VII: H. Hagmar, L. A. Tuan, O. Carlson, R. Eriksson, "A Survey of Voltage Stability Indicators Based on Local Synchronized Phasor Measurements" in *Proc. 2018 North American Power Symposium (NAPS)*, Fargo, ND, 2018. DOI: 10.1109/NAPS.2018.8600607.

Additional paper produced during the PhD studies but not included in the thesis:

H. Hagmar, L. A. Tuan, O. Carlson, R. Eriksson, "Integration Aspects of Full Converter Wind Turbines and the Impact on Long-term Voltage Stability" in *Proc. 2019 IEEE Power and Energy Society General Meeting*, Atlanta, GA, USA, 2019. DOI: 10.1109/PESGM40551.2019.8973939.

1.7 Thesis outline

The thesis is organized as follows:

Chapter 2 develops the definitions used in the rest of the thesis and also provides an overview of current practices in security and stability assessment, with an emphasis on voltage stability and VID methods. Moreover, a general framework for electric power system control and the main challenges involved are presented. Parts of the chapter are based on the summary and results

established in Paper VII.

Chapter 3 provides a theoretical background on the concepts of ML and NNs. The concept of RNNs with LSTM is introduced. A general framework of RL and DRL is then presented and a few basic theoretical concepts and the proximal policy optimization algorithm is introduced.

Chapter 4 introduces the overall functionality of the proposed RDS-ACT. Furthermore, the test system used in evaluating the developed RDS-ACT is presented and its general characteristics are briefly discussed.

Chapter 5 presents both theoretical and numerical comparisons of the static and dynamic security margins under various load configurations. The chapter is mainly based on the results established in Paper III.

Chapter 6 introduces the concept of DVSM and discusses and illustrates the difference to conventional voltage security margin (VSM) using transient P-V curves. A method for fast estimation of the DVSM is then proposed and tested. The chapter is mainly based on the methodology and the results established in Paper V.

Chapter 7 presents a voltage instability prediction method based on an RNN using LSTM. The method allows system operators to continuously assess and predict whether the present system state is stable or will evolve into an alert or an emergency state in the near future. The chapter is mainly based on the methodology and the results established in Paper IV.

Chapter 8 presents a DRL-based method for preventive control to ensure sufficient security margins in an electric power system. The method is capable of simultaneously controlling the switching of reactive shunts as well as performing load curtailment to alleviate system stress in highly loaded areas. The chapter is mainly based on the methodology and the results established in Paper II.

Chapter 9 presents a DRL-based method for emergency control to mitigate LTVS events. The method use and control ancillary services provided by DR and ESS to mitigate instability. The chapter is mainly based on the methodology and the results established in Paper I.

Chapter 10 highlights the key conclusions of the thesis and provides ideas for future research.

All figures included in this thesis have been created by the author.

Chapter 2

Stability assessment and control of electric power systems

This chapter describes the definitions and criteria of stability phenomena that are used in this thesis. The chapter has an emphasis on assessment methods aimed at voltage stability, which is the main focus of the thesis. The overall drivers for different instability phenomena are discussed along with a presentation of the current practices of stability and security assessment. A brief overview of the different methods used in voltage instability detection is presented. Parts of the chapter are based on the summary and results established in Paper VII.

2.1 Classification of stability phenomena

Power system stability has traditionally been classified by the most common system variables in which instability can be observed, namely; voltage stability, rotor angle stability, and frequency stability [7]. However, the dynamic behavior of power systems has gradually changed due to an increasing penetration of converter-interfaced generation (CIG), loads, and transmission devices. In a report published in 2020 by a joint task force set up by IEEE Power System Dynamic Performance Committee and CIGRE, the classic definitions and classifications of basic stability terms were updated to also incorporate the effects of fast-response power electronic devices [65]. In addition to the original stability categories, two new stability classes were introduced, namely converter-driven stability and resonance stability.

Each stability phenomenon and its drivers are briefly explained in the following sections. Although the developed tools in this thesis can be generalized to other types of stability phenomena, the main focus has been the assessment and control of voltage stability events, with an additional emphasis on LTVS. Thus, in the following sections, a focus on methods and analysis used in voltage stability assessment is presented. An overview of the different stability phenomena is illustrated in Fig. 2.1.

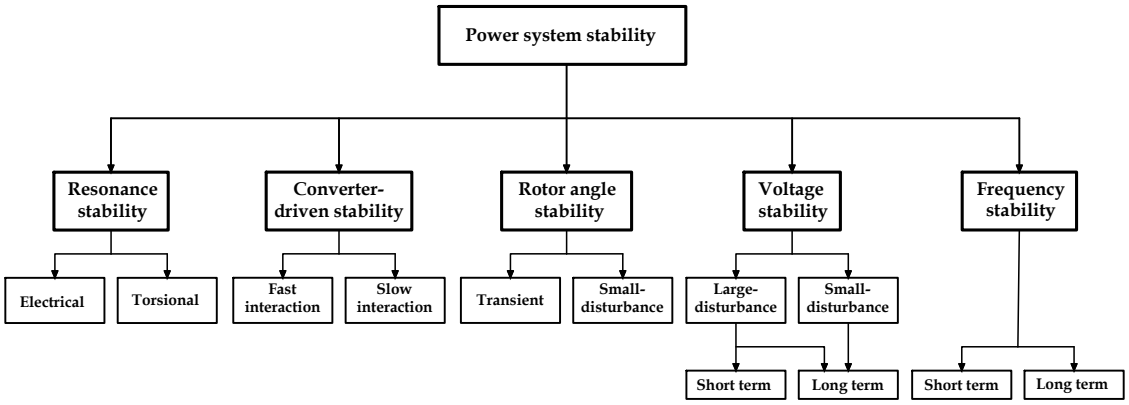

Figure 2.1: Updated classification of typical power system stability phenomena, adapted from [66].

2.1.1 Rotor angle stability

Rotor angle stability refers to the ability of synchronous machines in a power system to remain in synchronism after being subjected to a disturbance [7]. The rotor angle stability depends on the ability to maintain and restore equilibrium between electromagnetic torque and mechanical torque of each synchronous machine in the system. Rotor angle stability is further divided into two subcategories; small signal (caused by small disturbances) rotor angle stability and transient (caused by large disturbances) rotor angle stability.

2.1.2 Frequency stability

Frequency stability refers to the ability of a power system to maintain a steady frequency following a disturbance that causes a significant imbalance between generation and load [7]. It depends on the ability to maintain and restore equilibrium between system loads and generations, with a minimum of unintentional loss of load. Instability in the form of sustained frequency swings may lead to tripping of generating units and/or loads. Frequency stability, as identified in Fig. 2.1, may be divided into a short-term phenomenon or a long-term phenomenon. During frequency deviations, the time frame of the processes and devices that are activated will range from a fraction of a second, such as the response of under-frequency load shedding and generator controls and protections, to several minutes, corresponding to the response of devices such as prime mover energy supply systems and load voltage regulators.

2.1.3 Voltage stability

Voltage stability refers to the ability of a power system to maintain steady voltages at all buses in the system after being subjected to a disturbance from a given initial operating condition. It depends on the ability to maintain/restore equilibrium between load demand and load supply from the power system [7]. Voltage instability in a power system may lead to loss of loads or disconnection of other components such as tripping of transmission lines or generators due to too low voltage levels.

The concept of *voltage collapse* is generally referred to as a sudden event, often initiated by a larger disturbance or by a sequence of events, leading to a blackout or abnormally low voltages in the whole, or larger parts, of a power system. Historic events of voltage collapses are relatively few, but the related costs to society are extremely high [12]. The voltage stability phenomenon is complex and often requires a full network representation for its analysis [7]. Reactive power plays an important role in voltage stability, and voltage stability problems are often related to the incapability of the system to provide sufficient reactive power to keep system voltages at nominal levels. However, the *main* driver for voltage instability in power systems is the loads and the load restoration that follows a disturbance [12]. Historically, the majority of all voltage instability incidents experienced so far have resulted from larger disturbances, such as the loss of generation or transmission capacity. The time frames of voltage stability problems caused by larger disturbances are generally divided into a short-term or a long-term phenomenon, as identified in Fig. 2.1.

- *Short-term voltage stability*: involves dynamics of fast-acting load components such as induction motors, loads with power electronic interfaces, and HVDC converters [7]. The study period of interest is in the order of a couple of seconds, and the analysis generally requires detailed time-domain (dynamic) simulations. One of the most common causes has been attributed to induction motor stalling and HVDC links connected to weak alternating current (AC) systems. Stalling induction motors draw a very high reactive current, stressing the grid significantly and may cause cascading faults, resulting in a swift voltage collapse [67].

- *Long-term voltage stability*: involves the analysis of slower acting equipment such as tap-changing transformers, thermostatically controlled loads, and generator current limiters. The study period of a typical LTVS event may extend to several minutes [7]. Following a typical disturbance with a loss of generation or transmission capacity, the loss of generation is redirected and the remaining transmission lines have to carry a larger current, causing increased active and reactive power losses and voltage drops. The reduced system voltages initially affect voltage-dependent loads, which are then restored by the action of, for example, load tap changers (LTCs), distribution voltage regulators, motor slip adjustments, or thermostatic load restoration. The restored loads will further increase the active and reactive power flow through a mainly inductive transmission system, which increases the reactive power losses and further deteriorates the system voltages. The system may eventually collapse when the load dynamics attempt to restore the loads beyond the capability of the generators and the transmission network.

The above-mentioned examples are generally referred to as a *large-disturbance voltage stability events*. Less common are voltage instability caused by small disturbances, such as incremental changes in system load. Since system operators continuously ensure that sufficient margins are kept to fulfill the N-1 contingency criterion, small-disturbance events are very seldom the sole cause of voltage instability. How-

ever, the methodology used in the small-disturbance analysis is often valuable in the analysis of the voltage security margins, which is further discussed in Section 5.1.4.

2.1.4 Resonance stability

Resonance refers to periodically and oscillatory exchanges of energy which, in the case of insufficient dissipation of energy in the flow path, may grow and cause magnification of voltage/current/torque magnitudes. When such magnitudes exceed specified thresholds, it is said that resonance instability has occurred [65]. Resonance stability is divided into two possible forms: (1) a torsional resonance due to resonance between series-compensated lines and the mechanical torsional frequencies of turbine-generators shafts, and (2) an electrical resonance between series-compensated lines and the electrical characteristics of generators.

2.1.5 Converter-driven stability

The dynamic behavior of CIG is significantly different from that of conventional synchronous generators, due to the predominant voltage source converter interface with the grid [65]. CIG is controlled through control loops and algorithms with fast response times, such as the phase-locked-loop and inner-current control loops. The controls of CIG will affect both electromechanical dynamics of machines and electromagnetic transients of the network, which in turn may lead to unstable power system oscillations of a wide frequency range.

Converter-driven stability is divided into two subgroups; fast-interaction converter-driven stability and slow-interaction converter-driven stability. Fast-interaction converter-driven stability refers to instabilities that involve system-wide stability problems driven by fast dynamic interactions of the control systems of power electronic-based systems with fast-response components of the power system. These include the transmission network, the stator dynamics of synchronous generators, or other power electronic-based devices. The slow-interaction converter-driven stability refers to system-wide instabilities driven by slow dynamic interactions of the control systems of power electronic-based devices with slow-response components of the power system. These include the electromechanical dynamics of synchronous generators and some generator controllers.

2.2 Security assessment and instability detection

Security assessment refers to the daily operational and planning activities of system operators to ensure a secure and stable operation of a power system. The main goal is to continuously ensure a *secure normal (operating) state* in the system. A system is said to be operated in a normal state if both the *load constraints* and the *operating constraints* are satisfied [12]. The load constraints ensure that the load demand is met by the generation in the system, while the operating constraints ensure that minimum or maximum limits, in terms of variables such as line currents and bus voltages, are satisfied.

In the event of a disturbance in the system, the system may either settle to (i) a (new)

secure operating state, (ii) an (unsecure) alert state, or (iii) end up in an emergency state. Ensuring that a power system can be operated securely with respect to *all* possible disturbances would not be practically or economically feasible. In practice, the power system security is assessed with respect to a set of credible disturbances, generally referred to as contingencies. Power systems are almost exclusively operated according to the N-1 contingency criterion, meaning that the system should be able to withstand the loss of any single component, such as transmission or generation capacity, without the system entering an emergency state. A system capable to handle such an event without entering an emergency state is said to be *secure*. A system that cannot handle such an event without entering an emergency state is said to be operated in an *unsecure* state.

With reference to dynamical system theory, an operating state (or equilibrium) x^* is called *stable* if all solutions with an initial condition close to x^* remain near x^* for all time [12]. An equilibrium that is *not* stable is called *unstable*. Thus, a system can be stable at a current operating state, while at the same time not fulfilling the security criterion of handling a larger disturbance and still maintaining stability. An unstable system can be exemplified by a power system drifting towards voltage instability, driven by the action of load restoration devices.

The different operating states are illustrated in Fig. 2.2 and in Fig. 2.3, where the state space of a power system is reduced and illustrated in only two dimensions. In Fig. 2.2, the system is operated in the secure region. A disturbance causes the system state to transition to the unsecure region. Here, the system is still stable, but if it is further disturbed it may be at risk of becoming unstable with major blackouts as a possible result. The system can then be transferred back to the secure region through preventive control actions and sufficient remedial actions. Alternatively, in Fig. 2.3, the system is operated in an unsecure state. Following a disturbance, the

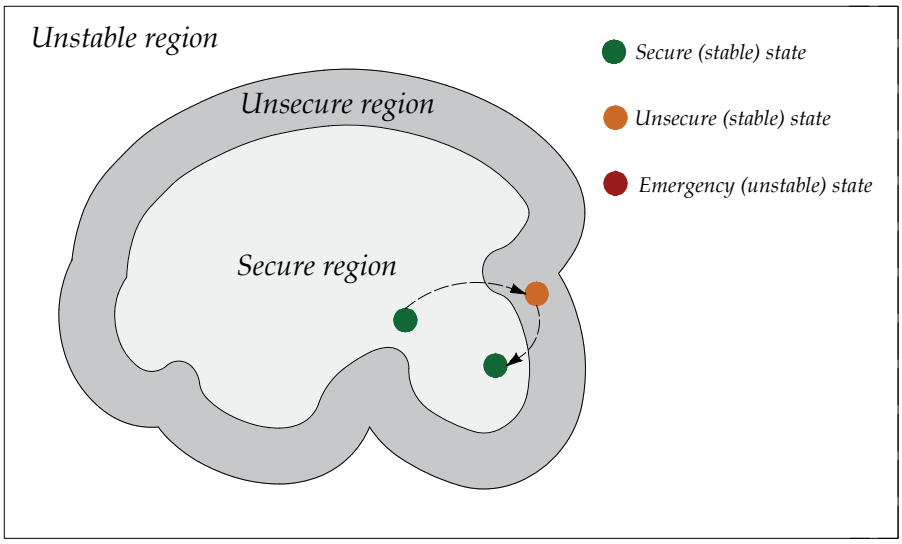

Figure 2.2: An example of a normal operating state, followed by a disturbance. Through preventive control, the system operating state is restored to normal.

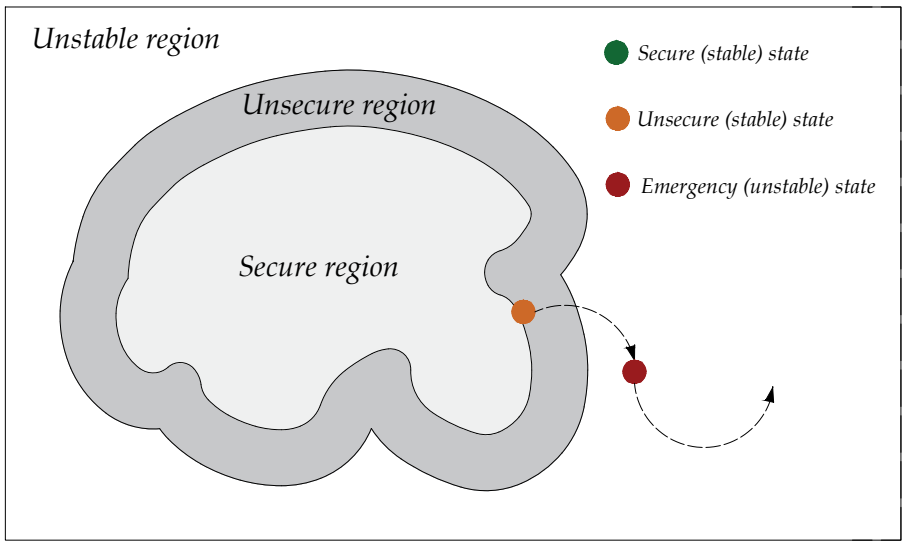

Figure 2.3: An example of an unsecure operating state, followed by a disturbance. The system state ends up in an emergency state, and without sufficient emergency control actions, the system will collapse.

system becomes unstable and enters an emergency state, and without sufficiently fast counteractions and emergency control, a larger system collapse may be imminent.

Generally, the stability assessment of a system operator can be divided into two different but complementary lines of defense used to avoid instability: *preventive* and *emergency* methods. The preventive methods are mainly used during normal operation to ensure that the power system is operated securely according to the N-1 contingency criterion. Emergency methods are instead used when a disturbance (or several) has already occurred. The aim of these methods is typically to perform instability detection, allowing system operators to identify imminent instability and trigger fast remedial actions. In the following subsections, the main practices in these two applications of stability assessment are presented.

2.2.1 Preventive assessment techniques

Security assessment involves assessing a power system's ability to undergo disturbances for a given operating point. The level of accuracy in the assessments varies, where simpler methods are based on static assessment methods and assessing the post-contingency long-term equilibrium [12]. A static load flow assessment of a power system will have no solution if there is an absence of a post-contingency long-term equilibrium, which provides a simple way to check whether an equilibrium of the system exists. However, this simple approach suffers from several drawbacks. An absence of a post-contingency long-term equilibrium may be the result of purely numerical problems that are not related to the system being unstable. Other drawbacks include the lack of information regarding the nature and the location of the problem [12].

To overcome these issues, and to also ensure that a power system can handle the

dynamic event following a large disturbance, system operators often use DSA. DSA refers to the analysis to determine whether or not a power system can meet security and reliability criteria in both transient and steady-state frames [68]. Commonly, time-domain analysis is used to test the power system's dynamic response for a range of different contingencies [16]. In general, DSA includes assessing not only voltage stability, but also any criteria such as thermal overloading, transient stability, or frequency stability.

More than just ensuring that the *current* operating condition is secure, the preventive assessments should also provide information to system operators of the *margin* to instability. Security margins are related to system stress, generally in terms of an increased transfer of active power. The security margins are also characterized by the direction of system stress in the parameter space by load increases and generation scheduling of various buses [12]. Two main approaches are used to compute the security margins of a power system: the PCLL and the SOL [12, 69]. The PCLL is evaluated by estimating the loadability limit of a post-contingency operating point, where a solution path is traced by iteratively increasing the system stress until the system's critical point is reached. The characteristics of the iteratively increased system stress in the post-contingency setting are similar to that of the slow load restoration that typically follows in a LTVS event. The SOL instead refers to the most stressed *pre-contingency* operating state in which the system can withstand a specified set of contingencies. In [13], the SOL is defined to encompass three types of different information:

- *A direction of system stress*: the system stress is a combination of load demand increase and/or generation rescheduling. These are quantities that the system operator can observe and control in the pre-contingency state.
- *Operator controller actions while the system is stressed*: the actions taken by system operators or controllers as a response to the increased system stress imposed in the pre-contingency situation. Typically, their role is to keep the voltage profile within limits and maximize the available reactive reserves. Compensation switching, generation redispatch, adjustments of generator voltages, and secondary voltage control are typical examples of actions likely to increase system security.
- *Post-contingency corrective controls*: the actions taken by SPSs in the system after the system has been subjected to a disturbance. Some are less intrusive such as compensation switching or increases in generator voltage set-points, while other actions such as load shedding or system separation are typically only used as a last resort to avoid a system collapse.

Computing the transition from a *pre-contingency* operating point to a *post-contingency* operating point is numerically infeasible when using static assessments and when the system is close to the system collapse point. Thus, the SOL generally requires either time-domain or QSS simulations to replicate the dynamic response following a disturbance. The SOL provides a security margin with respect to not only voltage instability, but also other stability-related phenomena such as rotor angle stability

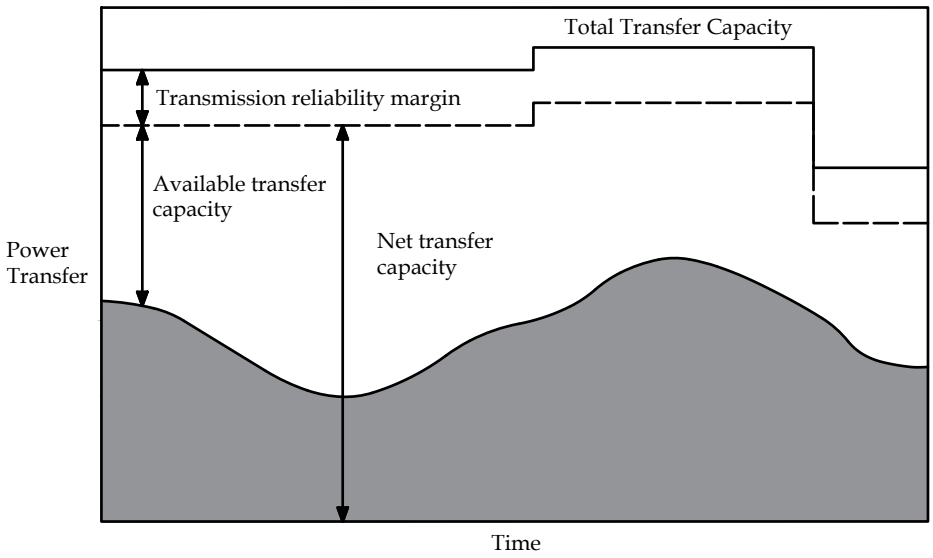

Figure 2.4: Different measures of transfer capacities and reliability margins under changing operating conditions.

or inter-area oscillations. In some parts of this thesis which specifically concern the assessment of voltage stability the more specific term of DVSM will be used.

The security margins are used to compute the transfer capacities in a power system, which sets the limit for how much power can be transmitted through the system. The different margins and capacities are illustrated in Fig. 2.4 with the following definitions [70]:

- **Total transfer capacity (TTC)**: the maximum transmission of active power between different areas/subsystems which is permitted with respect to a given security criterion (most commonly the N-1 criterion).
- **Transmission reliability margin (TRM)**: the security margin that takes into account the uncertainties on the computed TTC values. The TRM is in some cases arbitrarily determined, but is generally based on [70]:
  1. Unintended deviations of physical flows during operation due to the physical functioning of load-frequency regulation.
  2. Emergency exchanges between system operators to cope with unexpected unbalanced situations in real-time.
  3. Inaccuracies in, for instance, data collection, models, and measurements.
- **Net transfer capacity (NTC)**: defined as NTC = TTC - TRM and is the maximum exchange possible between two areas when taking into account the uncertainties in the TTC estimation.
- **Available transfer capacity (ATC)**: the maximum incremental transfer capacity possible between two parts of a power system without violating the

security margins.

The ability to accurately assess and compute the transfer capacities and the related security margins is of very high importance to a system operator. Different methods for estimating the security margins in a power system, as well as comparisons between static and dynamic estimation approaches, are presented in Chapter 5.

2.2.2 Emergency assessment techniques

The previously described methods fall under the category of *preventive* assessment methods. In the case of more severe incidents, such as the occurrence of multiple simultaneous contingencies, or in the event that the preventive methods have not been accurate or fast enough, the system operators have to rely on emergency methods to quickly detect instability, and then initiate suitable control actions to steer the system back into a normal operation again. The aim of such emergency assessment techniques is thus to detect the onset of instability itself, rather than its consequences [43].

The time scales of different stability phenomena will have an impact on the assessment methods that are feasible to use. For instance, in transient angle stability, the time scale of a typical instability event often ranges between a few hundred milliseconds up to a maximum of a few seconds, while that of a long-term voltage instability may have a time scale that ranges several minutes. Any emergency assessment technique aimed at transient stability will thus have to automatically trigger control actions, while those for long-term voltage instability may be used to warn system operators, which then manually choose suitable actions to protect the system stability. From here on, this thesis focuses on emergency assessment techniques used for VID methods, which are used to detect the onset of (mainly long-term) voltage instability.

One of the most simple methods for VID, yet a commonly used one, is to monitor voltage magnitudes in the power system [43]. If the voltages drop below a certain threshold value, it can provide an indication that the system may be headed towards a collapse. However, using system voltages as an indicator of voltage instability can be problematic. In Fig. 2.5, the evolution of two transmission bus voltages is presented for the modified Nordic32 test system after a disturbance, which illustrates the problem of using voltage magnitudes as indicators for voltage instability. The Nordic32 test system is further presented in Section 4.2 and is a commonly used test system for voltage stability studies. The disturbance of concern is a tripped transmission line between buses 4032 and 4044. The voltage instability is gradually developed after the disturbance, driven by components such as LTCs and over-excitation limiters (OELs). The voltage magnitudes are relatively high following the disturbance, but at some point, the mechanism of load power restoration causes the system to deteriorate to such a degree that the total power consumed in the system is reduced instead of restored for each attempt of load restoration. Through the reactive power support of generators and other voltage control devices, the voltage magnitudes are kept at a relatively high level for a long time after the actual disturbance. To overcome this issue, there has been significant research in

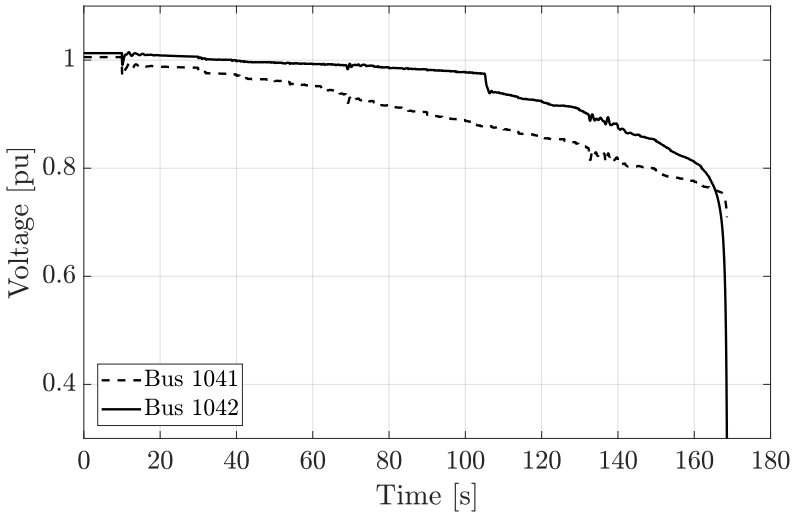

Figure 2.5: Evolution of bus voltages during a voltage instability event.

the development of new and faster methods for VID. The development in phasor measurement technology also allows several new perspectives and methods for wide-area monitoring and control of power systems. The methods differ in their approach, both in complexity and the required level of system observability. An overview of common methods used in VID is presented in Section 2.3.

2.3 An overview of VID methods

The aim of VID methods is to as fast as possible detect, or even predict, the onset of voltage instability. Numerous different methods have been proposed in the literature where the complexity and the requirements of the available measurement infrastructure vary significantly. The methods may also significantly differ in the accuracy and speed of the detection, as well as robustness to errors and other functionality. Available methods for VID can be divided into two different main categories:

1. *VID methods based on local measurements*: A VID method is in this thesis defined to be local if it relies on measurements from only two or fewer buses. Thus, those VID methods where measurements are required on both sides of, for instance, a transmission corridor, are also considered to be local VID methods.

2. *VID methods based on the observability of the whole region*: These methods are generally more accurate than VID methods that are based on local measurements. However, as the name indicates, they require full or close to full observability of the monitored region, and the measurements used in these models should preferably be filtered through a state estimator causing increased computation time and complexity.

The following section is a brief overview of a few certain methods for VID, illustrated in Fig. 2.6. The number of different methods for VID is vast and the overview here is not meant to be exhaustive but is rather intended to provide an overview

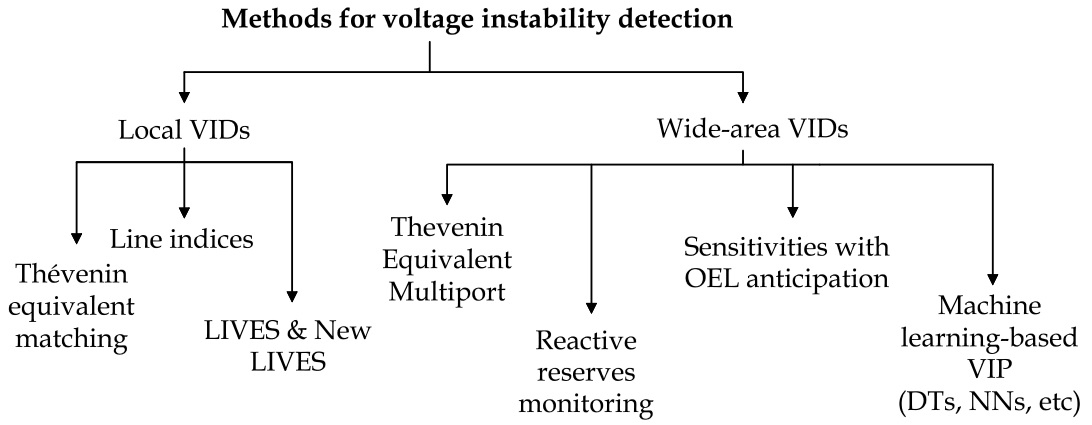

Figure 2.6: An overview of different types of methods aimed for VID.

of the available methods and their advantages and disadvantages. More thorough overviews and comparisons for different types of VID methods can be found in [43, 71, 72] and in paper IV.

2.3.1 Thévenin equivalent matching

VID methods based on Thévenin equivalent (TE) matching use the TE impedance as an indicator of the margin to voltage instability [73]. Considering the simple two-bus system in Fig. 2.7, it can be shown that the maximum transferable apparent power in the system occurs when

$$|\overline{Z}_{th}| = |\overline{Z}_L| \tag{2.1}$$

where $\overline{Z}_{th}$ is the complex TE impedance, and $\overline{Z}_L$ is the complex load impedance. The relationship between the TE equivalent impedance and the other system parameters may be stated as:

$$\overline{E}_{th} = \overline{V}_L + \overline{Z}_{th} \cdot \overline{I} \tag{2.2}$$

where $\overline{E}_{th}$ is the TE voltage, and $\overline{V}_L$ and $\overline{I}$ are the complex load voltage and current, respectively. Using the relationship in (2.2), the values of $\overline{Z}_{th}$ can be estimated. The real and imaginary values of $\overline{E}_{th}$ and $\overline{Z}_{th}$ in (2.2) result in four unknowns, requiring phasor measurements to be taken at two or more times to solve for the unknown parameters. By tracking and comparing the TE impedance and the load impedance, the system operator can assess the load margin to instability, for each bus in the system that the index is computed for. An indication of instability occurs when the load impedance becomes lower than the TE impedance. It should be noted that this margin is *not* the same as the VSM or the DVSM presented in Section 5, as it only provides a margin to instability in the system with respect to a theoretic *load increase*, and not an N-1 security margin.

2.3.2 Line VID methods

Line VID methods are based on measurements being available on both sides of a two-port transmission line. The number of different line VID methods is vast, although

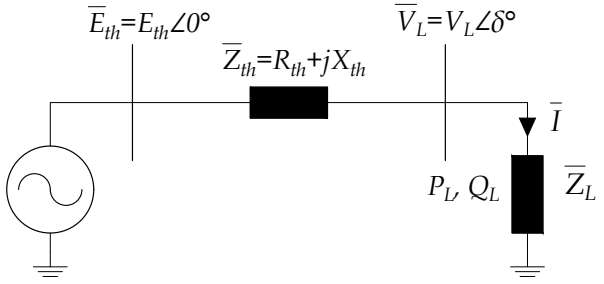

Figure 2.7: A two-bus Thévenin equivalent circuit.

many are based on the same concepts. Over time, several line VID methods based on the concept of maximal transferable power have been developed. These are similar to the TE-based methods for the bus VID methods, with the difference being that measurements are required at each end of a transmission line. An example of an early developed line VID was the transmission path stability index in [74], which showed that the maximum power transfer occurs when the voltage drop equals the load-side voltage. Other line VID methods based on similar concepts are the voltage collapse proximity indicators in [75], where four indicators are developed, based on the maximum transferable power and the maximum possible line losses that may occur over a transmission line.

Other formulations of line VID methods include methods that are based on solutions to the classical voltage-power equation for a two-bus system with negligible resistance, later presented in (5.1). It can be shown that the maximum power transfer occurs when the value of the inner square root in (5.1) is zero. A line index called L_{mn} based on this formulation is presented in [76]. The index can be formulated as:

$$\frac{4XQ_r}{[V_s \sin(\theta - \delta]^2} = L_{mn} \leq 1.0 \tag{2.3}$$

where θ is the impedance angle, X is the line reactance, Q_r is the reactive power at the receiving end, δ is the voltage angle difference between the two nodes, and V_s indicates the sending end voltage magnitude. Instability is indicated whenever the stability index L_{mn} exceeds a value of 1.

2.3.3 LIVES and new LIVES

In [77, 78], a method called local identification of voltage emergency situations (LIVES) is introduced and tested. The LIVES stability criterion is based on monitoring the change in the secondary voltage after a tap decrease on the primary side ($\Delta r < 0$) of a LTC transformer, which simplified may be stated as:

$$\frac{\Delta V_2}{\Delta r} < 0 \tag{2.4}$$

where ΔV_2 is the change in the secondary voltage. Thus, if a tap decrease leads to a negative change in ΔV_2, this indicates an unstable condition. Further, the criterion

indirectly takes into account the effect of other taps acting in the system as it can observe the net effect of various LTCs over a cycle of tap operations. In [79], this concept is developed further by monitoring the stability condition of (2.4), solely from the transformer bus, given that the primary voltage $\overline{V}_1$ and current $\overline{I}_1$ phasor measurements are available. The decreasing tap change is measured indirectly as a conductance increase seen from the primary side, whilst the secondary voltage is indirectly monitored as an increase of consumed active power, ΔP. The new index, denoted as the new LIVES Index (NLI) is formulated as:

$$NLI = \frac{\Delta P}{\Delta G_1} > 0 \tag{2.5}$$

where

$$G_1 = Re\{\overline{I}_1/\overline{V}_1\}$$

Simulations show promise during several different grid conditions and topologies, allowing early indication of impending voltage collapses. The method is further tested in [80], where the method is extended and applied for distance relays of transmission lines feeding weak areas.

2.3.4 Thévenin equivalent multiport

In [81], it was shown that the TE matching methods do not work properly for multi-load systems. To overcome these difficulties, a concept called *coupled* single-port circuits was introduced, where an additional term modeling the coupling effects of generators and other load variations of other buses was added. In [82], it was found that the coupled single-port model may still yield underestimations if loads are not proportionally increasing. A modified coupled single-port model was then proposed to handle the underestimations that occur if loads are not proportionally increasing. Another extension was proposed in [83], which could better incorporate the dynamic nature of the grid equivalence in the estimations. Here, adjustments to the equivalent parameters of the coupled single-port model were calculated from two consecutive phasor measurements at the corresponding bus to capture the power system evolution. In [84], an improved method for TE impedance matching was proposed which could anticipate the activation of OELs to increase the speed of the VID method.

A clear drawback of any TE matching methods is that they cannot incorporate and foresee the load restoration that follows after a disturbance in the system, which is often the main driver for instability. It should be noted that these methods included in this section rely on PMUs covering *all* relevant generation and load buses, requiring a well-developed PMU configuration.

2.3.5 Reactive reserves monitoring

Voltage instability is highly related to the incapability of the system to provide sufficient reactive power reserves to maintain voltages around nominal values. By monitoring the reactive power reserves of system components, such as synchronous

generators and static var compensators, the remaining reserves can be used as an indicator for VID [43]. Several studies have examined reactive power reserves as an indicator for VID, examples including [85–88]. However, a difficulty in using reactive power reserves for VID is that reactive power cannot be effectively transmitted over long distances on inductive transmission lines. Hence, although reactive reserves do exist in certain areas of a power system, it is infeasible to transmit them to the areas with low system voltages. Thus, methods using reactive reserves monitoring need to only take into account that reactive power reserves need to be close to the affected area to be effective [86, 87].

2.3.6 Sensitivities with OEL anticipation

In [11], a method using the sensitivities of reactive power generation to reactive power loads is considered. The method fits a set of algebraic equations to the sampled state of a power system, which is either gathered directly from wide-area phasor measurements or from a state estimator. Then, the sensitivities are computed to identify when a combination of load powers has passed through the system's maximum. A change of sign in the computed sensitivities is an indicator that the system has become unstable. The method does not require explicit modeling of the system loads but will take into account LTC and OEL activation, either directly from measurements or by anticipation techniques. The method was tested and compared to conventional TE matching methods in [89], where the method proved to be significantly faster than conventional TE methods to identify voltage instability.

2.3.7 Machine learning-based VIP

The concept of VIP significantly differs from conventional VID methods. Most methods for VID attempt to detect when the system is near, or has already reached, the point of maximum load power. However, when this point is reached, instability can develop quickly and the remaining time for system operators to initiate emergency control actions may be too short. In contrast, machine learning-based methods for VIP attempt to *predict* the onset of instability. This would allow system operators to get an indication, almost instantaneously after a disturbance, whether that disturbance is going to cause a voltage collapse in the near future. These methods are generally based on off-line training of a chosen ML algorithm to be able to correlate a certain state space with the future state of the system. In Chapter 7, a new method for VIP is proposed and presented.

2.4 Overview of power system stability control

Electric power system stability control is complex and its applications differ significantly with respect to the stability phenomena that the control is adapted for, the type of control, its location, and the scope of the control task. Furthermore, power system stability control must be cost-effective, robust, and have the capability to operate appropriately over a wide range of operating and disturbance conditions [90]. In this section, a brief presentation of various concepts and classifications commonly used in power system stability control is presented.

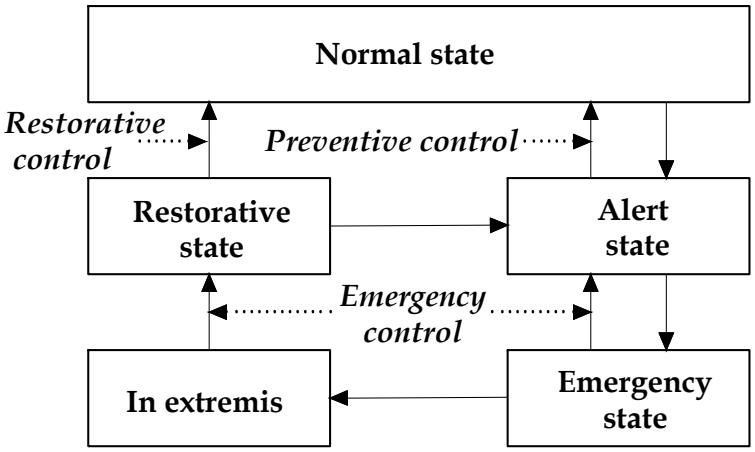

Figure 2.8: Power system operating states and controls, adapted from [91].

Power system stability control may, in the same way as for stability and security assessment be divided into *preventive* and *emergency* control applications [92]. The preventive control applications refer to the actions used by system operators to ensure that the system can maintain security and sufficient security margins, and they are issued whenever security criteria or margins are exceeded. Similarly, emergency control actions are used whenever the system has entered an emergency state, and the system operator is required to initiate fast actions to ensure that the system's stability is restored. Here, the main goal is often to avoid a system collapse or a larger disconnection of the grid, and more significant and expensive control actions are available for the system operator. Another type of control, called *restorative control* is often defined and is involved in restoring the system back to normal operation by reconnecting all facilities and restoring system load. Restorative control is generally issued after emergency control has managed to avert system instability after a large disturbance [92]. In Fig. 2.8, the different operating states and the control used to transfer the system between states are illustrated. In addition to the normal, the alert, and the emergency states earlier illustrated in Fig. 2.2, we here also show the *restorative state* and the *in extremis state*. In a restorative state, the operating constraints are generally satisfied, but not the load constraints (e.g., due to a blackout or a load shedding action). For the in extremis state, both the load constraints and the operating constraints are violated, and a major system collapse may be imminent if not sufficient actions are taken to stabilize the system.

Traditionally, power system stability has been maintained mainly through *local control*. This includes control actions such as under-voltage load shedding, excitation control, generator tripping, and reactive power switching, where local measurements in combination with feedback control and relays are used to maintain the controlled system parameter locally. *Wide-area control* is less common, but applications in the area have increased with the development of synchronized phasor measurement and better communication technology [90]. Wide-area control allows system operators to observe the whole system state and offer increased controllability and more efficient overall control. They may augment local controls, or provide adaptive functions

rather than the typically crude primary control used in local control schemes.

The majority of local control schemes are either based on different types of continuous feedback controls (e.g., the voltage control of excitation systems) or are rule-based (e.g., under-voltage load shedding schemes used to disconnect load whenever the measured voltage is below a certain threshold). There are also more advanced types of control schemes, including model-based control (e.g., model predictive control) that uses a model of the power system or the controlled component to improve the control. Other examples include different types of learning-based control, which is a model-free based control that learns an optimal control scheme through training on large sets of measured (or simulated) data. In this thesis, the focus is on using RL, which is a ML-based control, further discussed in Section 3.4.

Chapter 3

Machine learning and deep learning methods

This chapter provides a theoretical background of machine learning, with a focus on neural networks and reinforcement learning. The general structure and training aspects of neural networks are presented along with an introduction to recurrent neural networks and the long short-term memory architecture. Finally, the concept of Markov decision processes is introduced followed by an introduction to reinforcement learning and the proximal policy optimization algorithm. The presented theory is not intended to be exhaustive or cover all aspects of machine learning, but is rather focused on including the theory required for understanding the tools developed in this thesis.

3.1 Overview of machine learning methods

ML can be defined as a set of methods and statistical models used to perform specific tasks without using explicit instructions, instead relying on patterns and inference from data. ML algorithms differ in their approach, the type of task or problem that they are intended to solve, and the type of data used as input and output in the training. ML is often divided into three main types of learning, namely: *supervised learning*, *reinforcement learning*, and *unsupervised learning*.

In supervised learning, the ML algorithm is trained on a set of data that contains *both* the inputs and the desired outputs to perform tasks such as classification or regression. One of the simplest supervised learning algorithms is linear regression, which uses a set of N input-output training data pairs $\{(\boldsymbol{x}_1, \boldsymbol{y}_1)...(\boldsymbol{x}_N, \boldsymbol{y}_N)\}$. A linear regression model has the general form of $f(\boldsymbol{x}) = \boldsymbol{x}W + \boldsymbol{b}$, which is used to generate a linear function mapping each $\boldsymbol{x}_i$ to each $\boldsymbol{y}_i$. When training the linear regression model, the aim is to find the most suitable parameters for the weight matrix W and the bias vector $\boldsymbol{b}$, to minimize a chosen objective function (commonly the average squared error). Once the algorithm is trained, it should be capable to predict or estimate similar outputs from now unseen input data. In more general cases where

the relation between $\boldsymbol{x}$ and $\boldsymbol{y}$ is not necessarily linear, one may need to develop a nonlinear function mapping the inputs to the outputs [93]. In this thesis, we will mainly rely on various methods based on NNs to develop these nonlinear functions, which are further detailed in Section 3.2.

RL is an ML-based framework used for finding solutions to sequential decision-making problems. A decision-maker (an "*agent* ") attempts to learn an optimal policy by continuous interaction with an environment, where actions leading to high rewards are reinforced through various training schemes. The environment that RL uses is typically stated in the form of an Markov decision process (MDP), which provides a framework of how to model different state transitions and rewards functions based on the taken actions. The learning of RL algorithms is typically less stable than supervised learning algorithms and the issue of choosing whether to exploit a relatively good policy or explore new types of actions (generally referred to as the *exploration versus exploitation trade-off*) are two significant challenges with RL. In this thesis, we mainly use an algorithm called policy proximal optimization, which is further detailed in Section 3.4.5.

Unsupervised learning is used to find hidden patterns and structures in data, where the output data is unlabeled. This type of learning algorithm has not been used in the thesis and is therefore not further described in following the sections.

3.2 Neural networks and deep learning

Feedforward NNs, also known as multilayer perceptrons, are loosely inspired by the neurons in the human brain and their ability to classify and learn events from input data. The strength of NNs lies both in their capability of learning and approximating nonlinear functions and the scaling performance of the methods when trained on large sets of data. The universal approximation theorem, a famous theorem in NN mathematics, states that a feed-forward NN with a single hidden layer can approximate *any* given function arbitrarily well, provided that sufficiently many neurons are available in the hidden layer [94].

A typical NN with a single layer of hidden units is presented in Fig. 3.1. The NN consists of connected nodes, known as artificial neurons, stacked in different layers. Neurons in one layer only connect to neurons of the immediately preceding and the immediately following layers. The input layer receives external data, while the layer that produces the final result is denoted as the output layer. Between these layers are the hidden layers, where each neuron typically has a nonlinear activation function, which imitates the actions of synapses in a biological brain. The connections between each neuron are represented by weight parameters, and the aim of training a NN is to adjust these weights so that the network can form an effective mapping between the inputs to the outputs. In the following sections, we provide a general overview of the steps needed in training the network. We provide the NN model for a single hidden layer, which can then be generalized to NN models using multiple hidden layers, generally referred to as deep networks.

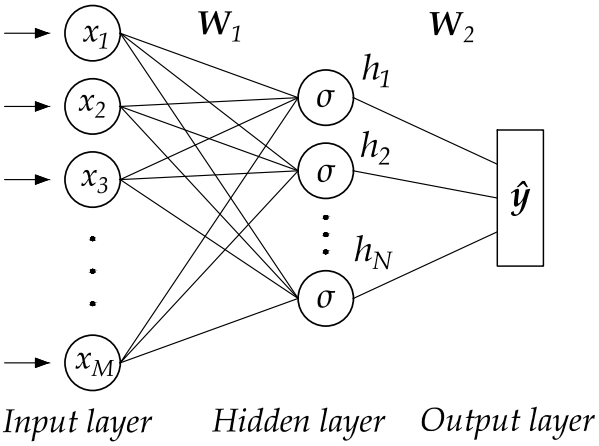

Figure 3.1: A simple NN with a single hidden layer.

3.2.1 The forward pass

The *forward pass* in a NN is used to compute the outputs, both in training as a means to adjust the weight parameters, but also in actual implementations where the outputs of the network are used as estimates or predictions. Following [93], the forward pass of the single hidden layered NN is presented below. Superscripts M and N refer to the number of input features and the number of hidden neuron cells in the hidden layer, respectively. Vector notation is used, meaning that, for instance, the output of the hidden layer $\boldsymbol{h}(\boldsymbol{x})$ is not the output of a single neuron cell, but the output of a vector of N neuron cells. The operation is summarized here for a single sample by the following steps.

The input layer first passes a row vector $\boldsymbol{x} \in \mathbb{R}^M$ of inputs through the weight matrix $\boldsymbol{W}_1 \in \mathbb{R}^{N \times M}$, illustrated by the lines connecting each of the cells in Fig. 3.1. The outputs of the hidden layer are computed by applying an element-wise nonlinear function to the weighted input values:

$$\boldsymbol{h}(\boldsymbol{x}) = \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) \tag{3.1}$$

where σ is a nonlinear activation function, and $\boldsymbol{b}_1 \in \mathbb{R}^N$ is a bias term for the first weight matrix. Common activation functions used in the hidden layers of NNs are the hyperbolic tangent function (Tanh) or the rectified linear unit function (ReLU) function. The outputs of the hidden layer are passed through the second weight matrix $\boldsymbol{W}_2$, which is used to compute the outputs $\boldsymbol{z}$:

$$\boldsymbol{z} = \boldsymbol{W}_2 \boldsymbol{h} + \boldsymbol{b}_2 \tag{3.2}$$

where $\boldsymbol{b}_2$ is a bias term for the second weight matrix. The dimensions of $\boldsymbol{W}_2$, $\boldsymbol{b}_2$, and consequently $\boldsymbol{z}$ depend on the application of the NN and the number of target values. A final activation function is applied in the output layer to generate the estimated target values: $\hat{\boldsymbol{y}} = f(\boldsymbol{z})$. For binary classification, a sigmoid activation function is applied on the sum of the estimated outputs, generating $\hat{y}$ ranging from

0 to 1:

$$\hat{y} = \frac{1}{1 + e^{-z}} \tag{3.3}$$

In multiclass classification, a softmax activation function is applied that normalizes the output into a probability distribution consisting of K number of probabilities proportional to the exponents of the input numbers:

$$\hat{\boldsymbol{y}} = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.4}$$

for $i = 1, ..., K$ and $\boldsymbol{z} = (z_1, ..., z_K) \in \mathbb{R}^K$, where K is the number of target values. In regression, $\hat{y}$ is simply a linear combination of z. Other types of activation functions are possible, but in supervised learning these are by far the most common types.

3.2.2 Loss and learning

The aim of training a NN is to tune the weight matrices connecting each layer of neurons such that the network can accurately model the relationship between the inputs and the target values. A suitable loss function $L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is first applied to the difference between the estimated target vectors and the true target vectors that are used in the training set. To use a NN for classification, cross-entropy (log-loss) functions are commonly used, while for regression purposes other means of loss, such as the mean squared error (MSE), are applied.

The tuning of the parameters in the weight matrices is performed iteratively using gradient-based optimization algorithms, which refers to moving along an error gradient towards some minimum level of error of a defined objective function J. Using the whole training set to update the weight parameters is commonly referred to as batch gradient descent, in which the objective function can be defined as:

$$J = \frac{1}{S} \sum_{i=1}^{S} L(\hat{\boldsymbol{y}}, \boldsymbol{y}) \tag{3.5}$$

where S is the total number of samples in the training set. In batch gradient descent, the objective function is defined as the average loss over the *whole* training set. However, batch gradient descent is often computationally inefficient and requires significant computational memory when a network is trained on large sets of data. To overcome this issue, training methods that use smaller subsets of the total training set are often used. In *mini-batch gradient descent*, a smaller batch is used to update the parameters. When all batches in the whole training set have been used to update the weight parameters, it is said that the network has been trained for one *epoch*.

The gradient of the objective function is computed with respect to the weight parameters and the bias vectors. A method called *backpropagation*, where the computed gradient is passed back along each hidden layer, is used to update the weight parameters connecting each layer. The weight parameters are updated iteratively in small steps, and it is common to train a NN for several epochs to reach good performance on the training set.

3.2.3 Overfitting and validation

The aim in training a NN is not to fit the data on the *specific* training set, but rather to generalize the training so that the network can provide good mapping on yet unseen data. A common problem when training NNs is overfitting, where the network is trained too closely on the specific training set, with the result that it fails to fit additional data or predict future observations accurately.

To ensure that the network is not overfitting on the training data, the performance of the network on a separate validation set is commonly monitored. In case the validation loss starts to increase, the training of the network can either be stopped, or various regularization techniques can be applied to avoid overfitting. Many regularization techniques are based on limiting the capacity of the networks, by adding different penalties to the loss function. A very common and simple regularization function is the $L2$ parameter norm penalty, also known as weight decay. This technique penalizes large numbers of the weight parameters by adding a regularization term $\Omega = \lambda \frac{1}{2m} \sum_{i=1}^{k} ||\boldsymbol{w}||_i^2$ to the objective function. The value of λ controls how strict the regularization is, and m is the number of training samples. A too small value of λ may result in an insufficient regularization with overfitting as a result, while a too large value may result in *underfitting* or high variance, with poor performance on both the training set and the validation set. Another popular regularization technique is called *dropout*, where a certain percentage of the connections between each layer are randomly masked (or "dropped"). This technique ensures that the network does not rely too heavily on certain connections.

3.2.4 Hyperparameters and network depth

Hyperparameters are parameters that control the training of the NN, and include parameters such as the learning rate of the optimization algorithm, the number of epochs the network is trained, or the structure and number of neurons in the hidden layers. The hyperparameters play a crucial part in the performance of the NN and a hyperparameter search is generally conducted where different combinations of hyperparameters are assessed and tested [93].

Deeper network architectures, with several hidden layers ordered in chain-like structures, are often able to use fewer hidden units per layer, and thus far fewer parameters, compared to more shallow networks used in the same applications. Deeper architectures are also found to better generalize the performance on unseen data. However, the drawback of deeper architectures is that these also tend to be harder to optimize [93].

3.2.5 Recurrent neural networks

In many applications, the capability to provide accurate classifications or estimations at a specific time t would be even more accurate if it was possible to account for previous or historic data. RNN provide this capability by using sequence-based networks that are adapted for processing sequences of input data, capable of utilizing both current and past data [95]. The concept of RNNs is not new but has received an increasing amount of attention in recent years. In power systems, RNNs have

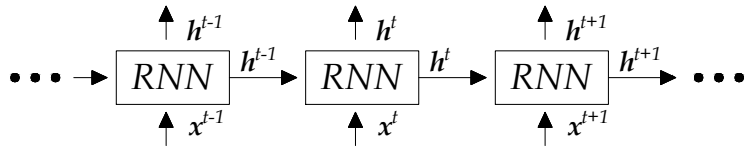

Figure 3.2: A sequence of consecutive RNN blocks.

previously been used in applications ranging from market forecasting [15], transient stability assessment [16], and in power quality assessments [17].

A typical RNN sequence is illustrated in Fig. 3.2, where each block has a directed connection to the following block in the sequence. The main difference from a feedforward NN is that the output of each block depends not only on the input vector at the current time step but also on the output from previous blocks in the sequence. If the block is the first one in the sequence, the inputs are made up solely by the input vector. Depending on the architecture of the RNN, the output vector of each block can be used for both classification at the current time step, and/or used as an input to the following block. Each block contains interior connections, weights, and non-linear activation functions. The block's complexity ranges from those of simple RNNs using conventional neuron layers, to more advanced structures such as the LSTM. RNNs are generally trained using an approach called *backpropagation-through-time*. Backpropagation-through-time is similar to the backpropagation used in training a conventional NN, with the difference that the error gradients also have to be propagated back in time through the RNN sequence [96].

3.2.6 Long short-term memory

The standard implementation of RNN has difficulties in capturing long-term dependencies of events that are significantly separated in time. When the error signal is passed back through many RNN sequences, it tends to either blow up or vanish [97]. This is effect is generally recognized as *vanishing* or *exploding gradients*. In an LSTM network, the information and the error gradients can be propagated through time within an internal state memory cell, making the network capable of memorizing features of significance over time [98].

A typical LSTM block is illustrated in Fig 3.3. The state memory cell, illustrated by the light grey area, is controlled by nonlinear gating units that regulate the flow in and out of the cell [99]. Following [98] and [99], the forward operation of an LSTM block is summarized below. It should be noted that each block consists of a number of hidden LSTM cells. Vector notation is used, meaning that, for instance, the hidden state vector $\boldsymbol{h}^t$ is not the output of a single LSTM cell at time t, but the output of a vector of N LSTM cells. The operation of an LSTM block at a time t may then be summarized by:

$$\boldsymbol{f}^t = \sigma\left(\boldsymbol{W}_f\boldsymbol{x}^t + \boldsymbol{U}_f\boldsymbol{h}^{t-1} + \boldsymbol{b}_f\right) \tag{3.6}$$

$$\boldsymbol{i}^t = \sigma\left(\boldsymbol{W}_i\boldsymbol{x}^t + \boldsymbol{U}_i\boldsymbol{h}^{t-1} + \boldsymbol{b}_i\right) \tag{3.7}$$

$$\tilde{\boldsymbol{c}}^t = \tanh\left(\boldsymbol{W}_c\boldsymbol{x}^t + \boldsymbol{U}_c\boldsymbol{h}^{t-1} + \boldsymbol{b}_c\right) \tag{3.8}$$

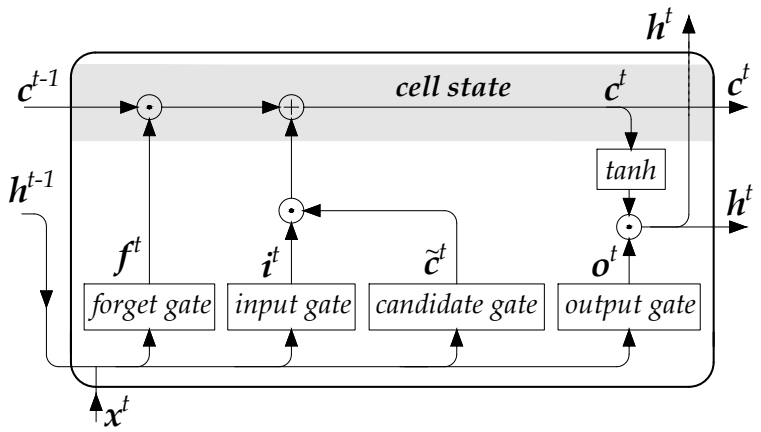

Figure 3.3: Detailed schematics of an LSTM block.

$$\boldsymbol{c}^t = \boldsymbol{f}^t \odot \boldsymbol{c}^{t-1} + \boldsymbol{i}^t \odot \tilde{\boldsymbol{c}}^t \tag{3.9}$$

$$\boldsymbol{o}^t = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}^t + \boldsymbol{U}_o \boldsymbol{h}^{t-1} + \boldsymbol{b}_o\right) \tag{3.10}$$

$$\boldsymbol{h}^t = \boldsymbol{o}^t \odot \tanh(\boldsymbol{c}^t), \tag{3.11}$$

where element-wise multiplication is denoted by $\odot$, σ is the logistic sigmoid function, tanh is the hyperbolic tangent function, and with the following variables:

- $\boldsymbol{x}^t \in \mathbb{R}^M$: input vector to an LSTM block
- $\boldsymbol{h}^t$,$\boldsymbol{h}^{t-1} \in \mathbb{R}^N$: output vector at time t respectively t-1
- $\boldsymbol{f}^t \in \mathbb{R}^N$: activation vector of the forget gate
- $\boldsymbol{i}^t \in \mathbb{R}^N$: activation vector of the input gate
- $\tilde{\boldsymbol{c}}^t \in \mathbb{R}^N$: vector of the candidate gate
- $\boldsymbol{c}^t \in \mathbb{R}^N$: cell state memory vector
- $\boldsymbol{i}^t \in \mathbb{R}^N$: activation vector of the output gate

where $\boldsymbol{W}$, $\boldsymbol{U}$, and $\boldsymbol{b}$ represent the weight matrices and bias vectors for each gate. The superscripts M and N refer to the number of inputs and hidden LSTM cells in each LSTM block, respectively.

The information stored in the state memory cell is regulated by the operation of the different gates, as illustrated in Fig. 3.3. By the operation of (3.6), the forget gate controls what information should be stored from the previous memory cell state, and what can be discarded as irrelevant. The input gate and candidate gate control and update the memory cell state with new information by the operation of (3.7)–(3.8). In (3.9), the state memory cell is first updated by an element-wise multiplication of the previous cell state memory vector and the resulting vector of the forget gate. Then, the state memory cell is updated with new values provided by an element-wise

multiplication of the resulting vectors from the input gate and the candidate gate. Equations (3.10)–(3.11) show how the hidden state is updated by the operation of the output gate, modulated by the updated cell state memory vector.

An LSTM *network* may then be constructed by creating a sequence of several LSTM blocks. A partition of an LSTM sequence is illustrated in Fig. 3.4, where each block has a directed connection to the following block in the sequence. If the block is the first one in the sequence, the past system state is initialized with a preset value. For a deep LSTM network, with several stacked layers, the inputs to the deeper layers consist of the hidden states of LSTM blocks of previous layers. The cell state memory is only passed along the time sequence between LSTM blocks of the same layer. Typically, for classification purposes, an output vector $\boldsymbol{y}$ is generated by applying a nonlinear function of the hidden state implemented by a separate feedforward NN. Depending on the application of the network, output vectors may be computed for a single, or several, LSTM block's hidden states.

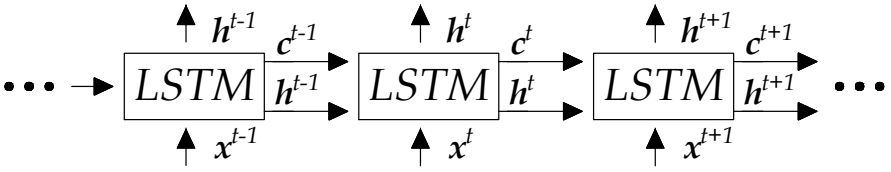

Figure 3.4: An LSTM sequence with a directed connection between the blocks.

3.3 Markov decision processes

An MDP is a mathematical framework used for modeling control problems. It is generally used for time-discrete sequential problems where the outcome of the actions is partly stochastic and partly influenced by the decision-maker [100]. Mathematically, an MDP is defined by the tuple $\mathcal{S}$, $\mathcal{A}$, $\mathcal{T}$, $\mathcal{R}$ and γ:

- $\mathcal{S}$: a set of states called the *state space*.
- $\mathcal{A}$: a set of actions called the *action space*.
- $\mathcal{T}$: the *state transition dynamics distribution* satisfying the Markov property.
- $\mathcal{R}$: a *reward function* $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ which depends on the current state and the action taken at that state.
- γ: a discount factor $\gamma \in [0, 1]$.

The agent-environment interface used in an MDP is illustrated in Fig. 3.5. The decision-maker is generally referred to as the "*agent*" and the system it operates and interacts with is called the "*environment*". The agent monitors the state of the environment and selects actions that affect it, in turn giving rise to new states. The actions and the environment also give rise to rewards, which the agent generally seeks to maximize over time through its choice of actions [101]. The terms of *agent*,

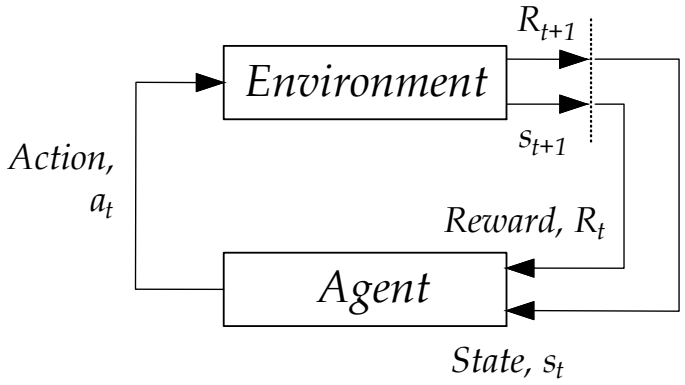

Figure 3.5: The agent-environment scheme used in an MDP.

environment, and *action* correspond to the terms used in control theory of *controller*, *controlled system* (or plant), and *control signal*, respectively.

The agent and the environment interact in a sequence of time steps ($t = 0, 1, 2, 3..$). At each time step t, the agent receives a representation of the environment's state, $s_t \in \mathcal{S}$, and by following a defined policy selects an action $a_t \in \mathcal{A}$. For the next time step, as a consequence of the taken action, the agent receives a numerical reward $R_t \in \mathcal{R}$, and finds itself in a new state s_{t+1}. The continuous interaction between the agent and the environment gives rise to a sequence of states, actions, and reward pairs. If there exists a natural notion of a final time step in an MDP, it is referred to as an *episodic* task. If the MDP does not break naturally into identifiable episodes, it is referred to as a *continuous* task.

The dynamics of the MDP are modelled by the state transition dynamics distribution $p(s_{t+1} = s'|s_t = s, a_t = a)$, which models the probability of a transition from state s to the next state s' when taking action a at time t. An important property in an MDP is the *Markov property*: $p(s_{t+1}|s_1, a_1, ..., s_t, a_t) = p(s_{t+1}|s_t, a_t)$, which, if satisfied, states that the state must include information about *all* aspects of past agent-environment interactions. The expected reward for any state-action pair can be modelled by the reward function $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R} : r(s, a) = \mathbb{E}[R_t|s_{t-1} = s, a_{t-1} = a]$. The reward provides a quantitative measure used to improve the policy of the agent. Actions and policies are said to be optimal if they lead to the maximum cumulative reward over time.

3.4 Reinforcement learning

RL relies on an MDP framework for modeling sequential decision-making problems, but different classes of RL algorithms make different mathematical assumptions. The main task in RL is always for the agent to learn an optimal policy π, which allows it to maximize the expected future reward. The policy maps states to actions and can be defined to be either stochastic or deterministic depending on the design of the RL algorithm. A stochastic policy specifies probabilities $\pi(a|s)$ for taking each action in each state, which may be mathematically formulated by $\pi_\theta(a|s) : \mathcal{S} \longrightarrow \mathcal{P}(\mathcal{A})$ where $\mathcal{P}(\mathcal{A})$ is the set of probability measures on $\mathcal{A}$ and $\pi(a|s)$ is the conditional probability density of taking action a in state s associated with the policy. Conversely, for a

deterministic policy, the actions taken by the policy are always fixed for each state in the environment: $\pi(s) : \mathcal{S} \longrightarrow (\mathcal{A})$. RL methods may also be divided into methods that use a model-based or a model-free approach. Model-based RL explicitly models the state transition dynamics distribution $\mathcal{T}$ and the reward function $\mathcal{R}$, which are built from dynamic programming methods. Model-free RL techniques do not model the state transition dynamics distribution or the reward function and instead rely exclusively on samples generated from the environment.

In general, the RL agent seeks to maximize the expected return, where the return, denoted G_t, is defined as some specific function of the reward sequence. In general, future rewards are discounted and the return can be formulated (for episodic tasks) as:

$$G_t^\gamma = \sum_{k=t}^{T} \gamma^{k-t} R(s_k, a_k) \tag{3.12}$$

where the discount term γ is ranging from 0 to 1 depending on how much future returns should be discounted. The majority of RL algorithms attempt to learn an approximation of the so-called value function, which represents the expected return if one starts in a particular state, and then acts according to the learned policy after that. The value function can be mathematically stated as:

$$V^\pi(s) = \mathbb{E}\left[G_t^\gamma | s_t = s; \pi\right] \tag{3.13}$$

Another often used function used in training RL agents is the action-value function, which is defined as the expected total return in state s when taking action a and *then* following the policy:

$$Q^\pi(s, a) = \mathbb{E}\left[G_t^\gamma | s_t = s, a_t = a; \pi\right] \tag{3.14}$$

There exist many RL algorithms that learn to develop optimal policies in different ways, such as Q-learning [102], SARSA [103], or policy-based methods such as REINFORCE [104] and natural gradients [105]. It is often the environment and the design of the control that governs which algorithm is best suited for the task at hand.

3.4.1 Value-based methods versus policy-based methods

RL algorithms that attempt to learn the action-value function are commonly referred to as value-based methods, which include popular RL algorithms such as Q-learning or SARSA. For instance, in Q-learning, the agent tries to learn the action-value function for all possible states and actions that the agent might operate in. Once it has learned the action-value function (or Q-function), it can act greedily and choose the actions that have the highest expected reward. To encourage exploration, random actions are generally selected with a certain (diminishing) probability. An advantage of many value-based methods is that they can learn on data sampled *off-policy*. This means that the agent can continue to learn from data that is sampled with an old policy. Thus, old samples do not necessarily need to be discarded after the agent has been trained on them but can be stored and used to improve the

training of the agent later on. This typically improves the sample efficiency of the algorithm, making value-based methods suitable for environments where training data are more limited.

The basic formulation of policy-based methods does not explicitly require information about the action-value function, but instead directly learns the policy. This is advantageous for certain environments, where the action-value function may be complex to learn, but the policy itself might be relatively simple to learn from data. Most policy-based methods learn *on-policy*, which means that once the policy has been updated on a batch of samples, the data needs to be discarded and new sampled, causing such methods to be typically less sample efficient. An advantage of many policy-based methods is that they can formulate stochastic policies. Stochastic policies have many advantages, which include natural exploration due to the stochastic nature of the policy, efficient handling of continuous action spaces (and possibly hybrid action spaces), and avoidance of convergence issues that are common in value-based methods. A drawback of policy-based methods is that they tend to have relatively slow convergence and have high variance during learning.

3.4.2 Exploration and exploitation

One of the main challenges in RL, which is not present in supervised or unsupervised learning schemes, is the trade-off between exploring new actions and exploiting already learned knowledge. An RL agent should prefer actions that it has tried in the past and found to be effective, but to discover such actions, it has to try actions that have not been selected before. Thus, the agent must exploit its knowledge in order to obtain a reward, but it must also explore to make better action selections in the future [101]. Neither exploration nor exploitation can be pursued by the agent exclusively without failing at the task of learning an optimal policy.

3.4.3 Deep reinforcement learning

The basic formulation of RL is only applicable on low-dimensional environments and with a discrete state space. For such environments, tabular methods, where the value function and the action-value function can be learned directly for each possible state, are effective. However, for most actual control applications, the number of feature dimensions is large and even infinite for environments with continuous state spaces.

To handle these issues, it is common to use some form of function approximation that maps the state space to either a value function or a policy. One of the most popular choices, which has proven effective in solving many advanced control tasks, has been to combine RL with deep learning, then referred to as DRL. While deep learning generally refers to methods that use several hidden NN layers, DRL is more synonymous with a RL algorithm that uses any architecture based on NNs to approximate the state space. The state-space approximation is used differently depending on the DRL algorithm at hand. For value-based methods, such as deep Q-learning, it is used to map states to action-values, which are then used to define a policy. In policy gradient methods, the deep learning approximation instead maps

the state directly to the policy. In this thesis, mainly policy gradient methods are used and examples of DRL architecture used for such applications are presented in Chapter 8 and 9.

3.4.4 Policy gradient and actor-critic methods

Policy gradient methods are a class of model-free RL algorithms that learn a parameterized policy π_θ that can select actions without requiring a value function [101]. These methods seek to maximize a defined objective function $J(\theta)$ parametrized by θ, so their updates approximate gradient ascent:

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)} \tag{3.15}$$

where $\widehat{\nabla J(\theta_t)}$ represents a stochastic estimate of the gradient of the objective function with respect to θ_t [101] and α is the learning rate used in the optimization. One of the most commonly used gradient estimator in RL has the following form:

$$\widehat{\nabla J(\theta_t)} = \hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(a_t|s_t) \hat{A}^\pi_\phi(s_t, a_t) \right] \tag{3.16}$$

where the expectation $\hat{\mathbb{E}}_t [...]$ indicates an empirical average over a batch of samples drawn from the MDP, $\nabla_\theta \log \pi_\theta(a_t|s_t)$ is the gradient of the parametrized policy, and $\hat{A}^\pi_\phi$ is an estimator of the advantage function at time step t. The estimated advantage function can be written as:

$$\hat{A}_t = \hat{Q}^\pi(s_t, a_t) - \hat{V}^\pi_\phi(s_t) \tag{3.17}$$

where $\hat{V}^\pi_\phi$ and $\hat{Q}^\pi$ are estimates of the value function and the action-value function, respectively. The action-value function $\hat{Q}^\pi$ can be estimated in different ways. A straightforward method is to use the sample return G^γ_t from (3.12) which provides an unbiased estimate of $\hat{Q}^\pi$. The drawback of using the sample return as an estimate of $\hat{Q}^\pi$, is that the estimate can have high variance, resulting in slow convergence of the algorithm. A method to achieve lower variance, at the cost of introducing some bias to the estimate, is to use methods based on temporal difference (TD). An alternative way of formulation the advantage function based on a TD formulation is:

$$\begin{aligned} \hat{A}_t &= \hat{Q}^\pi_\phi(a_t, s_t) - \hat{V}^\pi_\phi(s_t) = \\ &= \underbrace{R_t + \hat{V}^\pi_\phi(s_{t+1}) - \hat{V}^\pi_\phi(s_t)}_{\delta\text{-error}} \end{aligned} \tag{3.18}$$

where the action-value function is recursively defined as the sum of the immediate reward R_t after taking action a_t in state s_t and the estimated discounted value of the subsequent state $\gamma\hat{V}^\pi_\phi(s_{t+1})$. Since $\hat{V}^\pi_\phi(s_{t+1})$ is an estimated approximation, this formulation will introduce some bias to the advantage estimate, but it will simultaneously reduce the variance. When this recursive expression is used to form the advantage estimate, it is commonly referred to as the δ-error (or TD error). The value function used to compute the advantage function in (3.17) and (3.18) is generally unknown and has to be learned simultaneously as the policy. If the value

function is learned in addition to the policy and the δ-error is used to approximate the advantage function, the algorithm is usually referred to have an Actor-Critic architecture. The policy π_θ is estimated by the *actor* while the value function $\hat{V}_\phi^\pi$ is estimated by the *critic*.

The value function can be learned simultaneously as the policy, commonly by minimizing a new cost function $L(\phi)$, based on the mean-squared error (or some other loss function) between the true value function $V^\pi(s_t)$ and its approximation $\hat{V}_\phi^\pi(s_t)$:

$$L(\phi) = \mathbb{E}_\pi \left[\left(V^\pi(s_t) - \hat{V}_\phi^\pi(s_t) \right)^2 \right] \tag{3.19}$$

In the case the δ-error is used to form the advantage function, the value function can be learned by minimizing the mean-squared error (or some other loss function) of the sampled and computed δ-errors.

3.4.5 The proximal policy optimization algorithm

The basic formulation of policy gradient methods suffers from two main weaknesses: 1) sample inefficiency and 2) robustness issues during training. Policy gradient methods are in general on-policy methods, which means that they always need to sample transitions following the *current* policy. Once the policy has been updated by training on the current batch of transitions, the samples need to be discarded and new transitions sampled. This causes on-policy methods to be less sample efficient compared to off-policy methods (such as Q-learning) where the state-value function can still continue to be learned from old transition data. A tempting solution is to either run multiple steps of optimization on the same batch of data or to use a large learning rate to improve the learning speed. However, doing so is not well-justified, and empirically it often leads to destructively large policy updates [106].

The proximal policy optimization (PPO) algorithm, first presented in [106], presents a solution to these problems. In this paper, we use the "clipped" version of the PPO algorithm, where the objective function is defined as:

$$J^{clip}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{3.20}$$

where r_t is a probability ratio given by:

$$r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{3.21}$$

and θ_{old} refers to the vector of policy parameters used in sampling the transitions and thus before any update of the policy parameters. The clipped objective function ensures that one does move *too* far away from the current policy, which allows one to run multiple epochs of gradient ascent on the samples without causing destructively large policy updates. The r_t-ratio is always equal to 1 before the first epoch, when current policy $\pi_\theta(a_t|s_t)$ is the same as was used to sample the transitions $\pi_{\theta_{\text{old}}}(a_t|s_t)$. The intuition of (3.20) is best explained in two parts; when the advantage function is positive and negative, respectively. The two objective functions are illustrated in Fig. 3.6 for both a positive and a negative advantage function. The red circle

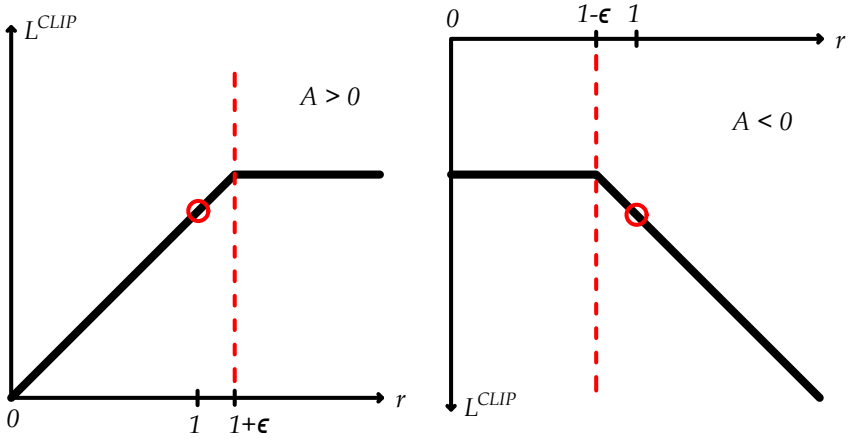

Figure 3.6: A single time step of the clipped surrogate function as a function of the probability ratio r_t, for $A > 0$ and $A < 0$, respectively. The red circle indicates the starting point of the optimization for the first epoch, i.e. r_t=1.

indicates the starting point of the optimization, whereas the dotted line indicates the limit to how much the ratio r_t can change before it is limited.

- *Positive advantage function*: a positive advantage function for a given state-action pair indicates that the taken action yielded a higher reward than was anticipated by the critic network. Thus, the objective function will increase if that action becomes more likely in that specific state; that is, if $\pi_\theta(a|s)$ increases, or equivalently if the r_t ratio increases above 1. However, the min-term puts a limit to how much the probability can increase, and whenever $\pi_\theta(a|s) > (1+\epsilon)\pi_{\theta_{old}}(a|s)$ the min term activates and the gradient of the objective function is reduced to zero.

- *Negative advantage function*: a negative advantage function for a given state-action pair indicates that the taken action yielded a lower reward than was anticipated by the critic network. The objective function will increase if that action becomes *less* likely when in that specific state; that is, if $\pi_\theta(a|s)$ decreases, or equivalently if the r_t ratio decreases below 1. Again, the min-term puts a limit to how much the probability can decrease, and whenever $\pi_\theta(a|s) < (1-\epsilon)\pi_{\theta_{old}}(a|s)$ the min-term activates and the gradient of the objective function is reduced to zero.

3.4.6 Adaptations for continuous-discrete control

To handle a hybrid actions space of both continuous and discrete actions, an implementation similar to the one introduced in [107] can be used. A hybrid policy $\pi_\theta(\mathbf{a}|s)$ is defined as a state-dependent distribution that jointly models discrete and continuous random variables. Independence between action dimensions, denoted by

a^i, is assumed and the hybrid policy can be written as:

$$\begin{aligned} \pi_\theta(\mathbf{a}|s) &= \pi_\theta^c(\mathbf{a}^\mathcal{C}|s)\pi_\theta^d(\mathbf{a}^\mathcal{D}|s) = \\ &= \prod_{a^i \in \mathbf{a}^\mathcal{C}} \pi_\theta^c(a^i|s) \prod_{a^i \in \mathbf{a}^\mathcal{D}} \pi_\theta^d(a^i|s) \end{aligned} \tag{3.22}$$

where $\mathbf{a}^\mathcal{C}$ and $\mathbf{a}^\mathcal{D}$ are subsets of action dimensions with continuous and discrete values respectively (where $\mathcal{C}$ and $\mathcal{D}$ represent continuous respectively discrete action spaces), and $\mathbf{a}$ is a vector of both discrete and continuous actions. Each component of the continuous policy π_θ^c can be represented as a normal distribution:

$$\pi_\theta^c(a^i|s) = \mathcal{N}\left(\mu_{i,\theta}(s), \sigma_{i,\theta}^2(s)\right) \tag{3.23}$$

where $\mu_{i,\theta}$ and $\sigma_{i,\theta}$ are the parametrized mean value and standard deviation of each continuous action dimension. Each component of the discrete policy π_θ^d can be described as a Bernoulli distribution parameterized by state-dependent probabilities $P_{i,\theta}(s)$:

$$\pi_\theta^d(a^i|s) = Bernoulli^i\left(P_\theta^i(s)\right) \tag{3.24}$$

where θ are the parameters of the policy components that we want to optimize. It should be noted that other types of distributions, such as a Beta distribution for a continuous action space, or a Softmax distribution for multi-class discrete actions are also possible options.

Chapter 4

Description of a real-time dynamic stability assessment and control tool

In this chapter, the overall functionality of the proposed real-time dynamic stability assessment and control tool is presented. Furthermore, the test system used in evaluating the developed monitor and control methods is presented and its general characteristics are briefly discussed. The real-time dynamic stability assessment and control tool is based on four different developed methods presented in subsequent chapters. The first method is developed for preventive monitoring, where a method for fast security margin estimation is developed. The second method is developed for monitoring emergency events, where a method for voltage instability prediction based on a long short-term memory network is developed. The third method is developed for preventive control, where a method based on deep reinforcement learning is used to control and ensure a sufficient security margin. The final method is developed for emergency control, where a method based on deep reinforcement learning is used to stabilize the system in case of emergency events and larger disturbances. In Chapters 6 - 9, each of the developed methods will be presented in more depth.

4.1 Overview of the Real-time Dynamic Stability Assessment and Control Tool (RDS-ACT)

This thesis aims to develop a RDS-ACT that can support system operators with better information on the current stability and security margins of their power systems. In case of too small margins towards the security boundary, or in the case of larger disturbances, the tool can also suggest optimized control actions to steer the system into a secure and/or stable operation again. An overview of the proposed RDS-ACT in the setting of a monitoring and control system in a power system is presented in Fig. 4.1. The functionality of the RDS-ACT is divided to handle the two complementary tasks, namely: i) *emergency* monitoring and control and ii) *preventive* monitoring and control.

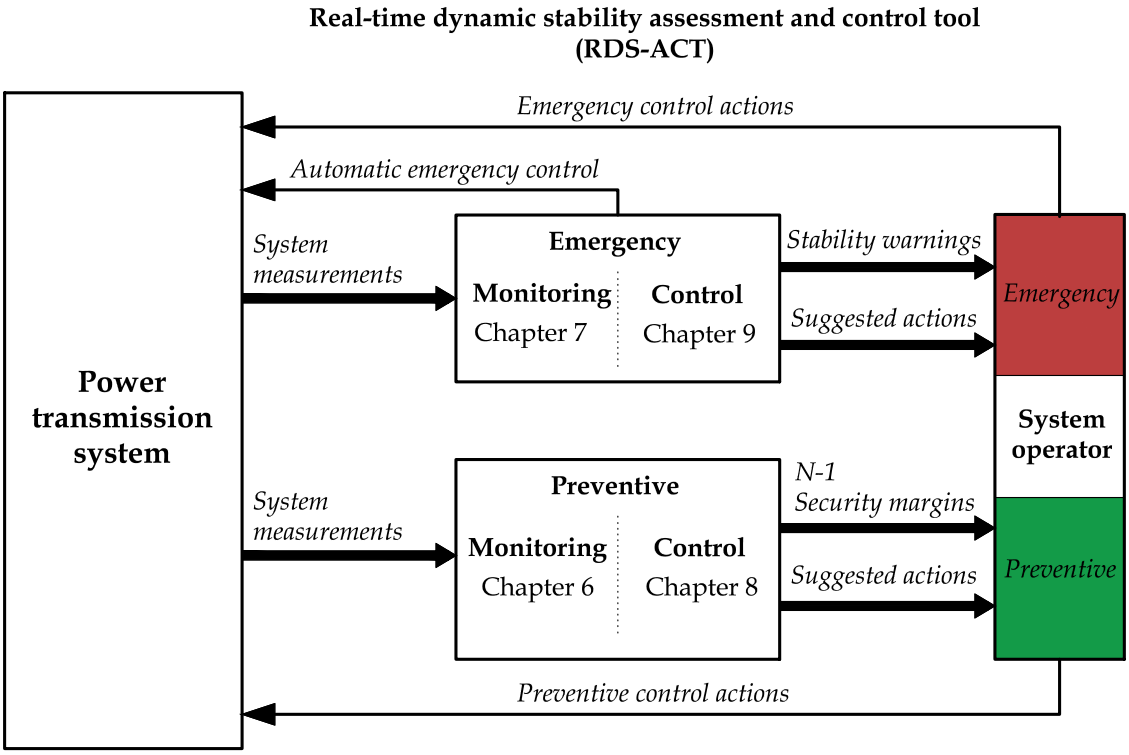

Figure 4.1: An overview of the proposed RDS-ACT and the signals and control actions in a monitoring system.

All developed methods are intended to use real-time system measurements sampled directly from a power transmission system. The requirements on the system measurements, such as measurement update rates or the availability of phasor measurements, are discussed in each of the chapters covering the functionality of the methods. To assure that errors and missing values are filtered out, measurements are typically assumed to always be preceded by a state estimator. Circuit breaker statuses are assumed to always be known. In this work, simulated measurement data are used exclusively.

4.1.1 Preventive monitoring and control

The proposed method for preventive *monitoring* is used to provide fast and robust estimations of the DVSM. The method uses a regression-based NN to provide a qualified estimate of the actual DVSM. In addition, a second NN is used to provide a classification of which contingency will be dimensioning for the system. The estimates from the NNs are then used in a method called dual binary search, which is used to validate the actual DVSM using time-domain simulations. The ML-based approach is thus only proposed to support the estimation of the DVSM, while the actual DVSM is always validated through actual time-domain simulations.

The proposed method for preventive *control* is based on using a DRL framework which in real-time can suggest suitable control actions whenever the security mar-

gins are below a predefined threshold. The developed DRL control uses a hybrid control scheme that is capable of simultaneously adjusting both discrete action variables (switching of shunt reactive power devices) and continuous action variables (controlling power flows by load and generation rescheduling in stressed system areas). The system operator can choose to automatically initiate these actions in the system or assess the suggested actions and then take the actions manually.

4.1.2 Emergency monitoring and control

The proposed method for emergency monitoring is based on using an RNN with LSTM for VIP, from now on abbreviated into LSTM-VIP. The method is designed to take current and historic measurements to assess whether the *current* state may cause voltage instability several minutes into the future. As time progresses and if new events occur in the system, the network updates the assessment continuously. Stability warnings are then passed to the system operator, which can initiate emergency control actions. The network is also adapted to be able to indicate where in the system instability emerges allowing more cost-effective countermeasures.

The proposed method for emergency *control* is based on using a DRL framework aimed to mitigate long-term voltage instability. Once trained, the DRL control can continuously assess the system stability and suggest fast and efficient control actions to system operators in case of voltage instability. The DRL control is trained to use system services from DR and ESS as a more efficient and flexible alternative to stabilize the system, compared to e.g., conventional load shedding. As time is critical in many emergency events, the actions suggested by the DRL controller may be required to be activated automatically in the system to avoid a system collapse. However, in long-term voltage instability events, where the time between the disturbance occurs and a possible unstable state may develop is relatively long, the system operators may have time to manually assess the suggested actions by the DRL emergency controller before activating them.

4.2 Test system

The methods are tested on an updated version of the classic Nordic32 test system, presented in [108]. The system is fictitious but captures similar voltage dynamics as in the Swedish and the Nordic power system. A one-line diagram of the test system is presented in Fig. 4.2. The system is divided into four different regions:

- *"North"*: mainly consists of hydro generation and some smaller loads.
- *"Central"*: the largest load center with significant generation of thermal power generation.
- *"Eq"*: an equivalent of an external system connected to the "North".
- *"South*: an area with thermal generation which is loosely connected to the "Central" area.

Table 4.1: Active power generation and load for the Nordic 32 test system.

Area	**Generated power (MW)**	**Load (MW)**
"North"	4,628.5	1,180.0
"Central"	2,850.0	6,190.0
"South"	1,590.0	1,390.0
"Eq."	2,437.4	2,300.0
Total	11,505.9	11,060.0

The modified Nordic32 test system has long transmission lines of 400 kV and 220 kV nominal voltage. The test system also includes a representation of regional systems operating at 130 kV. Table 4.1 provides the active power generation and load in each area and for the whole test system.

The system is heavily loaded with large power transfers mainly between the areas "North" and "Central". The transferable power is limited by the reactive power capabilities of generators in both of these areas. A disturbance in the transmission capacity connecting the "North" and the "Central" areas is critical for the stability of the system. A reduced transmission capacity, caused by, for instance, a disconnected transmission line, would increase the strain on the remaining lines in the system. The increased current in the remaining transmission lines would increase the reactive power losses in the system, which would cause lower system voltages. Voltage-dependent loads are restored through the actions of LTCs. The load restoration has an adverse effect on the voltage stability of the system as the restored loads will cause increased stress on the remaining lines in the system and may cause the system to deteriorate further. Reactive power limits of generators are enforced by OELs which further limits the capability to maintain nominal voltages in the system.

For the developed modified Nordic32 system, two developed operating points were presented in [108]. "Operating point A" is an unsecure operating point and an outage of either a larger thermal generating unit in the "Central" area or a disconnection of a transmission line connecting the "North" and "Central" areas may cause instability. "Operating point B" is a secure operating point that should be able to withstand the loss of any single component. Both the two different operating points are used as starting values in different chapters of the thesis. All time-domain simulations have been performed using the power system simulation software PSS®E with its built-in dynamical models.

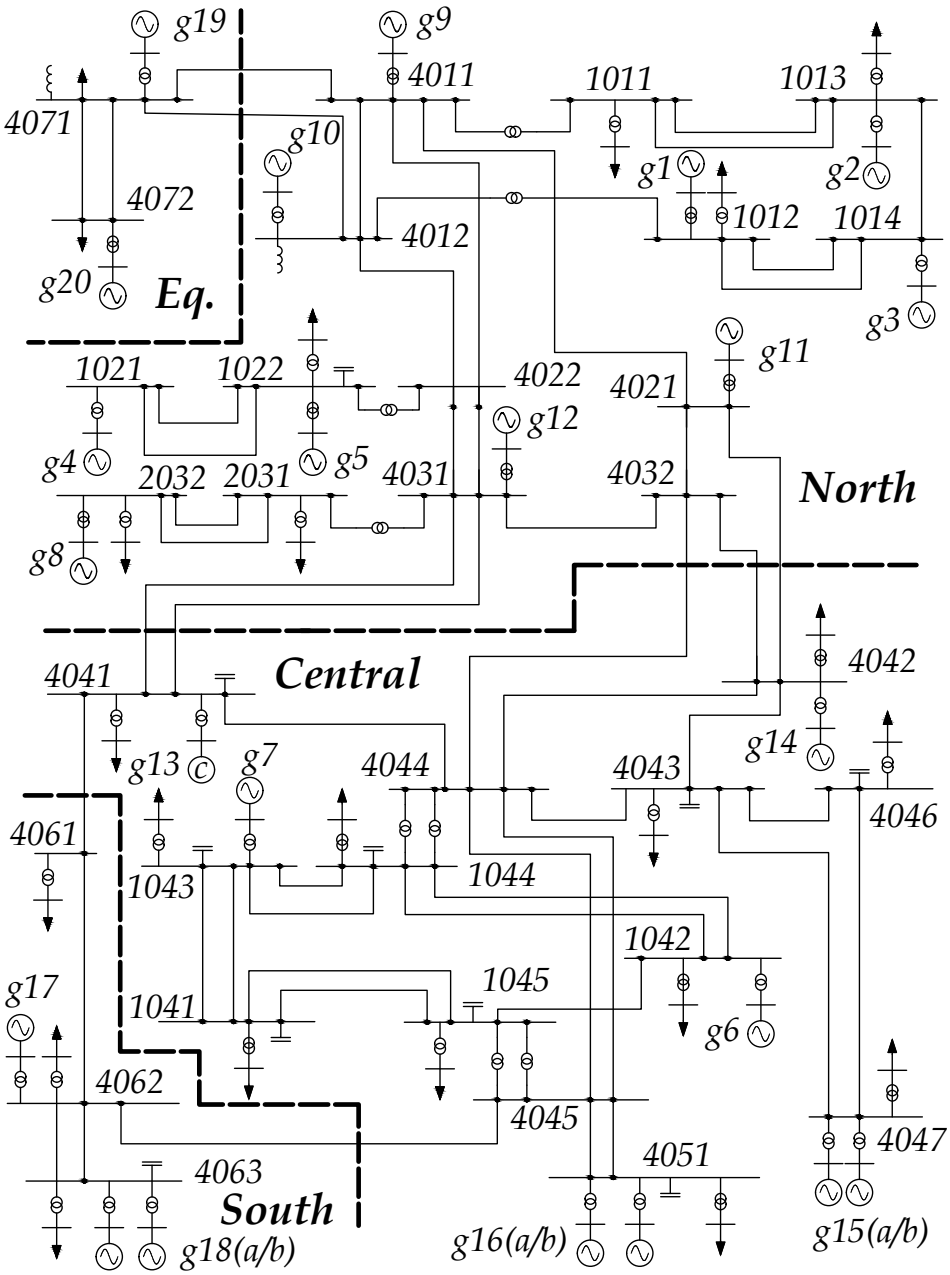

Figure 4.2: The Nordic32 test system used in testing the developed algorithms, adapted from [108].

Chapter 5

Comparison of dynamic and static security margins

In this chapter, the differences between static and dynamic security margins are established. First, the theoretical difference between the two methods is established and then visualized using the concept of transient P-V curves. Second, numerical comparisons are provided to further examine the differences between the different security margins under various load configurations and types of disturbances. The chapter is mainly based on the results established in Paper III.

5.1 Theoretical comparison of SOL and PCLL

In this section, the theoretical difference between the SOL and the conventional PCLL is established and then visualized using the concept of transient P-V curves (first introduced in [109]). A small test system's dynamic response following a disturbance is used in the analysis.

5.1.1 Small test system

The small 2-bus test system in Fig. 5.1 is used in the analysis. It consists of a controlled sending end voltage ($E\angle 0$), supplied by a voltage source. A complex load ($P + jQ$) is fed through a number of lines represented by inductances with the total reactance of X_t. A popular method in static voltage stability analysis is to use P-V curves, where the receiving end voltage is plotted with respect to an increasing level of active power transfer in the system. In the following figures in this section, P-V curves for the case when $E = 1.05$ pu, $X_t = 0.4$ pu, and a fully active power load are illustrated. An additional P-V curve is plotted in each figure for a N-1 case when one line has been disconnected (increasing X_t to 0.5 pu). Assuming lossless transmission, the curves are developed using the voltage equation for a two-bus system, given by [12]:

$$V = \sqrt{\frac{E^2}{2} - QX_t \pm \sqrt{\frac{E^4}{4} - X_t^2P^2 - X_tE^2Q}} \tag{5.1}$$

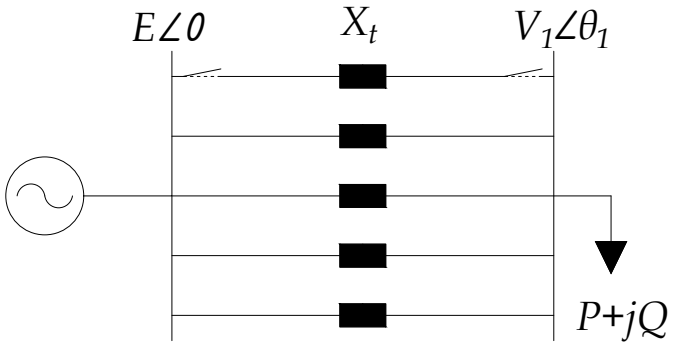

Figure 5.1: Simple 2-bus system used in the analysis.

where the upper part of each P-V curve corresponds to the solution of (5.1) with the plus sign, while the lower part of each curve corresponds to the solution with the minus sign.

The voltage instability mechanism is mainly driven by loads and the impact of load modeling in voltage stability analysis is imperative [12]. The power consumption of loads is affected by the system voltages and different load models are often used to characterize this relationship. One often used model is the *exponential load model*, which is given by:

$$P = zP_0 \left(\frac{V}{V_0}\right)^{\alpha} \tag{5.2}$$

$$Q = zQ_0 \left(\frac{V}{V_0}\right)^{\beta} \tag{5.3}$$

where P_0 and Q_0 are the active and reactive power consumed at voltage V equal to the reference voltage V_0 when $z = 1$. z is a dimensionless and independent variable indicating the actual loading of the system [12]. The voltage dependency is modeled by the α and β parameters, where $\alpha = \beta = \{0, 1, 2\}$ represents constant power (MVA), constant current, and constant impedance characteristics, respectively.

5.1.2 Estimating SOL and PCLL

The differences in how the PCLL and the SOL are estimated are illustrated in Fig. 5.2 using the developed P-V curves. The security margin is defined as the change in loading from an initial operating condition (OC) to the N-1 critical point. It should be noted that in real applications, the limit is often significantly smaller due to the other stopping criteria such as too low system voltages. Furthermore, in the following examples, only a single contingency (or disturbance) is considered when computing the PCLL and the SOL. However, in actual implementations, *all* larger contingencies should be assessed and the one that results in the *lowest* level of security margin is then generally used.

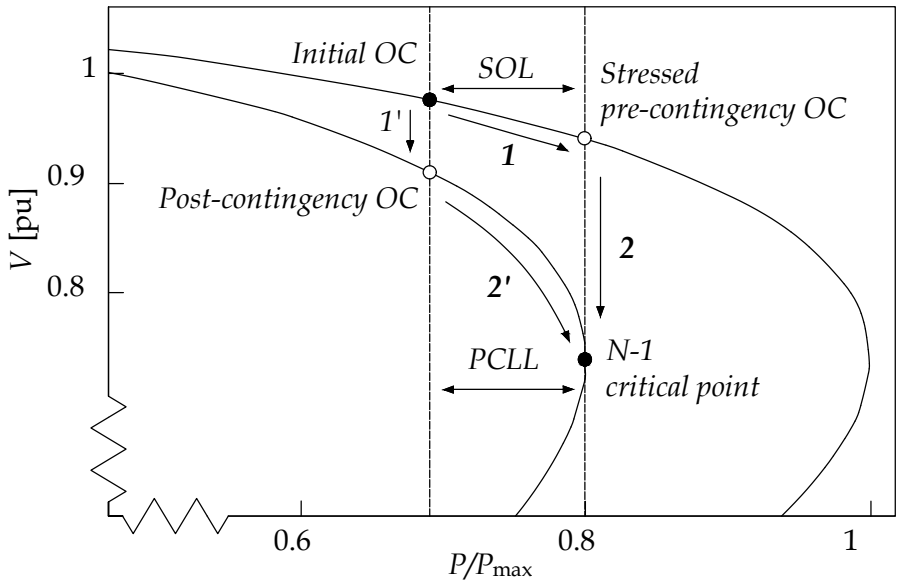

Figure 5.2: Difference in the estimation process for the PCLL and SOL illustrated using P-V curves.

- *PCLL estimation*: in PCLL estimation, the post-contingency OC is found by first introducing a contingency on the initial OC, which is followed by solving the resulting load flow study. This is illustrated in Fig. 5.2 by the movement along arrow **1′**. Once the post-contingency OC is found, the stability limit is then traced along the solution path by iteratively increasing the system stress until the critical point is reached, moving along the arrow **2′**. Continuation power flow (CPF) methods are preferably used to avoid convergence problems close to the collapse point [14,110]. The distance between the pre-contingency operating point and the N-1 critical point represents the PCLL.

- *SOL estimation*: the steps of estimating the SOL are conceptually different from the PCLL, where instead the dynamic security of the system is being tested with an increasing level of pre-contingency stress in the system, illustrated by arrow **1** in Fig. 5.2 [13]. For every new pre-contingency operating point (an increase in system stress), a dynamic simulation is initiated where the system response following a disturbance is studied. The simulation runs until the system stabilizes or becomes unstable. The final pre-contingency operating point that is tested and still provides a stable operating point is illustrated by moving along arrow **2** in Fig. 5.2. The distance between the initial operating point and the last pre-contingency operating point that can still handle a dimensioning contingency without causing instability then represents the SOL.

5.1.3 Load response after a disturbance

Loads are often recognized to maintain constant power characteristics in a long-term *system* perspective but do not generally behave as such following a disturbance. As-

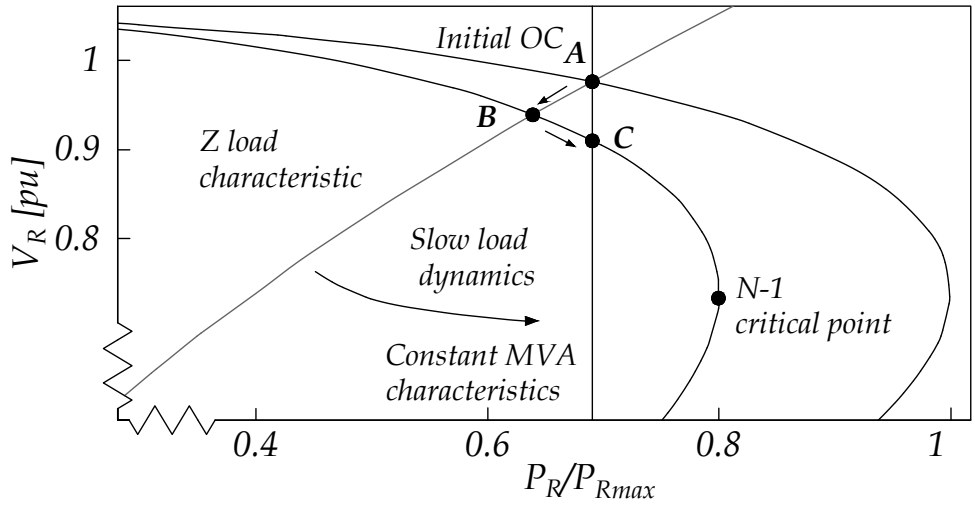

Figure 5.3: Example of slow load restoration after a contingency.

suming a sudden voltage change, loads will initially drop according to their instantaneous characteristics [109]. Then, the impedance or the drawn current is adjusted to restore the load to its original level; a process that can be exemplified by, for instance, the automatic changes in the slip of induction motors or by changes in tap positions to increase the voltage for loads behind LTCs.

A typical load restoration event following a disturbance is illustrated in Fig. 5.3. The initial OC is located at ***A***. Instantly after a disturbance, the load is assumed to have constant impedance characteristics, which results in a change in operating point from ***A*** to ***B***. Load restoration dynamics then change the operating point from ***B*** to ***C***, which corresponds to the same initial load level as point ***A***.

5.1.4 Transient P-V curves and fast load dynamics

The overall load restoration after a disturbance is generally assumed to act significantly slower than the dynamics of other system components, such as the dynamics of generators and excitation systems. The PCLL is based on this time-scale decomposition, where short-term dynamics, such as generator and excitation dynamics, are assumed to be in equilibrium. It was proven in [111], that if the system starts at a stable equilibrium and is *slowly* stressed towards the collapse point without encountering oscillations or other limit-induced events (e.g. reactive power limits for generators), the static equations are sufficient to locate the exact collapse point experienced by the dynamic system. Thus, if it is possible to assume that the short-term dynamics are in equilibrium and if no other limit-induced events occur, the loadability limit of the post-disturbance system can be found even though only static estimation methods are used to trace the security margin.

Load dynamics of induction motors and power electronic loads, such as chargers for electric vehicles, are inherently fast. For these components, the load is often restored in a time frame within a second, similar to other generator and excitation dynamics [12, 112]. Thus, if the system is dominated by loads with fast load restoration,

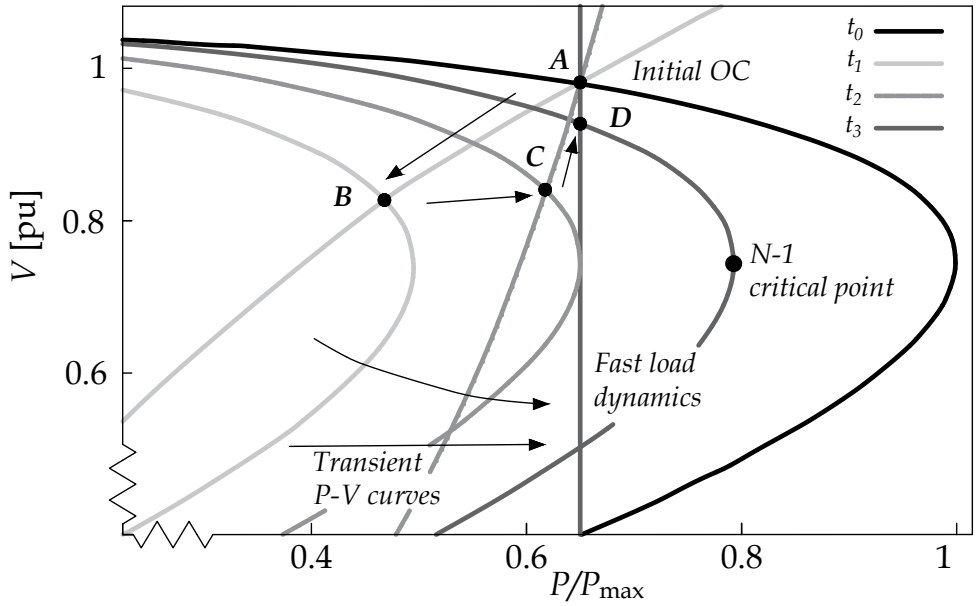

Figure 5.4: Transient P-V curves for a secure initial OC.

the assumption that the short-term dynamics are in equilibrium may no longer be accurate. In [109], a concept called transient P-V curves was adopted to allow visualization and analysis of short-term dynamics using classical P-V curves. Here, the same approach is used when the difference between PCLL and SOL is illustrated. In the analysis, the post-disturbance P-V curve is not fixed in time but is allowed to be affected by short-term system dynamics of, for instance, excitation systems. Nor is the load curve fixed in time, which allows the load restoration that follows after a disturbance to be illustrated. As the main purpose here is to provide a principal understanding of the concept, the transient P-V curves in the following figures are hypothetical and are not based on simulated values. Similarly, the curves illustrating the *fast* load restoration dynamics from a constant impedance load to a constant power load are drawn to allow a better understanding of the concept.

In Fig. 5.4, the dynamic response following a disturbance is illustrated for a secure initial OC. The load restoration curves and the transient P-V curves are illustrated using different shades of grey, where a lighter shade indicates closer in time after the disturbance. The time just after a disturbance is indicated by t_1; t_2 relates to a short time after the disturbance; t_3 relates to the time when all short-term dynamics are in equilibrium. The load is assumed to have long-term constant power characteristics, but just after a disturbance, the load will initially change to a constant impedance characteristic. Then, by fast load restoration, the load is quickly restored to the pre-disturbance level.

The initial OC is found in point ***A***. Just after the disturbance (at t_1), the short-term dynamics of system components such as generators or excitation systems will not yet have stabilized, which has the effect of shifting the post-disturbance P-V curve to the left. As a result of the initial load characteristics and the shifted P-V curve, the operating point moves along the arrow to operating point ***B***. After

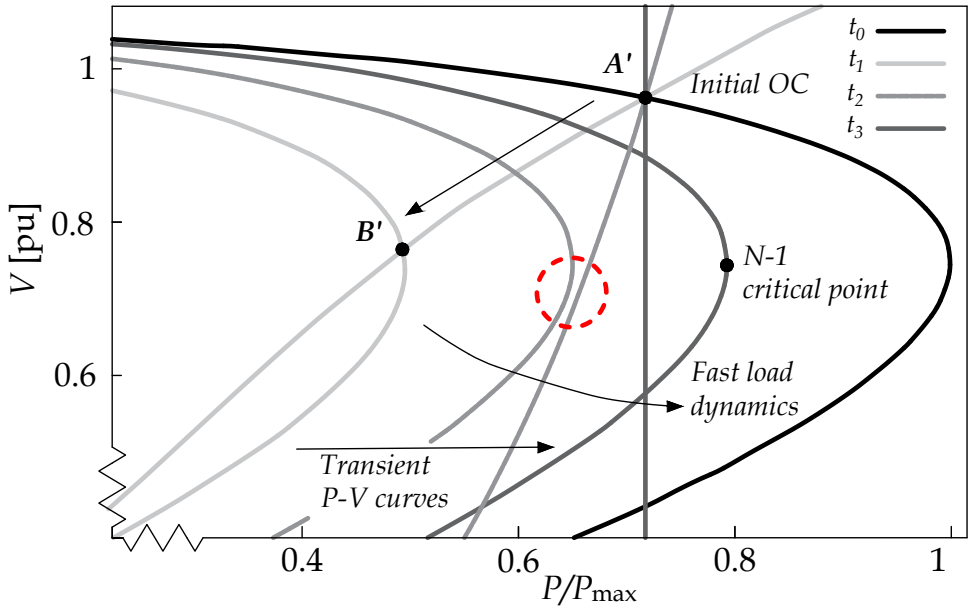

Figure 5.5: Transient P-V curves for an unsecure initial OC.

the shifted operating point, two separate dynamic responses are initiated. The system dynamics of mainly generators and excitation systems begin to stabilize, by for example restoring terminal voltages, which has the effect of restoring the P-V curve towards the final post-disturbance P-V curve. Simultaneously, the fast load dynamics is restoring the load from the initial load characteristics back to the pre-contingency load level. As an effect of the system dynamics and the load restoration, the operating point moves along the arrows from $\boldsymbol{B}$ to $\boldsymbol{C}$, then finally from $\boldsymbol{C}$ to $\boldsymbol{D}$. In this case, the system was found to be stable even after the disturbance with the new operating point $\boldsymbol{D}$.

In Fig. 5.5, the same system is slightly more stressed, with a higher level of initial transferred power. Just after the contingency, the operating point moves along the arrow from $\boldsymbol{A'}$ to $\boldsymbol{B'}$, by same the reasoning as in the previous example. However, due to the fast load dynamics, there exists no intersection between the curves at t_2, and without any emergency control actions, the system stability would be lost. The example in Fig. 5.5 illustrates a type of event that the SOL could identify and take into account, which is not possible when estimating the PCLL. It should be noted that the P-V curve for the N-1 case and the load characteristic at t_3 still intersect in this case, indicating that when using the PCLL, it would seem that the initial OC was still secure.

5.1.5 SOL versus PCLL

The analysis in the previous section showed that the SOL is preferred over the PCLL in power systems with a large share of loads with fast restoration dynamics. Furthermore, the closer power systems are being operated to the limits of operation, as for the event illustrated in Fig. 5.5, the more likely it is that the system will become unstable during the transient state after a disturbance.

Instability caused by the short-term dynamics that follows a disturbance can generally be divided into three different instability mechanisms [12, 66]:

- *Loss of post-disturbance equilibrium of short-term dynamics*: Typically exemplified by the stalling of induction motors after a disturbance causing the transmission impedance to increase. Due to the increased transmission impedance, the mechanical and electrical torque curves of the motor may not intersect, causing the system to lack a post-disturbance equilibrium, similar to the case illustrated in Fig. 5.5.

- *Lack of attraction towards the stable post-disturbance equilibrium of short-term dynamics*: Typically exemplified by transient angle instability and the loss of synchronism by one (or several) generators following a too slow fault clearing.

- *Oscillatory instability of the post-disturbance equilibrium*: Typically exemplified by rotor angle stability, in which the equilibrium between electromagnetic torque and mechanical torque of synchronous machines in the system is disturbed. Instability may be caused by increasing angular swings of some generators leading to their loss of synchronism with other generators [66].

Typically, time-domain simulations are required to capture the short-term dynamics after a large disturbance. SOLs computed using QSS simulations can not account for the short-term dynamics that follow after a disturbance and are thus better suited to only monitor long-term voltage instability phenomena. Extensions of the QSS model have been proposed that are capable of also incorporating frequency dynamics that take place over the same time scale as a long-term voltage instability event [38, 113]. Combinations of time-domain simulations and QSS, as proposed in [38], can use time-domain simulations to model the system during the short-term period following a disturbance, followed by QSS simulations used to simulate the long-term interval after the short-term effects are finalized. However, short-term instability may also be induced by long-term dynamics, where the system degradation caused by long-term instability eventually can trigger the above-mentioned short-term events [12]. It should be noted that SOLs computed by combinations of time-domain simulations and QSS, as proposed in [38], cannot capture this type of event.

5.2 Methodology for comparing SOL with PCLL

In this section, a methodology used to allow a fair comparison of the PCLL and the SOL is presented. A direct comparison between the SOL and PCLL is not trivial, as one is computed using a static model of the system, while the other is typically estimated using a dynamic model. Thus, although it is well-known that the PCLL and the SOL may produce significantly different estimations of the security margin, the difference in the results can be caused by both *how* the simulations were conducted, as well as owing to the fact that the SOL can account for the system's dynamic response after a disturbance.

To address this issue, a methodology is here developed that allows the two secu-

rity margins to be fairly compared and to isolate the root cause of the difference between them. Furthermore, a comparison for a large range of different static and dynamic load configurations and disturbance scenarios that are based on the developed methodology is presented. It should be noted that other aspects, such as post-disturbance controls and generation characteristics of, for instance, converter-interfaced generation, will also have a significant impact on the difference between the two security margins. The two security margin methods are analyzed mainly with respect to the following stability criteria: short-term and long-term voltage stability and rotor (transient) angle stability.

All simulations have been tested on the slightly modified version of the Nordic32 test system, detailed in Section 4.2. The security margins are computed by increasing the loading in the area "*Central*", while the generation in the area "*North*" is increased by a corresponding quantity. The starting point for all scenarios is the secure "operating point B" as defined in [114]. To ensure numerical stability during the dynamical simulation runs, a short integration step of 0.001 seconds was used in the simulations. In certain sensitive scenarios, such as when the simulations resulted in a non-converging dynamic simulation, the integration step was at times varied to provide a converging simulation.

5.2.1 Load models

In this section, the load models used in the comparison are presented. The power consumption of loads is generally affected by the system voltages and different load models are often used to characterize this relationship. A traditional load model used in both static and dynamic stability analysis is the ZIP model, which is made up of three components: constant impedance (Z), constant current (I), and constant power (P). The characteristics of the ZIP model is given by [12]:

$$P = zP_0 \left[a_P \left(\frac{V}{V_0} \right)^2 + b_P \frac{V}{V_0} + c_P \right] \tag{5.4a}$$

$$Q = zQ_0 \left[a_Q \left(\frac{V}{V_0} \right)^2 + b_Q \frac{V}{V_0} + c_Q \right] \tag{5.4b}$$

where $a_P + b_P + c_P = a_Q + b_Q + c_Q = 1$ for $a_x \geq 0$, P_0 and Q_0 are the real and reactive powers consumed at a reference voltage V_0, given that $z = 1$. V is the actual voltage and z is a variable indicating the actual loading of the system [12]. The constants a_x, b_x, and c_x represent the share of constant impedance, constant current, and constant power of the load, respectively.

Although simple and widely used in security analysis [115], the ZIP model cannot model any dynamic behavior of the loads themselves. The significance of induction motor loads and other fast-acting dynamic loads are often highlighted in system stability studies. Induction motors (IMs) are characterized by fast load restoration dynamics (often in the time frame of a second) and have a high reactive power demand. Induction motors are also prone to stalling, which may cause the motor to draw high reactive currents from the grid, resulting in a deteriorating effect on the system stability [12]. In PSS®E, a complex load (CLOD) model can be used

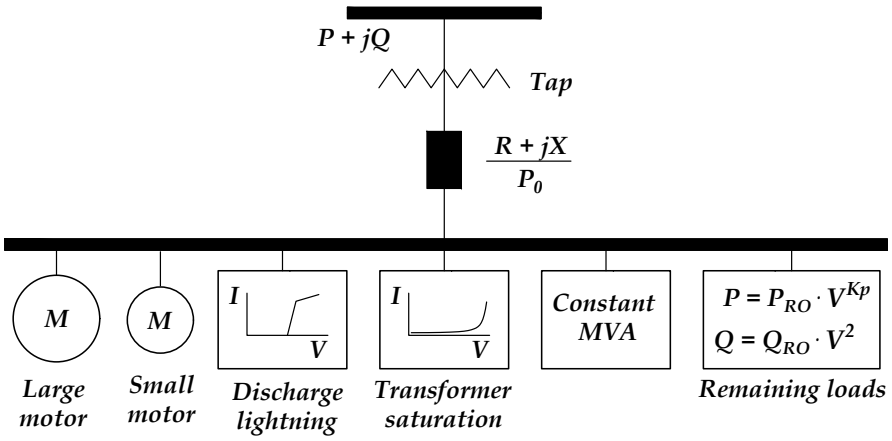

Figure 5.6: Overview of the CLOD model [116].

to represent a bundled mix of loads with different dynamic load characteristics into a single model. The CLOD models a composition of various load types including induction motors and several static loads but requires only eight parameters, which is achieved by internally using typical manufacturer data for each load type. The CLOD model was chosen as it provides a simple yet efficient solution to model different configurations of common load types, including induction motors, when no detailed dynamics data were available. In Fig. 5.6, a schematic of the CLOD model is presented. The transformer and feeders connecting the system bus to the load bus are modeled as a single impedance. At the load bus, different percentages of large and small IMs, discharge lighting loads, transformer saturation, and constant power loads can be modeled. The remaining part of the load is modeled as a polynomial load where the voltage dependency of the active load is controlled through a constant K_p. The performance curves of the two motor models, the discharge lightning model, and the transformer saturation model are further detailed in [116].

It should be noted that all constant power loads in PSS®E are modeled as constant power only for a certain range of load voltages. When voltages drop below a threshold, by default 0.7 per unit in PSS®E, the constant power loads instead follow a function based on the magnitude of the bus voltage, further detailed in [117].

5.2.2 Methodology and adaptations

To ensure that the difference in the computed security margins was not caused by differences in how the simulations were conducted, but rather by the fact that the SOL could better account for the system's dynamic response after a disturbance, a few adaptations of the methods were required. Instead of using CPF methods to compute the PCLL, a method that *slowly* ramps up the system stress in a dynamic simulation setting is adopted; an approach similar to the one used to compute the (pre-contingency) loadability margins in [114]. This approach allows the PCLL to be performed in a dynamic setting while mimicking how the system stress would have been increased if it would have been performed in a static setting. The advantage of

adopting this methodology is that the loading and the generation set points could be increased in the *exact* same way for both the computation of the PCLL and the SOL.

In [114], when computing the PCLL, the authors increased the system stress in small increments over time but did not evaluate whether the system had stabilized before continuing to stress the system. This could result in that additional system stress was being added to an already unstable system and that the PCLL became overestimated. For instance, long-term voltage instability events typically last several minutes, and a significant amount of system stress could thus have been added to the system while the system's maximum load power point had already been exceeded. To address this issue, an adaptive method to analyze whether the system had stabilized was adopted. To achieve this, the timer settings of LTCs and OELs were monitored continuously throughout the simulation. The LTC timers are activated whenever the voltage magnitude at a controlled bus is below (or above) a certain controlled bound. The OEL timers are activated whenever the field current of a generator is above a certain threshold. The specific timer settings for the OELs are computed by a function based on the magnitude of which the threshold is exceeded, see [118]. If the controlled voltages, respectively the field currents, are restored within the controlled bounds, respectively the field current thresholds, the timers are reset. These two components have the longest timer settings in the test system, and if all timers were reset for a given time (3 seconds) after a disturbance (or a load increase), the system was assumed to have stabilized.

5.2.3 Steps for PCLL computation

The steps used in computing the PCLL were the following:

- *Initialize dynamic simulation and introduce contingency*: The PCLL computation was initialized by applying a chosen contingency in the system in a dynamic simulation from the base case. The dynamic simulation then ran until the system was fully stabilized.

- *Increase system stress*: Once the system had stabilized after the initial disturbance, the system stress was increased in small increments of 1 MW, which was distributed among all the loads in the "*Central*" area. To reduce the required simulation time, the system stress was for certain fault scenarios (scenarios A and B) initially increased in larger increments (5 MW), since lower stress levels were found to not cause instability in the system. The different fault scenarios are further discussed in Section 5.3.1. The power factor of each load was kept constant. Simultaneously, the load change was compensated by the primary frequency response of the generators in the system. The added load for both the PCLL and SOL was computed as a nominal load increase at 1.0 pu voltage to ensure that the same amount of load was added for both methods regardless of the current load voltage in the dynamic simulation. Increased active line losses caused by the increased system stress were also automatically compensated by the generators' primary frequency response, while the reactive line losses were automatically compensated by the generators' excitation systems.

- *Check stability criterion*: After the increased system stress, the simulation continued to run until the system either stabilized or until a stability criterion was violated. The system was considered unstable if *any* bus voltage in the system was lower than 0.9 pu. Although the modified Nordic32 test system is characterized by sensitivity towards voltage instability, other types of instability can violate the stability criterion. For instance, transient angle instability can cause locally low voltages due to lost synchronism of certain generators.

- *Re-iterate*: The system stress was increased until the system eventually violated the stability criterion. The difference in loading from the base case to the final stable operating point in the stressed post-contingency system represented the computed PCLL.

5.2.4 Steps for SOL computation

The SOL was computed similarly to the PCLL, but by instead stressing the system in its pre-contingency configuration and *then* introducing the disturbance. The steps used in the computation of the SOLs were the following:

- *Initialize simulation and increase system stress*: When the ZIP model was used to model the loads, the dynamic simulation was initialized directly at the beginning of the simulation. The system stress was then increased in its *pre-contingency* base case in small increments of 1 MW, in the same way as was done for the PCLL computation in its post-contingency configuration. The small step size in system stress was chosen to allow the illustration of the security margins using P-V curves. In more general applications, faster methods such as the binary search method described in [13] or the dual binary search method described in [119], can otherwise be used to compute the SOL. *Adjustments for the CLOD model:* For the scenarios using the CLOD model, the increased system stress was required to be added in a static load flow scenario, which was then converted for dynamical studies. The increased load was distributed in a similar manner as when using the ZIP load model, except that the load was added in a static load flow scenario instead of during a dynamic simulation. However, the increased load could now not be automatically compensated by the generators' primary frequency response, and the increased load was instead distributed and compensated by increasing the generation set-points of all the hydro generators in the "North" and "Eq" regions, see Fig. 4.2. The distribution of the increased generation was based on the rated capacity of each generator and a generator with a higher rated capacity received a larger share of the increased generation. Increased active line losses caused by the increased system stress were compensated by an increase in the generation of the slack bus generator, "g20". While this distribution can be assumed to be relatively similar to how the primary frequency control of the governors would have compensated for the increased load, it will cause a small difference in how the system stress is increased between the two load models.

- *Introduce disturbance and check stability*: A disturbance was then applied in the system. A final end time of the dynamic simulation of 1,000 seconds was

used. The system was considered unstable if *any* bus voltage in the system was lower than 0.9 pu at the end of the simulation. The simulation was also stopped in advance if any bus voltage decreased below 0.7 pu (still allowing the system to first stabilize for 20 seconds after the disturbance).

- *Re-iterate*: If the system was stable, the system was reinitialized to the last pre-contingency state, followed by increasing the system stress by an additional 1 MW and applying the same disturbance. The SOL is then computed from the difference in loading from the base case to the final stable operating point in the stressed post-contingency system.

5.3 Simulation results and discussion

In this section, the results of the numerical comparison between the PCLL and the SOL are presented. Three different contingency scenarios were tested. The results of PCLL and the SOL computation are presented for each contingency scenario and each load model configuration. It should be noted that the different types of disturbances were chosen to exemplify the difference between the two security margin methods under various conditions. In real applications, all relevant contingencies that might be dimensioning for the security margin should be analyzed. In general, system operators perform contingency filtering (or selection) as it would be computationally infeasible to analyze *all* possible disturbances that might occur [13]. Furthermore, the direction of the system stress and the load-generation configuration should be representative of the specific system in consideration.

5.3.1 Contingency scenarios and loading scenarios

The following contingency scenarios were examined:

- **Scenario A:** A three-phased fault for 40 milliseconds, followed by tripping the faulted line. The faulted line is the one connecting the two areas "*North*" and "*Central*" between bus 4032 to bus 4044.
- **Scenario B:** A longer three-phased fault for 100 milliseconds, followed by tripping the faulted line. The faulted line is the one connecting the two areas "*North*" and "*Central*" between bus 4032 to bus 4044.
- **Scenario C:** Tripping of generator "g7" located at bus 1043 in the "*Central*" area.

For each of the contingency scenarios, different combinations of the ZIP load were tested for both the PCLL and the SOL. In addition, the SOL was computed for different compositions when the CLOD model was used to model the loads in the system. Adjusting the load levels *during* a dynamic simulation, which was required when computing the PCLL, was not feasible when using the CLOD model, as it requires that its internal parameters are recomputed whenever the load composition changes. Thus, the CLOD model was analyzed only with respect to the SOL. Furthermore, the CLOD model was found to be generally numerically unstable for

longer fault clearing times. Therefore, a comparison of the results for Scenario A is only provided with a fault clearing time of 40 milliseconds.

5.3.2 Simulation results

The PCLL and SOL results for each scenario and each load configuration using the ZIP model are presented in Table 5.1. The SOL results for scenario A and different configurations of the CLOD model are presented in Table 5.2. The largest difference between the PCLL and SOL is found for cases with a high share of constant power characteristics of the active part of the loads. For instance, for scenario 1A with fully constant power characteristics for the active part of the load and fully constant impedance characteristics for the reactive part of the load, the SOL was only 28 MW, while the PCLL was found to be 275 MW. The difference between the two security margin methods then reduces rapidly with a decreasing level of constant power characteristics on the active part of the load. Already at slightly lower levels of constant power loads, for instance, in Scenario 4A, the difference between the SOL and the PCLL becomes close to negligible. In Fig. 5.7, the post-disturbance P-V curves of the transmission side of bus 1041 are illustrated, respectively, for Scenario 1A. The P-V curves are computed by sampling the voltage magnitude when the system had stabilized after each dynamic simulation. Here, with a fully constant power characteristic of the active part of the loads, the P-V curves are almost identical for both the PCLL and the SOL up until the collapse point for the SOL.

Table 5.1: Computed PCLL and SOL for different loading and contingency scenarios.

	Constant								
	MVA	**I**	**Z**	***Scenario A***		***Scenario B***		***Scenario C***	
Scenario number	(P/Q) [%]	(P/Q) [%]	(P/Q) [%]	**PCLL** [MW]	**SOL** [MW]	**PCLL** [MW]	**SOL** [MW]	**PCLL** [MW]	**SOL** [MW]
1	100/0	0/0	0/100	275	28	275	4	351	71
2	95/0	5/0	0/100	340	88	340	86	353	128
3	90/0	10/0	0/100	341	146	341	144	357	196
4	80/0	20/0	0/100	364	362	364	260	365	362
5	50/0	50/0	0/100	387	386	387	387	380	378
6	95/0	5/50	0/50	280	55	280	48	283	85
7	80/0	20/50	0/50	359	240	359	233	357	356
8	50/0	50/50	0/50	382	382	382	381	375	372
9	0/0	100/0	0/100	425	424	425	425	407	405
10	0/0	50/0	50/100	465	464	465	456	439	438
11	0/0	20/0	80/100	488	488	488	489	457	458
12	0/0	0/0	100/100	504	504	504	505	471	471

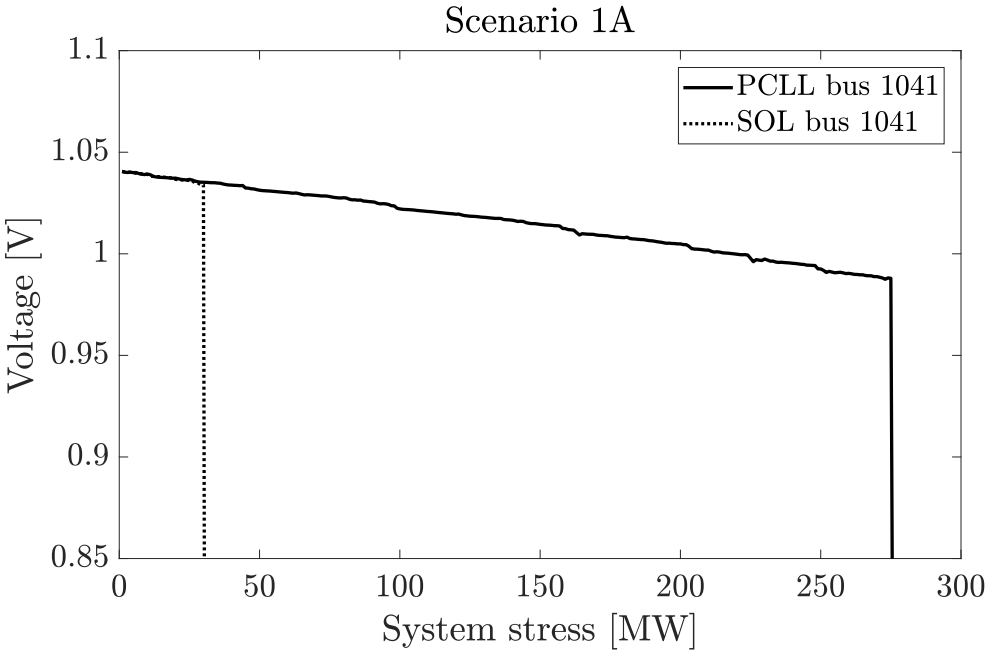

Figure 5.7: *P-V* curves computed at bus 1041 for scenario 1A.

The difference between the PCLL and the SOL is more significant for Scenario B when a longer fault clearing time was used in the simulations. For instance, in Scenario 1B, the SOL was estimated to be only 4 MW, compared to 275 MW for the PCLL. With reference to the discussion with the transient *P-V* curves presented in Section 5.1.4, a longer fault clearing time would have the effect of shifting the post-disturbance *P-V* curve for a longer time to the left, causing the system to lack attraction towards a stable post-disturbance equilibrium. Yet again, the difference between the two security margins decreases rapidly as the share of constant active power loads decreases. For instance, in Scenario 5B with a 50% share constant active power load, and the remaining part of the active load being of constant current characteristics, the SOL and the PCLL are almost identical. The post-disturbance *P-V* curves of scenario 5B on the transmission side of bus 1041 are illustrated in Fig. 5.8. The figure shows that although the computed *P-V* curves of the SOL are slightly below that of the PCLL, the two security margins find almost the same critical point of the system.

For all cases, except when the load is of fully constant power characteristics, the *P-V* curves computed using the SOL are *slightly* below the ones computed using the PCLL. Although the initial response of the excitation systems used in the Nordic32 test system is fast, there is an integrating part of the control system which takes a longer time until the voltage magnitudes of the generators are restored to their pre-disturbance set-point (differing slightly due to the droop in the automatic voltage regulation). In the PCLL case, this voltage restoration is allowed to fully stabilize after the initial disturbance before the system stress is added to the system. This is not the case for the SOL, in which the system is stressed before the disturbance is applied to the system. In turn, this causes LTCs and OELs to act earlier for a

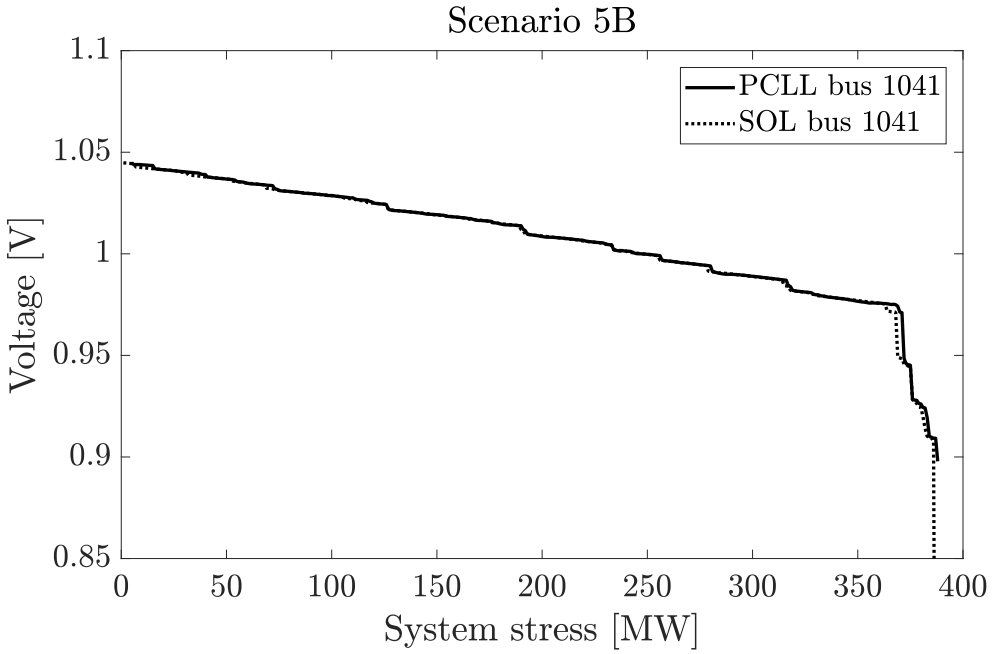

Figure 5.8: *P-V* curves computed at bus 1041 for scenario 5B.

lower level of system stress, causing the magnitude of the post-disturbance voltages to be generally lower.

In Scenario C, the chosen contingency was the disconnection of the generator "g7", located in the "*Central*" area. Once again, the largest difference between the PCLL and SOL is found for cases with a high share of constant power characteristics of the active part of the loads. For load scenarios with a larger share of either constant current or constant impedance characteristics of the active part of the load, the difference between the two security margins becomes negligible.

In Table 5.2, the computed SOLs for scenario A when using different configurations of the CLOD model are presented. The scenarios are generated by varying the load composition, consisting of large induction motors (LIMs), small induction motors (SIMs), discharge lightning (DL), transformer saturation (TS), constant power loads (MVA), and the remaining load which is of constant impedance characteristics ($K_p = 2$). Unsurprisingly, the computed SOL was the lowest when there was a large share of motor loads in the system. When the loads were modeled with a *too* high share of motor loads, such as scenario 13A, the computed SOL for the base case was negative. There was a relatively large difference between the computed SOL for scenario 17A with a 35% share of LIM loads and 25% share SIM loads, and scenario 20A with a 25% share of LIM loads and 35% share of SIM loads. LIM loads generally draw a higher reactive current during instances of low system voltages than SIM loads, which may have caused the computed SOL to differ from 47 MW for scenario 17A to 120 MW for scenario 20A.

In most scenarios where the CLOD model was used and for the level of system stress

Table 5.2: SOLs for different load configurations of the CLOD model

	CLOD model parameters						Scenario A
Scenario number	LIM [%]	SIM [%]	DL [%]	TS [%]	MVA [%]	Remaining (K_p=2) [%]	**SOL** [MW]
13	35	35	5	5	10	10	-
14	30	30	5	5	10	20	60
15	25	25	5	5	10	30	372
16	35	30	5	5	10	15	28
17	35	25	5	5	10	20	47
18	35	20	5	5	10	25	59
19	30	35	5	5	10	15	37
20	25	35	5	5	10	20	120
21	20	35	5	5	10	25	365

that made the system unstable, the system crashed during the transient state just after the disturbance. The CLOD models were found to be particularly sensitive towards long fault clearing times, and the Nordic32 test system consistently crashed when using a longer fault clearing time (such as 0.1 seconds). The difference between the two security margins is thus likely even greater if breakers with longer fault clearing times can be assumed to dominate the system. However, in a few scenarios, the long-term load restoration in the system was the main driver for instability. One of these cases, scenario 15A, is illustrated in Fig 5.9, which shows the development of bus voltages over time for different levels of system stress. For the lower system stress levels of 150 MW and 372 MW, the system is able to satisfy the given stability criterion, although the 372 MW level causes the system voltages to drop significantly. However, for a system stress level of 373 MW, the long-term load restoration and the activation of OELs cause the system to lose stability after about 500 seconds.

5.3.3 Discussion

The results in the previous section show that although the *same* operating point has been used as a starting point for all scenarios, the PCLL and the SOL differ significantly depending on the current load configuration and the type of fault that is considered. The largest difference between the two security margin methods was found when either the loads were of high constant power characteristics or consisted of a large penetration of induction motor loads. These results thus confirm the well-known fact that loads with fast restoration dynamics (where a constant power characteristic can be considered a theoretic extreme case) will deteriorate the system stability, and illustrate how significant this impact may be on the computed security margins.

The main conclusions of this study, that high penetration of loads with fast restoration dynamics will result in a difference between the PCLL and the SOL, should generalize well to other types of power systems. However, care should be taken

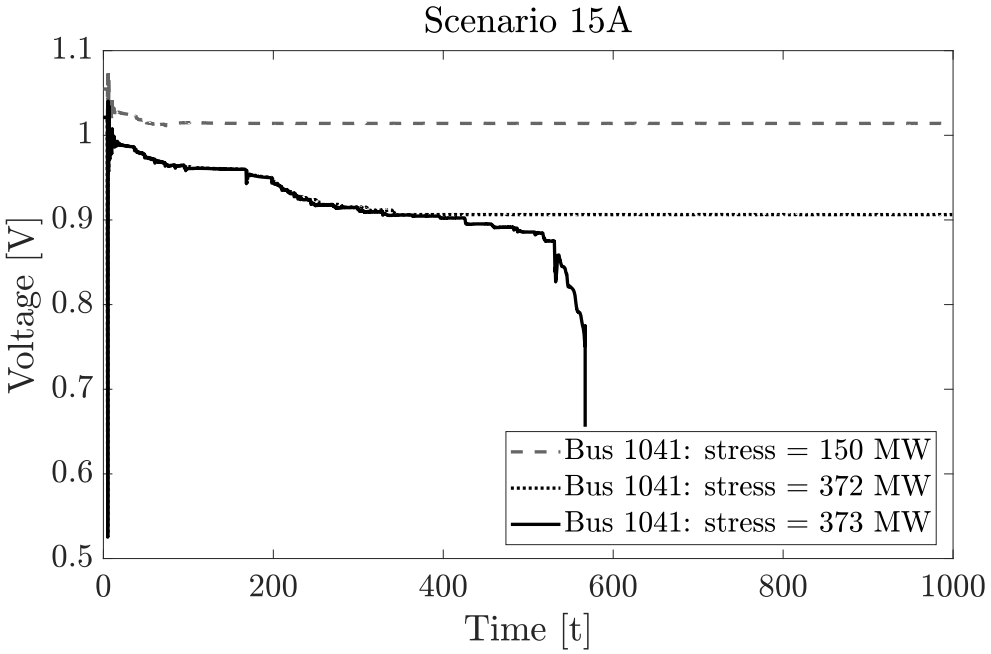

Figure 5.9: Voltage evolution for bus 1041 for Scenario 15A for different levels of system stress.

when generalizing the *specific* results of this study to real power systems with different characteristics. For instance, although the difference between the SOL and the PCLL in this study was found to be negligible whenever the share of constant power characteristic of the active part of the loads was lower than 50%, this is not necessarily the case for other systems with different dynamics. System operators would thus be required to perform a similar analysis on their specific systems to analyze during what specific loading scenarios the PCLL and the SOL start to differ.

The stability assessment practice of many system operators is, to the author's best knowledge, to compute security margin estimations based on the PCLL, often in combination with DSA. While DSA can provide certain types of security margins based on indices such as the transient energy functions [120], it does not provide an accurate measure of the loadability limit to the point where the system can no longer remain secure. If system operators continue to rely on conventional security margins computed by the PCLL, it is important to at least verify the reliability of those security margins to avoid either overly optimistic security margins or to avoid the need of adding unnecessary large reliability margins to the computed security margins. To account for modeling inaccuracies, transmission reliability margins are often added to ensure that modeling inaccuracies will not cause the system to be operated unknowingly in a non-secure operating state. Thus, if more accurate methods to determine security margins are used, such as the SOL, these reliability margins may theoretically be reduced and the system operators could more efficiently utilize the existing transmission capacity.

Dynamic load modeling may also become increasingly important in the future, as more loads are expected to be controlled through power electronically-controlled

interfaces. These types of loads, such as electric vehicle chargers, inhibit very fast dynamic responses after disturbances [121]. Despite this, dynamic load models are still relatively unused in the industry. In a large survey study from 2013 on international industry practice on power system load modeling, it was shown that about 70% of system operators and utilities still only used static load models for power system stability studies [122]. A drawback of advanced load models is that the load composition is often partly unknown to system operators, and it is thus more straightforward to use the simplified static load models. Another drawback is the increase in computational requirement during simulations, which reduces their applicability in real-time applications. However, although complex load models do not necessarily need to be used in real-time applications, sensitivity analyses can preferably be performed using these models, so that the impact of various degrees of motor loads and other types of loads on the stability of a system can be studied.

While this study focused on the impact of different load models, converter-interfaced generation and other power electronic devices in the power system will also have a significant impact on the computed security margins. Although a growing share of renewable generation is often challenging from a planning perspective due to the intermittency of the energy source, the converter interface may in fact mitigate some of the short-term instability phenomena. For instance, with proper design of the converter controls, such components can contribute in providing fast voltage/reactive power control or active power control for fast frequency responses. The impact of such components, also in combination with loads with fast restoration dynamics, deserves further attention in future research.

Chapter 6

Fast dynamic voltage security margin estimation

This chapter presents a method for fast dynamic voltage security margin estimation based on the methodology and the results established in Paper V. The method is proposed to be included as a preventive monitoring application in the developed real-time dynamic stability assessment and control tool, earlier presented in Chapter 4.

6.1 Introduction

In Chapter 5, the circumstances when the SOL is preferred to the conventional PCLL was presented. However, SOL estimation is computationally demanding, where multiple time-domain simulations (or combinations with QSS) are required to trace the security limit for a range of different contingencies. In this chapter, a methodology for fast estimation of the DVSM is proposed to overcome the computational difficulties when estimating the margin. The DVSM is computed identically as the SOL, with the difference that only voltage stability limits are taken into account for the stability criterion.

The method uses two different NNs to provide both an estimate of the actual DVSM at a specific OC, and to determine the dimensioning contingency for the system with respect to the DVSM. These *estimated* values are then used as starting points in a method called dual binary search to significantly reduce the required computational time in computing the *actual* DVSM. The method is developed to mitigate inconsistency issues associated with ML methods under new or unseen operating conditions.

6.2 Methodology for fast estimation of the DVSM

The NNs are trained on a training set consisting of i) computed values of DVSM for a range of different OC, and ii) the dimensioning contingency for the same OCs. Once the NNs are trained, they can almost instantaneously provide estimations

of the DVSM for a certain OC, and classify which contingency is most likely to be dimensioning for the system security margin. The first step of the method is the generation of credible OCs and estimations of the DVSM for a set of credible contingencies. The method is tested on the Nordic32 test system found in Fig. 4.2 with all data and models as presented in [108]. After a representative training set has been generated, the training scheme of the two NNs is presented. Each step in the methodology is described in the following subsections.

6.2.1 Generation of training data

The training data for the NNs were generated using PSS®E 34.2.0 with its in-built dynamic models [118]. Here, full time-domain simulations have been used, but the methodology could also be generalized for combinations of QSS and full time-domain simulations. The steps of generating the training data are illustrated in the flowchart in Fig. 6.1 and can be summarized as follows:

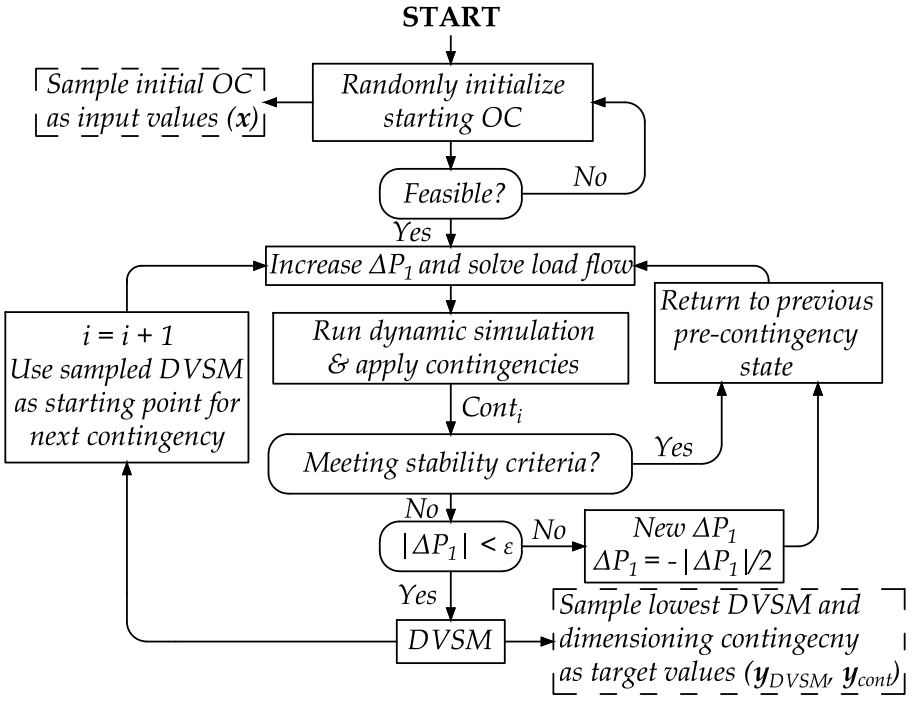

Figure 6.1: Flowchart of the generation of training data for the DVSM and the NN.

(i) *Choose initial operating conditions*: All initial OCs were randomly generated around the stable operating point of the simulated Nordic32 system denoted as "operating point B" in [108]. The total load in the system for each initial OC was generated by multiplying *all* the active loads randomly from the *same* uniform distribution (80 % of the original load as the lower limit, 95 % of the original load as the upper limit). Then, each *individual* load was varied by again multiplying the now updated load value with a random variable generated from a new uniform distribution (this time with 80% as the lower limit, and 120 % as the upper limit). The power factor of all loads was kept

constant. The total change in loading was then randomly distributed among all the generators in the system. The generated initial OCs were first solved using a conventional full Newton-Raphson load flow solution, which served as a starting point for the dynamical simulation. In the case the system was not found feasible, the initial OC was re-initialized.

(ii) *Increase system stress and solve load flow*: The system stress was then increased for the secure initial OC by increasing the power transfer between the two areas "North" and "Central". The increased system stress was achieved by increasing the loads in the "Central" area with a total of $\Delta P_1 = 200$ MW, while simultaneously increasing the generation in the "North" area by the same amount. The power factors of each load were again kept at the initial values. The distribution of the added load and generation was based on the initial load or the rated capacity of each generator. Thus, a bus with a larger initial load, or a generator with a higher rated capacity, received a larger share of the increased load and generation. The required generation that could not be supplied by the regular generators was distributed to the slack bus generator in the system, g20, see Fig. 4.2. After the loads and generation were updated, the load flow was reiterated which served as a starting point for the time-domain simulations. To avoid numerical and stability issues when increasing the system stress of the static system, the system stress was increased in small increments where a load flow solution was solved for each increment.

(iii) *Run time-domain simulation and test for security*: A time-domain simulation was then initialized for the first contingency. In the relatively small Nordic32 test system, the same single contingency was found to be dimensioning for almost all different initial OCs. To test the possibilities of using a NN to classify the dimensioning contingency, two different contingencies were handpicked as they were found to be dimensioning for different OCs. The tested contingency type was a three-phased fault on a transmission line applied for 0.1 seconds, followed by tripping the faulted line which was then kept tripped during the remaining time of the simulation. The lines between the buses "$4031 - 4041$" and "$4032 - 4044$", connecting the "*North*" and "*Central*" areas were used, see Fig. 4.2 for reference. Each simulation then ran for a total of 500 seconds. The system was considered secure if, at the end of each simulation, *all* transmission bus voltages were above 0.90 pu.

Each dynamic simulation ran for a total of 500 seconds but was in the case of a major voltage collapse stopped in advance. The simulation time was chosen to ensure that the system either fully stabilized or collapsed. It should be noted that the required simulation time is dependent on the power system in consideration, and it is likely that different simulation times would be required in actual implementations of the algorithm. The system was considered secure if, at the end of each simulation, *all* transmission bus voltages were above 0.90 pu.

(iv) *Re-iterate and test other contingencies*: In case the system was found secure

for the tested contingency, the system stress was increased again with ΔP_1, followed by another security test. In case the system was *not* found secure, the previously added system stress was reduced by half, and the process was re-initialized. This process of iteratively updating the system stress and testing for security continued until the increase in system stress was below a precision value of $\epsilon = 5$ MW. The DVSM was then computed by taking the difference in system loading between the initial OC and the secure system with the highest level of system stress.

Once the DVSM for the first contingency was computed, the same procedure was repeated for the second contingency. To save computational time, the estimated DVSM for the *first* contingency was used as a *starting point* for the estimation of the second contingency. If the system at that level of system stress was found secure for the second contingency, the simulation was stopped. Otherwise, the search algorithm continued until a new smaller value of the DVSM was found.

(v) *Sampling the input values and target values*: An input vector $\boldsymbol{x}$ consisting of measurements of all bus voltage magnitudes and angles, and active and reactive power flows were sampled from each one of the initial OCs. The choice of which input values to include in the training was based on the results in [123], which found that bus voltage magnitudes and angles were found to be the best combination of inputs when estimating the VSM using a NN. The active and reactive power flows were then added as additional inputs as this was found to increase the accuracy in the estimations even further. Two target vectors $\boldsymbol{y}_{DVSM}$ and $\boldsymbol{y}_{Cont}$ were generated by sampling the DVSM for each case, and the contingency that was dimensioning for the specific case, respectively. The previously described steps were re-iterated until a sufficiently large training set was generated.

6.2.2 Architecture of the neural networks

The architecture of the two NNs used in this thesis are presented in Fig. 6.2 and the specific details regarding the architecture and the training parameters of the two NNs are specified in Table 6.1. In the training phase, the two NNs take the same vector of input values, which are forwarded to each of the hidden layers through a set of weights, illustrated by the lines connecting each of the neurons. The output of each neuron in the hidden layer is computed using a non-linear activation function on the sum of all the inputs, which is then forwarded to the output layers. The ReLU was used as the non-linear activation function for the two NNs. For the NN estimating the DVSM, the outputs are forwarded to a regression layer with a linear activation function. For the NN responsible for ranking the contingencies, the outputs are forwarded to a layer with a softmax activation used for classification. The softmax activation function is generally used for multi-class classification but generally works well also for binary classification as is the case here. The softmax activation function outputs a probability vector, where each class is given a certain probability. The probability vector can then be used to rank the contingencies in

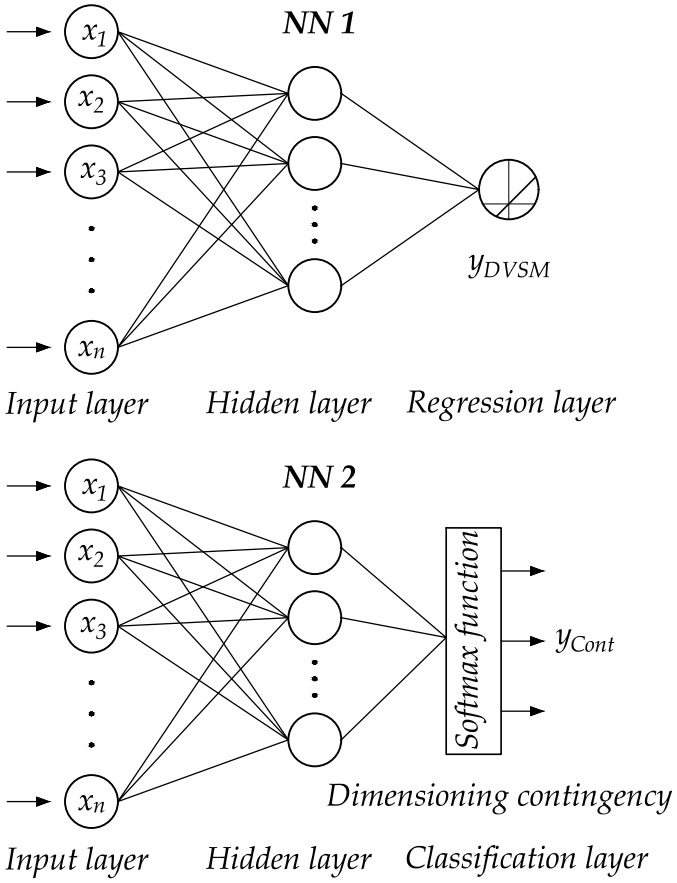

Figure 6.2: The two NNs trained to evaluate the DVSM and the buses with the lowest margin to instability.

order of which most likely will become dimensioning.

In the training phase, the networks use the true target vectors y_{DVSM} and y_{Cont}, while during the test or prediction phase, the network estimates the DVSM and the ranked contingencies by generating the vectors $\hat{y}_{DVSM}$ and $\hat{y}_{Cont}$ for the current OC. The supervised training approach aims to update and learn the suitable values for the weights connecting each layer, implicitly modeling the non-linear relationship between the inputs and outputs.

6.2.3 Training

Different data sets were used in the training, validation, and testing of the method. The training data has the dimension (364 x 6,000), where the dimensions represents the number of inputs, and the total number of training cases, respectively. Each network was trained for a maximum number of epochs, where an epoch is finished when all the cases in the training set have been used to update the network parameters. To reduce overfitting on the data, ridge regression (also known as L2 regularisation) was used to ensure the data does not rely too heavily on any single

Table 6.1: Design and hyperparameters used in training.

	Parameter	NN 1 / NN 2
Data	Number of inputs	364 / 364
	Training cases	6,000 / 6,000
	Validation cases	400 / 400
	Test cases	400 / 400
Architecture	Hidden layers	1 / 1
	Final activation function	Linear / Softmax
	Hidden cells	128 / 32
	Hidden layer activation	ReLU / ReLU
Training	Max Epochs	1,000 / 3,000
	Learning rate (α)	$1 \cdot 10^{-6}$ / $1 \cdot 10^{-5}$
	Dropout	0 % / 50 %
	L2 parameter	0.01 / 0.01
	Optimizer	Adam / Adam [124]
	Loss metric	MSE / Categorical cross-entropy

feature. To further reduce overfitting, a technique called dropout was applied where a certain percentage of the connections between each layer were masked/dropped, to ensure that the model does not rely too heavily on certain connections. The MSE was used as a metric for the NN estimating the DVSM, while the categorical cross-entropy loss function was used for the NN classifying the dimensioning contingency. An adaptable algorithm for gradient-based optimization, Adam, was used in training the network [124]. The learning rate was the only parameter that was specifically tuned for the algorithm, while the remaining used the default values according to [124].

It should be noted that both the training parameters and the architecture of the two networks have been iteratively tuned to increase the regression and classification accuracy. A deeper architecture with more hidden layers was found to not increase the performance for the specific test case and training set size. Other hyperparameters and network architectures would likely have better performance for other test systems than the Nordic32. By increasing the training set size further and spending even more effort in tuning the networks, even better accuracy could theoretically be achieved.

6.2.4 Fast DVSM estimation and dual binary search

In [13], a binary search method was proposed to estimate the DVSM. Here, an alternative approach denoted as the *dual binary search method* is proposed, which should be able to increase the computational speed of the DVSM. The trained NNs in Fig. 6.2 takes the same set of measurements and generates: (i) an estimated value of the DVSM, and (ii) an estimated ranked order of the contingencies that most probably will be dimensioning for the current OC. The estimated DVSM is used as a

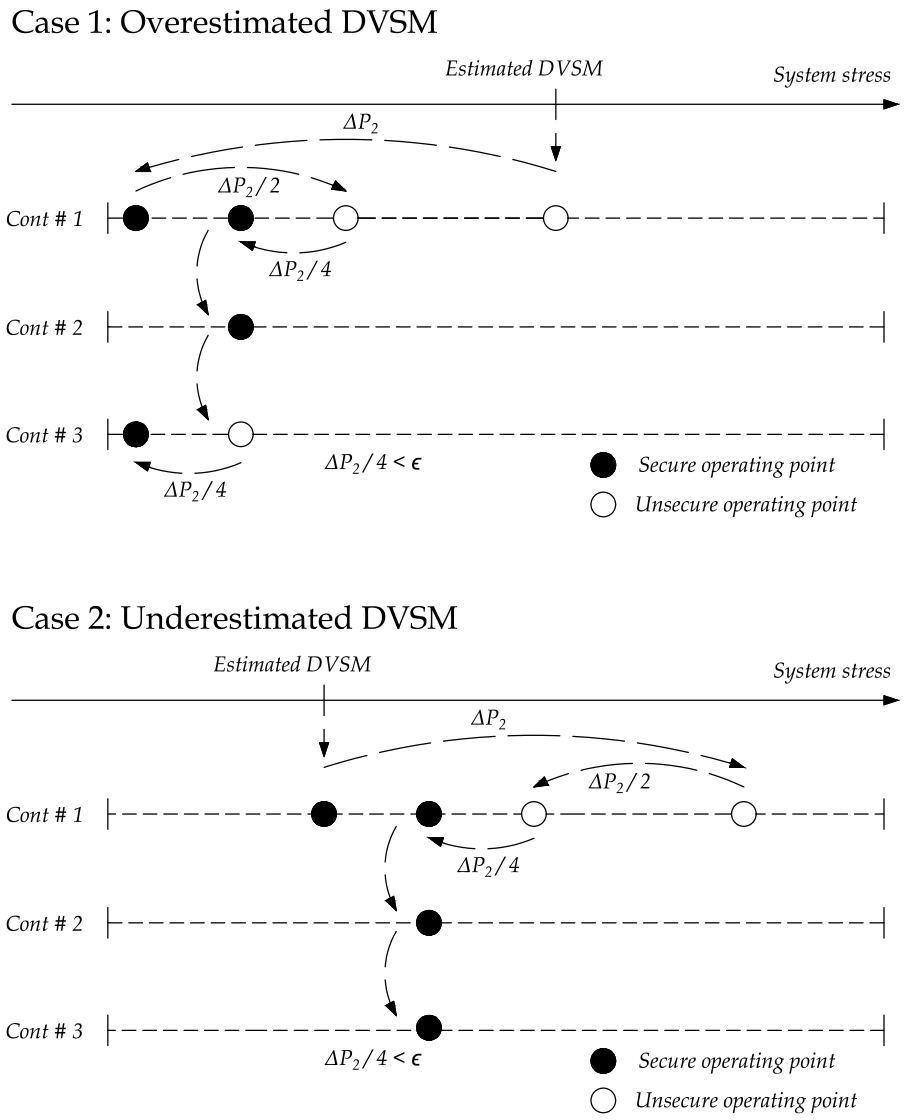

Figure 6.3: Dual binary search for multiple contingencies.

qualified estimate of the real DVSM, which is validated through actual time-domain simulations. The dual binary search method is then used to take advantage of the estimated DVSM and the dimensioning contingency to reduce the computational time when validating the real DVSM for the system.

The dual binary search method is illustrated for two cases in Fig. 6.3. Case 1 illustrates the estimation process for an overestimated value of the DVSM, while Case 2 illustrates the estimation process for an underestimated value of the DVSM. Black dots indicate secure operating points and white dots indicate unsecure operating points. The estimated DVSM is always the starting point for the search of the *actual* DVSM of the system. The system stress is increased to this point iteratively using the approach explained in section (ii) to avoid convergence problems.

Once the stressed static base case is found, a time-domain simulation is initiated for the *highest* ranked contingency by the second NN, which is the contingency that most likely will be dimensioning for the DVSM. The initial estimated DVSM level is then tested for the chosen contingency. In case it is stable (respectively unstable), the system stress is increased (respectively decreased) with a certain value represented by ΔP_2. A value of ΔP_2 equal to the MSE of the estimated values for the DVSM is proposed, which should represent a reasonable uncertainty and step size for the estimation. If the new operating point is found to be secure, the system stress is again increased with ΔP_2. In case it is not found to be secure, which happens in the example illustrated in Fig. 6.3, the system stress is reduced by $\Delta P_2/2$. The dual binary search is then continued until a secure operating point is found and when the step size in system stress change is smaller than a specified precision level ($\boldsymbol{\epsilon}$).

This level of system stress is then tested for the other contingencies, in ranked order, until all lower-ranked contingencies have been tested and found secure. For both the cases illustrated in Fig. 6.3, this level of system stress for the second-ranked contingency was found to result in a secure operating point. A third and final ranked contingency is then tested, which for Case 1 in Fig. 6.3 is found to be unsecure. The system stress is thus reduced further for this case, resulting in a secure operating point which then constitutes the dimensioning DVSM for the system. It should be noted that for Case 1, the contingency ranking was not perfect, with the result that an extra time-domain simulation was required.

6.3 Results and discussions

In the following section, the results from the regression and classification on the test set for the two NNs are presented. Furthermore, the reduction in computational effort is compared between the conventional tracing method and the proposed dual binary search method. Finally, practical applications and discussions related to DVSM estimation are presented.

6.3.1 Regression and classification accuracy

The prediction accuracy for the NN estimating the DVSM is presented in Fig. 6.4, where the estimated DVSM is plotted with respect to the real DVSM for the test set. The diagonal line indicates where the points should lie in case the estimated DVSM perfectly matches the real DVSM. Table 6.2 lists the mean and maximum error of the estimations in percentage, as well as the MSE presented in MW. The results indicate that the NN is generally capable of accurately estimating the DVSM given an initial OC, with a mean error for the test set of 1.49 %. The maximum estimation error was found to be 10.96 %, while the MSE was estimated to be 13.35 MW.

The classification accuracy of the NN used in ranking the dimensioning contingency is presented in Table 6.3 in the form of a confusion table. Each number in each row represents the instances of the real dimensioning contingencies, while each number in each column represents the instances of the predicted dimensioning contingencies.

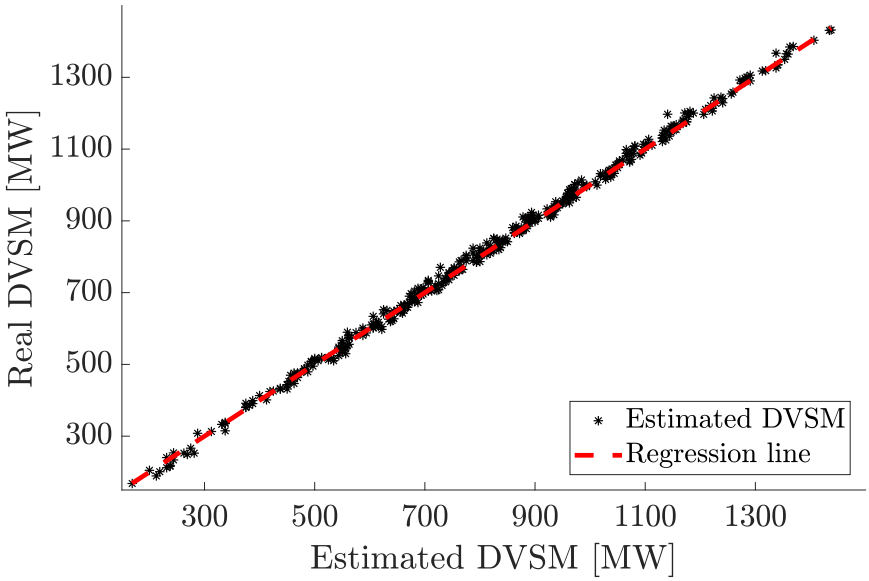

Figure 6.4: Prediction accuracy of estimating the DVSM.

Table 6.2: Regression results for estimating the DVSM.

Mean estimation error [%]	Maximum estimation error [MW]	Mean squared error [MW]
1.49	10.96	13.35

The conditional probabilities of correctly classifying the dimensioning contingency are presented in the column furthest to the right. Similarly, the conditional probabilities of a dimensioning contingency actually *belonging* to the predicted class are presented in the bottom row of the table. The total accuracy for the classification is presented in the rightmost corner of the table, and an accuracy of 91.3 % was provided for the test set. Thus, in about 9 instances of 10, the NN is capable of classifying which contingency that will be dimensioning for the DVSM for a specific OC. It should again be mentioned that the estimation and classification results could be enhanced further by either increasing the training set size or by a more careful exploration of suitable hyperparameters for the training of the networks.

Table 6.3: Confusion table showing real and predicted dimensioning contingencies.

		Predicted		
		L4044 - L4032	**L4044 - L4032**	**Accuracy**
Real	**L4044 - L4032**	104	23	81.9 %
	L4044 - L4031	12	261	95.6 %
	Accuracy	89.6 %	91.9 %	91.3 %

6.3.2 Computational efficiency

In this section, the computational efficiency is compared between the proposed fast dual binary search method and the more conventional tracing method that was used in generating the training set (see Section 6.2.1 for reference). The proposed fast dual binary search method, explained in Section 6.2.4, uses the estimated DVSM value and the dimensioning contingency as a starting point to validate the real DVSM. The computational efficiency is measured as the average number of time-domain simulations required in estimating the DVSM. The results of using the two different methods are presented in Table 6.4. The average number of time-domain simulations required in estimating the DVSM using the *conventional* tracing method was found to be 15.3, while the corresponding number using the proposed dual binary search method was 4.7. The reduction in the average number of time-domain simulations required was thus -69.2 % when the proposed method was applied.

Table 6.4: Reduction in computational effort using the proposed method.

Average number of time-domain simulations		
Conventional tracing method [No.]	**Dual binary search method** [No.]	**Relative reduction in computation time** [%]
15.3	4.7	-69.2 %

It should be noted that the *exact* comparison in computational efficiency between the two methods is of comparatively little interest, as it mainly applies to the specific test case used in this thesis. For instance, the computational savings are probably significantly higher in most real applications, where a larger range of contingencies may be dimensioning for the DVSM. Furthermore, in real applications where the range of the DVSM may be larger than what has been used here, the conventional search algorithm would require significantly more time-domain simulations to find the true DVSM. Similarly, it is also possible to further enhance both the conventional search algorithm and the dual binary search algorithm by, for instance, choosing more suitable values of ΔP_1, or increasing the precision value of $\boldsymbol{\epsilon}$. The most notable result is instead that the computational effort in estimating the DVSM can be reduced from requiring a large number of time-domain simulations, to only requiring a few. Although a few time-domain simulations would still take some time to compute for a real power system, it should be possible to provide sufficiently fast estimations of the DVSM to classify it as a "*near real-time*" estimation.

6.3.3 Impact of sudden topology change

In any real application, the performance of a NN is dependent on its generalization capability. This refers to the capability of the NN to generalize the learning from the actual training set to other, yet unseen, cases. In this section, the performance of the NNs to generalize their estimations when subjected to test cases where unplanned topology changes have taken place is examined.

For simplicity, only topology changes in the form of opened transmission lines are considered. To ensure that the Nordic32 test system is still secure despite the topology changes, only topology changes in the "North" region, see Fig. 4.2, were considered. Furthermore, only transmission lines between buses served by two parallel transmission lines were used in generating the test set with topology changes. A new test set of 400 cases was then generated in the same manner as explained in Section 6.2.1, with the difference that the topology changes were now added randomly.

In Fig. 6.5, the prediction accuracy of the NN estimating the DVSM is presented when the network has been trained on two different training sets. Case 1 presents the prediction accuracy when *no* unseen cases with topology change have been included in the training set. Case 2 presents the prediction accuracy when a few (100) training cases with topology change have been included in the training set. For Case 2, the two NNs were re-trained on the updated training set using the same training approach as previously described. Table 6.5 lists the mean error of the estimations in percentage for each case, as well as the MSE presented in MW. The result for Case 1 indicates that a sudden topology change will significantly affect the accuracy of the predictions. Although many cases were accurately predicted, the number of outliers increased significantly. The prediction accuracy was higher for Case 2, even though only a very small number of cases with the topology changes were added to the training set. The classification accuracy of the NN used in ranking the dimensioning contingency was also affected significantly for the two cases, with a total classification accuracy of 55 % and 78.5 % for case 1 and case 2, respectively.

The results highlight the importance of obtaining a representative training set and also taking into account the possibilities of unplanned topology changes. It was seen, that by the inclusion of even a very small set of training cases with various topology changes, the prediction accuracy could be increased significantly. Thus, in the event of an unplanned change in the system, the system operator could quickly generate a small training set on the new OC, and then retrain the NNs on the generated data. It should be noted that the proposed robust methodology of always validating the estimations of the NNs with *actual* time-domain simulations reduces the impact of these types of erroneous estimations. The main impact of a poor estimation of the DVSM will be that the time to validate it will increase.

Table 6.5: Regression results of the DVSM estimation for the two cases.

	Mean estimation error [%]	Mean squared error [MW]
Case 1	12.97	134.38
Case 2	7.33	63.59

6.3.4 Discussions and practical applications

The proposed method is aimed to be used as an online tool for system operators to estimate a power system's dynamic voltage security margin. The method does not necessarily have to replace conventional VSM estimation, but may instead be used

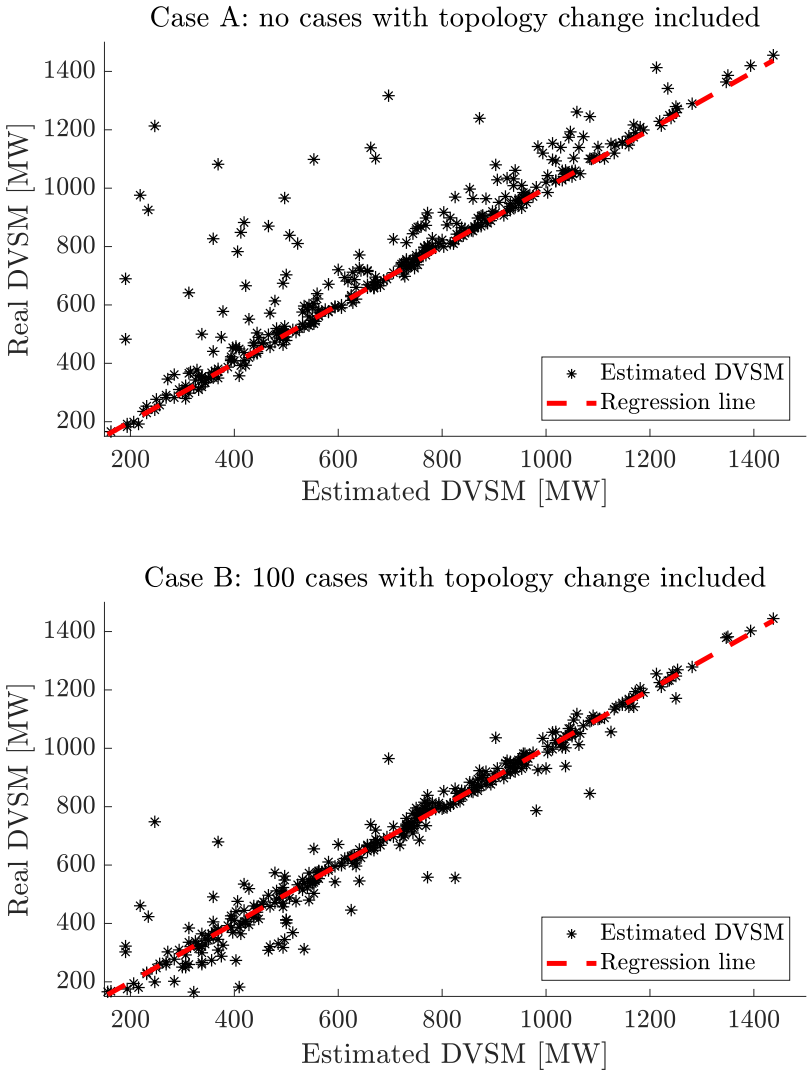

Figure 6.5: Prediction accuracy of the NN during sudden topology changes. With and without trained on a training set with a 100 training cases with topology changes included.

as an additional source of information to system operators to provide better and more accurate estimates of the total transmission capacity in their systems.

Theoretically, the DVSM estimates by the NNs could be used directly to provide real-time estimates of the security margin. However, despite years of research, examples where these methods have been practically applied in system operators' monitoring and control systems are, to the authors' best knowledge, very few. From a system operator view, an inferior method that *always* works are generally preferable to a superior method that in some instances does not. The proposed method is thus suggested to utilize the advantages of ML, while still ensuring that the method always provides good estimates regardless of the current OCs.

The results in the previous section indicated that by using the proposed method, the number of required time-domain simulations to estimate the DVSM could be reduced to only a few, allowing system operators to estimate the DVSM in a time

frame that could be defined as "near real-time". The update frequency of security margins will affect the required transmission reliability margins as the underlying system continuously change between the assessments. The actual estimation speed will still be affected by a range of different factors such as the computational speed of the hardware being used, the size of the specific power system in consideration, or the required precision (i.e. the value of ϵ).

Measurements of bus voltage magnitudes and angles, as well as active and reactive power, have been assumed to be available, either directly from measurements or from state estimations of the system. However, to ensure that missing values and errors are filtered out, all measurements should preferably be preceded by a state estimator. To adapt to the evolving operating conditions and self-rectify any bad predictions, the two NNs should be trained continuously during operation. Using approaches such as stochastic gradient descent, the NNs weight parameters can continuously be tuned to increase the robustness and accuracy.

Chapter 7

Voltage instability prediction using a long short-term memory network

This chapter presents an algorithm for real-time voltage instability prediction based on the developed methodology and the resulting findings in paper IV and paper VI. The methodology for training the voltage instability prediction tool to indicate where in the system the instability would emerge was first presented in paper VI. The final voltage instability prediction tool is mainly based on the proposed architecture presented in paper IV. The method is proposed to be included as an emergency monitoring application in the developed real-time dynamic stability assessment and control tool, earlier presented in Chapter 4.

7.1 Introduction

In Chapter II, the voltage instability event and the need for emergency monitoring systems were discussed. It was further discussed that conventional methods for VID may be too slow to detect instability in time for system operators to initiate sufficient emergency control actions. An alternative option is to use ML-based methods for VIP, which can predict the onset of instability only seconds after a disturbance has occurred in a system.

These methods are trained to indirectly correlate a post-disturbance state and learn its dynamical trajectory, to be able to directly assess whether the system is heading towards instability. Most previously developed methods for VIP have in common that only instantaneous measurements are used as inputs to the VIP algorithms. These inputs represent the "state signal" that the ML algorithm uses to predict the future state. Ideally, the state signal should summarize all relevant information required to determine the future state of the system. A state signal achieving this is said to have Markov property [101]. However, the dynamic response of a power system cannot be modeled as a first-order Markov process using only the static

states provided by available measurements in the power system. Rather, the future state of the system also depends on a range of unknown state variables such as the rotor speed of generators, tap positions, or rotor slips of induction motors.

In response to these limitations, a new method based on an RNN with LSTM is proposed. LSTM networks excel at capturing long-term dependencies [99], which is an inherent aspect in long-term voltage stability [43]. The methodology and test results for the proposed method, from here on denoted as LSTM-VIP, are presented in the following sections.

7.2 Methodology for LSTM-VIP

The proposed method for real-time VIP is based on off-line training of an LSTM network on a large data set consisting of time-domain simulation responses following a set of credible contingencies. The method is aimed to be used as a supplementary warning system that can assess the current state of the system in real-time. The LSTM network takes real-time and historic measurements and attempts to assess whether the *current* state will cause voltage stability issues several minutes into the future. As time progresses and if new events occur in the system, the network updates the assessment continuously. The network is also adapted to be able to indicate *where* in the system instability emerges, following the approach developed in [125], allowing more cost-effective countermeasures.

The first step of the method is the off-line generation of credible OCs and contingency scenarios using time-domain simulations. The method is generic, but is here tested on the Nordic32 test system with all data and models as presented in [108]. After a representative training set is generated, training of the LSTM network is performed. Each step in the methodology is described in the following subsections.

7.2.1 Generation of training data

The generation of a training set is a critical step and a range of different initial OCs and contingencies were included to generate a representative training set. Dynamic simulations were performed using PSS®E 34.2.0 with its built-in models [118]. The steps of generating the training data are illustrated as a flowchart in Fig. 7.1 and can be summarized as follows:

(i) *Randomly initialize OC*: For the Nordic32 system, the initial OCs were randomly generated around the operation point denoted as "operating point B" in [108]. A large number of possible OCs were simulated by randomly initiating the loads from a uniform distribution around the base case load levels (80 % of the original load as a lower limit and 120 % as an upper limit), while the power factor was kept constant. The total load change was distributed among the generators based on a weighted random distribution, where a higher rated capacity of a generator results in a higher probability to cover a larger share of the total load change. All generation that could not be supplied by the regular generators were distributed to the slack bus at g20, see Fig. 4.2.

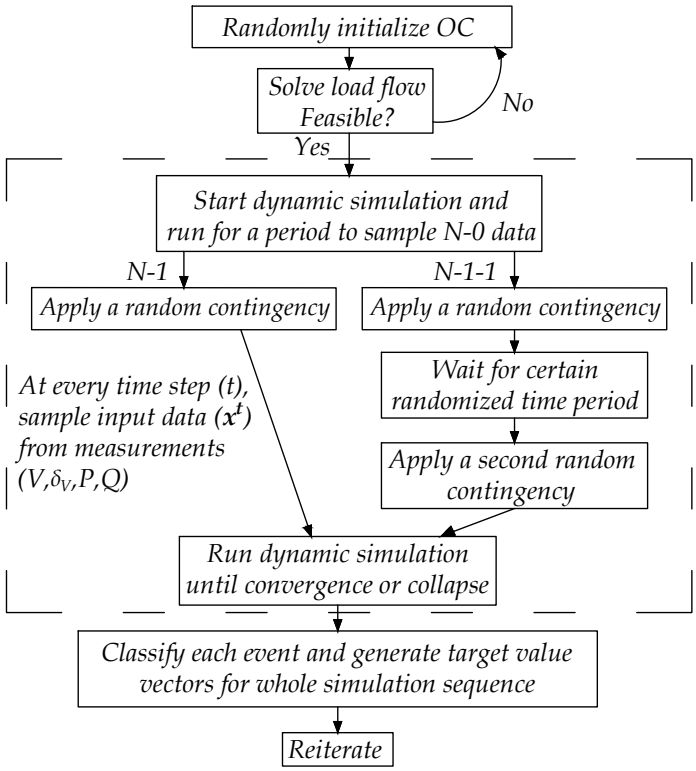

Figure 7.1: Flowchart for generating input data and target values.

In real applications, more delicate methods for efficient database generation and more careful generation of relevant OCs should be used [16, 126], where for instance the impact of unit commitment and topology changes are taken into account.

(ii) *Solve and check for feasibility*: The generated OCs were solved with a power flow simulator, which served as a starting point for the dynamical simulation. If the system load flow did not converge, the initial OC was re-initialized.

(iii) *Start dynamic simulation and introduce contingencies*: Two separate dynamic simulations were then initiated for the N-1 and the N-1-1 cases. The process is illustrated in Fig. 7.2. For each of the two cases, the system runs without any contingencies for 65 seconds to generate a sufficient amount of N-0 data for the LSTM network to train on. At $t = 66$ seconds, the *same* first contingency was applied to both of the cases. After an additional uniformly distributed random time in $[10-30]$ seconds after the first contingency, a secondary consecutive contingency was applied for the N-1-1 cases. Events resulting in several (near-)simultaneous contingencies were not taken into account (N-k events).

The considered contingencies in the simulations were either (i) tripping of a generator, or (ii) a three-phased fault during 0.1 seconds, followed by tripping the faulted line, which was then kept tripped during the remaining time of the simulation. The first contingency was chosen to be a major fault, meaning

a fault on any transmission line connecting the different main areas in the system (excluding the "Eq." area, see Fig. 4.2), *or* any larger thermal generator in the "Central" area. The second contingency, for the N-1-1 cases, included tripping of *any* transmission line in the whole system, excluding lines in the "Eq." area. No variations of load and generation were taken into account during the dynamic simulations as these, in the relatively short time of the simulations, are presumed to have a small impact on the system stability. In real settings, depending on the system and the experience of the operator, all relevant contingencies can be used in the training.

(iv) *Sample inputs and run until stopping criteria*: For each of the two cases, an input vector $\boldsymbol{x}^t$ consisting of measurements of all bus voltage magnitudes (V_{mag}) and angles (V_θ), active and reactive power flows (P_{flow}, Q_{flow}), were sampled every second ($\Delta t = 1s$) and saved in a data file. The value of $\Delta t = 1s$ is dependent on the possible measurement update rate in the actual system and will determine the rate the estimations are available to the system operator. No information regarding the type and location of applied the contingencies were sampled, as this information implicitly can be learned by the LSTM network. For instance, the LSTM network should be able to correlate a zero power flow in a transmission line with that line being out of service.

Each dynamic simulation ran for a total of 560 seconds, but was, in the case of a major voltage collapse, stopped in advance. The simulation interval of 560 seconds was chosen to allow time for *all* dynamic events to occur and for the system to either fully stabilize or collapse.

(v) *Classify each event and generate target value vectors*: For each case, a sequence of true target value vectors $\boldsymbol{y}^1, ..., \boldsymbol{y}^{560}$ was generated for every time step in the time-domain simulation. Each $\boldsymbol{y}^t$ in these sequences represents the classification of the system if the system is allowed to run from time t up until 560 seconds without any changes to the current system. As time progresses and new events occur, the class of $\boldsymbol{y}^t$ may change. The sequences consist of multi-dimensional vectors where the actual class is encoded using one-hot (binary) encoding.

The classification was performed according both to the severity and the location of the system degradation at the *end* of the time-domain simulation. The system was defined as stable if *all* transmission bus voltage magnitudes were above or equal to 1 pu, in an alert state if *any* transmission bus voltage magnitude ranged between $0.9 < V < 1.0$ pu, and in an emergency state if *any* transmission bus voltage magnitude was below 0.9 pu. Overvoltages were not taken into account.

The target values for the *alert* cases were also classified according to *where* the lowest bus voltage magnitudes were found at the end of each dynamic simulation. The Nordic32 test system is predefined into four different regions, namely: "Eq", "North", "Central", and "South" [108]. The regions "North", "South", and "Eq." were found to be stable regions, and no alert events were

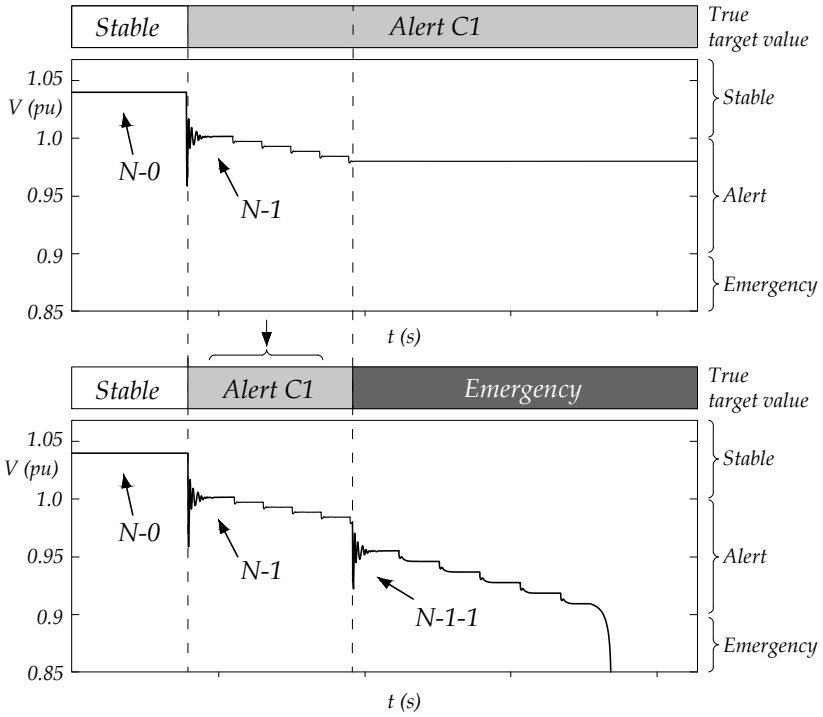

Figure 7.2: Example of classification of an N-1 and an N-1-1 case.

found in these regions for any of the simulated cases. To test the capability of the network to also indicate where instability emerges in the system, the "Central" area was divided into three separate regions (indicated by **C1**, **C2**, **C3** in Fig. 7.3). The classification for each time step of each simulation belonged then to one of 5 different possibilities. Either the whole system was predicted stable; it ended up in an emergency state; or an alert state was predicted in one of the three defined regions (**C1**, **C2**, or **C3**) where the *lowest* occurring transmission bus voltage was found.

The classification process is illustrated in Fig. 7.2. The target values are always classified as stable up until the first contingency. From different combinations of OCs and contingencies, the system may then end up being in a stable state, an alert state in area **C1**, **C2**, or **C3**, or in an emergency state. For the N-1 case, the sequence of true target value vectors from the time of the contingency to the end of the simulation are classified depending on which of these five states the systems end up in. For the example of the N-1 case in Fig. 7.2, the system ends up in an alert state in the **C1** area. For the N-1-1 case, the target values are classified as stable up until the first contingency. The target values are then gathered from the N-1 case, using the end state of that simulation for classifying the state *between* the first and the consecutive contingency. After the second consecutive contingency, the system runs until it either collapses or until 560 seconds. Depending on this final state, the sequence of true target value vectors from the second contingency until the

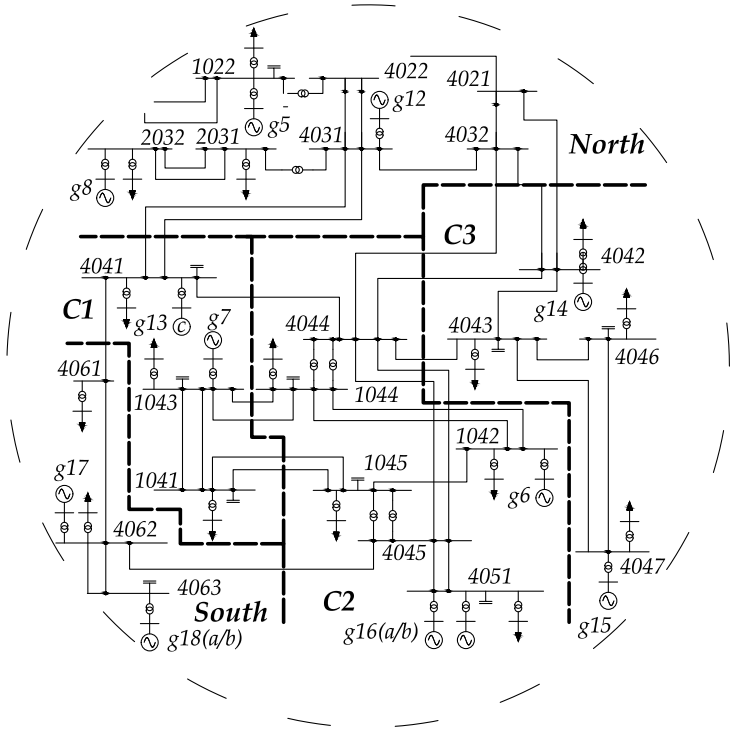

Figure 7.3: One-line diagram of Nordic32 system with subareas.

end of the simulation is classified. In the example in Fig. 7.2, an emergency state is reached. Note that the scales in Fig. 7.2 are different from those in the simulations for easier interpretation.

It should be noted that the classification of the different states (stable, alert, emergency) could be performed more intricately to satisfy other criteria of stability. For instance, these could be related to a minimum level of loadability of the system in its post-disturbance state. The loadability limit could then be computed by, for instance, parameterized continuation methods such as the CPF method [14], or by certain line indicators [71]. Other stability criteria could include the capability of the system in its post-disturbance state to handle yet another disturbance.

(vi) *Reiterate*: The described steps are reiterated until a sufficiently large training set is generated.

7.2.2 Architecture of the LSTM network

The proposed LSTM network architecture, shown in Fig. 7.4, is generally referred to as a "many-to-one" architecture, where previous measurements in the time sequence are used for the classification in the final block. The network consists of three stacked LSTM layers which are used to capture different levels of features from the inputs. Each LSTM block consists of 32 individual LSTM cells. The first layer of LSTM blocks takes a generated sequence of input vectors as inputs; then by mathematical

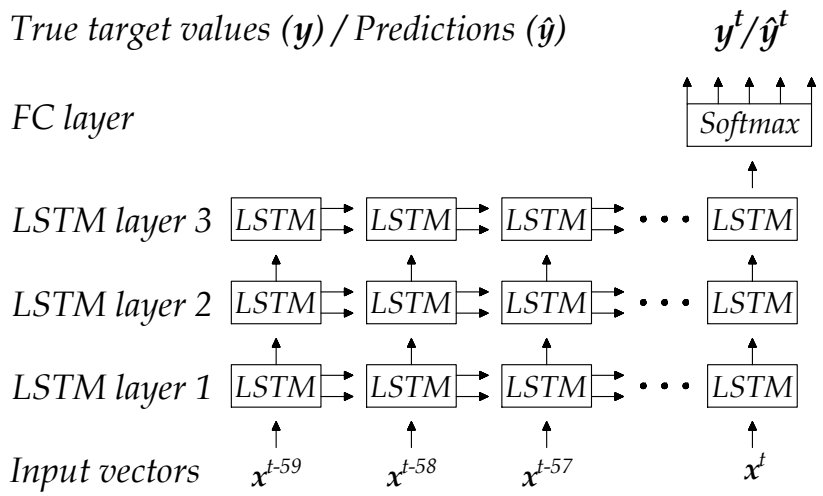

Figure 7.4: The proposed LSTM network architecture.

operation as presented in Section 3.2.6, the output of each block is forwarded both to the following block in the sequence, as well as to the upper layer of LSTM blocks. The inputs to the deeper layers consist only of the hidden states of LSTM blocks of previous layers, while both the hidden state and the cell state memory is passed along the time sequence between LSTM blocks of the same layer.

The LSTM network is designed to take sequences of 60 time steps of measurements as inputs. The internal architecture of each LSTM cell and functionality of the nonlinear gating units as presented in Section 3.2.6, allows the LSTM network to fully utilize and pass forward the information from the first to the final time step in the sequence. The third layer of LSTM blocks only passes the output forward along the time sequence. The output layer at time t is a fully connected network with softmax activation for classification. In training, the network uses the true target vector $\boldsymbol{y}^t$ at time t, while during the test or prediction phase, the network estimates a prediction vector $\hat{\boldsymbol{y}}^t$ at time t. The interpretation of the prediction problem is further explained in section 7.2.4.

7.2.3 Training the LSTM network

Different data sets were used for training, validation, and testing of the method on a mix of N-1 and N-1-1 cases. The training data set has the following dimension (135,000×364×560), where the dimension represents the number of training cases, the number of inputs, and the total interval in seconds for each simulation, respectively.

Before training, a process generally referred to as sequence preprocessing was performed to prepare batches of sequences with suitable length. The network is designed to take a sequence of 60 time steps of measurements as inputs and subsequences with a length of 60 time steps $(\boldsymbol{x}^{t-59}, ..., \boldsymbol{x}^t)$ were thus extracted from the 560 seconds long simulation intervals, for different values of t. For each subsequence of input

Table 7.1: Design and hyperparameters used in training.

	Parameter	Values and size
Data	Simulation interval	560
	Input dimension	364
	Input data type	V_{mag}, V_θ, P_{flow}, Q_{flow}
	Target classes	5
	Training cases (N-1+N-1-1)	45,000 + 90,000
	Validation cases (N-1/N-1-1)	5,000 / 10,000
	Test cases (N-1/N-1-1)	10,000 / 10,000
Architecture	LSTM layers	3
	LSTM sequence length	60
	FC activation function	Softmax
	LSTM hidden cells	32
	LSTM Activation function	Tanh
Training	Max Epochs	500
	Learning rate (α)	0.0001
	Dropout & recurrent dropout	50 % / 50 %
	Optimizer	Adam [124]
	Loss metric	Categorical cross-entropy

vectors, a corresponding target value ($\boldsymbol{y}^t$) at time t was gathered. The sequence preprocessing was performed 120 times for *each* training and validation case by varying t between values of $t = [60, 180]$. The lower bound of t is required to always allow historic data to be included into the sequence. The LSTM network could have been trained on the whole simulation interval by increasing the upper bound of t from 180 to 560. However, since the method is proposed to be used in fast VIP applications, there is less usefulness of predicting instability long after the contingencies have occurred.

The generated subsequences were then used to train the LSTM network. Due to memory limitations, a method called mini-batch gradient descent was utilized where mini-batches of 1000 subsequences were used separately to train the network. The training was performed for a maximum of 500 epochs. An epoch is finished when all generated batches have been used to update the network parameters. An adaptable algorithm suitable for gradient-based optimization of stochastic objective functions, more commonly known as "Adam" was used in training the network [124]. The algorithm used default parameters according to [124], except for the learning rate which was tuned. The loss function on which the optimizer is applied is the categorical cross-entropy function, which is suitable for multi-classification problems. To avoid overfitting the data, two regularization techniques were used during the training. First, early stopping was implemented, and the training of the network was stopped in case the performance on the validation set did not improve after six epochs. Second, a technique called dropout was applied, where a certain percentage of the connections between inputs and the LSTM cells were randomly masked (or "dropped") to reduce overfitting on the data. Both conventional dropout and recur-

rent dropout between consecutive blocks were applied during the training phase.

All other parameters related to the training of the network are presented in Table 7.1. The LSTM network was trained and implemented in Python, using the Keras library with TensorFlow backend. The architecture and parameters used to train the network have been iteratively tuned to increase the classification accuracy. It should be noted that the tuning could be extended even further to allow an even better classification accuracy.

7.2.4 Interpretation and intuition of the VIP problem

By the proposed training and architecture of the LSTM network, a classification problem is solved where the *current* system state space is separated into different regions. Every state on a trajectory to a stable, alert (in **C1**, **C2**, or **C3**), or emergency state is labeled accordingly. The LSTM network is then trained on this data to implicitly learn these asymptotic properties of solutions and the trajectories of the system state. Once trained, the network can correlate the inputs, current and historic measurements, with a certain state-space region and trajectory, allowing warnings of voltage instability only moments after a contingency have occurred in a system. The classification is performed under the assumption that the current system is unchanged, meaning that no additional contingencies or changes in generation and load configuration will occur. However, as time progresses, new observations are used as inputs to the LSTM network to continuously update and incorporate such changes in the system.

This VIP problem should be interpreted as a fixed horizon prediction problem, where the prediction horizon always is the final state given by the trajectories of the (dynamical) system. This interpretation assumes that the simulation horizon of the generated time-domain simulations are sufficiently long so that extending the simulation horizon even further, for this particular system beyond 560 seconds, would not change the partitioning of the state space.

7.3 Results

In this section, the classification accuracy of the LSTM network is presented for two different test sets, one containing only N-1 cases, the other containing N-1-1 cases. Each test set was composed of $10,000$ cases of dynamic simulations. The test results of the predictions are presented using categorical accuracy, where the *indices* of the true target values are compared to the argument maxima of the predictions. The accuracy at *each* time step is then calculated over time for each of the two test sets.

The data were fed into the network in the form of a rolling window, with subsequences generated in the exact same manner as described in Section 7.2.3. As time t progresses, new measurements entered the network from the rightmost block in the input layer and were shifted to the left in each time increment. Since the LSTM network require a sequence of 60 time steps of data, no predictions were made before $t = 60$. To facilitate the presentation in the following figures, a new time index T is

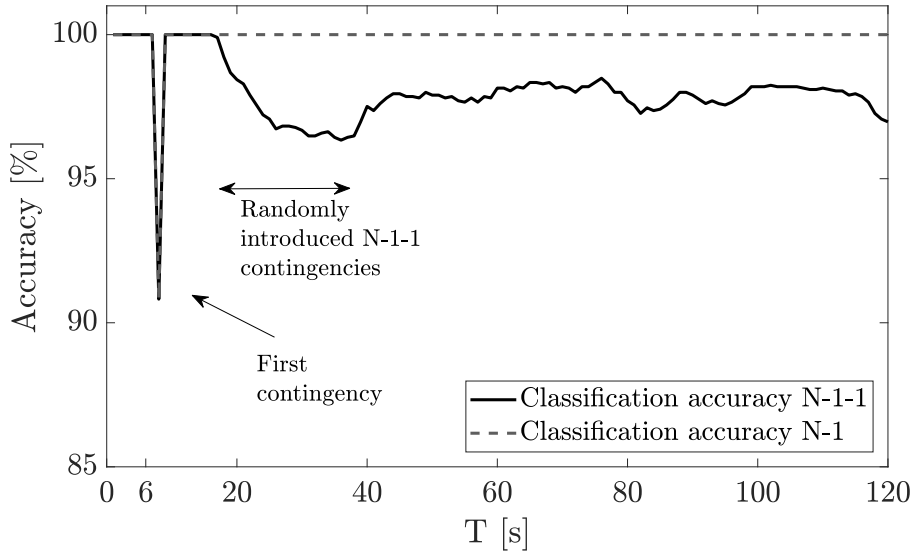

Figure 7.5: Classification accuracy over time for the proposed LSTM network.

introduced here. The relationship between the two time indices is $T = t - 60$. The LSTM network's performance for VIP is not only tested during the short "just after disturbance" state but during a longer period of the dynamic trajectory the system takes following the disturbances. This is performed to test the network's capability to incorporate new observations and improving its assessment as time progresses throughout a voltage instability event. The classification accuracy is only plotted for 120 seconds after T to better visualize the changes in accuracy after the contingencies.

The classification accuracy *over time* is presented in Fig. 7.5. The classification accuracy for the N-1 test set dropped significantly at $T = 6$ seconds, which is the same instant that the first contingency is applied. The large drop in classification accuracy can be attributed to low bus voltages instantaneously following the first contingency, which the LSTM network has learned to correlate to a voltage instability event. The large drop in accuracy only remained for a single measurement point. After the first contingency, the classification accuracy increased and remained constant at 100 % for the rest of the simulations. The classification accuracy for the N-1-1 test set was identical up until the time when the consecutive contingencies were randomly applied. During this time, illustrated by the arrows in Fig. 7.5, the classification accuracy decreased slightly. Since these contingencies do not occur at the same time instant in each test case, the same instant drop in accuracy as for the N-1 cases was not seen. The accuracy then gradually increased and stabilized at around 97-98 %.

The results show that the LSTM network can classify and predict future stability almost perfectly for the N-1 contingency cases and with good accuracy for the N-1-1 cases. To examine which cases were misclassified, the prediction accuracy for the two test sets, evaluated at $T = 50$ seconds, are presented in Table 7.2 in the form of a

confusion table. Each number in the column in the table represents instances of the predicted classes and each number in the row represents the instances of the actual classes. The (empirical) conditional probabilities of correctly classifying a certain state is presented in the column furthest to the right. Similarly, the conditional probability of a state *actually* belonging to the predicted state is presented in the bottom row of the table. The total accuracy is presented in the lower right corner of the table. The accuracy for all N-1 cases is 100 % and no cases are falsely classified. For the N-1-1 test set, the lowest classification accuracy occurred for the alert states. After inspection of the falsely classified cases, it was found that several of these were borderline cases where the transmission bus voltage magnitude used in the classification were very close to what was used in the other classes. The highest classification accuracy occurred for the emergency cases with 99.8 %.

It should be noted that the test and training sets were weighted with more cases ending up in certain classes than others. It is thus probable that the results are slightly biased with higher accuracy for these classes, and that the classification accuracy of the other classes may be lower as an effect.

7.4 Sensitivity analysis

In this section, a sensitivity analysis of various hyperparameters and other aspects that will affect the performance of the LSTM network is presented. First, the impact of the sequence length of the LSTM network is examined and compared to that of a conventional NN. The impact of different measurement update rates and is then examined, followed by a study of the network's generalization capability.

7.4.1 Impact of sequence length

In this section, the performance of the sequence-based approach is tested and compared against a conventional feedforward NN, which only uses a single snapshot of measurements as inputs. Further, to test the impact of a *shorter* time sequence, the results of an LSTM network using a time sequence of 30 time steps, instead of 60, are presented.

To allow a fair comparison between the two approaches, the feedforward NN used in this comparison was designed to be as similar as possible to the LSTM network. Essentially, the design of the NN in the comparison is identical to the *final* time step in the LSTM network presented in Fig. 7.4, with the difference that each layer consists of a hidden layer of neurons. The designed NN has thus three hidden unit layers, each layer with 32 hidden nodes. The same FC layer with a softmax activation function was used. The training for the NN was performed identically as for the LSTM network, with the exception that instead of a sequence of input values, a single snapshot was used. The LSTM network using a shorter time sequence was trained identically to that of the longer LSTM network with the exception that a shorter sequence of 30 instead of 60 time steps was used.

In Fig. 7.6, the classification accuracy on the N-1-1 test set is presented for the two

Table 7.2: Confusion table showing prediction results and accuracy of the LSTM network evaluated at $T = 50$ seconds.

		Predicted states (N-1 / N-1-1)					
		Stable state		**Alert state**		**Emergency state**	**Accuracy**
Classification		All areas	C1	C2	C3	All areas	
Actual states **Stable state**	All areas	2,766 / 1,147	0 / 36	0 / 11	0 / 8	0 / 8	100 / 94.8 %
Alert state	C1	0 / 0	856 / 562	0 / 3	0 / 0	0 / 5	100 / 98.6 %
	C2	0 / 5	0 / 5	1,874 / 1,222	0 / 0	0 / 109	100 / 91.1 %
	C3	0 / 0	0 / 0	0 / 12	0 / 208	0 / 0	- / 94.5 %
Emergency state	All areas	0 / 0	0 / 0	0 / 10	0 / 0	4,504 / 6,649	100 / 99.8 %
Accuracy		100 / 99.6 %	100 / 93.2 %	100 / 97.1 %	- / 96.3 %	100 / 98.2 %	**100 / 97.9 %**

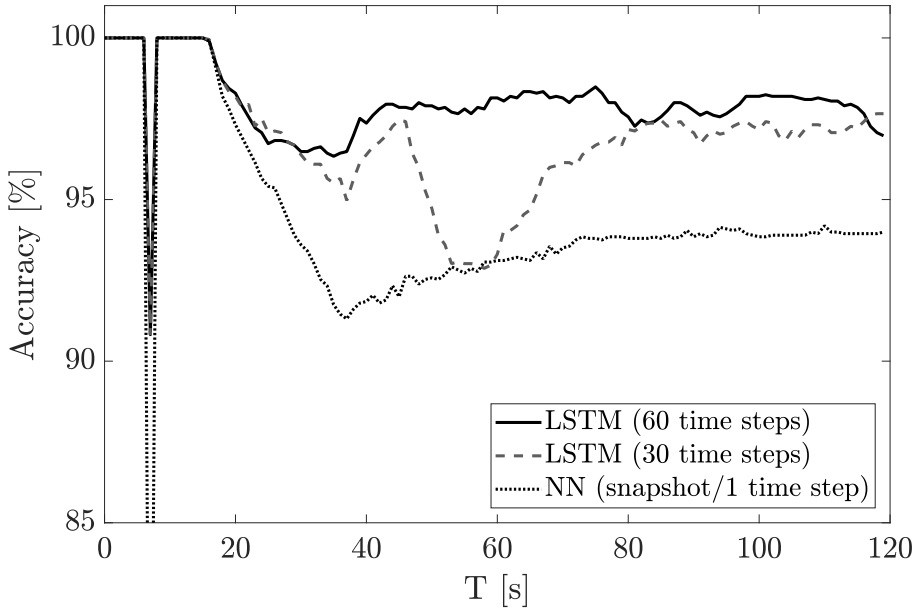

Figure 7.6: Impact of sequence length on classification accuracy.

LSTM networks with the different time sequence length and for the conventional NN. The classification accuracy for the conventional NN was around 93 % after all the consecutive contingencies had been applied, while that of the proposed LSTM network is around 97-98 %. The results show that the performance of the LSTM network using 60 time steps in the sequence significantly exceeded that of the conventional NN, generally providing better classification accuracy over the whole time frame of the simulation cases.

The classification accuracy of the LSTM network using a shorter sequence was similar to the one using a longer sequence, with the difference of a large drop in classification accuracy occurring at around $T = 46$ seconds, see Fig. 7.6. The accuracy declined for 20 seconds and was then restored to around 97 % accuracy. A similar decline in classification accuracy, though less significant, can be noted for the LSTM network using the longer time sequence at $T = 76$. Thus, a decline in classification accuracy started exactly 60 respectively 30 seconds after the consecutive contingencies were introduced (at $T = 16$) for the two networks, corresponding to the network's respective sequence length. One explanation of these results is that the LSTM networks utilize information concerning the contingency and *pre-contingency* state to enhance the classification accuracy. When the networks starts to lose the information about the pre-contingency state, the chance of a misclassification increases. The results strengthen the hypothesis that a long sequence LSTM network could be used to enhance the state signal to provide better classification accuracy. Theoretically, an even longer sequence could be used to increase the accuracy even further. However, this would increase the computational cost of training, and a balance between classification accuracy and computational cost should be sought.

7.4.2 Impact of measurement update rate

The performance of the LSTM network is in this section tested for different values of the measurement update rate. The performance is compared between the previously assumed available measurement update rate of $\Delta t = 1s$ and the slower update rates of $\Delta t = 3s$ and $\Delta t = 5s$. Due to the slower update rates, the architecture and the number of LSTM blocks along the time sequence had to be reduced accordingly. The original LSTM network was designed to take subsequences of 60 time steps of measurements as inputs. Thus, for the LSTM network adapted for $\Delta t = 3s$, the number of LSTM blocks along the time sequence was reduced to a third (20 blocks along the time sequence), while the number of blocks for the LSTM network adapted for $\Delta t = 5s$ was reduced to a fifth (12 blocks along the time sequence). The LSTM networks adapted for the new measurement update rates were then trained identically to the original LSTM network, with the difference that now only every third, respectively fifth, measurement in each generated subsequence were passed on the networks.

The classification accuracy for the different values of Δt is presented in Fig. 7.7 using the same N-1-1 test set as in previous sections. The results show that the performance when using a measurement update rate of $\Delta t = 1s$ exceeds those using a slower update rate. The largest difference can be identified during the period when the second consecutive contingencies are applied, which indicates that a lower value of Δt is especially valuable for classification during the short time that follows a disturbance. It should be noted that due to the slower update rates of Δt, there is no dip in the classification accuracy following the first contingency.

A larger value of Δt may also increase the *time* it takes to accurately predict instability, as new measurements are being passed less frequently to the LSTM network. In Table 7.3, the average time, after a contingency, to accurately predict the future state of the system is presented for the different values of Δt. The average time is only presented for the time it takes to correctly classify the system states following the *second* consecutive contingency, since correct classification following the first contingency was almost instantaneous in all test cases. The time was computed as the averaged passed time after the second contingency, up until the time when the LSTM network could consistently and accurately predict the state of the system. For cases that took *longer* time than 100 seconds to be correctly classified, a detection time of 100 seconds was assumed to avoid skewed averaged values.

The average time to correctly predict the system state was found to be 6.6 seconds for the proposed LSTM architecture using a measurement update rate of $\Delta t = 1s$. The corresponding values for the LSTM networks using the slower update rates of $\Delta t = 3s$ and $\Delta t = 5s$, were 8.7 seconds and 10.7 seconds, respectively. The longer time longer time to accurately predict instability for the slower update rates of Δt can be attributed partly to a lower classification accuracy, and partly to the fact that measurements are being updated less frequently.

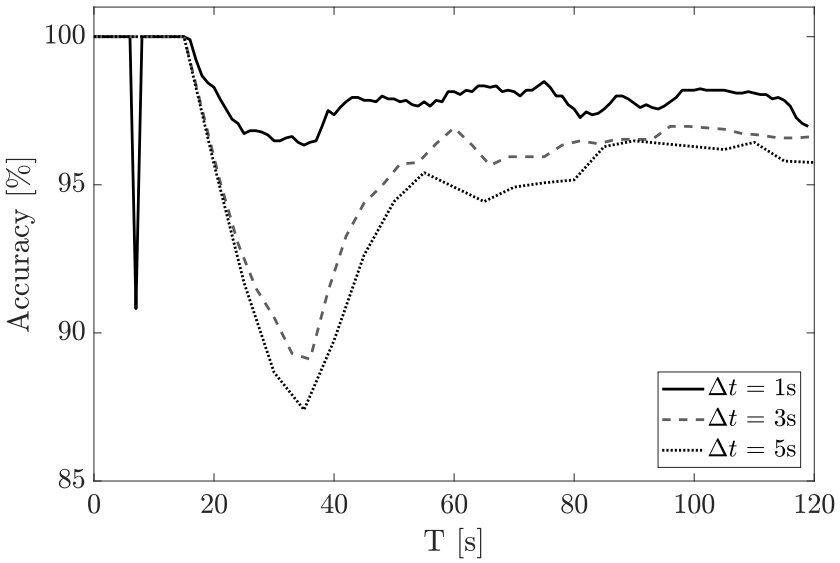

Figure 7.7: Classification accuracy over time for different values of Δt.

Table 7.3: Average time to predict the onset of voltage instability.

	Measurement update rates		
	$\Delta t = 1s$	$\Delta t = 3s$	$\Delta t = 5s$
Average prediction time [s]	6.6	8.7	10.7

7.4.3 Generalization capability and training set requirement

The generalization capability of a ML method refers to the capability to generalize the learning from the actual training set to other, yet unseen, cases. Such capability is especially valuable in overcoming the combinatorial increase of complexity in the training when N-1-1 cases are also considered [127].

In Fig. 7.8, the classification accuracy is presented on the N-1-1 test set when the LSTM network have been trained on three different training sets. The results are presented when the network was trained on i) the full training set with all N-1 and N-1-1 cases included, ii) a smaller training set with all N-1 cases but where only a small batch $(5,000)$ of N-1-1 cases have been included, and iii) a training set where the network is *only* trained on N-1. The same training approach as previously described were used. According to Fig. 7.8, the classification accuracy was significantly reduced when no N-1-1 cases are included in the test set. When including the small batch $(5,000)$ of N-1-1 cases, the classification accuracy increased significantly. However, the accuracy is still lower than when the full training set is used. Thus, the importance of obtaining a representative training set is still imperative if a high classification accuracy is to be achieved.

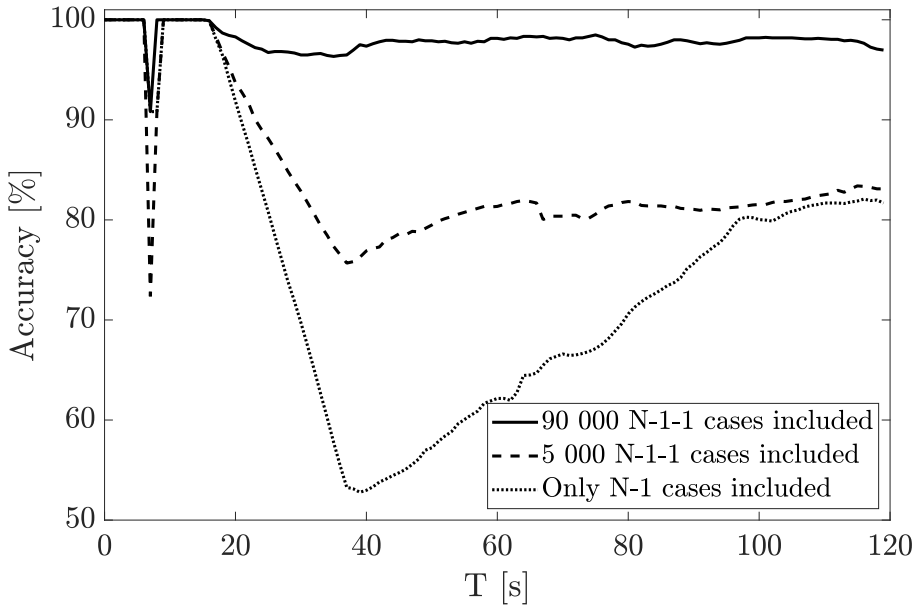

Figure 7.8: Classification accuracy over time when varying the number of N-1-1 cases included in the training data.

7.5 Practical applications and requirements

The method is proposed to be used as an online tool for system operators to predict the system's future stability given the current state. It should be stressed that the method is not proposed to replace conventional VID methods, but rather function as a supplementary tool to provide early warnings. The instantaneous prediction capability of the proposed method has to be weighed against the possibility of misclassification of the system's future stability. When comparing the proposed method to other conventional indicators for VID (see [43]), it is important to remember that these might be more accurate once instability detected, but generally take *significantly* longer time to indicate instability, thus reducing the time frame that system operators have to steer the system back into stable operation.

The proposed method is mainly intended for predicting mid-term or long-term voltage instability where system operators will have the possibility to act on the warnings provided by the network. Theoretically, the method could be adapted to also handle short-term voltage instability. However, this would require more frequent measurement updates to ensure that the onset of short-term instability is detected in time. Because of the difference in the dynamical trajectories of the system for the two different types of instability events, training a separate LSTM network would likely provide better performance. Furthermore, the signals provided by the network would have to automatically trigger emergency controls, since the available time for system operators to act on the signals would be too short for manual control actions.

For the proposed method to be effective in prediction of long-term voltage instability, measurement updates should be available within a few seconds. Here, a measurement update rate each second have been assumed to be available. As was found in Section 7.4.2, slower measurement update rates lead both to lower classification ac-

curacy and slower predictions. To assure that errors and missing values are filtered out, measurements should always be preceded by a state estimator. However, state estimates from a non-linear state estimator based on remote terminal units may be too slow to be effective. Thus, time-synchronized measurements from wide-area phasor measurements filtered through a linear state estimator would be preferred.

The softmax classifier of the LSTM network outputs a probability vector, where each class is given a certain probability. It should be noted that this probability vector does not provide a *true* representation of the model confidence. However, it can still be useful as a proxy by system operators to track the network's confidence in each prediction. Thus, the operator can use the probability vector directly in an online interface to track the network's belief in each prediction. Alternatively, argument maxima or other functions could be used to present the most probable prediction of the network, or, for instance, to avoid predictions of falsely labeled stable states.

The *practical* classification accuracy of the proposed method will be affected by many aspects and will generally be lower than on a simulated test set. One of the more important aspects are modeling errors, including erroneous system parameters or inaccurate modeling of parameter values for dynamic models. Such aspects will introduce a difference between the simulated and the actual dynamic response after a contingency.

Chapter 8

Security margin control using deep reinforcement learning

This chapter develops a real-time control method based on deep reinforcement learning aimed to determine the optimal control actions to maintain a sufficient secure operating limit. The results and the developed methodology were established in paper II. The control method is proposed to be included as a preventive control method in the developed real-time dynamic stability assessment and control tool, earlier presented in Chapter 4.

8.1 Introduction

Maintaining a secure operation of an electrical power system is a major aspect of power system operation. To be able to optimally activate the correct preventive control measures whenever the security of the system is threatened can both increase the operational efficiency of an electrical power system, as well as allow an operation closer to the actual limits of the system. Previous implementations of DRL in electric power system control have mainly been focused on emergency control, which aims at controlling the system back into a stable state after a disturbance has already occurred [56]. Typically, preventive control problems such as maintaining a certain security margin have been defined as static optimization problems. DRL is better suited to handle sequential decision-making problems but with relatively small adjustments, preventive control problems can easily be adjusted to fit a sequential decision problem where the advantages of the DRL framework can be utilized.

While previous DRL implementations in electric power system control have achieved good performance on the presented test systems, some limitations in terms of practical control remain. To achieve efficient control, system operators are typically required to simultaneously control both discrete (e.g., switching of a shunt capacitor) and continuous (e.g., the level of active power generation rescheduling and load curtailment) action variables. However, state-of-the-art DRL algorithms such as deep Q-networks, or deep deterministic policy gradients are generally designed

to only control *either* discrete *or* continuous action variables, and all previous implementations of DRL in electric power system control have been reduced to rather simple control schemes where only a single type of action space has been considered.

In this chapter, the lack of preventive control methods is addressed and a new method based on DRL aimed at determining the optimal control actions to maintain a sufficient SOL is introduced. Furthermore, a hybrid control scheme, which is capable of simultaneously adjusting both discrete and continuous action variables is presented. Several aspects of the DRL control, including robustness to different simulation scenarios and noise in the inputs, are evaluated. From here on, the control agent used in controlling and maintaining a sufficient SOL is referred to as the *hybrid DRL agent*, due to its capability of handling multiple action spaces simultaneously. The methodology and test results for the proposed method are presented in the following sections.

8.2 Framework for a real-time control of secure operating limits

In this section, the framework used in training the real-time SOL control algorithm is presented. The algorithm works by continuously monitoring the current state of the system through a set of measurements. If the hybrid DRL agent assesses that the system is not secure, or has a too small margin towards the security boundary, actions are initiated to steer the system to a more secure state. Other types of control, such as maintaining system voltages within acceptable limits, are not included in the developed control, but could theoretically be added as additional control goals. The hybrid DRL agent is trained on different loading scenarios where the security state of the system is varied to reflect the different operating conditions that may occur in a real power system.

8.2.1 Test system and security margin definitions

All simulations have been tested on the slightly modified version of the Nordic32 test system, detailed in Chapter 4 and again illustrated in Fig. 8.1. The security margin that the DRL agent attempts to control and maintain is based on the SOL. In Chapter 5, the SOL was shown to provide a more accurate measure of the security margin than other methods that are based on static assessments of the system, especially when the system is characterized by a larger share of loads with fast restoration dynamics.

The system stress used to compute the SOL is typically defined as a combination of load demand increase and/or generation rescheduling, which are quantities that the system operator can observe and control in the pre-contingency state. The SOL is then computed with respect to a set of credible contingencies. The SOL is in this study determined by a search process similar to the "binary search" proposed in [13], which is slightly differing from the "dual binary search" method earlier presented in Chapter 6. The process is illustrated in Fig. 8.2 and is based on searching through a narrowing interval by iteratively testing the system security with respect to a

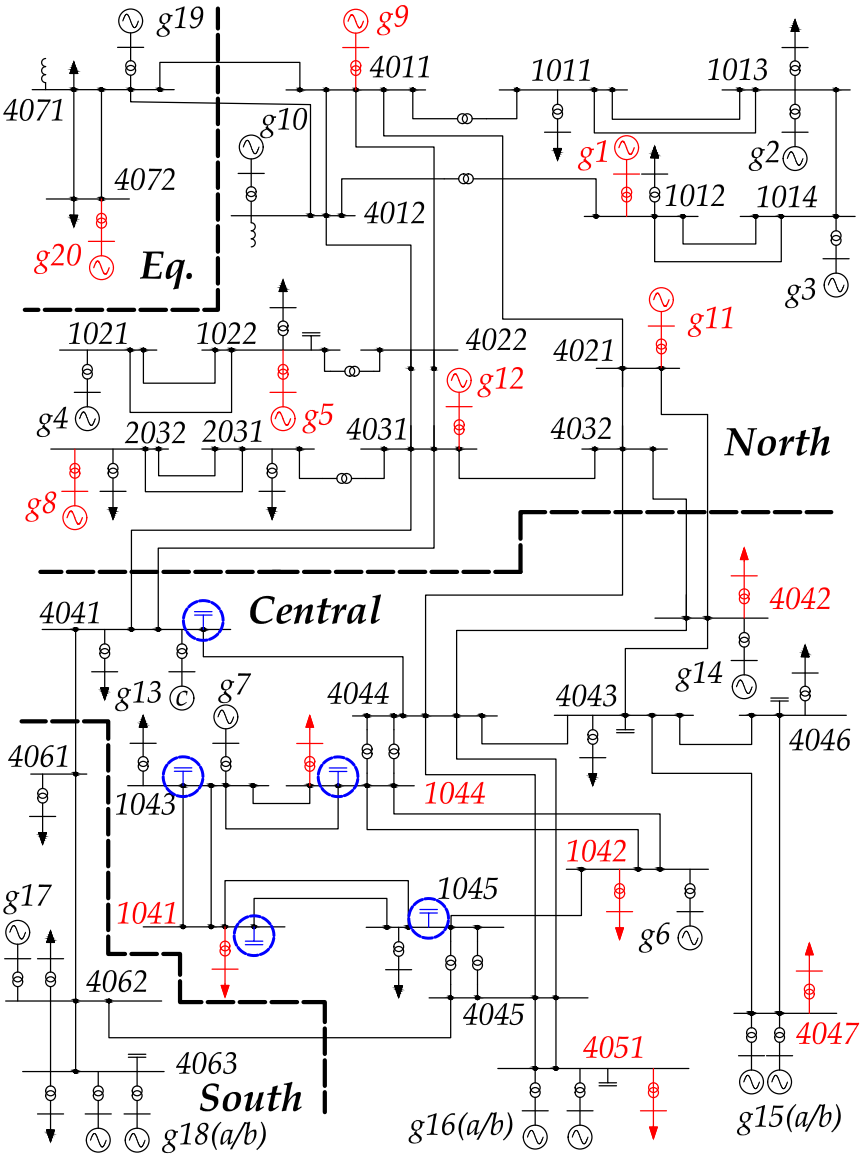

Figure 8.1: One-line diagram of the modified Nordic32 system [114]. Loads and generators included in the continuous action space are marked in red, while shunt capacitors participating in the discrete action space are marked in blue.

dimensioning fault and different levels of system stress. The upper part in Fig. 8.2 illustrates the search process when the state was not secure, while the lower part in Fig. 8.2 illustrates the search process when the state was secure. Black dots indicate a secure state, while white dots indicate a state that is not secure. The search process is exemplified for a secure starting state. If the starting state is found secure, the system stress is increased by a total of ΔP. If the new state was found to be secure, the system stress was again increased with ΔP. In case it was not found to be not secure, the system stress was instead reduced by $\Delta P/2$. The search process continued until a secure operating point was found and when the step size in system stress change was sufficiently small.

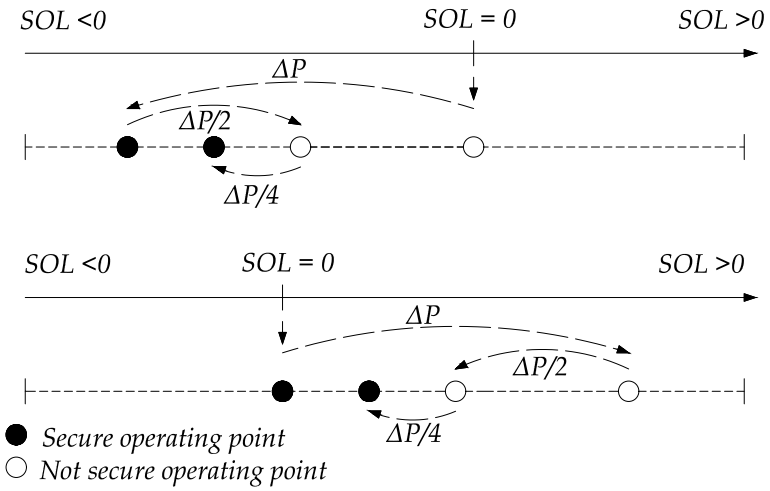

Figure 8.2: Illustration of the search process for the SOL for a secure and a non-secure initial operating point.

In the following simulations, the system stress is achieved by increasing the loads in the "Central" area of the modified Nordic32 test system, while simultaneously adjusting the generation in the "North" area with the same amount. The power factors of all loads were kept at their initial values and the distribution of the added load and generation were scaled by the initial load or the rated capacity of each generator. The dimensioning contingencies of concern will be the three largest generators that are located in the "Central" area, namely: generator *g14*, *g15*, and *g16*. The contingency resulting in the *lowest* SOL is always dimensioning for the system. An initial step size of $\Delta P = 128$ MW was used in the search process for the SOL and the process was stopped whenever the step size in system stress is equal to 1 MW.

A relatively simple stability criterion was used to determine whether a state was secure or not. The system was considered secure if, at the end of the post-contingency evaluation, *all* transmission bus voltages were above 0.90 pu. Although the modified Nordic32 test system is characterized by sensitivity towards long-term voltage instability, other types of instability can violate the defined stability criterion. For instance, transient angle instability can cause locally low voltages due to lost synchronism of certain generators. Frequency stability has not been included in the analysis but has been assumed to be stabilized by automatic frequency control actions of generators after a disturbance. All dynamic simulations ran for a maximum of 600 seconds but were stopped in advance if the case either collapsed (any bus voltage below 0.7 pu) or if the system stabilized early. This approach should ensure that the system had either stabilized or become unstable at the end of each simulation.

8.2.2 MDP formulation for security margin control

The SOL control problem is defined as an episodic MDP. At the beginning of each episode, the DRL agent receives a representation of the system state s_t through a set of measurements. Depending on the current policy π_θ, the DRL agent picks

different actions which are activated in the system. The taken actions and the transition dynamics distribution causes the system state to change ($s_t \rightarrow s_{t+1}$) and give rise to a reward R_t. If the DRL agent managed to restore the SOL above the defined threshold, or if a maximum number of time steps have been reached, the episode ends. Otherwise, the DRL agent continues observing new states, actions, and rewards. The MDP for the SOL control problem is defined as:

- *States*: a state vector s_t consisting of measurements of the bus voltage magnitudes of all buses in the system and, active and reactive power flows on all transmission lines and transformers. The current time step t of the episode was also added to the state vector. Neural networks are sensitive to input perturbations so to make the DRL agent more robust toward such errors, all state values were also randomly perturbed by multiplying each value with a random number sampled from a normal distribution with a mean of 1 and a standard deviation of 0.001. All states were then normalized by subtracting the mean of each state value and then dividing its standard deviation. The mean and standard deviation of each state value was computed from previously sampled states and a list with a maximum of 10 000 historic states was stored. Once 10 000 states were added to the list, the mean and standard deviation used for normalizing states became fixed.

- *Actions*: the DRL agent can activate either continuous and/or discrete action variables. The continuous actions were used to reschedule power generation and perform load curtailment to reduce the transfer of power through the system. This was achieved by reducing the active load in the "Central" area at certain participating load buses: *4042*, *4047*, *4051*, *1041*, *1042*, *1044*, while simultaneously increasing the generation in the "North" area at certain participating generators: *G1*, *G5*, *G8*, *G9*, *G11*, *G12*, *G20*. The active load decrease was distributed on each of the participating load buses based on their initial load *before* the change, while the power factors of all loads were kept constant. The discrete action variables included the switching of an additional 100 MVAr of reactive power support from any, or several, shunt capacitors. The discrete actions (D_1-D_5) controlled the shunt capacitors at the following buses: D_1: *1041*, D_2: *1043*, D_3: *1044*, D_4: *1045*, D_5: *4041*. The participating buses and equipment for the continuous and the discrete actions are all marked in red respectively blue, in the line diagram in Fig. 8.1.

- *State transition*: The state transition dynamics are deterministic and governed by a set of differential and algebraic equations used to build the dynamic model in PSS®E [118].

- *Rewards*: The reward R_t for the taken actions was computed by a combination of the resulting SOL and the costs for the continuous (C_{cont}) and the discrete (C_{disc}) actions, respectively. *Any* activation of the discrete actions contributed to a negative reward of -5, representing the cost of the mechanical wear that is involved in switching the shunt capacitors. The cost for the continuous actions contributed to a negative reward of -0.1 per adjusted MW in the power transfer.

Changing a total of ± 200 MW would thus result in a negative contribution to the reward of −20. This negative reward reflects the system cost of market adjustments or load/generation curtailment. The control goal is to always restore the SOL to a value equal to or above 30 MW, which would ensure that a sufficient security margin is achieved and that the N-1 contingency criterion always would be satisfied with some margin to account for possible inaccuracies. If the SOL was below 30 MW, a negative reward of -50 was added to the total reward for that time step. The final reward when accounting for the resulting SOL was then computed as:

$$R_t = \begin{cases} C_{cont} + C_{disc} + \text{SOL} - 50, & \text{if SOL} \leq 30 \text{ MW} \\ C_{cont} + C_{disc}, & \text{otherwise} \end{cases} \tag{8.1}$$

In this study, the reward is unitless, but should in real applications reflect the actual monetary cost of different actions and the corresponding rewards when the control goal is either achieved or missed.

8.2.3 Training data generation

An overview of the steps involved in training data generation and the training of the DRL agent is illustrated in Fig. 8.3. The different steps are detailed in the sections below. To speed up training, the data generation was parallelized and multiple CPU cores were used to generate data.

1. *Generate initial operating condition*: A large range of different initial OCs was generated to serve as training data for the algorithm. All loads in the system were randomly and individually varied by multiplying the active load value with a random variable generated from a uniform distribution (80 % of the original load as lower limit, 130 % of the original load as upper limit). The power factors of all loads were kept constant. The total change in loading was distributed proportionally among all the generators in the system based on the initial active power produced by each generator. The generated initial OCs were solved by a full Newton-Raphson load flow and were re-initialized in case the load flow did not converge.

2. *Sample state s_t and action a_t from the policy and move to next state*: Once an initial OC was generated, the state s_t was sampled from the system and passed to the actor network. The actor network outputs parameters that form the current hybrid policy $\pi_\theta(\mathbf{a}|s)$ from which a set of actions were sampled. The actor network and how it is used to form the hybrid policy is further detailed and discussed in Section 8.2.4. Once the actions were sampled and activated in the system, a new Newton-Raphson load flow was computed which formed the state transition from $s_t \rightarrow s_{t+1}$.

3. *Compute the SOL*: Once the load flow for the new state was solved, the SOL was computed following the steps defined in Section 8.2.1.

4. *End episode and save transitions*: The episode was ended if either the SOL

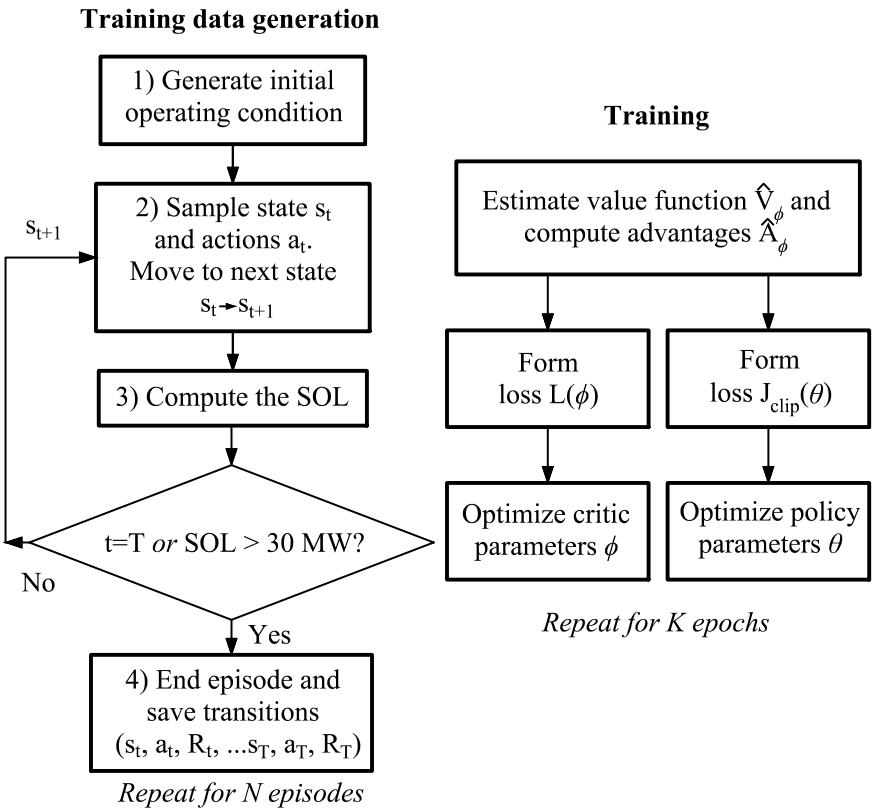

Figure 8.3: Flowchart showing the generation of training data and the training of the actor and critic network.

was restored above 30 MW (the defined security margin for when the system was assumed to be secure) or if the current time step was equal to T=8. At the end of all episodes, the transition data gathered during the episode (s_t,a_t,R_t,...,s_T,a_T,R_T) was stored and later used during training. The training data generation was reiterated for a total of $N = 64$ episodes before it was trained on the generated data.

8.2.4 Architecture of actor and critic network

The actor network is illustrated in Fig. 8.4 and further detailed in Table 8.1. It shares a common hidden layer, then a separate hidden layer is used for each type of activation function. The network outputs parameters used in defining the Normal distribution (3.23) and the Bernoulli distributions (3.24) that are used for the continuous and discrete action variables, respectively. The Normal distribution is parametrized by a mean value μ_{cont} and a standard deviation σ_{cont}, while the Bernoulli distribution is parametrized by a single probability parameter ranging from 0-1. The mean value μ_{cont} is computed using a linear activation function in the final layer, and the standard deviation σ_{cont} is computed using a Softplus activation function that ensures that the value never becomes negative, while the Bernoulli distribution is achieved using a sigmoid activation function in the final layer that ensures that the output is bounded between 0 and 1.

The critic network is separate from the actor network and consists of a simple NN

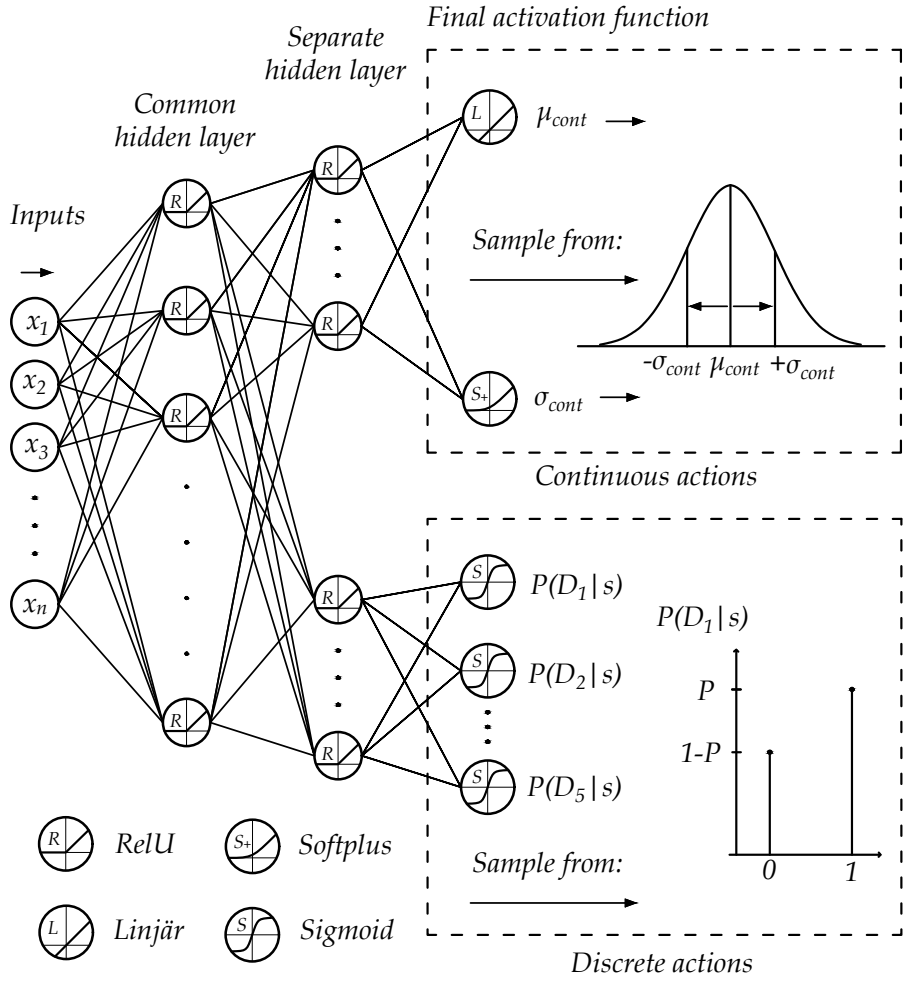

Figure 8.4: Architecture of the actor network.

with two hidden layers and a linear final activation function, further detailed in Table 8.1.

8.2.5 Training actor and critic networks

Once a batch of training data was sampled, the actor and critic networks were trained. The training was performed using the software Tensorflow in Python which automatically computes the gradients on the defined cost functions. The critic network was first used to estimate the value function $\hat{V}_\phi^\pi$ of each state. The returns G_t^γ were estimated using (3.12) and the advantage estimates $\hat{A}_t$ were estimated using (3.17) and the estimated value of the state $\hat{V}_\phi^\pi$ given by the critic network. A γ of 0.99 was used to compute the returns. The final objective function for the actor network was computed by taking the mean value of all samples for all N episodes on J^{clip}. The critic objective function was defined as the mean squared error (MSE) of the value function error, computed from (3.19). The final critic objective function was formed by taking the mean value of all samples for all N episodes on the MSE of the value function error.

Table 8.1: Design and hyperparameters used in training

		Parameter	Values
Architecture	*Critic*	Number of inputs	499
		Neurons in first layer	128
		Neurons in second hidden layer	64
		Final activation function	Linear
		Hidden layer activation	RelU
	Actor	Number of inputs	499
		Neurons in common hidden layer	128
		Neurons in each separate hidden layer	64
		Final activation for μ_{cont}	Linear
		Final activation for σ_{cont}	Softplus
		Final activation for $P(D_x\|s)$	Sigmoid
		Hidden layer activation	RelU
	Training	Max Epochs (K)	10
		PPO clip parameter (ϵ)	0.2
		Discount factor (γ)	0.99
		Optimizer	RMSprop [128]

Table 8.2: Learning rates over training iterations.

	α_{actor}	α_{critic}
Training iteration$\leq$ 250	$1 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
Training iteration $>$ 250	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$

Once the actor and critic objectives were formed, they were optimized using the RMSprop algorithm, which is an adaptable algorithm suitable for gradient-based optimization of stochastic objective functions [128]. The actor objective function was maximized with respect to θ, while the critic network objective function was minimized with respect to ϕ. The training was performed for $K = 10$ epochs on the whole batch of N episodes simultaneously. It should be mentioned that although a search of suitable hyperparameters was conducted, the performance could have been improved even more by further optimizing training parameters such as the learning rate or the number of hidden neurons in each layer. To speed up training and to stabilize it during later stages, the learning rate was adjusted as the number of training iterations increased and is specified in Table 8.2. After the networks were trained, the sampled training data were discarded and new were generated. The training results are presented in the following section.

8.3 Test sets and training results

The hybrid DRL agent was trained for a total of 600 training iterations, corresponding to 38,400 different episodes and a total of 68,900 samples (each episode consisted of up to 8 time steps/samples, depending on the episode length). The training performance is presented in Fig. 8.5. The total episode reward is presented in sub-figure (i), the final SOL of the episode is presented in sub-figure (ii), and the number of episode time steps is presented in sub-figure (iii). The red line shows a centered moving average computed over the mean value of 500 episodes. To better visualize the results, only every *100th* value during the training is illustrated in the figure. The results show that the performance improved rapidly until around 19,000 episodes, after which the policy managed to achieve a SOL above the threshold value of 30 MW using only a single time step for a majority of the episodes. After this, the performance continued improving by mainly optimizing the level of action activation for each of the scenarios.

In Fig. 8.6, the development of the different policy parameters is presented. The policy parameter governing the standard deviation σ_{cont} increased at first, which was then followed by the mean value μ_{cont} being adjusted. After the model was trained on approximately 40,000 samples, the mean value μ_{cont} stabilized. After that, the model improved mainly by reducing its exploration rate (the standard deviation σ_{cont} and the randomness of the discrete actions). In sub-figure (iii) of Fig. 8.6, the probability of the taken action D_3 is illustrated, showing that the policy became more and more certain of whether the discrete action should be activated or kept inactivated. Similar training development for the other discrete actions was observed as well.

8.3.1 Test sets

The hybrid DRL control was tested on three different test sets to evaluate its performance to handle different types of seen and unseen scenarios. Each of the test sets is detailed below.

1. **Test set 1**: Data generated in the same way as for the training data, but using a deterministic policy instead.

2. **Test set 2**: Introducing new unseen OCs by increasing the variation of the generation and load configurations. Instead of randomly adjusting each load between 80 % to 130 % as specified in Section 8.2.3, the OCs were adjusted randomly between 70 % to 140 %.

3. **Test set 3**: Introducing larger measurement errors by multiplying each state value with a random number with a mean of 1 and a standard deviation of 0.01.

During training, the actor used a stochastic policy which allowed it to automatically explore the available action space. While the exploration rate (the standard deviation of the continuous action space and the randomness of the discrete action spaces) of the agent decreased during the final part of the training, it would require

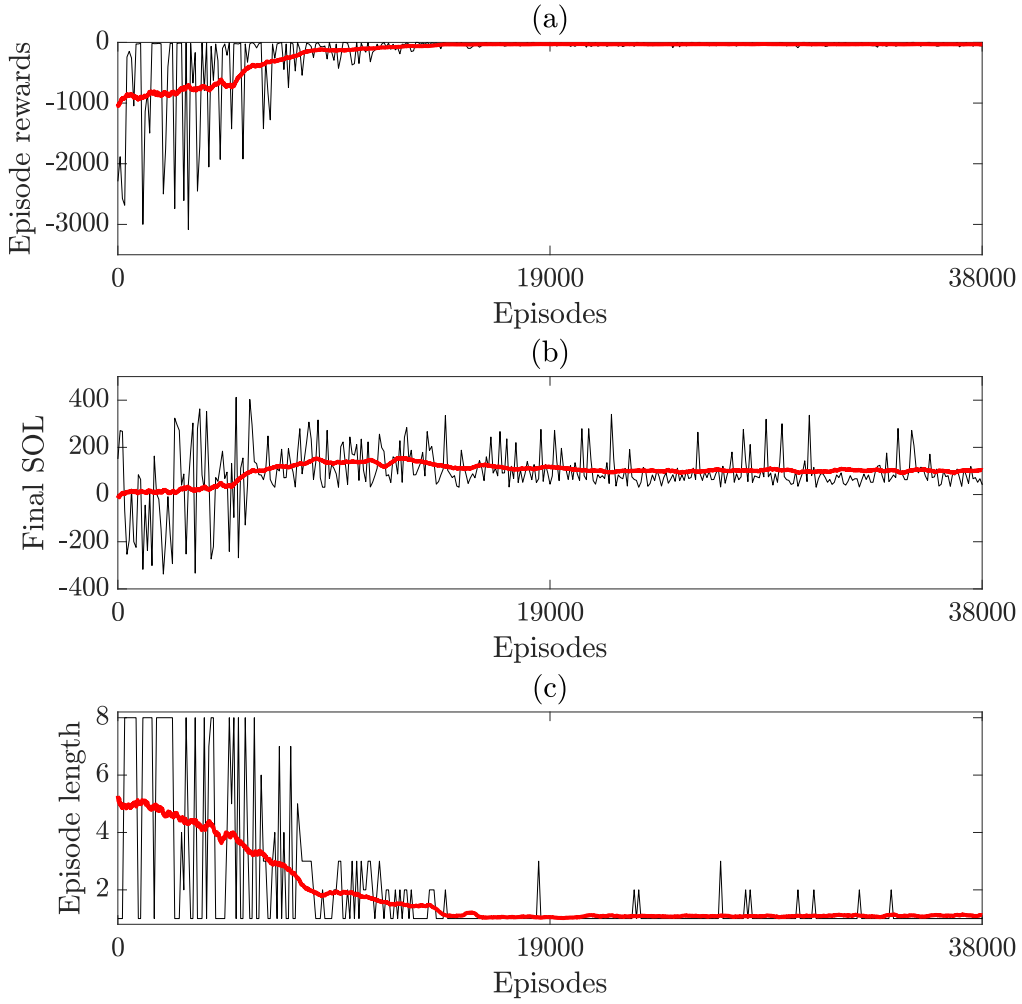

Figure 8.5: Performance and development over training samples and episodes: Subfigures showing (i) episode rewards during all episodes; (ii) final SOL; (iii) episode length. The red line indicates a moving average computed over the mean of 500 episodes. For better visualization, every 100th value is illustrated.

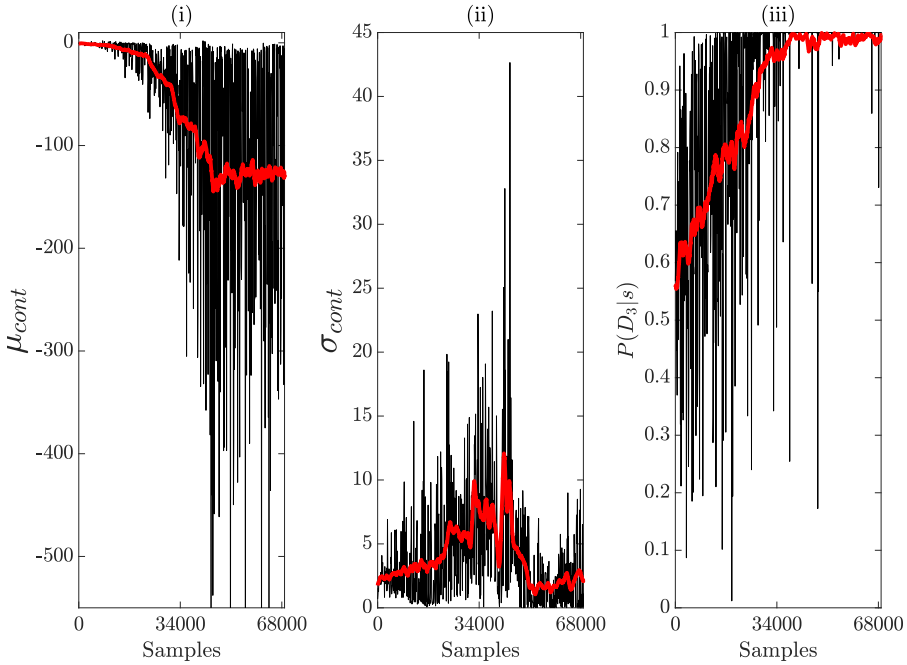

Figure 8.6: Development of policy parameters over episodes. Sub-figures showing (i) the mean value output μ_{cont}; (ii) the standard deviation output σ_{cont}; (iii) the probability of taking the discrete action $P(D_3|s)$. For better visualization, every 100th value is illustrated.

a significantly longer training time for it to reach a point where it essentially converged towards a *fully* deterministic policy. When implementing the policy online it is generally more suitable to transform the control policy into a deterministic one and always pick the actions that with the *highest probability* are optimal. When testing the algorithm, the continuous action was thus controlled directly by the mean value μ_{cont}. Each of the discrete actions was activated whenever any of the defined Bernoulli probabilities satisfied $P(D_i|s) \geq 0.5$.

8.3.2 Test results

The performance of the hybrid DRL control when tested on the different test sets is presented in Table 8.3, each consisting of 200 episodes. For test set 1, the average episode length was 1.09 steps, indicating that the hybrid DRL control managed to ensure a sufficient security margin in a single time step for a majority of all scenarios. For test set 2, where new unseen OCs were introduced by increasing the variation by which the loads and generation were initialized, showed good performance as well. The average episode length increased slightly to 1.12 steps, indicating that some of the unseen OCs forced the hybrid DRL control to require a few more steps before the SOL was restored. The average total episode reward for each of these test sets was -28.6 and -26.3, respectively. For test set 3, where the impact of larger measurement errors on the state values was evaluated, the performance dropped slightly. The average episode reward was reduced to -34.7, while the average episode length increased to 1.19 steps. Thus, for several of the scenarios in the test set, the hybrid DRL control required slightly more time steps before the SOL was restored above the threshold. The results also indicate the importance of incorporating

Table 8.3: Average performance of the hybrid DRL control.

	Test set 1	Test set 2	Test set 3
Episode reward	-28.6	-26.3	-34.7
Episode length [steps]	1.09	1.12	1.19

random errors that exist in real power systems, but which are generally not present when training the method on purely simulated data.

8.4 Comparing the hybrid DRL control to controls based on discrete action spaces

The main advantage of the proposed hybrid DRL architecture is the capability to simultaneously adjust *both* discrete and continuous action variables. To evaluate the impact of this feature, the developed hybrid control is compared to two other control policies which are only capable of controlling a discrete action space: one which is based on a rule-based look-up table control, while the other is based on DRL but only adapted for discrete action spaces. Each of the two methods and their actions is discussed in the following subsections.

8.4.1 Rule-based look-up table control

Conventional methods for preventive control typically rely on system operators to choose actions by matching the current system state with the nearest system state defined in a preventive control look-up table. The difficulty of assessing raw measurements from the system requires system operators to pre-process the system state, generally by first computing the security margin and then, in case the security margin is below a defined threshold, taking measures to restore it. Thus, using a control based on a rule-based look-up table, a system operator would have to (i) first estimate the security margin, (ii) possibly activate actions to restore it, and finally (iii) re-evaluate the security margin to ensure that it is above the defined threshold. In comparison, the developed hybrid DRL control can take actions by *directly* monitoring the system state, without the need for the additional first step (i) with time-consuming data pre-processing and computation of the SOL.

Since the rule-based look-up table require an initial computation of the SOL before any actions can be initiated, a penalty is added to be able to compare its performance to that of the hybrid DRL control. The pre-processing of measurement data and the initial computation of the SOL corresponds to an additional episode step for the hybrid DRL control. Thus, a penalty of -50 (the penalty added for the DRL agent at every time step it does not achieve its control goal) and an added time step for each episode, are added to the performance of the rule-based look-up table. However, for the scenarios when the initially estimated SOL was *above* the threshold value of 30 MW, no actions had to be taken and thus no penalty was added for those scenarios. Depending on the initially estimated SOL, different actions (A1-A6) were activated, each specified in Table 8.4. The actions activated controlled

Table 8.4: Discrete actions for the discrete-DRL control and the rule-based look-up table.

Action number	Initially estimated SOL [MW]	Actions activated	
		Load curtailment [MW]	Switching of reactive shunts
A1	SOL > 30	0	None
A2	$0 <$ SOL ≤ 30	-50	D_1
A3	$-100 <$ SOL ≤ 0	-100	$D_1 + D_2$
A4	$-250 <$ SOL ≤ -100	-200	$D_1 + D_2 + D_3$
A5	$-350 <$ SOL ≤ -250	-350	$D_1 + D_2 + D_3$
A6	SOL ≤ -350	-500	All

load curtailment and/or switching of reactive shunts which refer to the continuous respectively discrete actions specified in Section 8.2.2. The actions and their SOL activation levels were designed to *always* ensure that the SOL was restored above the threshold of 30 MW using a single action.

8.4.2 DRL control adapted for discrete actions

The DRL control that only handles a discrete action space is developed using the same PPO algorithm as for the hybrid control but is adapted only for a discrete action space. The discrete DRL control can choose from the *same* actions (A1-A6) defined in Table 8.4 as for the rule-based look-up control, but each of the actions is now instead chosen directly by the DRL agent without the need of an initially estimated SOL. The discrete DRL actor network has an identical architecture as the one used for the hybrid control, but instead of forming outputs used in defining the Normal and Bernoulli distributions, it uses a Softmax activation function in the final layer. The used Softmax function normalizes the outputs into a categorical probability distribution consisting of six numbers (the available discrete actions A1-A6) where each probability is proportional to the exponents of the input numbers, according to (3.4). From the defined categorical probability distribution, different discrete actions could then be sampled. The same hyperparameters and number of training episodes as were used for the hybrid DRL control and as is defined in Table 8.1 were used during training, with the exception that a lower learning rate for the actor and critic network of $\alpha_{actor} = 1 \cdot 10^{-5}$ and $\alpha_{critic} = 5 \cdot 10^{-5}$ was used, respectively. The lower learning rates were used to ensure that the discrete DRL agent did not converge to a sub-optimal policy due to a too high initial learning rate.

8.4.3 Results

In Table 8.5, the total episode reward and the episode length for the rule-based look-up table and the discrete DRL control are presented for each of the defined test sets. For each metric and test set, the percentage difference in performance compared to the hybrid DRL control is also presented in parenthesis after each value. The results show that the proposed hybrid DRL control performed significantly better on all of

Table 8.5: Average performance when using a rule-based look-up table control. Values in parenthesis presents the percentage increase to the average performance of the hybrid DRL control.

	Rule-based look-up table		*Discrete DRL control*	
	Episode reward	Episode length [steps]	Episode reward	Episode length [steps]
Test set 1	-72.6 (153.7 %)	1.77 (61.9 %)	-35.9 (25.2 %)	1.10 (1.4 %)
Test set 2	-62.3 (136.4%)	1.65 (47.3 %)	-33.3 (26.4 %)	1.11 (-0.9%)
Test set 3	-72.6 (109.0 %)	1.77 (47.7 %)	-39.9 (14.7 %)	1.16 (-3.3 %)

the defined test sets. For instance, the (negative) average reward increased from 109.0 % for test set 3, up to 153.7 % for test set 1 when the rule-based look-up table control was used. In the case of the discrete DRL control, the (negative) average reward increased from 14.7 % for test set 3, up to 25.2 % for test set 1.

The episode length for the rule-based look-up table was significantly higher for all of the test sets, mostly caused by the requirement of the look-up table control to take two steps (one to estimate the initial SOL, and one to verify that the SOL had been restored) whenever the initial SOL was below the threshold value of 30 MW. The average episode length for the discrete DRL control was in a similar range as for the hybrid DRL control for the different test sets. However, although both the hybrid DRL control and the discrete DRL control required a similar number of time steps in each episode, the hybrid DRL control achieved a significantly better total episode reward on each test set. This indicates that the possibility of both adjusting discrete and continuous action variables results in a more flexible control policy that more efficiently can adjust the SOL of a power system. In Fig. 8.7, a histogram showing the total episode reward difference between (a) the hybrid DRL control and the look-up table control, and (b) the hybrid DRL control and the discrete DRL control, is presented for the different scenarios included in test set 1. The results show that in 76.5 % and 73.5 % of all scenarios respectively, the hybrid DRL control achieved a better performance than the other types of control.

Finally, it should be stressed that the choice of method to compare the hybrid DRL control to is not trivial. A typical choice would be to evaluate it against some optimization-based control. However, evaluating a dynamic security margin (the SOL) with respect to a number of different contingencies and choosing from a wide range of different actions is a highly non-linear and non-convex optimization problem. Solving such a problem with optimization-based methods would either require significant simplifications in the model or would be too time-consuming to achieve in the time required by system operators, making the comparison impractical.

8.4.4 Practical aspects and requirements

The developed hybrid DRL agent is proposed to be used as an online tool for system operators to control and ensure a sufficient SOL in real-time. While the DRL agent can be applied online to automatically activate actions, it can also be used to suggest actions that system operators after evaluation can manually choose to activate.

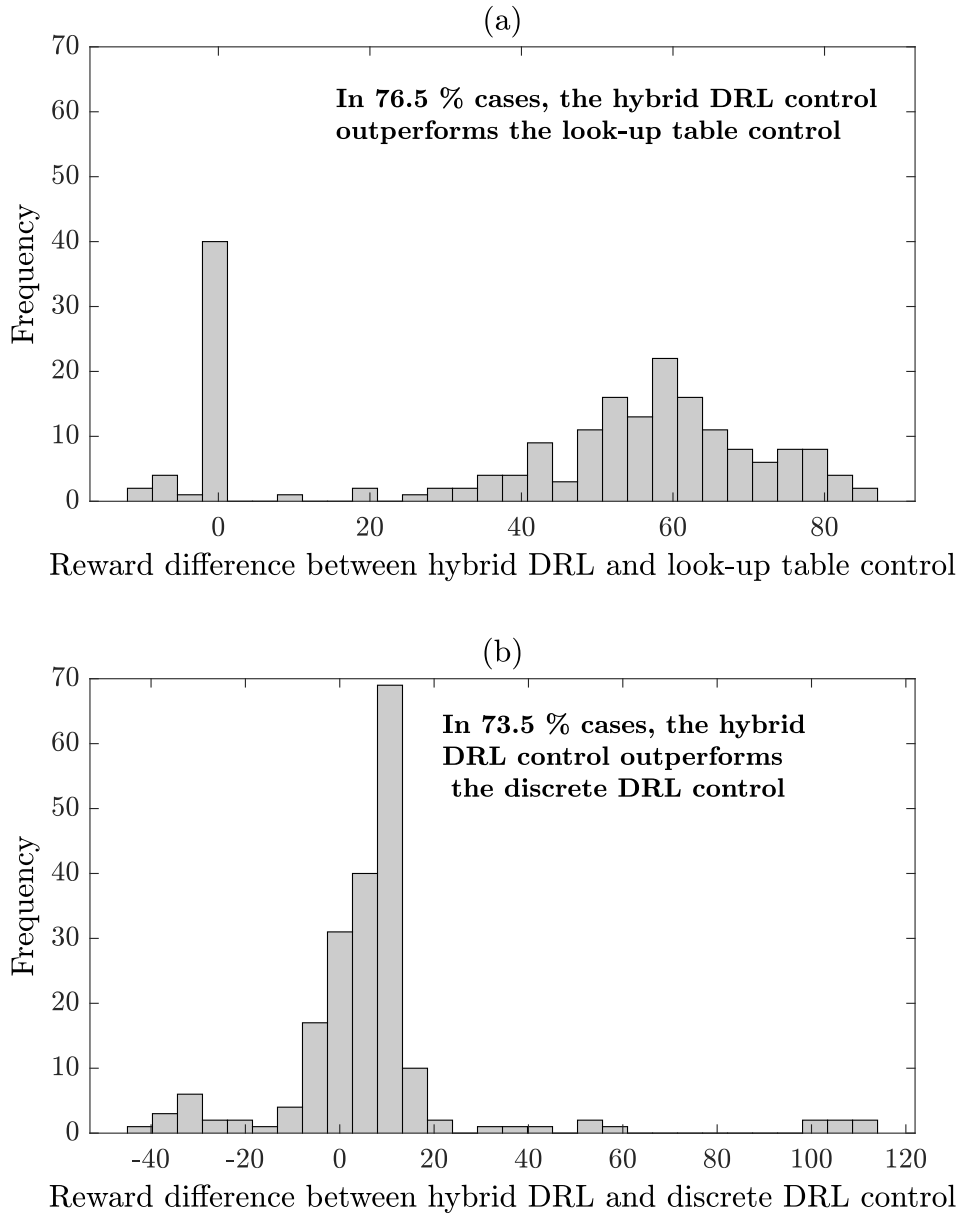

Figure 8.7: Histogram showing the episode reward difference between (a) the hybrid DRL control and the look-up table control and (b) the hybrid DRL control and the discrete DRL control, given for test set 1.

Theoretically, it could also be used as a method that could complement how system operators today develop their preventive control look-up tables. Instead of manually assessing the effectiveness of different actions on a few typical system scenarios, the DRL framework could be used to more efficiently evaluate what range of actions will be efficient for a larger range of different scenarios and automate the evaluation process.

Finally, the time consumption during data generation and the high requirement of computational power should be addressed. The training of the hybrid DRL agent is time-consuming, both due to the large requirement of training data and the fact that training scenarios were generated using full dynamic simulations. The parallelization of the data generation on different CPU cores *significantly* increased the efficiency by which the data was generated. Furthermore, the use of the PPO algorithm, which allowed multiple epochs of training on the same batch data, further increased the efficiency of the training. A possibility to further speed up the generation of training data is to compute the SOL based on QSS simulations or combinations of QSS and dynamic simulations, as suggested in [13] or [38]. While QSS methods, or combinations of QSS and dynamic simulations, do not provide as accurate estimations of the SOL as if it had been computed using a full dynamic simulation, it is still significantly more accurate than most methods today that are based on static estimations of the security margin.

Chapter 9

Deep reinforcement learning for long-term voltage stability control

This chapter develops a real-time control method based on deep reinforcement learning that can suggest optimized control actions to system operators to mitigate voltage instability. The results and the developed methodology were established in paper I. The control method is proposed to be included as an emergency control method in the developed real-time dynamic stability assessment and control tool, earlier presented in Chapter 4.

9.1 Introduction

In this chapter, a method based on DRL is developed for control to mitigate long-term voltage instability. LTVS control in large-scale power systems is a highly non-linear and non-convex optimal decision-making problem. At every time instant, the controller (in DRL: the agent) should assess the state of the system and choose an action that can most efficiently stabilize it to the lowest system cost. The complexity lies both in interpreting the state of the system *and* to determine what action is most optimal to take in that current state.

The developed method of DRL control for LTVS is based on off-line training on a large data set consisting of dynamical simulations for a range of different disturbance and load scenarios. By training a DRL agent on those scenarios, a policy π_θ is developed which can form a mapping from the state (provided by measurements in the system) to an action. Once the DRL agent has been trained, it can in real-time suggest optimized control actions to system operators to mitigate voltage instability. Furthermore, the feasibility of using future system services from e.g. DR and ESS as a more economic and flexible alternative to stabilize the system is evaluated. The study specifically examines the capability of the DRL control to account for the uncertainty involved in using such services (e.g. the price and the availability) as an alternative to conventional load shedding. Finally, an evaluation of the method's robustness and capability of handling scenarios that have not been included in the

training data of the algorithm. This is important since the number of states and the possible combinations with different disturbance scenarios are very large for real power systems.

9.2 DRL for long-term voltage stability control

The training data are generated using PSS®E 35.0.0 with its in-built dynamical models [118]. All simulations have been tested on the slightly modified version of the Nordic32 test system, detailed in [114] and illustrated in Fig. 4.2. In the following sections, all the details in generating the training data and the development of the DRL control are presented.

9.2.1 Training data generation

An overview of the steps involved in training data generation and the training of the DRL agent is illustrated in Fig. 9.1. The different steps in the training data generation are detailed in the sections below.

(1) *Generate initial operating condition*: For the Nordic32 system, the initial OCs were randomly generated around the non-secure operation point denoted as "operating point A" in [114]. All loads in the system were randomly and individually varied by multiplying the active load value with a random variable generated from a uniform distribution (95 % of the original load as lower limit, 105 % of the original load as upper limit). The power factor of all loads was kept constant. A load flow solution was then computed where any changes in total load in the system were compensated by the slack bus generator, *g20*, see Fig. 9.2.

(2) *Introduce random disturbance*: Once an initial OC was generated, a dynamic simulation was initialized and a single larger disturbance was introduced. The DRL agent was trained to handle different types of disturbances, and with the same probability for each scenario, either a line was tripped between buses (i) *4032-4044*, (ii) *4032-4042*, (iii) *4031-4041*, (iv) *4021-4042*, or the tripping of either (v) generator *g6*, or (vi) generator *g7*. The disturbances were chosen as they were proven to cause significant system stress in the "Central" area, and without suitable control actions would in most load scenarios cause long-term voltage instability. In an actual application, preferably all disturbances that are likely to cause long-term voltage instability should be evaluated and included in the training. However, this would require significantly more training data to achieve satisfactory results and without a loss of generalization, the study was reduced to include only the previously mentioned disturbances.

(3) *Sample state s_t and action a_t from the policy and move to next state*: The state was then sampled from the system and passed to the actor network. The actor network outputs parameters that form a current policy $\pi_\theta(a|s)$ from which an action is sampled. That action was then activated in the system and the simulation continued to run until the next time step, which forms the state

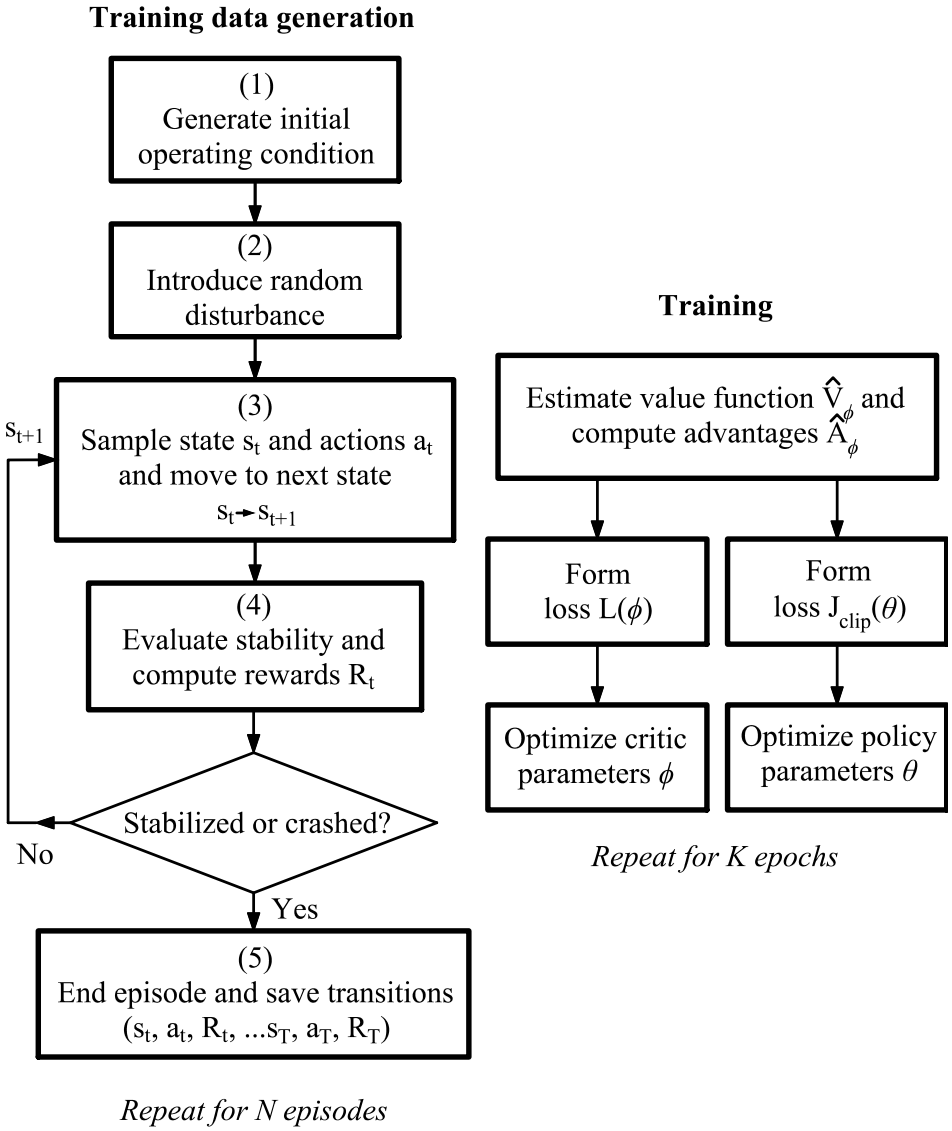

Figure 9.1: Flowchart showing the generation of training data and the training of the actor and critic network.

transition from $s_t \rightarrow s_{t+1}$. The time between each step in the simulation was 5 seconds (while the integration step size in the dynamical simulation in PSS®E was 0.05 seconds). The states and actions are further discussed in section 9.2.2 and 9.2.3, respectively.

(4) *Evaluate stability and compute rewards* R_t: Once the dynamical simulation reached the next time step, the stability of the system was evaluated. If any transmission system bus voltage (V_{TS}) were below 0.7 pu, the system was assumed to be unstable and the episode was terminated. If any V_{TS} at that time point was below 0.90 pu, the system was *also* assumed to be unstable. At every time step, a reward was also computed. The reward R_t at each time step was a combination of the cost for the taken action (C_a); a smaller penalty (-1) if any V_{TS} were below 0.90 pu; or a larger penalty (-500) if the system had become unstable. The action cost C_a is further discussed in Section 9.2.3.

The reward at every time step was then computed as:

$$R_t = \begin{cases} C_a - (500 \cdot 0.99^t) & \text{if unstable} \\ C_a, & \text{else if all } V_{TS} \geq 0.90 \text{ pu} \\ C_a - (1 \cdot 0.99^t), & \text{else if any } V_{TS} < 0.90 \text{ pu} \end{cases} \tag{9.1}$$

The penalties were multiplied by a discounting factor of 0.99^t, resulting in lower negative penalties if instability and low system voltages occurred later rather than early in an episode. In this study, the reward is unitless, but should in real applications reflect the actual monetary cost of different actions and the corresponding rewards when the control goal is either achieved or missed.

(5) *End episode and save transitions*: All episodes ran for a maximum of $T = 1000$ seconds unless the system become unstable and was terminated in advance. At the end of all episodes, the transition data (s_t,a_t,R_t,...,(s_T,a_T,R_T) was stored and later used during training.

9.2.2 States

The states were sampled from measurements taken from the dynamic simulation and consisted of a vector of i) bus voltages magnitudes, ii) active power flows, and iii) reactive power flows of all (and in between) buses in the system. While the relatively slow sample rate would allow measurements to be sampled from SCADA systems, the availability and use of phasor measurement units would ensure a higher modeling accuracy through the time-synchronized measurements. To also capture the dynamics of the system, *previous* observations from time step t-1 were stacked and included in the state vector (thus doubling the length of the state vector). To stabilize training, the state vectors were normalized by subtracting the mean value of each state value by its mean and then dividing by the standard deviation. The mean and standard deviation of each state value was computed from previously sampled states and a list with a maximum of 10 000 sampled states was stored. Once 10 000 sampled states were added to the list, the mean and standard deviation used for normalizing states became fixed.

9.2.3 Actions

To stabilize the system in case of instability, the DRL agent could activate load curtailment resources that, for instance, ESS and DR could provide to system operators. ESS and DR can essentially be viewed as available load curtailment that has been procured through a market system [129]. Using such services would allow system operators to alleviate stress in a power system similar to that of load shedding. The difference is mainly i) that a higher degree of flexibility is available, where the activation level of the load curtailment based on ESS/DR can be typically be taken in much smaller steps than load shedding, and ii) that the impact on the end-users would be significantly less than when using forced load shedding.

The DR and ESS are modeled implicitly by allowing the DRL agent to adjust the load levels at two participating load buses within a certain range. The two load buses that participate are located at bus 1044 and bus 1045, see Fig. 9.2. The

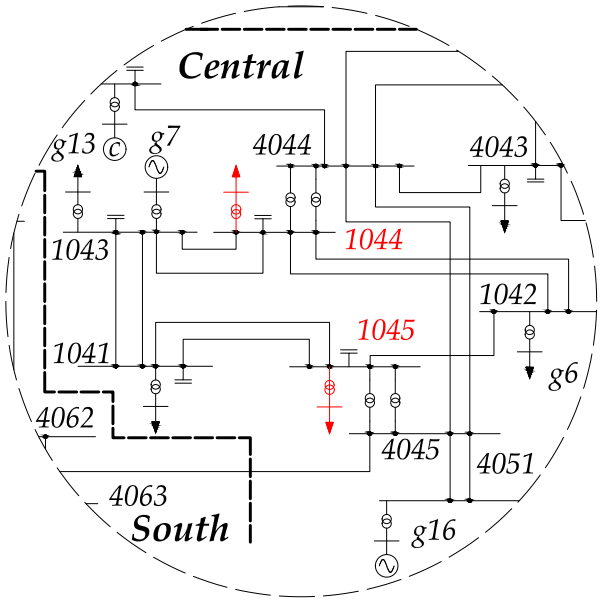

Figure 9.2: Zoomed in one-line diagram of the modified Nordic32 system (see the original in Fig. 4.2). Load buses 1044 and 1045 that participate in the load curtailment are marked in red.

availability and the price of market-based system services provided by DR and ESS are typically varying; an uncertainty that needs to be included in the training of the DRL agent. To model this, the level of load curtailment that is available at each of the two participating buses is varied at the beginning of each disturbance scenario. The capacity of load curtailment is determined by sampling from a random uniform distribution with a lower level of 300 MW, and an upper level of 500 MW. Furthermore, the price of activating the load curtailment is also varied between the two participating buses which is achieved by randomly varying the price for each bus at the beginning of each disturbance scenario. The price of activating load curtailment at each bus is also determined by sampling from a random uniform distribution with a lower cost of -0.1/MW, and an upper level of -0.2/MW.

The DRL agent then controls the *total* level of load curtailment that is taken at each time step. The bus with the lowest price is activated first, but if it has not sufficient capacity in adjusting its load, the other bus (with a higher price) will be activated as well. The range of load curtailment capacity was chosen to ensure that all of the load and disturbance scenarios could be stabilized if sufficient load curtailment was utilized, while still adding uncertainty in where the load curtailment was activated. The price variation models the market-based system, where the price of ESS/DR will typically vary depending on availability. Thus, the chosen approach ensures that there will be an uncertainty in *where* the actions are activated, to *what level* the actions are available, and also to *what cost* to the system. Depending on which bus the load curtailment is activated may also impact the effectiveness of the control action to mitigate instability, which is an additional uncertainty that the DRL agent needs to account for.

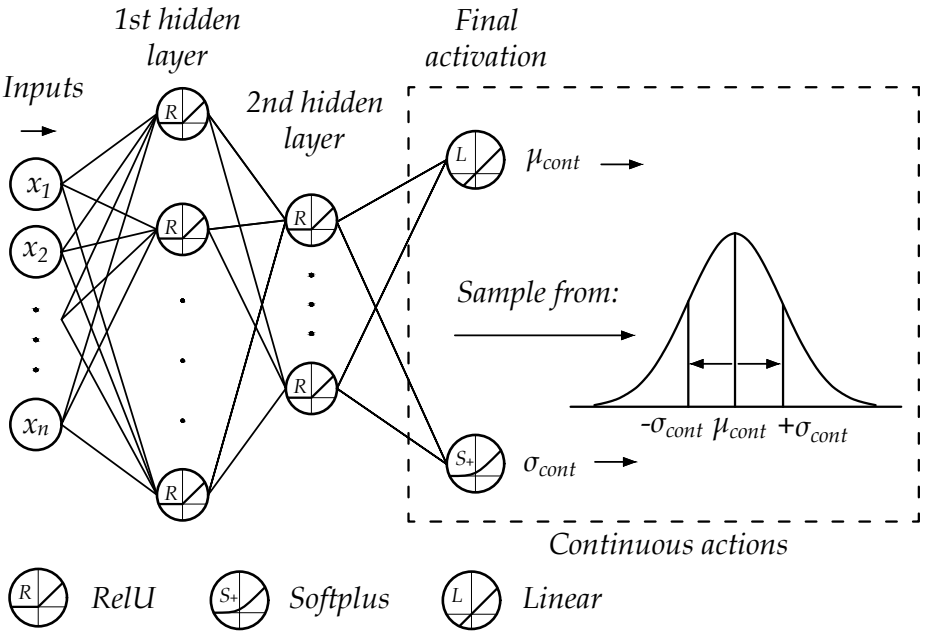

Figure 9.3: Architecture of the actor network.

9.2.4 Architecture of actor and critic network

The actor network, illustrated in Fig. 9.3 and further detailed in Table 9.1, forms the mapping from states to the policy $\pi_\theta(a|s)$, from which actions are sampled. The network has two hidden layers followed by a final activation layer with two different activation functions used to form the outputs. The network outputs parameters used in defining a Normal distribution $\mathcal{N}$ from which the policy is defined and actions are sampled:

$$\pi_\theta(a|s) = \mathcal{N}\left(\mu_\theta(s), \sigma_\theta^2(s)\right) \tag{9.2}$$

The Normal distribution is parametrized by a mean value μ_θ and a standard deviation σ_θ, where the mean value μ_θ is computed using a linear activation function in the final layer, while the standard deviation σ_θ is computed using a softplus activation function that ensures that the value never becomes negative. The critic network is separate from the actor network and consists of a simple NN with a single hidden layer and a linear final activation function, further detailed in Table 9.1.

9.2.5 Training of actor and critic networks

Once a total of $N = 64$ episodes were sampled, the actor and the critic networks were trained. The critic network was first used to compute the value $\hat{V}_\phi^\pi$ of each state. Once the value of each state was computed, the estimated advantage of each state was computed using (3.17). The cost function used to train the actor network is computed using (3.20) on all samples for all N episodes. Once the cost function for all samples was computed, the final $J^{clip}(\theta)$ was computed by taking the mean of those values. The cost function used to train the critic network, $L(\phi)$, is computed by taking the mean squared error on all δ-errors from (3.17) for all

Table 9.1: Design and hyperparameters used in training.

		Parameter	Values
Architecture	*Critic*	Number of inputs	976
		Neurons in hidden layer	128
		Final activation function	Linear
		Hidden layer activation	ReLU
	Actor	Number of inputs	976
		Neurons in 1st hidden layer	64
		Neurons in 2nd hidden layer	32
		Final activation for μ_{cont}	Linear
		Final activation for σ_{cont}	Softplus
		Hidden layer activation	ReLU
Training		Epochs (K)	5
		PPO clip parameter (ϵ)	0.2
		Optimizer	Adam [124]
		Batch size (N)	64

samples and all N episodes, followed by computing the mean of those values. The training was performed using the software Tensorflow in Python which automatically computes the gradients on the defined cost functions. The training was performed for $K = 5$ epochs on the whole batch of N episodes simultaneously. The values of the learning rates and other hyperparameters used in the training are specified in Table 9.1. Once the networks were trained on the stored transition data, new training data were generated and the old data that were sampled with the old policy were discarded.

9.3 Simulations and results

9.3.1 Training results

The DRL agent was trained for a total of 200 training iterations, corresponding to a total of 12 800 episodes, after which the performance converged. The training performance is shown in Fig. 9.4, where the episode rewards and whether the episode resulted in a crash or a stable state at the end of the simulation, is shown. The red line shows a centered moving average computed over the mean value over 250 episodes. The results in sub-figure (i) show that the performance improved rapidly until around 4 000 episodes, after which the policy managed to avoid system collapses completely. After this, the performance continued improving by mainly optimizing the level of action activation for each of the scenarios.

9.3.2 Test sets

During training, the DRL agent used a stochastic policy which allowed it to automatically explore the available action space. However, when using it online it is

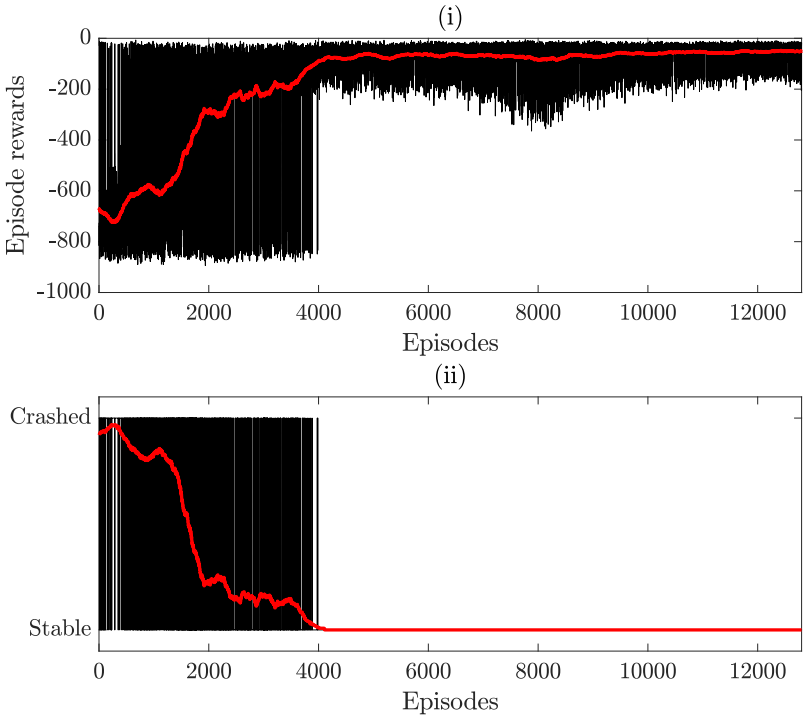

Figure 9.4: Performance and development over episodes: Sub-figures showing (i) episode rewards during all episodes; (ii) whether the episode resulted in a crash or a stable state at the end of the simulation. The red line indicates a moving average computed over the mean of 250 data points.

more suitable to transform the policy into a deterministic one and always pick the actions that with the *highest probability* are optimal. When testing the algorithm, the continuous action was thus not sampled from a normal distribution, but rather controlled directly by the mean value μ_{cont} which was one of the outputs of the actor network. The trained DRL agent was tested on three different test sets. A total of 100 test scenarios were computed for each test set. The test sets used were defined as:

- *Test set 1*: Data generated in the same way as for the training data, but using a deterministic policy instead.
- *Test set 2*: Introducing new unseen OCs by increasing the variation of the generation and load configurations. Instead of randomly adjusting each load between 95 % to 105 % as specified in Section 9.2.1, the OCs were adjusted randomly between 90 % to 110 %.
- *Test set 3*: Introducing new unseen OCs by introducing a disturbance that was not used in training the DRL agent. The new disturbance is the tripping of the line between the buses *4011-4021*. The same variation of generation and load configuration as during training was used.

The performance of the developed DRL agent was also compared to that of a rule-based load shedding protection scheme. The load shedding protection scheme acts

Table 9.2: Average performance required for different test sets and control methods.

	Mean episode reward		Difference
	DRL control	*Load shedding*	[%]
Test set 1	-37.9	-86.7	128.6 %
Test set 2	-36.5	-114.2	213.9 %
Test set 3	-16.7	-22.3	33.5 %

whenever *any* transmission system voltage is below 0.90 pu. In that case, a total of 100 MW load is removed from the system, divided equally between the loads located at bus 1044 and 1045. To allow a fair comparison, the cost for activating the load shedding (C_a) was chosen to -0.15/MW, which is the mean value of the varying price for activation of the DR/ESS resources used by the DRL agent.

9.3.3 Test performance

The average reward on the different test sets is presented in Table 9.2 and is computed as the mean episode reward of all test scenarios. In the final column, the relative difference between the DRL control and the load shedding scheme is presented. The results show that the DRL agent managed to get a significantly lower negative average reward compared to the load shedding control scheme on *all* different test sets. For instance, in test set 1, the load shedding scheme resulted in a 128.6 % higher *negative* reward compared to when the DRL control was. Although not being trained on the load and disturbance scenarios found in test set 2 and test set 3, the DRL agent managed to generalize its learning to these scenarios and still find a significantly more efficient control policy than for the load shedding scheme. The improvement was smallest on test set 3 (33.5 %) when a new disturbance that was not included in the training data was used to stress the system. It should be noted that all test scenarios in each test set were successfully controlled to stable states, both for the DRL control and the rule-based load shedding scheme.

In Table 9.3, the average required load curtailment for each test set and control method is presented. This metric represents how much load each control method required to be curtailed before the system stabilized. Once again, the relative difference between the two control methods is presented in the final column in the table. The results show that the DRL agent required significantly less load curtailment to stabilize the system compared to the load shedding scheme for all of the test sets. For instance, for test set 2, the load shedding required 259.4 % more load in average to be curtailed compared to what was used by the DRL control. The differences between the DRL control and the control that is achieved with load shedding are exemplified in Fig. 9.5 and Fig. 9.6. In Fig. 9.5 the voltage magnitude at bus 1041 is shown for one of the test scenarios in test set 1. The voltage magnitude over time is presented for the cases i) when the DRL control is used, ii) when the load shedding control is used, iii) and when no control is used. In Fig. 9.6, the load at the controlled load buses 1044 and 1045 are also shown, which shows the difference in how the load is controlled by the DRL control and when using a load shedding control.

Table 9.3: Average load curtailment required for different test sets and control methods.

	Average load curtailment [MW]		Difference
	DRL control	*Load shedding*	[%]
Test set 1	190.9	560.0	193.4 %
Test set 2	166.1	597.0	259.4 %
Test set 3	124.5	144.0	15.7 %

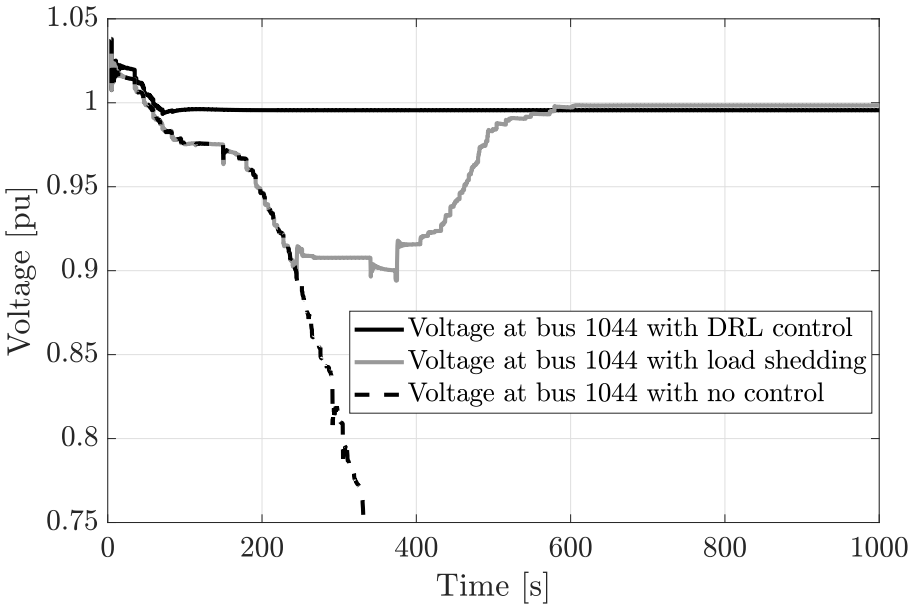

Figure 9.5: Voltage magnitude at bus 1041 over time for different control schemes.

For the given scenario, the system will collapse after around 330 seconds if no control is initiated. For the case with load shedding, a total of 200 MW is shed from the system. The load shedding is activated once at around 250 seconds, and then another activation occurs at around 380 seconds, which can be seen from the relatively large steps in load reduction in Fig. 9.6. After the second activation of the load shedding, the system voltages are restored in the system, which can be seen in Fig. 9.5. For the DRL control, the load curtailment is activated *directly* after the disturbance and in smaller increments, with no need to wait for the system to degrade before the control is activated. The load at bus 1045 is reduced by approximately 130 MW, after which the system is stabilized. The DRL control also manages to achieve a more satisfactory post-disturbance voltage magnitude profile, where the voltage magnitude is kept closer to the nominal pre-disturbance level. Thus, although the DRL control required a smaller amount of load curtailment, it achieved both a faster and more efficient control for the given scenario. The smoother control that is possible when utilizing load curtailment resources from DR and/or ESS also provided a more efficient way to mitigate voltage instability to a low system cost.

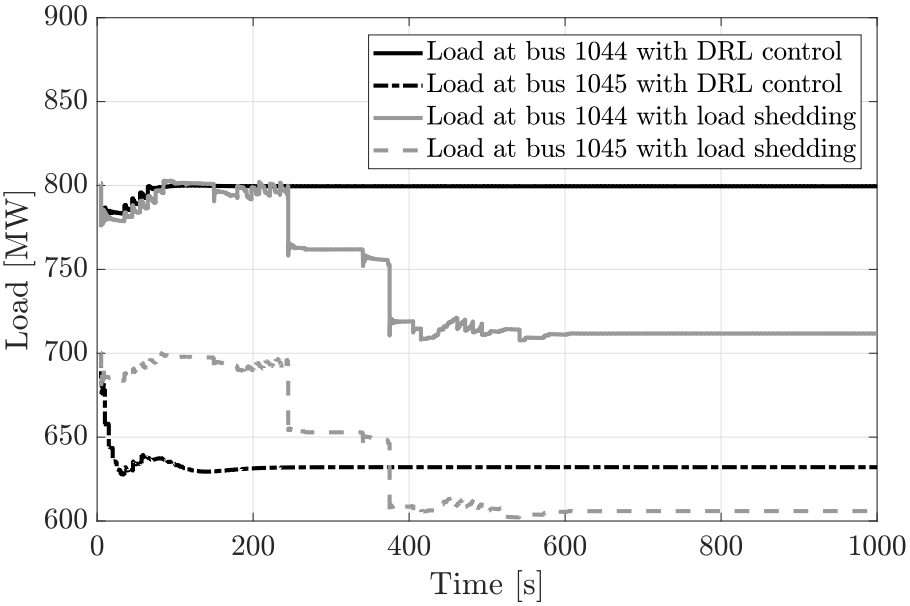

Figure 9.6: Load development at bus 1044 and bus 1045 for the developed DRL control and for a load shedding control scheme.

9.3.4 Performance with an action activation threshold

One of the advantages of the DRL control is that *any* level of load curtailment can be activated at each time step, while typically a load shedding scheme is activated in significantly larger blocks of load curtailment. However, while the DRL agent is trained to minimize the control actions once the system has stabilized, it is difficult to train the action (controlled by the mean value μ_{cont}) to *fully* converge to zero. Furthermore, when evaluating the DRL agent's performance on stable disturbance scenarios (i.e. disturbance scenarios that would en up being stable despite *no* load curtailment being activated), it was found that the DRL agent did (unnecessarily) activate a small level of load curtailment. The reasons for this behaviour can mainly be explained by the fact that the DRL agent was trained on a majority of cases that were unsecure, which can be observed by noting the number of crashed scenarios at the beginning of the training by the red line in sub-figure (ii) in Fig. 9.4.

Avoiding unnecessary activation of load curtailment will be important if the DRL control is to included in any real control systems. The main solution would be to train the DRL agent on more stable scenarios to make it more robust in handling such scenarios. This could however be combined with an action activation threshold, to make sure that only relatively significant action signals are activated in the system. To test this feature, an activation threshold of ± 10 MW of load curtailment was applied for the DRL control. Thus, any control action from the DRL agent with a lower magnitude than 10 MW resulted in *no activation* of load curtailment, while any control action larger or equal 10 MW was activated.

The action activation threshold was tested on the same test sets that were developed in Section 9.3.2. The results are presented in Table 9.4, where the mean episode reward and the average required load curtailment is presented when the action acti-

Table 9.4: Average performance and load curtailment when using an action activation threshold.

	Mean episode reward	Average load curtailment [MW]
Test set 1	-50.3 (32.7 %)	171.6 (-10.1 %)
Test set 2	-41.3 (13.2 %)	114.2 (-29.1 %)
Test set 3	-12.4 (-25.8 %)	22.3 (-82.1 %)

vation threshold was activated. The percentage difference in performance compared to case when no action activation threshold was implemented (while still using the DRL control) is presented in parenthesis after each value. The level of average load curtailment reduced significantly compared to the the case when no action activation threshold was implemented, for all of the sets. The difference was most significant for test set 3, where the average load curtailment was reduced by -82.1 %. However, the mean episode reward worsened for test 1 and test 2, where the (negative) mean episode reward increased by 32.7 % and 13.2 %, respectively. The higher negative rewards were caused by the transmission system voltages remaining below 0.9 pu for a longer time during the post-disturbance state, which resulted in a higher negative reward for those scenarios.

In Fig. 9.7, the action activation threshold is exemplified for a test scenario when both the action activation was used and when it was turned off. When the action activation threshold was used, the load curtailment was activated only in a short period after the disturbance, while in the case when no threshold was used, the load curtailment continued (albeit with low low levels of activation) up to around 200 seconds after the disturbance occured in the system. In Fig. 9.8, the resulting voltage magnitude at bus 1041 is shown for i) the case with no action threshold, ii) with the action threshold used, and iii) when no control is used for the same test scenario. The results show a significantly lower post-disturbance voltage magnitude over time when the action activation threshold was used, but system still manages to remain stable.

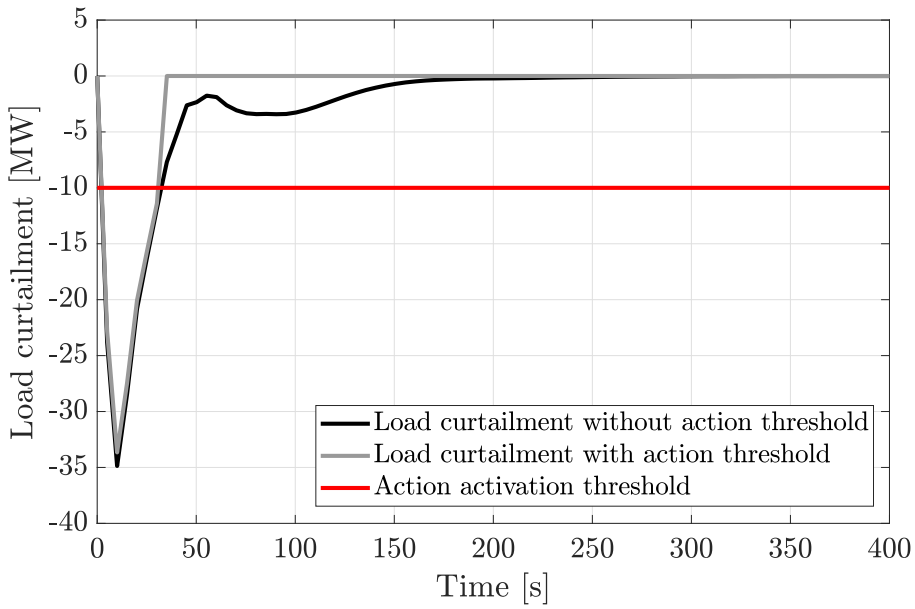

Figure 9.7: Load curtailment activated for the DRL control with and without action activation threshold implemented. Time axis limited for better visualization or results.

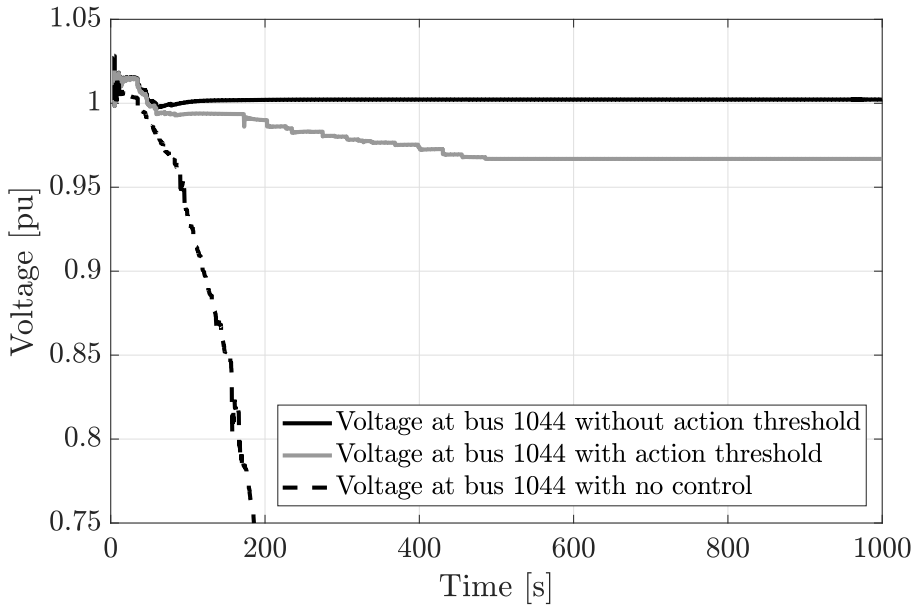

Figure 9.8: Voltage magnitude at bus 1041 over time with or without an action activation threshold.

Chapter 10

Conclusions and future research

This chapter summarizes the conclusions and refers back to the developed research questions and the aim of the thesis. Finally, a discussion of ideas and concepts for future research work is presented.

10.1 Conclusions

The main aim of this thesis has been to develop a new tool for real-time dynamic stability assessment and control that can support system operators and allow more efficient utilization of the transmission grid. The developed tool is comprised of two new methods for use in stability and security assessment, as well as two methods aimed at control to maintain a secure and stable power system. In Section 1.2, five different research questions were identified based on the problem overview which provided a scientific basis for the thesis. Each of the defined research questions and how this thesis has attempted to address them are detailed below.

The purpose of the first research question was to examine the impact that higher penetration of power electronic interfaced equipment has on methods for security margin assessment. This was studied in the thesis both theoretically and numerically, where the difference between the SOL and the simpler method PCLL was evaluated. The theoretical difference between the SOL and the PCLL was first illustrated using a method referred to as transient P-V curves. The results from the numerical comparison showed that the PCLL typically overestimates the security margin in scenarios when the system is dominated by loads with fast load restoration dynamics, and that the SOL in these circumstances provides a more accurate estimate of the actual security margin. The results further indicated that the difference between the methods is smaller for cases when a fast fault-clearing time is possible.

The purpose of the second research question was to evaluate how current methods for security margin can be improved, taking into account assessment speed, accuracy, and robustness. This was studied in the thesis by developing a methodology for fast estimation of the DVSM to overcome the computational difficulties involved

in its computation. The method uses a regression-based NN to provide a qualified estimate of the actual DVSM. Thereafter, a second NN is used to classify which contingency will be dimensioning for the system security margin. The estimates from the NNs are used in a method referred to as dual binary search which is used to *validate* the actual DVSM using time-domain simulations. The ML-based approach is thus only proposed to *support* the estimation of the DVSM, while the actual DVSM is always validated through actual time-domain simulations. This two-step approach is proposed with the aim to overcome robustness issues and uncertainty of using ML-based methods, while still allowing near real-time estimations of the DVSM. The results are promising and the trained NNs provides good estimations of both the DVSM and classifications for the dimensioning contingency. The accurate estimations used in combination with the proposed dual binary search method were found to successfully reduce the required number of time-domain simulations, which would allow system operators to overcome the main practical difficulties of estimating the DVSM.

The purpose of the third research question was to examine how future voltage instability detection methods could be improved, taking into account the intricate dynamics during a voltage collapse, while at the same time being fast enough to be used in real-time. This has been addressed in the thesis by developing a new approach for online VIP using an LSTM network capable of utilizing a sequence of measurements to improve classification accuracy. Once trained, the LSTM network can allow system operators to continuously assess and predict whether the present system state is stable or if it will evolve into an alert or an emergency state in the near future. The network is also adapted to be able to indicate *where* instability emerges, allowing system operators to perform more cost-effective control measures. The LSTM network was proposed to improve the available state signal by implicitly learning the dynamical trajectories of a power system following a disturbance. The results are encouraging and the proposed method is shown to have high accuracy in predicting voltage instability. Almost all N-1 contingency test cases were predicted correctly, and N-1-1 contingency test cases were predicted with over 95 % accuracy only seconds after a disturbance. The impact of the sequence length of the LSTM network was tested and it indicates that a longer sequence provided a significantly better classification capability than both a feedforward NN and a network using a shorter sequence.

The purpose of the fourth research question was to evaluate how future control systems could be designed, taking into account the non-linearity and the complexity of electric power systems. To address this, a DRL framework was developed which is based on training a DRL agent by having it interact with a simulated power system and taking various actions to achieve the control goal. Through continuous interactions with the power system, the DRL agent was trained to achieve an effective control policy that could be used to choose actions in real-time. Two different approaches of the DRL control were developed: one preventive control aimed at maintaining SOL and one emergency control aimed at mitigating long-term voltage instability in the case of larger disturbances. Both of the developed DRL agents were found to achieve a good performance on the developed test sets and outperformed

the more conventional control schemes to which they were compared.

The purpose of the final research question was to examine how robustness issues and inconsistencies in predictions and control could be mitigated when using data-driven methods. This part has been separately examined in each of the developed studies. In all of the different studies, the developed monitoring and control methods have been tested on test scenarios that had not been included in the training set on which the methods were trained. This could include test data where the inclusion of measurement errors, new types of disturbances, new OCs, or combinations of these aspects were tested. While some of the developed methods show robustness towards these issues, others result in significantly reduced performances. An alternative approach to using the direct estimates from the data-driven methods is to simply use them as an *aid* in the estimation process. This was evaluated in Chapter 6, where the DVSM estimations of the NNs were only used to support and speed up the estimation process and actual time-domain simulations were always used to validate the results.

10.2 Observations and future research

Data-driven methods, and especially methods based on DRL, are relatively new concepts in power system monitoring and control. Practical implementations are few and if they do exist they typically take the form of research work rather than actual systems used in practical operations. In this section, identified challenges and future research directions are discussed.

- A first challenge concerns the acceptance of data-driven methods by system operators and planners in the electric power system industry. Traditional approaches in monitoring and control are typically model-based and they explicitly account for well-known physical laws. Considering the practical consequences of failure of data-driven methods, transitioning toward systems fully depending on such methods will be a challenge and there is a need to demonstrate and assure system operators and field experts that these methods are efficient and reliable. A reasonable first step would be to use data-driven methods in parallel with more traditional approaches, which would allow system operators to assess their accuracy while still relying on conventional monitoring and control systems. A second step would be to only use traditional methods when the data-driven methods are not confident enough in their predictions/estimations. The development and testing of such prototype methods where real systems and real system data are used is a necessary step for the future adoption of data-driven methods.

- One of the major barriers to practically applying NNs and DRL in safety-critical systems (such as electric power systems) is the inability to anticipate the behavior of NNs [130]. While NNs might have high prediction accuracy on unseen test data, they can be highly vulnerable to so-called adversarial examples, where small input perturbations may lead to poor performance. A possibility to strengthen the confidence of system operators in data-driven

methods is to use verification methods of the NN behavior, such as the methods proposed in [130–132]. Such methods can provide performance guarantees of when the NN-based methods are reliable to use, bridging the robustness issues and removing a major barrier toward their adoption in actual power systems. Another approach includes the use of physics-informed neural networks, in which NNs are trained to solve supervised learning tasks while still respecting given physical laws that are described by nonlinear partial differential equations [133–135]. This approach allows the use of prior knowledge of the system dynamics and physical relationships which can compensate for the deficiency of limited training data.

- The demand for robust data-driven methods in monitoring and control are typically *less* critical for preventive applications compared to emergency applications. For instance, if a preventive control action is sub-optimal and does not instantaneously restore the system's security, the result is that the system will remain unsecure for a slightly longer time. Only in the (very unlikely) event of a disturbance occurring at the same time as the system is temporarily unsecure will there be any significant consequences to the system itself. This impact can be compared to an emergency event, where a misclassification of the system's stability, or a poorly chosen action to stabilize the system, could have disastrous results on the system with possible major blackouts as a consequence. Hence, the robustness requirements of the developed methods will typically vary and preventive methods based on ML can be assumed to have fewer barriers to practical implementation.

- All the developed methods in this thesis require simulation models to generate the training data. Even though simulation models are continuously improving, they do arguably not always provide the right answer to how a system would actually behave during an adverse event. A method that is based solely on simulated data and disturbance scenarios can never be more accurate than the model itself. That is, if a system model is not accurate, it is impossible for a control that is based on that system model to be fully accurate. Thus, it is imperative to continue verifying the dynamic behavior of existing models, especially as the power system is now transitioning into a system further dominated by distributed generation and new types of loads. Finally, it should be noted that this is a challenge for *any* method that uses simulation models and is not exclusively a problem for data-driven methods.

- The generation of training data is an imperative part of any data-driven method. The methods presented in this thesis have mainly been developed using full dynamic simulations, which are time-consuming despite that relatively small test systems have been used. If the developed methods are to be scaled up, the computational cost and time in generating training data may become a challenge even if significant computational resources are available. Furthermore, the performance of the presented methods is also heavily relying on the quality of the underlying data. While historical data and system states can be used as a large part of the training data, it will also be necessary

to include forecasted and generated data. To ensure a good performance in any state the system might be operating in, the developed datasets must be balanced between secure/stable and insecure/unstable operating points and take into consideration all possible combinations of loads, generations, and topology variations that can be expected. Methods for efficient and representative database generation, examples including [33, 126, 136, 137], will thus be important to overcome such challenges.

Bibliography

[1] IRENA, "Renewable power generation costs in 2020," International Renewable Energy Agency, Abu Dabhi, 2021. [Online]. Available: https://www.irena.org/publications/2021/Jun/Renewable-Power-Costs-in-2020

[2] GWEC, "Global wind report 2021," Global Wind Energy Council, 2021. [Online]. Available: https://gwec.net/global-wind-report-2021/

[3] Energimyndigheten, "Energimyndigheten: Statistikdatabas," 2022. [Online]. Available: http://pxexternal.energimyndigheten.se/pxweb/sv/Vindkraftsstatistik/

[4] Svenska kraftnät, "Systemutvecklingsplan 2020–2029," Statnett, Fingrid, Energinet, Svenska kraftnät, 2019. [Online]. Available: https://www.svk.se/om-oss/rapporter-och-remissvar

[5] ENTSO-E, "Picasso," 2020. [Online]. Available: https://www.entsoe.eu/network_codes/eb/picasso/

[6] Svenska kraftnät, "Nordic grid development plan 2019," Svenska kraftnät, 2019. [Online]. Available: www.svk.se/om-oss/rapporter-och-remissvar

[7] P. Kundur *et al.*, "Definition and classification of power system stability ieee/-cigre joint task force on stability terms and definitions," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387–1401, Aug 2004.

[8] Nord Pool, "Market members," Nord Pool, 2021, accessed: 2022-06-01. [Online]. Available: https://www.nordpoolgroup.com/en/the-power-market/The-market-members/

[9] Svenska kraftnät, "Our activities," Svenska kraftnät, 2021, accessed: 2022-06-01. [Online]. Available: https://www.svk.se/en/about-us/our-activities/

[10] S. Brahma *et al.*, "Real-time identification of dynamic events in power systems using PMU data, and potential applications—models, promises, and challenges," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 294–301, 2017.

[11] M. Glavic and T. Van Cutsem, "Wide-area detection of voltage instability from

synchronized phasor measurements. part I: Principle," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1408–1416, 2009.

[12] T. Van Cutsem and C. Vournas, *Voltage stability of electric power systems*. Boston: Kluwer Academic Publishers, 1998.

[13] T. Van Cutsem, C. Moisse, and R. Mailhot, "Determination of secure operating limits with respect to voltage collapse," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 327–335, Feb 1999.

[14] V. Ajjarapu and C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis," *IEEE Transactions on Power Systems*, vol. 7, no. 1, pp. 416–423, Feb 1992.

[15] S. C. Savulescu, *Real-Time Stability in Power Systems : Techniques for Early Detection of the Risk of Blackout.*, ser. Power Electronics and Power Systems. Springer International Publishing, 2014.

[16] I. Konstantelos *et al.*, "Implementation of a massively parallel dynamic security assessment platform for large-scale grids," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1417–1426, May 2017.

[17] T. Van Cutsem and R. Mailhot, "Validation of a fast voltage stability analysis method on the Hydro-Quebec system," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 282–292, Feb 1997.

[18] M. Zima and G. Andersson, "Model predictive real-time control of electric power systems under emergency conditions," in *Real-Time Stability in Power Systems*, S. Savulescu, Ed. Springer, 2004, pp. 367–385.

[19] Q. Huang *et al*, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1171–1182, 2020.

[20] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, p. 529–533, 2015.

[21] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–359, 10 2017.

[22] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[23] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3389–3396.

[24] J. L. Cremer, "Probabilistic dynamic security assessment for power system

control," Ph.D. dissertation, Imperial College London, 2020.

[25] J. Tong and L. Wang, "Design of a dsa tool for real time system operations," in *2006 International Conference on Power System Technology*, 2006, pp. 1–5.

[26] R. Schainker, G. Zhang, P. Hirsch, and C. Jing, "On-line dynamic stability analysis using distributed computing," in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008, pp. 1–7.

[27] S. Smith *et al.*, "On-line transient stability analysis using high performance computing," in *ISGT 2014*, 2014, pp. 1–5.

[28] Abdul-Rahman, Khaled *et al.*, "Application of high performance computing in california iso's on-line dsa system," in *2017 IEEE Power Energy Society General Meeting*, 2017, pp. 1–7.

[29] L. Wehenkel, "Machine learning approaches to power-system security assessment," *IEEE Expert*, vol. 12, no. 5, pp. 60–72, 1997.

[30] ——, *Automatic Learning Techniques in Power Systems*. Boston, MA: Springer, 1998.

[31] T. Van Cutsem *et al.*, "Decision tree approaches to voltage security assessment," *IEE Proceedings C - Generation, Transmission and Distribution*, vol. 140, no. 3, pp. 189–198, May 1993.

[32] K. Sun *et al.*, "An online dynamic security assessment scheme using phasor measurements and decision trees," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1935–1943, Nov 2007.

[33] V. Krishnan, J. D. McCalley, S. Henry, and S. Issad, "Efficient database generation for decision tree based power system security assessment," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2319–2327, Nov 2011.

[34] H. Khoshkhoo and S. M. Shahrtash, "Fast online dynamic voltage instability prediction and voltage stability classification," *IET Generation, Transmission & Distribution*, vol. 8, no. 5, pp. 957–965, May 2014.

[35] C. Liu *et al.*, "A systematic approach for dynamic security assessment and the corresponding preventive control scheme based on decision trees," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 717–730, 2014.

[36] Y. Mansour *et al.*, "Large scale dynamic security screening and ranking using neural networks," *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 954–960, May 1997.

[37] A.A da Silva *et al.*, "Power systems reliability evaluation using neural networks," *International journal of engineering intelligent systems for electrical engineering and communications*, vol. 9, no. 4, pp. 219–226, 2001.

[38] T. Van Cutsem, M.-E. Grenier, and D. Lefebvre, "Combined detailed and quasi steady-state time simulations for large-disturbance analysis," *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 9, pp. 634 – 642, 2006.

[39] A. Sittithumwat and K. Tomsovic, "Dynamic security margin estimation using artificial neural networks," in *IEEE Power Engineering Society Summer Meeting,*, vol. 3, Chicago, IL, July 2002, pp. 1322–1327.

[40] N. Amjady, "Dynamic voltage security assessment by a neural network based method," *Electric Power Systems Research*, vol. 66, no. 3, pp. 215 – 226, 2003.

[41] J. Shang, J. Zhang, W. Zhao, and J. Liu, "ANN based dynamic voltage security assessment for a practical power system," in *2007 International Power Engineering Conference (IPEC 2007)*, Singapore, Dec 2007, pp. 794–798.

[42] M. V. Baghmisheh and H. Razmi, "Dynamic voltage stability assessment of power transmission systems using neural networks," *Energy Conversion and Management*, vol. 49, no. 1, pp. 1 – 7, 2007.

[43] M. Glavic and T. Van Cutsem, "A short survey of methods for voltage instability detection," in *Proc. (IEEE) PES General Meeting*, Detroit, MI, Jul 2011, pp. 1–8.

[44] R. F. Nuqui, "State estimation and voltage security monitoring using synchronized phasor measurements," Ph.D. dissertation, Virginia Tech, 2001.

[45] R. Diao *et al.*, "Decision tree-based online voltage security assessment using PMU measurements," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 832–839, May 2009.

[46] H. Khoshkhoo and S. M. Shahrtash, "On-line dynamic voltage instability prediction based on decision tree supported by a wide-area measurement system," *IET Generation, Transmission & Distribution*, vol. 6, no. 11, pp. 1143–1152, November 2012.

[47] M. L. Scala, M. Trovato, and F. Torelli, "A neural network-based method for voltage security monitoring," *IEEE Transactions on Power Systems*, vol. 11, no. 3, pp. 1332–1341, Aug 1996.

[48] P. Geurts and L. Wehenkel, "Early prediction of electric power system blackouts by temporal machine learning," in *Proceedings of the ICML98/AAAI98 Workshop on "Predicting the future: AI Approaches to time series analysis*, 1998, pp. 24–26.

[49] L. Wehenkel, C. Lebrevelec, M. Trotignon, and J. Batut, "Probabilistic design of power-system special stability controls," *Control Engineering Practice*, vol. 7, no. 2, pp. 183–194, 1999.

[50] P. Geurts and L. Wehenkel, "Temporal machine learning for switching con-

trol," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2000, pp. 401–408.

[51] F. Capitanescu and T. Van Cutsem, "Preventive control of voltage security margins: a multicontingency sensitivity-based approach," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 358–364, 2002.

[52] H. Cai, H. Ma, and D. J. Hill, "A data-based learning and control method for long-term voltage stability," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 3203–3212, 2020.

[53] Q. Li, Y. Xu, and C. Ren, "A hierarchical data-driven method for event-based load shedding against fault-induced delayed voltage recovery in power systems," *IEEE Trans. Industrial Informatics*, vol. 17, no. 1, pp. 699–709, 2021.

[54] D. Ernst, M. Glavic, and L. Wehenkel, "Power systems stability control: reinforcement learning framework," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 427–435, 2004.

[55] Z. Zhang, D. Zhang, and R. C. Qiu, "Deep reinforcement learning for power system applications: An overview," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 1, pp. 213–225, 2020.

[56] M. Glavic, "(Deep) Reinforcement learning for electric power system control and related problems: A short review and perspectives," *Annual Reviews in Control*, vol. 48, pp. 22–35, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578819301014

[57] R. Diao *et al.*, "Autonomous voltage control for grid operation using deep reinforcement learning," in *2019 IEEE Power Energy Society General Meeting (PESGM)*, 2019, pp. 1–5.

[58] S. Wang *et al.*, "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4644–4654, 2020.

[59] B. L. Thayer and T. J. Overbye, "Deep reinforcement learning for electric transmission voltage control," in *2020 IEEE Electric Power and Energy Conference (EPEC)*, 2020, pp. 1–8.

[60] J-F, Toubeau *et al.*, "Deep reinforcement learning-based voltage control to deal with model uncertainties in distribution networks," *Energies*, vol. 13, no. 15, p. 3928, 2020.

[61] J. Zhang *et al.*, "Deep reinforcement leaming for short-term voltage control by dynamic load shedding in china southem power grid," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[62] C. X. Jiang, Z. Li, J. H. Zheng, and Q. H. Wu, "Power system emergency con-

trol to improve short-term voltage stability using deep reinforcement learning algorithm," in *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, 2019, pp. 1872–1877.

[63] Y. Hashmy, Z. Yu, D. Shi, and Y. Weng, "Wide-area measurement system-based low frequency oscillation damping control through reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5072–5083, 2020.

[64] S. Zarrabian, R. Belkacemi, and A. A. Babalola, "Reinforcement learning approach for congestion management and cascading failure prevention with experimental application," *Electric Power Systems Research*, vol. 141, pp. 179–190, 2016.

[65] N. Hatziargyriou *et al.*, "Stability definitions and characterization of dynamic behavior in systems with high penetration of power electronic interfaced technologies," IEEE PES, Tech. Rep., 2020.

[66] ——, "Definition and classification of power system stability revisited & extended," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3271–3281, 2021.

[67] A. Boričić, J. L. R. Torres, and M. Popov, "Comprehensive review of short-term voltage stability evaluation methods in modern power systems," *Energies*, vol. 14, no. 14, 2021.

[68] K. Morison, Lei Wang, and P. Kundur, "Power system security assessment," *IEEE Power and Energy Magazine*, vol. 2, no. 5, pp. 30–39, Sep. 2004.

[69] C. Vournas and T. Van Cutsem, "Online voltage security assessment," in *Real-Time Stability in Power Systems - Techniques for Early Detection of the Risk of Blackout*. Springer, Jul. 2014, ch. 10, pp. 305–333.

[70] ETSO: European Transmission System Operators, "Definitions of transfer capacities in liberalised electricity markets," accessed: 2022-06-01. [Online]. Available: https://www.entsoe.eu/fileadmin/user_upload/_library/ntc/entsoe_transferCapacityDefinitions.pdf

[71] S. Yari and H. Khoshkhoo, "A comprehensive assessment to propose an improved line stability index," *International Transactions on Electrical Energy Systems*, vol. 29, no. 4, p. e2806, 2019.

[72] J. Modarresi, E. Gholipour, and A. Khodabakhshian, "A comprehensive review of the voltage stability indices," *Renewable and Sustainable Energy Reviews*, vol. 63, pp. 1–12, 2016.

[73] K. Vu, M. M. Begovic, D. Novosel, and M. M. Saha, "Use of local measurements to estimate voltage-stability margin," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1029–1035, 1999.

[74] F. Gubina and B. Strmcnik, "Voltage collapse proximity index determina-

tion using voltage phasors approach," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 788–794, 1995.

[75] M. Moghavvemi and O. Faruque, "Real-time contingency evaluation and ranking technique," *IEEE Proc. - Generation, Transmission and Distribution*, vol. 145, no. 5, pp. 517–524, 1998.

[76] M. Moghavvemi and F. M. Omar, "Technique for contingency monitoring and voltage collapse prediction," *IEE Proc. - Generation, Transmission and Distribution*, vol. 145, no. 6, pp. 634–640, 1998.

[77] C. Vournas and T. V. Cutsem, "Local identification of voltage emergency situations," *IEEE Transactions on Power Systems*, vol. 23, pp. 1239 – 1248, 09 2008.

[78] C. D. Vournas, C. Lambrou, and M. Kanatas, "Application of Local Autonomous Protection Against Voltage Instability to IEEE Test System," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3300–3308, July 2016.

[79] C. Vournas, C. Lambrou, and P. Mandoulidis, "Voltage stability monitoring from a transmission bus PMU," in *2017 IEEE Manchester PowerTech*, June 2017, pp. 1–1.

[80] C. Vournas and P. Mandoulidis, "On-line voltage stability monitoring," in *Circuits and Systems (ISCAS), 2018 IEEE Int. Symposium on.* IEEE, 2018, pp. 1–5.

[81] Y. Wang, *et al.*, "Voltage stability monitoring based on the concept of coupled single-port circuit," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2154–2163, 2011.

[82] J. Liu and C. Chu, "Wide-area measurement-based voltage stability indicators by modified coupled single-port models," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 756–764, March 2014.

[83] B. Cui and Z. Wang, "Voltage stability assessment based on improved coupled single-port method," *IET Generation, Transmission Distribution*, vol. 11, no. 10, pp. 2703–2711, 2017.

[84] A. Perez, H. Jóhannsson, J. Østergaard, M. Glavic, and T. Van Cutsem, "Improved Thévenin equivalent methods for real-time voltage stability assessment," in *2016 IEEE International Energy Conference (ENERGYCON)*, 2016, pp. 1–6.

[85] Lixin Bao, Zhenyu Huang, and Wilsun Xu, "Online voltage stability monitoring using var reserves," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1461–1469, Nov 2003.

[86] B. Leonardi and V. Ajjarapu, "Investigation of various generator reactive power reserve (grpr) definitions for online voltage stability/security assess-

ment," in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, July 2008, pp. 1–7.

[87] ——, "An approach for real time voltage stability margin control via reactive power reserve sensitivities," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 615–625, May 2013.

[88] H. Nguyen Duc, I. Kamwa, L.-A. Dessaint, and H. Cao-Duc, "A novel approach for early detection of impending voltage collapse events based on the support vector machine," *International Transactions on Electrical Energy Systems*, vol. 27, no. 9, p. e2375, 2017.

[89] M. Glavic and T. Van Cutsem, "Wide-area detection of voltage instability from synchronized phasor measurements. part II: Simulation results," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1417–1425, 2009.

[90] C. W. Taylor, "Power system stability controls," in *Power System Stability and Control*. Taylor & Francis Group, Apr. 2012, ch. 13, pp. 192–211.

[91] P. Kundur, "Power system stability," *Power system stability and control*, vol. 10, 2007.

[92] M. Glavic, R. Fonteneau, and D. Ernst, "Reinforcement learning for electric power system decision and control: Past considerations and perspectives," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6918 – 6927, 2017, 20th IFAC World Congress.

[93] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[94] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251 – 257, 1991.

[95] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[96] P. J. Werbos *et al.*, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[97] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[98] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[99] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 28, no. 10, pp. 2222–2232, Oct 2017.

[100] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[101] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (second edition)*. Cambridge, Massachusetts London, England: The MIT Press, 2018.

[102] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[103] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. Citeseer, 1994, vol. 37.

[104] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[105] S. M. Kakade, "A natural policy gradient," *Advances in neural information processing systems*, vol. 14, 2002.

[106] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[107] M. Neunert *et al.*, "Continuous-discrete reinforcement learning for hybrid control in robotics," in *Conference on Robot Learning*. PMLR, 2020, pp. 735–751.

[108] T. Van Cutsem, M. Glavic, W. Rosehart, and J. A. dos Santos, "Test systems for voltage stability analysis and security assessment," 2015, IEEE/PES Task Force PES-TR19. [Online]. Available: http://resourcecenter.ieee-pes.org/pes/product/technical-publications/PESTR19

[109] M. K. Pal, "Voltage stability conditions considering load characteristics," *IEEE Transactions on Power Systems*, vol. 7, no. 1, pp. 243–249, Feb 1992.

[110] H.-D. Chiang *et al.*, "CPFLOW: a practical tool for tracing power system steady-state stationary behavior due to load and generation variations," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 623–634, May 1995.

[111] I. Dobson, "The irrelevance of load dynamics for the loading margin to voltage collapse and its sensitivities," in *Bulk power system voltage phenomena - III: Voltage stability, security & control*, 1994, pp. 509–518.

[112] C. Dharmakeerthi, N. Mithulananthan, and T. Saha, "Impact of electric vehicle fast charging on power system voltage stability," *International Journal of Electrical Power & Energy Systems*, vol. 57, pp. 241 – 249, 2014.

[113] M.-E. Grenier, D. Lefebvre, and T. Van Cutsem, "Quasi steady-state models for long-term voltage and frequency dynamics simulation," in *2005 IEEE Russia Power Tech*, 2005, pp. 1–8.

[114] T. Van Cutsem *et al.*, "Test systems for voltage stability studies," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 4078–4087, 2020.

[115] A. Arif *et al.*, "Load modeling—a review," *IEEE Transactions on Smart Grid*,

vol. 9, no. 6, pp. 5986–5999, 2018.

[116] *PSS®E 35.0.0 Program Application Guide: Volume II*, Siemens Power Technologies International, Schenectady, NY, Sep. 2019.

[117] *PSS®E 35.0.0 Program Application Guide: Volume I*, Siemens Power Technologies International, Schenectady, NY, Sep. 2019.

[118] *PSS®E 34.2.0 Model Library*, Siemens Power Technologies International, Schenectady, NY, Apr. 2017.

[119] H. Hagmar, R. Eriksson, and L. Tuan, "Fast dynamic voltage security margin estimation: concept and development," *IET Smart Grid*, vol. 3, Apr 2020.

[120] J. Jorge L, "Online dynamic security assessment," in *Real-Time Stability in Power Systems*. Springer, 2014, pp. 159–197.

[121] D. Mao, K. Potty, and J. Wang, "The impact of power-electronics-based load dynamics on large-disturbance voltage stability," in *2018 IEEE Power Energy Society General Meeting (PESGM)*, 2018, pp. 1–5.

[122] J. V. Milanovic *et al.*, "International industry practice on power system load modeling," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3038–3046, 2013.

[123] D. Q. Zhou, U. D. Annakkage, and A. D. Rajapakse, "Online monitoring of voltage stability margin using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1566–1574, Aug 2010.

[124] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv e-prints*, p. arXiv:1412.6980, Dec 2014.

[125] H. Hagmar, L. A. Tuan, O. Carlson, and R. Eriksson, "On-line voltage instability prediction using an artificial neural network," in *2019 IEEE Milano PowerTech*, June 2019 (Accepted), pp. 1–6.

[126] F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient database generation for data-driven security assessment of power systems," *IEEE Transactions on Power Systems*, pp. 1–1, 2019.

[127] P. Mitra, V. Vittal, B. Keel, and J. Mistry, "A systematic approach to n-1-1 analysis for power system security assessment," *IEEE Power and Energy Technology Systems Journal*, vol. 3, no. 2, pp. 71–80, June 2016.

[128] Tensorflow Keras Optimizers: RMSprop. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop

[129] F. Rahimi and A. Ipakchi, "Demand response as a market resource under the smart grid paradigm," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 82–88, 2010.

[130] A. Venzke and S. Chatzivasileiadis, "Verification of neural network behaviour: Formal guarantees for power system applications," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383–397, 2021.

[131] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1–7.

[132] Z. Zhang *et al.*, "Physics-constrained robustness verification of intelligent security assessment for power systems," *IEEE Transactions on Power Systems*, pp. 1–1, 2022.

[133] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[134] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *2020 IEEE Power Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[135] G. E. Karniadakis *et al.*, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[136] I. Konstantelos *et al.*, "Using vine copulas to generate representative system states for machine learning," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 225–235, 2019.

[137] A. Venzke, D. K. Molzahn, and S. Chatzivasileiadis, "Efficient creation of datasets for data-driven power system applications," *Electric Power Systems Research*, vol. 190, p. 106614, 2021.