공학박사 학위논문

Designing An Active Distribution Network Operation Strategy for Practical Power Systems Based on Safe Deep Reinforcement Learning

실 계통 적용을 고려한 안전 강화학습 기반의 능동 배전망 운영전략에 대한 연구

2023년 8월

서울대학교 대학원

전기 · 정보공학부

오 석 화

Designing An Active Distribution Network Operation Strategy for Practical Power Systems Based on Safe Deep Reinforcement Learning

지도 교수 윤 용 태

이 논문을 공학박사 학위논문으로 제출함

2023년 6월

서울대학교 대학원

전기 · 정보공학부

오 석 화

오석화의 공학박사 학위논문을 인준함

2023년 6월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

Abstract

Designing An Active Distribution Network Operation Strategy for Practical Power Systems Based on Safe Deep Reinforcement Learning

Seok Hwa Oh
School of Electrical and Computer Engineering
The Graduate School
Seoul National University

The primary objective of this research is to propose a methodology for Distribution System Operators (DSOs) to maintain real-time security in a distribution system infused with renewable energy sources, along with the provision of a safe reinforcement learning (RL) approach to resolve this as an optimal operation problem.

To this end, the research initially assesses the need, authority, and role of the DSO in a distribution system

environment transformed by the penetration of renewable energy sources and defines the objective of active system operation for maintaining system stability by the DSO. Furthermore, this study endeavors to integrate the actual physical system and the system in a simulation environment into a single Cyber-Physical System (CPS), defining the distribution power system environment where real-time network management of DSO takes place.

The assets or devices that the DSO can control vary according to each system environment; however, this study focuses intensively on two controlling options: distribution system reconfiguration through switches within the system and alternation of system current flow using energy storage devices, formalizing each of these as optimization problems.

Moreover, this research considers these optimization problems as control decision problems over continuous time and reformulates them as Markov Decision Processes (MDPs), designing a reinforcement learning algorithm to resolve them. Depending on the objectives and the characteristics of the target data in the field of reinforcement learning (RL), a variety of algorithm forms can be designed. In this study, a Dueling Deep Q-learning algorithm suitable for the proposed system operation

methodologies is designed.

While numerous studies have sought to resolve power system operation issues using reinforcement learning algorithms, this study scrutinizes problems that can occur when applying these beyond the simulation environment to practical power systems. Among these, the study extends the original MDP to a Constrained Markov Decision Process (CDMP) to handle safety constraints of control decisions required in actual systems within the reinforcement learning algorithm, designing a safety module to handle equality constraints and an adaptive cost function for inequality constraints.

Consequently, by simulating the designed safe reinforcement learning model in the IEEE 123-bus test system, it is proven to display more effective performance when considering multiple operation strategies simultaneously and taking into account the safety of reinforcement learning.

By adopting the reinforcement learning framework proposed in this paper for real-time distribution system operation, DSOs can design reinforcement learning algorithms to resolve the requirements derived from physical power systems. Furthermore, by ensuring that the algorithm's decision minimizes

the violation of physical system stability constraints, it can be utilized as one of the system operation strategies to counteract the increasing complexity of the distribution system.

Keyword : Safe Reinforcement Learning, Active Distribution System Operator, Cyber-Physical System, Distribution Network Reconfiguration, multi-ESS Operation.

Student Number : 2017-26165

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1. Background and Motivation

In this section, the necessity of an active distribution system operator (DSO) in future distribution power system environments that include numerous distributed renewable energy sources (DRESs) is articulated. Further, the necessity for short-term operation of distribution networks within such environments and its specific contents will be discussed. It is demonstrated that to implement this, the physical power system in the real world coupled with the simulation environment can be defined as a single Cyber-Physical System (CPS).

1.1.1. Need For Active Distribution System Operator

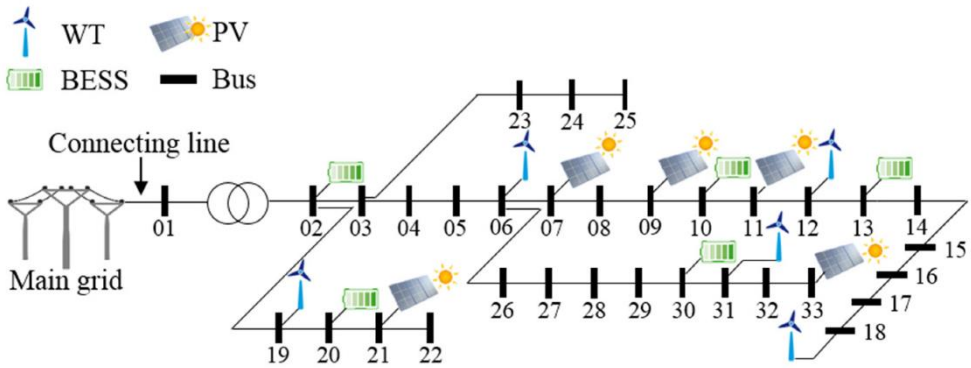

Figure 1.1 IEEE 33-bus distribution network with a high penetration of distributed renewable energy sources [1]

There are several ongoing significant transformations of contemporary power system, a transformation that is poised to intensify in the future. The key trends catalyzing this transformations can be encapsulated as follows [2]:

- Changes in the sources and characteristics of electricity generation

Transitioning from large-scale power plants towards smaller, distribution renewable energy sources (DRESs) across all system levels is significantly affecting the traditional electricity generation landscape. This shift, paired with the decommissioning of large thermal power plants, introduces

uncertainties in power generation and reduces system inertia due to the common use of power electronic converters in DRES generators, thereby complicating frequency regulation and control.

- New types of electricity loads and transformations in load profiles

Emerging load types, such as electric vehicles, along with changing load profiles influenced by the ability to generate electricity locally, the application of electronics and controls in residential, commercial, and industrial settings, and the increasing involvement of loads in electricity markets and power system control are reshaping the demand side of the electricity equation.

- The advent and implementation of smart grid technologies

Progress in communication infrastructure, innovative instrumentation and measurement technologies, and the surge in accessible data due to devices such as phasor measurement units (PMU) and smart meters is transforming the way power systems are monitored and managed.

- The rise of microgrids and energy communities

Emerging as unique entities within power systems, these groups, consisting of interconnected loads and DRES-based power generation within a specific area, function as a single controllable unit with respect to the grid. Their ability to operate both in grid-connected mode and as autonomous entities provides an additional layer of flexibility in power system control, particularly in restoration operations.

- The proliferation of electricity storage technologies

From large-scale storage connected to the transmission system to smaller devices linked to the distribution system, microgrids, energy communities, and individual load sites, storage technologies have emerged as key enablers for future power system operation. They offer the potential to mitigate generation-load imbalances under uncertain conditions, while also serving as crucial control devices throughout various power system operational states.

– Need for Active Network Management and Distribution System Operator

Traditionally, distribution networks have been passive systems receiving power from transmission networks and delivering it to loads. However, risk of new problems threatening distribution-system reliability and stability is increasing as DRESs have become increasingly introduced. To prevent such situations, DRES grid-connection rules conventionally have been designed according to the "fit-and-forget" principle, which is a planning approach to managing distribution networks to ensure high network reliability in worst-case operating scenarios without requiring any active control actions. However, because this approach may lead to distribution-network overinvestment, it is not economically feasible to adopt rapidly expanding DRESs [3].

In recent years, technological and institutional advances have led to a new phase in distribution system operation called "Active Network Management" (ANM), thereby ensuring the reliability of distribution networks while efficiently increasing hosting capacity. Active management in a distribution system is defined as short-term control and management of the system

elements including DRESs and energy storage systems (ESSs), using advanced information and communications technology (ICT) infrastructure for on-line network status measurements and bidirectional communication between all the network elements and a central operating system [4], ANM is a comprehensive concept including active fault management, active voltage control, and active power flow management, and various techniques have been studied to implement ANM, which has enabled distribution system operators (DSOs) to solve various network-constraint problems with operation-level solutions not included in planning-level principles. Furthermore, [4] claimed that network management at the operation stage was less expensive and, therefore, more economical.

1.1.2. Cyber-Physical System Framework for Power System Operation

- General CPS Concept [7]

A Cyber-Physical System (CPS) represents a paradigmatic

product emerging from the Industry 4.0, assuming a critical function due to its capacity to intertwine the physical and virtual realms through the provision of real-time data processing services [8]. A CPS, more explicitly, facilitates the equipping of a physical system with a virtual system, serving as a monitoring mechanism. It allows for the analysis of data procured from the physical environment within the virtual domain, thereby informing decisions that influence the trajectory of the physical world. Consequently, a CPS enables the consolidation, dissemination, and collaboration of information, along with real-time monitoring and global optimization of systems [9]. Modern industry boasts an array of applications based on CPSs, encompassing areas such as smart grids, healthcare, aviation, digital manufacturing, and robotics. Evidence in the literature suggests that CPS incorporates a variety of facets including, but not limited to, Networked Control Systems (NCSs), wireless sensor networks, and smart grids.

The constitution of a CPS involves a physical system and a cyber system, derived from the integration of physical processing, sensing, computation, communication, and control [10]. The typical architecture of a CPS is depicted in Figure 1.2.

The physical system comprises physical processes, sensors, and actuators, while the cyber system entails communication networks, computing, and control centers. Physical processes are customarily perceived as a plant under the control of a cyber system. Concerning the other components, their functionalities are as follows:

1) Sensors: These are employed for the acquisition of real-time data.
2) Actuators: The execution of control commands by corresponding actuators facilitates the realization of desired physical actions.
3) Computing and control center: This component is tasked with receiving data measured by sensors. Through the analysis of the received data, the control center formulates relevant control decisions, ensuring the correct execution of physical processes.
4) Communication network: This component furnishes a communication platform for the control center and physical system. Specifically, measurements obtained by sensors are transmitted over the communication network to the

control center. Control signals or decisions are subsequently relayed from the control center to actuators via the communication network.

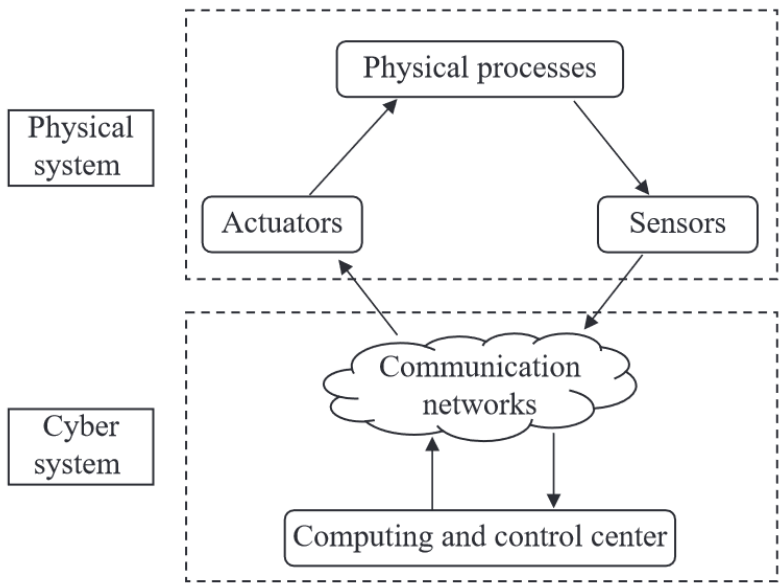

Figure 1.2 General CPS architecture [7]

- Cyber-Physical Power System

The application of CPS concept to power systems also has been a long-standing research topic. Around the turn of the millennium, there was a demand for future energy systems capable of the modular integration of DRES into power systems, as well as the implementation of customer choices by energy

users. In particular, [11] articulated the necessity of systemically embedding cyber technologies that can monitor, communicate, and control the physical system to adaptively meet objectives such as flexibility, efficiency, sustainability, reliability, and security, in response to the time-variant system states of these future energy systems. Moreover, they modularized power plants and loads into a generalized form, defined an interaction control protocol between each module by the system operator, facilitating system-wide controllability and stabilization, and proposed a cooperative sensing and communication scheme ensuring system-wide observability.

Such a module-based cyber-physical energy system was depicted as shown in Figure 1.3 by [12].

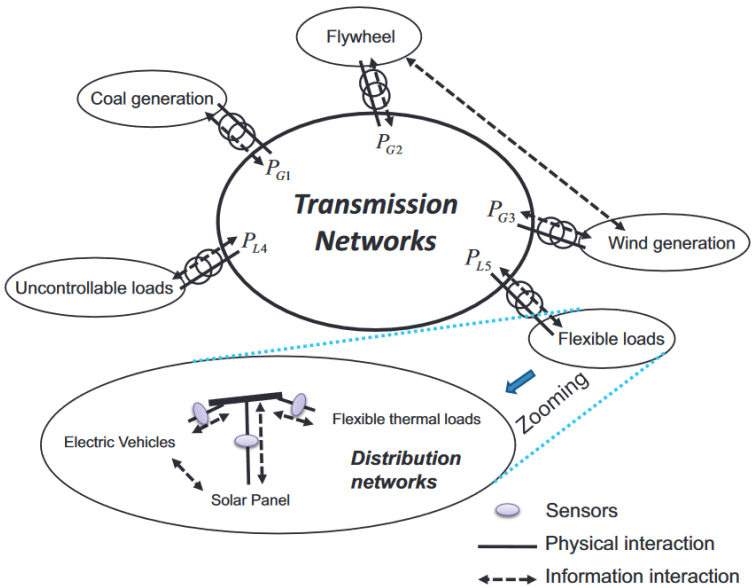

Figure 1.3 A module-based architecture for cyber-physical energy systems [12]

To construct a Cyber-Physical Power System as described above, it is necessary to formalize a dynamic model based on the physical characteristics of each component, with a limited number of sensible states and well-defined action inputs. This task is extensively carried out in the aforementioned prior studies. However, discussing it in detail falls outside the scope of this dissertation. It is assumed here that a DSO, based on such a Cyber-Physical Power System, can collect the necessary information of the network at intervals ranging from several minutes to an hour, as well as the DSO can take control decisions

and/or actions to maintain network stability.

1.1.3. Role and Structure of the Proposed Distribution System Operator

- Power system states and control action of DSOs

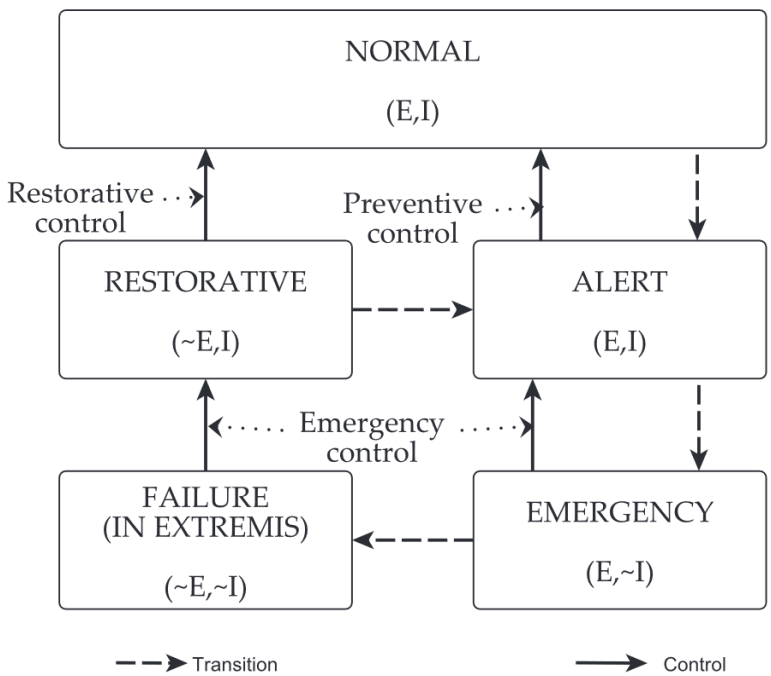

Figure 1.4 Power system operating states [2]

Electric power systems encounter diverse control challenges

across various operational states and time frames. The categorization of these operating states most commonly adopted is proposed in [5]. which defines states based on the system's adherence to equality (E) and inequality (I) constraints. As depicted in Figure 1.4, each state of the power system is delineated based on whether these constraints are satisfied, signified by the symbol '~' if not. Equality constraints represent the balance between generation and load demand, while inequality constraints convey the physical limits of the power system components. These limitations are typically outlined in terms of current and voltage magnitudes, as well as active, reactive, and apparent powers that a system component can bear without damage [6]. Figure 1.2 further exhibits the control mechanisms employed within electric power systems. Beyond preventive, emergency, and restorative controls, control in the normal operating state is also crucial due to the continuous minor variations in generations and loads observed in this state [2].

In this research, the role defined for the DSO is to maintain the power system in Normal state. It is posited that the target distribution power system could transition into an Alert or even Emergency state due to unpredictable DRES generation or load

patterns. Consequently, the DSO holds the authority and responsibility to execute Preventive and Emergency controls in such situations. It should be noted that failure and restorative states, which occur when equality constraints are violated, along with associated emergency and restorative controls, undoubtedly fall within the purview of the DSO's responsibilities. However, as they extend beyond the scope of this research, they will not be covered in this research.

- An integrated CPS Framework for power system operation

There are distinct advantages, and indeed necessity in this research, to applying the previously defined concept of CPS to power system operation. The concept of power system operation includes not only the actual manipulation of physical components but also setting and changing control objectives that have a significant impact on system reliability, both of which fall under the role of an active DSO. In other words, the active DSO is defined across the physical and cyber layers from the perspective of a CPS and is responsible for communication and interpretation between these two layers. However, many

previous studies tend to use concepts and operating actions belonging to each layer interchangeably.

Particularly when, as targeted in this research, a DSO intends to include such as artificial intelligence models in the operating system, which are not physical components or their corresponding simulated ones, this is not feasible from the perspective of simple power system simulation. In this research, following [13], we propose to redefine the power system by distinguishing it into the physical layer, information-based cyber layer, and the mediating interaction layer, for which we propose the following structure.

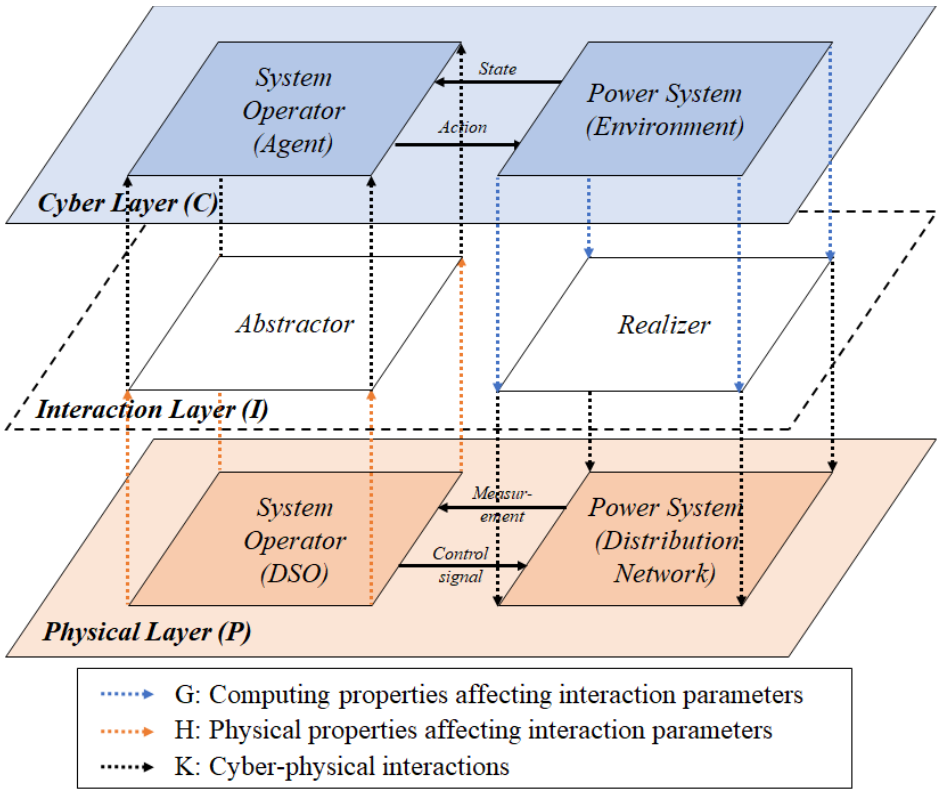

Figure 1.5 Proposed CPS structure for a distribution power system

The Cyber-Physical Power system in this dissertation is defined by three layers: the physical layer composed of physical components, the cyber layer based on the simulated model of the physical components, and the interaction layer that facilitates communication and interpretation of information and signals between the two.

Specifically, the physical layer includes traditional power system components like generators, loads, lines, and DRES such as PVs, WTs, ESSs. Moreover, considering the focus on the distribution power system, the interconnection point with the transmission power system is also included. The cyber layer consists of a simulated system, taking into account the controllability and visibility of the DSO, which is grounded on the physical power system of the physical layer. In the context of the CPS, the simulated system focuses more on controlling the physical system rather than constructing a 'mirroring model' that faithfully reproduces the physical system. This naturally leads to the role of the interaction layer in the proposed CPS. The interaction layer defines the computing and communication functionalities of the given CPS while facilitating mapping, real-time interaction, and, consequently, effective coordination between the physical layer and the cyber layer, which do not necessarily correspond one-to-one.

Figure 1.5 provides a graphical summary of the above-mentioned content. The elements composing the Figure 1.5 are as follows [13].

C : Cyber layer, or computing properties of a CPS

P : Physical layer, or physical properties of a CPS

I : Interaction layer, or Interaction parameters of a CPS

$G: C \times t \rightarrow I$: mapping between the sets C and I

$H: P \times t \rightarrow I$: mapping from the P to I

K: Cyber-Physical interaction; inverse mapping from a subset of I to a subset of P or C.

In the context of CPS operation, [13] argues that it necessitates the assurance of three key properties, collectively referred to as S3: safety, security, and sustainability. Among these, this dissertation will primarily focus on the aspect of safety (avoidance of hazards). Issues related to security (assurance of integrity, authenticity, and confidentiality of information) and sustainability (maintenance of long-term operation of CPSs using green energy sources) such as the much-discussed cyber-attack, i.e., false data injection in recent ICT-based power system operations, exceed the scope of this paper.

The emphasis here is that when the DSO manages and handles the physical power system, it always transpires within a cyber-physical power system through the interaction layer. Although an

informational gap exists between the cyber layer and the physical layer in CPS, the CPS discussed in this dissertation is assumed to be a closed one, precluding the intrusion of external signals into the gap. Thus, the data collected from the physical layer and arrived at the cyber layer can always be trusted, and contrarily, control or operation signals transmitted from the cyber layer can always be received by the corresponding parts of the physical layer.

1.2. Main Contribution

As explained in the previous section, today's distribution networks are necessitating a shift from traditional passive to proactive, active network operation strategies due to their evolving environment. This shift occurs not only in components requiring control within the network but also where these controls' timescales vary, indicating the need for a DSO capable of supervising and aligning these controls. Consequently, this dissertation explores a novel active distribution network operation strategy from the DSO's perspective, and the primary contributions are as follows:

- This dissertation has formalized problems of active network management methods using assets owned by DSO, specifically distribution network reconfiguration and multi-BESS operation, as optimization problems.
- This dissertation has demonstrated that the defined problems can be reformulated as corresponding Markov Decision Problems, allowing the introduction of reinforcement learning for solving them.
- This dissertation has designed reinforcement learning algorithms and neural network structures suitable for each problem's characteristics.
- This dissertation has examined potential issues when applying RL algorithms to safety-critical power systems and introduced a safe reinforcement learning framework. We designed safety algorithms to correspond appropriately to each system constraint and formalized them for each control method.

1.3. Dissertation Outline

This section provides the dissertation outline. Chapter 1

elucidates the necessity for Active Network Management in response to the dynamic changes in the distribution system environment and discusses the active DSO, the entity responsible for its execution. It further redefines the power distribution system as a Cyber-Physical System, conceptualized as a framework for the DSO to operate within. In Chapter 2, we describe the ANM methodologies that the DSO can deploy utilizing its own assets. We articulate the modeling and formulation of optimization problems with respect to Network reconfiguration, enabled by sectionalizing switch, and the preservation of system stability through the utilization of multi-ESS. Further, we scrutinize strategies for joint operation when more than one methodology can be employed. Chapter 3 introduces reinforcement learning as a method from the DSO's perspective for solving problems defined in Chapter 2, and redefines each ANM method within the RL framework. Chapter 4 evaluates the potential issues that may arise when seeking to apply artificial intelligence algorithms, including reinforcement learning, to physical power systems, and examines safe RL as a methodology to address these challenges. It also designs safety algorithms that can be applied to the RL model proposed in Chapter 3. Chapter 5 designs and conducts a case study to evaluate the methodologies

proposed previously and analyses the results. Chapter 6 offers concluding remarks along with suggestions for future research extensions. The Appendix presents detailed data for simulation and mathematical techniques.

Chapter 2. Short-term Network Operation of DSO

In this chapter, we examine short-term network control methods that a DSO can execute using its own assets with the objective of maintaining network stability. Specifically, we will investigate two methods: Distribution Network Reconfiguration (DNR) using sectionalizing switches and the operation of multiple Energy Storage Systems (ESS).

2.1. Distribution Network Reconfiguration

2.1.1. Concept of Distribution Network Reconfiguration

In self-sufficient distribution systems wherein most of the demand can be supplied internally by DRESs, system status can change considerably in a short time window owing to the variation of DRES generation outputs, which usually depends on the unpredictable weather conditions. To promptly cope with the distribution network constraint violation problem occurring in

normal operation situations, the presence of a DSO is necessary. The DSO can change the power flow by constructing another topology while maintaining the radiality of the distribution system by changing the switch status online. In principle, it takes a way to alleviate the line overflow or bus voltage problem by connecting more loads to the branch where the output of DRESs exceeds the tolerable power generation. From a practical viewpoint, the online network reconfiguration requires a distribution power system with several normally opened switches that can be remotely controlled (i.e., remotely controlled switches (RCSs)), which DSOs have the authority to operate, and a surveillance system such as SCADA to collect the network status data. In this case, the DSO can use the online collected data as the input of the algorithm proposed in this paper and can manipulate the RCSs of the distribution network with the outputted optimal topology of the network. It is expected that this will be done by automated ANM systems.

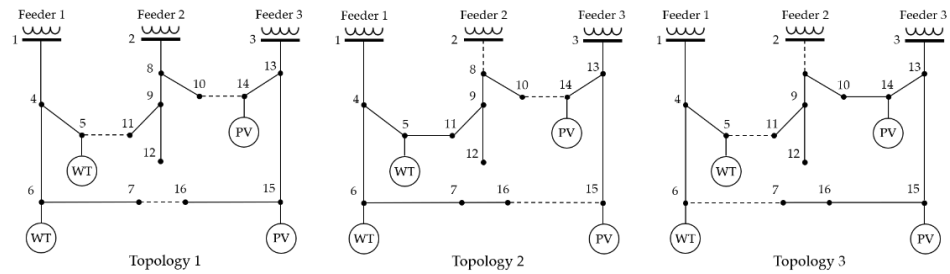

Figure 2.1 Simple power distribution network with DRESs

Assume that DRESs such as wind turbines (WTs) and photovoltaics (PVs) have been installed in a 3-feeder simple power distribution network [15], as shown in Figure 1, and the network is in the normal operation as in Topology 1. If the WT generation is temporarily increased, a reverse flow which may violate the thermal constraint occurs at Line 1-4 under Feeder 1. Existing DSOs can overcome this problem by 1) investing a large budget to increase line capacity by reinforcing Line 1-4, or 2) curtailing the generation of WTs and compensating DRES owners for generation opportunity costs. If online network reconfiguration is possible, this threat can be solved by changing the network switching status and reconfiguring the system topology as Topology 2. In another case, if the power distribution system is operating in Topology 2 and the PV generation is very

high while the WT generation is low, it can threaten the thermal constraint on Line 3–13. Similarly, this problem can be solved by changing system topology to Topology.

Because finding the optimal topology through network reconfiguration is basically a nonlinear combinatorial optimization problem, it is very difficult using existing optimization solvers to compute network topologies fast enough to apply them to online network reconfigurations. Therefore, the brute–force search method that solves the power flow for all possible switch statuses at every timestep is unsuitable because computation time exponentially increases according to number of switches. Previous studies have used various heuristic algorithms such as greedy algorithms, genetic algorithms (GAs), evolution programming (EP), artificial neural networks (ANNs), and branch exchange algorithms, or approximated the problem to break it down. However, only a few studies have attempted to access distribution network reconfiguration online. Reference [16] has proposed the online approach, their study merely extended previous research while using the existing heuristic method proposed by [17] to minimize the objective function which is composed of energy loss, Expected System Average

Interruption Frequency Index (ESAIFI), Expected Energy Not Supplied (EENS).

2.1.2. Modeling and Formulation of Network Reconfiguration

- Objective Function

$$\min_{\alpha_t} \sum_{t} C_t^{GC} + C_t^{SW} \tag{2.1}$$

C_t^{GC} : Generation curtailment cost

C_t^{SW} : switch operation cost

- Radiality constraints:

Most distribution systems must operate in radial topologies to facilitate the coordination of the protection system. To formally define the radiality of a power network, it is necessary to characterize it as an undirected graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$, where $\mathcal{N}$ represents the set of buses, and $\mathcal{N}$ signifies the set of lines, including lines with a sectionalizing switch.

The spanning tree constraints [18] are adopted here to ensure network radiality, that is, maintain the graph $\mathcal{G}$ as a forest

structure, in each time period:

$$\beta_{ij,t} + \beta_{ji,t} = \alpha_{ij,t}, \quad \forall (i,j) \in \mathcal{E}, t \in T \tag{2.2}$$

$$\sum_{j:(i,j)\in\mathcal{E}} \beta_{ij,t} = 1, \quad \forall i \in \mathcal{N} \setminus \mathcal{R}, t \in T \tag{2.3}$$

$$\beta_{ij,t} = 0, \quad \forall i \in \mathcal{R}, (i,j) \in \mathcal{E}, t \in T \tag{2.4}$$

where $\alpha_{ij,t} \in \{0,1\}$, 1 if line (i,j) is connected in time t, 0 otherwise, and $\beta_{ij,t} \in \{0,1\}$, 1 if node j is the parent node of node i in time t, 0 otherwise. $\mathcal{R}$ is set of substation nodes which connected with external grid such as transmission network. Note that, since variables $\beta_{ij,t}$ are set as binary, Eq. (2.2)−(2.4) allow us to treat $\alpha_{ij,t}$ as continuous variables by adding:

$$0 \le \alpha_{ij,t} \le 1, \quad \forall (i,j) \in \mathcal{E}, t \in T \tag{2.5}$$

We will define a set $\mathcal{E}^{rad}$ which is composed with radial topologies of the network $\mathcal{G}$, that is, if $\boldsymbol{\alpha} = \{\alpha_{ij} | \alpha_{ij}, \forall (i,j) \in \mathcal{E}\}$ satisfies Eq. (2.2)−(2.5), $\boldsymbol{\alpha} \in \mathcal{E}^{\mathrm{rad}}$.

- DRES constraints

The DRESs addressed in this paper consist of PVs and WTs,

both of which are treated as static generators. The original generation values $\hat{P}_{t,i}^{DRES}$ and $\hat{Q}_{t,i}^{DRES}$, which are dependent on weather conditions, cannot be determined by the DSO. However, in this research, we assume that the DSO can curtail a portion or all of these generation if needed for network stability. In this case, the DSO must pay a curtailment cost C_t^{GC}, proportional to the curtailment amount, to each DRES owner. Consequently, the final generation after curtailment process of each DRESs are represented as $P_{t,i}^{DRES}$ and $Q_{t,i}^{DRES}$.

$$0 \le P_{t,i}^{DRES} \le \hat{P}_{t,i}^{DRES} \le \bar{P}_i^{DRES} \tag{2.6}$$

$$0 \le Q_{t,i}^{DRES} \le \hat{Q}_{t,i}^{DRES} \le \bar{Q}_{t,i}^{DRES} \tag{2.7}$$

$$\left(\hat{P}_{t,i}^{DRES}\right)^2 + \left(\hat{Q}_{t,i}^{DRES}\right)^2 \le \left(\bar{S}_i^{DRES}\right)^2 \tag{2.8}$$

- Power flow constraints

The active DSO can obtain state information, such as the voltage and angle at each bus. Hence, the power flow related constraints can be included in the optimization as follows:

$$P_{t,i}^{T} + P_{t,i}^{DRES} + P_{t,i}^{ESS} - P_{t,i}^{L} - P_{t,i}^{F} = 0, \qquad \forall i \in \mathcal{R} \tag{2.9}$$

$$Q_{t,i}^{T} + Q_{t,i}^{DRES} + Q_{t,i}^{ESS} - Q_{t,i}^{L} - Q_{t,i}^{F} = 0, \qquad \forall i \in \mathcal{R} \tag{2.10}$$

$$P_{t,i}^{DRES} + P_{t,i}^{ESS} - P_{t,i}^{L} - P_{t,i}^{F} = 0, \qquad \forall i \in \mathcal{N} \setminus \mathcal{R} \tag{2.11}$$

$$Q_{t,i}^{DRES} + Q_{t,i}^{ESS} - Q_{t,i}^{L} - Q_{t,i}^{F} = 0, \qquad \forall i \in \mathcal{N} \setminus \mathcal{R} \tag{2.12}$$

where

$$P_{l,t}^{F} = |V_{t,i}| \sum_{j} |V_{t,i}| \left[G_{ij,t} \cos(\delta_{t,i} - \delta_{t,j}) + B_{ij,t} \sin(\delta_{t,i} - \delta_{t,j}) \right] \tag{2.13}$$

$$Q_{l,t}^{F} = |V_{t,i}| \sum_{j} |V_{t,i}| \left[G_{ij,t} \sin(\delta_{t,i} - \delta_{t,j}) - B_{ij,t} \cos(\delta_{t,i} - \delta_{t,j}) \right] \tag{2.14}$$

Unlike other buses in the distribution system, buses belonging to the root node can receive active and reactive power from an external power grid such as the transmission system, represented by $P_{t,i}^{T}$ and $Q_{t,i}^{T}$ respectively. Power balancing equations must be satisfied at all bus i in the system.

Moreover, the power flows in the distribution lines should not exceed their respective line capacities. These constraints can be represented as follows:

$$\left(P_{l,t}^{F}\right)^2 + \left(Q_{l,t}^{F}\right)^2 \leq (\bar{S}_l)^2 \tag{2.15}$$

where $\bar{S}_l$ is capacities of each distribution line l.

Finally, the voltages of all buses in the distribution system should be within the regulation range as follows:

$$V^{LL} \leq |V_{t,i}| \leq V^{UL} \tag{2.16}$$

where V^{LL} and V^{UL} are the lower limit and the upper limit of the voltage regulation range, respectively.

- Switch Operating Action Modeling

The switching action SW consists of an array with 0s (to open switches) and 1s (to close them). If the DSO wants to extract the action from the DQN prediction, output-layer values first must be arranged into an array form and then replaced with 1s as many switches as the DSO wants to close in the order of highest values, and the remaining elements are replaced with 0s. The determined action is then input to the test system to implement and activate or deactivate each switch and to update SW_{tr} by comparing the switch status to that of the previous action stored in the log. The log-update formula is as follows:

$$ch_i^t = 1 - \mathrm{XOR}\left(sw_i^t, sw_i^{t-1}\right) \tag{2.17}$$

$$SW_{tr}^t = \left\{ sw_{tr,i}^t = \min\left(T_{sw}, ch_i^t \cdot \left(sw_{tr,i}^{t-1} + 1\right)\right) \middle| \forall i \in N_{SW} \right\} \tag{2.18}$$

where ch_i^t indicates whether the operation state sw_i^t at time t differs from sw_i^{t-1} at time $t-1$. If both values are the same, ch_i^t has a value of 1 and increments $sw_{tr,i}^t$ by 1 as long as it is smaller than T_{sw}. One the contrary, if ch_i^t has a value of 0, it resets $sw_{tr,i}^t$ to 0.

2.2. DSO-owned Multi-ESS Operation

2.2.1. Concept of DSO-owned ESS

The function of ESS is to store electrical energy and subsequently supply it to power grids when necessary [19]. There is a wide array of methods by which ESS can be utilized within the distribution power system, which can be broadly categorized into two groups, namely, technical-support oriented and profit-making oriented. This research is described from the perspective of the DSO, a system operator that does not pursue profits; therefore, we concentrate on the former. In this case, ESS are considered as assets of power grids, with sizing and placement decisions aimed

at enhancing system performance, such as reducing frequency deviation, peak shaving, voltage support, and integration of DRESs, among others. These applications vary with ESS discharge duration and power capacity, as illustrated in Figure. 2.2 [20]. Despite their diverse applications, ESS are often considered from the perspective of planning costs, rather than operation, due to the high initial costs and the stability issues at the device level. However, in this study, we will assume that the DSO holds the authority and responsibility for controlling the ESS located at a given location within the distribution network, meaning we will not consider costs from a planning perspective, including degradation costs.

2.2.2. Modeling and Formulation of multi-ESS Operation

- Objective function

$$\min_{P_{e,t}} \sum_{t} C_t^{GC} + C_t^{ESS} \tag{2.19}$$

C_t^{GC} : Generation curtailment cost

C_t^{ESS} : ESS operation cost

- BESS Constraints

A BESS consists of a battery cell, which charges/discharges a direct current, and a power conditioning system (PCS), which converts a direct current into an alternating current.

$$0 \le P_{e,t}^{ch} \le \bar{P}_e^{ch} u_e^{ch}, \qquad e \in E \tag{2.20}$$

$$0 \le P_{e,t}^{dch} \le \bar{P}_e^{dch} u_e^{dch}, \qquad e \in E \tag{2.21}$$

$$P_{e,t} = P_{e,t}^{ch} u_e^{ch} - P_{e,t}^{dch} u_e^{dch}, \qquad e \in E \tag{2.22}$$

$$u_e^{ch} + u_e^{dch} = 1, \qquad u_e^{ch}, u_e^{dch} \in \{0,1\} \tag{2.23}$$

The state-of-charge (SOC) of the BESS should remain within the ranges of the following equation to avoid damage to the battery due to deep discharging and overcharging.

$$SOC_e^{min} \le SOC_{e,t} \le SOC_e^{max} \tag{2.24}$$

Also, SOC over time can be calculated as follows:

$$SOC_{e,t} = SOC_{e,t-1} + \frac{\eta_e^{ch} P_{e,t}^{ch} u_e^{ch}}{Batt_e} - \frac{P_{e,t}^{dch} u_e^{dch}}{\eta_e^{dch} Batt_e} \quad (2.25)$$

$$SOC_{e,0} = SOC_e^{init} \quad (2.26)$$

where $\eta_e^{ch} \in (0,1)$ and $\eta_e^{dch} \in (0,1)$ are charging efficiency and discharging efficiency of BESS, respectively.

Chapter 3. Reinforcement Learning for Distribution Network Operation

3.1. Concept of Reinforcement Learning

- Preliminary

Reinforcement Learning is a method of learning through trial-and-error, which entails 1) direct interaction with the environment, 2) self-education over time, and 3) ultimate attainment of a predefined goal. Specifically, Reinforcement Learning designates any decision-maker (learner) as an "agent" and all elements external to the agent as the "environment." The dynamics of interaction between the agent and the environment are characterized through three essential elements: 1) state s, 2) action a, and 3) reward r [21].

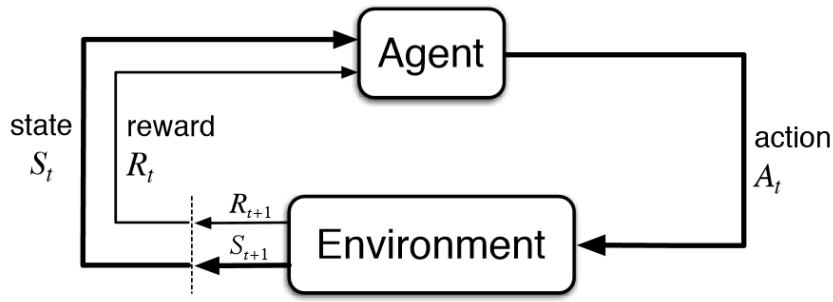

Figure 3.1 Reinforcement learning architecture [21]

At any given timestep t, the state of the environment is denoted as s_t. Thus, the agent examines s_t and takes a corresponding action a_t. In response, the environment modifies its state from s_t to s_{t+1} and furnishes the agent with a feedback reward r_t. The decision-making process of the agent is formalized through the definition of a "policy."

A policy π is a mapping function that links any observed state s to the action a taken from that state. A policy is termed deterministic if, for all states s, the probability of selecting an action a is equal to 1, i.e., $p(a|s) = 1$. Conversely, the policy is deemed stochastic if there exists one or more state s such that $p(a|s) < 1$. Regardless of the case, we can represent the policy π as a probability distribution of potential actions selected from a

specific state.

$$\pi(s) = \{p(a_i|s) \mid \forall a_i \in \mathcal{A} \ \wedge \sum_i p(a_i|s) = 1\} \tag{3.1}$$

In the equation, $\mathcal{A}$ represents the action space, or all potential actions, of the policy π. For ease of understanding, we assume that the action space is discrete, given that the case for continuous action spaces can be directly inferred using integral notation. Moreover, it is presumed that the next state s_{t+1} and feedback reward r_t are entirely determined by the current state−action pair (s_t, a_t), irrespective of prior history. Any Reinforcement Learning problem that satisfies this "memoryless" condition is known as a Markov Decision Process (MDP). Thus, the dynamics, or the model, of a Reinforcement Learning problem are fully defined by specifying all transition probabilities $p(a_i|s)$.

- The definition of Markov Decision Process

In the framework of Reinforcement Learning (RL), we consider a prototypical MDP encapsulated by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where the elements stand for the state set $\mathcal{S}$, action set $\mathcal{A}$, transition probability function $\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, discount factor γ in the interval [0, 1], respectively. An agent

navigates through the MDP, adhering to its policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which maps states to actions.

The overarching aim of an RL agent entails optimizing its policy to ensure the maximization of the expected cumulative discounted reward, formally denoted as $J(\pi) = \mathbb{E}_{\pi}[\sum_t \gamma^t r_t]$, where t ranges from 0 to the horizon T. Here, s_0 follows the initial state distribution $\rho_0(s_0)$, a_t is derived from the policy $\pi(s_t)$, the subsequent state s_{t+1} conforms to the transition probability $p_t = \mathcal{P}(s_{t+1}|s_t, a_t)$, and r_t signifies the reward at time t, given by $r_t = \mathcal{R}(s_t, a_t, s_{t+1})$

In this framework, we define the state−action value function for a policy π, denoted as $Q_{\pi}(s, a)$. It essentially represents the expected cumulative discounted reward when starting in state s_0, performing action a_0, and following policy π thereafter. Thus, $Q_{\pi}(s, a) = \mathbb{E}_{\pi}[\sum_t \gamma^t r_t | s_0 = s, a_0 = a]$.

- The solution of Markov Decision Process

The immediate reward r_{t+1} does not represent the long−term profit, we instead leverage a generalized return value G_t at time step t:

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-t-1} r_T = \sum_{i=0}^{T-t-1} \gamma^i r_{t+1+i} \tag{3.2}$$

where $\gamma \in [0,1]$ is a discounted factor. The agent becomes farsighted when γ approaches to 1 and vice versa the agent becomes shortsighted when γ is close to 0.

We can define a state value function V and state−action value function Q as

$$V(s) = \mathbb{E}[G_t | s_t = s] \tag{3.3}$$

$$Q(s,a) = \mathbb{E}[G_t | s_t = s, a_t = a] \tag{3.4}$$

Here, policy π represents the probability of taking each action a in a given state s, therefore the following relationship is established between the state value function $V(s)$ and the state−action value function $Q(s,a)$.

$$\sum_a \pi(a|s) = 1 \tag{3.5}$$

$$\sum_a \pi(a|s) Q_\pi(s,a) = V_\pi(s) \tag{3.6}$$

The defined state value function and state−action value function, decomposed by immediate reward r_{t+1} and discounted future reward $\sum_{i=1}^{T-t-1}\gamma^i r_{t+1+i}$, is called the Bellman equation. Assuming that $V(s)$ and $Q(s,a)$ follow the policy π, Eq. (3.3) by substituting Eq. (3.2), we can derive the following expression.

$$V_\pi(s) = \mathbb{E}_\pi[r_{t+1} + \gamma V_\pi(s_{t+1})|s_t = s] \tag{3.7}$$

Solving an MDP problem is equivalent to finding the optimal value function V^* and/or optimal state−value function Q^* below, where the policy π is written as the optimal policy π^*.

$$V^*(s) = \max_\pi V_\pi(s) \tag{3.8}$$

$$Q^*(s,a) = \max_\pi Q_\pi(s,a) \tag{3.9}$$

Then, the following relationship holds between Q∗ and V∗.

$$Q^*(s) = \mathbb{E}[r_{t+1} + \gamma V^*(s_{t+1})|s_t = s, a_t = a] \tag{3.10}$$

The MDP, when considered as a general stochastic control problem, can be resolved in principle through Dynamic

Programming (DP). The optimal policy in DP represents the solution to an optimization problem with an objective function designed to minimize the current cost and future expected value of the cost-to-go. If the expected value function can be computed exactly and is tractable, conventional optimization methodologies can be employed to obtain the optimal solution. However, if this is not the case, issues arise regarding the guarantee of the solution's optimality.

On the other hand, Approximate Dynamic Programming (ADP) [22] refers to a heuristic solution method for solving problems by approximating the value function in dynamic programming or by searching for policies within a parametric family. If the value function is approximated using a deep neural network, this is equivalent to deep reinforcement learning methodologies [23].

- Deep Q Network

Deep reinforcement learning (deep RL) is a subfield of machine learning that combines RL and deep learning. When the Value functions for the proposed MDP problem are high dimensional objects, we can use a deep Q-network $Q(s, a; \theta)$ with parameters θ to approximate them. To estimate this network, we optimize the

following sequence of loss functions at iteration i.

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta_i)\right)^2\right] \tag{3.11}$$

where θ^- represents the parameters of a fixed and separate target network, which periodically copies parameters of online network [24]. The update of the online network parameters is done by gradient descent, and the specific gradient value can be obtained as follows:

$$\begin{aligned}\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,a,r,s'}\Big[r + \gamma \max_{a'} Q(s', a'; \theta^-) \\ - Q(s, a; \theta_i)\nabla_{\theta_i} Q(s, a; \theta_i)\Big]\end{aligned} \tag{3.12}$$

This approach is model free in the sense that the states and rewards are produced by the environment. It is also off-policy because these states and rewards are obtained with a behavior policy (epsilon greedy in DQN) different from the online policy that is being learned.

- Dueling Deep Q Network

Within the reinforcement learning, a multitude of Q-network-

based algorithms exist. Nonetheless, in this study, we will adopt the Dueling Deep Q Network (Dueling DQN) presented by [25]. The dueling architecture adopted in this algorithm is noted for its ability to identify the correct action more swiftly during policy evaluation in environments where redundant or similar actions are introduced to the learning problem.

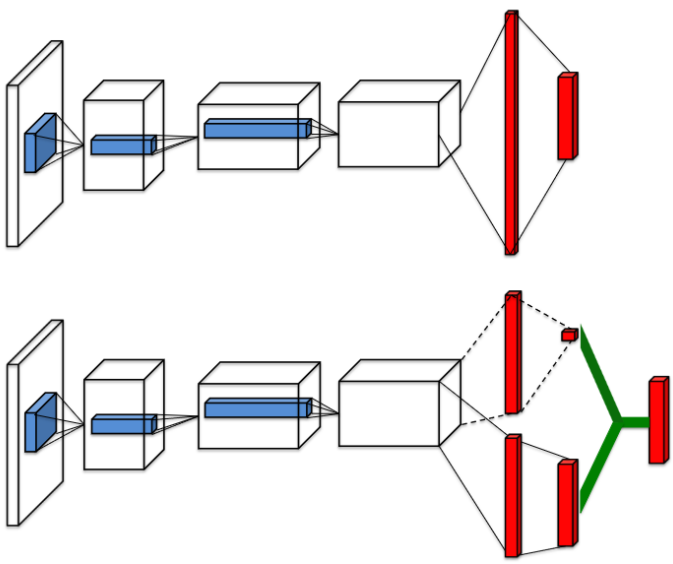

Figure 3.2 Single stream Q-Network (top) and the dueling Q-Network structure (bottom) [25]

The Dueling DQN shares its propensity to approximate the Q-value through a neural network with conventional Q-learning.

However, it builds on the observation that there is no need to estimate the value for each action choice. Consequently, the lower layers of the dueling network remain identical to the original DQN [24], yet from the mid-layer onwards, instead of a single sequence, the network utilizes two sequences (or streams) of fully connected layers. Each of these stream results in the estimation of the state value V and advantage function A, respectively. The advantage function A is defined as follows:

$$A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s) \tag{3.13}$$

Subsequently, the Q-value in the Dueling DQN is obtained as

$$Q(s,a;\theta,\phi,\psi) = V(s;\theta,\psi) + \left(A(s,a;\theta,\phi) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\phi) \right) \tag{3.14}$$

The subtraction of the average of Advantage values is intended for the identification of V and A, which are not determined in Eq. (3.13). While the network structure of the Dueling DQN may seem slightly altered from the original DQN, it is noteworthy that Eq. (3.14) can be inserted as part of the Deep Q-learning algorithm in the existing work by [24] without the

need for any separate algorithmic step.

3.2. Reinforcement Learning for Power System Operation

In this research, we assume that all activities in the DSO are performed on the CPS defined in section 1.1, so we can redraw the original RL architecture depicted in Figure 3.1 as follows.

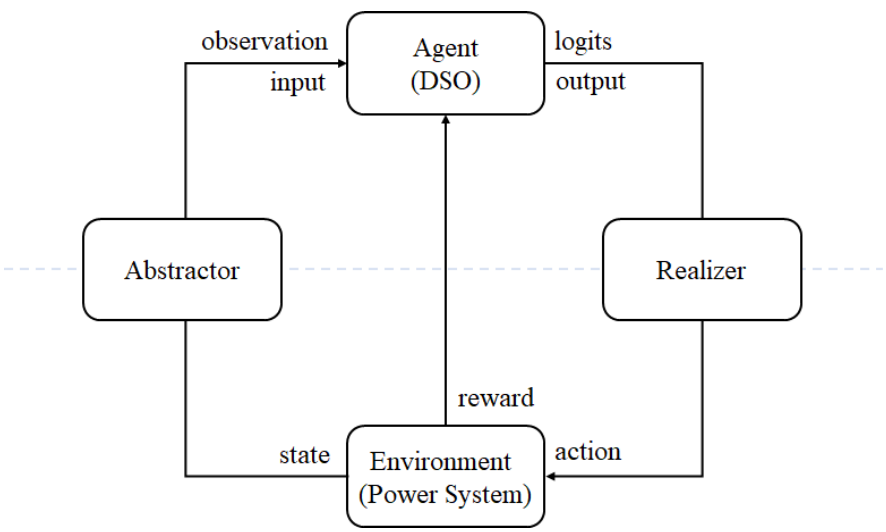

Figure 3.3 RL architecutre in cyber-physical power system

The vanilla RL model intended for DSO in the Figure 3.3 consists of four components, with the 'environment' contained

within the Figure 3.1 corresponding to the simulated power system, and the 'agent' representing the DSO operating this CPS including distribution power system. Furthermore, Figure 3.3 incorporates two additional components termed as the 'Abstractor' and the 'Realizer.' These components not only undertake domain-specific roles but also accentuate the context of the CPS environment.

The *Abstractor* is responsible for data collection and conversion to model input, along with the preprocessing steps, such as data integrity checking and normalization for learning stability. Conversely, the *Realizer* is tasked with transforming the RL model output into a physically meaningful and valid action, including performing network constraint checks, which will be addressed in Chapter 4. Like conventional RL, the proposed RL model also aims to iteratively learn an optimal policy based on the data obtained through the interaction between the DSO, acting as an agent, and the simulated power system, serving as the environment. This process can be described following the procedures within the CPS framework [26].

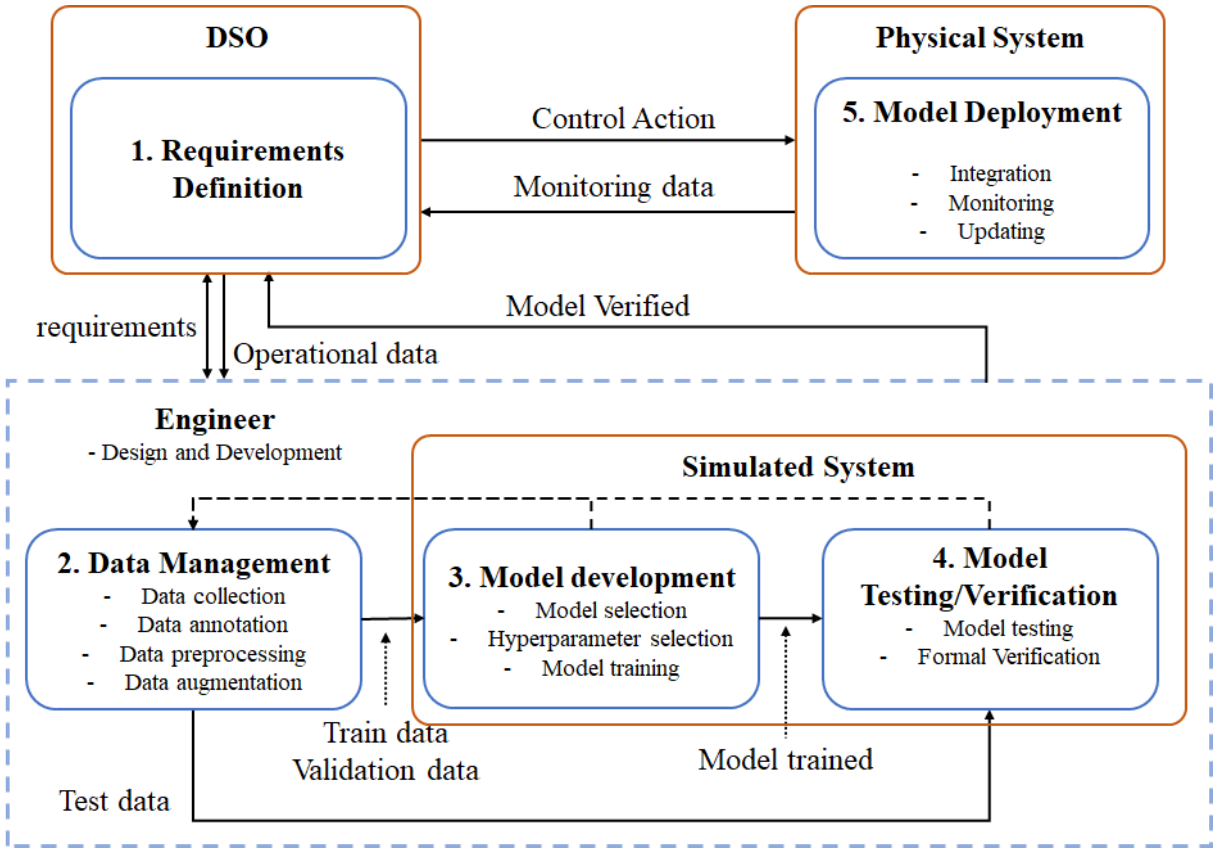

Figure 3.4 Lifecycle of a reinforcement learning algorithm for distribution system operation

The procedure from the DSO applying the RL algorithm to control the physical system is divided into five stages, and each process can be repeated several times until a complete model is achieved.

1) **Requirements Definition**: DSO receives events or requirements occurring in the power network's physical layer and defines the requirements for the RL algorithm model. Along with the necessary operational data, this information is transmitted to the engineer responsible for developing the RL model.
2) **Data Management**: Engineers, who receive requirements and operational data from the DSO, organize and preprocess this data to maintain its integrity and suitability for training the RL model. If the application of reinforcement learning, a form of artificial intelligence discussed in this paper, is desired, this stage includes the formulation of the MDP.
3) **Model Development:** Considering the given requirements and the nature of the data, an appropriate model is selected and specifically designed. Considering the computing budget, one or more sets of hyperparameters are

determined, followed by model training on a simulated system in the cyber layer.

4) **Model Testing/Verification:** The trained model is tested to ascertain its functionality in the test environment, using validation data and ensuring the satisfaction of formal requirements, such as safety constraints. The final verification is performed using test data. The verified RL model is then returned to the DSO by the engineer.

5) **Model Deployment:** The DSO utilizes the verified model to attempt control of the physical system, continuously monitoring the network's status and collecting monitoring data. If necessary, the DSO requests an update of the RL model from the engineer.

.

3.3. Reinforcement Learning for Network Reconfiguration

In this section, the DNR problem, defined in the form of an optimization problem in Section 2.1, is redefined as a MDP to solve with a RL model. Additionally, this section involves defining the

proposed RL model as well as the neural network structure for learning.

- MDP definition

$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle^{DNR}$

$\mathcal{S}$: $\{P_i^{DRES}, P_j^L, V_b, P_l^F, SW_{tr}, SW \mid \forall i \in N_{DRES}, \forall j \in N_{Load}, \forall b \in B, \forall l \in L\}$

$\mathcal{A}$: $\{sw_s | sw_s \in \{0,1\}, \forall s \in N_{SW}\}$

$\mathcal{R}$: $\mathcal{R}^{DNR}(\cdot)$

γ : γ^{DNR}

The state space $\mathcal{S}$ of the DNR problem should include information that the DSO requires for optimal decision-making. Herein, it incorporates the generation of DRES, including PV and WT, which influence system stability in the next timestep, along with load demand. Further, it comprises current system status indicators, such as bus voltages V_b and line loadings P_l^F, and factors particularly influencing the DNR problem, like the switching log SW_{tr} and switch status SW containing current network topology information. The action space $\mathcal{A}$ comprises decisions for each timestep in the DNR problem, specifically the on/off states of each

switch sw_s. The reward function $\mathcal{R}^{DNR}$ is explained subsequently, and the discount factor is defined as a real value $\gamma^{DNR} \in [0,1]$.

At this juncture, it is worth noting that formalizing the transition probability $\mathcal{P}$ in this problem is not required. In an MDP, the transition probability for each timestep is defined as $p_t = \mathcal{P}(s_{t+1}|s_t, a_t)$. However, the state s of the DNR problem depends on DRES generation or load demand values, which are challenging to accurately predict as they are based on weather conditions or human behavior and vary almost independently of the action a in this problem, i.e., the switch status change. In a sense, this could be due to a lack of necessary information to fully comprehend the transition model (for instance, weather prediction data), which could make the MDP seem partially observable. However, an advantage of the RL approach over the Dynamic Programming (DP) approach is precisely that it does not require an accurate transition probability function. Moreover, it can derive an approximated value function and/or optimal policy from sampled data, even without using this function. This same principle is applicable in subsequent Section 3.4 as well.

– Neural network structure

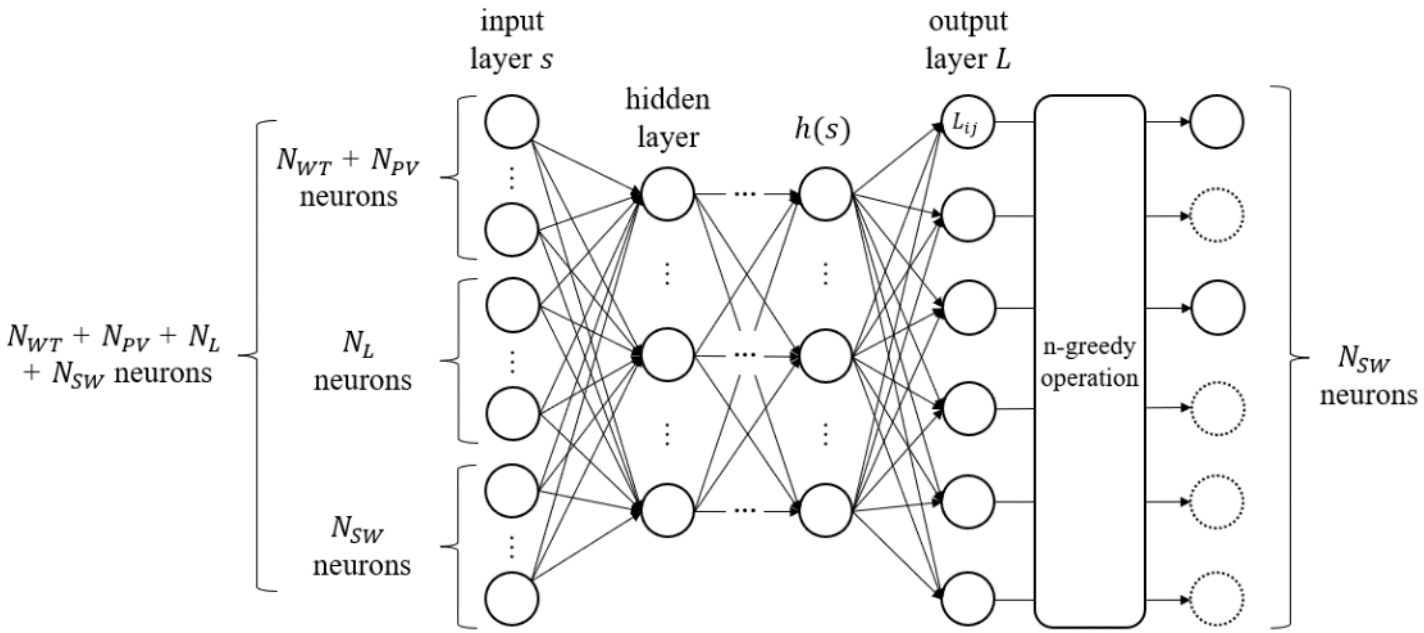

Figure 3.5 Neural network structure for DNR

– Search space configuration with given network topology

In this study, every sectionalizing switch on the network can have two statuses: on or off. If a distribution power network has N_{SW} switches, the number of actions the DSO can take in each state will be $2^{N_{SW}}$. This number increases exponentially as the size of the network or more precisely, the number of available switches increases, making it difficult to find valid switch statuses when the search space becomes very large. However, since the distribution network is assumed to radially operate, search space reduction can be performed using the graph theory. The radiality of the power system means that the network has a tree structure.

According to the graph theory, it is known that for the graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ which has root components $\mathcal{R}$, must satisfy $|\mathcal{N}| = |\mathcal{E}| + |\mathcal{R}|$ to be a forest graph. Therefore, when number of feeders of the system is F, number of buses is B, and number of lines is L then number of switches to be opened to satisfy the radiality of the system is determined as $N_{SW,o} = L - B + F$. That is, search space size is reduced from $2^{N_{SW}}$ to ${}_{N_{SW}}C_{N_{SW,o}}$. Figure 3.6 compares the original and smaller search spaces, where size was reduced according to N_{SW}, in case of $N_{SW,o} = 3$.

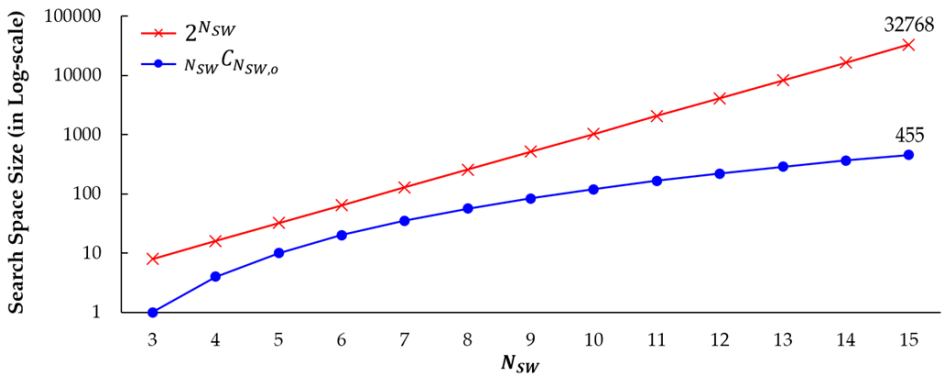

Figure 3.6 Search space size comparison

– Neural network structure configuration

The representation of actions in a MDP by a neural network, serving as a function approximator, differs depending on the reinforcement learning algorithm employed. For instance, in the widely recognized value-based algorithm, Deep Q-Network (DQN) [24], [27], the neural network is utilized as a Q-function approximator, training each logit in the network's output layer to correspond to each action's Q-value, i.e., $Q(s,a)$. This architectural design has been validated across various test cases.

However, associating each node in the output layer to each action, while intuitive, presents issues concerning scalability. Specifically, as the action space enlarges, the number of parameters that must be trained increases, thereby impacting the

convergence of the reinforcement learning model. In the context of the DNR problem discussed in this section, should each output node represent the entire network's switch status, the size of the output layer would escalate exponentially.

To counter this issue, this dissertation proposes a specialized RL model for DNR problem, based on the conventional DQN model. This model defines each output layer node of the neural network as the action of an individual switch rather than the action of the entire network. Consequently, the size of the output layer corresponds with N_{SW}, and each output node value, o_{SW}, can be defined as a Q−value relating solely to the respective switch's action. In situations like the one presented in this research, where the DSO possesses comprehensive knowledge of the network's graph structure, the number of switches to be opened, $N_{SW,o}$, can be predetermined. Switches are then opened to meet the required number, n, by sorting the output layer node values corresponding to each switch in descending order. This method extends the standard DQN's greedy policy, which selects the action with the highest Q−value, to select n actions, which can be termed as $n-greedy$.

Ultimately, the action returned to the network, serving as the

reinforcement learning model's environment, is manifested as a binary array of size N_{SW}, aligning with the previous definition in the MDP.

- Reward algorithm

Reward is a criterion for evaluating the action in given states and is one of the factors causing Q-values of DQN to be converged. Therefore, providing an appropriate reward algorithm is very important for DQN convergence and learning speed. In some environments of reinforcement learning such as the frozen lake problem, the reward is given only when the agent reaches a certain state as a goal. However, in the proposed environment, like in the cart-pole problem, the environment should be maintained within a certain range; a reward should be paid at each timestep if this condition is satisfied. Specifically, as defined in Section 2.1, the primary purpose of the agent in the proposed problem is to maintain the network radiality. Furthermore, the reconfigured network must meet general power network constraints such as maintaining line loading or bus voltage within a certain range for given power generations and loads. From this principle, we developed the reward algorithm for the proposed

DQL model. Given state s_t and action a_t, the total reward r_t^{DNR} in each timestep t can be represented as follows:

$$r_t^{DNR} = \begin{cases} \left[r_{init}^{DNR} + \left(\sum_{l}^{L} p_l^{line} + \sum_{b}^{B} p_b^{bus} + \sum_{s}^{SW} p_s^{sw} \right) \right]^+, & if\ network\ is\ radial \\ p^{fail}, & otherwise \end{cases} \quad (3.15)$$

$$p_l^{line} = -\frac{w_l}{|L|} \left(\left[\frac{(P_l^F)^2 + (Q_l^F)^2}{(\bar{S}_l)^2} - 1 \right]^+ \right)^{1/2}, \qquad \forall l \in L \quad (3.16)$$

$$p_b^{bus} = -\frac{w_v}{|B|} ([V^{LL} - V_b]^+ + [V_b - V^{UL}]^+), \qquad \forall b \in B \quad (3.17)$$

$$p_s^{sw} = -\frac{w_{sw}}{|N_{SW}|} \left[1 - \frac{sw_{tr,s}}{T_{sw}} \right]^+, \qquad \forall s \in N_{SW} \quad (3.18)$$

where

I_i= loading of line i (in percent)

V_j = nodal voltage of bus j (in p.u.)

w_l = line loading penalty weight

w_v= bus voltage penalty weight

w_{sw} = switch degradation penalty weight

If the reconfigured network is radial, the agent is given a certain reward r_{init}^{DNR} for each timestep; otherwise, it is given a negative reward (i.e., penalty) p^{fail} and the episode of the simulation will be

terminated immediately. Even if the reconfigured network is radial, when the line capacity and/or bus voltage constraint of the network is violated, the agent will be penalized p_l^{line} and p_b^{bus} according to the degree of violation and corresponding weight factors of each violation. p_l^{line} is calculated by assuming the capacity of the distribution network line is 100 in percentage and multiplying weight factor w_l by the violation of each line l. Similarly, p_b^{bus} is calculated assuming the stable bus voltage is in the range $V^{LL} \leq V_b \leq V^{UL}$ and multiplying weight factor w_v by the violation of each bus b. Furthermore, penalty term p_s^{sw} impedes frequent operation of sectionalizing switches. We use switching log data $SW_{tr} = \{sw_{tr,s}^t | \forall s \in N_{SW}\}$ to recognize how much time has elapsed since previous operation of each switch sw. If $sw_{tr,s} < T_{sw}$, p_s^{sw} is calculated by multiplying weighting factor w_{sw} by the difference between $sw_{tr,s}$ and T_{sw}. The penalty weight factors can be adjusted in a practically by whoever uses this model, depending on the distribution network environment to be applied.

3.4. Reinforcement Learning for multi-ESS Operation

In this section, the multi-ESS operation problem, defined in the form of an optimization problem in Section 2.2, is redefined as a MDP to solve with a RL model. Additionally, this section involves defining the proposed RL model as well as the neural network structure for learning.

- MDP definition

$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle^{ESS}$

$\mathcal{S}$: $\left\{ P_i^{DRES}, P_j^L, V_b, P_l^F, SOC_e \middle| \forall i \in N_{DRES}, \forall j \in N_{Load}, \forall b \in B, \forall l \in L, \forall e \in N_{ESS} \right\}$

$\mathcal{A}$: $\{ P_e^{ESS} | P_e^{ESS} \in [-\bar{P}_e^{dch}, \bar{P}_e^{ch}], \forall e \in N_{ESS} \}$

$\mathcal{R}$: $\mathcal{R}^{ESS}(\cdot)$

γ : γ^{ESS}

The MDP for Multi-ESS operation problem does not significantly diverge from that delineated in the previous section for the DNR problem. However, given the characteristics of the problem at hand, factors associated with switches have been

removed from the state space $\mathcal{S}$, while the SOC value of each ESS SOC_e have been added. The action space $\mathcal{A}$ is constituted by the decisions on the charging/discharging power P_e^{ESS} of each ESS, depending on its PCS capacity $\bar{P}_e^{dch}$ and $\bar{P}_e^{ch}$. The reward function $\mathcal{R}^{ESS}$ is explained subsequently, and the discount factor is defined as a real value $\gamma^{ESS} \in [0,1]$.

– Neural network structure

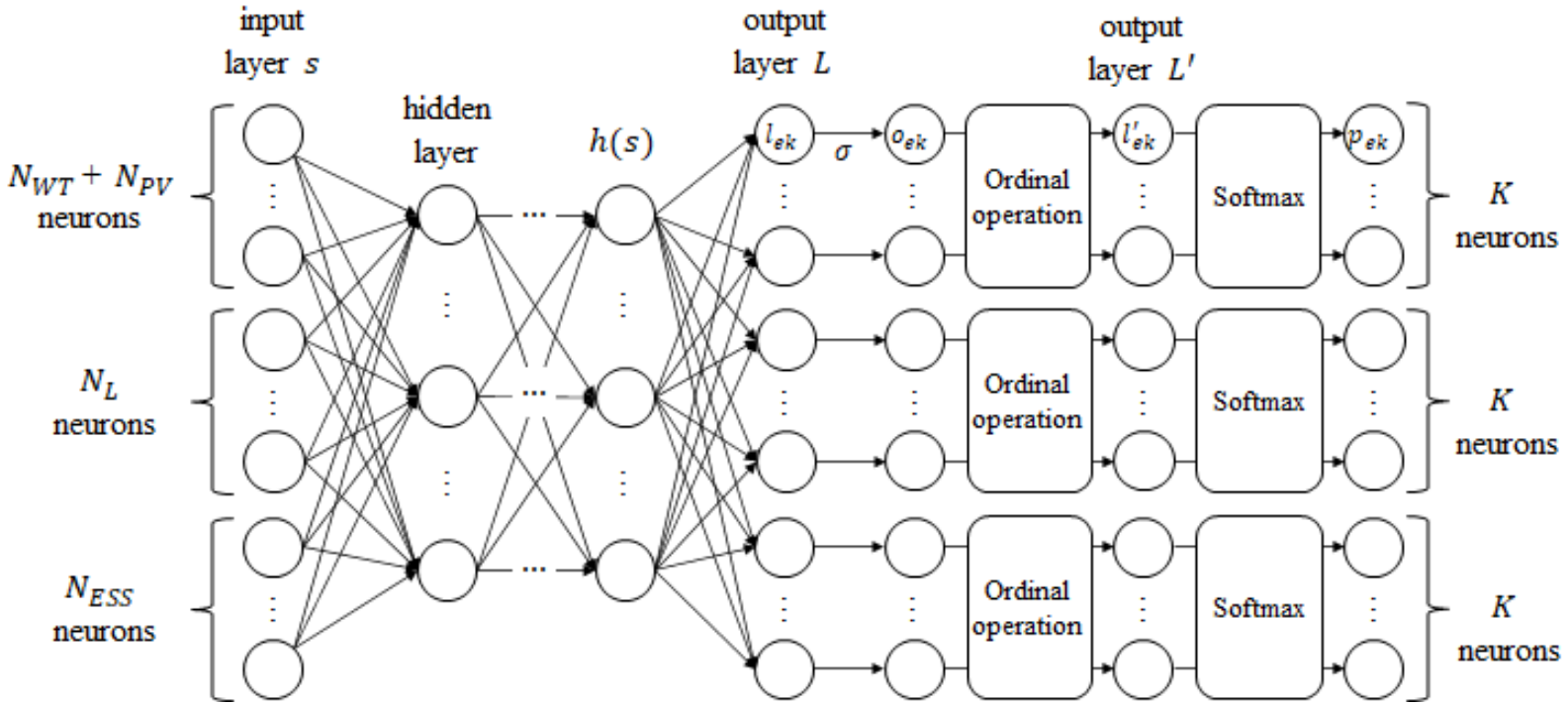

Figure 3.7 Neural network structure for ESS operation

– Neural network structure configuration

When building a neural network to determine the charging/discharging power of multiple BESSs, issues of scalability, as mentioned in the previous section, may arise. This may not be a problem if the number of ESSs in the system is small, but in this study, a device-decoupled structure [28] was adopted to avoid scalability issues in more general environments. In this device-decoupled neural network structure, the number of parameters in our proposed method increases linearly with the number of ESSs, contrasting with the vanilla Q-learning-based approach, where the number of parameters grows exponentially.

Additionally, we discretize the ESS control action into K discrete actions. As a result, the original continuous action is limited to discrete quantized actions. However, when there are enough quantized actions (e.g., $K \geq 11$), [29] presented these discretized actions can represent significantly more flexible distributions than Gaussian in practice. Intuitively, a discrete policy can represent multi-modal action distribution, while Gaussian is, by design, unimodal.

Without loss of generality, we can set the action a_e^{ESS} as a normalized form of $P_e^{ESS} \in [-\bar{P}_e^{dch}, \bar{P}_e^{ch}]$, that is, $a_e^{ESS} \in [-1, 1]$. Each

dimension of the action space is discretized into K equally spaced quantized actions. The set of quantized actions for any BESS e is $a_e^{ESS} = \left\{\frac{2k}{K-1} - 1\right\}_{k=0}^{K-1}$. Moreover, we adopt an ordinal representation for all the discrete actions of each BESS to encode the natural ordering between the discrete actions. Specifically, each subset of the output layer nodes corresponding to BESS e is first pre-processed as follows:

$$o_{ek} = sigmoid(l_{ek}) \tag{3.19}$$

$$l'_{ek} = \sum_{m \le k} \ln o_{em} + \sum_{m > k} \ln(1 - o_{em}), \qquad \forall e \in N_{ESS} \tag{3.20}$$

where l'_{ek} is the transformed output value after the ordinal encoding. Then the probability of ESS e taking action k (out of K) can be calculated as $p'_{ek} = \exp(l'_{ek}) / \sum_k^K \exp(l'_{ek})$.

- Reward algorithm

The reward function $r_t^{ESS} = \mathcal{R}^{ESS}(s_t, a_t, s_{t+1})$ of the multi-ESS operation problem is defined as follows.

$$r_t^{ESS} = r_{init}^{ESS} + \left(\sum_l^L p_l^{line} + \sum_b^B p_b^{bus} + \sum_e^{N_{ESS}} p_e^{ESS} \right) \tag{3.21}$$

$$p_l^{line} = -\frac{w_l}{|L|}\left(\left[\frac{(P_l^F)^2 + (Q_l^F)^2}{(\bar{S}_l)^2} - 1\right]^+\right)^{1/2}, \qquad \forall l \in L \tag{3.22}$$

$$p_b^{bus} = -\frac{w_v}{|B|}([V^{LL} - V_b]^+ + [V_b - V^{UL}]^+), \qquad \forall b \in B \tag{3.23}$$

$$p_e^{ESS} = -\frac{w_e}{|N_{ESS}|}([SOC_e^{min} - SOC_e]^+ + [SOC_e - SOC_e^{max}]^+), \quad \forall e \in N_{ESS} \tag{3.24}$$

where:

I_i= loading of line i (in percent)

V_j = nodal voltage of bus j (in p.u.)

w_l = line loading penalty weight

w_v= bus voltage penalty weight

w_e = BESS SOC violation penalty weight

The reward function in this case does not significantly deviate from the r_t^{DNR} defined in the previous section. However, there is no equality constraint regarding the maintenance of network radiality in this case. Moreover, a penalty p_e^{ESS} associated with a constraint related to the BESS SOC Eq. (2.24) is imposed, resulting in the assignment of a corresponding penalty coefficient w_e.

Chapter 4. Application of Reinforcement Learning in Practical Systems

4.1. Challenges of Reinforcement Learning for Physical Systems

While RL has demonstrated significant effectiveness across numerous artificial domains, it is only recently beginning to manifest success within practical scenarios. Despite the strides in RL research, particularly within the realm of power systems, the application of these advancements to physical systems often encounters hurdles due to the prevalence of certain assumptions that seldom hold in practical settings.

Certain studies have tried to comprehensively understand these inherent challenges. For instance, [30] argue that the primary obstacle lies in the disconnect between the casting of contemporary experimental RL setups and the generally undefined complexities of physical systems. They further propose that these difficulties can be encapsulated by a set of

challenges that currently impede the widespread application of RL in practical scenarios. Broadly, these challenges include:

1. Being able to learn on live systems from limited samples.

2. Dealing with unknown and potentially large delays in the system actuators, sensors, or rewards.

3. Learning and acting in high-dimensional state and action spaces.

4. Reasoning about system constraints that should never or rarely be violated.

5. Interacting with systems that are partially observable, which can alternatively be viewed as systems that are non-stationary or stochastic.

6. Learning from multi-objective or poorly specified reward functions.

7. Being able to provide actions quickly, especially for systems requiring low latencies.

8. Training off-line from the fixed logs of an external behavior policy.

9. Providing system operators with explainable policies.

These difficulties (or challenges) may arise, at least to some extent, in all types of practical physical systems. Each of these points is re-described in the context of power distribution networks, the subject of this dissertation:

1. As a social and national infrastructure, the power system has conservative and restricted data dissemination from the physical system.
2. Delayed actuator and sensor data and/or task rewards can compromise the stability of the RL model or induce control actions that do not align with the system state.
3. The number of devices that a power system operator can or must control is substantial. Without adequate decomposition of the problem, the system would require a large state and action space for optimal operation.
4. As a physical system where reliability is paramount, maintaining safety by satisfying system constraints that could lead to system failure is critical in a power system.
5. While contemporary power systems are gradually improving, a significant amount of power data is still not collected or standardized in a format that can be

immediately utilized.

6. The cost and/or reliability index traditionally used in power systems may be difficult to employ directly as rewards for effective reinforcement learning. The reward function should consider its physical meaning. Still, it is challenging to design a reward that satisfies both its effect and significance, particularly for multi-objective tasks.
7. As the proportion of uncontrollable resources like DRESs within the system increases, fast actions by system operators are required. However, many parts of the power system subject to control are not yet digitized and must be manually operated.
8. The most valuable elements for training a reliable RL algorithm for a power system, such as fault data of the system, are difficult to collect online if they are not from a simulated environment, necessitating the inevitable use of past log data.
9. As power system operators have the authority and responsibility for network operation, they may be hesitant to use RL algorithms if the associated policy is

not explainable, to understand and hedge potential risks associated with the use of the RL algorithm.

Many of these difficulties, especially challenges 2, 3, 7, and 8, are difficult to generalize from a singular physical system such as a distribution power network. Therefore, when a DSO wishes to apply artificial intelligence to the system they manage, they must fully consider the characteristics of the given system and address these challenges accordingly.

Several challenges, particularly challenges 1, 2, 5, 7, and 9, may clearly a problem for individual DSOs but are difficult to resolve immediately. To solve this problem, entities with a larger agenda and decision-making power over the power system, such as TSOs or governments, need to be involved, and it needs to be overcome gradually through long-term capital investment to make the information needed by DSOs more transparently.

Addressing the aforementioned challenges requires an approach to power system operation that considers entities at a higher level or physical devices at a lower level (e.g., generators, sensors, metering devices). However, this goes beyond the scope of this dissertation.

In this dissertation, Challenge 1 is addressed in Chapter 1 by defining CPS which based on the power system modeling and assuming their reliability. Chapter 2 indirectly tackles Challenges 3 and 6 by adequately defining DSO's control schemes. This chapter focuses on the challenges, especially Challenge 4 which about satisfying environmental constraints, that can be handled from the individual DSO's perspective. It aims to clarify these and explore ways to improve the RL algorithm discussed in previous chapters.

4.2. Concept of Safe Reinforcement Learning

– General Concept of Safe RL

The challenge of handling constraints in modeling the optimal operation of the distribution network remains a significant one. [31] asserts that in most model-free methods, constraints are modeled as negative rewards within the framework of MDP, using penalty methods. However, the determination of an optimal penalty coefficient to strike a balance between constraint violation and the reward is a challenging task. Furthermore, even when using a large penalty coefficient, penalty methods typically

cannot ensure strict adherence to constraints.

This poses a risk not only within the domain of distribution power systems, but also when applying RL algorithms to safety-critical fields like robotic control that are intimately tied to physical systems. Before Safe RL was established as a distinct field of research, RL researchers have been exploring mathematical methodologies to address such safety issues. This led to the emergence of the concept of 'Safe Reinforcement Learning', as named by [32], [33].

Though there isn't a full academic consensus on the precise definition of Safe RL yet, according to [32], Safe RL strives to "learn policies that maximize the expectation of return in problems where it is imperative to ensure reasonable system performance and/or adhere to safety constraints during the learning and/or deployment process."

- Safety level definition

To encode the safety constraints in RL framework, we define a constraint cost function $\mathcal{C}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$. Starting with the weakest guarantee, we introduce three levels of safety as [34]: soft constraints, probabilistic constraints, and hard constraints,

as depicted in Figure 4.1. We note that these safety levels are often mixed in practice.

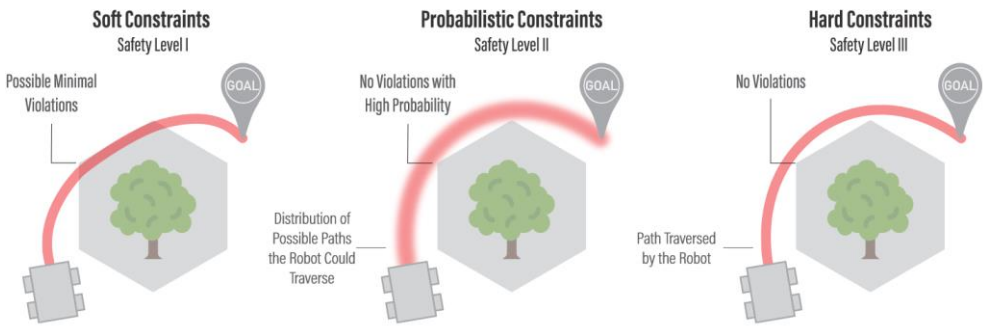

Figure 4.1 Illustration of the different safety levels [34]

Safety Level I: Soft Constraints

The system encourages adherence to constraints; however, no guarantee is provided. This is typically achieved by incorporating a penalty term into the objective (or value) function, which discourages contravention of constraints by imposing a significant cost. Formally, this can be represented as:

$$\mathcal{C}(s_t, a_t, s_{t+1}) \leq \epsilon_c \tag{4.1}$$

where ϵ_c is a non−negative cost threshold according to the constraint c. Alternatively, although $\mathcal{C}(s_t, a_t, s_{t+1})$ is a step−wise

quantity, some approaches only aim to provide guarantees on its expected value $\mathbb{E}[\cdot]$ on a trajectory level:

$$J_c = \mathbb{E}\left[\sum_{t=0}^{T} \mathcal{C}(s_t, a_t, s_{t+1})\right] \leq d_c \tag{4.2}$$

where J_c represents the expected total constraint cost, and d_c represents the threshold for cumulative constraint cost.

Safety Level II: Probabilistic Constraints (Chance Constraints)

The system adheres to probabilistic constraints, where the maximum probability dictating the operator's compliance with the constraints is established. This can be mathematically expressed as:

$$\Pr(\mathcal{C}(s_t, a_t, s_{t+1}) \leq 0) \geq p^c \tag{4.3}$$

where $\Pr(\cdot)$ denotes the probability, and $p^c \in (0,1)$ signifies the likelihood of satisfying the constraint. When p^c equals 1, the chance constraint in the above equation aligns with the hard constraint in Safety Level III.

Safety Level III: Hard Constraints

The system complies with hard constraints, which the operator must consistently respect. This can be formulated as:

$$\mathcal{C}(s_t, a_t, s_{t+1}) \leq 0 \tag{4.4}$$

– Proposed Safe RL framework

Before discussing the level and specific formulation of constraints associated with proposed ANM methods in the distribution power system, we would like to mention how constraints can be handled under the RL architecture discussed in Section 3.2. To deal with constraints in RL architecture, Figure 3.3 must be modified as follows.

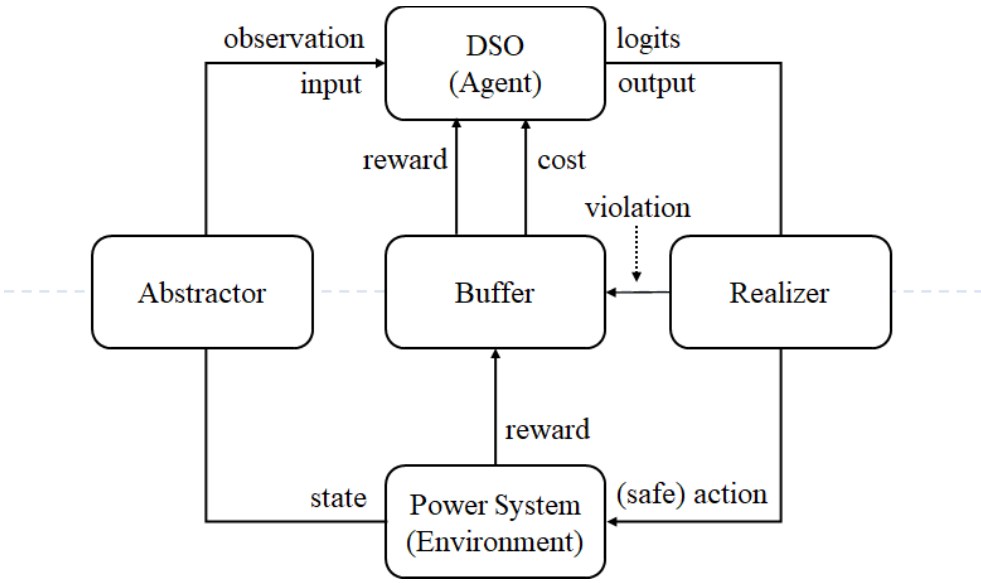

Figure 4.2 RL architecture with constraints

In Figure 4.2, the addition of the Buffer component is evident. Within our Safe RL framework, the Realizer examines the safety output of the RL Agent, verifying the presence and extent of any constraint violations, with the results then transferred to the Buffer component. Furthermore, the Realizer can, if needed, not merely confirm violations but also utilize this information to modify the Agent action into a safe action before passing it on to the environment.

The Buffer is situated between the Environment and the Agent. It stores experiences in the form of tuples, thereby enabling the Agent to learn from cumulative offline experiences.

In the event of receiving violation information from the Realizer, the Buffer transforms this into a cost format and stores it. From the Agent's perspective, this system changes the evaluation of its intended actions from a single value (reward) to two separate values (action and cost). This bifurcation can be utilized to meet safety requirements.

The determination of the safety level that each constraint must satisfy can vary based on system characteristics. In this study, equality constraints are treated as hard constraints under safety level III, while inequality constraints are treated as soft constraints under safety level I. However, this distinction doesn't imply that inequality constraints are physically less important than equality constraints. This separation is proposed with the intention to facilitate more efficient learning during the RL process by distinguishing between fundamentally different constraints, the satisfaction of which are defined in terms of binary (discrete) and continuous terms respectively.

4.3. Formulation of Safe Reinforcement Learning

4.3.1. Safety Module for Equality Constraint

Among the constraints covered in this dissertation, the equality constraint that must be respected is the network radiality constraint in the DNR problem. To this end, this section proposes a safety module as in Figure 4.3. This concept, referred to as a *shield* in some studies, such as [35], is designed to internally validate and, if necessary, appropriately adjust the initial action derived from the agent's policy. This ensures a stringent satisfaction of the network radiality constraint. Prior research on DNR, such as [36], has addressed this issue in studies on network reconfiguration, aimed at determining new topology via heuristic methods.

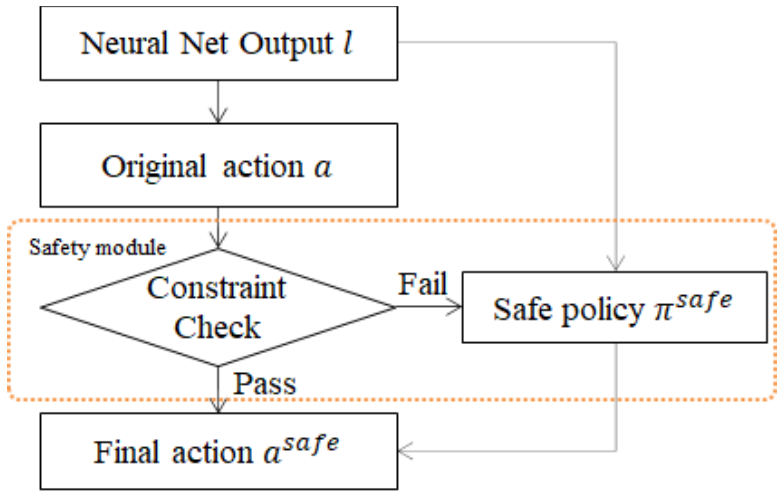

Figure 4.3 Flowchart of safety module

The output layer values of the neural network are used to determine the priority of each switch to open. This means, the violation of the network radiality constraint can be interpreted as an inconsistency between the device-level requirement of switch priority and the network/system-level requirement of network radiality, due to any given reason. Hence, a safety module, which strictly satisfies the network radiality constraint, must be designed to leverage the existing output layer values. Prior to the application of the resulting topology in the distribution network, the radiality of this topology should be verified. If necessary, through minimal modifications, it should be

transformed into a safe action, thus ensuring radiality.

Drawing inspiration from [36], this paper presents a method to derive a safe action from the input output values, in the following sequence. This is also referred to as a safe policy, denoted as π^{safe}.

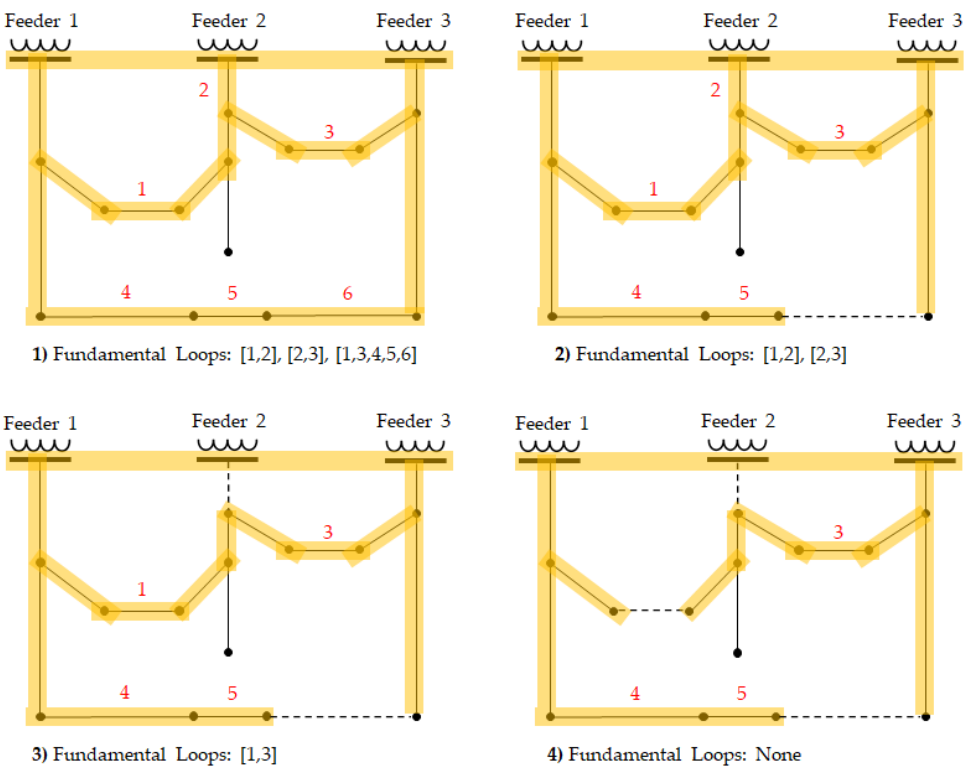

Figure 4.4 An example of applying safe policy for DNR

1) Identify the fundamental loops in the network (the minimal loops that occur in the network when all switches are closed)

and compile a list of switches included in each loop.

2) Start opening switches from the one with the highest value, and remove from the list, the fundamental loop that includes this switch.

3) If the selected switch in step 2) is shared by two fundamental loops, open this switch, remove the line that includes this switch, and merge the two fundamental loops into a new one.

4) Repeat the above steps until all fundamental loops have been eliminated.

Through this methodology, we can determine a switching action that always satisfies the network radiality, while still relying on the priority value of each switch.

- SafeFallback method

Additionally, we aim to maximize sample efficiency and learning effectiveness by employing the SafeFallback method [37] during the learning of the aforementioned equality constraint and the results obtained from the safety module. The core principles of the SafeFallback method are as follows: 1) If the initially

obtained action is safe, the experience with this action is stored in the buffer. 2) If the initially obtained action is not safe and other safe action is subsequently obtained from the safety module, both the experiences with the initial action and the safe action are stored. In this case, a radiality violation penalty is assigned to the original unsafe action, as in Section 3.3, instead of the reward. The pseudocode is provided in Algorithm 1.

Algorithm 1: SafeFallback for Network Radiality Constraint

Require:
distribution network as a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$

1: Input: initialize policy π, initialize radial topology set $\mathcal{E}^{\mathrm{rad}}$, initialize safe fallback policy π^{safe} , initialize replay buffer D

2: for each sample step **do**
3: Observe state s and select action a according to π
4: find network topology $\boldsymbol{\alpha}$ with selected action a
5: **if** $\boldsymbol{\alpha} \in \mathcal{E}^{\mathrm{rad}}$ **then**
6: keep selected action a as safe action a^{safe}
7: **else**
8: get safe action a^{safe} from the safe fallback policy π^{safe}
9: **end**
10: Execute $\boldsymbol{\alpha}^{safe}$ according to a^{safe} in the environment
11: Observe next state s', reward r and done signal d
12: $D \leftarrow D \cup \left(s, a^{safe}, r, s', d\right)$
13: **if** $a^{safe} \neq a$ **then**
14: $D \leftarrow D \cup \left(s, a, p^{rad}, s', d\right)$ with radiality violation penalty p^{rad}
15: If s' is terminal, reset environment state
16: **End for**

4.3.2. Adaptive Penalty for Inequality Constraints

The safety module proposed earlier has the advantage of satisfying the system's equality constraint at safety level III as a hard constraint. However, this perspective is primarily from the utilization, application, and deployment of RL. Conversely, an agent in the learning process can become decoupled from the originally defined RL formulation, the MDP problem, due to the safety module. Consequently, it may face an unstationary problem that it considers the environment is not fixed while interacting with or may possibly take reckless actions relying solely on the safety module. In such a case, the reinforcement learning model can experience a phenomenon where its robustness declines with respect to the environment's stochasticity, which could not be observed in the data used for learning.

To address these problems, some Safe RL research has utilized a method that simultaneously designs a manual safety module and imposes a corresponding (constant) penalty when this module is activated [38]. Through this approach, we can provide the agent with a signal about the constraint. However,

this approach still faces two problems: 1) it can provide information about a specific action's constraint violation, but lacks information about which actions should be taken to avoid violating the constraint, and 2) it is difficult to determine an appropriate penalty coefficient to balance the relative importance of obtaining a reward and satisfying a constraint.

In this study, inspired by [28], we aim to redefine the given problem as a Constrained Markov Decision Process (CMDP) problem with added constraints to the original MDP. We plan to establish a Lagrange function for this and treat the Lagrange multiplier λ as a penalty coefficient, which will be iteratively updated alongside the neural network parameters.

- Constrained Markov Decision Process

A CMDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{C} \rangle$. This represents an MDP as defined in Section 3.1, with the addition of a constraint function C, which can be defined as $\mathcal{C}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R} \cup \{+\infty\}$, where the $+\infty$ value could be used to assign a constraint that must not be violated, if needed. Given that the cost function $\mathcal{C}$ shares the same structure as the reward function $\mathcal{R}$, a cost value function V^c can be equivalently defined with respect to the cost,

just as the value function V is defined from the reward.

$$G_t^c = c_{t+1} + \gamma c_{t+2} + \cdots + \gamma^{T-t-1} c_T = \sum_{i=0}^{T-t-1} \gamma^i c_{t+1+i} \tag{4.5}$$

$$V^c(s) = \mathbb{E}[G_t^c | s_t = s] \tag{4.6}$$

$$V_\pi^c(s) = \mathbb{E}_\pi[c_{t+1} + \gamma V_\pi^c(s_{t+1}) | s_t = s] \tag{4.7}$$

From this, the optimal policy of CMDP can be obtained by solving the following optimization problem.

$$\max_\pi V_\pi(s), \qquad s.t.\ V_\pi^c(s) \leq \overline{V^c} \tag{4.8}$$

where $\overline{V^c}$ is a threshold value of constraint cost c. we note that this value can be determined in considering of required safety level for constraints.

- Adaptive penalty for Deep Q-learning

To solve the above CMDP problem, the experience collected by the agent during the reinforcement learning process changes

from (s_t, a_t, s_{t+1}, r_t) to $(s_t, a_t, s_{t+1}, r_t, c_t)$.

Also, we will rewrite Eq.(4.8) as a sample based expectation form to solve within the RL framework.

$$\max_{\pi} \mathbb{E}_{s\sim D}[V_\pi(s)], \qquad s.t.\ \mathbb{E}_{s\sim D}[V_\pi^c(s)] \le \overline{V^c} \tag{4.9}$$

As in [28], the Lagrange function of the constrianed optimization problem can be written as:

$$\begin{aligned} \mathcal{L}(\pi, \boldsymbol{\lambda}) &= \mathbb{E}_{s\sim D}[V_\pi(s)] + \lambda(\overline{V^c} - \mathbb{E}_{s\sim D}[V_\pi^c(s)]) \\ &= \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t \big(\mathcal{R}(s_t, a_t, s_{t+1}) - \lambda \mathcal{C}(s_t, a_t, s_{t+1})\big)\right] + \lambda\overline{V^c} \end{aligned} \tag{4.10}$$

A separate network other than Q needs to be introduced to estimate the V^c in the above equation. It is not necessary for this network to follow the same dueling Q-network structure that we introduced in Chapter 3. However, for reasons of programming convenience, this study has introduced a dueling Q-network of the same structure $Q^c(s, a; \theta^c, \phi^c, \psi^c)$ for the constraint cost c, which can be estimated and updated in the same way as Eq. (3.11)-(3.12).

The method of multipliers can be used to solve the constrained optimization problem. At $k-$th iteration, given a multiplier $\lambda_k \geq 0$, we can maximize $\mathcal{L}(\cdot, \lambda_k)$, over policy domain thereby obtaining a policy π_k. We can iteratively update λ_k as follows:

$$\lambda_{k+1} = [\lambda_k - \delta_\lambda \nabla_\lambda \mathcal{L}]^+ = \left[\lambda_k + \delta_\lambda \left(\mathbb{E}_{s\sim D}\left[V_{\pi_k}^c(s)\right] - \overline{V^c}\right)\right]^+ \tag{4.11}$$

where δ_λ is the step size for the λ update process. Along with the SafeFallback algorithm in previous section, the pseudocode of constrained deep Q−learning is provided in Algorithm 2.

Algorithm 2: Constrained Deep Q Learning

1: Input: Initialize network parameters and Lagrange multiplier λ^j
2: repeat
3: **for** each sample step **do**
4: $D \leftarrow$ **Algorithm 1**: *SafeFallback*
5: **end for**
6: **for** each gradient step with sample batch B **do**
7: Update dueling Q-network parameters θ, ϕ, ψ
8: Update constrained dueling Q-network parameters θ^c, ϕ^c, ψ^c
9: $\lambda \leftarrow [\lambda + \delta_\lambda \sum_B \left(V_{\phi^c} - \overline{V^c}\right)/|B|]$
10: **end for**
11: **until** coverage

We summarize the discussion until now as follows. In Chapter 2, we proposed DNR and multi-ESS operation as two short-term control schemes that can be utilized by DSOs, and in Chapter 3, we redefined each scheme under the RL framework. In Chapter 4, we designed a safety module and an adaptive penalty term that can explicitly consider safety constraints in the framework. The final revised RL architecture from Figure 4.2 is illustrated in Figure 4.5. The DSO (Agent) has one or more network control options, and each option is paired with corresponding devices in the distribution network (Environment). Between the two entities, the Abstractor preprocesses the environment's sensor or measured data into the agent's observation or input, and the Realizer performs verification through the safety module on the agent's policy output, converts it into a safe action, and delivers it to the environment. The experiences generated in this process are stored in the Buffer and used to learn the agent's policy.

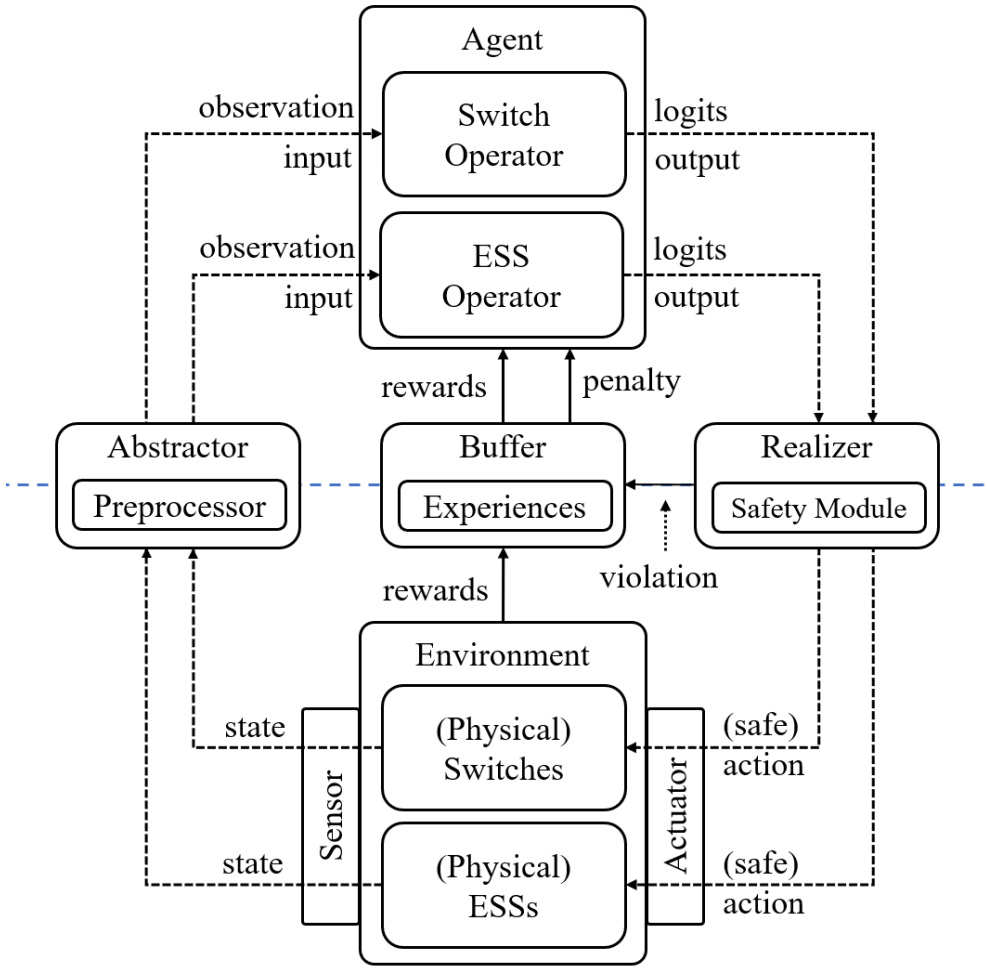

Figure 4.5 Proposed Safe RL framework

In the next chapter, we will implement the above architecture on the IEEE test feeder network and verify its effectiveness.

Chapter 5. Case Study

5.1. Simulation Settings

5.1.1. Test Network Configuration

A modified version of IEEE 123-bus test system [39] is used to verify the scalability of the proposed algorithm and shown in Figure 5.1. The original IEEE network has only one feeder and 6 sectionalizing switches, however, we added one more feeder in front of bus 27 and 11 more switches for maximizing the effectiveness of network reconfiguration. Also, 9 WTs and 7 PVs were added with generation capacities of 400kW respectively, as well as 8 ESSs with 300kW PCS capacity and 900kWh battery capacity. Base voltages were set to 4.16 kV. The number and capacity of DRESs in both test networks are determined considering whether they can demonstrate both effectiveness of network reconfiguration and self-sufficiency of power distribution system.

The locations of PVs, WTs, and switches are from [40] and the locations of ESS are from [41]. More detailed distribution system data are summarized in Appendix A.

We note that there are 10 fundamental loops in this test network, as shown in Figure 5.2. From this, we can see that this network needs to open 10 switches out of a total of 17 to be a radial network, and as mentioned in Section 3.3, the action space size if using the vanilla RL algorithm is $_{17}C_{10} = 19448$.

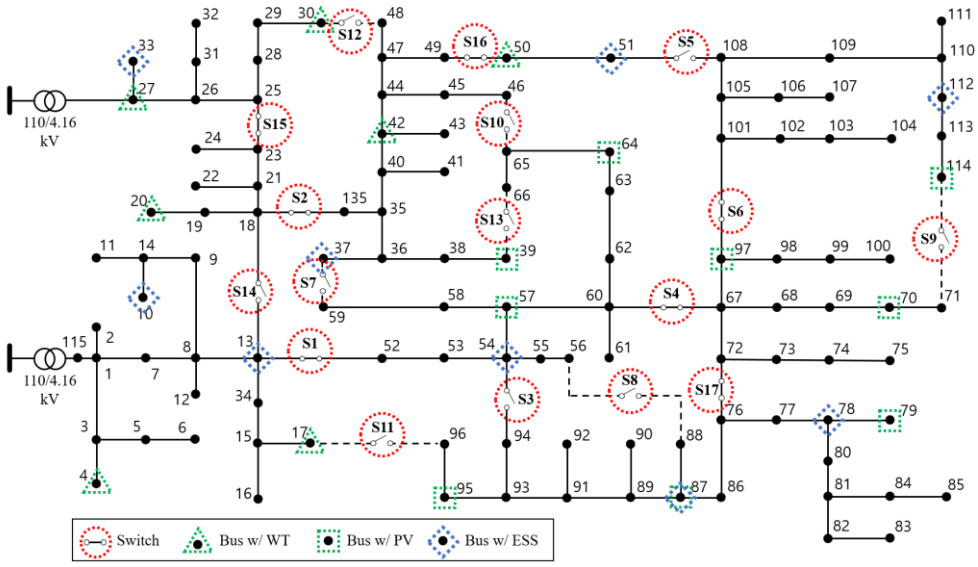

Figure 5.1 Modified IEEE 123-bus test system network

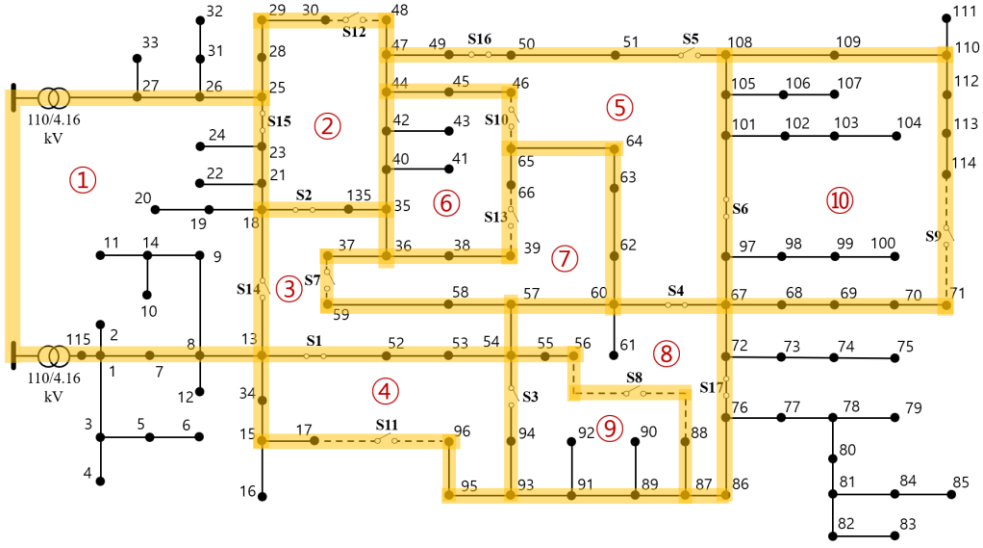

Figure 5.2 Fundamental loops of the modified IEEE 123-bus test system network

5.1.2. Input Data Configuration

Power generation data, presented in Figure 5.5, were imported from 2015 generation data recorded hourly at Hangyeong Wind Farm Part #1 by Korea Southern Power Co., Ltd. [42] and Jindo Solar Power Plant by Korea Rural Community Corporation [43]. Load data, presented in Figure 5.6, were imported from the dataset named "Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States" recorded by the Office of Energy Efficiency & Renewable Energy (EERE) [44]. All data were normalized and were added to appropriate randomized noise in the range of ±10% when applied to each DRES and/or load for to enhance model robustness. The time window length of all data is 1 hour.

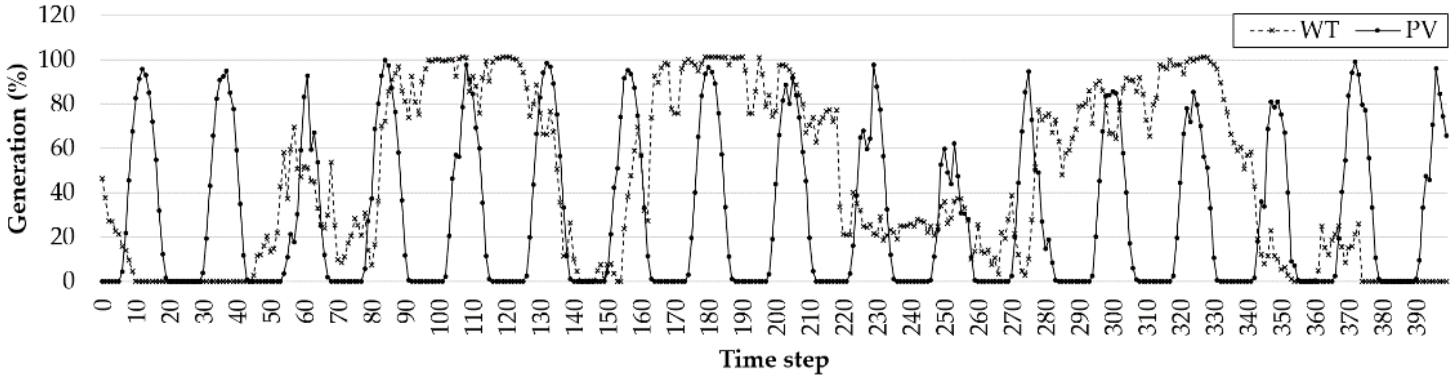

Figure 5.3 DRES power generation (%) over 400 timesteps [42], [43]

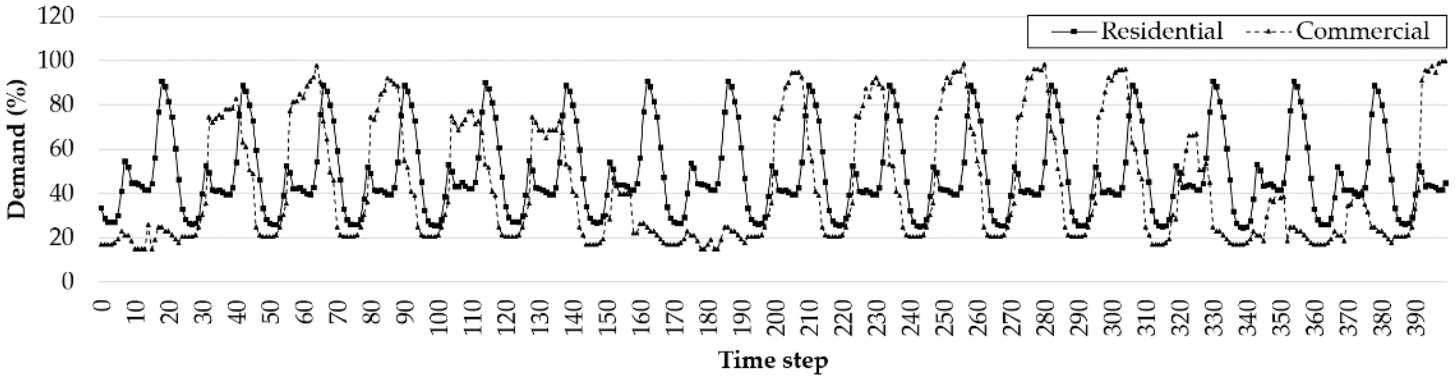

Figure 5.4 Load demand (%) over 400 timesteps [44]

5.1.3. Reinforcement Learning Configuration

In machine learning, choosing optimal hyperparameters for a learning algorithm is the most crucial aspect of training. Because such parameters cannot be optimized with internal training, they must be tuned by whoever designs them according to the proposed model and data features. Hyperparameters are composed of several variables that determine network structure and training. Hyperparameters used to implement the Deep-Q Learning algorithm designed in Chapter 3 are listed in Table 5.1.

Table 5.1 Hyperparemeters for RL

Hyper Parameter	Value	Description
$r_{\text{init}}^{DNR}, r_{\text{init}}^{ESS}$	1	Initial reward value for each timestep. Agent obtains this reward finally if there is no violation in reconfigured network.
w_l	0.8	Weight of line loading violation penalty.
w_v	0.8	Weight of bus voltage violation penalty.
w_s	0.2	Weight of frequent switch operation penalty.
w_e	0.2	Weight of SOC violation penalty.
p^{fail}	-1	Final reward for the agent if the reconfigured network is not radial.
no. of hidden layers	4	Number of hidden layers of DQN. The Q network in lower layers has 2 hidden layers, and V and A network has 2 hidden layers respectively.

Hidden layer size	32	Number of nodes of each hidden layer.
Learning rate	5E-5	Learning rate used by optimizer.
M	1E5	Total number of randomly sampled episodes required to train DQN.
T	24	Maximum length of each episode.
Batch Size (N_{Batch})	64	Number of training cases over which each update of parameters is computed.
Optimizer	Adam	Parameter optimization model. Adam is one of the most popular and strongest optimizers for training deep neural networks. Its basic idea is to combine the advantages of RMSProp and SGD with momentum.

Also, we must determine the hyperparameters for Safe RL that designed in Chapter 4.

Table 5.2 Additional hyperparameters for safety algorithm

Hyper Parameter	**Value**	**Description**
$\overline{V^c}$	1	Maximum cost value function threshold.
λ_0	1	Weight of initial Lagrange coefficient for adaptive penalty term.
Learning rate δ_λ	1E-6	Learning rate used to update λ
no. of hidden layers	4	Number of hidden layers of DQN for cost.

5.1.4. Simulation Case Design

- Case 1: Results depending on DSO's assets

  - 1-A. Base

    This case is designed to demonstrate the situation where the DSO has no controllable assets for operation.

  - 1-B. Only DNR

    This case is designed to demonstrate the situation where DSO can control the sectionalizing switches in the network to change the topology of the network.

  - 1-C. Only ESS

    This case is designed to demonstrate the situation where DSO can fully control the BESSs in the network.

  - 1-D. DNR+ESS

    This case is designed to demonstrate the situation where DSO can control the sectionalizing switches as well as BESSs in the network.

- Case 2: Results depending on applying safety algorithm

  ■ 2-A. Base

  This case is same with Case 1-A.

  ■ 2-B. DNR

  This case is same with Case 1-B.

  ■ 2-C. Safe DNR

  This case is designed to demonstrate the effect of Safe algorithms which proposed in Chapter 4, to compare with case 2-B.

  ■ 2-D. DNR+ESS

  This case is same with Case 1-D.

  ■ 2-E. Safe DNR+ESS

  This case is designed to demonstrate the effect of Safe algorithms which proposed in Chapter 4, to compare with case 2-D.

All program codes are written and compiled in the Python 3.9 environment while using PyTorch 1.13.1 to build RL algorithms. Furthermore, pandapower 2.11.1 elements and functions were used to implement the changing switching status in test distribution system and solve the power flow of the system. All simulations were conducted using a personal computer (PC) equipped with 3.59-GHz AMD Ryzen 5 3600 6-Core central processing unit (CPU), 32 GB of random-access memory (RAM), and a 64-bit Windows® 11 operating system.

5.2. Simulation Results and Analysis

This section provides the simulation results of the given 400 timestep operation, and analysis for that.

5.2.1. Results depending on DSO's assets

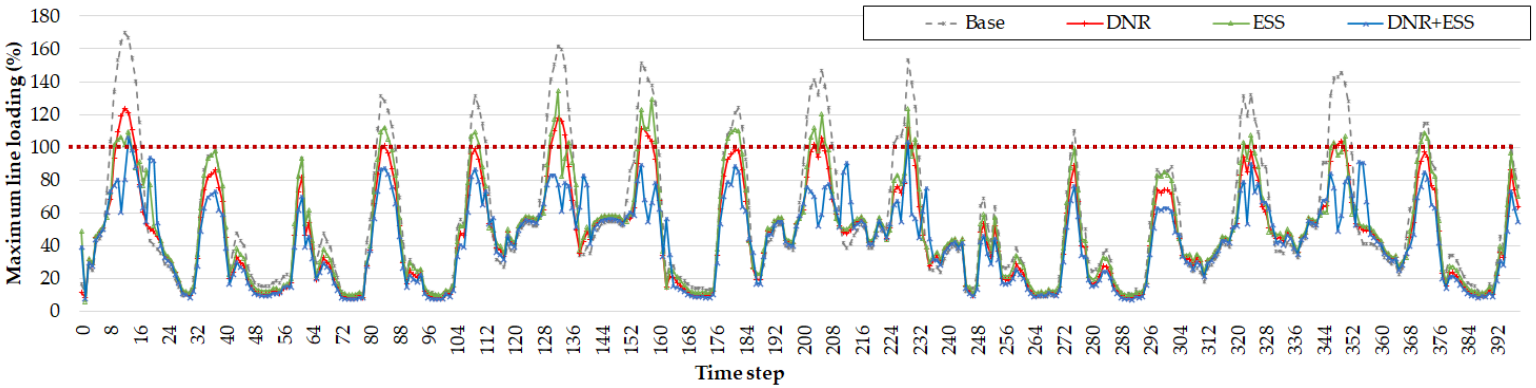

Figure 5.5 Maximum test distribution network line lodings for test dataset (Case 1)

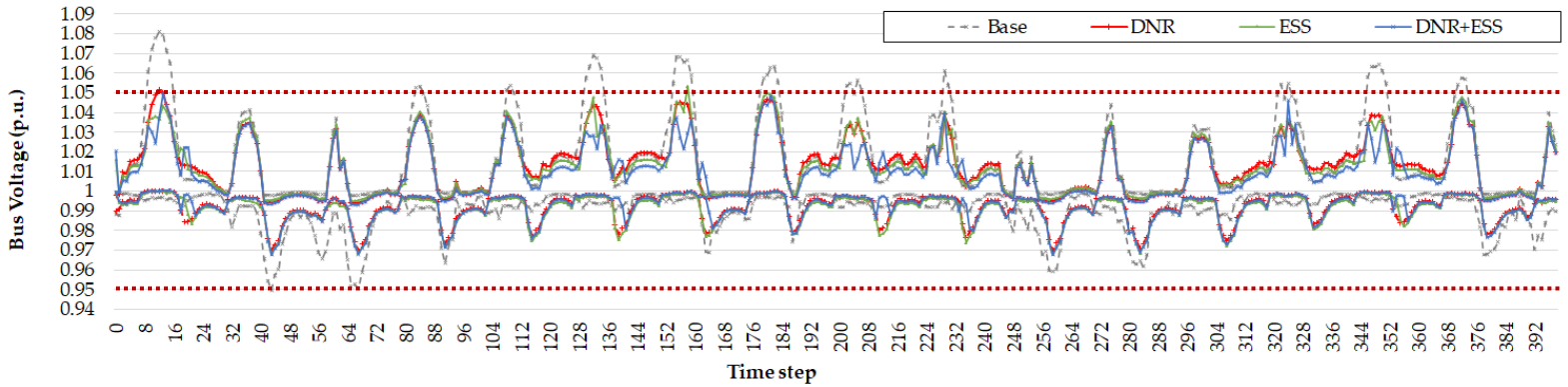

Figure 5.6 Maximum and minimum test distribution network bus voltages for test dataset (Case 1)

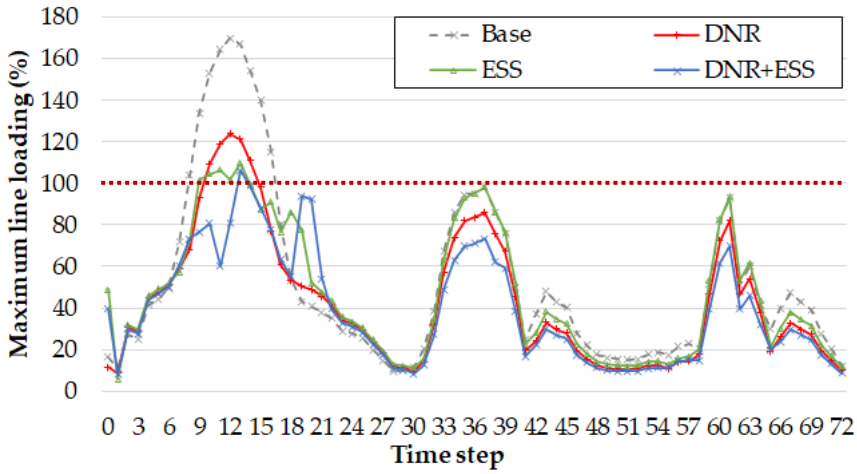

Figure 5.7 Part of Figure 5.5

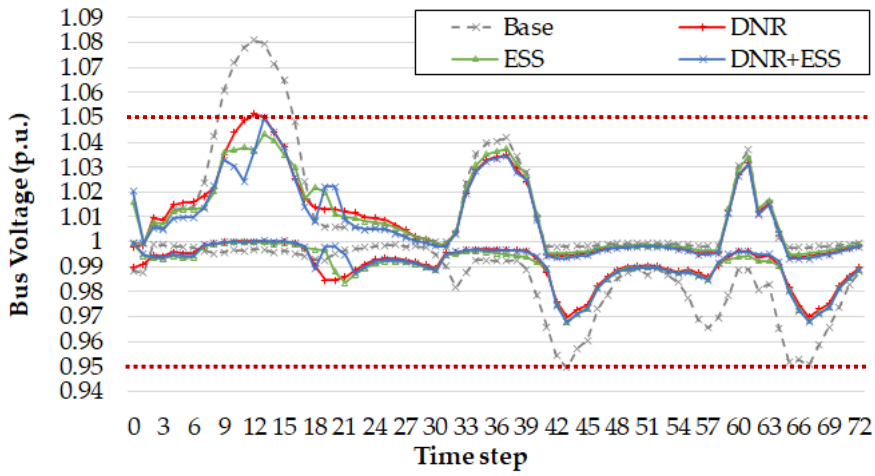

Figure 5.8 Part of Figure 5.6

The results of Simulation Case 1 is illustrated in Figures 5.5 through 5.8. As depicted in the graphs, in Case 1-A where the DSO takes no operating action, violations occur in maximum

line loading and both maximum and minimum bus voltage. If network operation attempts are made by deploying DNR or ESS, as in Case 1-B and 1-C, these violations are partly alleviated, albeit not completely. Moreover, when both DNR and ESS are utilized as in Case 1-D, it is evident that the system is maintained in a substantially more stable state compared to when each is used separately.

The aforementioned graphs only present the time-varying system states for each case; therefore, examining the overall system state at a particular snapshot for each case could prove beneficial. Utilizing a python library, named *plotly*, we were able to observe each node voltage and line loading that comprise the entire system.

Table 5.3 Network index for each subcase in Case 1 snapshot

Case	**1-A (Base)**	**1-B (DNR)**	**1-C (ESS)**	**1-D (DNR+ESS)**
Maximum Line Loading (%)	153.68	125.07	97.58	73.24
Maximum Bus Voltage (p.u.)	1.067	1.053	1.045	1.037

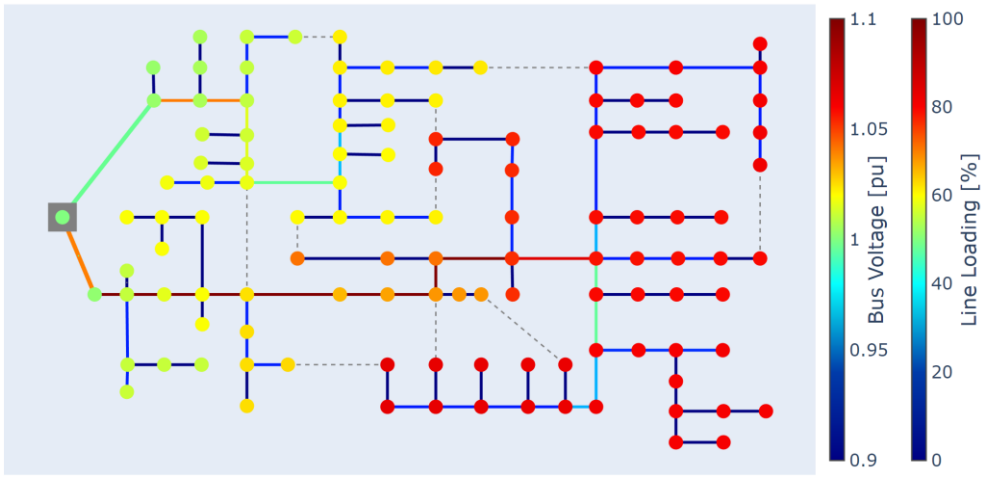

Figure 5.9 Network snapshot from Case 1−A.

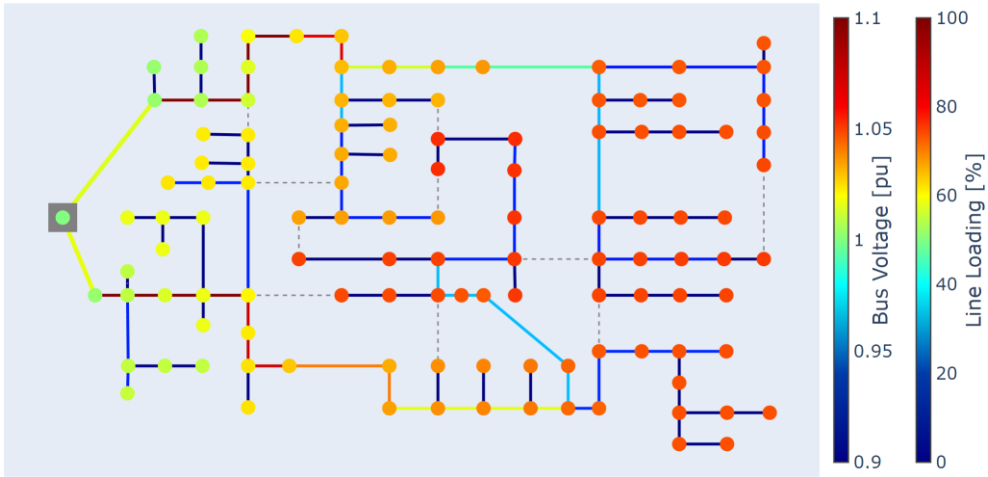

Figure 5.10 Network snapshot from Case 1−B.

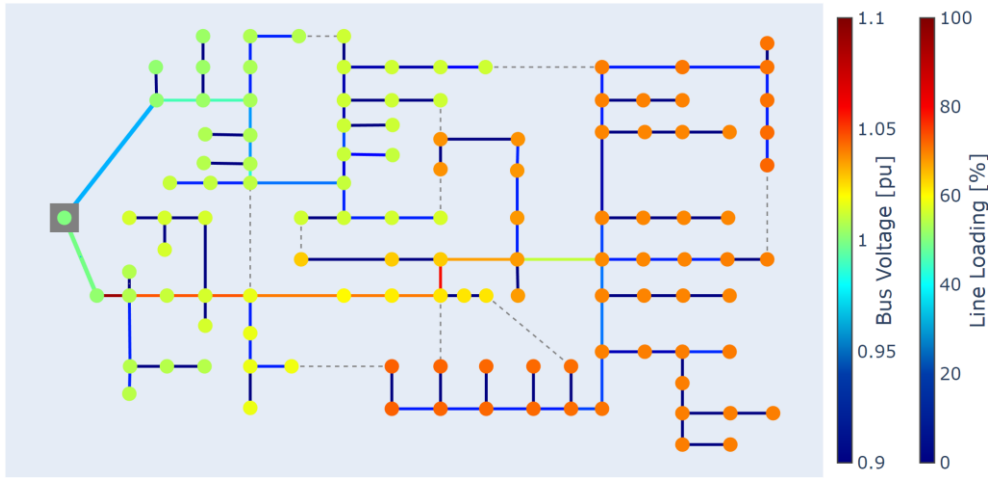

Figure 5.11 Network snapshot from Case 1−C.

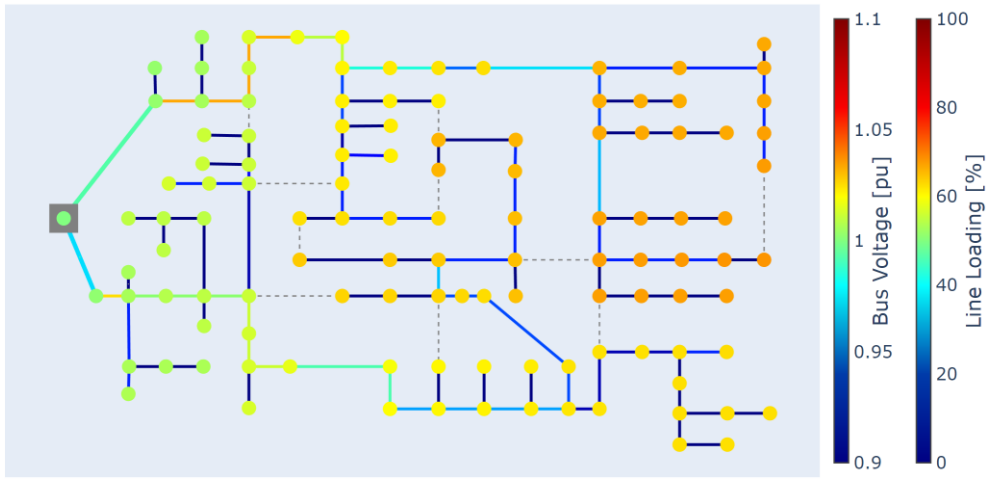

Figure 5.12 Network snapshot from Case 1−D.

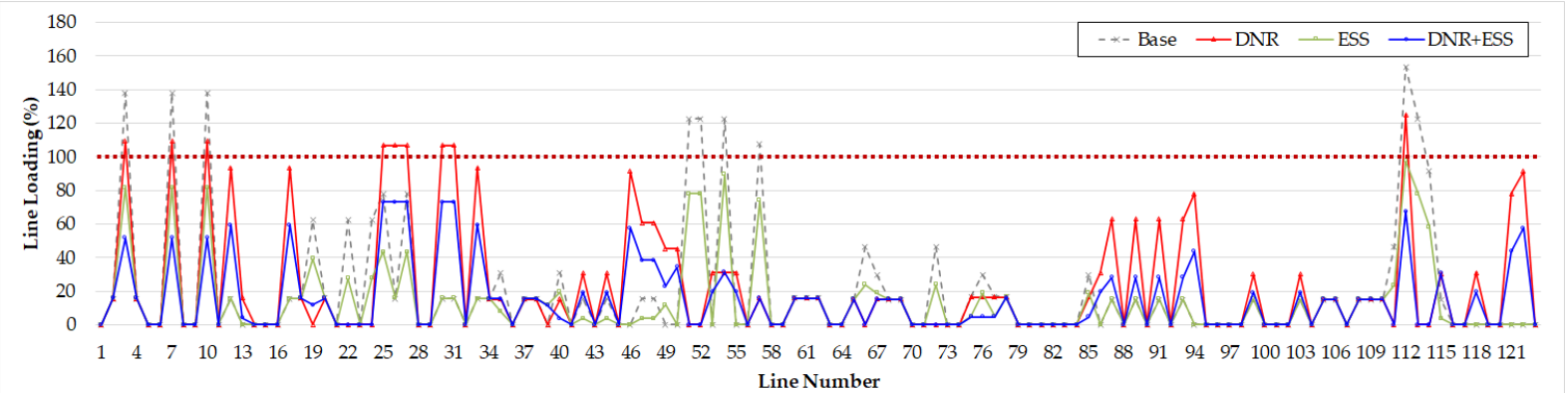

Figure 5.13 Entire network line loadings of Fig. 5.9−5.12

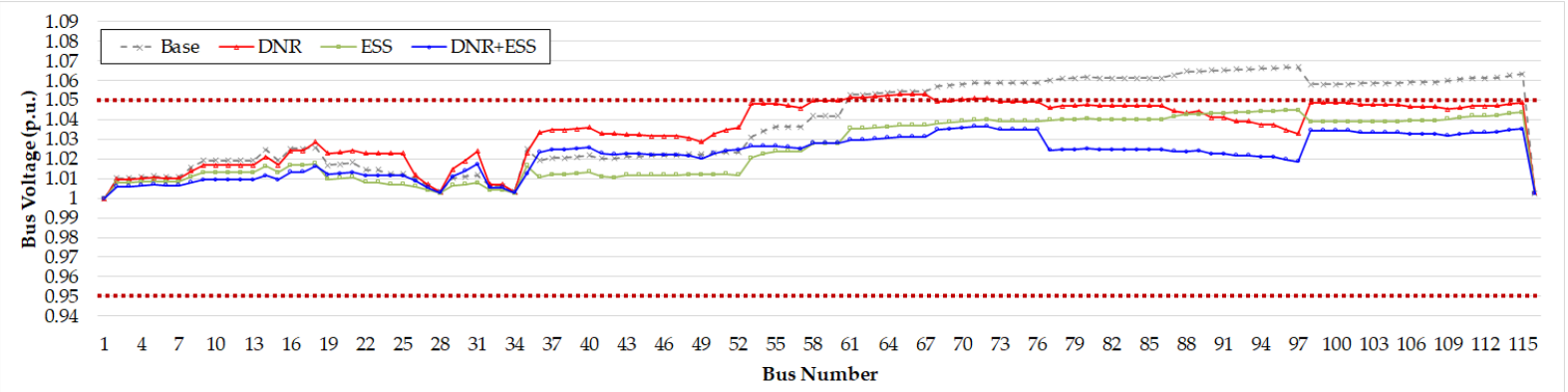

Figure 5.14 Entire network bus voltages of Fig. 5.9−5.12

Among the results, Figures 5.10 and 5.11 warrant special attention as they allow us to discern the differences between two methodologies for mitigating the system constraint violation problem depicted in Figure 5.9. When compared to Figure 5.9, the DNR method in Figure 5.10 could resolve some overvoltage and overflow issues. However, it was unable to fundamentally address the overflow in lines near the feeder where reversal power flows converge. In contrast, the ESS method in Figure 5.11 appears to handle such problems more effectively than the DNR method. Nevertheless, since the topology of the system remains unchanged in this case, although the overall level of overflow and overvoltage decreases, the pattern itself does not undergo a substantial transformation.

5.2.2. Results of Using Safety Constraints

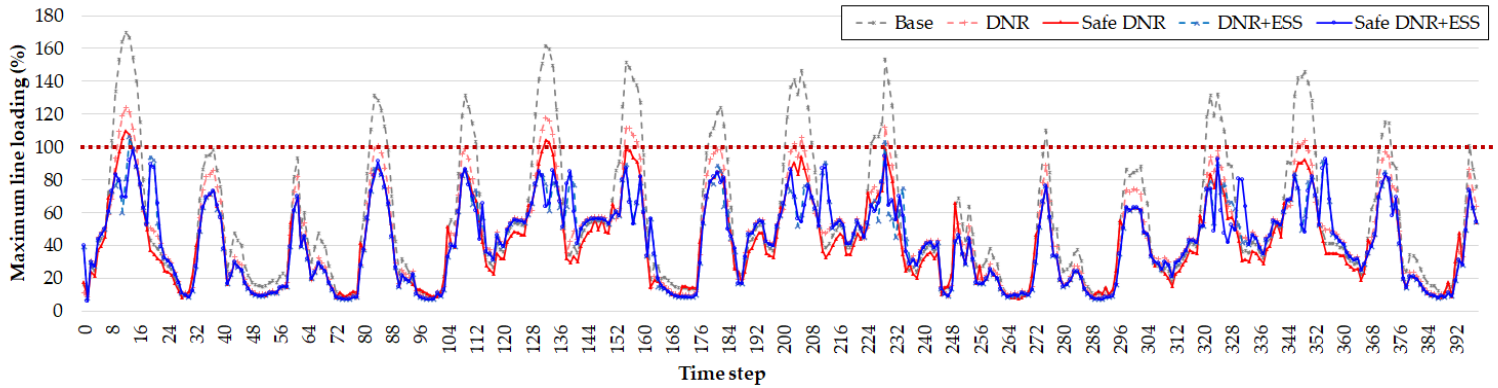

Figure 5.15 Maximum test distribution network line lodings for test dataset (Case 2)

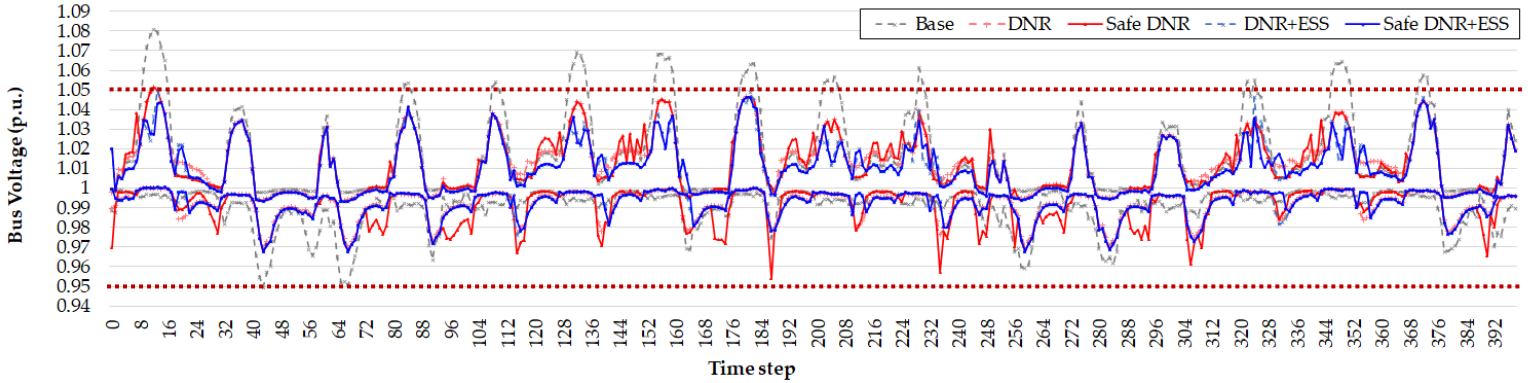

Figure 5.16 Maximum and minimum test distribution network bus voltages for test dataset (Case 2)

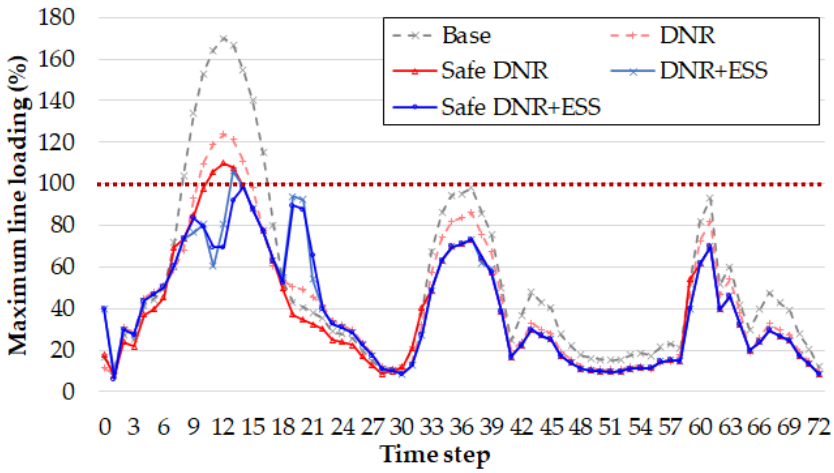

Figure 5.17 Part of Figure 5.15

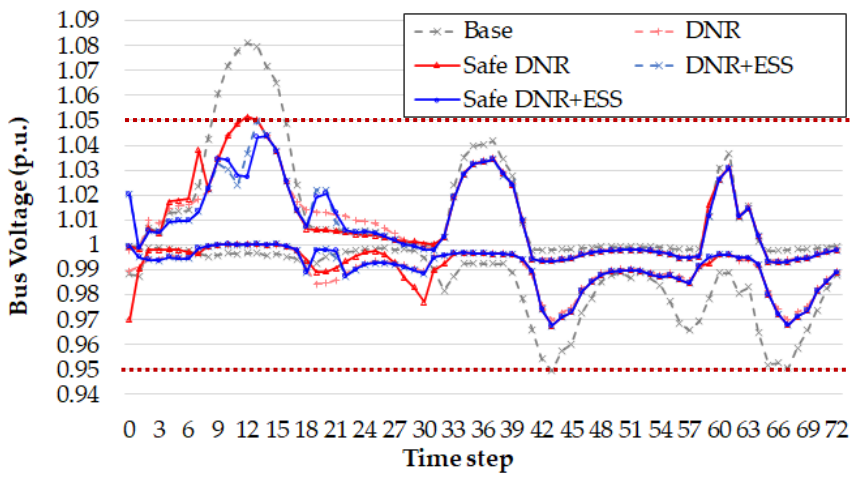

Figure 5.18 Part of Figure 5.16

In Simulation Case 2, we sought to examine the changes resulting from the application of the safety algorithm, especially in the DNR problem. While we verified in the previous chapter

that the safety module can strictly satisfy the equality constraint, in this simulation, we additionally found that the algorithm trained with the application of the safety module shows better performance. This difference is more noticeable when comparing Case 2-B and Case 2-C, where DNR is used alone, as opposed to the difference between Case 2-D and Case 2-E, where sufficient resources are available. This can be attributed to the safety module providing the agent with more useful samples, potentially playing a guiding role in the training process. We note that in Case 2-E, where all proposed assets and algorithms from this dissertation were ultimately applied, not a single network constraint violation occurred in the test dataset.

Chapter 6. Conclusions and Future Extensions

6.1. Conclusions

This dissertation's primary aim was to develop a method for Distribution System Operators (DSOs) to ensure short-term stability in a distribution network permeated with renewable energy sources. This was achieved by introducing a safe reinforcement learning (RL) strategy that treats this issue as an optimal operation problem.

The study first embarked on evaluating the necessity, jurisdiction, and function of the DSO in an environment of the distribution system which is redefined by the ingress of renewable energy sources. The primary objective of the DSO, in maintaining system stability through active system management, was established in this context. Moreover, the research successfully merged the real physical system and the system within a simulated environment into a unified Cyber-Physical System (CPS). This CPS served as the backdrop for the

implementation of such system management.

Though the controllable factors for DSOs differ depending on the specific system environment, this research intensively focused on two approaches: the reconfiguration of the distribution system via sectionalizing switches and the modification of the system's current flow using energy storage systems (ESSs). Each method was formally defined as an optimization problem.

In addition, this study reframed these optimization problems as decision-making processes over multiple time, subsequently reformulating them as MDP problems. An RL algorithm was developed to address these, and the unique Dueling Deep Q-learning algorithm was specifically designed to align with the proposed system operation methodologies.

Finally, the designed safety reinforcement learning model is simulated on the modified IEEE 123 node test system. It is demonstrated that the proposed model performs more effectively when more than one operational strategy is adopted simultaneously and when the safety of reinforcement learning is considered.

6.2. Future Works

This research commences with the framework for DSOs to perform active network management at the near real-time scale, demonstrating the feasibility of secure operation through reinforcement learning methodologies. However, the research assumed that each control method transpires actions or decisions at the same time scale, specifically every hour. In practice, methods available to the DSO likely occur at different time scales. Consequently, future research necessitates the exploration of DSO strategies under such circumstances.

Relevant to this problem is the concept of Multi-agent Reinforcement Learning (MARL) in the field of reinforcement learning. MARL is an actively researched area in academia nowadays, which addresses competition and cooperation strategies of multiple agents which are homogeneous or heterogeneous, and dealing with asymmetry in observable information during training and deployment. Thus, it can be posited that MARL possesses a high potential for application in future power systems.

Appendix A. Network data

All elements are based on IEEE123 Node Test Feeder [39]. However, this dissertation assumes a balanced three-phase system and analyzes the given system as a single-phase, and some figures are changed accordingly. All power system components in the simulation are implemented based on the *pandapower* 2.11.1 library in Python 3.9 language, as far as supported.

Load data

Node	Load Type	kW	kVAr
1	PQ	40	20
2	PQ	20	10
4	PQ	40	20
5	I	20	10
6	Z	40	20
7	PQ	20	10
9	PQ	40	20
10	I	20	10
11	Z	40	20
12	PQ	20	10
16	PQ	40	20
17	PQ	20	10
19	PQ	40	20
20	I	40	20
22	Z	40	20
24	PQ	40	20
28	I	40	20

29	Z	40	20
30	PQ	40	20
31	PQ	20	10
32	PQ	20	10
33	I	40	20
34	Z	40	20
35	PQ	40	20
37	Z	40	20
38	I	20	10
39	PQ	20	10
41	PQ	20	10
42	PQ	20	10
43	Z	40	20
45	I	20	10
46	PQ	20	10
47	I	105	75
48	Z	210	150
49	PQ	105	75
50	PQ	40	20
51	PQ	20	10
52	PQ	40	20
53	PQ	40	20
55	Z	20	10
56	PQ	20	10
58	I	20	10
59	PQ	20	10
60	PQ	20	10
62	Z	40	20
63	PQ	40	20
64	I	75	35
65	Z	105	75
66	PQ	75	35
68	PQ	20	10
69	PQ	40	20
70	PQ	20	10
71	PQ	40	20
73	PQ	40	20
74	Z	40	20
75	PQ	40	20
76	I	245	180

77	PQ	40	20
79	Z	40	20
80	PQ	40	20
82	PQ	40	20
83	PQ	20	10
84	PQ	20	10
85	PQ	40	20
86	PQ	20	10
87	PQ	40	20
88	PQ	40	20
90	I	40	20
92	PQ	40	20
94	PQ	40	20
95	PQ	20	10
96	PQ	20	10
98	PQ	40	20
99	PQ	40	20
100	Z	40	20
102	PQ	20	10
103	PQ	40	20
104	PQ	40	20
106	PQ	40	20
107	PQ	40	20
109	PQ	40	20
111	PQ	20	10
112	I	20	10
113	Z	40	20
114	PQ	20	10

Line data

Number	Bus1	Bus2	Length(ft)
1	1	2	175
2	1	3	250
3	1	7	300
4	3	4	200
5	3	5	325

6	5	6	250
7	7	8	200
8	8	12	225
9	8	9	225
10	8	13	300
11	9	14	425
12	13	34	150
13	13	18	825
14	14	11	250
15	14	10	250
16	15	16	375
17	15	17	350
18	18	19	250
19	18	21	300
20	19	20	325
21	21	22	525
22	21	23	250
23	23	24	550
24	23	25	275
25	25	26	350
26	25	28	200
27	26	27	275
28	26	31	225
29	27	33	500
30	28	29	300
31	29	30	350
32	31	32	300
33	34	15	100
34	35	36	650
35	35	40	250
36	36	37	300
37	36	38	250
38	38	39	325
39	40	41	325
40	40	42	250
41	42	43	500
42	42	44	200
43	44	45	200
44	44	47	250
45	45	46	300

46	47	48	150
47	47	49	250
48	49	50	250
49	50	51	250
50	51	108	1500
51	52	53	200
52	53	54	125
53	54	55	275
54	54	57	350
55	55	56	275
56	57	58	250
57	57	60	750
58	58	59	250
59	60	61	550
60	60	62	250
61	62	63	175
62	63	64	350
63	64	65	425
64	65	66	325
65	67	68	200
66	67	72	275
67	67	97	250
68	68	69	275
69	69	70	325
70	70	71	275
71	72	73	275
72	72	76	200
73	73	74	350
74	74	75	400
75	76	77	400
76	76	86	700
77	77	78	100
78	78	79	225
79	78	80	475
80	80	81	475
81	81	82	250
82	81	84	675
83	82	83	250
84	84	85	475
85	86	87	450

86	87	88	175
87	87	89	275
88	89	90	225
89	89	91	225
90	91	92	300
91	91	93	225
92	93	94	275
93	93	95	300
94	95	96	200
95	97	98	275
96	98	99	550
97	99	100	300
98	101	102	225
99	101	105	275
100	102	103	325
101	103	104	700
102	105	106	225
103	105	108	325
104	106	107	575
105	108	109	450
106	109	110	300
107	110	111	575
108	110	112	125
109	112	113	525
110	113	114	325
111	18	35	375
112	115	1	400
113	13	52	400
114	60	67	350
115	97	101	250
116	54	94	400
117	37	59	400
118	56	88	400
119	71	114	800
120	46	65	400
121	17	96	400
122	30	48	400
123	39	66	400

Line Configuration

Parameter	Value	Description
c_nf_per_km	10.75	line capacitance (line-to-earth) in nano Farad per km
r_ohm_per_km	0.306	line resistance in ohm per km
x_ohm_per_km	0.33	line reactance in ohm per km
max_i_ka	0.35	maximum thermal current in kilo Ampere

References

[1] S. W. Kim, J. Kim, Y. G. Jin, and Y. T. Yoon, "Optimal Bidding Strategy for Renewable Microgrid with Active Network Management," *Energies*, vol. 9, no. 1, Art. no. 1, Jan. 2016.

[2] M. Glavic, "(Deep) Reinforcement learning for electric power system control and related problems: A short review and perspectives," *Annu. Rev. Control*, vol. 48, pp. 22–35, Jan. 2019.

[3] S. Heinen, D. Elzinga, S.–K. Kim, and Y. Ikeda, "Impact of Smart Grid Technologies on Peak Load to 2050," OECD, Paris, Aug. 2011.

[4] J. Zhang, H. Cheng, and C. Wang, "Technical and economic impacts of active management on distribution network," *Int. J. Electr. Power Energy Syst.*, vol. 31, no. 2, pp. 130–138, Feb. 2009.

[5] T. E. Dy Liacco, "Real–time computer control of power systems," *Proc. IEEE*, vol. 62, no. 7, pp. 884–891, Jul. 1974.

[6] M. Glavic, R. Fonteneau, and D. Ernst, "Reinforcement Learning for Electric Power System Decision and Control: Past Considerations and Perspectives," *IFAC–Pap.*, vol. 50, no. 1, pp. 6918–6927, Jul. 2017.

[7] W. Duo, M. Zhou, and A. Abusorrah, "A Survey of Cyber Attacks on Cyber Physical Systems: Recent Advances and Challenges," *IEEECAA J. Autom. Sin.*, vol. 9, no. 5, pp. 784–800, May 2022.

[8] Y. Lu, "Cyber Physical System (CPS)–Based Industry 4.0:

A Survey," *J. Ind. Integr. Manag.*, vol. 02, no. 03, p. 1750014, Sep. 2017.

[9] H. Fan, M. Ni, L. Zhao, and M. Li, "Review of cyber physical system and cyber attack modeling," in *2020 12th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Sep. 2020, pp. 1-5.

[10] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on cyber-physical systems," *IEEECAA J. Autom. Sin.*, vol. 4, no. 1, pp. 27-40, Jan. 2017.

[11] M. D. Ilić, L. Xie, U. A. Khan, and J. M. F. Moura, "Modeling of Future Cyber-Physical Energy Systems for Distributed Sensing and Control," *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 40, no. 4, pp. 825-838, Jul. 2010.

[12] L. Xie, Y. Zhang, and M. D. Ilic, "Multi-scale Integration of Physics-Based and Data-Driven Models in Power Systems," in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, Apr. 2012, pp. 129-137.

[13] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring Safety, Security, and Sustainability of Mission-Critical Cyber-Physical Systems," *Proc. IEEE*, vol. 100, no. 1, pp. 283-299, Jan. 2012.

[14] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, "Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison," *Engineering*, vol. 5, no. 4, pp. 653-661, Aug. 2019.

[15] S. Civanlar, J. J. Grainger, H. Yin, and S. S. H. Lee, "Distribution feeder reconfiguration for loss reduction," *IEEE Trans. Power Deliv.*, vol. 3, no. 3, pp. 1217-1223, Jul.

1988.

[16] D. P. Bernardon, A. P. C. Mello, L. L. Pfitscher, L. N. Canha, A. R. Abaide, and A. A. B. Ferreira, "Real-time reconfiguration of distribution network with distributed generation," *Electr. Power Syst. Res.*, vol. 107, pp. 59-67, Feb. 2014.

[17] D. P. Bernardon, V. J. Garcia, A. S. Q. Ferreira, and L. N. Canha, "Electric distribution network reconfiguration based on a fuzzy multi-criteria decision making algorithm," *Electr. Power Syst. Res.*, vol. 79, no. 10, pp. 1400-1407, Oct. 2009.

[18] S. Lei, Y. Hou, F. Qiu, and J. Yan, "Identification of Critical Switches for Integrating Renewable Distributed Generation by Dynamic Network Reconfiguration," *IEEE Trans. Sustain. Energy*, vol. 9, no. 1, pp. 420-432, Jan. 2018.

[19] C. K. Das, O. Bass, G. Kothapalli, T. S. Mahmoud, and D. Habibi, "Overview of energy storage systems in distribution networks: Placement, sizing, operation, and power quality," *Renew. Sustain. Energy Rev.*, vol. 91, pp. 1205-1230, Aug. 2018.

[20] B. Yang *et al.*, "Optimal sizing and placement of energy storage system in power grids: A state-of-the-art one-stop handbook," *J. Energy Storage*, vol. 32, p. 101814, Dec. 2020.

[21] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. in Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 1998.

[22] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, 2007.

[23] A. Agrawal, S. Barratt, S. Boyd, and B. Stellato, "Learning Convex Optimization Control Policies." arXiv, Dec. 19, 2019. Accessed: May 25, 2023. [Online]. Available: http://arxiv.org/abs/1912.09529

[24] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Feb. 2015.

[25] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling Network Architectures for Deep Reinforcement Learning." arXiv, Apr. 05, 2016.

[26] A. Pereira and C. Thomas, "Challenges of Machine Learning Applied to Safety-Critical Cyber-Physical Systems," *Mach. Learn. Knowl. Extr.*, vol. 2, no. 4, Art. no. 4, Dec. 2020.

[27] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning." arXiv, Dec. 19, 2013.

[28] Y. Wang, Y. Xu, J. Li, J. He, and X. Wang, "On the Radiality Constraints for Distribution System Restoration and Reconfiguration Problems." arXiv, Mar. 15, 2020. Accessed: May 26, 2023. [Online]. Available: http://arxiv.org/abs/1912.05185

[29] Y. Tang and S. Agrawal, "Discretizing Continuous Action Space for On-Policy Optimization," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, Art. no. 04, Apr. 2020.

[30] G. Dulac-Arnold *et al.*, "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis," *Mach. Learn.*, vol. 110, no. 9, pp. 2419-2468, Sep. 2021.

[31] H. Li and H. He, "Learning to Operate Distribution

Networks With Safe Deep Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 1860–1872, May 2022.

[32] J. Garc ı a and F. Fernandez, "A Comprehensive Survey on Safe Reinforcement Learning," *J. Mach. Learn. Res.*, 2015.

[33] S. Gu *et al.*, "A Review of Safe Reinforcement Learning: Methods, Theory and Applications." arXiv, Feb. 20, 2023.

[34] L. Brunke *et al.*, "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning." arXiv, Dec. 06, 2021.

[35] H. Odriozola-Olalde, M. Zamalloa, and N. Arana-Arexolaleiba, "Shielded Reinforcement Learning: A review of reactive methods for safe learning," in *2023 IEEE/SICE International Symposium on System Integration (SII)*, Jan. 2023, pp. 1–8.

[36] B. Enacheanu, B. Raison, R. Caire, O. Devaux, W. Bienia, and N. HadjSaid, "Radial Network Reconfiguration Using Genetic Algorithm Based on the Matroid Theory," *IEEE Trans. Power Syst.*, vol. 23, no. 1, pp. 186–195, Feb. 2008.

[37] G. Ceusters, L. R. Camargo, R. Franke, A. Now , and M. Messagie, "Safe reinforcement learning for multi-energy management systems with known constraint functions," *Energy AI*, vol. 12, p. 100227, Apr. 2023.

[38] S. Jeon, H. T. Nguyen, and D.-H. Choi, "Safety-Integrated Online Deep Reinforcement Learning for Mobile Energy Storage System Scheduling and Volt/VAR Control in Power Distribution Networks," *IEEE Access*, pp. 1–1, 2023.

[39] IEEE, "Resources - IEEE PES Test Feeder." https://cmte.ieee.org/pes-testfeeders/resources/ (accessed

May 29, 2023).

[40] S. H. Oh, Y. T. Yoon, and S. W. Kim, "Online reconfiguration scheme of self-sufficient distribution network based on a reinforcement learning approach," *Appl. Energy*, vol. 280, p. 115900, Dec. 2020.

[41] H. S. V. S. K. Nunna, A. Sesetti, A. K. Rathore, and S. Doolla, "Multiagent-Based Energy Trading Platform for Energy Storage Systems in Distribution Systems With Interconnected Microgrids," *IEEE Trans. Ind. Appl.*, vol. 56, no. 3, pp. 3207-3217, May 2020.

[42] Korea Southern Power Co. The Korea Southern Power wind power generation real time monitoring information. Available online: https://www.data.go.kr/dataset/15012779/fileData.do (accessed May 30, 2023).

[43] Korea Rural Community Corporation. Jindo Solar power plant generation data; 2015. Available online: https://www.data.go.kr/dataset/15005796/fileData.do (accessed May 30, 2023).

[44] EERE. Commercial and residential hourly load profiles for all TMY3 locations in the United States. Available online: https://openei.org/doe-opendata/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-unitedstates (accessed May 30, 2023).

초 록

실 계통 적용을 고려한 안전 강화학습 기반의 능동 배전망 운영전략에 대한 연구

오 석 화
전기·정보공학부
공과대학원
서울대학교

본 연구의 목적은 신재생에너지원이 유입된 배전계통의 안정성을 실시간으로 관리하기 위한 배전계통운영자(DSO) 방법론과 이를 풀 수 있는 안전 강화학습 기법을 제안하는 것이다.

이를 위해 본 연구에서는 우선 신재생에너지원의 유입으로 인해 변화하는 배전계통 환경에서의 배전계통운영자의 필요성, 그리고 권한과 역할에 대해 검토하며, 배전계통운영자가 계통 안정성을 유지하기 위한 능동적 계통 운영의 목적을 정의한다. 나아가 이러한 관리가 이루어지는 배전계통 환경을 하나의 가상 물리 시스템(CPS)으로 정의하여 실제의 물리적 계통과 시뮬레이션 환경 상의 계통을 하나로 종합하고자

하였다.

배전계통운영자가 제어할 수 있는 대상은 각 계통 환경에 따라서 달라지나, 본 연구에서는 계통 내 스위치를 통한 배전계통 재구성, 에너지저장장치를 활용한 계통 내 조류량 변경의 두 가지 방법론을 집중적으로 다루고자 하였으며, 이들 각각을 최적화 문제로 정식화하였다.

아울러, 본 연구에서는 이러한 최적화 문제를 연속된 시간에서의 제어 결정 문제로 보고 마르코프 결정 프로세스(MDP)로 재정식화하였으며, 이를 해결하기 위한 강화학습 알고리즘을 설계하였다. 강화학습 분야에는 목적 및 대상이 되는 데이터의 특성에 따라 다양한 형태의 알고리즘을 설계할 수 있으며, 본 연구에서는 제안된 각 계통 운영 방법론에 맞는 Dueling Deep Q-learning 알고리즘을 설계하였다.

한편, 현재까지 강화학습 알고리즘을 활용해 전력 시스템의 운영 문제를 해결하고자 한 연구가 다수 있어왔으나, 이를 시뮬레이션 환경이 아닌 실제의 물리적 시스템에 적용하고자 할 때에 발생할 수 있는 문제점에 대해 검토하였다. 이중 본 연구에서는 실제 계통에서 요구되는 제어 결정의 안정성 문제를 강화학습 알고리즘에 반영하기 위해 기존의 마르코프 결정 프로세스를 제약된 마르코프 결정 프로세스(Constrained Markov Decision Process)로 확장하여, 등호 제약조건을 다루기 위한 안정성 모듈 및 부등호 제약조건을 다루기 위한 적응 비용 함수를 설계하였다.

결과적으로 설계된 안전 강화학습 모델을 IEEE 123 모선 시험

계통에서 시뮬레이션 함으로써, 하나 이상의 운영전략을 동시에 취할 경우, 또 강화학습의 안전성을 고려할 경우 보다 효과적인 성능을 보임을 입증하였다.

배전계통운영자는 본 논문에서 제안하는 실시간 배전계통 운영을 위한 강화학습 프레임워크의 도입을 통해, 실제 물리적 계통으로부터 도출되는 요구사항을 해결하기 위한 강화학습 알고리즘을 설계할 수 있으며, 또 해당 알고리즘의 결정이 물리적 계통의 안정성 제약 위배를 최소화하도록 함으로써, 증가하는 배전계통의 복잡성에 대응하기 위한 하나의 계통 운영 전략으로서 취할 수 있을 것이다.

주요어 : 능동적 배전계통운영자(Active DSO), 가상 물리 시스템(Cyber-Physical System), 배전계통 재구성, 다중 에너지저장장치 운용, 안전 강화학습

학 번 : 2017-26165