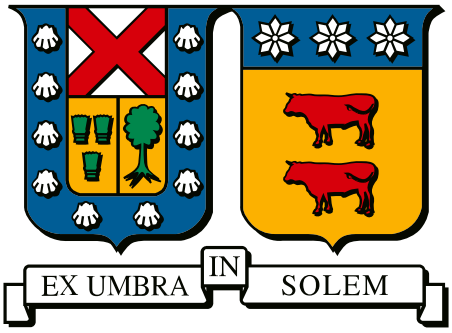


Universidad Técnica Federico Santa María
Department of Electrical Engineering

Physics-informed Machine Learning for the Sparse and Distributed Operation of Active Distribution Networks under Uncertainty

Author:
Miguel Angel
Huerta Quian

Thesis Director:
Dr. Alejandro Alberto
Angulo Cárdenas

A thesis submitted in partial fulfillment of the requirements for the degree of

MSc Electrical Engineering

April 23, 2025

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título; ☒ Tesis de Postgrado;

Título del trabajo: Physics-informed Machine Learning for the Sparse and Distributed Operation of Active Distribution Networks under Uncertainty

Nombre del candidato(a): Miguel Angel Huerta Quian

Carrera / Grado: Magíster en Ciencias de la Ingeniería Eléctrica

Campus: Casa Central Valparaiso **;** **Departamento**: Ingeniería Eléctrica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Alejandro Angulo, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 19/06/2025 ; **Firma:** _____

Estudiante o Candidato(a):

Fecha: 19/06/2025 ; **Firma:** _____

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Abstract

Legacy distribution systems need to adjust to emerging operational contexts. The rise in distributed generation will introduce efficiency challenges, and the limited communication capabilities of these systems hinder centralized control methods. Academic research has thus focused on distributed control strategies for photovoltaic generators to enhance operations and handle uncertainties. Traditional literature approaches utilize distribution system models for optimizing control parameters, which presents a computational challenge. This thesis work presents a new framework for defining local operational policies for distributed generators by integrating machine learning algorithms into a control structure for real–time operational solutions. The work focuses on the design, training and integration of an intelligent agent that optimizes local operation under conditions of uncertainty, while minimizing energy consumption and voltage violations. The intelligent agent takes advantage of the physical structure of the system and causal relationships, processing the uncertainty of the system and historical data to adjust local controllers. Computational experiments demonstrate the effectiveness of the proposed method, surpassing state–of–the–art frameworks in scenarios of high penetration of renewable energies. The approach guarantees high economic efficiency and effective voltage regulation, while addressing the physical interpretability of machine learning models in energy systems. Sensitivity studies and simulation results confirm that the methodology outperforms traditional methods and manages distribution networks efficiently under conditions of uncertainty.

Acknowledgements

Antes que nada, quiero dar las gracias a mis padres y abuelos por haber sido mi mayor apoyo a lo largo de mi formación. Gracias por su amor incondicional, por brindarme siempre lo necesario para seguir adelante, por su paciencia infinita y por sus palabras de aliento en cada obstáculo que he enfrentado. Sus enseñanzas y valores han sido la base de todo lo que he logrado, y por ello les estaré eternamente agradecido.

A mis amigos, con quienes compartí esta increíble etapa universitaria, gracias por estar siempre ahí. Por las largas jornadas de estudio, por las risas y las experiencias vividas, por el apoyo en los días difíciles y por las celebraciones cuando logramos nuestras metas. Sin ustedes, este camino no habría sido el mismo.

A mi profesor guía, Alejandro Angulo, quien ha estado a mi lado durante todo este recorrido académico. Gracias por su dedicación y paciencia para orientarme en cada paso y por ser una fuente constante de inspiración. Su apoyo y confianza en mí han sido esenciales para poder alcanzar mis metas. Siempre le estaré agradecido por ello.

Contents

List of Figures

List of Tables

Chapter 1

Motivation

Integrating renewable energy sources to address the continuously escalating demand for power systems has emerged as a viable solution. Such an integration harbors numerous advantages, including mitigating environmental impact by displacing non-renewable energy resources. However, a substantial challenge is also associated with it: ensuring a reliable supply of electrical power that meets acceptable quality standards [1]. This issue presents several technical challenges, with voltage regulation being the most critical, as it often restricts the integration of distributed generation (DG) into distribution networks.

Within this context, traditional utility equipment's slow control actions do not suffice for comprehensive voltage regulation [2]. Conventional strategies depend on Volt/VAr algorithms to orchestrate the functioning of tap–changing transformers, capacitor banks, and voltage regulators. They aim to respond to alterations in feeder load level to sustain voltages within safe parameters and, possibly, minimize feeder losses. Although Volt/VAr control strategies have exhibited efficacy on traditional feeders, they have also been recognized as potentially inadequate when extensive amounts of unregulated photovoltaic (PV) generation increasingly penetrate distribution feeders [3]. A potential resolution to the problem highlighted above is integrating intelligence into the DGs to control their generation set–points [4], which delineate the functioning of the voltage source converters (VSCs). Such an approach is proposed as an alternative to enhance the flexibility of established distribution networks significantly, necessitating minimal equipment and infrastructure upgrades.

Along these lines, multiple works have proposed control strategies to define DG dispatch, focusing on designing policies that can effectively manage the uncertainty inherent in the data, and creating models and algorithms that can effectively utilize distributed energy resources (DERs) in real–time. This work belongs to the latter group. This approach is distinguished by utilizing real-time operation techniques, which involve designing algorithms that adjust the output of DGs as actions that can be executed during time instants in which partial or total uncertainty is disclosed. This concept promotes the development of modeling policies based on the uncertainty of the problem. Examples are policies that are based on the continuous calculation of the reactive power of DGs as a function of voltage [5, 6], making use of defined functional structures. Similarly, proposals have recently emerged that delegate the handling of uncertainty to more complex Machine Learning (ML)-based structures [7]. These approaches are a substantial improvement over the strategy proposed by IEEE 1547.8 [8], but carry with them a number of simplifications that define a field of improvement with respect to real-time controller design.

In order to take advantage of the benefits of using an ML-based agent for uncertainty management and to bridge the gap between the modeling of the system and its real-world operation, our research focuses on the development of an agent endowed with the ability to update an affine control policy. In this context, we refer as agent to an computational model based on artificial neural networks (ANN) that autonomously determines control actions based on predefined system inputs and objectives, while maintaining the interpretability of its decisions through the physical principles of the system. To this end, we integrate physics-informed neural networks (PINNs) with optimization

techniques to improve the agent's decision-making process. PINNs are ANN that use the physical laws of a system, such as partial differential equations, to guide model training [9]. This practice has proven to be especially effective for problems with limited data but well-understood physical dynamics. Recent literature suggests that the incorporation of Karush–Kuhn–Tucker (KKT) conditions provide a natural way to link PINN physics-based training with optimization models [10]. By defining the optimality conditions of an optimization problem, KKT conditions allow the development of analytical expressions that, with proper reinterpretation, serve to guide agent training. This integration allows the agent to dynamically adjust the control parameters, optimizing system performance and achieving economic efficiency superior to that of existing models.

1.1 Hypothesis

Given an active distribution network equipped with controllable DGs, a weak communication infrastructure that imposes communication constraints between components, and assuming that the system can be partitioned and controlled coordinatedly by controllers associated with the active components, whose main characteristic is having partial access to uncertain states' history and parameters, we posit the following:

> *It is possible to manage an active distribution network by applying distributed control techniques and optimization procedures under a wide range of real-time operating conditions. Coordination of these processes with physics-informed artificial intelligence algorithms allows for optimal dispatch of spatially distributed generation sources even under high levels of uncertainty. The application of this framework yields more efficient performance than the control configuration recommended by the IEEE-1547 standard.*

1.2 Objectives

Main objective

The main objective of this work is to develop a framework for the centralized definition of local operating policies for DGs by incorporating ML algorithms into the control structure of a distribution system. The framework aims to optimize system operation under uncertainty, ensuring global optimality in local operations while minimizing energy consumption and voltage violations. A key aspect of the approach is the integration of the system's physical structure and causal relationships into the intelligent agent's design, making the agent physically interpretable. This design allows the agent to capture the inherent physics of the distribution network and transfer this knowledge to define the operational parameters of local controllers in real-time, ensuring effective system performance. The scenarios are specifically designed with high solar energy penetration to emulate future operating conditions, enhancing the agent's ability to improve traditional voltage control methods and contributing to more efficient, responsive, and sustainable power distribution methodologies.

Specific objectives

1. In the context of distribution networks, review the current literature on optimal power flow, voltage control strategies, and optimization under uncertainty.
2. In the context of neural networks, carry out a literature review of ANN architectures and training mechanisms based on the physics of the systems. In particular, the aim is to develop guidelines that define the agent, together with a training environment that allows the inherent physics of the system to be captured under conditions of uncertainty.
3. Define and implement an optimal power flow model suitable for partitioning and integrating the response of the intelligent agent using standard solvers or real-time simulators, considering the equilibrium equation for the closed-loop controlled distribution network.
4. Design and develop out-of-sample computational experiments using a set of IEEE instances under different distributed generation and uncontrolled load scenarios.

Chapter 2

Background

This chapter outlines the theoretical framework and the fundamental modeling techniques used throughout this work. First, the conventional and non–conventional mathematical notation used for the mathematical formulation of the operating models is stated. Secondly, some background information on active distribution networks is presented, emphasizing the challenges and opportunities they present, together with the operating structures and regulatory framework that defines the operating standards. Likewise, the section continues with the presentation of the background on the optimal power flow problem, which from its fundamental formulation, derived formulations are developed and presented, allowing the construction of linear models for its subsequent use. Finally, the background related to artificial neural networks and their link with the resolution of optimization problems and optimal operation under uncertainty is presented. The topologies of existing networks are formalized, along with establishing their interaction with the operating environment according to their designated task, as well as defining training and operation techniques derived from the optimization theory used later in this proposal.

2.1 Mathematical notation

For the mathematical development of the models, three types of elements are considered: variables and parameters, written in roman and math italic font; vectors and matrices, written in roman bold and math bold font; and sets, written in calligraphic font.

Linear algebra

Operator $(\cdot)^\top$ corresponds to the transpose, $(\cdot)^\mathsf{H}$ means to the Hermitian transpose, and $(\cdot)^*$ corresponds for its complex conjugate. The operator $|\cdot|$ has different meanings depending on its application: in a scalar it returns the absolute value; in a phasor it returns the magnitude; and in a matrix it means the cardinality. For a given $N \times 1$ vector $\mathbf{x}$: x_i represents the i^{th} element and $\mathbf{x}^m$ denotes its m^{th} element–wise power. Let $\mathbf{y}$ be another $N \times 1$ vector, then $\mathbf{x} \odot \mathbf{y}$ and $\mathbf{x}/\mathbf{y}$ denote element–wise multiplication and quotient respectively (same notation applies for matrices). For an $N \times N$ matrix $\mathbf{A}$, with eigenvalues $\lambda_1, \dots \lambda_N$ the spectral radius $\rho(\cdot)$ is defined as $\rho(\mathbf{A}) \coloneqq \max\{|\lambda_1|, \dots |\lambda_N|\}$. For an $M \times N$ matrix $\mathbf{A}$, the Frobenius norm is $||\mathbf{A}||_F = \sqrt{\mathrm{Tr}(\mathbf{A}^\mathsf{H}\mathbf{A})}$.

2.2 Active distribution networks

An active distribution network (ADN) is an electric power distribution network designed to incorporate advanced technology and intelligence to improve the efficiency and reliability of electricity distribution in the face of scenarios with a high penetration of variable renewable resources (VREs). Traditionally, distribution networks have been passive, meaning that power flows in only one direction from the transmission grid to end users. In an ADN, however, DERs such as renewable and non-renewable DG, distributed energy storage systems (DESSs), plug-in electric vehicles (PEVs), and users are integrated into the grid and actively managed to optimize their performance and contribution to the network [11]. The interaction between the different agents that make up an ADN is depicted in Figure 2.1.

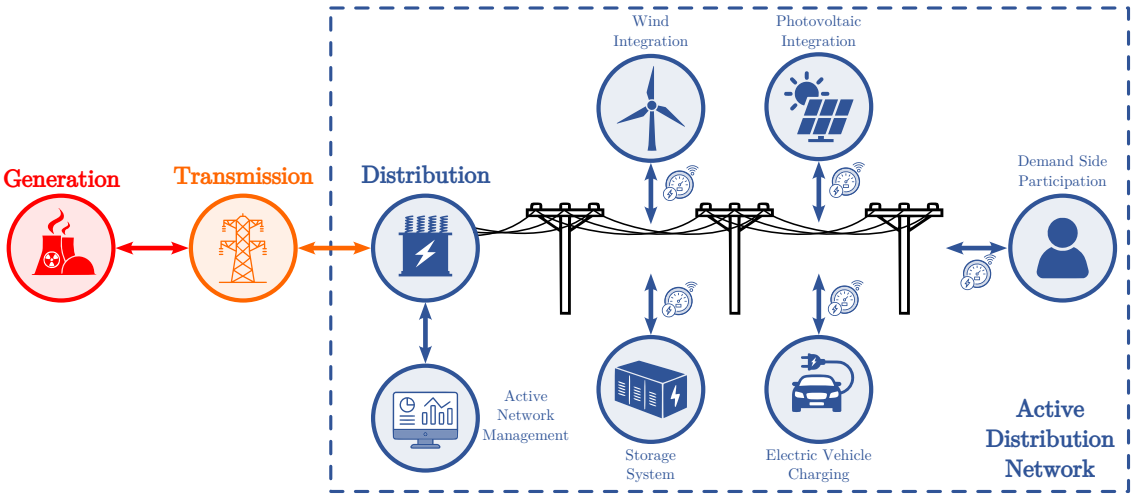


Figure 2.1: Structure of the ADNs [12].

The integration and participation of multiple agents, each with a certain degree of uncertainty, in the energy management of an electricity system introduces several challenges as well as opportunities. These complexities are especially pronounced when it comes to coordinating various energy sources in order to maintain the stability of the power system. As evidence of this, a key challenge is the effective management of DERs, which are inherently variable and less predictable than conventional energy sources. This variable nature of VREs requires the application of advanced forecasting techniques and local response strategies to ensure that generation matches demand in real time. In addition, ensuring the reliability of energy supply is becoming increasingly complex as the number of decentralized agents and VREs increases. Finally, ensuring compliance with the regulatory framework represents another critical challenge, as the integration of DERs involves coordinating a wide range of technologies and standards frameworks. This complexity requires robust control mechanisms and interoperability solutions to ensure quality of service.

Despite these challenges, the integration of DERs exhibits substantial opportunities to improve the efficiency and sustainability of the power network. The integration of multiple stakeholders in the management of a system can significantly improve energy resilience by decreasing reliance on centralized power plants, fostering a more diversified energy portfolio [13]. These energy resources provide localized generation, which allows for the reduction of transmission losses. In addition, the integration of DERs innovation in demand response solutions leading to more flexible and adaptive network management in pursuit of increasing the sustainability of the clean energy share of the overall energy mix, contributing to the reduction of greenhouse gas emissions and aligning with global climate goals [14]. To manage the system to meet these objectives, the ADNs feature advanced sensors, communication systems, and automation technologies that enable the network to detect and respond to uncertainty in real-time [15]. These technologies enable the network to actively manage the distribution of electricity and balance the generation and demand, improve the stability and reliability of the grid, and support the integration of renewable energy sources.

2.2.1 Operation of ADNs under uncertainty

The increasing integration of DERs into distribution systems has resulted in the inadequacy of traditional practices to effectively manage the high variability of these power sources [16]. In addition to the intrinsic variability of VREs, the high penetration of DERs in distribution systems causes reverse power flows, thereby increasing the likelihood of voltages exceeding the limits prescribed by regulation [15]. To meet these challenges, ADN decision–makers must design effective policies that accommodate scenarios with high uncertainty, prevailing a good balance in the trade-off between cost-effective and reliable solutions [17, 18]. Along these lines, several authors have focused their efforts on the design of policies that can effectively manage the uncertainty inherent in the data and/or on the creation of models that can effectively utilize DERs in real–time, which requires a thorough understanding of the system dynamics as well as the dynamics and limitations of the controllable devices.

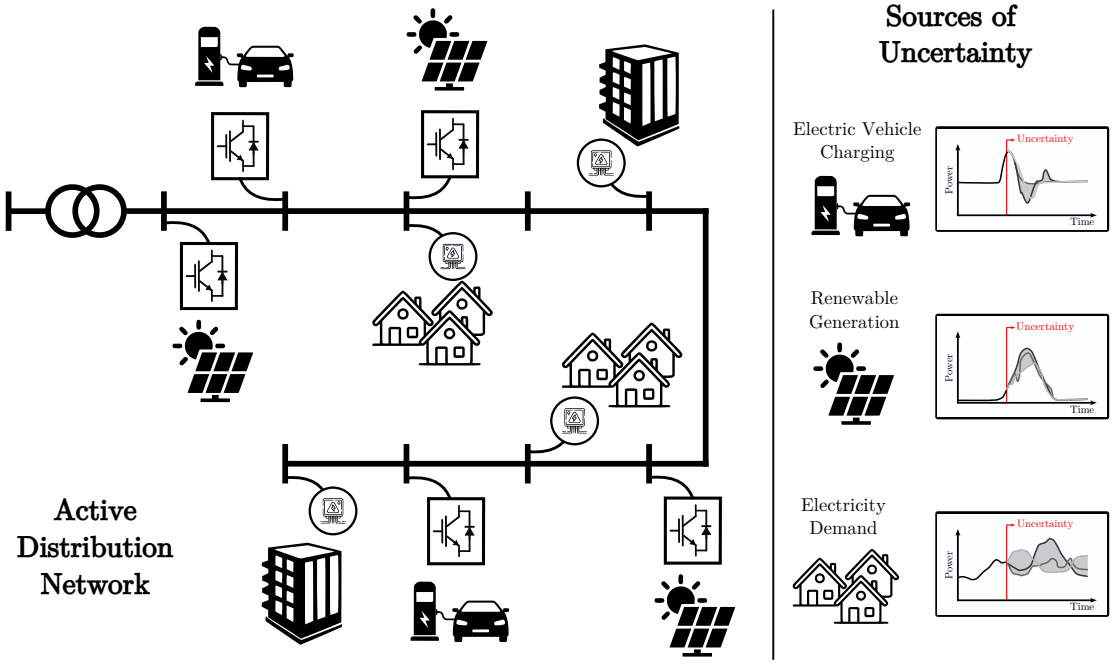


Figure 2.2: ADN subject to different sources of uncertainty.

The first approach depends on traditional OPF scheduling, where the DGs' power set–points are considered pre–determined decisions before the manifestation of uncertainty. This methodology employs a centralized (or decentralized) entity responsible for determining the operational set–points of the DGs within the operating window. These set–points are established based on historical observations of uncertainty and the global information of the network. An optimization model, incorporating the OPF as a sub–problem, is solved to accomplish this. In contrast, the second approach is distinguished by utilizing real-time operation techniques, which involve designing techniques that adjust to the output of the DGs as actions that can be executed during time instants in which partial or total uncertainty is disclosed. This concept promotes the development of modeling policies based on the uncertainty of the problem. In the context of ADNs, a frequently employed policy is the continuous calculation of the reactive power of the DGs as a function of the maximum available power [19] or voltage [4–6, 20–22]. Consequently, traditional applications are considered within this approach. Both approaches make instantaneous decisions. The former involves determining DG output, while the latter involves determining control policy parameters. Moreover, treating uncertainty in the calibration of adjustable policies is akin to the conventional OPF–based approach.

2.2.2 Modeling PV DGs

Recent literature highlights the potential for managing and controlling DGs as a leading and proficient alternative for the effective operation of ADNs [2]. Most studies concentrate on solar and wind generators interfaced by a controlled voltage source converter. However, we focus solely on PV generation, for which we provide some modeling context. In steady–state operation, PV VSCs of size $\bar{s}^{\mathrm{DG}}$ may inject complex power $(p_g^{\mathrm{DG}}, q_g^{\mathrm{DG}})$ in four modes, as shown in Figure 2.3.

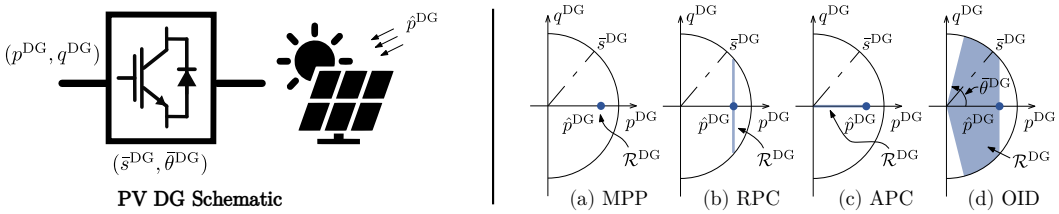


Figure 2.3: Operational chart of different operating modes for the PV DGs [23].

We denominate them maximum power point (MPP), reactive power control (RPC), active power control (APC), and optimal inverter dispatch (OID). In all operating modes, the maximum active power $\hat{p}^{\mathrm{DG}}$ depends on solar irradiance and thus is uncertain. They are mathematically described as the set:

(a) Maximum power point (MPP):

$$\mathcal{R}_g^{\mathrm{DG}}(\hat{p}_g^{\mathrm{DG}}) = \left\{ \left(p_g^{\mathrm{DG}}, q_g^{\mathrm{DG}}\right) \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} p_g^{\mathrm{DG}} = \min\left\{\hat{p}_g^{\mathrm{DG}}, \bar{s}^{\mathrm{DG}}\right\} \\ q_g^{\mathrm{DG}} = 0 \end{array} \right\}, \qquad \forall g \in \mathcal{N}^{\mathrm{DG}} \tag{2.1a}$$

(b) Reactive power control (RPC):

$$\mathcal{R}_g^{\mathrm{DG}}(\hat{p}_g^{\mathrm{DG}}) = \left\{ \left(p_g^{\mathrm{DG}}, q_g^{\mathrm{DG}}\right) \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} p_g^{\mathrm{DG}} = \hat{p}_g^{\mathrm{DG}} \\ \left(p_g^{\mathrm{DG}}\right)^2 + \left(q_g^{\mathrm{DG}}\right)^2 \leq \left(\bar{s}_g^{\mathrm{DG}}\right)^2 \end{array} \right\}, \qquad \forall g \in \mathcal{N}^{\mathrm{DG}} \tag{2.1b}$$

(c) Active power control (APC) at unity power factor:

$$\mathcal{R}_g^{\mathrm{DG}}(\hat{p}_g^{\mathrm{DG}}) = \left\{ \left(p_g^{\mathrm{DG}}, q_g^{\mathrm{DG}}\right) \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} 0 \leq p_g^{\mathrm{DG}} \leq \min\left\{\hat{p}_g^{\mathrm{DG}}, \bar{s}^{\mathrm{DG}}\right\} \\ q_g^{\mathrm{DG}} = 0 \end{array} \right\}, \qquad \forall g \in \mathcal{N}^{\mathrm{DG}} \tag{2.1c}$$

(d) Optimal inverter dispatch (OID) with constrained minimum power factor:

$$\mathcal{R}_g^{\mathrm{DG}}(\hat{p}_g^{\mathrm{DG}}) = \left\{ \left(p_g^{\mathrm{DG}}, q_g^{\mathrm{DG}}\right) \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} 0 \leq p_g^{\mathrm{DG}} \leq \hat{p}_g^{\mathrm{DG}} \\ \left(p_g^{\mathrm{DG}}\right)^2 + \left(q_g^{\mathrm{DG}}\right)^2 \leq \left(\bar{s}_g^{\mathrm{DG}}\right)^2 \\ \left|q_g^{\mathrm{DG}}\right| \leq \tan\left(\bar{\theta}_g\right) p_g^{\mathrm{DG}} \end{array} \right\}, \qquad \forall g \in \mathcal{N}^{\mathrm{DG}} \tag{2.1d}$$

where $\mathcal{N}^{\mathrm{DG}} \subseteq \mathcal{N}$ is the set of buses with a DG, while $\bar{\theta}_g$ is the limit of the power factor angle on (2.1d). In the set of equations that model the operation, the impact of uncertainty in the model is made explicit. The bound on DGs' active and reactive power output is subject to uncertainty. Note that (2.1d) defines a tractable cone, i.e., a direct product of half–spaces and second–order cones. Therefore, it is suitable for second-order cone programming (SOCP) optimization models.

2.2.3 Distributed control using DER

The integration of DER into distribution networks poses a major challenge: to efficiently manage resources with inherent uncertainties to ensure system reliability, stability and efficiency, while maximizing the benefits offered by these resources [24]. Effective DER control and management strategies are essential to meet this challenge, enabling the use of resources to be optimized to ensure that overall system performance meets desired objectives. The literature highlights multiple options for designing operational policies for DER integration, with droop control emerging as a particularly notable strategy, as highlighted in Figure 2.4.

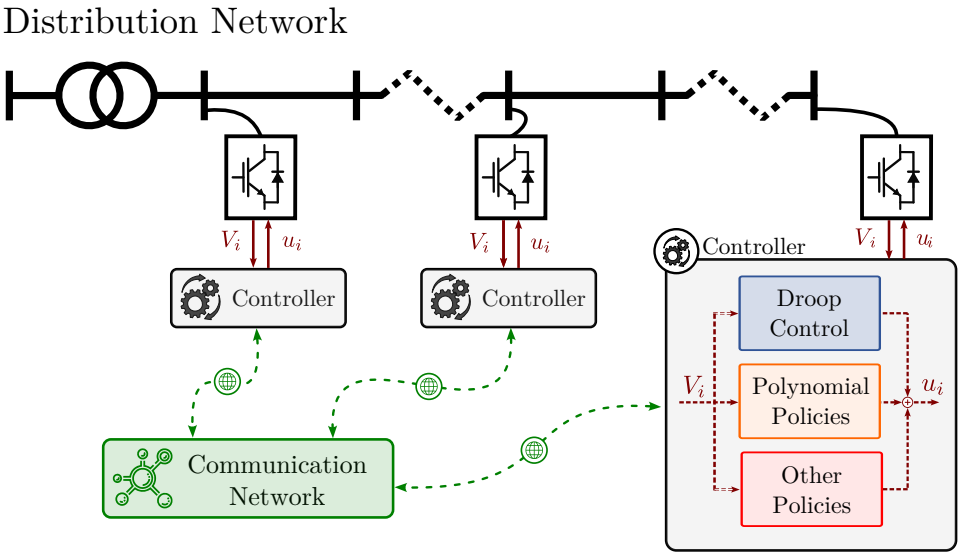


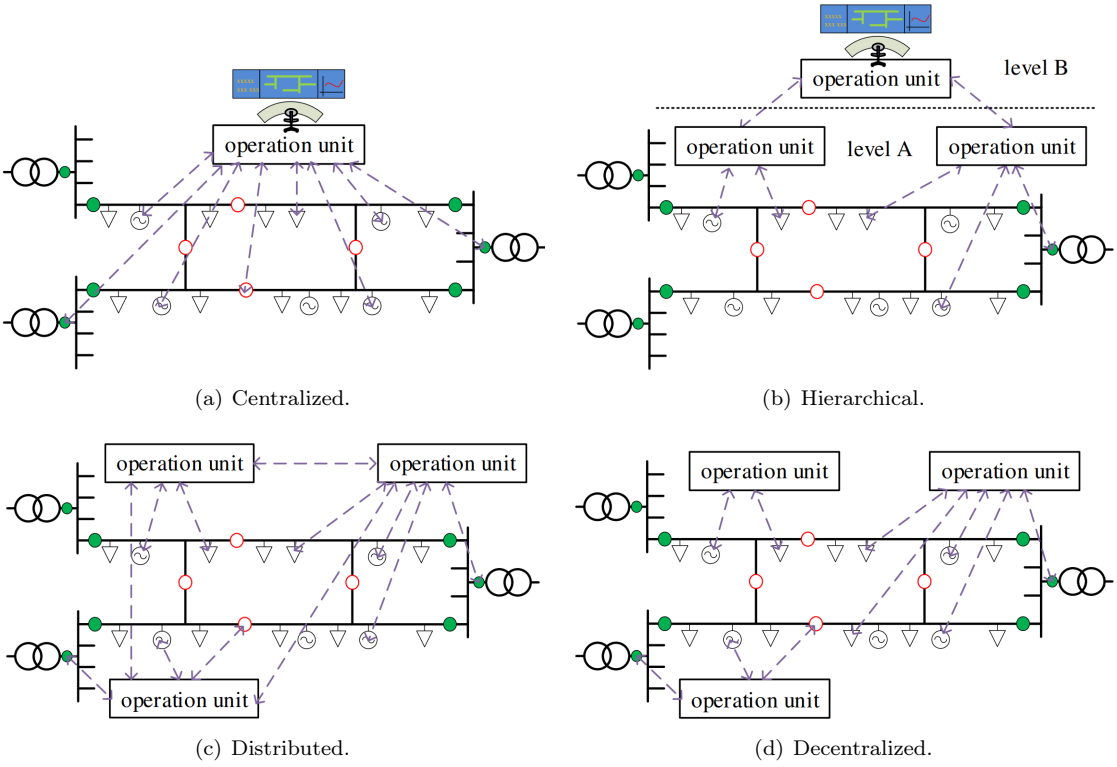
Figure 2.4: Example of an advanced multi-level voltage control strategy [24].

Also known as proportional control, this control structure is based on the concept of manipulating a non-increasing continuous variable through a specific input. This manipulation is balanced by adjusting the shape of the droop policy. Its utilization spans various applications involving linear [6], piecewise linear [8], and quadratic [25] functions. The classic example in the electrical engineering realm involves managing systematic frequency through power fluctuation in the dispatch of generators [26]. On the same line, numerous droop control methodologies have been adopted to reach primary control goals like stability and load sharing within microgrids [27]. Recent times have seen the proposition of many droop control tactics aimed at achieving voltage regulation in microgrids and ADNs, calling this strategy Volt/VAr control [4].

In this line, the IEEE 1547.8 Standard [8] endorses the implementation of droop control for dispatching DGs using fixed control policies that depend on the equipment characteristics to aid voltage regulation in distribution networks by modifying the reactive power set–point of inverters based on the local voltage deviation. On the one hand, this local approach is beneficial as it does not require global information to adjust the control parameters. However, neglecting the overall system's information can lead to significantly sub–optimal operational performance, which in some cases, can even become unstable with steep slopes of the control rules [28,29]. To incorporate global attributes into the Volt/VAr control design, numerous studies employ heuristics [30–32], optimization models [4,33], and associated control policies [5,6] to derive parameters that most accurately represent and control the system's dynamics within an operating window. In turn, understanding the operation of the distribution system as a nonlinear and dynamic environment, other proposals have resulted in assigning the definition of the functional form of control to algorithms based on artificial intelligence [7,34]. These nonlinear controllers, through latent variables encoding intelligent agents, define the operating point to operate under optimal and feasible conditions using the input attributes.

2.2.4 Operation architecture

The management of resources and operation of distribution networks can be executed via various architectures. These distinct frameworks vary in the degree of data utilization for decision-making procedures and their level of independence concerning other constituents within the system [35]. Provided below is a concise exposition of topologies frequently identified within the literature, attached at a schematic level in Figure 2.5.



(a) Centralized. (b) Hierarchical.

(c) Distributed. (d) Decentralized.

Figure 2.5: Operation architecture [35].

In general terms, these operating architectures have the following characteristics:

(a) Centralized operation architecture

In a centralized architecture, decision-making and control are concentrated in a central entity or control center. This central authority has complete visibility and control over the network, making decisions and issuing control commands to various components. This architecture enables efficient coordination and optimization of network operations, allowing for centralized monitoring, control, and management. However, it may need to be more resilient to failures or bottlenecks at the central authority.

(b) Hierarchical operation architecture

A hierarchical architecture divides the operation into multiple levels, each with specific responsibilities and decision-making authority. Higher–level entities, such as a central control center, oversee and coordinate the activities of lower–level entities, such as regional or local control centers. This architecture allows for a balance between centralized control and decentralized autonomy. It enables efficient coordination, scalability, and local decision–making while maintaining overall system–wide coordination. In this architecture, there is still a need for a centralized entity to monitor the overall performance of the system.

(c) Distributed operation architecture

In a distributed architecture, decision–making and control are distributed among multiple geographically dispersed entities within the ADN. Each entity operates autonomously and makes decisions based on local information and objectives, along with limited information

provided by its neighbors. Communication and coordination among these entities are critical to achieving system-wide optimization and stability. Distributed architectures offer high resilience, fault tolerance, and scalability but require sophisticated coordination and communication protocols.

(d) Decentralized operation architecture

In a decentralized architecture, decision–making and control authority is distributed among multiple entities or components within the ADN. Each entity, such as a local or DER controller, has a certain degree of autonomy and makes independent decisions based on local information. Communication and coordination between these entities is nil. Decentralized architectures provide flexibility, resilience, and adaptability to local conditions, but lacking coordination mechanisms, they do not guarantee overall network performance.

Regardless of the operational architecture employed, the communication system is the fundamental infrastructure for this type of architecture. It facilitates real–time monitoring, control actions, fault detection, and system–wide coordination. An unwavering commitment to a dependable and efficient communication system is of utmost importance in attaining ADN goals, including enhanced reliability, efficiency, successful integration of DERs, and effective network operations management.

2.2.5 Communication network

Communication network refers to the infrastructure and protocols that facilitate data exchange and coordination between various devices, systems and stakeholders in an operating environment. In the context of ADNs, the communication network is a fundamental component for the active management of the energy resources distributed throughout the system, since it enables the integration of monitoring, control and signal probing schemes that allow the development of sophisticated control strategies and structures [36]. In Table 2.1, some technologies commonly used in the field of power systems for the development of a communication network are presented.

Table 2.1: Communication techniques employed in the power system [36].

Technology	Advantages	Disadvantages
Power line carrier	Communication function over existing power lines; small investment; wide coverage; Easy installation and maintenance.	Narrow bandwidth; distribution transformers will block the power line carrier signal; three–phase power lines have greater signal loss; poor reliability.
Optical fiber communication.	High reliability, anti–interference ability, long communication distance, high data bandwidth.	High construction cost; shunt coupling inconvenience of de-multiplexing and multiplexing; incapable of transmit signals while powering the equipment.
Wireless public network	High reliability; simplifying the communication network design and debugging process for power company.	Lower real–time data transmission rate; the financial cost is closely related to the amount of data transmitted.
Micro power wireless networks	Easy installation and wiring; does not depend on the network of telecommunications operator, large amount of network nodes, high redundancy.	Short communication range; low data transmission rate, high failure rate of the single–point equipment.
Field bus	High anti–interference ability, high reliability, providing real–time data transmission.	Short communication range limited to a single LAN.

The technologies discussed are designed to enable seamless data exchange among smart meters, utilities, and other stakeholders involved in managing distribution grid assets. By offering secure and reliable communication channels, the network supports the effective operation of the ADN, allowing for optimal grid management and the integration of renewable energy sources. Additionally, the choice and implementation of specific architectural frameworks depend on the technology type and the intended operational structure. The primary distinctions between these frameworks are based on the hierarchy and interconnections among the involved parties.

2.2.6 Regulatory framework

A regulatory framework is defined as a set of policies and guidelines on the operation of a system that ensure its efficiency, reliability and sustainability. In the context of ADNs, these guidelines focus on facilitating the integration of DERs into the system, along with ensuring equitable access to all market participants, maintaining grid stability and encouraging energy sharing. In addition, they address issues related to data privacy, cybersecurity and interoperability standards to create a transparent and competitive environment that benefits both utilities and consumers. Thus, below are some definitions of the regulatory frameworks that underpin this proposal:

IEEE 1547.8 Standard

The IEEE 1547.8 standard delineates the intricacies of advanced controls and communication methods for inverters that support the electrical grid, along with established best practices for managing multiple inverters and microgrids. It provides leading-edge insights into the behavior and interactions of DER groups with grid equipment, encompassing operational and safety aspects, including unintentional islanding. The standard also covers the response of the interconnection system to abnormal conditions and presents real-world application examples, such as cutting-edge protection practices and advanced unintentional islanding methodologies. Adherence to the practices stipulated in the IEEE 1547.8 standard should catalyze the evolution of sophisticated hardware and software, facilitate their implementation acceptance, and ultimately contribute to heightened penetration levels of DER.

IEC 61850 Standard

The IEC 61850 standard delineates communication protocols pertinent to intelligent electronic devices stationed within and in between electrical substations. This specification encompasses aspects of device communication such as data modeling, messaging, and network architecture. The prescriptions within this standard are paramount to command and control systems employed across electricity generation, transmission, distribution, and consumption. It fortifies interoperability, reliability, and efficiency, concurrently enabling the incorporation of renewable energy sources and distributed energy resources into the grid. Additionally, it stipulates cybersecurity prerequisites pertinent to smart grids.

Communication protocols

The ADNs in modern power systems rely on various communication protocols to integrate and manage DERs. Protocols such as MODBUS and DNP3 are commonly used for industrial and utility-specific needs, while IEC 61850 and IEC 60870-5-104 are international standards for substation automation and telecontrol [36]. Lightweight protocols like MQTT and CoAP are emerging for IoT-based applications, and Zigbee is often used for home automation. OPC UA offers a robust, vendor-neutral option. The choice of protocol often depends on the specific needs of the application, such as data rate, latency, and security.

2.3 Optimal power flow

The optimal power flow problem (OPF) is a fundamental power system problem that consists of finding the optimal generation dispatch among a set of generators in a power system. The objective of the OPF is to minimize the total operating cost related to generation cost, system losses, and quality of service while satisfying a set of operational constraints, such as power demand, technical limits of voltage and capacity of transmission lines, and stability and security constraints. In turn, this problem is integrated as a subproblem in many standard applications of electrical engineering, such as economic dispatch, unit commitment, state estimation, stability evaluation, Volt/VAr control, demand response, etc.

By the nature of the problem, the OPF is computationally complex and has been proven to be NP–Hard in the general case, as demonstrated in [37] for a hardness proof based on a reduction from an instance of the optimal power flow to a complex quadratic optimization problem. However, practical applications of OPF require tractable and accurate representations of the problem. To this end, many works leverage solution algorithms and approximations, including linear programming (LP) approaches such as the conventional DC OPF [38], *LinDistFlow* [39], and first-order Taylor (FOT) approximations [40], which provide more accurate results than DC OPF. Other approaches apply semidefinite relaxations based on SOCP, proposed in [41] for radial networks, and semidefinite programming (SDP), proposed in [42] for general networks.

2.3.1 OPF formulations

To solve the OPF, it is necessary to have a model of the AC power flow equations that describes the steady state of the power system. In [43], two alternative models are proposed for this purpose: the bus injection model (BIM) and the branch flow model (BFM). We introduce the following definitions to analyze these models and provide some structural properties for both BIM and BFM. Considering that an electrical network can be represented through a directed graph of $\mathcal{N}^+$ nodes and $\mathcal{E}$ branches, such that $\mathcal{G} = (\mathcal{N}^+, \mathcal{E})$ and $\mathcal{E} \subseteq \mathcal{N}^+ \times \mathcal{N}^+$, it is possible to describe the interactions of the electrical system using the following physical expressions:

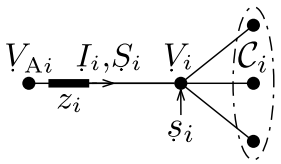


Figure 2.6: Notation summary for a directed network.

$$\underline{V}_{A_i} - \underline{V}_i = z_i \underline{I}_i, \qquad \forall\, i \in \mathcal{E} \tag{2.2a}$$

$$\underline{I}_i = \sum_{j \in \mathcal{C}_i} \underline{I}_j, \qquad \forall\, i \in \mathcal{N} \tag{2.2b}$$

$$\underline{s}_i = \underline{V}_i \underline{I}_i^{\mathsf{H}}, \qquad \forall\, i \in \mathcal{N} \tag{2.2c}$$

where (2.2a) corresponds to the application of Ohm's law on each line, while (2.2b) and (2.2c) define the current and power injection balances at each node. From these expressions, multiple definitions of sets, parameters, and variables around node i are derived, which allows us to define the operation of the network. The induced sets are denoted as $\mathcal{N}$ and $\mathcal{C}_i$, and correspond to the set of nodes of the system excluding the slack bus ($\mathcal{N} = \mathcal{N}^+ \backslash 0$), and the set of downstream neighboring members that node i possesses. On the other hand, in the expression (2.2a), the attribute z_i characterizes the branch connecting node i with the one seen upstream, whose physical interpretation underlies the line impedance. Similarly, in the expressions presented, two types of variables are used to define the operating state of the network: those associated with the nodes, such as the voltage at bus i ($\underline{V}_i$), the voltage seen at i–bus upstream ($\underline{V}_{A_i}$) and the complex power injected at bus i ($\underline{s}_i$); and those associated with each branch, these being the current ($\underline{I}_i$) and complex power ($\underline{S}_i$) flowing through the line connecting i–bus with the upstream bus. The explicit definition of the variables and parameters of the system is shown in Figure 2.6.

Bus injection model

Presenting as main drawback the Hermitian transpose, the canonical BIM poses the substitution of this operator in the expression (2.2c) as follows:

$$\dot{s}_i = \sum_{j \in \mathcal{N}_i} \dot{V}_i \left(\frac{\dot{V}_i - \dot{V}_j}{z_i} \right)^*, \qquad \forall\, i \in \mathcal{N} \tag{2.3}$$

where $\mathcal{N}_i$ is the vector of nodes neighboring node i, such that $\mathcal{N}_i = \mathcal{C}_i \cup \{A_i\}$. The reinterpretation used in the BIM model eliminates the complex transposition on the vector $\dot{I}_i$ to calculate the power injected in member i, replacing it with a summation over the line powers. The optimization model that defines the operation of the system is thus defined as follows:

$$\underset{\mathbf{x}}{\text{minimize}} \;\; f(\mathbf{x}), \tag{2.4a}$$

$$\text{subject to} \quad \underline{s}_i \leq \sum_{j \in \mathcal{N}_i} \dot{V}_i \left(\frac{\dot{V}_i - \dot{V}_j}{z_i} \right)^* \leq \bar{s}_i, \qquad \forall i \in \mathcal{N}^+ \tag{2.4b}$$

$$\underline{V}_i \leq |\dot{V}_i| \leq \bar{V}_i, \qquad \forall i \in \mathcal{N}^+ \tag{2.4c}$$

where $f(\cdot)$ is the cost function, expressed as a function of the vector of decision variables $\mathbf{x} = [\mathbf{V}]^\top$. The parameters $\underline{s}$ and $\bar{s}$ represent the power injection limits at each bus, while $\underline{\mathbf{V}}$ and $\bar{\mathbf{V}}$ represent the vector of limits on the magnitude of the voltage.

Branch flow model

Based on the characteristics of a directed graph, the BFM reformulates the expressions (2.2) by injecting power into a simplified equivalent, as follows:

$$\sum_{j \in \mathcal{C}_i} \dot{S}_j = \dot{s}_i + \dot{S}_i - z_i |\dot{I}_i|^2, \qquad \forall i \in \mathcal{N} \tag{2.5a}$$

$$\dot{V}_{A_i} - \dot{V}_i = z_i \dot{I}_i, \qquad \forall i \in \mathcal{E} \tag{2.5b}$$

$$\dot{S}_i = \dot{V}_{A_i} \dot{I}_i^*. \qquad \forall i \in \mathcal{E} \tag{2.5c}$$

In contrast to the approach used in (2.2), (2.5) relies on line flows to define the other state variables that support the operation of the system. This allows to rewrite the operation problem as follows:

$$\underset{\mathbf{x}}{\text{minimize}} \;\; f(\mathbf{x}), \tag{2.6a}$$

$$\text{subject to} \quad \text{(2.5a)}, \text{(2.5b)}, \text{(2.5c)}, \tag{2.6b}$$

$$\underline{s}_i \leq \dot{s}_i \leq \bar{s}_i, \qquad \forall i \in \mathcal{N}^+ \tag{2.6c}$$

$$\underline{V}_i \leq |\dot{V}_i| \leq \bar{V}_i, \qquad \forall i \in \mathcal{N}^+ \tag{2.6d}$$

where the vector of decision variables $\mathbf{x} = [\dot{\mathbf{S}}, \dot{\mathbf{I}}, \dot{\mathbf{V}}, \dot{\mathbf{s}}]^\top$. In addition, the cost function $f(\cdot)$ is written in terms of the decision variables of the optimization problem.

Equivalence between BIM and BFM

Although the BIM and the BFM are defined by different sets of equations in terms of their own variables, both are models that support their formulation with Kirchhoff's law for the same electrical network. Thus, it is posited that these power grid modelings are equivalent if there is a bijection between their solution sets. This relationship is demonstrated in [44], where the authors formally establish the procedure that links the expressions of both models in both directions. Roughly speaking, if we consider the definitions (2.5b) and (2.5c) provided by the BIM, it is possible to manipulate them to reconstruct the expression (2.3), and thus obtain the BFM. Conversely, by eliminating $\dot{I}_i$ and $\dot{S}_i$ in (2.5) we obtain the BIM from the BFM.

2.3.2 OPF convex relaxations

The use of convex relaxations to solve OPFs offers significant advantages. By obtaining a feasible solution in the convex domain, it represents a global optimum for the OPF. If the solution is infeasible, it establishes an infeasible lower bound for the original domain [45]. This duality leads to consider the accuracy, efficiency and stability of these methods. High–quality convex relaxations are desirable because the OPF appears as a sub–problem of more complex optimal control problems [3]. For this reason, recent attention has been given to SDP and SOCP relaxations.

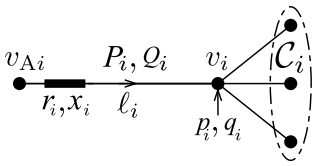


Figure 2.7: Notation summary for a directed network with convex relaxation.

The SOCP relaxations are practical for OPF, providing high accuracy especially in radial networks [46] using both BIM and BFM methods, with BFM being more stable [43]. However, SOCP does not guarantee accuracy when voltage upper limits are linked to optimality [47], which poses problems in high solar radiation scenarios where voltage increases with power injections. For this reason we chose linear models for the development of this proposal. However, the convex relaxation of the BFM that supports the development of one of the linearized models used in the proposal is developed below.

$$\sum_{k \in \mathcal{C}_i} P_k = p_i + P_i - r_i \ell_i, \qquad \forall i \in \mathcal{N} \tag{2.7a}$$

$$\sum_{k \in \mathcal{C}_i} Q_k = q_i + Q_i - x_i \ell_i, \qquad \forall i \in \mathcal{N} \tag{2.7b}$$

$$v_i = v_{A_i} - 2r_i P_i - 2x_i Q_i - \left(r_i^2 + x_i^2\right) \ell_i, \qquad \forall i \in \mathcal{E} \tag{2.7c}$$

$$v_{A_i} \ell_i \geq P_i^2 + Q_i^2. \qquad \forall i \in \mathcal{E} \tag{2.7d}$$

Known as *DistFlow* [48], this formulation uses as auxiliary variables the quadratic magnitudes of voltage ($v_i = |V_i|^2$) and current ($\ell_i = |I_i|^2$) to efficiently decompose the power flows per line and power injection per bus into their active and reactive parts. Therefore, by employing the quadratic variables it is possible to decompose the power balance expression (2.5a) into its active and reactive part, described in (2.7a) and (2.7b) respectively. In the same way, Ohm's law applied on the line impedance is rewritten through the relationships of active power with the resistance, reactive power with the reactance and quadratic current with the impedance, as shown in (2.7c). Finally, the expression (2.5c) defining the power flows is convexized by means of the rotated Lorentz cone, giving rise to the expression (2.7d). The development of this procedure is presented explicitly in [49]. The convexification of the solution space thus brings with it new notations of variables and parameters which are presented in Figure 2.7. As a result of this relaxation, the SOCP optimization problem seen below is obtained:

$$\underset{\mathbf{x}}{\text{minimize}} \;\; f(\mathbf{x}) \tag{2.8a}$$

$$\text{subject to} \quad \text{(2.7a)}, \text{(2.7b)}, \text{(2.7c)}, \text{(2.7d)}, \tag{2.8b}$$

$$P_i^2 + Q_i^2 \leq \bar{S}_i^2, \qquad \forall i \in \mathcal{E} \tag{2.8c}$$

$$\underline{V}_i^2 \leq v_i \leq \bar{V}_i^2, \qquad \forall i \in \mathcal{N}^+ \tag{2.8d}$$

where, instead of limiting the power injected into each bus, the power flowing through the lines is limited with its maximum admissible value, denoted as $\bar{S}_i$. While working with the quadratic magnitude of the voltage as a decision variable, the constraint (2.8d) uses the nominal voltage limits in a quadratic way. Furthermore, as in the previous models, the objective function $f(\cdot)$ still captures the operating costs in terms of the decision variables $\mathbf{x} = [\mathbf{P}, \mathbf{Q}, \boldsymbol{\ell}, \mathbf{v}, \mathbf{p}, \mathbf{q}]^\top$.

2.3.3 Linear models for radial networks

Generalizing the conception of linear models in distribution networks, it is introduced and considered that $\mathcal{F} : \mathbb{R}^{2n} \to \mathbb{R}^n$ such that the solution of

$$\mathbf{v} = \mathcal{F}(\mathbf{p}, \mathbf{q}), \tag{2.9}$$

where for each pair $(\mathbf{p}, \mathbf{q})$ there exists a single $\mathbf{v}$ satisfying the conditions (2.3). This allows us to assume that the existence and uniqueness conditions are satisfied and we focus on the construction of linear approximations of (2.9) for radial networks, such that

$$\mathbf{v} = \mathbf{B}\mathbf{u} + \mathbf{m}, \tag{2.10}$$

where $\mathbf{u} \coloneqq [\,\mathbf{p}, \mathbf{q}\,]^\top$ and the column vector $\mathbf{m}$ represents an intercept.

LinDistFlow (LDF)

Being the linearization of formulation (2.7), the *LinDistFlow* model [39] considers line currents negligible, which redefines the quadratic BFM as:

$$\sum_{k \in \mathcal{C}_i} P_k = p_i + P_i, \qquad \forall i \in \mathcal{N} \tag{2.11a}$$

$$\sum_{k \in \mathcal{C}_i} Q_k = q_i + Q_i, \qquad \forall i \in \mathcal{N} \tag{2.11b}$$

$$v_i = v_{A_i} - 2r_i P_i - 2x_i Q_i. \qquad \forall i \in \mathcal{E} \tag{2.11c}$$

By eliminating the current ratios, the resulting formulation consists only of linear active and reactive power balance ratios at the nodes and the voltage drop across the feeder. This suggests that (2.11) has a compact formulation. Let $\hat{\mathbf{A}} \coloneqq [\,\mathbf{a_0}\ \mathbf{A}\,] \in \mathbb{R}^{|\mathcal{N}| \times (|\mathcal{N}|+1)}$ be the element to node incidence matrix, where a_0 is its first column, and $\mathbf{A} \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$ the bus incidence matrix. Therefore

$$\mathbf{p} = \mathbf{A}^\top \mathbf{P}, \tag{2.12a}$$

$$\mathbf{q} = \mathbf{A}^\top \mathbf{Q}, \tag{2.12b}$$

$$\mathbf{A}\mathbf{v} + v_0 \mathbf{a}_0 = \operatorname{diag}(\boldsymbol{r})\mathbf{P} + \operatorname{diag}(\boldsymbol{x})\mathbf{Q}. \tag{2.12c}$$

The variable v_0 corresponds to the quadratic magnitude of the utility voltage. Taken to the functional form described in (2.10), we define $\mathbf{F} \coloneqq \mathbf{A}^{-1}$, such that we obtain:

$$\mathbf{F}\mathbf{a_0} + \mathbf{1} = \mathbf{0}. \tag{2.13}$$

Using this definition the problem can be rewritten as:

$$\boldsymbol{v} = v_0 \mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q}, \tag{2.14}$$

where matrices $\mathbf{R} \coloneqq \mathbf{F}\operatorname{diag}(\boldsymbol{r})\mathbf{F}^\top$ and $\mathbf{X} \coloneqq \mathbf{F}\operatorname{diag}(\boldsymbol{x})\mathbf{F}^\top$ are symmetric positive definite and have positive entries [50]. Then we write (2.14) in compact matrix form becoming

$$\mathbf{v} = \begin{bmatrix} \mathbf{R} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} + v_0 \mathbf{1}. \tag{2.15}$$

Now set $\mathbf{B} = [\,\mathbf{R}, \mathbf{X}\,]$ and $\mathbf{m} = v_0 \mathbf{1}$ and we obtain a linear model as (2.12).

2.4 Artificial neural network

The ANNs are computational models inspired by neurons in the brain, designed to recognize patterns, make decisions and solve complex problems. Composed of interconnected neurons, ANNs process information using statistical and mathematical methods. Neurons receive inputs, produce outputs encoded by nonlinear activation functions, and transmit them to other neurons, as shown in Figure 2.8. The strength of the connection between neurons, known as the weight, determines the degree of influence of one neuron on another and is adjusted during learning to improve accuracy on the designated task. In recent decades, neural networks have become vital in machine learning, excelling in the fields of image processing [51, 52] and language [53, 54], and, more recently, in the modeling and simulation of physical systems, thanks to its ability to handle complex, high-dimensional data, common in science and engineering.

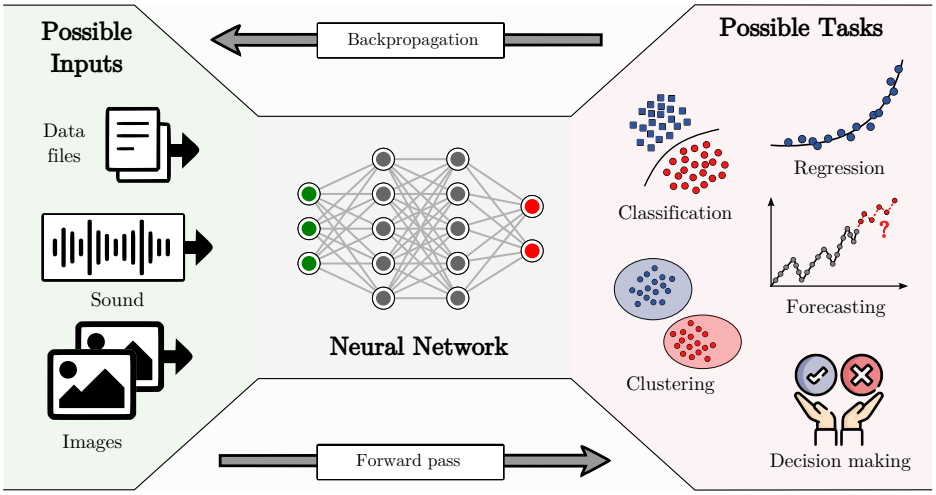


Figure 2.8: Conceptualization of a ANN. On the left are the possible input sources, in the center the ANN composed of non-linear relationships between layers, and on the right the possible tasks it can perform.

In recent years, PINNs have been a breakthrough in solving nonlinear systems, especially those governed by partial differential equations (PDEs) [9, 55, 56]. PINNs integrate the physical laws governing over the system directly into the learning process of the neural network by incorporating them into the loss function of the network. This allows PINNs to solve both forward and inverse problems efficiently, even in cases where traditional numerical methods may struggle due to high dimensionality or the need for fine discretization [57,58]. Moreover, the combination of PINNs with optimization frameworks, such as the KKT conditions, enables a more comprehensive approach to solving constrained optimization problems [59–65]. By incorporating KKT conditions into the training process, PINNs guarantee optimality conditions on neural network performance, providing a robust tool for tackling problems with high dimensionality and low uncertainty.

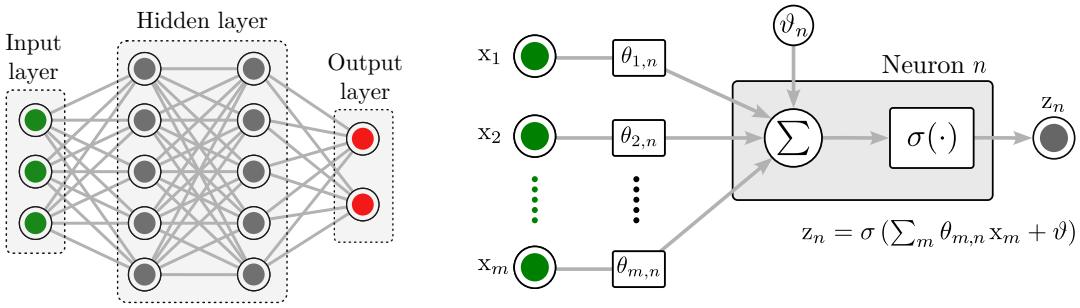
While PINNs offer significant advantages over traditional optimization methods, especially in handling nonlinear systems and high-dimensional problems, they also come with certain disadvantages. One of the key benefits of PINNs is their ability to integrate uncertainty directly into the modeling process, allowing for more flexible and adaptive solutions under varying conditions [66]. This is particularly valuable in fields like energy management, where optimal operation must account for uncertainties such as fluctuating demand and supply. However, neural networks, including PINNs, often require substantial computational resources and large datasets for effective training, which can be a limitation in practice [67]. Moreover, in general terms, the interpretability of neural network models is often less intuitive compared to classical optimization methods, which are often more straightforward in terms of their mathematical formulation and solution space [68,69]. Given this background, one of the objectives of this proposal is the development of an intelligent agent based on ANN that integrates physical components for its interpretability in its conception and action, in addition to being efficient and robust in the face of system uncertainty.

2.4.1 Artificial neural network structures

ANNs are versatile computational models that can be configured in various ways depending on the specific requirements of a given application. The structure of these networks directly impacts their performance and suitability for different tasks, such as image processing, time-series prediction, among others. In this section, we will provide a comprehensive overview of three widely used neural network architectures: dense neural networks (DNN), also known as fully connected networks; convolutional neural networks (CNNs), which are particularly effective in image and spatial data analysis; and temporal convolutional networks (TCNs), specialized for sequence modeling and temporal data. Knowing the characteristics and applications of these architectures will allow us to discern their contributions to the design of the agent used in this proposal.

Dense neural network

Being the most fundamental type of neural network architecture, DNNs have a design and conception inspired by the functioning of the human brain, seeking to emulate the interaction between biological neurons to perform a given task [70]. Its functional structure is organized by stacking layers composed of neurons, where each neuron of a layer is connected to all the neurons of the previous and following layers, as shown in Figure 2.9.



(a) Structure (b) Forward pass calculation

Figure 2.9: Dense neural network conceptualization.

Thus the computation of the output signal of any n neuron is defined by the weighted signals received from the previous layer together with the nonlinear activation that defines the neuron's performance. The generalization of this procedure at the layer level is described below:

$$\mathbf{z} = \sigma(\boldsymbol{\theta} \cdot \mathbf{x} + \boldsymbol{\vartheta}^\top), \tag{2.16}$$

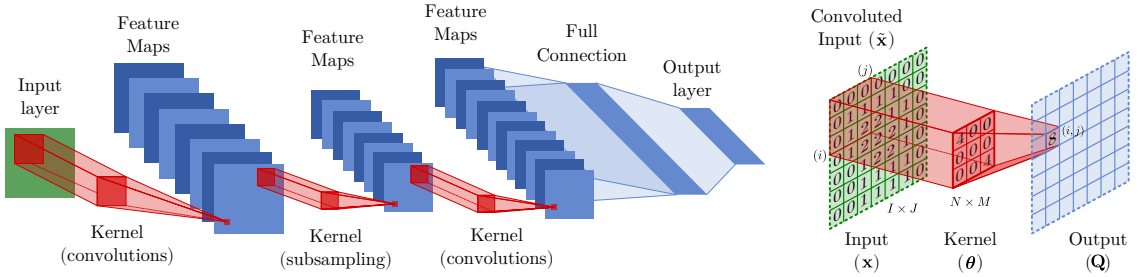
where $\boldsymbol{\theta}$ represents the matrix of interconnection weights between the two layers, $\boldsymbol{\vartheta}$ denotes the bias vector of the preceding layer; $\sigma(\cdot)$ the nonlinear activation function; $\mathbf{x}$ and $\mathbf{z}$ are the input and output vectors associated with the current layer, respectively. This interconnected structure of nonlinear activations allows DNNs to learn complex data patterns by adjusting the weights between layers, similar to how neurons in the brain strengthen or weaken their connections in response to stimuli. In machine learning, the process associated with adjusting weights is known as error backpropagation–based training, which traditionally uses stochastic gradient descent to formulate:

$$\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\vartheta} \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\vartheta} \end{bmatrix} - \eta \begin{bmatrix} \mathbf{x}^\top \nabla_{\mathbf{z}} f^{\text{loss}}(\mathbf{z}) \odot \sigma'(\mathbf{z}) \\ \nabla_{\mathbf{z}} f^{\text{loss}}(\mathbf{z}) \odot \sigma'(\mathbf{z}) \end{bmatrix}, \tag{2.17}$$

where $f^{\text{loss}}(\cdot)$ represents a generic loss function, and η is learning rate. This training method is not limited to DNNs alone, but is a generalized approach applicable to virtually any neural network architecture [71]. This versatility underscores the robustness and adaptability of back–propagation as a fundamental technique in deep learning, making it a cornerstone of modern ANN training in a wide range of applications.

Convolutional neural network

CNNs are a class of ANNs designed on the basis of the matrix convolution principle, whose application has been extended to handle data with spatial hierarchies, such as images and videos [72]. Unlike DNNs, CNNs use kernels associated with channels to perform matrix convolutions on the input data. These channels are usually several, such that their kernels are designed and trained to identify different types of latent patterns and features [73]. The mathematical procedure behind this type of architecture is shown in Figure 2.10.



(a) Structure

(b) Convolution process

Figure 2.10: Convolutional neural network conceptualization.

Using the graphical scheme in Figure 2.10, for any indexing (i, j), the matrix $\mathbf{Q}$ resulting from the convolution between the input data submatrix $\tilde{\mathbf{x}}$ and the kernel interconnecting two layers $\boldsymbol{\theta}$ is:

$$\mathbf{Q}_{M,N} = \underbrace{\begin{bmatrix} \mathrm{x}_{i,j} & \mathrm{x}_{i,j+1} & \cdots & \mathrm{x}_{i,j+N} \\ \mathrm{x}_{i+1,j} & \mathrm{x}_{i+1,j+1} & \cdots & \mathrm{x}_{i+1,j+N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x}_{i+M,j} & \mathrm{x}_{i+M,j+1} & \cdots & \mathrm{x}_{i+M,j+N} \end{bmatrix}}_{\tilde{\mathbf{x}}} \odot \underbrace{\begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,N} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{M,1} & \theta_{M,2} & \cdots & \theta_{M,N} \end{bmatrix}}_{\boldsymbol{\theta}}. \tag{2.18}$$

Note that, in order to unify the notation with the definition of the different topologies, the paper will use the variable $\boldsymbol{\theta}$ to define the adjustable weights interconnecting two layers.

The convolutional matrix is a sub–matrix of the input matrix $\mathbf{x}$ such that $\tilde{\mathbf{x}} \subset \mathbf{x}$. In this sense, the dimensions of the input matrix $\mathbf{x}$ are arbitrary, being represented by $I \times J$. The dimension of this sub-.matrix depends on the dimension given for the kernel, being $N \times M$ for the presented case. Similarly, the resulting matrix dimension depends on the handling of the boundary conditions of the sub–matrix, which are usually associated with the padding or not of the convolution matrix [74]. Under these guidelines, the performance of a CNN layer can be understood as:

$$\mathrm{z}_{i,j} = \sigma\left(\sum_{m}^{M}\sum_{n}^{N} \mathbf{Q}_{m,n} + \vartheta\right), \tag{2.19}$$

where ϑ is the bias of the interconnection of between two layers. The result of the convolution, in this case two-dimensional, complying with the dimensions of the input matrix. This allows us to establish that the structure of the resulting vector is given by:

$$\mathbf{z} = \begin{bmatrix} \mathrm{z}_{1,1} & \cdots & \mathrm{z}_{1,j} & \cdots & \mathrm{z}_{1,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathrm{z}_{i,1} & \cdots & \mathrm{z}_{i,j} & \cdots & \mathrm{z}_{i,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathrm{z}_{M,1} & \cdots & \mathrm{z}_{M,j} & \cdots & \mathrm{z}_{M,N} \end{bmatrix}, \tag{2.20}$$

It should be noted that in a CNN structure, the procedure described in (2.19) is sequential in nature, since as the structure is deepened, the data processing uses the results of the previous layer. Moreover, that for the case under study, the dimension of $\mathbf{z}$ is worked to be equal to $\mathbf{x}$, but with a proper manipulation of the matrix operations that compose the expression (2.18), reductions or increases in dimensionality of the output sample of each layer can be realized, which is widely used for decision making and training strategy designs of this type of ANN [75].

Temporal convolutional network

Based on the principles underpinning CNNs, the TCNs offer a robust alternative to recurrent neural networks (RNNs) for sequence modeling, using one-dimensional convolutions to effectively capture sequential dependencies [76]. Unlike RNNs, which process data sequentially and are prone to problems with information forgetting due to vanishing gradients, TCNs handle sequences in parallel, improving both training efficiency and model performance [77]. Parallel processing is achieved by means of causal convolutions supported by dilation's performed on the input data, as exemplified in Figure 2.11.

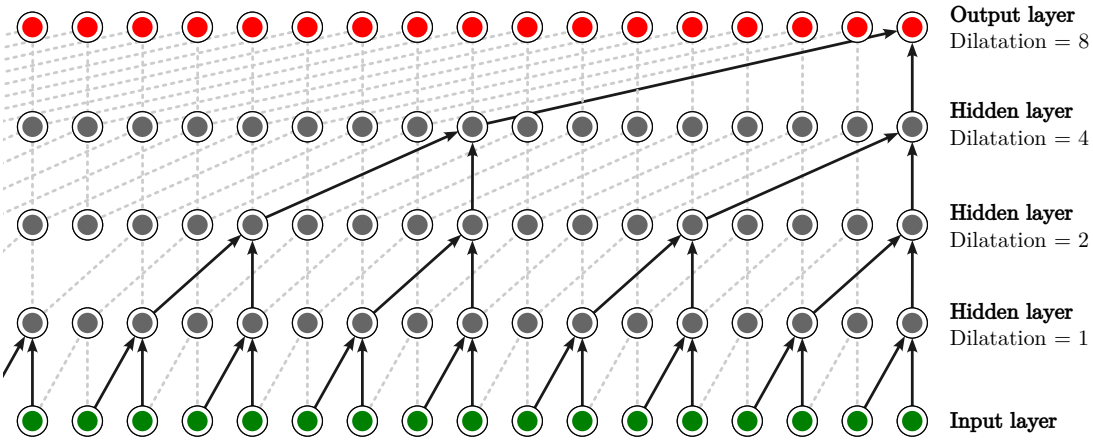


Figure 2.11: Temporal convolutional network conceptualization [76].

The key to the effectiveness of TCNs lies in their use of causal convolutions, which ensure that the output at any time the output of each layer is influenced only by past and present inputs. This causal relationship can be formally expressed using the transform exposed for the definition of $\mathbf{T}$,

$$\mathbf{T} = \begin{bmatrix} \mathrm{x}_1 & \mathrm{x}_2 & \dots & \mathrm{x}_N \\ \mathrm{x}_2 & \mathrm{x}_3 & \dots & \mathrm{x}_{N+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x}_{M-N} & \mathrm{x}_{M-N+1} & \dots & x\mathrm{x}_M \end{bmatrix}, \tag{2.21}$$

such that the computation of a convolutional layer is defined as,

$$\mathbf{z} = \sigma\left(\mathbf{T} \cdot \boldsymbol{\theta} + \boldsymbol{\vartheta}\right), \tag{2.22}$$

where $\mathbf{x}$ is the input vector defining the context window of dimensions $M \times 1$, while $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ are the one-dimensional kernel and bias of a channel interconnecting two layers of a TCN. In this matrix representation, $\mathbf{T}$ captures the causal nature in which future data do not influence past outputs. This translates to $\mathbf{T}$ being of $(M - N) \times N$ dimensions. The representation (2.22) corresponds to the computation of only one of the layers composing the TCN structure, with this procedure being repeated on the processed vector of the previous layer as the depth of the TCN structure is advanced.

Enhancing this configuration, dilation in the causal convolutions introduces jumps or gaps in the filter application, as seen below:

$$\mathbf{T}_d = \begin{bmatrix} \mathrm{x}_1 & \mathrm{x}_{1+d} & \cdots & \mathrm{x}_{N\cdot d} \\ \mathrm{x}_2 & \mathrm{x}_{2+d} & \cdots & \mathrm{x}_{1+N\cdot d} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x}_{M-N\cdot d} & \mathrm{x}_{M-(N-1)d} & \cdots & \mathrm{x}_M \end{bmatrix}. \tag{2.23}$$

This alteration expands the convolutional kernel's reach across the input vector without increasing the number of parameters, thereby allowing the network to capture longer-range dependencies more effectively [78]. The dilation factor d, increases as one advances in the depth of the structure, such that it follows a dilation basis D as seen in Figure 2.11. The combination of causal structure and dilation in TCNs thus balances computational efficiency with a deep temporal understanding, essential for complex sequence analyses.

2.4.2 Advanced techniques in neural network training

Training a neural network is the foundational process by which a model learns to map inputs to desired outputs by adjusting its internal parameters, usually weights and biases, through iterative optimization. The training process consists of feeding the model's performance to the design objectives, using for this purpose iterative algorithms such as error back–propagation. This error, measured by a loss function, quantifies the distance between model predictions and reality. In this sense, the definition of this loss function is paramount to the performance of the model once trained, as it directly influences how well the model captures the underlying patterns in the data [79]. A well-chosen loss function ensures that the model optimizes towards the most relevant features and objectives, while an inadequate one may lead to poor generalization or convergence to suboptimal solutions.

Several advanced methodologies have emerged in the recent literature to improve traditional neural network training. Physics-informed machine learning (PIMLs), for example, integrate physical laws-such as differential equations-directly into the solving of the machine learning algorithm [80], improving model accuracy by guiding the learning process toward physically consistent and meaningful solutions [9]. PIMLs ensure that predictions remain based on the physical principles governing the system, which is critical for applications in scientific computing and engineering simulations. In addition, optimization-based techniques, such as regularization mechanisms and hot-start transfer learning, offer effective ways to train neural networks. Regularization mechanisms promote the principle of parsimony for the trained model [81]. While transfer learning, combined with warm bootstrapping, accelerates learning by leveraging knowledge from previously trained models, which improves performance on new but related tasks [82]. These approaches significantly improve training efficiency, especially in complex or data-limited scenarios, enabling faster and more robust convergence in the face of uncertainty.

Physics-informed machine learning

Physical science problems often involve data-intensive tasks such as spatiotemporal data modeling, causal reasoning, computer vision, and probabilistic inference. ML has shown significant success in addressing these challenges, leading to increased interest in applying ML to advance scientific discovery in physics [83]. Physics-informed machine learning (PIML) combines the predictive power of ML with the explanatory strength of physics by integrating knowledge of the underlying physical processes into model design and parameter setting. This approach offers greater flexibility, generalizability, and computational efficiency compared to traditional numerical or purely physics-based methods, while maintaining physical plausibility [84]. By embedding physical laws into the learning process, PIML enhances interpretability, robustness, and generalization, enabling more accurate predictions and providing deeper insights into the physical mechanisms driving the system.

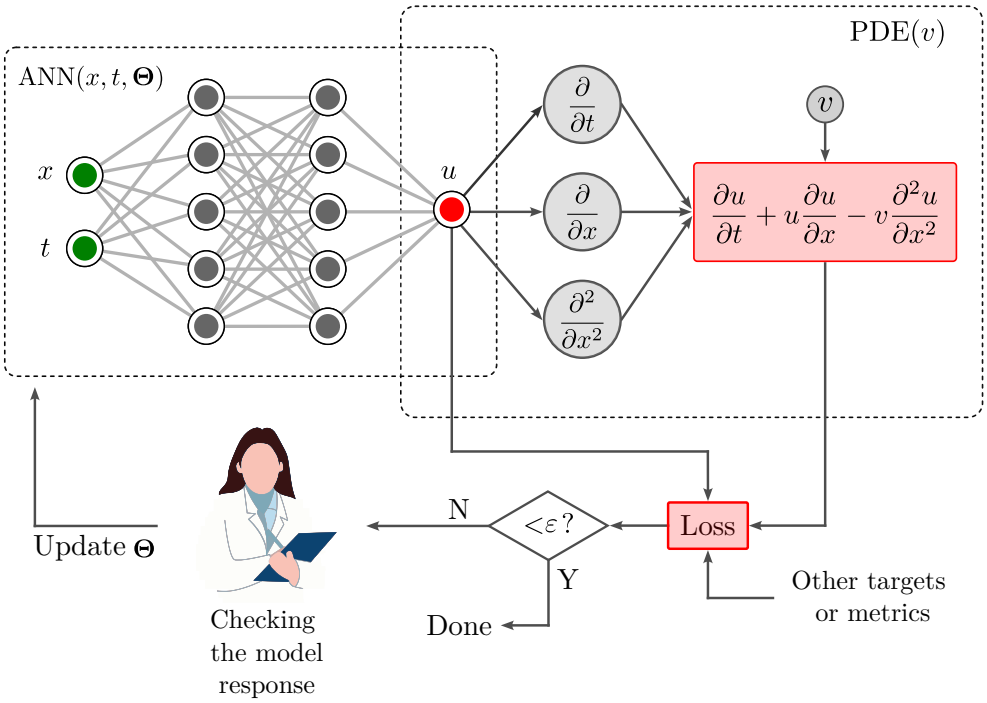


Figure 2.12: Physics-informed neural networks [9].

Building on the strengths of PIML, PINNs extend this concept by integrating the physics of the operating environment directly into ANNs [55]. PINNs incorporate physical laws, often expressed as PDEs, into the network's architecture by embedding these equations into the loss function during training [56]. This ensures that the learned solution not only minimizes prediction error but also adheres to the known physics of the system, as depicted in Figure 2.12. By leveraging the underlying physical model, PINNs can operate effectively with limited or even unsupervised data, reducing the need for large datasets [9]. This makes them especially useful in fields where system dynamics are well understood and can be mathematically modeled, such as engineering, fluid dynamics, and climate science. As a key extension of the PIML framework, PINNs combine the flexibility of neural networks with the explanatory power of physics, offering a powerful tool for tackling complex scientific and engineering problems.
There are two main methods for integrating physics principles into ANNs. These approaches differ in how they incorporate physics into the model, either through the design of the network itself or by incorporating expressions that complement the training process. These two key strategies for integrating physics into ANNs are described below.

- **Integration of physics into the ANN design.** This method integrates physics principles directly into the ANN architecture, customizing it for specific applications. For example, studies such as [85–88] focus on incorporating time- and space-based dynamics within the neural network structure. By taking into account how the system behaves over time and in different spatial dimensions, these designs ensure that the network better matches the physical behavior of the system in the real world. This approach improves the accuracy and predictive capability of the model by creating a more physically representative framework.

- **Integration of physics in training through the loss function.** The second approach integrates physical laws into the training process through the loss function. By generating complementary expressions, the physical delineations governing the behavior of the system are introduced. Traditionally, this method is linked to systems whose physical description can be realized with the PDEs [55],, where terms representing the residuals of the differential equations are added to the loss function to ensure that the model predictions comply with the known physical laws [56], as seen in Figure 2.12. Recent literature has extended this approach, linking it to optimization problems [89]. In particular, through the use of KKT

conditions, optimization concepts such as constraints and objectives have been adapted in the ANN training process, thus offering a new working paradigm to address the efficient operation of systems under constraints [10,90,91]. Starting from the generalized formulation presented in [92] for the KKT conditions, the following describes the reinterpretation of these conditions for their use in the training of an ANN. Let be the following optimization problem:

$$\underset{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}), \tag{2.24a}$$

$$\text{subject to} \quad g_i(\mathbf{x}) = 0, \qquad \forall i = 1,\dots,m \tag{2.24b}$$

$$h_j(\mathbf{x}) \leq 0, \qquad \forall j = 1,\dots,n \tag{2.24c}$$

where the decision variable $\mathbf{x}$ is contained in the feasible solution region $\mathcal{X} \subseteq \mathbb{R}^n$, while $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ correspond to the objective function, equality conditions and inequality conditions, respectively. If these functions are considered as differentiable, the Lagrangian defining the optimality conditions is:

$$f_\nabla(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \nabla_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i \nabla_{\mathbf{x}} g_i(\mathbf{x}) + \sum_{i=1}^{n} \nu_i \nabla_{\mathbf{x}} h_i(\mathbf{x}), \tag{2.25}$$

where the vectors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are the dual variables associated with the equality and inequality constraints, respectively. Under this set of definitions, the following optimality conditions are established:

$$g_i(\mathbf{x}^*) = 0, \qquad i = 1,\dots,m \tag{2.26a}$$

$$h_i(\mathbf{x}^*) \leq 0, \qquad i = 1,\dots,n \tag{2.26b}$$

$$\nu_i \geq 0, \qquad i = 1,\dots,n \tag{2.26c}$$

$$\nu_i h_i(\mathbf{x}^*) = 0, \qquad i = 1,\dots,n \tag{2.26d}$$

$$\nabla_{\mathbf{x}} f(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i^* \nabla_{\mathbf{x}} g_i(\mathbf{x}^*) + \sum_{i=1}^{n} \mu_i^* \nabla_{\mathbf{x}} h_i(\mathbf{x}^*) = 0. \tag{2.26e}$$

The set of expressions (2.26) corresponds to the KKT conditions, where $\mathbf{x}^*$ is the optimal value that the decision variables can take. The expressions (2.26a) and (2.26b) represent the primal conditions of the problem, and evaluate that the solution is feasible given the original problem constraints. While expressions (2.26c) and (2.26d) ensure the dual feasibility and complementary slack of the Lagrange multipliers associated with the inequality constraints. The former restricts the Lagrange multipliers to strictly positive values, while the latter sets their values to zero if the constraint is active. Finally, the expression (2.26a) ensures that the gradient of the Lagrangian function with respect to the decision variables is zero. Adapting these functional forms, the authors of [89] propose the following reformulation for integration into the loss function:

$$f^{ss}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\nu}) = \sum_{x\in\mathbf{x}} \left| \frac{\partial}{\partial x} f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i \frac{\partial}{\partial x} g_i(\mathbf{x}) + \sum_{i=1}^{n} \nu_i \frac{\partial}{\partial x} h_i(\mathbf{x}) \right|, \tag{2.27a}$$

$$f^{slk}(\mathbf{x}, \boldsymbol{\nu}) = \sum_{i=1}^{n} \left| \nu_i h_i(\mathbf{x}) \right|, \tag{2.27b}$$

$$f^{prim}(\mathbf{x}) = \sum_{i=1}^{m} \max\{0, g_i(\mathbf{x})\} + \sum_{i=1}^{n} \left| h_i(\mathbf{x}) \right|. \tag{2.27c}$$

This reinterpretation is based on the minimization of the residuals generated by the expressions defending the KKT conditions. In the case where the condition is strictly equal to zero, we seek to minimize the absolute value, while in the cases of inequality, we penalize when this is not satisfied. This allows a more structured and mathematically grounded adjustment of the loss function, making the training process more aligned with the specific needs of the system under analysis.

These two approaches often complement each other and are widely used to improve the efficiency and accuracy of neural network training [93]. Proof of this is provided in [94], where the quantitative improvements generated by the integration of physics into ANNs are exemplified. In the example developed, shown in Figure 2.13, the differences between conventional training and PINNs are depicted. In the former case, neural networks can produce accurate solutions with generalization problems due to overfitting to the available data. In contrast, PINNs are able to integrate the physical laws governing the system into their performance, significantly improving the accuracy of predictions and effectively managing complex systems. This approach shifts the paradigm from viewing neural networks as black boxes to a more transparent process based on formalized mathematical principles.

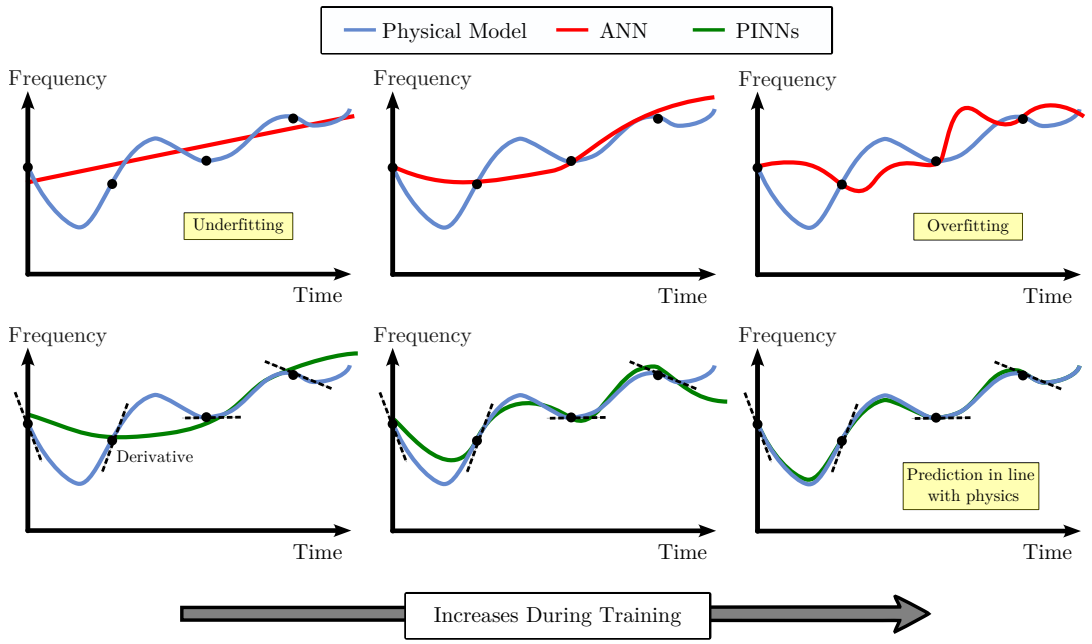


Figure 2.13: Comparison between ANNs and PINNs in the training process [94].

Regularization

Regularization is a crucial technique in statistical modeling and machine learning, grounded in the principle of parsimony, which advocates for simpler models with fewer parameters [95]. According to this principle, models with fewer parameters are favored when they achieve similar predictive accuracy, as they tend to be more interpretable, less prone to overfitting, and more likely to generalize well to new data. In complex models like neural networks, regularization is essential for controlling model complexity by penalizing large or unnecessary parameters, allowing the model to capture only the most relevant patterns in the data [96]. Classical regularization methods include Lasso and Ridge regularization, both of which impose penalties on the magnitudes of the model's parameters. Lasso regularization, or L_1–regularization, imposes a penalty on the absolute values of the parameters, which encourages sparsity by driving some weights to exactly zero, effectively selecting a subset of features. The Lasso loss function is expressed as:

$$f^{L_1}(\mathbf{z}, \boldsymbol{\Theta}) = f^{\text{loss}}(\mathbf{z}) + \lambda_{\text{L}} \sum_{\theta \in \boldsymbol{\Theta}} |\theta| \tag{2.28}$$

where λ_{L} represents the penalty factor function, while $\boldsymbol{\Theta}$ represents the set of trainable weights of an ANN. The regularization term penalizes the absolute magnitude of the weights, causing the model to zero out smaller weights while keeping larger, more impactful weights. In contrast, Ridge regularization, or L_2–regularization, adds a penalty proportional to the square of the parameters, which shrinks large weights but generally retains all features.

The Ridge loss function is defined as:

$$f^{L_2}(\mathbf{z}, \mathbf{\Theta}) = f^{\text{loss}}(\mathbf{z}) + \lambda_{\text{L}} \sum_{\theta \in \mathbf{\Theta}} \theta^2 \tag{2.29}$$

The regularization term penalizes larger weights and controls the strength of the penalty. By shrinking large weights toward zero, Ridge regularization reduces model complexity without enforcing sparsity, as all weights are reduced but not eliminated. In ANN, Ridge and Lasso regularization serve as effective weight penalization techniques to control model complexity and enhance generalization.

Transfer learning for warm-starting

Transfer learning is a technique in machine learning where a model trained on a large dataset for one specific task is adapted to solve a different but related task [97]. Instead of starting the learning process from scratch, the pre-trained model leverages the knowledge it gained from the source task, which allows for faster training and reduces the amount of data required for the new task. This is accomplished by fine-tuning only specific layers of the model, while the rest of the network retains the information from the initial training [98]. Transfer learning is especially useful in scenarios where large labeled datasets are not readily available for the target task or where computational resources are constrained. In Reinforcement Learning, for example, transferable knowledge can include expert policies, value functions, or learned dynamics, all of which can be adapted to improve performance in new tasks [99]. The process of transferring knowledge must be carefully aligned with the differences between the source and target domains to ensure successful adaptation and enhanced efficiency [100].

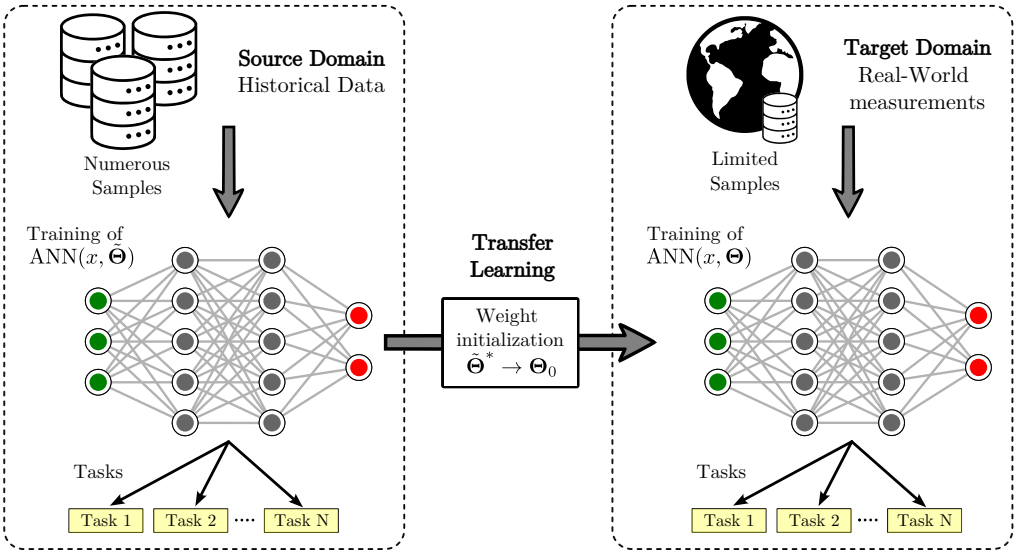


Figure 2.14: Conceptualization of transfer of learning for warm start.

A key application of transfer learning is its role in warm-starting models that operate in online, real-world environments, where models need to continuously adapt to new data without retraining from scratch, as depicted in Figure 2.14. In these scenarios, transfer learning enables models to quickly adjust to evolving conditions by fine-tuning only a subset of parameters while retaining most of the pre-learned knowledge [82, 101]. By providing a warm start, transfer learning allows models to stay responsive, reduce downtime, and enhance sample efficiency compared to traditional learning approaches [102, 103]. This seamless integration of new information ensures that models maintain high performance while adapting to changes, making transfer learning an essential tool for real-time, continuously evolving applications.

Chapter 3

Methodology

This chapter focuses on the characterization of the problem addressed together with the proposed solution approach. First of all, the framework considered is presented as a starting point, demarcated by the interconnection between the different layers of operation that make up the cyber-structure that supports the operation of the ADN. Through the presentation of this cyber-structure, the elements that make up each of the operating layers will be described and modeled in detail, along with their linkage to the operating environment. This includes the presentation of the modeling of the distribution system that supports the integration of renewable energies in the system, the modeling of distributed generators with their respective local operation policy, the stability conditions that ensure the viability of the distributed operation of the system, operation cost, among others.

As a second milestone, the discussion will lead to the methodological proposal proposed as a solution to the exposed problem. In this sense, the main problem to be addressed is the development of a methodology for a high performance control tuning that ensures stability and optimality in the operation under uncertainty. To this end, instead of employing a monolithic optimization model that defines the parameters that compose the control policy, this proposal advocates the use of ML-based agents informed by system physics, to determine the parameters that define the optimal set-points of the DGs. Based on this premise, the integration mechanisms of the agent with the operating policies and the operating environment will be presented. This includes from the design criteria on which the construction of the agent and the database is based, to the integration of the physical laws that govern the operation of the distribution system in the training of the model. This last point will establish the link between the agent and the operation problem.

3.1 Problem statement

The objective of this proposal is to develop a control strategy for DGs to enhance performance under uncertainty. To this end, it is imperative to establish the context over which the system operation takes place. We assume an ADN with limited communication infrastructure and edge devices with low computational capacity, causing delays in information exchange. As a result, the control strategy must ensure real-time operation without relying on direct communication between buses and devices, limiting available information and complexity on the control structure. However, to achieve near-optimal operational performance it is necessary to have some kind of network-wide information available. To this end, we will use the delineations introduced by [5] and include an additional control layer that will update the parameters defining the control structure as system uncertainty is partially revealed. This additional layer allows the development of sophisticated models that capture and transmit the global optimality conditions to the local operation. The configuration of the update intervals is designed to withstand the typical delays and transient communication failures associated with the technologies employed for this task [104].

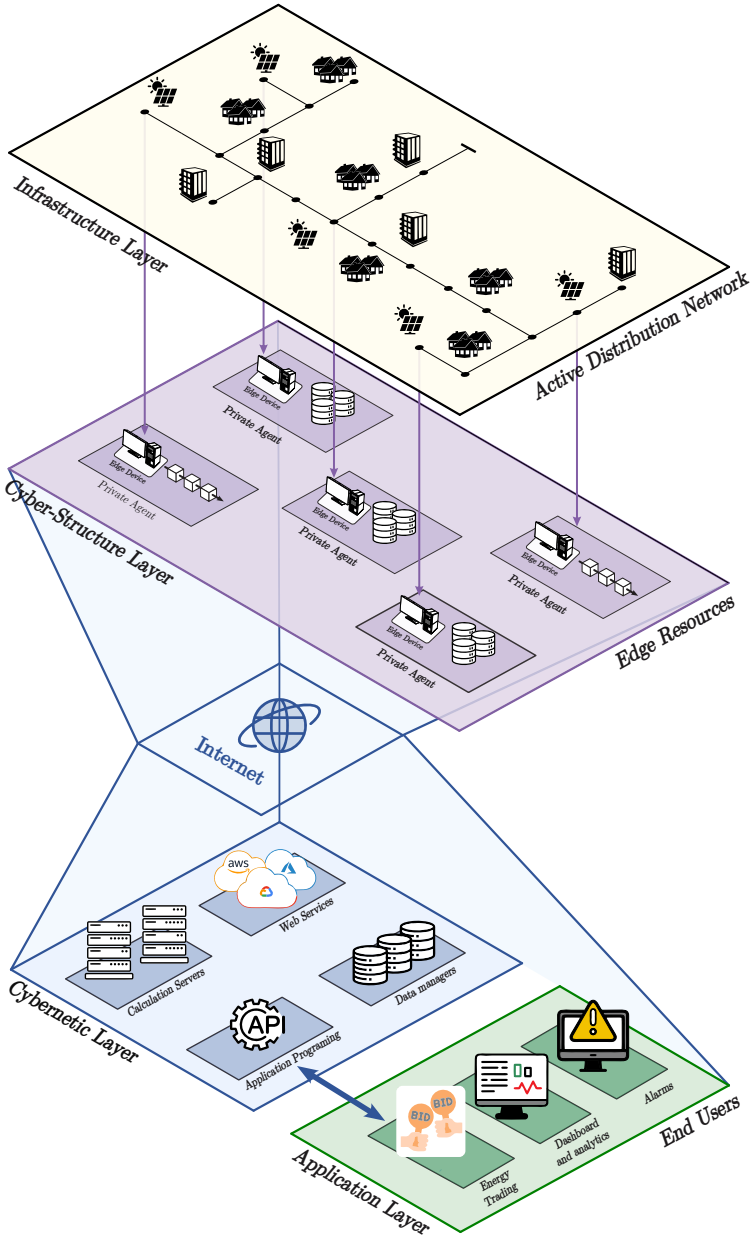


Figure 3.1: Framework of an ADN.

The schematic presented in Figure 3.1 captures the scope of the framework used for the of this proposal. The diagram presents a framework consisting of three interconnected layers: infrastructure, distributed edge resources, and centralized services. The first layer models the interactions between various ADN components, including lines, demands, and DG. The second layer encompasses the control elements interacting with edge resources, such as storage, computing, and other relevant components. Within this layer, control systems make decisions that significantly impact overall system performance. Finally, the third layer illustrates the centralized services, ranging from data storage and management to server computing services and application development. This layered conceptualization enables the development of sophisticated control strategies, as demonstrated in the case study.

Edge devices play a key role in this framework. They act as computing nodes, which, in addition to performing the necessary calculations to deploy local control strategies, ensure the secure and transparent transfer of critical energy data through communication mechanisms, such as those discussed in Table 2.1. For the development of the first task, each edge device hosts the particular functional structure that defines the local control policies at the same time. Meanwhile, the second task consists in transferring local uncertainties to the centralized system for global decision making and receiving updated control commands. The joint action of these tasks allows to adapt the global operational instructions to the local policies governing the optimal operation of the DGs. The following sections develop in detail the modeling and design scopes of the proposed solution.

3.1.1 Active distribution network modeling

The delineations presented in Section 2.3 provide the foundation for constructing the model that describes the operation of the ADN. Focusing on a radial network equipped with controlled DGs, the scope of the model is defined by detailing the interactions between system elements. This includes how the DGs will be represented, considering their operational behavior and control mechanisms to ensure optimal and stable operation. Therefore, as a frame of reference we will consider that the radial network can be described by means of an addressed graph of $\mathcal{N}^+$ nodes and $\mathcal{E}$ branches, such that $\mathcal{G} = (\mathcal{N}^+, \mathcal{E})$ and $\mathcal{E} \subseteq \mathcal{N}^+ \times \mathcal{N}^+$. Under this reference, the subset of nodes that has controlled DG is denoted as $\mathcal{N}^{\text{DG}}$, such that $\mathcal{N}^{\text{DG}} \subseteq \mathcal{N}^+$. Having established these guidelines, we now proceed with the description of the discrete ADN operation.

Definition of discrete-time system states

As a first step, we have described the quasi-stationary modeling that defines the operation of the system. For this we define the sub–index k associated to the discrete time of the simulation period; such that the dynamics of the power system can be described in the following set of quasi–stationary closed–loop equations:

$$\mathbf{x}_k = f^{\text{AC}}(\mathbf{u}_k, \boldsymbol{\xi}_k^{\text{L}}), \tag{3.1a}$$

$$\tilde{\mathbf{u}}_k = f^{\text{ctrl}}(\mathbf{x}_{k-1}, \boldsymbol{\xi}_{k-1}), \tag{3.1b}$$

$$\mathbf{u}_k = \text{proj}_{\mathcal{R}(\boldsymbol{\xi}_k)}\left(\tilde{\mathbf{u}}_k\right), \tag{3.1c}$$

$$\mathbf{x}_k \in \mathbb{R}^{|\mathcal{N}|}, \mathbf{u}_k \in \mathbb{R}^{2|\mathcal{N}^{\text{DG}}|}, \boldsymbol{\xi}_k^{\text{L}} \in \mathbb{R}^{2|\mathcal{N}|}, \boldsymbol{\xi}_k \in \mathbb{R}^{2|\mathcal{N}|+|\mathcal{N}^{\text{DG}}|}.$$

In this context, $\mathbf{x}_k$, $\mathbf{u}_k$, and $\boldsymbol{\xi}_k^{\text{L}}$ represent the deviations of the voltage magnitude, the discrete-time control actions, and the uncertainty associated with the system power demand, respectively. The voltage magnitude deviations are understood as the difference between the measurement $\mathbf{v}_k$ and the set–point value $\bar{\mathbf{v}}$ established for the operating window, i.e., $\mathbf{x}_k = \mathbf{v}_k - \bar{\mathbf{v}}$. The control actions correspond to the adjustments made by the local controller to modify the operating set–point and minimize the impact of uncertainty on the system.

The dispatch of the DGs is defined as the sum of the nominal set–point for active and reactive power, $\bar{\mathbf{u}}_k \coloneqq [\bar{\mathbf{p}}_k^{\mathrm{DG}}\, \bar{\mathbf{q}}_k^{\mathrm{DG}}]^\top$, and the intervention from the local controller, $\Delta\mathbf{u}_k \coloneqq [\Delta\mathbf{p}_k^{\mathrm{DG}}\, \Delta\mathbf{q}_k^{\mathrm{DG}}]^\top$. Therefore, the total dispatched power is given by

$$\tilde{\mathbf{p}}_k^{\mathrm{DG}} = \bar{\mathbf{p}}_k^{\mathrm{DG}} + \Delta\mathbf{p}_k^{\mathrm{DG}}, \tag{3.2a}$$

$$\tilde{\mathbf{q}}_k^{\mathrm{DG}} = \bar{\mathbf{q}}_k^{\mathrm{DG}} + \Delta\mathbf{q}_k^{\mathrm{DG}}. \tag{3.2b}$$

Together they form the vector $\tilde{\mathbf{u}}_k \coloneqq [\tilde{\mathbf{p}}_k^{\mathrm{DG}}\, \tilde{\mathbf{q}}_k^{\mathrm{DG}}]^\top$. The uncertainty associated with power demand, including both active and reactive components, is denoted by $\boldsymbol{\xi}_k^{\mathrm{L}} \coloneqq [\mathbf{p}_k^{\mathrm{L}}\, \mathbf{q}_k^{\mathrm{L}}]^\top$. To account for renewable energy integration, the overall system uncertainty is extended to include the variability of renewable generation, such that $\boldsymbol{\xi}_k \coloneqq [\boldsymbol{\xi}_k^{\mathrm{L}}\, \hat{\mathbf{p}}_k^{\mathrm{DG}}]^\top$.

This comprehensive definition of parameters allows for a clear interpretation of the equations governing the system. Equation (3.1) provides the fundamental control framework, linking the system state $\mathbf{x}$, control actions $\mathbf{u}$, and uncertainty $\boldsymbol{\xi}$. The plant's dynamics are captured in (3.1a), which models the nonlinear alternating current (AC) power flow equations for balanced systems. In (3.1b) and (3.1c), the control actions are computed and projected onto the feasible operating region of the system. Under the assumption of mild conditions as stated in [40], we ensure that for every combination of control actions $\mathbf{u}$ and uncertainty $\boldsymbol{\xi}$, there exists a unique system state $\mathbf{x}$, allowing us to reformulate the plant dynamics described by (3.1a). This formulation leads to the development of a functional control policy, denoted as $f^{\mathrm{ctrl}}(\cdot)$, which calculates a vector of reference control actions $\mathbf{u}$. The structure of (3.1) simplifies the interpretation of the interactions between $\mathbf{x}$, $\mathbf{u}$, and $\boldsymbol{\xi}$, while incorporating uncertainty as a key factor influencing the control policy. The importance of addressing uncertainty in this model is supported by the findings in [5], which demonstrate its significant impact on DG dispatch performance.

Communication layer to the devices

Various configurations allow the development of control strategies for active management within a ADN. Given the delimitation's established at the beginning of this chapter, adding an additional control layer for adjusting the parameters governing the local control structure makes it imperative to develop mechanisms that facilitate efficient data management. This task is further constrained by a limited communication infrastructure due to typical delays and transient communication failures [104]. Considering these factors, a centralized approach will be adopted, in which information from the various operating nodes is communicated and collected to generate updated control set–points asynchronously. These updated control parameters and set–points will be transmitted to the local controllers using industry standard methods. A centralized entity with access to global system information allows the implementation of advanced algorithms and models for parameter assignment, such as data-driven approaches [5] or ML-based models, as proposed in this work.

According to the schematic in Figure 3.1, the development of this framework takes place between the second and third layers, by means of edge devices with Internet access. The edge devices, in addition to performing the local control actions that command the performance of the DGs, must report as uncertainty is revealed, the local energetic data of interest to the centralized system. The centralized computation along with the storage and management of system energy data is performed on a single server computer service, with sufficient capacity for the resolution of ML-based models informed by system physics and monolithic optimization problems.

3.1.2 Operation of distributed generators

In the context of this proposal, precise modeling of DGs is critical to ensuring the quality and stability of the system. Accurately capturing their operational behavior is imperative for the effective integration of operating policies. The following key guidelines support the safe integration of well-regulated DGs, beginning with the identification of their feasible operational region, the definition of local operating policies, and the establishment of conditions that guarantee the global stability of the system.

DGs feasible region

To simplify the mathematical notation, throughout the paper we have assumed at most one DG controlled per bus. However, the presented approach can be generalized without much effort to cases with more than one DG per bus. For each DG, we have delimited its feasible dispatch region to the following representation:

$$\mathcal{R}_g^{\mathrm{DG}}(\hat{p}_g^{\mathrm{DG}}) = \left\{ \left(p_g^{\mathrm{DG}},\, q_g^{\mathrm{DG}}\right) \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} 0 \le p_g^{\mathrm{DG}} \le \hat{p}_g^{\mathrm{DG}} \\ \left(p_g^{\mathrm{DG}}\right)^2 + \left(q_g^{\mathrm{DG}}\right)^2 \le \left(\bar{s}_g^{\mathrm{DG}}\right)^2 \end{array} \right\}. \quad \forall g \in \mathcal{N}^{\mathrm{DG}} \tag{3.3}$$

Unlike (2.1d), the feasible region of the controlled DG is constrained only by the available active power and the converter's rated power. This definition simplifies the integration of DGs into the operational framework, as the feasible dispatch region introduced by the DGs is formed by the intersection of half-spaces and second–order cones, making it convex.

Local control policies

To take advantage of a systematic methodology for the design proposed in the previous section, it is essential to limit $f^{\mathrm{ctrl}}(\cdot)$ to a specific class of functions. To achieve a balance between flexibility and simplicity, the paper [5] proposes the following functional form.

$$f^{\mathrm{ctrl}}(\mathbf{x}, \boldsymbol{\xi}) = \mathbf{G}\mathbf{x} + \sum_{j=1}^{d} \mathbf{D}_j \Gamma^j(\boldsymbol{\xi})\,, \tag{3.4}$$

where the standardization used consists of

$$\Gamma\left(\boldsymbol{\xi}\right) = \frac{\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}_{\mathcal{H}}}{\sigma\left(\boldsymbol{\xi}_{\mathcal{H}}\right)}. \tag{3.5}$$

The matrices $\mathbf{G}$ and $\mathbf{D}_0, \ldots, \mathbf{D}_d$ comprise the control parameters, where the scaling function $\Gamma(\cdot)$ normalizes the values of $\boldsymbol{\xi}$ using the historical expected value and standard deviation, denoted with $\bar{\boldsymbol{\xi}}_{\mathcal{H}}$ and $\sigma(\boldsymbol{\xi}_{\mathcal{H}})$. These control parameter matrices are block-structured as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}^{p^{\mathrm{G}}} \\ \mathbf{G}^{q^{\mathrm{G}}} \end{bmatrix}, \quad \mathbf{D}_j = \begin{bmatrix} \mathbf{D}_j^{p^{\mathrm{L}} \to p^{\mathrm{G}}} & \mathbf{D}_j^{q^{\mathrm{L}} \to p^{\mathrm{G}}} & \mathbf{D}_j^{\hat{p}^{\mathrm{G}} \to p^{\mathrm{G}}} \\ \mathbf{D}_j^{p^{\mathrm{L}} \to q^{\mathrm{G}}} & \mathbf{D}_j^{q^{\mathrm{L}} \to q^{\mathrm{G}}} & \mathbf{D}_j^{\hat{p}^{\mathrm{G}} \to q^{\mathrm{G}}} \end{bmatrix}, \tag{3.6}$$

for $j = 1, \ldots, p$, where each entry in (3.6) is an $|\mathcal{N}| \times |\mathcal{N}|$ diagonal matrix whose naming superscript indicates the corresponding component–wise interactions between $\boldsymbol{\xi} \to \mathbf{u}$. Compactly, we write $\mathbf{D} = [\mathbf{D}_0, \ldots, \mathbf{D}_p]$, and we refer to $\mathbf{D}$ as the matrix of polynomial policy parameters. The action to be taken by each DG is given by:

$$\tilde{\mathbf{u}} = \bar{\mathbf{u}} + f^{\mathrm{ctrl}}(\mathbf{x},\, \boldsymbol{\xi}). \tag{3.7}$$

The control structure considers the difference between the measured voltage $\mathbf{v}$ and the set–point voltage $\bar{\mathbf{v}}$, together with the estimated set–point dispatch power $\bar{\mathbf{u}}$ for the temporary operation window. As a result of this operation, the control system is provided with the ability to integrate the newly revealed local uncertainty into the performance of each DG. This allows the system

to mitigate the impact of the uncertainty on the optimal operation of the system with proper parameter tuning. Finally, the control policy introduces a control action projection stage to ensure that the control actions are valid. The use of this is made explicit in (3.1c), such that the following operator is introduced

$$\text{proj}_{\mathcal{R}^{\text{DG}}(\boldsymbol{\xi})}(\mathbf{s}) \coloneqq \underset{\mathbf{r}\in\mathcal{R}^{\text{DG}}(\boldsymbol{\xi})}{\arg\min}\,(||\mathbf{r}-\mathbf{s}||_2^2), \tag{3.8}$$

which projects its argument to the set of feasible actuation $\mathcal{R}^{\text{DG}}(\boldsymbol{\xi})$, the definition of which is given by the following expression:

$$\mathcal{R}^{\text{DG}}(\boldsymbol{\xi}) \coloneqq \bigcap_{g}^{\mathcal{N}^{\text{DG}}} \mathcal{R}_g^{\text{DG}}(\hat{p}_g^{\text{DG}}) \tag{3.9}$$

By unifying the presented development, a compact and simplified structure is obtained that allows defining an efficient and robust control policy in the face of uncertainty, being ideal for real–time operation.

Stability conditions

Introducing operating policies that may modify the nominal dispatch of devices does not guarantee system stability. As a result, one of the key constraints in the design phase is ensuring that local actions do not compromise overall system stability. By employing the linear approximation of power flow equations [40] and disregarding the projection (3.1c), the closed–loop dynamic system can be defined as follows:

$$\mathbf{x}_{k+1} = \mathbf{B}\mathbf{G}\mathbf{x}_k + \mathbf{B}\sum_{j=1}^{d}\mathbf{D}_j\Gamma^j(\boldsymbol{\xi}_k) + \mathbf{B}\boldsymbol{\xi}_k^{\text{L}} + \mathbf{m}, \tag{3.10}$$

where, following the presented development (2.15), $\mathbf{B} \in \mathbb{R}^{|\mathcal{N}|\times 2|\mathcal{N}|}$ and $\mathbf{m} \in \mathbb{R}^{2|\mathcal{N}|}$ are known systematic parameters associated with the distribution system. Note that in the linearized system, the same reference system is used for the power injections and the uncertainty associated with the requested demand. From the elementary control theory of discrete-time linear systems [105], the exponential stability of (3.10) can be achieved by restricting the spectral radius of $\mathbf{BG}$, such that:

$$\rho(\mathbf{BG}) < 1. \tag{3.11}$$

The set defined by (3.11) is convex and can be expressed as a linear matrix inequality which can be handled by SDP. Adopting the assumptions of [5], it is assumed that the SOC approximation on (3.11) is a sufficient condition for bounded input bounded state stability (BIBS):

$$||\mathbf{BG}||_F \leq 1. \tag{3.12}$$

This approach takes advantage of the fact that any vector-induced matrix norm serves as an upper bound on the spectral radius, leading to a computationally efficient reformulation while maintaining stability. In [6], the authors showed that, under mild assumptions, condition (3.12) is sufficient to ensure BIBS of the nonlinear system (3.1) with a pure voltage droop controller. Assuming that the polynomial and uncontrolled terms of the control policy remain bounded, the extension of the stability results of [6] is ruled by:

$$||\mathbf{B}\sum_{j=1}^{d}\mathbf{D}_j\Gamma^j(\boldsymbol{\xi}_k)\,||_2 + ||\mathbf{B}\boldsymbol{\xi}_k^{\text{L}} + \mathbf{m}||_2 \leq M, \tag{3.13}$$

where $M \in \mathbb{R}$. Furthermore, we assume that the error between the linear approximation and the nonlinear AC power flow model is bounded, i.e., there exists $\delta \in \mathbb{R}$ such that:

$$||f^{\text{AC}}(\mathbf{x}_k + \boldsymbol{\xi}_k^{\text{L}}) - \mathbf{B}(\mathbf{x}_k + \boldsymbol{\xi}_k^{\text{L}}) - \mathbf{m}||_2 < \delta, \tag{3.14}$$

where δ is the error induced by the linear approximation of the AC power flow. With these assumptions, it is straightforward to extend the stability proof of [6] to show that condition (3.12) ensures BIBS stability of system (3.1) when $\mathbf{D} \neq 0$.

3.1.3 System operating cost

The operational cost captures the total net power consumption and voltage violations across the distribution network, expressed as:

$$f(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}) \coloneqq \zeta(\mathbf{u}, \boldsymbol{\xi}) + \lambda e(\mathbf{x}) \tag{3.15}$$

Here, $\zeta(\mathbf{u}, \boldsymbol{\xi})$ denotes the net power supplied by the utility at the slack bus, $e(\mathbf{x})$ represents voltage violations, and λ is a positive scalar weighting factor. Instead of treating the utility's power injection as an independent variable, $\zeta(\mathbf{u}, \boldsymbol{\xi})$ is expressed as the difference between system losses and the net power injected by loads and DGs. Therefore,

$$\zeta(\mathbf{u}, \boldsymbol{\xi}) = \phi(\mathbf{u}, \boldsymbol{\xi}^{\mathrm{L}}) - \mathbf{1}^{\top}\left(\mathbf{u} + \boldsymbol{\xi}^{\mathrm{L}}\right) \tag{3.16}$$

where $\phi(\mathbf{u}, \boldsymbol{\xi}^{\mathrm{L}})$ represents system losses. In radial networks, these losses can be approximated using a convex quadratic function:

$$\phi(\mathbf{u}, \boldsymbol{\xi}^{\mathrm{L}}) \approx \left(\mathbf{u} + \boldsymbol{\xi}^{\mathrm{L}}\right)^{\top} \tilde{\mathbf{R}} \left(\mathbf{u} + \boldsymbol{\xi}^{\mathrm{L}}\right) \tag{3.17}$$

Note that $\tilde{\mathbf{R}} \coloneqq \mathrm{diag}(\mathbf{R}, \mathbf{R})$ is a symmetric positive definite matrix with non–negative entries. Finally, the voltage violation function is defined as

$$e(\mathbf{x}) \coloneqq ||(\check{\mathbf{x}} - \mathbf{x})^{+}||_1 + ||(\mathbf{x} - \hat{\mathbf{x}})^{+}||_1 \tag{3.18}$$

where $\check{\mathbf{x}}$ and $\hat{\mathbf{x}}$ are the lower and upper bounds on $\mathbf{x}$, respectively. This formulation only penalizes voltage values outside the permissible operating range (e.g., $\pm 5\%$). Under normal conditions, the main objective is to minimize the total power consumption of the distribution network, but given the context of the work it is necessary to consider voltage limit violations.

3.1.4 Optimal system operation

After defining the design conditions underlying the integration of an additional control layer to determine the parameters defining the operating policy, the operating conditions of the system are defined through the set of expressions (3.19).

$$\psi(\mathbf{G}, \mathbf{D}, \bar{\mathbf{u}}, \bar{\mathbf{v}}, \boldsymbol{\xi}) \coloneqq$$

$$\mathbf{u} = \bar{\mathbf{u}} + \mathbf{G}\mathbf{x} + \sum_{j=1}^{d} \mathbf{D}_j \Gamma^j(\boldsymbol{\xi})\,, \tag{3.19a}$$

$$\mathbf{x} = \mathbf{v} - \bar{\mathbf{v}} \tag{3.19b}$$

$$\mathbf{v} = \mathbf{B}(\mathbf{u} + \boldsymbol{\xi}^{\mathrm{L}}) + \mathbf{m}, \tag{3.19c}$$

$$\mathbf{u} \in \mathcal{R}^{\mathrm{DG}}(\boldsymbol{\xi}) \tag{3.19d}$$

Assuming known the parameters ($\mathbf{G}$, $\mathbf{D}$, $\bar{\mathbf{u}}$, $\bar{\mathbf{v}}$) that guarantee the optimal operation under the uncertainty $\boldsymbol{\xi}$, the following optimization problem is established

$$\underset{\mathbf{y}}{\text{minimize}} \quad f(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}^{\mathrm{L}}) \tag{3.20a}$$

$$\text{subject to} \quad \mathbf{y} \in \psi(\mathbf{G}, \mathbf{D}, \bar{\mathbf{u}}, \bar{\mathbf{v}}, \boldsymbol{\xi}) \tag{3.20b}$$

where $\mathbf{y} = [\mathbf{x},\ \mathbf{u}]^{\top}$ is the vector of optimized state variables under uncertainty $\boldsymbol{\xi}$. The optimization problem (3.20) measures the performance of the operation under uncertainty by integrating the cost function as the objective to be minimized, subject to closed-loop control system equilibrium conditions defined in (3.19a) to (3.19d). Note that, in the context that the operation problem (3.20) is formulated, operation under uncertainty is defined as the performance of the control policy under partially revealed uncertainty, emulating real-time operating conditions.

3.2 Solution approach

Under the context provided at the beginning of the chapter, together with the mathematical formulation that supports its conception, the development of an efficient and robust methodology that allows adjusting the local control parameters to ensure the stability and optimality of the system operation is still a pending task. The systematic presentation of the operating model together with the operating conditions of the DGs allows discerning that the complexity of the adjustment of local parameters arises from the uncertainty inherent to the operation of the DERs and the need to continuously adapt to these changing conditions. In this sense, multiple works in the literature have defined operational and technical frameworks to meet this challenge. We highlight the contributions made in [5], which, in addition to proposing the operation policy used in this proposal, offers a solution to this problem by dynamically managing uncertainty by means of data-driven adaptive polyhedral uncertainty sets (DDUS), for the development of a robust optimization model.

In this work, we propose a novel solution that addresses these challenges by developing an ML-based agent capable of real-time adaptation of affine control policies. This agent integrates PINNs with optimization techniques, guided by KKT conditions to ensure that the physical constraints of the system are respected while optimizing performance. By combining PINNs with KKT conditions, our approach not only improves the interpretability of the model, but also enhances its ability to manage uncertainty more effectively and robustly. The following section develops in detail the modeling of the agent, its integration with the operation and system policies, design considerations that define its architecture, and the training methodology that incorporates the physical laws that define the behavior of the ADN.

3.2.1 Integration of agent into operating policy

Once the application for which it is necessary to design the agent is known, its integration with the operation framework is subject to the reception of asynchronous measurements from the system nodes, and to the display of the operation parameters computed for the update of the local control policies. Following these ideas, in Figure 3.2 the interaction of the agent with the system is schematized.

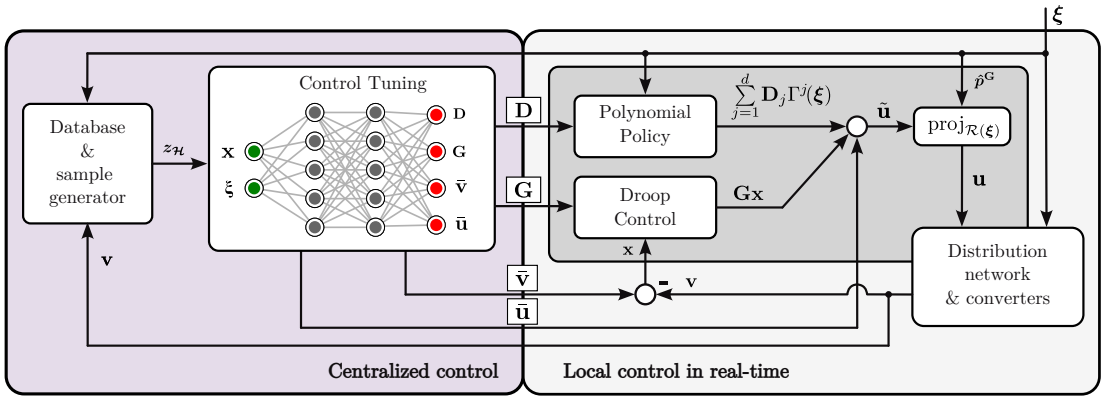


Figure 3.2: Proposed control block diagram.

The presented schematic includes some preliminary design aspects to be commented. First, the centralized control and local control actions are separated in real–time. The centralized control is framed by the energy data management of the system, together with the availability of the agent to perform the calculation of the parameters that define the local operation policies. On the other hand, the local control refers to the provision, in each DG, of the control scheme that follows from the control policy described in Section 3.1.2. The separation of tasks allows to discern how adding an additional layer for parameter determination helps the development of sophisticated models to capture and transmit the global optimality conditions to the local operation.

Secondly, as part of the agent design, the input and output components of the agent are appreciated. The agent input vector $\boldsymbol{z}_{\mathcal{H}}$ is constructed from historical measurements of variability of voltage magnitude $\mathbf{x}_{\mathcal{H}}$ and historical uncertainty $\boldsymbol{\xi}_{\mathcal{H}}$, such that $\boldsymbol{z}_{\mathcal{H}} \coloneqq [\mathbf{x}_{\mathcal{H}}\, \boldsymbol{\xi}_{\mathcal{H}}]^{\top}$ is provided. The scopes of the definition of the input vector $\boldsymbol{z}_{\mathcal{H}}$ will be discussed in Section 3.2.2. As output the agent reports the polynomial control parameters, droop control and the set–points of voltage and dispatched power, respectively. The design scopes associated with the database and agent are discussed in the following sections.

3.2.2 Database structure

The management of energy data is a primary task in the context of this work. The development of sophisticated models that capture and transmit global optimality conditions to local operating environments is based on providing an overview of the system, generally linked to measurements and historical records of the operating status. In this sense, the scheme in Figure 3.2 integrates into the centralized control actions this task, which is performed by two fundamental components: a dynamic database and advanced scenario generators. The database serves as a repository of historical energy measurements, where each entry consists of both the uncertainty associated with power demand and generation at each bus and the corresponding voltage magnitudes. In order to limit the computational resources allocated for its development, the database maintains a fixed-length historical data window at all times, which introduces the control parameter T_S. This ensures that the most relevant and recent information is always available for system analysis and decision making, so that in each simulation window, the system automatically updates the database by adding new data and removing an equivalent amount of the oldest data, as shown in Figure 3.3. This mechanism ensures that the database is not overloaded with outdated information, which could lead to inaccurate or ineffective operational decisions.

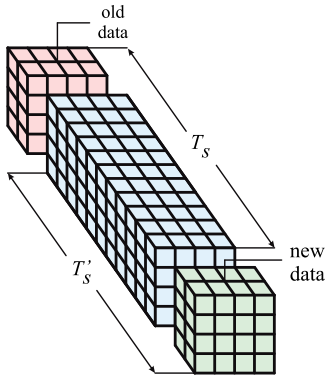


Figure 3.3: Database update.

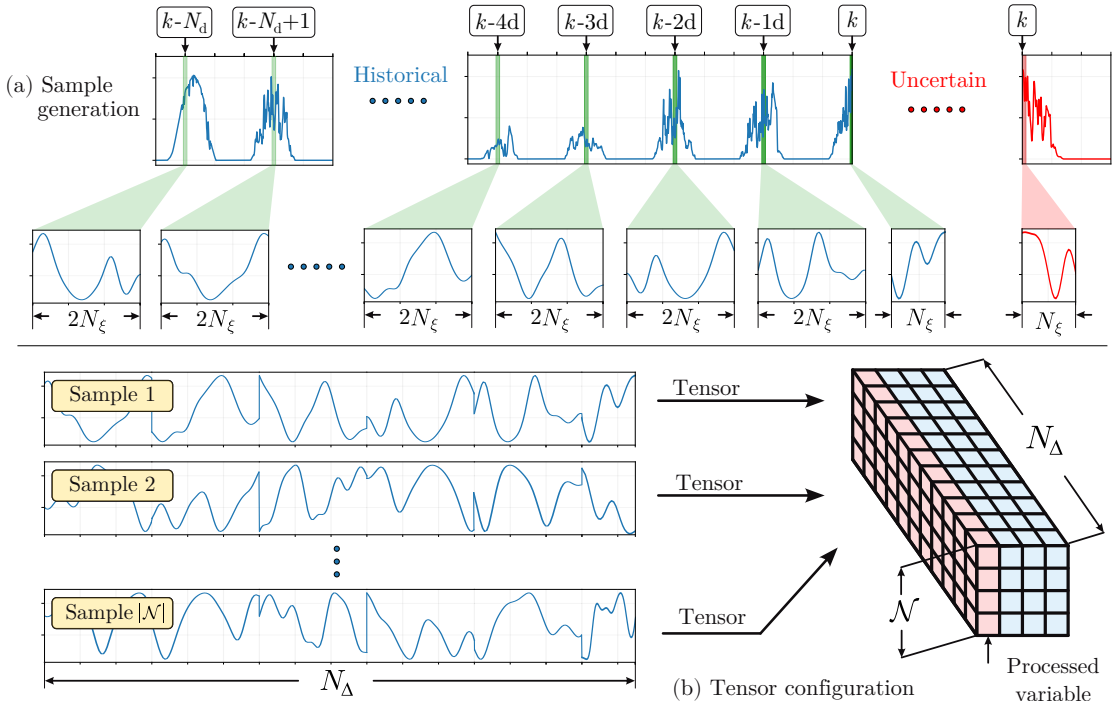


Figure 3.4: Scheme for sample generation. (a) Sample generation with temporal correlation based on daily seasonality. (b) Stacking of information for the generation of data tensor. Each time series corresponds to a row of the processed data stack.

The database is complemented by the scenario generator compatible with the agent's actions. This entity performs the task of transforming the most recent data from the database into a representative sample of the current state of the system, in order to make future decisions made by the agent, which are reflected in the construction of the parameters that define local control policies. In this sense, in this work we recognize that given the nature of the data that compose the database, there are temporal correlations that allow us to establish a more sophisticated design of the generated sample. In particular, the construction of each sample follows the guidelines presented in the Figure 3.4, where the temporal composition of the information extracts seeks to rescue the correlation with the previous instant and the daily seasonality that has a time series of uncertainty [106]. For this purpose, the length of the context window N_Δ is defined from the length of the real-time control action window N_ξ and the number of correlated days N_d, so that it is satisfied:

$$N_\Delta = (2N_\mathrm{d} + 1)\, N_\xi, \tag{3.21}$$

For a operating horizon T, such that $t \in \mathcal{T} \coloneqq \{0, 1, \ldots, T\}$, each instant t is associated with two complementary time intervals: the historical window and the uncertainty window. In this document, we denote these intervals using the sub–index "$\mathcal{H}$" and "$\mathcal{U}$", respectively. For simplicity, we omit explicit references to their relationship with the operating instant under analysis. The subindex $\mathcal{H}$ represents the context window provided to the agent, defined in accordance with $h \in \mathcal{H}$, such that $\mathcal{H} \coloneqq \{0, 1, \ldots, N_\Delta\}$. Similarly, the subindex $\mathcal{U}$ denotes the uncertainty window, whose definition is also based on $k \in \mathcal{U}$, such that $\mathcal{U} \coloneqq \{0, 1, \ldots, N_\xi\}$. The set of attributes provided as input sample is contained in the set $\mathcal{Z}$, such $\boldsymbol{z}_\mathcal{H} \in \mathcal{Z} \times \mathcal{N} \times \mathcal{H}$. This provides the generated sample with two other fundamental properties for the generalization of the agent's response: variability in the energy attributes that characterize the system and a generalized view of the spatial distribution of the nodes. The variability in the energetic attributes refers to the diversification of the time series used for the construction of $\boldsymbol{z}_\mathcal{H}$, such as active and reactive power demand uncertainty, solar generation uncertainty and voltage magnitude variability, respectively. While the generalized view of the system is provided in the ordering of the time series in the sample structure. The unification of these attributes with the time component is represented in Figure 3.5, where the sample data structure is referred to as a tensor. Hence, this data tensor is characterized by three dimensions: spatial, characteristic and temporal; linked to the cardinality of the set of buses $|\mathcal{N}|$, number of energetic attributes and contextualization window. This structured approach to data management and scenario generation allows the system to continuously adapt to changing conditions and make data–driven decisions that optimize overall performance.

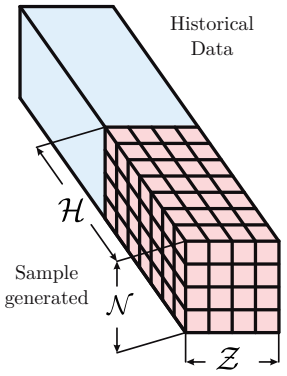


Figure 3.5: Composition of a generated sample.

3.2.3 Design criteria and agent structure

When designing an ML-based agent to determine the control parameters that define an adaptive local operation policy, it is essential to take into account the dynamic nature of the task. The agent must be able to periodically update the local controller parameters as uncertainty is revealed, knowing how to effectively adapt to evolving operating conditions in order to maintain system optimality. This objective in control operation presents a major design challenge, given the complexity of the task and the nonlinearity of the operating environment in which it is performed.

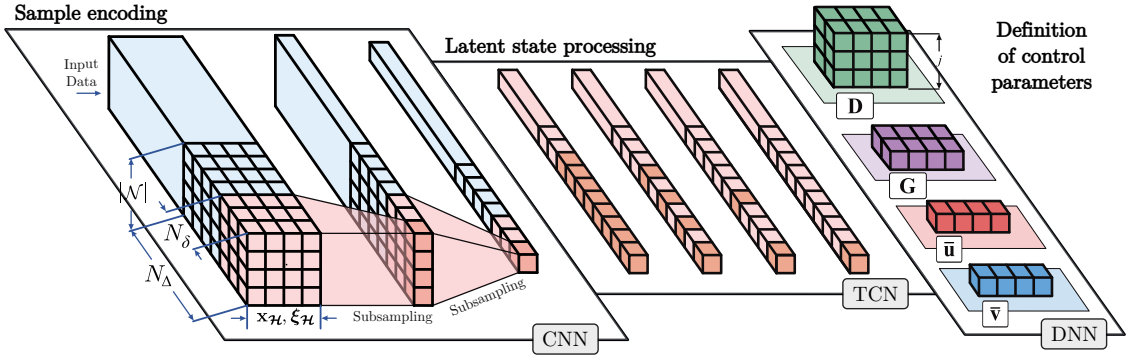


Figure 3.6: Structure of the ML–Agent.

Given the background developed in the previous sections, the architecture seen in Figure 3.6 is proposed for the agent. The key aspects of this design will be defined below, starting with the coding of the sample data, followed by the processing of the latent variables and concluding with the configuration of the output layer responsible for parameter definition.

- **Sample encoding:** Having the first task of understanding the input data in a lower dimensional latent representation, the agent employs a convolutional encoder as the first major processing block. This encoder employs a series of convolutional layers that, with proper Kernel handling, is designed to reduce the spatial and feature dimensionality of the input tensor by extracting a vector that encapsulates the intrinsic features of the sample along the temporal dimension. The underlying idea behind this design is presented below:

$$\mathbf{Q}_{|\mathcal{N}|,|\mathcal{Z}|,k} = \begin{bmatrix} \mathrm{p}^{\mathrm{L}}_{1,k} & \mathrm{q}^{\mathrm{L}}_{1,k} & \hat{\mathrm{p}}^{\mathrm{DG}}_{1,k} & \mathrm{x}_{1,k} \\ \mathrm{p}^{\mathrm{L}}_{2,k} & \mathrm{q}^{\mathrm{L}}_{2,k} & \hat{\mathrm{p}}^{\mathrm{DG}}_{2,k} & \mathrm{x}_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{p}^{\mathrm{L}}_{|\mathcal{N}|,k} & \mathrm{q}^{\mathrm{L}}_{|\mathcal{N}|,k} & \hat{\mathrm{p}}^{\mathrm{DG}}_{|\mathcal{N}|,k} & \mathrm{x}_{|\mathcal{N}|,k} \end{bmatrix} \odot \begin{bmatrix} \theta^{p^{\mathrm{L}}}_{1,k} & \theta^{q^{\mathrm{L}}}_{1,k} & \theta^{\hat{\mathrm{p}}^{\mathrm{DG}}}_{1,k} & \theta^{\mathrm{x}}_{1,k} \\ \theta^{p^{\mathrm{L}}}_{2,k} & \theta^{q^{\mathrm{L}}}_{2,k} & \theta^{\hat{\mathrm{p}}^{\mathrm{DG}}}_{2,k} & \theta^{\mathrm{x}}_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ \theta^{p^{\mathrm{L}}}_{|\mathcal{N}|,k} & \theta^{q^{\mathrm{L}}}_{|\mathcal{N}|,k} & \theta^{\hat{\mathrm{p}}^{\mathrm{DG}}}_{|\mathcal{N}|,k} & \theta^{\mathrm{x}}_{|\mathcal{N}|,k} \end{bmatrix}, \quad \forall k \in [h, \ldots, h + N^{\mathrm{CNN}}_{\delta}], \forall h \in \mathcal{H} \tag{3.22}$$

  Let h be any time instant belonging to the temporal dimension of the input tensor. The encoding process is carried out by means of 3D convolutions that incorporate part of the vencincity of the temporal measurements. This is achieved by designing the Kernel with three dimensions: the number of sample features, the buses of the system, and the length that defines this temporal neighborhood, given by the $N^{\mathrm{CNN}}_{\delta}$ parameter. Thus, by way of explanation, in (3.22) we define the projection of the 3D convolution on a 2D convolution for a time instate k of the convolved data.

  This view allows us to discern how the tensor data are presented in the convolution process with the kernel, as well as to emphasize how the dimensionality is reduced given the output operation defined in (3.23).

$$\mathrm{z}_h = \sigma\left(\sum_{i}^{\mathcal{N}}\sum_{j}^{\mathcal{Z}}\sum_{k}^{h:h+N_{\delta}} \mathbf{Q}_{i,j,k} + \vartheta,\right), \qquad \forall h \in \mathcal{H} \tag{3.23}$$

  The number of rescued features is defined by the number of filters used, denoted by N^{CNN}_{f}. Note that each filter adds a processing channel, extending in one dimension the expressions that operate by convolutions. The structure of the resulting vector of latent variables is thus:

$$\mathbf{z} = \left[\mathrm{z}_1, \mathrm{z}_2, \ldots, \mathrm{z}_{N_{\Delta}}\right]^{\top}. \tag{3.24}$$

- **Latent state processing:** After defining the structure of the **z**-coded tensor, the second stage of the architecture is built following temporal processing. This stage employs temporal convolutions with dilation, in pursuit of efficiently capturing information from the entire context window. Given the data processing presented in the first stage, the temporal length of the encoded tensor matches that of the input tensor, both being N_Δ. Under this assertion, the number of layers in the TCN-based structure is determined by:

$$N_{\mathrm{C}}^{\mathrm{TCN}} = \left\lceil \log_D \left(\frac{(N_\Delta - 1)(D-1)}{(N_\delta^{\mathrm{TCN}} - 1)} + 1 \right) \right\rceil \tag{3.25}$$

The function $N_{\mathrm{C}}^{\mathrm{TCN}}$ calculates the number of layers needed to ensure the TCN's receptive field encompasses the entire input sequence. The term inside the logarithm scales the difference between the input sequence length and the kernel size, adjusted by the dilation basis $D \geq 2$. Taking the logarithm with base d and rounding up with $\lceil \cdot \rceil$ yields the minimum number of layers required for the desired coverage. For simplicity, a single value will be used for both the kernel size and the number of filters in the convolutions, such that $N_\delta = N_\delta^{\mathrm{CNN}} = N_\delta^{\mathrm{TCN}}$ and $N_f = N_f^{\mathrm{CNN}} = N_f^{\mathrm{TCN}}$ throughout the remainder of the paper. Under this assumption, the specific causal relationships of a dilated TCN structure can be defined as follows:

$$\mathbf{T} = \begin{bmatrix} \mathrm{z}_1 & \mathrm{z}_{1+d} & \cdots & \mathrm{z}_{N_\delta \cdot d} \\ \mathrm{z}_2 & \mathrm{z}_{2+d} & \cdots & \mathrm{z}_{1+N_\delta \cdot d} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{z}_{N_\Delta - N_\delta \cdot d} & \mathrm{z}_{N_\Delta - (N_\delta - 1)d} & \cdots & \mathrm{z}_{N_\Delta} \end{bmatrix}, \qquad \boldsymbol{\theta}^{\mathrm{TCN}} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{N_\delta} \end{bmatrix}, \tag{3.26}$$

where the calculation of the output vector of a layer of the TCN structure is given by:

$$\mathbf{y}^{\mathrm{TCN}} = \sigma \left(\mathbf{T} \cdot \boldsymbol{\theta}^{\mathrm{TCN}} + \boldsymbol{\vartheta} \right) \tag{3.27}$$

Note that the task described in (3.27) is repeated in each of the layers that compose the TCN structure, such that the results of the predecessor layers are used for this purpose. This precedent makes the dilation factor d applied in a layer $l \in [1, \ldots, N_{\mathrm{C}}^{\mathrm{TCN}}]$ to be a $d = D^l$, which translates into the characteristic shape seen in the neuron activation of Figure 3.6.

- **Definition of control parameters:** Once the vector with the processed temporal characteristic is obtained, the output layers are designed in order to define the parameters of local control. Following the guidelines defined in the Section 3.1.2, we know that as a design condition the agent must return four outputs associated to the control parameters $\mathbf{G}$ and $\mathbf{D}$, and the voltage $\bar{\mathbf{v}}$ and power $\bar{\mathbf{u}}$ set-points. For the development of this task we will use blocks built by dense layers, such that for each output we have N_{D} layers of N_{n} neurons, whose output is supported by a reshaping to arrange them in the desired matrix form. Under this assumption, each output layer returns:

$$\mathbf{G} = \sigma_{\mathbf{G}}(\boldsymbol{\theta}_{\mathbf{G}} \cdot \mathbf{y}^{\mathrm{TCN}} + \boldsymbol{\vartheta}_{\mathbf{G}}^{\top}), \tag{3.28a}$$

$$\mathbf{D} = \sigma_{\mathbf{D}}(\boldsymbol{\theta}_{\mathbf{D}} \cdot \mathbf{y}^{\mathrm{TCN}} + \boldsymbol{\vartheta}_{\mathbf{D}}^{\top}), \tag{3.28b}$$

$$\bar{\mathbf{v}} = \sigma_{\bar{\mathbf{v}}}(\boldsymbol{\theta}_{\bar{\mathbf{v}}} \cdot \mathbf{y}^{\mathrm{TCN}} + \boldsymbol{\vartheta}_{\bar{\mathbf{v}}}^{\top}), \tag{3.28c}$$

$$\bar{\mathbf{u}} = \sigma_{\bar{\mathbf{u}}}(\boldsymbol{\theta}_{\bar{\mathbf{u}}} \cdot \mathbf{y}^{\mathrm{TCN}} + \boldsymbol{\vartheta}_{\bar{\mathbf{u}}}^{\top}), \tag{3.28d}$$

where the sub–index in the activation function, weights and bias of the interconnection is used to differentiate the components involved in each operation. Under the described delineations [5], we can assign a bounded linear activation function for $\sigma_{\bar{\mathbf{v}}}(\cdot)$ and $\sigma_{\bar{\mathbf{u}}}(\cdot)$; while for the control parameters $\mathbf{G}$ and $\mathbf{D}$ it must be fulfilled that $\mathbf{G} \leq 0$ and $\mathbf{D} \geq 0$. The sign condition that prevails in each parameter translates into $\sigma_{\mathbf{G}}(x) = -\max\{0, x\}$ and $\sigma_{\mathbf{D}}(x) = \max\{0, x\}$. Note that the function applied in both cases is called $\mathrm{relu}(x) = \max\{0, x\}$ in the ML context.

By endowing the agent with this structure, its conceptualization is linked to the characteristics of the system and, consequently, its structure has physical interpretability. Moreover, the parameters defining the control actions are associated to sensitivities with uncertainty, so that the agent's output also has physical interpretability with respect to the system.

3.2.4 Training informed by system physics

Once the integration mechanisms that ensure the operability of the agent in the distribution system have been established, the pending task is to establish under which operating criteria the parameters that define the ANN will be adjusted. This task is essential in the context of the proposal, since it defines the guidelines on which the agent will adjust the weights that define its performance, and consequently its overall system performance. In this sense, using an extended version of (3.19), explicitly describing the LDF power flows, we can define the following set of expressions for the system operation:

$$\tilde{\psi}(\mathbf{G}, \mathbf{D}, \bar{\mathbf{u}}, \bar{\mathbf{v}}, \boldsymbol{\xi}) \coloneqq$$

$$\mathbf{u} = \bar{\mathbf{u}} + \mathbf{G}\mathbf{x} + \sum_{j=1}^{d} \mathbf{D}_j \Gamma^j(\boldsymbol{\xi}), \tag{3.29a}$$

$$\mathbf{x} = \mathbf{v} - \bar{\mathbf{v}} \tag{3.29b}$$

$$\mathbf{P} = \mathbf{F}^\top \left(\mathbf{A}_{\mathrm{DG}}^\top \mathbf{p}^{\mathrm{DG}} + \mathbf{A}_{\mathrm{L}}^\top \mathbf{p}^{\mathrm{L}}\right) \tag{3.29c}$$

$$\mathbf{Q} = \mathbf{F}^\top \left(\mathbf{A}_{\mathrm{DG}}^\top \mathbf{q}^{\mathrm{DG}} + \mathbf{A}_{\mathrm{L}}^\top \mathbf{q}^{\mathrm{L}}\right) \tag{3.29d}$$

$$\mathbf{v} = 2\mathbf{F}\mathbf{R}\mathbf{P} + 2\mathbf{F}\mathbf{X}\mathbf{Q} + v_0\mathbf{1} \tag{3.29e}$$

$$\mathbf{P}^2 + \mathbf{Q}^2 \leq \bar{\boldsymbol{S}}^2 \tag{3.29f}$$

$$\check{\mathbf{x}} - \boldsymbol{\varepsilon}^- \leq \mathbf{x} \leq \hat{\mathbf{x}} + \boldsymbol{\varepsilon}^+ \tag{3.29g}$$

$$\boldsymbol{\varepsilon}^- \geq \mathbf{0}, \boldsymbol{\varepsilon}^+ \geq \mathbf{0}, \tag{3.29h}$$

$$\mathbf{u} \in \mathcal{R}^{\mathrm{DG}}(\boldsymbol{\xi}) \tag{3.29i}$$

In the set of expressions (3.29), the definition of the LDF is extended to explicitly integrate the power flows by lines. This operation is performed by including in the constraint set the line power limitation. For this purpose, the active and reactive power balances are defined for each bus, where the participation of the control actions and the uncertainty are also defined explicitly by means of their incidence matrices $\mathbf{A}_{\mathrm{DG}}$ and $\mathbf{A}_{\mathrm{L}}$, respectively. In turn, this formulation integrates as auxiliary variables $(\boldsymbol{\varepsilon}^-, \boldsymbol{\varepsilon}^+)$ for the reformulation of the nonlinear function that quantifies the cost of voltage deviations (3.18), such that:

$$\tilde{e}(\boldsymbol{\varepsilon}^-, \boldsymbol{\varepsilon}^+) \coloneqq \left[\mathbf{1}^\top \left(\boldsymbol{\varepsilon}^- + \boldsymbol{\varepsilon}^+\right)\right] \tag{3.30}$$

This redefines the operating costs as follows:

$$\tilde{f}(\mathbf{u}, \boldsymbol{\varepsilon}^-, \boldsymbol{\varepsilon}^+, \boldsymbol{\xi}) \coloneqq \zeta(\mathbf{u}, \boldsymbol{\xi}^{\mathrm{L}}) + \lambda \tilde{e}(\boldsymbol{\varepsilon}^-, \boldsymbol{\varepsilon}^+) \tag{3.31}$$

Under these definitions, the conjunction between the control parameter assignment problem and the system operation can be understood by means of the following optimization problem:

$$\underset{\mathbf{w}, \tilde{\mathbf{y}}}{\text{minimize}} \quad \tilde{f}(\mathbf{u}, \boldsymbol{\varepsilon}^-, \boldsymbol{\varepsilon}^+, \boldsymbol{\xi}) \tag{3.32a}$$

$$\text{subject to} \quad \tilde{\mathbf{y}} \in \tilde{\psi}(\mathbf{G}, \mathbf{D}, \bar{\mathbf{u}}, \bar{\mathbf{v}}, \boldsymbol{\xi}) \tag{3.32b}$$

$$||\mathbf{B}\mathbf{G}||_2 \leq 1 - \varepsilon \tag{3.32c}$$

where $\mathbf{w} = [\mathbf{G}, \mathbf{D}, \bar{\mathbf{u}}, \bar{\mathbf{v}}]^\top$ and $\tilde{\mathbf{y}} = [\mathbf{x}, \mathbf{u}, \boldsymbol{\varepsilon}^-, \boldsymbol{\varepsilon}^+]^\top$ correspond to the first and second stage variables respectively. In this sense, the first stage is associated with the agent's performance in defining the parameters, while the second stage is its performance in the operation under uncertainty, such that its linkage is given by a closed functional form linked to the control policy (3.29a).

Thus, it is possible to pose the KKT conditions on the performances instead of the parameters predicted by the agent. Moreover, note that the variables $(\boldsymbol{\varepsilon}^{-}, \boldsymbol{\varepsilon}^{+})$ can be expressed from $\mathbf{x}$ by means of:

$$\boldsymbol{\varepsilon}^{-} \coloneqq \max\{0,\, \check{\mathbf{x}} - \mathbf{x}\} \tag{3.33a}$$

$$\boldsymbol{\varepsilon}^{+} \coloneqq \max\{0,\, \mathbf{x} - \hat{\mathbf{x}}\} \tag{3.33b}$$

Therefore, under the definitions provided, in (3.34) the development of the KKT conditions of the expressions linked to the action on the system is presented.

$$f_{p}^{ss} = \left| \left(\boldsymbol{\partial}^{\boldsymbol{\zeta}}_{\mathbf{p}^{\mathrm{DG}}} + \lambda \boldsymbol{\partial}^{\boldsymbol{\varepsilon}}_{\mathbf{p}^{\mathrm{DG}}} \right) + \boldsymbol{\nu_1} \left(2\mathbf{P} \boldsymbol{\partial}^{\mathbf{P}}_{\mathbf{p}^{\mathrm{DG}}} \right) + \boldsymbol{\nu_2} \mathbf{1}^{\top} + \boldsymbol{\nu_3} \left(2\mathbf{p}^{\mathrm{DG}} \right) \right| \tag{3.34a}$$

$$f_{q}^{ss} = \left| \left(\boldsymbol{\partial}^{\boldsymbol{\zeta}}_{\mathbf{q}^{\mathrm{DG}}} + \lambda \boldsymbol{\partial}^{\boldsymbol{\varepsilon}}_{\mathbf{q}^{\mathrm{DG}}} \right) + \boldsymbol{\nu_1} \left(2\mathbf{Q} \boldsymbol{\partial}^{\mathbf{Q}}_{\mathbf{q}^{\mathrm{DG}}} \right) + \boldsymbol{\nu_3} \left(2\mathbf{q}^{\mathrm{DG}} \right) \right| \tag{3.34b}$$

$$f_{S}^{slk} = \left| \boldsymbol{\nu}_1 \left(\mathbf{P}^2 + \mathbf{Q}^2 - \bar{\boldsymbol{S}}^2 \right) \right| \tag{3.34c}$$

$$f_{\mathcal{R}^{\mathrm{DG}}}^{slk} = \left| \boldsymbol{\nu}_2 \left(\mathbf{p}^{\mathrm{DG}} - \hat{\mathbf{p}}^{\mathrm{DG}} \right) \right| + \left| \boldsymbol{\nu}_3 \left(\left(\mathbf{p}^{\mathrm{DG}} \right)^2 + \left(\mathbf{q}^{\mathrm{DG}} \right)^2 - \left(\bar{\mathbf{s}}^{\mathrm{DG}} \right)^2 \right) \right| \tag{3.34d}$$

$$f_{S}^{prim} = \max \left\{ 0,\, \mathbf{P}^2 + \mathbf{Q}^2 - \bar{\boldsymbol{S}}^2 \right\} \tag{3.34e}$$

$$f_{\mathcal{R}^{\mathrm{DG}}}^{prim} = \max \left\{ 0,\, \mathbf{p}^{\mathrm{DG}} - \hat{\mathbf{p}}^{\mathrm{DG}} \right\} + \max \left\{ 0,\, \left(\mathbf{p}^{\mathrm{DG}} \right)^2 + \left(\mathbf{q}^{\mathrm{DG}} \right)^2 - \left(\bar{\mathbf{s}}^{\mathrm{DG}} \right)^2 \right\} \tag{3.34f}$$

The notation $\partial^{\mathcal{Y}}_{\mathcal{X}}$ is used to develop a partial derivative of the function $\mathcal{Y}$ in terms of the variable $\mathcal{X}$. The dual Lagrangian is solved by a subgradient method [60] that computes a sequence of multipliers $(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \boldsymbol{\nu}_3)$ by solving a sequence of training samples and adjusting the multipliers using the scenarios in which some constraint is violated, i.e.

$$\boldsymbol{\nu}_1 \leftarrow \boldsymbol{\nu}_1 + \eta \cdot \max \left\{ 0,\, \mathbf{P}^2 + \mathbf{Q}^2 - \bar{\boldsymbol{S}}^2 \right\} \tag{3.35a}$$

$$\boldsymbol{\nu}_2 \leftarrow \boldsymbol{\nu}_2 + \eta \cdot \max \left\{ 0,\, \mathbf{p}^{\mathrm{DG}} - \hat{\mathbf{p}}^{\mathrm{DG}} \right\} \tag{3.35b}$$

$$\boldsymbol{\nu}_3 \leftarrow \boldsymbol{\nu}_3 + \eta \cdot \max \left\{ 0,\, \left(\mathbf{p}^{\mathrm{DG}} \right)^2 + \left(\mathbf{q}^{\mathrm{DG}} \right)^2 - \left(\bar{\mathbf{s}}^{\mathrm{DG}} \right)^2 \right\} \tag{3.35c}$$

Note that the update rate between each training step is the same as the learning rate configured for agent training. The set of expressions (3.34) is developed from the active and reactive power control actions. This is done to simplify the representation of the line-flow constraint in (3.34a) and (3.34b). Note that the matrix formulation incorporates the definitions (3.34d) and (3.34f) to realize the development of the feasible region (3.29i). Therefore, the KKT conditions associated with the LDF are complemented with the integration of the control stability condition (3.12) by the following expression:

$$f^{\mathrm{ctr}} = \max \left\{ 0,\, (1 - \varepsilon) - ||\mathbf{BG}||_2 \right\} \tag{3.36}$$

The loss function based on the physics of the environment is defined as,

$$f^{\mathrm{PINNs}} = f_{p}^{ss} + f_{q}^{ss} + f_{S}^{slk} + f_{\mathcal{R}^{\mathrm{DG}}}^{slk} + f_{S}^{prim} + f_{\mathcal{R}^{\mathrm{DG}}}^{prim} + f^{\mathrm{ctr}} \tag{3.37}$$

As a regularization mechanism, the L_1–norm is integrated over the weights composing the agent,

$$f^{\mathrm{L}} = f^{\mathrm{PINNs}} + \lambda_{\mathrm{L}} \sum_{\theta \in \boldsymbol{\Theta}} |\theta| \tag{3.38}$$

The expression (3.38) captures the behavior of the power flow model, along with control feasibility and stability conditions. With a due management of the scenarios with which the agent is trained, the model captures the behavior of the inherent uncertainty of the system, adjusting its weights for the definition of the parameters that define the optimal operation under uncertainty. The scope of the data management and training strategy will be discussed in the development of the computational experiments.

Chapter 4

Computational experiments

This chapter provides a detailed overview of the experimental design and presents the results obtained. It begins with a description of the key considerations that make up the simulation environment, including the methodology, the data used for the experiments, and the agent training strategy. It then presents the experiments that capture the scope of the proposal, defined in two phases: agent training and sensitization, and out-of-sample operational performance.

In the first phase, we present results linked to the training strategy, with a focus on model convergence under specific hyperparameters and bounded search spaces. This analysis is further enhanced by exploring the role of partial transfer learning as an acceleration mechanism in the agent's training convergence. The second phase shifts to analyzing the agent's out-of-sample performance in an operational setting. Here, alternative control strategies from existing literature serve as reference points, allowing for a comparative evaluation of the proposed and alternative approaches across three IEEE test feeder scenarios.

4.1 Simulation environment

The framework presented here is designed to simulate the operating environment, forming the foundation for computational testing and model training. This simulation integrates the control policy into the operating environment, which is essential for analyzing its impact on the ADN. The framework comprises three main components: the simulation methodology, database processing, and ANN training methodology. The simulation methodology, the first component, outlines the approach used to develop the simulation, employing a rolling horizon technique. This approach dynamically updates local control parameters based on newly available information. The database processing component details the structure in which data is stored and classified. It explains how specific subsets are constructed from this data, allowing for agent training and out-of-sample performance evaluation. Finally, the agent training methodology defines the scope of the algorithm's design post-model fitting. It describes how historical data subsets are utilized to deploy the agent within the operational environment, facilitating performance measurement as the model undergoes training.

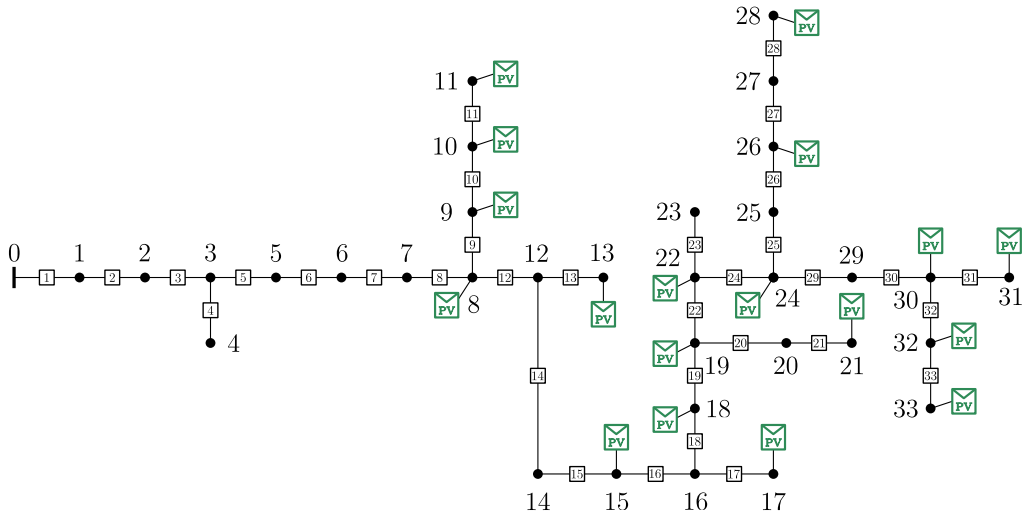


Figure 4.1: Illustration of the spatial arrangement of DGs in the 34-bus test feeder.

Extensive computational experiments have been conducted using modified versions of three IEEE test feeders: the 4-bus, 34-bus, and 123-bus feeders [107]. These modifications include balancing line segments using Carson's equation and Kron reduction, and converting distributed loads into balanced three-phase loads. Similarly, each feeder was randomly populated with controlled DGs at different locations. Specifically, for the 4-bus feeder, DGs were installed at buses 1, 2, and 3. For the 34-bus feeder, DGs were placed at buses 8, 9, 10, 11, 13, 15, 17, 18, 19, 21, 22, 24, 26, 28, 30, 31, 32, and 33. For the IEEE 123-bus feeder, DGs were installed at numerous locations: buses 9, 10, 15, 16, 18, 19, 21, 23, 27, 29, 30, 31, 32, 33, 34, 36, 37, 38, 40, 41, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 57, 58, 59, 62, 63, 64, 65, 67, 68, 69, 72, 73, 75, 76, 78, 79, 81, 82, 83, 84, 85, 86, 87, 89, 91, 93, 94, 95, 98, 99, 102, 103, 105, 106, 108, 110, 112, 113, and 119. An example of the spatial arrangement of the DGs in the system is depicted in Figure 4.1. Note that a systematic criterion for DG assignment is beyond the scope of this paper.

4.1.1 Simulation methodology

Using dynamic optimization techniques, the simulation methodology used for the development of this work is based on a rolling horizon approach to develop the real-time dispatch process. Figure 4.2 exposes the division of the time base comprising the development of in-sample and out-of-sample tasks.

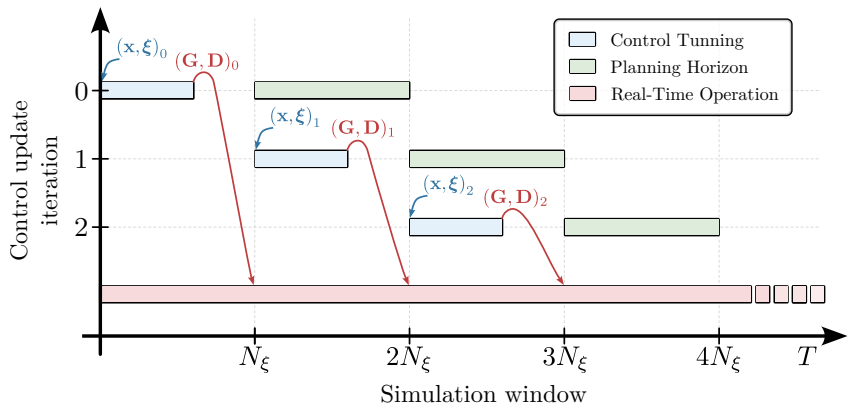


Figure 4.2: Rolling horizon concept.

The in–sample period (blue color), is operated within the additional control layer. This implies that the actions performed are of a centralized nature, such as sample generation or determination of control parameters with the agent in a delimited computing time. In this sense, the control actions taken are planned for a given operating horizon (green color). Meanwhile, in the out-of-sample period (red color) comprises local operation based on the control policies determined in the previous stage and the uncertainty revealed in real time. Figure 4.3. gives a detailed view of the coordinated operation between the two stages for a given operating horizon.

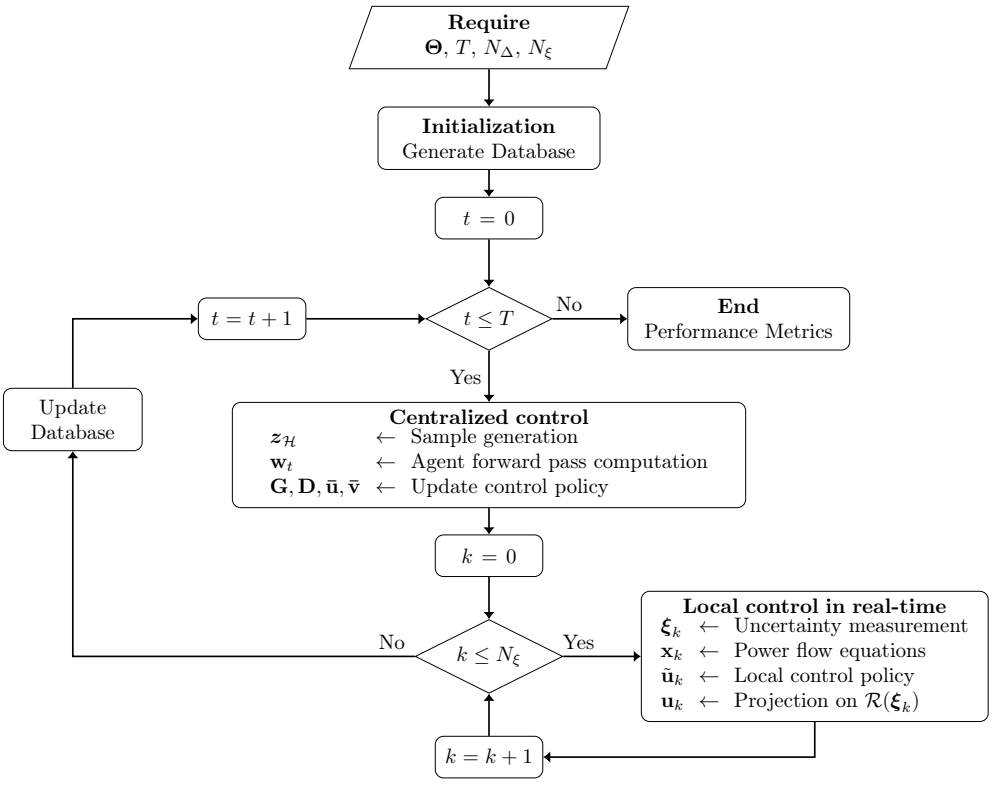


Figure 4.3: Simulation flowchart.

In Figure 4.3, concepts such as control horizon index and subscript of each instant in the uncertainty horizon are introduced. The first of these, represents at which instant of the rolling horizon the simulation is located, denoted as $t \in \{0, 1, \ldots, T\}$. While the second index does the same to define over which instant of the uncertainty window the local control action is performed, such that $k \in \{0, 1, \ldots, N_\xi\}$. So, the simulation methodology is composed of an outer control loop that performs the centralized control actions, and an inner loop that simulates the uncertainty window and performs the local control actions commanded by the centralized control, respectively.

In the outer simulation loop, the operations to be taken here and now for the interval t are performed, which includes updating the database, constructing samples and predicting the vector $\mathbf{w} = [\mathbf{G}, \mathbf{D}, \bar{\mathbf{u}}, \bar{\mathbf{v}}]^\top$ by the agent. Meanwhile, the inner simulation loop incorporates the concept of the quasi-stationary control loop introduced in (3.1) to measure the performance of the centralized control decisions. The performance of each configuration is evaluated using accurate power flow equations. This stage of operation concludes with the uncertainty data updated and communicated to the centralized controller. At the end of the simulation, key performance metrics are provided.

4.1.2 Database processing

The structure of the data set is based on the measurements used in [5]. This dataset is characterized by having power demand profiles for all the bus of the systems under study, in addition to including solar energy availability profiles for a subset of bis where the controlled DG generation facility is considered. With a time seed of 6 seconds, and a total of 67 sampled days, the dataset has a total of 964,800 individual samples.

As reported in [5], the original time series is the product of processing various household energy data from the Austin region (USA). For each household, real and reactive power (P-Q) load measurements are available, along with solar energy injection data, if applicable. On this basis, the individual loads were aggregated into equivalent bus injection profiles using an affine combination of the load time series, followed by a moving average low-pass filter to simulate the effect of overlap. In addition, seeking to capture other systemic phenomena, the database used integrates in its demand profiles the uncontrolled action of electromobility [108] and DGs operating in MPP, scaled to create cases with 100% PV penetration and 10% electric vehicles penetration relative to total system demand. This results in negative values for system demand at certain time intervals.

Table 4.1: Database summary.

Object	Parameter description	Symbol	Value
Database	Time step (seconds)	–	6
	Number of days	–	67
	Number of individual samples	–	964,800
Sets	Training set (samples)	$\mathcal{M}^{\text{train}}$	604,380
	Validation set (samples)	$\mathcal{M}^{\text{val}}$	172,680
	Testing set (samples)	$\mathcal{M}^{\text{test}}$	86,340
	Out-of-sample simulation set (samples)	$\mathcal{M}^{\text{sim}}$	100,800

To prepare the dataset for model training and out-of-sample evaluation, a 9:1 split was applied, resulting in 864,000 samples (60 days) for historical data and 100,800 samples (7 days) for uncertainty data, used in system simulation and performance evaluation, denoted as $\mathcal{M}^{\text{sim}}$. The uncertainty subset includes the most recent data, representing the final week of recorded measurements. Historical data is used to build samples for agent training, with data properties complemented by adding voltage levels for each system bus in response to sample-specific uncertainties. Following the procedure described in Section 3.2.2, we produce 863,400 instances formed by the tuple $(\boldsymbol{z}_{\mathcal{H}}, \boldsymbol{\xi}_{\mathcal{U}})$, associated with the input and uncertainty on which the agent and control policy act, respectively. In a 7:2:1 ratio, the instances are divided into training ($\mathcal{M}^{\text{train}}$), validation ($\mathcal{M}^{\text{val}}$) and test ($\mathcal{M}^{\text{test}}$) sets for cross-validation and metric computation. The summary of each subset and the properties present in the database are presented in Table 4.1.

4.1.3 Agent training methodology

The training methodology proposed for agent adjustment is structured around three interconnected algorithms that progressively address the phases of training, validation and testing of the model, in addition to the adjustment of the dual variables. In this sense, the Figure 4.4 provides a macro-level view of the training process, integrating the training, validation and dual variable adjustment phases within a loop executed over N_{epochs} epochs.

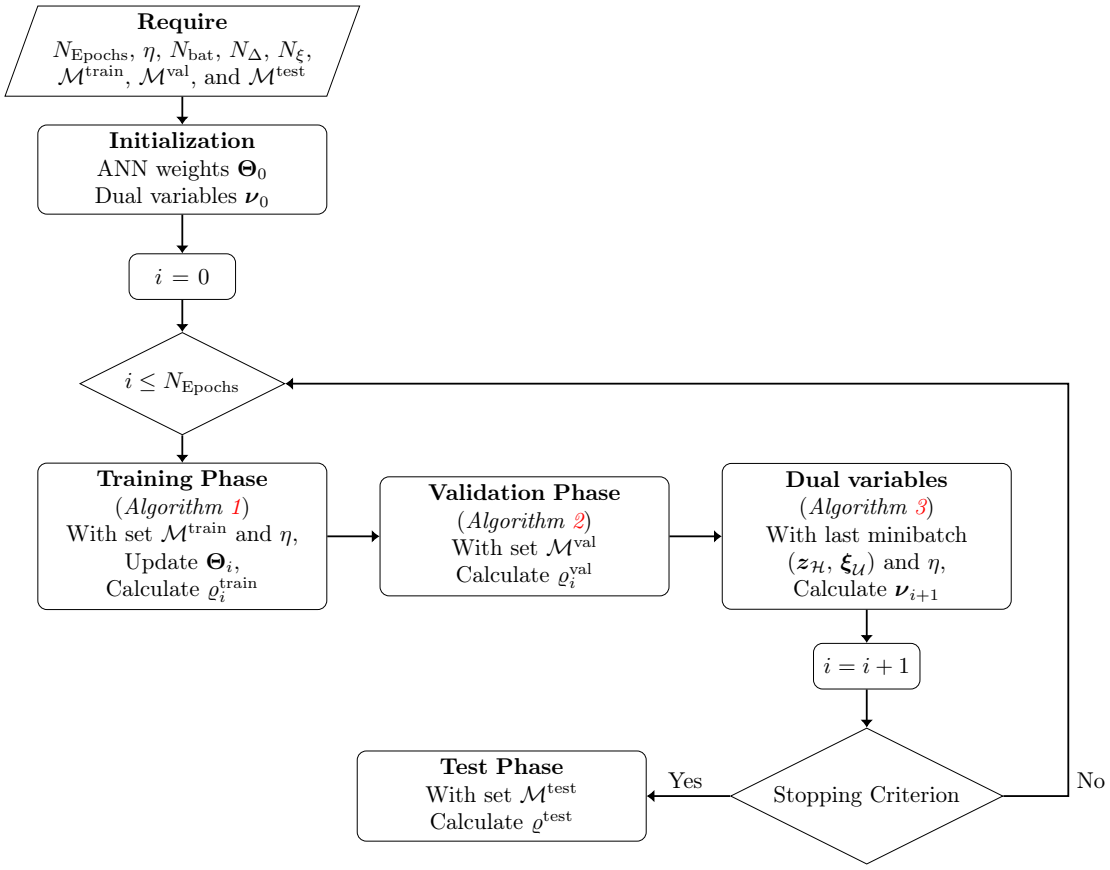


Figure 4.4: Agent tuning flowchart.

During each epoch, the weights $\boldsymbol{\Theta}$ are updated in the training phase and their performance is evaluated in the validation phase, while the dual variables $\boldsymbol{\nu}$ are updated at the end of the epoch. Using performance metrics, along with cross-validation [109], ensures a systematic approach to optimization that avoids overfitting on the training set. Once the model meets the convergence criteria, the algorithm calculates the final performance metric on the test set $\mathcal{M}^{\text{test}}$, providing an objective measure of its generalizability and overall effectiveness. Note that, in each of the phases comprising the agent adjustment, the ANN performance measurement is represented by the calculation of ϱ^{train}, ϱ^{val} and ϱ^{test}, respectively.

Deepening each one of the phases that compose the training algorithm, the document develops Algorithms 1, 2, and 3. In the first two developments, the generation of mini–batch is from the $\mathcal{M}^{\text{train}}$ and $\mathcal{M}^{\text{val}}$, respectively, is presented as a preliminary stage. The use of the mini–batches takes a fundamental role in the convergence of the model, since it smoothes the training process associated with the downward gradient [110]. The selection of the size adopted by the mini–batches, represented by the parameter N_{bat}, follows the recommendations made in [111].

Algorithm 1 Training phase

Require: $\boldsymbol{\Theta}$, η, N_{bat}, N_{Δ}, N_{ξ}, and $\mathcal{M}^{\text{train}}$

$\boldsymbol{\Theta}_{s=0} \leftarrow \boldsymbol{\Theta}$
$\mathcal{M}^{\text{bat}} \leftarrow$ Creation of subsets of size N_{bat} from random clusters of $\mathcal{M}^{\text{train}}$.
for each sample $\{\boldsymbol{z}_{\mathcal{H},s},\, \boldsymbol{\xi}_{\mathcal{U},s}\}_{s=1}^{\mathcal{M}^{\text{bat}}}$ **do**
    $\boldsymbol{z}_{\mathcal{H},s} \leftarrow$ Get input sample considering N_{Δ}.
    $\mathbf{w}_s \leftarrow$ Agent forward pass computation with $\boldsymbol{z}_{\mathcal{H},s}$ and $\boldsymbol{\Theta}_s$.
    Update of control policy with $\mathbf{w}_s$.
    for $k = 0, 1, 2, \ldots, N_{\xi}$ **do**
        $\boldsymbol{\xi}_{k,s} \leftarrow$ Uncertainty measurement from $\boldsymbol{\xi}_{\mathcal{U},s}$.
        $\mathbf{x}_{k,s} \leftarrow$ LDF equations (2.15)
        $\tilde{\mathbf{u}}_{k,s} \leftarrow$ Local control policy (3.4)
    $f_s^{\text{L}} \leftarrow$ Compute loss function using (3.38).
    $\boldsymbol{\Theta}_s \leftarrow$ Update weights using state-of-the-art optimizer with learning rate η.
    $\varrho_s \leftarrow$ ANN performance metrics
    $s \leftarrow s + 1$
    if stoping criterion is met **then**
        break
return $\boldsymbol{\Theta}_s$, ϱ_s

Algorithm 1 describes the training phase, which results in the update of the $\boldsymbol{\Theta}$ weights. For each pair of data $(\boldsymbol{z}_{\mathcal{H}},\, \boldsymbol{\xi}_{\mathcal{U}})$ belonging to a sample, the control parameters $\mathbf{w}$ are defined and their performance in the system is evaluated by means of the LDF equations. The idea behind this simplification lies in suppressing the non-linearity of the power flow equations, in order to speed up the training process [112]. Note that the internal loop emulating the out-of-sample operation does not use the projection process, since by construction, this process is captured in the loss function (3.4). The learning framework allows the use of state–of–the–art optimizers for the update of the ANN's weights, e.g., Adaptive Moment Estimation (Adam) [113], Root Mean Square Propagation (RMSprop) [110], Adadelta [114]. Training metrics are continuously updated throughout this process, allowing the algorithm to monitor progress and evaluate convergence.

Algorithm 2 Validation phase

Require: $\boldsymbol{\Theta}$, N_{bat}, N_{Δ}, N_{ξ}, and $\mathcal{M}^{\text{val}}$

$\mathcal{M}^{\text{bat}} \leftarrow$ Creation of subsets of size N_{bat} from random clusters of $\mathcal{M}^{\text{val}}$.
for each sample $\{\boldsymbol{z}_{\mathcal{H},s},\, \boldsymbol{\xi}_{\mathcal{U},s}\}_{s=1}^{\mathcal{M}^{\text{bat}}}$ **do**
    $\boldsymbol{z}_{\mathcal{H},s} \leftarrow$ Get input sample considering N_{Δ}.
    $\mathbf{w}_s \leftarrow$ Agent forward pass computation with $\boldsymbol{z}_{\mathcal{H},s}$ and $\boldsymbol{\Theta}$.
    Update of control policy with $\mathbf{w}_s$.
    for $k = 0, 1, 2, \ldots, N_{\xi}$ **do**
        $\boldsymbol{\xi}_{k,s} \leftarrow$ Uncertainty measurement from $\boldsymbol{\xi}_{\mathcal{U},s}$.
        $\mathbf{x}_{k,s} \leftarrow$ LDF equations (2.15).
        $\tilde{\mathbf{u}}_{k,s} \leftarrow$ Local control policy (3.4).
    $f_s^{\text{L}} \leftarrow$ Compute loss function using (3.38).
    $\varrho_s \leftarrow$ ANN performance metrics
    $s \leftarrow s + 1$
    if stoping criterion is met **then**
        break
return ϱ_s

Similarly, Algorithm 2 describes the validation phase, used to perform cross-validation at each of the epochs. For this purpose, the validation algorithm receives the model already adjusted for the current iteration, and on this it simulates and evaluates its performance in simulation using the validation set. By construction, the elements that compose the validation set have not been used to calculate the gradient and subsequent update of the agent, so they are out-of-sample examples. As a stopping criterion, the early stopping [115] is integrated in the cross-validation.

Algorithm 3 Updating dual variables

Require: $\boldsymbol{\Theta}$, η, $\boldsymbol{\nu}$, $\boldsymbol{z}_{\mathcal{H}}$, and $\boldsymbol{\xi}_{\mathcal{U}}$

1: $\mathbf{w} \leftarrow$ Agent forward pass computation with $\boldsymbol{z}_{\mathcal{H}}$ and $\boldsymbol{\Theta}$.
2: Update of control policy with $\mathbf{w}$.
3: **for** $k = 0, 1, 2, \ldots, N_{\xi}$ **do**
4: $\quad \boldsymbol{\xi}_k \leftarrow$ Uncertainty measurement from $\boldsymbol{\xi}_{\mathcal{U}}$.
5: $\quad \mathbf{x}_k \leftarrow$ LDF equations (2.15).
6: $\quad \tilde{\mathbf{u}}_k \leftarrow$ Local control policy (3.4).
7: $\boldsymbol{\nu} \leftarrow$ Updating dual variables according to (3.35).
8: **return** $\boldsymbol{\nu}$

Finally, Algorithm 3 describes the process of updating the dual variables associated with the inequality constraints of the power transfer limit per line (3.29f) and the feasible region of the DGs (3.29i). Using the last mini-batch used in the training stage, the subgradient method [60] is applied to update the dual variables. To simplify the use of hyperparameters employed by the model, the learning rate used on the agent to perform this update is configured. The training process ends when the convergence criteria are satisfied, indicating that the model has reached a steady state within the given training and validation set. However, this does not guarantee optimal performance, as the model could over- or under-fit the specific training data. To address this, the state of the art presents the generation of multiple training sets within a limited search space [116] as a measure. After training, the model's performance is rigorously tested on a separate test set, ensuring that it not only converges, but also maintains robustness and accuracy in a variety of scenarios.

4.2 Agent training

The training methodology described in the diagram in Figure 4.4 is implemented in Python 3.10.9, together with the Tensorflow library version 2.14.0. The platform used for testing is characterized as a server with an Intel Xeon(R) Gold 5218 processor @2.30GHz×32 and 16GB×4 of ram at 2666MHz.

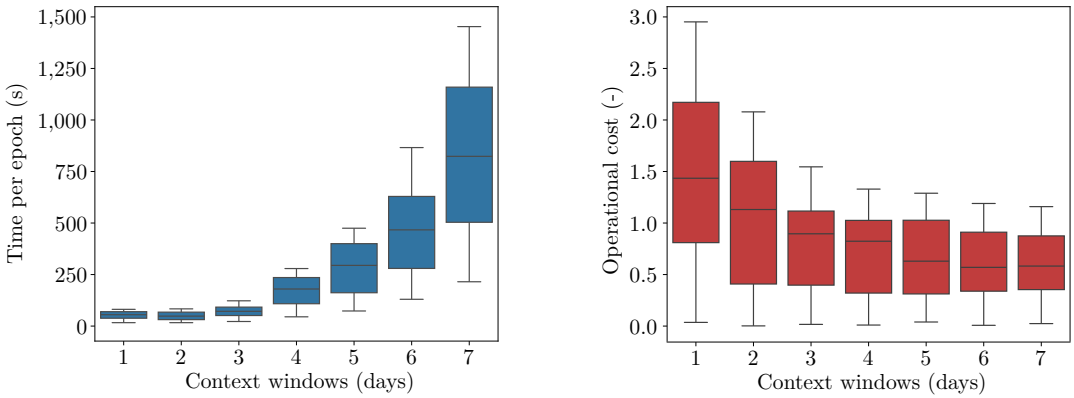
Table 4.2: Search space of ANN's hyper–parameters.

Object	Hyper-parameter description	Symbol	Search space
Enviroment	Simulation window	N_{ξ}	150
	Number of days observed	N_{d}	1–7 day
	Context window	N_{Δ}	150–2,250
Optimizer	State-of-the-art optimizer	Adam	–
	Learning rate	η	1×10^{-6}
	Batch size	N_{bat}	32
Agent	Dilatation base	D	2–3
	Kernel size	N_{δ}	150–300
	No. filters	N_f	32–64
	No. dense layers	N_{D}	1–3
	No. neurons per layer	N_{n}	10–30
	Penalty factor	λ_{L}	0.10–0.25

The agent training process was conducted within a search space defined by the data in Table 4.2. This search space encompassed the parameters and ranges relevant to the optimization of the agents, pigeonholed into which element of the system they characterize. The selection of hyper–parameters and settings that optimize the operation is obtained from the random search process. The early stopping criterion was set to 25 iterations to avoid over-fitting; while the Glorot uniform initialization was used for the initialization of the weights [117]. Likewise, the Adam optimizer is used for the weight update process. The training process is broken down into four sensitivity analyses, associated with the impact of the length of the contextualization window, the impact of regularization, the evolution of the control parameters during the training process, and the convergence of the agent.

4.2.1 Impact of days considered in N_{Δ}

As explained in the design stage, the input sample captures the intra-hour seasonality of the energy data used to adjust the control parameters. Under this idea, the following analysis seeks to elucidate how many days of observation are necessary to capture the system behavior. Delimiting the test to the 4-bus IEEE test feeder, we obtain the pareto profiles seen in Figure 4.5 is obtained.



(a) Time per epoch *vs.* context window

(b) Operational cost *vs.* context window

Figure 4.5: Agent performance *vs.* context window.

The horizontal axis of both graphs contains the observation days considered, which, by means of the ratio (3.21), is transformed into the time length of the sample. The vertical axis of the left plot reports the value of the loss function applied on the test set once the model is trained. Meanwhile, the vertical axis of the right graph reports the average time of the training process. In each case, the boxplots contain the information of the 10 models with convergence, evaluating their performance on the test set.

The observed results suggest, in the first instance, that as the number of observation days is reduced, the time spent on training is reduced. This is because, as the temporal length of the sample increases, the number of causal convolutional layers required to compute the temporal tensor, according to (3.25), increases, which increases the number of required parameters and associated gradient. These operations have a direct impact on the computational load of the platform, causing saturation in the calculation process and, therefore, a slower evaluation of the samples. On the platform in use, these phenomena cause the behavior of the computation time vs. the number of days of context to have an exponential relationship. Regarding the out-of-sample performance, measured through the evaluation of (3.38), the pareto allows us to conclude that models trained with less than two days do not fully capture the behavior of the system. This is due to the fact that the input samples are composed of a low variability of the possible values that the energy data can adopt, which generates instances in which the prediction of the control parameters is suboptimal. On the other hand, scenarios over four days have a better performance by improving the variability of the energy data, but increasing the sample length too much decreases the computational efficiency of the training process. This outlines for us the design of the agents for the other test feeders, setting the observation days to three, which is equivalent to a sample length of 450 data per sample.

4.2.2 Effect of regularization

As is made explicit in the design stage and subsequent conceptualization of the objective function, the need to incorporate regularization mechanisms is made explicit in order to improve the robustness and interpretability of the model, to minimize the risk of overfitting when learning overly complex representations from training data. To address this task, in the objective function (3.38) we have included the L_1–norm on the agent's weights in order to encourage the use of the minimum amount of these in the final configuration of the model. Consequently, the following study analyzes the impact of this regularization mechanism on the final design of the model. The number of active weights in each structural block of the agent is analyzed, according to the composition shown in Figure 3.6. As in the preliminary study, the analysis is carried out on a IEEE 4–bus feeder test, the results of which are presented in Table 4.3.

Table 4.3: Effect of regularization on the design of the agent.

Agent block	**Weights**				
	Per design	**With regularization**		**Without regularization**	
Sample encoding	86,400	56,684	(65.61%)	85,198	(98.61%)
Latent state processing	28,000	6,962	(24.86%)	24,629	(87.96%)
Definition of control parameters	99,351	54,394	(54.75%)	97,458	(98.09%)
Total	213,751	118,040	(55.22%)	207,285	(96.97%)

The models in this study follow established guidelines for optimal agent performance. For a fair comparison, both models achieve similar out-of-sample performance in the physical component of the training function (3.37), with 0.86 and 0.89 for the regularized and non-regularized versions, respectively. Table 4.3 presents the total weights per design and the active weight percentages in brackets. The L_1-norm effectively prunes weights by encouraging sparsity, reducing active weights by over 30% in each block, with the TCN block benefiting the most. This reduction limits the influence of less significant connections, promoting a more efficient model structure. Overall, active weights decrease by more than 44%, significantly reducing computational complexity. This streamlined architecture not only improves computational efficiency but also enhances the model's robustness against uncertainty, contributing to better generalization in unseen scenarios.

4.2.3 Stabilization of control actions

To evaluate how the agent stabilizes the action of the control policy as it adjusts its weights, the following study analyzes the behavior of the dispatch in out-of-sample instances for different training periods. Similarly to the previous study, the sensitivity analysis involved in this study will be limited to the IEEE 4-bus test feeder, where the configuration given to the agent corresponds to the model with the best out-of-sample performance given the delineations in the previous study. The scope associated with the convergence of the model is discussed a posteriori. Thus, in Figure 4.6, the uncertainty used for the development of each out-of-sample simulation is shown. The time series presented correspond to the first day of the $\mathcal{M}^{\text{sim}}$ set.

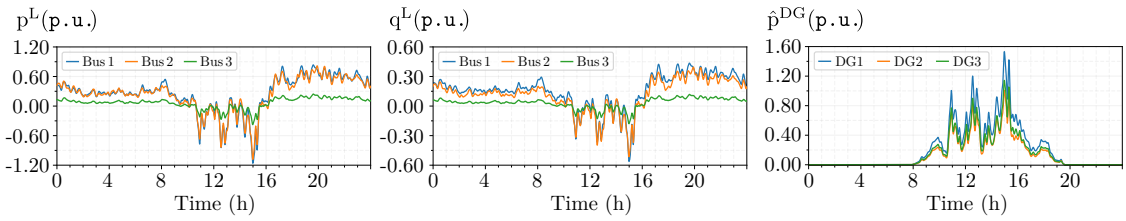


Figure 4.6: Stabilization of control actions, uncertainty used in simulation.

Meanwhile, Figure 4.7 presents the three simulations performed in a grid layout. Each column shows the training instant used to simulate the system, while each row shows the injection of active and reactive power, together with the voltage response of the system.

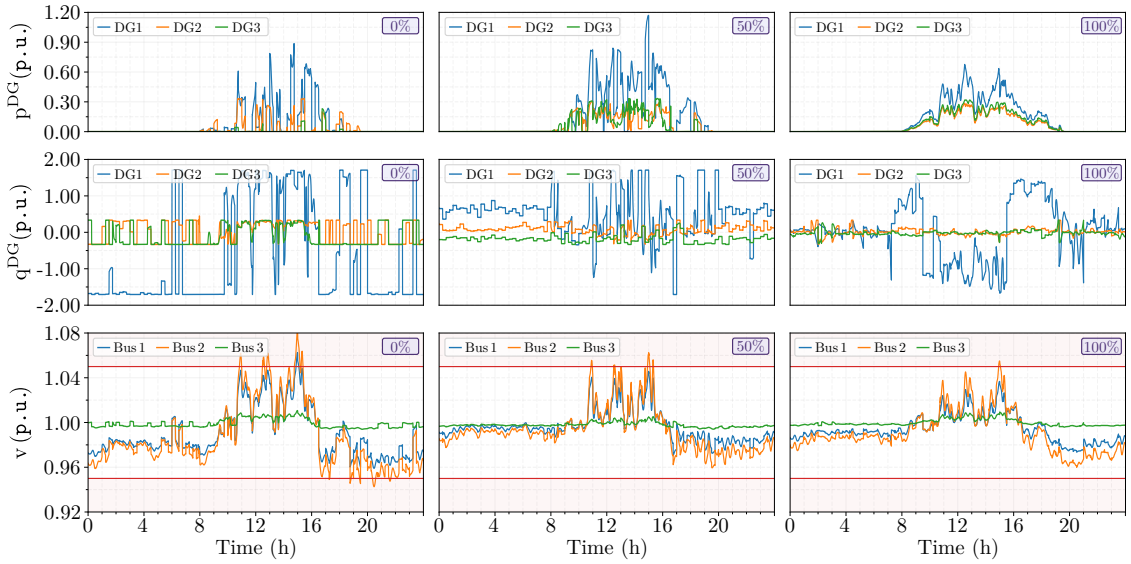


Figure 4.7: Stabilization of control actions, power dispatch and voltage response.

After the visual analysis, a clear stabilization of the control actions can be observed as the training process converges. When the model is first initialized, the behavior of the power dispatch is completely random, because the dynamics behind the system are totally unknown to the agent. This causes considerable voltage deviations at certain times. Note that the simulation flow seen in Figure 4.3 integrates the projection of the dispatch into the feasible region, which is why the operation of the system is still stable, but not optimal. As the training process advances, the model gradually acquires part of the dynamics that define the system. This is represented in the central column, where the DGs gradually align the dispatch generated according to the available power. Even so, there are intervals in which the power generation deviates from optimal operation, being zero or with many oscillations. After the objective of obtaining optimal operation, the adjustment process means that the voltage deviations and the active power spill are considerably reduced. Finally, when the training is completed, the agent manages to define the optimal control policies for the operation of the system. The active power dispatch no longer presents anomalous intervals, now following the trend presented by the available power. In the same way, reactive power dispatch complements active power, injecting or consuming power as appropriate in order to reduce voltage deviations. Likewise, for the case under study, optimal operation is achieved by pouring out active power in order to have reactive power available to control the voltage.

Table 4.4 complements the analysis with metrics typical of the operation of a distribution network, such as measuring the average voltage violation per bus (AVV), hourly average utility power injection (HUPI), hourly curtailed PV power (HCPVP), and hourly network losses (HNL).

Table 4.4: Comparison of out-of-sample performance at different stages of training.

Training Stage	**AVV $\times 10^{-7}$**	**HUPI (kW)**	**HCPVP (kW)**	**HNL (kW)**
Initialized	1907.55	464.61	411.77	72.76
Mid-training	261.03	305.17	252.33	26.36
Convergence	53.45	299.33	246.50	23.31

The results in the table numerically validate the visual analysis carried out. Progression in training goes hand in hand with optimization of the operation. The deviations in voltage, power injection from the feeder, power spillage from the DG's and energy transmission losses decrease as the agent learns to manage the system's energy around an efficient operating point.

4.2.4 Convergence analysis

After having carried out the main sensitivity analyses that comprise the performance of the proposed agent, we proceed to evaluate the convergence of each agent associated with the IEEE test feeders. As the test feeders have different capacities, numbers of lines and controllable DG implemented, the time and number of epochs required to achieve agent convergence is asymmetric. Consequently, to normalize the representation of the agent training process for each system, it will be presented on a percentage basis for each of its axes. In this way, for each of the systems under study, the results are presented in an arrangement of sub-figures with the components of greatest impact in the training process.

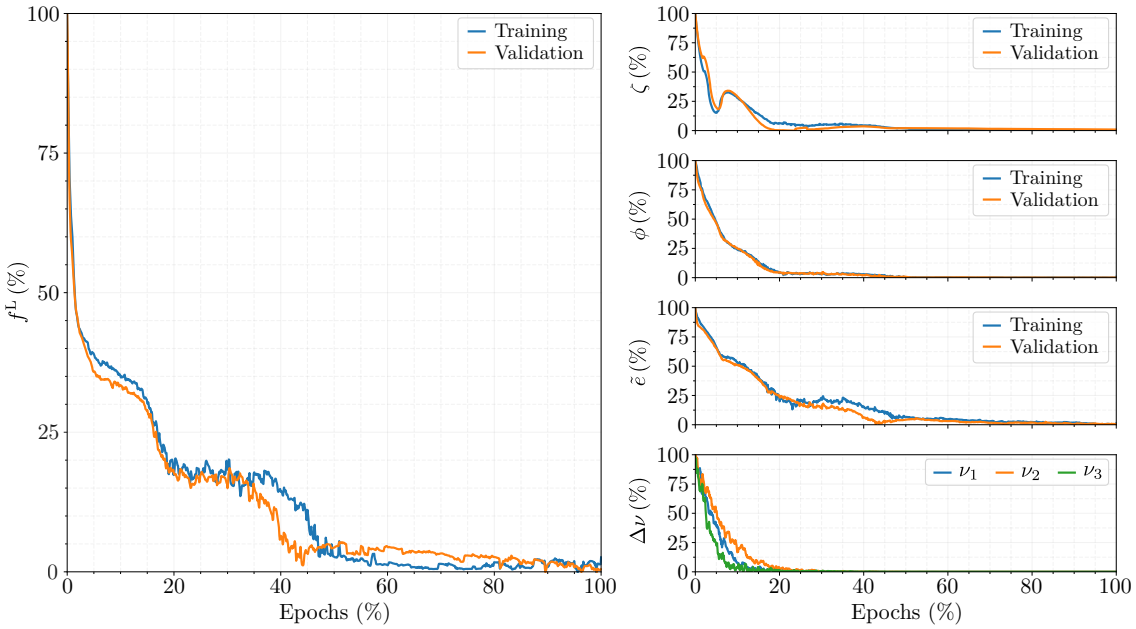


Figure 4.8: Agent convergence in the IEEE 4-bus test feeder.

Figure 4.8 makes use of this configuration to represent the results of the IEEE 4 bus feeder test. On the left-hand side, the behavior of the training function (3.38) is graphed during the parameter adjustment process. This function is implicitly composed of the power of the utility (3.16), the losses (3.17), the voltage violations (3.30) and the updating of the dual variables (3.35), so it is decided to present these in the rows on the right-hand side of Figure 4.8. Under this arrangement, the actions of the agent during the training process are made explicit. In the initial part, from the start to 20% of the training process, the agent adjusts the parameters that define the control actions to minimize non-compliance with the converter's operating limits, available power and distribution lines. This means that the dual variables do not need to be updated a posteriori, the $\Delta\nu$ being zero, as can be seen in the lower right part of the arrangement of figures.

The rest of the training is aimed at optimizing the operation of the system. This is a non-linear task, so there are intervals in which the training stagnates due to local optima. In Figure 4.8, this is exemplified in two intervals of the training, between 20–40% and 60–100%, respectively. In these intervals, the agent updates its parameters in order to manipulate the balance of the costs of the objective function, as reflected in the first three rows on the right. For the first interval, the costs of voltage violations oscillate around a value until they manage to leave the local optimum. This gives way to the interval between 40% and 60% where the minimization of the operating cost is

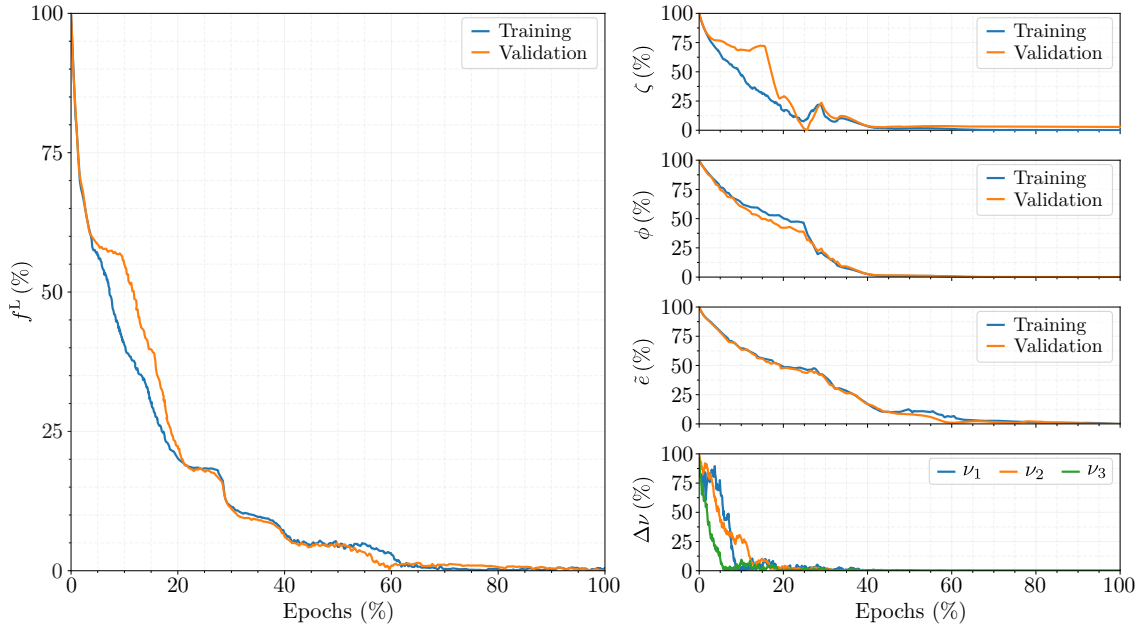
again sustained. The final stage of the training, once again balances the reduction of the different costs of the objective function of the optimization problem, until the training termination criterion is met. Note that, throughout this process, the intricate behavior of each cost function component is reflected in the training function, clear examples of which are the sustained minimization, the initial oscillation of the utility dispatch, and the slight increase in voltage deviations over the training set between 30%–50% of the adjustment process. This transfer of behavior is given by the construction of the adjustment function, which underlies the formulation of the optimization problem through the KKT conditions. Table 4.5 shows the values used to normalize the axes of each of the sub-figures.

Table 4.5: Reference values in the training process of the IEEE 4–bus feeder test.

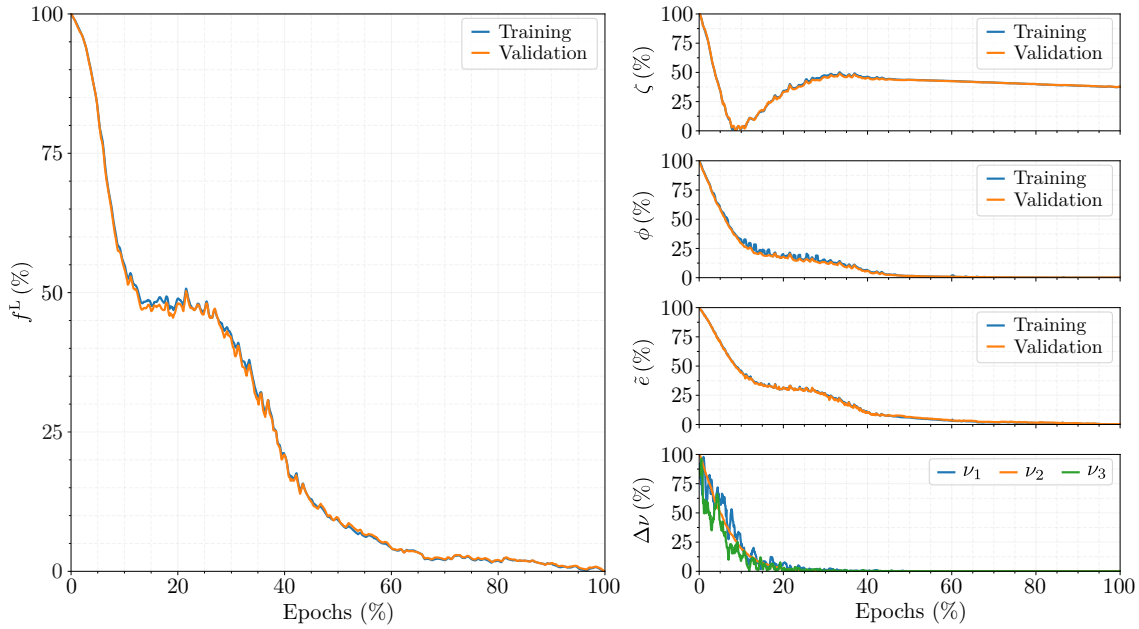
Variable	Values	
	min	**max**
Epoch	0.00	2,576.00
f^{L}	0.67	11,970.83
ζ	10.15	280.31
ϕ	0.22	77.18
$\tilde{e}$	12.14	8,395.25
$\Delta\nu$	0.00	183.97

The variables listed in Table 4.5 represent the training function f^{L} in equation (3.38), the different components of the objective function such as ζ, which denotes the cost of the utility's power injection in equation (3.16), ϕ, which represents the losses in equation (3.17), and $\tilde{e}$, which corresponds to the voltage deviations in equation (3.30). Additionally, $\Delta\nu$ represents the update component of the dual variable in equation (3.35). Note that, in each of these expressions, the average is calculated over the time window, number of buses and/or lines, and number of samples sets, respectively. In addition, they are subject to standard normalization and system in per unit, marking these differences with respect to the AVV, HUPI, HCPVP and HNL metrics.

The tabulated values provide us with more complementary information regarding the training. In the first place, the number of epochs used until convergence allows us to estimate the total time of the training process. Considering that each sample is executed in around 1.2 `s`, and that each epoch contains approximately 2000 samples, the training of the model lasts between one and two months. Likewise, the minimums and maximums of the training function and its components seen from the objective function allow us to analyze their impact on the training. In particular, voltage violations have a maximum value with an order of magnitude greater than those of the utility's power injection and losses. This is because voltage deviations are strongly affected by non-compliance with DG limits, which initially generate unfeasible scenarios where the bus voltage is well above or below the technical limits. This initial difference between magnitudes of the components of the objective function allows us to conclude that the initial stage of the training is strongly influenced by the component of the voltage deviation in (3.34a) and (3.34b), the complementary conditions (3.34c) and (3.34d), and the primal constraints (3.34e) and (3.34f) associated with the feasibility of the solution. After the adjustment of the dual variables, including after 20% of the training process, optimization under the stationary KKT conditions (3.34a) and (3.34b) takes precedence. In this interval, the primal and complementary conditions are already met, and the difference in magnitude between the voltage deviations and the power injection of the utility and losses is attenuated. This means that the training updates the weights of the agent by balancing and minimizing the three components, arriving at the configuration of parameters that best minimizes the cost of operation.



(a) Glorot initialization



(b) Warm–starting

Figure 4.9: Agent convergence in the IEEE 34-bus test feeder.

From the analysis we have just carried out, it can be concluded that the scalability of the training methodology is computationally expensive. The training times required for larger distribution systems, with a greater number of controllable DGs, increase significantly. To overcome this difficulty, we will take advantage of the standardized structure given to the agent to transfer part of the learning obtained in the model associated with the IEEE 4–bus feeder test. In particular, the input and output tensor dimension of TCN–based latent state processing is compatible between the three IEEE feeder tests. This allows the weights already entered on the IEEE 4–bus feeder test system to be transferred and frozen in the other systems, given that the task performed at this stage is the same regardless of the system. Under this premise, Figure 4.9 presents the result using traditional initialization, in Figure 4.9(a), and initialization with transfer learning for a warm start, in Figure 4.9(b).

Unlike the sensitivity studies, in which multiple initializations were carried out to choose the one with the best performance, in the IEEE 34–bus feeder test we will only initialize one agent for the comparison of traditional initialization and warm start. When using traditional initialization, it can be seen in Figure 4.9(a) that the training follows a behavior similar to that previously analyzed. First, the agent accommodates the dual variables instead of operating in the feasible region of operation and then the operation of the system is optimized. In the optimization stage there are also local optima, which are circumvented by manipulating the control parameters in order to vary the costs of the different components of the objective function. Furthermore, it can be seen once again how the behavior of the components of the objective function of the optimization problem directly influences the behavior of the training function, a clear example being the behavior of the validation component between the figure on the left and the first row of the figure on the right.

On the other hand, when using warm start in the TCN structure, Figure 4.9(b), two significant changes occur in the training process. There is a slight acceleration in the computational calculation of each epoch, since fewer weights are trained and consequently fewer gradients are calculated; and the convergence is much faster. Both behaviors translate into an acceleration of the training process. As in the training process with traditional initialization, the dual variables are the first to be optimized. This is done from the beginning up to 30% of the training, where there is no longer a relevant update value. Once the dual variables have been adjusted, and as in the previous analyses, the training completes the optimization of the system. In the particular case of Figure 4.9(b), it presents a single period of stagnation around 10% and 30% of the training process. In this interval, the internal balance of the components of the objective function of the operational problem lies in increasing the participation of the utility in order to minimize losses and voltage deviations. After this period, a gradual and sustained reduction in the loss function and the components of the objective function is observed. Table 4.6 shows the values related to the normalization of both cases studied.

Table 4.6: Reference values in the training process of the IEEE 34–bus feeder test.

Variable	Glorot initialization		Warm–starting	
	min	max	min	max
Epoch	0.00	2,323.00	0.00	1,312.00
f^{L}	1.15	985,222.99	1.11	756,236.22
ζ	10.22	234.55	6.01	181.55
ϕ	0.86	119.56	0.89	90.16
$\tilde{e}$	18.14	903,958.25	17.71	533,499.12
$\Delta\nu$	0.00	315.22	0.00	243.91

The tabulated data complements the visual analysis, providing information on the minimums and maximums recorded in each case. Firstly, there is a significant reduction in the number of epochs needed for convergence. This is because, by using and setting already trained weights, the training efforts are focused on optimizing the rest of the agent's parameters. Consequently, the training process is more efficient, leading to a decrease of around 40% in training time. This indicates that a large part of the efforts of the training algorithm are focused on the adjustment of the latent state processing stage. Secondly, the maximum values recorded for the components of the objective function are lower in the case of warm start. This is because a warn start can be understood as an initialization point where the problem of parameter adjustment is better conditioned, which leads to a faster and more stable convergence of the optimization process, by reducing sensitivity to initial conditions and facilitating the search for an optimal minimum in the parameter space. The equivalence in the performance of the trained models is reflected in the minimum values tabulated in Table 4.6. Usually linked to the final values of the training, it can be seen that these present equivalent values and orders of magnitude between the models. In this way, the results of this analysis validate the use of warn start between trained agents under different feeders.

The study ends by presenting the behavior of the agent trained for the IEEE 123-bus feeder test. Taking as a reference the results obtained for the IEEE 34-bus feeder test, in this system the agent is only trained with the warn start, as can be seen in Figure 4.10

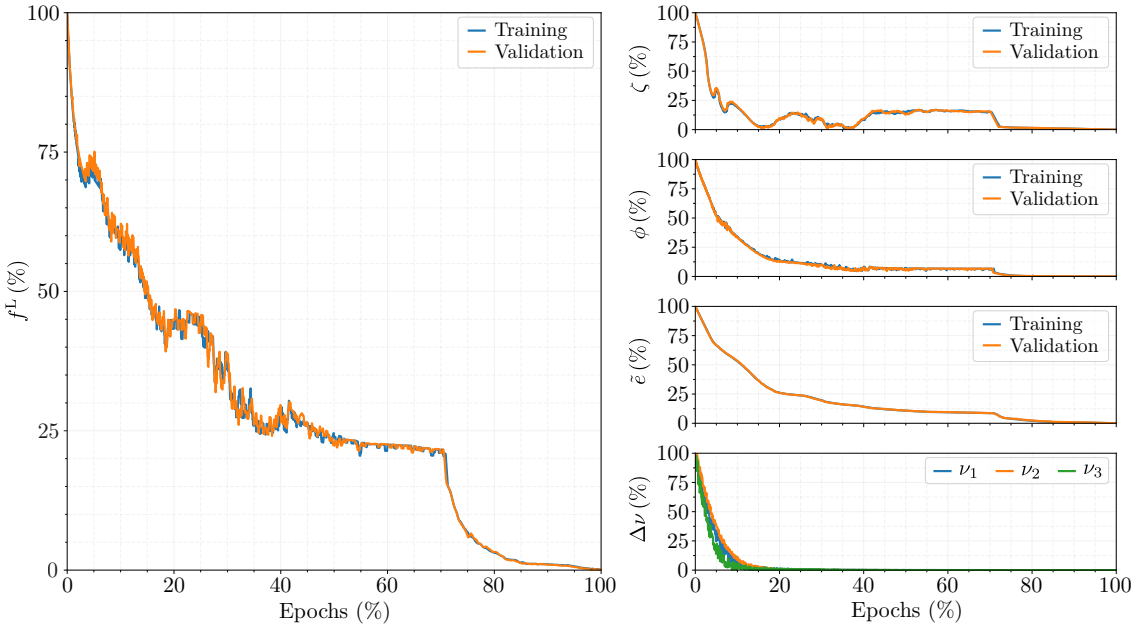


Figure 4.10: Agent convergence in the IEEE 123-bus test feeder.

This is reflected in the behavior of the curves in Figure 4.10, where, as in the case of the agent with a warn start of the IEEE 34-bus feeder, the percentage of training allocated to the adjustment of the dual variables is close to 15%. Once this threshold is exceeded, the parameters are adjusted to comply with the feasible regions of operation. After this, the training algorithm enters the operation optimization process, which involves adjusting the parameters so that their impact on the objective function minimizes the value of (3.32) in its in-sample evaluation. In the particular case of this system, the behavior of the loss function is strongly linked to the cost of the utility's participation, which is adjusted according to the reduction of system losses and voltage violations. Table 4.7 shows the minimum and maximum values recorded for each of the variables surveyed in the set of graphs in Figure 4.10.

Table 4.7: Reference values in the training process of the IEEE 123–bus feeder test.

Variable	Values	
	min	**max**
Epoch	0.00	2,393.00
f^{L}	2.34	1,458,215.25
ζ	28.77	451.22
ϕ	2.21	145.23
$\tilde{e}$	48.78	1,212,061.33
$\Delta\nu$	0.00	682.99

The values tabulated in Table 4.7 are the agent's frame of reference for the IEEE 123–bus feeder test. Again, as the distribution system increases in size, the execution time of a sample, and consequently the training time, increases in proportion to the amount of information stored in the input tensor and the number of controllable DGs in the system. For this particular case, each sample takes around 1.8 s to execute. This means that the training process, starting from a warm state, takes approximately 3 months. Finally, the minimum and maximum values allow us to evaluate the convergence of the agent. In particular, as observed in the previous model, the maximum values increase due to the increase in controllable elements and system capacity. While

the minimum values give indications of convergence according to the early stopping criterion. These values have a non–zero value, but are significantly lower than the maximum recorded. Note that, despite recording minimum values other than zero, this does not necessarily mean that the agent is operating far from the optimum, since the value associated with the loss function constructed from the KKT is close to zero, with a small gap associated with optimality.

The results associated with the convergence of the models indicate that the training informed by physics in the construction and conceptualization of the objective function achieves its goal. In addition to training the models, the structure of the agent and underlying training process is physically interpretable. Furthermore, it is demonstrated how the warm start mechanisms considerably accelerate the convergence of the large-scale models. To conclude the convergence analyses, Table 4.8 shows the performance of each model on the test set.

Table 4.8: Performance on the test sets of the agents associated with each IEEE test feeder.

Test feeder	f^{L}	ζ	ϕ	$\tilde{e}$
IEEE 4–bus	0.60	10.86	0.21	11.15
IEEE 34–bus	1.21	7.12	1.01	16.33
IEEE 123–bus	2.40	31.01	2.87	45.22

The results tabulated in Table 4.8 are consistent in order of magnitude with respect to the minimum values presented above. This indicates that, on a set totally exogenous to the training, each of the agents manages to obtain control parameters in such a way that optimal conditions prevail in the operation of the system.

4.3 Out-of-sample performance

This section evaluates the out-of-sample performance of the trained agents using the simulation methodology outlined in Figure 4.3. All strategies, except for the proposed one, use 400 scenarios for parameter adjustment. The performance of each strategy is compared against various control methodologies from the literature, using standard metrics in distribution systems: AVV, HUPI, HCPVP, and HNL. The results, presented in Table 4.9, are derived from the $\mathcal{M}^{\mathrm{sim}}$, which includes the last seven days of uncertainty from the database.

Table 4.9: Comparison of the proposal *vs.* literature models in the out-of-sample operation.

Test feeder	**Strategies**	**AVV $\times 10^{-7}$**	**HUPI (kW)**	**HCPVP (kW)**	**HNL (kW)**
IEEE 4–bus	IEEE–1547	769.51	251.65	96.26	26.54
	SOPF	12.15	541.11	414.29	30.02
	NCognizant	5.32	584.91	429.52	31.31
	Robust	7.64	414.76	259.36	26.17
	Proposed	2.04	433.25	262.12	24.66
IEEE 34–bus	IEEE–1547	637.46	−196.82	0.00	42.05
	SOPF	105.38	723.17	556.17	98.77
	NCognizant	96.49	765.09	568.27	105.83
	Robust	0.00	392.49	195.67	62.59
	Proposed	0.00	367.18	183.94	51.91
IEEE 123–bus	IEEE–1547	11245.20	−252.75	80.54	153.28
	SOPF	471.15	1212.15	1344.18	56.33
	NCognizant	374.41	1387.25	1389.70	50.26
	Robust	75.71	195.77	486.40	107.11
	Proposed	53.88	203.25	477.21	101.22

The schemes used for this comparison are the IEEE-1547 standard [8], the determination of set-points by stochastic optimization of the OPF (SOPF), the optimization of local voltage sag control considering a network cognizant approach [6] (NCognizant), robust co-optimization of the control structure [5] (Robust) and the proposal of this work, respectively.

In the tabulated results we observe that NCognizant, SOPF, Robust and Proposed achieve satisfactory voltage regulation. This is because, by construction, these strategies integrate knowledge of the system into the definition of the dispatch, which is totally contrary to IEEE–1547, where the dispatch is defined through the characteristics of the DG and the local measurement of the voltage. Among these strategies, the proposed approach produces the most economically optimal solution by minimizing the other metrics, i.e., HUPI, HCPVP, and HNL. This statement is made explicit in Figure 4.11, which shows the probability distribution of average daily operating costs, which considers a weighted sum of AVV and HUPI, as formulated in (3.31).

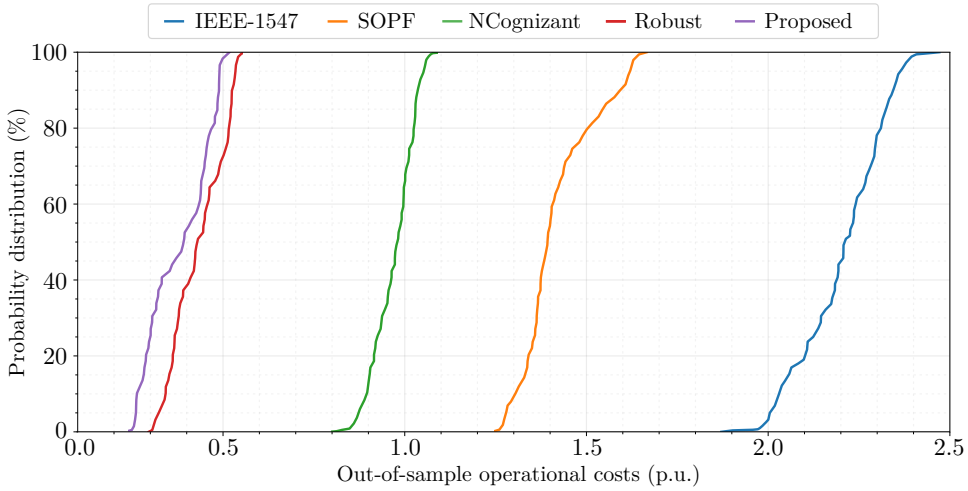


Figure 4.11: Out–of–sample performance for daily average operational costs, IEEE 34–bus test feeder.

This figure shows how each control methodology impacts the economic performance of the system. The results of the computational experiments suggest that the proposed strategy captures a multi-objective approach that achieves more optimal decisions from an economic point of view while regulating voltage effectively, because the proposed strategy systematically presents a lower operational cost compared to the strategies in the literature. When comparing the methodologies for determining control structure parameters (3.4), the null hypothesis assumes that the agent-based approach defines parameters with superior operational performance compared to the robust method. The significance test yields a p–value of 3.98×10^{-5}, which is below 0.05, indicating that the proposed methodology results in significantly lower operating costs than the robust approach. This improvement stems from the enhanced modeling of uncertainty and the incorporation of system physics into the training function design.

4.3.1 Analysis of the intra-hour operation

This section aims to evaluate the intra-hourly operational behavior. For this purpose, a typical day of the 34-bus test feeder is selected for analysis. This study involves the comparison of three control methodologies: the IEEE–1547 standard, a NCognizant model from the literature, and the proposed approach developed in this work. The study is limited to these strategies as they are representative of the spectrum of economic solutions seen in the previous section.

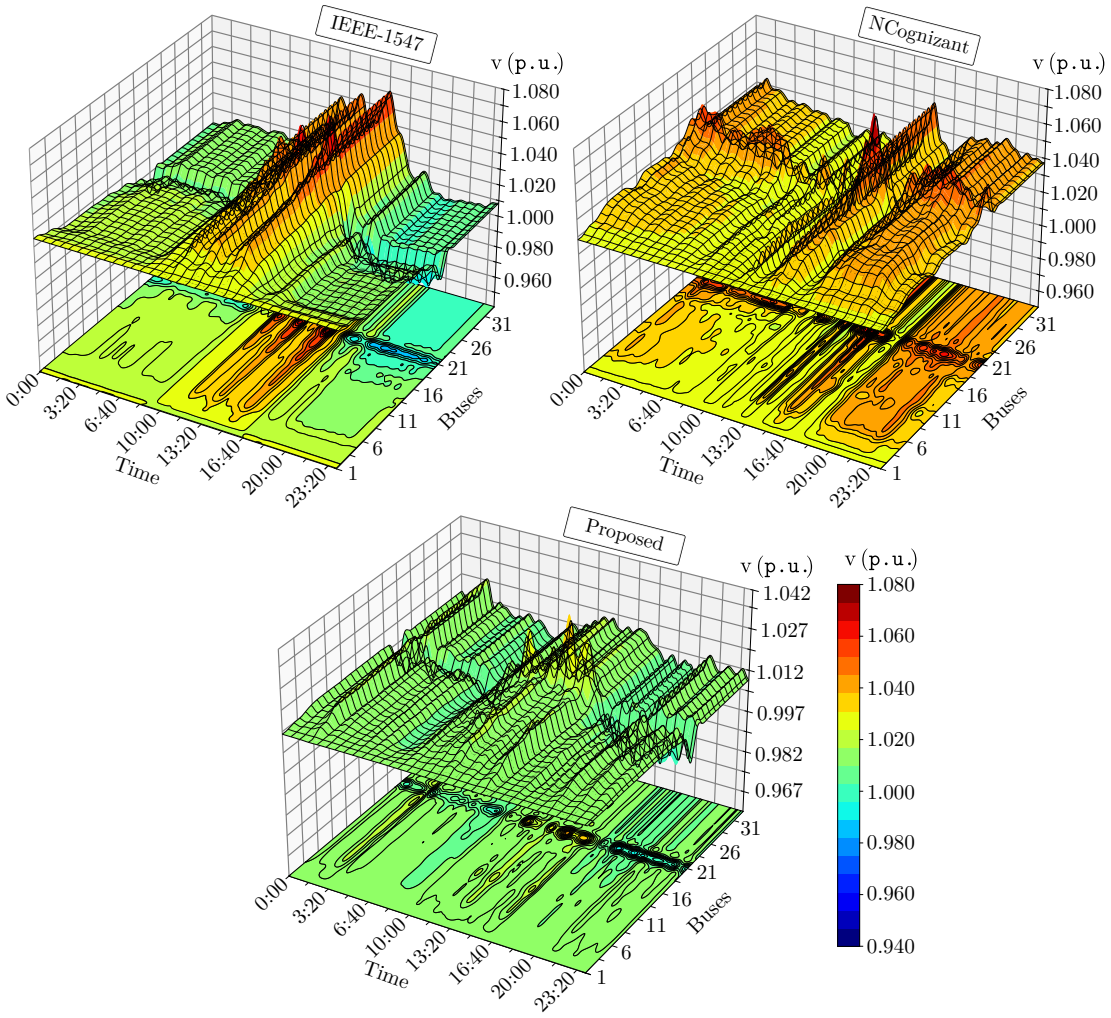


Figure 4.12: Voltage response comparison: literature models *vs.* proposed.

The Figure 4.12 shows three surfaces, where in each one the axes are time, buses, and voltage response. The figures in the upper left and right are associated with the IEEE–1547 standard and the NCognizant model, while the one at the bottom is obtained with the proposal. The color map used is common in all surfaces, and indicates the voltage level reached on each bus for each instant of the day. The behavior observed allows us to distinguish multiple events. In the first place, the strategy proposed by the IEEE 1547 standard does not manage the voltage effectively, generating multiple instances where the maximum voltage level is violated on more than one bus. These instances where there is a voltage violation are concentrated in the interval from 10:50 `am` to 3:20 `pm`, for buses 20 and 21. The NCognizant model mitigates voltage deviations, presenting an average voltage profile slightly higher than that achieved with the IEEE 1547 standard. In particular, this control strategy eliminates voltage deviations in bus 21, and for bus 20 it reduces the interval where voltage deviations exist, being present between 2:20 `pm` and 3:20 `pm`, with 1.079 `p.u.` as maximum voltage recorded. The proposed methodology does not present voltage deviations for the day in question, reaching a maximum of 1.040 `p.u.` for bus 21 at 2:20 `pm`. These differences support the results in Table 4.9, where the methodologies analyzed present different orders of magnitude in the AVV indicator.

When subjecting the systems to the same uncertainty, the voltage response depends solely on how the dispatch of the DGs is carried out. Figure 4.13 and 4.14 show the dispatch of active and reactive power of the DGs. Both figures show three surfaces, where in each one the axes are time, the id of each generator, and the power dispatch. As with the previous figure, the arrangement of figures includes in the upper left and right the dispatch associated with the IEEE–1547 standard and the NCognizant model, while the one in the lower part is obtained with the proposal; all subject to a common color map indicating the maximum and minimum dispatch of each device.

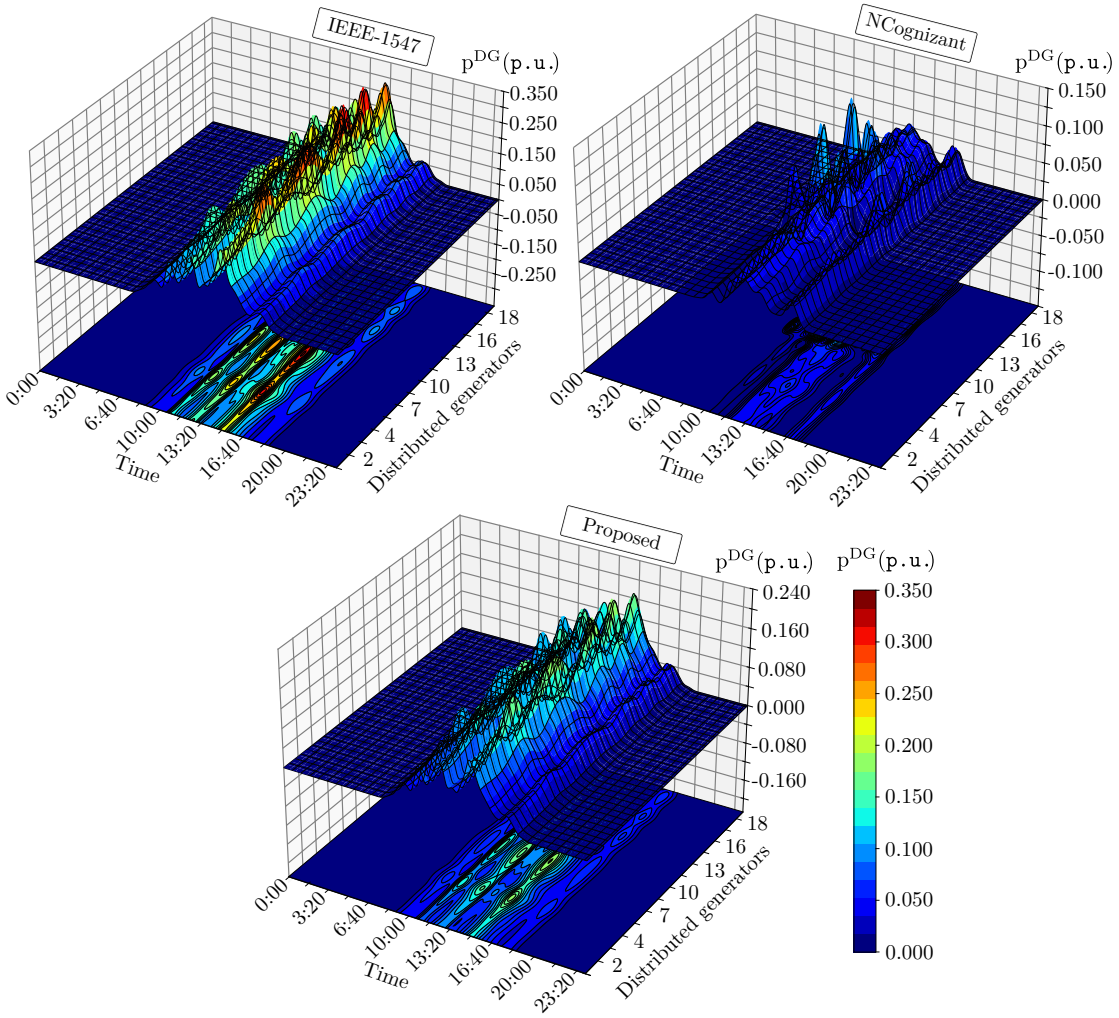


Figure 4.13: Active power dispatch comparison: literature models *vs.* proposed.

The active power dispatch surfaces of each control methodology provide distinct operating guidelines. Under the IEEE-1547 standard, minimizing active power spill is the primary objective. In contrast, both the model from the literature and the proposed approach treat active power spill as an additional control mechanism for system energy management. As a result, operation under the IEEE-1547 standard typically sets the available power as the operating point, whereas the other two models follow dispatch determined by their respective control structures. Notably, the NCognizant model focuses solely on voltage drop control, while the proposed approach also incorporates voltage droop control and a polynomial component to account for uncertainty. These differences in dispatch definition lead to variations in both the magnitude and shape of the resulting figures.

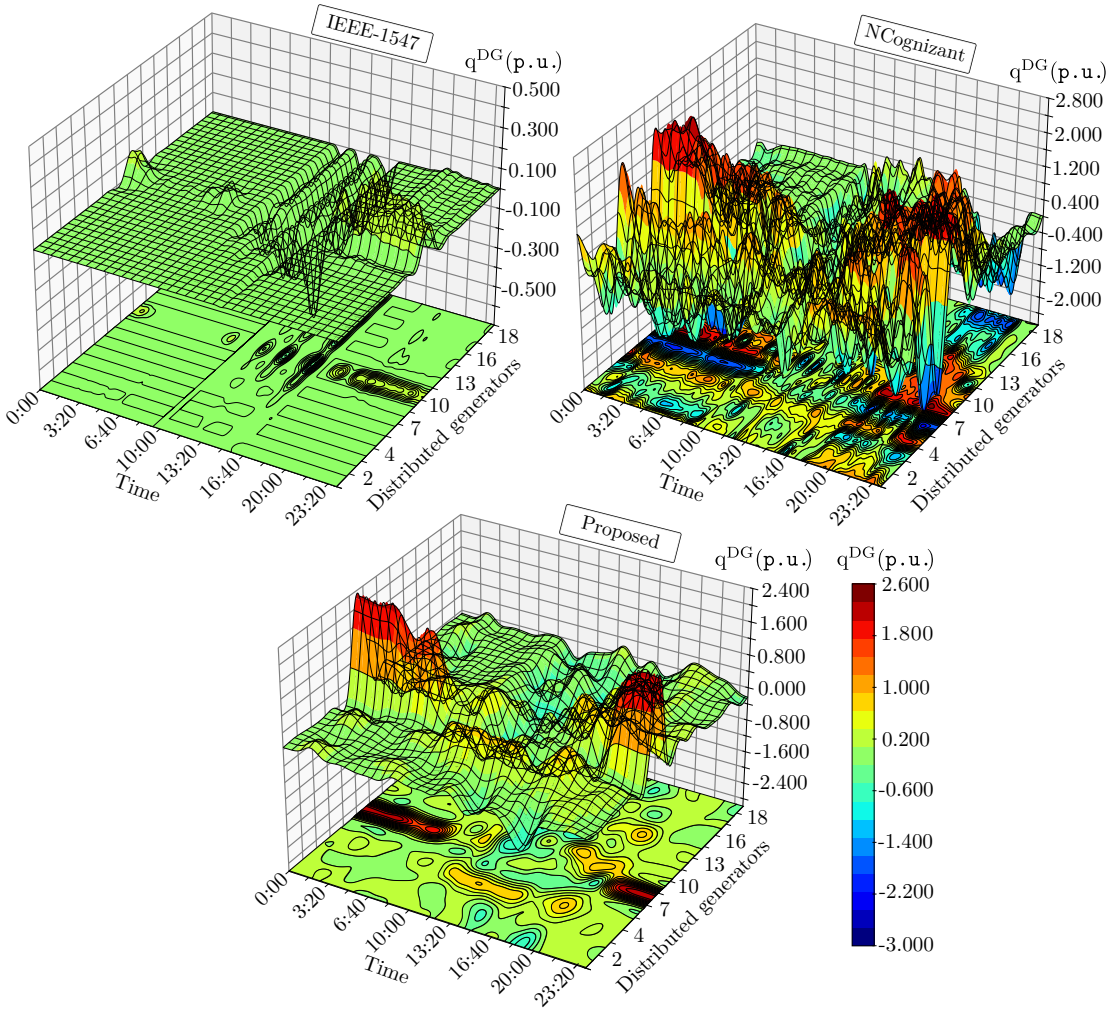


Figure 4.14: Reactive power dispatch comparison: literature models *vs.* proposed.

The reactive power dispatch surfaces for each control methodology exhibit erratically. In the IEEE-1547-based operating methodology, dispatch is governed by the converter's available power and local voltage. This approach limits reactive power utilization to minimize active power dumping, resulting in underuse. Conversely, both the literature model and the proposed method treat reactive power as a resource for system energy management. The NCognizant model employs robust optimization to define operating policies within a given window but lacks regularization for reactive power, constraining it only to the feasible region (3.3), leading to the irregular surface seen in the upper right. In contrast, the proposed model leverages a physics-based training algorithm to determine the operating policy. By incorporating multiple adjustment scenarios and agent-driven regularization mechanisms, it achieves a smoother response compared to its predecessors.

The analysis of voltage response, power dispatch, and system management under uncertainty clearly demonstrates the advantages of the proposed model over the IEEE-1547 standard and the NCognizant approach. While IEEE–1547 fails to mitigate voltage violations and NCognizant reduces but does not eliminate them, the proposed methodology maintains voltage within safe limits throughout the day. The dispatch surfaces further highlight this superiority, with the proposed model integrating voltage drop control and polynomial adjustments in the control structure, together with the physics-based training algorithm. These features not only smooth reactive power dispatch but also optimize active power utilization, balancing system stability and energy efficiency. Altogether, these results confirm that a proposed framework provides the most robust, reliable and adaptable solution for distributed generation control, making it the best option among the methodologies analyzed.

Chapter 5

Conclusions and future work

This document presents a new framework for the centralized definition of local operating policies of DGs based on the transfer of the underlying physics of the distribution system to an intelligent agent. Using a control structure present in the literature, the work focuses its efforts on the development of a methodological proposal for the design, integration and training of an intelligent agent that defines the operating parameters that achieve global optimality in local operation under uncertainty.

Unlike similar approaches that rely on optimization strategies to manage operational uncertainty, we have explored the integration of ML algorithms to characterize functional forms that provide real–time operational solutions. Specifically, given a control structure that enables centralized decision-making followed by local execution, we explicitly define the underlying monolithic optimization problem governing the system's optimal operation. This problem aims to minimize total energy consumption and voltage violations. Based on this formulation, we derive and reformulate the KKT optimality conditions to establish a unified function for agent adjustment, capturing the system's intrinsic behavior along with the impact of local control policies and inherent operational uncertainties. The operation of the system is organized under a two–layer control hierarchy: a real-time control layer responsible for evaluating local control policies in the order of 10–100 `ms`; and a centralized adaptability layer responsible for periodically defining the updated parameters of the local controllers by means of a previously trained intelligent agent. This operational framework avoids communication overload and, at the same time, takes advantage of the asynchronous acquisition of information to define local operational instructions that ensure the overall optimality of the system. For the design of the intelligent agent, we establish a set of criteria based on the causal structure and topology of the system. The agent framework is organized into three key tasks: coding, processing and definition of control parameters. In the coding stage, the agent derives latent variables based on system uncertainty and historical operating states. These latent variables are then processed in subsequent stages, processing their temporal characteristics to determine the control parameters that define each local controller.

Numerous computational experiments show that the proposed approach safely and trustworthily integrates an intelligent agent into the operation of a distribution system, considerably surpassing similar state–of–the–art operating frameworks. The algorithm is tested in three modified versions of the IEEE feeders in high solar penetration scenarios with almost 100% solar penetration. The results reveal several characteristics:

1. We have investigated the operating conditions that allow for the safe and reliable integration of artificial intelligence algorithms for the active management of a distribution network. In the context of high penetration of distributed photovoltaic generation, we were able to train and put into operation an intelligent agent that captures and transfers global optimality to local control set–points, which translates into effective voltage regulation with high economic efficiency compared to the models in the literature. From this assertion, we can conclude that the study presents a consistent framework for the integration of ML-based technologies in energy systems.

2. One of the main concerns regarding the applicability of ML-based methods in energy systems is the physical interpretability of the model, as this is linked to the reliability of the model's deployment in the real world. This issue is addressed exhaustively in the conceptualization stage. In it, we transfer characteristics of the topology of the system and the causal nature of the operation of the system into the structure and training algorithm of the intelligent agent, managing to capture the physics inherent in the operation of a distribution network. This assertion is supported by the analysis of the agent's sensitivity and convergence. In both studies, it is empirically demonstrated how systemic performance is linked to the performance of the metrics used for training, meaning that as the agent converges, systemic performance improves.

3. The out-of-sample simulation results indicate that the proposed approach surpasses comparable methods based on proportional control and OPF. Utilizing standard distribution system metrics, we show that the proposed methodology efficiently manages a distribution network with high renewable energy penetration while minimizing operating costs. This is accomplished by embedding the system's physics into the intelligent agent's design and training, ensuring regularized and effective control actions under uncertainty.

4. The sensitivity studies are limited to the IEEE 4–bus test feeder. This is due to the limited scalability of the problem, caused by the high computational load involved in training an intelligent agent under the adopted approach. This is partially overcome with warm start strategies between agents from different systems, achieving reductions of 40% in training time. Note that real–time operation is not affected as it uses a trained model.

Based on the experimental results and the aforementioned observations, future research will address the following issues:

1. Exploring the construction of new control policies based on families of functions for real-time control rules. Using procedure [58], the option of constructing control policies based on the adjustment of a regularized operating curve can be explored.

2. The evaluation of the risk introduced by the delays and failures of the communication network by means of the integration of the model in an operational environment of real-time simulation.

3. A solution approach that avoids scalability problems. The construction of a distributed multi–agent operation environment is proposed, which locally hosts intelligent agents for the local operation of DGs or microgrids.

4. Explore the incorporation of a more precise OPF operation problem. In the current application, the based model uses SOCP-OPF relaxations on the LDF model that has the potential to improve the solution without a dramatic impact on computational manageability.

5. Extend the training operation horizon to explicitly incorporate regularization mechanisms on the control parameters.

Bibliography

[1] Nasif Mahmud and A Zahedi. Review of control strategies for voltage regulation of the smart distribution network with high penetration of renewable distributed generation. *Renewable and Sustainable Energy Reviews*, 64:582–595, 2016.

[2] K Turitsyn, P Sulc, S Backhaus, and M Chertkov. Options for Control of Reactive Power by Distributed Photovoltaic Generators. *Proc. IEEE*, 99(6):1063–1073, jun 2011.

[3] Fernando Mancilla-David, Alejandro Angulo, and Alexandre Street. Power management in active distribution systems penetrated by photovoltaic inverters: A data-driven robust approach. *IEEE Trans. Smart Grid*, 11(3):2271–2280, 2019.

[4] Sarthak Gupta. *Machine Learning and Quantum Computing for Optimization Problems in Power Systems*. PhD thesis, Virginia Tech, 2023.

[5] Juan Sepúlveda, Alejandro Angulo, Fernando Mancilla-David, and Alexandre Street. Robust co-optimization of droop and affine policy parameters in active distribution systems with high penetration of photovoltaic generation. *IEEE Trans. Smart Grid*, 13(6):4355–4366, 2022.

[6] Kyri Baker, Andrey Bernstein, Emiliano Dall'Anese, and Changhong Zhao. Network-cognizant voltage droop control for distribution grids. *IEEE Trans. Power Syst.*, 33(2):2098–2108, 2017.

[7] Bin Zhang, Di Cao, Weihao Hu, Amer MYM Ghias, and Zhe Chen. Physics-informed multi-agent deep reinforcement learning enabled distributed voltage control for active distribution network using pv inverters. *International Journal of Electrical Power & Energy Systems*, 155:109641, 2024.

[8] IEEE Standards. IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces. *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pages 1–138, apr 2018.

[9] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[10] Ferdinando Fioretto, TW Mak, Federico Baldo, Michele Lombardi, and Pascal Van Hentenryck. A lagrangian dual framework for deep neural networks with constraints. *arXiv preprint arXiv:2001.09394*, 2020.

[11] Mudathir Funsho Akorede, Hashim Hizam, and Edris Pouresmaeil. Distributed energy resources and benefits to the environment. *Renewable and sustainable energy reviews*, 14(2):724–734, 2010.

[12] Ali Ehsan and Qiang Yang. State-of-the-art techniques for modelling of uncertainties in active distribution network planning: A review. *Applied energy*, 239:1509–1523, 2019.

[13] Domagoj Badanjak and Hrvoje Pandžić. Distribution-level flexibility markets—a review of trends, research projects, key stakeholders and open questions. *Energies*, 14(20):6622, 2021.

[14] Sulman Shahzad and Elżbieta Jasińska. Renewable revolution: A review of strategic flexibility in future power systems. *Sustainability*, 16(13):5454, 2024.

[15] Seyed-Ehsan Razavi, Ehsan Rahimi, Mohammad Sadegh Javadi, Ali Esmaeel Nezhad, Mohamed Lotfi, Miadreza Shafie-khah, and João PS Catalão. Impact of distributed generation on protection and voltage regulation of distribution systems: A review. *Renewable and Sustainable Energy Reviews*, 105:157–167, 2019.

[16] Hongbin Sun, Qinglai Guo, Junjian Qi, Venkataramana Ajjarapu, Richard Bravo, Joe Chow, Zhengshuo Li, Rohit Moghe, Ehsan Nasr-Azadani, Ujjwol Tamrakar, et al. Review of challenges and research opportunities for voltage control in smart grids. *IEEE Trans. Power Syst.*, 34(4):2790–2801, 2019.

[17] Morteza Aien, Ali Hajebrahimi, and Mahmud Fotuhi-Firuzabad. A comprehensive review on uncertainty modeling techniques in power system studies. *Renewable and Sustainable Energy Reviews*, 57:1077–1089, 2016.

[18] Mohammad Hemmati, Behnam Mohammadi-Ivatloo, and Alireza Soroudi. Chapter 2 - Uncertainty management in decision-making in power system operation. In Shady H E [Abdel Aleem], Almoataz Youssef Abdelaziz, Ahmed F Zobaa, and Ramesh Bansal, editors, *Decision Making Applications in Modern Power Systems*, pages 41–62. Academic Press, 2020.

[19] Rabih A Jabr. Segregated linear decision rules for inverter watt-var control. *IEEE Trans. Power Syst.*, 36(3):2702–2708, 2020.

[20] Ilgiz Murzakhanov, Sarthak Gupta, Spyros Chatzivasileiadis, and Vassilis Kekatos. Optimal design of volt/var control rules for inverter-interfaced distributed energy resources. *arXiv preprint arXiv:2210.12805*, 2022.

[21] Na Li, Guannan Qu, and Munther Dahleh. Real-time decentralized voltage control in distribution networks. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 582–588. IEEE, 2014.

[22] K Baker, A Bernstein, E Dallanese, and C Zhao. Network-Cognizant Voltage Droop Control for Distribution Grids. *IEEE Trans. Power Syst.*, 33(2):2098–2108, mar 2018.

[23] F Mancilla-David, A Angulo, and A Street. Power Management in Active Distribution Systems Penetrated by Photovoltaic Inverters: A Data-Driven Robust Approach. *IEEE Trans. Smart Grid*, page 1, 2019.

[24] Edward J Smith, Duane A Robinson, and Sean Elphick. Der control and management strategies for distribution networks: A review of current practices and future directions. *Energies*, 17(11):2636, 2024.

[25] John W Simpson-Porco, Florian Dörfler, and Francesco Bullo. Voltage stabilization in microgrids via quadratic droop control. *IEEE Trans. Autom. Control*, 62(3):1239–1253, 2016.

[26] Allen J Wood, Bruce F Wollenberg, and Gerald B Sheblé. *Power generation, operation, and control*. John Wiley & Sons, 2013.

[27] Josep M Guerrero, Juan C Vasquez, José Matas, Luis García De Vicuña, and Miguel Castilla. Hierarchical control of droop-controlled ac and dc microgrids—a general approach toward standardization. *IEEE Trans. Ind. Electron.*, 58(1):158–172, 2010.

[28] Masoud Farivar, Lijun Chen, and Steven Low. Equilibrium and dynamics of local voltage control in distribution systems. In *52nd IEEE Conference on Decision and Control*, pages 4329–4334. IEEE, 2013.

[29] Xinyang Zhou, Masoud Farivar, Zhiyuan Liu, Lijun Chen, and Steven H Low. Reverse and forward engineering of local voltage control in distribution networks. *IEEE Trans. Autom. Control*, 66(3):1116–1128, 2020.

[30] Vito Calderaro, Gaspare Conio, Vincenzo Galdi, Giovanni Massa, and Antonio Piccolo. Optimal decentralized voltage control for distribution systems with inverter-based distributed generators. *IEEE Trans. Power Syst.*, 29(1):230–241, 2013.

[31] Afshin Samadi, Robert Eriksson, Lennart Söder, Barry G Rawn, and Jens C Boemer. Coordinated active power-dependent voltage regulation in distribution grids with pv systems. *IEEE Trans. Power Del.*, 29(3):1454–1464, 2014.

[32] Ankit Singhal, Venkataramana Ajjarapu, Jason Fuller, and Jacob Hansen. Real-time local volt/var control under external disturbances with high pv penetration. *IEEE Trans. Smart Grid*, 10(4):3849–3859, 2018.

[33] Ilgiz Murzakhanov, Sarthak Gupta, Spyros Chatzivasileiadis, and Vassilis Kekatos. Optimal design of volt/var control rules for inverter-interfaced distributed energy resources. *IEEE Trans. Smart Grid*, 2023.

[34] Yincheng Zhao, Guozhou Zhang, Weihao Hu, Qi Huang, Zhe Chen, and Frede Blaabjerg. Meta-learning based voltage control strategy for emergency faults of active distribution networks. *Applied Energy*, 349:121399, 2023.

[35] Friederike Wenderoth, Elisabeth Drayer, Robert Schmoll, Michael Niedermeier, and Martin Braun. Architectural and functional classification of smart grid solutions. *Energy Informatics*, 2:1–13, 2019.

[36] Ting Yang. Ict technologies standards and protocols for active distribution network. In *Smart power distribution systems*, pages 205–230. Elsevier, 2019.

[37] J Lavaei and S H Low. Zero Duality Gap in Optimal Power Flow Problem. *IEEE Trans. Power Syst.*, 27(1):92–107, feb 2012.

[38] B. Stott and O. Alsac. Fast decoupled load flow. *IEEE Trans. Power App. Syst.*, PAS-93(3):859–869, May 1974.

[39] M E Baran and F F Wu. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Trans. Power Del.*, 4(2):1401–1407, 1989.

[40] A Bernstein, C Wang, E Dall'Anese, J Le Boudec, and C Zhao. Load Flow in Multiphase Distribution Networks: Existence, Uniqueness, Non-Singularity and Linear Models. *IEEE Trans. Power Syst.*, 33(6):5832–5843, nov 2018.

[41] R A Jabr. Radial distribution load flow using conic programming. *IEEE Trans. Power Syst.*, 21(3):1458–1459, aug 2006.

[42] Xiaoqing Bai, Hua Wei, Katsuki Fujisawa, and Yong Wang. Semidefinite programming for optimal power flow problems. *International Journal of Electrical Power & Energy Systems*, 30(6):383–392, 2008.

[43] S H Low. Convex Relaxation of Optimal Power Flow—Part I: Formulations and Equivalence. *IEEE Trans. Control Netw. Syst.*, 1(1):15–27, mar 2014.

[44] Bose Subhonmesh, Steven H Low, and K Mani Chandy. Equivalence of branch flow and bus injection models. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1893–1899. IEEE, 2012.

[45] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, USA, 2004.

[46] S Bose, D F Gayme, K M Chandy, and S H Low. Quadratically Constrained Quadratic Programs on Acyclic Graphs With Application to Power Flow. *IEEE Trans. Control Netw. Syst.*, 2(3):278–287, 2015.

[47] L Gan, N Li, U Topcu, and S H Low. Exact Convex Relaxation of Optimal Power Flow in Radial Networks. *IEEE Trans. Autom. Control*, 60(1):72–87, jan 2015.

[48] M.E. Baran and F.F. Wu. Optimal capacitor placement on radial distribution systems. *IEEE Trans. Power Del.*, 4(1):725–734, 1989.

[49] Miguel Huerta, Orlando Gonzalez, Alejandro Angulo, and Fernando Mancilla-David. An active distribution network model including legacy and modern equipment: relaxations and approximations under optimal management. *NAPS 2024: 56th North American Power Symposium*, 2024.

[50] Xinyang Zhou, Lijun Chen, Masoud Farivar, Zhiyuan Liu, and Steven Low. Reverse and Forward Engineering of Local Voltage Control in Distribution Networks. *IEEE Trans. Autom. Control*, 2020.

[51] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[53] OpenAI. Gpt-4 technical report, 2023.

[54] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[55] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *arXiv preprint arXiv:1711.10561*, 2017.

[56] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

[57] Raban Iten, Tony Metger, Henrik Wilming, Lídia Del Rio, and Renato Renner. Discovering physical concepts with neural networks. *Physical review letters*, 124(1):010508, 2020.

[58] Alejandro Angulo, Miguel Huerta, and Fernando Mancilla-David. A neural network-aided functional model of photovoltaic arrays for a wide range of atmospheric conditions. *IEEE Trans. Ind. Informat.*, 20(2):2487–2496, 2024.

[59] Vinod Nair, Sergey Bartunov, Felix Gimeno, Ingrid Von Glehn, Pawel Lichocki, Ivan Lobov, Brendan O'Donoghue, Nicolas Sonnerat, Christian Tjandraatmadja, Pengming Wang, et al. Solving mixed integer programs using neural networks. *arXiv preprint arXiv:2012.13349*, 2020.

[60] Ferdinando Fioretto, Terrence W.K. Mak, and Pascal Van Hentenryck. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):630–637, Apr. 2020.

[61] Ivan Sosnovik and Ivan Oseledets. Neural networks for topology optimization. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 34(4):215–223, 2019.

[62] Ahmed S Zamzam and Kyri Baker. Learning optimal solutions for extremely fast ac optimal power flow. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2020.

[63] Kyri Baker. Emulating ac opf solvers for obtaining sub-second feasible, near-optimal solutions. *arXiv preprint arXiv:2012.10031*, 2020.

[64] Wanjun Huang, Xiang Pan, Minghua Chen, and Steven H. Low. Deepopf-v: Solving ac-opf problems efficiently. *IEEE Trans. Power Syst.*, 37(1):800–803, 2022.

[65] Stefano Markidis. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Frontiers in big Data*, page 92, 2021.

[66] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

[67] Maad M Mijwel. Artificial neural networks advantages and disadvantages. *Mesopotamian Journal of Big Data*, 2021:29–31, 2021.

[68] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

[69] Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760, 2021.

[70] Anders Krogh. What are artificial neural networks? *Nature biotechnology*, 26(2):195–197, 2008.

[71] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[73] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. Convolutional channel features. In *Proceedings of the IEEE international conference on computer vision*, pages 82–90, 2015.

[74] Md Amirul Islam, Matthew Kowal, Sen Jia, Konstantinos G Derpanis, and Neil DB Bruce. Position, padding and predictions: A deeper look at position information in cnns. *International Journal of Computer Vision*, pages 1–22, 2024.

[75] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[76] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[77] Yujie Liu, Hongbin Dong, Xingmei Wang, and Shuang Han. Time series prediction based on temporal convolutional network. In *2019 IEEE/ACIS 18th International conference on computer and information science (ICIS)*, pages 300–305. IEEE, 2019.

[78] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[79] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, pages 1–26, 2020.

[80] Chuizheng Meng, Sungyong Seo, Defu Cao, Sam Griesemer, and Yan Liu. When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797*, 2022.

[81] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Regularization for deep learning. *Deep learning*, pages 216–261, 2016.

[82] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pages 270–279. Springer, 2018.

[83] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1(1):1–34, 2020.

[84] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[85] Balthazar Donon, Benjamin Donnot, Isabelle Guyon, and Antoine Marot. Graph neural solver for power systems. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[86] Liang Zhang, Gang Wang, and Georgios B Giannakis. Real-time power system state estimation and forecasting via deep unrolled neural networks. *IEEE Trans. Signal Process.*, 67(15):4069–4077, 2019.

[87] Valentin Bolz, Johannes Rueß, and Andreas Zell. Power flow approximation based on graph convolutional networks. In *2019 18th ieee international conference on machine learning and applications (icmla)*, pages 1679–1686. IEEE, 2019.

[88] Balthazar Donon, Rémy Clément, Benjamin Donnot, Antoine Marot, Isabelle Guyon, and Marc Schoenauer. Neural networks for power flow: Graph neural solver. *Electric Power Systems Research*, 189:106547, 2020.

[89] Rahul Nellikkath and Spyros Chatzivasileiadis. Physics-informed neural networks for ac optimal power flow. *Electric Power Systems Research*, 212:108412, 2022.

[90] Ling Zhang, Yize Chen, and Baosen Zhang. A convex neural network solver for dcopf with generalization guarantees. *IEEE Trans. Control Netw. Syst.*, 9(2):719–730, 2021.

[91] Manish K Singh, Sarthak Gupta, Vassilis Kekatos, Guido Cavraro, and Andrey Bernstein. Learning to optimize power distribution grids using sensitivity-informed deep neural networks. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE, 2020.

[92] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[93] Bin Huang and Jianhui Wang. Applications of physics-informed neural networks in power systems-a review. *IEEE Trans. Power Syst.*, 2022.

[94] Spyros Chatzivasileiadis, Andreas Venzke, Jochen Stiasny, and Georgios Misyris. Machine learning in power systems: Is it time to trust it? *IEEE Power and Energy Magazine*, 20(3):32–41, 2022.

[95] Valentin Frank Ingmar Guenter and Athanasios Sideris. Robust learning of parsimonious deep neural networks. *Neurocomputing*, 566:127011, 2024.

[96] Saaketh Desai and Alejandro Strachan. Parsimonious neural networks learn interpretable physical laws. *Scientific reports*, 11(1):12761, 2021.

[97] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[98] Asmaul Hosna, Ethel Merry, Jigmey Gyalmo, Zulfikar Alom, Zeyar Aung, and Mohammad Abdul Azim. Transfer learning: a friendly introduction. *Journal of Big Data*, 9(1):102, 2022.

[99] Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[100] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, 2020.

[101] Huanru Henry Mao. A survey on self-supervised pre-training for sequential transfer learning in neural networks. *arXiv preprint arXiv:2007.00800*, 2020.

[102] Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

[103] Diego Klabjan and Xiaofeng Zhu. Neural network retraining for model serving. *arXiv preprint arXiv:2004.14203*, 2020.

[104] Murat Kuzlu and Manisa Pipattanasomporn. Assessment of communication technologies and network requirements for different smart grid applications. In *2013 IEEE PES innovative smart grid technologies conference (ISGT)*, pages 1–6. IEEE, 2013.

[105] Goodwin GC, Graebe SF, and Mario Salgado. *Control System Design*. Prentice Hall, 2001.

[106] Kumar Shivam, Jong-Chyuan Tzou, and Shang-Chen Wu. Multi-step short-term wind speed prediction using a residual dilated causal convolutional network with nonlinear attention. *Energies*, 13(7):1772, 2020.

[107] William H Kersting. *Distribution System Modeling and Analysis, Third Edition*. CRC Press, Taylor & Francis Group [Distributor, 3rd ed.,revised edition, Jan. 2012.

[108] Kejun Qian, Chengke Zhou, Malcolm Allan, and Yue Yuan. Modeling of load demand due to ev battery charging in distribution systems. *IEEE Trans. Power Syst.*, 26(2):802–810, 2010.

[109] Daniel Berrar et al. Cross-validation., 2019.

[110] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[111] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv e-prints*, page arXiv:1804.07612, Apr. 2018.

[112] Miguel Huerta, Alejandro Angulo, Juan Sepúlveda, and Fernando Mancilla-David. An artificial neural network approach to affine policy definition in active distribution networks. In *2024 IEEE 42th Central America and Panama Convention (CONCAPAN)*, pages 1–6. IEEE, 2025.

[113] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, Dec. 2014.

[114] Matthew D. Zeiler. ADADELTA: An adaptive learning rate method, 2012.

[115] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

[116] Tong Yu and Hong Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020.

[117] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.