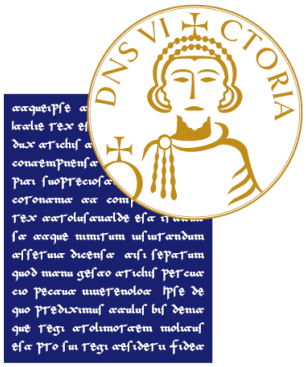
University of Sannio

Department of Engineering

Master's Degree in

**Electronics Engineering for Automation and Sensing**



# Deep Reinforcement Learning for Adaptive Bidirectional Electric Vehicle Charging Management (Vehicle-to-Grid)

**Supervisor:**
Prof. Carmela Bernardo

**Co-Supervisor:**
Dr. Antonio Pepiciello

**Candidate:**
Angelo Caravella
Student ID 389000016

Academic Year 2024–2025

# Contents

# Abstract in italian

L'adozione crescente dei **Veicoli Elettrici (EV)** in concomitanza con la sempre maggiore penetrazione di **Fonti di Energia Rinnovabile (RES)** intermittenti, presenta sfide significative alla **stabilità** e all'**efficienza della rete elettrica**. La tecnologia **Vehicle-to-Grid (V2G)** emerge come soluzione fondamentale, trasformando gli EV da carichi passivi a **risorse energetiche flessibili** capaci di fornire vari **servizi di rete**. Questa tesi affronta il complesso **problema di ottimizzazione multi-obiettivo** della gestione intelligente di carica e scarica degli EV, che intrinsecamente implica un equilibrio tra **benefici economici**, **esigenze di mobilità dell'utente**, **preservazione della salute della batteria** e **stabilità della rete** in condizioni stocastiche.

Di fronte alla complessa sfida di ottimizzare la ricarica dei veicoli elettrici (EV) in scenari Vehicle-to-Grid (V2G), un approccio che si limita a un singolo modello di controllo, come il Deep Q-Networks (DQN), risulterebbe inadeguato. La natura del problema, caratterizzata da molteplici obiettivi contrastanti (benefici economici, esigenze dell'utente, salute della batteria, stabilità della rete) e da una profonda incertezza; richiede un'analisi comparativa e rigorosa di un'ampia gamma di strategie di controllo. Per questo motivo, la ricerca si concentra sulla valutazione di un portafoglio diversificato di algoritmi, che include numerosi modelli di Deep Reinforcement Learning (DRL), approcci euristici e il Model Predictive Control (MPC). Questo metodo consente di mappare in modo completo il panorama delle soluzioni, identificando i punti di forza e di debolezza di ciascun approccio in relazione alle diverse sfaccettature del problema V2G.

In conclusione questa lavoro di tesi non si focalizza su un singolo modello, ma adotta un approccio comparativo su larga scala perché:

**Non esiste una soluzione unica**: La complessità del problema V2G rende improbabile che un solo algoritmo sia ottimale in tutte le condizioni.

**Si ricercano i compromessi**: L'obiettivo è comprendere i trade-off tra l'efficienza dei dati, la stabilità dell'addestramento, la robustezza all'incertezza e la complessità computazionale delle diverse famiglie di algoritmi.

**La validazione è più rigorosa**: Confrontare i modelli di DRL non solo tra loro ma anche con benchmark consolidati come le euristiche e l'MPC fornisce una misura molto più credibile del loro reale valore aggiunto.

# Abstract

The growing adoption of **Electric Vehicles (EVs)** in embracing with the ever-increasing incursion of sporadic **Renewable Energy Sources (RES)** presents substantial challenges to the **stability** and **efficiency** of the power grid. **Vehicle-to-Grid (V2G)** technology emerges as a key solution, transforming EVs from passive loads to **flexible energy resources** subject of providing assorted **grid services**. This thesis addresses the composite **multi-objective optimization problem** of smart EV charging and discharging management, which inherently involves a trade-off between **economic benefits**, **user mobility needs**, **battery health preservation**, and **grid stability** under stochastic conditions.
Faced with the complex challenge of optimizing electric vehicle (EV) charging in Vehicle-to-Grid (V2G) scenarios, an approach limited to a single control model, such as Deep Q-Networks (DQN), would be insubstantial. The nature of the problem, characterized by multiple running afoul objectives (economic benefits, user needs, battery health, grid stability) and profound uncertainty, requires a rigorous comparative analysis of a all-embracing range of control strategies.
For this argue, the research focuses on appraising a diverse portfolio of algorithms, including numerous Deep Reinforcement Learning (DRL) models, heuristic approaches, and Model Predictive Control (MPC). This method allows for a utter mapping of the solution landscape, identifying the strengths and weaknesses of each approach in relation to the different facets of the V2G problem.
Briefly, this thesis does not concentrate on a single paradigm, but embraces a **broad-spectrum comparative perspective** because:
**There is no universal remedy**: The intricacy of the V2G challenge makes it improbable that one algorithm will prove superior across all circumstances.
**We pursue equilibria**: The objective is to unveil the balances between data thriftiness, learning steadiness, resilience to unpredictability, and computational burden across diverse algorithmic families.
**Assessment is more stringent**: Juxtaposing DRL frameworks not only among themselves but also against established references such as heuristics and MPC yields a far more trustworthy appraisal of their genuine incremental merit.

# List of Acronyms

| Acronym | Description |
|---|---|
| **Artificial Intelligence & Control** | |
| A2C | Advantage Actor-Critic |
| AC | Actor-Critic |
| AI | Artificial Intelligence |
| AL-SAC | Augmented Lagrangian Soft Actor-Critic |
| ARS | Augmented Random Search |
| CL | Curriculum Learning |
| CMDP | Constrained Markov Decision Process |

| Acronym | Description |
|---|---|
| DDPG | Deep Deterministic Policy Gradient |
| DQN | Deep Q-Networks |
| DRL | Deep Reinforcement Learning |
| LQR | Linear Quadratic Regulator |
| LSTM | Long Short-Term Memory |
| MARL | Multi-Agent Reinforcement Learning |
| MDP | Markov Decision Process |
| MILP | Mixed-Integer Linear Program |
| MPC | Model Predictive Control |
| NN | Neural Network |
| PER | Prioritized Experience Replay |
| PPO | Proximal Policy Optimization |
| RL | Reinforcement Learning |
| SAC | Soft Actor-Critic |
| TD3 | Twin-Delayed Deep Deterministic Policy Gradient |
| TQC | Truncated Quantile Critics |
| TRPO | Trust Region Policy Optimization |
| **Electric Vehicles & Charging** | |
| AFAP | As Fast As Possible (Heuristic) |
| ALAP | As Late As Possible (Heuristic) |
| CAFA | Charge As Fast As Possible |
| CALA | Charge As Late As Possible |
| CPO | Charge Point Operator |
| EV | Electric Vehicle |
| G2V | Grid-to-Vehicle |
| SCP | Scheduled Charging Power |
| SoC | State of Charge |
| SoH | State of Health |
| V2B | Vehicle-to-Building |
| V2G | Vehicle-to-Grid |
| V2H | Vehicle-to-Home |
| V2M | Vehicle-to-Microgrid |
| V2V | Vehicle-to-Vehicle |
| VPP | Virtual Power Plant |
| **Power Grid & Energy Markets** | |
| ACE | Area Control Error |
| ARR | Area Regulation Requirement |
| DER | Distributed Energy Resources |
| DR | Demand Response |
| RES | Renewable Energy Sources |
| **Metrics & Technical Parameters** | |
| DC | Constant Current (charging phase) |
| CV | Constant Voltage (charging phase) |
| DoD | Depth of Discharge |
| MSE | Mean Square Error |

| Acronym | Description |
|---|---|
| OU | Ornstein-Uhlenbeck (stochastic process) |
| RMSE | Root Mean Square Error |

# Chapter 1

# Introduction

The shift toward electric mobility constitutes a pivotal element in worldwide strategies for the decarbonization of transportation; nevertheless, the widescale incorporation of electric vehicles into existing power networks introduces a multifaceted spectrum of hurdles and prospects that this thesis seeks to investigate.

### 1.0.1 Background and Relevance of Electric Vehicles and Vehicle-to-Grid

The surge of the Electric Vehicle (EV) market is accelerating a profound reconfiguration of modern mobility, with the promise of lowering carbon emissions while fostering greater energy efficiency[1]. This evolution is more than a technological trend: it underpins environmental sustainability by reducing dependence on fossil resources, alleviating the impacts of climate change through diminished greenhouse gas emissions, and improving air quality in densely populated areas. Yet, embedding millions of EVs into existing power systems is far from trivial. It can intensify peak demand, place additional stress on transmission and distribution networks, and trigger side effects such as voltage irregularities or higher line losses[2].
In this context, the **Vehicle-to-Grid (V2G)** concept emerges as a forward-looking and strategic pathway. Through bidirectional power exchange, V2G redefines EVs: no longer passive electrical loads, but mobile and flexible energy assets, able to deliver a spectrum of services to the power system[3]. This potential becomes even more compelling when one considers that, on average, EVs remain parked and unused for nearly 96% of the day, offering an ample time window to actively engage with the grid[4]. A further distinctive benefit lies in the rapid responsiveness of EV batteries, which makes them especially suitable for ancillary services demanding quick interventions, such as frequency regulation[5]. Alongside V2G, other schemes of bidirectional power flow have been proposed, each with its own scope:

1. **Vehicle-to-Home (V2H)**, where an EV sustains household demand during

[1] **orfanoudakis2022deep**.
[2] **orfanoudakis2022deep**; **salvatti2020electric**.
[3] **alfaverh2022optimal**.
[4] **evertsson2024investigating**.
[5] **alfaverh2022optimal**.

outages or periods of elevated prices, strengthening domestic energy resilience;

2. **Vehicle-to-Building (V2B)**, extending this logic to commercial or industrial facilities, enabling EVs to support load management and improve consumption efficiency; and

3. **Vehicle-to-Vehicle (V2V)**, which allows direct power transfer among EVs, a valuable feature for emergency charging or shared resources.

Taken together, these modalities highlight the versatility of EV batteries as distributed energy units, reinforcing both energy resilience and the transition toward a more sustainable energy ecosystem.

### 1.0.2 Challenges in EV Integration into the Electricity Grid and the Role of Artificial Intelligence

Modern electricity systems are increasingly shaped by the penetration of intermittent **Renewable Energy Sources (RESs)** such as wind and solar. Their variability generates pronounced swings in output and persistent mismatches between supply and demand, fuelling price volatility and complicating dispatch strategies. As a consequence, the stability and economic efficiency of the grid are continuously put under strain. Managing these fluctuations, while making rapid operational choices to balance the system and minimize costs, has proven difficult for conventional control frameworks[6].

The parallel rise of EV adoption and RES deployment has produced an environment marked by both uncertainty and complexity. In such conditions, traditional approaches are increasingly inadequate, prompting a growing reliance on methods rooted in artificial intelligence—and particularly in **Reinforcement Learning (RL)**. This shift alters the very nature of the grid: from a relatively predictable and centralized infrastructure to one that is decentralized, stochastic, and highly dynamic. Rule-based or deterministic controllers, designed for a past paradigm, are ill-suited to cope with this degree of volatility. The outcome is a pressing demand for adaptive and intelligent decision-making mechanisms. This transformation extends beyond the simple challenge of absorbing extra load or integrating new generators: it signals a genuine paradigm change towards a *smart grid*[7], where adaptive, real-time, and autonomous operation is no longer optional but vital to preserve efficiency, resilience, and reliability. In this light, RL appears not merely as a tool for optimization, but as an enabling technology for a cognitive and robust energy infrastructure, capable of navigating the uncertainties inherent in a decarbonized, electrified future. Against this backdrop, **Deep Reinforcement Learning (DRL)** has gained attention as an especially powerful approach. Its capacity to derive near-optimal strategies in dynamic and uncertain environments—without requiring a precise model of the system or flawless forecasts—makes DRL particularly well-suited for EV integration and advanced grid management[8].

[6] **orfanoudakis2022deep**; **minchala2025systematic**.

[7] **alhmoud2024review**.

[8] **orfanoudakis2022deep**; **shibl2023electric**.

### 1.0.3 Objectives and Contributions of the Thesis

This thesis addresses the complex multi-objective optimization problem inherent in Vehicle-to-Grid (V2G) systems. The overarching objective is to move beyond a purely theoretical analysis by actively developing, testing, and enhancing a high-fidelity simulation architecture. This platform serves as a digital twin to rigorously evaluate and compare advanced control strategies, balancing economic benefits, user mobility needs, battery health, and grid stability under realistic stochastic conditions.
More than a simple review of existing literature, this work focuses on the practical implementation and validation of a V2G simulation framework in Python. This tool is leveraged to demonstrate and explore novel perspectives for training intelligent agents. The main contributions are:

- **Enhancement of a V2G Simulation Architecture:** A significant contribution lies in the systematic testing, validation, and enhancement of the **EV2Gym** simulation framework. This work solidifies its role as a robust and flexible platform for benchmarking control algorithms, ensuring that the models for battery physics, user behavior, and grid dynamics are coherent and realistic for advanced research.

- **Exploration of Novel Reinforcement Learning Perspectives:** The validated simulation environment is used to investigate and implement advanced training methodologies for RL agents. A key focus is placed on techniques like **adaptive reward shaping**, where the reward function dynamically evolves during training to guide the agent towards a more holistic and robust control policy, overcoming the limitations of static reward definitions.

- **Practical Implementation of Advanced Control Paradigms:** The thesis demonstrates the practical transition from a theoretical, offline optimal controller to a realistic, online controller. Specifically, it details the implementation of an **offline MPC using Gurobi**, which acts as a "judge" with perfect foresight, and contrasts it with an **online MPC formulated in PuLP**, designed to operate as a real-time "controller" with limited future information, highlighting the trade-offs and challenges of real-world deployment.

### 1.0.4 Thesis Structure

The remainder of this thesis is organized as follows:

- **Chapter 2: Overview of Optimal Management of EV Charging and Discharging** provides foundational knowledge on V2G technology, the complex multi-objective nature of EV charging optimization, and presents a comprehensive review of state-of-the-art research approaches.
- **Chapter 3: The V2G Simulation Framework: A Digital Twin for V2G Research** details the architecture and core models of the simulation environment. This chapter describes the enhancements made to the framework, establishing it as the central experimental platform for implementing and evaluating the control agents analyzed in this work.
- **Chapter 4: Experimental Campaign and Results Analysis** This chapter presents the results of the comparative analysis between the different control strategies (DRL, MPC, heuristics). It analyzes the performance of novel training techniques and discusses the implications of the findings.
- **Bibliography** lists all cited references.

# Chapter 2

# State of the Art in Optimal V2G Management

## 2.1 The V2G Imperative: A Cornerstone of Europe's Green Transition

Our society stands at a critical juncture, facing the twin revolutions of decarbonizing transport and transforming our energy systems. This is not merely an ambition but a legally binding mandate, enshrined in frameworks like the **European Green Deal** and its ambitious **"Fit for 55"** package[1]. These policies impose a rapid phase-out of internal combustion engines and mandate a massive scale-up of renewable energy sources, as detailed in the revised Renewable Energy Directive (RED III)[2]. The proliferation of Electric Vehicles (EVs) sits squarely at the nexus of this challenge. Initially viewed with apprehension—a looming threat of massive, synchronized loads poised to destabilize fragile distribution networks—that perception is now obsolete. Today, we must see EVs not as a problem, but as a foundational pillar of the solution. This paradigm shift is embodied in the concept of **Vehicle-to-Grid (V2G)**. V2G is the critical enabling technology that transforms millions of EVs from passive energy consumers into an active, distributed, and intelligent grid asset. The key lies hidden in plain sight: private vehicles remain parked and connected for an astonishing 96% of their existence[3], representing a potential of terawatt-hours of mobile storage waiting to be harnessed.

The true power of V2G is not in the individual, but in the collective. A single EV's contribution is a whisper, but a coordinated fleet, managed by an aggregator, becomes a roar—a **Virtual Power Plant (VPP)**. This collective entity, with the lightning-fast response of battery inverters, can deliver a spectrum of critical services. This capability is the linchpin for stabilizing a grid increasingly reliant on the fluctuating whims of wind and sun, making the high renewable penetration targets of the EU feasible. The services enabled are foundational to the smart, resilient grid of tomorrow:

[1] **european_commission_2021_fit_for_55**.
[2] **RED_III_directive_2023**.
[3] **evertsson2024investigating**.

- **Frequency Regulation:** The grid's heartbeat. V2G fleets can inject or absorb power in seconds, instantly counteracting supply-demand imbalances to maintain the stable 50/60 Hz frequency, preventing cascading failures and blackouts[4].

- **Demand Response and Peak Shaving:** By intelligently shifting charging to off-peak hours and discharging during peak demand, V2G flattens the load curve. This reduces our reliance on expensive and polluting "peaker" plants and can defer trillions in grid infrastructure upgrades[5].

- **Renewable Energy Integration:** Perhaps the most profound impact. V2G fleets act as a giant, distributed sponge, absorbing surplus solar and wind energy that would otherwise be curtailed and wasted, and releasing it when the sun sets or the wind dies down. This directly supports the integration goals of RED III and mitigates intermittency[6].

This vision is no longer a distant prospect but is actively being codified into European law and technical standards. The landmark **Alternative Fuels Infrastructure Regulation (AFIR, EU 2023/1804)** now mandates that new public charging infrastructure must support smart and bidirectional charging capabilities. This legal requirement is given its technical teeth by specific standards; a delegated regulation specifies that from 2027, charging points must comply with **ISO 15118-20**, a standard that explicitly defines the communication protocols for bidirectional power transfer. This regulatory push is complemented by large-scale pilot projects like **'SCALE'** and **'V2G Balearic Islands'**, which are testing the technology's technical and economic viability on an industrial scale.
However, while the regulatory foundation is being laid, significant barriers to widespread adoption remain, creating a complex landscape that technology and policy must navigate together. Key challenges include:

- **Market and Economic Hurdles:** A clear, pan-European framework for remunerating EV owners for grid services is still absent. Critical issues like the **"double taxation"** of electricity—taxed both on charging and discharging—create significant economic disincentives and must be resolved.

- **Regulatory and Grid Access Rules:** The role of EV fleets as a flexibility resource is not yet uniformly recognized in electricity markets. Standardized procedures for grid connection, aggregator certification, and secure data exchange are still under development, hindering market access.

- **Technical and Consumer Barriers:** On the consumer side, concerns about accelerated **battery degradation** and its impact on vehicle warranties remain a primary obstacle. Furthermore, the reality is that not all EVs or chargers are currently equipped with the necessary hardware and software to support V2G.

[4] **alfaverh2022optimal**; **sadeghi2021deep**.
[5] **orfanoudakis2022deep**.
[6] **khan2024review**; **zou2021deep**.

Therefore, the central challenge—and the focus of this thesis—is not merely to enable V2G, but to do so *intelligently*. It requires a control strategy sophisticated enough to operate within this nascent regulatory framework, navigate its economic uncertainties, and overcome technical constraints to unlock the immense potential of EVs as a cornerstone of a sustainable energy future.

## 2.2 The Optimizer's Trilemma: Navigating a Stochastic World

While the potential is immense, orchestrating this symphony of distributed assets is a formidable challenge. The primary driver for an aggregator is economic viability, but pursuing profit in isolation is a recipe for failure. Optimal V2G management is a delicate balancing act, a genuine multi-objective optimization problem often framed as the "V2G trilemma": the simultaneous pursuit of **economic profitability**, the preservation of **battery longevity**, and the guarantee of **user convenience**.
This is not a simple trade-off. It is a dynamic problem steeped in **stochasticity** and **uncertainty** from multiple sources:

- **Market Volatility:** Electricity prices can fluctuate wildly based on unpredictable supply and demand.
- **Renewable Intermittency:** The output of solar and wind generation is inherently variable.
- **Human Behavior:** EV owners' arrival times, departure times, and energy needs are not deterministic; a driver might need to leave unexpectedly, a non-negotiable constraint that any intelligent system must respect.

This chaotic environment renders static, rule-based control systems obsolete. We need an approach that can learn, adapt, and make intelligent decisions in real-time under profound uncertainty. This is precisely the domain of Reinforcement Learning.

## 2.3 A New Paradigm for Control: Reinforcement Learning

To tackle the V2G challenge, we turn to Reinforcement Learning (RL), a field of machine learning concerned with how an intelligent agent learns to make optimal decisions through trial and error. Unlike traditional methods that require a perfect model of the world, RL learns directly from interaction, making it exceptionally robust.

### 2.3.1 The Language of Learning: Markov Decision Processes (MDPs)

The mathematical foundation of RL is the **Markov Decision Process (MDP)**, formally defined by the tuple $(S, A, p, R, \gamma)$. In the V2G context:

- $S$ is the state (a snapshot of the world: battery levels, electricity price, time).
- $A$ is the action (the decision: the charging/discharging rate for each EV).
- $p(s', r|s, a)$ is the environment's response (the probability of transitioning to a new state $s'$ and receiving reward $r$).
- $R$ is the reward (the feedback signal: profit generated, penalty for user dissatisfaction).
- $\gamma$ is the discount factor, balancing immediate vs. future rewards.

This framework rests on the **Markov Property**, which allows the agent to make decisions based solely on the current state.

### 2.3.2 Judging the Future: Value Functions and Actor-Critic Architectures

The agent's goal is to learn a **policy**, $\pi(a|s)$, a strategy for choosing actions. To do this, it learns **value functions**, which estimate the long-term value of being in a certain state ($v_\pi(s)$) or taking a specific action in a state ($q_\pi(s, a)$).
The **Actor-Critic** architecture provides an elegant way to learn the policy. It maintains two distinct components:

- **The Critic**: It learns the value function. Its job is to evaluate the actor's decisions.
- **The Actor**: It is the policy. Its job is to select actions, using the critic's feedback to improve its strategy over time.

This architecture is particularly powerful for V2G because it can directly learn a policy over a continuous action space, allowing for precise control of power. The agent's entire behavior, however, is shaped by the reward signal it receives. The complex art of designing this signal to align the agent's goals with our multi-faceted objectives is a critical discipline in itself, known as reward engineering.

### 2.3.3 Advanced Reward Engineering

Within a Reinforcement Learning (RL) framework, the architecture of the reward function becomes the linchpin of success. A simplistic formulation say, a single reward term tied to short-term profits, inevitably produces short-sighted and even harmful strategies.
By contrast, a carefully crafted reward must reflect the multi-dimensional landscape: penalties for excessive cycling, bonuses for aligning with user preferences, and incentives for contributing to grid stability.
Recent research has gone a step further by introducing **adaptive reward shaping**. Instead of relying on static weights for each component of the reward, these approaches allow the structure or the weights to evolve dynamically during training. One possible scheme begins by emphasizing profit, which enables the agent to

quickly acquire the basics of arbitrage. Once a plateau in performance is reached, the system progressively increases the penalty terms associated with battery wear or unmet user mobility targets. This staged adjustment steers the agent away from narrow, short-term gains and towards policies that remain robust in the long run, ultimately producing strategies that balance profitability, reliability, and sustainability[7].

### A Taxonomy of Reward Shaping Techniques

Reward shaping refers to the practice of enriching an environment's original reward signal,often sparse, delayed, or difficult to interpret—with additional terms that accelerate learning and provide intermediate guidance to the agent. Over the years, several methodologies have emerged, each grounded in different theoretical principles and suited to different problem settings. What follows is a taxonomy of the most relevant approaches in the context of V2G control.

**Potential-Based Reward Shaping (PBRS)** Perhaps the most theoretically rigorous approach, PBRS was introduced by Ng et al. and has the remarkable property of guaranteeing policy invariance: the optimal policy of the original Markov Decision Process is preserved, while convergence can be significantly accelerated[8]. The modified reward $R'$ is obtained as:

$$R'(s, a, s') = R(s, a, s') + F(s, s') = R(s, a, s') + \gamma\Phi(s') - \Phi(s) \tag{2.1}$$

where $\Phi : S \to \mathbb{R}$ is a potential function defined over the state space and $\gamma$ the discount factor. Intuitively, the shaping term $F$ rewards transitions that increase the potential $\Phi$. In a V2G setting, $\Phi(s)$ might assign higher values as the aggregate SoC of connected EVs approaches their target levels, thus supplying dense feedback for incremental progress without altering the long-term objective.

**Dynamic and Adaptive Reward Shaping** In contrast to PBRS, which prioritizes theoretical guarantees, adaptive approaches deliberately relax policy invariance to address the intricacies of multi-objective control. Here, the reward function itself evolves during training, either in response to the state of the system or according to a predefined schedule:

- **State-Dependent Shaping:** Reward weights adapt to the current state $s_t$. For example, the penalty associated with transformer overloading can be defined as $\lambda^{\text{tr}}(s_t)$, increasing exponentially as the load approaches a critical threshold. In this way, constraint violations are emphasized precisely when they become imminent.
- **Time- or Schedule-Based Shaping:** The relative importance of different reward components is varied across training episodes. An agent may initially be exposed to a reward function dominated by profitability,

[7] **wan2022dynamic**.
[8] **ng1999policy**.

before progressively incorporating penalties for user dissatisfaction and battery degradation. This staged modification closely mirrors the logic of curriculum-based training.

Such adaptive methods are particularly well-suited for scenarios, like V2G, where the notion of an "optimal" trade-off among competing objectives must itself be discovered rather than imposed from the outset.

**Curriculum Learning (CL)** Although strictly speaking a training paradigm rather than a reward shaping method, CL can be interpreted as an implicit form of shaping, since it gradually modifies both the environment and the associated reward structure. The agent is not immediately confronted with the full problem complexity, but instead progresses through a sequence of tasks of increasing difficulty. A possible curriculum for V2G might include:

1. **Phase 1:** Single EV, deterministic price signals, reward based solely on arbitrage profit.
2. **Phase 2:** Multiple EVs, stochastic pricing, introduction of user satisfaction penalties.
3. **Phase 3:** Full-scale environment including grid constraints, degradation costs, and the complete adaptive reward function.

This progression enables the agent to acquire foundational skills before addressing the most challenging aspects of the task, ultimately leading to more robust and transferable policies.

### Most Utilized Techniques in Reinforcement Learning for V2G

In the specific context of Vehicle-to-Grid management, the most effective and commonly used techniques are **Dynamic and Adaptive Reward Shaping** and **Curriculum Learning**.
The reason for their prevalence is rooted in the nature of the V2G problem itself. It is a multi-objective optimization problem with deeply intertwined and often conflicting goals (e.g., maximizing profit vs. minimizing battery wear, ensuring grid stability vs. guaranteeing user satisfaction).

- **Dynamic/Adaptive Reward Shaping** is exceptionally well-suited for V2G because the relative importance of each objective is not static; it is state-dependent. For example, satisfying a user's charging request is of little importance when they have 10 hours left, but it becomes critically important when they have 10 minutes left. An adaptive reward function that calculates an "urgency score" can capture this dynamic priority, which is impossible with a fixed-weight penalty. This allows the agent to learn a far more nuanced and realistic control policy.
- **Curriculum Learning** is widely used as a practical strategy to make the training of complex DRL agents for V2G feasible. Training an agent on the full, stochastic, multi-objective V2G problem from scratch is often unstable

and inefficient. By using a curriculum, the agent can first master basic concepts (like energy arbitrage) before moving on to handle complex constraints (like transformer limits and user deadlines), leading to more stable and effective final policies.

Conversely, **Potential-Based Reward Shaping (PBRS)** is less utilized for the overall V2G control problem. Its core strength—policy invariance—is actually a limitation here. The goal in V2G is not to find the optimal policy for a simple, predefined objective (like pure profit), but rather to discover a novel policy that represents the *best possible compromise* between all objectives. Dynamic shaping intentionally alters the learning objective to guide the agent to this superior compromise, a task for which PBRS is not designed.
To further refine this balance, recent research has explored **adaptive reward shaping**. Instead of using fixed weights for different components of the reward function, these methods dynamically adjust the weights or the structure of the reward during training. For example, an agent might initially be incentivized primarily by profit to learn the basic mechanics of arbitrage. As its performance plateaus, the penalty for battery degradation or for failing to meet user departure targets can be gradually increased. This guides the agent toward a more holistic and robust final policy, preventing a premature convergence to a suboptimal strategy that ignores long-term costs like battery health[9].

## 2.4 The Rise of Deep Reinforcement Learning for V2G Control

The fusion of RL with the representational power of deep neural networks gives us **Deep Reinforcement Learning (DRL)**, the state-of-the-art paradigm for V2G control. The journey of DRL algorithms applied to V2G is one of increasing sophistication and robustness.

- **The Leap to Continuous Control: DDPG:** The first major breakthrough for continuous control problems like V2G was the **Deep Deterministic Policy Gradient (DDPG)** algorithm[10]. As an actor-critic method, it could output precise, continuous power values. However, DDPG became notorious for its training instability and its crippling vulnerability to **overestimation bias**, where the critic systematically overestimates Q-values, leading the actor to converge on suboptimal policies[11].
- **Stabilizing the Foundation: TD3: Twin Delayed DDPG (TD3)** was developed specifically to address DDPG's flaws[12]. It introduces three crucial innovations: clipped double Q-learning to combat overestimation, delayed policy updates to stabilize training, and target policy smoothing to improve

[9] **wan2022dynamic**.
[10] **lillicrap2015continuous**.
[11] **orfanoudakis2022deep**; **alfaverh2022optimal**.
[12] **fujimoto2018addressing**.

robustness. These additions made it a much more reliable baseline for V2G tasks[13].

- **The State of the Art: Soft Actor-Critic (SAC):** SAC represents the current frontier, offering superior sample efficiency and stability[14]. Its core innovation is the **maximum entropy framework**. The agent's objective is not just to maximize the cumulative reward, but to do so while acting as randomly as possible. This entropy bonus encourages broad exploration, preventing the agent from prematurely converging to a narrow, suboptimal strategy. The resulting policies are not only high-performing but also more robust and adaptable to unforeseen changes in the environment, a critical feature for real-world deployment[15].

## 2.5 A Comparative Perspective on Control Methodologies

While DRL represents the cutting edge, it is crucial to contextualize it within the broader landscape.
**Model Predictive Control (MPC)** is the most powerful model-based alternative[16]. Its primary strength is its ability to handle constraints. However, its performance is fundamentally shackled to the accuracy of its internal model and forecasts[17]. In the V2G domain, creating an accurate model is nearly impossible due to non-linear battery dynamics, market volatility, and human unpredictability. Furthermore, solving the large-scale Mixed-Integer Linear Program (MILP) required at each time step becomes computationally intractable for large fleets[18].
Other methods, such as **meta-heuristic algorithms** (e.g., genetic algorithms), are typically used for offline scheduling and lack the real-time responsiveness required for dynamic V2G control[19].
In conclusion, the singular advantage of DRL is its inherent ability to learn and internalize the complex, non-linear trade-offs of the multi-objective V2G problem directly from data. This makes it uniquely suited to navigating the uncertainties of the real world. While other methods have their place, DRL stands out as the most promising technology for deploying the truly intelligent, autonomous, and robust V2G management systems required to achieve the ambitious energy and climate goals of the European Union.

[13] **liu2023optimal**; **wang2022multi**.
[14] **haarnoja2018soft**.
[15] **logeshwaran2022comparative**.
[16] **alsabbagh2022reinforcement**.
[17] **faggio2023design**.
[18] **schwenk2022computationally**.
[19] **ghosh2024optimal**; **kumar2024integration**.

Table 2.1: Comparative Analysis: DRL vs. Model Predictive Control (MPC) for V2G

| **Aspect** | **Deep Reinforcement Learning (DRL)** | **Model Predictive Control (MPC)** |
|---|---|---|
| **Paradigm** | Model-Free, learning-based. Learns optimal policy via trial-and-error. | Model-Based, optimization-based. Solves an optimization problem at each step. |
| **Strengths** | • Highly robust to uncertainty and stochasticity.<br>• No need for an explicit system model.<br>• Can learn complex, non-linear control policies.<br>• Fast inference time once trained. | • Explicitly handles hard constraints (safety guarantees).<br>• Proactive and anticipatory if forecasts are accurate.<br>• Well-established and understood. |
| **Weaknesses** | • Can be sample-inefficient during training.<br>• Lacks hard safety guarantees (an active research area).<br>• "Black box" nature can make policies hard to interpret. | • Performance is fundamentally tied to model and forecast accuracy.<br>• Computationally expensive at each time step (curse of dimensionality).<br>• Brittle to forecast errors and unmodeled dynamics. |
| **V2G Suitability** | Excellent for dynamic, uncertain environments with complex trade-offs. | Good for problems with simple dynamics and reliable forecasts, but struggles with real-world V2G complexity. |

# Chapter 3

# An Enhanced V2G Simulation Framework for Robust Control

Developing, validating, and benchmarking advanced control algorithms for Vehicle-to-Grid (V2G) systems is a task fraught with complexity. Real-world experimentation is often impractical due to prohibitive costs, logistical challenges, and risks to grid stability and vehicle hardware. To bridge the gap between theory and practice, a realistic, flexible, and standardized simulation environment is a scientific necessity. This thesis builds upon the foundation of **EV2Gym**, a state-of-the-art, open-source simulator designed for V2G smart charging research[1]. However, this work extends the original framework significantly, transforming it into a high-fidelity **digital twin** engineered not just for single-scenario optimization, but for the development and rigorous evaluation of **robust, generalist control agents**.

This enhanced framework provides a dual-pronged approach to experimentation: it allows for deep-dive analysis of agents specialized for a single environment, while also introducing a novel methodology for training and testing agents designed to generalize across a multitude of diverse, unpredictable scenarios. This chapter provides an in-depth tour of this extended architecture, its data-driven models, and its unique evaluation capabilities, establishing the methodological bedrock for the rest of this work.

## 3.1 Core Simulator Architecture

The framework retains the modular architecture of EV2Gym, which mirrors the key entities of a real-world V2G system. Its foundation on the OpenAI Gym (now Gymnasium) API remains a cornerstone, providing a standardized agent-environment interface defined by the familiar language of states, actions, and rewards[2].

The architecture consists of several interacting components:

- **Charge Point Operator (CPO):** The central intelligence of the simulation, managing the charging infrastructure and serving as the primary interface for

[1] **orfanoudakis2024ev2gym**.
[2] **brockman2016openai**.

the control algorithm (the DRL agent). The CPO aggregates system state information and dispatches control actions to individual chargers.

- **Chargers:** Digital representations of physical charging stations, configurable by type (AC/DC), maximum power, and efficiency. This allows for the simulation of heterogeneous charging infrastructures.
- **Power Transformers:** These components model the physical connection points to the grid, aggregating the electrical load from multiple chargers. Crucially, they enforce the physical power limits of the local distribution network and can model inflexible base loads (e.g., buildings) and local renewable generation (e.g., solar panels).
- **Electric Vehicles (EVs):** Dynamic and autonomous agents, each defined by its unique battery capacity, power limits, current and desired energy levels, and specific arrival and departure times.

The simulation process follows a reproducible three-phase structure: (1) **Initialization** from a comprehensive YAML configuration file, (2) a discrete-time **Simulation Loop** where the agent interacts with the environment, and (3) a final **Evaluation and Visualization** phase that generates standardized performance metrics.

## 3.2 Core Physical Models

The fidelity of the simulation is anchored in its detailed and empirically validated models, which are essential for developing control strategies robust enough for real-world application.

### 3.2.1 EV Model and Charging/Discharging Dynamics

The framework implements a realistic two-stage charging/discharging model that captures the non-linear behavior of lithium-ion batteries, simulating both the **constant current (CC)** and **constant voltage (CV)** phases. Each EV is defined by a rich parameter set: maximum capacity ($E_{max}$), a minimum safety capacity ($E_{min}$), separate power limits for charging and discharging ($P_{ch}^{max}, P_{dis}^{max}$), and distinct efficiencies for each process ($\eta_{ch}, \eta_{dis}$).

### 3.2.2 Battery Degradation Model

To address the critical issue of battery health in V2G operations, the simulator incorporates a semi-empirical battery degradation model. It quantifies capacity loss ($Q_{lost}$) as the sum of two primary aging mechanisms[3]:

- **Calendar Aging ($d_{cal}$):** Time-dependent capacity loss, influenced by the battery's average State of Charge (SoC) and temperature.

[3] **orfanoudakis2024ev2gym**.

- **Cyclic Aging ($d_{cyc}$):** Wear resulting from charge/discharge cycles, dependent on energy throughput, depth-of-cycle, and C-rate.

This integrated model allows for the direct quantification of how different control strategies impact the battery's long-term State of Health (SoH), enabling the training of agents that balance profitability with battery preservation.

### 3.2.3 EV Behavior and Grid Models

To ensure realism, the simulation is driven by authentic, open-source datasets. EV arrival/departure patterns and energy requirements are modeled using probability distributions derived from a large real-world dataset from **ElaadNL**. Grid conditions are similarly grounded in reality, using inflexible load data from the **Pecan Street** project and solar generation profiles from the **Renewables.ninja** platform[4].

## 3.3 A Dual-Pronged Evaluation Architecture

A key contribution of this thesis is the development of a sophisticated, dual-mode evaluation pipeline, which distinguishes between specialized and generalized agent performance. This is implemented through two primary execution scripts: `Single_Domain_Env.py` and `MultiScenarioEnv.py`.

### 3.3.1 Single-Domain Specialization

The `Single_Domain_Env.py` script is designed to train and evaluate "specialist" agents. In this workflow, a Reinforcement Learning agent is trained from scratch on a single, fixed configuration file. This approach is used to answer the question: "What is the optimal performance achievable for this specific, known environment?" It allows for a deep-dive analysis of an agent's ability to master one particular scenario, serving as a crucial baseline for performance.

### 3.3.2 Multi-Scenario Generalization

The `MultiScenarioEnv.py` script introduces a more challenging and realistic paradigm: training a single, "generalist" agent that must perform well across a diverse set of scenarios. This is achieved through two key innovations:

- **MultiScenarioEnv:** A custom Gymnasium environment that acts as a wrapper around multiple underlying `EV2Gym` instances. At the beginning of each training episode (i.e., on `reset()`), this environment randomly selects one of the provided configuration files. This forces the agent to learn a robust policy that is not overfitted to any single scenario's characteristics (e.g., number of chargers, grid capacity, or price volatility).

[4] **orfanoudakis2024ev2gym**.

- **CompatibilityWrapper:** A critical technical solution to handle the varying observation and action space sizes across different scenarios. Since a neural network policy has a fixed input and output size, this wrapper **pads** observations from smaller environments to a maximum size and **slices** action vectors from the agent to match the specific needs of the currently active environment. This enables a single agent to seamlessly control infrastructures of varying scales.

This multi-scenario training methodology is fundamental to developing agents that are truly robust and ready for deployment in the real world, where conditions are never static.

## 3.4 Software and Experimentation Workflow

The project's functionality is organized into a modular structure to facilitate clear and reproducible experimentation.

- `ev2gym/`: The core directory containing the simulator's heart.
  - `models/`: Defines the main environment (`ev2gym_env.py`) and the physical components (`ev.py`, `ev_charger.py`, `transformer.py`).
  - `baselines/`: Contains the classical control algorithms used for benchmarking, including heuristics (`heuristics.py`) and Model Predictive Control (`pulp_mpc.py`).
  - `rl_agent/`: Houses DRL-specific components, such as state space definitions (`state.py`) and reward functions (`reward.py`).
  - `data/`: Contains the input time-series data for EV arrivals, energy prices, and loads.
- `Compare.py`: A powerful utility script for pre-analysis and scenario comparison. It reads multiple YAML configuration files and generates summary tables and legends as images, allowing for a quick, visual comparison of experimental setups.
- `Single_Domain_Env.py`: The primary script for training and evaluating specialist agents on a single, user-selected scenario. It orchestrates the entire benchmark for one environment.
- `MultiScenarioEnv.py`: The script for training and evaluating robust, generalist agents. It utilizes the `MultiScenarioEnv` to train a single agent on a collection of scenarios and then evaluates its performance across each of them.

## 3.5 Evaluation Metrics

To ensure a fair and comprehensive comparison, all algorithms are evaluated against the same set of pre-generated scenarios (using a "replay" mechanism). The **mean** and **standard deviation** of performance are calculated across multiple simulation runs. The key metrics include:

- **Total Profit ($):** The net economic outcome, calculated as revenue from energy sales minus the cost of energy purchases.

$$\Pi_{\text{total}} = \sum_{t=0}^{T_{\text{sim}}} \sum_{i=1}^{N} \left( C_{\text{sell}}(t) P_{\text{dis},i}(t) - C_{\text{buy}}(t) P_{\text{ch},i}(t) \right) \Delta t$$

- **Tracking Error (RMSE, kW):** For grid-balancing scenarios, this measures the root-mean-square error between the fleet's aggregated power and a target setpoint.

$$E_{\text{track}} = \sqrt{\frac{1}{T_{\text{sim}}} \sum_{t=0}^{T_{\text{sim}}-1} \left( P_{\text{setpoint}}(t) - P_{\text{total}}(t) \right)^2}$$

- **User Satisfaction (Average):** The fraction of energy delivered compared to what was requested by the user, averaged across all EV sessions. A score of 1 indicates perfect service.

$$US_{\text{avg}} = \frac{1}{N_{\text{EVs}}} \sum_{k=1}^{N_{\text{EVs}}} \min \left( 1, \frac{E_k(t_k^{\text{dep}})}{E_k^{\text{des}}} \right)$$

- **Transformer Overload (kWh):** The total energy that exceeded the transformer's rated power limit. An ideal controller should achieve a value of 0.

$$O_{\text{tr}} = \sum_{t=0}^{T_{\text{sim}}} \sum_{j=1}^{N_T} \max(0, P_j^{\text{tr}}(t) - P_j^{\text{tr,max}}) \cdot \Delta t$$

- **Battery Degradation ($):** The estimated monetary cost of battery aging due to both cyclic and calendar effects.

$$D_{\text{batt}} = \sum_{k=1}^{N_{\text{EVs}}} (\text{CyclicCost}_k + \text{CalendarCost}_k)$$

## 3.6 Reinforcement Learning Formulation

The control problem is formalized as a Markov Decision Process (MDP), defined by the tuple $(S, A, P, R, \gamma)$.

### 3.6.1 State Space ($S$)

The state $s_t \in S$ is a feature vector providing a snapshot of the environment at time $t$. A representative state, as defined in modules like `V2G_profit_max_loads.py`, includes:

$$s_t = [t, P_{\text{total}}(t-1), \mathbf{c}(t, H), \mathbf{L}_1(t, H), \mathbf{PV}_1(t, H), \ldots, \mathbf{s}_1^{\text{EV}}(t), \ldots, \mathbf{s}_N^{\text{EV}}(t)]^T$$

where the components are:

- $t$: The current time step.
- $P_{\text{total}}(t-1)$: The aggregated power from the previous time step.
- $\mathbf{c}(t, H)$: A vector of **predicted future** electricity prices over a horizon $H$.
- $\mathbf{L}_j(t, H), \mathbf{PV}_j(t, H)$: Forecasts for inflexible loads and solar generation.
- $\mathbf{s}_i^{\text{EV}}(t) = [\text{SoC}_i(t), t_i^{\text{dep}} - t]$: Key information for each EV $i$, including its State of Charge and remaining time until departure.

### 3.6.2 Action Space ($A$)

The action $a_t \in A$ is a continuous vector in $\mathbb{R}^N$, where $N$ is the number of chargers. For each charger $i$, the command $a_i(t) \in [-1, 1]$ is a normalized value that is translated into a power command:

- If $a_i(t) > 0$, the EV is charging: $P_i(t) = a_i(t) \cdot P_{\text{charge},i}^{\max}$.
- If $a_i(t) < 0$, the EV is discharging (V2G): $P_i(t) = a_i(t) \cdot P_{\text{discharge},i}^{\max}$.

### 3.6.3 Reward Function ($R(s, a, s')$)

The reward function $R(t)$ encodes the objectives of the control agent. The framework allows for the selection of different reward functions from the `reward.py` module to suit various goals. Key examples include:

- **Profit Maximization with Penalties** (`ProfitMax_TrPenalty_UserIncentives`): This function creates a balance between economic gain and physical constraints.

$$R(t) = \underbrace{\text{Profit}(t)}_{\text{Economic Gain}} - \underbrace{\lambda_1 \cdot \text{Overload}(t)}_{\text{Grid Penalty}} - \underbrace{\lambda_2 \cdot \text{Unsatisfaction}(t)}_{\text{User Penalty}}$$

  The agent is rewarded for profit but penalized for overloading transformers and for failing to meet the charging needs of departing drivers.

- **Squared Tracking Error** (`SquaredTrackingErrorReward`): Used for grid service applications where precision is paramount.

$$R(t) = -\left(P_{\text{setpoint}}(t) - \sum_{i=1}^{N} P_i(t)\right)^2$$

  The reward is the negative squared error from the power setpoint, incentivizing the agent to minimize this error at all times.

By leveraging this enhanced framework, this thesis moves beyond single-scenario optimization to develop and validate an intelligent V2G control agent that is not only high-performing but also robust, adaptable, and ready for the complexities of real-world deployment.

### 3.6.4 A History-Based Adaptive Reward for Profit Maximization

To effectively steer the learning agent towards a policy that is both highly profitable and reliable, we have designed and implemented a novel, history-based adaptive reward function, named `FastProfitAdaptiveReward`. This function departs from traditional static-weight penalties and instead introduces a dynamic feedback mechanism where the severity of penalties is directly influenced by the agent's recent performance. The core philosophy is to aggressively prioritize economic profit while using adaptive penalties as guardrails that become stricter only when the agent begins to consistently violate operational constraints.

The total reward at each timestep $t$, $R_t$, is calculated as the net economic profit minus any active penalties for user dissatisfaction or transformer overload.

$$R_t = \Pi_t - P_t^{\text{sat}} - P_t^{\text{tr}} \tag{3.1}$$

#### Economic Profit

The foundation of the reward signal is the direct, instantaneous economic profit, $\Pi_t$. This component provides a clear and strong incentive for the agent to learn market dynamics, encouraging it to charge during low-price periods and discharge (V2G) during high-price periods.

$$\Pi_t = \sum_{i=1}^{N} \left( C_t^{\text{sell}} \cdot P_{i,t}^{\text{dis}} - C_t^{\text{buy}} \cdot P_{i,t}^{\text{ch}} \right) \Delta t \tag{3.2}$$

where $N$ is the number of connected EVs, $C_t^{\text{sell}}$ and $C_t^{\text{buy}}$ are the electricity prices, and $P_{i,t}^{\text{dis}}$ and $P_{i,t}^{\text{ch}}$ are the discharging and charging powers for EV $i$.

#### Adaptive User Satisfaction Penalty

The penalty for failing to meet user charging demands, $P_t^{\text{sat}}$, is not a fixed value. Instead, it adapts based on the system's recent history of performance. The environment maintains a short-term memory of the average user satisfaction over the last 100 timesteps. From this history, we calculate an average satisfaction score, $\bar{S}_{hist}$.

A *satisfaction severity multiplier*, $\lambda_t^{\text{sat}}$, is then calculated. This multiplier grows quadratically as the historical average satisfaction drops, meaning that if the system has been performing poorly, the consequences for a new failure become much more severe.

$$\lambda_t^{\text{sat}} = \lambda_{\text{base}}^{\text{sat}} \cdot (1 - \bar{S}_{hist})^2 \tag{3.3}$$

where $\lambda_{\text{base}}^{\text{sat}}$ is a base scaling factor (e.g., 20.0). A penalty is only applied if any departing EV's satisfaction, $S_k$, is below a critical threshold (e.g., 95%). The magnitude of the penalty is the product of the adaptive multiplier and the current satisfaction deficit.

$$P_t^{\text{sat}} = \lambda_t^{\text{sat}} \cdot (1 - \min(S_k)) \quad \forall k \in \text{EVs departing at } t \tag{3.4}$$

This creates a powerful feedback loop: a single, isolated failure in an otherwise well-performing system results in a mild penalty. However, persistent failures lead to a rapidly escalating penalty, forcing the agent to correct its behavior.

### Adaptive Transformer Overload Penalty

Similarly, the transformer overload penalty, $P_t^{\text{tr}}$, adapts based on the recent frequency of overloads. The environment tracks how often an overload has occurred in the last 100 timesteps, yielding an overload frequency, $F_{hist}^{\text{tr}}$.

This frequency is used to compute a linear *overload severity multiplier*, $\lambda_t^{\text{tr}}$. The more frequently overloads have happened, the higher the penalty for a new one.

$$\lambda_t^{\text{tr}} = \lambda_{\text{base}}^{\text{tr}} \cdot F_{hist}^{\text{tr}} \tag{3.5}$$

where $\lambda_{\text{base}}^{\text{tr}}$ is a base scaler (e.g., 50.0). If the total power drawn, $P_j^{\text{total}}(t)$, exceeds the transformer's limit, $P_j^{\max}$, a penalty is applied. This penalty consists of a small, fixed base amount plus the adaptive component, which scales with the magnitude of the current overload.

$$P_t^{\text{tr}} = P_{\text{base}} + \lambda_t^{\text{tr}} \cdot \sum_{j=1}^{N_T} \max(0, P_j^{\text{total}}(t) - P_j^{\max}) \tag{3.6}$$

This mechanism teaches the agent that while a rare, minor overload might be acceptable in pursuit of high profit, habitual overloading is an unsustainable and heavily penalized strategy.

### Rationale and Significance

This history-based adaptive reward function represents a significant advancement over static or purely state-based approaches. By making the penalty weights a function of the system's recent performance history, we provide a more nuanced and stable learning signal. The agent is not punished excessively for isolated, exploratory actions that might lead to a minor constraint violation. Instead, it is strongly discouraged from developing policies that lead to chronic system failures.

The intuition is to mimic a more realistic management objective: maintain high performance on average, and react strongly only when performance trends begin to degrade. This method is also computationally efficient, avoiding complex state-dependent calculations in favor of simple updates to historical data queues. Ultimately, this reward structure guides the agent to discover policies that are not only profitable but also robust and reliable over time, striking a more intelligent balance between economic ambition and operational safety.

## 3.7 Model Predictive Control (MPC)

The MPC, implemented in `mpc.py` and `eMPC.py`, solves an optimization problem at every time step over a prediction horizon $H$.

### 3.7.1 System Model

The system is modeled in linear state-space form. The state $\mathbf{x}_k \in \mathbb{R}^N$ is the vector of SoCs of all EVs at time $k$. The input $\mathbf{u}_k \in \mathbb{R}^{2N}$ is the vector of charging and discharging powers.

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k$$

The matrices $A_k$ (`Amono`) and $B_k$ (`Bmono`) are time-varying because they depend on which EVs are connected. $A_k$ is typically a diagonal identity-like matrix modeling the persistence of EVs. $B_k$ maps power to SoC change, including efficiencies and $\Delta t$.

### 3.7.2 Optimization Problem

At time $t$, the MPC solves:

$$\min_{\{\mathbf{u}_k\}_{k=t}^{t+H-1}} \sum_{k=t}^{t+H-1} \mathbf{f}_k^T \mathbf{u}_k$$

subject to:

$$\begin{aligned}
&\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k, \quad \forall k \in [t, t+H-1] && \text{(Dynamics)} \\
&\mathbf{x}_k^{\min} \leq \mathbf{x}_k \leq \mathbf{x}_k^{\max} && \text{(SoC limits)} \\
&\mathbf{0} \leq \mathbf{u}_k^{\text{ch}} \leq \mathbf{u}_k^{\text{ch,max}} \cdot \mathbf{z}_k && \text{(Charge limits)} \\
&\mathbf{0} \leq \mathbf{u}_k^{\text{dis}} \leq \mathbf{u}_k^{\text{dis,max}} \cdot (1 - \mathbf{z}_k) && \text{(Discharge limits)} \\
&\sum_{i \in \text{CS}_j} (u_i^{\text{ch}} - u_i^{\text{dis}}) + L_j(k) - PV_j(k) \leq P_j^{\text{tr,max}}(k) && \text{(Transformer limits)}
\end{aligned}$$

where $\mathbf{z}_k$ is a vector of binary variables to prevent simultaneous charge and discharge. The cost vector $\mathbf{f}_k$ contains the energy prices. The code formulates this problem compactly as $\mathbf{AU} \leq \mathbf{bU}$, where $\mathbf{U}$ is the vector of all actions over the horizon.

## 3.8 Offline Optimization with Gurobi

Gurobi is used to find the optimal offline (a posteriori) solution, providing a performance benchmark. The files `profit_max.py` and `tracking_error.py` define the optimization problem over the entire simulation horizon $T_{\text{sim}}$.

### 3.8.1 Decision Variables

- $E_{p,i,t}$: Energy in the EV at port $p$ of station $i$ at time $t$.
- $I_{p,i,t}^{\text{ch}}, I_{p,i,t}^{\text{dis}}$: Charging/discharging currents.

- $\omega^{\text{ch}}_{p,i,t}, \omega^{\text{dis}}_{p,i,t}$: Binary variables for operating modes.

### 3.8.2 Objective Function (Example: Profit Maximization)

$$\max \sum_{t=0}^{T_{\text{sim}}} \sum_{i=1}^{N_{CS}} \sum_{p=1}^{N_p} \left( C_{\text{sell}}(t) P^{\text{dis}}_{p,i,t} - C_{\text{buy}}(t) P^{\text{ch}}_{p,i,t} \right) \Delta t - \lambda \sum_{k \in \text{EVs departed}} (E^{\text{des}}_k - E_k(t^{\text{dep}}_k))^2$$

where $P = V \cdot I \cdot \eta$.

### 3.8.3 Main Constraints

- **Energy Balance:**

$$E_{p,i,t} = E_{p,i,t-1} + (\eta_{\text{ch}} V_i I^{\text{ch}}_{p,i,t} - \frac{1}{\eta_{\text{dis}}} V_i I^{\text{dis}}_{p,i,t}) \Delta t$$

- **Activation of Current:**

$$I^{\text{ch}}_{p,i,t} \leq M \cdot \omega^{\text{ch}}_{p,i,t} \quad , \quad I^{\text{dis}}_{p,i,t} \leq M \cdot \omega^{\text{dis}}_{p,i,t}$$

- **Mutual Exclusion:**

$$\omega^{\text{ch}}_{p,i,t} + \omega^{\text{dis}}_{p,i,t} \leq 1$$

- **Current and SoC Limits:**

$$I^{\min} \leq I_{p,i,t} \leq I^{\max} \quad , \quad E^{\min} \leq E_{p,i,t} \leq E^{\max}$$

- **SoC at Departure:**

$$E_{p,i}(t^{\text{dep}}) \geq E^{\text{des}}_{p,i}$$

## 3.9 Online MPC Formulation (PuLP Implementation)

The Model Predictive Control (MPC) implemented with PuLP solves a profit maximization problem at each time step $t$ over a finite prediction horizon $H$. This formulation is designed for online, real-time control, where decisions are made based on the current system state and future predictions.

### 3.9.1 Mathematical Formulation

At each time step $t$, the MPC controller solves the following optimization problem.

### Objective Function: Net Operational Profit

The objective is to maximize the total net operational profit over the control horizon $H$. This provides a comprehensive economic model that goes beyond simple energy arbitrage.

$$\max_{P^{\text{ch}},P^{\text{dis}},z} \sum_{k=t}^{t+H-1} \sum_{i\in\text{CS}} (\text{Revenues}_{i,k} - \text{Costs}_{i,k}) \tag{3.7}$$

The revenue and cost components are defined for each station $i$ at time step $k$ as:

- **Revenues** consist of:
  - Grid Sales Revenue (V2G): $c_k^{\text{sell}} \cdot P_{i,k}^{\text{dis}} \cdot \Delta t$
  - User Charging Revenue: $c^{\text{user}} \cdot P_{i,k}^{\text{ch}} \cdot \Delta t$
- **Costs** consist of:
  - Grid Purchase Cost: $c_k^{\text{buy}} \cdot P_{i,k}^{\text{ch}} \cdot \Delta t$
  - Battery Degradation Cost: $c^{\text{deg}} \cdot (P_{i,k}^{\text{ch}} + P_{i,k}^{\text{dis}}) \cdot \Delta t$

where $c_k^{\text{sell}}$ and $c_k^{\text{buy}}$ are the time-varying electricity prices, $c^{\text{user}}$ is the fixed price for the end-user, $c^{\text{deg}}$ is the estimated cost of battery degradation per kWh cycled, and $\Delta t$ is the time step duration.

### System Constraints

The optimization is subject to the following constraints for each station $i$ and time step $k \in [t, t+H-1]$.

**Energy Balance Dynamics.** The state of energy of the EV battery evolves according to:

$$E_{i,k} = E_{i,k-1} + \left(\eta^{\text{ch}} P_{i,k}^{\text{ch}} - \frac{1}{\eta^{\text{dis}}} P_{i,k}^{\text{dis}}\right) \cdot \Delta t \tag{3.8}$$

where the initial state $E_{i,t-1}$ is the currently measured energy level of the EV.

**Power Limits and Mutual Exclusion.** Charging and discharging powers are bounded by the EV's capabilities and controlled by a binary variable $z_{i,k}$ to prevent simultaneous operation.

$$0 \le P_{i,k}^{\text{ch}} \le P_i^{\text{ch,max}} \cdot z_{i,k} \tag{3.9}$$

$$0 \le P_{i,k}^{\text{dis}} \le P_i^{\text{dis,max}} \cdot (1 - z_{i,k}) \tag{3.10}$$

**State of Energy (SoE) Limits.** The battery energy level must remain within its physical operational window.

$$E_i^{\min} \le E_{i,k} \le E_i^{\max} \tag{3.11}$$

**User Satisfaction (Hard Constraint).** The desired energy level must be met at the time of departure. This is modeled as a hard constraint, reflecting a non-negotiable service requirement.

$$E_{i,k_{\text{dep}}} \geq E_i^{\text{des}} \tag{3.12}$$

where $k_{\text{dep}}$ is the predicted departure step of the EV within the horizon.

**Transformer Power Limit.** The total net power drawn from (or injected into) the grid by all charging stations must not exceed the transformer's maximum capacity.

$$\sum_{i \in \text{CS}} (P_{i,k}^{\text{ch}} - P_{i,k}^{\text{dis}}) \leq P^{\text{tr,max}} \tag{3.13}$$

## 3.10 Conceptual Comparison: PuLP MPC vs. Gurobi Offline Optimizer

While both the PuLP MPC and the Gurobi offline optimizer are used to solve the EV charging problem, they operate on fundamentally different principles and serve distinct purposes. This section provides a discursive comparison of their core concepts.

### 3.10.1 Core Philosophy: Controller vs. Judge

The most significant difference lies in their philosophy. The **PuLP MPC** is designed as a **controller**. It operates online, making decisions in real-time with incomplete information about the future (e.g., EV arrivals, price fluctuations beyond the prediction horizon). Its goal is to find a practical and robust strategy for the immediate future.
Conversely, the **Gurobi formulation** acts as a **judge**. It is an offline tool that solves the problem over the entire simulation period with perfect hindsight (a-posteriori). Its purpose is not to control the system in real-time, but to establish a theoretical performance benchmark—the "perfect score"—against which the performance of a practical controller like the MPC can be measured.

### 3.10.2 Objective Function: Operational Profit vs. Energy Arbitrage

The objectives, while both related to profit, reflect their different roles. The PuLP MPC maximizes a detailed **Net Operational Profit**, incorporating a realistic business model that includes revenue from end-users and operational costs like battery degradation. This makes its decisions economically grounded from a business perspective.
The Gurobi optimizer, on the other hand, typically maximizes profit from a simpler **energy arbitrage** model, focusing on the difference between buying and selling

electricity. While it includes a penalty for not meeting user demand, it does not explicitly account for the same level of operational economic detail as the MPC.

### 3.10.3 Handling of User Satisfaction: Hard vs. Soft Constraints

This distinction is critical from an operational standpoint. The PuLP MPC treats user satisfaction as a **Hard Constraint**. The EV *must* reach its desired energy level by its departure time. If the model determines this is impossible, the optimization problem becomes infeasible, signaling a failure to meet a mandatory service level agreement.
The Gurobi formulation treats user satisfaction primarily as a **Soft Constraint** via a penalty term in its objective function. This allows the optimizer to make a trade-off: it can choose to not fully charge a vehicle if the economic benefit of doing so (e.g., selling a large amount of energy to the grid at a high price) outweighs the penalty for customer dissatisfaction. This is useful for theoretical analysis but less practical for guaranteeing service.