Reinforcement Learning Control of MicroGrid Systems

Farooq Olanrewaju (g202404900)[1], Abubakar Abdulkarim (g202421140)[2]
[1]*Control & Instrumentation Eng. Dept.*, *King Fahd Univ. of Petroleum & Minerals*, Dhahran 31261, Saudi Arabia
[2]*Electrical Eng. Dept.*, *King Fahd Univ. of Petroleum & Minerals*, Dhahran 31261, Saudi Arabia

***ABSTRACT*—Microgrids combine renewable generation, dispatchable resources, energy storage, and local loads, and can operate either grid-connected or islanded. Rapid renewable intermittency, stochastic demand, and time-varying electricity prices make real-time energy management challenging for fixed-rule or strictly model-based controllers, particularly when reliability constraints must be respected. This paper proposes a reinforcement-learning (RL) energy management system (EMS) for a hybrid microgrid comprising photovoltaic and wind generation, battery storage, a dispatchable unit, and grid import/export. The control problem is cast as a continuous Markov decision process and implemented in an OpenAI Gym-compatible simulator driven by load and resource time-series (measured or synthetically generated). We benchmark a rule-based EMS against DRL agents trained with Proximal Policy Optimization (PPO), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Soft Actor-Critic (SAC). Across the studied scenarios, unstructured (random) actions produce frequent power-balance violations and load shedding, whereas learned continuous-control policies improve supply adequacy and reduce unmet demand and renewable curtailment while coordinating storage and grid trading within operational limits. The study also highlights the sensitivity of learned performance to environment fidelity, motivating future extensions that explicitly model degradation and outage/repair processes for deployment-ready evaluation.**

***Index Terms*—Microgrid, energy management system, reinforcement learning, deep reinforcement learning, continuous control, PPO, TD3, SAC, grid trading, reliability.**

I. Introduction

A microgrid is a controllable, localized portion of the distribution network that can operate connected to the main grid or in islanded mode [1], [2]. Microgrids typically integrate renewable generation mostly photovoltaic (PV) and wind, dispatchable generation, energy storage, and diverse loads both residential and industrial as shown in Fig.1. An energy management system (EMS) coordinates these assets to minimize operating cost while meeting reliability and power-quality requirements. However, increasing renewable penetration introduces fast variability and uncertainty; furthermore, component failures, grid outages, and time-varying electricity prices complicate optimal control and can degrade supply security if not handled explicitly [2].

Conventional EMS approaches include rule-based heuristics, dynamic programming, and model predictive control (MPC). These methods provide structured constraint handling, but typically require explicit models and forecasts; their performance may deteriorate under significant uncertainty, modeling mismatch, or rare-event contingencies. Motivated by the need for adaptive decision-making under uncertainty, recent work has explored reinforcement learning (RL) for microgrid energy management [3]. In parallel, open-source simulators such as `pymgrid` have lowered the barrier for RL-oriented EMS research by standardizing environments and interfaces for training and evaluation [4]. Nevertheless, many published RL studies employ simplified discrete states/actions and often omit practical phenomena such as battery degradation, repair costs, and multi-load interactions, which can materially change optimal operating strategies and the realism of performance claims.

This paper develops a continuous-control RL EMS that integrates local generation, storage, and grid trading to optimize cost-efficiency, reliability, and resilience under variable domestic and industrial loads. The EMS is formulated as a Markov decision process (MDP) with continuous observations and actions, enabling direct learning of practical setpoints using modern continuous-control DRL algorithms such as PPO, TD3, and SAC [5]–[7].

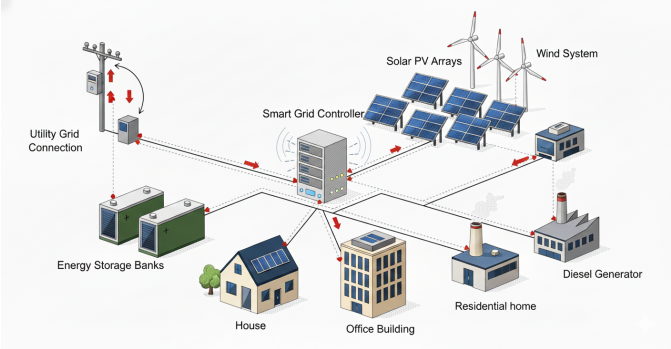

Figure 1. Microgrid system overview

A. Contributions

The main contributions are:

- A continuous-control MDP formulation for hybrid microgrid EMS with renewables, storage, dispatchable generation, and grid trading.
- A comparative analysis of modern DRL controllers (PPO/TD3/SAC) against rule-based baselines under variable residential and industrial demand.
- Development of a Python-based microgrid simulation and Gym-style environment built around realistic PV, wind, battery, grid, and failure models.

B. Paper organization

Section II reviews related work on optimization/MPC-based EMS and RL-based EMS, including safety and benchmarking considerations. Section III presents the microgrid model and MDP formulation. Section IV describes the learning algorithms and experimental methodology. Section V reports results and discussion. Section VI concludes and outlines future work.

II. Literature review

Microgrid EMS research broadly spans (i) optimization- and MPC-based methods that use explicit models and forecasts, and (ii) learning-based methods, particularly RL/DRL, that learn policies from interaction. Across both categories, the core challenge is balancing economics and reliability under uncertainty while respecting device and network constraints [2].

A. Optimization- and MPC-based EMS under uncertainty

Optimization-based EMS methods (deterministic, stochastic, or robust) are attractive because they encode constraints directly and can incorporate tariffs, reserves, and operational priorities. MPC extends this framework by repeatedly solving a constrained optimization problem over a receding horizon, enabling feedback through re-optimization as forecasts update. However, MPC performance depends strongly on model fidelity and forecast quality, and it can be stressed by high renewable variability, heterogeneous loads, and contingencies such as islanding and component failures. Moreover, multi-timescale operation like day-ahead planning plus real-time correction raises coordination questions; approaches that couple operational planning and real-time optimization via value-function or cost-to-go ideas attempt to address this gap by embedding longer-horizon consequences into real-time decisions [8]. These limitations motivate adaptive strategies that can react effectively even when explicit models are imperfect.

B. RL/DRL for microgrid energy management

RL formulates EMS as sequential decision-making, learning a policy that maps states to actions to maximize long-term return. Early work showed that RL can handle stochastic renewables and demand and can learn effective EMS policies without an explicit transition model [3]. As EMS models grow in dimensionality and nonlinearity, DRL becomes important; empirical studies comparing DRL algorithms for microgrid EMS with flexible demand report that algorithm choice and training stability significantly affect convergence and operational performance [9]. Even with this promising results, many EMS-RL studies simplify action spaces like discrete charge/discharge modes and omit important operational mechanisms such as outages/repairs and degradation, which can inflate performance estimates and reduce transferability.

C. Continuous-control DRL and actor–critic methods

Practical EMS decisions are often continuous (battery power, grid import/export, dispatchable generation). Continuous-control DRL is therefore a natural fit, avoiding coarse discretization that can reduce optimality and induce switching behavior near constraints. Modern actor–critic methods are widely used in continuous control: PPO stabilizes policy-gradient updates using clipped objectives [5], TD3 reduces overestimation bias using twin critics and delayed updates [6], and SAC optimizes a maximum-entropy objective to encourage exploration and improve robustness [7]. For EMS, these properties are relevant because the environment is non-stationary and may include rare but consequential events such as grid outages.

D. Safety, constraints, and feasibility in EMS-RL

EMS operation is safety-critical: actions must respect SOC bounds, power limits, and import/export capacities while maintaining supply to critical loads. Standard RL exploration does not guarantee constraint satisfaction, motivating safe and constrained RL. Survey work categorizes safe RL methods into approaches that modify the optimality criterion like risk-sensitive or constrained objectives and those that incorporate external knowledge such as shielding, action projection, or safety filters [10]. CMDP-based methods provide a principled framework by treating constraint violations as separate cost signals; constrained policy optimization is a representative approach that updates policies while enforcing constraints under trust-region style bounds [11]. In practice, EMS-RL studies often combine DRL with engineering safeguards to ensure physical feasibility during training and deployment.

E. Battery degradation and lifecycle-aware EMS

Battery storage is central to microgrid flexibility, but cycling accelerates degradation and changes the true economic optimum. Many EMS-RL studies treat storage as an ideal buffer with fixed capacity, which can overstate savings and lead to unrealistic dispatch patterns. Battery aging mechanisms and their dependence on operating conditions are well documented [12], motivating degradation-aware EMS that includes lifecycle costs or proxy penalties, throughput- or SOC-swing-based terms in the objective. From an RL point, adding degradation costs changes the reward landscape and can shift learned policies toward gentler cycling; however, the lack of standardized degradation models and evaluation protocols remains a key barrier to cross-paper comparability.

F. Benchmarking and open simulation environments

Because EMS-RL outcomes depend heavily on environment design and evaluation protocol, benchmarking and reproducibility are recurring concerns. Open-source environments support more credible comparisons by standardizing interfaces and scenarios. `pymgrid` provides an RL-oriented microgrid simulator aimed at tertiary EMS research [4], while OpenModelica Microgrid Gym offers a Gym-compatible environment for microgrid control experimentation [13]. Despite these advances, unified benchmarks that simultaneously capture continuous-control EMS setpoints, explicit outage/repair processes, degradation-aware storage modeling, and multi-load reliability metrics remain limited.

III. Methodology

This section presents the component-level mathematical models used to simulate the microgrid dynamics and defines the learning-based energy management strategy. The overall goal is to obtain a tractable yet physically meaningful environment where an RL agent can learn continuous setpoints for storage and grid exchange while respecting operational limits and reliability objectives.

A. Mathematical Models

We model the microgrid as a discrete-time system with time step Δt. At each step, renewable generation and loads are treated as exogenous inputs (measured or time-series driven), while the controllable assets (battery and grid exchange) are actuated by the EMS.

To improve transparency and reproducibility, we include representative plots of the renewable inputs and the corresponding model outputs. Specifically, synthetic profiles are used to sanity-check the PV, wind, and storage sub-models under controlled conditions, while real-data-driven profiles illustrate the time-series characteristics employed in evaluation scenarios (e.g., variability, intermittency, and realistic magnitudes).

1) Photovoltaic (PV) Model

The PV generation is computed using a commonly adopted irradiance–temperature performance model [14], [15]. The PV power output is given by

$$P = P_r \mu \frac{G}{G_{ref}} \left[1 + \gamma \left(T_{cell} - T_a\right)\right] \tag{1}$$

where P is the PV power output (kW), P_r is the rated PV power (kW), μ is a derating factor accounting for aggregate non-idealities (e.g., soiling, wiring losses, mismatch, and inverter losses), G is solar irradiance, and G_{ref} is the PV reference irradiance (typically standard test conditions). The bracketed term captures the first-order sensitivity of PV output to temperature: γ is the temperature coefficient, T_{cell} is the PV cell temperature, and T_a is the ambient temperature. This model provides a lightweight but effective mapping from weather inputs to available PV power, making it suitable for control-oriented simulation and RL training where many rollouts are required [14], [15].Fig. 2 illustrates a synthetic input/output example used to verify the sensitivity of PV output to irradiance and temperature. Figs. 3–4 show representative PV power profiles used in the data-driven evaluation scenarios.

Fig. 2 illustrates a synthetic input/output example used to verify the sensitivity of PV output to irradiance and temperature. Figs. 3–4 show representative PV power profiles used in the data-driven evaluation scenarios.

2) Wind Turbine Model

Wind generation is modeled using a piecewise power curve parameterized by cut-in, rated, and cut-out wind speeds [3]. The wind turbine output is defined as

$$P = \begin{cases} 0 & \text{if } v < v_{ci} \\ P_r \frac{v - v_{ci}}{v_r - v_{ci}} \Delta t & \text{if } v_{ci} \le v < v_r \\ P_r \Delta t & \text{if } v_r \le v < v_{co} \\ 0 & \text{if } v > v_{co} \end{cases} \tag{2}$$

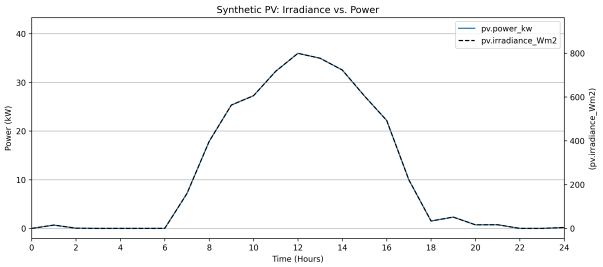

Figure 2. Representative synthetic PV input/output example: PV power computed from the irradiance–temperature model using synthetic weather inputs.

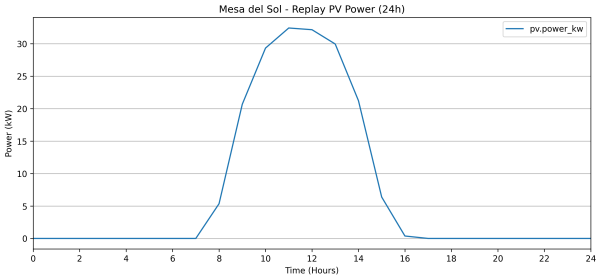

Figure 3. Example PV power profile from the Mesa Del Sol microgrid dataset used for data-driven simulations [16].

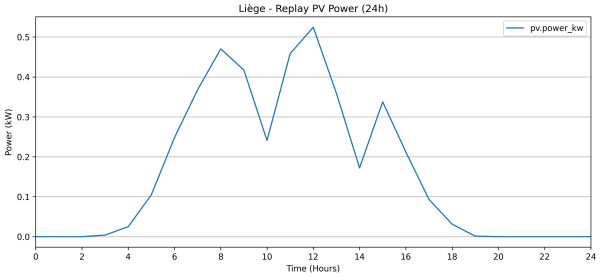

Figure 4. Example PV power profile from the Liège scenario used in evaluation, illustrating day-to-day variability in solar generation.

where v is the current wind speed (m/s), v_{ci}, v_r, and v_{co} are the cut-in, rated, and cut-out speeds (m/s), P_r is the rated wind turbine power (kW), and Δt is the time interval (s). The piecewise structure captures the physical operating regimes: no production at low wind speeds, a ramp-up region as aerodynamic power increases, a rated plateau due to generator/controls saturation, and shutdown at extreme winds for protection [3]. In our simulator, this model provides the available wind contribution to the instantaneous power balance at each step.

Fig. 5 verifies the expected cut-in/rated/cut-out operating regimes of the wind turbine model, while Fig. 6 shows a representative wind-speed profile used in the evaluation scenarios.

3) Battery Model

Battery dynamics are represented through the state of charge (SOC) and a simplified capacity aging model [17]. The SOC is defined as:

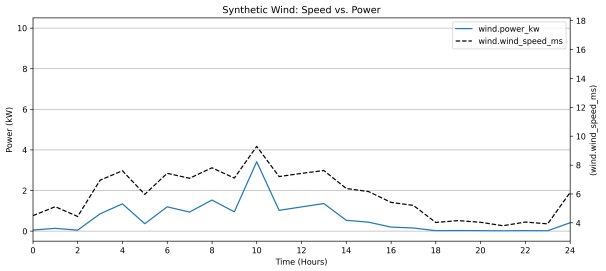

Figure 5. Representative synthetic wind input/output example: turbine power computed from the piecewise wind-speed power curve.

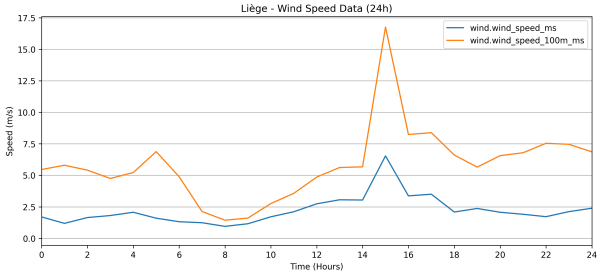

Figure 6. Example wind-speed time series from the Liège scenario used to generate wind power via the turbine model.

$$SOC(t) = \frac{C_s(t)}{C_n(t)} \tag{3}$$

where $C_s(t)$ is the current stored charge (C) and $C_n(t)$ is the nominal charge storage capacity (C). The SOC update under charging and discharging is modeled as

$$SOC(t+1) = SOC(t) + \mu_c \frac{P\Delta t}{C_s} \quad \text{(Charging)} \tag{4}$$

$$SOC(t+1) = SOC(t) - \mu_d \frac{P\Delta t}{C_s} \quad \text{(Discharging)} \tag{5}$$

where P is the battery power (kW) applied during Δt and μ_c and μ_d are charging/discharging coefficients that represent conversion losses. Operationally, $SOC(t)$ is constrained to remain within allowable limits (e.g., $SOC_{\min}$ and $SOC_{\max}$), and power setpoints are saturated to respect charge/discharge limits.

To reflect long-term performance degradation, the nominal capacity is updated using a throughput/SOC-swing degradation proxy [17]:

$$C_n(t) = C_n(t-1) - C_n(0)\,\varphi\,[SOC(t-1) - SOC(t)] \tag{6}$$

where φ is the aging coefficient. Although simplified compared to electrochemical aging models, this formulation introduces an explicit coupling between cycling behavior and usable capacity, enabling evaluation of control policies under non-ideal storage evolution [17].

Fig. 7 provides a sanity check of the SOC dynamics under charging and discharging, confirming that the model responds consistently to commanded power profiles.

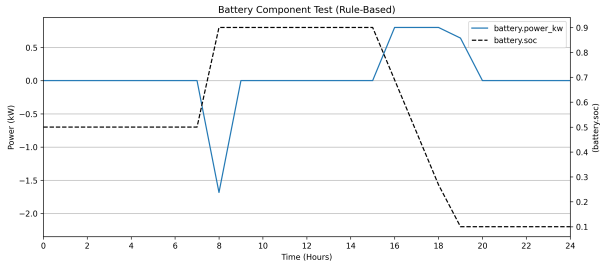

Figure 7. Battery model sanity check under a representative charge/discharge sequence, illustrating SOC evolution under the implemented update equations.

4) Grid Model and Power Balance

At each time step, the microgrid enforces power balance between supply and demand. We define aggregate generated and consumed power as

$$\sum P_{gen} = \sum P_{PV} + P_W + P_{BC} + P_{GS} \tag{7}$$

$$\sum P_{con} = \sum P_F + P_R + P_{BD} + P_{GB} \tag{8}$$

and the net power as

$$P_{net} = \sum P_{gen} - \sum P_{con}. \tag{9}$$

Here, P_{PV} and P_W are the PV and wind contributions, P_R and P_F denote residential and factory loads, P_{BC} and P_{BD} are battery charge/discharge powers, and P_{GS} and P_{GB} represent grid export (sell) and import (buy), respectively. The sign conventions are chosen such that $P_{net} \geq 0$ indicates a surplus (exporting is possible), while $P_{net} < 0$ indicates a deficit (importing is required).

The grid is treated as a slack bus with finite import/export capacities. When surplus power exists, export is limited by P_{GI} (maximum grid injection). If P_{net} exceeds this limit, excess generation must be curtailed:

$$P_{curt} = P_{net} - P_{GI} \quad \text{if } P_{net} > P_{GI}. \tag{10}$$

When a deficit exists, import is limited by P_{GE} (maximum grid extraction). If the deficit exceeds the import capability, unmet demand occurs:

$$P_{unmet} = P_{GE} - P_{net} \quad \text{if } P_{net} < P_{GE}. \tag{11}$$

The resulting quantities P_{curt} and P_{unmet} are key reliability/efficiency indicators and are explicitly penalized in the RL reward design.

B. Failure Modelling

To evaluate controller robustness under disturbances and rare events, we employ a time-varying failure rate model. The failure parameter θ is modulated by external conditions as

$$\theta = \theta_{base}\,[1 + \omega \max(0,\, \theta_{ext} - \theta_{thresh})] \tag{12}$$

where θ_{base} is the baseline failure rate, θ_{ext} is an external stress indicator, θ_{thresh} is a threshold above which external stress increases failure propensity, and ω controls sensitivity to externally induced failure.

Assuming a Poisson failure process, the probability of failure occurrence during an interval Δt is

$$P = 1 - e^{-\theta \Delta t}. \tag{13}$$

Following a failure event, the time-to-recover T is modeled as exponentially distributed with mean $MTTR$:

$$T \sim \text{Exp}(\lambda), \quad \lambda = \frac{1}{MTTR}. \tag{14}$$

This yields a parsimonious outage/repair mechanism that can be integrated into simulation rollouts to stress-test learned policies under reduced availability or islanded-like conditions.

C. Reinforcement Learning Controller

The EMS is formulated as a Markov decision process (MDP) in which the agent observes a state s_t and selects an action a_t at each time step to maximize expected discounted return. The state vector aggregates information required for real-time decisions, including (but not limited to) load levels (P_R, P_F), renewable availability (P_{PV}, P_W or their exogenous drivers), battery SOC and operational limits, and grid exchange limits. The action space is continuous and corresponds to real-valued setpoints for controllable power flows (e.g., battery charge/discharge command and/or grid import/export scheduling subject to saturation and SOC feasibility).

We implement the RL controller using the Stable-Baselines3 toolbox with actor–critic function approximation and a `MultiInputPolicy` to accommodate multi-field observations. We evaluate multiple widely used DRL algorithms—PPO, A2C, TD3, and SAC—using their standard library implementations [5]–[7]. For each method, separate neural networks are learned for the policy (actor) and value estimation (critic), and action outputs are clipped/projection-filtered to ensure physical feasibility (SOC bounds, power limits, and grid exchange constraints). Training is performed by rolling out the simulator over representative time-series scenarios and updating the policy to improve long-horizon performance.

D. Reward Function

The reward at each time step is designed to encode economic operation while strongly discouraging reliability violations and renewable wastage. The implemented reward is a weighted combination of four terms:

$$r_t = w_{\text{cost}}\, c_t - w_{\text{unmet}}\, P_{\text{unmet}} - w_{\text{curt}}\, P_{\text{curt}} - w_{\text{soc}}\, \Delta_{\text{SOC}} \tag{15}$$

where c_t is the total cost of operation (encompassing both operational expenses and repair costs), P_{unmet} is the total unmet power, P_{curt} is the total curtailed renewable power, and Δ_{SOC} is the total change in the State of Charge (SOC).

The weights $w_{(\cdot)}$ are constructed to balance the raw unit costs of each objective against the system's priorities. We assign specific unit penalty costs, $C_{\text{unmet}} = 3.5$ and $C_{\text{curt}} = 1.5$, to penalize unreliability and wastage, respectively. These costs are then scaled by priority factors (5.0, 10.0, and 0.1) to tune the agent's focus relative to the monetary cost c_t.

Table I
TRAINING CONFIGURATION USED FOR DRL AGENTS.

Item	Value
Episodes	1000
Episode horizon	7 days (168 h)
Control interval	60 min
Simulation step	60 min
Random seed	42
Reward constants	$C_{\text{unmet}} = 3.5,\ C_{\text{curt}} = 1.5$
Reward weights	$w_{\text{cost}} = 5,\ w_{\text{unmet}} = 10C_{\text{unmet}},\ w_{\text{curt}} = 0.1C_{\text{curt}},\ w_{\text{soc}} = 0$

In our implementation, the final weights are calculated as:

$$\begin{aligned} w_{\text{cost}} &= 5.0 \times 1.0 = 5.0 \\ w_{\text{unmet}} &= 10.0 \times C_{\text{unmet}} = 35.0 \\ w_{\text{curt}} &= 0.1 \times C_{\text{curt}} = 0.15 \\ w_{\text{soc}} &= 0.0 \end{aligned} \tag{16}$$

This formulation allows the agent to distinguish between the inherent cost of a violation (e.g., C_{unmet}) and the design priority assigned to minimizing that violation (e.g., the factor 10.0). The SOC deviation weight is currently set to zero ($w_{\text{soc}} = 0$) as a direct cost conversion for SOC fluctuations was not established for this experiment.

IV. RESULTS AND DISCUSSION

This section reports the performance of (i) a hand-crafted rule-based EMS and (ii) deep reinforcement learning (DRL) agents trained on the same Gym-compatible microgrid simulator. We present both synthetic scenarios (used for controlled sanity checks and stress-testing) and data-driven scenarios (used to expose the EMS to realistic intermittency and scaling). For all experiments, the simulator enforces operational constraints (SOC bounds, charge/discharge limits, and grid import/export limits) and includes component reliability/repair processes whose costs are accounted for in the operating cost.

A. Simulation and Training Setup

1) Episode structure and key metrics

Each episode spans a horizon of seven days with hourly control decisions. We report: (i) *total operating cost* (including operating expenses and repair/maintenance costs, with negative values indicating net revenue under the adopted sign convention), (ii) *unmet energy* (energy not served due to infeasible power balance under finite import/export and component outages), (iii) *curtailed energy* (renewable energy discarded due to surplus under finite export/storage), and (iv) qualitative reliability indicators such as downtime events and security-of-supply violations.

2) Microgrid sizing and operating costs

To make the simulation configuration explicit (and avoid embedding code in the manuscript), Table II summarizes the component ratings, unit costs, and base reliability parameters used throughout the reported experiments.

3) DRL training time

Table III reports the wall-clock training time observed for each algorithm for the fixed training above.

Table II
MICROGRID CONFIGURATION AND RELIABILITY PARAMETERS USED IN THE SIMULATOR (RATINGS, COSTS, AND REPRESENTATIVE REPAIR ASSUMPTIONS).

Component	Rating / Limit	Cost model	Base fail rate	Minor frac.	Minor derate	MTTR (minor/major)
PV	350 kW	0.02 $/kWh (O&M)	0.05/h	0.6	0.5	2 h / 8 h
Wind	200 kW	0.027 $/kWh (O&M)	0.08/h	0.7	0.6	6 h / 16 h
Diesel	0–200 kW	0.45 $/kWh + 1 $/h	0.05/h	0.5	0.7	4 h / 12 h
Battery	1644 kWh; ±103 kW	0.086 $/kWh (degradation proxy)	0.01/h	0.9	0.95	1 h / 4 h
Grid	±1000 kW	0.20 $/kWh (import/export)	0.01/h	0.8	0.9	2 h / 8 h

Table III
OBSERVED TRAINING TIME FOR EACH DRL ALGORITHM.

Algorithm	Training time
PPO	52 min 31.1 s
A2C	39 min 20.2 s
SAC	96 min 6.2 s
TD3	77 min 0.3 s

B. Rule-Based Control with Synthetic Data

We first evaluate a deterministic rule-based EMS to establish an interpretable baseline and to illustrate how explicit schedules interact with intermittency, grid constraints, and islanding. The three synthetic scenarios are designed to progressively stress feasibility: (i) normal sunny operation, (ii) forced night islanding, and (iii) scheduled grid purchasing.

1) Sunny-day scenario

In the sunny scenario (Fig. 8), the EMS charges the battery during the PV-rich midday window (11:00–15:00) and discharges during the evening peak window (18:00–21:00). The diesel generator is scheduled to run overnight (20:00–06:00) at a fixed setpoint (4.0 kW), while the grid remains available without islanding constraints. Under these conditions the rule-based policy achieved **no downtime** and a total cost of **$–80.86**, reflecting net revenue under the adopted sign convention.

2) Night-islanding scenario

In the islanded scenario (Fig. 9), the grid is disconnected during 22:00–04:59, forcing the microgrid to satisfy demand using local generation and storage. The battery discharge window is shifted to 22:00–04:59 and the diesel setpoint is reduced to 2.0 kW (20:00–06:00). This scenario highlights the primary limitation of fixed schedules under constraints: when islanding coincides with low renewable availability and limited dispatchable output, the system can experience *both* renewable curtailment (surplus at times when export is unavailable and storage saturates) and residual unmet energy during deficit periods. The logged totals were **$–74.19** and **5.34 kWh unmet energy**, consistent with the security-of-supply excursions visible in the overview plot.

3) Scheduled grid purchasing scenario

In the scheduled purchasing scenario (Fig. 10), the EMS imposes a fixed grid import setpoint during 02:00–04:59 (setpoint of -8.0 kW). Despite this schedule, the controller may still draw additional power outside the scheduled window when load demand requires it (subject to grid limits), and it exports energy during surplus periods. This policy produced **no downtime**, **0.00 kWh unmet energy**, and a total cost of **$–102.37**, indicating that scheduled purchasing plus opportunistic selling can improve net revenue when sufficient surplus occurs and the grid is available.

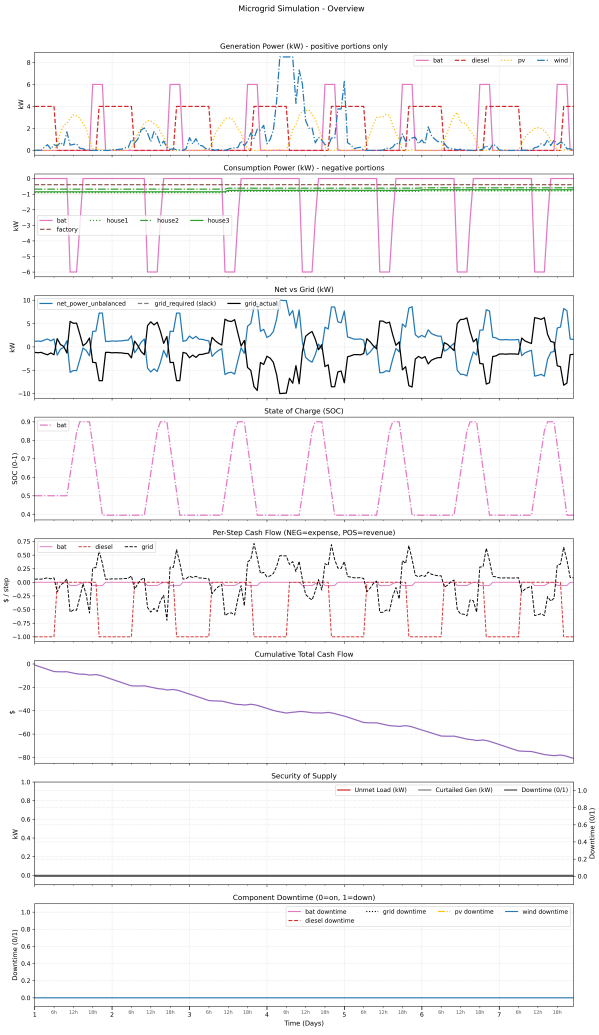

Figure 8. Rule-based control results overview on a sunny day scenario.

Table IV
SUMMARY OF RULE-BASED SYNTHETIC SCENARIOS. NEGATIVE TOTAL COST INDICATES NET REVENUE UNDER THE SIGN CONVENTION USED IN THE SIMULATOR.

Scenario	Total cost ($)	Unmet energy (kWh)
Sunny (scheduled charge/discharge + diesel)	-80.86	0.00
Night islanding (22:00–04:59)	-74.19	5.34
Scheduled grid buy (02:00–04:59)	-102.37	0.00

Key observation: These results confirm that hand-crafted schedules can be effective when operating conditions match

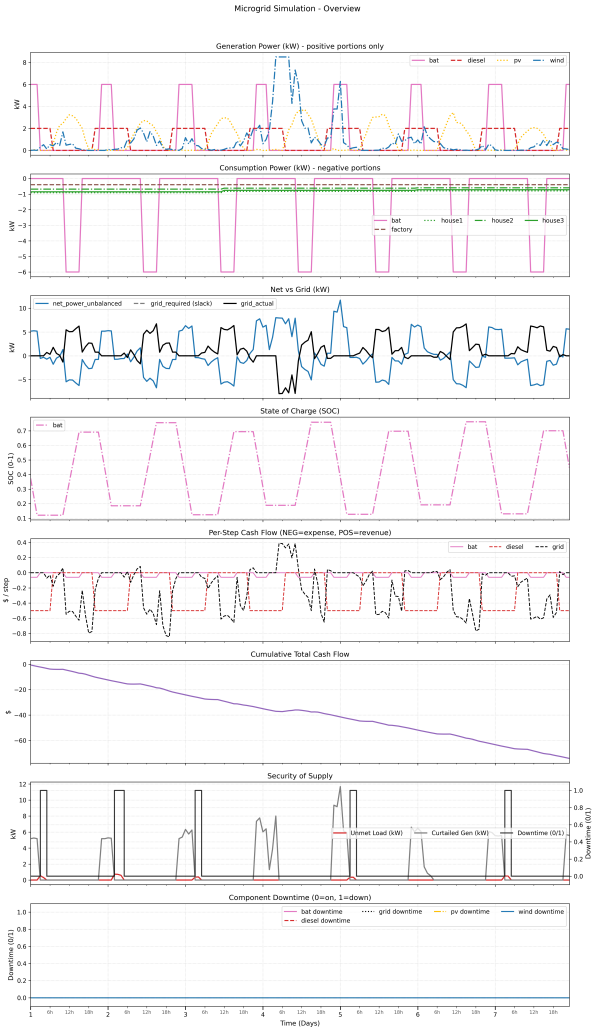

Figure 9. Rule-based control results overview on a scenario where the grid is islanded at night.

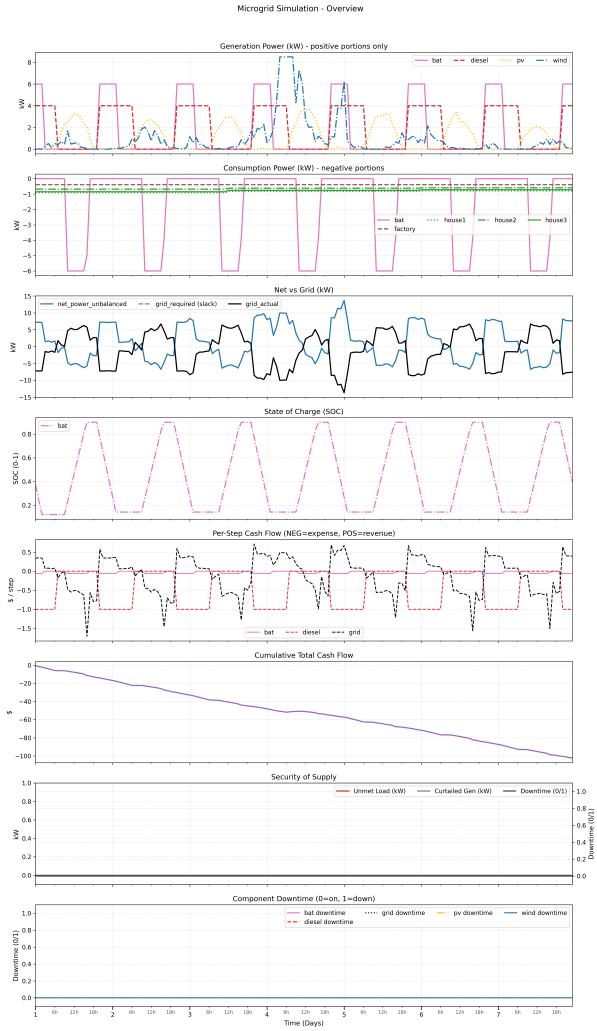

Figure 10. Rule-based control results overview on a scenario where power is bought from the grid at a scheduled time only (02:00am - 04:59am).

design assumptions, but performance can degrade under islanding and intermittency because the controller does not adapt its actions to the realized trajectory of renewables, load, and component outages.

C. Rule-Based Control with Real Datasets

We next evaluate the rule-based policy under two data-driven scenarios to highlight scaling effects and the importance of tuning rule magnitudes to the power level of the underlying dataset.

For Liège (Fig. 12), the battery charge/discharge magnitudes are small (charge -0.20 kW and discharge 0.15 kW) with SOC triggers (0.1/0.9) and scheduled operation windows (charge 09:00–14:00; discharge 21:00–04:59). For Mesa Del Sol (Fig. 11), the rule magnitudes are significantly larger (charge -30 kW and discharge 10 kW) with windows (charge 08:00–17:00; discharge 17:00–22:00). In both cases, the EMS achieved **0.00 kWh unmet energy**. However, the total cost differs strongly: **$–2489.59** (Mesa) versus **$–0.04** (Liège). This discrepancy is expected because the datasets operate at very different power scales (e.g., PV generation below 0.5 kW in one case versus tens of kW in the other), and cost accumulates roughly proportionally to energy throughput and trading volume.

Key observation: Rule magnitudes and schedules are not transferable across datasets without retuning. When the underlying power scale changes by orders of magnitude, fixed heuristics can remain feasible yet yield incomparable economic outcomes.

D. RL Control

We now assess learning-based EMS policies. Unless stated otherwise, all DRL agents are trained using the reward in Section III-E and evaluated on the same simulator configuration.

1) Random policy baseline

As a stress baseline, a random policy is evaluated in Fig. 13. The resulting behavior is characterized by frequent action changes, pronounced SOC fluctuations, intermittent grid islanding periods (observable as intervals where grid exchange becomes flat at zero), occasional diesel dispatch, and sporadic exporting. This baseline produces unstable operation and frequent security-of-supply excursions, confirming that naive

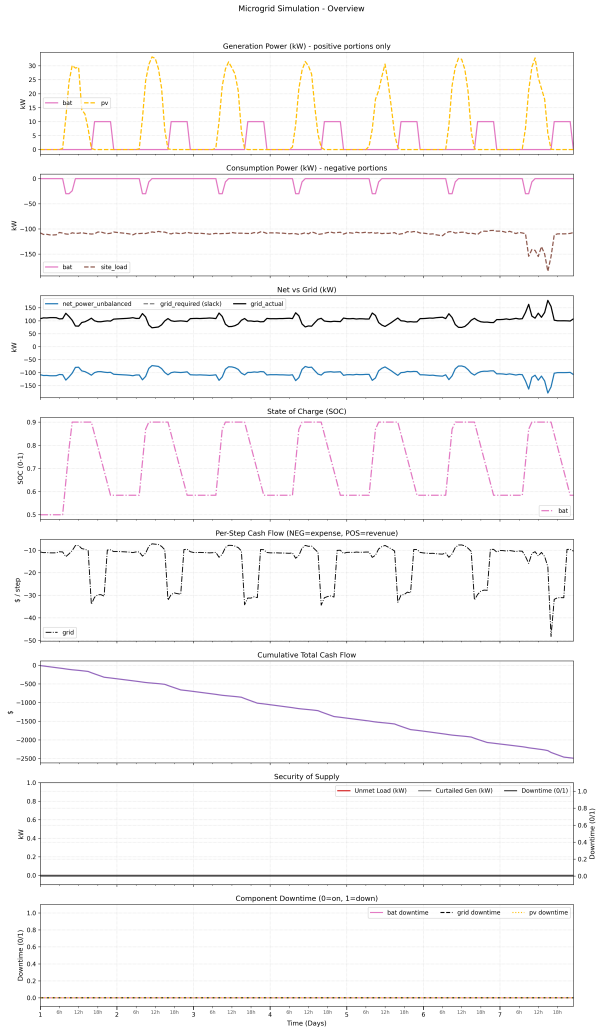

Figure 11. Rule-based control results overview on Mesa Del Sol dataset.

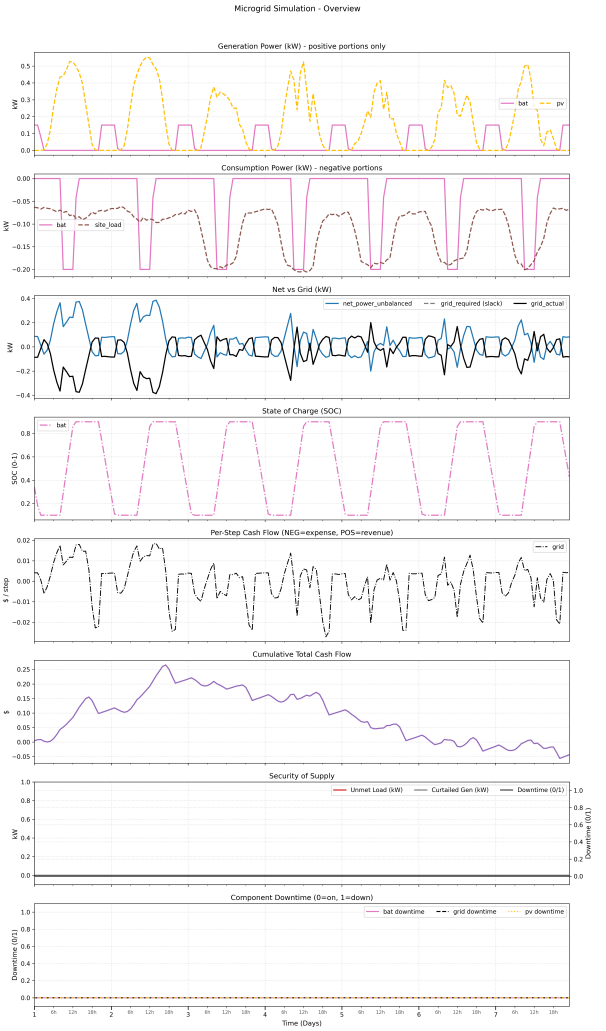

Figure 12. Rule-based control results overview on Liege dataset.

exploration is insufficient for reliable EMS operation in constrained microgrids.

2) PPO

PPO exhibits a relatively steady improvement in episodic reward during training (Fig. 14). In evaluation (Fig. 15), the learned policy avoids islanding actions and adopts a conservative strategy: it slowly charges the battery over the week-long horizon and rarely discharges it. Despite occasional component failures, the policy maintains supply adequacy without producing downtime events, suggesting that the learned policy relies primarily on grid availability and conservative reserve accumulation rather than aggressive arbitrage.

3) A2C

A2C shows a faster rise in episodic reward (Fig. 16) and is the fastest to train under the reported budget (Table III). In evaluation (Fig. 17), A2C learns to dispatch the diesel generator more actively while still charging the battery gradually and rarely using it for discharge. Similar to PPO, the policy avoids grid islanding actions and maintains reliable operation despite component failures.

4) SAC

SAC reaches a strong reward region early in training (Fig. 18). In evaluation (Fig. 19), SAC learns a qualitatively different strategy from PPO/A2C: it largely avoids diesel operation and instead coordinates battery charging and discharging to buffer renewable variability. This behavior is consistent with a policy that uses storage as the primary flexibility resource, while still maintaining reliability and exhibiting no downtime even when some components enter failure states.

5) TD3

TD3 training rewards (Fig. 20) show intermittent large negative drops despite generally strong performance. In evaluation (Fig. 21), TD3 quickly charges the battery and then uses it minimally, while also avoiding diesel operation. The learned policy maintains feasibility and avoids downtime despite component failures, but the reward volatility suggests sensitivity to rare events and/or occasional policy actions that trigger large penalties (e.g., transient unmet load or curtailment spikes).

E. Comparison and Key Observations

Table V summarizes the dominant qualitative behaviors observed across the evaluated controllers. A consistent trend

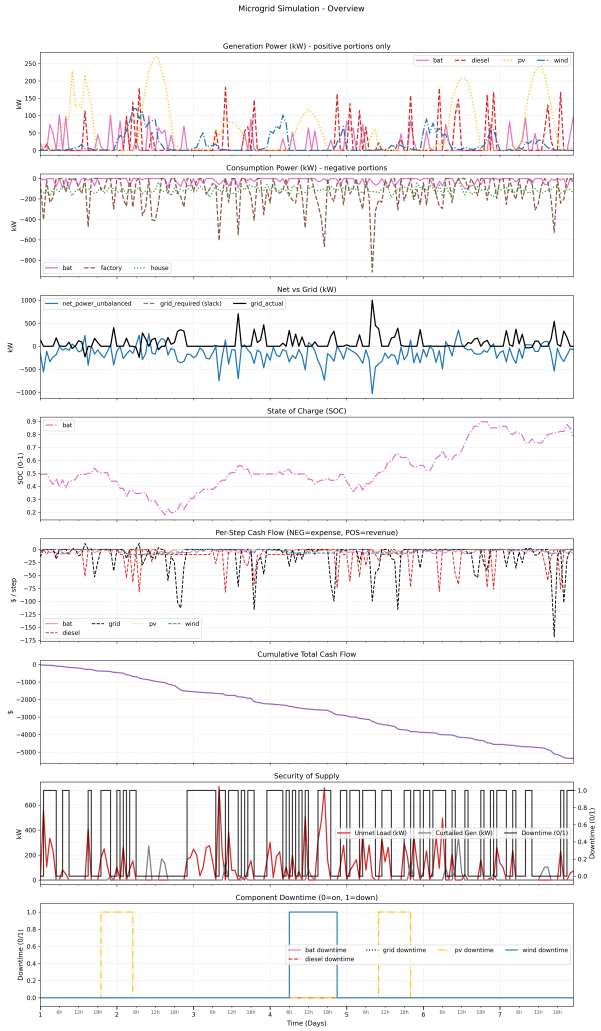

Figure 13. Random policy results overview on evaluation scenario.

Figure 14. PPO reward progression over training episodes.

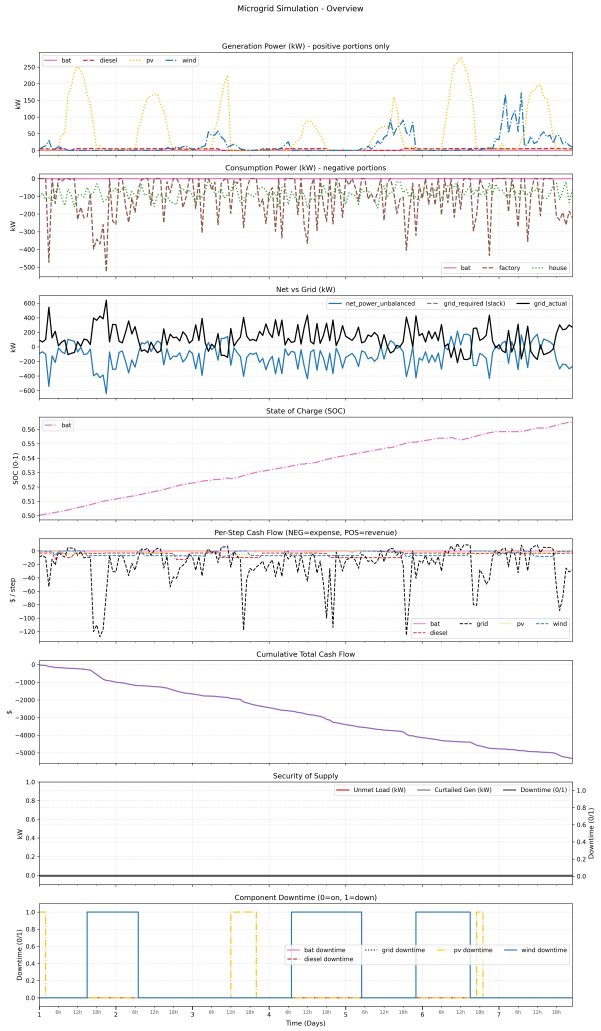

Figure 15. PPO results overview on evaluation scenario.

Figure 16. A2C reward progression over training episodes.

across PPO/A2C/TD3 is a conservative reliance on grid availability and reserve accumulation (charging without substantial discharge), while SAC more actively exploits storage to offset variability and reduce reliance on diesel.

This subsection presents a quantitative comparison of the evaluated controllers using cumulative metrics computed over the full evaluation horizon. Table VI summarizes the total operating cost, unmet energy, curtailed energy, and cumulative test return obtained by each policy.

Several important observations can be drawn from these results:

- **Reliability and feasibility:** All trained DRL controllers (PPO, A2C, SAC, and TD3) achieved *zero unmet energy and zero renewable curtailment* over the evaluation horizon. This confirms that the learned policies successfully respect the power-balance constraints and operate within grid import/export and storage limits. In contrast, the random policy incurred severe reliability violations, with approximately 15.8 MWh of unmet demand and 1.94 MWh of curtailed energy, rendering it operationally infeasible despite its comparable raw operating cost.
- **Cost versus reward interpretation:** The "Total Cost"

Table V
QUALITATIVE COMPARISON OF CONTROLLER BEHAVIORS (AS OBSERVED IN THE PLOTTED EVALUATION ROLLOUTS).

Controller	Battery cycling	Diesel usage	Avoids islanding
Rule-based (varies by scenario)	Scheduled	Scheduled	Scenario-dependent
Random policy	Erratic	Occasional	No
PPO	Charge-heavy, low discharge	Rare	Yes
A2C	Charge-heavy, low discharge	Yes	Yes
SAC	Active charge/discharge	Minimal	Yes
TD3	Fast charge, low discharge	Minimal	Yes

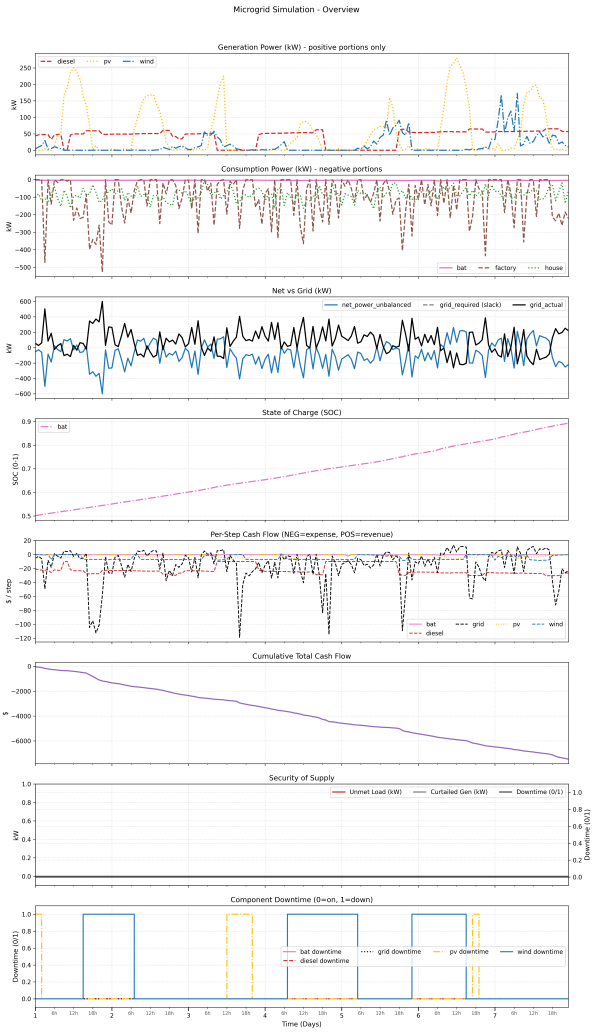

Figure 17. A2C results overview on evaluation scenario.

Figure 18. SAC reward progression over training episodes.

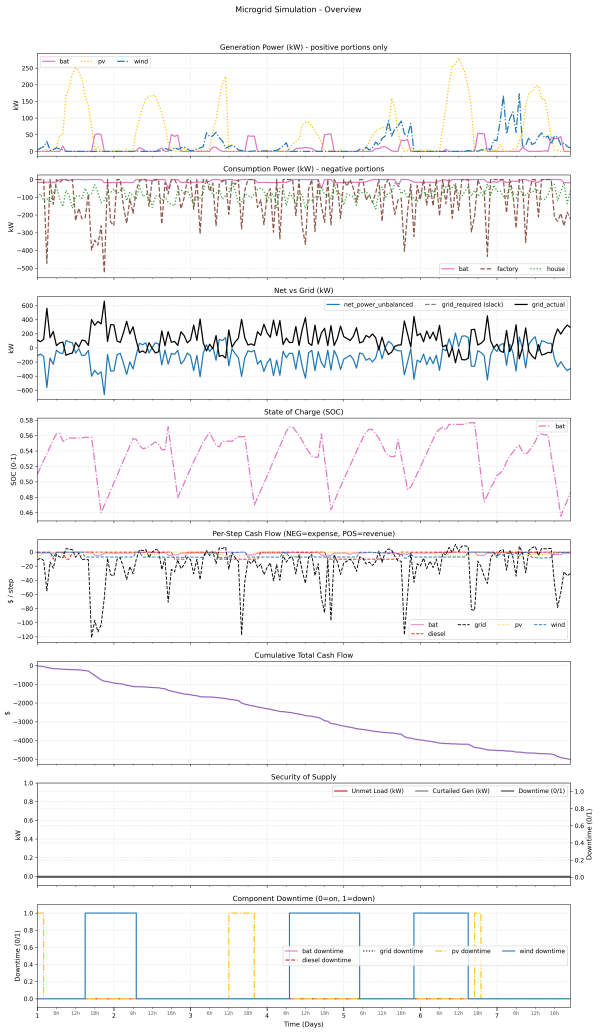

Figure 19. SAC results overview on evaluation scenario.

Figure 20. TD3 reward progression over training episodes.

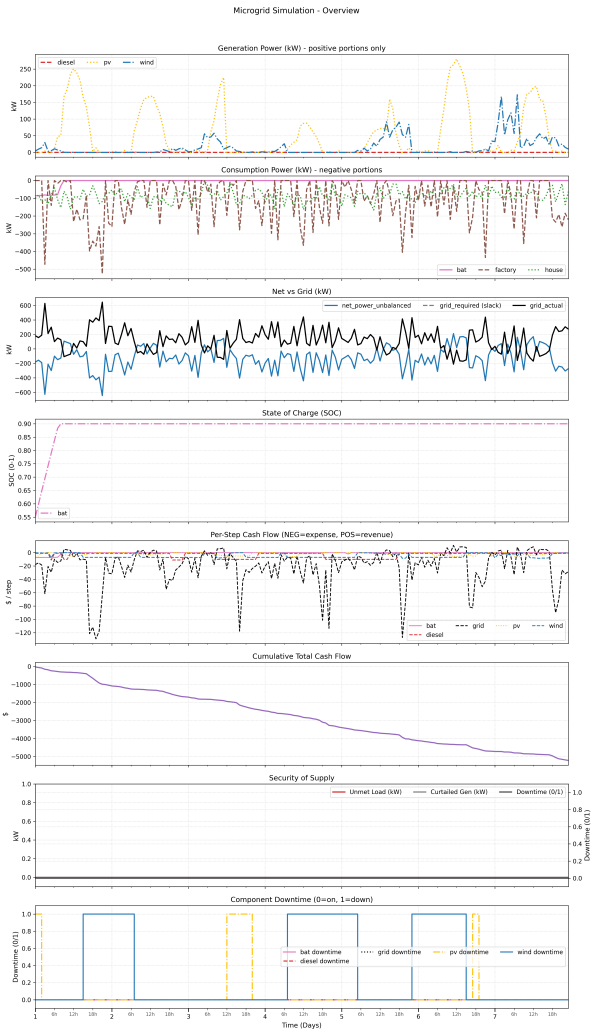

Figure 21. TD3 results overview on evaluation scenario.

Table VI
QUANTITATIVE COMPARISON OF EMS CONTROLLERS OVER THE EVALUATION HORIZON. NEGATIVE TOTAL COST VALUES FOLLOW THE SIMULATOR SIGN CONVENTION AND CORRESPOND TO HIGHER NET OPERATING EXPENSE.

Algorithm	Total Cost ($)	Unmet Energy (kWh)	Curtailed Energy (kWh)	Test Reward
PPO	−5,319.21	0.000	0.000	−26,596.03
A2C	−7,469.94	0.000	0.000	−37,349.71
SAC	−5,017.02	0.000	0.000	−25,085.12
TD3	−5,212.88	0.000	0.000	−26,064.41
Random	−5,365.85	15,798.16	1,053.46	−1,580,514.05

column reports cumulative operating and repair costs using the simulator sign convention, where more negative values correspond to higher net expenditure. However, this metric alone does not capture reliability. The test return directly reflects the RL objective in (**??**), where unmet demand and curtailment are heavily penalized. Consequently, the random policy—while appearing comparable in cost—exhibits an extremely poor test return due to the dominant penalties associated with unmet load and wasted energy.

- **Comparison among DRL algorithms:** Among the trained controllers, SAC achieves the highest (least negative) test return, indicating the best overall trade-off under the chosen reward formulation. PPO and TD3 yield similar performance, with slightly lower returns, while A2C exhibits the lowest test return among the trained agents. This trend is consistent with the observed qualitative behaviors: SAC actively coordinates battery charging and discharging to buffer renewable variability and avoid diesel usage, whereas PPO, A2C, and TD3 tend toward more conservative strategies that rely on grid availability and reserve accumulation.
- **Economic performance:** While A2C results in the most negative total cost value, this does not translate into superior reward performance. This discrepancy highlights the importance of multi-objective evaluation in microgrid EMS: minimizing monetary cost alone can be misleading if it is achieved through aggressive or inefficient dispatch patterns that do not align with the weighted objectives embedded in the RL reward.
- **Overall:** these results demonstrate that continuous-control DRL agents can reliably satisfy demand and eliminate curtailment while achieving competitive operating costs. In particular, SAC provides the most balanced performance across economic efficiency and reliability in the studied scenarios, whereas naive policies that ignore system constraints can appear economically attractive while being fundamentally unacceptable from a power-system perspective.

V. CONCLUSION AND FUTURE WORK

This paper presented a continuous control RL EMS for a hybrid microgrid integrating photovoltaic and wind generation, battery energy storage, a dispatchable generator, and grid import/export. The EMS problem was formulated as a continuous Markov decision process and implemented in an OpenAI Gym-compatible simulation environment that enforces operational constraints like state-of-charge bounds, charge/discharge limits, and grid trading limits while capturing time-varying renewable availability and heterogeneous residential and industrial demand. Using a rule-based controller as a baseline, we evaluated modern continuous-control DRL algorithms PPO, TD3, and SAC to quantify their capability to learn coordinated storage dispatch and grid-trading actions.

The results demonstrate that random actions policies produce frequent power-balance violations, unmet load events, and unstable operation, underscoring the difficulty of microgrid EMS under uncertainty. In contrast, trained RL policies improve supply adequacy and sustain power balance more consistently in the studied scenarios by learning to allocate renewable generation, storage charging/discharging, and grid import/export in a coordinated manner. The comparative analysis also indicates that algorithm choice influences policy smoothness and operational trade-offs, and that observed performance is sensitive to environment assumptions, reward shaping, and the fidelity of component models. These findings support the feasibility of continuous control RL for microgrid EMS while highlighting the need for careful benchmarking and realism to ensure deployment relevance.

Future work will focus on increasing modeling fidelity and strengthening evaluation rigor. First, we will incorporate richer contingency modeling, including component-specific failure modes, external-stress dependent outage rates, repair costs, and stochastic repair-time distributions, enabling resilience oriented training and assessment. Second, we will integrate degradation-aware storage models that capture cycle aging and calendar aging effects, allowing the EMS to explicitly optimize lifecycle cost rather than short-term energy arbitrage alone. Third, we will tighten coupling between the EMS layer and lower-level microgrid dynamics by interfacing with voltage/frequency control and converter constraints, enabling evaluation of how EMS decisions interact with stability margins during both grid-connected and islanded transitions.

VI. Data and Code Availability

All source code, simulation environments, and supporting materials for this work are publicly available in the GitHub repository titled microgrid-control-sim[1]. The repository includes a simple `README` file with instructions, enabling full reproduction of the experimental results and facilitating adaptation for future research projects.

References

[1] R. H. Lasseter, "MicroGrids," in *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings*, vol. 1, New York, NY, USA, Jan. 2002, pp. 305–308.

[2] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, and N. D. Hatziargyriou, "Trends in microgrid control," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1905–1919, 2014.

[3] E. Kuznetsova, Y.-F. Li, C. Ruiz, and E. Zio, "Reinforcement learning for microgrid energy management," *Energy*, vol. 59, pp. 133–146, 2013.

[4] G. Henri, T. Levent, A. Halev, R. Alami, and P. Cordier, "pymgrid: An open-source python microgrid simulator for applied artificial intelligence research," arXiv:2011.08004, 2020. [Online]. Available: https://arxiv.org/abs/2011.08004

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347, 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[6] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, Jul. 2018, pp. 1587–1596. [Online]. Available: https://proceedings.mlr.press/v80/fujimoto18a.html

[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, Jul. 2018, pp. 1861–1870. [Online]. Available: https://proceedings.mlr.press/v80/haarnoja18b.html

[8] J. Dumas, S. Dakir, C. Liu, and B. Cornélusse, "Coordination of operational planning and real-time optimization in microgrids," *Electric Power Systems Research*, vol. 190, p. 106634, 2021.

[9] T. A. Nakabi and P. Toivanen, "Deep reinforcement learning for energy management in a microgrid with flexible demand," *Sustainable Energy, Grids and Networks*, vol. 25, p. 100413, 2021.

[10] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 42, pp. 1437–1480, 2015. [Online]. Available: http://jmlr.org/papers/v16/garcia15a.html

[11] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, Aug. 2017, pp. 22–31. [Online]. Available: https://proceedings.mlr.press/v70/achiam17a.html

[12] J. Vetter, P. Novák, M. R. Wagner, C. Veit, K.-C. Möller, J. O. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, and A. Hammouche, "Ageing mechanisms in lithium-ion batteries," *Journal of Power Sources*, vol. 147, no. 1–2, pp. 269–281, 2005.

[13] S. Heid *et al.*, "OMG: A scalable and flexible simulation and testing environment toolbox for intelligent microgrid control," *Journal of Open Source Software*, vol. 5, no. 53, p. 2435, 2020.

[14] H. M. Ridha, C. Gomes, H. Hizam, M. Ahmadipour, A. A. Heidari, and H. Chen, "Multi-objective optimization and multi-criteria decision-making methods for optimal design of standalone photovoltaic system: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110202, 2021.

[15] F. Spertino, P. D. Leo, V. Cocina, and G. M. Tina, "Storage sizing procedure and experimental verification of stand-alone photovoltaic systems," in *2012 IEEE International Energy Conference and Exhibition (ENERGYCON)*, 2012, pp. 464–468.

[16] A. Bashir, C. Leap, A. Blumenthal *et al.*, "Power, voltage, frequency and temperature dataset from mesa del sol microgrid," Dryad Dataset, 2023. [Online]. Available: https://doi.org/10.5061/dryad.fqz612jzb

[17] S. Semaoui, A. H. Arab, S. Bacha, and B. Azoui, "Optimal sizing of a stand-alone photovoltaic system with energy management in isolated areas," *Energy Procedia*, vol. 36, pp. 358–368, 2013.

[1]Link: https://github.com/olanrewajufarooq/microgrid-control-sim