SEQUENTIAL DECISION MAKING AND REINFORCEMENT LEARNING CS 236018

FINAL PROJECT

DATE: MARCH 1, 2025

LECTURER: SARAH KEREN

SUBMITTED BY: ALMOG ANSCHEL
EDEN HINDI

Abstract

This project is part of the **SDRML** course of Technion's CS Faculty. We explore the application of RL for optimizing battery storage operations in a dynamic electricity market. Our work focuses on developing and evaluating an RL-based agent capable of efficiently managing energy storage by responding to stochastic demand fluctuations and real-time market price variations. To thoroughly assess our approach, we experiment with three demand models and two reward formulations, allowing us to capture diverse market conditions.

We begin by benchmarking a standard Soft Actor-Critic (SAC) agent and then introduce an enhanced version with a lookahead critic, utilizing $TD(n)$ learning to improve long-term reward estimation. This modification is designed to mitigate overestimation bias and enhance learning stability. Our experimental results indicate that the lookahead critic leads to more stable training and improved decision-making in settings with smooth demand variations, such as Gaussian models. However, we observe only modest improvements in highly dynamic environments, such as those governed by sinusoidal demand.

1 Methodology

1.1 Environment

The simulated environment models an electricity market wherein an agent manages a battery storage system. The state of the environment is represented by a continuous vector:

$$\text{State} = \begin{bmatrix} \text{SoC} & D_t & P_t \end{bmatrix},$$

where **SoC** denotes the battery's state of charge, D_t represents the household electricity demand, and P_t is the market price at time step t. The dynamics of the environment are captured by periodic functions with added stochastic noise. In our implementation, three demand functions are considered (a mixture of Gaussians, a sinusoidal function, and a step function) to simulate various usage scenarios.

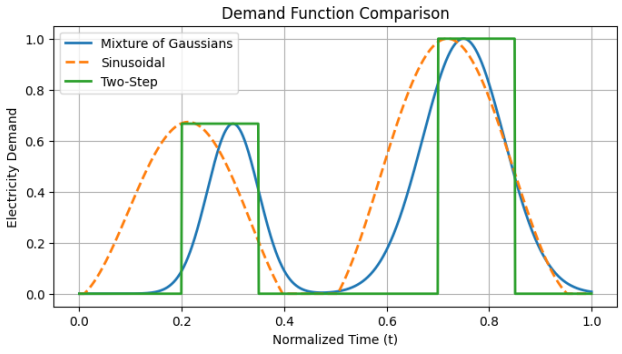

Figure 1: Visualization of the three demand functions

The market price is modeled as a linear function of demand:

$$P_t = m \cdot D_t + b + \mathcal{N}(0, \sigma)$$

where m and b are constants ensuring a proportional relationship between demand and price.

The agent operates in a continuous action space $A \in [-C, C]$, where C is the battery capacity. Positive actions correspond to charging the battery, while negative actions indicate discharging. The evolution of the battery's state is subject to capacity constraints and directly affects the reward structure. Two reward formulations are employed:

- **Profit:** $R = P_t \times$ sold energy - Rewards the agent solely based on the revenue.
- **Internal Demanad**: $R = P_t \times$ discharge amount $- \lambda \times \max(0, D_t -$ discharge amount$)$ - This encourages the agent to prioritize internal demand before selling energy.

1.2 Agent Architecture

Our agent is based on the Soft Actor-Critic (SAC) algorithm, which maximizes a maximum entropy objective to balance exploitation and exploration. The objective is given by:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right) \right]$$

where α controls the trade-off between reward maximization and entropy.

1.2.1 Standard Action Critic

The standard critic in SAC is tasked with estimating the soft Q-function:

$$Q^{\pi}(s_t, a_t) = \mathbb{E} \left[\sum_{k=t}^{\infty} \gamma^{k-t} \left(r(s_k, a_k) + \alpha \mathcal{H}(\pi(\cdot|s_k)) \right) \middle| s_t, a_t \right].$$

This estimation is achieved through Q-networks. In our implementation, we used twin critics to mitigate overestimation bias by taking the minimum of the two Q-value estimates. The update rule is reinforced by a soft target network update:

$$\theta_{\text{target}} \leftarrow \tau\theta + (1 - \tau)\theta_{\text{target}}$$

ensuring stability during training.

1.2.2 Lookahead Critic

During our experimentation, we observed high variance in some training runs, which motivated us to explore strategies for stabilizing the learning process. One key takeaway from the course was that $TD(0)$ introduces minimal variance, making it a robust choice for learning updates. We extended the concept to the $TD(n)$ method, integrating it into our Soft Actor-Critic (SAC). This resulted in the development of our lookahead critic which improves the estimation of long-term rewards.

Unlike traditional one-step temporal difference updates, which may suffer from high variance and limited foresight, the lookahead critic enables the network to anticipate rewards over a longer horizon. The critic "looks 10 steps ahead," leading to a more stable Q-value estimation. More importantly, instead of simply providing local value approximations, the critic now serves as a "guiding mechanism" for the actor, promoting more informed action selection.

The lookahead critic computes the Q-value target using $TD(\text{n})$:

$$y_t = \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n \left[Q(s_{t+n}, a_{t+n}) - \alpha \log \pi(a_{t+n}|s_{t+n}) \right],$$

where the accumulated rewards over n steps are combined with a bootstrapped Q-value at the n-th step.

By incorporating the lookahead critic, our agent should anticipate delayed rewards more accurately. This results in a more reliable Q-value estimation, which in turn guides the policy update more effectively.

2 Experiments

Firstly, we conducted an analysis of six trained agents, each utilizing different combinations of reward structures and demand models. Initially, we observed that agents optimized for profit exhibited high variance in their performance. However, we later identified that this was likely due to including early training rewards in our evaluation, before the policies had stabilized.

To ensure a meaningful comparison, we evaluated three demand functions under the profit reward structure, with an additional agent using the internal reward type with a Gaussian demand model. This setup allowed us to analyze how different demand patterns influence performance under each reward formulation.

Fig 2 visualize the accumulative rewards for all agents.

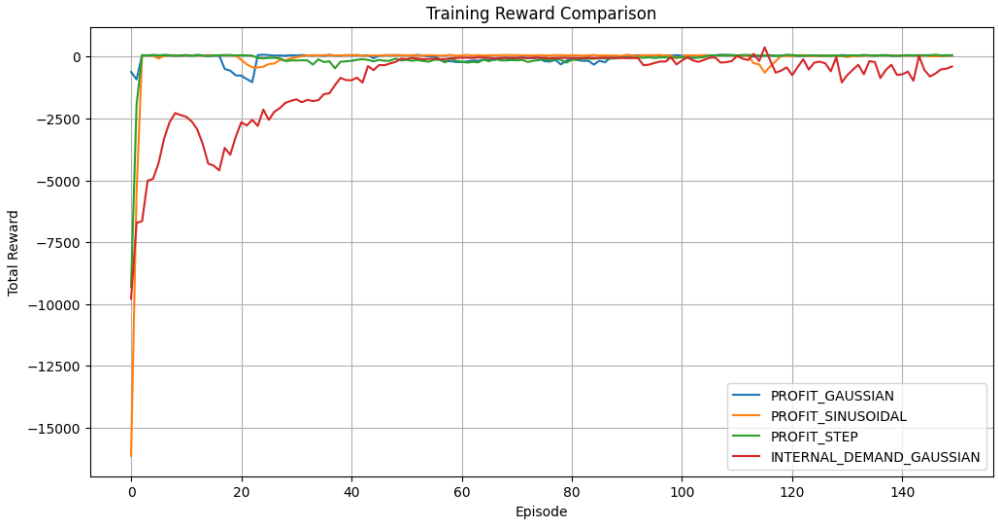

Figure 2: Accumulative rewards

Following this, Table 1 provides a summary of the results. From the table, we observe that each model

Reward Type	Demand Type	Avg Reward	Std Reward	Avg Reward (Eval)	Final Entropy
Profit	Gaussian	**-5.51**	142.17	**404.97**	0.016344
Profit	Sinusoidal	-121.87	449.99	521.62	**0.000482**
Profit	Step	-3.89	**75.32**	-10608.63	0.001561
Internal	Gaussian	-595.48	704.26	-1846.02	0.022507

Table 1: Comparison of SAC performance across different demand models and reward types.

exhibits unique advantages. While some configurations maximize profit, others demonstrate more stable behavior with lower variance.

Additionally, detailed training loss and critic loss curves for each agent can be found in the appendix.

To see if the lookahead critic got it right, we compare its performance in two different environments: profit with Gaussian demand and profit with Sinusoidal demand. This comparison allows us to analyze how the lookahead mechanism influences stability, reward accumulation, and policy efficiency across different demand patterns.

We hypothesize that the lookahead critic improves learning stability by reducing variance in Q-value estimation, leading to more consistent policy updates.

Figure 3 and Figure 4 visualizes the performance trends of both agents. We observe that the lookahead critic leads to more stable learning in the Gaussian demand setting, while in the Sinusoidal demand model, the agent managed to slightly outperform the vanilla one.

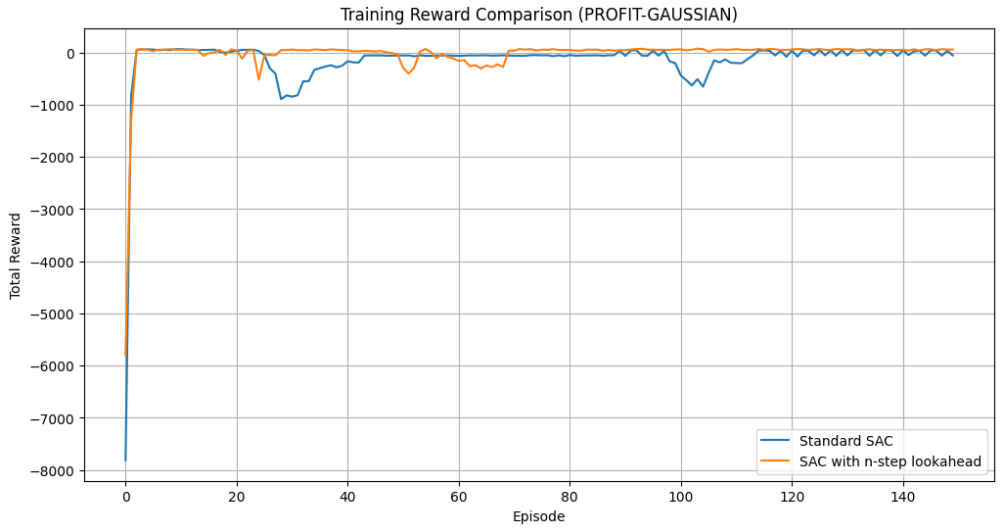

Figure 3: Performance comparison of SAC with and without the lookahead critic with Gaussian as demand models

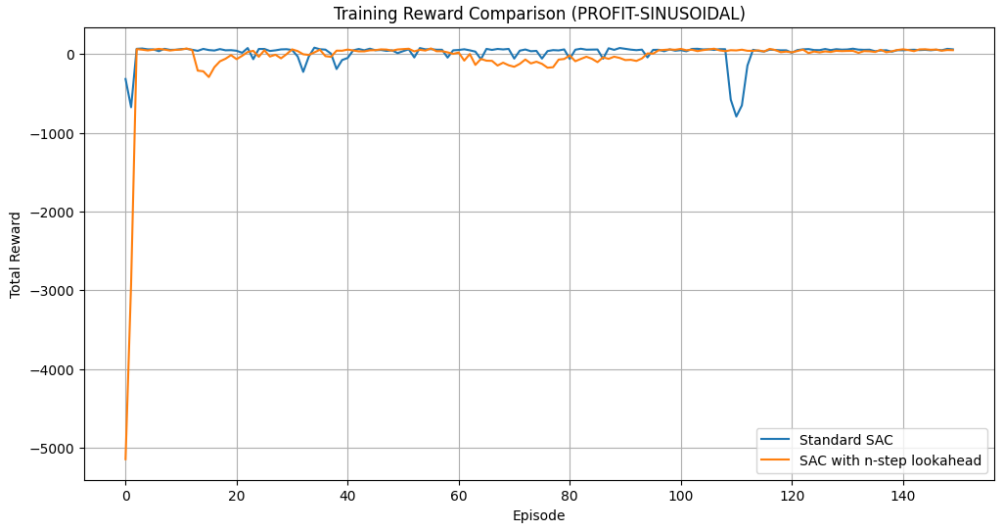

Figure 4: Performance comparison of SAC with and without the lookahead critic with Sinusoidal as demand models

To further quantify these findings, we provide a numerical comparison in Table 2, summarizing the key performance metrics.

Agent	Demand	Avg Reward	Std Reward	Avg Reward (Eval)
Standard	Gaussian	-69.727698	131.936480	-955.442219
Lookahead	Gaussian	**10.563807**	**106.514852**	**556.124964**
Standard	Sinusoidal	**20.250673**	128.490936	521.230944
Lookahead	Sinusoidal	-1.026224	**66.719927**	**562.418743**

Table 2: Comparison of SAC performance across different demand models and reward types.

These results suggest that the lookahead critic plays a significant role in stabilizing Q-value estimation, particularly in environments where demand patterns are less oscillatory. While its benefits are evident in Gaussian demand settings, further tuning might be required for environments with high periodicity like the Sinusoidal model.

3 Conclusion

In this work, we explored reinforcement learning-based optimization for battery storage management in dynamic electricity markets. We implemented an SAC-based agent with a lookahead critic that leverages $TD(n)$ learning to improve reward estimation and decision-making. Our results suggest that incorporating the lookahead critic improves training stability and performance, particularly in environments with smoother demand patterns such as Gaussian models.

However, our findings also highlight challenges in applying this approach across varying demand functions. While the lookahead critic demonstrated clear benefits in the Gaussian demand setting, it also showed slight improvements over the standard SAC agent in the Sinusoidal case. Additionally, due to potential implementation errors and architectural choices throughout the development process, we cannot fully ensure the reliability of all performance gains observed, as seen in high variances through the project.

Future work should aim to refine the lookahead critic by exploring adaptive step sizes, improved reward normalization techniques, and alternative architectures such as Transformer-based critics for capturing long-term dependencies.

References

[1] Kristopher De Asis, Juan F Hernandez-Garcia, G Zacharias Holland, and Richard S Sutton. Multi-step reinforcement learning: A unifying algorithm. *arXiv preprint arXiv:1703.01327*, 2017.

[2] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[3] CLAIR-LAB TECHNION. Sdmrl - course material. GitHub Repository, 2024. Available at https://github.com/CLAIR-LAB-TECHNION/SDMRL.

GitHub Link:

Project Repository

A Results of Runs

A.1

standard reward and critic losses of the a regular agent training on a 4 environments setups.

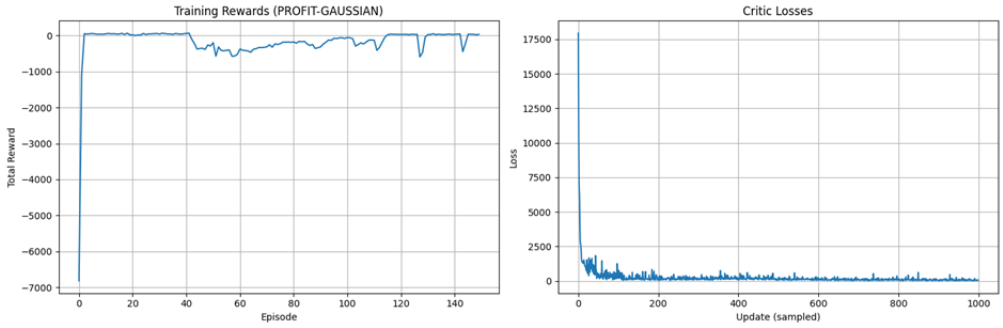

Figure 5: Agent training with Profit as reward and Gaussian as demand

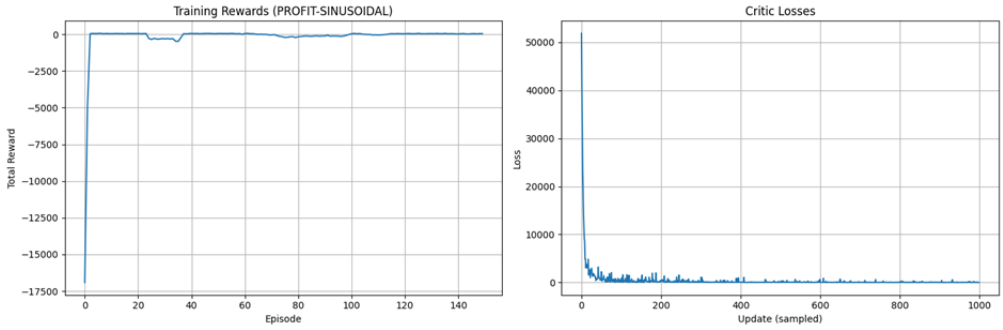

Figure 6: Agent training with Profit as reward and Sinusoidal as demand

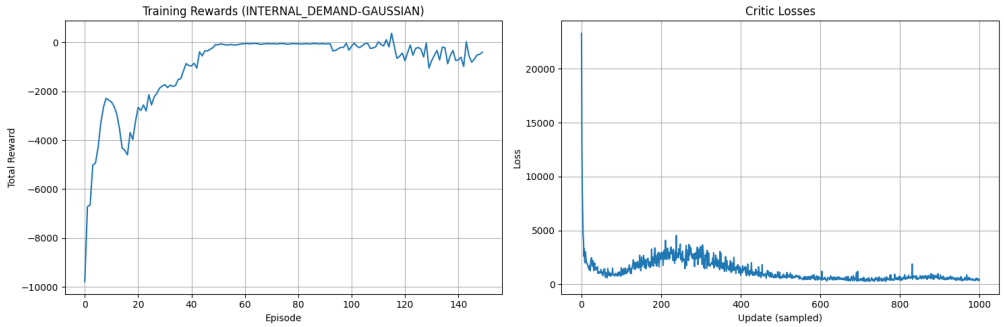

Figure 7: Agent training with Internal as reward and Gaussian as demand

A.2

We evaluated an agent across six different environment configurations to analyze how it responds to battery charge (SoC), electricity price, and demand fluctuations. In most cases, the agent exhibited the expected behavior—charging when prices were low and discharging during periods of high demand. However, the step function demand model with internal demand diverged significantly from the start, failing to learn an effective policy. This was one of the reasons we later decided to discard it from further analysis.

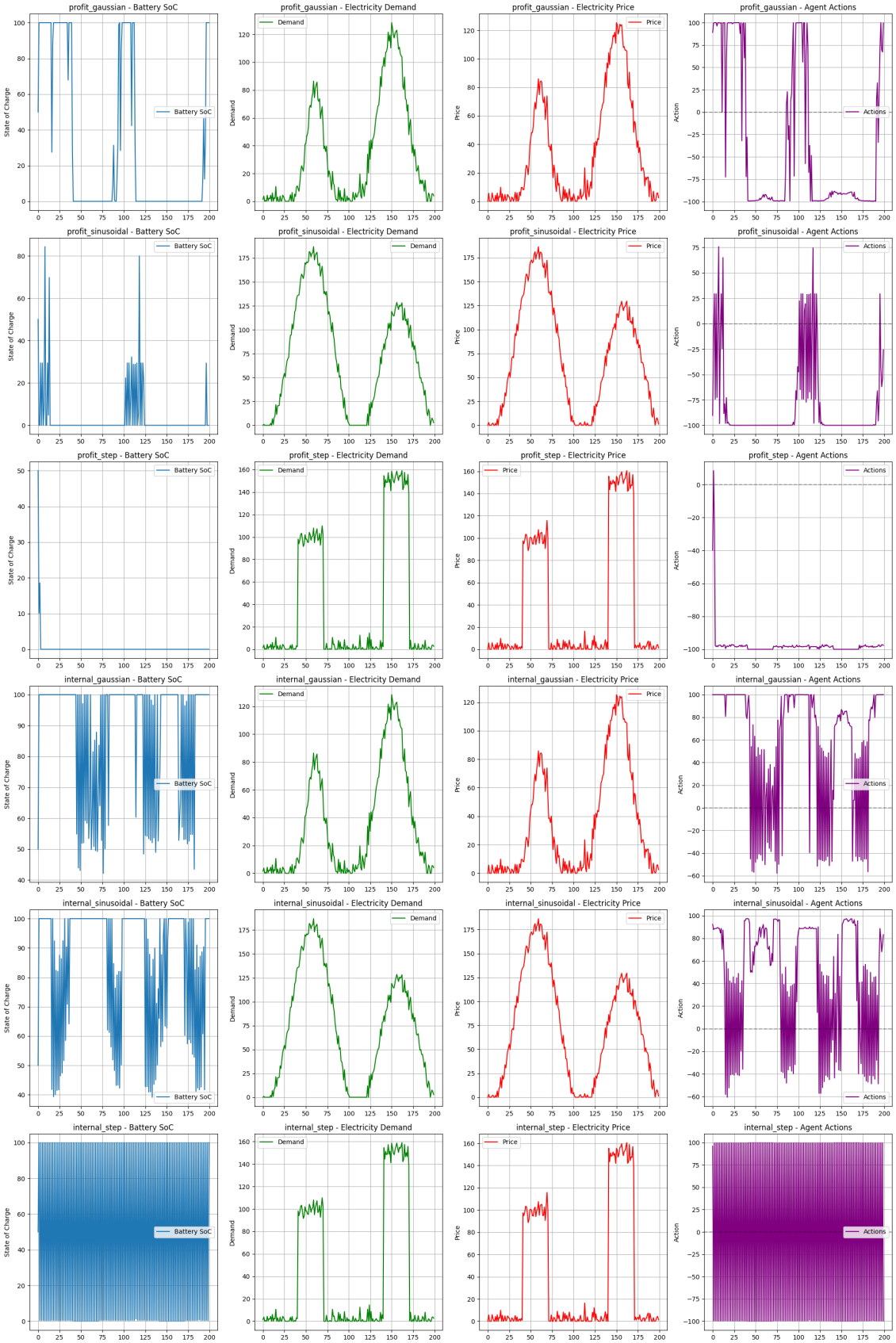

Figure 8: Action according to demand and price.