A Multi-Agent Deep Reinforcement Learning Framework for Home Energy Management Systems with Responsive Dynamic Pricing

1st Chenxi Qin
Department of Electrical and Electronic Engineering
The Hong Kong Polytechnic University
Hong Kong SAR
chenxi.qin@connect.polyu.hk

2nd Qin Wang
Department of Electrical and Electronic Engineering
The Hong Kong Polytechnic University
Hong Kong SAR
qin-ee.wang@polyu.edu.hk

3rd Zhaoyuan Wu
State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources
North China Electric Power University
Beijing, China
wuzy@ncepu.edu.cn

***Abstract*—In recent years, the advancement of smart grids and Internet of Things technologies has made the home energy management system (HEMS) a practical tool for improving residential energy efficiency. At the same time, electricity market liberalization is driving a transition of the power industry from fixed pricing to dynamic pricing. The use of dynamic prices based on individual consumption has become an important topic. This paper proposes a multi-agent deep reinforcement learning (DRL) framework for HEMS optimization, in which different types of loads are managed by specialized agents to better address their heterogeneous characteristics. Unlike conventional DRL-based HEMS approaches that treat electricity prices as fixed inputs, our framework embeds a responsive dynamic pricing environment into training. We develop a method to model how retailers adjust prices in response to customer demand. The HEMS interacts directly with the responsive environment during training, enabling it to learn strategies that perform well under diverse pricing conditions. The results of the case study show that our approach reduces customer costs more effectively than the DRL methods trained under fixed prices, demonstrating stronger generalizability across varying electricity price scenarios.**

***Index Terms*—Multi-agent deep reinforcement learning, home energy management, responsive dynamic prices, demand response.**

I. Introduction

With the advancement of smart grids and Internet of Things technologies, home energy management systems (HEMS) have become an important research direction, as they enhance the demand response potential of residential customers. HEMS can schedule smart appliances based on electricity prices and customer preferences, thereby improving residential energy efficiency and reducing electricity costs. As the price setters in the retail market, electricity retailers (ERs), who supply electricity to customers, can analyze smart meter data to understand customers' electricity consumption preferences [1]. Based on this analysis, they can design dynamic pricing strategies that adjust electricity rates according to customers' consumption behavior, thereby increasing profits. Therefore, HEMS should be capable of responding to varying electricity prices to effectively shield households from price volatility and strike an optimal balance between customer comfort and cost.

Currently, many studies have adopted various optimization methods to control HEMS. Ref. [2] employs binary particle swarm optimization to optimize the operation of smart appliances and electric vehicles (EVs) under RTP schemes. Ref. [3] uses the sparrow search optimization algorithm to control HEMS while ensuring grid stability. Ref. [4] utilizes a multi-objective differential evolution algorithm to optimize HEMS, considering three objectives: minimizing energy cost, peak-to-average ratio, and wasted photovoltaic energy. Ref. [5] employs a mixed-integer linear programming approach to model HEMS, considering the impact of battery cycle degradation in residential energy storage systems. However, these studies model HEMS as an optimization problem. When the electricity price changes, the HEMS must re-solve the optimization problem to obtain the corresponding optimal solution. When electricity prices fluctuate frequently, HEMS must repeatedly solve the optimization problem, resulting in an increased computational burden and longer computation times.

Deep reinforcement learning (DRL) enables agents to interact with the environment and evaluate outcomes using a reward function. This method is capable of solving complex nonlinear and non-convex problems, and supports an "online training, offline deployment" approach. Once training is completed, the DRL agent can be deployed offline. When faced with changing electricity price signals, the deployed agent can directly output decisions without the need for online optimization, resulting in a much faster response compared to traditional optimization algorithms. These features make DRL a popular approach for controlling HEMS. However, most current studies that use DRL to control HEMS do not adequately consider the price volatility that customers may face. Ref. [6], [7] both employ DRL approach to optimize HEMS. However, a common limitation of these studies is their reliance on static, predefined electricity pricing schemes, resulting in a fixed and passive pricing environment. This reduces the agents' ability to generalize and respond effectively to different electricity price scenarios.

Some studies have used changing electricity prices to train DRL agents, such as refs. [8]–[12]. However, the prices used in these studies are sourced from real-world datasets. Their variations are not affected by the agent's actions but are updated at each time slot based on the dataset. Thus the price changes are independent of the agent's actions. As a result, the agent cannot learn the relationship between its actions and changes in prices. In addition, other studies have adopted electricity price forecasting methods to predict possible future price scenarios in order to address potential price fluctuations. Study [13]

This study is supported by the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources (Grant No.LAPS25006).
979-8-3315-8138-1

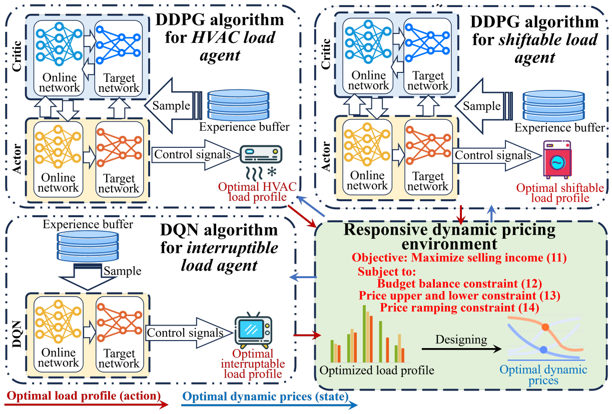

Fig. 1. The proposed multi-agent DRL framework

employs a neural network (NN)-based forecasting method to predict the electricity prices input to HEMS, and uses these predicted prices to train the DRL agent. Ref. [14] proposes a data-driven, DRL-based HEMS that uses a gate recurrent unit NN to forecast electricity prices. Agents then make scheduling decisions based on these predicted prices. Ref. [15] predicts the range of future RTP. This approach quantifies uncertainty and enhances the agent's generalization ability when dealing with RTP fluctuations. However, these approaches rely heavily on the accuracy of RTP forecasts, and the NN-based forecasting method inevitably introduces errors [16], [17]. Thus, the agent's optimization performance is affected when the actual electricity prices differ from the predicted values.

In summary, existing research on HEMS has not adequately accounted for the fact that dynamic pricing changes in response to agent's actions. This limitation restricts the practical applicability of HEMS in dynamic pricing environments. To address this issue, we propose a multi-agent DRL framework for HEMS with a responsive dynamic pricing environment, where price variations depend on the agents' actions. This allows the agent to learn the impact of its actions on electricity prices, thereby improving its robustness and generalization when facing different prices. Specifically, the contribution of this paper can be summarized as follows: **1)** We propose a multi-agent DRL framework for HEMS with a responsive dynamic pricing environment that enables agents to learn the impact of their actions on the changes of prices, thereby improving their generalization ability when facing different price scenarios. The responsive dynamic pricing environment is based on the Gurobi solver, which simulates the ER's pricing behavior. The environment adjusts prices according to changes in actions taken by the agents to maximize the ER's profit. **2)** We employ multi-agents to separately control the customer's shiftable load, heat, ventilation and air conditioning (HVAC) load, and interruptible load, allowing each agent to learn the distinct demand response characteristics of each load type. This approach better addresses their heterogeneous characteristics.

II. Multi-Agent DRL Framework with Responsive Dynamic Prices Environment

Fig. 1 illustrates the proposed multi-agent DRL framework, which consists of a shiftable load agent, an HVAC load agent, an interruptible load agent and the responsive dynamic pricing environment. The agents controlling the shiftable load and the HVAC load operate in continuous action spaces, as these loads can be adjusted continuously. Therefore, the deep deterministic policy gradient (DDPG) algorithm is employed for their training. In contrast, the control of the interruptible load involves discrete on/off decisions, for which the deep Q-network (DQN) algorithm is adopted. When the agents observe the current price curve provided by the dynamic pricing environment as the state, they train in parallel and then take actions to optimize the corresponding load profiles for maximizing rewards. The responsive dynamic pricing environment then updates the prices based on these load profiles aiming to maximize the ER's revenue. The agents subsequently re-optimize the load profiles according to the updated prices. This iterative process continues until both the optimal load profiles and optimal prices converge, achieving an equilibrium between the customer and the ER. The pseudocode of the training process is shown in Algorithm 1.

In this paper, each time slot is set to one hour. Therefore, the duration of each time slot Δt will be omitted in the following equations for simplicity.

A. State

The state set s consists of prices π_t for different time slots, which serve as observations for each agent, guide their training and act as input parameters. The state s can be represented as:

$$s = \{\pi_1, \pi_2, \cdots, \pi_t, \cdots, \pi_{24}\} \tag{1}$$

B. Action

1) Action Set for Shiftable Load Agent: For shiftable loads such as washing machines, dishwashers, and washer dryers, customers adjust their usage based on electricity prices to minimize their costs, the action space a_t^{SL} at time slot t is the power of shiftable load at time slot t, expressed as:

$$a_t^{SL} = p_t^{SL} \in [\theta_{min}^{SL} p_t^{SL,ori}, \theta_{max}^{SL} p_t^{SL,ori}] \tag{2}$$

where $p_t^{SL,ori}$ is the original power of shiftable load, θ^{min} and θ^{max} denote the minimum and maximum adjustable ranges, respectively.

2) Action Set for HVAC Load Agent: For HVAC loads, customers can adjust the output power in each time slot according to electricity prices. The action space for HVAC load agent a_t^{HVAC} at time slot t can be expressed as:

$$a_t^{HVAC} = p_t^{HVAC} \in [0, p_t^{HVAC,ori}] \tag{3}$$

where $p_t^{HVAC,ori}$ is the original power of HVAC load.

3) Action Set for Interruptible Load Agent: For interruptible loads, such as televisions and audio systems, customers adjust their on/off status according to electricity prices, so the action space for interruptible load agent a_t^{IL} is expressed as:

$$a_t^{IL} = \delta_t^{IL} \in \{0, 1\} \tag{4}$$

where 0/1 denotes the on/off status of interruptible load.

C. Reward

1) Reward for Shiftable Load Agent: The reward for the shiftable load agent has 3 components that can be expressed as follows.

$$R^{SL} = -\sum_{t=1}^{24}(R_t^{SL,cost} + R_t^{SL,step}) - R^{SL,total} \tag{5}$$

$$R^{SL,cost} = \pi_t \times p_t^{SL} \tag{6}$$

$$R_t^{SL,step} = c^{step} \times \max(0, \sum_{t=1}^{24} p_t^{SL,ori} - \sum_{i=1}^{t} p_i^{SL}) \tag{7}$$

$$R^{SL,total} = c^{total} \times |\sum_{t=1}^{24} p_t^{SL,ori} - \sum_{t=1}^{24} p_t^{SL}| \tag{8}$$

979-8-3315-8138-1

where (6) denotes the cost component, which calculates the electricity cost at time slot t. Since the shiftable load must satisfy the constraint that the total power before and after shifting remains equal as $\sum_{t=1}^{24} p_t^{SL} = \sum_{t=1}^{24} p_t^{SL,ori}$, we introduce both a stepwise penalty and a total power penalty in the reward function to guide the agent's optimization. These penalties guide the agent to adhere to the total energy constraint and avoid excessive deviation in the cumulative consumption profile during training. Where c^{step} and c^{total} are the coefficients for the stepwise and total power penalties, respectively. (7) is the stepwise penalty term, which ensures that the cumulative load at each time step remains close to the original cumulative load, helping the agent learn which specific steps may violate the power equality constraint. (8) is the total power penalty term, which prevents excessive deviation of the total load after shifting.

2) Reward for HVAC Load Agent: The customer's dissatisfaction with changes in HVAC load power can be represented by a quadratic function [18]. The reward for the HVAC load agent consists of the electricity cost term and dissatisfaction term:

$$R^{HVAC} = -\sum_{t=1}^{24} [\pi_t p_t^{HVAC} + \alpha^{HVAC} (p_t^{HVAC,ori} - p_t^{HVAC})^2 + \beta^{HVAC} (p_t^{HVAC,ori} - p_t^{HVAC})] \quad (9)$$

3) Reward for Interruptible Load Agent: For interruptible loads, the customer's dissatisfaction is proportional to the duration of interruption. Therefore, the reward for the interruptible load agent can be expressed as:

$$R^{IL} = -\sum_{t=1}^{24} (1 - \delta_t^{IL}) \pi_t p_t^{IL} - \gamma^{IL} \sum_{t=1}^{24} \delta_t^{IL} \quad (10)$$

where the first term represents the electricity cost, while the second term reflects the customer's discomfort caused by interruptions. p_t^{IL} denotes the load value of the interruptible load, and γ^{IL} represents the customer's discomfort associated with the interruptible load.

D. Environment of the multi-agent DRL

Once the three agents output their optimized actions, the responsive dynamic pricing environment adjusts electricity prices based on the optimized load profiles to maximize profit. The corresponding optimization problem is formulated as follows:

$$\max \sum_{t=1}^{24} \pi_t (p_t^{SL} + p_t^{HVAC} + p_t^{IL}) \quad (11)$$

$$\text{s.t.:} \quad \sum_{t=1}^{24} \pi_t = \Pi \quad (12)$$

$$\pi_t^{min} \le \pi_t \le \pi_t^{max} \quad (13)$$

$$|\pi_t - \pi_{t-1}| \le r_\pi^{max} \quad (14)$$

where dynamic prices π_t is the decision variable, Π is a predefined constant, π_t^{min} and π_t^{max} represent the lower and upper bounds of the electricity price, respectively, and r_π^{max} denotes the maximum allowable difference between adjacent prices. (12) is the budget balance constraint that ensures the sum of prices over all time periods is fixed, preventing the ER from setting excessively high prices. (13) imposes upper and lower constraints on the price at time slot t, while (14) is the ramping constraint that limits the price ramp between adjacent time periods to avoid excessive price fluctuations.

Algorithm 1 Multi-agent DRL for HEMS Optimization

Initialize DDPG agents for shiftable load and HVAC load, DQN agent for interruptible load
Initialize target networks and replay buffers for each agent
Initialize prices π_t for $t = 1, ..., 24$
for *episode* $e \leftarrow 1$ **to** E **do**
  for *time step* $t \leftarrow 0$ **to** 24 **do**
    Parallel Agent Actions:
    Each agent observes current state s_e and selects action:
    - Shiftable load: p_t^{SL} (DDPG)
    - HVAC load: p_t^{HVAC} (DDPG)
    - Interruptible load: δ_t^{IL} (DQN)
  end
  Execute actions and compute rewards R^{SL}, R^{HVAC}, R^{IL}
  Prices Update:
  Responsive dynamic pricing environment optimizes prices by solving: $\max \sum_{t=1}^{T} \pi_t (p_t^{SL} + p_t^{HVAC} + \delta_t^{IL} p_t^{IL})$
  s.t. (12)-(14)
  Observe new state s' with updated prices
  Store experiences and update agents:
  - Shiftable/HVAC agents: DDPG update
  - Interruptible agent: DQN update
end
Output: Equilibrium load profiles and prices

The optimized electricity prices are then observed by the agents as the new prices. Based on these updated prices, the agents further optimize the different types of loads. This process enables the agents to learn the impact of their actions on the environment, thereby improving the generalization ability of the multi-agent DRL-based HEMS.

E. DRL Algorithm

1) DDPG for Shiftable Load Agent and HVAC Load Agent: For the shiftable load agent trained using the DDPG algorithm, which consists of an actor network and a critic network, the actor network generates deterministic actions, while the critic network evaluates the Q-value. Target networks are employed to stabilize training through soft updates. An experience replay buffer is used to store transition tuples, which are randomly sampled during training to break temporal correlations and enhance learning stability. The actor network produces a single continuous output, bounded within $[\alpha^{min} p_t^{SL,ori}, \alpha^{max} p_t^{SL,ori}]$. This action represents the adjustment to the original load profile. The critic network, with identical hidden architecture, takes concatenated state-action pairs as input to estimate Q-values, guiding the actor toward cost-minimizing policies while maintaining $\sum_{t=1}^{24} p_t^{SL} = \sum_{t=1}^{24} p_t^{SL,ori}$ through threshold-based penalties.

The HVAC load agent adopts a similar DDPG architecture, with actions activated by a sigmoid function to represent the proportion (between 0 and 1) of the original HVAC load to maintain. This design ensures that the continuous output remains within the range $[0, p_t^{HVAC,ori}]$.

Both the shiftable load agent and the HVAC load agent utilize Ornstein-Uhlenbeck noise during exploration to achieve an effective balance between exploration and stability. Additionally, a soft update mechanism is employed to ensure gradual updates of the target network, while experience replay helps to reduce sample correlation. These strategies support stable convergence and effective policy learning in continuous control tasks.

TABLE I
PARAMETERS FOR DIFFERENT LOAD TYPES

Load Type	Parameters	Values
Shiftable load	c^{step}, c^{total},	0.01, 1.0
HVAC load	α^{HVAC}, β^{HVAC}	0.01, 0.021
Interruptible load	γ^{IL}	0.2

TABLE II
HYPERPARAMETER FOR DRL ALGORITHMS

Parameter	DDPG	DQN
Number of hidden layers	2*	2
Number of neurons per layer	128*	128
Learning rate	1×10^{-3}	1×10^{-3}
Discount factor	0.99	0.99
Soft update	0.0075	0.005
Mini batch	64	64
Replay buffer size	100000	100000

*:Both actor and critic networks use the same architecture.

2) DQN for Interruptible Load Agent: The DQN algorithm was selected for the interruptible load agent, as its discrete action space—representing binary decisions on/off—aligns naturally with DQN's discrete optimization capabilities. The DQN architecture employs two deep neural networks: a primary Q-network that learns action values and a target Q-network that provides stable learning targets, with periodic soft updates to mitigate training instability and oscillation. The training incorporates several DQN enhancements, including experience replay with a buffer to break temporal correlations, and an ϵ-greedy exploration strategy. This exploration schedule ensures comprehensive policy space investigation initially while progressively favoring exploitation of learned optimal strategies.

III. CASE STUDY

A. Case Study Setup

We selected the data of shiftable load, HVAC load, and interruptible load from publicly available datasets as the original input for the case study. The discomfort parameters for different load types are shown in Table I. They are chosen to more clearly demonstrate the performance of the proposed method. In practical applications, customers can set these parameters in the HEMS according to their own preferences. The adjustment limits θ^{SL}_{min} and θ^{SL}_{max} for the shiftable load are set to 0.8 and 1.2, respectively. The training parameters for DDPG and DQN are summarized in Table II. The sum of electricity prices Π is set to \$0.942. The upper and lower bounds of the electricity price are set to \$0.06/kWh and \$0.025/kWh, respectively, and the ramp limit is set to \$0.02. All simulations are performed using Python 3.9 on a PC equipped with a GTX4060 GPU, an AMD Ryzen 9 CPU @ 2.50 GHz, 16 GB RAM, and a Windows 11 operation system. The optimization of a responsive dynamic pricing environment is implemented using Gurobi 12.0.3.

B. Performance and Adaptability of the Multi-agent DRL Framework

As shown in Fig. 2, the training rewards of the shiftable, HVAC, and interruptible load agents consistently increase and converge within 100 episodes, demonstrating the learning efficacy and stability of the proposed multi-agent DRL framework. Fig. 3 illustrates the load profiles before and after adjustment by the multi-agent DRL approach under the final price. It can be observed that, after optimization, customer electricity consumption during high-price periods is reduced, with the total cost decreasing from \$19.91 before response to \$15.53 after response, achieving a 22% reduction. Fig. 4 shows the

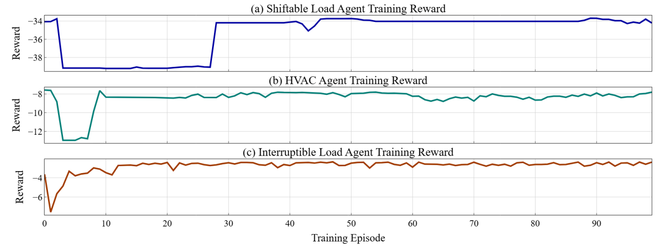

Fig. 2. Rewards convergence of multi-agent DRL

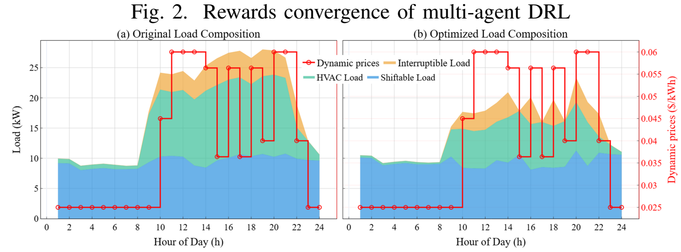

Fig. 3. Load profiles before and after optimization

adjustment of the three types of loads in each time slot. The shiftable load agent shifts loads from peak price periods (9:00–11:00 and 15:00–18:00) to lower price periods to reduce cost. The sum of shiftable load before response is 226.38 kW, while after response it is 224.62 kW, a difference of only 0.78%. This demonstrates the effectiveness of our step penalty and total power penalty mechanisms. The HVAC agent also decreases HVAC load power during high-price periods to balance cost and dissatisfaction. The interruptible agent responds at 14:00, 16:00, and 18:00—when both the price and interruptible load values are high—by interrupting the load accordingly.

Table III presents the costs for agents trained in the responsive dynamic pricing environment and those trained in the passive pricing environment under different price scenarios, with the same training hyperparameters. Price 1 refers to the original input price, price 2 is the final price obtained after iterative adjustment in the responsive dynamic pricing environment, and price 3 is a randomly generated price curve that satisfies the price constraints. In the passive pricing environment, agents are trained using price 1, which remains constant throughout the training process. Under price 1, both sets of agents achieve similar results and effectively reduce costs. However, when faced with price 2 and price 3, agents trained in the responsive dynamic pricing environment consistently outperform those trained in the passive pricing environment. This demonstrates that training agents in the responsive dynamic pricing environment enhances their generalization ability to different price scenarios, thereby increasing their practical value for real-world applications.

C. Analysis of the Responsive Dynamic Pricing Environment

Fig. 5(a) illustrates the evolution of prices in the responsive dynamic pricing environment during the training process. The price curves for the 1st, 30th, 50th, and 100th training episodes are shown. It can be observed that in the early stage of training (1st episode), the price curve remains relatively flat. As training progresses to the 30th episode, the system is able to identify peak load periods (from 10:00 to 22:00) based on the agents' actions, and increases the price during these periods. This reflects the environment's ability to recognize and respond to load patterns. With further training, at the 50th and 100th episodes, the price curve continues to adjust dynamically according to the agents' strategy, modifying both the timing

979-8-3315-8138-1

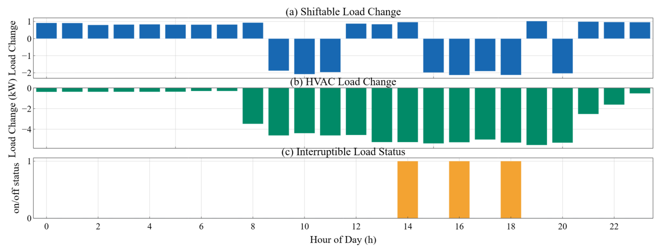

Fig. 4. Load adjustments per time slot

TABLE III
COMPARISON OF TRAINING ENVIRONMENTS ON COST PERFORMANCE

Load Type	Cost Scenario	Price 1	Price 2	Price 3
Shiftable	Original cost ($)	9.01	9.05	9.09
	Responsive environment ($)	**8.78**	**8.74**	**8.80**
	Passive environment ($)	8.78	8.98	8.96
HVAC	Original cost ($)	7.66	8.29	8.38
	Responsive environment ($)	4.43	**4.80**	**4.86**
	Passive environment ($)	**4.42**	6.45	6.67
Interruptible	Original cost ($)	2.29	2.57	2.56
	Responsive environment ($)	**1.51**	**1.99**	**1.86**
	Passive environment ($)	1.51	2.34	2.30

and magnitude of price peaks, and gradually converging to an equilibrium state. This process demonstrates that the responsive dynamic pricing environment possesses adaptive and learning capabilities, continuously optimizing its pricing strategy through ongoing interaction with agents. By emulating the real-world pricing mechanisms of ERs, this environment enables the agents to gradually understand the intrinsic relationship between their actions and price fluctuations, guiding them to adjust their actions in response to dynamic prices.

Fig. 5(b) and (c) illustrate the evolution of price volatility and ER's electricity sales profit throughout the training process. It can be observed that after approximately the 70th training episode, price volatility converges to a stable state near zero, indicating that the pricing strategy no longer undergoes significant adjustments. Meanwhile, ER's sales profit also converges to around $15.24. Combined with the results from Fig. 2 that the rewards of the three agents also stabilize during the same training period, it can be inferred that an equilibrium has been reached between agents and the responsive dynamic pricing environment. In summary, the simultaneous convergence of price, profit, and agents' rewards during training demonstrates the strong learning stability and policy consistency of the responsive dynamic pricing environment, eliminating the risk of cyclical oscillations in strategy.

IV. CONCLUSION

This paper presents a multi-agent DRL framework for HEMS under responsive dynamic pricing. Instead of treating prices as fixed, we model them as functions of household consumption to better reflect retail price formation for residential customers. Retailer pricing decisions are simulated using the Gurobi solver. The framework deploys three specialized agents to manage shiftable, HVAC, and interruptible loads, enabling tailored control strategies for each load type.

Case study results show that the proposed framework achieves coordinated convergence of price, load, and retailer profit during training, reaching an equilibrium state between the customer and the ER. Compared to agents trained in a fixed pricing environment, the proposed method demonstrates stronger adaptability and robustness when faced with different price curves, confirming its potential for application in real-world dynamic pricing

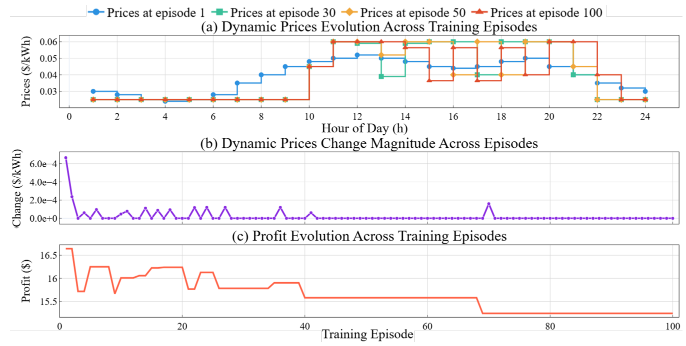

Fig. 5. Convergence behavior of the responsive dynamic pricing environment

environments. Beyond improving HEMS decision quality, the approach offers a practical pathway for coordinated optimization of customer-side resources within the electricity market.

REFERENCES

[1] C. Qin et al., "Designing personalized incentive-based demand response services based on smart meter data and NSGA-III-DE algorithm," *Energy*, p. 137454, 2025.

[2] K. Kanakadhurga et al., "Smart home energy management using demand response with uncertainty analysis of electric vehicle in the presence of renewable energy sources," *Appl. Energy*, vol. 364, p. 123062, 2024.

[3] L. Kong et al., "Smart home energy management optimization: An amended sparrow search algorithm for enhanced grid stability and cost efficiency," *Energy*, p. 136944, 2025.

[4] M. GhasemiGarpachi et al., "Multi-objective improved differential evolution algorithm-based smart home energy management system considering energy storage system, photovoltaic, and electric vehicle," *IEEE Access*, 2025.

[5] T. D. de Lima et al., "Optimizing home energy management systems: A mixed integer linear programming model considering battery cycle degradation," *Energy Build.*, vol. 329, p. 115251, 2025.

[6] Z. Wang et al., "Scalable energy management approach of residential hybrid energy system using multi-agent deep reinforcement learning," *Appl. Energy*, vol. 367, p. 123414, 2024.

[7] B. Wang et al., "Deep reinforcement learning method for demand response management of interruptible load," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3146-3155, 2020.

[8] Y. Liu et al., "Optimization strategy based on deep reinforcement learning for home energy management," *CSEE J. Power Energy Syst.*, vol. 6, no. 3, pp. 572-582, 2020.

[9] S. H. Mansour et al., "Deep reinforcement learning-based plug-in electric vehicle charging/discharging scheduling in a home energy management system," *Energy*, vol. 316, p. 134420, 2025.

[10] A. A. Amer et al., "DRL-HEMS: Deep reinforcement learning agent for demand response in home energy management systems considering customers and operators perspectives," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 239-250, 2022.

[11] G. Wei et al., "Deep Reinforcement Learning for Real-Time Energy Management in Smart Home," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2489-2499, 2023.

[12] L. Zhao et al., "Deep reinforcement learning-based joint load scheduling for household multi-energy system," *Appl. Energy*, vol. 324, p. 119346, 2022.

[13] H. Ding et al., "A safe reinforcement learning approach for multi-energy management of smart home," *Electr. Power Syst. Res.*, vol. 210, p. 108120, 2022.

[14] K. Ren et al., "A data-driven DRL-based home energy management system optimization framework considering uncertain household parameters," *Appl. Energy*, vol. 355, p. 122258, 2024.

[15] H. J. Kim et al., "Real-time price forecasting combined deep reinforcement learning for predictive home energy management system," *IEEE Internet Things J.*, 2025.

[16] B. Jiang et al., "A Hybrid KAN-ANN based Model for Interpretable and Enhanced Short-term Load Forecasting," *IEEE Trans. Artif. Intell.*, 2026.

[17] B. Jiang et al., "Multitask learning for enhanced short-term load forecasting in distribution networks under small sample conditions," *CSEE J. Power Energy Syst.*, 2025.

[18] X. Xu et al., "A multi-agent reinforcement learning-based data-driven method for home energy management," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3201-3211, 2020.

979-8-3315-8138-1