IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 October 2022, accepted 29 November 2022, date of publication 7 December 2022, date of current version 25 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3227450

METHODS

Multi-Agent Reinforcement Learning Based Actuator Control for EV HVAC Systems

SUNGHO JOO[1], DONGMIN LEE[1], MINSEOP KIM[1], TAEHO LEE[1], SANGHYEOK CHOI[1], SEUNGJU KIM[1], JEYEOL LEE[1], JOONGJAE KIM[2], YONGSUB LIM[1], AND JEONGHOON LEE[2]

[1]MakinaRocks, Seoul 06626, Republic of Korea
[2]Hanon Systems, Deajeon 34325, Republic of Korea

Corresponding author: Jeonghoon Lee (jlee@hanonsystems.com)

ABSTRACT While electric vehicles (EVs) continue to draw more attention as an alternative to traditional fossil fuel vehicles, the relatively short driving range of EVs is often pointed out as their biggest drawback. In terms of energy consumption, one of the most energy-intensive systems in EVs is the heating, ventilation, and air conditioning (HVAC) system. Most HVAC systems use On/Off or PID control for the actuators, but these control methods have low efficiency and are difficult to apply in multiple-input multiple-output systems. In this paper, we propose a novel multi-agent deep reinforcement learning (MADRL) method to efficiently control the low-level actuators of the EV HAVC systems. Through this method, multiple objectivs such as setpoint temperature, subcooling and efficiency can be considered simultaneously by giving independent rewards for each actuator agent. The proposed method is evaluated via a actual vehicle simulator, and experimental results show that the MADRL-based method consumes only 53% of the energy consumption of PID control on average in a transient phase.

INDEX TERMS Multi-agent reinforcement learning, energy consumption efficiency, HVAC, EV, RL.

NOMENCLATURE

Abbreviation

COP	Coefficient of Performance.
DOS	Degree of subcool.
EV	Electric Vehicle.
EXV	Electric eXpansion Valve.
HVAC	Heating, Ventilation, and Air Conditioning.
MADRL	Multi-Agent Deep Reinforcement Learning.
MARL	Multi-Agent Reinforcement Learning.
MG	Markov Game.
MIMO	Multiple-Input Multiple-Output.
PID	Proportional-Integral-Derivative.
RL	Reinforcement Learning.
SAC	Soft Actor-Critic.
SC	Subcooling.

Variables

β	Importance ratio in single agent reward function.
Δu_t^i	Change of control value of i^{th} actuator at time t.
δ	Weight of time penalty.
Γ^{target}	Target DOS.
Γ_t	DOS at time t.
λ	Range of convergence of temperature.
π_θ^i	policy of i^{th} actuator at time t.
ρ	Coefficient of reward function.
a_t^i	Action value of i^{th} actuator at time t.
C_t	Binary flag of the convergence of temperature.
L_Δ	Lower bound of change of control value.
L_i	Lower bound of operating range of i^{th} actuator.
M_1	Limit of convergence time in transient phase.
M_2	Limit of convergence time in steady phase.

The associate editor coordinating the review of this manuscript and approving it for publication was Christopher H. T. Lee.

N	Time of convergence.
r_t^{Γ}	Reward function of subcool at time t.
r_t^{Comp}	Reward function of compressor at time t.
r_t^{EXV}	Reward function of EXV at time t.
r_t^{Temp}	Reward function of temperature at time t.
r_t^{Work}	Reward function of work at time t.
s_t^i	State of i^{th} actuator at time t.
T^{target}	Target temperature.
T_t	Temperature at time t.
u_t^i	Control value of i^{th} actuator at time t.
U_{Δ}	Upper bound of change of control value.
U_i	Upper bound of operating range of i^{th} actuator.

I. INTRODUCTION

One of the biggest drawbacks of current electric vehicles (EVs) is their relatively short driving range compared to traditional fossil fuel vehicles. The simplest way to improve the short driving range is to increase battery capacity, and a significant amount of related research is underway [1]. Another approach is to improve the efficiency of the energy-consuming systems in EVs [2], [3], [4], [5]. One of the most energy-intensive systems in EVs is the heating, ventilation, and air conditioning (HVAC) system, which is a complex nonlinear thermo-fluid dynamics system composed of a variety of components such as a compressor, heat exchangers, and electric expansion valves (EXVs) to control vehicle climate conditions. The importance of the efficient control of the HVAC system in EVs is greater than that of conventional vehicles as the proportion of energy consumption by the HVAC system is larger in EVs [3].

Most HVAC systems use On/Off control and PID control [6], [7], [8]. On/Off control is simple but has problems such as low efficiency and a shortening of the lifespan of the components. PID control, also widely used for its simple implementation, does not require the dynamics of the target system and therefore can be applied to both linear and nonlinear systems. However, PID control involves a number of issues. First, in most cases, it is difficult to use PID control in multiple-input multiple-output (MIMO) systems as each control output is coupled with a single feature, which makes it hard to reflect complex objectives with multiple features in PID control. Second, PID control requires a setpoint that is typically based on human expertise and therefore might not be optimal. In particular, determination of the target degree of subcool (DOS), which greatly influences the efficiency of the system [9], [10], depends on the prior experience of humans. Further details about subcooling are explained in Section II. Third, since PID control is a type of feedback control that relies on errors from the current observation of the system, it is difficult to consider the entire trajectory. As a result, if the coefficients are not tuned well, oscillation can occur in the system.

Recently, approaches based on reinforcement learning (RL) to HVAC control have been studied. RL methods are designed to maximize the reward of the global trajectory [11] and are widely applied to various control problems. In HVAC systems, the policies of RL control a set of actuators, satisfying the multi-objective of the MIMO system via reward engineering [12], [13]. In Refs. [12], [13], the classical RL method SARSA has been applied to vehicle HVAC systems, showing promising performance compared to conventional control methods. However, the classical tabular RL method is inadequate for a large continuous system because of the curse of dimensionality [11]. In building HVAC systems, RL-based control has shown promising efficiency and generalization performance compared to baseline algorithms [14], [15], [16], [17], [18], [19], [20]. However, Refs. [14] and [18] can be used only for the discrete control problem, and while Refs. [15], [17], [18] propose a high-level control that outputs the desired setpoint temperature, the efficiency of the low-level subsystem actuator control to reach the setpoint temperature is not considered. Therefore, there is room for improving control at the low-level. Otherwise, the control methods proposed in [19] and [20] apply to the HVAC systems of buildings and are not applicable to vehicle HVAC systems.

In this paper, to address the above mentioned challenges, we propose a novel control method based on multi-agent deep reinforcement learning (MADRL) for the EV HVAC system, with the following features. First, our method is based on a multi-agent architecture, which enables the use of actuator-specific reward functions to minimize the energy consumption of the MIMO system. Second, using the novel reward functions, our method finds the setpoints needed for feedback control without human expertise. Third, our method achieves more energy-efficient control than conventional methods in terms of the entire trajectory while achieving comparable target convergence performance. To the best of our knowledge, this research is the first attempt to use deep reinforcement learning for the continuous control of low-level subsystem actuators in the EV HVAC system.

The remaining sections are as follows. In Section II, we introduce our problem's objective functions, the concept of subcooling, and deep reinforcement learning. In Section III, we explain our MADRL-based method including state representation, action representation, and reward functions. In Section IV, we show that our proposed method outperforms the grid search and conventional PID control in efficiency while meeting a target temperature. As a training environment, we employed the commercial simulation software GT-SUITE® [21], which is widely used for industrial vehicle HVAC modeling. Conclusion and further works are provided in the last section.

II. PROBLEM STATEMENTS AND PRELIMINARIES

A. PROBLEM STATEMENTS

The HVAC system provides a cool or warm airflow into the cabin through a variety of heat exchanges for passengers' thermal comfort during driving. The heat exchangers

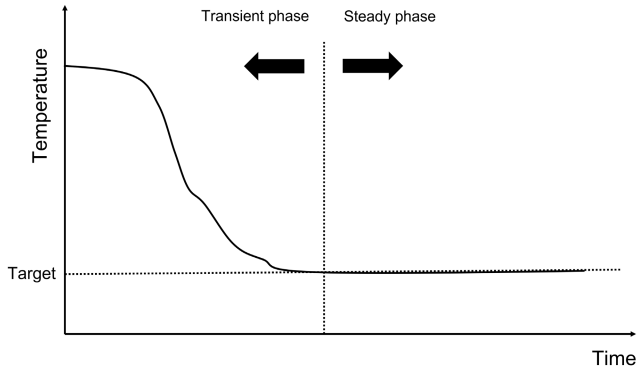


FIGURE 1. Phases of control in the HVAC system.

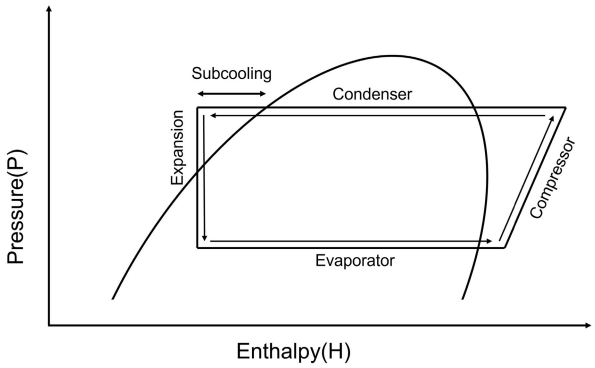


FIGURE 2. Pressure-enthalpy diagram for the vapor-compression cycle.

(e.g., evaporator, heater core, low temperature radiator, etc.) transport thermal energy that is converted into the change in cabin air temperature.

The HVAC system has multiple modes depending on the usage for high performance and efficiency. Each mode adopts a different circuit design by closing or opening multi-way valves. Because of such distinct circuit design, each mode involves a different set of actuators, and therefore, every mode has its own specific objective and constraints.

1) OBJECTIVE

In this paper, we consider the cabin air conditioning (A/C) mode, which is used when the cabin demands low-temperature air. The actuators involved in this mode are the compressor and one EXV. The objective of the mode is to meet the given setpoint of the air temperature from the evaporator while minimizing the work. Regarding the work, only that done by the compressor is considered, since other work such as cooling fan work or blower work are relatively small.

To be more specific, the objective can be divided into two parts: before and after reaching the target temperature, as shown in Fig. 1. Before reaching the target temperature (we call this stage the transient phase), the system operates to find the optimal trajectory of the control inputs, which minimizes the weighted sum of the total work done and time taken until the convergence of the temperature. The system should reach the target temperature within a given time:

$$\begin{aligned} &\min \sum_{t=1}^{N} W_t + \delta N \\ &\text{s.t. } |T_{N-1} - T^{target}| \le \lambda, \\ &\quad |T_N - T^{target}| \le \lambda, \\ &\quad N \le M_1 \end{aligned} \tag{1}$$

where δ determines the weight of the time penalty and λ is the range of convergence of the temperature.

After reaching the target temperature (we call this stage the steady phase), the system operates to maintain the temperature within the desired range while conducting minimum work. The system becomes steady by fixing the inputs of the actuators to certain values. While there can be multiple input combinations that maintain the same target temperature, each of them may differ in efficiency. Therefore, in the steady phase, the objective is to find the optimal combination of control inputs rather than finding the trajectory. This optimal combination should satisfy Eq. (2):

$$\begin{aligned} &\min \sum_{t=N+1}^{M_2} W_t \\ &\text{s.t. } |T_t - T^{target}| \le \lambda, \\ &\quad \forall t \in \{N+1, \ldots, M_2\} \end{aligned} \tag{2}$$

2) USE OF SUBCOOLING

A liquid is subcooled when it exists at a temperature below its normal condensing point. Fig. 2 shows that subcooling (SC) exists to the upper left of the saturation line in the pressure-enthalpy diagram. In HVAC systems, condenser SC has a significant effect on the coefficient of performance (COP) [22]. In other words, maintaining the DOS at the target level is equivalent to achieving a certain level of efficiency. In conventional control methods, an EXV is controlled to reach the desired DOS using PID control or model predictive control to improve the COP [9], [10], [23], [24]. Our method is first validated to control the EXV to meet the target DOS provided by experts as an auxiliary target of efficiency. In the steady phase, the system is expected to reach not only the target temperature but also the optimal DOS. The objective function of the steady phase can be modified as follows:

$$\begin{aligned} &\min \sum_{t=N+1}^{M_2} W_t \\ &\text{s.t. } |T_t - T^{target}| \le \lambda, \\ &\quad |\Gamma_t - \Gamma^{target}| \le \lambda, \\ &\quad \forall t \in \{N+1, \ldots, M_2\} \end{aligned} \tag{3}$$

However, as stated in [23], finding the optimal DOS is difficult and often requires numerous assumptions. Moreover, if the system is newly introduced or updated, finding the optimal DOS requires trial and error. In such cases when

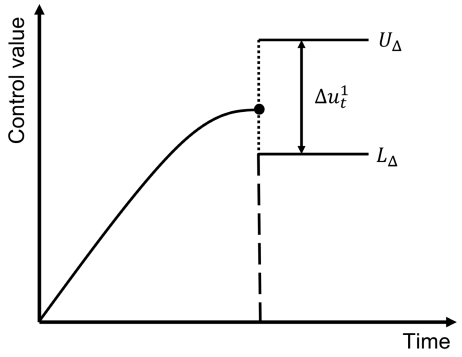


FIGURE 3. Slew rate constraint.

the target DOS is unknown, our method can be trained to directly minimize the work. Details are further discussed in Section IV.

3) CONSTRAINTS

Due to physical limits and safety issues, the compressor has a limit on the available changes in actuation per time, which is called the slew rate constraint. The slew rate constraint can be expressed as Eq. 4, where u_t^1 and u_t^2 are the control values of the compressor and EXV at time step t, respectively:

$$L_\Delta \leq \Delta u_t^1 \leq U_\Delta, \quad \forall t \tag{4}$$

Also, each actuator's operating range has physical limits. This constraint is shown in Eq. 5.

$$\begin{aligned} L_1 \leq u_t^1 \leq U_1, \quad \forall t \\ L_2 \leq u_t^2 \leq U_2, \quad \forall t \end{aligned} \tag{5}$$

B. PRELIMINARIES

1) REINFORCEMENT LEARNING

We formulate the HVAC control problem as a Markov decision process [11] with a tuple of states (S, A, p, r, γ), where S is the continuous state space, A is the continuous action space, $p : S \times A \times S \to \mathbb{R}_{\geq 0}$ is the unknown state transition dynamics that denotes the probability density of the next state $s' \in S$ given $s \in S$ and $a \in A$, and $r : S \times A \to \mathbb{R}$ is a reward function. $\gamma \in [0, 1)$ is a discount factor of future rewards, and $\pi(a|s)$ is a stochastic policy of action a given state $s \in S$. The agent chooses action a_t based on policy $\pi(a_t|s_t)$ for every time step given state s_t and reaches the next state s_{t+1} following the stochastic transition dynamics $p(s_{t+1}|s_t, a_t)$.

The goal of RL is to maximize the expected cumulative reward from the current policy π for every time step:

$$\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \tag{6}$$

Using the policy function $\pi(a|s)$, the state value function V or action value function Q (or Q-function) is obtained to approximate the expected cumulative reward. The state value function $V^\pi(s)$ of a policy π is the expected cumulative reward starting from the state s upon executing π:

$$V^\pi(s) \triangleq \mathbb{E}_\pi \left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r(s_\tau, a_\tau) \mid s_t = s \right] \tag{7}$$

Next, the Q-function $Q^\pi(s, a)$ of a policy π is the expected return starting from state s, taking action a, and then following π:

$$Q^\pi(s, a) \triangleq \mathbb{E}_\pi \left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r(s_\tau, a_\tau) \mid s_t = s, a_t = a \right] \tag{8}$$

2) SOFT ACTOR-CRITIC

Recently, deep neural networks have been widely used to train the policy or value function of RL algorithms [25], [26], [27], [28]. In this work, we deployed soft actor-critic (SAC) [29], which is a state-of-the-art model-free RL algorithm for continuous control domains. SAC has a high sample efficiency, as the algorithm is trained based on maximum entropy [30] with an actor-critic architecture [31]. Moreover, SAC tends to converge more stably and requires less hyperparameter tuning than other RL algorithms.

In SAC, the goal of the agent is to maximize not only the expected sum of the rewards from the current policy π but also the expected entropy of the policy:

$$\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \Big(r(s_t, a_t) + \alpha H\big(\pi(\cdot \mid s_t)\big) \Big) \right] \tag{9}$$

where $H\big(\pi(\cdot \mid s_t)\big) = \mathbb{E}_{a\sim\pi}[-\log\big(\pi(a|s)\big)]$. The critic, which is the soft Q-function Q_θ of a policy π_ϕ, is trained by minimizing the critic objective (Eq. 10), while the actor π_ϕ is updated by minimizing the actor objective (Eq. 11),

$$\begin{aligned} J_Q(\theta) \triangleq \mathbb{E}_{(s_t,a_t)\sim D} \Bigg[\frac{1}{2} \Bigg(Q_\theta(s_t, a_t) \\ - \Big[r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V_{\hat{\theta}}(s_{t+1}) \Big] \Bigg)^2 \Bigg] \end{aligned} \tag{10}$$

$$J_\pi(\phi) \triangleq \mathbb{E}_{s_t\sim D} \Big[\mathbb{E}_{a_t\sim\pi_\phi} \big[\alpha \log \pi_\phi(a_t \mid s_t) - Q_\theta(s_t, a_t) \big] \Big] \tag{11}$$

where $\hat{\theta}$ is the set of parameters of the target network. For a better exploration, the temperature coefficient α is automatically adjusted for the maximum entropy policy by minimizing the α objective,

$$J(\alpha) \triangleq \mathbb{E}_{a_t\sim\pi_t} \big[-\alpha \log \pi_t(a_t \mid s_t) - \alpha H_0 \big] \tag{12}$$

where H_0 is the desired minimum expected entropy threshold.

3) MULTI-AGENT RL

Multi-agent reinforcement learning (MARL) is widely studied in control systems with multiple components [32], [33], [34], [35], [36], [37]. MARL enables a more delicate design of the action and reward functions compared to single-agent

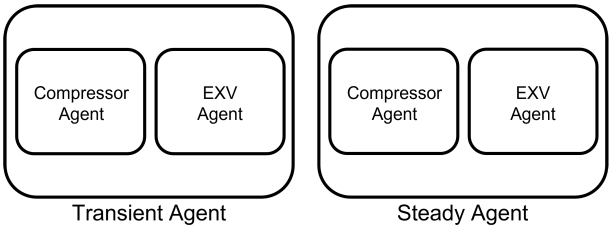


FIGURE 4. Agent diagram.

RL algorithms. To define our MARL problem, we introduce a Markov game (MG) [32], [34], [38], [39]. The MG is a framework that generalizes the Markov decision process for multiple agents interacting simultaneously in a shared environment. MG is defined with the tuple $(N, S, \{A^i\}, p, \{r^i\})$, where N denotes the number of interacting agents ($N > 1$), S is the continuous state space, $A : A^1 \times \cdots \times A^N$ is the joint action space which is the collection of the continuous action spaces of individual agents $i \in \{1, \ldots, N\}$, $p : S \times A \times S \to \mathbb{R}_+$ is the unknown state transition probability, and $r : r^1 \times \cdots \times r^N$ is the reward function. In MARL, joint action $a : a^1 \times \cdots \times a^N$ and joint policy $\pi(a|s) = \Pi_i \pi^i(a^i|s^i)$ are defined with the collection of the actions and policies of individual agents. A MARL agent chooses joint action a_t based on joint policy $\pi(a_t|s_t)$ for every time step given state s_t and reaches the next state s_{t+1} following stochastic transition dynamics $p(s_{t+1}|s_t, a_t)$. The goal of MARL is same as the goal of RL (Eq. 6).

III. METHODS

A. MULTI-AGENT IN THE HVAC SYSTEM

As mentioned in Section II-A, our model has two different objectives, Eq. 1 for the transient phase and Eq. 2 for the steady phase. Each objective is optimized by its own agent, namely a transient agent for the transient phase and a steady agent for the steady phase. Each agent is composed of a compressor agent and an EXV agent, which are responsible for the corresponding actuators of the system. With an identical architecture for the compressor and EXV agents (Fig. 4), the transient and steady agents are trained for their respective phase using their own novel reward functions. Each reward function reflects the objectives defined in Eq. 1 and Eq. 3.

The compressor agent contains a policy network π^1_θ whose primary objective is to find the optimal control value of the compressor to meet the target temperature. Similarly, the EXV agent contains a policy network π^2_θ whose primary objective is to find the optimal control value of the EXV to meet the target DOS. In general, the compressor has a larger influence on the system than the EXV. Before the compressor converges, most of the features including compressor work, temperature, and SC predominantly depend on the compressor. Only after the target temperature is reached and the compressor converges can the EXV agent observe the change in SC that is induced solely by itself. Then it becomes possible to control the EXV agent to meet the target DOS with appropriate feedback. In other words, EXV control requires long-term exploration, in which the agent learns to meet the target DOS after the convergence of the target temperature.

In the case of single-agent RL, the agent outputs control values of both compressor and EXV from a single neural network. Also, the reward is a single scalar value unified from r^1_t and r^2_t (Eq. 13).

$$r^{single}_t = (1-\beta)r^1_t + \beta r^2_t, \quad \beta \in [0, 1] \tag{13}$$

In this case, the agent tends to learn to control the more dominant component, which is the compressor, while struggling with the less dominant component, the EXV. One possible solution to address this issue is to adjust each reward's relative importance with a ratio constant, β. Unfortunately, finding the right value of β requires extensive hyperparameter searching with exponentially growing costs for increasing numbers of actuators. By splitting each actuator into independent agents, the relative magnitude of the reward is no longer an issue. Likewise, this structure can be easily expanded to HVAC systems with more actuators by simply adding more agents.

B. STATE REPRESENTATION

The compressor and EXV agents share most of their important features from system observations. Common features are as follows.

$$[T_{t-1}, T_t, \Gamma_{t-1}, \Gamma_t, T_t - T^{target}, \Gamma_t - \Gamma^{target}, u^1_{t-1}, u^2_{t-1}] \tag{14}$$

Here, $T_{t-1}, T_t, \Gamma_{t-1}, \Gamma_t$ are observed values of the temperature and SC of the current and previous time steps, respectively. As the HVAC system reacts gradually rather than instantly, information of the previous time steps is required for the agents to decide the next action. This nature of the HVAC system can be easily inferred from a mathematical modeling of each system component, where most of the dynamics are differential equations with time [40], [41]. In the above equation, the current error of temperature and SC are denoted by $T_t - T^{target}$ and $\Gamma_t - \Gamma^{target}$. The agents can decide the magnitude and direction of the action based on the current error. Moreover, the control values of the previous step are also included as (u^1_{t-1}, u^2_{t-1}). To make the most of the information available, each agent's current state contains the other agent's action that was taken one step before. The control values of different actuators are normalized with their upper and lower limits for a relative scale, as follows.

$$\begin{aligned} u^1_{t-1} &\leftarrow \frac{u^1_{t-1} - L_1}{U_1 - L_1} \\ u^2_{t-1} &\leftarrow \frac{u^2_{t-1} - L_2}{U_2 - L_2} \end{aligned} \tag{15}$$

Besides the common features, the EXV agent receives additional information, C_t, which is a binary flag indicating whether the target temperature is reached. C_t, as given by Eq. 16, is 1 when the error of the temperature is below λ

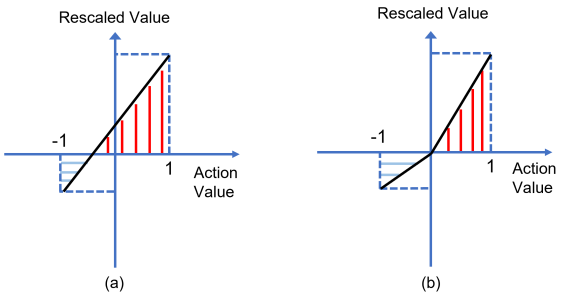


FIGURE 5. Compressor action mapping function. (a) Naive approach and (b) proposed approach.

for two consecutive action steps and otherwise 0. The reward changes depending on C_t, and thus C_t provides a clear understanding of the status. More details about the reward function are explained in Section IV-D.

$$C_t = \begin{cases} 1, & \text{if } |T_t - T^{target}| < \lambda \\ & \text{and } |T_{t-1} - T^{target}| < \lambda \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

To sum up, the state of each agent is as follows:

$$\begin{aligned} s_t^1 &= [T_{t-1}, T_t, \Gamma_{t-1}, \Gamma_t, T_t - T^{target}, \\ &\quad \Gamma_t - \Gamma^{target}, u_{t-1}^1, u_{t-1}^2] \\ s_t^2 &= [T_{t-1}, T_t, \Gamma_{t-1}, \Gamma_t, T_t - T^{target}, \\ &\quad \Gamma_t - \Gamma^{target}, u_{t-1}^1, u_{t-1}^2, C_t] \end{aligned} \tag{17}$$

C. ACTION REPRESENTATION

The action values (a_t^1, a_t^2) of the policy networks (Eq. 18), which are in the $[-1, 1]$ range, are converted to control values via mapping functions. For clarity, we call the output of the agent as the action value and the input of the actuator as the control value.

$$\begin{aligned} a_t^1 &= \pi_\theta^1(s_t^1) \in [-1, 1] \\ a_t^2 &= \pi_\theta^2(s_t^2) \in [-1, 1] \end{aligned} \tag{18}$$

As each actuator needs a different control value, different mapping functions are applied. For the compressor agent, the action value a_t^1 is mapped to the control value u_t^1 considering two types of constraints. The first constraint (Eq. 5) is the operating range of the component, which is the maximum and minimum rpm value of the compressor. The other constraint (Eq. 4) is the slew rate, which is the allowed change of rpm per second for the safety of the component. The compressor mapping function is as follows:

$$\begin{aligned} u_{range} &= \begin{cases} \min(U_1, u_{t-1}^1 + U_\Delta) - u_{t-1}^1, & \text{if } a_t^1 > 0 \\ u_{t-1}^1 - \max(L_1, u_{t-1}^1 + L_\Delta), & \text{if } a_t^1 < 0 \end{cases} \\ u_t^1 &= u_{t-1}^1 + a_t^1 \cdot u_{range} \end{aligned} \tag{19}$$

The compressor action value a_t^1 is first rescaled to the domain of an actuator control value using the slew rate and the operating range. The rescaled value is then added to the previous control value u_{t-1}^1 to get the next control value u_t^1. In this way, we can assure that the control value always stays in the allowed range.

One notable thing to address is that the mapping function is asymmetric. When a_t^1 is rescaled with an affine transformation to the change in the control value, the allowed range of increment and decrement can differ [Fig. 5(a)]. If a_t^1 is rescaled uniformly in this range, the portion of either increment or decrement can be relatively larger than the other, which can be problematic for the training process. Specifically, the agent collects training data with a random policy in the early stages of the training. The distribution of the training data can be biased towards either increment or decrement if the range is unbalanced, and this can greatly influence the training time and quality. To address this issue, the action value is rescaled differently depending on its sign. Positive action values always increase the control value within the allowed range, while negative action values work in the opposite way [Fig. 5(b)].

While a_t^1 is rescaled to Δu_t^1, a_t^2 is directly rescaled to u_t^2 using the operating range of the EXV component. The slew rate is not considered in this case. The EXV mapping function is as follows.

$$\begin{aligned} u_t^2 &= 0.5\, a_t^2 \cdot (U_2 - L_2) \\ &\quad + 0.5(U_2 + L_2) \end{aligned} \tag{20}$$

D. REWARD FUNCTION

The reward function is designed to enable multi-objective control in the current MIMO system. As described in Fig. 4, two different control agents are trained, one for each corresponding phase. The objectives of each phase are expressed as reward components.

1) REWARD FUNCTION FOR THE TRANSIENT AGENT

The reward function for the transient agent in Eq. 21 is based on Eq. 1 from Section II. The reward function has two components, r_t^{Temp} and r_t^{Work}. The compressor agent and EXV agent receive rewards composed of a temperature reward and work reward. In the transient phase, the convergence of SC is not considered; hence, the only termination condition for the transient phase is the convergence of temperature. The reward function of the compressor and EXV agents is r_t^1 and r_t^2, respectively. Note that the ρ variables determine the shape and scale of the reward function.

$$\begin{aligned} r_t^{Temp} &= -(\rho_1 + \rho_3) + \rho_1 e^{-\rho_2 \cdot (T_t - T^{target})^2} \\ &\quad + \rho_3 e^{-\rho_4 \cdot (T_t - T^{target})^2} \\ r_t^{Work} &= -(\rho_1 + \rho_3) + \frac{\rho_1 + \rho_3}{500 - 4000}\left(W_t - 4000\right) \\ r_t^1 &= \begin{cases} r_t^{Temp} + r_t^{Work} - \rho_5, & \text{if } t = M_1 \text{ and } C_t = 0 \\ r_t^{Temp} + r_t^{Work}, & \text{otherwise} \end{cases} \\ r_t^2 &= r_t^{Work} \end{aligned} \tag{21}$$

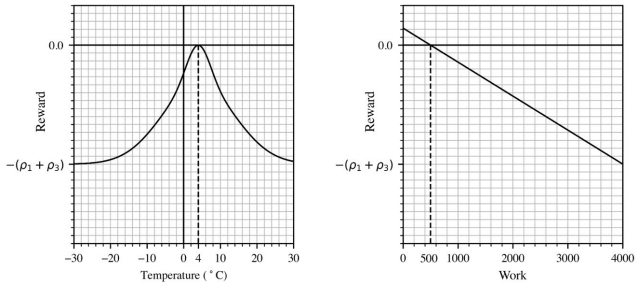


FIGURE 6. Reward function of the transient agent with a target temperature of 4 °C, showing (left) the temperature reward and (right) the work reward.

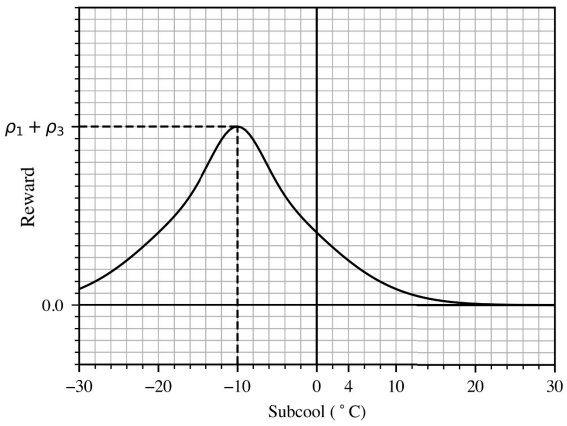


FIGURE 7. Subcool reward function for steady phase with a target degree of subcool −10 °C.

The r_t^{Temp} component is related to the convergence of temperature to the target value. As T_t approaches T^{target}, r_t^{Temp} converges to 0. In contrast, r_t^{Temp} converges to $-(\rho_1+\rho_3)$ as T_t gets further away from T^{target}. Otherwise, the r_t^{Work} component is related to the work done by the compressor and linearly decreases as the work W_t increases. W_t is linearly mapped to $[-(\rho_1+\rho_3), 0]$ using its minimum and maximum values, which are 500 W and 4000 W. The reason for mapping temperature and work rewards to the same range is to prevent any reward from being overly dominant. In a multi-objective problem, an imbalance between reward components may cause some objectives to be ignored in training (Fig. 6).

Each agent's reward function is designed to satisfy the objectives of the transient phase (Eq. 1). In the compressor agent case, the reward r_t^1 consists of a sum of r_t^{Temp} and r_t^{Work} in order to reach the target temperature while minimizing the work. To reflect the time constraint, the agent receives a large negative penalty ρ_5 if the model fails to reach the target temperature by the end of the episode. On the other hand, r_t^2 is the same as r_t^{Work}. The EXV only takes the work reward because SC is not considered in the transient phase and the EXV control value has only a subtle influence on the change in temperature. It is worth noting that the objective of the transient phase is to meet the target temperature, not the target SC.

One important thing to address in the transient phase is the sign of the rewards. Contrary to the steady phase, every reward in the transient phase has a negative value. With a negative reward, the total episode reward decreases as the length of the episode increases. Therefore, negative rewards force the agent to balance between terminating the episode early and minimizing the work, which paves the way to achieve our objective function in Eq. 1. This property of negative rewards is used to reflect the objective of faster convergence.

2) REWARD FUNCTION FOR THE STEADY AGENT

The reward function for the steady agent in Eq. 22 is based on Eq. 3 from Section II. This reward function is designed to be two-fold due to the trajectory of the temperature. As the steady phase comes after the transient phase (Fig. 1), the steady agent should be trained not only in the steady phase but also in the transient phase, and thus should have two reward functions. When the temperature is far from its target ($C_t = 0$, transient phase), the reward function is designed to reach the target temperature as fast as possible to enter the steady phase. Then after reaching the target temperature ($C_t = 1$, steady phase), the reward function changes to meet the objectives of the steady phase (Eq. 3). Compared to the transient agent, an additional reward component r_t^Γ is used for the convergence of the target SC in this case. Also, the sign of both reward components for the steady agent is positive, for the following reasons. As the objective of the steady phase is to maintain the target temperature and SC in a stable manner, the steady agent should be enhanced to stay in the desired state as long as possible. With a positive reward, we can compensate the agent as much as necessary. Note that composing the two reward functions of this agent with a single sign is important; if negative and positive rewards are mingled, the agent will struggle to understand the true effect of the action. Therefore, the positive reward for the steady agent's transient phase is applied, differing from the negative reward for the transient agent.

$$
\begin{aligned}
r_t^{Temp} &= \rho_1 e^{-\rho_2 \cdot (T_t - T^{target})^2} \\
&\quad + \rho_3 e^{-\rho_4 \cdot (T_t - T^{target})^2} \\
r_t^{Work} &= \frac{\rho_1 + \rho_3}{500 - 4000}\left(W_t - 4000\right) \\
r_t^{\Gamma} &= \rho_1 e^{-\rho_2 \cdot (\Gamma_t - \Gamma^{target})^2} \\
&\quad + \rho_3 e^{-\rho_4 \cdot (\Gamma_t - \Gamma^{target})^2} \\
r_t^1 &= \begin{cases} r_t^{Temp}, & \text{if } C_t = 0 \\ r_t^{Temp} + r_t^{Work}, & \text{if } C_t = 1 \end{cases} \\
r_t^2 &= \begin{cases} r_t^{Work}, & \text{if } C_t = 0 \\ r_t^{\Gamma}, & \text{if } C_t = 1 \end{cases}
\end{aligned} \tag{22}
$$

The shapes of the temperature and work reward for the steady agent are the same as the model for the transient agent. They are only translated to positive values. Here, r_t^{Temp} converges to $\rho_1+\rho_3$ as T_t approaches T^{target} and converges to 0 otherwise. And r_t^{Work} is linearly mapped to $[0, \rho_1+\rho_3]$ using

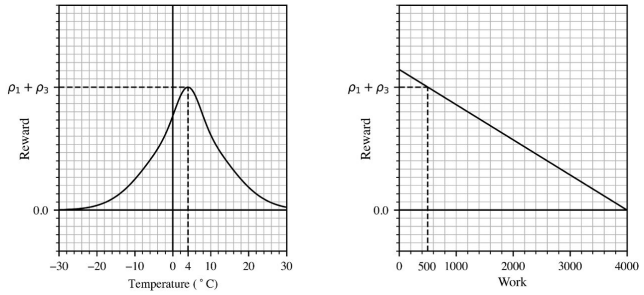


FIGURE 8. Reward function of the steady agent with a target temperature of 4 °C, showing (left) the temperature reward and (right) the work reward.

the minimum and maximum values of W_t. Similar to r_t^{Temp}, r_t^{Γ} converges to $\rho_1 + \rho_3$ as Γ_t reaches Γ^{target} and converges to 0 otherwise (Fig. 7). Before reaching the target temperature ($C_t = 0$), r_t^1 is the same as r_t^{Temp}. After reaching the target temperature ($C_t = 1$), the compressor agent receives both r_t^{Temp} and r_t^{Work}. The reason for adding r_t^{Work} when $C_t = 1$ is to train the steady agent to keep the temperature in the desired range ($|\Delta T| < \lambda$). As r_t^{Work} is positive, r_t^1 is always much higher when the temperature is in the desired range, from which we can be assured that the convergence of the temperature has a higher priority than efficiency.

In this case, r_t^2 is the same as r_t^{Work} before reaching the target temperature. After reaching target temperature, the EXV agent receives r_t^{Γ}, with which it is trained to reach the target SC. When the target SC is unknown, the EXV agent can be trained with r_t^{Work} after reaching the target temperature ($C_t = 1$). Here, the EXV agent is trained to find the action that minimizes the work while maintaining the temperature. As a byproduct, the resultant SC can be used as a target SC value. An agent trained without target SC is validated in Section IV-2.

E. MODEL ARCHITECTURE AND TRAINING PROCESS

Fig. 9 illustrates our MADRL training framework for the HVAC system control. First, both compressor and EXV agents receive the tuple (s_t, a_t, s_{t+1}) by interacting with the HVAC simulator. Using the tuple, the reward functions of the compressor and EXV agents calculate r_t^1 and r_2^t, respectively, and store the transition (s_t, a_t, r_t, s_{t+1}) in their experience replay memory. Then, a mini-batch of k transitions are randomly sampled from each experience replay memory and given to each network. The networks output the current action value, current Q-value, and target Q-value for each agent. These outputs are passed to multiple loss functions that calculate actor loss, critic loss, and α loss. The parameters of each network are updated using the gradient of the corresponding loss. After the training, in the inference, the actor network is separated and takes the state as an input and outputs control values for the compressor and EXV.

Both the compressor and EXV agents are based on SAC [29]. The network architecture of our model is as follows. Both agents have identical structures except for the size of their inputs. As stated in Section III-B, the EXV agent receives an additional state C_t. Similar to the original SAC paper, our model uses twin Q-networks with a target smoothing coefficient τ of 0.01. Both the actor and critic of the agents are fully connected networks with 2 hidden layers with 50 neurons and use ReLU as the activation function. The learning rate η is 0.0003 for the critic, actor, and entropy. For the stability of the training, the gradient is clipped at 5. Also, the Adam optimizer [42] is used to update the parameters. More details about the model hyperparameters are given in Table 1.

The pseudocode of the training process is described in Algorithm 1. The MADRL training algorithm first starts by initializing the neural networks. In line 2, the parameters of

TABLE 1. Model hyperparameter table.

Hyperparameter	Value
Batch size	64
η	0.0003
τ	0.01
λ	0.5
H_0	0

Algorithm 1 Training Algorithm

1: **for** each component i **do**
2: Initialize critic network parameters θ_1^i, θ_2^i and actor network parameter ϕ^i
3: Initialize target critic network parameters $\bar{\theta}_1^i \leftarrow \theta_1^i, \bar{\theta}_2^i \leftarrow \theta_2^i$
4: Initialize alpha α^i
5: Initialize experience replay memory M^i
6: **end for**
7: **for** each iteration **do**
8: Obtain the state s_0^1, s_0^2
9: **for** each environment step **do**
10: Obtain control actions $a_t^1 \sim \pi_\phi^1(a_t^1 \mid s_t^1), a_t^2 \sim \pi_\phi^2(a_t^2 \mid s_t^2)$
11: Calculate rewards $r_t^1(s_t^1, a_t^1), r_t^2(s_t^2, a_t^2)$
12: Obtain new states $s_{t+1}^1 \sim p(s_{t+1}^1 \mid s_t^1, a_t^1), s_{t+1}^2 \sim p(s_{t+1}^2 \mid s_t^2, a_t^2)$
13: **for** each component i **do**
14: $M^i \leftarrow M^i \cup (s_t^i, a_t^i, r_t^i, s_{t+1}^i)$
15: **if** $M_{size} \geq B_{size}$ **then**
16: Randomly sample k transitions from experience replay memory M^i
17: Update $\theta_j^i \leftarrow \min_{\theta^i} J_Q(\theta_j^i)$ for $j \in 1, 2$
18: Update $\phi^i \leftarrow \min_{\phi^i} J_\pi(\phi^i)$
19: Update $\alpha^i \leftarrow \min_{\alpha^i} J(\alpha^i)$
20: Update $\bar{\theta}_j^i \leftarrow \tau\theta_j^i + (1-\tau)\bar{\theta}_j^i$ for $j \in 1, 2$
21: **end if**
22: **end for**
23: **end for**
24: **end for**

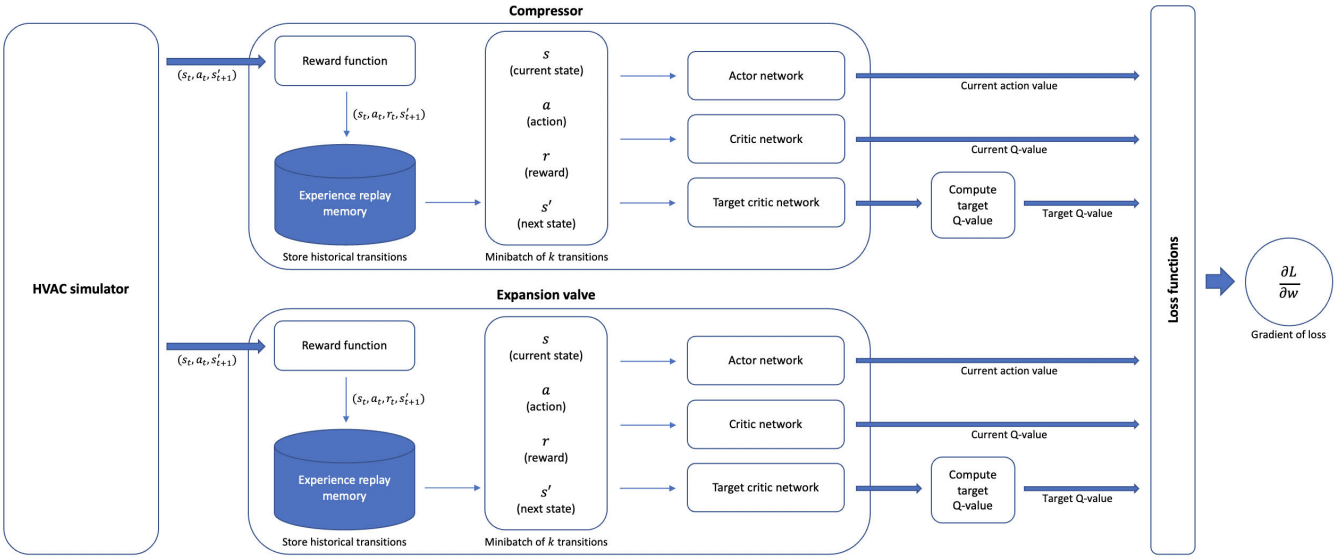


FIGURE 9. MADRL training framework.

the actor and critic networks use Xavier initialization. Then the parameters of the target critic network are synchronized with the critic network. Also, the coefficient of entropy α and experience replay memory M are initialized.

The training process consists of multiple nested loops, where the outer loop denotes the training episode for each iteration, the second loop denotes the training step of each episode, and the innermost loop denotes the training of each component. In line 8, the initial states s_0^1, s_0^2 are obtained from the HVAC simulator. In lines 10–12, using the current state s_t, the action, reward, and next state are obtained, which forms the transition sample (s_t, a_t, r_t, s_{t+1}) for every time step. In line 14, the transition sample is stored in the replay memory, and in lines 15–19, a mini-batch is randomly sampled from the replay memory and the networks are updated. In line 20, the target critic network is soft-updated with hyperparameter τ.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

In this section, the MADRL-based control algorithm is validated for the EV HVAC system. The HVAC system is formulated with the commercial software GT-SUITE® [21], which has been widely employed by vehicle manufacturers and component suppliers. As one of the key aims of this research is to facilitate our implementation in an industrial setup, it is natural to choose a widely used industrial simulator. In our experiment, the vehicle thermal system circuit is implemented as explained in Section II. For experiments, the thermal system circuit requires additional conditions such as vehicle speed and ambient temperature; these conditions are chosen here based on a common scenario using the A/C circuit. In particular, the target evaporator outlet temperature, which is the main control objective, is set within a range of 4−12°C with an interval of 2 degrees. The time step unit is 3 seconds, and one episode has 200 time steps. For each setpoint temperature experiment, RL agents are trained 500 episodes each. Details of the conditions are summarized in Table 2.

TABLE 2. Experimental conditions.

Condition	Value
Vehicle Speed	50 km/h
Ambient Temperature	30 °C
Target Temperature	4∼12 °C
Action Interval	3 sec/step
Time steps per episode	200 step

B. EXPERIMENTAL RESULTS

As explained in Section III, the transient and steady agents of the MADRL-based control model are trained separately. In this section, we evaluate our model in terms of the temperature convergence and the work efficiency. Also, we show that our method works well without a target DOS and can even find an effective target DOS. Additionally, we compare the MADRL-based control model with conventional PID control.

1) CONVERGENCE AND EFFICIENCY EVALUATION

The performance of temperature convergence is evaluated differently for each phase. In the transient phase, the ability to reach the target temperature within a limited time is tested (Fig. 10). The time limit (t_{limit}) is 40 steps, equivalent to 120 s. For every target temperature, the transient agent succeeds to reach the target within the given time. In the steady phase, the ability to reach both target temperature and SC is evaluated. Fig. 11 shows the case when the target temperature is 6 °C; the rest of the experimental results are presented in Fig. 12.

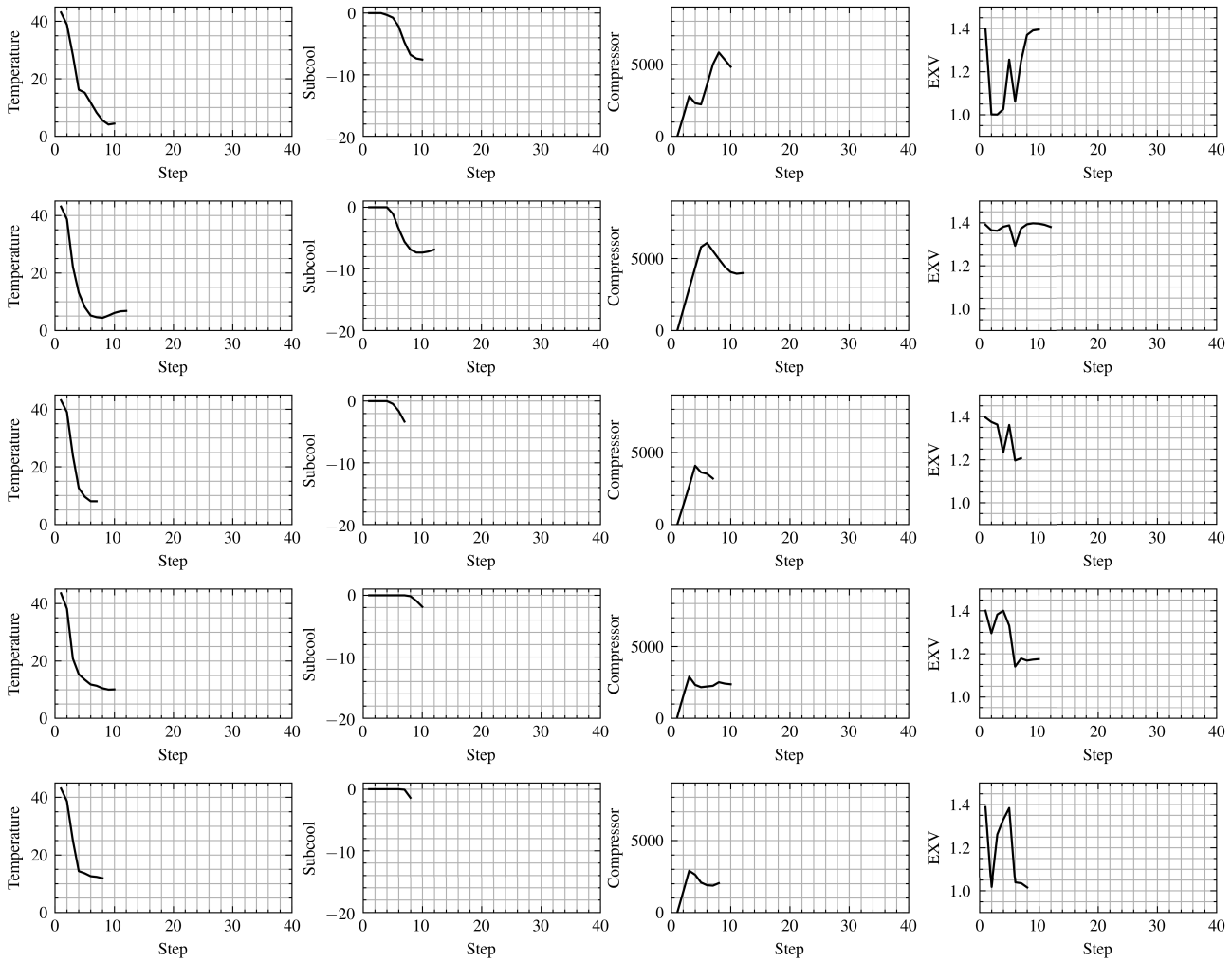


FIGURE 10. Evaluation of transient agent in transient phase.

TABLE 3. Ratio of work done between different methods in the transient phase (until $C_t = 1$).

Temperature	Steady agent	Transient agent	PID
4	0.26	0.11	1.00
6	0.61	0.35	1.00
8	0.52	0.25	1.00
10	1.50	0.96	1.00
12	0.81	0.96	1.00

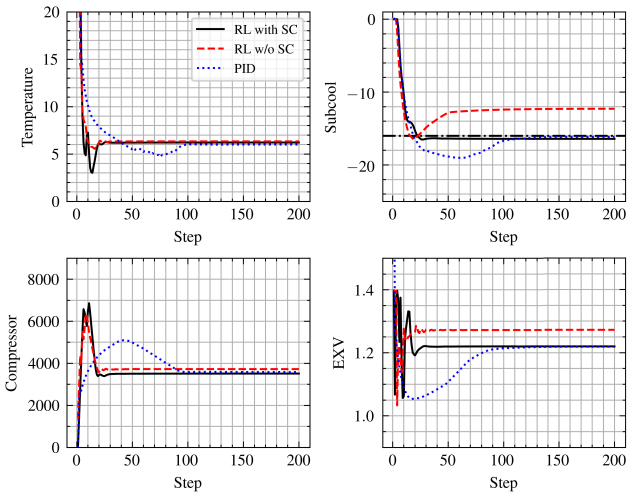


FIGURE 11. Comparison of MADRL steady agent and PID control trajectory for 6°C.

The target DOS corresponding to the target temperatures is obtained by domain experts. In Fig. 11, we can see that the trajectory of the MADRL control model with a SC target converges to the target temperature and the target SC. As a result, the control values reach almost the same control values as PID control by the end. For every experiment in Fig. 12, the agents succeed to reach the target temperature and target SC with less than 0.5 °C error.

In terms of efficiency, we evaluate the work done by each control model in each phase. In Table 3, the work done by the MADRL control model is compared with PID control for the transient phase. Comparing the result of the transient agent and PID control, the MADRL control model significantly improves the efficiency. The transient agent generally shows better performance in the lower temperature zone where the

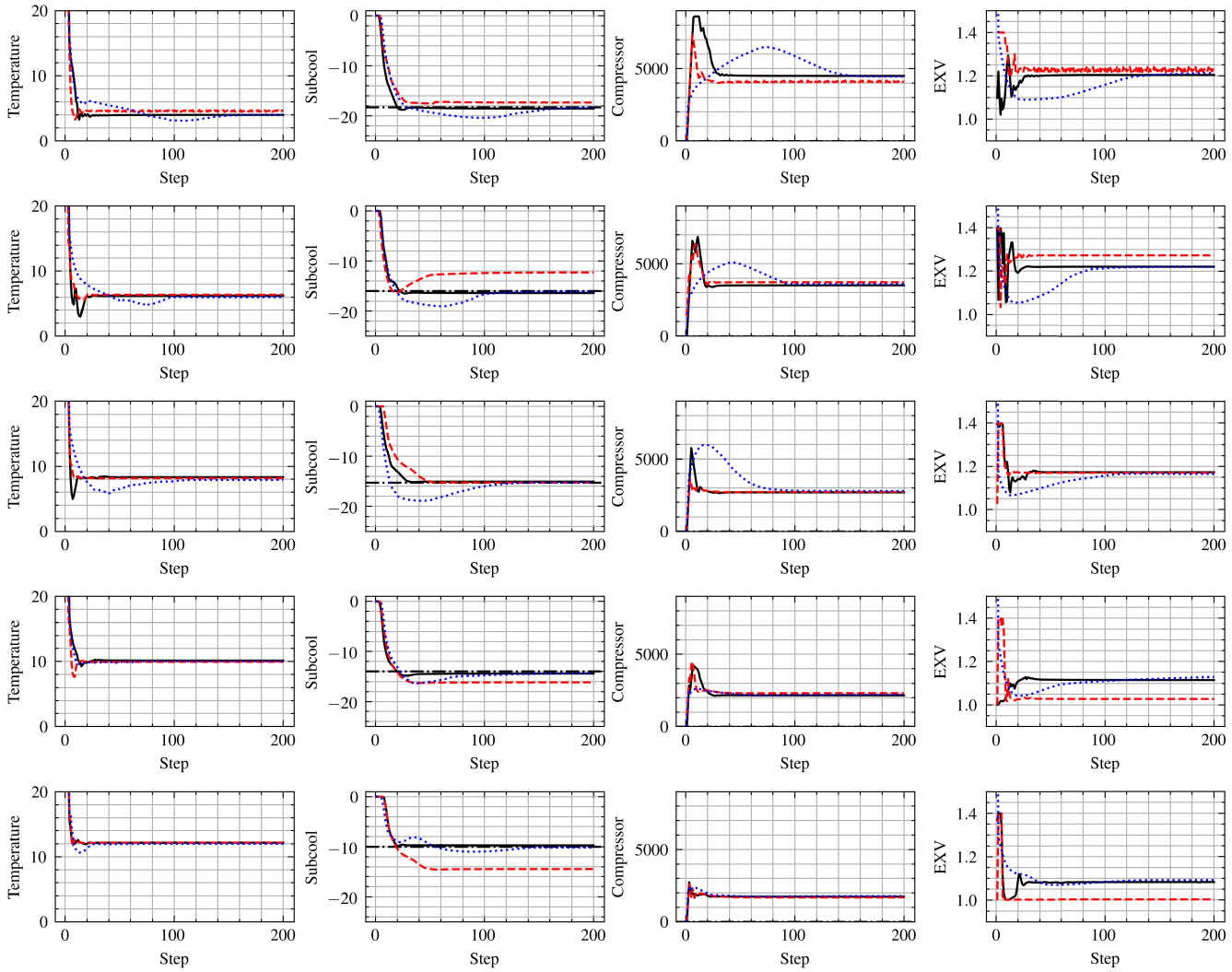


FIGURE 12. Evaluation of steady agent with subcool target, without subcool target and PID control in steady phase: red dash line is RL control without subcool, blue dot line is PID control, solid black line is RL control with subcool, and black dash dot line is target subcool.

TABLE 4. Time steps of each method in the transient phase (until $C_t = 1$). Final refers to the final settling time.

Temperature	Steady agent	Transient agent	PID	PID (Final)
4	15	10	67	132
6	19	12	38	92
8	12	7	18	81
10	13	10	11	11
12	8	8	9	23

TABLE 5. Comparison of work in the steady phase.

Temperature	RL w/ SC	RL w/o SC	Grid Search	PID
4	1.01	0.92	1.37	1.00
6	0.97	1.02	2.05	1.00
8	0.96	0.97	1.07	1.00
10	0.98	1.04	1.14	1.00
12	0.98	1.02	1.05	1.00

cooling load is greater than in the high temperature zone. In particular, when the target temperature is 4°C, the transient agent consumes only 11% of the energy compared to that by PID control.

Such performance improvement is largely due to faster convergence. As shown in Fig. 10, Fig. 12, and Table 4, PID control converges to the target temperature more slowly than the transient agent. One of the reasons for this slow convergence is an overshoot that is easily observable in the PID control. The overshoot of control is observable not only in the temperature but also in the SC. The overshoot of SC also contributes to the inefficiency of the system, where the proper SC promises a better COP. Conversely, the trajectories of the transient agent show little or no overshoot.

When we compare the work done between the steady agent and the transient agent in Table 3, the transient agent shows better efficiency, indicating that the reward functions are working as intended. As the transient agent is penalized if the work done by the compressor is high, the transient agent

TABLE 6. Grid search result.

Temperature		Compressor	EXV	Work	Subcool
Target	Final				
4	4.8	4000	1.2	1680.4	-17.5
	3.3	5000	1.2	2105.9	-18.8
	4.0	6000	1.4	2567.7	-11.5
6	5.4	5000	1.4	2081.4	-7.9
	6.3	7000	1.6	3092.4	0
	5.7	8000	1.6	3431.3	0
8	7.5	3000	1.2	1246.3	-14.7
	7.8	4000	1.4	1680.4	0
	8.7	5000	1	1940.8	0
	8.4	5000	1.6	2285.8	0
	9.0	6000	1.8	2303.4	0
	8.5	6000	1	2363.4	-18.6
	7.2	6000	1.6	2598.4	0
	8.3	6000	1.8	3043.0	0
	8.5	6000	2	3075.9	0
	8.2	7000	1	2668.2	-19.0
	7.8	7000	1.8	3136.1	0
	7.9	7000	2	3128.4	0
	8.0	8000	1	2954.3	-19.7
	7.4	8000	1.8	3519.4	0
	7.6	8000	2	3526.6	0
10	9.8	3000	1	1181.5	-16.8
	10.6	3000	1.4	1239.2	0
	9.2	4000	1	1562.8	-17.5
	9.7	4000	1.6	1705.5	0
	10.1	4000	1.8	1711.3	0
	10.3	4000	2	1714.8	0
	9.2	5000	2	2304.9	0
12	11.3	2000	1	1034.3	-15.8
	12.4	2000	1.2	967.8	0
	11.4	3000	1.6	1261.9	0
	11.8	3000	1.8	1274.3	0
	12.0	3000	2	1287.6	0

utilizes the compressor more mildly to use less work. Lastly, in the steady phase, the steady agent trained with the SC target is compared with PID control in Table 5. Specifically, the work values are estimated based on the average of the last 10 steps of the trajectory in the steady phase. The steady agent shows a similar performance compared to PID control. This result is consistent with the trajectory of both methods, as they converge to the same temperature and DOS.

2) TRAINING THE MADRL MODEL WITHOUT A SUBCOOLING TARGET

As mentioned in Section III, conventional PID control requires a target DOS for feedback control, whereas the MADRL-based control model can be trained without a target DOS. Without a SC target, the agent is expected to reach the target temperature while minimizing the work. Since the target DOS is only used in the steady phase, this experiment is only for the steady agent.

We first validate if the model successfully reaches the target temperature. In Fig. 12, for every case, the agent reaches the target temperature successfully. In Table 5, we compare the work done by the MADRL control model trained without a SC target and that by PID control. Our model shows comparable efficiency for all cases, averaging 99% efficiency compared to PID control. Especially when the target temperature is 4°C, the MADRL control model outperforms by 8%. Also, it is worth noting that the model converges to SC similar to PID control in most cases. Even when the converged DOS differ, the control model shows reasonable efficiency, varying by less than 5% from the PID control results even in the worst case. The DOS targets used in this work are optimal values found by domain experts, and finding them requires wide experience and numerous heuristics. If the DOS targets are not optimal, the performance gap between PID control and our method will be even greater.

For further validation, we conducted a brief grid search (Table 6). The grid search is executed by maintaining a fixed set of actions until observations converge. The resolutions of the actions are 1000 rpm and 0.2 mm for the compressor and EXV, respectively. Each grid search result is grouped by the final temperature. In the results, each group contains a final temperature value differing less than 1°C from the target temperatures of our experiment. In Table 5, the best grid search results are selected that satisfy the convergence condition ($|\Delta T| \leq 0.5$). Compared to the grid search results, both the MADRL control model and PID control achieve greater efficiency for all target temperatures. In fact, the grid search results show that finding the appropriate DOS for each temperature is difficult. For target temperatures of 4 and 6°C, less than 3 combinations of actions are found to reach the desired temperature. Applying a finer resolution would result in better performance but also increase the cost of computation. Moreover, if the number of actuators increases, the cost increases exponentially.

V. CONCLUSION

In this paper, we proposed a MADRL-based control method for an EV HVAC system. The key conclusions of this work are as follows: First, through the multi-agent architecture, various objectives such as temperature, subcooling and efficiency can be simultaneously set. Second, the proposed method reduce the energy consumption in the transient phase to 53% of the PID control. Third, experiment show that it is possible to control with a similar level of efficiency without optimal setpoint subcooling only through the reward function. Furthermore, the converged SC values can be used as new setpoints.

Our study was validated in the A/C mode under a simulated environment. In future research, we will validate our model in the heat pump mode with more actuators. Also, we expect to generalize our model in a real EV with minimum tuning.

ACKNOWLEDGMENT

(Sungho Joo, Dongmin Lee, and Minseop Kim contributed equally to this work.)

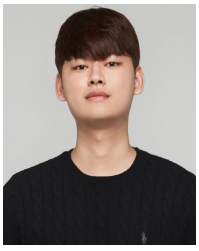
REFERENCES

[1] W. Chang, M. Lukasiewycz, S. Steinhorst, and S. Chakraborty, "Dimensioning and configuration of EES systems for electric vehicles with boundary-conditioned adaptive scalarization," in *Proc. Int. Conf. Hardw./Softw. Codesign Syst. Synth.*, Sep. 2013, pp. 1–10.

[2] J. Pouladi, M. B. B. Sharifian, and S. Soleymani, "Determining charging load of PHEVs considering HVAC system and analyzing its probabilistic impacts on residential distribution network," *Electr. Power Syst. Res.*, vol. 141, pp. 300–312, Dec. 2016.
[3] K. Vatanparvar and M. A. Al Faruque, "Battery lifetime-aware automotive climate control for electric vehicles," in *Proc. 52nd Annu. Design Autom. Conf.*, Jun. 2015, pp. 1–6.
[4] T. Zhang, C. Gao, Q. Gao, G. Wang, M. Liu, Y. Guo, C. Xiao, and Y. Y. Yan, "Status and development of electric vehicle integrated thermal management from BTM to HVAC," *Appl. Thermal Eng.*, vol. 88, pp. 398–409, Sep. 2015.
[5] E. Paffumi, M. Otura, M. Centurelli, R. Casellas, A. Brenner, and A. Jahn, "Energy consumption, driving range and cabin temperature performances at different ambient conditions in support to the design of a user-centric efficient electric vehicle: The quiet project," in *Proc. 14th SDEWES Conf.*, Dubrovnik, Croatia, 2019, pp. 1–6.
[6] H. Nasution and M. N. W. Hassan, "Potential electricity savings by variable speed control of compressor for air conditioning systems," *Clean Technol. Environ. Policy*, vol. 8, no. 2, pp. 105–111, May 2006.
[7] J. D. Zhang, G. H. Qin, B. Xu, H. S. Hu, and Z. X. Chen, "Study on automotive air conditioner control system based on incremental-PID," *Adv. Mater. Res.*, vols. 129–131, pp. 17–22, Aug. 2010.
[8] B. C. Ng, I. Z. M. Darus, H. M. Kamar, M. N. M. Lazin, and M. Hussein, "Dynamic modeling of an automotive air conditioning system and an auto tuned PID controller using extremum seeking algorithm," in *Proc. IEEE Symp. Comput. Informat. (ISCI)*, Apr. 2013, pp. 92–97.
[9] G. Pottker and P. S. Hrnjak, "Effect of condenser subcooling of the performance of vapor compression systems: Experimental and numerical investigation," Tech. Rep., 2012.
[10] L. Xu and P. S. Hrnjak, "Potential of controlling subcooling in residential A/C system," Tech. Rep., 2014.
[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[12] M. J. Kasbi, B. Sallans, and G. Russ, "A new approach in controlling the compressor of the vehicle air conditioning system," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2006, pp. 484–491.
[13] J. Brusey, D. Hintea, E. Gaura, and N. Beloe, "Reinforcement learning-based thermal comfort control for vehicle cabins," *Mechatronics*, vol. 50, pp. 413–421, Apr. 2018.
[14] T. Wei, Y. Wang, and Q. Zhu, "Deep reinforcement learning for building HVAC control," in *Proc. 54th Annu. Design Autom. Conf.*, Jun. 2017, pp. 1–6.
[15] Y. Wang, V. Kirubakaran, and H. Biao, "A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems," *Processes*, vol. 5, no. 3, p. 46, Sep. 2017.
[16] N. K. Dhar, N. K. Verma, and L. Behera, "Adaptive critic-based event-triggered control for HVAC system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 178–188, Jan. 2018.
[17] Y. Chen, L. K. Norford, H. W. Samuelson, and A. Malkawi, "Optimal control of HVAC and window systems for natural ventilation through reinforcement learning," *Energy Buildings*, vol. 169, pp. 195–205, Jun. 2018.
[18] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, and K. PohLam, "A deep reinforcement learning approach to using whole building energy model for HVAC optimal control," in *Proc. Building Perform. Anal. Conf.*, vol. 3, 2018, pp. 22–23.
[19] G. Gao, J. Li, and Y. Wen, "DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8472–8484, Sep. 2020.
[20] L. Yu, Y. Sun, Z. Xu, C. Shen, D. Yue, T. Jiang, and X. Guan, "Multi-agent deep reinforcement learning for HVAC control in commercial buildings," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 407–419, Jan. 2021.
[21] *GT-SUITE®*, Gamma Technologies, Westmont, IL, USA, 2020.
[22] J. Glos, F. Solc, and P. Vaclavek, "Model-based electronic expansion valve feed-forward control for electrified automotive vapor compression refrigeration system," in *Proc. 46th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2020, pp. 2050–2056.
[23] E. Hervas-Blasco, M. Pitarch, E. Navarro-Peris, and J. M. Corberán, "Study of different subcooling control strategies in order to enhance the performance of a heat pump," *Int. J. Refrig.*, vol. 88, pp. 324–336, Apr. 2018.
[24] M. Pitarch, E. Hervas-Blasco, E. Navarro-Peris, J. Gonzálvez-Maciá, and J. M. Corberán, "Evaluation of optimal subcooling in subcritical heat pump systems," *Int. J. Refrig.*, vol. 78, pp. 18–31, Jun. 2017.
[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[26] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
[27] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[29] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.
[30] B. D. Ziebart, *Modeling Purposeful Adaptive Behavior With the Principle of Maximum Causal Entropy*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 2010.
[31] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.
[32] P. Hernandez-Leal, B. Kartal, and E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Auto. Agents Multi-Agent Syst.*, vol. 33, pp. 750–797, Oct. 2019.
[33] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.
[34] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artif. Intell. Rev.*, vol. 55, pp. 895–943, Feb. 2022.
[35] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, "Emergent tool use from multi-agent autocurricula," 2019, *arXiv:1909.07528*.
[36] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, and C. Hesse, "Dota 2 with large scale deep reinforcement learning," 2019, *arXiv:1912.06680*.
[37] O. Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
[38] L. Busoniu, R. Babuska, and S. B. De, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Feb. 2008.
[39] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1994, pp. 157–163.
[40] Q. Zhang, S. Stockar, and M. Canova, "Energy-optimal control of an automotive air conditioning system for ancillary load reduction," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 67–80, Jan. 2016.
[41] Y. Xie, Z. Liu, K. Li, J. Liu, Y. Zhang, D. Dan, C. Wu, P. Wang, and X. Wang, "An improved intelligent model predictive controller for cooling system of electric vehicle," *Appl. Thermal Eng.*, vol. 182, Jan. 2021, Art. no. 116084.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

SUNGHO JOO received the B.S. and M.S. degrees in mechanical engineering from Seoul National University (SNU), where he studied the application and analysis of deep learning into physics-based simulations. As part of his graduate studies in the Simulation-Driven Structure Design Laboratory, SNU, he conducted research on deep learning-based industrial image alignment with the Samsung Semiconductor Laboratory. He is currently a Machine Learning Engineer with MakinaRocks. His main research interest includes applications of machine learning in real world problems.

DONGMIN LEE received the B.S. degree in computer science and engineering from Hanyang University. He is currently a Machine Learning Engineer with MakinaRocks. Prior to joining MakinaRocks, he worked as a Research Assistant Intern at the Robot Learning Laboratory and the Biointelligence Laboratory, Seoul National University. He is currently interested in applying machine learning technology and MLOps technology to real world problems.



MINSEOP KIM received the B.S. and M.S. degrees in computer science from Sogang University. He is currently a Machine Learning Research Engineer with MakinaRocks. Prior to joining MakinaRocks, he was employed at a Biotech Startup as a Research Intern, developing robot control algorithms with reinforcement learning. He is currently interested in solving issues in various sectors by applying reinforcement learning to real-world environments.

TAEHO LEE received the B.S. degree in electronics and communications engineering and the M.S. degree in computer science from Hanyang University, in 2013 and 2015, respectively. Among his ten year research career, he has been working as a Senior Machine Learning Research Engineer with MakinaRocks, since 2021. Prior to MakinaRocks, he worked as a Machine Learning Specialist at the Technical University of Denmark and as a Senior Researcher at the AI Laboratory of LG Electronics until he moved to Denmark. His research interests include at the intersection of machine learning, scientific simulation, and control.

SANGHYEOK CHOI is currently pursuing the bachelor's degree in industrial engineering and business administration from Seoul National University. He is currently a Machine Learning Research Engineer Intern with MakinaRocks. His current research interests include machine learning, reinforcement learning, and their applications to real world problems.

SEUNGJU KIM is currently pursuing the master's degree in automotive engineering from Hanyang University. He is currently a Machine Learning Research Engineer Intern with MarkinaRocks. His current research interests include reinforcement learning, distributed learning, and deep dynamics modeling.

JEYEOL LEE received the B.Sc. and M.Sc. degrees in computer engineering from Dankook University, in 2016 and 2018, respectively. He is currently a Machine Learning Engineer with MakinaRocks and also leads the Robot Offline-Programming Team. His current research interests include reinforcement learning, metaheuristic optimization, and robotics.

JOONGJAE KIM received the B.S. degree in electrical engineering from Purdue University. He was an electrical engineer at Hyundai Heavy Industry. He is currently a Senior Research Engineer with Hanon Systems Company. He is focusing on applying AI technologies into vehicle thermal energy management system control.

YONGSUB LIM received the Ph.D. degree from the School of Computing, KAIST. He continued his work as a Postdoctoral Researcher at the Data Mining Laboratory, Seoul National University. He is currently a Co-Founder and a Chief Data Scientist with MakinaRocks. Leading the ML Solution Team, he is focusing on developing machine learning technology to solve challenging problems in industry and MLOps technology for continuous deployment and operation. Prior to MakinaRocks, he was a Data Scientist at SK Telecom specializing in advanced analytics and machine learning for industrial data.

JEONGHOON LEE received the B.S. and M.S. degrees in mechanical engineering from Hanyang University, Seoul, and the Ph.D. degree from the School of Mechanical and Aerospace Engineering, KAIST, where he studied thermal comfort evaluation inside a passenger vehicle compartment using 3-D image reconstruction. He is a Technical Fellow with Hanon Systems Company and played a leading role in applying CO2 sensors, humidity sensors, and external variable compressor control to mass production in vehicles. He is currently focusing on industrial artificial intelligence, as well as simulation and test using domain knowledge-based AI dynamics modeling and embedding it into control units.