IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 November 2023, accepted 2 December 2023, date of publication 5 December 2023,
date of current version 12 December 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3339826*

RESEARCH ARTICLE

# Deep Q-Network-Based Controller for Cabin Cooling System of Electric Vehicles

**WANSIK CHOI AND CHANGSUN AHN, (Member, IEEE)**
School of Mechanical Engineering, Pusan National University, Busan 46241, Republic of Korea

Corresponding author: Changsun Ahn (sunahn@pusan.ac.kr)

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF–2022R1A2C1004894).

**ABSTRACT** The efficiency of the thermal management system of electric vehicles is important because the thermal management system requires a significant amount of electric energy. Therefore, controllers of the thermal management system should be designed considering the efficiency. This paper proposes a deep Q-network-based controller for the thermal management system in electric vehicles. The deep Q-networks were designed to control each actuator and the observation signals, the action signals, and the reward function were designed to achieve requirements. The controller regulates cabin and evaporator air temperature by adjusting the compressor and cooling fan speed while minimizing energy consumption and adhering to system constraints. Unlike previous studies, this design process considers practical implementation, including a high-fidelity plant model, essential constraint conditions, and multiple objectives. Test results show lower energy consumption and better temperature regulation performance than a heuristically designed rule-based controller. This method can optimize thermal management system performance in electric vehicles, which have increased complexity and number of thermal loads that conventional control methods cannot adequately address.

**INDEX TERMS** Air conditioning, deep learning, electric vehicles, Q learning, reinforcement learning.

## I. INTRODUCTION

The role of the thermal management system (TMS) is more crucial in electric vehicles (EVs) than in conventional internal combustion engine vehicles due to the increased number of thermal loads requiring temperature regulation using refrigeration systems. In conventional vehicles, only the passenger cabin requires a refrigeration system for temperature regulation. However, in EVs, the power electronics, battery, and passenger cabin all require refrigeration for temperature regulation. As a result, TMS is one of the most power-consuming systems in EVs. Thus, a well-designed controller for TMS that minimizes energy consumption can maximize the driving range of EVs, given their limited battery capacity.

Conventionally, simple control methods such as proportional–integral–derivative (PID) control [1], [2], [3], and thermostat control [4], [5], [6] have been widely used for TMS in vehicles due to their lightweight, safe, and reliable nature, especially for conventional vehicles. However, these approaches are not sufficient for TMS in EVs due to the greater complexity involved. Unlike conventional vehicle refrigeration cycle systems, EV refrigeration cycle systems should consider the temperatures of battery and power electronics when the systems control cabin temperature. As a result, the controllers for TMS in EVs require the capability to consider complex requirements.

Another noteworthy characteristic of TMS in EVs is their large heat capacity. For instance, the battery and cabin have large heat capacities, causing the temperature changes to be slow. While this property can present a challenge for high responsive control performance, it can be beneficial for disturbance attenuation due to large inertia. Additionally, immediate cooling demand does not need to be met due to the large heat capacities, allowing for the temporal load distribution to the cooling supply system.

Considering the characteristics of the TMS in EVs, fuzzy logic controllers (FLCs) could be a solution. Since FLCs are highly flexible and easy to utilize knowledge with

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

multi-valued logic, many studies about environment condition control have used FLCs [1], [7], [8], [9], [10], [11]. However, it is hard to consider the trade-off between temperature regulation performance and power consumption of the TMS explicitly. On the other hand, horizon-based optimal control methods, which perform optimal planning using system dynamics and information on expected disturbances for a future horizon, are promising candidates for TMS controllers. For instance, many studies have employed model predictive control (MPC) [12], [13], [14], [15]. However, since a control model with many states leads to high computational loads and often prevents real-time implementation, simplified control models are often employed in MPC. Nonetheless, given the complexity of the TMS, including nonlinearity and numerous states, using MPC with such a simple control model may lead to performance degradation.

Another horizon-based optimal control method for TMS controller design is reinforcement learning (RL). RL is a methodology for designing an optimal policy in a given Markov decision process, where the optimal policy is computed for the infinite horizon in a stochastic sense. In contrast to MPC, RL searches for a policy that maximizes the cumulative reward without relying on a control model. This is achieved by utilizing reward signals extracted directly from the true plant, making RL less dependent on the complexity of the plant compared to MPC.

Although there have been numerous publications on the use of RL for controlling building energy systems [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], there are only a few studies on its application to thermal management controllers of EVs [26], [27], [28], [29], [30]. This is primarily due to the recent growth in the market share of EVs, which has brought about the need for energy-optimal control of TMS. As a result, the current state of RL research in TMS applications of EVs is relatively primitive in terms of practicality. Most papers on RL-based controllers for TMS of EVs have demonstrated controller designs using a simplified plant [26], [27], [28], [29] or a reward function that does not consider practical requirements [26], [27], [28], [29], [30]. To the authors' knowledge, there are no robust papers on RL-based TMS controller design that employ sufficiently complex plant and reward functions that consider practical constraints.

This study proposes a TMS controller for EVs that utilizes a deep Q-network (DQN) to minimize energy consumption while maintaining temperature regulation performance and to consider system constraints to ensure feasibility. The objective of the controller is to swiftly regulate cabin and evaporator air temperature to the desired values while minimizing power consumption and adhering to system constraints by controlling the compressor and the cooling fan.

To achieve the objective, we designed a DQN structure with two separated DQNs to handle the high dimensionality due to multiple control inputs and the nature of the DQN, the compressor, and the cooling fan. Also, we modified the DQNs to reduce the uncertainty of data and to boost the efficiency of training. In addition, we designed a reward function to consider the system constraints such as refrigerant pressure limitation and the boundary of the action signals.

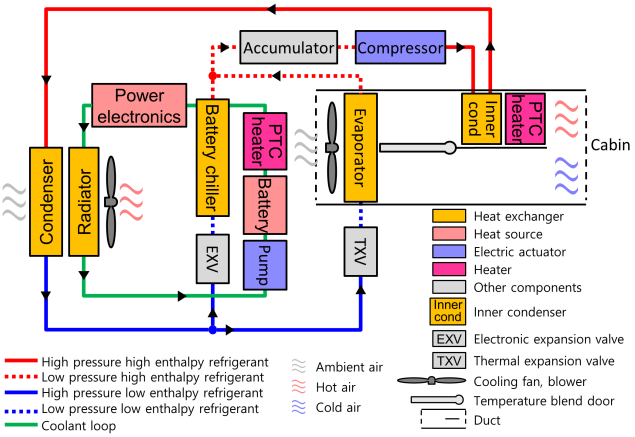


**FIGURE 1.** TMS of the EV.

In contrast to existing research on RL-based TMS controllers for EVs [21], [22], [23], [24], [25], this study presents a comprehensive design that fully considers the feasibility of practical implementation as mentioned earlier.

## II. TMS OF EV

The target plant is a TMS of a medium-sized EV. The TMS controls the temperatures of the cabin air, battery, and electronics by transferring heat via refrigerant and coolant. The TMS has multiple modes for efficient cooling and heating. In this study, we focus on the air conditioning mode, which is one of the most power-consuming tasks of the system. Fig. 1 shows the system's structure with the air conditioning mode.

The TMS with the air conditioning mode works as follows: to cool the cabin air, the refrigerant is cooled by controlling the compressor and the cooling fan speeds. The cooled refrigerant cools the evaporator outlet air temperature by absorbing heat via the evaporator. If the evaporator outlet air is too cold, the temperature blend door increases the amount of the air that absorbs heat from the inner condenser to heat the duct outlet air slightly. If the heat from the inner condenser is not sufficient, the positive temperature coefficient (PTC) heater increases the temperature to the appropriate level. To cool the battery and the electronics, the pump circulates the coolant to absorb heat and dump the heat at the radiator. If the radiator is not sufficient to dump the heat, the electronic expansion valve is opened and the refrigerant absorbs the heat from the coolant at the battery chiller.

## III. CONTROLLER DESIGN

To control the TMS, several actuators must be taken into consideration. In this study, the proposed controllers regulate the compressor and cooling fan speeds, which govern the refrigeration process. The thermal expansion valve is excluded from the controllers since it is typically controlled

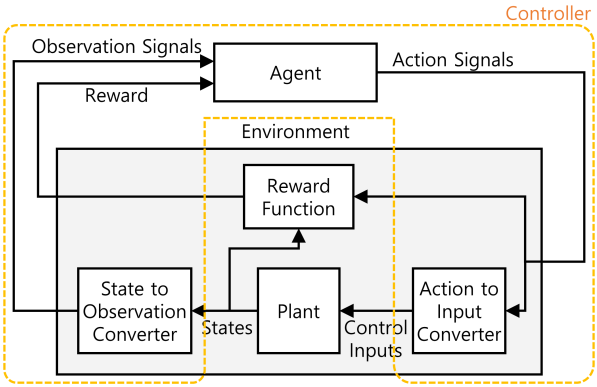


**FIGURE 2.** Controller design framework using RL.

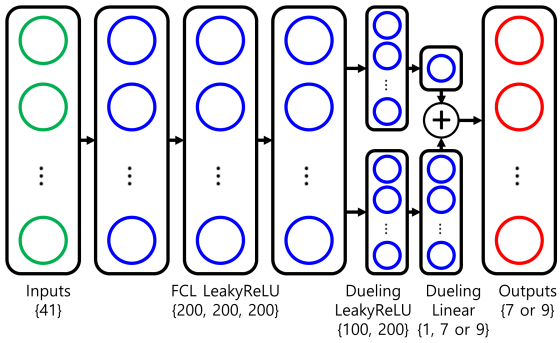


**FIGURE 3.** Structure of the DNN of the DQNs.

automatically, and the blower speed is not included since it is typically set by the passengers. Other actuators, such as the temperature blend door and PTC heater, are controlled by simple controllers.

The proposed controllers are designed using RL. Fig. 2 illustrates the controller design framework using RL. From the perspective of RL, the framework consists of an agent and an environment. The agent learns a policy that is a decision-making rule and makes decisions to achieve its goal. The environment is a set of observations, actions, and a reward function. From the perspective of the controller, the controller comprises the agent, the state-to-observation converter, and the action-to-input converter.

The agent sends action signals to the environment, which determines the control inputs of the plant. However, action signals may not be equivalent to the physical control inputs, as they are often designed to improve reinforcement learning (RL) training. To address this, an action-to-input converter is used to convert the action signals into the appropriate control inputs. The states of the plant are then updated, and to ensure that they are measurable and to improve RL training, a state-to-observation converter converts states into observation signals. The reward function calculates the immediate goal based on the action signals and states. The agent considers the observation signals when giving action signals to the environment. Finally, the agent learns the optimal policy from the interaction data, which includes observation signals, action signals, and rewards.

### A. AGENT

This study uses the DQN algorithm to control the TMS for three reasons. First, DQN is an RL algorithm that does not require a simplified control model for model training. Second, DQN is flexible in the number of observations due to its use of deep neural networks (DNNs), which is important for practical requirements that may require additional observations. Lastly, DQN is advantageous over policy gradient-based algorithms such as deep deterministic policy gradient, as it can avoid local optima due to its discretization of the action space, which allows for more aggressive policy updates.

As the proposed controllers have two control inputs, there exist two independent action spaces. Integrating these into a single action space that considers all possible cases for the DQN is impractical due to the high dimensionality of the resulting action space. Thus, we separated the DQN into a compressor DQN and a cooling fan DQN. The compressor DQN handles 7 cases of the compressor's quantized action signal, while the cooling fan DQN handles 9 cases of the cooling fan's quantized action signal.

Both DQNs use multi-layer perceptron DNNs with a leaky rectified linear unit activation function for the hidden layers. Additionally, the dueling network architecture is employed to improve performance by separating the advantages of each action from the action value [31]. Fig. 3 shows the detailed structure of the DNNs using fully connected layers (FCL), with the compressor DQN having 7 output nodes and the cooling fan DQN having 9 output nodes.

Algorithm 1 outlines the training process for the separated DQNs used in TMS. The approach is based on double DQN [32], which mitigates the problem of overestimation bias. To handle the separated DQNs, one DQN explores while the other DQN follows the greedy policy. After each episode, the two DQNs swap their roles, which helps reduce the uncertainty of the samples and facilitate efficient training. During training, the DQN with the greedy policy explores with a minimum exploration rate $\epsilon_{min}$ to avoid local optima.

To make the training process more efficient, the DQNs are updated multiple times in a single step, thereby minimizing the waiting time for observation signals. However, several updates at early steps can lead to overfitting. To mitigate this issue, the DQNs are slowly updated during the initial stages of training. Specifically, when the replay buffer is less than 5 percent full, the DQNs are updated once every 10 steps. When the buffer is between 5 and 10 percent full, the DQNs are updated once per step. Once the buffer is over 10 percent full, the DQNs are updated four times per step. Note that Algorithm 1 does not include the slow updates.

When training DQN, the weights of the target action value functions $\hat{Q}$ are periodically updated with the weights of

**Algorithm 1** Training DQNs for TMS

Initialize action value functions $Q_{comp}$ and $Q_{cfan}$
Initialize target action value functions $\hat{Q}_{comp}$ and $\hat{Q}_{cfan}$
Initialize experience replay memories $D_{comp}$ and $D_{cfan}$ to capacity $N$
Initialize step counter for $\boldsymbol{\epsilon - greedy}$, $k = 0$
Initialize loss pass filtered losses, $L_{comp}$, $L_{cfan}$ as 1.0
Set parameter for the $\boldsymbol{\epsilon - greedy}$, $\boldsymbol{\epsilon}_{min}$, $\boldsymbol{\epsilon}_{upper}$, $\boldsymbol{\epsilon}_{lower}$, $K_{\boldsymbol{\epsilon}}$
Set parameters for the adaptive soft target update, $\boldsymbol{\rho}_0 \boldsymbol{\rho}_L$, $K_{\boldsymbol{\rho}}$, $\boldsymbol{\rho}_{min}$
Set the discount factor $\boldsymbol{\gamma}$, the update number $N_{upt}$, and the learning rate $\boldsymbol{\alpha}$ For episode = 1: $M$ do
  $\boldsymbol{\epsilon} = \text{clip}\left(\frac{1}{K_{\boldsymbol{\epsilon}}(k-1)+1}, \boldsymbol{\epsilon}_{lower}, \boldsymbol{\epsilon}_{upper}\right)$
  If mod(episode,2) is 1
    $\boldsymbol{\epsilon}_{comp} = \epsilon$, $\boldsymbol{\epsilon}_{cfan} = \boldsymbol{\epsilon}_{min}$
  Else
    $\boldsymbol{\epsilon}_{comp} = \boldsymbol{\epsilon}_{min}$, $\boldsymbol{\epsilon}_{cfan} = \epsilon$
  End
  Initialize observation $\boldsymbol{\phi}$
  For $t$ =1:$T$ do
    $a_{comp} = \epsilon - greedy\left(Q_{comp}, \boldsymbol{\epsilon}_{comp}\right)$,
    $a_{cfan} = \epsilon - greedy\left(Q_{cfan}, \boldsymbol{\epsilon}_{cfan}\right)$
    $\boldsymbol{\phi}_j^{'}, r, d, \tau = env\left(a_{comp}, a_{cfan}\right)$
    Store $\left(\boldsymbol{\phi}, a_{comp}, r, \boldsymbol{\phi}_j^{'}, d\right)$ in $D_{comp}$
    Store $\left(\boldsymbol{\phi}, a_{cfan}, r, \boldsymbol{\phi}_j^{'}, d\right)$ in $D_{cfan}$
    For k=1:$N_{upt}$
    For $n = [comp, cfan]$
      Sample random minibatch $\left(\boldsymbol{\phi}_j, a_{n,j}, r_j, \boldsymbol{\phi}_j^{'}, d_j\right)$ from $D_n$
      Set $y_j = r_j + (1-d_j)\gamma \hat{Q}_n\left(\boldsymbol{\phi}_j^{'}, argmax_{A'}\left(Q_n\left(\boldsymbol{\phi}_j^{'}, A'\right)\right)\right)$
      Calculate loss $L = mean\left(\left(y_j - Q_n\left(\boldsymbol{\phi}_j, a_{n,j}\right)\right)^2\right)$
        Update $Q_{comp}$ using gradient descent to minimize $L$
        $L_n = \boldsymbol{\rho}_L L_n + \left(1-\boldsymbol{\rho}_L\right) L$
        If $L_n > 1$
          $\boldsymbol{\rho} = \boldsymbol{\rho}_{min}$
        Else
          $\boldsymbol{\rho} = max\left(\rho_0 exp\left(-K_{\boldsymbol{\rho}} \cdot L_n\right), \boldsymbol{\rho}_{min}\right)$
        End
          $\hat{Q}_n \leftarrow \boldsymbol{\rho}\hat{Q}_n + (1-\rho)Q_n$
      End
    End
      $\boldsymbol{\phi} = \boldsymbol{\phi}_j^{'}$
      $k = k + 1$
    End
  End

$Q$. However, determining the appropriate period for these updates can be challenging. To address this, we employed the soft target update method, which gradually updates $\hat{Q}$ with a target update rate $\rho$ (similar to a low-pass filter) [33]. Despite the use of soft target updates, we occasionally observed loss divergence. To prevent this, we applied an adaptive target update rate. Specifically, when the low-pass filtered loss is high, $\rho$ decreases, and when the low-pass filtered loss is low, $\rho$ increases. This approach effectively suppressed divergence.

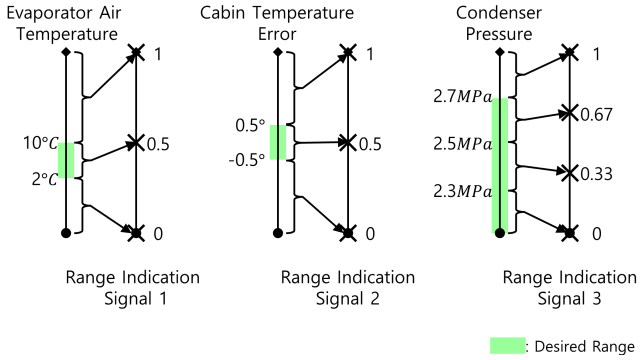


**FIGURE 4.** Range indication signals.

### B. OBSERVATION SIGNALS

The observation signals in this study comprise three types of signals: base signals, rate of change (ROC) signals, and supporting signals. The base signals provide information on ambient and cabin conditions, refrigerant and coolant states, and actuator states. These signals are selected based on their relevance to the TMS and their measurability in a real-world TMS of an EV. Table 1 lists the 18 base signals (No. 1-18). As the TMS has slow dynamics and the base signals alone cannot fully represent the system, the ROC of the base signals is chosen as the second type of observation signal to support the notice of system states. Table 1 also lists the 18 ROC signals (No. 19-36).

The supporting signals are designed signals that consist of range indication signals and rescaled signals. Range indication signals indicate the range of three signals with a desired range, namely, the evaporator air temperature, the cabin temperature error, and the condenser pressure. These signals explicitly provide information to DNNs on whether the signals are in the desired range, as shown in Fig. 4. Table 1 lists the three range indication signals (No. 37-39).

The first range indication signal is generated from the evaporator air temperature, with a desired range of 2 to 10°C. The signal value is 0 if the temperature is lower than the range, 0.5 if the temperature is within the range, and 1 if the temperature is higher than the range. The second range indication signal is similar but for the cabin temperature error, with a desired range of -0.5 to 0.5°C.

In contrast, the condenser pressure has a strict upper limit of 2.7 MPa to prevent leakage, but no lower boundary for the desired range. The desired range is from negative infinity to 2.7 MPa, divided into three sub-ranges to provide information on the approach to the upper limit. If the pressure is less than 2.3 MPa, the range indication signal value is 0, and if the pressure is less than 2.5 MPa and greater than or equal to 2.3 MPa, the signal value is 0.33. If the pressure is less than

**TABLE 1. Observation signals.**

| No. | Name | Type | Unit |
|---|---|---|---|
| 1 | Ambient temperature | Base | $°C$ |
| 2 | Vehicle velocity | Base | $m/s$ |
| 3 | Solar intensity | Base | $W/m^2$ |
| 4 | Condenser pressure | Base | $kPa$ |
| 5 | Cabin relative humidity | Base | $\%$ |
| 6 | Evaporator air temperature | Base | $°C$ |
| 7 | Desired evaporator air temperature | Base | $°C$ |
| 8 | Evaporator air temperature error | Base | $°C$ |
| 9 | Duct air temperature | Base | $°C$ |
| 10 | Cabin air temperature | Base | $°C$ |
| 11 | Desired cabin air temperature | Base | $°C$ |
| 12 | Cabin air temperature error | Base | $°C$ |
| 13 | LTR coolant temperature | Base | $°C$ |
| 14 | Compressor power | Base | $W$ |
| 15 | Compressor speed | Base | RPM |
| 16 | Cooling fan speed | Base | RPM |
| 17 | PTC power | Base | $W$ |
| 18 | Temperature blend door | Base | - |
| 19 | ROC of ambient temperature | ROC | $°C/s$ |
| 20 | ROC of vehicle velocity | ROC | $m/s^2$ |
| 21 | ROC of solar intensity | ROC | $W/m^2s$ |
| 22 | ROC of condenser pressure | ROC | $kPa/s$ |
| 23 | ROC of cabin relative humidity | ROC | $\%/s$ |
| 24 | ROC of evaporator air temperature | ROC | $°C/s$ |
| 25 | ROC of desired evaporator air temperature | ROC | $°C/s$ |
| 26 | ROC of evaporator air temperature error | ROC | $°C/s$ |
| 27 | ROC of duct air temperature | ROC | $°C/s$ |
| 28 | ROC of cabin air temperature | ROC | $°C/s$ |
| 29 | ROC of desired cabin air temperature | ROC | $°C/s$ |
| 30 | ROC of cabin air temperature error | ROC | $°C/s$ |
| 31 | ROC of LTR coolant temperature | ROC | $°C/s$ |
| 32 | ROC of compressor power | ROC | $W/s$ |
| 33 | ROC of compressor speed | ROC | RPM/$s$ |
| 34 | ROC of cooling fan speed | ROC | RPM/$s$ |
| 34 | ROC of PTC power | ROC | $W/s$ |
| 36 | ROC of temperature blend door | ROC | $s^{-1}$ |
| 37 | Range indication of evaporator air temperature | Supporting | - |
| 38 | Range indication of cabin temperature error | Supporting | - |
| 39 | Range indication of condenser pressure | Supporting | - |
| 40 | Evaporator air temperature rescaled based on desired range | Supporting | - |
| 41 | Cabin temperature error rescaled based on desired range | Supporting | - |

2.7 MPa and greater than or equal to 2.5 MPa, the signal value is 0.67, and if the pressure is greater than or equal to 2.7 MPa, the signal value is 1.

As DQN uses DNNs, the observation signals are normalized. However, this process can decrease performance because the magnitude of the desired ranges is reduced. Therefore, rescaled signals are used to magnify the desired range for the evaporator air temperature and the cabin temperature error, as shown in Fig. 5. Table 1 lists the two rescaled signals (No. 40-41).

The first rescaled signal for the evaporator air temperature has a lower bound of 0°C and an upper bound of 12°C, which

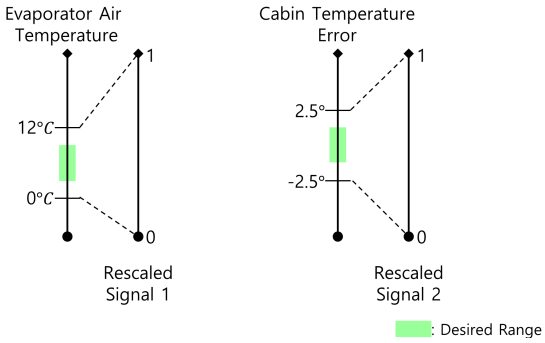


**FIGURE 5. Temperature signals rescaled based on desired ranges.**

are slightly wider than the desired range to ensure the signal is valid when the evaporator air temperature approaches the desired range sufficiently. Similarly, the second rescaled signal for the cabin temperature error has a lower bound of −2.5°C and an upper bound of 2.5°C.

The observation signals are generated from the states by the state-to-observation converter, which selects the base signals based on their relevance to the TMS and measurability. The converter then computes the ROC of the base signals and extracts supporting signals from them. In total, the observation signals comprise 41 signals, which include the 18 base signals, the 18 ROC signals, the 3 range indication signals, and the 2 rescaled signals.

### C. ACTION SIGNALS

The action signals determine the control inputs, which consist of compressor speed and cooling fan speed. As our proposed controllers are based on DQN, the action space must be discrete. However, if we quantize the compressor and cooling fan speeds, handling the rate of change constraints of the control inputs becomes difficult, and utilizing the whole continuous space of the control input becomes impossible. Therefore, we design the action space as the discretized rate of changes of the compressor and cooling fan speed. This approach ensures that the constraints of the rate of change are naturally satisfied, and the controllers can utilize the entire continuous space of the control inputs.

To determine the compressor speed, we quantize the rate of change of the compressor speed into −500, −375, −250, −125, 0, 125, and 250 RPM/s. Similarly, to determine the cooling fan speed, we quantize the rate of change of the cooling fan speed into −1000, −750, −500, −250, 0, 250, 500, 750, and 1000 RPM/s. Finally, the action-to-input converter computes integrals of the action signals to convert these action signals into control inputs.

### D. REWARD FUNCTION

The reward function was designed to achieve the objective of controlling the cabin and evaporator temperatures to desired ranges as quickly as possible while minimizing power consumption and satisfying constraints. The reward function

consists of seven terms, as shown in (1).

$$r = f_{cab}\left(e_{cab}\right) + f_{evap}\left(T_{evap}\right) + f_{cond}\left(p_{cond}\right) + f_{bnd}\left(\boldsymbol{u}, \boldsymbol{a}\right) - \alpha_1 P - \alpha_2 \sum abs\left(\boldsymbol{a}\right) - \alpha_3 \sum abs\left(\dot{a}\right) \tag{1}$$

In (1) $r$ represents reward, $\alpha_1$, $\boldsymbol{\alpha}_2$, and $\boldsymbol{\alpha}_3$ are weighting parameters, $f_{cab}$ calculates the reward for the cabin temperature error, $e_{cab}$, $f_{evap}$ calculates the reward related to the evaporator air temperature, $T_{evap}$, $f_{cond}$ calculates the reward consider the condenser pressure, $p_{cond}$, and $P$ is the sum of the power consumption of the compressor, PTC heater, and cooling fan. Additionally, $f_{bnd}$ gives a reward related to the control inputs, $\boldsymbol{u}$, and $\boldsymbol{a}$is a vector of the normalized action signals (from -1.0 to 1.0), with $\dot{a}$ being the rate of change vector of $\boldsymbol{a}$.

The first term of the reward function, $f_{cab}\left(e_{cab}\right)$, was designed to control the cabin temperature error to 0. As the absolute error becomes smaller, the reward increases linearly. When the absolute error is within 0.5°C, an additional reward is given to encourage controllers to regulate the absolute error in a sufficiently small range. The second term, $f_{evap}\left(T_{evap}\right)$, was designed to control the evaporator air temperature within the desired range of 2 to 10°C. As the evaporator air temperature gets closer to the desired range, the reward increases linearly. When the temperature is within the desired range, the reward is a constant positive value. If the temperature is less than 2°C, an additional negative reward is given to prevent water vapor from freezing. The third term, $f_{cond}\left(p_{cond}\right)$, is related to the upper limit of the condenser pressure to avoid refrigerant leakage. This term is normally 0, but when the pressure approaches the upper limit, the reward decreases dramatically. The fourth term, $f_{bnd}\left(\boldsymbol{u}, \boldsymbol{a}\right)$, is related to the lower and upper limits of the control inputs. This term normally returns 0 but returns a negative reward that is proportional to the number of control inputs that exceed the limits. The fifth term, $-\alpha_1 P$, is related to power consumption. The reward decreases linearly as power consumption increases to minimize power consumption. The sixth term, $-\alpha_2 \sum abs\left(\boldsymbol{a}\right)$, decreases linearly as the magnitude of each action signal increases to reduce the magnitudes of the action signals. Finally, the seventh term, $-\alpha_3 \sum abs\left(\dot{a}\right)$, decreases linearly as the magnitude of the rate of change signals of the action signals increases to reduce the chattering of the action signals.

The first four terms of the reward function are visualized in Fig. 6. The fourth term is visualized for one actuator with a normalized action signal $a$ and a normalized control input $u$. With the reward function, the RL algorithm learns the policy to maximize performance while considering the constraints implicitly.

### E. TRAINING AND RESULTS

A high-fidelity Simulink model of the TMS of a medium-sized EV, shown in Fig. 7, was used to train the DQNs through simulation. This model accounts for driving conditions such as the driving cycle, ambient air temperature and relative humidity, solar intensity, and desired cabin temperature.

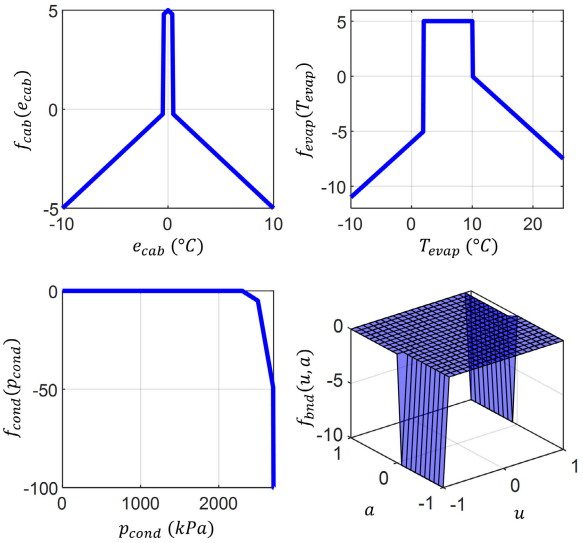


**FIGURE 6.** Graphs of the reward function terms.

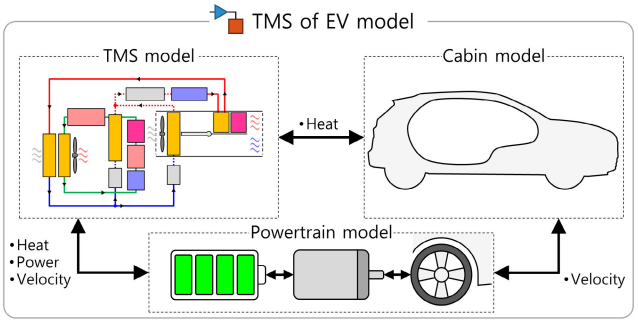


**FIGURE 7.** Training and validation platform in Simulink.

**TABLE 2.** Driving conditions for training.

| Driving Conditions | Value |
|---|---|
| Driving cycle | US06 or SC03 |
| Ambient air temperature | $U_{[20,45]}$ °C |
| Ambient air relative humidity | $U_{[0,100]}$ % |
| Solar intensity | $U_{[500,1000]}$ W/m2 |
| Target cabin temperature | $20 + 0.5 \cdot U\{0,8\}$ °C |

For the implementation of the DQNs, Python and Tensorflow are employed. Additionally, the MATLAB engine is used to manage the Simulink model and TCP/IP is used to acquire the experience data and determine the actions.

To efficiently train the DQNs, driving conditions were randomly initialized at the beginning of each episode. One of the US06 and SC03 driving cycles was randomly selected, and the ambient temperature, ambient relative humidity, solar intensity, and target cabin temperature were initialized using the uniform random distribution $U$. The details of the driving conditions are presented in Table 2.

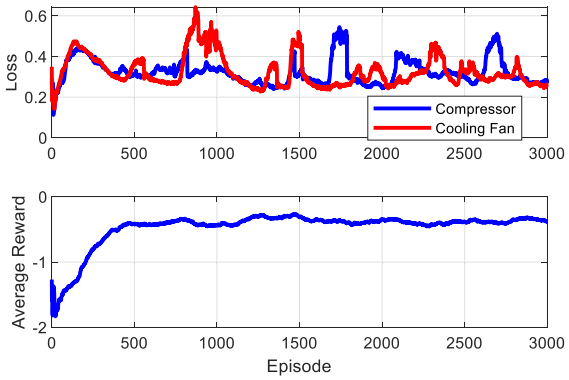


**FIGURE 8.** Loss graphs of the compressor DQN and cooling fan DQN and average reward graph for each episode.

The training process proceeded as follows: First, the DQNs and model parameters, including the driving conditions, were initialized, and the model simulation was started using Python. Subsequently, the model transmitted initial observation via TCP/IP to Python, which calculated the action signals using the DQNs and observation signals and then transmitted them back to the model. The model performed a one-step simulation using the received action signals and sent the next observation signals to Python. After that, Python updates the DQNs as introduced in Algorithm 1. This process was repeated until the training was completed.

The DQNs were trained 3000 episodes, and the loss and moving average of the expected reward are presented in Fig. 8. The loss graph shows occasional rapid increases in the loss, but it did not diverge due to the soft target update with the adaptive target update rate. The mean reward, which is the average of the reward values in the experience replay buffer, increased until the episode number reached around 1500. After that, the mean reward slightly oscillated.

## IV. CONTROLLER VALIDATION

### A. VALIDATION ENVIRONMENT

To validate the effectiveness of the proposed controller, we used the same platform for training as a validation platform, but with a different driving cycle and specified ambient conditions. The validation platform's driving conditions are outlined in TABLE 3. Fig. 9 shows the driving cycle for testing that influences power usage and temperature rise. To evaluate the performance of the proposed controller, a heuristically designed rule-based controller composed of several PID controllers is employed as a baseline controller.

### B. TEST RESULTS

Fig. 10, 11, and 12 depict the results obtained from Test 1, Test 2, and Test 3, respectively. The blue lines in each figure represent the outcomes of the rule-based controllers, whereas the red lines represent those of the RL-based controllers.

In the results of Test 1, the compressor speed was rapidly increased to reach the desired temperature level, which is a smart decision to increase the total reward. This is because most of the power consumption occurs during the transient phase and keeping the temperature level uses less power. Because of the low ambient air temperature, the evaporator

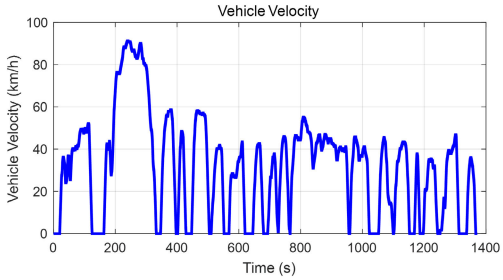


**FIGURE 9.** UDDS driving cycle used for controller validation.

**TABLE 3.** Driving conditions for controller validation.

| Driving Conditions | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 |
|---|---|---|---|---|---|
| Driving cycle | UDDS | UDDS | UDDS | UDDS | UDDS |
| Ambient air temperature | 24°C | 32°C | 40°C | 32°C | 40°C |
| Ambient air relative humidity | 50% | 50% | 50% | 90% | 90% |
| Solar intensity | 750W/m$^2$ | 750W/m$^2$ | 750W/m$^2$ | 200W/m$^2$ | 900W/m$^2$ |
| Target cabin temperature | 22°C | 22°C | 22°C | 22°C | 22°C |

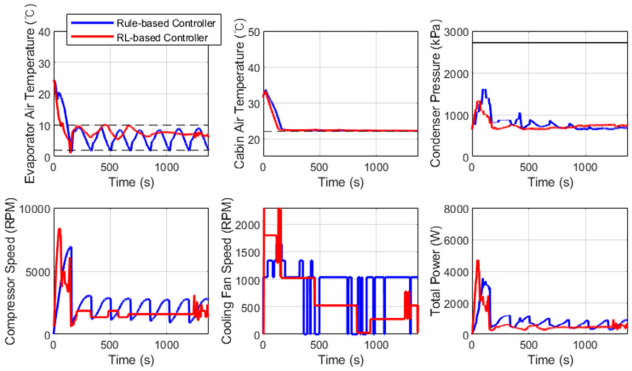


**FIGURE 10.** Test 1 result (Ambient air temperature: 24 °C).

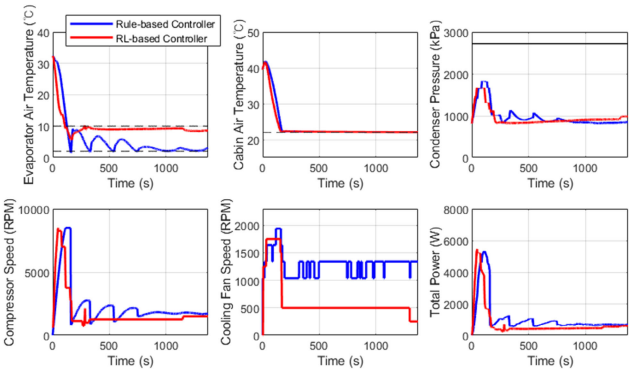


**FIGURE 11.** Test 2 result (Ambient air temperature: 32 °C).

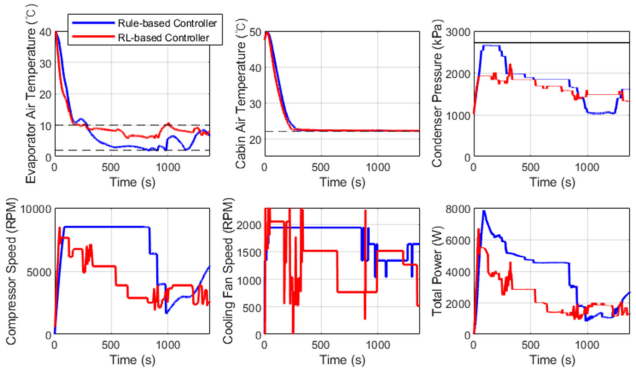


**FIGURE 12.** Test 3 result (Ambient air temperature: 40 °C).

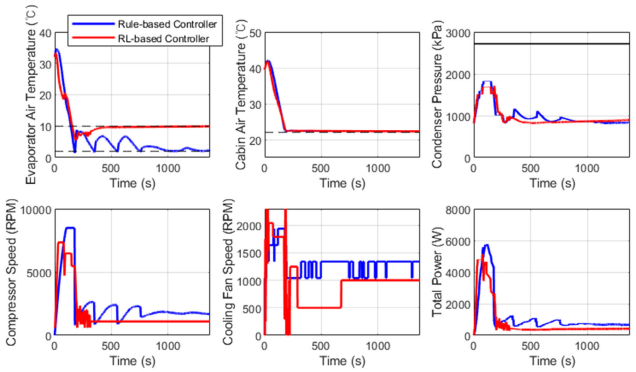


**FIGURE 13.** Test 4 result (Rainy Tropics).

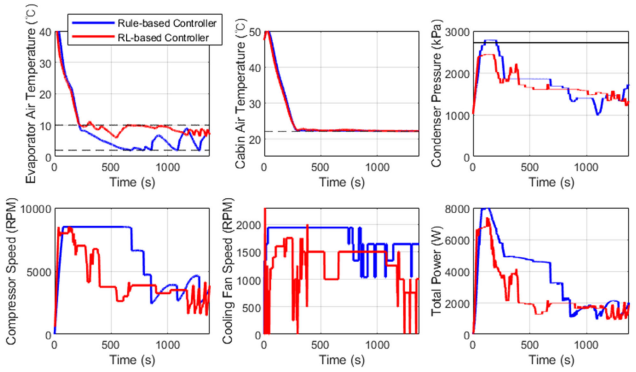


**FIGURE 14.** Test 5 result (Very hot and humid summer).

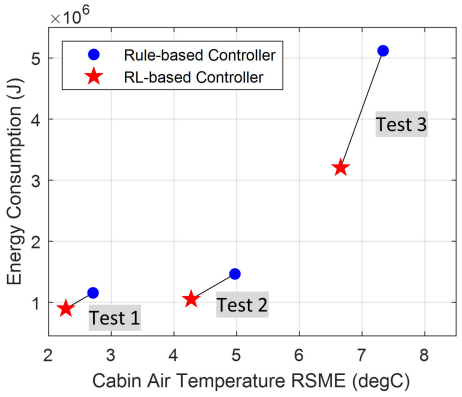


**FIGURE 15.** Performance of the RL-based controllers compared with the rule-based controllers.

air temperature could be easily lowered with smaller power consumption than in higher ambient air temperature cases.

In the results of Test 2, the characteristics were similar to those of Test 1, but the ambient air temperature was higher, requiring more energy to reduce the cabin temperature to the desired value than in Test 1. However, keeping the temperature at the desired range in Test 2 did not require much more energy than in Test 1. This indicates that most of the energy in the TMS is used during the transient phase.

In Test 3, the ambient air temperature was extremely high, resulting in slightly different behaviors from the lower ambient temperature cases. Because of the high ambient temperature, the condenser pressure was high to dump the heat from the system to the outside of the vehicle.

Fig. 13 and Fig.14 preset results from Test 4 and Test 5 which were conducted to test the controller with specific weather conditions. Test 4 emulates weather conditions when it rains in the tropics, while Test 5 emulates very hot and humid summer weather. Test 4 and Test 5 show similar results to Test 2 and Test 3.

The proposed controller does not violate the significant constraint of the condenser pressure. Additionally, the proposed controller is able to control the evaporator air temperature within the desired range most of the time, but it also adaptively handles the boundary of the evaporator air temperature to acquire future rewards.

In summary, the RL-based controllers exhibit similar patterns for each test scenario. Initially, the compressor speed rises rapidly, causing the cabin temperature to drop quickly to the target value. Once the evaporator temperature reaches a suitable level, the compressor speed slows down, leading to reduced power consumption. Additionally, to ensure efficient operation and prevent refrigerant leakage, the cooling fan speed increases when the pressure in the refrigeration system rises to ensure sufficient removal of latent heat.

In terms of both temperature tracking performance and energy consumption, the RL-based controller outperforms the rule-based controllers, as illustrated in Fig. 15 and Fig. 16. The horizontal axis represents the temperature regulation performance, and the vertical axis represents the energy consumption. In the figures, the lower left corner indicates higher performance. The marker of the RL-based controller was located closer to the lower left corner than the rule-based controller marker, indicating better performance in terms of both temperature tracking and energy consumption.

## V. CONCLUSION

This study proposed an RL-based TMS controller for EVs that minimizes energy consumption while maintaining temperature regulation performance. Unlike existing research on RL-based TMS controllers for EVs, this study presents a

comprehensive design process that fully considers the feasibility of practical implementation by using a high-fidelity plant model that accurately represents the high nonlinearity and complexity of the TMS of EVs and considering essential constraints such as refrigerant pressure limit, speed limit of the actuators, acceleration limit of the actuators, and temperature limit. Multiple objectives were considered by adding rewards for cabin and evaporator temperatures as well as energy consumption.

The proposed controller has limitations. The controller needs to be trained again when the target TMS is changed, and the action space should be discretized since the controller is based on DQN. Furthermore, the training result could be unstable due to uncertainties arising from the separated DQNs. Nevertheless, the simulation results show that the proposed controller outperforms the rule-based controller in terms of energy consumption and temperature regulation performance. The proposed controller can swiftly regulate the cabin and evaporator air temperature to the desired values while minimizing power consumption and adhering to system constraints. The proposed controller also exhibits robustness to uncertainties such as changing driving conditions.

In conclusion, the proposed RL-based TMS controller for EVs can significantly improve the energy efficiency of TMS while maintaining temperature regulation performance. The comprehensive design process ensures practical feasibility, making the proposed controller a promising candidate for future TMS implementations in EVs.

The application of the proposed RL-based TMS controller for EVs can lead to significant energy savings and increased driving range, which can improve the EV's overall efficiency and reduce its environmental impact. The results of this study can be expanded upon by exploring the use of RL-based controllers for other components of the EV, such as the powertrain and battery management systems, to further optimize the energy consumption and overall performance of the vehicle. Additionally, the design process presented in this study, which considers practical constraints and objectives, can serve as a framework for future research on RL-based controllers for TMS and other applications in EVs.

## APPENDIX A
## DETAILS OF THE TMS OF EV

The target plant of this paper is the TMS of the EV. Although this paper has worked with the Simulink model, we present wiring diagrams to give a further understanding of the TMS. Fig. 17, 18, and 19 display the wiring diagrams we supposed. All components are powered by direct current (DC) from the battery. The main controller acquires information about the TMS and determines the actuators' target via a control area network (CAN) bus as shown in Fig. 17. The wiring diagram of the actuators is shown in Fig. 18. The compressor operates with a brushless DC (BLDC) motor that requires $360V$. The controller measures the current of each phase to estimate the rotation speed and power consumption and controls the rotation speed to the desired speed. The PTC heater requires

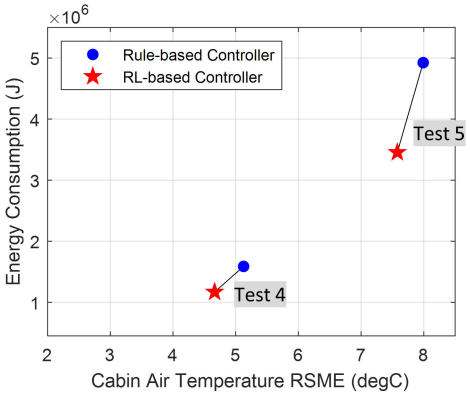


**FIGURE 16.** Performance of the RL-based controllers compared with the rule-based controllers at specific weather conditions.

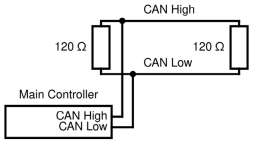


**FIGURE 17.** Wiring diagram of the main controller.

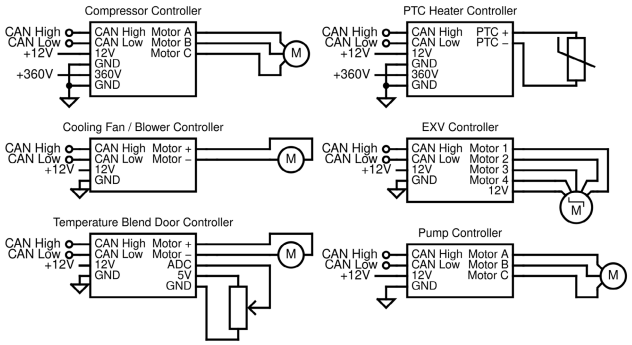


**FIGURE 18.** Wiring diagram of the actuator controllers.

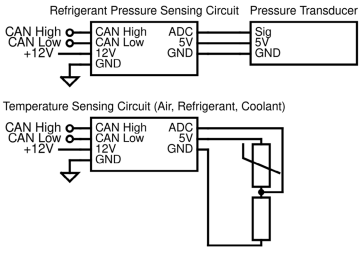


**FIGURE 19.** Wiring diagram of the sensors.

$360V$ to operate. The controller measures the current to the PTC heater to estimate the power consumption and controls heat generation. The cooling fan and blower operate using DC motors without any sensors. As a result, the controller approximates the rotational speed and current using look-up tables and determines the motor inputs. The electronic expansion valve (EXV) uses a stepper motor to control the pressure drop of the refrigerant. The controller controls the

**TABLE 4. Parameters of the algorithm.**

| Parameter | Value | Note |
|---|---|---|
| $\alpha_1$ | 0.0025 | Coefficient for the power ($P$) related term in the reward function |
| $\alpha_2$ | 5.0 | Coefficient for the normalized action signals ($\boldsymbol{a}$) related term in the reward function |
| $\alpha_3$ | 5.0 | Coefficient for the rate of change of the action signals ($\dot{\boldsymbol{a}}$) related term in the reward function |
| $N$ | 80000 | Capacity of the experience replay memory |
| $\epsilon_{min}$ | 0.05 | Probability for random action selection in the greedy policy |
| $\epsilon_{upper}$ | 0.7 | Maximum probability for random action selection during exploration |
| $\epsilon_{lower}$ | 0.1 | Minimum probability for random action selection during exploration |
| $K_\epsilon$ | 0.00002 | Parameter for decaying probability of random action selection |
| $\rho_0$ | 0.01 | Maximum target update rate |
| $\rho_L$ | 0.995 | Loss smoothing parameter |
| $K_\rho$ | 5.0 | Parameter for adaptive target update rate |
| $\rho_{min}$ | 0.000002 | Minimum target update rate |
| $M$ | 3000 | Number of episodes |
| $\gamma$ | 0.98 | Discount factor |
| $N_{upt}$ | 4 | Number of updates per step |
| $\alpha$ | 0.002 | Learning rate |

EXV to reach the desired level of superheat. The temperature blend door changes its angle using a DC motor and measures its angle using a potentiometer. The controller determines the motor input to control the door to the desired position. The pump operates with a BLDC motor, and its controller functions similarly to the compressor's controller, except it requires a 12V. Fig. 19 shows the wiring diagram for the sensors. The refrigerant pressure sensors are pressure transducers, converting pressure into voltage, and it is measured via the analog-digital converter (ADC). Similarly, the temperature sensors use thermistors where sensing resistors are connected in series and the circuit measures voltage between the thermistor and the resistor through the ADC.

## APPENDIX B
## PARAMETERS OF THE ALGORITHM

The parameters of the algorithm that we used are shown in Table 4. The reward function is scaled by a factor of 0.05 to improve the training process.

## CONFLICT OF INTEREST DECLARATION

The authors declare that they have no known conflicting interests that could have affected the results of this study.

## REFERENCES


[1] J. Wang, D. An, and C. Lou, "Application of fuzzy-PID controller in heating ventilating and air-conditioning system," in *Proc. Int. Conf. Mechatronics Autom.*, Jun. 2006, pp. 2217–2222.

[2] 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.

[3] 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.

[4] R. Kamyar and M. M. Peet, "Optimal thermostat programming for time-of-use and demand charges with thermal energy storage and optimal pricing for regulated utilities," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2714–2723, Jul. 2017.

[5] E. Vrettos, C. Ziras, and G. Andersson, "Fast and reliable primary frequency reserves from refrigerators with decentralized stochastic control," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2924–2941, Jul. 2017.

[6] X. Yan, J. Fleming, and R. Lot, "A/C energy management and vehicle cabin thermal comfort control," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11238–11242, Nov. 2018.

[7] B. S. K. K. Ibrahim, M. A. N. Aziah, S. Ahmad, R. Akmeliawati, H. M. I. Nizam, A. G. A. Muthalif, S. F. Toha, and M. K. Hassan, "Fuzzy-based temperature and humidity control for HV AC of electric vehicle," *Proc. Eng.*, vol. 41, pp. 904–910, Jan. 2012, doi: 10.1016/j.proeng.2012.07.261.

[8] S. Hussain, H. A. Gabbar, D. Bondarenko, F. Musharavati, and S. Pokharel, "Comfort-based fuzzy control optimization for energy conservation in HVAC systems," *Control Eng. Pract.*, vol. 32, pp. 172–182, Nov. 2014, doi: 10.1016/j.conengprac.2014.08.007.

[9] L. M. Escobar, J. Aguilar, A. Garcés-Jiménez, J. A. G. De Mesa, and J. M. Gomez-Pulido, "Advanced fuzzy-logic-based context-driven control for HVAC management systems in buildings," *IEEE Access*, vol. 8, pp. 16111–16126, 2020, doi: 10.1109/ACCESS.2020.2966545.

[10] M. Wozniak, A. Zielonka, A. Sikora, M. J. Piran, and A. Alamri, "6G-enabled IoT home environment control using fuzzy rules," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5442–5452, Apr. 2021, doi: 10.1109/JIOT.2020.3044940.

[11] M. Woźniak, J. Szczotka, A. Sikora, and A. Zielonka, "Fuzzy logic type-2 intelligent moisture control system," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121581, doi: 10.1016/j.eswa.2023.121581.

[12] J. Lopez-Sanz, C. Ocampo-Martinez, J. Alvarez-Florez, M. Moreno-Eguilaz, R. Ruiz-Mansilla, J. Kalmus, M. Gräeber, and G. Lux, "Nonlinear model predictive control for thermal management in plug-in hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3632–3644, May 2017.

[13] Y. Masoudi and N. L. Azad, "MPC-based battery thermal management controller for plug-in hybrid electric vehicles," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 4365–4370.

[14] M. R. Amini, H. Wang, X. Gong, D. Liao-McPherson, I. Kolmanovsky, and J. Sun, "Cabin and battery thermal management of connected and automated HEVs for improved energy efficiency using hierarchical model predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 1711–1726, Sep. 2020.

[15] M. R. Amini, I. Kolmanovsky, and J. Sun, "Hierarchical MPC for robust eco-cooling of connected and automated vehicles and its application to electric vehicle battery thermal management," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 316–328, Jan. 2021.

[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. Wang, K. Velswamy, and B. Huang, "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, Aug. 2017.

[18] T. Wei, Y. Wang, and Q. Zhu, "Deep reinforcement learning for building HVAC control," in *Proc. 54th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2017, pp. 1–6.

[19] 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.

[20] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, and K. P. Lam, "A deep reinforcement learning approach to using whole building energy model for HVAC optimal control," in *Proc. Building Perform. Anal. Conf. SimBuild*, vol. 3, 2018, pp. 22–23.

[21] 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.

[22] 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.
[23] X. Fang, G. Gong, G. Li, L. Chun, P. Peng, W. Li, X. Shi, and X. Chen, "Deep reinforcement learning optimal control strategy for temperature setpoint real-time reset in multi-zone building HVAC system," *Appl. Thermal Eng.*, vol. 212, Jul. 2022, Art. no. 118552.
[24] Q. Fu, X. Chen, S. Ma, N. Fang, B. Xing, and J. Chen, "Optimal control method of HVAC based on multi-agent deep reinforcement learning," *Energy Buildings*, vol. 270, Sep. 2022, Art. no. 112284.
[25] D. Weinberg, Q. Wang, T. O. Timoudas, and C. Fischione, "A review of reinforcement learning for controlling building energy systems from a computer science perspective," *Sustain. Cities Soc.*, vol. 89, Feb. 2023, Art. no. 104351.
[26] 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.
[27] A. B. Kumar, "Battery thermal management for an urban electric freight vehicle using reinforcement learning," M.S. thesis, Dept. Mech. Eng., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2020.
[28] G. Chen, "Policy gradient reinforcement learning-based vehicle thermal comfort control," Coventry Univ., Tech. Rep., 2021.
[29] G. Huang, P. Zhao, and G. Zhang, "Real-time battery thermal management for electric vehicles based on deep reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 14060–14072, Aug. 2022.
[30] S. Joo, D. Lee, M. Kim, T. Lee, S. Choi, S. Kim, J. Lee, J. Kim, Y. Lim, and J. Lee, "Multi-agent reinforcement learning based actuator control for EV HVAC systems," *IEEE Access*, vol. 11, pp. 7574–7587, 2023.
[31] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.
[32] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 2094–2100.
[33] 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*.

**WANSIK CHOI** received the B.S. degree in mechanical engineering from Pusan National University, in 2016, where he is currently pursuing the Ph.D. degree with the School of Mechanical Engineering. He has experience in mechatronics and artificial intelligence systems. His research interest includes control and estimation of automotive systems, in particular, using AI-based methodologies.

**CHANGSUN AHN** (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, Seoul, South Korea, in 1999 and 2005, respectively, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2011. He is currently a Professor with Pusan National University, South Korea. His research interest includes automotive control/estimation. Recently, he has focused on autonomous vehicle control.