Policy gradient reinforcement learning-based vehicle thermal comfort control

Chen, Gaobo

Award date:
2021

POLICY GRADIENT REINFORCEMENT LEARNING-BASED VEHICLE THERMAL COMFORT CONTROL

GAOBO CHEN

A thesis submitted in partial fulfilment
of the University's requirements for the degree of
Doctor of Philosophy

January 2020 – version 1.0

ABSTRACT

Reinforcement learning (RL) methods have been developed to deal with numerous real world tasks including applications that focus on the climate controls for different indoor environment including office, classroom, house and car cabin. Recent research applying in car cabin climate control is based on the State-Action-Reward-State-Action (SARSA) algorithms to train an artificial agent that can automatically maintain the thermal conditions that satisfy occupant comfort. However, the SARSA-based RL approaches usually spend 2.9 to 6.3 years of simulated learning experience on training a near-optimal control policy. This cost is not negligible in comparison with the lifetime of vehicles. Alternatively, the family of policy gradient reinforcement learning (PGRL) algorithms has potential to accelerate the training process and acquire less learning experience.

Hence, the main aim of this thesis is to apply PGRL approaches in learning vehicle climate control and assess if the resulting controller can maximally achieve occupant comfort with reasonable energy consumed by the thermal conditioning system. In order to achieve this main goal, a multilayer perceptron (MLP) based neural network with softmax output layer is used as thermal control policy, the PGRL schemes basically maximize received rewards to compute the gradients to update the weights of this control policy. Two primitive PGRLs are applied and compared: the Monte-carlo policy gradient (MCPG) and mean actor critic (MAC). However, the main difficulty of using primitive PGRL methods is that the learning step size computed by direct gradient-descent rules does not always improve the policy. This issue can be solved by employing two typical PGRL approaches: trust region policy optimization (TRPO) and proximal policy optimization (PPO).

The experiment shows that TRPO and PPO approaches can improve the sample efficiency and with a reduced simulated learning time of 0.63 years. The PPO based training scheme statistically yields higher episodic reward per learning trial than the alternative PGRLs. Additionally, the PPO-based controller achieves occupant comfort averagely in 3.8 minutes, and maintains 77.94% time spent on the comfort. Compared to the SARSA-based controls with pre-selected testing scenarios, the PPO-based one achieves 92.3% occupant comfort which is higher than the 67% achieved by the SARSA-based controller. Moreover, the state representation is non-Markovian due to its dependence on the time steps. As the validation shows that increasing the episode duration from 1000 to 5000 s can significantly improve the comfort maintaining performance and averaged episodic rewards. A

Markovian state representation is then introduced to mitigate state dependence on time-step, the case with 4×10^3 s duration shows that using Markovian state representation can improve comfort percentage from 53.58% to 64.32%. But this improvement is lower than 77.94% by the non-Markovian training case with 5×10^3 s episode time. The trade-off is that the non-Markovian learning case consumes 20% more simulated time in estimating comfort-oriented controller that maintains 13% more time spent on comfort.

Therefore, the PPO-based PGRL climate controller can significantly improve the occupant comfort percentage to above 77%, while using less simulated learning time (0.63 years) with the non-Markovian state representation. The simulated time is much less compared to the vehicle's lifetime. Furthermore, other innovative policy gradient techniques, such as, Actor Critic with experience replay and Trust Region-Guided PPO have potentials to further reduce the 0.63 years of learning sample by PPO method, and a more realistic human thermal comfort model is needed.

ACKNOWLEDGMENTS

At the beginning of my thesis, I would like to express my sincere gratitude to my Director of Studies, Prof. James Brusey, for giving me this precious opportunity to independently carry out this research. His expertise, guidance and passion lead me to successfully overcome all the hardness moment throughout this research journey. Also, I would like to thank my second supervisor, Prof. Elena Gaura, for continuously motivating me through her constructive criticism to carry out my research.

I am also extremely grateful for the unforgettable moment that I have shared with my colleagues in the lab: Dr Alexandra Petre, Dr Kojo Sarfo Gyamfi, Gene Palencia, Nicolas Melinge, Dr Ross Wilkins, James Westcott, and Ross Drury. Thank you to Zihao Liu, Xiaoyang Zhang, Jin Ren and Susan Lasslett for their kindness.

Finally, I would like to express my love and gratitude to my family members, especially my mum and dad for their long-term care and support. I cannot get this research done without their dedication and faith on me.

CONTENTS

1 INTRODUCTION 1
- 1.1 Research aim 1
- 1.2 Research questions 4
- 1.3 Contribution to knowledge 6
- 1.4 Thesis Structure 7

2 BACKGROUND 9
- 2.1 Markov Decision Process 9
- 2.2 Reinforcement learning with value-based approaches 12
  - 2.2.1 Value functions and policies 13
  - 2.2.2 Classical learning algorithms 14
  - 2.2.3 Value function approximation methods 16
- 2.3 Reinforcement learning with policy gradient approaches 18
  - 2.3.1 The policy and trajectory 18
  - 2.3.2 A Monte Carlo-based formulation of policy gradient 20
  - 2.3.3 Application of advantage function 22
  - 2.3.4 Mean Actor Critic method 23
  - 2.3.5 Trust region policy optimization and Proximal policy optimization approaches 23
  - 2.3.6 The representative policy gradient reinforcement learning applications 24
- 2.4 Reinforcement learning applications for indoor thermal conditioning 27
- 2.5 Summary 30

3 POLICY GRADIENT REINFORCEMENT LEARNING BASED VEHICLE CLIMATE CONTROL 33
- 3.1 Problem statement 34
  - 3.1.1 Policy network and control variables 35
  - 3.1.2 Car cabin thermal environment 36
  - 3.1.3 Reward function definition 38
- 3.2 Experimental settings and evaluation methods 39
- 3.3 Results and discussions 42
  - 3.3.1 Description for resulting data illustration 43
  - 3.3.2 Policy gradient methods comparison 44
  - 3.3.3 Comparison between TRPO and PPO 50
- 3.4 Chapter Summary 59

4 A MARKOVIAN STATE REPRESENTATION FOR LEARNING HVAC CONTROL 61
- 4.1 Non-Markov decision process of PGRL HVAC system 62

4.1.1 Problem statement 62
4.1.2 The NMDP and the termination time 63
4.1.3 Experiment setting 64
4.1.4 Results and analysis 64
4.2 MDP-based model for PGRL HVAC system 67
4.2.1 Results of MDP-based training and testing 68
4.3 Chapter Summary 71
5 POLICY GRADIENT REINFORCEMENT LEARNING FOR A SIMULATION BASED ON CLIMATIC WIND TUNNEL EXPERIENCE 75
5.1 The information of state, control action and thermal model 76
5.2 RL control policy and baseline control implementation 78
5.3 Experiment setting 81
5.3.1 Equivalent temperature and reward function 81
5.4 Results and discussion 83
5.4.1 The settings for environmental factors 83
5.4.2 The episodic rewards throughout the learning trials 85
5.4.3 Warm-up test in cold climate 85
5.4.4 Cool-down test without solar radiation on cabin roof 90
5.4.5 Cool-down test with solar radiation on cabin roof 94
5.4.6 Comfort percentage and power consumption 99
5.5 Chapter summary 105
6 CONCLUSIONS 107
6.1 Answers for Research questions 107
6.2 Future work 109
A APPENDIX 115
A.1 Algorithms 115
A.1.1 Traditional value-based RLs and simple policy gradient method 115
A.1.2 Advantage actor critic and Mean actor critic 116
A.1.3 Trust region policy optimization 116
A.1.4 Proximal policy optimization 123

BIBLIOGRAPHY 127

LIST OF FIGURES

Figure 2.1 A cliff walk game 11
Figure 2.2 A cart-pole balancing task 11
Figure 2.3 Tabular Q-function upgrade 15
Figure 2.4 Policy inside the Reinforcement Learning 18
Figure 2.5 Simulations of robotics locomotion control: "Humanoid", "Ant walk", "Half-cheetah running", "Lion running", "Atlas robot walking upstairs" and "Ball-throw task" (clockwise-listed) via Policy gradient reinforcement learning, 25
Figure 3.1 A fully connected feed-forward policy network based HVAC control 34
Figure 3.2 Averaged episodic rewards against learning trials by MCPG, MAC, TRPO and PPO methods 45
Figure 3.3 Averaged episodic comfort rate against learning trials by MCPG, MAC, TRPO and PPO methods 45
Figure 3.4 Thermal comfort testing results of the MCPG, MAC, TRPO and PPO control policies acting on 4×10^3 randomly selected initial states and 200 pre-selected ones from earlier work 46
Figure 3.5 Power consumption testing results of the MCPG, MAC, TRPO and PPO control policies acting on 4×10^3 randomly selected initial states and 200 pre-selected ones from earlier work 46
Figure 3.6 Occupant's equivalent temperature warm-up processes by MCPG, MAC, TRPO and PPO control policies under the environment, cabin air and mass temperature in 1°C 48
Figure 3.7 Occupant's equivalent temperature cool-down processes by MCPG, MAC, TRPO and PPO control policies under the environment temperature in 40°C and cabin air, mass temperature in 45°C respectively 48
Figure 3.8 Comfort rate results by TRPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$ 51

Figure 3.9 Power consumption testing results by TRPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$ 51
Figure 3.10 Warm-up process of occupant equivalent temperature by TRPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 1°C environment, cabin air and interior mass temperature 52
Figure 3.11 Physical temperatures of cabin T_c and interior mass T_m in the warm-up process, trained with 1×10^4 learning episodes 52
Figure 3.12 Cool-down process of occupant equivalent temperature by TRPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 40°C environment temperature and 45°C cabin air, interior mass temperature 53
Figure 3.13 Physical temperatures of cabin T_c and interior mass T_m in the cool-down process, trained with 1×10^4 learning episodes 53
Figure 3.14 Comfort rate results by PPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$ 54
Figure 3.15 Power consumption testing results by PPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$ 54
Figure 3.16 Warm-up process of occupant equivalent temperature by PPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 1°C environment, cabin air and interior mass temperature 55
Figure 3.17 Physical temperatures of cabin T_c and interior mass T_m in the warm-up process, trained with 1×10^4 learning episodes 55
Figure 3.18 Cool-down process of occupant equivalent temperature by PPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 40°C environment temperature and 45°C cabin air, interior mass temperature 56
Figure 3.19 Physical temperatures of cabin T_c and interior mass T_m in the cool-down process, trained with 1×10^4 learning episodes 56
Figure 3.20 TRPO and PPO HVAC policy warm-up performance 58

Figure 3.21 TRPO and PPO HVAC policy cool-down performance 58
Figure 4.1 Averaged episodic rewards with episode duration $T = 5 \times 10^3$ s and $T = 4 \times 10^3$ s against learning trials 64
Figure 4.2 Averaged episodic rewards with episode duration $T = 2.5 \times 10^3$ s and $T = 1 \times 10^3$ s against learning trials 65
Figure 4.3 Episodic comfort rate against learning trials with episode duration $T = \{1 \times 10^3, 2.5 \times 10^3, 4 \times 10^3, 5 \times 10^3$ s 66
Figure 4.4 Comfort testing performance by TRPO & PPO policies trained by 4×10^3 episodes with durations ranging from 1×10^3 s to 5×10^3 s; each case includes 4×10^3 random initial states 67
Figure 4.5 Power consumption testing performance by TRPO & PPO policies trained by 4×10^3 episodes with durations ranging from 1×10^3 to 5×10^3 s; each case includes 4×10^3 random initial states 67
Figure 4.6 Averaged episodic rewards against learning trials by MDP and NMDP state representation, with an averaged episode duration 4×10^3 s 68
Figure 4.7 Averaged episodic comfort rate against learning trials by MDP and NMDP state representation, with an averaged episode duration 4×10^3 s 69
Figure 4.8 Comfort testing performance by PPO policies trained by Markov and non-Markov represented state with total simulated training time $\{1.6 \times 10^7, 4 \times 10^7, 2 \times 10^7, 5 \times 10^7,$ secs 69
Figure 4.9 Power consumption testing performance by PPO policies trained by Markov and non-Markov represented state with total simulated training time $\{1.6 \times 10^7, 4 \times 10^7, 2 \times 10^7, 5 \times 10^7,$ secs 70
Figure 4.10 Warm-up process of occupant equivalent temperature by using PPO policies respectively trained by Markov and non-Markov state representation with 1.6×10^7 s, 2×10^7 s of simulated training time, starting from the environment, cabin air and mass temperature in 1°C 72

Figure 4.11 Cool-down process of occupant equivalent temperature by using PPO policies respectively trained by Markov and non-Markov state representation with 1.6×10^7 s, 2×10^7 s of simulated training time, starting from the environment temperature 40°C, cabin air and mass temperature 45°C 72
Figure 5.1 The averaged episodic rewards against 5×10^3 learning trials 85
Figure 5.2 Driver and passenger's mean body equivalent temperatures in the warm-up process 87
Figure 5.3 The 0 to 25 min of occupants' head ET in the warm-up process 88
Figure 5.4 The 0 to 25 min of occupants' torso equivalent temperatures in the warm-up process 88
Figure 5.5 Driver and passenger's feet equivalent temperatures in the warm-up process. This shows that the bang-bang method achieves the comfort very slow 89
Figure 5.6 The 0 to 25 min of the occupants' feet ET warm-up process in Figure 5.5 above 90
Figure 5.7 The 0 to 25 min of the occupants' mean body ET in the cool-down process (no solar radiation) 91
Figure 5.8 The 0 to 25 min of the occupants' head ET in the cool-down process (no solar radiation) 92
Figure 5.9 The 0 to 25 min of the occupants' torso ET in the cool-down process (no solar radiation) 92
Figure 5.10 Driver and passenger's feet ET in the cool-down process 93
Figure 5.11 The 0 to 25 min of the occupants' feet ET cool-down process in Figure 5.10 above 93
Figure 5.12 Driver and passenger's mean body ET in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) 95
Figure 5.13 The 0 to 25 min of driver and passenger's mean body ET cool-down process (solar radiation) in Figure 5.12 above 95
Figure 5.14 Driver and passenger's head equivalent temperatures in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) 96

Figure 5.15 Driver and passenger's torso equivalent temperatures in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) 97
Figure 5.16 The 0 to 25 min of driver and passenger's torso ET cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) in Figure 5.15 above 97
Figure 5.17 Driver and passenger's feet equivalent temperatures in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) 98
Figure 5.18 The 0 to 25 min of driver and passenger's feet ET cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) in Figure 5.17 above 98
Figure 5.19 Comfort rate of driver's body positions 101
Figure 5.20 Comfort rate of passenger's body positions 102
Figure 5.21 Neutral comfort rate of driver's body positions 102
Figure 5.22 Neutral comfort rate of passenger's body positions 103
Figure 5.23 The HVAC power consumption for maintaining occupants' body thermal comfort, the mean power by PPO-RL and bang-bang approximately equal 269.3 *W* and 554.7 *W* 104

LIST OF TABLES

Table 3.1 Model constants 38
Table 3.2 Meta parameters 40
Table 3.3 Comparison between applied RLs 49
Table 3.4 Average percentage of time spent in comfort and HVAC power consumption (with time duration of 1×10^3 s) of TRPO & PPO based HVAC control policies by different learning trials 59
Table 4.1 Averaged comfort percentage and power consumption of TRPO & PPO based HVAC control policies trained under different learning episode durations (1×10^3s to 5×10^3s) 66

Table 4.2 Averaged comfort percentage and power consumption of PPO based HVAC control policies respectively trained under Non-Markovian and Markovian representations with total simulated training time of 1.6×10^7 s, 4×10^7 s, 2×10^7 s, 5×10^7 s 71

Table 5.1 The HVAC control variables and corresponded discrete spaces (V.A.T refers to vent air temperature) 80

Table 5.2 The control variable settings for the double bang-bang controller (unit for I_b is Ampere A, T_{vdc} to T_{vp} is Celsius °C) 81

Table 5.3 Human body's thermal comfort zone constrained by Nilsson's equivalent temperatures (Summer and Winter season) [Nil04] 83

Table 5.4 The range of ambient temperature T_{env}, roof temperature T_r, ambient humidity ϕ_{env} and car velocity V_{car} values for warm-up/cool-down testing cases 84

Table 5.5 Time (min) taken to achieve neutral comfort zone and the percentage of time reduced due to PPO-RL compared with bang-bang controller in the warm-up process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator. 89

Table 5.6 Percentage of time in neutral comfort zone (during 0 to 160 min) and comfort improvement due to PPO-RL compared with bang-bang controller in the warm-up process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator. 90

Table 5.7 Time (min) taken to achieve neutral comfort zone and the percentage of time reduced due to PPO-RL compared with bang-bang controller in the cool-down (no solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator. 94

Table 5.8 Percentage of time in neutral comfort zone (during 0 to 160 min) and comfort improvement due to PPO-RL compared with bang-bang controller in the cool-down (no solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator. 94

Table 5.9 Time (min) taken to achieve neutral comfort zone and the percentage of time reduced due to PPO-RL compared with bang-bang controller in the cool-down (with solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator. 99

Table 5.10 Percentage of time in neutral comfort zone (during 0 to 160 min) and comfort improvement due to PPO-RL compared with bang-bang controller in the cool-down (with solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator. 99

Table 5.11 Percentage of time that the PPO-RL and bang-bang controller can maintain occupants in the thermal comfort zone during 0 to 83.4 min. These results are averaged over 5×10^3 random initial states used in the learning trials. 101

Table 5.12 Percentage of time that the PPO-RL and bang-bang controller can maintain occupants in the **neutral thermal comfort** zone during 0 to 83.4 min. These results are averaged over 5×10^3 random initial states used in the learning trials. 103

ACRONYMS

HVAC	Heating, Ventilation, Air Conditioning
MRT	Mean Radiant temperature
ET	Equivalent Temperature
AT	Air Temperature
RL	Reinforcement Learning
MDP	Markov Decision Process
NMDP	Non-Markov Decision Process
SARSA	State Action Reward State Action
PGRL	Policy Gradient Reinforcement Learning
MCPG	Monte Carlo Policy Gradient
GAE	Generalized Advantage Estimation
A2C	Advantage Actor Critic
A3C	Asynchronous Advantage Actor Critic
SGD	Stochastic Gradient Descent
MAC	Mean Actor Critic
TRPO	Trust Region Policy Optimization
PPO	Proximal Policy Optimization
NPG	Natural Policy Gradient
ANN	Artificial Neural Network
MLP	Multi-Layer Perceptron
LSTM	Long-Short-Term Memory
RNN	Recurrent Neural Network
UAV	Unmanned Aerial Vehicle
KL	Kullback-Leibler
K-FAC	Kronecker-Factored Approximate Curvature
ACKTR	Actor Critic using Kronecker factored Trust Region
ACER	Actor Critic with Experience Replay
TRGPPO	Trust Region-Guided Proximal Policy Optimization

1 INTRODUCTION

1.1 RESEARCH AIM

The primary aim of a vehicle climate control system is to keep occupants comfortable while minimising the use of energy. A secondary aim is to ensure that windshield glass is free from condensation that might obscure the view. To do this, the system blows hot or cold air that is either recirculated or refreshed from the external area. This is ducted to vents positioned at the feet section, in the dashboard at middle height, and to defrost vents near the windshield. Other advanced functions may include controlling indoor air quality and humidity. This integrated system, comprising such air-thermal conditioning functions, is known as the Heating, Ventilation and Air-Conditioning (HVAC) system. The main objective of the HVAC system is to create a safe and comfortable indoor environment for humans [McD06], [Ene17]. Safety refers to controlling air quality, including airborne particles, oxygen and noxious gas levels, to guarantee the health of occupants within strict air conditioning standards [JAW07], [BAAB10]. Thermal comfort is a term specific to individual sensation of indoor space climate since different people tend to have different comfort perception in the same indoor environment for physical and psychological reasons [Tal+13]. Thermal comfort mainly concerns the exchange of heat between the human body and its environment. The individual's thermal comfort experience is related to body conditions, including gender, age, weight, indoor activity, clothing resistance and heat convection of skin, body mass, thermoregulation, psychological adaptation [RVL15]. Other dynamic environmental factors include air temperature, humidity, velocity, pressure and circulation [MR13], [DTN10]. These factors make it difficult to provide universal rules to define conditions of indoor thermal comfort [Cro+15].

Currently, most vehicle cabin thermostats are manually operated by the occupants when they are not satisfied with the air quality and thermal conditions of the cabin. For example, occupants need to adjust the settings of cabin temperature, blower speed and ventilation.

These manual operations might be distracting for the drivers when focusing on traffic or road conditions. It is essential that drivers not be distracted by trivial actions to ensure traffic safety. Despite the manual adjustments, the HVAC system may not provide the occupant's desired comfort under various circumstances. For example, the current commercial HVAC system merely inputs warm or cold air when the cabin air temperature is below the set-point, but cannot assess whether the user's preferred thermal condition is satisfied. Other environmental factors such as solar load, environment temperature, cabin humidity rate, air recirculation velocity also affect occupant's perceptions of thermal comfort. Therefore, it is important to identify the circumstances in which the occupant's comfort is or is not maintained. Some recent experiments regarding vehicle cabins have shown that it is realistic to formulate rational human-vehicle cabin heat exchange models [SSU16], [Lee+15], [MWL18] to model the relations between HVAC settings to occupant's thermal preferences.

Relevant researches by Hintea [Hin+13], [Hin14] and Fojitlin [Foj+17] have proposed using equivalent temperature [Nil04] to estimate occupant's thermal comfort regions based on cabin air and interior mass temperatures, blower speed, recirculation. This comfort model makes it possible to design an artificial agent that can identify and maintain thermal comfort for cabin occupants using the HVAC system. The intelligent vehicle HVAC control technique, for example, a fuzzy logic-based controller [Ibr+12] can automatically maintain a comfortable electric vehicle (EV) cabin temperature and relative humidity level, regardless of the time-varying surrounding climate. The challenge for EV climate control is that the use of air conditioning system can significantly reduce the driving range in winter climate [Zha+17]. Because the EV's HVAC system consumes large amount of electricity to pump the heated air into the cabin, and a positive temperature coefficient (PTC) resistance heater device is normally used to generate the heat for car cabin. Therefore, it is important to manage the energy efficiency of the EVs' air conditioning system [Qi14]. An energy management solution for sophisticated EV HVAC systems is based on a model predictive control (MPC) strategy to estimate power demands during normal driving in real-time [Eck+16]. Other practical approaches to HVAC control include machine learning methods of Predictive Mean Vote (PMV) [Fer+12], evolutionary, artificial neural networks [LD05] and reinforcement learning approaches [VCN19]. Therefore, the balance

between energy usage and cabin climate conditioning tends to be an essential issue for EV HVAC system control applications.

Among the climate control systems using machine learning, the reinforcement learning (RL)-based HVAC controllers have been applied in various indoor situations including car cabins [Bru+18], offices [Zha+18], classrooms [Val+19] and buildings [Yan+15]. These control tasks aim to create a comfortable environment by controlling for air temperature, carbon dioxide level, humidity and ventilation. Existing RL-based HVAC applications commonly employ traditional value-based approaches (such as the tabular action values, function approximation) to estimate HVAC control policies. However, the value-based approaches usually converge slowly to the optimal control policy [Dua+16]. Instead, recent RL approaches focus on using policy gradient-based reinforcement learning to accelerate the learning process to solve various real-world control tasks (primarily for model-free control) [Sch+17]. These include complex robotics locomotor skills, playing Atari video games, balancing pendulums, training a bipedal robot to walk [Tas+18], [Bro+16], [PS06]. Such cases indicate that policy gradient methods have potentials of dealing with real-world industrial problems, including the indoor thermal control tasks. Based on the car cabin RL HVAC system developed by Brusey [Bru+18], this work proposes to use policy gradient reinforcement learning (PGRL) as the primary approach to improving the performance of resulting HVAC controllers.

The main aim of this thesis is to develop a machine learning system with PGRL methods to train HVAC controllers that can maximally offer thermal comfort to passengers while maintaining the energy cost of the vehicle HVAC system. Meanwhile, the learning system can significantly reduce the number of simulation samples consumed in the training process.

The reinforcement learning (RL) process indicates the interactions between an agent and environment through trial and error steps: observing states from the environment, taking actions by consulting the policy, receiving rewards. The architecture of PGRL is to learn a control policy that decides actions the agent takes to achieve or maintain specific target states corresponding to maximum received rewards. Hence the policy updating rule is to adjust policy's behaviour in choosing the future actions that can maximize the rewards. However, the primitive PGRL approaches to update policy follows a gradient-descent rule which does not always improve the policy. A state-of-the-art approach, namely the trust region policy optimization (TRPO) [Sch16] introduces

a constrained learning step to ensure positive update for policy. Therefore, it is useful to employ primitive and TRPO-based PGRLs to train HVAC control policies, and compare their performance in satisfying the set point of thermal comfort and energy consumption.

In reinforcement learning, the way that an agent observes states and selects actions, namely the Markov decision process (MDP). In this process, the effects of actions depend only upon the current state; this also means that the knowledge of the current state is sufficient for optimal control [SB18]. Based on the vehicle thermal model by Brusey [Bru+18], this work has defined information of cabin state and agent actions, but the state is represented in a non-Markov decision process (NMDP). Therefore, this vehicle thermal control model has potentials to investigate what improvements the MDP-based state representation can bring to the PGRL HVAC learning system, and how it impacts the comfort maintaining and energy consumption performance of the learnt HVAC control policies.

The first challenge is choosing effective policy gradient reinforcement learning (PGRL) algorithms to consume less simulated samples on training vehicle HVAC controller that can effectively maintain comfort, cool down and warm up the cabin. The second challenge is to consider the Markov property in a PGRL framework and examine whether it improves the comfort maintaining and energy consumption performance of the resulting HVAC controller with the same amount of training samples.

1.2 RESEARCH QUESTIONS

This thesis focuses on subsequent sets of research questions as responses to the aims mentioned above:

> **RQ 1.1** Can the vehicle HVAC agent, trained by PGRL schemes, reduce the time taken to achieve occupant thermal comfort and keep reasonable energy consumed by the HVAC system compared to the SARSA based learning scheme?
>
> **RQ 1.2** Can the PGRL HVAC training scheme learn an optimal control policy within a reasonable number of training samples?
>
> The HVAC system is vital in creating a safe and comfortable cabin environment for the occupants. An intelligent HVAC controller is able to sense the cabin state, including cabin temperatures, vent airflow and decide which set of control actions to take.

This intelligent control agent can also effectively offer desired thermal comfort to occupants while keeping energy-efficiency of the HVAC system. This work proposes using policy gradient reinforcement learning (PGRL) to find an optimal control policy that can maximally achieve occupant thermal comfort while keeping energy efficiency compared with earlier work done by SARSA-based RL [Bru+18] (for RQ 1.1). Since some state-of-the-art RL-based HVAC control tasks [Pet18], [Bru+18], [Val+19] spend 2.9 to 10 years of simulated time on training the controllers, hence less number of the learning experience is expected in this PGRL HVAC system compared to these works (for RQ 1.2). This term "simulated time" denotes the duration from the point of view of the simulated environment. This term's value corresponds to the number of state-action sequences observed multiplied by the size of the simulation time step. For example, training an intelligent agent for the cart-pole balancing task in Figure 2.2 need to sample sequences of time-enumerated states. So, the simulator needs to imitate the physical dynamics of the cart-pole model per time step, namely the changes of cart velocity, pole angular velocity over a timescale. The cumulated timescales of these generated samples refer to the simulated time, usually not equal to real-world time spent on the simulation. Another example shows the terminology of real-world time scenario in training Go game agent [Sil+17], which takes 40 days of real-world time rather than simulated time. In this thesis, the PGRL case takes approximately 40 to 60 minutes of real-world time for the HVAC agent training processes equivalent to two years of simulated time.

Chapter 3 presents the learning results of applied PGRL solutions, and energy, comfort performance by resulting control policies. Section 3.3 answers RQ 1.1 by comparing the performance of resulting HVAC controllers estimated by PGRL methods in terms of 1) averaged percentage of time spent on occupant comfort 2) the time taken to respectively achieve and maintain occupant thermal comfort in cool-down and warm-up processes 3) energy consumption of HVAC system. The answer for RQ 2.2 focuses on using less simulated time of training samples to learn the resulting controllers because the simulated time need to be kept within the vehicle's lifetime.

The following sets of research questions focus on investigating what impacts the Markov property can have on the learning capability of the proposed PGRL HVAC system.

> **RQ 2.1** Is the learning performance of PGRL HVAC negatively impacted by a non-Markovian cabin state representation?
>
> **RQ 2.2** Can the Markovian-represented cabin state improve the energy efficiency by using the same number of training experience in a non-Markovian state representation?
>
> The Markov decision process (MDP) suggests a typical agent-environment interaction model in reinforcement learning (RL) framework. Still, some real-world problems do not always obey the MDP rules (Markov property) due to a lack of state information [WL95]. In traditional RL approaches (for example Q-learning), the Markov property is essential in determining what state information the agent can use to decide control actions. Section 4.1 of Chapter 4 answers RQ 2.1 by showing the presence of non-Markov property in state representation and how it impacts the learning capability of PGRL HVAC system. Section 4.2 answers RQ 2.2 by comparing the comfort and energy consumptions performed by non-Markov and Markov based policies. This section shows that the fixed ending time results in non-Markov represented cabin states and proposes an MDP-represented states collection process for PGRL HVAC system

Then the last research question focus on the PGRL application in an occupant-oriented car cabin thermal simulation model approximated by the climatic wind tunnel dataset. This question can better investigate the PGRL performance in realistic thermal comfort conditioning.

> **RQ 3** Can the PGRL-based HVAC controller reduce the time taken and power consumption to achieve occupant thermal comfort in a climatic wind tunnel simulation model compared to a bang-bang method?

1.3 CONTRIBUTION TO KNOWLEDGE

There are two main contributions

> 1. This thesis presents a policy gradient reinforcement learning (PGRL)-based comfort-oriented, energy efficient, heating and

cooling HVAC controller that outperforms existing RL-based vehicle HVAC controllers. The performance is measured here in terms of maximizing the proportion of time spent in comfort for the occupant while minimizing the energy consumed by the HVAC system. The training samples spent on learning this controller are significantly reduced by $70-90\%$ compared to earlier work based on SARSA RL scheme. This controller is a multilayer perceptron (MLP)-based neural network that can decide control actions according to the cabin thermal state it observes.

2. A method to represent cabin environment state in a way that fulfils Markov decision process, so that it helps to improve the learning performance of PGRL HVAC system. The resulting policies yield competitive performance in achieving occupant comfort and energy efficiency compared with non-Markovian state representation cases.

1.4 THESIS STRUCTURE

Rest part of this thesis is listed as follows:

Chapter 2 reviews classic reinforcement learning approaches, the background of policy gradient methods with corresponded applications, and existing reinforcement learning solutions for thermal conditioning. Also, four typical policy gradient methods being applied to the vehicle HVAC control problem: Monte-Carlo policy gradient (MCPG), Mean actor critic (MAC), Trust region policy optimization (TRPO) and Proximal policy optimization (PPO).

Chapter 3 describes the vehicle HVAC control model, combined with proposed PGRL methods. This chapter presents the learning capabilities of applied PGRL methods in achieving optimal control policies, results of testing cases by the estimated controllers, and evaluations of control performance in extreme cold and hot weather conditions. (contribution 1)

Chapter 4 discusses the Non-Markovian state representation in the PGRL HVAC learning system through analysis and experiments. Compares the capabilities of Non-Markovian and Markovian modelled PGRL HVAC systems in learning optimal

policies and rewards maximizations. Testing results by Markovian and Non-Markovian PGRL HVAC agents are compared. (contribution 2)

Chapter 5 discusses the PGRL application in an HVAC system for a car cabin thermal model based on climatic wind tunnel dataset. This chapter also presents comparison results between RL-based and bang-bang HVAC controllers. And the model focus on simulations of occupant's thermal status. (supplementary to contribution 1)

Chapter 6 presents answers to the research questions, conclusions from presented researches and experiments, suggests potential research directions for future works

2

BACKGROUND OF REINFORCEMENT LEARNING METHODS AND APPLICATIONS TO INDOOR ENVIRONMENT THERMAL CONDITIONING

This chapter mainly reviews the research materials related to reinforcement learning (RL) fundamentals and state-of-the-art RL algorithms and their applications on indoor environment thermal conditioning controls. The first three sections start with introducing the Markov Decision Process (MDP) as the basic model for RL framework. Specifically, reviewing the fundamental value-based and policy gradient-based RL methods regarding to solving problems in continuous domains, then specifying the state-of-the-art policy gradient-based RL approaches and their advantages in solving realistic control problems (such as locomotion). The last section of this chapter mainly covers state-of-the-art RL applications in relevant heating, ventilation and air conditioning (HVAC) control tasks for various indoor environments. By the end of this chapter, the potential benefits of choosing the policy-based RL as the solutions for vehicle HVAC control are identified as the basic aspects of research questions.

2.1 MARKOV DECISION PROCESS

Markov Decision Process (MDP) [SB18] is a mathematical framework used for modelling decision making under certain environmental rules. An MDP can be represented by a tuple consisting of states $\mathcal{S}$, actions $\mathcal{A}$, transition probabilities $\mathcal{P}$, rewards $\mathcal{R}$ and discount factor γ, and the definition of an MDP is given by

- $\mathcal{S}$: set of observable environment states
- $\mathcal{A}$: set of possible actions from which the agent choose an action for each time step
- $\mathcal{P}$: state transition distribution $p(s_{t+1} \mid s_t, a_t)$ determines next state s_{t+1} resulted from current state s_t and action a_t
- $\mathcal{R}$: immediate reward $R(s_t, a_t)$ received from current state and action s_t, a_t

- $\gamma \in [0, 1]$: the discount factor which represents the weight between current reward and future reward

The MDP events start from an initial state $s_0 \in \mathcal{S}$ drawn from a specific environment; the agent observes a state $s_t \in \mathcal{S}$ after time t. Then the agent selects an action $a_t \in \mathcal{A}$ and executes it. This process results in a state transition over a unit time step, resulting in observation s_{t+1}, receiving an immediate real-valued reward $R(s_t, a_t)$. State transition probability $p(s_{t+1} \mid s_t, a_t)$ governs the likelihood of receiving s_{t+1} as a result of executing action a_t on s_t. By repeatedly observing state, selecting and executing actions, the agent receives a sequence of states, actions and rewards $s_0, a_0, r_0, s_1, a_1, r_1, \cdots$. The sum of received discounted rewards are

$$R(s_0, a_0) + \gamma \cdot R(s_1, a_1) + \gamma^2 \cdot R(s_2, a_2) + \cdots \tag{2.1}$$

where discount factor γ ($\gamma \leqslant 1$) acts as a weight balancing the importance of current and future rewards. Given an MDP environment, the goal of reinforcement learning is to find a function that selects actions $a_0, a_1, a_2, \cdots$ over time steps, thus maximizing the expected discounted sum of rewards in equation 2.1. The Markov property indicates the fact that the future observations of the MDP depend only on current observation, this means that the conditional probability of a future state $p(s_{t+1} \mid s_t, s_{t-1}, \cdots, s_0)$ conditioned on both the current state s_t and history $\{s_{t-1}, \cdots, s_0 \mid t > 2\}$ equals to $p(s_{t+1} \mid s_t)$ which is only conditioned on current observation.

Where the set of actions $\mathcal{A}$ and state observations $\mathcal{S}$ can be either finite or infinite for different cases. An example of finite discrete MDP is the cliff walk game [SB18] shown in Figure 2.1, a robot is assigned to find the path from initial cell S to the target G without falling off the cliff. This robot can move in four directions (up/right/down/left) in each cell while either receiving reward -1 for each step or -100 for falling off the cliff. The maximized reward reflects two possible paths in the figure: a safe path and an optimal route. It is not difficult to figure out that each future state s' on the cliff walk grid depends on current observation s. Figure 2.2 shows a classical cart-pole balancing benchmark that can be used as a continuous observation MDP model [Dua+16]. A pole is attached to a cart by an unactuated joint which allows it to swing freely. The cart can move along a horizontal frictionless track by applying a force with left/right directions and certain magnitude. The state of this system comprises four values: the position of the cart relative to the centre of the track x, the velocity of

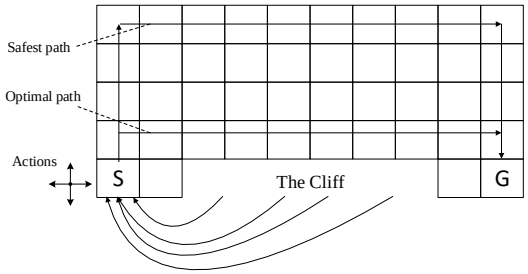

Figure 2.1: A cliff walk game

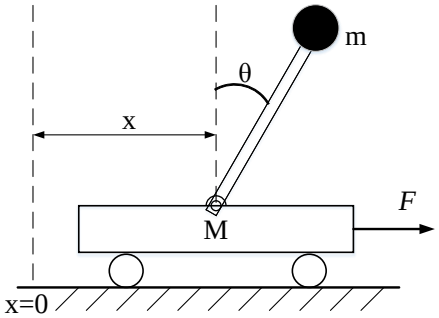

Figure 2.2: A cart-pole balancing task

cart $\dot{x}$, the angle of the pole θ and angular velocity $\dot{\theta}$. Given an initial state $s_0 = \{x_0, \dot{x}_0, \theta_0, \dot{\theta}_0\}$ when the pole is upright, the subsequent controlling goal is to prevent it from falling over. The physical model of deducing acceleration $\ddot{x}$, $\ddot{\theta}$ with respect to horizontal force and other parameters in Figure 2.2 can be specified in simple polynomials [LYB07]. Also, the update of the observed state depends on the instant velocity $\dot{x}$, acceleration of velocity $\ddot{x}$, angular velocity $\dot{\theta}$ and acceleration of angular velocity $\ddot{\theta}$ over the unit time-step Δt. Based on a state transition process [LYB07], the next-step state $\{\hat{x}, \hat{\dot{x}}, \hat{\theta}, \hat{\dot{\theta}}\}$ is only conditioned on current state $\{x, \dot{x}, \theta, \dot{\theta}\}$ and force F. But in real cases, the pole joint and the cart wheels can slip, thus not always resulting in the exact future state $\hat{s}$ as the update equation does. Hence the stochastic behaviour of this system can be modelled as state transition probability $p(\hat{s}_0, a_0, s_0)$ where a_0 =left/right F. Consider the effect of executing certain action in each transition step, the conditional probability of observing the future state s_{t+1} only depends on current state s_t and action a_t. The distribution of observing the future state for current observation and action:

$$p(s_{t+1}, a_t, s_t) = p(s_{t+1} \mid a_t, s_t) \cdot p(a_t \mid s_t) \cdot p(s_t) \tag{2.2}$$

where the conditional probability $p(a_t \mid s_t)$ can be referred to as a mapping from state to action $\mathcal{S} \to \mathcal{A}$. This mapping is also known as a policy function $\pi : \mathcal{S} \to \mathcal{A}$, which is considered to be the final learning target. There are two common policy models. One is named as stochastic policy which samples action according to a distribution over a given state: $a \sim \pi(a \mid s)$. Another is deterministic policy in which action a equals to policy $\pi(s)$. The following sections will mostly discuss the role of a policy inside the value-based and policy-based reinforcement learning.

2.2 REINFORCEMENT LEARNING WITH VALUE-BASED APPROACHES

Markov Decision Process (MDP) is introduced as a framework of achieving a goal by interaction with the environment. The decision-maker is called the agent that interacts with the environment, and these interactions comprise processes of selecting actions, observing environment's responding to these actions and presenting new situations to the agent. By doing such interactions with the environment, the agent is seeking opportunities to maximize received rewards through its choice of action over time.

The previous section has indicated that these interactions can be sampled into a sequence or trajectory over time steps: $s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_t, a_t, r_t, \cdots$. The simplest form of agent-environment interaction is the finite MDP, in which the sets of states $\mathcal{S}$, actions $\mathcal{A}$ and rewards $\mathcal{R}$ have finite elements. The last section has also indicated the distribution of receiving next state as $p(s_{t+1} \mid s_t, a_t)$. Consider the reward r_{t+1} received after executing action a_t on state s_t, the probability of receiving consecutive state and reward refers to $p(s_{t+1}, r_{t+1} \mid s_t, a_t)$. A general form of defining such transition probability is

$$p(s', r \mid s, a) = \Pr\left\{s_{t+1} = s', r_{t+1} = r \mid s_t = s, a_t = a \right. \tag{2.3}$$

where $s, s' \in \mathcal{S}$, $r \in \mathcal{R}$ and $a \in \mathcal{A}$. As the choice of action a and state s is in a finite space, the summation of all possible transition probabilities must follow the rule

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1 \tag{2.4}$$

which indicates the dynamics of finite MDP environment. With these four arguments, it is easy to characterize the state-action ($Q(s, a)$) or state value ($V(s)$) functions with respect to the received rewards. These two functions are useful in evaluating how good current policy is, and assist the RL scheme in improving the policy further.

2.2.1 *Value functions and policies*

Most reinforcement learning algorithms estimate a value function [SB18], a value function is a value of states or state-action pair that can estimate how much benefit the agent can have when reaching a certain state or execute a specific action inside a given state. This benefit is relevant to the rewards that the agent can expect to receive in the future when taking specific actions in a given state. More precisely, the value function is related to the policy which decides the actions. As mentioned above, a policy maps state to the probability of selecting possible action from a given action space $\mathcal{A}$ according to the distribution of actions $a \sim \pi(a \mid s)$ where $a \in \mathcal{A}$ for each $s \in \mathcal{S}$. The value $v^{\pi}(s)$ of a given state s under a policy π is the expected rewards following policy π from state s,

$$v^{\pi}(s) = \mathbb{E}_{s,a,r\sim\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right] \text{for all } s \in \mathcal{S} \tag{2.5}$$

where the agent samples the expected return of rewards with a specific policy π at any time step t, and v_{π} denotes state-value function for policy π. Consider the value of executing action a in state s under the policy π, we can derive a similar equation of expected rewards starting from given state s, taking action a with respect to policy π:

$$q^{\pi}(s, a) = \mathbb{E}_{a,s,r\sim\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right] \tag{2.6}$$

where q_{π} is called action-value function for policy π. Both $v^{\pi}(s)$ and $q^{\pi}(s)$ can be estimated from the agent's experience of observing states, performing actions and receiving rewards. In order to get the highest rewards for state, action values, the scenario of optimal policy is introduced. For example, if a policy π can lead to expected returns greater than or equal to the rest ones, then policy π is the

optimal policy marked as π^*. Therefore the optimal state, action-value functions are induced as follows

$$v^*(s) = \max_{\pi} [v^{\pi}(s)] \qquad q^*(s) = \max_{\pi} [q^{\pi}(s, a)] \tag{2.7}$$

The bellman equation [Bel66] shows the recursive optimality equation for state action values with respect to transition probability $p(s', r \mid s, a)$

$$v^*(s) = \max_{a} \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v^*(s') \right] \tag{2.8}$$

$$q^*(s, a) = \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma \max_{a'} q^*(s', a') \right] \tag{2.9}$$

As most RL cases account for the impact of executing action as part of the reward function, estimating the action-value function $q^*(s, a)$ is an essential procedure for both value-based and policy gradient-based RLs. The subsequent section mainly covers traditional value-based RL approaches.

2.2.2 *Classical learning algorithms*

There exist two main categories of learning action value functions: on-policy and off-policy. A typical on-policy example is named "State Action Reward State Action" (SARSA), which is essentially estimating $q^{\pi}(s, a)$ for current policy π, with respect to the state-action pairs $\{s, a\}$. This algorithm uses policy π, state-action value $Q(S, A)$ to derive action a for current state s and execute it, then receiving reward r and next state S', next updating current state-action value $Q(s, a)$ by

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma Q(s', a') - Q(s, a) \right] \tag{2.10}$$

where α is a learning rate. Algorithm 3 includes detail of SARSA where ϵ-greedy method allows randomly selected actions to keep minimal explorations.

A typical off-policy RL algorithm is named as Q-learning [WD92] which uses current policy π and state-value $Q(s, a)$ to derive action a for current state s, then executing it to receive next state s', finding the

maximum next state-action value $Q(s', a')$ and updating Q function as follows:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \Bigg(\tag{2.11}$$

An alternative method Expected SARSA [VS+09] calculates the expected action value of the next state rather than simply finding the maximum next state-action pairs $\max_{a'} Q(s', a')$. Hence the formula is slightly changed into such form

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \sum_{a' \in \mathbb{A}} \pi(a' \mid s') Q(s', a') - Q(s, a) \right] \Bigg(\tag{2.12}$$

where this algorithm takes into account the likelihood of choosing each action is under current policy, thus eliminating the variance due to random selection of action a'. In the appendix, Algorithm 4 lists details of Expected SARSA where the ϵ-greedy method allows the probability ϵ to randomly select actions from the given action space to enable exploration in finding potential optimal policies. For example, probability $1 - \epsilon$ for choosing greedy action $a = \arg\max_{a \in \mathbb{A}} Q(s, a)$ and probability ϵ for choosing random action a. However, these algorithms are limited to MDP cases with finite states and actions space. For this reason, state-action values $Q(s, a)$ essentially consists a table of Q-function values. Given current state s and action a, both on-policy and off-policy learning approaches require the action-value function of next state s' to update current state-action value $Q(s, a)$. Figure 2.3 illustrates the process of updating the state-action value in terms of discrete action and state spaces. It is understood that this finite state-action strategy can not be used to solve continuous observation problems directly. As for this issue, an empirical solution is to approximate state value functions ($V(s)$ or $Q(s, a)$) from experience generated by a known policy π.

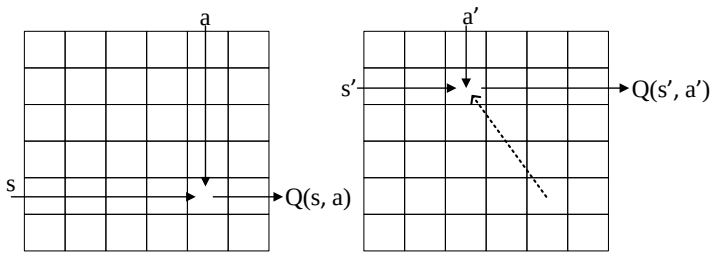

Figure 2.3: Tabular Q-function upgrade

2.2.3 *Value function approximation methods*

The significant advantage of using approximation methods is that the approximate state action value is not represented by a table but as a parameterized function with specific weight vector from real value domain. This application helps to improve the classic SARSA or Q-learning algorithms to deal with continuous state problems in real-world applications [BD95], [Doy00], [GWZ99], [MPD02]. In this case, we can write $\hat{v}_{\boldsymbol{w}}(s)$ as the approximate value of state s with weight vector $\boldsymbol{w}$, hence letting $\hat{v}_{\boldsymbol{w}}(s) \approx v^{\pi}(s)$. Where $\hat{v}_{\boldsymbol{w}}$ can be linear function in features of state s weighted by vector $\boldsymbol{w}$, or computed by a multi-layered neural network with connection weight $\boldsymbol{w}$ of all the layers. Sutton's chapter describes the approximation models regarding to linear and non-linear categories [SB18]. A general way is to minimize the value error between $\hat{v}_{\boldsymbol{w}}(s)$ and $v^{\pi}(s)$ below

$$\mathrm{VE}(\boldsymbol{w}) = \sum_{s \in \mathcal{S}} [v^{\pi}(s) - \hat{v}_{\boldsymbol{w}}(s)]^2 \tag{2.13}$$

An ideal goal of minimizing $\mathrm{VE}(\boldsymbol{w})$ is finding a global optimum weight vector $\boldsymbol{w}^*$ yielding the smallest error $\mathrm{VE}(\boldsymbol{w}^*)$ against all possible weight $\boldsymbol{w}$. This process, known as value prediction, is generally implemented by stochastic gradient descent (SGD) methods. Given weight vector with a fixed number of real-valued elements $\boldsymbol{w} = [w_1, w_2, \cdots, w_d]^{\mathsf{T}}$, and a differentiable approximate function $\hat{v}_{\boldsymbol{w}}(s)$ with weight $\boldsymbol{w}$ and states $s \in \mathcal{S}$, we can iteratively compute the gradient update along with discrete time steps $t = 0, 1, 2, 3, \cdots$ so that the simple SGD based case is derived as follows

$$\begin{aligned} \boldsymbol{w}_{t+1} =& \boldsymbol{w}_t - \frac{1}{2}\alpha \nabla_{\boldsymbol{w}} [v^{\pi}(s) - \hat{v}_{\boldsymbol{w}}(s)]^2 \\ =& \boldsymbol{w}_t + \alpha [v^{\pi}(s) - \hat{v}_{\boldsymbol{w}}(s)] \nabla_{\boldsymbol{w}} \hat{v}_{\boldsymbol{w}}(s) \end{aligned} \tag{2.14}$$

where α ($\alpha \in [0, 1]$) is a positive step size. Similarly, the approximation of state-action value $\hat{Q}_{\omega}(s, a)$ weighted by vector ω, implicates an update of ω

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + \alpha \left[q^{\pi}(s, a) - \hat{Q}_{\omega}(s, a)\right] \nabla_{\omega} \hat{Q}_{\omega}(s, a) \tag{2.15}$$

where the approximation of state-action value $\hat{q}_{\boldsymbol{w}}(s, a)$ or state value $\hat{v}_{w}(s)$ can either be modelled by a linear equation or non-linear artifi-

cial neural networks (ANNs) [SB18]. The linear model approximates value function by inner product between weight and feature vector:

$$\hat{v}_{\boldsymbol{w}}(s) = \sum_{i=1}^{d} w_i x_i(s) \quad \text{or} \quad \hat{Q}_{\omega}(s,a) = \sum_{i=1}^{d} \omega_i x_i(s,a) \tag{2.16}$$

where $x(s)$ and $x(s,a)$ are $d-$dimensional feature values represented by certain basis functions. There exists several representative approaches to represent the feature functions, for example, the tile coding method [Sut96] and radial basis function [SSR97].

The ANNs with fully-connected deeply-layered architectures act as universal approximators [HSW89] that are useful in approximating non-linear multi-dimensional objectives. A practical deep Q-learning proposed by Mnih [Mni+13] has been used to learn the Atari video game strategies, a convolutional neural network [LBH15] has been applied to extract state information from each frame of the video, and a multilayer perceptron (MLP) feed-forward neural network $\hat{Q}_{\theta}(s,a)$ with weights θ is introduced to approximate state-action values. The process of updating weights θ is done by minimizing the difference between current state-action value $\hat{Q}_{\theta}(s,a)$ and target Y^Q. According to the Temporal difference (TD) learning [SB18], the next time-step reward r_{t+1} and optimal action-value $\max_a \hat{Q}_{\theta_t}(s_{t+1},a)$ forms the target .

$$\begin{aligned} \theta_{t+1} &= \theta_t + \alpha \left[Y_t^Q - \hat{Q}_{\theta_t}(s_t,a_t) \right] \nabla_{\theta_t} \hat{Q}_{\theta_t}(s_t,a_t) \\ \text{where} \quad Y_t^Q &= r_{t+1} + \gamma \max_a \hat{Q}_{\theta_t}(s_{t+1},a) \end{aligned} \tag{2.17}$$

The state-value function $\hat{v}_w(s)$ can be represented by a value network with weights w, thus using TD-learning to minimize error between current state value $\hat{v}_w$ and target Y^v

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t + \alpha \left[Y_t^v - \hat{v}_{\boldsymbol{w}_t}(s_t) \right] \nabla_{\boldsymbol{w}_t} \hat{v}_{\boldsymbol{w}_t}(s_t) \\ \text{where} \quad Y_t^v &= r_{t+1} + \gamma \hat{v}_{\boldsymbol{w}_t}(s_{t+1}) \end{aligned} \tag{2.18}$$

where the approximation can use dataset batches for each update of $\boldsymbol{w}$ and θ. The following section discusses a method that can directly learn and optimize the policy π instead of consulting value functions.

2.3 REINFORCEMENT LEARNING WITH POLICY GRADIENT APPROACHES

As mentioned above, the traditional reinforcement learning approaches are based on estimating state-action value $Q(s, a)$ (utility function) either in tabular or approximator form. An optimal policy is achieved by comparing action values $\pi^* : a \leftarrow \arg\max_{a \in \mathbb{A}}(s, a)$. A representative application such as Deep Q Network [Mni+13] has been successfully used in managing HVAC system inside buildings [WWZ17], [Val+19]. However, such value-based RLs fail in training robotics locomotion tasks [Sch+17] and has limitations in dealing with continuous action space, because the utility function usually deal with discrete action space. So this section mainly discusses a novel category of RL approaches named as policy gradient, this method directly learns the policy $\pi_\theta(\cdot \mid s)$ by maximizing received rewards. The following sections start by introducing the simple Monte-Carlo policy gradient (MCPG) as the prototype version, next the application of advantage function to reduce variance, then trust region policy optimization and proximal policy optimization as state-of-the-art policy gradient methods.

2.3.1 *The policy and trajectory*

Reinforcement learning can be generalized as an interaction between an intelligent agent and environment. And the history of such interactions can be modelled as sequences of states, actions and rewards. Figure 2.4 illustrates a fully-observable RL sequence, which includes state transitions, actions selected by policy, and relations with stochastic process. The sequence begins by observing an initial state s_0 from

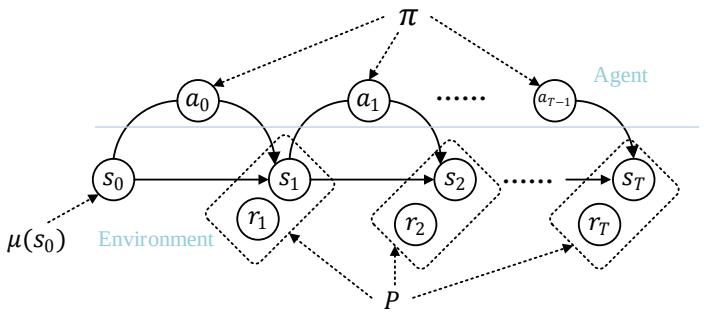

Figure 2.4: Policy inside the Reinforcement Learning

certain distribution $\mu(s_0)$ of the environment, the agent chooses an

action a_0 based on policy $\pi(a_0 \mid s_0)$. The policy π is a distribution used by the agent to sample actions. After executing the action a_0, the environment transits to the next state s_1 and receiving reward r_1. This process is subject to transition distribution $P(s_1, r_1 \mid s_0, a_0)$ due to potential interference from the environment. After that the agent observes s_1, samples a_1 from policy π, executes a_1 and receives s_2 this process can be generalized by equations

$$\begin{aligned} s_0 \sim \mu(s_0) \qquad a_0 \sim \pi(a_0 \mid s_0) \qquad s_1, r_1 \sim P(s_1, r_1 \mid s_0, a_0) \\ a_1 \sim \pi(a_1 \mid s_1) \qquad s_2, r_2 \sim P(s_2, r_2 \mid s_1, a_1) \quad \cdots\cdots \\ a_{T-1} \sim \pi(a_{T-1} \mid s_{T-1}) \qquad s_T, r_T \sim P(s_T, r_T \mid s_{T-1}, a_{T-1}) \end{aligned} \tag{2.19}$$

Where s_T denotes the terminal state that ends receiving next state, action, reward in certain RL cases, for example, the cart-pole balancing task is ended if the pole falls down, thus generating a trajectory (**episode**) of finite actions, states, rewards. Each trajectory begins with an initial state s_0 sampled from the distribution $\mu(s_0)$ of the environment. For each time step $t = 0, 1, 2, 3, \cdots$, the agent selects action a_t sampled from distribution given by stochastic policy $\pi(a_t \mid s_t)$. Then executing the action, the environment generates next state s_{t+1} and reward r_{t+1} (or $R(s_{t+1}, a_{t+1})$) according to state transition $P(s_{t+1}, r_{t+1} \mid s_t, a_t)$. The episode ends when reaching a terminal state s_T. Therefore, the agent receives a trajectory of states, rewards and actions $(\tau = s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_T)$ which corresponds to subsequent distribution

$$\tau \sim P_\pi(\tau) = \mu(s_0) \cdot \prod_{t=0}^{T-1} \pi(a_t \mid s_t) P(s_{t+1}, r_{t+1} \mid s_t, a_t) \tag{2.20}$$

The expected summation of rewards

$$E_\pi [R(\tau)] = E_\pi \left[\sum_{t=0}^{T-1} \gamma^t \cdot R(s_t, a_t) \right] \Bigg(\tag{2.21}$$

where $\gamma \in [0, 1]$ is a discount factor, T is the length of a specific episode, and the learning problem is to estimate a policy π^* which satisfies the condition of maximizing episodic rewards

$$\pi^* \leftarrow \arg\max_\pi E [R(\tau)] \tag{2.22}$$

where the expectation is taken over such trajectories and with notations of rewards summation. The stochastic policy model $\pi : \mathcal{S} \to \mathcal{A}$

is to produce a reliable policy π denoting a mapping from state to distributions of actions ($a \sim \pi(a \mid s)$). In this case, we use the parameterized stochastic policy π_θ specified by parameter $\theta \in \mathbb{R}^d$. For example, a neural network can represent the parameterized policy, θ corresponds to the weights and biases of the network. However, this policy network's parameterization depends on the type of action space in the MDP, such as continuous or discrete action space. In the discrete action space, we use a neural network that outputs action probabilities through a softmax output layer [ST09], and the continuous actions are sampled from a Gaussian distribution [SB18]. The following section mainly discusses how the policy parameter θ is updated for the maximization of rewards.

2.3.2 *A Monte Carlo-based formulation of policy gradient*

As mentioned above, the policy estimator from equation 2.22 and parameterized policy hypothesis, the episodic reinforcement learning problem becomes an optimization of policy parameter $\theta \in \mathbb{R}^d$

$$\theta \leftarrow \arg\max_\theta E_{\tau\sim\pi_\theta(\tau)} [R(\tau) \mid \pi_\theta] \tag{2.23}$$

A general way to implement this estimator is to repeatedly calculate the gradients of the policy's performance (resulting rewards) with respect to policy parameter θ. The goal of this optimization is to maximize the expected rewards $E_\pi [R(\tau)]$. A practical approach [BB+00] is to calculate the gradient of the expected rewards

$$\nabla_\theta \mathbb{E} [R(\tau)] = \mathbb{E} \left[\sum_{k=0}^{T-1} (\nabla_\theta \log \pi_\theta (a_k \mid s_k)) \cdot \sum_{t=k}^{T-1} \gamma^{t-k} \cdot R(s_t, a_t) \right] \Bigg(\tag{2.24}$$

Which is equivalent to the maximization of state-action value $Q(s, a)$ termed as

$$\nabla_\theta \mathbb{E} [R(\tau)] = \mathbb{E} \left[\sum_{k=0}^{T-1} (\nabla_\theta \log \pi_\theta (a_k \mid s_k)) \cdot Q^\pi(s_t, a_t) \right] \Bigg(\tag{2.25}$$

This function indicates the policy gradient approach with likelihood term, which optimizes the policy through maximizing value functions [Sut+00]. This form is also named as Monte-Carlo policy gradient (MCPG) since it uses full trajectory of state, action and reward samples.

However, the MCPG causes high variance in gradient estimation, this is likely to degrade the optimization of policy parameters. A primitive approach to mitigate the variance is to subtract a baseline function $b(s_t)$ [BB01] from the sampled cumulative rewards

$$b(s_t) = \mathbb{E}\left[r_t + \gamma \cdot r_{t+1} + \cdots + \gamma^{T-1-t} \cdot r_{T-1}\right]$$

where r_t equals to $r(s_t, a_t)$, hence the policy gradient with baseline function follows subsequent form

$$\begin{aligned}
&\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}\left[R(\tau)\right] \\
&\quad \approx \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{k=0}^{T-1}\left(\nabla_\theta \log \pi_\theta\left(a_k \mid s_k\right)\right) \cdot \left(\sum_{t=k}^{T-1} \gamma^{t-k} \cdot R(s_t, a_t) - b(s_t)\right)\right]
\end{aligned} \tag{2.26}$$

Also, Peters et al [PS06] and Riedmiller et al [RPS07] proposed another form of baseline function which computes $b(s_t)$ according to the expected cumulative gradients of policies over multiple episodes. The core of the learning process is to implement the following gradient descent iteration to update the policy

$$\theta_{n+1} = \theta_n + \alpha_n \cdot \nabla_\theta \mathbb{E}\left[R(\tau)\right]|_{\theta=\theta_n} \tag{2.27}$$

where $\alpha_n \in \mathbb{R}^+$ denotes learning rate and $n \in \mathbb{N}$ denotes the iteration. To formulate such gradients over cost function $\mathbb{E}\left[R(\tau)\right]$, we need to collect system's experience of states, actions and rewards. With a *stochastic policy*, we can generate probability distribution (written as $\pi(a \mid s)$) over actions. Since the policy contains parameter θ, we can specify this policy as $\pi_\theta(a \mid s)$. As cost function $\mathbb{E}\left[R(\tau)\right]$ comprises system's experience shortly named as trajectory $\tau = [\boldsymbol{s}_{0:T-1}, \boldsymbol{a}_{0:T-1}]$, this is generated with respect to policy $a_k \sim \pi_\theta(a_k \mid s_k)$ and state transition $p(s_{k+1} \mid s_k, a_k)$.

The Algorithm 5 in appendix details process of optimizing policy parameter θ through maximization of cumulative rewards. This MCPG approach has been widely applied in robotics motor primitive controls such as motor task planning and learning to hit a baseball for robotic arm [PS06], [PS08b]. Although MCPG is practical in such application cases, the drawback is that the cumulative rewards do not necessarily measure the positive policy update. For example, the maximization of state-action value $Q^\pi(s, a)$ does not always mean that the corresponded state value $V^\pi(s)$ is therefore improved by action a.

Therefore, the policy gradient method is increasing the probability of better-than-average actions (meaning these actions lead to higher rewards and state values) and decreasing the chance of getting worse-than-average actions. The next section focuses on using an advantage function $A(s, a)$ instead of solely using action-value function $Q(s, a)$ or state-value function $V(s)$.

2.3.3 *Application of advantage function*

By definition, the advantage function $A(s, a)$ is the difference of state-action value $Q^\pi(s, a)$ and state value $V^\pi(s)$

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s) \tag{2.28}$$

which indicates whether executing action a on state s is **better than the average**. Therefore, the policy gradient estimator with the application of advantage function has the following form

$$\nabla_\theta J = \mathbb{E}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) A^\pi(s_t, a_t)\right]\Bigg(\tag{2.29}$$

where the value of $A^\pi(s_t, a_t)$ is estimated via the TD-residual of the state-value function, let $\hat{A}_t$ be the estimate of $A^\pi(s_t, a_t)$, the sum of k of residual $\delta_t^V = r(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t)$ indicates the advantage estimator $\hat{A}_t^{(k)}$ according to Mnih et al. [Mni+16]

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = -V(s_t) + \gamma^k V(s_{t+k}) + \sum_{l=0}^{k-1} \gamma^l r(s_{t+l}, a_{t+l}) \tag{2.30}$$

as $k \to \infty$, the above term $\hat{A}_t^{(\infty)} \approx -V(s_t) + Q(s_t, a_t)$. Furthermore, the general advantage estimator [Sch+15b], is defined as exponentially-weighted average of k-step advantage estimator with $\lambda \in [0, 1]$

$$\hat{A}_t^{GAE(\gamma,\lambda)} = (1-\lambda)\left(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \cdots + \lambda^{k-1} \hat{A}_t^{(k)}\right)\Bigg(\tag{2.31}$$

where the application of λ makes good balance between variance and bias, so as to accurately estimate the value function $V(s_t)$. As mentioned in section 2.2.3, the value function approximation can be either minimizing difference between cumulative rewards and value estimator with N samples per learning batch or minimizing TD error with a batch of N samples [Sch16]. This policy gradient with advantage

function is named as advantage actor critic (A2C) [Mni+16], and the learning process is listed in Algorithm 6 at the appendix part.

2.3.4 *Mean Actor Critic method*

The section above estimates advantage function $A(s,a) = Q^{\pi}(s,a) - V(s)$ by approximating the state value $\hat{V}^{\pi}(s)$, this learning form follows an Actor-Critic architecture which has two separate parts: learning the policy $\pi_{\theta}()$ and approximating state-action values $Q(s,a)$. However the empirical way of estimating the advantage function is to use a TD-error $(r + \gamma \cdot V(s') - V(s))$ and this method has relatively high variance due to the dynamics of the environment. This section focuses on using state-action value estimate $\hat{Q}_{\theta}(s,a)$ to approximate advantage function instead of sampling from TD-error of value functions. The Mean Actor Critic (MAC) [Asa+17] method provides an advantage estimation solution, which only needs the state-value function $Q^{\pi}(s,a)$

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s\sim\pi}\left[\nabla_{\theta} \sum_{a\in\mathcal{A}} \pi_{\theta}(a \mid s) Q^{\pi}(s,a)\right]\Bigg(\tag{2.32}$$

In which, this method computes the policy-weighted average over all Q-values rather than only using the sampled states and actions, thus significantly reducing the variance caused by the stochastic policy. The explicit estimation of state-action value $Q^{\pi}(s,a)$ is based on the Expected SARSA [VS+09] to iteratively calculate the TD-error of $Q^{\pi}(s,a)$. Hence, the detail of MAC is listed in Algorithm 7 in appendix section.

2.3.5 *Trust region policy optimization and Proximal policy optimization approaches*

The empirical policy gradient approaches (MCPG, A2C, MAC) estimate policy parameters by maximizing the sample of rewards or state-action values episodically. The stochastic gradient descent (SGD) is a fundamental learning framework utilized to compute the policy gradient. This upgrade is usually scaled by a fixed learning-step times the first-order gradients of policy parameters. However, the expected rewards of a trajectory $\mathbb{E}_{\tau\sim\pi}[R(\tau \mid \pi_{\theta})]$ usually present a non-convex property such as multiple stationary points [She+19]. In such circumstances, the empirical first-order gradient method does not guarantee

convergence to the globally optimal policy [BR19]. Specifically, if the learning step size is too large, the overestimated policy is likely to result in negative rewards update and cause degradations within the learning trials; otherwise, a tiny step size usually slows down the learning speed. Therefore, it is essential to "properly adjust" the learning step size to keep a non-negative policy update along the non-convex curvature of expected return $\mathbb{E}_{\tau\sim\pi}[R(\tau \mid \pi_\theta)]$.

Therefore, two state-of-the-art optimization approaches, namely the trust region policy optimization (TRPO) [Sch+15a], [Sch16] and the proximal policy optimization (PPO) [Sch+17] are introduced as the main PGRL approaches. The TRPO method firstly uses Kullback-Leibler (KL) divergence [Joy11] to approximate the lower boundary of policy update, secondly derives the constrained optimization process in terms of natural gradient framework [Ama98]. Alternatively, the PPO method performs multiple epochs of policy gradient update per data sample (batches of states, actions and rewards being used to estimate policy π_θ and advantage function $A(s, a)$) and uses clipped ratio probability to adjust the learning step size. Moreover, PPO is compatible with the first-order optimization such as the Adam optimizer [KB14], hence being more straightforward than the TRPO method.

The calculation details for TRPO are listed in Algorithm 8 to Algorithm 9 in the appendix section. Also, the PPO method with its hyperparameter settings are listed in Algorithm 10 in the appendix section.

2.3.6 *The representative policy gradient reinforcement learning applications*

In terms of reinforcement learning application tasks, the robotic locomotions [KBP13] are the most challenging categories due to complicated movement dynamics, high-dimensionality of action and observation space. For example, the bipedal walking and robotic locomotion control tasks, including training a humanoid to stand up and walk [Tas+18], which are simulations close to real-world problems. Also, the policy gradient approaches have been proved to solve locomotion control tasks, such as the half-cheetah problem [Waw09] employs the natural actor-critic method [PVS03], [PS08a] as the learning scheme. The first three locomotion control modules in Figure 2.5 are among the typical RL benchmark models by OpenAI [Bro+16]. For example, the humanoid task is to have optimal control of the torques, velocity

and actuator forces of 21 joints of this humanoid robot. Usually, the value-based RL method, such as the Deep Q Network (DQN) [Mni+13] is incapable of learning optimal locomotor skills according to Duan's locomotion control simulation tasks [Dua+16]. However, the recently

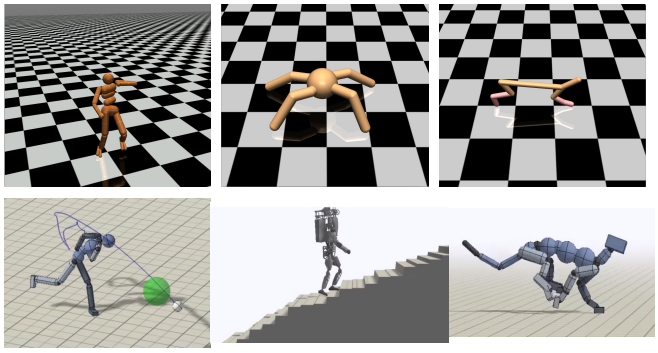

Figure 2.5: Simulations of robotics locomotion control: "Humanoid", "Ant walk", "Half-cheetah running", "Lion running", "Atlas robot walking upstairs" and "Ball-throw task" (clockwise-listed) via Policy gradient reinforcement learning,

developed TRPO and PPO methods perform very well in doing such locomotion control tasks; additionally, Kurutach et al [Kur+18] has proposed a model-ensemble TRPO method to reduce the sample complexity. Peng's work [Pen+18] includes PPO application in training an intelligent multi-task agent that can deal with various robotics dynamics skills, such as a human character throwing a ball to the random targets, Atlas robot walking upstairs, and lion running in Figure 2.5. additionally, this work presented a multi-skill policy that can perform corresponded skills for different robotic tasks.

Another concurrent PPO application is for a wheel-legged robot control task by Chen et al. [Che+18]. This task aims to train an agent to control a wheel-legged robot to target position without colliding with obstacles. The agent is equipped with convolutional networks to extract height information of barriers and decides corresponded actions, such as stretching legs to drive over a short but wide obstacle, lifting the body to drive over a high but narrow obstacle, moving around to avoid a big obstacle. The reward definition shows that the robot is trained to reach the target by using less time and reducing invalid actions. Also, environment randomization is applied to improve the agent's ability to observe task-relevant components under different environments. The result has shown that the PPO method improves the success rate by using the same batch of samples, hence improving the data efficiency.

The RL methods have been used in controlling aircraft, for example, an autonomous helicopter control by Abbeel et al. [Abb+07] uses differential dynamic programming to perform an aerobatic manoeuvre, and Ng [Ng03] has implemented a policy search combined with reward shaping to learn the control strategies of a helicopter. The following works of literature concern the policy gradient applications in aircraft control.

Koch et al. [Koc+19] uses DDPG [Lil+15], TRPO, and PPO approaches to train a quadcopter attitude controller. A Proportional, Integral Derivative (PID) controller is used as a comparison. This quadcopter can fly and move through all three dimensions by adjusting the four motors' rotation speed. This task aims to control the quadcopter to accurately reach a setpoint as soon as possible and keep a small and stable acceleration of angular velocity. The result shows that PPO based learning outperforms TRPO, DDPG and PID in terms of success rate, rising time and stability. Another attitude control of fixed-wing Unmanned aerial vehicle (UAV) by Bohn et al. [Bøh+19] also proposed using PPO as the representing learning method. This task aims to train a PPO-based agent to control this aircraft's aileron and elevator deflection angles, throttle force to maintain desired airspeed, pitch and roll angles. The controller's performance is examined by comparing the success rate of reaching a set-point (bounds of desired airspeed, pitch and roll angles), time spent on reaching and staying in the set-point bounds, and control variations. The results show that the PPO-based controller achieves a higher success rate than PID control and performs competitively in the speed of reaching set-point. Still, the PID controller has lower control variation below severe turbulence.

Lopes et al. [Lop+18] has employed the PPO approach to training a quadcopter controller based on a robot virtual simulation environment. This task includes training an intelligent agent that can control the signal to adjust the propeller thrust force to fly the drone to the target position. The quadcopter equipped with the resulting policy has shown its capability to reach a fixed target position, recovering from a harsh initial position (when the propeller is vertical to the ground) and tracking a moving target.

2.4 REINFORCEMENT LEARNING APPLICATIONS FOR INDOOR THERMAL CONDITIONING

As reinforcement learning (RL) methods serve as popular solutions for indoor Heating, Ventilation and Air Conditioning (HVAC) control, most of the existing application cases mainly focus on using value-based approaches (such as Q-learning, SARSA-λ) to estimate building climate controllers, while several state-of-the-art studies have proposed using PGRLs.

Li et al. [LX15] proposed a multi-grid method that combined both coarse and fine discretizations to accelerate Q-learning convergence in an HVAC control policy optimization problem. The result shows a faster learning speed of multi-grid Q-learning, and the resulting HVAC controller can reduce the energy cost and improve comfort satisfaction in the first 12 weeks of learning time.

Wei et al. [WWZ17] employed a deep neural network-based Q-learning [Mni+13] approach to training the agent for controlling the air velocity of a building HVAC system. The goal of using this RL-based control is to keep air temperature within the desired range and reduce electricity cost. This experiment is conducted on three simulated multi-zone building environments (on the *EnergyPlus* software platform), and the agent is trained with 100 months (8.3 years) of simulated data. According to the evaluation of trained agent, this deep reinforcement learning application can maintain the percentages of desired temperatures above 98% for the tested multi-zone building environments and significantly reduce electricity cost from 19% to 52% of the corresponded baseline cases. Based on the same simulation platform, Valladares et al. [Val+19] improved the learning architecture by using double Q-learning [VHGS16] approaches to training an intelligent agent for controlling air-conditioning units and ventilation fans. Also with the aims of maintaining thermal comfort and air quality for indoor environment and reducing the energy cost by the air conditioning and ventilation system. The agent is firstly trained with 10 years past experience of climate data on simulated environments of laboratory room and classroom. The testing result shows that this agent can give satisfactory thermal control and balance performance by achieving good Predictive Mean Vote (PMV) [Fan+70] indexes representing occupant's thermal comfort, reduce the carbon-dioxide level by 10% and energy consumption by $4-5\%$.

In terms of energy consumption management, Yang et al. [Yan+15] developed an RL-based photovoltaic-thermal (PV/T) array and geothermal heat pump control to satisfy the heating demand while reducing fossil fuel consumption. This work basically uses a Q-learning scheme combined with experience replay to train controllers for different control loops, such as the PV/T energy output, compensating ground heat through the borehole and maintaining an optimal operation temperature difference between the source and load side. This RL-based control can increase the net power output by 11.4% and satisfy the heating demand in the third year compared with the standard rule-based controller. Another kind of building thermal system is based on controlling a mixing loop of heat supply. Overgaard et al. [Ove+19] have used Q-learning with eligibility traces to learn a mixed loop controller with data collected from a real office building. The state-action value function is approximated by the radial basis function. The resulting controller outperforms standard industrial controller by saving 20.5%.

Barrett et al. [BL15] has also developed an RL-based HVAC controller to autonomously keep the room temperature close to the occupant's set point. And this work has estimated an occupancy prediction model by real-world data. With the information provided by an occupancy prediction model, this agent is trained by a Q-learning framework to minimize energy cost and improve occupant comfort. The resulting HVAC controller demonstrates a 10% energy reduction compared to the programmable controller in the heating task. Another specific occupant behaviour based building climate control by Fazenda et al. [Faz+14] applied an RL-based HVAC controller to automatically adjust thermostat settings to keep thermal comfort of the human body. The simulated human is modelled by an $\alpha-$fuzzy logic model, and the simulated behaviour indicates the occupant's interactions with the thermostat, such as working or going out of the building, adjusting thermostats when the occupant is uncomfortable. The training os simulated behaviour is based on prior devised schedules of occupant's interactions with thermostats. The fuzzy-logic element was used for modelling set-point temperature selections. The controller represented by a neural network was trained with a Q-learning framework to examine Bang-bang control and set-point adjustment control, respectively. The resulting HVAC controller can offer the pre-heat for the room in accordance with the occupant's behaviours.

The application of simple Monte-Carlo policy gradient (MCPG) [Sut+00], [PS06] for a building HVAC system control was implemented in Jia's work [Jia+19]. An HVAC control policy is trained on a building simulation model (*EnergyPlus* software platform) to control the supply air temperature and airflow rate. The result showed that policy gradient-based control outperforms the baseline controller in achieving and keeping room temperature in the comfort zone. Also, Wang et al. [WVH17] has employed the MCPG RL method as the learning framework for optimizing a building HVAC controller. The control policy is designed using the Long-Short-Term Memory (LSTM) network, and the HVAC control simulation is conducted via the communication between Building Controls Virtual Test Bed (BCVTB) tool to a simulated office building on the *EnergyPlus* platform. The LSTM HVAC control policy is trained using two days of simulation data sampled by 5 minutes, two other practical methods: ideal PMV baseline and variable controls, are implemented as comparisons with the trained LSTM HVAC control policy. The validation results indicate the MCPG-based controller averagely improves comfort (quantized by PMV) by 15% and energy efficiency by 2.5% compared to the rest two control methods.

To improve the learning efficacy of primitive policy gradient RLs such as MCPG, advantage actor critic (A2C). Based on Schulman's Trust region policy optimization (TRPO) [Sch+15a], Wang's next work [WVH18] proposed a proximal actor critic method, which uses Kullback-Leibler (KL) constraint as a loss function to constrain policy learning objective in addition to the Monte-carlo policy gradient method. A recurrent neural network (RNN) controller is designed for an HVAC heating coil system. The control action is to adjust the hot water flow rate valve so as to keep the air temperature at a certain set point. This RL-based HVAC controller can achieve the desired air temperature faster than two baseline controllers: the proportional-integral (PI) and linear quadratic regulator (LQR) ones. Meanwhile, the RL-based controller can achieve the lowest integral square and absolute errors compared with PI and LQR controllers.

Zhang et al. [Zha+18] employed a policy gradient-based RL named as asynchronous advantage actor critic (A3C) [Mni+16] to integrate the building energy model into a model-based optimal control module. The agent is doing online control to real HVAC system, receiving data feedback from the real system to calibrate the building energy model. The agent is being trained off-line with feedback from cal-

ibrated model. This framework has been implemented in an office building; the resulting agent controls the radiator's supply water temperature to achieve indoor thermal comfort and save 15% heating energy. And Zhang's following study [ZL18] used the A3C learning scheme combined with calibrated building model to trained a radiant heating system controller inside an office building. The evaluation showed that this RL-based controller saved 16% to 19% energy than rule-based control.

Apart from the RL applications in thermal control for buildings, a recent work by Brusey and Diana [Bru+18] has used RL for vehicle HVAC system control. The cabin's thermodynamics is described for the vector of cabin state and HVAC actions. In this case, the reward function is related to two factors: the equivalent temperature [Nil04] for indicating comfort and energy consumption. This system outperforms the fuzzy-logic and standard bang-bang controller in terms of comfort percentage. However, this RL learning system uses 6.8 years of simulated time to train a working control policy, such a long simulated time cost is not feasible in real-world applications.

2.5 SUMMARY

This chapter mainly includes reinforcement learnings from traditional approaches to state-of-the-art policy gradient reinforcement learnings (PGRLs), introduce some novel PGRL applications in robotics control tasks, and state-of-the-art RL-based thermal control tasks. The issue is that most of these thermal control tasks are using Q-learning or SARSA-based RLs. Some cases cost 6.8 to 10 years of past experience to learn the optimal control policy, and this means that the time of learning experience is not feasible. Correspondingly, the PGRLs have subsequent advantages

- Policy gradient is compatible with a wider range of problems, especially for the RL tasks (continuous action space, high dimensionality) with difficulties of learning Q-function
- Policy gradient is more efficient in using training dataset compared to Q-learning and SARSA
- The policy directly gives action outputs without consulting the state-action values

However, the simple policy gradient method yields several issues

- High variance: bad actions cause a positive update of policy parameter
- Unconstrained learning step size: first-order gradient step size may violate the good policy parameter

As for the variance reduction, a practical method is to use the value function $V(s)$ as the baseline, and cumulative rewards $R(\tau)$ subtract value function $R(\tau) - V(s)$ indicates how good or bad the action is as compared to average. This form of subtracting value function is known as the advantage function. As for the learning step-size, this chapter includes TRPO and PPO approaches to ensure monotonic policy improvement.

The following chapter details the applications of PGRL methods in a vehicle cabin control problem, shows the resulting HVAC policy's capability of achieving comfort and energy consumption.

3

POLICY GRADIENT REINFORCEMENT LEARNING BASED VEHICLE CLIMATE CONTROL

Autonomous car cabin climate controller is designed to recognize the pattern of thermal conditions and make correct climate conditioning strategies to satisfy occupant thermal comfort settings. Existing reinforcement learning (RL) based HVAC control approaches utilize the Q-learning or SARSA-based methods to estimate action-value functions, then consulting control strategies from the estimated state-action values. A recent work [Bru+18] has developed a tile-coding based SARSA learning scheme to train a thermal conditioning agent being able to select control actions, such as heating/chilling the cabin air, fan speed for airflow and circulation rate according to the observed temperatures of the environment, cabin air and interior mass. These control actions are designated to adjust the cabin air and mass temperatures to achieve a preset thermal comfort condition for occupants. Based on real car cabin thermal data measurements, Hintea [Hin+14] has developed a human body's thermal model for vehicle Heating, Ventilation and Air Conditioning (HVAC) control. The resulting agent from previous work [Bru+18] has averagely achieved a 67% duration of thermal comfort and 0.77 kW power consumption over 200 different testing cases (1×10^3 s simulated time for each case). However, this approach cannot satisfy the occupant's thermal comfort under cold or hot surrounding temperatures outside the car cabin. As policy gradient reinforcement learnings (PGRL) are widely developed to solve numerous robotics control tasks that cannot be solved using value-based approaches, including the deep q networks (DQN) category. It is believed that PGRL families can estimate a promising agent being able to maintain the cabin thermal comfort under extreme cold and hot surroundings, while consuming reasonable amount of energy. These requirements account for subsequent research questions:

RQ1.1 Can the vehicle HVAC agent, trained by PGRL schemes, reduce the time taken to achieve occupant thermal comfort and keep reasonable energy consumed by the HVAC system compared to the SARSA based learning scheme?

RQ1.2 Can the PGRL HVAC training scheme learn an optimal control policy within a reasonable number of training samples?

These questions aim to investigate the impact of using PGRL methods in training HVAC controller, which learns from the cabin environment and thermal condition rewards. This chapter's main contribution is evaluating the benefits of proposed policy gradient based reinforcement learnings, and examining corresponded performance in satisfying thermal control and energy cost. The proposed PGRL methods include primitive Monte-Carlo policy gradient (MCPG), Mean Actor Critic (MAC), Trust region policy optimization(TRPO) and Proximal policy optimization (PPO), with details in chapter 2. This chapter also examines how long does it take for the agent to learn an optimal policy when being trained by corresponded PGRL algorithm.

3.1 PROBLEM STATEMENT

The overall system consists of two parts: the cabin environment and RL agent. The reinforcement learning framework allows the agent (policy) to explore the cabin environment by visiting its state. The state is commonly known as sensation, which serves as input to the RL agent. After observing a state, the agent chooses an action to activate HVAC control options for heating, chilling or preserving current thermal conditions. Meanwhile, the RL framework yields a reward

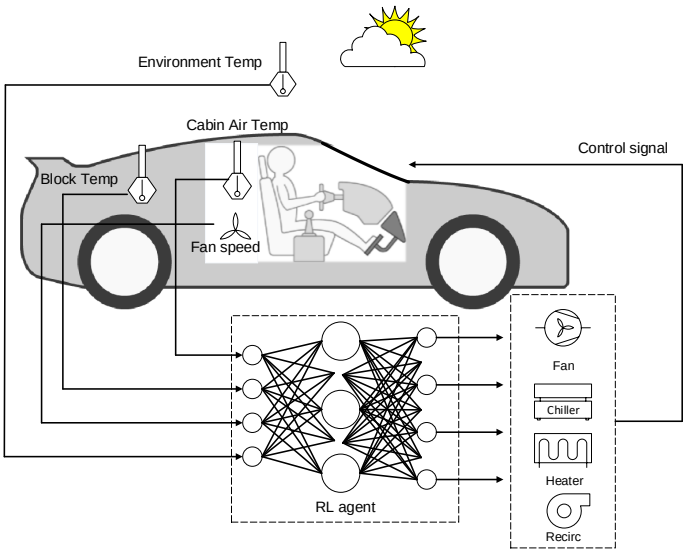

Figure 3.1: A fully connected feed-forward policy network based HVAC control

$R(s, a)$ used to evaluate how good the current state-action pair is. For example, the occupant comfort level and energy consumption can be used as reward information. The objective of RL framework is to learn

a decision process to produce behaviour that maximizes predefined reward function. Figure 3.1 presents a simple thermal control process by feeding the information of car cabin air flow and temperatures to the RL agent, which activates HVAC signals of internal conditioning temperatures, fan airflow and recirculation.

3.1.1 *Policy network and control variables*

The policy (agent) $\pi_\theta(a \mid s)$ is a fully-connected multilayer perceptron (MLP) neural network with a softmax output layer. As the control actions are sorted from combinations of vent air temperature T_i, ventilation speed v_i and recirculation ratio A_r in a finite space $\mathcal{A}$ (where $[T_i, v_i, A_r] \in \mathcal{A}$), and the input states are continuous by linear approximation equation 3.9. It is practical to choose a softmax output layer [Muroo] [ST09] for this MLP policy-based agent to yield distinct distributions of all possible actions when dealing with continuous thermal states. For example, the policy yields probability of choosing a specific HVAC control action a_j ($a_j \sim \pi_\theta(\cdot \mid s)$) from finite space $\mathcal{A}$ when giving an input state s observed by the agent:

$$\pi_\theta(a_j \mid s) = \frac{\exp\left(\phi_\theta^j(s)\right)}{\sum_{k=1}^{N_a} \exp\left(\phi_\theta^k(s)\right)} \tag{3.1}$$

Where $\phi_\theta(s)$ is an MLP neural network with full-layer weights θ and N_a output units, the input and output layers respectively consist of 4 and 60 neurons. The number of output layer nodes correspond to the scale of state vector $[T_c, T_m, T_{env}, \dot{v}_i]^T$ and the total number of actions. The deep layers neurons of $\phi_\theta(s)$ network use hyperbolic tangent activation function $\tanh(x)$. The upgrade of policy network parameter θ follows the gradient descent approach assisted by the gradient optimizer [KB14] to overcome possible saddle points in stochastic objective functions. The optimization process is to update weight θ by maximizing expected rewards or advantage value functions resulted from actions which as selected by policy π_θ. So the policy gradient process essentially estimates gradient of policy parameter $\Delta\theta$ over the loss function $J(\theta)$ as follow

$$\Delta\theta \leftarrow \mathbf{E}\left[\sum_{a\sim\pi}^{\infty} \nabla_\theta \log(\pi_\theta(a \mid s)) \cdot A^\pi(a, s)\right] \tag{3.2}$$

and according to chapter 2, the optimization approaches for target $J(\theta)$ above are different. The MCPG and MAC are using Adam optimizer [KB14] to directly compute the gradient $\nabla_\theta J(\theta)$. However, the TRPO approach applies a natural gradient method to adjust its policy network parameter

$$\theta = \theta_{old} + \sqrt{\frac{2\delta}{g^T F^{-1} g}} F^{-1} g \quad \text{where} \quad g = \nabla_\theta \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta old}(a_t \mid s_t)} R_t \right] \Big|_{\theta=\theta_{old}} \tag{3.3}$$

where the Fisher information matrix F is approximated by a conjugate gradient process. A KL-divergence boundary and a positive reward update are used to examine whether this estimated θ leads to positive update over the old one θ_{old}. The details of this TRPO [Sch+15a], [Sch16] process can be found in Algorithm 9 in the appendix section. The PPO [Sch+17] algorithm applies a clip objective function to compare the probability ratio $w_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$ with the clip ratio boundary $1-\epsilon$ and $1+\epsilon$ to directly constrain the updated policy network parameter. The clip objective function is shown below

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(w_t(\theta)\hat{A}_t, clip\left(w_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{3.4}$$

and the details are included in Algorithm 10 in the appendix section. The control actions include three control variables: vent air temperature T_i, vent airflow v_i and recirculation flap position A_r. Where these actions are selected from subsequent sets: T_i (ranging from 7 °C to 60 °C), v_i (airflow speed from 1 $l\,s^{-1}$ to 100 $l\,s^{-1}$) and A_r (zero to full recirculation).

- Action vector: $a = [T_i, v_i, A_r]^T$
- Finite action space $\mathcal{A}$: $T_i \in [7, 20.25, 33.5, 46.75, 60]$ °C, $v_i \in [1, 34, 67, 100]\,l\,s^{-1}$, $A_r \in [0, 0.5, 1]$, The combination of these sets forms a finite space including sixty $(5 \times 4 \times 3)$ specific actions
- Cabin state (in continuous space): $s = [T_c, T_m, T_{env}, \dot{v}_i]^T$

3.1.2 *Car cabin thermal environment*

The physical environment is modelled on thermodynamic processes inside a car cabin when it transits. This model accounts for simple heat

conduction, convection and radiation process based on Lee's lumped capacity model [Lee+15]. Applying this scenario into a simple mathematical model [Bru+18] that extensively formulates how thermal and air exchanges impact car cabin temperatures, thus further influencing occupant thermal comfort. This developed car cabin model mainly relies on following equations to keep heat balance:

$$\dot{Q}_h + I_{in}(T_{env} - T_c) = I_{fan}(T_x - T_c) \tag{3.5}$$

$$C_c \frac{dT_c}{dt} = I_{fan}(T_x - T_c) + \dot{Q}_{sol} + \dot{Q}_{occ} + \frac{T_m - T_c}{R_m} + \frac{T_{env} - T_c}{R_c} \tag{3.6}$$

$$C_m \frac{dT_m}{dt} = \frac{T_c - T_m}{R_m} \tag{3.7}$$

Where the temperatures are: cabin air T_c, cabin block T_m, or named as interior mass temperature, environment air T_{env}, mixed air T_x, which depends on the recirculation rate. The recirculation rate is a ratio of environment input air I_{in} and cabin-recirculated hot or cool air I_{fan}. Moreover equation 3.6 indicates step update for cabin air temperature and equation 3.7 indicates step update for cabin block temperature. The solar load Q_{sol} and occupant load Q_{occ} are maintained at 150 W and 120 W. The change of heat pump energy per unit time is denoted as $\dot{Q}_h$, where a positive $\dot{Q}_h$ means heating power and a negative value means cooling power. The absolute value of $\dot{Q}_h$ is considered as the unit energy W_h consumed by the HVAC system, while the blower energy cost is negligible. The interior mass thermal resistance R_m and capacitance R_c, cabin thermal resistance R_c are constants listed in Table 3.1, the cabin capacitance C_c is calculated by using cabin air volume V_c, air density ρ_c, capacitance factor k and specific heat c_p with the equation below:

$$C_c = C_c \times \rho_c \times k \times c_p \tag{3.8}$$

These constants are validated by testing cool-down and warm-up processes inside a Jaguar model XJ sedan car [Hin+14]. The cabin environment state, alternatively known as observation, comprises temperatures for cabin air T_c, interior mass T_m, external environment T_{env}, and airflow $\dot{v}_i$ (which determines the volume of air I_{fan} being heated or chilled per unit time). Hence forming a state vector $s = [T_c, T_m, T_{env}, \dot{v}_i]^T$, which the RL agent observes. While thermodynamic

Table 3.1: Model constants

Cabin volume V_c	$2.5\,m^3$
Cabin capacitance factor k	8
Solar load $\dot{Q}_{sol}$	$150\,W$
Occupant load $\dot{Q}_{occ}$	$120\,W$
Cabin resistivity R_c	$1/5.741626794 \times 4 \quad K\,W^{-1}$
Interior mass resistivity R_m	$1/(75 \times 1.08) \quad W^{-1}$
Interior mass capacitance C_m	$450 \times 0.02 \times 7850 \quad J\,K^{-1}$

process over a unit time Δt for the state being observed at time t $s_t = [T_c(t), T_m(t), T_{env}(t), \dot{v}_i(t)]^T$, yields subsequent update equation

$$T_c(t+\Delta t) = T_c(t) + \Delta t \cdot \frac{dT_c(t)}{dt} \quad T_m(t+\Delta t) = T_c(t) + \Delta t \cdot \frac{dT_m(t)}{dt}$$
$$T_{env}(t+\Delta t) = T_{env} \qquad \dot{v}_i(t+\Delta t) = v_i(t) \tag{3.9}$$

where environment temperature T_{env} is regarded as a fixed value, and v_i refers to the airflow control signal received from t to $t+\Delta t$.

3.1.3 *Reward function definition*

The measures of occupant comfort used in this thesis are named equivalent temperature T_e. The validations [Hin+14] based on data collected from real car shows that an equivalent temperature model [Nil04] is the most accurate one in representing cabin thermodynamics. Following equations indicate the derivation of equivalent temperature

$$T_e = \begin{cases} 0.5(T_c + T_m); \dot{v}_c \leqslant 0.1\,m\,s^{-1} \\ \\ 0.55T_c + 0.45T_m + \frac{0.24 - 0.75\sqrt{\dot{v}_c}}{1+I_{cl}}(36.5 - T_c); \dot{v}_c > 0.1\,m\,s^{-1} \end{cases} \tag{3.10}$$

Where mean radiant temperature T_r equals interior mass temperature T_m and clothing insulation I_{cl} is set as a constant value of 0.7. The airflow $\dot{v}_c$ is calculated by using the cross-sectional area (A_{cs}) of the vents, and the number of fans N_{fan} on the dashboard

$$\dot{v}_c = \frac{v_i}{N_{fan} \times A_{cs}} \tag{3.11}$$

where there are two fans and the cross-sectional area is $5.04 \times 10^{-3} m^2$ according to data provided by a car HVAC device [Foj+16].

The learning goal is to maximize the time duration in thermal comfort zone (when T_{eq} within $T_{target} \pm 1°C$) while reducing energy consumption and noise caused by a blower. This trade-off between thermal comfort and energy, airflow cost use can be expressed as the following reward function

$$R_c(s) = \begin{cases} 0; & \text{If } T_e \in T_{target} \pm 1°C \\ -1; & \text{otherwise} \end{cases} \tag{3.12}$$

$$R(s,a) = R_c(s) - R_d(s) - \frac{E(s,a)}{w_d}; \tag{3.13}$$

$$R_d(s) = \mid T_e - T_{target} \mid; \qquad E(s,a) = \mid \dot{Q}_E \mid + 2v_i \tag{3.14}$$

R_c and E denote thermal comfort reward and energy, airflow cost, respectively, w_d denotes energy divisor used as the weight for the energy consumption part in the reward function. If choosing a small value for energy divisor, for example w_d equals 100, the resulting RL HVAC controller tends to sacrifice the thermal comfort to save the energy. If choosing a large number, for example w_d equals 4000, the resulting RL HVAC controller tends to ensure the comfort prior to saving the energy. Hence, w_d acts as a trade-off between comfort and energy. In this experiment, the energy divisor w_d is chosen between 3×10^3 to 3×10^4 according to the reward function setting in Brusey's research [Bru+18]. The term $R_d(s)$ acts as the penalty for getting far from the target equivalent temperature. And $R_d(s)$ is inspired by the reward shaping technique in locomotion task, representing a reward function by calculating the difference between observed joint angles and the desired angle values [Raj+17]. This RL-based task employs a similar approach to calculate difference of equivalent temperatures between the observed one T_e and target T_{target}.

3.2 EXPERIMENTAL SETTINGS AND EVALUATION METHODS

At the beginning of each agent training episode, the initial state is randomly selected from uniform distributions over the range for each feature: temperatures of cabin $T_c \in [0, 50]\,°C$, Interior mass $T_m \in [0, 50]\,°C$, environment $T_{env} \in [0, 40]\,°C$ and vent airflow $v_i \in [1, 100]\,l\,s^{-1}$. This

information of cabin state is sufficient for the definition of reward function and RL framework. Another important setting is the episode length, limiting the number of rewards and observations received by the agent. In this case, the episode length is uniformly set to 500, where the period for each step Δt (by equation 3.9) equals 10s. Given such fundamental definitions of state and episode information, we further implement proposed policy gradient methods (MCPG, MAC, TRPO and PPO) to train the policy. Meanwhile, meta parameters are essential in controlling the learning processes. For example, reward discount factor γ constrains future received rewards, learning step size, and MLP neural network determines how quickly the training proceeds. The GAE discount factor λ helps reduce the variance of advantage functions [Sch16]. The entropy coefficient $C_{\mathcal{H}}$ allows certain explorations over the training. Table 3.2 indicates meta parameters for each policy gradient algorithm: The performance of PGRL controllers

Table 3.2: Meta parameters

Algorithm name	MCPG	MAC	TRPO	PPO
Discount factor γ	0.95	0.98	0.98	0.98
Policy learning step size α	3.0×10^{-3}	2.0×10^{-3}	Adaptive	2.0×10^{-4}
Value function learning step size	None	5.0×10^{-3}	1.0×10^{-4}	2.0×10^{-4}
No. of the hidden layer (policy)	1	1	2	2
No. of neurons per hidden layer (policy)	100	100	96	96
No. of the hidden layer (critic)	None	2	2	2
No. of neurons per hidden layer (critic)	None	64	96	96
GAE discount factor λ	None	None	0.98	0.98
Entropy coefficient $C_{\mathcal{H}}$	None	None	0.01	0.01

trained with proposed algorithms is evaluated in terms of trials (proportional to episode numbers) taken to optimize the working agent, cumulative rewards per episode, the time steps of thermal comfort state sequence and averaged power.

It is understood that the higher reward values usually indicate lower fan speed, energy cost and longer duration of comfort according to its definition. To compare the training performance of all proposed PGRLs, the averaged episodic rewards and thermal comfort period ratio against training iterations are respectively illustrated. The resulting PGRL controllers are tested with start states randomly selected for all training episodes and 200 pre-selected randomised ones from the SARSA-based car cabin climate control simulation [Bru+18]. The testing procedures are listed as follows

- Randomly select i^{th} start state $s^i_{t_0}$ beginning from time step t_0 as the initial state for i^{th} episode

Algorithm 1 Training thermal control policy with PGRLs

1: Initialize Policy $\pi_\theta(\cdot \mid s)$ where $\theta \leftarrow \theta_0$
2: Initialize Value or State-action function: $V_\omega(s)$ or $Q_\omega(s, a)$ where $\omega \leftarrow \omega_0$
3: **for** i=0,1,2,... until the end of the training process **do**
4: Initialize cabin state s_0^i, i.e.randomly select temperatures of cabin T_c, interior mass T_m, environment T_{env} and airflow v_i from defined space, thus $s_0^i = [T_c, T_m, T_{env}, v_i]$
5: Present initial state to policy $\pi_\theta(\cdot \mid s)$, obtain HVAC control action a_0^i and execute it, then observing updated cabin state s_1^i, recurrently receiving full episode of states $\{s_0^i, s_1^i, s_2^i, \cdots, s_T^i$ and actions $\{a_0^i, a_1^i, a_2^i, \cdots, a_T^i$ until the ending time step T
6: Calculating rewards $\{r_0^i, r_1^i, r_2^i, \cdots, r_T^i$ according to equation 3.14
7: Present sequence of states, actions, rewards to MCPG 5, MAC 7, TRPO 9 and PPO 10 learning algorithms to respectively obtain an update of policy $\hat{\pi}_\theta$ and critic network $\hat{Q}_\omega$ or $\hat{V}_\omega$ (where MAC uses state-action function $Q_\omega(s, a)$, while TRPO, PPO use value function $V_\omega(s)$ to estimate advantage $A(s, a)$)
8: Update $\pi_\theta(\cdot \mid s) \leftarrow \hat{\pi}_\theta$, $Q_\omega \leftarrow \hat{Q}_\omega$ or $V_\omega \leftarrow \hat{V}_\omega$
9: **end for**
10: **Return**: optimal policy π_θ^*

- Resulting policy $\hat{\pi}_\theta(\cdot \mid s)$ observes state, select and execute control actions for time period Δt, then observing the updated state after Δt and recurrently select, execute actions for next period of Δt $\cdots\cdots$ hence yielding sequence of states, actions and rewards until the ending time $t_0 + T$
- As the sequence (episode) of cabin state $\left[s_{t_0}^i, s_{t_0+\Delta t}^i, \cdots, s_T^i\right]$ and corresponded control actions $\left[a_{t_0}^i, a_{t_0+\Delta t}^i, \cdots, a_T^i\right]$ can respectively indicate equivalent temperatures T_e (by equation 3.10) and energy consumption $\dot{Q}_h$ (by equation 3.5), then calculating the comfort period percentage and mean power consumption over the whole episode time t_0 to $t_0 + T$
- Recurrently test policy $\pi_\theta(\cdot \mid s)$ on the following $i + 1^{th}$, $i + 2^{th}$,$\cdots$ start state $s_{t_0}^{i+1}, s_{t_0}^{i+2}, \cdots$ and calculate comfort period percentage, mean power consumption for $i + 1^{th}, i + 2^{th} \cdots$ episodes. In the end, this test scenario yields a statistical dataset of comfort period percentage and power consumption corresponded to randomised initial cabin states.

Moreover, this testing procedure examines explicitly whether the PGRL HVAC controllers can keep the occupant thermal comfort (de-

fined as equivalent temperature T_e) under extreme hot or cold environmental temperatures. These two circumstances are known as cool-down and warm-up control processes. The PGRL agent activates the HVAC system to cool cabin air in hot weather for the cool-down case. Conversely, a warm-up case means that the PGRL agent activates HVAC system to warm cabin air in cold weather. For the cool-down testing case, the cabin air and interior mass temperatures are initialized to 45°C, the environment (ambient) temperature is set to 40°C. However, the warm-up case sets the cabin, interior mass and environment temperatures to 1°C. The warm-up testing case also evaluates whether the PGRL based HVAC controller can introduce warm air into the cabin. All initial vent airflow rate is set to $1\ l\,s^{-1}$ (litre/ second) in default.

3.3 RESULTS AND DISCUSSIONS

According to Hintea's work [Hin+14] of defining human's thermal comfort inside cabin environment, the setting of target equivalent temperature equals to 24°C ensures the occupant's thermal comfort feelings in both summertime and wintertime climate. Although another earlier work [Bru+18] shows that a SARSA-based RL HVAC agent can effectively achieve and maintain the occupant's equivalent temperature within 24 ± 1°C when the cabin needs to be cooled from hot weather. While in the warm-up case, even though increasing the duration of each training episode by 20%, this RL HVAC controller achieves a final equivalent temperature around 13°C (lower than the target 24°C) in a cold environment. Another problem with the SARSA based RL HVAC controller is that it takes 2×10^8 seconds (approximately 6.3 years) of learning time in the simulation. While this duration is almost equivalent to the estimated lifetime of a real car, hence it is not efficient to learn an optimal control policy with the cost of a car's lifetime. Therefore the proposed policy gradient RLs aim to solve subsequent problems

- Thermal conditioning objectives: the optimized policy (agent) can identify both cold and hot thermal conditions, respectively executing warm-up and cool-down control tasks, maintain equivalent temperature (ET) within the comfort zone 24 ± 1°C (where 24°C is the target)

- Learning efficiency: to examine whether the proposed PGRL scheme estimates optimal control policy while consuming less learning time (equivalent to less number of learning episodes)
- A wider application range: resulting agent satisfies the thermal setting of 24°C target equivalent temperature when being given arbitrary initial cabin thermal states sampled from space of cabin air T_c ($T_c \in [0, 50]\,°C$), interior mass T_m ($T_m \in [0, 50]\,°C$), environment T_{env} ($T_{env} \in [0, 40]\,°C$) temperatures

This chapter mainly presents the processes of optimizing HVAC control policies by corresponded PGRL algorithms in terms of episodic rewards, comfort rate against learning trials, and testing cases covering the above requirements.

3.3.1 *Description for resulting data illustration*

As mentioned above, the PGRL-based task is to learn an intelligent thermal controller which can activate the cabin HVAC system to achieve target thermal comfort within a reasonable time. More importantly, the PGRL algorithm applications need to account for the number of iterations (learning trials) to upgrade policy parameter: $\theta_{i+1} = \theta_i + \Delta\theta_i$. This section briefly introduces several comparison metrics for learning outcomes of different RL-based approaches and their testing performances.

- Episodic rewards: the summation of rewards $(r_0, \cdots, r_T)$ received in an episode of thermal states $\{s_0, a_0, r_0, \cdots, s_t, a_t, r_t, \cdots, s_T\}$ with time limit T
- Episodic comfort rate: inside an episode time, the **proportion** of time that the cabin ET is maintained within the comfort zone ($24 \pm 1°C$).
- Power consumption: the power that the HVAC control actions inside each episode (with a certain length of time limit) averagely consume to adjust cabin thermal conditions

In the following comparison section, the **averaged episodic rewards** in Figure 3.2 indicates how the episodic rewards vary as the policy parameter being upgraded throughout 4×10^3 learning iterations. Similarly, the **averaged episodic comfort rate** in Figure 3.3 indicates the proportion of thermal comfort time that the PGRL HVAC control

policy can achieve over the policy learning iterations. Both graphs use points (scatter dots) to represent the episodic rewards and comfort rate against the iterations due to the oscillations. For example, the episodic reward steadily increase from -1500 to -1000 during 500^{th} to 600^{th} iterations but decrease to -1200 at 650^{th} iteration then increasing to -998 at 680^{th} (caused by policy exploration or degraded policy update). If using solid or dashed curves to represent the learning trials, such oscillations can blur the observation of the results. The following scatter graphs use green, red, purple and blue points to respectively denote the Monte-Carlo policy gradient(MCPG), Mean actor critic (MAC), Trust region policy optimization (TRPO) and Proximal policy optimization (PPO). These approaches maximise expected rewards or advantage values to update the parameterized policy π_θ. Either by primitive stochastic gradient descent (SGD) or monotonic non-decreasing rewards methods, such as natural gradient by the TRPO or clip ratio method by the PPO.

3.3.2 *Policy gradient methods comparison*

Figure 3.2 shows the episodic rewards against learning trials performed by MCPG, MAC, TRPO and PPO method. Each method's trial is averaged over results of using ten different energy divisor w_d values, which are uniformly chosen from 3×10^3 to 3×10^4. Because the energy divisor acts as a weight balance between comfort and energy-efficiency reward in equation 3.14. The result shows that both MCPG and MAC indicate episodic rewards ranging from -5000 to -3000; the mean reward of MCPG and MAC respectively equals to -4244.1 and -4461.7. Conversely, TRPO and PPO learning processes indicate episodic rewards, most of which are above -1000 and constantly converge to the value around -500. The averaged episodic reward values of TRPO and PPO respectively correspond to -1085.6 and -978.68 (lower than -500 due to explorations) for the entire learning trials. Compared to the MCPG and MAC methods, the learning trials of TRPO and PPO show increasing rewards from -5000 to -500, this means that the control policies are being improved to comply with the objectives of maintaining equivalent temperature in the comfort zone ($24 \pm 1°C$). Moreover, Figure 3.3 presents the thermal comfort conditioning performance of the applied methods. The TRPO and PPO trials of thermal comfort rate mostly converge to 0.9, meaning that the control policies can maintain target thermal comfort around 90% of

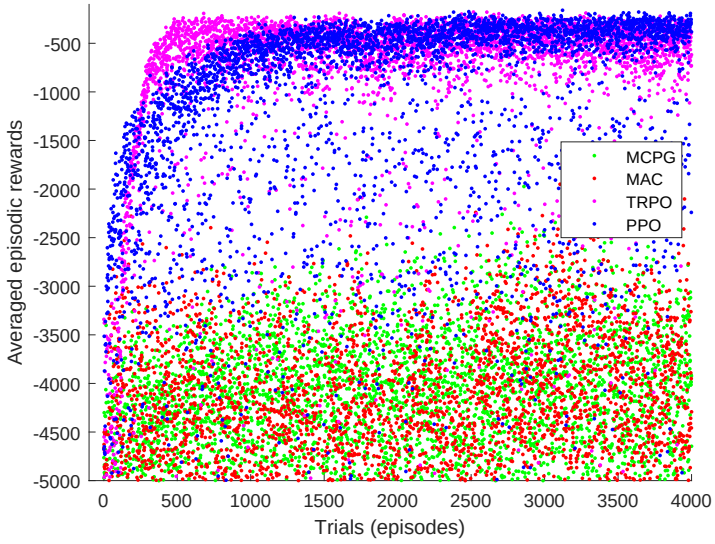

Figure 3.2: Averaged episodic rewards against learning trials by MCPG, MAC, TRPO and PPO methods

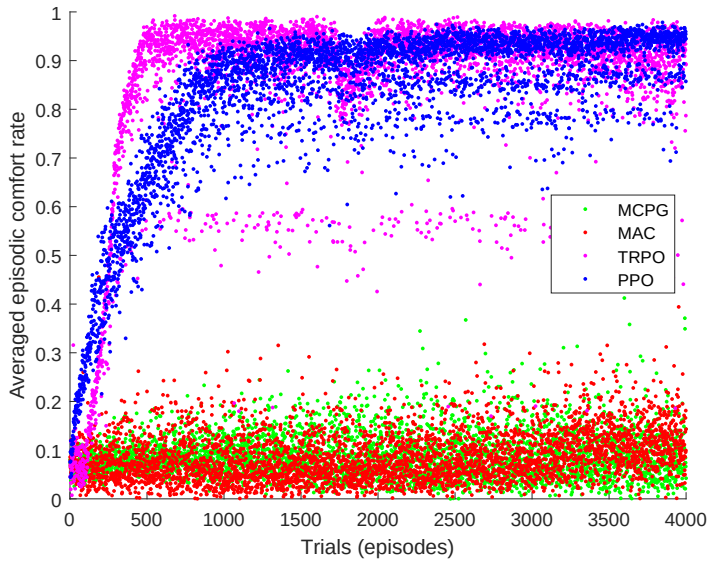

Figure 3.3: Averaged episodic comfort rate against learning trials by MCPG, MAC, TRPO and PPO methods

the full episode duration (5×10^3 s). The mean thermal comfort rate of TRPO and PPO respectively equals 0.8890 and 0.9002, which approximately outperform MCPG and MAC approaches by 70%. However, the MCPG and MAC trials indicate convergence to 0.0884 and 0.0864.

The policies $\pi_\theta(\cdot \mid s)$ are being optimized until the end of learning trials. So the next procedure is to examine how these policies handle different initial temperature cases. A practical testing case is to randomly select temperatures of cabin T_c, interior mass T_m from 0 to 50 °C, environment T_{env} from 0 to 40 °C, and airflow v_i from 1 to

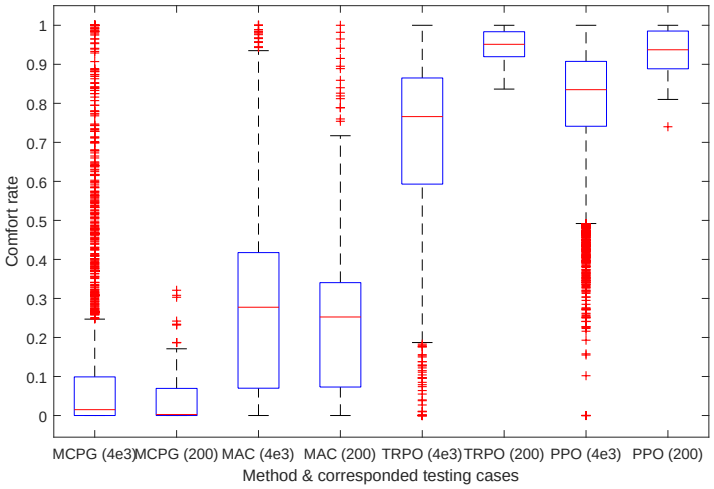

Figure 3.4: Thermal comfort testing results of the MCPG, MAC, TRPO and PPO control policies acting on 4×10^3 randomly selected initial states and 200 pre-selected ones from earlier work

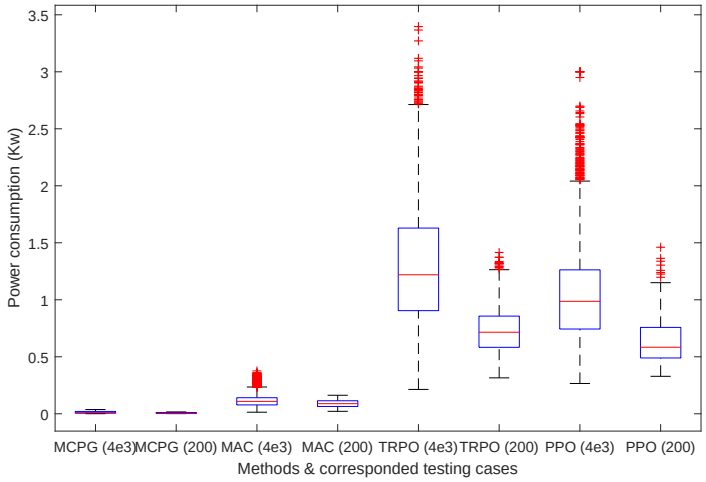

Figure 3.5: Power consumption testing results of the MCPG, MAC, TRPO and PPO control policies acting on 4×10^3 randomly selected initial states and 200 pre-selected ones from earlier work

100 l s^{-1}. Given this information as an initial state of cabin environment, the controllers (policies) are being tested throughout episode time steps of 1×10^3 s, then respectively assessing comfort rate and averaged power performance of this duration. The following box plot by Figure 3.4, Figure 3.5 demonstrate the performance of comfort and HVAC energy consumption with respect to the applied RLs. As the policies of MCPG, MAC, TRPO and PPO are being optimized throughout 4×10^3 learning episodes, the corresponding testing scenario includes the same amount of randomly selected initial states while using the same random seeds. Box plot in Figure 3.4 respectively

illustrates the comfort rate concentrations of testing MCPG, MAC, TRPO, PPO controllers on 4×10^3 randomly initialized episodes and 200 pre-selected ones. The MCPG based HVAC controller maintains less than 30% comfort duration for most cases, while the red cross marks denote outliers, indicating 5% test cases with a comfort percentage higher than 30%. Although the MAC controller improves the rate of comfort to 30%, most cases are centred around the value domain between 10% to 40% and averagely yield 28.18% comfort duration. Moreover, the 200 pre-selected testing cases of MCPG and MAC controllers separately yield 4.54% and 26.89% comfort. However, TRPO cases indicate that more than half of the comfort percentage results range from 60% to 87%, hence averagely indicating 71.53% duration in the thermal comfort zone. The corresponded pre-selected cases result in 94.76% comfort duration. Among the applied methods, PPO cases show that most of the comfort percentage values range from 75% to 90%, of which the mean value is 78.95%. Meanwhile, Figure 3.5 statistically compare power consumptions corresponded to the thermal comfort testing cases in Figure 3.4. Accordingly, both MCPG and MAC-based policies consume a small amount of energy in conditioning the cabin thermal status. While over the 4×10^3 randomly initialized cases, TRPO and PPO-based policies respectively consume averaged power of 1.30 kW and 1.05 kW, which are higher but more reasonable than the MCPG and MAC cases. Furthermore, Table 3.3 details the power consumption and the thermal comfort rate information for the applied RL algorithms, where the testing results of the SARSA-based HVAC controller are sorted from earlier work [Bru+18]. Therefore, it is clear that TRPO and PPO methods significantly outperform the SARSA, MCPG and MAC reinforcement learning approaches regarding to the endurance of achieving target thermal comfort.

Consider the extreme summer or winter weather of the environment, subsequent testing procedures directly present cool-down and warm-up tests in addition to the statistical results of comfort and power consumption. Figure 3.6 and Figure 3.7 show the processes of maintaining target thermal comfort (equivalent temperature (ET)) in extreme cold and hot environment (surrounding) temperatures while using policies estimated by MCPG, MAC, TRPO and PPO framework. In the warm-up test case, the temperatures of cabin T_c, interior mass T_m, environment T_{env} are initialized to 1°C, and airflow rate starts with $1l\,s^{-1}$, then employing estimated policies to achieve target comfort zone (equivalent temperature 24 ± 1°C). Figure 3.6 indicates that

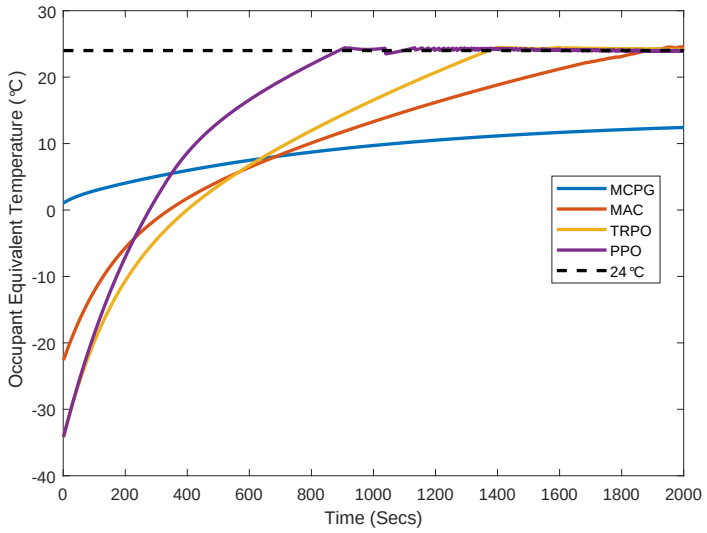

Figure 3.6: Occupant's equivalent temperature warm-up processes by MCPG, MAC, TRPO and PPO control policies under the environment, cabin air and mass temperature in 1°C

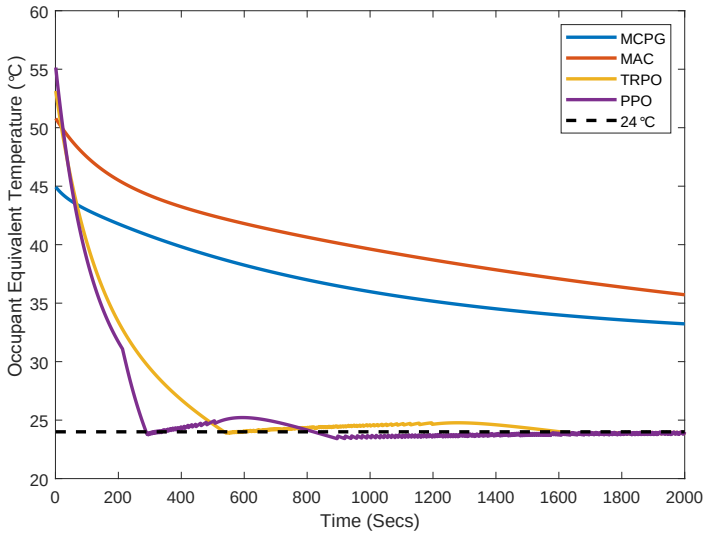

Figure 3.7: Occupant's equivalent temperature cool-down processes by MCPG, MAC, TRPO and PPO control policies under the environment temperature in 40°C and cabin air, mass temperature in 45°C respectively

the PPO method can increase the equivalent temperature to the target in around 15 minutes, and this is faster than TRPO's 22.6 minutes and MAC's 31 minutes. However, MCPG fails to reach the target due to choosing the lowest airflow rate ($1 l\, s^{-1}$) and heating power. Also, the airflow rate determines ETs according to equation 3.6, a higher airflow rate can pump more warm air into the cabin to increase the cabin temperature. However, the ET definition in equation 3.10 indicates

that a higher airflow rate can lower the ET when having low cabin air and interior mass temperatures. In the early stages of warm-up process, the MAC, TRPO and PPO choose airflow values 34 $l\,s^{-1}$, 67 $l\,s^{-1}$, 67 $l\,s^{-1}$, which lower the ETs below zero.

In the cool-down test case, the temperatures of cabin T_c, interior mass T_m are initialized to 45°C, environment T_{env} is set to 40°C, and airflow rate v_i starts from 1 $l\,s^{-1}$. Figure 3.7 shows that the PPO-based control can cool down the cabin, decrease ET from 55°C to the target in around 5 minutes, faster than TRPO controller. Over the 2×10^3 s, neither MAC nor MCPG succeeds in conditioning ET to the target comfort zone (24 ± 1°C). Like the warm-up cases, the MCPG controller selects the lowest airflow rate $1 l\,s^{-1}$ and cooling power, resulting in the slowest rate of decreasing ET. Similarly, during the cool-down process, the MAC controller selects lowest power to cool down the cabin, although it chooses a higher airflow rate of 34 $l\,s^{-1}$. It can be figured out from Figure 3.5 and Figure 3.3 that the MCPG and MAC-based RLs are monotonically reducing the energy cost but sacrificing the occupant's thermal comfort to improve the rewards. While TRPO and PPO maximally exploit the chances of improving policies, which can equally satisfy comfort and reduce energy cost. In conclusion, the trust region policy optimization (TRPO) and prox-

Table 3.3: Comparison between applied RLs

Algorithm name	SARSA	MCPG	MAC	TRPO	PPO
Averaged rewards $\bar{R}(\tau)$	––	-4244.1	-4461.7	-1085.6	-978.68
% Time spent in comfort (full)	––	8.87%	28.18%	71.53%	**78.95%**
Average HVAC power (kW) (full)	––	0.0129	0.1192	1.3012	1.0484
% Time spent in comfort (200)	67%	4.54%	26.89%	**94.76%**	93.42%
Average HVAC power (kW) (200)	0.77	0.0069	0.0888	0.7582	**0.6578**

imal policy optimization (PPO) methods yield control policies that can widely maintain occupant's thermal status within the comfort zone. By contrast, primitive learning methods such as Monte-Carlo policy gradient (MCPG) and mean actor critic (MAC) are simply reducing energy cost rather than exploring the chances of conditioning thermal comfort. However, this experiment has yet to investigate the performance of TRPO, PPO-based HVAC control policies. It is understood that the learning trials denote the episodes used to estimate HVAC control policies, and a promising RL framework is designated to acquire beneficial HVAC control policies while consuming as fewer data samples as possible. The following section details comparisons

between TRPO and PPO based HVAC controllers with respect to the number of training episodes.

3.3.3 *Comparison between TRPO and PPO*

To investigate whether increasing the number of learning episodes (trials) can also improve the resulting policies' performance. This section shows the testing cases of control policies estimated by the learning trials with 4×10^3, 1×10^4 and 2×10^4 iterations. In the following graphs, for example, the term "4e3 full" means that the policy estimated after 4×10^3 trials is fully tested by 4×10^3 random start states used in the learning processes, and "4e3 (200)" means the control policy is tested by the 200 pre-selected start states from previous work [Bru+18].

Figure 3.8 and Figure 3.9 present the comfort percentage and power consumption testing results by policies estimated by 4×10^3, 1×10^4 and 2×10^4 training episodes. The box plot results show that most of the comfort percentage range from 70% to 90%. The averaged testing results are 76.66%, 76.43%, 76.53% for the full test, and 93.17% 93.23% 93.23% for the pre-selected test; these groups of testing results indicate that increasing number of learning trials to 2×10^4 do not significantly improve the thermal conditioning performance of TRPO HVAC control policy. Accordingly, the power consumption testing results in Figure 3.9 averagely yield 1.3316 kW, 1.2654 kW, 1.2697 kW for the full range tests, and 0.7235 kW 0.6539 kW, 0.6594 kW for the pre-selected tests. Also, the change of power consumption from 1.3316 kW to 1.2697 kW demands an extra 1.6×10^4 episodes from 4×10^3 to 2×10^4. Meanwhile, the TRPO control policies are tested under the circumstances of extreme cold and warm environment weather. Graphs of Figure 3.10 and Figure 3.12 indicate warm-up and cool-down processes done by TRPO policies trained with 4×10^3, 1×10^4 and 2×10^4 trials. The warm-up process starts with 1°C of cabin temperature T_c, interior mass temperature T_m and environment temperature T_{env}, while cool-down starts with 45°C temperatures of cabin T_c, interior mass T_m and 40°C of environment T_{env}. The equivalent temperature (ET) starts from −33.1°C in the warm-up case, and the ET starts from 55°C in the cool-down case are caused by the high airflow rate v_i according to equation 3.10. In other words, the very negative ET (−33.1°C) and high ET values (55°C) respectively denote extreme cold and hot thermal feelings of the occupant, rather than

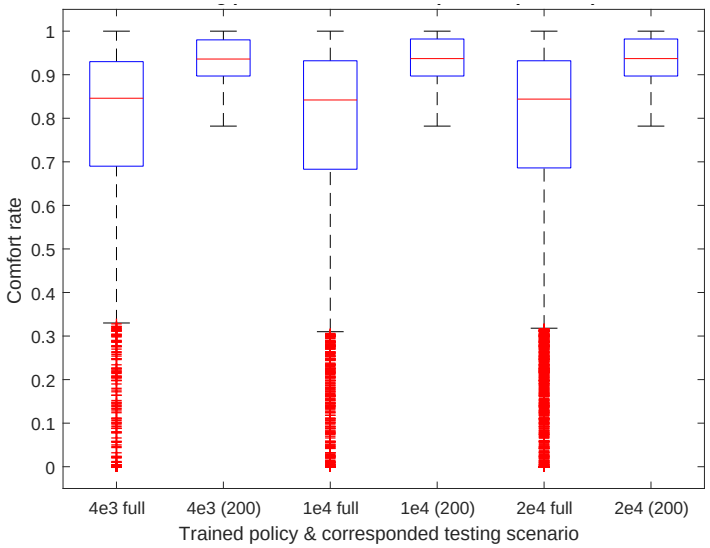

Figure 3.8: Comfort rate results by TRPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$

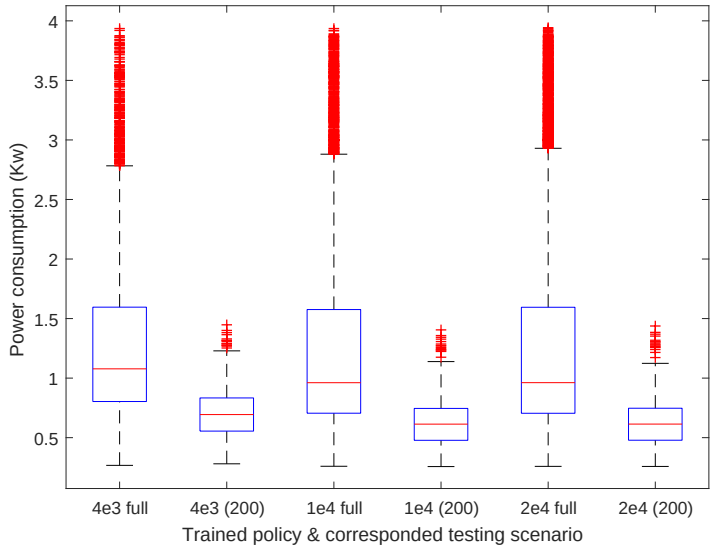

Figure 3.9: Power consumption testing results by TRPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$

indicating actual physical temperatures. Figure 3.10 shows that the TRPO-based controller warms up the ET to the target zone ($24 \pm 1°C$), and achieves the target thermal comfort in about 22.6 minutes for all the resulting TRPO control policies. The cool-down process in Figure 3.12 shows that the ET decreases to the target in around 9.5 minutes for all resulting policies. Further information of temperatures of cabin T_c and interior mass T_m in the warm-up and cool-down tests are given in Figure 3.11 and Figure 3.13; these two indicate that the

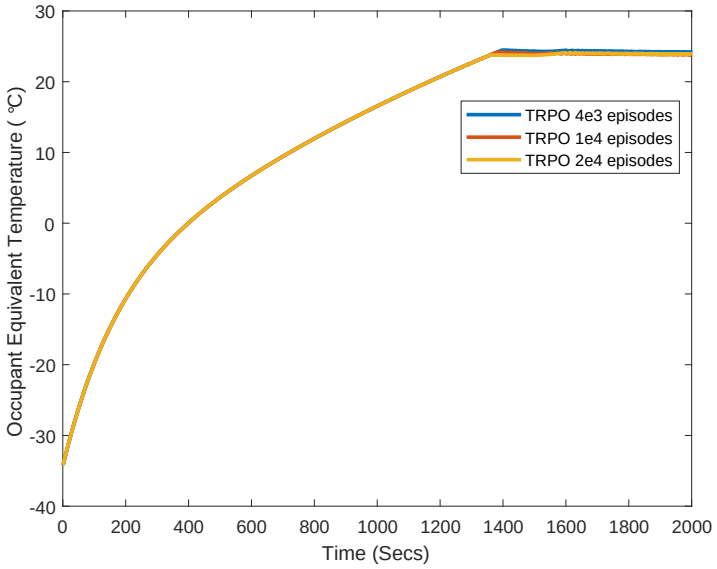

Figure 3.10: Warm-up process of occupant equivalent temperature by TRPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 1°C environment, cabin air and interior mass temperature

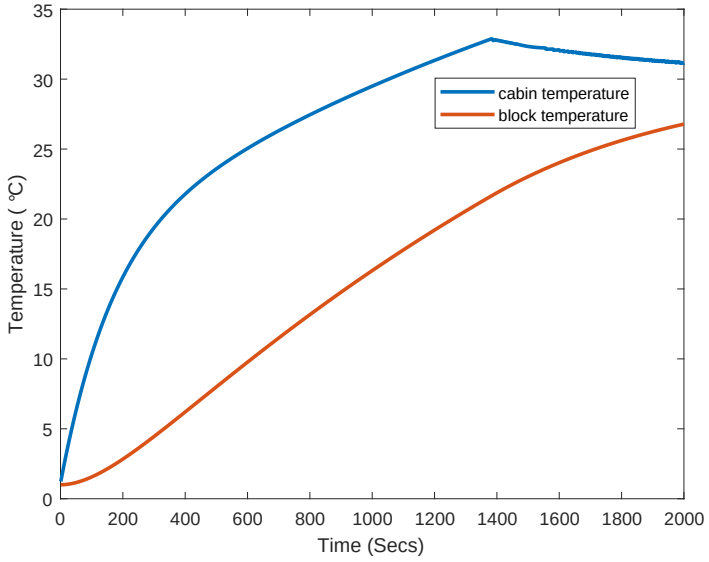

Figure 3.11: Physical temperatures of cabin T_c and interior mass T_m in the warm-up process, trained with 1×10^4 learning episodes

cabin environment is respectively being heated from 1°C to physical temperatures above 25°C, and being cooled down from 45°C to the temperature around 30°C. The fact that both T_c and T_m are above 24°C is reasonable because the airflow rate v_i also decides the equivalent temperature (ET) according to equation 3.10.

As mentioned above, the TRPO learning scheme makes it trivial to increase the number of trials over 4×10^3 for training a working

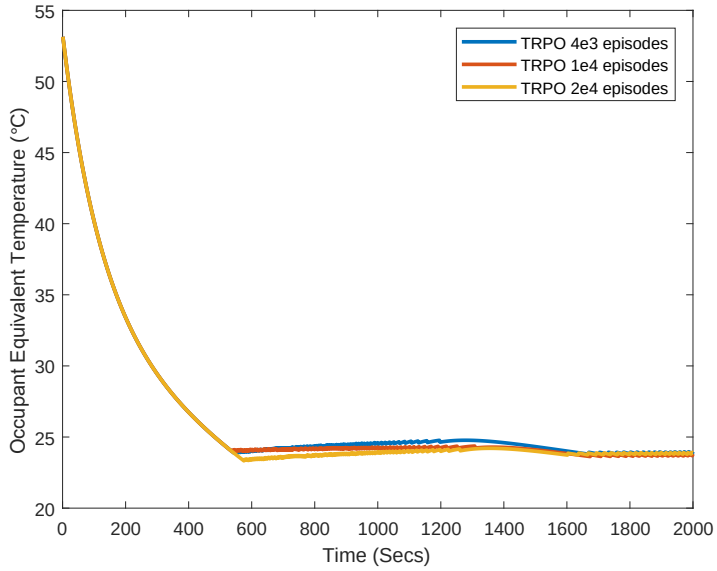

Figure 3.12: Cool-down process of occupant equivalent temperature by TRPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 40°C environment temperature and 45°C cabin air, interior mass temperature

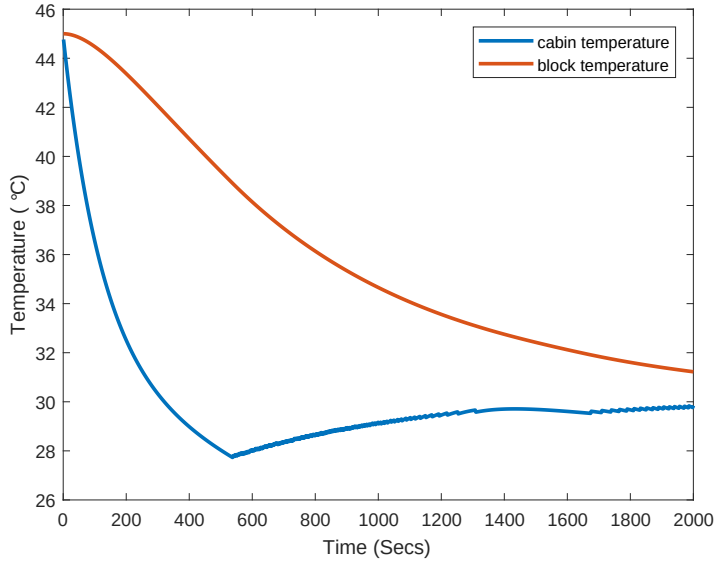

Figure 3.13: Physical temperatures of cabin T_c and interior mass T_m in the cool-down process, trained with 1×10^4 learning episodes

RL HVAC control policy. The following part mainly examines the impact of increasing learning trials for PPO based HVAC controllers. Accordingly, Figure 3.14 and Figure 3.15 demonstrate testing results of comfort rate and power consumption by policies respectively resulted from trials of 4×10^3, 1×10^4 and 2×10^4 in the PPO learning scheme. Conversely, the comfort percentage results are mainly distributed be-

tween 70% to 90%, individually yield 77.94%, 75.67%, 80.58% mean comfort for the full cases, and 92.30%, 90.69%, 92.73% for pre-selected cases. While the power consumptions in Figure 3.15 respectively in-

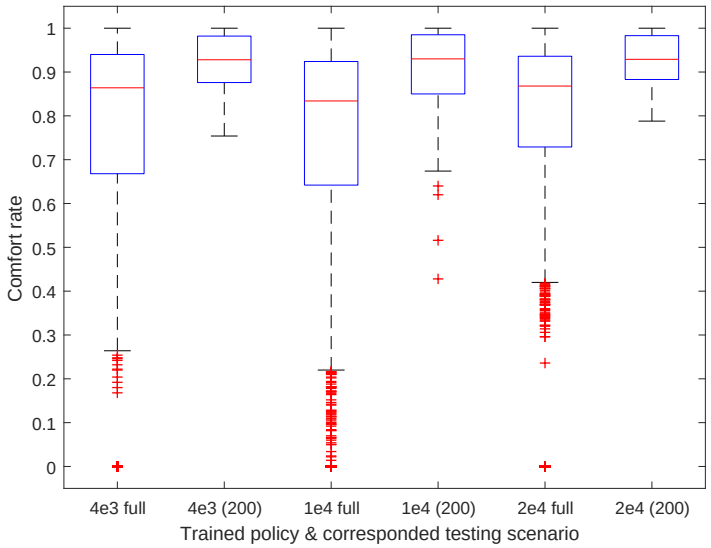

Figure 3.14: Comfort rate results by PPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$

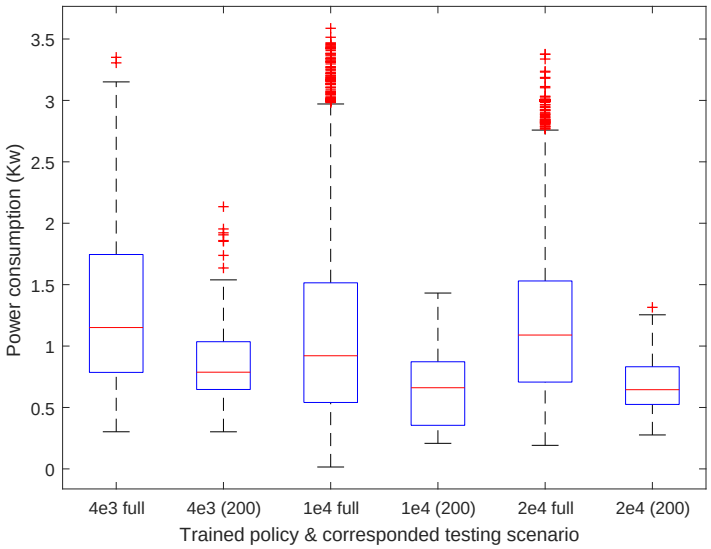

Figure 3.15: Power consumption testing results by PPO policies respectively trained by 4×10^3, 1×10^4 and 2×10^4 episodes, with energy divisor $w_d = 3 \times 10^4$

dicate 1.2871 kW, 1.1236 kW, 1.2049 kW mean powers for the fully tested cases with 4×10^3, 1×10^4, 2×10^4 random start states, and yield mean power values of 0.8713 kW, 0.6686 kW, 0.6866 kW for pre-selected tests. Similar to the TRPO testing results, increasing the

learning trials from 4×10^3 to 1×10^4 or 2×10^4 does not significant improve comfort duration and power efficiency. The PPO control poli-

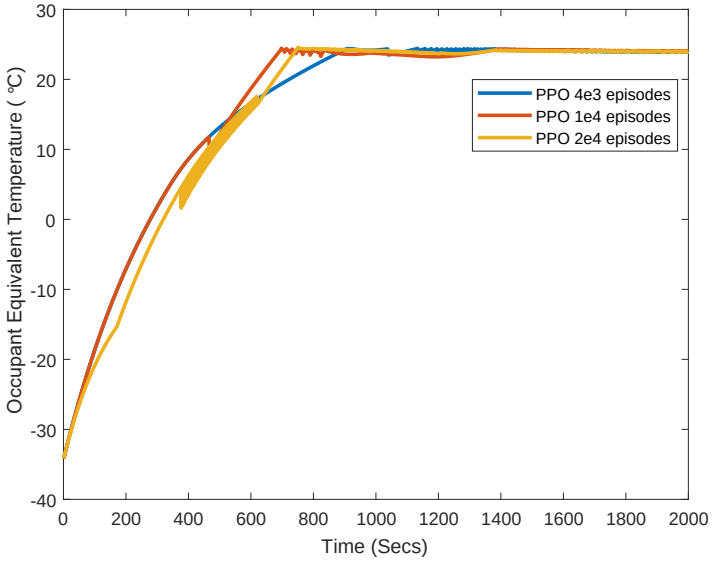

Figure 3.16: Warm-up process of occupant equivalent temperature by PPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 1°C environment, cabin air and interior mass temperature

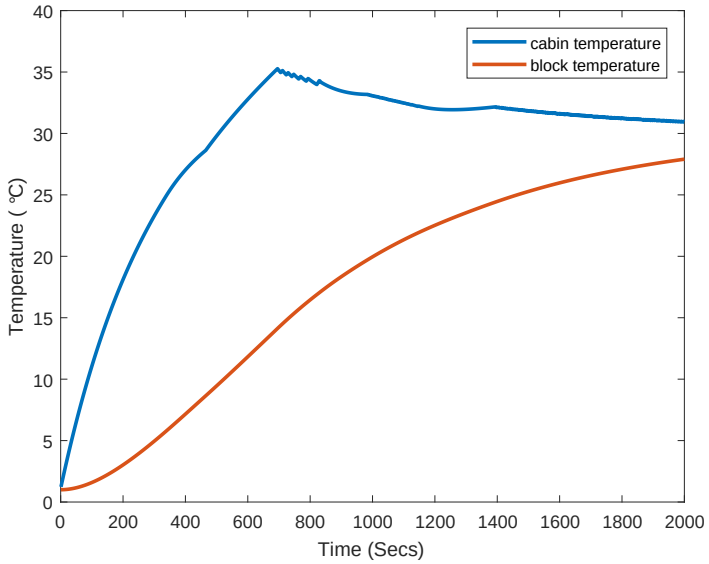

Figure 3.17: Physical temperatures of cabin T_c and interior mass T_m in the warm-up process, trained with 1×10^4 learning episodes

cies are also tested under the circumstances of extreme cold and warm surroundings. Figure 3.16 and Figure 3.18 indicate warm-up and cool-down processes done by PPO policies estimated by 4×10^3, 1×10^4 and 2×10^4 trials. The warm-up process starts with 1°C temperatures

of cabin T_c, interior mass and environment T_{env}. The cool-down case starts with 45°C temperatures of cabin T_c, interior mass T_m and 40°C of environment T_{env}. The equivalent temperature (ET) starts from −33.1°C in the warm-up case, and the ET starts from 55°C in the cool-down case caused by the high airflow rate v_i according to equation 3.10. Figure 3.16 shows the process of warming car cabin to the

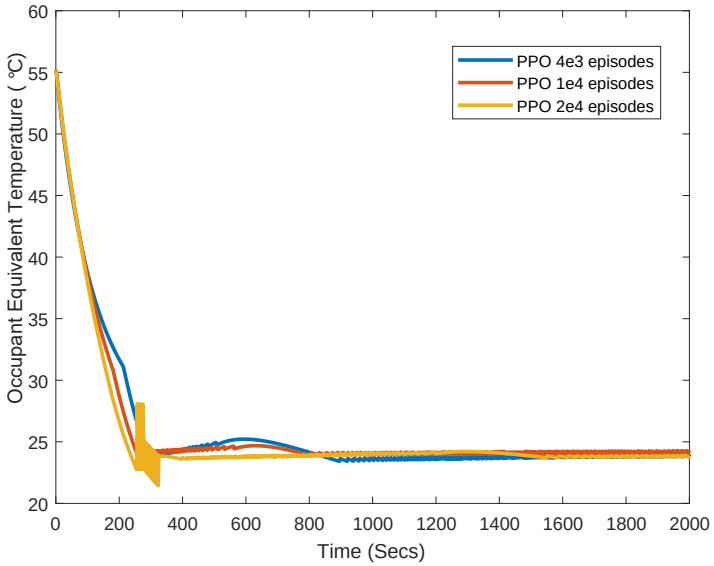

Figure 3.18: Cool-down process of occupant equivalent temperature by PPO control policies (estimated after 4×10^3, 1×10^4 and 2×10^4 trials) under 40°C environment temperature and 45°C cabin air, interior mass temperature

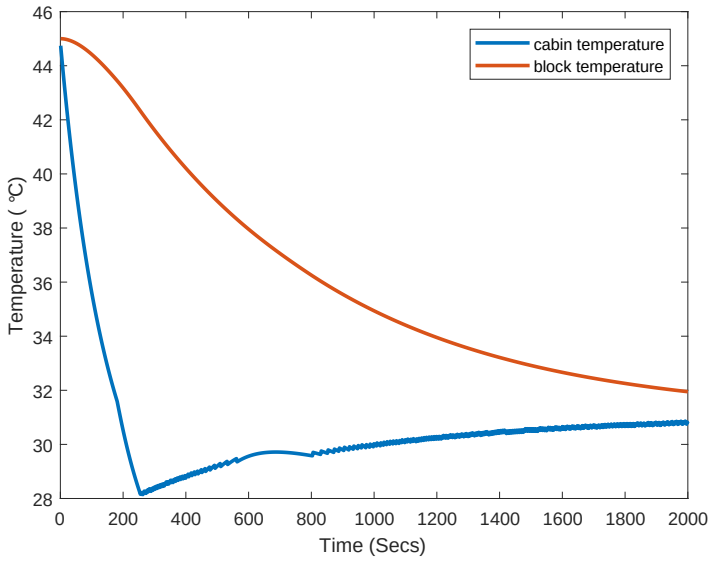

Figure 3.19: Physical temperatures of cabin T_c and interior mass T_m in the cool-down process, trained with 1×10^4 learning episodes

comfort region (ET equals to $24 \pm 1°C$) of occupants, where the warm-up is done by policy estimated from 1×10^4 trial reaches the target in 12.5 minutes, earlier than the cases by 4×10^3 and 2×10^4 trials. Meanwhile, the cool-down in Figure 3.18 shows that ET decreases to the target in around 5 minutes for all resulting policies. Further cabin air and interior mass temperature information are provided in Figure 3.17 and Figure 3.19; these two indicate that the cabin environment is respectively being heat from 1°C to the temperature above 25°C, and being cooled down from 45°C to the temperature around 30°C. It is reasonable for both T_c and T_m above 24°C, because the airflow rate v_i also decides the equivalent temperature (ET) according to equation 3.10. It is clear that increasing trials from 4×10^3 to 2×10^4 only improves the average comfort percentage by 2.64% and reduces power consumption by 0.0822 kW. The extra 1.6×10^4 episodes (each duration is 5×10^3 s) correspond to 8×10^7 s of simulated time, increasing the number of data samples used in the learning process.

Table 3.4 details the averaged episodic comfort percentage and power consumption results in terms of TRPO and PPO-based HVAC policies estimated by learning trials with number 4×10^3, 1×10^4 and 2×10^4. By comparing the results of fully-tested cases, the PPO-based agents generally consume less energy than TRPO ones and averagely maintain longer comfort duration. And the pre-selected cases show that the TRPO agent averagely outperforms PPO with negligible increase of comfort rate.

In the previous section, Figure 3.6 and Figure 3.7 have shown that the PPO-based controller spends 15 minutes on the warm-up task and 6 minutes on the cooling-down task; conversely, the TRPO agent takes 22.5 minutes to warm the cabin and 9 minutes to cool down. This section also compares the best TRPO and PPO policies individually estimated by 4×10^3, 1×10^4, 2×10^4 trials; among these estimates, PPO based controller by 1×10^4 trials have the best performance. The following Figure 3.20 and Figure 3.21 specifically compare the warm-up and cool-down processes done by TRPO and PPO agents (estimated by 1×10^4 trials). These cases have shown that the PPO-based HVAC controller saves 10 and 4.3 minutes in the warm-up and cool-down tests compared to the TRPO one, therefore being more competitive than TRPO based control. Combine this aspect with power consumption and comfort conditioning performance from Table 3.4, PPO-based HVAC controller generally outperforms the TRPO one due to faster control performance in cool-down and warm-up tests and

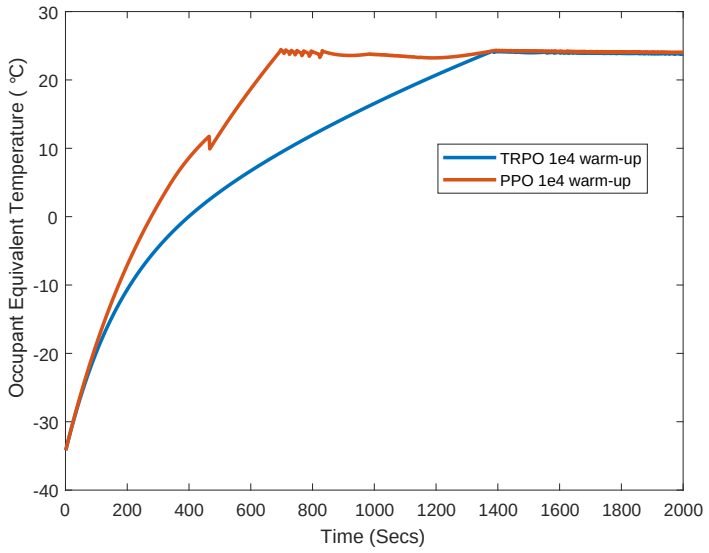

Figure 3.20: TRPO and PPO HVAC policy warm-up performance

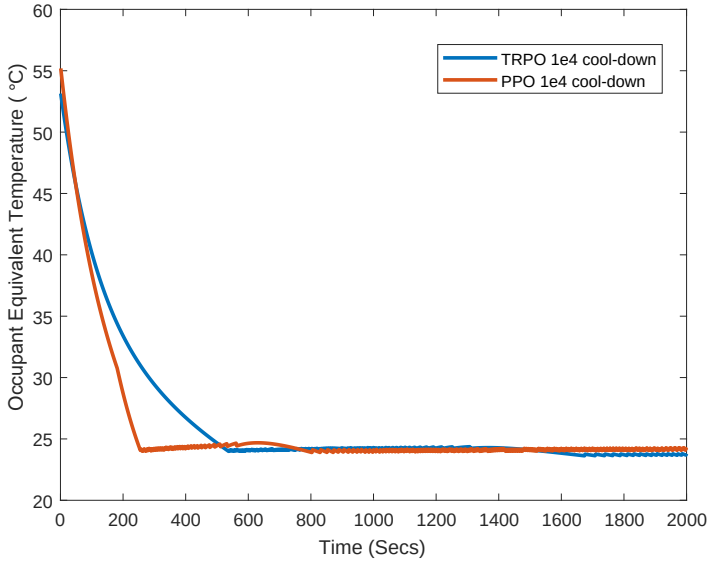

Figure 3.21: TRPO and PPO HVAC policy cool-down performance

almost equally competitive performance in dealing with numerous start states to maintain cabin climate comfort and energy efficiency. More importantly, this section has justified that increasing the number of learning trials over 4×10^3 does not significantly improve the PPO and TRPO based HVAC agents; therefore, using only 4×10^3 episodes can practically estimate optimal HVAC control policies. As mentioned above, each episode time duration is 5×10^3 s, hence 4×10^3 episodes corresponds to simulated times of 2×10^7 s (around 0.63 years), which

is much lower than the times of 2×10^8 s (6.3 years) used to simulate a SARSA based HVAC agent in earlier work [Bru+18].

Table 3.4: Average percentage of time spent in comfort and HVAC power consumption (with time duration of 1×10^3 s) of TRPO & PPO based HVAC control policies by different learning trials

No. of trials	4×10^3		1×10^4		2×10^4	
testing scenario	full	pre-select	full	pre-select	full	pre-select
TRPO comfort(% time)	76.66	93.17	76.43	93.23	76.53	93.23
TRPO power(kW)	1.3316	0.7235	1.2654	0.6539	1.2697	0.6594
PPO comfort(% time)	77.94	92.30	75.67	90.69	80.58	92.73
PPO power(kW)	1.2871	0.8713	1.1236	0.6866	1.2049	0.6866

3.4 CHAPTER SUMMARY

This chapter has presented a set of experiments that extensively investigate the impact of applying the policy gradient reinforcement learning (PGRL) methods to estimate control policies for car cabin heating, ventilation, air conditioning (HVAC). The control policy is required to achieve and maintain occupant thermal comfort while reducing energy cost. The experiments mainly introduce four PGRL methods, which are Monte-Carlo policy gradient (MCPG), mean actor critic (MAC), trust region policy optimization (TRPO) and proximal policy optimization (PPO) to deal with this control.

The results show that the HVAC policies trained by TRPO and PPO generally outperform the MCPG and MAC in terms of maintaining occupant comfort. There are two categories of testing cases: “full” means the initial cabin states are fully selected from its own learning trial, "pre-select" corresponds to 200 pre-selected initial cabin states generated from SARSA based RL for HVAC controller [Bru+18]. For each initial cabin state, the policies are tested through a time step of 10^3 s in order to calculate the percentage of duration that offers comfort to the occupant and the average power consumption.

The results show that policies trained by TRPO and PPO can respectively achieve average percentages of 71% and 78.95% duration spent on occupant comfort. These two are around 40% to 60% higher than the cases achieved by MCPG and MAC methods, and 24% to 26% higher than the SARSA-based approach in pre-selected testing cases. The energy consumptions of PPO and TRPO agent respectively corresponds to 1.30 kW and 1.05 kW under the full testing cases. Still,

MCPG and MAC reduce energy cost to an extremely low level (below 0.15 kW) rather than exploring the chances to achieve comfort.

In the cool-down and warm-up processes, the PPO-based HVAC agents are generally faster in achieving occupant comfort than TRPO. However, both MCPG and MAC-based HVAC controllers fail to achieve occupant thermal comfort (equivalent temperature $24 \pm 1°C$) under cold and warm climate. Furthermore, the number of learning trials for both TRPO and PPO methods are reduced to 4×10^3 (2×10^7 s simulated time), which is equivalent to 10% simulated time of the SARSA-based learning system [Bru+18], and therefore maintains better sample efficiency. The reason primitive PGRL methods (MCPG, MAC) fail to achieve good HVAC control policies is basically due to the fixed learning step size. Because the policy gradient is a non-convex optimization with multiple local optima, having a fixed learning step size can easily get trapped in bad local optima.

This chapter aims to answer both RQ 1.1 and RQ 1.2. For RQ 1.1, the answer is Yes; among all the applied PGRLs, the TRPO and PPO based methods significantly outperform MCPG and MAC in the tests of maintaining comfort, cooling and warming the cabin from extreme hot and cold weather conditions. The TRPO and PPO significantly improve comfort percentage to 77.94% (in Table 3.4) compared with MCPG and MAC-based cases' performance. This means that the occupant's comfort (equivalent temperature equals to 24 ± 1 °C) can be achieved in 3.7 minutes on average (the duration for each testing case is 1×10^3 s). Moreover, the PPO agent is slightly faster than TRPO in both warm-up and cool-down tests by saving 6 and 10 minutes to reach the thermal comfort region while consuming almost the same amount of energy.

The answer for RQ1.2 is Yes because all these optimal policies can be estimated with 4×10^3 episodes (2×10^7 s simulated time, only 10% of the SARSA-based RL), less than earlier work done by the SARSA-based RL. It is also clear that increasing the number of trials over 4×10^3 does not significantly improve PGRL HVAC agents' performance in comfort and energy efficiency.

The next chapter describes how the cabin state is represented in a Non-Markov decision process when using a time-dependent model to sample episodes and introduces a method to mitigate the time-dependence to represent the states to satisfy the Markov decision process.

4

A MARKOVIAN STATE REPRESENTATION FOR LEARNING HVAC CONTROL

The decision tasks solved by reinforcement learning (RL) based techniques are usually described as Markov decision processes (MDPs), which indicate the agent's observing state and receiving reward as the output of an environment, taking action then observing the next state and reward. Alternatively, such agent-environment interactions are defined to satisfy the Markov property; namely, future observations are only conditioned on the current state. The MDPs are useful for modelling a wide range of control tasks, such as the traditional pendulum-balancing [LYB07] and multi-robot patrolling management [PR13]. However, the agents in some tasks observe inadequate information from the state, thus unable to identify the state. These are known as non-Markov decision processes (NMDP) [WL95], which also widely exist in real-world applications. A typical example is the helicopter control with inaccurate position data provided by the sensors [Ng03].

Usually, the control of the MDP-based task can be effectively trained, as the agent decides action by only observing the current state. As for the vehicle HVAC control task in this thesis, the state information includes temperatures of cabin, block (interior mass), ambient (environment), and airflow rate. The action comprises vent air temperature (heating or cooling), fan speed and recirculation flap position. According to prior simulation results with different episode termination time, we can observe variations of estimated state-action values, which is the probability distribution over possible next states seems to depend on the information of the entire history of states. Therefore, the cabin state can be represented in a Non-Markov decision process as the amount of time passed does matter. To present the impacts of non-Markovian state representation in training PGRL HVAC controller, this chapter focus on subsequent research questions:

RQ2.1 Is the learning performance of PGRL HVAC negatively impacted by a non-Markovian cabin state representation?

RQ2.2 Can the Markovian state representation improve the energy efficiency by using the same number of training experience in a non-Markovian state representation?

These questions aim to investigate how the cabin state is represented in a non-Markov decision process (NMDP) and what negative impacts the NMDP can have on the PGRL HVAC learning system. Then introducing a Markov state representation for training PGRL HVAC controller, comparing the comfort and energy performance with non-Markov represented cases.

4.1 NON-MARKOV DECISION PROCESS OF PGRL HVAC SYSTEM

4.1.1 *Problem statement*

As mentioned above, the MDP intuitively represents a control task. At each time point, the agent directly observes the state of the environment and the effects of actions depend only upon the action and current state. For instance, having a sequence of observations received from time point 0 to $t-1$: $\{s_0, s_1, \cdots, s_{t-1}\}$ where $t > 2$. The Markov property allows distribution of X_t given the entire history $s_0, \cdots, s_{t-1}$ of the past only depends on the immediate past state s_{t-1}:

$$P(X_t = s_t \mid X_{t-1} = s_{t-1}, \cdots, X_0 = s_0) = P(X_t = s_t \mid X_{t-1} = s_{t-1})$$

When executing actions, the state and reward received at point t only depend on observation and action at $t-1$, thus the distribution of state s_t and reward r_t given the whole history of states and actions $s_0, a_0, s_1, a_1, \cdots, a_{t-1}, s_{t-1}$ only depends on immediate state s_{t-1} and action a_{t-1}

$$P(s_t, r_t \mid s_{t-1}, a_{t-1}, \cdots, s_1, a_1, s_0, a_0) = P(s_t, r_t \mid s_{t-1}, a_{t-1})$$

This property also indicates that events beyond s_t are independent of past states and actions $\{s_{t-1}, a_{t-1}, \cdots, s_0, a_0\}$ and how much time has passed does not influence the probability distribution of possible next states.

Conversely, some real-world control tasks are non-Markov due to insufficient information in identifying the states. This issue is caused by the fact that the agent's sensor cannot fully observe the whole information from the current state of the environment [WL95]. As a typical class of non-Markov decision processes (NMDP), the partially observable MDP (POMDP) is widely applied in modelling hidden state problems, including machine maintenance, elevator control, space navigation [Cas98]. These are typical cases in which the agent can

only observe and collect part information of the actual state from the environment. The ways of validating whether specific problem models hold the Markov property are either based on statistical analysis or prior knowledge of the model because Markov property usually comes into a model as an assumption [Ros14].

4.1.2 *The NMDP and the termination time*

When using a fixed termination time for the cabin simulation, the problem is non-Markovian. To explain this, consider that the Markov property says that the state and action (combined) contain enough information to determine the probability distribution over possible next states. In other words, no other information, such as the entire history of states, would provide a better (or even different) estimate of the next state probability distribution. This can be stated as

$$P(s_{t+1} \mid s_t, a_t) = P(s_{t+1} \mid s_t, a_t, \cdots, s_0, a_0) \tag{4.1}$$

The Markov property also means that how much time has passed does not matter. For two arbitrary trajectories that pass through the same state $s_u = s_t$ and action $a_u = a_t$, the resulting distribution over next states is the same,

$$P(s_{u+1} \mid s_u, a_u) = P(s_{t+1} \mid s_t, a_t) \tag{4.2}$$

To see that this is true, consider a process that enters state s_t but then stays there for several time steps. Logically, the history makes no difference to the next state distribution regardless of how many time steps pass, if the Markov property holds.

On the other hand, for the cabin model with a fixed termination time, the next state distribution changes for the last time step to be 1 for the absorbing or terminal state and zero elsewhere. Clearly, the amount of time passed does matter and thus this is a non-Markov decision process. When applying a reinforcement learning algorithm to such a non-Markov problem, a correct estimate of the utility of a state action combination will tend to vary depending on the episode time, even for deterministic chains. To see this, consider a discrete time chain with state space $\{0, \cdots, 10\}$, a reward function $R(10) = 1$ and zero elsewhere, and actions $-1, 0, +1$. Assuming an optimal policy and with a time limit of 9, the utility $Q(0, +1)$ might be estimated as 1 at time zero but 0 at any time after this. Thus it can be seen

that a time limit can affect the utility function. The variation will not necessarily change which policy is considered optimal but it may cause the algorithm to converge to that policy more slowly.

4.1.3 *Experiment setting*

To validate the assumption of non-Markovian cabin thermal model, the following experiment examines the learning outcomes by using different durations to generate corresponded sequences (episodes) of states, actions and rewards for the PPO learning scheme (an efficient and effective PGRL method). The episode duration is set to $\{1 \times 10^3, 2.5 \times 10^3, 4 \times 10^3, 5 \times 10^3$ s. The learning outcome is evaluated by the cumulative sum of rewards received in each learning episode, and rate of cabin states that satisfy occupant comfort.

4.1.4 *Results and analysis*

The following graphs illustrate the learning outcomes of respectively using 1×10^3 s, 2.5×10^3 s, 4×10^3 s and 5×10^3 s as the episode durations. Figure 4.1 and Figure 4.2 demonstrate the episodic reward against the learning trials with corresponded episode duration. As

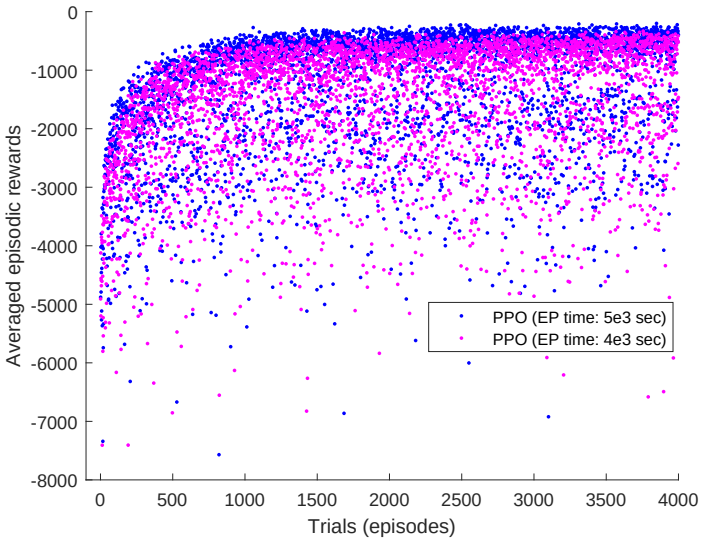

Figure 4.1: Averaged episodic rewards with episode duration $T = 5 \times 10^3$ s and $T = 4 \times 10^3$ s against learning trials

mentioned in section 3.3.1 from Chapter.3, the "episodic reward" defines the sum of rewards received in a single episode. The proximal policy optimization (PPO) is employed as the training method, the

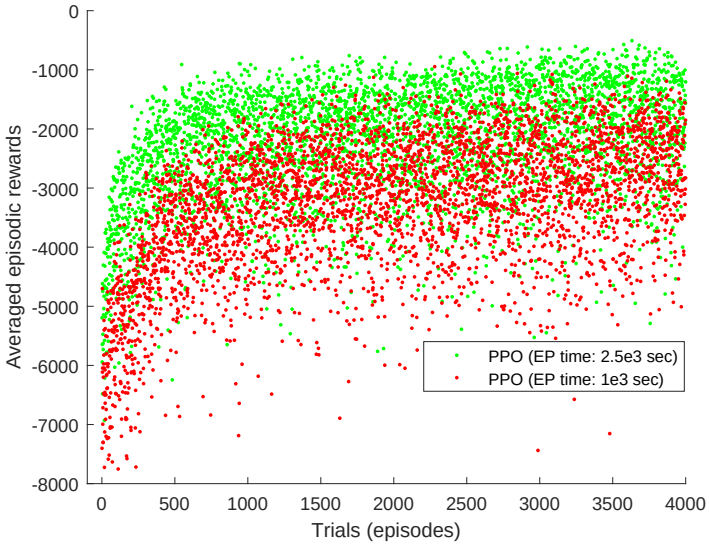

Figure 4.2: Averaged episodic rewards with episode duration $T = 2.5 \times 10^3$ s and $T = 1 \times 10^3$ s against learning trials

learning trial for each case is averaged over the results of using ten different random seeds. Both Figure 4.1 and Figure 4.2 show that increasing the episode duration from 1×10^3 s to 5×10^3 s can significantly improve the episodic rewards in the learning processes. For example, the learning case with episode duration 2.5×10^3 s achieves averaged episodic reward -2007.1, which is higher than the one by 1×10^3 s duration with averaged reward -2914.6. The averaged episodic reward achieved by 5×10^3 s and 4×10^3 s approximately correspond to -1059.5 and -1297.6. It is clear that the longer episode duration case is likely to achieve higher episodic rewards.

The comfort rate denotes the proportion of time spent on maintaining occupant thermal comfort. The Figure 4.3 illustrates comfort rate against learning trials with corresponded episode durations. Like the episodic reward cases, the increasing episode duration T leads to improvement in maintaining the occupant comfort. The training case with 5×10^3 s episode duration indicates the capability of learning HVAC control policies that can steadily achieve around 90% time spent on occupant comfort. As the duration T decreases, the learnt HVAC control policies tend to achieve a lower occupant comfort rate. The details of averaged comfort percentage achieved by 5×10^3 s and 4×10^3 s approximately correspond to 89.15%, 82.62%, 67.25% and 48.66%. The randomly selected initial cabin states cause the variations of episodic rewards and comfort rate in the learning processes.

According to the comparisons of episodic rewards and comfort maintaining capability, the episode duration determines how good

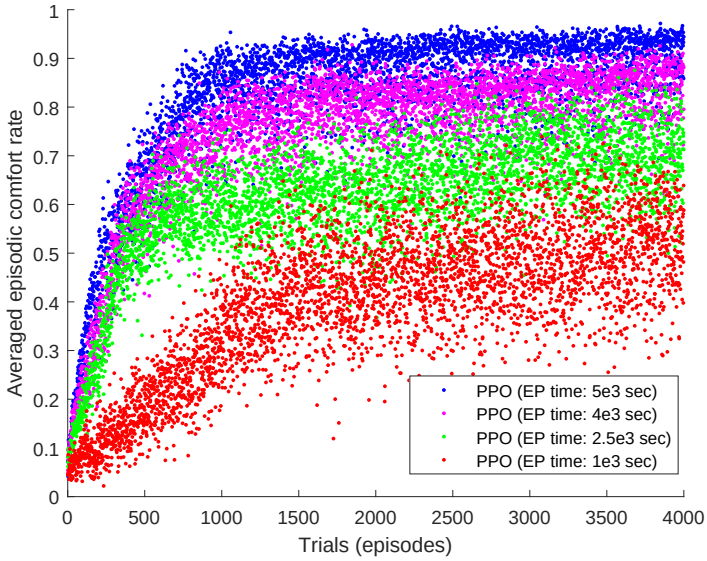

Figure 4.3: Episodic comfort rate against learning trials with episode duration $T = \{1 \times 10^3, 2.5 \times 10^3, 4 \times 10^3, 5 \times 10^3 \quad s$

the control policy is learnt after a certain number of learning trials. Subsequent testing cases examine the comfort keeping and energy consumption performance of control policies trained by 4000 episodes with durations ranging from 1×10^3 to 5×10^3 s. Figure 4.4 and Figure 4.5 present the comfort rate and power consumption testing results for the TRPO and PPO policies trained with the corresponded episode duration (time limit). In terms of comfort keeping performance, the box plot by Figure 4.4 shows an ascending comfort percentage as the training episodes' duration increases from 1×10^3 to 5×10^3 s. The averaged comfort percentages by PPO-based controller generally exceed the corresponded performance by TRPO-based one. According

Table 4.1: Averaged comfort percentage and power consumption of TRPO & PPO based HVAC control policies trained under different learning episode durations (1×10^3s to 5×10^3s)

Episode duration (Secs)	1×10^3s	2.5×10^3s	4×10^3s	5×10^3s
TRPO % Time in comfort	11.71	16.16	23.36	71.53
PPO % Time in comfort	27.44	49.39	55.61	78.95
TRPO average HVAC power (kW)	0.333	0.651	0.634	1.3012
PPO average HVAC power (kW)	0.613	0.636	0.901	1.0484

to the comfort percentage details in Table 4.1, the ascending episode duration improves the comfort maintaining performance. Meanwhile, the energy consumption increases as the episode duration increases from 1×10^3 to 5×10^3 s. Because more energy is needed to maintain thermal comfort in the extra time. Therefore, the episode duration

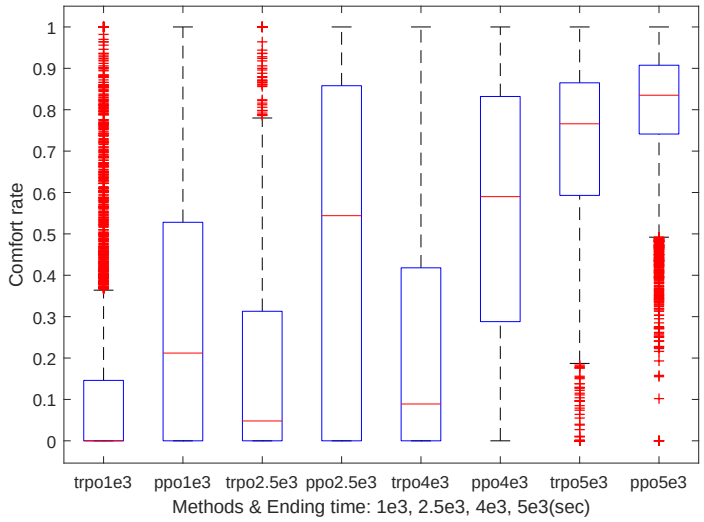

Figure 4.4: Comfort testing performance by TRPO & PPO policies trained by 4×10^3 episodes with durations ranging from 1×10^3 s to 5×10^3 s; each case includes 4×10^3 random initial states

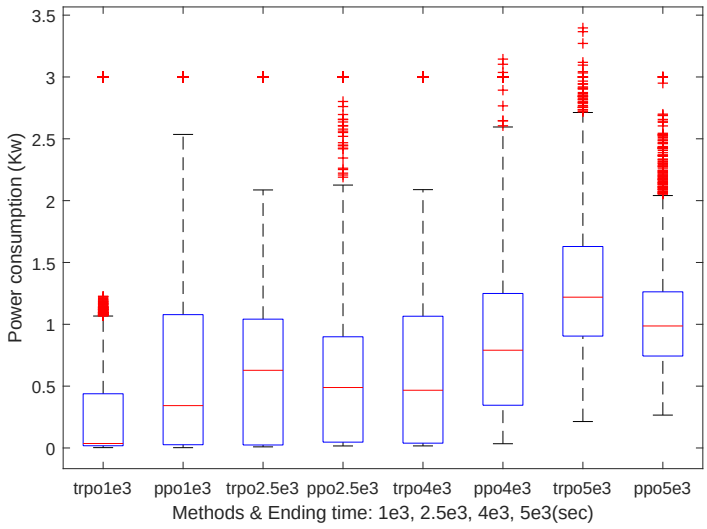

Figure 4.5: Power consumption testing performance by TRPO & PPO policies trained by 4×10^3 episodes with durations ranging from 1×10^3 to 5×10^3 s; each case includes 4×10^3 random initial states

can determine the PGRL-based HVAC control policy's performance; namely, an episode's fixed time limit needs to be long enough to better estimate the control policy.

4.2 MDP-BASED MODEL FOR PGRL HVAC SYSTEM

How do we change problem so that it has the Markov Property? The answer is that we remove the time-limit and instead add to every

non-terminal state a constant probability ν of jumping to the terminal state, such that,

$$\nu = P(\text{terminal} \mid s) \tag{4.3}$$

for all $s \in S$. It is easy to see that this resolves the problem introduced by the time-limit since no additional history is needed to determine whether this jump to the terminal state should take place. Another way to see this is that the probability of jumping to the terminal state is the same regardless of the time.

The expected length (duration) of an episode will correspond to the mean of an exponential distribution with parameter ν. Thus the expected episode length is simply $1/\nu$. [Wik21]

4.2.1 *Results of MDP-based training and testing*

The following two graphs Figure 4.6, Figure 4.7 respectively demonstrate the averaged episodic reward and comfort rate against learning trials. And each case is averaged over multiple trails of using ten different random seeds. Over the 4000 trials, MDP training case achieves

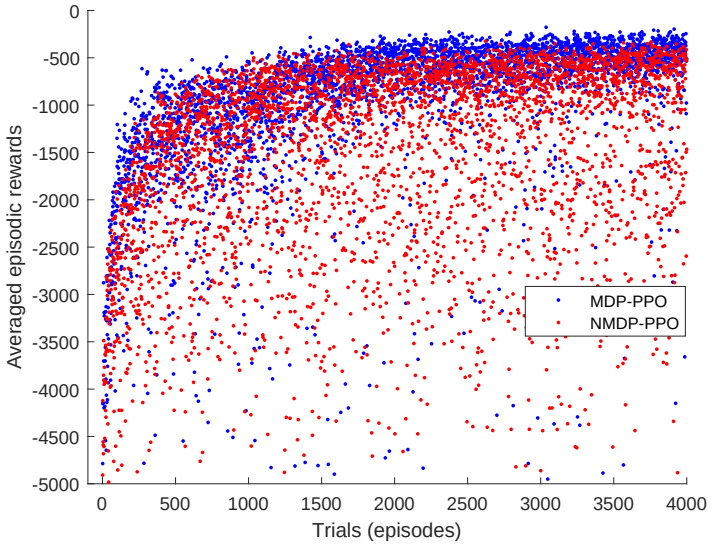

Figure 4.6: Averaged episodic rewards against learning trials by MDP and NMDP state representation, with an averaged episode duration 4×10^3 s

averaged episodic reward -796.04, which is higher than the NMDP one's averaged episodic reward -1263.7. In terms of the comfort rate performance, due to application of ending probability $P(\text{terminal} \mid s)$, the MDP case shows higher variance than NMDP's case. But the com-

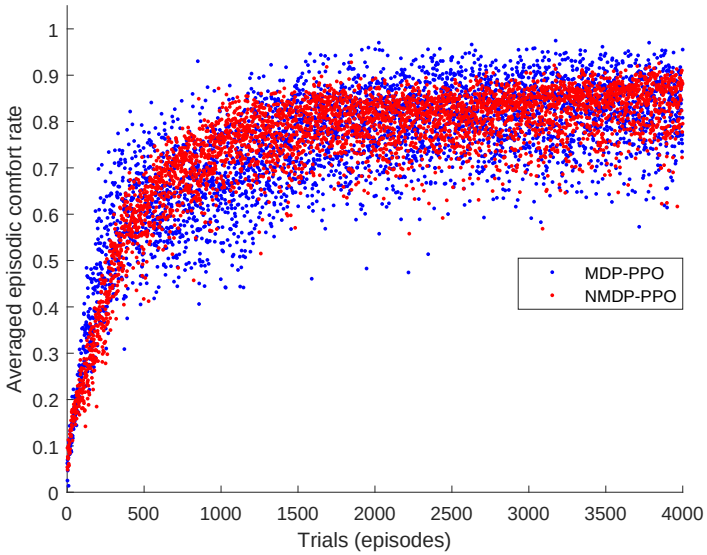

Figure 4.7: Averaged episodic comfort rate against learning trials by MDP and NMDP state representation, with an averaged episode duration 4×10^3 s

fort percentages that the MDP and NMDP training cases averagely achieve are corresponded to 81.67% and 82.62%. Also, the testing

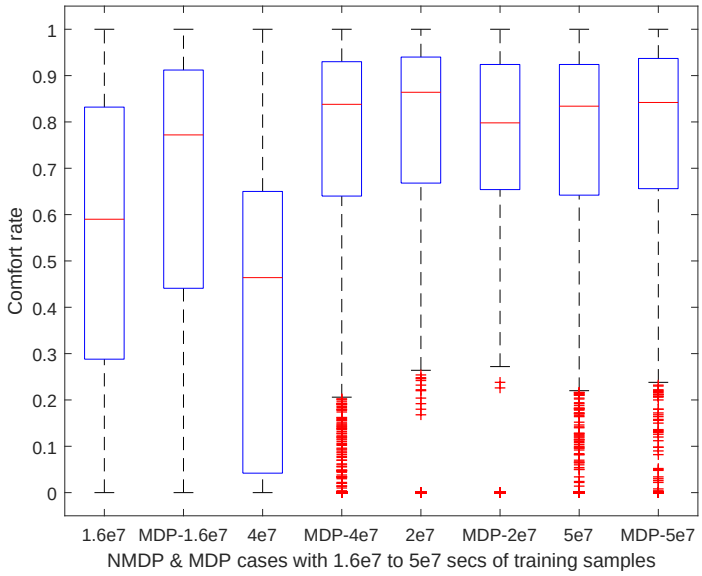

Figure 4.8: Comfort testing performance by PPO policies trained by Markov and non-Markov represented state with total simulated training time $\{1.6 \times 10^7, 4 \times 10^7, 2 \times 10^7, 5 \times 10^7,$ secs

scenarios focus on comparing the performance of resulting HVAC control policies respectively trained with Markov and Non-Markov representations. Moreover, the number of learning trials (episodes) is selected from $\{4 \times 10^3, 1 \times 10^4$ and averaged episode duration from

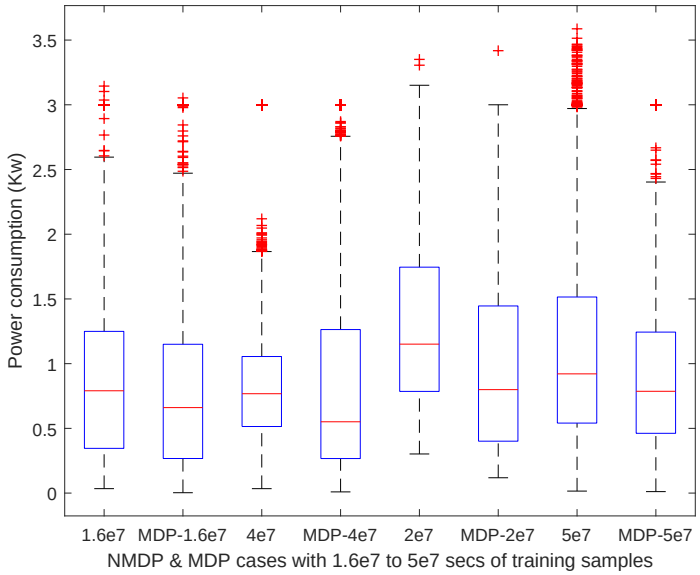

Figure 4.9: Power consumption testing performance by PPO policies trained by Markov and non-Markov represented state with total simulated training time $\{1.6 \times 10^7, 4 \times 10^7, 2 \times 10^7, 5 \times 10^7,$ secs

$\{4 \times 10^3, 5 \times 10^5$ secs, hence yielding the total simulated training time ranges from 1.6×10^7 to 5×10^7 secs. Based on the training settings above, Figure 4.8 and Figure 4.9 respectively show the comfort maintaining and energy consumption testing performance of the resulting HVAC control policies. The definition of labels, for example, "1.6e7" and "MDP-1.6e7" correspond to the HVAC control policies trained with Non-Markovian and Markovian state representation using 1.6×10^7 s of training samples.

The results of comfort test have shown that the Markov representation significantly improve the comfort rate of control policies trained with Non-Markov representation of 1.6×10^7 and 4×10^7 simulated time. The comfort rate is improved by 10.72% and 31.39% respectively. However, the Markov representation slightly drives down the comfort rate of the control policies trained with 2×10^7 and 5×10^7 simulated time. In terms of power consumptions, the application of Markov representation can generally reduce HVAC system's energy cost while achieving high comfort percentage. According to the results detailed in Table 4.2, the average power consumptions are reduced below 1.0 kW for the MDP-based training cases, although the comfort maintaining performance of HVAC control policies trained with 4×10^3 s average episode duration is lower than the case with 5×10^3 s.

To examine the resulting HVAC control policies in terms of dealing with cold and hot cabin climate conditions, the warming-up and

Table 4.2: Averaged comfort percentage and power consumption of PPO based HVAC control policies respectively trained under Non-Markovian and Markovian representations with total simulated training time of 1.6×10^7s, 4×10^7s, 2×10^7s, 5×10^7s

Average episode durations (Secs)	4×10^3 s		5×10^3 s	
No. of learning episodes (trials)	4×10^3	1×10^4	4×10^3	1×10^4
NMDP-PPO average % time in comfort	53.58	40.69	**77.94**	**75.67**
MDP-PPO average % time in comfort	**64.32**	**72.08**	75.88	73.34
NMDP-PPO average HVAC power (kW)	0.8919	0.8470	1.2871	1.1236
MDP-PPO average HVAC power (kW)	**0.8200**	**0.8437**	**0.9901**	**0.9300**

cooling-down processes are made to show the time taken to achieve the target equivalent temperature ($24 \pm 1°C$). Four control policies are respectively trained with MDP and NMDP state representations by using 1.6×10^7 and 2×10^7 s of simulated time. Figure 4.10 illustrates the warm-up cases done by these four policies, the MDP-based ones achieve target equivalent temperature (ET) faster than the NMDP ones, individually saves 5.7 and 6.9 minutes. For the NMDP case trained with 1.6×10^7 s of simulated samples, ET variations observed at early stages are resulted from the alternating control actions of low and high vent airflow rate (v_i is selected from 1 and 67 $l\,s^{-1}$ alternatively). Accordingly, Figure 4.11 indicates the cool-down processes where the policy trained with MDP representation and 2×10^7 s of simulated time can maximally drive down the time spent on achieving the target ET. Combined with the testing cases using 2×10^7s of simulated time in the box plot of Figure 4.8, these cool-down and warm-up results indicate that the application of MDP-based simulation does not significantly improve the comfort maintaining performance in the NMDP case. Therefore, the Markov represented PGRL HVAC system can generally reduce power consumption, achieve high comfort rate and satisfy cool-down, warm-up requirements.

4.3 CHAPTER SUMMARY

This chapter has presented a set of experiments focusing on investigating how the Markovian state representation impacts the process of using policy gradient reinforcement learning (PGRL) to estimate of a vehicle cabin's heating, ventilation, air conditioning (HVAC) system. These experiments firstly validate the presence of Non-Markov property due to a time-dependent cabin state representation of the HVAC system, secondly introduce a random ending-time event to

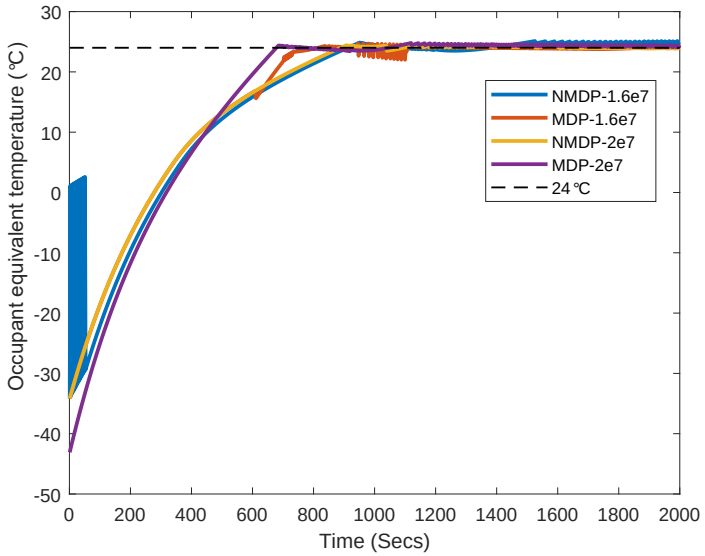

Figure 4.10: Warm-up process of occupant equivalent temperature by using PPO policies respectively trained by Markov and non-Markov state representation with 1.6×10^7 s, 2×10^7 s of simulated training time, starting from the environment, cabin air and mass temperature in 1°C

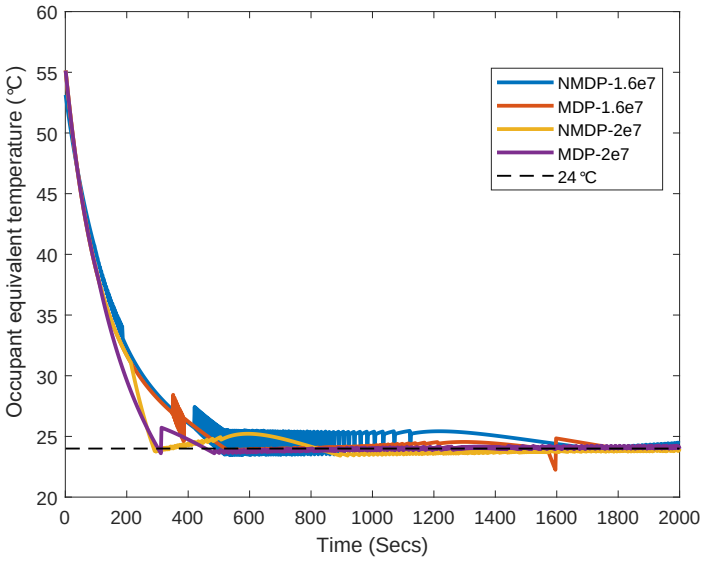

Figure 4.11: Cool-down process of occupant equivalent temperature by using PPO policies respectively trained by Markov and non-Markov state representation with 1.6×10^7 s, 2×10^7 s of simulated training time, starting from the environment temperature 40°C, cabin air and mass temperature 45°C

mitigate the time-dependence of episode states so as to represent the states in a way that satisfies Markov property. The learning and testing performance of the PGRL HVAC system are compared with respect to Non-Markovian and Markovian representation of states.

The validation results show that increasing episode durations (time steps) can improve the rewards and comfort percentages, namely the longer duration of each learning episode can improve the control performance in achieving higher percentage of occupant comfort, hence the control task is dependent on episode time steps. To mitigate this impact, the random ending-time event enables episode states to terminate at arbitrary time steps rather than using fixed ending-time. This change also helps to represent the states to fulfil the Markov decision process (MDP). The comparison results indicate that this MDP representation helps to individually reduce the average power consumptions from 0.8919 kW to 0.8200 kW and 1.2871 kW to 0.9901 kW for the training cases with 1.6×10^7 and 2×10^7 secs of total simulated time. Correspondingly, the comfort percentages are 64.32%

This chapter aims to answer research questions 2.1 and 2.2. The answer for RQ 2.1 is Yes, because the episode duration can impact comfort control performance of the resulting policy, for example, longer episode duration helps to improve the comfort percentage. And by relevant definition of non-Markov decision process, the fixed episode duration is the extra information or knowledge that can be used to improve this HVAC control task, hence this case is non-Markovian.

The answer for RQ 2.2 is Yes, because the application of MDP represented states can generally reduce the average power consumption over the training cases with 1.6×10^7 and 2×10^7 secs of total simulated time.

5

POLICY GRADIENT REINFORCEMENT LEARNING FOR A SIMULATION BASED ON CLIMATIC WIND TUNNEL EXPERIENCE

The previous chapters focus on the policy gradient reinforcement learning (PGRL) applications in a transient thermal model based on the linear approximation technique. This thermal model imitates a simple car cabin thermodynamic, which approximates the cabin air temperature (AT) and interior mass temperature over time. However, inside a car cabin environment, the cabin relative humidity, radiant heat exchange, cabin air temperature, airflow and clothing type also impact occupants' thermal comfort experience. As the occupants' body constantly exchanges thermal radiation with the cabin environment, the occupants' body surface temperature will be non-uniform and vary depending on factors such as clothing area, airflow, surrounding climate, relative humidity and metabolic heat. A mean radiant temperature (MRT) measurement is then introduced to denote the average surface temperature of the occupants' body sections, such as head, torso and feet. In a real car cabin environment, the MRT can accurately indicate occupants' body surface temperature rather than using an interior mass temperature in chapter 3. Based on the data collected in a climatic wind tunnel experiment [BR20], a linear regression model is then used to approximate the occupant-oriented car cabin thermodynamics. This chapter focuses on the PGRL HVAC comfort control application in this climatic wind tunnel simulation model, and compares its comfort conditioning performance with the bang-bang control technique. The challenge for the RL HVAC system is to identify the MRT, AT and airflow of occupants' body sections to achieve and maintain comfort. This chapter investigates the subsequent research question:

RQ 3: Can the PGRL-based HVAC controller reduce the time taken and power consumption to achieve occupant thermal comfort in a climatic wind tunnel simulation model compared to a bang-bang method?

Section 5.1 details the state and control information used in PGRL framework, then lists the linear regression model as cabin thermal simulator. Section 5.2 presents the control policy network and bang-bang HVAC control algorithm. Section 5.3 details initial state settings, equiv-

alent temperature and related reward function. Finally, section 5.4 shows the testing cases, including warm-up, cool-down, and examines the comfort maintaining and power consumption results.

5.1 THE INFORMATION OF STATE, CONTROL ACTION AND THERMAL MODEL

The subsequent list indicates the variables that can be observed by the reinforcement learning (RL) agent in a single state. The observable information includes measurements from both the cabin and environment.

- Air temperatures of front driver and passenger's head, torso and foot locations (noted as T_{dh}, T_{dt}, T_{df}, T_{ph}, T_{pt}, T_{pf} in °C)
- Mean radiant temperatures measured at driver and passenger's head, torso and foot locations (noted as M_{dh}, M_{dt}, M_{df}, M_{ph}, M_{pt}, M_{pf} in °C)
- Temperature of the windshield at driver's side (T_{wd} in °C)
- The relative humidity inside the car cabin (ϕ_c)
- Air velocities ($m\,s^{-1}$) measured at driver and passenger's head, torso and foot locations (noted as v_{dh}, v_{dt}, v_{df}, v_{ph}, v_{pt}, v_{pf})
- The external cabin roof temperature T_r
- The level of ambient relative humidity ϕ_{env} (environment outside the car)
- Ambient air temperature T_{env}
- The temperatures of the dashboard surface at the driver, passenger, and the central locations (noted as T_{dd}, T_{dp} and T_{cd})
- Car velocity V_{car} ($km\,h^{-1}$)

In total, a state comprises 27 observable variables ranging from air and mean radiant temperatures to car velocities, these consist a vector known as the state or observation

$$s = \left[T_{dh}, \cdots, T_{dt}, M_{dh}, \cdots, M_{pf}, \cdots, V_{car}\right] \tag{5.1}$$

The RL agent observes the state information above, then chooses control actions to condition the cabin climate. These control strategies

include adjusting the vent air temperatures, blower rotation speed, recirculation rate and defrost option. Other factors, such as car velocity and cabin roof temperature, also influence the thermal status inside the cabin. The experiment control options which can be decided by the HVAC system are as follows

- The electric current flow in the blower I_b
- The fresh / recirculation option A_{fr} $[0, 1]$(where "0" is for fresh air, "1" is for recirculation)
- Distribution setting (defrost or neutral option) A_{dist} $[0, 1]$ ("0" for neutral and "1" for defrost)
- The vent air temperatures for the driver and passenger: located at driver's central and left side, passenger's central and right side (noted as T_{vdc}, T_{vdl}, T_{vpc}, T_{vpr})
- The air temperatures at driver and passenger's floor (noted as T_{adf} T_{apf})
- The air temperature near the HVAC recirculation inside the cabin T_{rc}
- The air temperatures of the HVAC vent ducts for both driver and passenger sides (noted as T_{vd} and T_{vp})

where the electric current of the blower I_b determines the blower's fan rotating speed to control the rate of vent air flow for the occupants. Other control options, such as vent air, recirculation, and HVAC duct temperatures indirectly affect the air and mean radiant temperatures in the occupants' space. An HVAC control vector comprises twelve variables ranging from electric current I_b to vent duct temperature T_{vp} mentioned above

$$z = \left[I_b, A_{fr}, A_{dist}, T_{vdc}, \cdots, T_{vpr}, \cdots, T_{vd}, T_{vp}\right] \tag{5.2}$$

As for the car cabin thermal model, prior research [BR20] uses a linear function to imitate the thermodynamics model of the car cabin. Several sets of real time data (the state, control measurements indicated above) are used in the training and validations for the resulting cabin thermal simulator, and the mean square error is limited below 3%. This function estimates cabin thermal state in terms of prior observations

and control inputs, the subsequent equation indicates the recurrent state approximation process

$$X_{t+1} \approx f(X_{t-1}, X_t, U_t, w) \tag{5.3}$$

where the vector X comprises 23 variables of the state s (equation 5.1 above), excluding these four variables: car velocity V_{car}, cabin roof temperature T_r, ambient air temperature T_{env} and ambient relative humidity ϕ_{env}. These four measurements are beyond the control range of the HVAC system, however, the ambient air temperature and solar radiation still affect the temperature distribution inside the top section of the car cabin. For example, intense solar radiation can significantly drive up the cabin roof temperature and the surface temperature of the front dashboard. So, the control input vector U includes these four environmental factors (V_{car}, T_r, T_{env}, ϕ_{env}) and the HVAC control options listed above (ranging from I_b to T_{vd}). And w represents the weight parameter of this linear function. The previous cabin thermal state X_{t-1} preserves certain inertia strength to affect the current cabin thermal state X_t. For instance, an increasing airflow in the previous observation tend to increase for the next short period, although current control input U_t is changed. This model is then named as "CWT-cabin-env" simulator for the following experiment results sections.

5.2 RL CONTROL POLICY AND BASELINE CONTROL IMPLEMENTATION

As the control actions are selected from multiple discrete spaces indicated in Table 5.1, the multilayer perceptron (MLP)-based policy network $\pi_\theta(a \mid s)$ is then fully connected with multiple output layers. Still, the input states are estimated by the thermal model above. Compared to the setting in chapter 3, the only difference is that the output layers need to be compatible with the scales of different discrete control spaces. For example, the policy respectively estimates action distributions for blower amperage I_b and vent air temperature T_{vdc}, and according to Table 5.1 below, the number of units for output layer is 7 and 26. Hence, a 1×7 and a 1×26 scaled output layers are

needed. This MLP neural network $\phi(s)$ (parameterized by θ) applies a softmax equation to calculate all possible action distributions

$$\pi_\theta(a_j^i \mid s) = \frac{\exp\left(\phi_j^i(s)\right)}{\sum_{k=1}^{N_j} \exp\left(\phi_j^k(s)\right)} \tag{5.4}$$

where a_j^i refers to the i^{th} action at the j^{th} discrete space, and the value for j can be chosen from 1 to 12. The optimization process is to update weight θ by maximizing expected rewards or advantage value functions resulted from actions selected by policy π_θ. So the policy gradient process essentially estimates the gradient of policy parameter $\Delta\theta$

$$\Delta\theta \leftarrow \mathbf{E}\left[\sum_{a\sim\pi}^{\infty} \nabla_\theta \log(\pi_\theta(a \mid s)) \cdot A^\pi(a, s)\right] \tag{5.5}$$

and according to previous chapters, the proximal policy optimization (PPO) is chosen as the main reinforcement learning approach. Apart from the scale of input/output layers, other hyperparameters of the PPO are kept with the same values in chapter 3 (Table 3.2). Subsequent Table 5.1 lists discrete spaces for all control variables, the number of discrete values indicates the number of elements in this discrete space. For example, the space for blower amperage I_b has seven values uniformly sampled in space $[0, 12]$, hence obtaining a finite space $\{0, 2, 4, 6, 8, 10, 12\}$ with seven values. So, the outputs of control policy $\pi_\theta(a_j^i \mid s) \mid_{j=1, i=\{1,2,\cdots,7\}}$ correspondingly denote distributions of selecting I_b from this discrete space $\{0, 2, 4, 6, 8, 10, 12\}$. The advantage of using multiple discrete space here is that this can reduce the number of output layer nodes to 295. If using the combination of multiple discrete spaces, the number of output layer nodes will exceed a million.

As a common technique applied in commercial thermostats, the bang-bang method is a simple but effective control solution to achieve and maintain a desired air temperature. A simple bang-bang thermal controller is based on temperature feedback to switch control signal between two control actions. For example, switching between electric heating power on and off states when the temperature is lower or higher than a certain setpoint. To avoid too rapid value changes in the control outputs, double set-points are introduced in a bang-bang control. For example, if the desired temperature is 20 °C, the thermostat can turn the heater on when temperature drops below 18 °C, but not

Table 5.1: The HVAC control variables and corresponded discrete spaces (V.A.T refers to vent air temperature)

Control variable	Min	Max	No. of discrete values
I_b blower amperage	0 A	12 A	7
A_{fr} fresh/recirculation	0	1	2
A_{dist} neutral/defrost	0	1	2
T_{vdc} V.A.T at driver's central side	0 °C	50 °C	26
T_{vdl} V.A.T at driver's left side	0 °C	50 °C	26
T_{vpc} V.A.T at passenger's central side	0 °C	50 °C	26
T_{vpr} V.A.T at passenger's right side	0 °C	50 °C	26
T_{adf} A.T of driver's floor	0 °C	45 °C	36
T_{rc} A.T of recirculation	-10 °C	45 °C	26
T_{apf} A.T of passenger's floor	0 °C	45 °C	36
T_{vd} A.T of duct at driver's side	0 °C	60 °C	41
T_{vp} A.T of duct at passenger's side	0 °C	60 °C	41

switch it off until the temperature rises above 22 °C. Therefore, in this cabin thermal model, a double bang-bang controller [RWP15], [Hua94] is developed to maximally maintain the comfortable air temperatures for both occupants. The subsequent thermal control Algorithm 2 indicates a double bang-bang thermostat which decides specific cooling or heating control actions based on the feedback of mean air temperature. The feedback mean air temperature thresholds for heating/cooling mode are 19 °C and 23 °C (below 19 °C switch on heating, over 23 °C switch to cooling mode). There are four HVAC control options ranging from maximal cooling to the maximal heating power options listed in Table 5.2.

Algorithm 2 Double bang-bang HVAC control process

1: Receive last HVAC control Z_{pre} as the input
2: Observe air temperatures $\left[T_{dh}, T_{dt}, T_{df}, T_{ph}, T_{pt}, T_{pf}\right]$
3: Calculate mean air temperature $\bar{T}_{air}$ over driver and passenger
4: **if** $\bar{T}_{air} < 10°C$ **then**
5: output strong heating option Z_{sh}
6: **else if** $10°C \leqslant \bar{T}_{air} < 19°C$ **then**
7: output medium heating option Z_{mh}
8: **else if** $23°C \leqslant \bar{T}_{air} < 32°C$ **then**
9: output medium cooling option Z_{mc}
10: **else if** $\bar{T}_{air} \geqslant 32°C$ **then**
11: output strong cooling option Z_{sc}
12: **else**
13: output last control option Z_{pre}
14: **end if**

Table 5.2: The control variable settings for the double bang-bang controller (unit for I_b is Ampere A, T_{vdc} to T_{vp} is Celsius °C)

Control variables	I_b	A_{fr}	A_{dist}	T_{vdc}	T_{vdl}	T_{vpc}	T_{vpr}	T_{adf}	T_{rc}	T_{apf}	T_{vd}	T_{vp}
Z_{sc}: Strong cooling	10	0	0	12	12	15	12	15	35	12	20	38
Z_{mc}: Medium cooling	8	0	0	16	16	18	15	20	30	20	25	32
Z_{mh}: Medium heating	8	1	0	22	22	30	20	25	15	27	26	31
Z_{sh}: Strong heating	10	1	1	28	28	32	25	30	12	30	28	28

5.3 EXPERIMENT SETTING

As mentioned in chapters 2 and 3, the policy gradient reinforcement learning (PGRL) framework updates policy based on the states, actions and rewards information received from each episode (trial). The number of learning trials is set as 5×10^3 and the time steps for each episode is 5×10^3 s (approximately 83.4 minutes). At the beginning of each episode, the state variables are randomly selected from different intervals.

- initial air, mean radiant temperatures are randomly selected from [-8, 40] °C, and the difference between each is constrained within ± 4°C
- relative humidities of cabin and environment are randomly selected from [12, 70]%
- vent air velocities of passenger and driver are randomly selected from [0, 1] $\mathrm{m\,s^{-1}}$
- car velocity starts from 0 $\mathrm{km\,h^{-1}}$ but variates between 50 to 100 $\mathrm{km\,h^{-1}}$ in this episode later on. (section 5.4.1 below)

where the differences between initial air and mean radiant temperatures of occupants are constrained within a range. The initial cabin thermal status is closed to the ambient environment as the car is parked for a long enough time.

5.3.1 *Equivalent temperature and reward function*

As mentioned in chapter 3, the equivalent temperature (ET) is applied to decide whether the cabin provides a comfortable thermal status. The equation below estimates ET for occupants' different body sections. T_a represents air temperature in occupants' body section, for example, driver's head temperature T_{dh}. M can represent driver's head mean

radiant temperature M_{dh} and v_a equals air velocity at driver's head section v_{dh}. The clothing insulation coefficient I_{cl} keeps the same value in chapter 3. Therefore, driver's head equivalent temperature T_{edh} is calculated as follow

$$T_e = \begin{cases} 0.5(T_a + M); v_a \leqslant 0.1\ m\,s^{-1} \\ \\ 0.55T_a + 0.45M + \frac{0.24-0.75\sqrt{v_a}}{1+I_{cl}}(36.5 - T_a); v_a > 0.1\ m\,s^{-1} \end{cases} \tag{5.6}$$

This equation also calculates driver's torso to passenger's feet ETs (T_{edt} to T_{epf}). The reward function defines the penalty for driving the ET out of the corresponded comfort zones. According to Table 5.3, when the head ET is above the "hot but comfortable" boundary (22.4 °C in winter) and below the "too hot" boundary (27.9 °C in winter), the reward function presents a -1 for head ET. However, if torso ET is above the "cold but comfortable" temperature and below the "hot but comfortable" temperature (known as neutral comfort), the occupant does not feel cold or hot in the comfort zone.

- T_{th}: when ET is above this value, the occupant feels too hot (uncomfortable)
- T_{hc}: when ET is above this but below T_{th}, the occupant feels hot but comfortable
- Neutral comfort zone: when ET is below T_{hc} and above T_{cc}, the occupant feels not too hot or too cold in the thermal comfort zone
- T_{cc}: when ET is below this but higher than T_{tc}, the occupant feels cold but comfortable
- T_{tc}: when ET is below this value, the occupant feels too cold (uncomfortable)

The T_{tc} to T_{th} values also depend on different climate according to Table 5.3 below. Based on the definition of comfort boundaries, the reward function presents 0 for "neutral comfort condition" and -1

for "cold but comfortable" or "hot but comfortable" conditions. The reward detail is listed

$$r(T_e) = \begin{cases} 0 & \text{if} \quad T_{cc} \leqslant T_e \leqslant T_{hc} \\ -1 & \text{if} \quad T_{tc} \leqslant T_e < T_{cc} \quad \text{or} \quad T_{hc} < T_e \leqslant T_{th} \\ -2 & \text{else} \end{cases} \tag{5.7}$$

So the total reward value equals the sum of occupants' body comfort rewards and the energy consumed by cool/warm control actions of the HVAC system.

$$R(s, a) = r(T_{edh}) + r(T_{edt}) + r(T_{edf}) + r(T_{eph}) + r(T_{ept}) + r(T_{epf}) - \omega_e \left[Q_d + Q_p\right] \tag{5.8}$$

where ω_e is the energy weight factor selected from the domain $[0, 1/3000, 1/1000, 1/300]$. The air heating/cooling power for driver Q_d and passenger Q_p is calculated with respect to the airflow velocity and heat transfer of hot or cold air [Lee+15], [TJ+15]. In the following testing simulations, the ω_e value equals $1/1000$.

Table 5.3: Human body's thermal comfort zone constrained by Nilsson's equivalent temperatures (Summer and Winter season) [Nil04]

	too cold (< °C)		cold but comfortable(< °C)		hot but comfortable (> °C)		too hot (> °C)	
Climate	summer	winter	summer	winter	summer	winter	summer	winter
Face	10.9	11.5	18.7	17.0	26.6	22.4	34.4	27.9
Thigh	17.0	17.9	21.2	22.3	25.5	26.8	29.8	31.2
Feet	17.0	17.9	21.2	22.3	25.5	26.8	29.8	31.2
Torso	13.7	14.7	21.6	22.0	25.2	26.2	32.0	30.0
full-body	20.5	18.6	23.6	21.4	26.7	24.3	29.9	27.1

5.4 RESULTS AND DISCUSSION

5.4.1 *The settings for environmental factors*

As mentioned above, the external factors, such as car velocity V_{car}, roof temperature T_r, ambient temperature T_{env} and ambient relative humidity ϕ_{env}, also impact cabin thermal status when executing the HVAC control actions (Table 5.1). This section presents the external information based on the data samples collected in the climatic wind tunnel experiment [BR20].

- T_{env}: Winter ambient temperature values are randomly selected between -13 to -10 °C, summer ambient temperature values are between 34 to 36.5 °C.
- ϕ_{env}: Winter ambient relative humidity values are randomly selected from 60% to 70%, the summer cases are randomly selected between 5% to 20%.
- T_r: Winter roof temperature values are selected between -9 to -4 °C, summer cloudy values are from 33 to 36.5 °C, the summer sunny values increase from 40 to 80 °C after 60 min
- V_{car}: Car velocity is randomly selected between 50 km to 100 km over the simulation time

Table 5.4: The range of ambient temperature T_{env}, roof temperature T_r, ambient humidity ϕ_{env} and car velocity V_{car} values for warm-up/cool-down testing cases

	Winter	Summer(cloudy)	Summer (Sunny)
T_{env}	$[-13, -10]$ °C	$[34, 36.5]$ °C	$[34, 36.5]$ °C
T_r	$[-9, -4]$ °C	$[33, 36.5]$ °C	40 increases to 80 °C
ϕ_{env}	$[60, 70]$ %	$[5, 20]$ %	$[5, 20]$ %
V_{car}	$[50, 100]$ km	$[50, 100]$ km	$[50, 100]$ km

As the cabin roof can absorb solar radiation to increase its surface temperature far above the ambient air temperature. Based on the data measurements, the above T_r setting shows an example case that the roof temperature significantly increases to eighty degrees after one hour. The main challenge is that the intense solar radiation warms up the top section of car cabin, and consequently raises the air temperatures of occupants' torso to head sections. Therefore, these cold and warm environment conditions examine whether a thermal controller can offer desired thermal comfort for occupants. The data of car velocity is also randomly selected between 50 to 100 km for all climate cases. In terms of the over-heat cabin roof condition, the cabin roof receives intense solar radiation to drastically warm up the surface material temperature. Therefore, the cabin environment can be heated to an extreme hot condition. This setting aims to test whether the resulting RL HVAC controller can cool down occupants and maintain a cooler thermal comfort condition.

5.4.2 *The episodic rewards throughout the learning trials*

As mentioned in chapter 3, the episodic reward value indicates the summation of state-action rewards that the agent can receive in an episode with a limited time-step. In the PGRL framework, the way to update control policy is based on the maximization of received episodic rewards. As for the initial state value of each episode, sec-

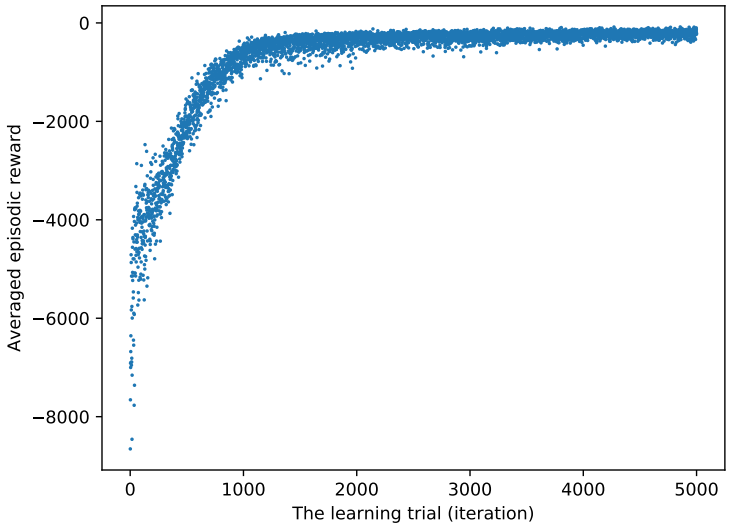

Figure 5.1: The averaged episodic rewards against 5×10^3 learning trials

tion 5.3 above has mentioned random initialization intervals. Figure 5.1 indicates a steady convergence to episodic reward value -258.89, also these increasing episodic rewards are averaged over the learning trial results by using ten different random seeds and fixed energy weight ω_e equals 1/1000. An increasing episodic reward means that the RL-controller tends to maintain a longer proportion of comfort for occupants and consume less energy. The total simulation time is 2.5×10^7 s.

5.4.3 *Warm-up test in cold climate*

Based on the data of ambient winter climate indicated above, this part demonstrates relevant graphs of testing results. The following graphs present how the HVAC controllers achieve and maintain the desired comfort represented by the equivalent temperatures (ETs). The comfort zone is constrained by ETs mentioned in Table 5.3. For example, if occupant's head ET is between 11.5 to 27.9°C at winter

time, the occupant then feels comfortable. Therefore, the following graphs use black solid and dashed lines as the ET boundaries, and use a green line to indicate the comfort zone's baseline (mean value of the upper and lower ET boundary). The purple solid and dashed lines respectively represent the "hot but comfort" and the "cold but comfortable" ET boundaries. The area constrained by the purple solid and dashed lines are named as the neutral comfort zone. According to Table 5.3 and Nilsson's work [Nil04], the neutral comfort zone does not make the occupants feel too cold or hot in the thermal comfort zone.

The following graphs illustrate the occupants' head, torso and feet ET, the mean body ET is then calculated as the mean value of head to feet ETs. The abbreviation "eqt" in the graphs refers to equivalent temperature, "AT" refers to air temperature, "MRT" is mean radiant temperature. The mean body ETs of both occupants in Figure 5.2 indicates that the PPO-based HVAC controller can warm up both occupants' body sections to the neutral comfort zone. The RL controller can consistently maintain this comfortable thermal status for a long enough time. As a comparison, the bang-bang controller takes longer (more than 28 min) to achieve the target comfort and does not fully maintain the ETs in the neutral comfort zone. Table 5.5 details the time spent on achieving the target comfort zone, the PPO-RL method significantly reduces the time to achieve the neutral comfort zone. In Table 5.6, compared with the bang-bang method, RL-based controller improves the percentage of neutral comfort above 90%.

Figure 5.3 indicates that the resulting PPO-RL controller can drive up the equivalent temperature of both occupants' head from temperature below zero to the neutral comfort zone in around three minutes, this is faster than the bang-bang. As a comparison, the red-dashed curve indicates that the bang-bang HVAC controller drives up the driver's head ET in a short time, but its fluctuation exceeds the upper comfort boundary. After fifty minutes, the bang-bang controller maintains the driver's head ETs in stable fluctuation. However, driver's head ETs are in the "cold but comfortable" region (ETs below 17°C according to Table 5.3). The blue-dashed curve represents the passenger's head ET achieved by the bang-bang control, the result shows that the bang-bang also raises passenger's head ET to the comfort region and maintains it with a reduced fluctuation amplitude. However, the bang-bang controller drives up the ETs above the upper boundary of comfort, thus making the occupants feel too hot for several minutes, then

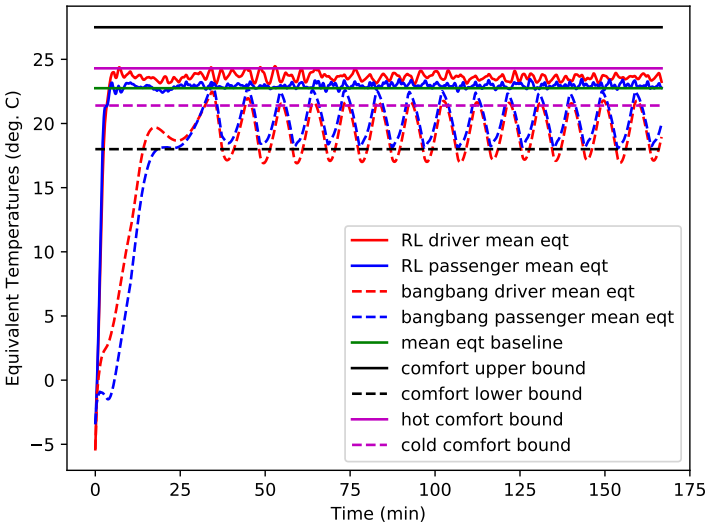

Figure 5.2: Driver and passenger's mean body equivalent temperatures in the warm-up process

cooling down the cabin and keeping ETs within the comfort region. However, the PPO-RL controller precisely achieves the target ET which does not make occupants feel too hot or too cold in the comfort zone. Also the airflow rate determines ETs according to the definition, hence higher airflow can pump more warm air into the cabin so as to increase the cabin temperature, but the definition of ET indicates that higher airflow rate can lower the ET when having low air and mean radiant temperatures.

Similar to the head warming-up case above, Figure 5.4 indicates that the resulting PPO-RL controller can drive up the ETs of both occupants' torso from a temperature below zero to the neutral comfort zone in around four minutes, this is also faster than the bang-bang method. Meanwhile, the bang-bang control method also succeed in the warming up process and maintains both occupants' torso ETs within the neutral comfort region. The red and blue-dashed curves show that the RL-based HVAC controller can raise both occupants' torso ET to the baseline ET within eight to ten minutes. However, the bang-bang HVAC controller raises the driver's torso ET over the baseline, thus providing too hot thermal sensation for occupants. When torso ET is over 27 °C (shown in Table 5.3 above), the occupant is likely to feel too hot. Therefore, in Figure 5.4, the bang-bang controller offers stable neutral comfort for driver's torso after 40 minutes. The driver's torso

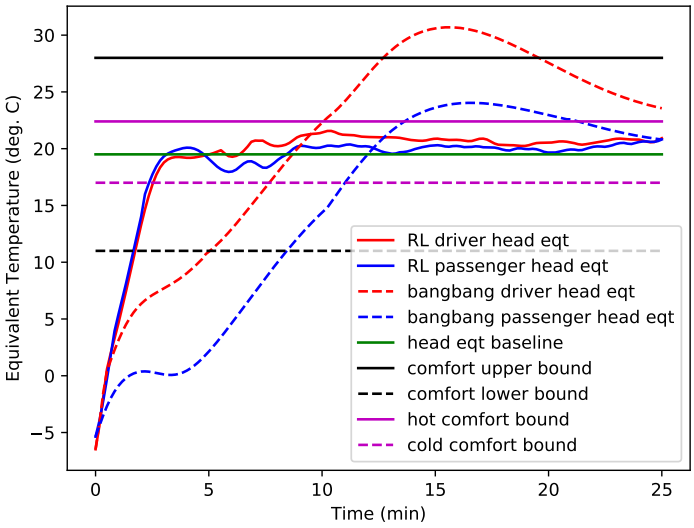

Figure 5.3: The 0 to 25 min of occupants' head ET in the warm-up process

AT and MRT fluctuations by bang-bang control are corresponded to the ETs curve above.

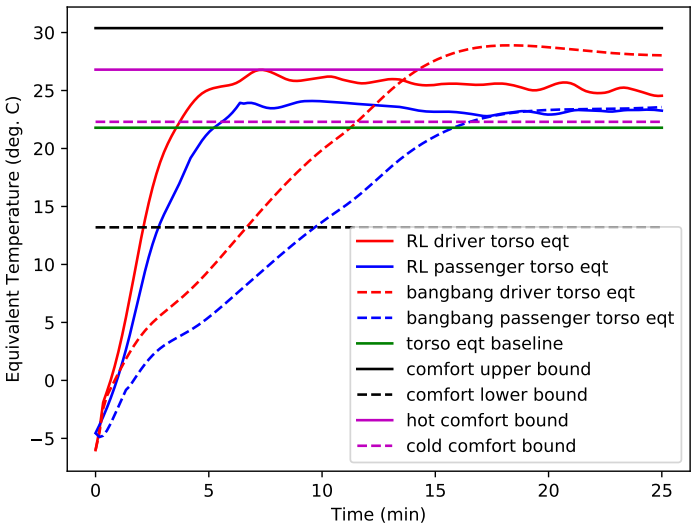

Figure 5.4: The 0 to 25 min of occupants' torso equivalent temperatures in the warm-up process

The occupants' feet ET curves in Figure 5.5 indicate the time lag between PPO-RL and bang-bang HVAC controllers. Red and blue solid curves indicate that the PPO-RL HVAC controller drastically raises the feet ET to the baseline comfort equivalent temperature in less than five minutes according to Figure 5.6. But the passenger's feet ET exceeds

the upper comfort boundary for a short duration, this is caused by a drastically increasing AT of the passenger. As a comparison, the red and blue-dashed curves indicate that the bang-bang HVAC controller takes more than fifty minutes to achieve comfort ET zone. Such a big time lag can also be observed in AT and MRT curves in subsequent graphs. Therefore, the PPO-based HVAC controller can consistently provide warm thermal comfort to the occupants. However, the bang-bang control takes a longer time to achieve the target comfort.

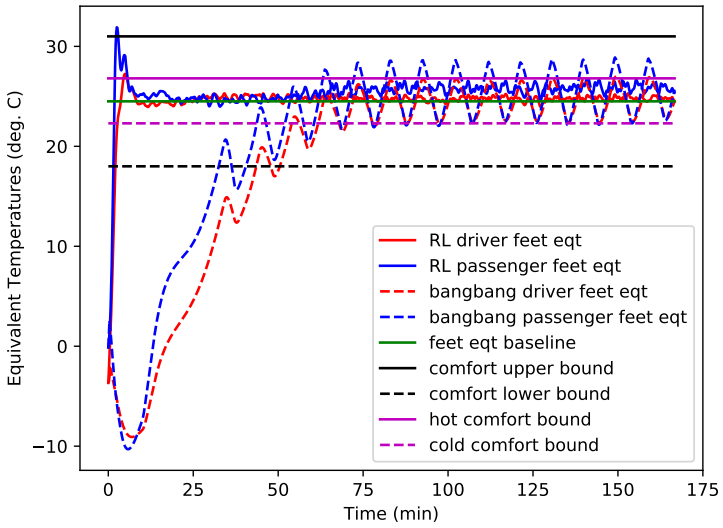

Figure 5.5: Driver and passenger's feet equivalent temperatures in the warm-up process. This shows that the bang-bang method achieves the comfort very slow

Table 5.5: Time (min) taken to achieve neutral comfort zone and the percentage of time reduced due to PPO-RL compared with bang-bang controller in the warm-up process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator.

	Driver			Passenger		
Method	Bang-bang	PPO-RL	% increase	Bang-bang	PPO-RL	% increase
Head	7.70	2.70	-64.9%	11.0	2.50	-77.3%
Torso	11.5	3.83	-66.7%	16.5	5.70	-65.5%
Feet	54.0	2.83	-94.8%	43.6	1.83	-95.7%
Mean body	32.3	3.50	-89.2%	32.0	3.67	-88.5%

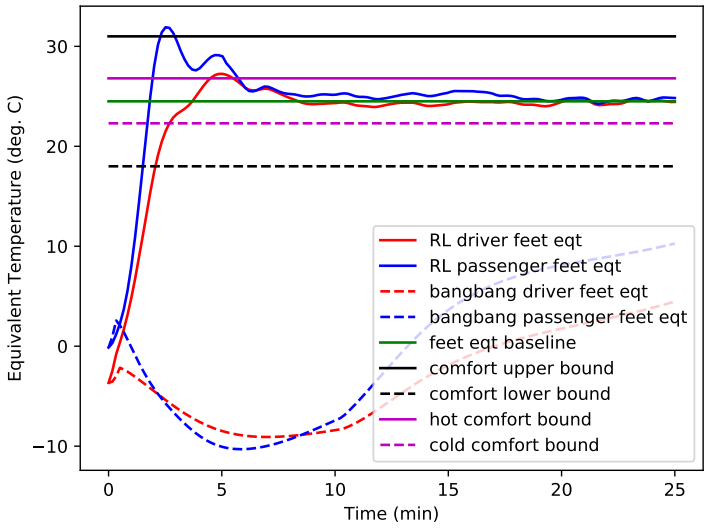

Figure 5.6: The 0 to 25 min of the occupants' feet ET warm-up process in Figure 5.5 above

Table 5.6: Percentage of time in neutral comfort zone (during 0 to 160 min) and comfort improvement due to PPO-RL compared with bang-bang controller in the warm-up process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator.

	Driver			Passenger		
Method	Bang-bang	PPO-RL	% increase	Bang-bang	PPO-RL	% increase
Head	6.20%	98.4%	1490%	34.8%	98.5%	183.05%
Torso	14.3%	97.7%	583%	19.0%	96.6%	408%
Feet	60.3%	98.0%	62.5%	46.3%	97.0%	110%
Mean body	13.5%	96.5%	614.8%	23.5%	97.8%	316%

5.4.4 *Cool-down test without solar radiation on cabin roof*

Subsequent graphs indicate cool-down testing cases without solar radiation in Table 5.4. The curves in this section indicate the cool-down cases without solar radiation. The subsequent Figure 5.7 shows both occupants' mean body ETs in the cool-down testing case. The PPO-RL controller cools down driver and passenger's body sections to the neutral comfort zone in less than two minutes, this process reduces more than 80% of time spent by the bang-bang controller. However, the passenger's mean body ET is cooled around 1 to 2 °C below the "cold but comfortable" boundary. So the percentage of time in neutral comfort zone is lower than the bang-bang application according to Table 5.8. Still, the results in Table 5.7 indicate that the

PPO-RL reduces 40% to 90% of time taken to achieve neutral thermal comfort. The proportion of time spent on neutral thermal comfort in Table 5.8 indicates obvious cool-down task improvements by the PPO-RL controller. For example, compared with bang-bang results, about 80% to 275% relative more time is spent in neutral comfort. Figure 5.8

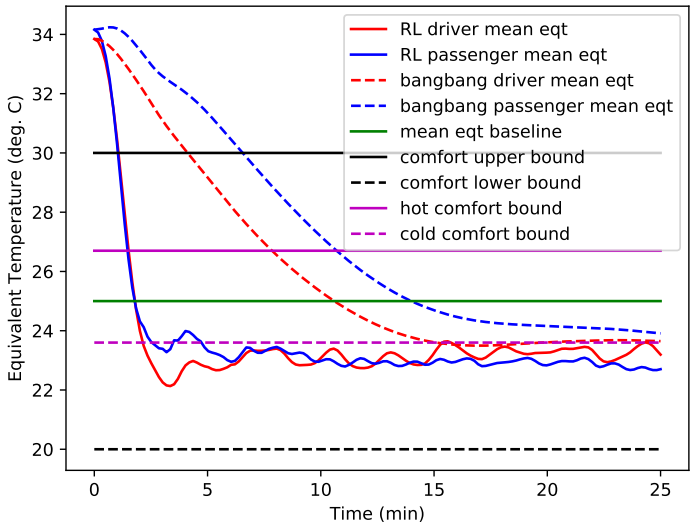

Figure 5.7: The 0 to 25 min of the occupants' mean body ET in the cool-down process (no solar radiation)

indicates that the resulting PPO-RL controller can cool down both occupants' head ET to the baseline ET in around eight minutes, this cool-down process also achieves stable neutral thermal comfort. As a comparison, the red and blue-dashed curves indicates that the bang-bang HVAC controller can cool down both occupants' head ET below the comfort baseline. And according to Table 5.3, when the head ET is below 18.7 °C the occupant is likely to feel cold in the summer season. So, the double-bang-bang method can not consistently offer the neutral thermal comfort in this cool-down task.

The occupants' torso cool-down results in Figure 5.9 show corresponding fluctuations in the bang-bang control part. Still, the RL HVAC controller can drastically cool down both occupants' torso ET from 34 °C to the neutral comfort baseline ET in around five minutes. However the bang-bang controller achieves colder ET in the comfort zone and obvious temperature fluctuations. Similar to the cool-down results above, the bang-bang method can not preserve stable cool AT and MRT of occupants' torso. Because the bang-bang controller might drive up obvious fluctuations when its heating mode is turned on.

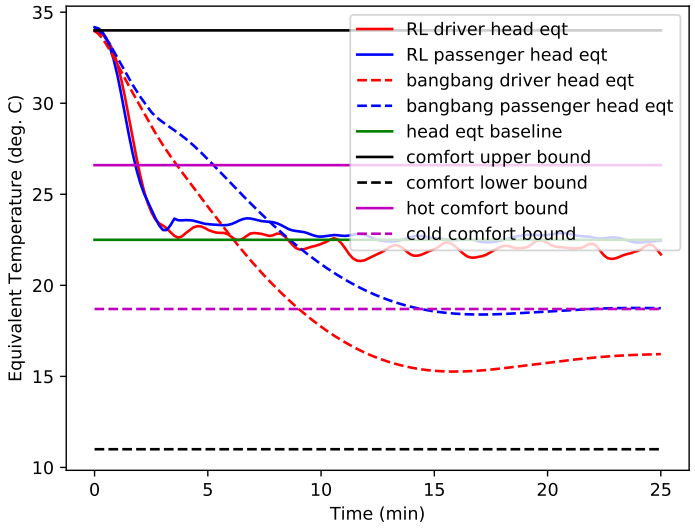

Figure 5.8: The 0 to 25 min of the occupants' head ET in the cool-down process (no solar radiation)

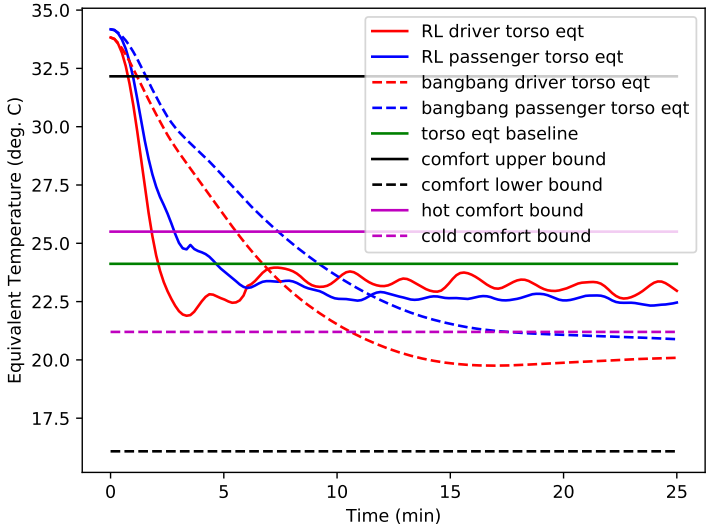

Figure 5.9: The 0 to 25 min of the occupants' torso ET in the cool-down process (no solar radiation)

The following graphs present occupants' feet cool-down results. In Figure 5.10, the red and blue solid curves indicate that the RL HVAC controller can cool down both occupants' feet to the neutral comfort in around two minutes. And both occupants' feet ETs are maintained in the neutral comfort region. As a comparison, the red and blue-dashed curves indicate that the bang-bang HVAC controller takes more than forty minutes to achieve the comfort zone. Such a big time

lag can also be clearly observed in Figure 5.11 for the beginning one hour time. Therefore, in the hot climate, the PPO-RL HVAC controller can offer stable cooling comfort to the cabin occupants. Bang-bang HVAC controller fails to achieve stable cooler temperatures, because the heating mode causes high fluctuations for temperature variations.

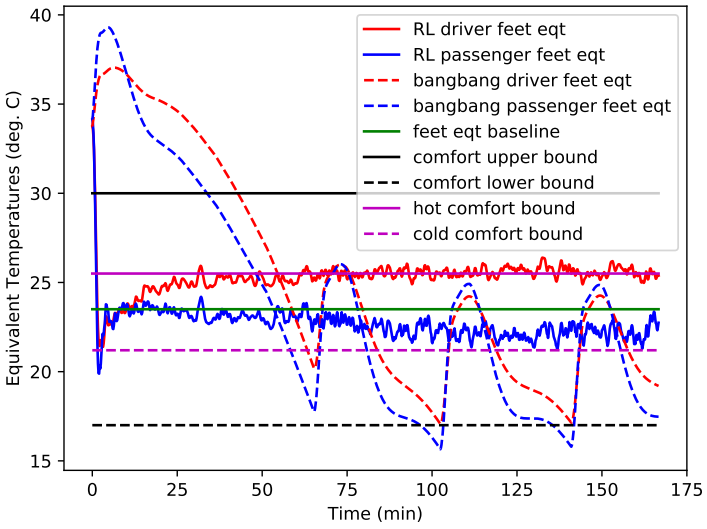

Figure 5.10: Driver and passenger's feet ET in the cool-down process

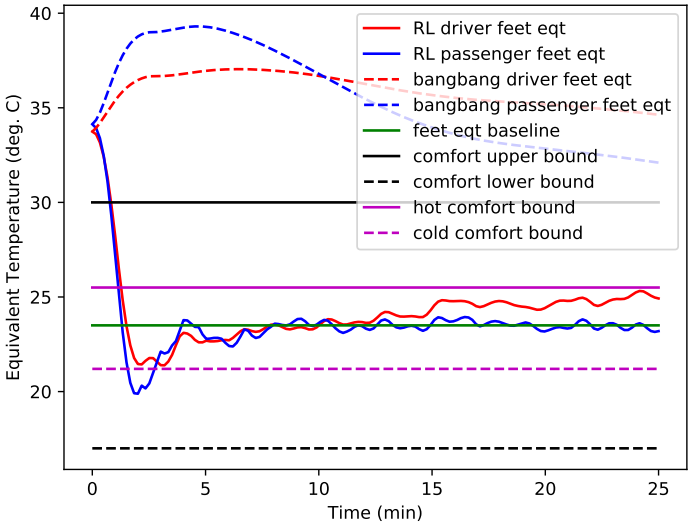

Figure 5.11: The 0 to 25 min of the occupants' feet ET cool-down process in Figure 5.10 above

Table 5.7: Time (min) taken to achieve neutral comfort zone and the percentage of time reduced due to PPO-RL compared with bang-bang controller in the cool-down (no solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator.

	Driver			Passenger		
	Bang-bang	PPO-RL	% increase	Bang-bang	PPO-RL	% increase
Head	3.67	2.00	-45.5%	5.33	1.83	-65.7%
Torso	5.67	1.83	-67.7%	7.50	2.83	-62.3%
Feet	54.7	1.33	-97.6%	48.2	1.17	-97.6%
Mean body	10.2	1.67	-83.6%	13.5	1.50	-88.9%

Table 5.8: Percentage of time in neutral comfort zone (during 0 to 160 min) and comfort improvement due to PPO-RL compared with bang-bang controller in the cool-down (no solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator.

	Driver			Passenger		
Method	Bang-bang	PPO-RL	% increase	Bang-bang	PPO-RL	% increase
Head	37.0%	98.8%	167%	48.0%	98.9%	106%
Torso	55.0%	98.9%	79.8%	53.7%	98.3%	83.1%
Feet	29.1%	61.2%	110%	26.3%	98.6%	275%
Mean body	13.3%	26.6%	100%	21.1%	1.50%	-92.9%

5.4.5 *Cool-down test with solar radiation on cabin roof*

As mentioned above, the solar radiation can warm up the cabin roof to increase the air temperature of the top section of the cabin. Therefore, based on real-time data samples, Table 5.4 introduces a testing case with roof temperatures increasing above 80°C after one hour. By introducing an intense solar radiation, this experiment aims to examine the RL HVAC controller's cooling down capability. One challenge is that the occupants' head and torso section can be warmed up by the over heated cabin roof material. Figure 5.12 shows the occupants' mean body ETs by RL and bang-bang methods. The PPO-RL HVAC controller cools down driver and passenger's body sections to the neutral comfort zone in less than one minute according to Figure 5.13, this process reduces around 80% of time spent by the bang-bang controller. The passenger's mean body ET (blue solid curve) is cooled below the "cold but comfortable" boundary before 75 min. However, the blue and red-dashed curves present frequent ET fluctuations between 21 to 27 °C, hence the bang-bang controller cools down mean body ET below the comfort lower boundary. The results in Table 5.10 indicate that the PPO-RL reduces 20% to 90% of time taken to achieve neutral

thermal comfort, and Table 5.9 presents RL improvements in maintaining neutral thermal comfort. The following Figure 5.14 shows

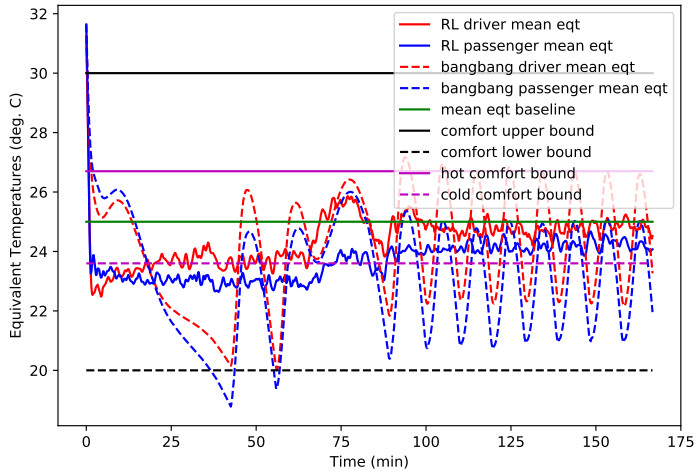

Figure 5.12: Driver and passenger's mean body ET in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min)

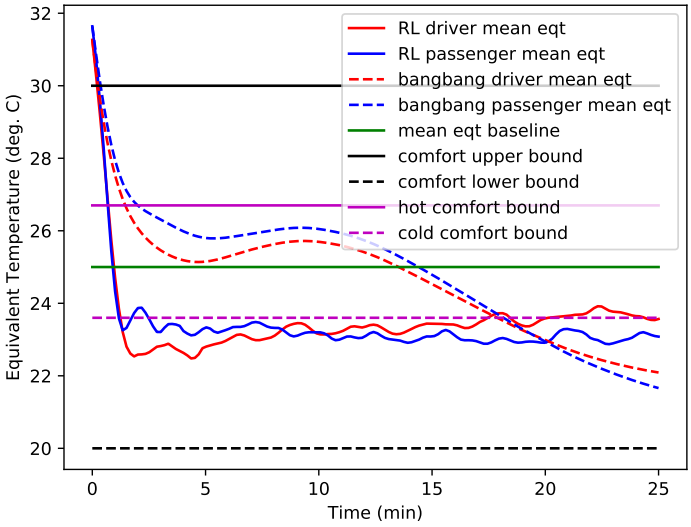

Figure 5.13: The 0 to 25 min of driver and passenger's mean body ET cool-down process (solar radiation) in Figure 5.12 above

occupants' head equivalent temperatures (ETs) in the solar radiation cool-down process. The red and blue solid curves indicate that the RL HVAC controller cools down head ET to the neutral baseline ET during the initial one hour. When the cabin roof receives significant solar radiation after one hour, the head ET is still maintained in the

comfort zone, although it slightly raises above the "hot but comfortable" ET boundary. As a comparison, the bang-bang controller can not preserve cooler comfort, and the over-heated roof increases both occupants' head ETs according to the red and blue-dashed curves in Figure 5.14. The following Figure 5.15 shows occupants' torso equiva-

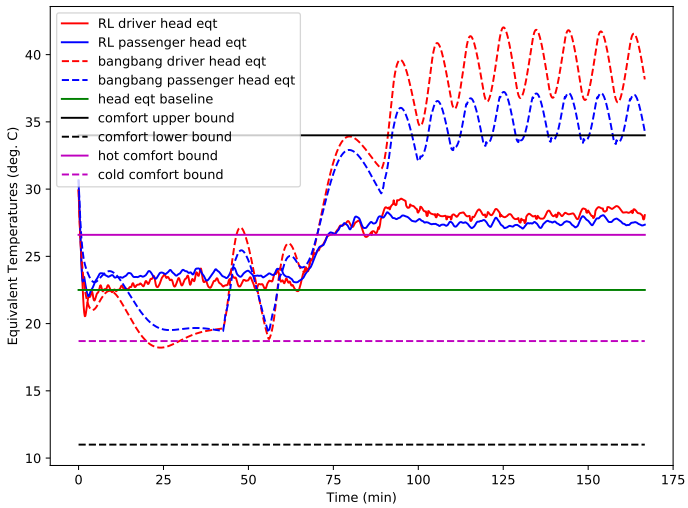

Figure 5.14: Driver and passenger's head equivalent temperatures in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min)

lent temperatures (ETs) in the solar radiation cool-down process. The red and blue solid curves indicate that the RL HVAC controller also cools down torso ET to the neutral baseline ET during the initial one hour. When the cabin roof receives significant solar radiation after one hour, the torso ET is still maintained in the comfort zone, although the driver's torso ET slightly raises above the baseline. As a comparison, the bang-bang controller can not preserve cooler comfort, and the over-heated roof increases both occupants' torso ETs, according to the red and blue-dashed curves in Figure 5.15. Still, the RL HVAC controller achieves the occupant's neutral thermal comfort faster than the bang-bang according to Figure 5.16.

The following Figure 5.17 shows occupants' feet equivalent temperatures (ETs) in the solar radiation cool-down process. The red and blue solid curves indicate that the RL HVAC controller also cools down feet ET to the neutral baseline ET before solar radiation starts heating the roof. When the cabin roof receives significant solar radiation after one hour, the feet ET is still maintained in the comfort zone, although passenger's feet ET slightly drops below the baseline. As

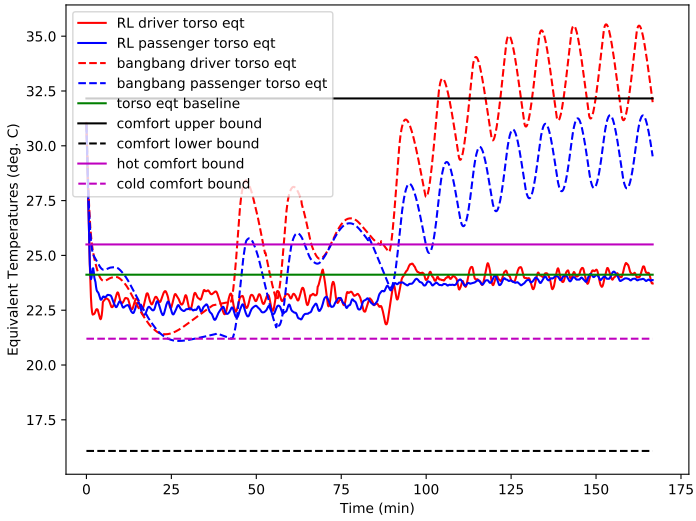

Figure 5.15: Driver and passenger's torso equivalent temperatures in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min)

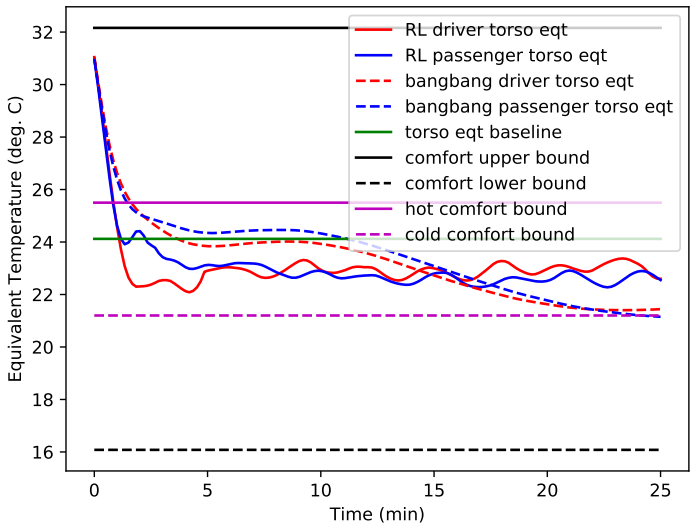

Figure 5.16: The 0 to 25 min of driver and passenger's torso ET cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) in Figure 5.15 above

a comparison, the bang-bang controller over cools down occupants' feet, and the feet ET decreases below the lower comfort boundary according to the red and blue-dashed curves in Figure 5.17. Also, the RL-based controller achieves the neutral thermal comfort according to Figure 5.18. These three sections (5.4.3, 5.4.4, 5.4.5) present the

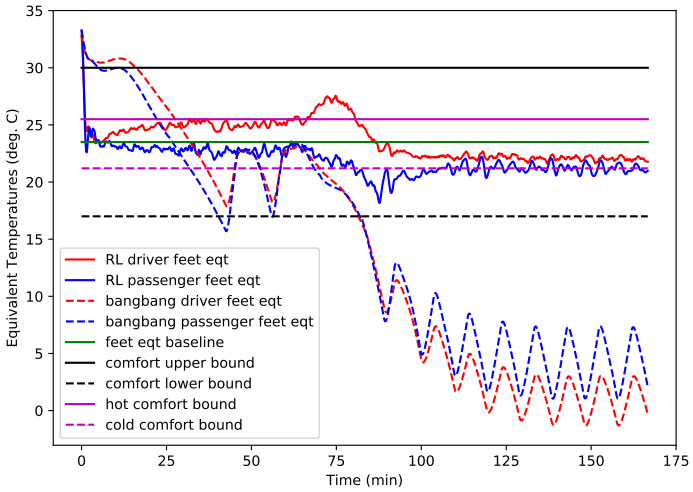

Figure 5.17: Driver and passenger's feet equivalent temperatures in the cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min)

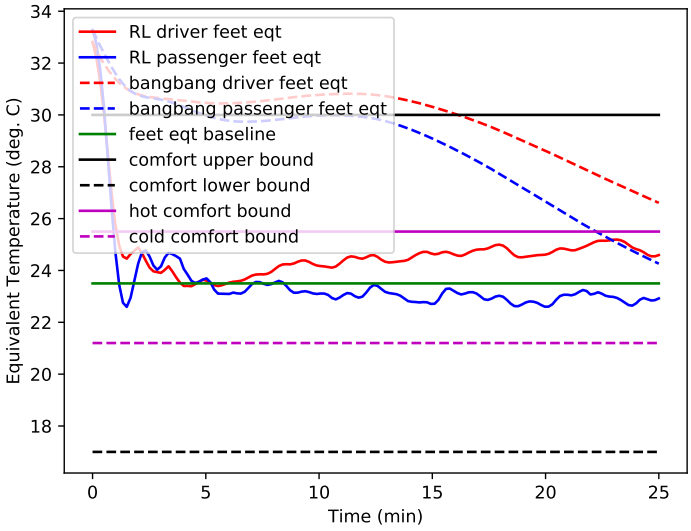

Figure 5.18: The 0 to 25 min of driver and passenger's feet ET cool-down process (solar radiation heat cabin roof from 40 °C to 80 °C after 60 min) in Figure 5.17 above

results of using PPO-RL and bang-bang methods to warm up and cool down the cabin occupants to the comfort zone in the winter and summer climate. Section 5.4.3 results show that the RL-based HVAC averagely warms up the occupants' body to the neutral comfort zone in around 3.3 min. This value is 89% less than the time by the bang-bang HVAC. And RL-based HVAC controller averagely maintains 97%

Table 5.9: Time (min) taken to achieve neutral comfort zone and the percentage of time reduced due to PPO-RL compared with bang-bang controller in the cool-down (with solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator.

	Driver			Passenger		
Method	Bang-bang	PPO-RL	% increase	Bang-bang	PPO-RL	% increase
Head	0.5	0.5	0%	0.83	0.67	-19.3%
Torso	1.67	1.0	-40.1%	1.50	0.83	-44.7%
Feet	27.7	1.17	-95.8%	22.2	1.0	-95.5%
Mean body	4.0	0.83	-79.3%	13.8	0.83	-94.0%

Table 5.10: Percentage of time in neutral comfort zone (during 0 to 160 min) and comfort improvement due to PPO-RL compared with bang-bang controller in the cool-down (with solar radiation) process. These results are based on 5×10^3 simulated trials (episodes) using "CWT-cabin-env" simulator.

	Driver			Passenger		
Method	Bang-bang	PPO-RL	% increase	Bang-bang	PPO-RL	% increase
head	34.6%	45.1%	30.3%	41.6%	43.6%	4.80%
Torso	34.4%	99.4%	189%	39.4%	99.5%	153%
Feet	17.1%	89.3%	422%	16.1%	66.0%	310%
Mean body	25.8%	81.6%	216%	35.4%	53.3%	50.6%

time of neutral thermal comfort compared to the 27% by bang-bang controller. Sections 5.4.4 and 5.4.5 indicate cool-down tasks including and excluding solar radiation on cabin roof. The results in section 5.4.4 indicates that the RL HVAC controller averagely takes 1.8 min to cool down the occupants to the neutral comfort zone. Section 5.4.5 shows that RL HVAC controller can averagely maintain a cooler thermal comfort when solar radiation starts to heat the cabin roof after one hour. The RL-based HVAC controller averagely achieves 72% neutral comfort compared to the 30% by bang-bang controller. Therefore, the PPO-RL method outperforms the bang-bang approach in the warm-up and cool-down tasks.

5.4.6 *Comfort percentage and power consumption*

This section includes the resulting HVAC controllers' performance in maintaining occupants' full body sections' comfort under various ambient climate conditions. The PGRL HVAC control policy $\pi_\theta(\cdot \mid s)$ is being optimized throughout the 5×10^3 learning trials (each trial or episode starts with random state information). Therefore, the testing case includes the same amount of randomly selected initial

states while using the same random seeds. Similar to the experiment setting in section 5.3, this section generates a testing case by randomly selecting the initial air, mean radiant temperatures of driver and passenger sections from -8 to 40 °C, environment and cabin roof from -8 to 40 °C, and all body sections' airflow v_i from 0 to 1 $m\,s^{-1}$. The difference between each is constrained within ±4°C. Given this information as initial car cabin state for each episode, the controllers (policies) are being simulated throughout a fixed time-step of 5×10^3 s (approximately equals 83.4 minutes). Based on the proportion of time that the RL-based or bang-bang controller can maintain occupants' comfort and neutral thermal comfort, subsequent graphs indicate the thermal comfort rate distributions for occupants.

- Comfort rate: the proportion (or percentage) of time that the HVAC controller can maintain occupants' body equivalent temperatures (ET) in the thermal comfort zone, but the occupant might feels cold or hot when ET is below T_{cc} or above T_{hc} (section 5.3.1)
- Neutral comfort rate: the proportion (or percentage) of time that the HVAC controller can maintain occupants' body equivalent temperatures (ET) in the neutral thermal comfort zone. Therefore, the occupant won't feel too cold or hot when ET is maintained between T_{cc} to T_{hc}.

The box plot in Figure 5.19 shows the comfort rate concentrations of using RL-based and bang-bang HVAC controllers to maintain driver's head, torso and feet body sections on 5×10^3 randomly initialized episodes. The PPO-RL HVAC controller maintains more than 90% comfort duration for driver's body sections. However, for more than 75% of full-body cases, bang-bang controller can only keep the comfort rate above 60%. The feet section comfort results are centred around value domain between 50% to 90% and averagely yield 71.53% comfort. As for the torso section, the black cross marks outliers, indicating 5% test cases with comfort percentage higher ranging from 30% to 80%. Therefore, the PPO-RL method generally achieves higher comfort proportion compared to the bang-bang method. Table 5.11 below denotes the mean comfort percentage values of driver's body sections.

Similar to the driver's body sections, Figure 5.20 presents the comfort rate concentrations of using RL-based and bang-bang HVAC controllers to maintain passenger's head, torso and feet body sections on 5×10^3 randomly initialized episodes. The PPO-RL HVAC controller

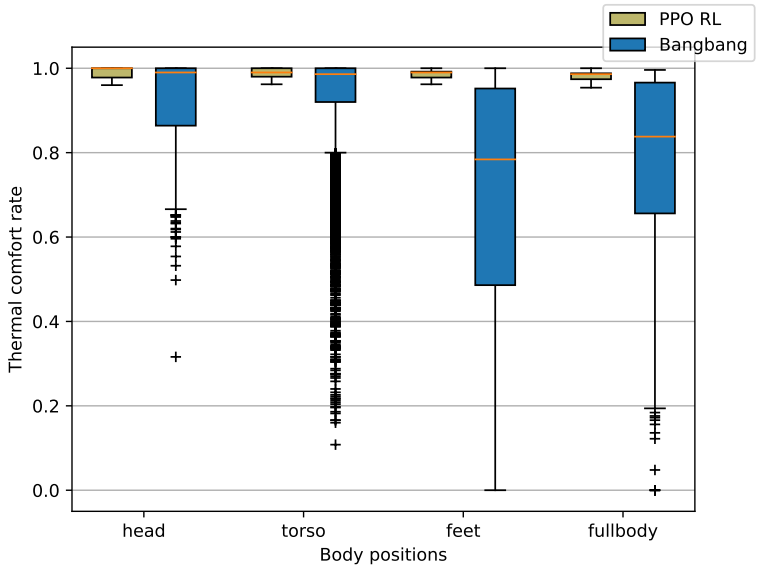

Figure 5.19: Comfort rate of driver's body positions

still maintains more than 90% comfort duration for passenger's body sections. As for the full-body testing case, the bang-bang controller can only keep the comfort rate above 40%. The bang-bang controller's feet section comfort results are centred around value domain between 60% to 90% and averagely yield 76.67% comfort duration. As for torso and head sections, the black cross marks denote outliers, indicating 5% test cases with comfort percentage higher ranging from 40% to 80%. Therefore, the PPO-RL method generally achieves higher comfort proportion compared to the bang-bang method. Also, Table 5.11 details the mean comfort percentage values of passenger's body sections.

Table 5.11: Percentage of time that the PPO-RL and bang-bang controller can maintain occupants in the thermal comfort zone during 0 to 83.4 min. These results are averaged over 5×10^3 random initial states used in the learning trials.

Section	Head		Torso		Feet		Mean body	
Method	bang-bang	PPO-RL	bang-bang	PPO-RL	bang-bang	PPO-RL	bang-bang	PPO-RL
Driver	93.32%	99.13%	91.32%	98.89%	71.53%	98.62%	79.57%	98.31%
Passenger	96.76%	99.13%	95.25%	98.63%	76.67%	98.63%	90.92%	98.35%

Figure 5.21 shows the box plot of neutral comfort rate by using RL-based and bang-bang HVAC controllers to maintain driver's head, torso and feet body parts on 5×10^3 randomly initialized episodes. The PPO-RL HVAC controller maintains more than 80% neutral comfort duration for driver's head to feet sections. The black cross marks denote outliers which means 5% cases ranging from 10% to 80%

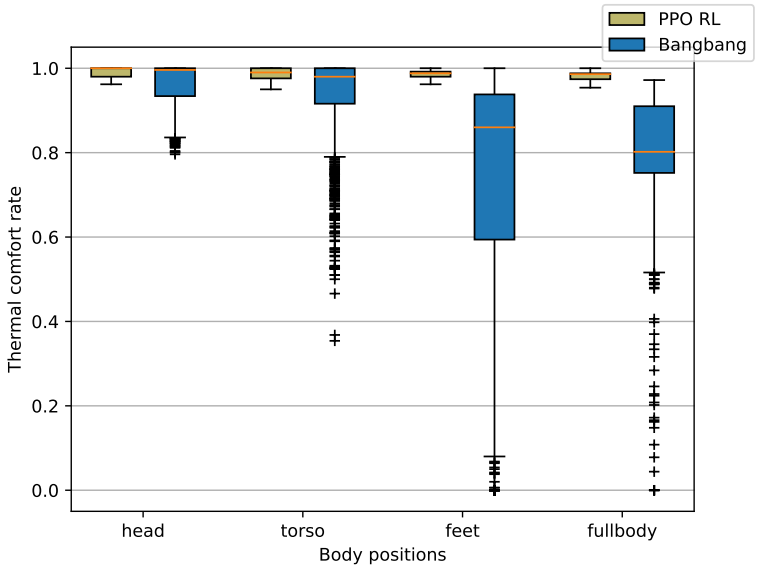

Figure 5.20: Comfort rate of passenger's body positions

neutral comfort. However, the bang-bang controller maintains the neutral comfort rate below 80% for most cases of driver's body sections. As mentioned in Table 5.12, the driver's head, torso and feet neutral comfort rate results are approximately 60% less than the PPO-RL results. And the bang-bang averagely achieve 28.87% neutral comfort for driver's full-body section.

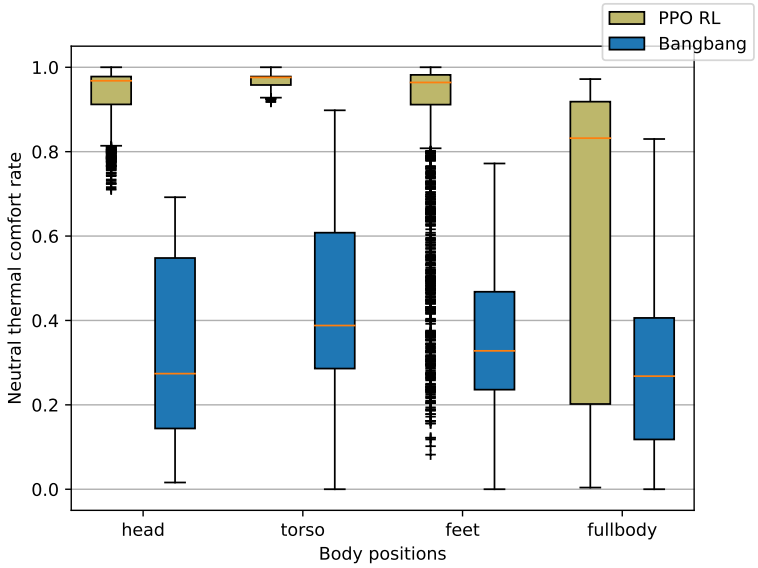

Figure 5.21: Neutral comfort rate of driver's body positions

Figure 5.22 shows the box plot of neutral comfort rate by using RL-based and bang-bang HVAC controllers to maintain passenger's head,

torso and feet body parts on 5×10^3 randomly initialized episodes. The PPO-RL HVAC controller can also maintain more than 70% neutral comfort duration for passenger's head to feet sections. The black cross marks denote outliers which means 5% cases ranging from 22% to 70% neutral comfort. However, the bang-bang controller still maintains the neutral comfort rate below 80% for most cases of passenger's body sections. Table 5.12 also indicates that the driver's head, torso and feet neutral comfort rate results are approximately 50% less than the PPO-RL results. The bang-bang averagely achieve 30.62% neutral comfort for driver's full-body section. This is less than the 53.62% by the RL method. Therefore, the PPO-RL method generally achieves higher neutral comfort proportion compared to the bang-bang method.

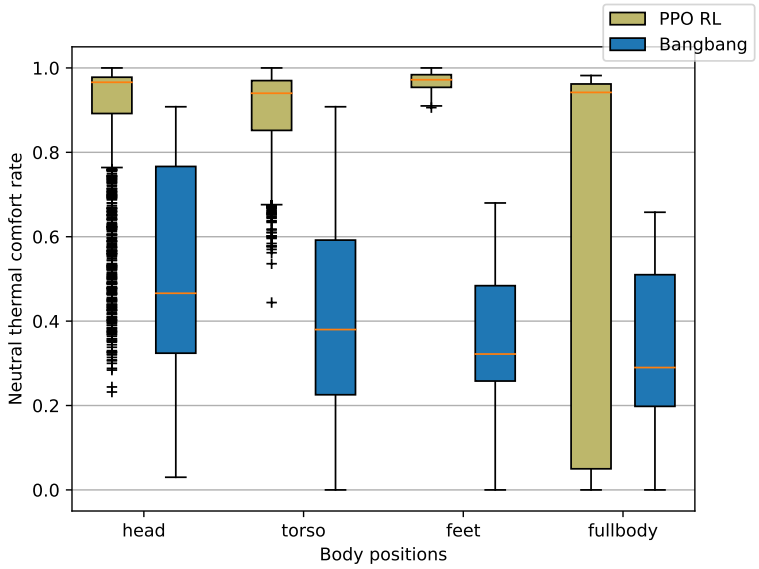

Figure 5.22: Neutral comfort rate of passenger's body positions

Table 5.12: Percentage of time that the PPO-RL and bang-bang controller can maintain occupants in the **neutral thermal comfort** zone during 0 to 83.4 min. These results are averaged over 5×10^3 random initial states used in the learning trials.

Section	Head		Torso		Feet		Mean body	
Method	bang-bang	PPO-RL	bang-bang	PPO-RL	bang-bang	PPO-RL	bang-bang	PPO-RL
Driver	32.67%	94.68%	40.83%	97.11%	32.76%	90.76%	28.87%	62.78%
Passenger	51.44%	92.73%	38.86%	90.25%	35.71%	96.81%	30.62%	53.62%

As mentioned in section 5.3.1 above, in the comfort conditioning processes, the HVAC system consumes energy to either cool down or warm up the occupants air temperatures and mean radiant temperatures to achieve thermal comfort. The energy Q is estimated

with respect to the mass of hot or cold airflow and the amount of energy spent on heating or cooling the circulated air [Lee+15]. The box plot in Figure 5.23 indicates the energy consumption results by using RL-based and bang-bang HVAC controllers to maintain the comfort of occupants' body sections on 5×10^3 randomly initialized episodes. The PPO-RL controller consumes less energy than the bang-

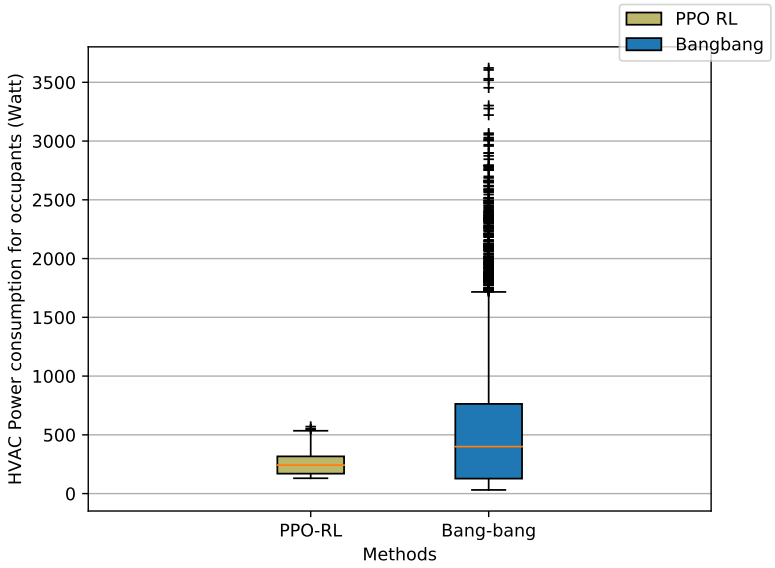

Figure 5.23: The HVAC power consumption for maintaining occupants' body thermal comfort, the mean power by PPO-RL and bang-bang approximately equal 269.3 W and 554.7 W

bang method, as the PPO-RL control averagely consumes 269.3 W less than bang-bang's 554.7 W for the 5×10^3 randomly initialized episodes. Compared with the power consumption values shown in chapter 3, the PPO-RL controller for this "CWT-cabin-env" simulator generally consumes much less power. The reason of this difference is that the airflow amount in "CWT-cabin-env" simulator is less than the cabin model in chapter 3. Because the HVAC system for this CWT cabin model only needs to control the air temperatures and mean radiant temperatures for occupants' body sections rather than the entire cabin's air and interior mass (block) temperatures. Therefore, the PGRL HVAC controller's mean power consumption for cooling or warming the occupants in this "CWT-cabin-env" simulator is fewer than the PGRL HVAC's power consumption in chapter 3.

5.5 CHAPTER SUMMARY

This chapter has presented a set of experiments that extensively investigate the impact of applying the proximal policy optimization (PPO) method as the policy gradient reinforcement learning (PGRL) framework to estimate control policies for a climatic wind tunnel simulation-based car cabin heating, ventilation, air conditioning (HVAC) system. The control policy is also required to achieve and maintain occupants' thermal comfort while reducing energy cost. The experiments mainly compare the PPO-RL approach with the bang-bang control.

The results show that the PPO-RL method generally outperforms the bang-bang controller. More specifically, RL-based controller achieves the target neutral comfort faster than the bang-bang, and RL method can significantly improve the comfort rate for occupants. Therefore, the answer to RQ 3 is listed as follows

- Cool-down and warm-up tests for occupants' body sections: section 5.4.3 indicates that RL-based controller averagely takes 3.32 minutes to achieve neutral comfort reducing the time by 87.27% compared to the bang-bang, RL maintains 97% neutral comfort and improves 258.2% compared to the bang-bang. In section 5.4.4, RL-based controller averagely takes 1.8 minutes to achieve neutral comfort reducing the time by 89% compared to the bang-bang, RL maintains 14.1% neutral comfort and improves 3.5% compared to the bang-bang. In section 5.4.5, RL-based controller averagely takes 0.83 minutes to achieve neutral comfort reducing the time by 87% compared to the bang-bang, RL maintains 67.5% neutral comfort and improve 133.3% compared to the bang-bang.

- Testing cases for random initial cabin thermal state: RL-based method averagely achieves 58.2% neutral thermal comfort for occupants' mean body, while bang-bang only maintains 29.75%. RL-based method averagely consumes 269.3 W compared to bang-bang's 554.7 W.

Therefore, the PPO-RL method significantly improves the comfort maintaining performance and reduces power consumption when compared with traditional bang-bang control technique.

6

CONCLUSIONS

This thesis has investigated policy gradient reinforcement learning (PGRL) approaches to a vehicle Heating, Ventilation and Air Conditioning (HVAC) control task that aims to maximally offer thermal comfort to occupants. In order to employ PGRL as the learning schemes, the corresponded HVAC control policy is designed as a multilayer perceptron (MLP) fully-connected neural network with softmax output layer that can select actions with respect to action distributions. And the learning process is basically maximizing the received rewards so as to update the weights of control policy throughout all the learning trials.

This research examines the implementations of employed PGRL methods with corresponded policy optimization techniques. Among the applied PGRL algorithms, the most suitable learning technique is the proximal policy optimization (PPO) that can ensure non-decreasing rewards in order to maintain policy improvements. The resulting HVAC controller trained by PPO-based PGRL scheme outperforms the State-Action-Reward-State-Action (SARSA)-based controllers by achieving occupant comfort faster and maintaining the higher comfort rate. Based on the PPO-based PGRL HVAC system, the state representation is validated non-Markovian due to the fixed time limit for each episode. The application of an ending state probability then represents the model into an MDP, thus slightly reducing the power consumption and still maintaining high comfort rate. A further experiment focuses on the PGRL application in a car cabin climate control simulation based on climatic wind tunnel experience. In terms of the occupants, the PGRL HVAC controller achieves the neutral comfort faster than the bang-bang method.

This thesis aimed to answer subsequent research questions:

6.1 ANSWERS FOR RESEARCH QUESTIONS

RQ1.1 Can the vehicle HVAC agent, trained by PGRL schemes, reduce the time taken to achieve occupant thermal comfort and keep reasonable energy consumed by HVAC system compared to the SARSA-based learning scheme?

Ans: Yes. The vehicle HVAC controller trained by PPO-based PGRL algorithm can significantly improve the comfort percentage to 77.49% in the corresponded full range testing case. This outperforms the controller trained by trust region policy optimization (TRPO), Mean actor critic (MAC) and Monte-carlo policy gradient (MCPG). Compared with the SARSA-based HVAC controller with corresponded testing cases, the PPO-based one achieves 92.3% of comfort which is higher than 67% by the SARSA-based one, the averaged time taken by the PPO-based control to achieve the occupant comfort is 3.8 minutes, faster than 5.5 minutes done by SARSA-based controller. The energy consumed by PPO-based HVAC system is 0.8713 kW higher than 0.77 kW by the SARSA-based system. This consumption is reasonable, because PPO-based HVAC controller achieves far more comfort rate than SARSA does.

RQ1.2 Can the PGRL HVAC training scheme learn an optimal control policy within a reasonable number of training samples?

Ans: Yes. The number of training episodes is 4000 and each episode comprises 5×10^3 s of past states experience information. Hence, the overall simulated time is 2×10^7 s equivalent to 0.63 years. This result is lower than the 6.3 years of simulated time consumed by the SARSA-based one.

RQ2.1 Is the learning performance of PGRL HVAC negatively impacted by a non-Markovian cabin state representation?

Ans: Yes. The cabin state representation is non-Markovian, because the state observed in the future does not only depend on current state, action but also depends on the episode time-steps. This non-Markovian state representation is validated by comparing the cases that increasing the episode duration from 1×10^3 s to 5×10^3 s can improve the averaged episodic rewards and performance of comfort control. Therefore, the impact of non-Markovian state representation results in longer episode duration (over 4×10^3 s) for training comfort-oriented control policies.

RQ2.2 Can the Markovian-represented cabin state improve the energy efficiency by using the same number of training experience in a non-Markovian state representation?

Ans: Yes. The Markovian state representation introduces an event which has a stationary probability to end the episode when observing

a new state, hence the state is no longer represented to depend on the fixed time limit. The Markovian state representation in the training case with episode duration of 4×10^3 s improves the comfort percentage from 53.58% to 64.32%, and mitigates power consumption from 0.8919 kW to 0.8200 kW. The Markovian representation in the training case with episode duration of 5×10^3 s reduces power consumption from 1.2871 kW to 0.9901 kW, although slightly drives down comfort percentage from 77.94% to 75.88%.

RQ 3: Can the PGRL-based HVAC controller reduce the time taken and power consumption to achieve occupant thermal comfort in a climatic wind tunnel simulation model compared to a bang-bang method?

Ans: Yes. In the cool-down and warm-up testing cases, the PPO-RL HVAC averagely takes 1.98 min to achieve neutral thermal comfort, faster than bang-bang's 17.9min. In the randomly initialized cases, during 0 to 83.3 min, the PPO-RL HVAC controller can averagely maintain 58.2% neutral comfort for occupants' mean body sections, while the bang-bang only maintains 29.75%. But PPO-RL method averagely consumes 269.3 W less than bang-bang's 554.7 W.

6.2 FUTURE WORK

There are several aspects that can be improved for this PGRL HVAC system. As for the reinforcement learning applications, the trust region policy optimization (TRPO) method has a step of using conjugate gradient to estimate Fisher matrix vector product. However a Kronecker-factored approximate curvature (K-FAC) method [MG15] basically uses Kronecker-factored approximation to Fisher information matrix to perform an efficient natural gradient update. The combination of K-FAC and TRPO is named Actor Critic using kronecker-Factored Trust Region (ACKTR) [Wu+17] which has potential to improve TRPO's performance in learning vehicle HVAC control policy. Another possible direction is combining experience replay function with TRPO, namely Actor Critic with Experience Replay (ACER) [Wan+16] that can significantly reduces number of training experience compared with TRPO. In this thesis the proximal policy optimization (PPO) outperforms the rest three applications, but under some poor initializations the PPO is possible to yield convergence to bad local optima. The Trust Region-Guided PPO (TRGPPO) [Wan+19] is newly proposed to adaptively adjust the surrogate clip range and enables better explorations,

sample efficiency than PPO. These learning algorithms have potential to improve the learning efficiency, hence driving down total simulated time below 0.63 years.

Another aspect is about the thermodynamics of cabin environment, because current model is using simple lumped capacitance model, and Torregrosa 's dual zone model [TJ+15] can be more accurate in describing the thermodynamics. Lastly, the thermal comfort model used in this thesis is severely affected by the vent airflow rate, for example, when cabin temperature is close to zero, the high airflow rate drives down occupant's equivalent temperature 10°C to 20°C below zero, but this is unlikely to happen in real-world situations. It is therefore important to build up accurate human comfort model like the Gaussian classification model [Pet18] that can be calibrated to indicate human thermal comfort in real car cabin environment.

Certificate of Ethical Approval

Applicant:

Gaobo Chen

Project Title:

Modelling and Optimisation of a Solar Drying facility

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

09 March 2016

Project Reference Number:

P41688

Certificate of Ethical Approval

Applicant:

Gaobo Chen

Project Title:

Policy-based and deep reinforcement learning application in vehicle heating, ventilation, air conditioning systems control

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

12 January 2018

Project Reference Number:

P65223

Certificate of Ethical Approval

Applicant:

Gaobo Chen

Project Title:

Policy-based reinforcement learning application in heating, ventilation and air conditioning systems control

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

18 September 2018

Project Reference Number:

P75775

A

APPENDIX

A.1 ALGORITHMS

A.1.1 *Traditional value-based RLs and simple policy gradient method*

Algorithm 3 Simple SARSA [SB18] algorithm for a finite state $\in \mathcal{S}$, action $\in \mathcal{A}$ space

Initialize: $Q(s, a)$ for $s \in \mathcal{S}$, a$\in \mathcal{A}$ arbitrarily, and Q(terminal-state,)=0
Repeat: for each episode
    Initialize starting state s_0 for s
    choose a from s using policy derived from $Q(s, a)$ by ϵ-greedy
    Repeat:(for each step of the episode):
        Take action a, observe R, s'
        choose a' from s' using policy derived from $Q(s, a)$ by ϵ-greedy
        $Q(s, a) \leftarrow Q(s, a) + \alpha\,[R + \gamma Q(s', a') - Q(s, a)]$
        $s \leftarrow s'; a \leftarrow a'$
    Until s is terminal
Until the end of training

Algorithm 4 Simple Q-learning, expected SARSA [SB18] algorithms for finite state $\in \mathcal{S}$, action $\in \mathcal{A}$ space

Initialize: $Q(s, a)$ for $s \in \mathcal{S}$, a$\in \mathcal{A}$ arbitrarily, and Q(terminal-state,)=0
Repeat: for each episode
    Initialize starting state s_0 for s
    Repeat:(for each step of the episode):
        choose a from s using policy derived from Q by ϵ-greedy
        Take action a, observe R, s'
        If: Q-learning
        $Q(s, a) \leftarrow Q(s, a) + \alpha\,[R + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
        If: Expected SARSA
        $Q(s, a) \leftarrow Q(s, a) + \alpha\big[R + \gamma \sum_{a' \in \mathbb{A}} \pi(a' \mid s') Q(s', a') - Q(s, a)\big]$
        $s \leftarrow s'$
    Until s is terminal
Until the end of training

Algorithm 5 Simple Monte-Carlo policy gradient [Sut+00], [SB18]

Initialize policy parameter θ_0, learning rate α
for k=0,1,2,... until convergence **do**
  generate trajectory on policy π_{θ_k} with start state s_0
  for step: t= 0,...,T-1 **do**
    start state s_0
    get action $a_t \sim \pi_{\theta_k}(\cdot \mid s_t)$
    execute action a_t on current state s_t receive reward r_t and next state s_{t+1} $(t < T-1)$
  end for
  receive trajectory $\tau = \{s_0, a_0, r_0, s_1, \cdots, s_t, \cdots, s_T\}$
  compute the returns $R_t(\tau) = \sum_{n=t}^{T-1} \gamma^{n-t} \cdot \frac{1}{T} \cdot r(s_n, a_n)$
  calculate baseline $b(s_t)$
  update policy parameter:
  $\theta_{k+1} \leftarrow \theta_k + \alpha \cdot \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \cdot (R_t(\tau) - b(s_t))$
end for
Return: optimal policy parameter $\theta*$

A.1.2 *Advantage actor critic and Mean actor critic*

The Expected SARSA [VS+09] is a practical way to estimate TD-error of $Q^\pi(s, a)$

$$\begin{aligned}\mathbb{E}[\delta] &= r(s_t, a_t) + \gamma \cdot \mathbb{E}[Q(s_{t+1}, a_{t+1})] - Q(s_t, a_t) \\ &= r(s_t, a_t) + \left[\gamma \cdot \sum_{a' \in \mathcal{A}} \pi_\theta(a' \mid s_{t+1}) Q(s_{t+1}, a')\right] \Big(- Q(s_t, a_t)\end{aligned} \tag{A.1}$$

where let $\hat{Q}_\omega(s, a)$ be the accurate estimate of $Q(s, a)$, the learning of $\hat{Q}_\omega(s, a)$ is to minimize $\mathbb{E}[\delta]$.

A.1.3 *Trust region policy optimization*

The trust region optimization (TRPO) [Sch+15a], [Sch16] incorporates the ideas of natural policy gradient [Kak02], importance sampling [SB18] and KL divergence constraint approximation into a conjugate gradient optimization problem. Specifically, this method follows subsequent procedures:

- Estimating the local approximation $L_\pi(\pi')$ of policy improvement $\eta(\pi') - \eta(\pi)$ which refers to the difference of expected rewards resulted from policy π to the newly updated one π'

Algorithm 6 Advantage Actor Critic [Mni+16]

Initialize policy parameter θ_0, learning rate α, value function $\hat{V}_w$ with weight w
for k=0,1,2,... until convergence **do**
  generate trajectory on policy π_{θ_k} with start state s_0 for T steps
  for step: t= 0,...,T-1 **do**
    start state s_0
    get action $a_t \sim \pi_{\theta_k}(\cdot \mid s_t)$
    execute action a_t on current state s_t receive reward r_t and next state s_{t+1} $(t < T-1)$
    Compute $R_t = \sum_{i=0}^{T-t} \gamma^i r(s_t, a_t) + \gamma^{T-t-1} \hat{V}_w(s_{T-1})$ at time step t
    Advantage $\hat{A}(s_t, a_t) = R_t - \hat{V}_w(s_t)$
  end for
  update $w \leftarrow \arg\min_w \sum_{t=0}^{T-1} \left\| R_t - \hat{V}_w(s_t) \right\|^2$
  update policy parameter:
  $\theta_{k+1} \leftarrow \theta_k + \alpha \cdot \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \cdot \hat{A}(s_t, a_t)$
end for
Return: optimal policy parameter $\theta*$

Algorithm 7 Mean Actor Critic with experience replay [Asa+17]

Initialization: Policy $\pi_\theta(a \mid s)$ weight θ, critic network $\hat{Q}_\omega(s, a)$ weight ω
Hyperparameter: learning rate α_A, α_C for policy and critic respectively, discount factor γ, episode length T, size of experience replay buffer D and batch size N, epoch number M for updating critic $\hat{Q}_\omega(s, a)$ (where $D > T$)
repeat
  Inputs: sampling observations, actions and rewards $s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_T, r_T$ from the system by policy $\pi_\theta(a_t \mid s_t)$
  Store: replay buffer stores the episode samples $\{s, a, r, s', a'\}$
  for each updating epoch **do**
    Fix parameter: ω^-
    Random batch: $\{s_t, a_t, r_t, s_{t+1}, a_{t+1}\}_{N \times 1}$
    Update: $\omega \leftarrow \arg\min_\omega \left[r_t + \gamma \cdot \mathbb{E}\left[\hat{Q}_{\omega^-}(s_{t+1}, a_{t+1}) \right] - \hat{Q}_\omega(s_t, a_t) \right]^2$
  end for
  Update: $\omega^- \leftarrow \omega$
  Update: policy $\theta \leftarrow \arg\max_\theta \left[\sum_{a \in A} \nabla_\theta \pi(a \mid s; \theta) \hat{Q}_\omega(s, a) \right]$
until The policy converge or exceeding the maximum episodes
Return: optimal policy parameter $\theta^* = \theta$

- Using KL divergence as the constraint of local approximation $L_\pi(\pi')$, then plugging this constraint optimization into a natural policy gradient (NPG) algorithm process
- Practical TRPO algorithm is combining NPG with conjugate gradient and line search to approximate the Fisher information matrix to estimate the learning step of policy gradient

Given a stochastic policy $\pi(a \mid s) \in [0, 1]$ which maps state to action distributions, the objective function $\eta(\pi)$ is denoted by the expected discounted reward:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \cdots} \left[\sum_{t=0}^{\infty} \gamma^t \cdot r(s_t) \right] \Bigg(= \mathbb{E}_{\tau \sim \pi} [R(\tau)] \tag{A.2}$$

where the initial state $s_0 \sim \rho_0$, action at time step t $a_t \sim \pi(a_t \mid s_t)$ and $s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$. Recall the definitions of state-action function $Q_\pi(s, a)$, the state value function $V_\pi(s)$ and advantage function $A_\pi(s, a)$ are given

$$Q_\pi(s, a) = \mathbb{E}_{s_{t+n}, a_{t+n}, \cdots} \left[\sum_{n=0}^{\infty} \gamma^n \cdot r(s_{t+n}) \right] \Bigg($$

$$V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \cdots} \left[\sum_{n=0}^{\infty} \gamma^n \cdot r(s_{t+n}) \right] \Bigg($$

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$$

where $a_t \sim \pi(a_t \mid s_t), s_{t+1} \sim P(s_{t+1} \mid s_t, a_t), t \geqslant 1$

The expected reward of another different policy $\tilde{\pi}$ subtract expected return of current policy π yields following equation:

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \cdot A_\pi(s_t, a_t) \right] \Bigg(\tag{A.3}$$

where the notation $\mathbb{E}_{\tau \sim \tilde{\pi}}[\cdots]$ indicates actions being sampled by policy $\pi(\cdot \mid s_t)$. This formula can be written into a sum over states instead of time steps

$$\begin{aligned}
\eta(\tilde{\pi}) &= \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s \mid \tilde{\pi}) \sum_{a} \tilde{\pi}(a \mid s) \gamma^t A_\pi(s, a) \\
&= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid \tilde{\pi}) \sum_{a} \tilde{\pi}(a \mid s) A_\pi(s, a) \\
&= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a \mid s) A_\pi(s, a)
\end{aligned} \tag{A.4}$$

where $\rho_{\tilde{\pi}} = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \cdots$ is notified as discounted visitation frequency. For every state s, a non-negative expected advantage function $\sum_a \tilde{\pi}(a \mid s) A_\pi(s, a) \geqslant 0$ is guaranteed to increase current policy performance $\eta(\pi)$ or keep it constant when the expected advantage is zero. However, the inaccurate advantage function approximation unavoidably yields the negative expected advantage $\sum_a \tilde{\pi}(a \mid s) A_\pi(s, a) < 0$ due to the variance of the SGD. In addition, the visitation frequency $\rho_{\tilde{\pi}}(s)$ on $\tilde{\pi}$ makes equation A.4 more complicated to optimize directly. If ignoring the change of state visitation density due to policy changes from $\pi \to \tilde{\pi}$, thus replacing $\rho_{\tilde{\pi}}$ by ρ_π, a local approximation to $\eta(\tilde{\pi})$ is written as

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a \mid s) A_\pi(s, a) \tag{A.5}$$

where the local approximation uses the current state visitation frequency ρ_π, but still satisfies the condition of $L_{\pi_0}(\tilde{\pi} = \pi_0) = \eta(\tilde{\pi} = \pi_0)$, and if given a parameterized policy π_θ with differentiable parameter θ, we can derive the subsequent differential equation

$$\begin{aligned} \nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta) \mid_{\theta=\theta_0} &= \nabla_\theta \eta(\pi_\theta) \mid_{\theta=\theta_0} \\ &= \nabla_\theta \mathbb{E}_{\tau \sim \pi_{\theta_0}} \left[\frac{\pi_\theta(\tau)}{\pi_{\theta_0}(\tau)} R(\tau) \right] \Big|_{\theta=\theta_0} \end{aligned} \tag{A.6}$$

which indicates that an update of policy $\pi_{\theta_0} \to \tilde{\pi}$ that improves $L_{\pi_{\theta_0}}$ also improves η. Based on this issue, Kakade [KL02] has developed a boundary theorem regarding to a conservative policy iteration problem. In this case, π_{old} $\pi' = \arg\max_{\pi'} L_{\pi_{\text{old}}}(\pi')$ denotes current policy and deterministic mixture policy. The upgraded new policy π_{new} is denoted by

$$\pi_{\text{new}}(a \mid s) = (1 - \alpha)\pi_{\text{old}}(a \mid s) + \alpha \pi'(a \mid s) \tag{A.7}$$

and the boundary is found below:

$$\begin{aligned} &\eta(\pi_{\text{new}}) \geqslant L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2, \\ &\text{where } \epsilon = \max_s \left| \mathbb{E}_{a \sim \pi'(a|s)} \left[A_\pi(s, a) \right] \right| \end{aligned} \tag{A.8}$$

In Schulman's TRPO research [Sch+15a], this boundary can be extended to general stochastic policy upgrade cases with the definition of expected advantage $\bar{A}(s)$ and the probability of taking different ac-

tions $P(a \neq \tilde{a} \mid s)$ at state s. Firstly the scenario of expected advantage of a new policy $\tilde{\pi}$ over the old one π at state s is given by

$$\bar{A}(s) = \sum_{a} \tilde{\pi}(a \mid s) A_{\pi}(s, a) = \mathbb{E}_{a \sim \tilde{\pi}(\cdot|s)} \left[A_{\pi}(s, a) \right] \tag{A.9}$$

which can be plugged into the terms defined in equation A.4 and equation A.5, hence obtaining the theoretical and local advantage update:

$$\begin{aligned} \eta(\tilde{\pi}) &= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a \mid s) A_{\pi}(s, a) = \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \bar{A}(s) \\ L_{\pi}(\tilde{\pi}) &= \eta(\pi) + \sum_{s} \rho_{\pi}(s) \sum_{a} \tilde{\pi}(a \mid s) A_{\pi}(s, a) = \eta(\pi) + \sum_{s} \rho_{\pi}(s) \bar{A}(s) \end{aligned}$$

where the theoretical update $\eta(\tilde{\pi})$ is sampling states using policy $\tilde{\pi}$ and local approximation $L_{\pi}(\tilde{\pi})$ using π, these twos terms can also be written with respect to $\tilde{\pi}$ and π:

$$\begin{aligned} \eta(\tilde{\pi}) &= \eta(\pi) + \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^{t} \bar{A}(s_t) \right] \Bigg(\\ L_{\pi}(\tilde{\pi}) &= \eta(\pi) + \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^{t} \bar{A}(s_t) \right] \Bigg(\end{aligned} \tag{A.10}$$

While the difference between $L_{\pi}(\tilde{\pi})$ and $\eta(\tilde{\pi})$ satisfies following scenario:

$$|\eta(\tilde{\pi}) - L_{\pi}(\tilde{\pi})| = \sum_{t=0}^{\infty} \gamma^{t} \Big| \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\bar{A}(s_t) \right] \Bigg(- \mathbb{E}_{\tau \sim \pi} \left[\bar{A}(s_t) \right] \Bigg(\Big| \leqslant \frac{4\alpha^2 \gamma \epsilon}{(1-\gamma)^2} \tag{A.11}$$

where α represents the total variation divergence of policy pair $D_{TV}(\pi(\cdot \mid s) \parallel \tilde{\pi}(\cdot \mid s))$, ϵ denotes maximal advantage $\max_{a,s} |A_{\pi}(s, a)|$ and γ is the discounted factor. As the total variation divergence of two distributions p and q satisfies is less than their KL divergence $D_{TV}(p \parallel q)^2 \leqslant D_{KL}(p \parallel q)$. If accounting for the maximal total variation divergence $D_{TV}^{max}(\pi \parallel \tilde{\pi})$, the constraint defined in equation A.11 can be rewritten as

$$\begin{aligned} \eta(\tilde{\pi}) \geqslant L_{\pi}(\tilde{\pi}) - C \cdot D_{TV}^{max}(\pi \parallel \tilde{\pi})^2 \geqslant L_{\pi}(\tilde{\pi}) - C \cdot D_{KL}^{max}(\pi \parallel \tilde{\pi}) \\ \text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2} \end{aligned} \tag{A.12}$$

According to the definition of local approximation in equation A.6, it is understood that a policy update improves local approximation $L_\pi(\tilde{\pi})$ also improves actual expected rewards $\eta(\pi)$. Combined with terms of equation A.12, performing a maximization of $L_\pi(\tilde{\pi}) - C \cdot D_{KL}^{max}(\pi \parallel \tilde{\pi})$ also guarantees maximization of $\eta(\tilde{\pi})$. Algorithm 8 details a strait forward solution of optimizing the policy iteratively while ensuring a non-decreasing expected return $\eta(\tilde{\pi})$. Where the core objective is to estimate a policy π satisfying

Algorithm 8 Policy update that ensures non-negative improvement of expected return η [Sch+15a]

1: Initialize Policy π_0
2: **for** i=0,1,2,... until convergence **do**
3: Compute advantage values $A_{\pi_i}(s, a)$
4: Plug $A_{\pi_i}(s, a)$ into
5: $L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a \mid s) A_{\pi_i}(s, a)$
6: With the definition $C = \frac{4\gamma}{(1-\gamma)^2} \max_{a,s} |A_\pi(s, a)|$
7: Update policy $\pi_{i+1} = \arg\max_\pi \left[L_{\pi_i}(\pi) - C \cdot D_{KL}^{max}(\pi_i, \pi) \right]$
8: **end for**
9: **Return**: optimal policy parameter $\pi*$

$\max_\pi \left[L_{\pi_{old}}(\pi) - C \cdot D_{KL}^{max}(\pi_{old} \parallel \pi) \right]$. While directly using penalty coefficient C in the term $\pi_{i+1} = \arg\max_\pi \left[L_{\pi_i}(\pi) - C \cdot D_{KL}^{max}(\pi_i, \pi) \right]$ can result in tiny learning step as discount factor γ is close to zero [Ach+17]. In order to avoid small step size while using penalty coefficient C,the problem is derived into a KL divergence constrained optimization between the new policy and old policy

$$\max_\pi \left[L_{\pi_{old}}(\pi) \right] \quad \text{subject to} \quad D_{KL}^{max}(\pi_{old} \parallel \pi) \leqslant \delta \tag{A.13}$$

where δ is the upper boundary of KL constraint. Using the policy parameterization $\pi_\theta(\cdot \mid s)$ instead of $\pi(\cdot \mid s)$, therefore the following terms with policy parameterization are simplified as follows

$$\begin{aligned} L_{\theta_{old}}(\theta) &= L_{\pi_{\theta_{old}}}(\pi_\theta), \quad \eta(\theta) = \eta(\pi_\theta) \\ D_{KL}(\theta_{old}, \theta) &= D_{KL}(\pi_{\theta_{old}} \parallel \pi_\theta) \\ A_{\pi_\theta}(s, a) &= A_\theta(s, a), \quad \rho_{\pi_\theta}(s) = \rho_\theta(s) \end{aligned}$$

Therefore the problem becomes an optimization with respect to policy parameter θ and D_{KL}^{max} can be practically replaced by an averaged term $\bar{D}_{KL}$ and according to the background of natural policy gradi-

ent [Kak02], [Raj+17]. The simplified objective functions are presented below

$$\max_{\theta} [L_{\theta_{old}}(\theta)] \quad \text{subject to} \quad \bar{D}_{KL}(\theta_{old}, \theta) \leqslant \delta \tag{A.14}$$

this constraint optimization problem can be solved by natural gradient approach. The first step is to compute linear approximation to $L_{\theta_{old}}(\theta)$ and quadratic approximation to $\bar{D}_{KL}(\theta_{old}, \theta)$ respectively, therefore the terms of (A.14) are approximated as follows

$$\begin{aligned} &\max_{\theta} [\nabla_\theta L_{\theta_{old}}(\theta) \mid_{\theta=\theta_{old}} \cdot(\theta - \theta_{old})] \\ &\text{subject to} \quad \frac{1}{2}(\theta - \theta_{old})^T F (\theta - \theta_{old}) \leqslant \delta \\ &\text{where } F = \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \left[\nabla_\theta \log \pi_\theta(a \mid s) (\nabla_\theta \log \pi_\theta(a \mid s))^T\right] \mid_{\theta=\theta_{old}} \end{aligned} \tag{A.15}$$

the term F represents a fisher information matrix which is positive definite and symmetric. According to the definition of natural gradient [Ama98], the complete update for policy parameter θ is given

$$\theta = \theta_{old} + \sqrt{\frac{2\delta}{g^T F^{-1} g}} F^{-1} g \quad \text{where} \quad g = \nabla_\theta L_{\theta_{old}}(\theta) \mid_{\theta=\theta_{old}} \tag{A.16}$$

According to definition in equation A.6, the objective of maximizing term $L_{\theta_{old}}(\theta)$ can be denoted in terms of an importance sampling form:

$$\max_{\theta} [L_{\theta_{old}}(\theta)] \leftarrow \max_{\theta} \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta old}(a_t \mid s_t)} R_t\right] \tag{A.17}$$

where R_t can be replaced by the advantage function estimator $\hat{A}_t$ or generalized one $\hat{A}_t^{GAE(\gamma,\lambda)}$ according to section.2.3.3. In order to maintain policy exploration, an entropy term $\mathbb{H}$ can be added to the objective function $L_{\theta_{old}}(\theta)$. According to the definition of policy entropy in Ahmed et al [Ahm+19], the term is indicated as follow

$$\mathbb{H}_\theta(s_t) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s_t)} [-\log \pi_\theta(a \mid s_t)] \tag{A.18}$$

and this term is usually weighted with a coefficient $C_{\mathcal{H}}$.

Algorithm 9 TRPO algorithm [Sch16], [Sch+15a]

Initialize Policy parameter θ_0, value function weight ω_0, entropy coefficient $C_{\mathcal{H}}$, KL upper bound δ (0.01 in default), GAE discount λ
for k=0,1,2,... until convergence **do**
  Collect trajectory τ_k on policy π_{θ_k}
  Compute advantage values $A_{\theta_k}(s,a)$ or $A^{GAE(\gamma,\lambda)}_{\theta_k}(s,a)$
  Estimating policy gradient
  $g_k \leftarrow \nabla_\theta \left[L_{\theta_{old}}(\theta) + C_{\mathcal{H}} \cdot \mathbb{H}_\theta(s)\right]|_{\theta=\theta_{old}}$
  Fisher information matrix $F_k^{-1} g_k$ by conjugate gradient
  Natural gradient update step $\Delta_{\theta_k} \approx \sqrt{\frac{2\delta}{g_k^T F_k^{-1} g_k}} F_k^{-1} g_k$
  Perform Line search with exponential decay to estimate final update
  for j=0,1,2,...,L do **do**
    Compute update $\theta = \theta_k + \alpha^j \Delta_{\theta_k}$
    if $L_{\theta_k}(\theta) \geqslant 0$ and $\bar{D}_{KL}(\theta \parallel \theta_k) \leqslant \delta$ **then**
      accept the update $\Delta_k = \alpha^j \Delta_{\theta_k}$
      break
    end if
  end for
  Update policy parameter $\theta_{k+1} = \theta_k + \Delta_k$
end for
Return: optimal policy parameter $\theta*$

A.1.4 *Proximal policy optimization*

As mentioned in last chapter, the policy optimization objective is constrained by the KL-divergence of old and updated policies.

$$\begin{aligned} &\max_\theta \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta old}(a_t \mid s_t)} \hat{A}_t\right] \\ &\text{subject to} \quad \hat{\mathbb{E}}_t \left[KL\left[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)\right]\right] \leqslant \delta \end{aligned} \tag{A.19}$$

where the TRPO approach uses linear approximation to the objective function and quadratic approximation to the constraint, then using conjugate gradient to approximate fisher information matrix for the natural gradient update of policy parameter θ. Conversely, this constrained problem can be represented in an unconstrained optimization objective

$$\max_\theta \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta old}(a_t \mid s_t)} \hat{A}_t - \beta KL\left[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)\right]\right] \tag{A.20}$$

However, the choice of penalty coefficient β impacts the performance and it is difficult to use a fixed β that can generally ensure improved

policy over the whole learning trials. To deal with this constrained policy optimization problem by simply using first-order gradient, Schulman et al. [Sch+17] proposed a simple clip surrogate objective in order to constrain the change that moves probability ratio $w_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$ far from 1. The clip objective function is shown below

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(w_t(\theta)\hat{A}_t, clip\left(w_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{A.21}$$

where the term $clip\left(w_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t$ can avoid excessive policy update by clipping the probability ratio and always keep the policy update within the interval $[1-\epsilon, 1+\epsilon]$. Practical choices for ϵ between 0.1 and 0.2 according to studies by Schulman et al [Sch+17]. The algorithm 10 shows valid optimization process. The Adam optimizer [KB14] can efficiently calculate the gradient of objective function $L^{CLIP}(\theta)$.

Algorithm 10 The proximal policy optimization (PPO) algorithm [Sch+17]

Initialize Policy parameter θ_0, value function weight ω_0, clip ratio ϵ, entropy coefficient $C_{\mathcal{H}}$, GAE discount λ
for k=0,1,2,... until convergence **do**
  generate trajectory τ_k on policy π_{θ_k} with start state s_0
  for step: t= 0,...,m **do**
    start state s_0
    get action $a_t \sim \pi_{\theta_k}(\cdot \mid s_t)$
    execute action a_t on current state s_t receive reward r_t and next state s_{t+1} ($t \leqslant m-1$)
    record trajectory $\tau_k = \{s_0, a_0, r_0, s_1, \cdots, s_m\}$ into memory set $\mathcal{D}$
  end for
  $\theta_{old} \leftarrow \theta_k$
  for each update step **do**
    Sampling mini-batch with N samples of $\{(s_i, a_i, r_i, s_{i+1})\}$ from $\mathcal{D}$
    Computing value approximator by using TD(λ)
    Target value function $y_{i+1} = r(a_i, s_i) + \gamma \cdot V_\omega(s_{i+1})$
    $\omega \leftarrow \arg\min_\omega \|y_{i+1} - V_\omega(s_i)\|^2$
    Compute advantage function $\hat{A}_i$ or $\hat{A}_i^{GAE(\gamma,\lambda)}$ w.r.t $V_\omega(s_i)$ and $R(\tau)$
    $w_i(\theta_k) \leftarrow \frac{\pi_{\theta_k}(a_i|s_i)}{\pi_{\theta_{old}}(a_i|s_i)}$
    $L^{CLIP}(\theta_k) = \hat{\mathbb{E}}_t[\min\left(w_i(\theta_k)\hat{A}_i, clip(w_i(\theta_k), 1-\epsilon, 1+\epsilon)\hat{A}_i\right) + C_{\mathcal{H}} \cdot \mathbb{H}_\theta(s_i))]$
    Estimating policy gradient update $\theta_{k+1} \leftarrow \arg\max_\theta L^{CLIP}(\theta_k)$
  end for
end for
Return: optimal policy parameter $\theta*$

BIBLIOGRAPHY

[Abb+07] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. "An application of reinforcement learning to aerobatic helicopter flight". In: *Advances in neural information processing systems*. 2007, pp. 1–8 (cit. on p. 26).

[Ach+17] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. "Constrained policy optimization". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 22–31 (cit. on p. 121).

[Ahm+19] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. "Understanding the impact of entropy on policy optimization". In: *International Conference on Machine Learning*. 2019, pp. 151–160 (cit. on p. 122).

[Ama98] Shun-Ichi Amari. "Natural gradient works efficiently in learning". In: *Neural computation* 10.2 (1998), pp. 251–276 (cit. on pp. 24, 122).

[Asa+17] Kavosh Asadi, Cameron Allen, Melrose Roderick, Abdelrahman Mohamed, George Konidaris, Michael Littman, and Brown University Amazon. "Mean actor critic". In: *stat* 1050 (2017), p. 1 (cit. on pp. 23, 117).

[BL15] Enda Barrett and Stephen Linder. "Autonomous hvac control, a reinforcement learning approach". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2015, pp. 3–19 (cit. on p. 28).

[BAAB10] Wafa Batayneh, Omar Al-Araidah, and Khaled Bataineh. "Fuzzy logic approach to provide safe and comfortable indoor environment". In: *Int. J. Eng. Sci. Technol* 2.7 (2010) (cit. on p. 1).

[BB01] Jonathan Baxter and Peter L Bartlett. "Infinite-horizon policy-gradient estimation". In: *Journal of Artificial Intelligence Research* 15 (2001), pp. 319–350 (cit. on p. 21).

[BB+00] Jonathan Baxter, Peter L Bartlett, et al. "Reinforcement learning in POMDP's via direct gradient ascent". In: *ICML*. Citeseer. 2000, pp. 41–48 (cit. on p. 20).

[Bel66] Richard Bellman. "Dynamic programming". In: *Science* 153.3731 (1966), pp. 34–37 (cit. on p. 14).

[BR19] Jalaj Bhandari and Daniel Russo. "Global Optimality Guarantees For Policy Gradient Methods". In: *arXiv preprint arXiv:1906.01786* (2019) (cit. on p. 24).

[Bøh+19] Eivind Bøhn, Erlend M Coates, Signe Moe, and Tor Ame Johansen. "Deep Reinforcement Learning Attitude Control of Fixed-Wing UAVs Using Proximal Policy Optimization". In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2019, pp. 523–533 (cit. on p. 26).

[BD95] Steven J Bradtke and Michael O Duff. "Reinforcement learning methods for continuous-time Markov decision problems". In: *Advances in neural information processing systems*. 1995, pp. 393–400 (cit. on p. 16).

[Bro+16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016) (cit. on pp. 3, 24).

[Bru+18] James Brusey, Diana Hintea, Elena Gaura, and Neil Beloe. "Reinforcement learning-based thermal comfort control for vehicle cabins". In: *Mechatronics* 50 (2018), pp. 413–421 (cit. on pp. 3–5, 30, 33, 37, 39, 40, 42, 47, 50, 59, 60).

[BR20] James Brusey and Matteo Rostagno. *Efficient Cabin Model for Simulating Thermal and Acoustic Behaviour of Car Cabins*. [Online; accessed 14-April-2021]. 2020. URL: https://www.domus-project.eu/wp-content/uploads/2020/06/D1.5-PubSum.pdf (cit. on pp. 75, 77, 83).

[Cas98] Anthony R Cassandra. "A survey of POMDP applications". In: *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*. Vol. 1724. 1998 (cit. on p. 62).

[Che+18] Xi Chen, Ali Ghadirzadeh, John Folkesson, Mårten Björkman, and Patric Jensfelt. "Deep reinforcement learning to acquire navigation skills for wheel-legged robots in complex environments". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3110–3116 (cit. on p. 25).

[Cro+15] Cristiana Croitoru, Ilinca Nastase, Florin Bode, Amina Meslem, and Angel Dogeanu. "Thermal comfort models for indoor spaces and vehicles—Current capabilities and future perspectives". In: *Renewable and Sustainable Energy Reviews* 44 (2015), pp. 304–318 (cit. on p. 1).

[DTN10] Noël Djongyang, René Tchinda, and Donatien Njomo. "Thermal comfort: A review paper". In: *Renewable and sustainable energy reviews* 14.9 (2010), pp. 2626–2640 (cit. on p. 1).

[Doyoo] Kenji Doya. "Reinforcement learning in continuous time and space". In: *Neural computation* 12.1 (2000), pp. 219–245 (cit. on p. 16).

[Dua+16] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. "Benchmarking deep reinforcement learning for continuous control". In: *International Conference on Machine Learning*. 2016, pp. 1329–1338 (cit. on pp. 3, 10, 25).

[Eck+16] Julian Eckstein, Christopher Lüke, Frederik Brunstein, Patrick Friedel, Ulrich Köhler, and Ansgar Trächtler. "A novel approach using model predictive control to enhance the range of electric vehicles". In: *Procedia Technology* 26 (2016), pp. 177–184 (cit. on p. 2).

[Ene17] Diana Enescu. "A review of thermal comfort models and indicators for indoor environments". In: *Renewable and Sustainable Energy Reviews* 79 (2017), pp. 1353–1379 (cit. on p. 1).

[Fan+70] Poul O Fanger et al. "Thermal comfort. Analysis and applications in environmental engineering." In: *Thermal comfort. Analysis and applications in environmental engineering*. (1970) (cit. on p. 27).

[Faz+14] Pedro Fazenda, Kalyan Veeramachaneni, Pedro Lima, and Una-May O'Reilly. "Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems". In: *Journal of Ambient Intelligence and Smart Environments* 6.6 (2014), pp. 675–690 (cit. on p. 28).

[Fer+12] Pedro M Ferreira, Sérgio M Silva, António E Ruano, Aldric T Negrier, and Eusebio ZE Conceicao. "Neural network PMV estimation for model-based predictive control of HVAC systems". In: *Neural Networks (IJCNN), The 2012*

International Joint Conference on. IEEE. 2012, pp. 1–8 (cit. on p. 2).

[Foj+17] Miloš Fojtlín, Jan Fišer, Jan Pokorný, Aleš Povalač, Tomáš Urbanec, and Miroslav Jícha. "An innovative HVAC control system: Implementation and testing in a vehicular cabin". In: *Journal of Thermal Biology* 70 (2017), pp. 64–68 (cit. on p. 2).

[Foj+16] Miloš Fojtlín, Michal Planka, Jan Fišer, Jan Pokorný, and Miroslav Jícha. "Airflow measurement of the car HVAC unit using hot-wire anemometry". In: *EPJ web of conferences*. Vol. 114. EDP Sciences. 2016, p. 02023 (cit. on p. 39).

[GWZ99] Chris Gaskett, David Wettergreen, and Alexander Zelinsky. "Q-learning in continuous state and action spaces". In: *Australasian Joint Conference on Artificial Intelligence*. Springer. 1999, pp. 417–428 (cit. on p. 16).

[Hin14] Diana Hintea. "Reinforcement Learning-based Thermal Comfort for Vehicle Cabins". PhD thesis. Coventry University, 2014 (cit. on p. 2).

[Hin+13] Diana Hintea, James Brusey, Elena I Gaura, John Kemp, and Neil Beloe. "Comfort in Cars-Estimating Equivalent Temperature for Comfort Driven Heating, Ventilation and Air Conditioning (HVAC) Control." In: *ICINCO (1)*. 2013, pp. 507–513 (cit. on p. 2).

[Hin+14] Diana Hintea, John Kemp, James Brusey, Elena Gaura, and Neil Beloe. "Applicability of thermal comfort models to car cabin environments". In: *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Vol. 1. IEEE. 2014, pp. 769–776 (cit. on pp. 33, 37, 38, 42).

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366 (cit. on p. 17).

[Hua94] Bin-Juine Huang. "Transient performance of solar systems with a bang-bang controller". In: *JOURNAL-CHINESE SOCIETY OF MECHANICAL ENGINEERS* 15 (1994), pp. 409–409 (cit. on p. 80).

[Ibr+12] BSKK Ibrahim, MAN Aziah, S Ahmad, R Akmeliawati, HMI Nizam, AGA Muthalif, SF Toha, and MK Hassan. "Fuzzy-based temperature and humidity control for HV AC of electric vehicle". In: *Procedia Engineering* 41 (2012), pp. 904–910 (cit. on p. 2).

[Jia+19] Ruoxi Jia, Ming Jin, Kaiyu Sun, Tianzhen Hong, and Costas Spanos. "Advanced Building Control via Deep Reinforcement Learning". In: *Energy Procedia* 158 (2019), pp. 6158–6163 (cit. on p. 29).

[JAW07] Sherry Everett Jones, Robert Axelrad, and Wendy A Wattigney. "Healthy and safe school environment, Part II, Physical school environment: Results from the School Health Policies and Programs Study 2006". In: *Journal of School Health* 77.8 (2007), pp. 544–556 (cit. on p. 1).

[Joy11] James M Joyce. "Kullback-leibler divergence". In: *International encyclopedia of statistical science* (2011), pp. 720–722 (cit. on p. 24).

[Kak02] Sham M Kakade. "A natural policy gradient". In: *Advances in neural information processing systems*. 2002, pp. 1531–1538 (cit. on pp. 116, 122).

[KL02] Sham Kakade and John Langford. "Approximately optimal approximate reinforcement learning". In: *ICML*. Vol. 2. 2002, pp. 267–274 (cit. on p. 119).

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on pp. 24, 35, 36, 124).

[KBP13] Jens Kober, J Andrew Bagnell, and Jan Peters. "Reinforcement learning in robotics: A survey". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274 (cit. on p. 24).

[Koc+19] William Koch, Renato Mancuso, Richard West, and Azer Bestavros. "Reinforcement learning for uav attitude control". In: *ACM Transactions on Cyber-Physical Systems* 3.2 (2019), p. 22 (cit. on p. 26).

[Kur+18] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. "Model-ensemble trust-region policy optimization". In: *arXiv preprint arXiv:1802.10592* (2018) (cit. on p. 25).

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436 (cit. on p. 17).

[Lee+15] Hoseong Lee, Yunho Hwang, Ilguk Song, and Kilsang Jang. "Transient thermal model of passenger car's cabin and implementation to saturation cycle with alternative working fluids". In: *Energy* 90 (2015), pp. 1859–1868 (cit. on pp. 2, 37, 83, 104).

[LX15] Bocheng Li and Li Xia. "A multi-grid reinforcement learning method for energy conservation and comfort of HVAC in buildings". In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2015, pp. 444–449 (cit. on p. 27).

[LD05] Jian Liang and Ruxu Du. "Thermal comfort control based on neural network for HVAC application". In: *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*. IEEE. 2005, pp. 819–824 (cit. on p. 2).

[Lil+15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971* (2015) (cit. on p. 26).

[LYB07] Yang Liu, Hongnian Yu, and Brian Burrows. "Optimization and Control of a Pendulum-driven Cart-pole System". In: *2007 IEEE International Conference on Networking, Sensing and Control*. IEEE. 2007, pp. 151–156 (cit. on pp. 11, 61).

[Lop+18] Guilherme Cano Lopes, Murillo Ferreira, Alexandre da Silva Simões, and Esther Luna Colombini. "Intelligent Control of a Quadrotor with Proximal Policy Optimization Reinforcement Learning". In: *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*. IEEE. 2018, pp. 503–508 (cit. on p. 26).

[MWL18] Yiyi Mao, Ji Wang, and Junming Li. "Experimental and numerical study of air flow and temperature variations in an electric vehicle cabin during cooling and heating". In: *Applied Thermal Engineering* 137 (2018), pp. 356–367 (cit. on p. 2).

[MG15] James Martens and Roger Grosse. "Optimizing neural networks with kronecker-factored approximate curvature". In: *International conference on machine learning*. 2015, pp. 2408–2417 (cit. on p. 109).

[McD06] Robert McDowall. *Fundamentals of HVAC systems*. Academic Press, 2006 (cit. on p. 1).

[MPD02] José Del R Millán, Daniele Posenato, and Eric Dedieu. "Continuous-action Q-learning". In: *Machine Learning* 49.2-3 (2002), pp. 247–265 (cit. on p. 16).

[MR13] Asit Kumar Mishra and Maddali Ramgopal. "Field studies on human thermal comfort—an overview". In: *Building and Environment* 64 (2013), pp. 94–106 (cit. on p. 1).

[Mni+16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning". In: *International conference on machine learning*. 2016, pp. 1928–1937 (cit. on pp. 22, 23, 29, 117).

[Mni+13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013) (cit. on pp. 17, 18, 25, 27).

[Mur00] Kevin P Murphy. "A survey of POMDP solution techniques". In: *environment* 2 (2000), p. X3 (cit. on p. 35).

[Ng03] Andrew Y Ng. "Shaping and policy search in reinforcement learning". PhD thesis. University of California, Berkeley Berkeley, 2003 (cit. on pp. 26, 61).

[Nil04] Håkan O Nilsson. *Comfort climate evaluation with thermal manikin methods and computer simulation models*. 2004 (cit. on pp. 2, 30, 38, 83, 86).

[Ove+19] Anders Overgaard, Brian Kongsgaard Nielsen, Carsten Skovmose Kallesøe, and Jan Dimon Bendtsen. "Reinforcement Learning for Mixing Loop Control with Flow Variable Eligibility Trace". In: *2019 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2019, pp. 1043–1048 (cit. on p. 28).

[Pen+18] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills". In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), p. 143 (cit. on p. 25).

[PS06] Jan Peters and Stefan Schaal. "Policy gradient methods for robotics". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 2219–2225 (cit. on pp. 3, 21, 29).

[PS08a] Jan Peters and Stefan Schaal. "Natural actor-critic". In: *Neurocomputing* 71.7-9 (2008), pp. 1180–1190 (cit. on p. 24).

[PS08b] Jan Peters and Stefan Schaal. "Reinforcement learning of motor skills with policy gradients". In: *Neural networks* 21.4 (2008), pp. 682–697 (cit. on p. 21).

[PVS03] Jan Peters, Sethu Vijayakumar, and Stefan Schaal. "Reinforcement learning for humanoid robotics". In: *Proceedings of the third IEEE-RAS international conference on humanoid robots*. 2003, pp. 1–20 (cit. on p. 24).

[Pet18] Alexandra Petre. "User feedback-based reinforcement learning for vehicle comfort control". PhD thesis. Coventry University, 2018 (cit. on pp. 5, 110).

[PR13] David Portugal and Rui P Rocha. "Distributed multi-robot patrol: A scalable and fault-tolerant framework". In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1572–1587 (cit. on p. 61).

[Qi14] Zhaogang Qi. "Advances on air conditioning and heat pump system in electric vehicles–A review". In: *Renewable and Sustainable Energy Reviews* 38 (2014), pp. 754–764 (cit. on p. 2).

[Raj+17] Aravind Rajeswaran, Kendall Lowrey, Emanuel V Todorov, and Sham M Kakade. "Towards generalization and simplicity in continuous control". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6550–6561 (cit. on pp. 39, 122).

[RPS07] Martin Riedmiller, Jan Peters, and Stefan Schaal. "Evaluation of policy gradient methods and variants on the cart-pole benchmark". In: *2007 IEEE International Symposium*

on Approximate Dynamic Programming and Reinforcement Learning. IEEE. 2007, pp. 254–261 (cit. on p. 21).

[Ros14] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014 (cit. on p. 63).

[RVL15] Ricardo Forgiarini Rupp, Natalia Giraldo Vásquez, and Roberto Lamberts. "A review of human thermal comfort in the built environment". In: *Energy and Buildings* 105 (2015), pp. 178–205 (cit. on p. 1).

[RWP15] A Ryniecki, Jolanta Wawrzyniak, and Agnieszka Anna Pilarska. "Basic terms of process control: the on-off control system". In: *Przemysł Spożywczy* 69.11 (2015), pp. 26–29 (cit. on p. 80).

[SSR97] Juan C Santamaría, Richard S Sutton, and Ashwin Ram. "Experiments with reinforcement learning in problems with continuous state and action spaces". In: *Adaptive behavior* 6.2 (1997), pp. 163–217 (cit. on p. 17).

[Sch16] John Schulman. "Optimizing expectations: From deep reinforcement learning to stochastic computation graphs". PhD thesis. UC Berkeley, 2016 (cit. on pp. 3, 22, 24, 36, 40, 116, 123).

[Sch+15a] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust region policy optimization". In: *International Conference on Machine Learning*. 2015, pp. 1889–1897 (cit. on pp. 24, 29, 36, 116, 119, 121, 123).

[Sch+15b] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. "High-dimensional continuous control using generalized advantage estimation". In: *arXiv preprint arXiv:1506.02438* (2015) (cit. on p. 22).

[Sch+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017) (cit. on pp. 3, 18, 24, 36, 124, 125).

[She+19] Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. "Hessian Aided Policy Gradient". In: *International Conference on Machine Learning*. 2019, pp. 5729–5738 (cit. on p. 23).

[Sil+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. "Mastering the game of go without human knowledge". In: *Nature* 550.7676 (2017), p. 354 (cit. on p. 5).

[ST09] David Silver and Gerald Tesauro. "Monte-Carlo simulation balancing". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 945–952 (cit. on pp. 20, 35).

[SSU16] Mihaela Simion, Lavinia Socaciu, and Paula Unguresan. "Factors which influence the thermal comfort inside of vehicles". In: *Energy Procedia* 85 (2016), pp. 472–480 (cit. on p. 2).

[Sut96] Richard S Sutton. "Generalization in reinforcement learning: Successful examples using sparse coarse coding". In: *Advances in neural information processing systems*. 1996, pp. 1038–1044 (cit. on p. 17).

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on pp. 4, 9, 10, 13, 16, 17, 20, 115, 116).

[Sut+00] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems*. 2000, pp. 1057–1063 (cit. on pp. 20, 29, 116).

[Tal+13] Mohammad Taleghani, Martin Tenpierik, Stanley Kurvers, and Andy Van Den Dobbelsteen. "A review into thermal comfort in buildings". In: *Renewable and Sustainable Energy Reviews* 26 (2013), pp. 201–215 (cit. on p. 1).

[Tas+18] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. "Deepmind control suite". In: *arXiv preprint arXiv:1801.00690* (2018) (cit. on pp. 3, 24).

[TJ+15] Bárbara Torregrosa-Jaime, Filip Bjurling, José M Corberán, Fausto Di Sciullo, and Jorge Payá. "Transient thermal model of a vehicle's cabin validated under variable ambient conditions". In: *Applied Thermal Engineering* 75 (2015), pp. 45–53 (cit. on pp. 83, 110).

[Val+19] William Valladares, Marco Galindo, Jorge Gutiérrez, Wu-Chieh Wu, Kuo-Kai Liao, Jen-Chung Liao, Kuang-Chin Lu, and Chi-Chuan Wang. "Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm". In: *Building and Environment* (2019) (cit. on pp. 3, 5, 18, 27).

[VHGS16] Hado Van Hasselt, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning". In: *Thirtieth AAAI conference on artificial intelligence*. 2016 (cit. on p. 27).

[VS+09] Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. "A theoretical and empirical analysis of Expected Sarsa". In: *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. IEEE. 2009, pp. 177–184 (cit. on pp. 15, 23, 116).

[VCN19] José R Vázquez-Canteli and Zoltán Nagy. "Reinforcement learning for demand response: A review of algorithms and modeling techniques". In: *Applied energy* 235 (2019), pp. 1072–1089 (cit. on p. 2).

[WVH17] Yuan Wang, Kirubakaran Velswamy, and Biao Huang. "A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems". In: *Processes* 5.3 (2017), p. 46 (cit. on p. 29).

[WVH18] Yuan Wang, Kirubakaran Velswamy, and Biao Huang. "A Novel Approach to Feedback Control with Deep Reinforcement Learning". In: *IFAC-PapersOnLine* 51.18 (2018), pp. 31–36 (cit. on p. 29).

[Wan+19] Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. "Trust Region-Guided Proximal Policy Optimization". In: *arXiv preprint arXiv:1901.10314* (2019) (cit. on p. 109).

[Wan+16] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. "Sample efficient actor-critic with experience replay". In: *arXiv preprint arXiv:1611.01224* (2016) (cit. on p. 109).

[WD92] Christopher JCH Watkins and Peter Dayan. "Q-learning". In: *Machine learning* 8.3-4 (1992), pp. 279–292 (cit. on p. 14).

[Wawo9] Paweł Wawrzyński. "Real-time reinforcement learning by sequential actor–critics and experience replay". In: *Neural Networks* 22.10 (2009), pp. 1484–1497 (cit. on p. 24).

[WWZ17] Tianshu Wei, Yanzhi Wang, and Qi Zhu. "Deep reinforcement learning for building HVAC control". In: *Proceedings of the 54th Annual Design Automation Conference 2017*. ACM. 2017, p. 22 (cit. on pp. 18, 27).

[WL95] Steven D Whitehead and Long-Ji Lin. "Reinforcement learning of non-Markov decision processes". In: *Artificial Intelligence* 73.1-2 (1995), pp. 271–306 (cit. on pp. 6, 61, 62).

[Wik21] Wikipedia contributors. *Exponential distribution — Wikipedia, The Free Encyclopedia*. [Online; accessed 14-April-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Exponential_distribution&oldid=1014032247 (cit. on p. 68).

[Wu+17] Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation". In: *Advances in neural information processing systems*. 2017, pp. 5279–5288 (cit. on p. 109).

[Yan+15] Lei Yang, Zoltan Nagy, Philippe Goffin, and Arno Schlueter. "Reinforcement learning for optimal control of low exergy buildings". In: *Applied Energy* 156 (2015), pp. 577–586 (cit. on pp. 3, 28).

[Zha+18] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, Siliang Lu, and Khee Poh Lam. "A deep reinforcement learning approach to using whole building energy model for hvac optimal control". In: *2018 Building Performance Analysis Conference and SimBuild*. 2018 (cit. on pp. 3, 29).

[ZL18] Zhiang Zhang and Khee Poh Lam. "Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system". In: *Proceedings of the 5th Conference on Systems for Built Environments*. ACM. 2018, pp. 148–157 (cit. on p. 30).

[Zha+17] Ziqi Zhang, Wanyong Li, Chengquan Zhang, and Jiangping Chen. "Climate control loads prediction of electric vehicles". In: *Applied Thermal Engineering* 110 (2017), pp. 1183–1188 (cit. on p. 2).