Reinforcement Learning-based Thermal Comfort Control for Vehicle Cabins

Brusey, J; Hintea, D; Gaura, E. and Beloe, N.

Reinforcement Learning-based Thermal Comfort Control for Vehicle Cabins

J. Brusey[a,1,*], D. Hintea[a,1], E. Gaura[a,1], N. Beloe[a,b,1]

[a] *Faculty of Engineering, Environment and Computing, Coventry University, Gulson Rd, Coventry, West Midlands CV1 2JH, United Kingdom*
[b] *Jaguar Land Rover Limited, Abbey Road, Whitley, Coventry, CV3 4LF, United Kingdom*

Abstract

Vehicle climate control systems aim to keep passengers thermally comfortable. However, current systems control temperature rather than thermal comfort and tend to be energy hungry, which is of particular concern when considering electric vehicles. This paper poses energy-efficient vehicle comfort control as a Markov Decision Process, which is then solved numerically using Sarsa(λ) and an empirically validated, single-zone, 1D thermal model of the cabin. The resulting controller was tested in simulation using 200 randomly selected scenarios and found to exceed the performance of bang-bang, proportional, simple fuzzy logic, and commercial controllers with 23%, 43%, 40%, 56% increase, respectively. Compared to the next best performing controller, energy consumption is reduced by 13% while the proportion of time spent thermally comfortable is increased by 23%. These results indicate that this is a viable approach that promises to translate into substantial comfort and energy improvements in the car.

Keywords: Thermal Comfort, Reinforcement Learning, Equivalent Temperature, Comfort Model, Energy Consumption

1. Introduction

Vehicle HVAC (Heating, ventilation, and air conditioning) systems aim to ensure that passengers are thermally comfortable. Traditionally, controllers for these systems are hand-coded and tuned to try to achieve this goal. However, there are a number of drivers for change:

1. Current systems only control cabin temperature whereas thermal comfort is also dependent on a multitude of other factors (such as radiant heat and airflow).

*Corresponding author
Email address: j.brusey@coventry.ac.uk (J. Brusey)

2. Past systems have relied on waste heat from the engine whereas electric vehicles produce much less heat and so a different design is required.
3. Current systems are energy hungry whereas electric and hybrid vehicles demand a much more energy efficient approach. **?** report that air conditioning systems reduce the fuel economy of fuel-efficient cars by about 50%.

These drivers for change make redesign of many parts of the vehicle comfort delivery system timely. As this comfort system design changes, the controller must also adapt to best make use of the available actuation options.

The main idea in this paper is to show that Reinforcement Learning (RL) reliably produces a controller that uses less energy while delivering better comfort than existing hand-coded approaches (Section 4). We also show that the trade-off between energy and comfort can be adjusted to suit situations that demand either more comfort or better energy efficiency (Section 4.3.1). The approach requires a model of the cabin environment and we provide a simple, empirically validated, lumped model of the cabin's thermal environment (Section 3.1). The problem is then defined in terms of the state space (Section 3.3), action space (Section 3.5) and reward function (Section 3.6). Issues and implementation ramifications of this approach are discussed in Section 5.

2. Related work

2.1. HVAC control methods in vehicles

Much of the work on HVAC control (**????**) remains rooted in thermal comfort models developed for home and office indoor environments. The best known comfort model is the Predictive Mean Vote (PMV) (**???**), which estimates comfort based on: environmental parameters (such as air temperature, mean radiant temperature, relative air velocity and relative humidity); and personal parameters (such as metabolic rate and clothing thermal resistance). For example, **?** derive a PMV-based fuzzy logic control mechanism, with rules like "if temperature is medium and activity is low, then PMV is near neutral".

Although many aspects of vehicle thermal environment control are derivative of that in buildings, the vehicle's thermal environment is transient and non-uniform (**?**). Thus it is recognised that what is appropriate in the thermal comfort model for a building may not be appropriate in a car (**??**).

While there are a number of thermal comfort models available, there is disagreement between these models about what contribution different parameters should have, or even what parameters to include (**?**). Moreover, there are clearly parameters that might be considered but are not generally included. For example, occupants may enter the vehicle with latent or stored heat, they may have a physiological condition (such as a fever), or they may have cultural or personal preferences (**?**). While there are many factors that can affect comfort, not all affect it equally. While air temperature remains central to comfort, as the number of sensors and intelligence of the controller within the car increases, it becomes possible to include more factors.

A number of additional *models*, *estimators*, and *predictors* populate the literature, typically accompanied by a strategy for HVAC control (e.g., **?** predicts comfort based on facial skin temperature and cabin air temperature; **?** proposed a zonal HVAC system driven on an occupant thermal comfort level based on sensor measurements, thermal comfort charts, the ASHRAE thermal scale, ISO 7730, the PMV index, the PPD index and their combination; **?** applies artificial intelligence methods to extract thermal comfort knowledge from the interaction between the passengers and the HVAC controls). Not surprisingly, most, if not all, of the proposed controllers are based on machine learning techniques. A prime reason is that car cabin comfort control is non-linear with respect to the observable state, for example: (a) the transfer of heat as a function of vent speed and vent temperature is non-linear; (b) any plant output limitation affects response in a non-linear fashion (**?**); (c) comfort models, such as Predicted Mean Vote (PMV) and equivalent temperature (ET), are a non-linear function of their inputs.

Fuzzy logic is a common HVAC control approach given the imprecise nature of comfort (**?????????**) and many fuzzy-logic controllers have been found to perform better than the traditional air temperature controllers. **?** demonstrated that even better results were obtained when the parameters of the comfort oriented fuzzy controller were optimised by a genetic algorithm. Such controllers are, however, computationally expensive and can be difficult to design.

2.2. Reinforcement learning-based control applications

? and **?** have examined the problem of optimising HVAC thermal comfort-based control through a RL-based technique in the context of buildings rather than cars. **?** developed and simulated a reinforcement learning-based controller using Matlab/Simulink. The reward is a function of the building occupants' thermal comfort, the energy consumption and the indoor air quality. The proposed controller was compared to a Fuzzy-PD controller and a traditional on/off controller (an evaluation approach also applied here). The results showed that, after a couple of simulated years of training, the reinforcement learning-based controller performed better in comparison to the other two controllers.

? highlight an issue with regard to reinforcement learning-based controllers—that of sufficient exploration. Taking random actions, even during short times, is unacceptable for a system deployed in a real environment and the authors recommend to exhaustively train the controller prior deployment and allow minimal or no exploration at all afterwards. This work provided inspiration and a good foundation for our work in vehicle cabins.

? have examined the problem of optimising comfort and energy using Q-learning with a state space that includes the time of day. They break the control problem down into: bang-bang control (when to turn the heater on or off) and set-point control (what temperature to request at what time). In their work, the tenant immediately responds to discomfort, which might seem unrealistic, but it provides similar input to the thermal comfort model used here. By including time, they neatly provide for pre-heating or cooling and this approach might also be used for the car cabin.

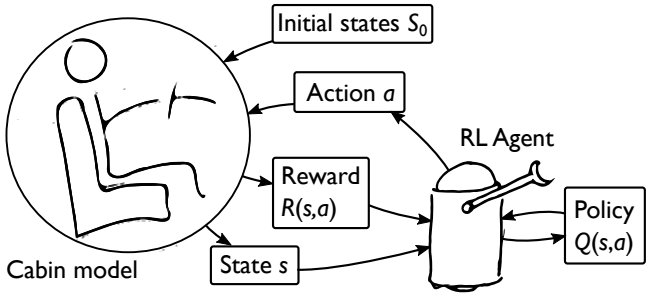

Figure 1: The process of finding an optimal policy with RL involves modelling the cabin environment T, identifying the state S and action A spaces, defining the distribution S_0 of initial states, and defining an appropriate reward function $\mathcal{R}(s, a)$.

Less recently, **?** have examined the problem of a simulated heating coil and combined a PI (proportional-integral) controller with an RL supervisor. They showed that the combined approach outperforms the base PI controller. This combination is similar to the approach here where the RL action is a vent temperature set-point that is passed to a base controller to achieve.

? provide a detailed review of computational intelligence approaches in the built environment and show that, for the built environment, a variety of adaptive control approaches have been tried and advanced approaches (such as RL) have led to improved comfort and energy savings.

This past work demonstrates that RL, while untested, may be appropriate in this domain.

3. Materials and methods

We formulate the cabin comfort control problem as a Markov Decision Process (MDP) with continuous states defined by the tuple $\langle S, S_0, A, T, \mathcal{R}, \gamma \rangle$, where S is the (infinite) set of states of the cabin environment from which a set of initial states $S_0 \subseteq S$ is drawn, A is a finite set of actions (e.g., setting the blend door position), $T\colon S \times A \to S$ is a deterministic environmental model that maps states and actions to subsequent states, $\mathcal{R}\colon S \times A \to \Re$ is a function expressing the reward for taking an action in a particular state, and γ is a discount factor such that, for $\gamma < 1$, a reward achieved in the future is worth less than a reward achieved immediately.

The solution of the MDP is a policy $\pi\colon S \to A$ or mapping from states to actions and, in particular, an optimal solution is one that maximises the long-term, discounted expected reward. In algorithms such as Q-learning and Sarsa(λ), rather than find the policy directly, we estimate the expected value or utility $Q^{\pi}(s, a)$ of each state, action combination when following policy π. This expected value is the immediate reward $\mathcal{R}(s, a)$ plus the discounted subsequent reward, which can thus can be defined recursively,

$$Q^{\pi}(s, a) = \mathcal{R}(s, a) + \gamma Q^{\pi}(T(s, a), \pi(T(s, a))). \tag{1}$$

We can then progress greedily towards the optimum policy by updating the policy π to be that which maximises Q^π, or,

$$\pi(s) \leftarrow \arg\max_{a \in A} Q^\pi(s, a). \tag{2}$$

Since the policy for any state is easy to calculate from Q^π, it does not need to be explicitly stored.

For finite state MDPs, algorithms such as Monte Carlo Exploring Starts (MCES) and Monte Carlo ε-soft **?**, §5.3,5.4 use repeated application of (1) and (2) to converge on the optimal policy. To avoid getting stuck in a local minima, they include some random exploration and this is sufficient to ensure that they always converge on the global optimum policy. For continuous state MDPs, $Q^\pi(s, a)$ must be approximated using a function $f\left(\vec{\theta}, s, a\right)$ parameterised by a vector $\vec{\theta}$ and algorithms, such as Sarsa(λ), that use this approach may not converge on the optimum policy but may oscillate (**?**).

A learning *episode* begins by selecting an initial state at random from the distribution $s_0 \sim S_0$ and then continues with the agent selecting an action and the cabin model returning a new state and reward until a maximum number of steps is reached. For some problems, it is possible to have a terminal state that ends the episode. However, this is not possible here, since the reaching comfort is not sufficient; the agent needs to efficiently maintain comfort as well. The initial state distribution should be comprehensive to avoid leaving parts of the state space unexplored. The agent is ε-greedy, which means that with probability ε it selects a random action and otherwise it selects according to the largest estimated utility for that state, as per (2).

Although it might be possible to implement a learning system directly in the car, prior works in this domain (such as, **?**) suggest learning in simulation first. In principle, the learnt policy can then be implemented in the car cabin either as a fixed policy or as a start point for continued learning. In this work, we only examine the system in simulation and implementing in the car is left to future work.

Given this basis for learning, we now define each aspect of the MDP, beginning with the model.

3.1. Cabin Thermal Environmental Model

Car cabin thermal modelling has been investigated by a number of authors (**??**), typically to examine the trade-off between comfort and energy use. Simple 1D models are appropriate for optimisation (e.g., **?** examines the effect of different coolant fluids) since they allow the consequences of changes to be quickly evaluated. Some simplifying assumptions are necessary and different works tend to make different assumptions about the cabin environment. For example, **?** include the effect of engine heat on supply and return ducts, whereas **?** include radiant heat effects for a multi-zone minivan. Our focus here is to provide a clearly described, simple model that might be expanded upon

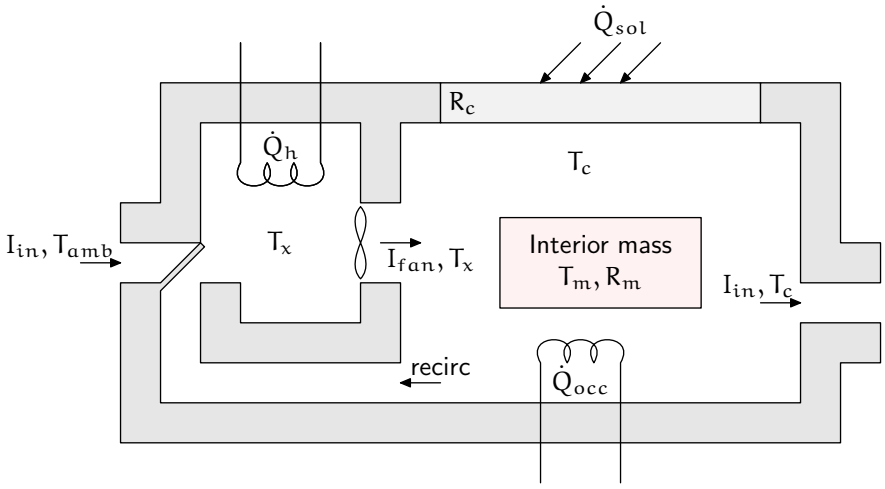

Figure 2: Schematic of the simplified cabin model used for learning a controller.

but which is validated against data from a real car in a climatic wind tunnel (Section 3.2).

Our simple cabin model is shown in Figure 2 and this corresponds to a system of three heat balance equations (heat in = heat out + heat stored),

$$\dot{Q}_h + I_{in}\left(T_{amb} - T_c\right) = I_{fan}\left(T_x - T_c\right) \quad (3)$$

$$I_{fan}\left(T_x - T_c\right) + \dot{Q}_{sol} + \dot{Q}_{occ} + \frac{T_m - T_c}{R_m} = \frac{T_c - T_{amb}}{R_c} + C_c k \frac{dT_c}{dt} \quad (4)$$

$$\frac{T_c - T_m}{R_m} = C_m \frac{dT_m}{dt} \quad (5)$$

where $\dot{Q}$ is the change in heat energy, I is the current (or mass flow of heated air), T is the temperature, R is the thermal resistivity, and C is the thermal capacitance. Subscripts are: h heat pump, *in* input air, *amb* ambient air, c cabin air, *fan* blower fan, x mixed air, *sol* solar load, *occ* occupant, and m interior mass. A cabin capacitance factor k is used to account for the difference between the experimentally observed capacitance of the cabin air and the theoretical thermal capacitance of air. This difference is probably due factors such as the air mixing time (which is otherwise assumed to be instantaneous in the model). The recirculation factor $\alpha = I_{in}/I_{fan}$ corresponds to the percentage of fresh air. Note that the mixing chamber heat storage is assumed to be negligible. For the purposes of this work, we take the work done (or energy consumed) by the HVAC to be simply $W_h = \left|\dot{Q}_h\right|$ and ignore the energy cost for the fan.

Model constants, shown in Table 1, were selected to best match the target car, a Jaguar model XJ sedan. This car was used for model validation in Section 3.2.

It is assumed that there is no air leakage. Nor is the vehicle velocity taken into account. In comparison with **?** this model does not deal with the internals

Table 1: Model constants

Cabin volume V_c	$2.5\,\mathrm{m}^3$
Cabin capacitance factor k	8
Solar load $\dot{Q}_{sol}$	$150\,\mathrm{W}$
Occupant load $\dot{Q}_{occ}$	$120\,\mathrm{W}$
Cabin resistivity R_c	$1/\left(5.741626794 \times 4.0\right)\ \mathrm{K.W}^{-1}$
Interior mass resistivity R_m	$1/\left(75 \times 1.08\right)\ \mathrm{K.W}^{-1}$
Interior mass capacitance C_m	$450 \times 0.02 \times 7850\,\mathrm{J.K}^{-1}$

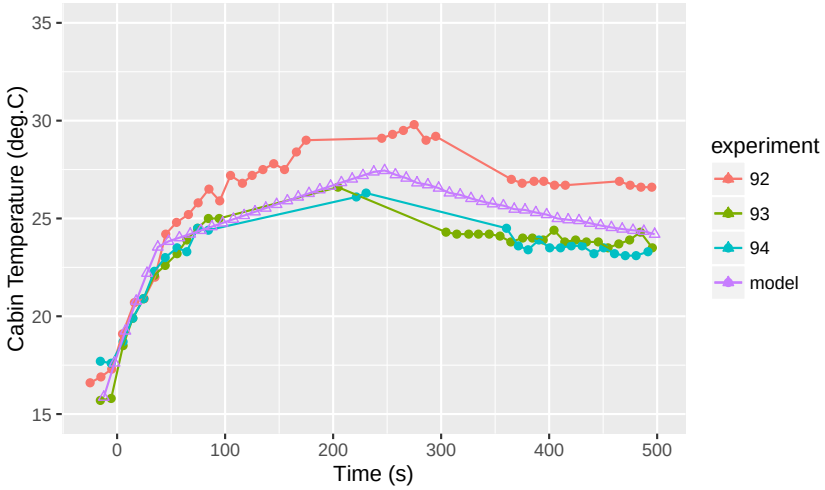

Figure 3: Car cabin warm-up experiment showing real and simulated (denoted 'model') results time-aligned at 18 °C. The time-series shows overshoot in the controller, probably due to lag in the in-car sensor. The proportional controller with an averaged sensor (see Section 4.1) is used with the simulated model and this produces a similar overshoot.

of the evaporator but rather considers the combined heat sum from a heat pump. Also, heat effects from the internal combustion engine (through the firewall or supply ducts) are not considered here (and may be inappropriate for an electric vehicle). **?** have a more sophisticated model that includes two zones for a minivan. In comparison to the work here, they include radiative heat transfer between cabin walls and the interior mass as well as between the cabin walls and the sky.

3.2. Model validation

The simulation data was compared to empirical data collected by the authors within various warm-up and cool-down scenarios (described in **?**). Figures 3 and 4, based on experiments with a Jaguar XJ in MIRA LTD's climatic wind tunnel, show warming (from cold) and cooling (from hot) the car cabin based on head-rest height temperature sensors over a number of experiments and also

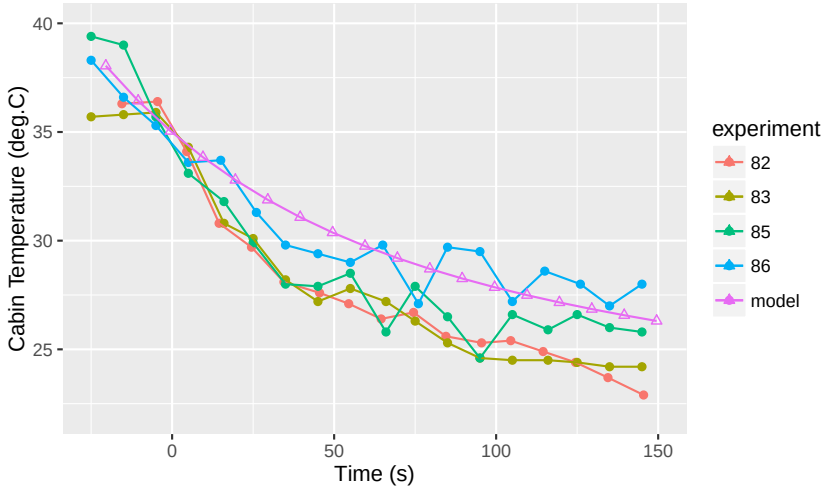

Figure 4: Car cabin cool-down experiment showing real and simulated (denoted 'model') results time-aligned at 35 °C.

showing the simulated or 'model' results. Simulated results are based on the bang-bang controller described further in Section 4.1. These graphs demonstrate that the simulation broadly matches the characteristics of the physical system and thus that a controller that performs well with the simulation is likely to work well in practise in terms of control of temperature.

Although the modelling of energy use is based on reasonable assumptions, at this stage we have no experimental data with which to validate the model. Energy use is difficult to estimate precisely in practise since, for example, latent heat from the engine is used to heat the cabin. Thus energy use in practise may differ from the simulation.

3.3. State representation

The state of the cabin environment is a vector comprising: the cabin air temperature T_c, the interior mass temperature T_m and the outside air temperature T_{amb}. Equivalent temperature (ET) is not an explicit component of the state but is computed using a formula (referring to sedentary conditions only, that is energy metabolism $< 70\,\mathrm{Wm}^{-2}$) introduced by **??**,

$$T_e = \begin{cases} 0.5(T_c + T_r), & \text{for air flow } \dot{v}_c \leq 0.1\,\mathrm{ms}^{-1} \\ 0.55T_c + 0.45T_r + \frac{0.24-0.75\sqrt{\dot{v}_c}}{1+I_{cl}}\left(36.5 - T_c\right), & \text{for } \dot{v}_c > 0.1\,\mathrm{ms}^{-1} \end{cases} \tag{6}$$

The air flow corresponding to the cabin occupant $\dot{v}_c$ is not directly available and it is estimated here by dividing the vent air flow v_i by 10. The value was selected based on cabin air flow measurements in the literature **?**. The mean radiant temperature T_r is assumed to be equal to the interior mass temperature

T_m. For this work, the clothing insulation I_{cl} is set to a constant value of 0.7 clo corresponding to long trousers and short sleeve, light-coloured blouse or shirt. Note that ET is provided as input to the 'et' variants of the hand-coded controllers but is not explicitly provided to the RL controller.

At the beginning of each controller training episode, the initial state vector is selected at random from a uniform distribution over the full range of values for each of:

- Interior temperature T_m: $[0, 50]$ ℃.
- Outside temperature T_o: $[0, 40]$ ℃.
- Cabin air temperature T_a: $[0, 50]$ ℃.

The representation of the state is minimal, sufficient (along with the action) for the reward function, and Markovian (in terms of the simulation). Some elements that are held constant in this model (such as the solar load) might also be included in the state vector if they were allowed to vary.

Selection of the initial state and range of states is influenced by the episode length and what is likely to occur. Episode length places a limit on the extreme values. For example, it might take more than 500 steps to achieve a comfortable state from a very high or low start temperature. From such a start point, any policy looks equally bad.

We also eliminate start states where the interior mass temperature is different from the cabin temperature by more than 30 ℃, as this situation is considered to be unlikely.

Function approximation is used by the Sarsa(λ) algorithm to avoid having to discretise the state and also to support a large state space. Function approximation involves defining a parameter vector $\theta = (\theta_1, \theta_2, \ldots)^T$ thus allowing Q to be approximated by a smooth function

$$\hat{Q}(s, a) = f_\theta(s, a).$$

The function approximator used in this case is tile coding and the configuration of the function approximator is further below.

3.4. Tile coding

The tile coding parameters used for this problem are presented in Table 2. In contrast with other work, rather than use a separate function approximator for each action, a single function approximator is used with tiles that span the combined state and action spaces. The tile coding used to represent the action-values included 30 tiles, 10 tiles integrating variables $(T_a, T_m, T_o, T_i, v_i, A_r)$ and 20 tiles integrating variables (T_a, T_i, v_i, A_r). Note that ET is not included (since it is not part of the state vector).

Table 2: Tile coding parameters used to learn the control policy.

Variable	Minimum	Maximum	Intervals
T_a	0	50	26
T_m	10	40	7
T_o	0	40	7
T_i	0	60	3
v_i	1	100	3
A_r	0	1	3

3.5. Action representation

The set of actions consists of a vector $(v_i, T_i, A_r)^T$ where each component of the vector takes on one of a small set of discrete values. Specifically, there are four possible vent air flows $v_i \in \{1, 34, 67, 100\}$ ℓs^{-1}. Five possible vent air temperatures can be selected, which are evenly defined over the range $T_i \in [7, 60]$ °C. Lastly, three recirculation flap positions are available $A_r \in \left\{0, \frac{1}{2}, 1\right\}$. This yields a total of 60 ($4 \times 5 \times 3$) possible actions.

3.6. Reward function

The learning goal is to maximise the time spent in comfort (defined here as when the occupant ET is 24 ± 1 °C) while minimising energy use. This can be expressed as the reward function,

$$\mathcal{R}(s,a) = \mathcal{R}_C(s) - E(s,a)/w \tag{7}$$

$$\mathcal{R}_C(s) = \begin{cases} 0 & \text{if } T_e \in 24 \pm 1\,^\circ\text{C} \\ -1 & \text{otherwise} \end{cases} \tag{8}$$

$$E(s,a) = \left|\dot{Q}_E\right| + 2v_i \tag{9}$$

where $\mathcal{R}_C$ is the penalty for being uncomfortable, E is the energy cost, $w = 30\,000$ is the energy weight divisor (which, in lay terms, means that a 1% improvement in comfort is equivalent to 300 W). This weight can be adjusted to give a different trade-off between energy and comfort (see Section 4.3.1). The above reward function could be further extended to include goals such as minimising fan noise or keeping the screen clear. Illegal states (where component values are out of bounds) are not explicitly penalised but act as an absorbing state with worst case penalty, which is sufficient to ensure that the learning agent avoids them.

3.7. Meta parameters

Meta parameters control the learning process and may affect how quickly learning proceeds. The first is the number of steps per episode, which is set at 500. This allows the agent to reach a comfortable state from any start state but also that the episode length is not so long that new start states are

rarely experienced. The reward discount factor $\gamma = 0.99$ ensures that a policy is appropriately rewarded for actions that do not produce immediate reward. Given that the reward function does not give reward for moving towards comfort (but only for reaching it), setting γ close to 1 allows the agent to learn to achieve comfort even from extreme initial temperatures. The learning rate $\alpha = 0.01$, exploration factor $\varepsilon = 0.16$ (for first 190 000 episodes and zero thereafter), and eligibility trace decay $\lambda = 0.98$ were decided by looking at the performance over the first 2 000 episodes, as discussed in Section 4.3.1.

3.8. Evaluation method

The performance of the RL controller is tested using a set of 200 randomly pre-selected start states $S_T \subset S_0$ at regular intervals during learning. This set is referred to as the *test scenario set*. This approach provides a standard test that can be used for all controllers to provide fair comparison while ensuring that the test is reasonably comprehensive over possible start states.

4. Evaluation

The RL-based controller is evaluated by comparing its performance with: a bang-bang controller, a proportional controller, a commercial controller, and a fuzzy-logic controller. For each controller, three possible temperature sensors T_s are simulated: the true cabin air temperature (air), the average of cabin and interior mass temperatures (avg), and the equivalent temperature (et). All controllers actuate as per the action representation (see Section 3.5).

4.1. Bang-bang, proportional and commercial controllers

The first three hand-coded controllers are somewhat similar. The bang-bang controller blows the maximum fan rate to cool or warm the cabin until it is within 1 ℃ of the target, at which point it blows the minimum fan rate and tries to match the target temperature. The proportional controller is similar but it reduces the fan speed exponentially $v_i = 100 - 99 \exp\left(\frac{|T_s - 24|}{10}\right)$ as the sensor temperature nears the target temperature. The commercial controller is based on a commercial specification. This tends to use lower fan rates than the proportional controller, probably to avoid noise and vibration, but is otherwise quite similar.

4.2. Simple fuzzy logic controller

For the evaluation here, a simple fuzzy logic controller was implemented in Java using the fuzzylite library version 1.0 (**?**). Apart from the sensor temperature T_s , this controller also receives interior mass temperature T_m. Fuzzy set membership functions for input temperatures T_s, T_m are NEUTRAL (24 ± 1 ℃), COLD (below NEUTRAL) and HOT (above) with some ramped overlap between each range. For vent temperature, the sets are LOW (below 10 ℃), MEDIUM (around 20 ℃), HIGH (above 30 ℃) and for vent flow rate, LOW (below 30 ℓs^{-1}),

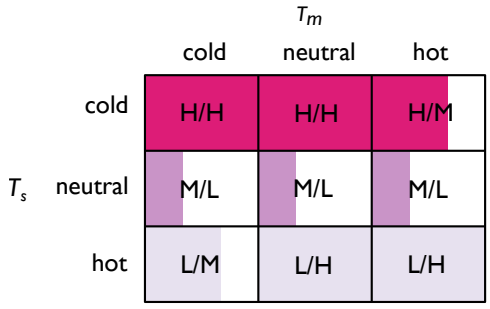

Figure 5: Simple fuzzy controller rules expressed as a table with outputs T_i/v_i. The two inputs (sensor T_s and interior mass temperature T_m) are used to derive control of the vent temperature T_i (HIGH, MEDIUM or LOW) and fan speed v_i (HIGH, MEDIUM, or LOW). E.g., if T_s is HOT and T_m is COLD, then T_i is set LOW and v_i is set MEDIUM.

MEDIUM (around $50\,\ell\mathrm{s}^{-1}$), and HIGH (above $70\,\ell\mathrm{s}^{-1}$) with similar ramped overlaps.

The fuzzy logic rules are summarised in Figure 5. These rules are slightly modified from those used by **?** and **?**.

4.3. Results

The relative performance of the RL controller compared with that of the hand-coded controllers is given in Table 3. The RL controller gives the largest (least negative) average per-step reward, uses less power and provides more comfort. This performance evolves during learning as shown in Figure 6. The RL controller achieves an average reward of -1.2 after 200 000 learning episodes (approximately 6.3 simulated years). Learning for the Sarsa(λ) algorithm (implemented in Java), corresponding to 200 000 episodes, completed in 85 minutes on a 2.9 GHz Intel® Core™ i7 processor.

These results translate into an average factor of 37% energy reduction over the test scenarios set when compared to the simple fuzzy logic-based controller, while thermal comfort was achieved and maintained successfully.

Figure 7 shows how each controller controls the occupant ET in a cool down scenario (45 °C cabin air, 45 °C block temperature and 20 °C outside temperature). Some oscillation in ET is caused by turning on and off the fan, due to ET's definition, which depends on air flow rate. The RL controller cools slightly more quickly and avoids the fluctuation in ET present in both other approaches and thus performs better overall.

4.3.1. Effect of parameter choices

Learning parameters (such as $\alpha, \varepsilon, \lambda$) affect the RL learning rate. For example, Figure 8 shows how the mean reward over episodes 1 000–2 000 changes with the learning rate α and that a rate of 0.01 produces the fastest learning.

Table 3: Reward, comfort and energy performance of the controllers over the test scenario set for commercial, bang-bang, proportional, fuzzy logic and RL agents. Sensors for the manual agents are cabin temperature (air), an average of cabin and interior mass (avg), or equivalent temperature (et).

Agent	Average reward	% Time Spent in Comfort	Average HVAC power
commercial-avg	−2.9	5%	1.4 kW
commercial-et	−2.9	5.1%	1.4 kW
bang-bang-et	−2.8	28%	2 kW
commercial-air	−2.8	6.6%	1.3 kW
fuzzy-avg	−2.7	2.6%	0.94 kW
fuzzy-air	−2.5	2.2%	0.76 kW
proportional-et	−2.3	18%	0.91 kW
bang-bang-air	−2.3	13%	0.72 kW
proportional-air	−2.2	12%	0.59 kW
proportional-avg	−2.2	17%	0.7 kW
fuzzy-et	−2.1	42%	1.2 kW
bang-bang-avg	−1.6	55%	0.88 kW
rl	−1.2	67%	0.77 kW

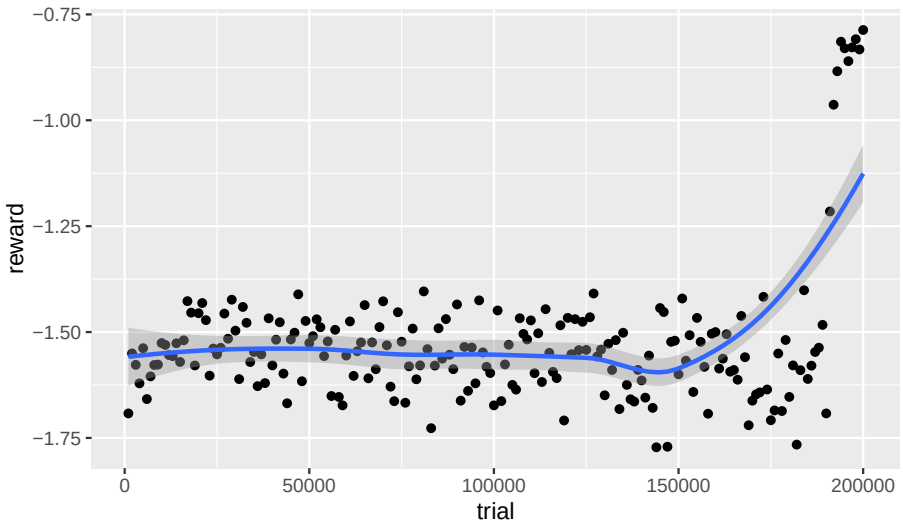

Figure 6: Policy performance during learning for Sarsa(λ). The first 190 000 episodes are with exploration $\varepsilon = 0.16$ while the rest are with no exploration $\varepsilon = 0$. A LOESS fit of the reward (with shaded 0.95 confidence interval) is also shown.

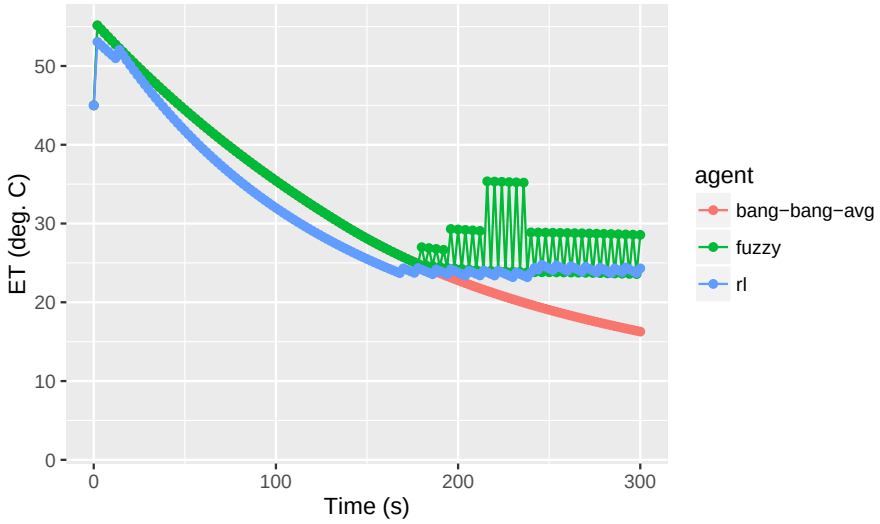

Figure 7: Comparison of how each agent responds to an initially warm cabin.

Similar experiments reveal best values for λ (0.98) and ε (0.16). Although these parameter choices are suitable during early stages of learning, different values may be better later on. In particular, reward performance improves substantially if exploration is turned off $\varepsilon = 0$ in the later stages of learning.

The weighting of energy versus comfort in the reward function can make a significant difference to the performance of the resulting policy. The tradeoff being made is reflected in Figure 9, which shows the performance for policies learnt with different energy divisor values in terms of energy use and percentage comfort. The black line drawn in the graph corresponds to the trade-off curve (or Pareto optimal front) and shows a progressive change in balance between comfort and energy as the energy divisor w is increased. Mostly, comfort and energy use increases as the energy divisor w is increased. However, there is some backtracking (e.g., at $w = 10^{4.3}$) that suggests that the policy learnt for some divisors is sub-optimal.

5. Discussion

There are several limitations of the RL controller as currently described. First, not all factors relevant to thermal comfort are simulated or included in the ET comfort metric, such as humidity, clothing level, or the metabolic work rate of the subject. Of these, possibly the most significant is humidity. Incorporating humidity into the model could be valuable since it also helps identify window fogging and thus allows a penalty for fogging to be included in the reward function. If a certain thermal comfort model leads to sub-optimal comfort when implemented as a controller, then this implies that there might

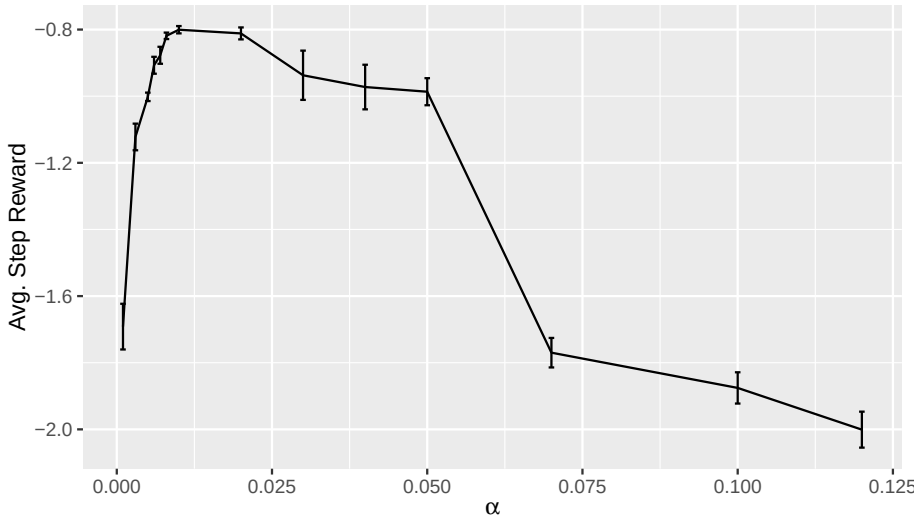

Figure 8: Mean reward for the test scenario set obtained over episodes 1000 to 2000 shows that a small, but not too small, learning rate $\alpha = 0.01$ provides peak performance. Error bars show the two-tail 95% confidence interval.

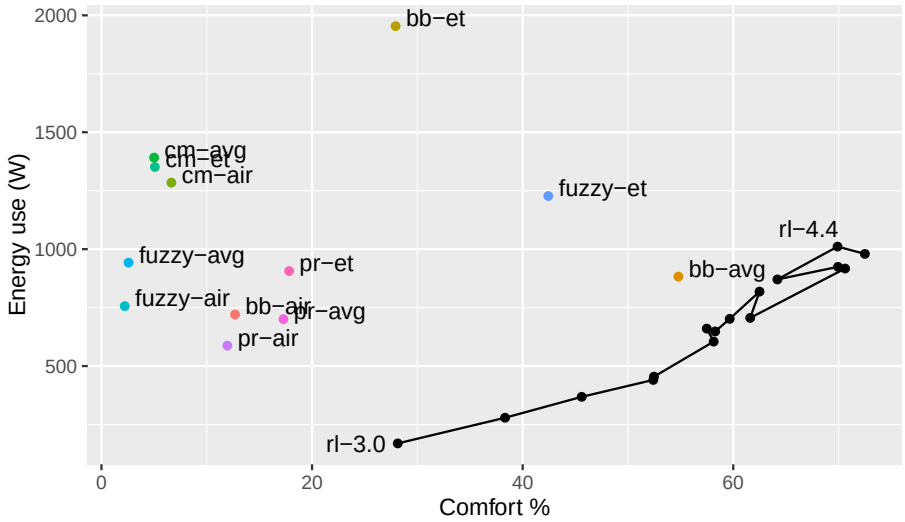

Figure 9: Effect of energy weight on comfort and energy showing the different trade-offs made to either increase comfort or reduce energy use. The black line connects results for the RL agent with an exponentially increasing energy divisor ($\log_{10} w = [3, 4.5]$ in 0.1 steps) and corresponds to the Pareto optimal front. Other agents are: bang-bang (bb), commercial (c), fuzzy, and proportional (pr). The sensor type for each agent is either cabin air (air), average of cabin and interior mass (avg), or equivalent temperature (et).

be a problem with the comfort model and thus help identify which parameter or feature is missing. Given the diversity of opinions about comfort models and relative importance of parameters in the literature, this iterative approach seems best.

Second, the hand-coded controllers shown in this paper may not perform as well as current state-of-the-art HVAC controllers. Although we tried a commercial controller, this performs poorly in simulation. Although this may suggest that the simulation is imperfect or that the reward function does not take into account important factors, it also seems likely that there is room for improvement. To understand how much of an improvement can be obtained, side-by-side in-car comparison is needed.

Third, some users may prefer less fan noise, even at the expense of being thermally uncomfortable. Furthermore, adjusting the fan speed or recirculation setting constantly may be distracting. On the other hand, some users prefer to hear the fan as it reassures them that the HVAC is actively attempting to restore comfort. An advantage of our approach is that a range of user types can be catered for by using different reward functions with added penalties for such things as fan noise. Note that the fact that the commercial controller performs poorly in terms of thermal comfort could be due to a deliberate design decision to constrain fan noise.

5.1. Pathway to implementation in the car

Occupant ET, which is used here as a proxy for comfort, cannot be directly measured in a real car cabin and the need for a proxy inspired the development of a Virtual Thermal Comfort Sensor (VTCS) (**?**). VTCS makes use of a distributed set of sensors to estimate ET based on a machine-learning approach. Note that all learning occurs off-line and thus little computation is required to implement the VTCS approach in the car. In principle, VTCS can be used to estimate comfort for different zones such as upper and lower body as well as different passenger positions.

A key consideration in the development of a controller is the accuracy of the sensor. No matter how good the controller, inaccurate measurements will lead to incorrect control. The VTCS approach has an additional advantage that it becomes possible to integrate a set of inexpensive sensors to accurately estimate ET rather than rely on a single sensor.

The RL agent developed in this paper is designed to sit on top of existing low-level controllers (such as those that control the speed of the compressor motor). This approach has the advantage that it makes the RL controller generic and retains any existing low-level safety mechanisms.

Implementing in the car provides an opportunity to receive feedback from the end-user. This feedback might come in the form of manual temperature adjustments. Such feedback can be incorporated as a penalty in the reward function and thus enable some learning of preferences. It is unclear whether learning of preferences in this way would occur quickly enough.

6. Conclusions and future work

Our results show that the RL-based controller delivers better comfort (67% time in comfort versus 55% for the bang-bang controller with averaged sensor) more efficiently (0.77 kW for RL versus 0.88 kW for the bang-bang controller). Note that the exact level of energy use may vary from this in practise since the energy use aspect of the simulation has not been fully validated. The performance of the RL controller is striking for two reasons: First, the reward function does not 'coach' the agent towards the solution; reward is only provided when comfort is reached. Second, the RL controller is not explicitly informed of the current ET but still manages to control it in a stable way.

There are a number of opportunities for future work. As discussed in Section 5, some of the limitations of the approach are due to the simulator and the controller might be improved by enhancing its realism. However, work to date on integrating with a Dymola-based cabin simulation (**?**) has shown that ensuring that the simulation is sufficiently fast remains a key challenge. There are several options to improve the simulation to make it more realistic. For example, humidity is a key factor in thermal comfort and enables identifying screen fogging. Furthermore, a zoned approach to the simulation would allow differential control of comfort for different parts of the body and for different seat positions. Testing in the car is another avenue for future work that would allow better comparison against existing controllers.

Actuation has become more complex with the introduction of heated and cooled surfaces. Although it makes sense for radiant and blown-air systems to work in concert, no current system attempts this. Similarly, natural ventilation can be used to reduce cabin temperatures in hot climates with minimal energy consumption. This work opens the door to development of a holistic controller that integrates such disparate actuators.

From the cabin HVAC designer's perspective, the RL approach raises the abstraction level from coding boolean or fuzzy rule sets towards making decisions about how to best model occupant comfort and its relative importance versus noise level, screen clarity, and energy efficiency. As this work shows, the resulting controller can be expected to substantially improve over manually coded designs.

Acknowledgements

The Low Carbon Vehicle Technology Project (LCVTP) was a collaborative research project between leading automotive companies and research partners, revolutionising the way vehicles are powered and manufactured. The project partners included Jaguar Land Rover, Tata Motors European Technical Centre, Ricardo, MIRA LTD., Zytek, WMG and Coventry University. The project included 15 automotive technology development work-streams that will deliver technological and socio-economic outputs that will benefit the West Midlands Region. The £19 million project was funded by Advantage West Midlands (AWM) and the European Regional Development Fund (ERDF).

References