Control Engineering Practice 164 (2025) 106458

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Safe and time-efficient exploration in Reinforcement Learning-based control of a vehicle thermal systems

Prasoon Garg [a,*], Emilia Silvas [b,c], Frank Willems [b,c]

[a] DAF Trucks, N.V., Eindhoven, The Netherlands
[b] Eindhoven University of Technology, Department of Mechanical Engineering, Control Systems Technology, Eindhoven, The Netherlands
[c] TNO Mobility and Built Environment, Helmond, The Netherlands

ARTICLE INFO

Keywords:
Reinforcement Learning
Safe exploration
Control barrier function
Gaussian process regression

ABSTRACT

Reinforcement Learning has achieved huge success with various applications in controlled environments. However, limited application is seen in real-world applications due to challenges in guaranteeing safe system operation, required experiment time, and required a-priori system knowledge and models in existing methods. In this work, we propose a novel exploration method, which addresses simultaneously the challenges associated with safe and time-efficient exploration while dealing with system uncertainty. This method integrates a reciprocal Control Barrier Function and an on-line learned Gaussian Process Regression model. For safe system operation, we leverage the information from the reciprocal Control Barrier Function to limit the step size of the agent's actions, when approaching the safety boundary. To make this exploration process time-efficient, we use the information gain metrics that are calculated using the estimation of the action-values by an on-line learned Gaussian Process Regression model to determine the direction of the agent's actions. We demonstrate the potential of our exploration method in simulation and on a vehicle test-bench for efficiency-optimal calibration of a thermal management system for battery electric vehicles. To quantify the benefits in terms of safety, optimality, and time efficiency, we benchmark our exploration method with random and uncertainty-driven exploration methods in a simulation environment. For the studied test case, the proposed exploration method satisfies the safety constraint and it converges to within 1.25% of the true optimal action while requiring 28% and 18% lower experiment time compared to the random and uncertainty-driven exploration methods, respectively. For the proposed method, its performance is also demonstrated on a vehicle test bench. Experimental results show that the maximal thermal system efficiency is realized within 2% of the true optimum, while effectively dealing with the safety constraints.

1. Introduction

1.1. Challenges in Reinforcement Learning applications

Reinforcement Learning (RL) (Sutton & Barto, 2018) is a learning paradigm where an agent learns to take optimal actions in interaction with its environment. The two key elements of all RL algorithms are exploitation and exploration. To determine the optimal actions, a RL agent should explore actions other than the current optimal action in the action space to improve its estimation of the action-values. To date, RL algorithms have achieved a tremendous success in applications within a simulated ecosystem, for example, video games (Atari Mnih et al., 2013, Game of Go Silver et al., 2016) and web recommender systems (Afsar et al., 2022). On the other hand, RL is also interesting for learning-based control of the physical systems. Especially, model-free RL has the potential to minimize the control development time by saving time required in generating models a-priori in existing model-based control approaches. Model-free RL approach can learn models in an autonomous manner by interacting with its environment without a prior system model. This approach also has the potential to automate the control development process and significantly minimize the expert involvement in the process. Moreover, it can be robust to changes in the real-world operating conditions and disturbances as it learns from the real data. There is a growing body of literature that has explored RL-based control for engineering systems in the simulation environment, for example, process control (Nian et al., 2020), robotics (Kormushev et al., 2013), vehicle energy management systems (Qi et al., 2019), automotive powertrain control (Norouzi et al., 2023) and autonomous vehicles (Aradi, 2022). However, a limited

* Corresponding author.
E-mail address: prasoon.garg08@gmail.com (P. Garg).

https://doi.org/10.1016/j.conengprac.2025.106458
Received 22 September 2024; Received in revised form 23 May 2025; Accepted 11 June 2025
Available online 2 July 2025

Table 1
A relative comparison of state-of-the-art exploration methods and the proposed method in this work. The benchmark method for the comparison is the random exploration method marked in bold. (Y) is yes, and (N) is no. (0) is the benchmark, while (+) represents an improvement over the benchmark, and (−) represents a decline in comparison to the benchmark. UCB is upper confidence bound, GPR is Gaussian Process Regression and rCBF is reciprocal control barrier function.

Method classification	Method	Evaluation criteria		
		Explicitly deal with safety	Reduction in experiment time	Reduction in a-priori system knowledge required
Random	**$\epsilon-$greedy** (Sutton & Barto, 2018)	**N**	**0**	**0**
Uncertainty-driven	UCB (Guo et al., 2020; Wu et al., 2016)	N	+	0
	Thompson Sampling (Guo et al., 2020; Urteaga & Wiggins, 2017)	N	+	0
	GPR (Kuss & Rasmussen, 2003)	N	+	0
Safety-driven	Model-based on-policy exploration: (Alshiekh et al., 2018; Yu et al., 2019) (Berkenkamp et al., 2017; Zhao et al., 2022)	Y	−	−
	Model-based off-policy exploration: (Gros et al., 2020; Hunt et al., 2021) (Wagenmaker & Pacchiano, 2023; Zhu & Kveton, 2022)	Y	−	−
	Model-free off-policy using rCBF (Marvi & Kiumarsi, 2021)	Y	+	+
This work	Model-free off-policy using rCBF and GPR	Y	+	+

number of RL applications is seen in the real-world operation of these systems due to multiple challenges in the exploration process, which are Garcıa and Fernández (2015):

1. Guarantee safe operation, i.e., satisfy safety constraints;
2. Required experiment time, i.e., time required to collect data for generating models and time required in exploration to learn the optimal policy;
3. Required a-prior system knowledge, that consists of information on the system's physical limits and input constraints. This information is typically derived from historical data and models of the existing, related systems, or both.

1.2. Exploration in Reinforcement Learning

Multiple studies have investigated different exploration methods for RL in varying applications. An overview of these methods can be found in Ladosz et al. (2022). The existing exploration methods can be categorized into three types: (i) Random exploration, (ii) Uncertainty-driven exploration; and (iii) Safety-driven exploration. In Table 1, we present a brief review of the main characteristics of state-of-the-art exploration methods.

1.2.1. Random exploration

Typically, random exploration using a $\epsilon-$greedy policy is applied for systems without safety constraints (Sutton & Barto, 2018). In the case of systems with safety constraints, random exploration is applied in the simulation environment where there is no risk of hardware damage. The advantage of using the random exploration is that it can converge to global optimal solutions in the long-term, i.e., if every state–action pair is visited a large number of times. Its downsides include potential unsafe system operation without prior system knowledge and large experiment time. For systems with safety constraints, random exploration can be used only if the safe action space is known a-prior. However, this is often difficult to determine for new and complex real-world systems where the safe action space varies with changes in operating conditions, for example, varying ambient conditions.

1.2.2. Uncertainty-driven exploration

Multiple studies have investigated uncertainty-driven exploration strategies to reduce the experiment time required for exploration while learning the optimal policy. The most commonly studied strategies are upper confidence bound (UCB) (Sutton & Barto, 2018), Thompson Sampling (Thompson, 1933) and Gaussian Process Regression (GPR)-driven exploration (Kuss & Rasmussen, 2003). In all these methods, the action-values are learned sequentially from the data samples, and the uncertainty in the action-values drives the exploration process. The agent explores the actions with the highest uncertainty in the corresponding action-value estimates and reduces the exploration of actions with low uncertainty. This results in quicker convergence to optimal actions compared to random exploration. As far as safety is concerned, these methods also require an understanding of the system's physical limits and input constraints to determine the safe action space, which is non-trivial to determine for new and complex real-world systems (Guo et al., 2020; Urteaga & Wiggins, 2017; Wu et al., 2016).

1.2.3. Safety-driven exploration

The last decade has seen an increasing interest in exploration strategies that deal with safety constraints (Brunke et al., 2022; Garcıa & Fernández, 2015). Most studies found on safe exploration use a prior system model, for example, tabular model (Alshiekh et al., 2018), state-space model (Yu et al., 2019), data-driven model using Gaussian Process Regression (Berkenkamp et al., 2017) and first-principle models (Gros et al., 2020; Hunt et al., 2021; Zhao et al., 2022). This approach needs an accurate system model to determine the safe control inputs under a wide range of operating conditions, which is challenging to generate for complex systems. Typical examples of modeling techniques used are first principle and data-driven modeling. For complex systems, generating system models using first principles is challenging and time-consuming. Therefore, data-driven models are often used. Due to the black-box nature of data-driven models, (very) limited system knowledge is needed to make these models. Nevertheless, data-driven modeling requires significant data, resulting in large experiment times.

Limited work has been found on model-free safe exploration. In Marvi and Kiumarsi (2021), a reciprocal control barrier function (rCBF) is used for safe exploration using model-free RL. rCBF is an adapted version of Control Barrier Function (CBF), which has been extensively studied for formal proofs of safety for dynamical systems in the field of control theory (Anand et al., 2021; Prajna & Jadbabaie, 2004; Wieland & Allgöwer, 2007). For safe exploration, the reward is augmented with a rCBF value to indicate proximity to the safe boundary. However, safety is treated as a soft constraint in this formulation, which can result in constraint violation because it is difficult to interpret the individual contribution of reward and rCBF term in the feedback signal. Therefore, there is a non-zero probability of violating the safety constraint to determine the unsafe action.

Most of the existing work on safe exploration follows an on-policy approach where one policy is used for both exploration and exploitation (Alshiekh et al., 2018; Berkenkamp et al., 2017; Yu et al., 2019; Zhao et al., 2022). The disadvantage of using on-policy approach is that it learns action-values for a near-optimal policy, which is always exploring. Off-policy approach overcomes this limitation by employing two policies, one that is exploratory and the other that becomes the optimal policy. The explorative policy is called the behavior policy, while the learned policy is called the target policy. In Wagenmaker and Pacchiano (2023), Zhu and Kveton (2022), off-policy approach is used for safe exploration, where the behavior policy is learned from a historical dataset that provides a-prior information on the expected action-value estimations and unsafe actions. The downside of this approach is that the system's safe operation depends on the data's coverage, i.e., values of states and actions, in the historical dataset. Therefore, it is challenging to maintain safe operation in conditions that are not contained in the historical dataset.

A most common feature of the existing works on safe exploration is that they require a significant a-priori system knowledge either in the form of system models or a historical dataset, which is then used to determine the safe action space. Uncertainty-driven methods reduce the experiment times by exploring actions that maximize information on the reward function, however, they can result in unsafe operation. To the extent of our knowledge, limited work exists that ensures safe operation of the systems with safety constraints while minimizing the experiment times, as illustrated in Table 1.

1.3. Research objective and main contribution

The objective of this work is to develop an exploration method for RL-based control that maintains safe operation of the system during learning and minimizes the required experiment time to determine the optimal policy. To minimize overall development time and required expert knowledge, we assume that a prior system model is not required.

The main contribution of this work is a novel Safe and Information-seeking exploration (Safe-ISE) method that integrates techniques from the fields of Control Theory and Machine Learning. The proposed Safe-ISE method learns action-values from the data generated by agent-environment interaction during the exploration process using an online learned Gaussian Process Regression (GPR) model (Williams & Rasmussen, 2006). This model is used to estimate actions that give most information on the reward value in the studied operating region, i.e. actions that reduce model uncertainty and increase expected improvement in the action-values. This so-called Information Gain approach aims to reduce experiment time. To guarantee safe operation, a reciprocal Control Barrier Function (rCBF) (Ames et al., 2019; Marvi & Kiumarsi, 2021) is introduced that limits the step-size of the agent's action when approaching the safety boundary. We demonstrate the potential of the proposed method for the calibration of a battery electric vehicle thermal system to optimize its steady-state operation in simulation and on a vehicle test-bench.

1.4. Outline

This paper is organized as follows. The problem formulation is stated in Section 2. Section 3 briefly reviews the different methods that are applied in this work. The fully integrated concept for the novel safe and time-efficient RL exploration method is presented in Section 4. This method is applied for the calibration of an automotive thermal management system. Section 5 introduces the studied system and associated control problem. For the proposed method, simulation results are presented in Section 6 and compared with the results of the random and uncertainty-driven exploration methods. Next, in Section 7 the proposed method is implemented on a vehicle test bench and experimental results are discussed. Finally, Section 8 summarizes the main conclusions and gives directions for future research.

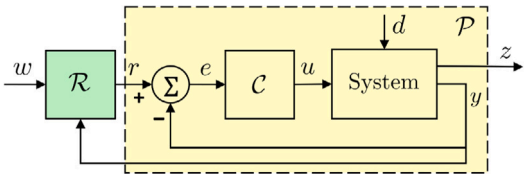


Fig. 1. A general control schematic. Here, u is the feedback control action, z is the system performance, y are the measured outputs, w are the external inputs to the control system, d are the unknown disturbances to the system, $e = r - y$ is the control error, C is the feedback controller and $\mathcal{R}$ is the setpoint generator.

2. General problem formulation

To develop safe and time-efficient RL for systems with limited prior knowledge, we consider a nonlinear dynamical control system,

$$\dot{x} = f(x, u, t) \tag{1}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ is the system internal state vector, $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, f is a n-dimensional system function vector. We consider a classical control architecture with a feedback controller C for reference tracking and rejection of external disturbances d, and a setpoint generator $\mathcal{R}$ for optimizing system performance by smart choice of r, as shown in Fig. 1. This architecture is most commonly used in the control of real-world systems.

The process of determining optimal settings for controllers $\mathcal{R}$ that maximizes system performance z and C is defined as control calibration, which is a challenging and most time-consuming task in the control development process (Garg et al., 2023). Typically, the first step in control calibration process is to determine the optimal control settings for a stationary operating point. The control objective is to learn the optimal steady-state reference setpoints r, which maximize system performance, defined by z, while the system internal state x with safety constraints stays within the set of safe states $\mathcal{X}_\varsigma \subset \mathcal{X}$ at all times.

Few assumptions are made in this work to clearly define the scope of the problem. We assume that the system dynamics are not known a-priori. However, we assume that the system internal states x with safety constraints are measurable. Further, we assume that the safety constraint on x poses either the lower or the upper constraint on r. We also assume that the feedback controller C, which is an asymptotic stabilizing feedback controller, is available such that $e = r - y \to 0$ at a steady-state operating point. We assume that there exists a safe initial reference setpoint $r(t_0)$ and it is known a-priori. $r(t_0)$ is generally known or derivable from historical data of similar existing systems. For generality, we also assume that the system performance function defined by $z(r)$ can have one or multiple optimums.

3. Methodology

In this section, we present how we combine concepts from the field of Control Theory and Machine Learning for the formulation of safe and time-efficient RL, as illustrated in Fig. 2. First, the definitions of safe set and safe control are presented. Then, we discuss the approach to formulate safety constraints using the reciprocal control barrier function. Next, we formulate the control calibration problem as a Contextual Bandits problem. We then present how we use the Gaussian Process Regression model to estimate action-values. Lastly, we present the proposed Safe and Information-seeking (Safe-ISE) exploration method.

3.1. Safe control and safe set

We use the concept of forward invariance to define safe control i.e., the system internal states always evolve and stay in the safe set $\mathcal{X}_\varsigma$ (Geurts, 1998). This implies that $x \in \mathcal{X}_\varsigma$ during the exploration and exploitation by the RL agent. To define the safe set $\mathcal{X}_\varsigma$ as a function of system internal state x, we use the approach defined in Ames et al. (2019).

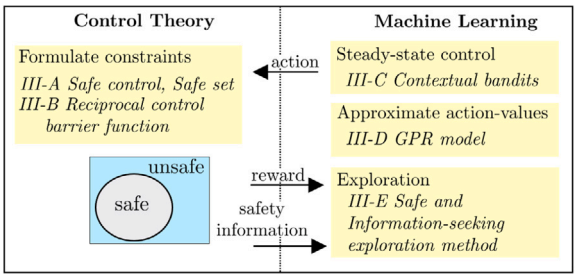


Fig. 2. Overview of concepts used for developing Safe-ISE method. GPR stands for Gaussian Process Regression. Numbers in the prefix represents the corresponding section numbers.

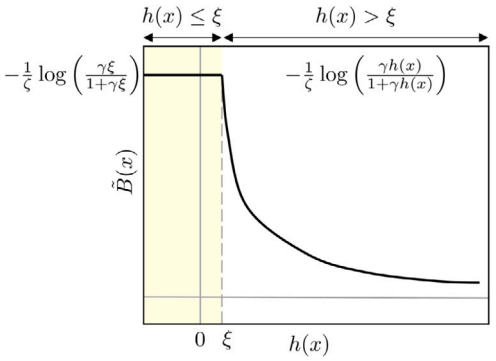


Fig. 3. Proposed $\tilde{B}(x)$ function.

Definition 3.1 (*Safe Set*)**.** $\mathcal{X}_\varsigma$ is defined as a superlevel set of a continuously differentiable function $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ such that:

$$\begin{aligned} \mathcal{X}_\varsigma &= \{x \in \mathcal{X} \subset \mathbb{R}^n \mid h(x) \geq 0\} \\ \partial\mathcal{X}_\varsigma &= \{x \in \mathcal{X} \subset \mathbb{R}^n \mid h(x) = 0\} \\ \text{Int}(\mathcal{X}_\varsigma) &= \{x \in \mathcal{X} \subset \mathbb{R}^n \mid h(x) > 0\} \end{aligned} \tag{2}$$

where $\partial\mathcal{X}_\varsigma$ is the boundary of the safe set and $\text{Int}(\mathcal{X}_\varsigma)$ is the interior of the safe set.

Here, $h(x)$ is used to formulate the safety constraint on the system internal state x as described in Marvi and Kiumarsi (2021). The constraints are formulated as,

$$h(x) \geq 0 \tag{3}$$

where,

$$h(x) = \begin{cases} \overline{x} - x, & \text{if } x \text{ is upper bounded} \\ x - \underline{x}, & \text{if } x \text{ is lower bounded} \end{cases} \tag{4}$$

where, $\overline{x}$ is an upper bound on state x, $\underline{x}$ is a lower bound on state x and are assumed to be known a-priori. For real-world physical systems, $\overline{x}, \underline{x}$ are generally known from the operational guidelines of the components present in the system. $h(x) \geq 0$ holds true for all $x \in \mathcal{X}_\varsigma$ as given by Eq. (2).

3.2. Reciprocal control barrier function (rCBF)

rCBF is a modified form of a control barrier function introduced in Ames et al. (2014) and its value blows up as $x \to \partial\mathcal{X}_\varsigma$. We use rCBF to provide information on the unsafe system operation to the agent during the exploration. rCBF, represented by $\tilde{B}(x)$, is defined as a function of the constrained system state x and satisfies the following conditions,

1. $\tilde{B}(x) > 0 \ \forall x \in \mathcal{X}_\varsigma$
2. $\tilde{B}(x) \to \infty \ \forall x \in \partial\mathcal{X}_\varsigma$

The most common choices for $\tilde{B}(x)$ found in literature (Ames et al., 2019; Marvi & Kiumarsi, 2021) are logarithmic functions that satisfy the above two conditions. However, a common limitation of these functions is that $\tilde{B}$ is not well defined for $h(x) < 0$ and provide incorrect information about the unsafe states. Therefore, we propose a modified function whose magnitude stays large for unsafe states, i.e., $\tilde{B}(x) \to \infty \ \forall \ h(x) < 0$, see Fig. 3. It is expressed as,

$$\tilde{B}(x) = \begin{cases} -\frac{1}{\zeta}\log\left(\frac{\gamma h(x)}{1+\gamma h(x)}\right), & h(x) > \xi \\ -\frac{1}{\zeta}\log\left(\frac{\gamma\xi}{1+\gamma\xi}\right), & h(x) \leq \xi \end{cases} \tag{5}$$

where ξ is a small positive constant, which is introduced for the continuity of $\tilde{B}(x)$ around $h(x) = 0$. γ, ζ are tunables and affect the scaling and rate of change of $\tilde{B}(x)$ for $h(x) > \xi$, respectively. We exploit the exponential increase in the value of $\tilde{B}(x)$ as $h(x) \to 0$ to limit the exploration process approaching $\partial\mathcal{X}_\varsigma$. It is achieved by chosing the step size of the agent's action Δa approaching $\partial\mathcal{X}_\varsigma$ inversely proportional to $\tilde{B}(x)$. Δa is expressed as,

$$\Delta a \propto \frac{1}{\tilde{B}(x)} \tag{6}$$

3.3. Contextual Bandits problem

The output r of $\mathcal{R}$ in Fig. 1 has no impact on the future operating conditions w and future reference setpoints in steady-state operating conditions of the system. Therefore, we propose to use Contextual Bandits RL algorithm to learn the optimal steady-state reference setpoints r (see control problem in Section 2). In a Contextual Bandits problem, an action a in a stationary scenario, also called the context s, is independent of previous actions and does not affect future contexts and actions. This is consistent with steady-state control of the system around a stationary operating point. The objective of a Contextual Bandit RL problem is to determine the policy $\pi(s)$ by mapping the context s to optimal actions a^* that maximizes the agent's reward R (Sutton & Barto, 2018). To determine a^*, the average reward or action value $q(s, a)$ of taking an action a in context s is calculated. $q(s, a)$ is defined as,

$$q(s, a) = \frac{\sum_{k=1}^{k=t} R_k \cdot \mathbb{1}_{a_k=a}}{\sum_{k=1}^{k=t} \mathbb{1}_{a_k=a}} \tag{7}$$

where $\mathbb{1}_{a_k=a}$ is 1 when action a is taken and 0 otherwise. a^* in context s is defined as the action with the maximum action value and it is expressed as,

$$a^*(s) = \arg\max_a q(s, a) \tag{8}$$

3.4. Gaussian Process Regression for approximating action-values

For learning action-values $q(s, a)$, we use a sampled-based learning technique. In this technique, the agent incrementally learns action-values using the information from a sequence of data samples one at a time. We use the Gaussian Process Regression (GPR) model because it can learn from the data samples incrementally. The GPR model approximates $q(s, a) \ \forall \ a \in \mathcal{A}$ using the sampled data and we use its mean and uncertainty predictions to guide the exploration. The true action-values $q(s, a)$ can be written as,

$$q(s, a) = \hat{q}(s, a) + \eta \tag{9}$$

where η is the noise, s, a are the inputs to the model and $\hat{q}(s, a)$ is the action-value estimate. $\hat{q}(s, a)$ is distributed as a Gaussian Process, which is characterized by its mean function $\mu(s, a)$ and a covariance function $k([s, a], [s', a'])$ (Williams & Rasmussen, 2006) and it is defined as,

$$\hat{q}(s, a) \sim \mathcal{GP}\mu(s, a), k([s, a], [s', a']) \tag{10}$$

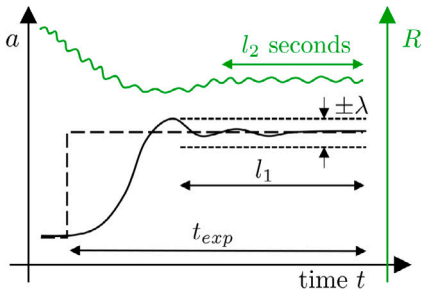


Fig. 4. Illustration of the approach used to determine the average reward. Dotted black line shows the agent's action a and solid black line shows the corresponding system output y.

where $[s'\ a']$ is the prediction context-action pair for which the GPR model is evaluated. To determine $q(s,a)$, the reward signal is averaged over a time duration of l_2 seconds when the system output y is stationary as shown in Fig. 4. The stationary condition is detected when the system output y corresponding to agent's action a is within the desired tolerance of $\pm\lambda$ from a for the duration of l_1 seconds. The total time required before $q(s,a)$ value is sampled is denoted by t_{exp}, which is equivalent to the time required for a single experiment. Due to the averaging of the reward signal, the observations $q(s,a)$ are assumed to be noise-free. The outputs of the GPR model are the predictions of mean $\mu_{\hat{q}}(s,a)$ and variance $\sigma_{\hat{q}}(s,a)$ in the estimated action-values $\hat{q}(s,a)$.

4. Safe and information-seeking exploration

The proposed RL-based control system with the Safe-ISE method is shown in Fig. 5. The corresponding pseudo-code is shown in Algorithm 1. The system signals are sampled with a sampling time of one second and time is represented by t. The iterations of the agent-environment interactions are represented by $i = 1, 2, 3, \ldots$, which is incremented only when an agent takes a new action. An off-policy learning approach is used for exploration, which consists of two different policies, i.e., behavior policy $\pi_b(s)$ and target policy $\pi_\tau(s)$. $\pi_b(s)$ can be used to generate data at every time instant i by exploration. Therefore, in order to learn the model for $q(s,a)$, we use an off-policy approach. $\pi_b(s)$ converges to an optimal policy over iterations and this converged policy is assigned as the target policy $a^*(s) \equiv \pi_\tau(s)$, which is a fixed policy. This $\pi_\tau(s)$ can then be used for deployment on the system.

4.1. Initialization

As a first step in the exploration, the environment context s_t is set to a constant value, which is equivalent to defining a steady-state operating point for the system. Thereafter, the hyperparameter settings for the different functions in Fig. 5 including rCBF, signal averaging, GPR model selection and detect stationary conditions are defined. The choice of hyperparameter settings are listed in Table 2, which are described later in Section 5.3. These settings are kept constant during the exploration process. To store data on actions a_i taken by the agent, corresponding action-values $q(s_i,a_i)$ and rCBF values $\tilde{B}(s_i,a_i)$, matrices are defined and initialized equivalent to zero before exploration begins. The agent then begins exploration by taking a safe initial action a_0 for iteration $i = 1$ followed by small steps Δa_0 around a_0 i.e., $a_0 + \Delta a_0$, $a_0 - \Delta a_0$ for $i = 2, 3$ respectively. It is assumed that a safe initial action a_0 is known from experience with similar systems. By taking small steps around a_0, the agent determines the action $a \in \{a_0 - \Delta a_0, a_0 + \Delta a_0\}$ results in x approaching the safe boundary $\partial\mathcal{X}_\varsigma$. For every action a_i during the exploration, the agent receives a measurement of the average context s_i, average reward i.e., $q(s_i,a_i)$ and average value of $\tilde{B}(s_i,x_t)$ i.e., $\bar{\tilde{B}}(s_i,x_i)$ using the signal averaging block. s_i, $\bar{\tilde{B}}(s_i,x_i)$ are calculated in a similar approach to $q(s_i,a_i)$, as illustrated in Fig. 4. First, the GPR model is learned after every agent-environment interaction as in Eq. (10). Second, to determine actions that approach $\partial\mathcal{X}_\varsigma$, we create an additional mapping from agent's action a_i to the $\bar{\tilde{B}}(s_i,x_i)$ value, i.e., $\hat{\bar{\tilde{B}}}(s_i,a_i) : a_i \rightarrow \bar{\tilde{B}}(s_i,x_i)$. This mapping provides information to the agent about actions lying away or close to the safe boundary $\partial\mathcal{X}_\varsigma$ under the assumption that x increases or decreases monotonically as a function of a in $\mathcal{X}_\varsigma$. As a result of this initial exploration, the agent determines the direction in actions approaching $\partial\mathcal{X}_\varsigma$ and estimates of the action-values as predicted by the GPR model shown in Fig. 6(a).

4.2. Define action space

After the initial exploration around a_0, the action space $\mathcal{A}_i$ for exploration in the next iteration, i.e., $i = 4$ is determined. Fig. 6(b) illustrates the process of defining the action space. For an informed exploration of the action space, we introduce the concept of information gain (IG) metrics, which is used to define $\mathcal{A}_i$. IG is defined as a combination of two different metrics of information: (1) Uncertainty in the action-value estimate i.e., standard deviation $\sigma_{\hat{q}}(s_i, a \in \mathcal{A}_i)$ and (2) Proximity to the current maxima (PM) of other local or global maxima in the action-values. The metric PM is adapted from the expected improvement acquisition function from the theory of Bayesian Optimization (Pelikan et al., 1999). Both these values are derived from the GPR model predictions of $\hat{q}(s_i,a_i)$. For the action space feasible for exploration at iteration i i.e., $\mathcal{A}_i$, $\mathrm{PM}(s_i, a \in \mathcal{A}_i)$ is defined as,

$$\mathrm{PM}(s_i, a \in \mathcal{A}_i) = \mu_{\hat{q}}(s_i, a \in \mathcal{A}_i) - \underbrace{\left(\mu_{\hat{q}}(s_i, a_i^*) - 1.96 \times \sigma_{\hat{q}}(s_i, a_i^*)\right)}_{\text{Minimum best prediction}} \tag{11}$$

where, a_i^* is the current optimal action (i.e., action with maximum action-value) at instant i. $\mu_{\hat{q}}(s_i, a_i^*) - 1.96 \times \sigma_{\hat{q}}(s_i, a_i^*)$ corresponds to the lowest value in 95% prediction interval of action value at a_i^*. A 95% prediction interval is chosen to make this metric more conservative such that the agent assumes a lower confidence in the current best action and favors exploration of other actions in the action space. PM compares the lower prediction value at the current best action with the mean predictions for all $a \in \mathcal{A}_i$. Larger values of PM correspond to other maximas and gives indication of presence of other maximas to the agent and potential actions that should be explored to converge to the global maxima.

$\mathcal{A}_i$ consists of the currently estimated safe action space $\mathcal{A}_{\varsigma,i}$ and small action space outside the safe action space $\mathcal{A}_i^+$ and it is expressed as,

$$\mathcal{A}_i = \mathcal{A}_{\varsigma,i} + \mathcal{A}_i^+ \tag{12}$$

where $\mathcal{A}_{\varsigma,i} \subset \mathcal{A}$ is the safe action space at ith iteration. $\mathcal{A}_{\varsigma,i}$ is defined as,

$$\mathcal{A}_{\varsigma,i} = [\underline{a}_i, \overline{a}_i] \tag{13}$$

where $\overline{a}_i$ is expressed as,

$$\overline{a}_i = \begin{cases} \max\{a_0, a_1, \ldots, a_{i-1}\} \text{ if } \hat{\bar{\tilde{B}}}(s_i, a_0 + \Delta a_0) > \hat{\bar{\tilde{B}}}(s_i, a_0) \\ \max(\{a \mid \mathrm{PM}(s, a \in \mathcal{A}) > \underline{\mathrm{PM}}\}) \\ \qquad\qquad \text{if } \hat{\bar{\tilde{B}}}(s_i, a_0 - \Delta a_0) > \hat{\bar{\tilde{B}}}(s_i, a_0) \end{cases} \tag{14}$$

Here, $\underline{\mathrm{PM}}$ is the lower bound on PM below which the agent does not explore the corresponding actions in the action space. $\underline{a}_i$ is defined as,

$$\underline{a}_i = \begin{cases} \min(\{a \mid \mathrm{PM}(s, a \in \mathcal{A}) > \underline{\mathrm{PM}}\}) \\ \qquad\qquad \text{if } \hat{\bar{\tilde{B}}}(s_i, a_0 + \Delta a_0) > \hat{\bar{\tilde{B}}}(s_i, a_0) \\ \min\{a_0, a_1, \ldots, a_{i-1}\} \text{ if } \hat{\bar{\tilde{B}}}(s_i, a_0 - \Delta a_0) > \hat{\bar{\tilde{B}}}(s_i, a_0) \end{cases} \tag{15}$$

IG is also calculated for $\mathcal{A}_i^+$ to determine if there exists a maxima outside $\mathcal{A}_{\varsigma,i}$ using the metrics PM and the corresponding uncertainty $\sigma_{\hat{q}}(s_i,a_i)$ in its action-value estimation. This approach prevents the

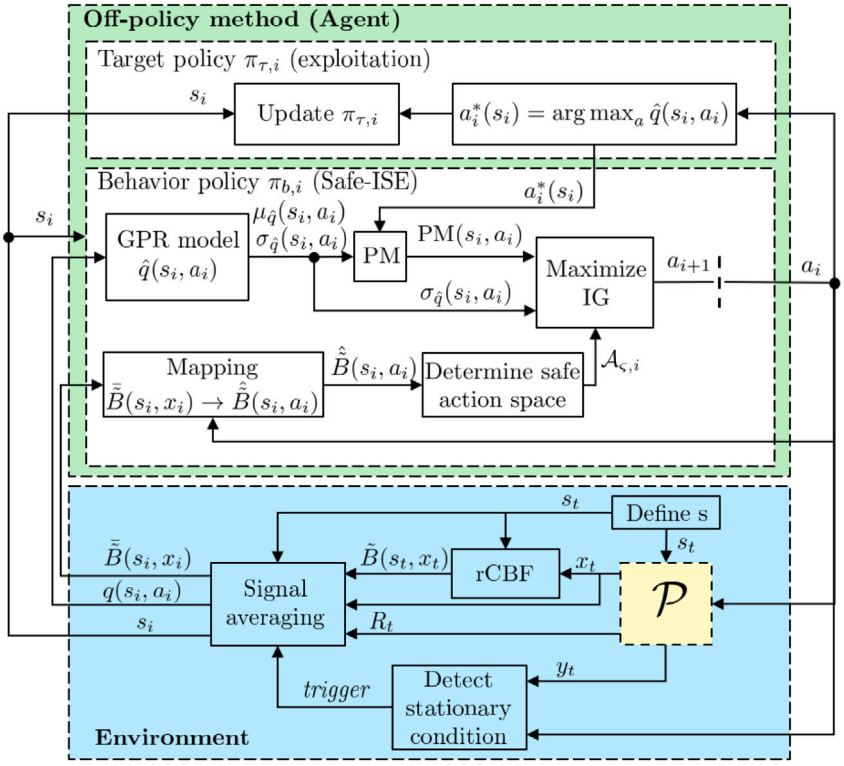
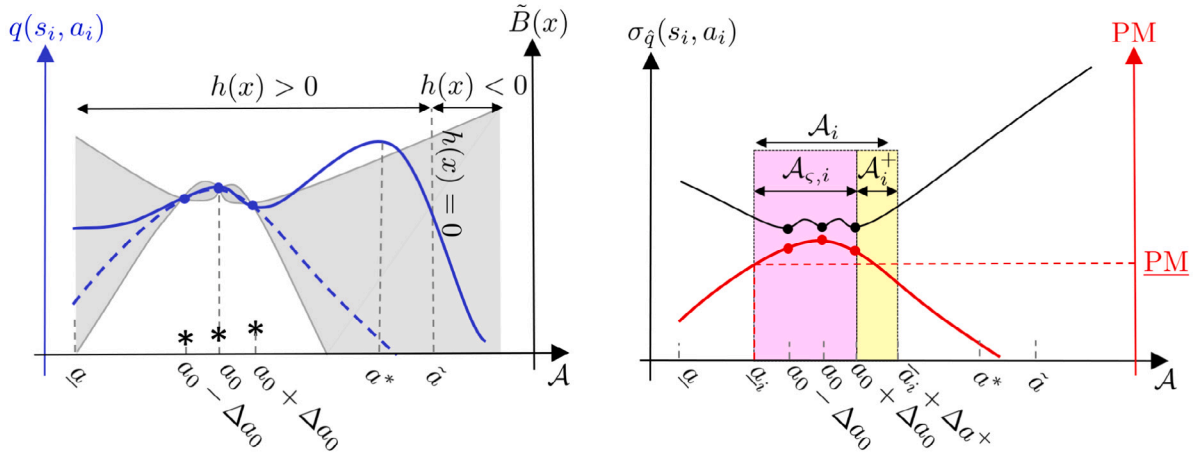


Fig. 5. Control schematic showing off-policy learning for exploration and exploitation. Blocks highlighted in green are the focus of this work. $i = 1, 2, 3, \ldots$ represents the iterations of the agent-environment interaction. $t = 1, 2, 3, \ldots$ represents the discrete time instances. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Action-value estimate using GPR. (b) Information gain metrics.

Fig. 6. Illustration of Safe-ISE method after initial exploration around the safe action a_0 at iteration i = 4. In Fig. (a), solid blue line shows the true action-value function, the dotted blue line is the mean prediction from GPR model, shaded region in gray shows the 95% prediction interval estimated by GPR model, $(*)$ represents the $\tilde{B}$ values, $(\bullet)$ in blue are action-values corresponding to agent's actions. In Fig. (b), the black line shows the standard deviation estimated by GPR and the red line shows the PM values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

agent from exploring actions approaching $\partial\mathcal{X}_\varsigma$ when no maxima exists as predicted by PM. $\mathcal{A}_i^+$ is defined as,

$$\mathcal{A}_i^+ = \begin{cases} [\overline{a}_i, \overline{a}_i + \Delta a^+] \text{ if } \hat{\tilde{B}}(s_i, a_0 + \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \\ [\underline{a}_i - \Delta a^+, \underline{a}_i] \text{ if } \hat{\tilde{B}}(s_i, a_0 - \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \end{cases} \tag{16}$$

where Δa^+ is the maximum action step size that that lies outside the safe action space and it is kept constant during exploration. It is a calibration parameter where its smaller values reduce the probability of the agent exploring actions approaching $\partial\mathcal{X}_\varsigma$.

4.3. Action selection

To choose its next action a_i, the agent compares the IG metrics for two action spaces. i.e., $\mathcal{A}_{\varsigma,i}$ and $\mathcal{A}_i^+$. The agent then chooses a_i from the actions that has the highest uncertainty i.e., $\sigma_{\hat{q}}(s_i, a)$ in action-value and for which the condition $\mathrm{PM}(s_i, a \in \mathcal{A}_i) > \underline{\mathrm{PM}}$ is satisfied. If the action satisfying these conditions lies in $\mathcal{A}_{\varsigma,i}$, the agent takes the action with maximum uncertainty as follows,

$$a_i = \arg\max_a \sigma_{\hat{q}}(s_i, a) \tag{17}$$

Algorithm 1: Safe and Information-seeking exploration (Safe-ISE)

Input: Define context s, safe initial action a_0, define hyperparameters in Table 2

Initialize $i = 0$, Δa^+ ;
Initialize $A(i)$, which is a vector to store values of all actions taken by the agent ;
Initialize $\mathcal{A}$, which is a set of discrete actions at which GPR model is evaluated;
Initialize $q(s,a)$, $\hat{\tilde{B}}(s,a)$ for $a \in \mathcal{A}$;
while *stopping criteria not met* **do**
  $i = i + 1$;
  for $i = 1,2,3$ **do**
    Explore around safe action: $a = a_0 \pm \Delta a_0$, $A(i) \leftarrow a$;
  end
  for $i \geq 4$ **do**
    $\mathrm{PM}(s_i, a \in \mathcal{A}_i) \leftarrow \mu_{\hat{q}}(s_i, \mathcal{A}_i) - \mu_{\hat{q}}(s_i, a_i^*) - 1.96 \times \sigma_{\hat{q}}(s_i, a_i^*)$;
    $\underline{a}_i \leftarrow \begin{cases} \min(\{a \mid \mathrm{PM}(s, a \in \mathcal{A}) > \underline{\mathrm{PM}}\}) \text{ if } \hat{\tilde{B}}(s_i, a_0 + \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \\ \min\{a_0, a_1, ..., a_{i-1}\} \text{ if } \hat{\tilde{B}}(s_i, a_0 - \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \end{cases}$;
    $\overline{a}_i \leftarrow \begin{cases} \max\{a_0, a_1, ..., a_{i-1}\} \text{ if } \hat{\tilde{B}}(s_i, a_0 + \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \\ \max(\{a \mid \mathrm{PM}(s, a \in \mathcal{A}) > \underline{\mathrm{PM}}\}) \text{ if } \hat{\tilde{B}}(s_i, a_0 - \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \end{cases}$;
    $\mathcal{A}_{\varsigma,i} \leftarrow [\underline{a}_i, \overline{a}_i]$;
    $\mathcal{A}_i^+ \leftarrow \begin{cases} [\overline{a}_i, \overline{a}_i + \Delta a^+] \text{ if } \hat{\tilde{B}}(s_i, a_0 + \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \\ [\underline{a}_i - \Delta a^+, \underline{a}_i] \text{ if } \hat{\tilde{B}}(s_i, a_0 - \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \end{cases}$;
    $\mathcal{A}_i \leftarrow \mathcal{A}_{\varsigma,i} + \mathcal{A}_i^+$;
    if $\mathrm{argmax}_a\, \sigma_{\hat{q}}(s_t, a \in \mathcal{A}_i) \in \mathcal{A}_{\varsigma,i}$ **then**
      $a_i \leftarrow \mathrm{argmax}_a\, \sigma_{\hat{q}}(s_i, a)$
    else if $\mathrm{argmax}_a\, \sigma_{\hat{q}}(s_t, a \in \mathcal{A}_i) \in \mathcal{A}_i^+$ and $\mathrm{PM}(\mathrm{argmax}_a\, \sigma_{\hat{q}}(s_t, a \in \mathcal{A}_i^+)) > \underline{\mathrm{PM}}$ **then**
      $a_i \leftarrow \begin{cases} \overline{a}_i + \epsilon_{\tilde{B}} / \hat{\tilde{B}}(s_i, \overline{a}_i) \text{ if } \hat{\tilde{B}}(s_i, a_0 + \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \\ \underline{a}_i - \epsilon_{\tilde{B}} / \hat{\tilde{B}}(s_i, \underline{a}_i) \text{ if } \hat{\tilde{B}}(s_i, a_0 - \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \end{cases}$
    $A(i) \leftarrow a_i$;
  end
  Observe $q(s_i, a_i), x_i$;
  $\tilde{B}(s_i, x_i) \leftarrow \begin{cases} -\frac{1}{\zeta} \log\left(\frac{\gamma h(x)}{1+\gamma h(x)}\right), & h(x) > \xi \\ -\frac{1}{\zeta} \log\left(\frac{\gamma \xi}{1+\gamma \xi}\right), & h(x) \leq \xi \end{cases}$;
  Update $\hat{\tilde{B}}(s_i, a_i)$;
  Update GPR model using data $A(i), q(s_i, a_i)$;
  Compute GPR predictions $\mu_{\hat{q}}(s_i, a \in \mathcal{A}), \sigma_{\hat{q}}(s_i, a \in \mathcal{A})$;
  $a_i^* \leftarrow \mathrm{argmax}_a\, \hat{q}(s_i, a \in \mathcal{A}_i)$;
end

On the other hand, if such an action lies in $\mathcal{A}_i^+$, the agent takes a step size inversely proportional to $\tilde{B}$ towards $\partial\mathcal{X}_\varsigma$, expressed as, a_i is expressed as,

$$a_i = \begin{cases} \overline{a}_i + \epsilon_{\tilde{B}} / \hat{\tilde{B}}(s_i, \overline{a}_i) \text{ if } \hat{\tilde{B}}(s_i, a_0 + \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \\ \underline{a}_i - \epsilon_{\tilde{B}} / \hat{\tilde{B}}(s_i, \underline{a}_i) \text{ if } \hat{\tilde{B}}(s_i, a_0 - \Delta a_0) > \hat{\tilde{B}}(s_i, a_0) \end{cases} \tag{18}$$

where $\epsilon_{\tilde{B}}$ is a constant of proportionality and a calibration parameter. This ensures a decaying step size in approaching the constraint boundary to avoid constraint violation.

4.4. Stopping criteria

The agent stops the exploration process if either of the following termination conditions is met:

1. No action exists satisfying criteria defined in Section 4.3,
2. Change in optimal action $\Delta a^* < \epsilon$ for p number of iterations. Here, ϵ is a small scalar. Decreasing ϵ increases the experiment time and probability to converge closer to the global optima and vice-versa. On the other hand, by decreasing p, the experiment time decreases with higher probability of getting stuck in local maxima,
3. Maximum allowable value of rCBF $\overline{\tilde{B}}$ reached i.e., $\hat{\tilde{B}}(s_i, a_i) \geq \overline{\tilde{B}}$,
4. Maximum number of experiments allowed is reached i.e., $N_{exp} \geq \overline{N}_{exp}$.

5. Application to an automotive heat pump system

To determine the potential of Safe-ISE method, we apply the proposed exploration method to an automotive heat pump system in simulation. The focus of this work is on the cabin cooling mode of the heat pump system shown in Fig. 7. The pressure-enthalpy diagram, also called the Mollier diagram, shown in Fig. 8 can be used to explain the operation of the heat pump in the cabin cooling mode. In this mode, the heat pump system extracts the heat energy from the cabin and releases it to the environment to meet the desired cabin temperature. The system consists of four actuators i.e., expansion valve u_{exv}, compressor u_{cmp}, fan u_{fan} and blower u_{blwr}, two heat exchangers i.e., evaporator (EVA) and outer heat exchanger (OHX) and accumulator (ACC). The air flowing across the outer heat exchanger present at the vehicle front extracts the heat from the hot refrigerant flowing across it. The refrigerant then condenses and changes its phase from vapor to liquid. If the refrigerant temperature goes below its saturation temperature at a given pressure p_3, it is in a subcooled liquid state. Here, the subcool temperature T_{sc} at a given working fluid pressure is defined as the temperature difference corresponding to $h_3(p_3) - h_{3'}(p_3)$, where h is enthalpy, $h_{3'}(p_3)$ is the enthalpy of the saturated working fluid in the liquid state, see Fig. 8.

The refrigerant changes its phase from liquid to two-phase mixture as it passes the expansion valve due to the pressure drop across the expansion valve. The two-phase refrigerant then extracts heat energy from hot air flowing across the evaporator and changes its phase to vapor. It is beneficial to keep the refrigerant in a saturated vapor state

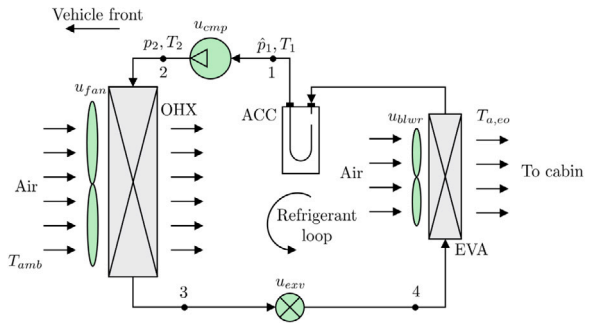


Fig. 7. Schematic of heat pump system in the cabin cooling mode. Symbols p, T represents the measured pressure and temperature signals. Subscript a, eo represents air at evaporator outlet and amb is ambient air. $\hat{p}_1$ represents a prediction from the virtual pressure sensor. Actuators are marked in green and the heat exchangers are marked in gray. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

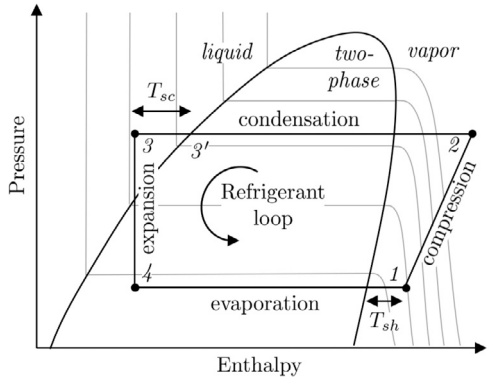


Fig. 8. Mollier diagram for the heat pump system in the cabin cooling mode. Isothermal conditions are indicated by gray lines.

at the outlet of the evaporator, as the presence of liquid droplets in the refrigerant going into the compressor can cause damage to its components. To ensure the safety of the compressor, an accumulator is introduced before the compressor, which extracts any remaining liquid droplets in the refrigerant. The refrigerant in a vapor state at low temperature and pressure is then compressed to vapor at high temperature and pressure by the compressor, closing the cycle. The refrigerant temperature at the compressor outlet is constrained, as high temperatures can damage the compressor due to high frictional forces. Also, the refrigerant pressure after the compressor is constrained by the operational guidelines of the heat pump system.

5.1. Control problem

The control objective is to determine the T_{sc} setpoints to realize the cooling demand of the passengers J_c while maximizing the heat pump efficiency J_η and maintaining safe operation of the system in steady-state operation. The heat pump efficiency J_η is directly correlated to T_{sc} (Yamanaka et al., 1997), which can controlled by the expansion valve. J_η is defined as the ratio of the rate of useful thermal energy exchanged and the rate of work consumed. It is often referred to as Coefficient of Performance (COP) and is expressed as,

$$J_\eta = \frac{\dot{Q}_{a,e}}{\dot{W}_{net}} = \frac{P_{eva}(T_{sc}, u_{cmp})}{P_{cmp}(u_{cmp}) + P_{fan}(u_{fan})} \tag{19}$$

where $\dot{Q}_{a,e}$ is the rate of thermal energy exchanged by air at the evaporator, $\dot{W}_{net}$ is the rate of useful work, P_{eva} is the cooling power delivered by the refrigerant at the evaporator, P_{cmp} is the required compressor power and P_{fan} is required fan power. The blower power P_{blwr} is omitted in Eq. (19) because the blower speed is held constant. The cooling demand J_c is defined as the absolute deviation in the setpoint value $r_{T_{a,eo}}$ and the actual value of $T_{a,eo}$ and it is expressed as,

$$J_c = |r_{T_{a,eo}} - T_{a,eo}| \tag{20}$$

To meet the cooling demand, i.e., minimize J_c, the air temperature at the evaporator exit $T_{a,eo}$ is controlled by the compressor. The common practice for designing heat pump control system is to introduce two single-input single-output (SISO) feedback control loops, consisting of setpoint generator $\mathcal{R}$ and feedback controller C. Here, T_{sc} and $T_{a,eo}$ are controlled by the expansion valve and the compressor speed, respectively. However, the interaction between the two SISO control loops makes the calibration of the controllers $\mathcal{R}, C$ challenging and time-consuming (Keir & Alleyne, 2007).

With regard to safety, the refrigerant pressure at the compressor exit i.e., p_2 should remain below a threshold value $\bar{p}_2$ as advised in the operating guidelines of the compressor. Factoring in the interaction between the two SISO control loops, the cost function for optimization is defined as a weightage sum of J_η and J_c. The resulting optimal control problem is expressed as,

$$\begin{aligned} &\min_{T_{sc}} -\left(d_1 \times J_\eta - d_2 \times J_c\right) \\ &\text{s.t.} \quad \dot{x} = f(x, u, t), \\ &\quad \bar{p}_2 - p_2 \geq 0, \end{aligned} \tag{21}$$

Here, $d_1 = 1$ and $d_2 = 1$ [1/K] are the weights used to scale the cost terms.

5.2. Contextual Bandits problem formulation

The heat pump control problem is formulated as a Contextual Bandits problem and its specifications are as following,

1. *Reward R:* The cost terms J_η and J_c provide relevant information on the system efficiency and the cooling demand, therefore, the reward is defined as the negative of the cost function defined in Eq. (21) and it is expressed,

$$R = J_\eta - J_c \tag{22}$$

2. *Context s:* The stationary operating point of the heat pump system can be defined by stationary conditions of the known external disturbances w, such as ambient air temperature T_{amb}, relative humidity of the air RH_{air}, cooling demand of the passenger and vehicle speed. For the method development, s is defined as a subset of w i.e., T_{amb} and RH_{air} expressed as,

$$s = \begin{bmatrix} T_{amb} & RH_{air} \end{bmatrix}^\top \tag{23}$$

3. *Action a:* The agent's action a is the T_{sc} setpoint i.e., $r_{T_{sc}}$, which is directly correlated with J_η and also impacts J_c. r can take real values. Therefore, we assume a continuous action space, i.e., $a \in \mathcal{A} \subseteq \mathbb{R}^+$.

A-priori information: For exploration, we assume a-priori information on: (1) Safe initial action $a_0 = 15$ °C, (2) Lower bound on action space $\underline{a} = 3$ °C and (3) Constraint on p_2, i.e., $\bar{p}_2 = 21$ bars. $\underline{a}$ is known from a physical understanding of the automotive heat pump system, which ensures that the refrigerant remains in the liquid phase. This value is consistent across all stationary operating conditions i.e., $w \doteq s$. For this application, the upper bound on the action space varies with the operating conditions and it is non-trivial to determine. Therefore, we assume that partial information is available on restricting action space for exploration.

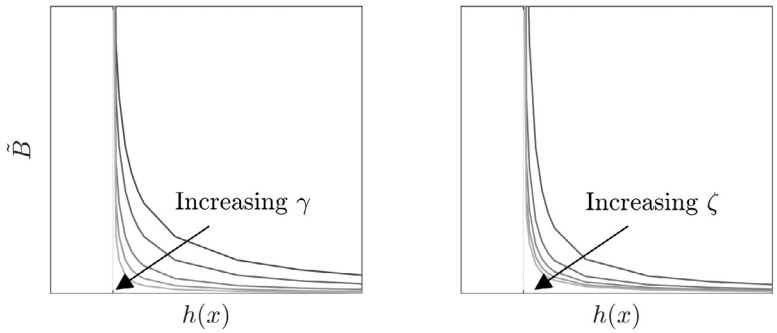


Fig. 9. Sensitivity of $\tilde{B}(x)$ to change in γ, ζ.

Table 2
List of hyperparameters for the Safe-ISE method in simulations.

Parameter	Description	Category	Value
Δa_0	Step-size in action for start-up	Initialization	0.5 °C
ζ	Scaling of rCBF	rCBF	10
γ	Rate of change of rCBF		0.0005
ξ	Small constant for continuity of rCBF		0.0001
l_1	Window length to detect stationary condition	Averaging of R signal	120 s
l_2	Window length for averaging R		60 s
λ	Allowable deviation from setpoint		0.5 °C
k	GPR - kernel function	GPR	Squared-exponential
$\sigma_{f,0}$	GPR - initial estimate of signal standard deviation		1
$\sigma_{l,0}$	GPR - initial estimate of length scale		3
$\sigma_{n,0}$	GPR - initial estimate of noise standard deviation		0.003
$\epsilon_{\tilde{B}}$	Learning rate rCBF	Exploration	30
$\underline{\text{PM}}$	Minimum value for PM		−0.30
Δa^+	Step size outside $\mathcal{A}_\zeta$		2 °C
p	Number of iterations to check for optimality	Stopping criteria	5
ϵ	Small constant for change in optimal action		0.01 °C
$\overline{\tilde{B}}$	Maximum allowable $\tilde{B}$		170
$\overline{N}_{\text{exp}}$	Maximum number of experiments		50

5.3. Hyperparameter selection

The hyperparameters of the Safe-ISE method are listed in Table 2. For initial exploration, a small value for Δa_0 is desired to avoid violation of safety constraint in case a_0 is close to the safe boundary. $\Delta a_0 = 0.5$ °C is chosen to account for inaccuracy in the measured value of T_{sc}. For shaping the rCBF, smaller values of γ, ζ are chosen by trial and error approach such that the rCBF function grows to a large value before $p_2 \to \bar{p}_2$ as shown in Fig. 9.

For the GPR model, a squared-exponential kernel function is chosen due to its suitability for smooth functions, which is consistent for the studied system. The parameters $\sigma_{f,0}, \sigma_{l,0}, \sigma_{n,0}$ represent the initial estimates for the standard hyperparameters of the GPR model in MATLAB. These parameters are optimized by the GPR toolbox in MATLAB as new data points are received during the exploration.

In order to sample true action-value measurements, the reward signal is averaged over $l_2 = 60$ seconds if the system output $y = T_{sc}$ stays stationary i.e., within the tolerance of $\lambda = 0.5$ °C to the agent's action for a window length of $l_1 = 120$ seconds. The exploration process after the initial exploration is controlled by the hyperparameters $\epsilon_{\tilde{B}}, \underline{\text{PM}}$ and Δa^+. For determining the step size in agent's action approaching the safe boundary, the choice of the parameter $\epsilon_{\tilde{B}}$ is very crucial. Smaller values of $\epsilon_{\tilde{B}}$ cause small changes in action step size resulting in slow learning process. However, larger values of $\epsilon_{\tilde{B}}$ can result in constraint violation due to larger step sizes. A small value of $\underline{\text{PM}}$ limits agent to look for peaks closer to current best while a larger value allows agent to explore peaks much smaller than the current best. A larger value of $\underline{\text{PM}}$ represents a conservative nature of the agent and places lower confidence in the action-value of the current best action. With this approach, the agent explores not just actions with larger uncertainty but also actions where a maxima could lie. Δa^+ determines the learning speed of the agent, where a smaller value promotes faster approach towards safe boundary and vice versa.

The termination of the exploration process is controlled by the parameters $p, \epsilon, \overline{\tilde{B}}$, and $\overline{N}_{exp}$. p, ϵ has a direct impact on optimality and number of experiments during exploration. A smaller value of p can result in early stopping of learning and has a higher probability of converging in a local maxima. Whereas a larger value of p can converge close to the most optimal action within the safe action space, however, it can result in larger number of experiments. On the other hand, ϵ can be derived from the desired system performance for example, its smaller value has a higher probability that the agent converges close to the most optimal action in the safe action space. However, it can result in larger number of experiment as the agent is more persistent in getting close to the optimal point. For safety, a tolerance of $h(x = p_2) = 1$ bar is chosen in p_2 with respect to $\bar{p}_2 = 21$ bars, which is equivalent to $\overline{\tilde{B}} = 170$ as calculated using Eq. (5). For the case where the agent does not meet the criteria imposed by $p, \epsilon, \overline{\tilde{B}}, \overline{N}_{exp}$ is introduced to terminate the learning process. $\overline{N}_{exp}$ has a similar impact as p, therefore, a larger value equivalent to 50 is chosen to strike a balance between optimality and the number of experiments.

6. Simulation results

6.1. Test-case

The benefits of the proposed RL-based control method for safety, optimality and time-efficiency are demonstrated in simulation. Table 3 shows the specifications of the stationary test case. For reference, we determine the true reward function as shown in Fig. 10. The true

Table 3
Specifications of the test case for simulation.

Parameter	Description	Value	Unit
T_{amb}	Ambient air temperature	31	[°C]
RH_{air}	Relative humidity of air	40	[%]
$T_{a,c}$	Desired cabin air temperature	20	[°C]
V_v	Vehicle speed	0	[km/h]
u_{fan}	Fan speed	50	[%]
u_{blwr}	Blower speed	30	[%]

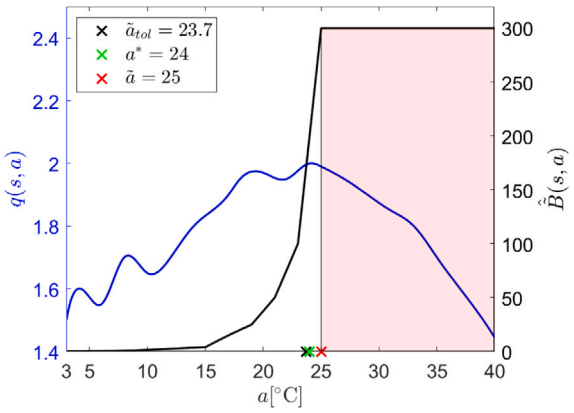


Fig. 10. Illustration of the true action-value function for the test case. The solid blue line shows the true action-value function, black solid line shows true $\hat{\hat{B}}$ and the shaded region in red represents $h(x) \leq 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

function is derived from the measurement data collected by making a sweep of different actions a and gathering corresponding $q(s,a)$ values the steady-state operating point. The underlying true expected reward function is non-convex and consists of multiple maxima.

The safe action space $\mathcal{A}_\varsigma = \{a \,|\, a < 25°C, a \in \mathbb{R}^+\}$, where the critical action $\tilde{a} = 25$ °C represents action at $h(x = p_2) = 0$. The action corresponding to the tolerance $h(x = p_2) = 1$ is $\tilde{a}_{tol} = 23.7$ °C. In the test case, the global optimal action $a^* = 24$ °C lies beyond $\tilde{a}_{tol} = 23.7$ °C while the most optimal action in $\mathcal{A}_\varsigma$ is $a = 23.7$ °C.

6.2. Safe-ISE method

The proposed method is applied to the test case specified above. Fig. 11 shows the exploration result at the end of the learning process. The agent begins the exploration process with $a_0 = 15$ °C and requires $N_{exp} = 18$ experiments for convergence. The agent converges to the safe and locally optimal action $a^*_{agent} = 23.7$ °C $\in \mathcal{X}_\varsigma$ without violating the safety constraint. The exploration is terminated when $\hat{\hat{B}}(a) \geq \overline{\hat{B}}$. The total experiment time T_{exp} required during exploration is calculated as,

$$T_{exp} = \sum_{i=1}^{i=N_{exp}} t_{exp,i} \tag{24}$$

where $t_{exp,i}$ is the time required for ith iteration in the agent-environment interaction. From the experimental data, it is found that the average value for t_{exp} is 300 s for varying actions taken by the agent. T_{exp} required by the agent is 1.5 h.

The exploration process is investigated more closely in Fig. 12. For $i = 4$, the agent determines the actions for which $\hat{\hat{B}}(a) < \hat{\hat{B}}(a_0)$ i.e., safe action space $\mathcal{A}_{\varsigma,i}$ and calculates the values of PM and σ. Initially, the GPR predictions do not provide sufficient information about the underlying action-value function. It is seen that PM($\forall a \in \mathcal{A}_i$) > $\underline{\text{PM}}$, which implies that the agent considers all $a \in \mathcal{A}_i$ as potential options to take in the next iteration. Also, the agent has a similar uncertainty in action-value for all $a \in \mathcal{A}_i \setminus \{a_0, a_1, \ldots, a_{i-1}\}$. For iteration $i = 11$,

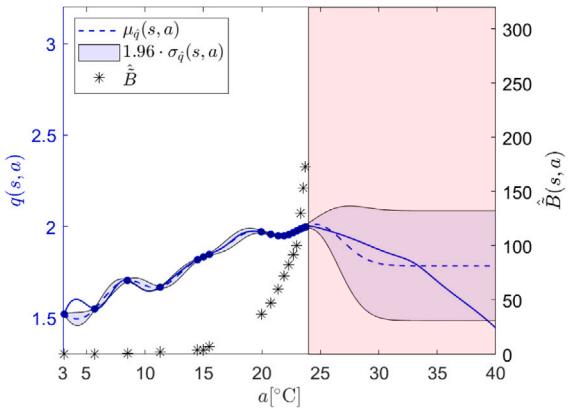


Fig. 11. Exploration results with the Safe-ISE method at $N_{exp} = 18$. (•) in blue are action-values corresponding to agent's actions, the solid blue line shows the true action-value function and the shaded region in red represents $h(x) \leq 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the PM and σ values are different for actions in the action space compared to that in iteration $i = 4$. For $a \in [3,7]$°C, PM($\mathcal{A}_i$) < $\underline{\text{PM}}$ with small uncertainty in action-value, which indicates the absence of global maxima in that action range. Among actions $a \in \mathcal{A}_i$, it is seen that $a \in \mathcal{A}_i^+$ has PM > $\underline{\text{PM}}$ and the highest uncertainty (see red box in Fig. 12), therefore, the agent takes the next action in $\mathcal{A}_i^+$ approaching $\partial\mathcal{X}_\varsigma$. For $i = 18$, the exploration process is terminated as $\hat{\hat{B}}(a) \geq \overline{\hat{B}}$.

6.3. Comparison with state-of-the-art exploration methods

We benchmark the proposed exploration method with two state-of-the-art exploration methods: (1) Random exploration and (2) Uncertainty-driven exploration. In random exploration, the agent chooses actions randomly from a pre-defined action space $\mathcal{A} \in [3,40]$ °C with no knowledge of safe action-space. Whereas in uncertainty-driven method, the agent chooses the action corresponding to the highest uncertainty in action-value estimate at every iteration from the action space $\mathcal{A} \in [3,40]$ °C. For both the methods, the exploration begins with taking the safe initial action $a_0 = 15$ °C. For a consistent comparison, we use the GPR model for action-value approximation and estimate the optimal action in a similar approach as in Safe-ISE method. Moreover, we use similar values for the hyperparameters of the GPR model and the agent's stopping criteria for the comparison study as shown in Table 4.

The comparison results between the state-of-the-art and proposed exploration methods are shown in Table 5. Fig. 13(a) shows the exploration result of applying the random exploration method. It is seen that without a-priori information on the safe action space, the agent randomly takes all actions in the maximum allowable action space $\mathcal{A}$ and violates the system safety constraint. Furthermore, it is seen that certain actions are not explored, which results in large uncertainties in their action-value estimates. Nonetheless, the random exploration method converges to $a = 24.2$ °C, which is close-to the global optimal action $a^* = 24$ °C with a negligible deviation from the true action-values at the cost of safety. Moreover, it requires approximately 38% larger experiment time compared to Safe-ISE.

The result of applying the uncertainty-driven exploration method is shown in Fig. 13(b). Similar to the random exploration, system safety constraints are violated as the agent explores actions with the largest uncertainty in the corresponding action-values in the maximum allowable action space. As expected, the uncertainty-driven exploration requires 12% lower experiment time compared to the random exploration with a marginal penalty on optimality. In comparison to the Safe-ISE method, it requires 22% larger experiment time. In summary,

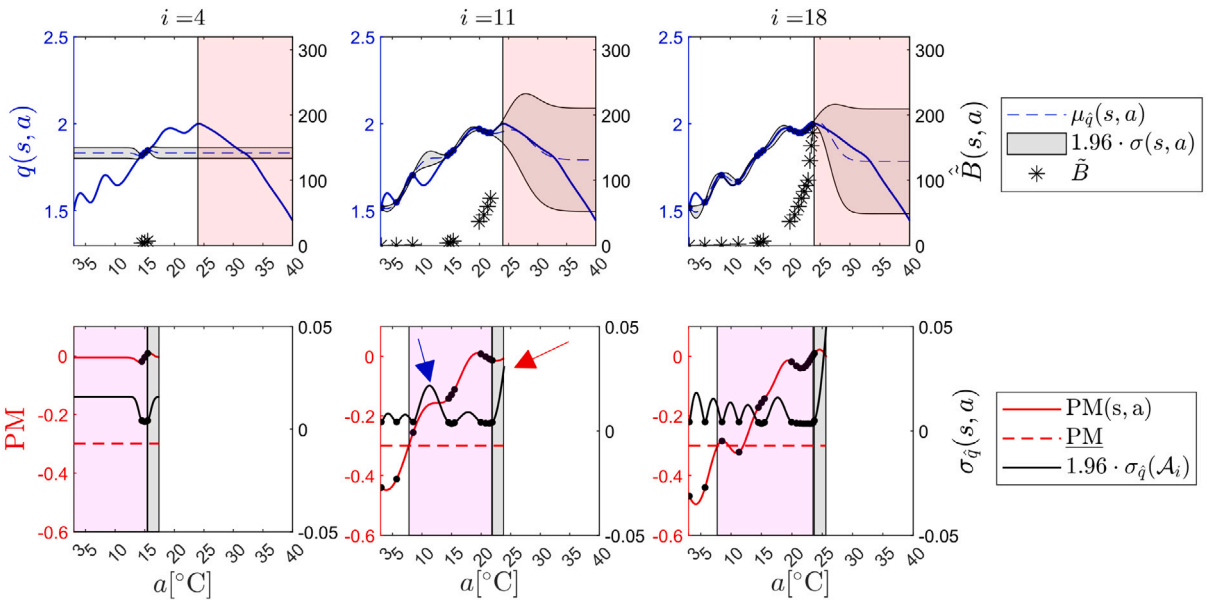


Fig. 12. Evolution of action-value estimate and information gain metrics over the exploration process of Safe-ISE method. (•) in blue are action-values corresponding to agent's actions, the solid blue line shows the true action-value function; the shaded regions in red, magenta and gray represents $h(x) \leq 0$, $\mathcal{A}_{\varsigma,i}$ and $\mathcal{A}_i^+$, respectively. The red arrow indicates the action with the highest uncertainty in $\mathcal{A}_i^+$ and the blue arrow indicates the action with the highest uncertainty in $\mathcal{A}_{\varsigma,i}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
List of hyperparameters for random and uncertainty-driven exploration methods.

Parameter	Description	Category	Value
δ	Small constant for numerical stability	Initialization	0.0001
l_1	Window length to detect stationary condition	Averaging of R signal	120 s
l_2	Window length for averaging R		60 s
λ	Allowable deviation from setpoint		0.5 °C
k	GPR - kernel function	GPR	Squared-exponential
$\sigma_{f,0}$	GPR - initial estimate of signal standard deviation		1
$\sigma_{l,0}$	GPR - initial estimate of length scale		3
σ_n	GPR - initial estimate of noise standard deviation		0.003
p	Number of iterations to check for optimality	Stopping criteria	5
ϵ	Small constant for change in optimal action		0.01 °C
N_{exp}	Maximum number of experiments		50

Table 5
Comparison of simulation results for Safe-ISE method with random and uncertainty-driven exploration methods. Text in bold represents the best performance among the studied methods.

Evaluation criteria	Exploration method		
	Safe-ISE	Random	Uncertainty-driven
Deviation from q^* ($\Delta q = q^* - q^*_{agent}$)	0.0021	0.01	**−0.0016**
Percentage deviation ($\Delta q/q^* \times 100\%$)	0.1%	0.5%	**−0.008%**
Deviation from a^* ($\Delta a = a^* - a^*_{agent}$)	0.3 °C	**0.2**°C	0.5 °C
Percentage deviation ($\Delta a/a^* \times 100\%$)	1.25%	**0.83%**	2.1%
Safe operation ($\tilde{B} \leq \overline{\tilde{B}}$)	Yes	No	No
Number of experiments N_{exp}	**18**	25	22
Total exploration time T_{exp}	**1.5 h**	2.08 h	1.83 h

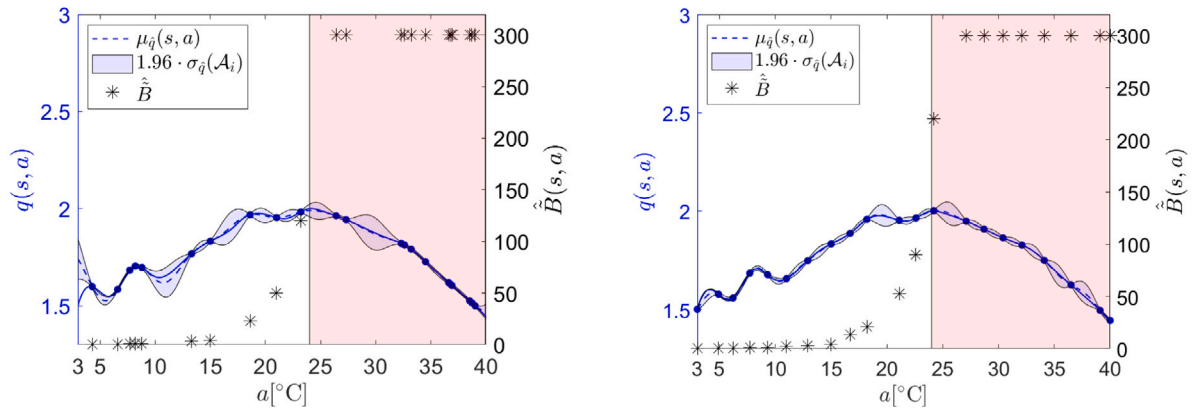
both the random and uncertainty-driven exploration methods does not maintain safe system operation and they require larger number of experiments compared to Safe-ISE method for converging close-to the optimal action.

7. Experimental validation on a vehicle test-bench

In this section, the initial results of implementing the proposed Safe-ISE method on a vehicle test-bench are presented. The vehicle test-bench is equipped with a chassis dynamometer inside a climatic chamber and it has the capabilities to realize a wide range of values for the ambient temperature and relative humidity. The vehicle is equipped with an ETAS rapid prototype system (RPS), for the RL-based controller implementation and validation. The RL-based control method is developed in MATLAB 2017b on a laptop.

7.1. Test cases

The proposed method is validated over two test cases: (1) Nominal operating point (OP1) and (2) Nominal operating point with different constraint bound (OP2) as listed in Table 6. The test cases are steady-state operating points characterized by stationary values of context



(a) Result of applying random exploration after $N_{exp} = 25$.

(b) Result of applying uncertainty-driven exploration after $N_{exp} = 22$.

Fig. 13. Simulation results with existing exploration methods. (•) in blue are action-values corresponding to agent's actions, the solid blue line shows the true action-value function and the shaded region in red represents $h(x) \leq 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

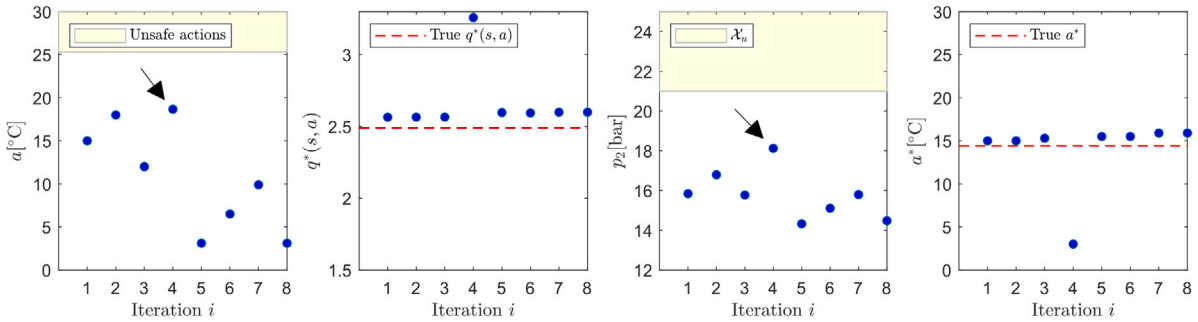


Fig. 14. Experiment test-case OP1: Results of incremental learning on the vehicle test-bench. (•) represent data corresponding to agent's actions. The arrow in black indicates the agent's action and corresponding increase in p_2 with relaxed constraint on system state p_2 compared to OP2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Specifications of the validation operating points (OPs). The underlined text emphasizes the change in the operating conditions compared to the nominal operating point OP1.

OP	Description	T_{amb} [°C]	RH_{air} [%]	Constraint Bound $\bar{p}_2$ [bar]	True optimal action a^* [°C]	Critical action $a_{p_2=\bar{p}_2}$ [°C]
OP1	Nominal	35	10	21	14.4	21.2
OP2	Nominal with different constraint bound	35	10	<u>19</u>	14.4	25.3

$s = \begin{bmatrix} T_{amb} & RH_{air} \end{bmatrix}^\top$ and safety constraint defined by $\bar{p}_2$. In order to find the true values of a^* and $a_{p_2=\bar{p}_2}$, a steady-state T_{sc} sweep is made. For individual value of T_{sc}, the corresponding data on $q(s,a)$ and p_2 is collected, which are fitted using the Gaussian Process Regression model. a^* is determined as follows,

$$a^*(s) = \underset{a=T_{sc} \in \mathcal{A}_{eval}}{\arg\max}\ \mu_{\hat{q}}(s,a) \tag{25}$$

where $\mu_{\hat{q}}(s,a)$ is the mean prediction of the GPR model and $\mathcal{A}_{eval}$ represents the set of T_{sc} values for which the GPR model is evaluated.

7.2. Validation results

The incremental learning by the agent on the vehicle test-bench for OP1 and OP2 is shown in Figs. 14 and 15, respectively. It can be seen that in both the test cases, the agent remains within the safe boundary by not exploring the unsafe actions. In OP1, the agent takes 8 different actions starting from an initial safe action and converges to action $a^*_{agent} = 15.3$ °C, which is within the desired 2% of the true $q^*(s,a)$ determined from the T_{sc} sweep. In OP2, the safety constraint on state p_2 is made more strict by changing its value from 21 bars to 19 bars. As seen from Fig. 15, the agent's behavior is more conservative and explores fewer actions approaching the safe boundary as compared to OP1. In OP1, the agent explores the action $a = 18.7$ °C due to relaxed state constraint unlike in OP2. Nonetheless, the agent converges to action $a^*_{agent} = 14.8$ °C, which is close to the true optimal $a^* = 14.4$ °C. Similar to OP1, the agent converges to within 2% of the true $q^*(s,a)$. These experimental results provide a proof-of-concept of a safe, time-efficient and automated calibration process using the proposed exploration method.

8. Conclusions and future work

In this paper, we present a novel method for safe and time-efficient exploration that requires no prior system model for Reinforcement Learning-based control of complex real-world physical systems. The safety constraint is formulated using the reciprocal control barrier function, which is used to determine the step-size of the agent's actions approaching the safe boundary during exploration. To minimize number of experiments, we combine the reciprocal control barrier function

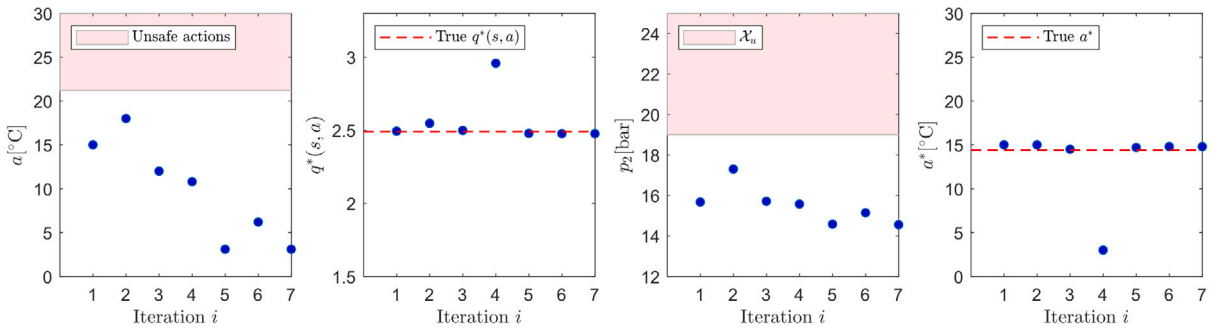


Fig. 15. Experiment test case OP2: Results of incremental learning on the vehicle test-bench. (•) represent data corresponding to agent's actions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with information gain metrics to determine actions which maximize information on the action-values. The required a-priori system knowledge is minimized with the proposed method, which highlights its applicability for control development of new systems. First, we demonstrated the benefits in safety, optimality and number of experiments of our proposed method on an automotive heat pump control system in simulation. For the studied test-case, the proposed method converges to the safe and locally optimal action with a deviation of 0.3 °C from the global optima without violating the safety constraint during the exploration. A significant reduction of 28% and 18% in experiment time is achieved with the proposed method in comparison to the existing exploration methods i.e., random and uncertainty-driven, respectively. Second, the proposed method was validated on a vehicle test-bench for optimizing the steady-state performance of the vehicle thermal system. The experimental results show that the proposed control method converges close to the true optimum in action-values with an accuracy of $\pm 2\%$ for the studied test-cases while ensuring system safety at all times during the exploration. Future work will focus on applying the approach to a wider range of operating conditions. This is an important step towards RL-based control for on-line learning in an on-road vehicle. Special attention will be paid to stability analysis of the system's closed-loop performance.

CRediT authorship contribution statement

Prasoon Garg: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Emilia Silvas:** Writing – review & editing, Supervision. **Frank Willems:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the company DENSO, Japan for financial support to perform this research and the Powertrain R&D team at DENSO Aachen Engineering Centre (AEC), Germany for the technical support and constructive discussions. Especially, Bart van Moergastel, Ron Puts, and Bastian Aust from the Energy Systems R&D team are acknowledged for their support with conducting the experiments.

References

Afsar, M. M., Crump, T., & Far, B. (2022). Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, *55*(7), 1–38.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., & Topcu, U. (2018). Safe reinforcement learning via shielding. In *AAAI conference on artificial intelligence*: *Vol. 32*, (pp. 2669–2678).

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. (2019). Control barrier functions: Theory and applications. In *IEEE European control conference* (pp. 3420–3431). IEEE.

Ames, A. D., Grizzle, J. W., & Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE conference on decision and control* (pp. 6271–6278). IEEE.

Anand, A., Seel, K., Gjærum, V., Håkansson, A., Robinson, H., & Saad, A. (2021). Safe learning for control using control lyapunov functions and control barrier functions: A review. *Procedia Computer Science*, *192*, 3987–3997.

Aradi, S. (2022). Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(2), 740–759.

Berkenkamp, F., Turchetta, M., Schoellig, A., & Krause, A. (2017). Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*, *30*, 908–911.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., & Schoellig, A. P. (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, *5*, 411–444.

Garcıa, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, *16*(1), 1437–1480.

Garg, P., Silvas, E., & Willems, F. (2023). Systematic hyperparameter selection in machine learning-based engine control to minimize calibration effort. *Control Engineering Practice*, *140*, Article 105666.

Geurts, F. (1998). *Abstract compositional analysis of iterated relations: A structural approach to complex state transition systems* (pp. 95–131). Springer.

Gros, S., Zanon, M., & Bemporad, A. (2020). Safe reinforcement learning via projection on a safe set: How to achieve optimality? *IFAC-PapersOnLine*, *53*(2), 8076–8081.

Guo, D., Ktena, S. I., Myana, P. K., Huszar, F., Shi, W., Tejani, A., Kneier, M., & Das, S. (2020). Deep bayesian bandits: Exploring in online personalized recommendations. In *ACM conference on recommender systems* (pp. 456–461).

Hunt, N., Fulton, N., Magliacane, S., Hoang, T. N., Das, S., & Solar-Lezama, A. (2021). Verifiably safe exploration for end-to-end reinforcement learning. In *International conference on hybrid systems: Computation and control* (pp. 1–11).

Keir, M. C., & Alleyne, A. G. (2007). Feedback structures for vapor compression cycle systems. In *2007 American control conference* (pp. 5052–5058). IEEE.

Kormushev, P., Calinon, S., & Caldwell, D. G. (2013). Reinforcement learning in robotics: Applications and real-world challenges. *Robotics*, *2*(3), 122–148.

Kuss, M., & Rasmussen, C. (2003). Gaussian processes in reinforcement learning. *Advances in Neural Information Processing Systems*, *16*.

Ladosz, P., Weng, L., Kim, M., & Oh, H. (2022). Exploration in deep reinforcement learning: A survey. *Information Fusion*, *85*, 1–22.

Marvi, Z., & Kiumarsi, B. (2021). Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control*, *31*(6), 1923–1940.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Nian, R., Liu, J., & Huang, B. (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, *139*, Article 106886.

Norouzi, A., Shahpouri, S., Gordon, D., Shahbakhti, M., & Koch, C. R. (2023). Safe deep reinforcement learning in diesel engine emission control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *237*(8), 1440–1453.

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Annual conference on genetic and evolutionary computation: Vol. 1*, (pp. 525–532).

Prajna, S., & Jadbabaie, A. (2004). Safety verification of hybrid systems using barrier certificates. In *International workshop on hybrid systems: Computation and control* (pp. 477–492). Springer.

Qi, X., Luo, Y., Wu, G., Boriboonsomsin, K., & Barth, M. (2019). Deep reinforcement learning enabled self-learning control for energy efficient driving. *Transportation Research Part C: Emerging Technologies*, *99*, 67–81.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*(3–4), 285–294.

Urteaga, I., & Wiggins, C. H. (2017). Bayesian bandits: balancing the exploration-exploitation tradeoff via double sampling. arXiv preprint arXiv:1709.03162.

Wagenmaker, A., & Pacchiano, A. (2023). Leveraging offline data in online reinforcement learning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Machine learning research: Vol. 202, International conference on machine learning* (pp. 35300–35338). PMLR.

Wieland, P., & Allgöwer, F. (2007). Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, *40*(12), 462–467.

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning: Vol. 2*, MA: MIT PRESS Cambridge.

Wu, Y., Shariff, R., Lattimore, T., & Szepesvári, C. (2016). Conservative bandits. In *International conference on machine learning* (pp. 1254–1262). PMLR.

Yamanaka, Y., Matsuo, H., Tuzuki, K., Tsuboko, T., & Nishimura, Y. (1997). Development of sub-cool system. *SAE Transactions*, *106*, 129–134.

Yu, M., Yang, Z., Kolar, M., & Wang, Z. (2019). Convergent policy optimization for safe reinforcement learning. *Advances in Neural Information Processing Systems*, *32*.

Zhao, Z., Xun, J., Wen, X., & Chen, J. (2022). Safe reinforcement learning for single train trajectory optimization via shield SARSA. *IEEE Transactions on Intelligent Transportation Systems*, *24*(1), 412–428.

Zhu, R., & Kveton, B. (2022). Safe optimal design with applications in off-policy learning. In *International conference on artificial intelligence and statistics* (pp. 2436–2447). PMLR.