2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Markov Decision Processes with Unknown State Feature Values for Safe Exploration using Gaussian Processes

Matthew Budd[1], Bruno Lacerda[1], Paul Duckworth[1], Andrew West[2], Barry Lennox[2] and Nick Hawes[1]

Abstract— **When exploring an unknown environment, a mobile robot must decide where to observe next. It must do this whilst minimising the risk of failure, by only exploring areas that it expects to be *safe*. In this context, safety refers to the robot remaining in regions where critical environment features (e.g. terrain steepness, radiation levels) are within ranges the robot is able to tolerate. More specifically, we consider a setting where a robot explores an environment modelled with a Markov decision process, subject to bounds on the values of one or more environment features which can only be sensed at runtime. We use a Gaussian process to predict the value of the environment feature in unvisited regions, and propose an *estimated* Markov decision process, a model that integrates the Gaussian process predictions with the environment model transition probabilities. Building on this model, we propose an exploration algorithm that, contrary to previous approaches, considers probabilistic transitions and explicitly reasons about the uncertainty over the Gaussian process predictions. Furthermore, our approach increases the speed of exploration by selecting locations to visit further away from the currently explored area. We evaluate our approach on a real-world gamma radiation dataset, tackling the challenge of a nuclear material inspection robot exploring an *a priori* unknown area.**

I. INTRODUCTION

For many tasks that autonomous mobile robots are well suited to, the robot may need to plan and navigate in an environment where there is an *a priori* unknown distribution of a *hazard*. This could be steep terrain (for planetary rovers), water depth or current (for underwater vehicles), or radiation exposure (for disaster recovery/nuclear inspection). In this paper, we provide a robust safe exploration approach to handle cases where the dynamics of this hazard are *unknown*.

We build upon the work in [1], extending it to handle probabilistic transition models and support more complex safety specifications. The aim of our exploration task is to maximise robot safety while minimising the expected cumulative cost to build a model of the dynamics of the environment features. The exploration is considered complete when the model reaches a user-specified degree of accuracy for the states that it can safely visit.

We use a Gaussian process (GP) [2] to model and predict the values of unknown environment features, since a GP provides a flexible, non-parametric method of function approximation. We assume the hazard is a smooth, stationary function, and predict mean and uncertainty across the domain – this is important as a safe approach requires taking uncertainty about the safety of states into account.

We present two new formalisms for describing uncertainty over unknown feature values in an MDP setting, and an algorithm which makes use of these for safe exploration. The first of these formalisms is an *MDP with Unknown Feature Values* (U-MDP) which we use to model the exploration problem. The second formalism is an *Estimated MDP* (Est-MDP) which we use to plan in an approximation of the U-MDP. In both models, there is a subset of state features with unknown values. In the U-MDP there is a deterministic mapping which defines the values of unknown state features at given known states. The Est-MDP approximates the U-MDP by replacing the deterministic mapping with a probabilistic mapping based on its current knowledge. Our new algorithm, *SafeEst-MDP*, exploits this probabilistic mapping, encoded in the Est-MDP transition function, to plan safe paths to states expected to be informative during exploration. These safe paths take into account the probability of falling into an unsafe state at each step needed to reach, and return from, one of these informative states.

The contributions of this paper are a new framework for tackling exploration tasks under uncertainty, and the evaluation of the framework on a real-world nuclear inspection dataset. We demonstrate that our method offers several advantages over existing safe approaches, including support for probabilistic action outcomes, and more expressive safety constraints. Our experimental results show that our proposed exploration approach significantly outperforms the SafeMDP algorithm presented in [1]. This is both in terms of the cost incurred, and distance travelled, to complete the exploration, and the number of measurements required to do so. Overall, our approach is significantly more efficient in producing a GP model that accurately predicts the unknown state feature values in the set of safe states.

II. RELATED WORK

MDPs are commonly used for planning under uncertainty for robots e.g. [3], [4], [5], and there has been previous work concerning safe exploration of this type of model. Safe exploration of MDPs (without explicitly modelling the unknown state feature values) has been addressed in [6], where a technique for the problem of exploring an MDP whilst ensuring returnability to the initial state is presented. GP exploration (without the added reachability constraints of an MDP) has been investigated in [7], where the value of an objective function is optimised while avoiding the risk of sampling the function where its value is below a safety threshold. Building upon these works, [1] introduced the *SafeMDP* algorithm for safe exploration of MDPs using

[1] Dept. of Engineering Science, University of Oxford; {mbudd, bruno, pduckworth, nickh}@robots.ox.ac.uk

[2] Dept. of Electrical and Electronic Engineering, University of Manchester; {andrew.west, barry.lennox}@manchester.ac.uk

GPs. SafeMDP reasons about the notions of *reachability* from and *returnability* to a set of states known to be “safe”. Our approach builds on this, proposing a novel model for the safe exploration of states in an MDP. Our *MDP with Unknown Feature Values* allows us to drop SafeMDP’s assumption of a deterministic transition function, and include GP estimates directly into the planning model. This facilitates the introduction of more complex safety specifications. We compare our algorithm with SafeMDP and show that we are able to more efficiently and safely explore the environment. Other works have built on [1], investigating goal-driven behaviour [8], [9]. We aim to add goal-driven behaviour in future work.

As well as MDP approaches that assume a fully known model of the environment, partially observable MDPs (POMDPs) have been used in robotics [10] to plan to gain information about the world [11], [12]. Similar to our setting, exploration problems in a sequential Bayesian optimisation framework have also been posed as POMDPs. Due to their greatly increased complexity, exact solutions to POMDPs are generally intractable and previous literature often makes use of Monte Carlo Tree Search (MCTS) techniques to generate approximate solutions.

Examples of POMDPs for path planning with a GP observation model can be found in [13], [14], [15]. These POMDP approaches are able to plan in a non-myopic manner by reasoning over beliefs, and aim to carry out Bayesian optimisation-based “informative path planning”. This is similar in many ways to our goal of planning safe paths to informative states – however, they do not consider a safety constrained setting as we do. Although one can interpret our use of a GP as maintaining belief over state values in much the same way as a POMDP, we do not directly reason about partial observability. Our path planning approach is therefore more myopic than a POMDP, but requires significantly less computation (giving better scalability to larger problems) and provides better guarantees on safety than is possible with a standard POMDP reward structure. It does this by separating the safety of paths (based on constrained reachability calculations) from the task of determining informative states to sample. The safe exploration task cannot be directly translated into a reward structure over a POMDP without its dimensionality growing unfeasibly large. Moreover, it is not possible to encode safety and reward into a single reward in a principled manner.

III. PRELIMINARIES

A. Markov Decision Processes & Constrained Reachability

An MDP is defined as a tuple $\mathcal{M} = \langle S, \overline{s}, A, T, c \rangle$, where S is a finite set of states; $\overline{s} \in S$ is the initial state; A is a finite set of actions; $T : S \times A \times S \to [0, 1]$ is a probabilistic transition function; and $c : S \times A \to \mathbb{R}_{\geq 0}$ is a cost function. Examples of cost functions are the expected time to execute an action, or the expected energy required to do so.

A *path* through an MDP is a sequence $w = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} ...$ where $T(s_i, a_i, s_{i+1}) > 0$ for all $i \in \mathbb{N}$. We denote the set of all paths of $\mathcal{M}$ starting from state s as $Path_{\mathcal{M},s}$. The choice of action to take at each step of the execution of an MDP is made by a *policy*. In this paper, we consider *deterministic, stationary* policies, defined as functions $\pi : S \to A$ that map each state $s \in S$ to the action to execute in s, and denote the set of all such policies as Π. Given an MDP $\mathcal{M}$ and a policy $\pi \in \Pi$, we can define a probability measure $Pr^{\pi}_{\mathcal{M},s}$ over the set of paths $Path_{\mathcal{M},s}$ [16]. Furthermore, for a measurable function $X : Path_{\mathcal{M},s} \to \mathbb{R}$, we write $E^{\pi}_{\mathcal{M},s}(X)$ for the expected value of X with respect to $Pr^{\pi}_{\mathcal{M},s}$.

In this work, we consider *cost-optimal constrained reachability* problems for which the probability of satisfaction might be less than one. These involve identifying a policy to reach a set of goal states whilst avoiding a set of forbidden states. More formally, let $G \subset S$ be a set of goal states and $F \subset S$ be a set of forbidden states. We define the set of paths that reach G whilst avoiding F as:

$$\begin{aligned} reach_{\neg F,G} = \{(s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} ...) \in Path_{\mathcal{M},s_0} \mid \text{ exists } i \in \mathbb{N} \\ \text{such that } s_i \in G \text{ and } s_j \notin F \text{ for all } j \leq i\}. \end{aligned} \tag{1}$$

We will consider policies that are cost-optimal, in the sense that they minimise the expected cumulative cost to reach either a goal state, or a state where reaching the goal is not possible. Given $w = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} ... \in Path_{\mathcal{M},s_0}$, we define l_w as the timestep until which cost will be accumulated for path w:

$$l_w = \begin{cases} \min_l \text{ s. t. } s_l \in G & \text{if } w \in reach_{\neg F,G} \\ \min_l \text{ s. t. } & \\ Pr^{\max}_{\mathcal{M},s_l}(reach_{\neg F,G}) = 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $Pr^{\max}_{\mathcal{M},s_l}(reach_{\neg F,G})$ denotes the supremum over Π of $Pr^{\pi}_{\mathcal{M},s_l}(reach_{\neg F,G})$. Note that the second condition in the definition of l_w encompasses the case where a forbidden state is visited before a goal state, and the case where the path never reaches a goal or a forbidden state. Finally, consider the function $cumul_{\neg F,G} : Path_{\mathcal{M},s} \to \mathbb{R}_{\geq 0}$ that maps a path $w = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \ldots$ to the cost accumulated up to l_w:

$$cumul_{\neg F,G}(w) = \sum_{i=0}^{l_w - 1} c(s_i, a_i). \tag{3}$$

Defining the set of policies that maximise the probability of reaching G whilst avoiding F as $\Pi^* = \{\pi \in \Pi \mid \pi = \arg\max_{\pi'} Pr^{\pi'}_{\mathcal{M},\overline{s}}(reach_{\neg F,G})\}$, the optimisation objective for constrained reachability can now be defined as finding the policy $\pi_{\neg F,G} \in \Pi$ that has minimal expected cumulative cost:

$$\pi_{\neg F,G} = \operatorname*{arg\,min}_{\pi \in \Pi^*} E^{\pi}_{\mathcal{M},\overline{s}}(cumul_{\neg F,G}). \tag{4}$$

The above optimisation problem is a variant of a safest and stochastic shortest path problem [17], with the minimal expected cost being known to converge to a finite value. In order to find $\pi_{\neg F,G}$, we encode the constrained reachability problem in *co-safe linear temporal logic* and use the approach presented in [4].

We will also be interested in the probability of reaching a set of states within a bound on the number of allowed timesteps. For notational consistency, we denote as $Pr^{\pi}_{\mathcal{M},s}\left(reach^{\leq n}_{\neg\emptyset,G}\right)$ the probability, starting in state s and under policy π, of reaching a state in G *within n steps*. We can calculate this probability by building the set $S^{\leq n}$ of states reachable by applying π from s for n timesteps, and determining the probability of reaching G in the sub-model of $\mathcal{M}$ with state space equal to $S^{\leq n}$.

B. Gaussian Process

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [2]. A GP model is of the form $f(s) \sim \mathcal{GP}(m(s), k(s,s'))$, and represents a probability distribution over functions, fully specified by its mean function $m(s)$ and kernel function $k(s,s')$. We can let $m(s) = 0$ without loss of generality.

Given a set of j noisy observations $\mathbf{y} = (f(s_1) + n_1, \ldots, f(s_j) + n_j)$ (where $n_j \sim \mathcal{N}(0, \sigma_n^2)$ is Gaussian observation noise) at the set of observed states $S_j = [s_1, \ldots, s_j]^\intercal$, we can use a GP to predict the value of the unknown function at all other states.

The predictive posterior is a Gaussian distribution with mean $\mu_j(s) = \mathbf{k}_j^\intercal(\mathbf{K}_j + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}$, covariance $\Sigma_j(s,s') = k(s,s') - \mathbf{k}_j^\intercal(\mathbf{K}_j + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}_j(s')$ and variance $\sigma_j^2(s) = k_j(s,s)$. For these equations, $\mathbf{k}_j = [k(s_1,s), \ldots, k(s_j,s)]^\intercal$, the positive semi-definite kernel matrix $\mathbf{K}_j = [k(s,s')]_{s,s' \in S_j}$, and $\mathbf{I} \in \mathbb{R}^{j \times j}$ is the identity matrix.

Regularity assumptions must be made about the similarity of the unknown function at nearby states. The choice of kernel and kernel hyperparameters encodes these regularity assumptions. Hyperparameters may include lengthscale and variance parameters, and their values may be fixed or assigned a prior distribution to encode prior knowledge. We optimise the marginal log likelihood to best estimate the GP hyperparameters online.

We make the standard modelling assumptions that the unknown function f has bounded norm in the Reproducing Kernel Hilbert Space associated with the chosen kernel function, and also that it is Lipschitz continuous with respect to some metric $d(\cdot,\cdot)$ on S [1], [8].

IV. PROBLEM FORMULATION

We define an MDP with Unknown Feature Values (U-MDP) $\mathcal{M}^o = \langle S^o, \overline{s}, A, T^o, c\rangle$ as an MDP where the state space is factored as $S^o = S_k \times S_e$, where $S_k = S_k^1 \times \ldots \times S_k^{n_k}$ is a set of n_k state features with *known* values and $S_e = S_e^1 \times \ldots \times S_e^{n_e}$ is a set of n_e state features with *unknown* values. Furthermore, there is an unknown mapping $o : S_k \rightarrow S_e$ that defines which values $o(s_k) \in S_e$ are observed at $s_k \in S_k$. In other words, the state of the U-MDP is defined as $(s_k, o(s_k)) \in S_k \times S_e$, where the mapping function o is *a priori* unknown. Finally, given that a state is uniquely defined by the value of the known state feature $s_k \in S_k$, the outcome of the transition function only represents the change in the known state feature. Formally, $T^o : (S_k \times S_e) \times A \times S_k \rightarrow [0,1]$, where $T^o((s_k, s_e), a, s'_k)$ represents the probability of moving to state $(s'_k, o(s'_k))$ given that action a was taken at state $(s_k, o(s_k))$. Note that this formalisation allows us to make the dynamics of the state features with known values dependent on the state features with unknown values, which allow us for richer modelling than previous works. We also define a safety function over the states as $\chi : S \rightarrow \{0,1\}$ where $\chi(s) = 1$ when s is considered safe and 0 otherwise. A simple safety function could be an upper-bound threshold $b \in \mathbb{R}$ on the value of a state feature in S_e, as used in Section VI. However, our approach allows for safety to be defined as an arbitrary Boolean function over all state features, something which is not possible with existing approaches. The sets of safe and unsafe states can then be defined as $\mathit{safe} = \{s \in S_k \times S_e \mid \chi(s) = 1\}$ and $\mathit{unsafe} = (S_k \times S_e) \setminus \mathit{safe}$, respectively.

Our approach is to use a GP, trained on observations up to timestep t, to iteratively estimate the mapping o between known state feature values and their corresponding unknown state feature values. To avoid dealing with continuous state spaces, which is outside of the scope of this work, and to simplify notation, we assume a single state feature with unknown values. This state feature can take values in a finite partition of $\mathbb{R}$, i.e. $S_e = \{I_1, \ldots, I_p\}$ where I_i are non-overlapping intervals of $\mathbb{R}$ such that $\cup_{i=1,\ldots,p} I_i = \mathbb{R}$. Thus, for an exploration timestep t, we define $\mathcal{GP}_t : S_k \times S_e \rightarrow [0,1]$ such that $\mathcal{GP}_t(s_k, I)$ is the probability, predicted by the GP taking into account the recorded observations up to timestep t, of $o(s_k) \in I$. To compute $\mathcal{GP}_t(s_k, I)$, we integrate the GP posterior at s_k over interval I. Further details will be given in V-B. The extension to multiple unknown value state features (using separate GPs or a multidimensional GP) is straightforward. Furthermore, with a slight abuse of notation, given a set of intervals $\mathcal{I} \in 2^{S_e}$ we define $\mathcal{GP}_t(s_k, \mathcal{I}) = \sum_{I \in \mathcal{I}} \mathcal{GP}_t(s_k, I)$. In particular, we will write $\mathcal{GP}_t(s_k, \mathit{safe})$ to denote the probability, according to the GP given the recorded observations up to timestep t, of s_k being safe.

The problem we tackle in this paper can now be defined as, given a user-defined safety function over the states of the U-MDP $\mathcal{M}^o$, observe sufficient states to estimate the unknown mapping o to a given accuracy ϵ, across the reachable states in *safe*, without visiting a state in *unsafe*. We assume that we have a known-safe starting set (and corresponding noisy observations), that is sufficient to give enough information to be able to start exploring. The starting set must also satisfy the reachability and returnability requirements that are a key part of the exploration algorithm.

We denote noisy observations of o as $\overline{o}$, and assume that the observation noise $\sigma_n \ll \epsilon$.

V. SAFE EXPLORATION FRAMEWORK

A. Overall Description

The flow diagram (Figure 1) provides a high-level description of the exploration approach. At each exploration step, the robot uses its current knowledge of the U-MDP to determine which *goal* state it should next visit in order to best improve its knowledge of other uncertain states. It uses an Estimated MDP (Est-MDP) to represent its current knowledge of the U-MDP and makes decisions based on this estimation. In the

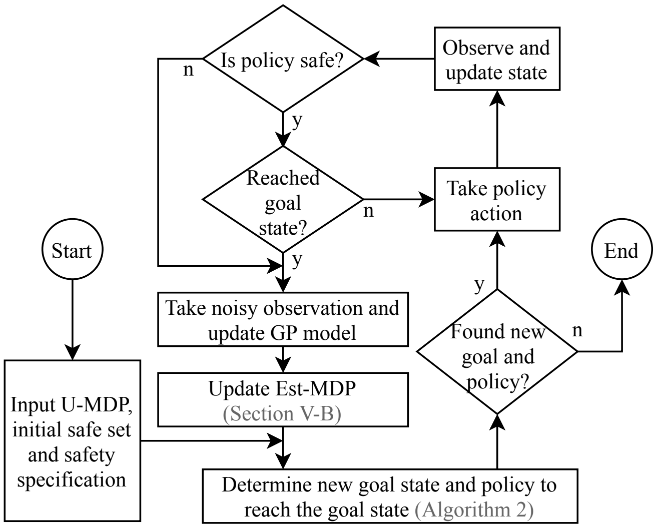
Start
Input U-MDP, initial safe set and safety specification
Determine new goal state and policy to reach the goal state (Algorithm 2)
Found new goal and policy?
n
End
y
Take policy action
Observe and update state
Is policy safe?
n
y
Reached goal state?
n
y
Take noisy observation and update GP model
Update Est-MDP (Section V-B)

Fig. 1. Flow diagram of the exploration method.

Algorithm 1 Safe Exploration (*SafeEst-MDP*)

```
Input: U-MDP $\mathcal{M}^o$, safety function $\chi$, kernel $k(s,s')$, start observations $\overline{o}(S^E_{k,0})$ Output: Explored U-MDP
1:  $s = (s_k, \overline{o}(s_k)) \leftarrow \overline{s}$;  $s_g \leftarrow s_k$;  $t \leftarrow 0$
2:  while $s_g \neq nil$ do
3:      if $s_k = s_g$ or $Pr^{\pi}_{\mathcal{M}^e_t,s}\left(reach^{\leq n}_{\neg\emptyset,unsafe}\right) > (1 - p_{\min})$ then
4:          $t \leftarrow t + 1$
5:          $S^E_{k,t} \leftarrow S^E_{k,t-1} \cup \{s_k\}$
6:          $\mathcal{GP}_t \leftarrow$ update and optimise $\mathcal{GP}_{t-1}$ with $\overline{o}(s_k)$
7:          $s_g,\ \pi \leftarrow$ ChooseGoal (Algorithm 2, Section V-C)
8:      end if
9:      $s' = (s'_k, \overline{o}(s'_k)) \leftarrow$ execute $\pi(s)$ and observe outcome
10:     $s \leftarrow s'$
11: end while
12: No new goal state identified, end exploration
```

following we define the Est-MDP and then describe how our algorithm selects goal states to explore.

We provide further detail on the approach in Algorithm 1. The algorithm receives a U-MDP $\mathcal{M}^o$ to be explored, and the starting set of known states $S^E_{k,0}$, with corresponding observations. To encode any prior knowledge about the environment features, a kernel function for the GP is also provided along with any prior distributions on kernel hyperparameters.

The exploration algorithm repeatedly finds new goal states to explore, until no more goals are available (line 2). Similar to the approach proposed by Turchetta et al. [1], the algorithm maintains a set of explored states $S^E_{k,t}$ that have been visited until timestep t (and must therefore be safe), which is passed to the goal selection algorithm. However, we make use of a richer underlying model and a more complex scoring function for deciding between potential goal states to observe.

The robot checks its current policy over an n-step horizon to determine the probability of remaining in safe states over the next n action-choice steps – this corresponds to checking whether the policy can continue being executed safely from the current state. The calculation is based on the current version $\mathcal{M}^e_t$ of the Est-MDP, obtained from $\mathcal{M}^o$ and $\mathcal{GP}_t$ as explained in Section V-B, i.e., we use the current approximation of the underlying model for each policy check. When the policy safety probability goes under a user-defined threshold p_{min} or the robot reaches its current goal (line 3), then the robot carries out the sampling procedure in lines 4–7. The current state is added to the explored set, the GP is updated with a new observation at the current state, and a new goal and policy are chosen according to Algorithm 2.

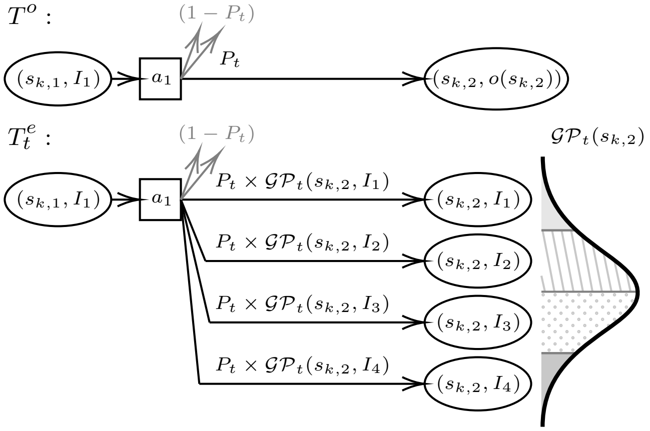
T^o :
$(1 - P_t)$
P_t
$(s_{k,1}, I_1)$
a_1
$(s_{k,2}, o(s_{k,2}))$
T^e_t :
$(1 - P_t)$
$\mathcal{GP}_t(s_{k,2})$
$(s_{k,1}, I_1)$
a_1
$P_t \times \mathcal{GP}_t(s_{k,2}, I_1)$
$(s_{k,2}, I_1)$
$P_t \times \mathcal{GP}_t(s_{k,2}, I_2)$
$(s_{k,2}, I_2)$
$P_t \times \mathcal{GP}_t(s_{k,2}, I_3)$
$(s_{k,2}, I_3)$
$P_t \times \mathcal{GP}_t(s_{k,2}, I_4)$
$(s_{k,2}, I_4)$

Fig. 2. Combination of probabilistic transition function and GP distribution.

If the policy check is satisfied and the current goal state has not been reached then the robot executes the current policy, observing the environment and updating the state accordingly (lines 9–10). Note that if the U-MDP transition function depends only on state features with known values S_k, then the policy action at $(s_k, \overline{o}(s_k))$ will be independent of the value of $\overline{o}(s_k)$. In such cases, it is only necessary to observe the unknown state feature values when updating the GP, not when updating the state for policy execution in line 9.

B. Estimated MDP

The *Estimated MDP* (Est-MDP) encodes both the probabilistic transition function of $\mathcal{M}^o$ and the GP model at timestep t. We define the Est-MDP transition function by weighting the transitions in $\mathcal{M}^o$ with probabilities of observing the different values of S_e, according to $\mathcal{GP}_t$. Formally, given U-MDP $\mathcal{M}^o$ and $\mathcal{GP}_t : S_k \times S_e \to [0,1]$ at timestep t, the Est-MDP at timestep t is defined as $\mathcal{M}^e_t = \langle S_k \times S_e, \overline{s}, A, T^e_t, c\rangle$ where:

$$T^e_t((s_k, I), a, (s'_k, I')) = T^o((s_k, I), a, s'_k)\mathcal{GP}_t(s'_k, I'). \tag{5}$$

Figure 2 illustrates the construction of T^e_t, where integration over the probability density is used to determine the transition probabilities to assign to each interval.

C. Choice of goal state

The choice of goal states makes use of the Est-MDP model, and is detailed by Algorithm 2. A good goal state should provide information when observed (i.e. have a high predictive variance before observation) relative to the cost taken to reach it, and importantly be safe to reach and return from.

The set of candidate goal states consists of states that are considered safe with high probability and that have a

Algorithm 2 ChooseGoal

```
Input: Est-MDP $\mathcal{M}^e_t$, $S^E_{k,t}$, $\mathcal{GP}_t$, current state $s$, $\chi$, $k(s, s')$
Output: New goal state $s_g$ and corresponding policy $\pi$
 1: $S^U_{k,t} \leftarrow S_k \setminus S^E_{k,t}$
 2: $S_{cand} \leftarrow \{s_k \in S^U_{k,t} \mid \mathcal{GP}_t(s_k, safe) > p_{min}$ and $\Sigma_t(s_k) > \epsilon\}$
 3: while $S_{cand} \neq \emptyset$ do
 4:     Take next batch of N states $S_N$ with highest uncertainty in $S_{cand}$
 5:     for $s_{cand} \in S_N$ do
 6:         $p_{reach} \leftarrow Pr^{\max}_{\mathcal{M}^e_t, s}(reach_{\neg unsafe, \{(s_{cand}, I) \mid I \in S_e\}})$
 7:         $p_{return} \leftarrow Pr^{\max}_{\mathcal{M}^e_t, s_{cand}}(reach_{\neg unsafe, \{(s_k, \overline{o}(s_k)) \mid s_k \in S^E_{k,t}\}})$
 8:         if $p_{reach} < p_{min}$ or $p_{return} < p_{min}$ then
 9:             Remove $s_{cand}$ from $S_N$
10:         end if
11:     end for
12:     if $S_N \neq \emptyset$ then
13:         $s_g \leftarrow \arg\max_{s_k \in S_N} \text{score}_t(s, s_k)$
14:         return $s_g$, and corresponding policy $\pi_{\neg unsafe, \{(s_g, I) \mid I \in S_e\}}$
15:     end if
16: end while
17: return $nil$
```

GP predictive variance of $\Sigma_t(s_k) > \epsilon$, as shown in line 2 of Algorithm 2. Other than the safety check, there are two additional checks that are carried out to identify a goal state among the set of candidate goal states: reachability and returnability. The state with the highest score (goal score function - Section V-D) that passes all 3 checks is returned as the new goal state. The policy returned is the policy generated from the reachability check for the chosen goal state, on the Est-MDP.

We calculate reachability and returnability probabilities p_{reach} and p_{return}, based on constrained reachability problems over the current Est-MDP with the forbidden set defined as the set of unsafe states. These calculations are carried out in batches of N states with the highest variance and therefore likely highest score, to avoid computing scores for too many low-scoring states. The reachability check (line 6) is performed between the current state s_t (as the initial state) and and a goal set of all states in the Est-MDP with known feature s_{cand}. The returnability check (line 7) is performed between s_{cand} (as the initial state)[1] and a goal set of the already explored (hence surely safe) states. These probabilities are compared to p_{min}, which can be considered as a measure of the level of risk that the robot is willing to accept during exploration, in line 8. If the algorithm reaches line 16, then the GP posterior at all safely reachable unexplored states has low enough predictive variance, and the goal choice algorithm returns nil, effectively terminating exploration.

D. Goal Scoring Function

The goal scoring function $\text{score}_t : S \times S_k \to \mathbb{R}$ is designed to indicate how beneficial a state s_k would be to visit and observe, given the current state s and the knowledge of the Est-MDP at timestep t. This score should take into account $\Sigma_t(s_k)$ (i.e. the GP's predictive variance at s_k), the optimal expected cost $c^{s_k}_s$ to reach $\{(s_k, I) \mid I \in S_e\}$ from current state s whilst avoiding unsafe states (i.e the expected cost to reach s_k under policy $\pi_{\neg unsafe, \{(s_k, I) \mid I \in S_e\}}$), and the reachability/returnability probabilities for s_k. A suggested scoring function is:

$$\text{score}_t(s, s_k) = (\Sigma_t(s_k))(c^{s_k}_s)^{-\gamma_1}(p_{reach}p_{return} - p^2_{min})^{\gamma_2}, \tag{6}$$

where the parameters γ_1 and γ_2 provide relative weightings on different parameters.

VI. Experiments

We validate our approach with two experiments based on the scenario of a mobile robot investigating a radioactive environment. For these experiments, we consider the MDP cost function to represent expected time for action execution, taken to be equivalent to the travel distance the transition entails.

A. Radiation Data Collection

We collected a real-world dataset of gamma radiation intensity observations using a mobile robot. The University of Lancaster Neutron Laboratory houses a 75 MBq californium-252 source encased in a steel cladded container of light water. The source produces direct gamma emission through spontaneous fission, as well as through neutron activation of other materials. The source is exposed to one side of the container, and a radiation field is produced locally within the laboratory environment.

With the source exposed, a Clearpath Jackal unmanned ground vehicle (UGV) was teleoperated from a remote location to explore the facility, using 2D LIDAR and optical cameras for operator situational awareness. The UGV was equipped with a compact Scionix CeBr_3 scintillator detector, coupled with a Mixed-Field Analyzer (MFA) from Hybrid Instruments. As the CeBr_3 detector is inherently unresponsive to neutrons, it discriminates between neutron and gamma fluxes. This radiation instrumentation was previously developed for integration with submersible robots for characterisation of the Fukushima Daiichi nuclear sites [18].

Individual photon count events are integrated over one second by the MFA, and recorded by the robot as an integer count per second at a frequency of 1 Hz. Three circuits of the facility and the source container were completed before returning to a starting location external to the laboratory. Through the use of SLAM (Simultaneous Localisation and Mapping), the radiation intensity was cross referenced to a spatial location, producing a dataset of integer counts and (x,y) locations shown in Figure 3, where the key shows counts per second values. We use 554 of the possible 1037 data points as we take the median count per second value when data points arise at the same (x,y) location.

We create a "ground-truth" model for reference, which is a GP trained on the 554 dataset points using a Matern 3/2 kernel. The GP posterior is also shown in Figure 3, and is a good fit for the expected behaviour of a radiation source as

[1] This is a notational simplification because $s_{cand} \in S_k$, hence it defines a set of possible initial states $\{(s_{cand}, I) \mid I \in S_e\}$. We consider this when calculating the returnability probability by calculating the constrained reachability probability for each possible initial state (s_{cand}, I), and weighing the results according to the probability of reaching (s_{cand}, I) when executing the optimal constrained reachability policy $\pi_{\neg unsafe, \{(s_{cand}, I) \mid I \in S_e\}}$.

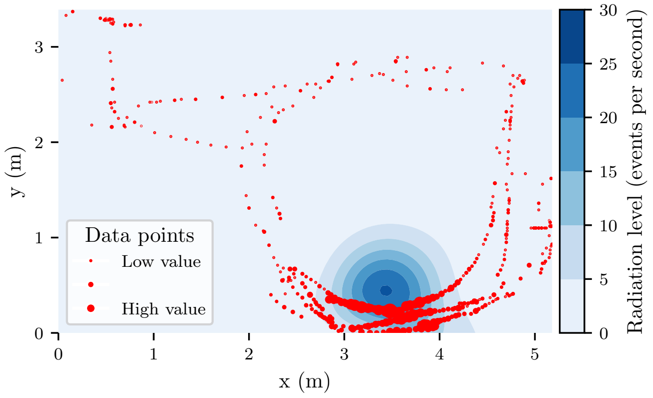

Fig. 3. Predicted posterior distribution over environment feature (gamma radiation level), based on GP model. The red dots are sample points in the dataset that can be observed.

it is symmetrical and falls off in a $1/r^2$ manner as physics would suggest.

B. Single Radiation Source Dataset

We create a U-MDP from the points in the dataset, resulting in a known state space of 554 states and a single unknown value state feature to represent the radiation counts per second value. Each known state therefore corresponds to an (x,y) location in the dataset, with the corresponding radiation count value comprising the value observed by the robot at the state. The transition function defined allows the robot to move deterministically to one of up to 6 neighbouring known states, which can be up to 0.4m away. Observations of the radiation level unknown state variable are taken directly from the dataset at the relevant locations. We specify an upper safety bound on the radiation level of 25 counts per second.

For all experiments below (including Section VI-C), the U-MDP transition functions are independent of the radiation level so observations are carried out only when updating the GP (as discussed at the end of Section V-A). We use PRISM [19] to solve MDPs, and GPflow [20] for our GP models. For the exploration algorithm, we set $p_{min} = 0.95$, relating to 95% confidence, and check candidate goal states in batches of N=3. A Matern 5/2 kernel is used for the exploration GP. The parameter weightings in the scoring function were $\gamma_1 = 1$ and $\gamma_2 = 0.25$. Setting $\gamma_1 = 1$ should result in the score function being roughly proportional to information gain per unit time, which is an intuitively reasonable factor to maximise for exploration.

We demonstrate the ability of SafeEst-MDP to safely learn the unknown feature over the environment by comparing GP posterior distributions between the GP produced by SafeEst-MDP and the full dataset ground truth GP. We carry out comparisons using the Kullback–Leibler (KL) divergence, a measure of the similarity of probability distributions. The KL divergence is calculated between the joint multivariate normal distributions that result from evaluating the two GP posterior distributions at all 554 states, giving a 554-dimensional multivariate normal distribution for each. Figure 4 shows

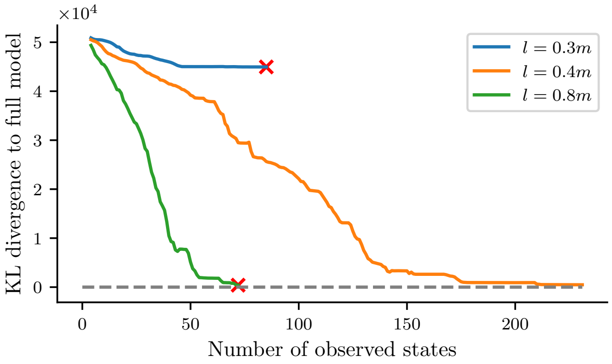

Fig. 4. KL divergence between the posterior distributions of the full dataset GP model and those from runs of the SafeEst-MDP algorithm with different GP lengthscales.

this KL divergence for the SafeEst-MDP GP model after a given number of explored states, for three different kernel lengthscales used. Our results show that a kernel lengthscale of $l = 0.3m$ causes the robot to be too cautious, as it terminates after 80 explored states believing that all remaining states are unsafe. On the other hand, a lengthscale of $l = 0.8m$ causes the robot to explore rapidly, but it enters an area of high radiation it is not expecting and fails its safety specification. Overall, this demonstrates that the lengthscale hyperparameter defines an upper limit on rate of change of radiation that the robot can expect.

The KL divergence converges towards zero for $l = 0.4m$ (showing that the robot is building a useful, predictive model) with only around 200 of the 554 available known states observed. It cannot completely converge to zero as, under the SafeEst-MDP algorithm, the robot is unable to sample from unsafe locations. A large prior distribution was set on the kernel variance hyperparameter to ensure that it did not optimise to zero while the robot sampled low-radiation points.

C. Multiple Radiation Source Map

Our second experiment poses the challenge of using the SafeEst-MDP exploration algorithm on a simulation of a map with multiple radiation sources. As radiation from multiple sources should be additive (ignoring low-level background radiation), we create the map as a linear combination of affine transformations of the ground truth GP in Section VI-A. This gives simulated maps of multiple sources differing in strength and location. The approach is justified by the behaviour of the GP fitted from the previous experiment, where the GP is able to accurately model a single radiation source in a way that prevents the robot breaking the safety specification.

Our algorithm is compared with the SafeMDP [1] exploration algorithm, on 8 different simulated maps. Each simulated map is a 5m x 5m grid world with deterministic transitions defined between states - this is to provide a fair comparison as SafeMDP can only handle deterministic state transitions. With state side length set to 0.2m, this gives a known state space of 625 states. The safety upper bound is set to 28 counts per second for this experiment.

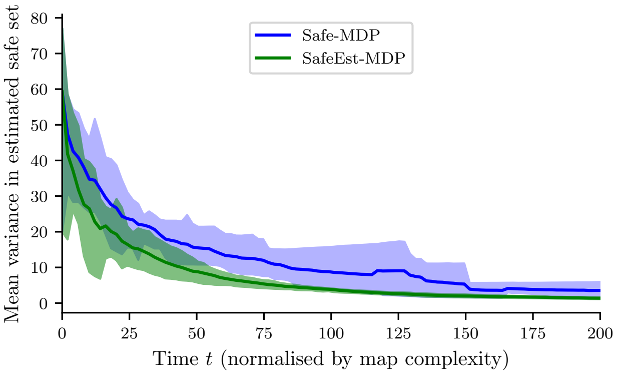

Fig. 5. Exploration progress vs time comparison for SafeEst-MDP and SafeMDP.

The results of this comparison are summarised in Figure 5, which shows the average predicted variance in the robot's current estimated safe set, at the specified time, for both algorithms. The exploration time has been normalised relative to the difficulty of the different maps, by scaling the time variable for each map to effectively make the mean completion time equal between maps. This has been done to give representative error bars on both algorithms' performance, accounting for the fact that each map has varying difficulty of exploration. On average, over all tested maps, our proposed algorithm is able to determine the radiation level at all reachable safe states (to within a defined standard deviation of 3 counts per second) in 52% of the time compared to the SafeMDP algorithm, while carrying out 56% fewer observations. It is able to determine the size of the ground-truth reachable safe set in a comparable amount of time.

Overall, these results show that SafeEst-MDP is more cost and sample efficient with respect to observations than similar approaches. This means it requires fewer, computationally expensive, GP update/optimisation and replanning sequences. The performance increase is largely due to SafeEst-MDP's ability to choose goal states that are multiple steps away from the currently explored set (SafeMDP will only choose goal states neighbouring the currently explored set), and the fact that it takes into account the expected cost to reach a goal state as well as the GP variance at that state.

VII. CONCLUSIONS

In this paper we presented SafeEst-MDP, a new MDP-based approach for safe exploration, and evaluated its effectiveness in a representative safety-constrained task using real-world data from a hazardous radioactive environment. We demonstrated that our method offers several advantages over existing safety approaches, both in terms of exploration performance and supporting more expressive safety constraints. Further illustration can be found in this paper's accompanying video.

Planning processing time is significantly increased for SafeEst-MDP compared to SafeMDP, due to the increased computational load of generating and solving Est-MDP models. However, this planning can be carried out comparatively less frequently due to its higher sample efficiency, and for many applications the heavy computation work can be performed remotely from the mobile robot.

There is scope for significant future work on applying our probabilistic exploration planning approach to other extensions of safe exploration, such as goal-driven exploration or more complex optimisation of the choice of efficient states to sample.

ACKNOWLEDGMENTS

This work has been funded by UK Research and Innovation and EPSRC through the Robotics and Artificial Intelligence for Nuclear (RAIN), and Offshore Robotics for Certification of Assets (ORCA) hubs [EP/R026084/1, EP/R026173/1].

REFERENCES

[1] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration in finite Markov decision processes with Gaussian processes," in *NeurIPS*, 2016.
[2] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
[3] S. Feyzabadi and S. Carpin, "Planning using hierarchical constrained Markov decision processes," *Aut. Rob.*, vol. 41, no. 8, 2017.
[4] B. Lacerda, F. Faruq, D. Parker, and N. Hawes, "Probabilistic planning with formal performance guarantees for mobile service robots," *IJRR*, vol. 38, no. 9, 2019.
[5] N. Gopalan, M. L. Littman, J. MacGlashan, S. Squire, S. Tellex, J. Winder, L. L. Wong, *et al.*, "Planning with abstract Markov decision processes," in *ICAPS*, 2017.
[6] T. M. Moldovan and P. Abbeel, "Safe exploration in Markov decision processes," in *ICML*, 2012.
[7] Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *ICML*, 2015.
[8] A. Wachi, Y. Sui, Y. Yue, and M. Ono, "Safe exploration and optimization of constrained MDPs using Gaussian processes," in *AAAI*, 2018.
[9] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration for interactive machine learning," in *NeurIPS*, 2019.
[10] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *AIJ*, vol. 101, no. 1-2, 1998.
[11] M. T. Spaan, T. S. Veiga, and P. U. Lima, "Decision-theoretic planning under uncertainty with information rewards for active cooperative perception," *JAAMAS*, vol. 29, no. 6, 2015.
[12] M. Lauri, J. Pajarinen, and J. Peters, "Information gathering in decentralized POMDPs by policy graph improvement," in *AAMAS*, 2019.
[13] R. Marchant, F. Ramos, and S. Sanner, "Sequential bayesian optimisation for spatial-temporal monitoring," in *UAI*, 2014.
[14] P. Morere, R. Marchant, and F. Ramos, "Sequential bayesian optimization as a pomdp for environment monitoring with UAVs," in *ICRA*, 2017.
[15] G. Flaspohler, V. Preston, A. P. M. Michel, Y. Girdhar, and N. Roy, "Information-guided robotic maximum seek-and-sample in partially observable continuous environments," *IEEE RA-L*, vol. 4, no. 4, 2019.
[16] J. Kemeny, J. Snell, and A. Knapp, *Denumerable Markov chains*, 2nd ed. Springer-Verlag, 1976.
[17] F. Teichteil-Königsbuch, "Stochastic safest and shortest path problems," in *AAAI*, 2012.
[18] M. Nancekievill, A. Jones, M. Joyce, B. Lennox, S. Watson, J. Katakura, K. Okumura, S. Kamada, M. Katoh, and K. Nishimura, "Development of a radiological characterization submersible ROV for use at Fukushima Daiichi," *IEEE Trans. Nucl. Sci.*, vol. 65, no. 9, 2018.
[19] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *CAV*, 2011.
[20] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman, "GPflow: A Gaussian process library using TensorFlow," *JMLR*, vol. 18, no. 40, 2017.