Bayesian Reinforcement Learning for Single-Episode Missions in Partially Unknown Environments

Matthew Budd **Paul Duckworth** **Nick Hawes** **Bruno Lacerda**
Oxford Robotics Institute, University of Oxford
{mbudd, pduckworth, nickh, bruno}@robots.ox.ac.uk

Abstract: We consider planning for mobile robots conducting missions in real-world domains where *a priori* unknown dynamics affect the robot's costs and transitions. We study single-episode missions where it is crucial that the robot appropriately trades off exploration and exploitation, such that the learning of the environment dynamics is *just enough* to effectively complete the mission. Thus, we propose modelling unknown dynamics using Gaussian processes, which provide a principled Bayesian framework for incorporating online observations made by the robot, and using them to predict the dynamics in unexplored areas. We then formulate the problem of mission planning in Markov decision processes under Gaussian process predictions as Bayesian model-based reinforcement learning. This allows us to employ solution techniques that plan more efficiently than previous Gaussian process planning methods are able to. We empirically evaluate the benefits of our formulation in an underwater autonomous vehicle navigation task and robot mission planning in a realistic simulation of a nuclear environment.

Keywords: Planning under Uncertainty, Gaussian Processes, Single-Episode Bayesian Reinforcement Learning

1 Introduction

Real-world mobile robots rarely have complete knowledge of their environment dynamics. When operating under uncertainty, they need to be able to incorporate their online *observations* of uncertain environment features into their plans. In this paper, we consider the single-episode setting where a robot must carry out a mission in an environment for which the dynamics are not fully known at deployment time. The mission is specified by a goal state(s) that the agent must eventually reach while minimising incurred cost under an environment feature that has *a priori unknown dynamics*. The robot's information gathering capabilities are limited, as the environment features are only observable at the robot's current state. An example would be a Geiger counter-equipped robot minimising its cumulative radiation exposure in an environment with an unknown radiation distribution. In this setting, it is infeasible to pre-plan for every possible environment that might be encountered. Moreover, even if it were possible to pre-train a standard reinforcement learning (RL) agent on every possible environment, it would lack the flexibility to *adapt* the robot's behaviour taking into account new observations it can collect during the mission.

To model continuous environment features, we follow previous works [1, 2, 3, 4] and use a Gaussian process (GP) [5] to predict unknown dynamics away from the agent's current location. GPs are well-suited for modelling spatio-temporal distributions by incorporating online measurements into the posterior distribution, along with a measure of predictive uncertainty. GP prior hyperparameters can be estimated from physical intuition or from a small dataset of similar environments. Similarly to [1, 2, 3, 4], our GP maintains a *belief* over the underlying dynamics of the environment. Rather than ensuring safe environment exploration or maximising information collected, we extend these works by formulating a unified Bayes-optimal framework for efficient online mission planning.

We therefore pose our task as online, model-based Bayesian RL (BRL) with a GP belief over the transition function. As all environmental uncertainty is then encapsulated within the transition function, we assume that the cost function is known given an instance of the environment dynamics. Continuing the previous example, the robot does not need to learn online that it incurs more damage

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

from higher levels of radiation. Our problem statement also assumes a discrete state space (except for unknown environment dynamics, which are continuous) and full observability of the current state.

The BRL formulation provides several advantages. First, it is known to optimally trade off exploration and exploitation [6], which is crucial in the single-mission setting we address: over-exploration may hinder mission performance or increase the risk of failure. Second, it allows us to intuitively encode the agent's transition function and local observability limitations, and reflects the existence of fixed (but *a priori* unknown) true underlying environment dynamics; third, it enables the use of efficient Monte-Carlo planning approaches developed for BRL.

Our contributions are to 1. formulate goal-driven planning in unknown environments as model-based BRL; 2. adapt the Bayes-adaptive Monte-Carlo planning (BAMCP) algorithm [6] for our GP-based goal-driven BRL problem formulation; and 3. use this formulation and algorithm to improve on previous GP planning approaches, both in expressibility and computational efficiency. In particular, we exploit techniques that allow us to efficiently sample possible environment dynamics from the GP to use during planning. To the best of our knowledge, we are the first to apply BRL with GP belief models to goal-driven planning in partially unknown environments.

2 Related Work

Non-myopic decision-making with an unknown transition function requires reasoning over possible observation (i.e. function evaluation) sequences. This task has been investigated from the perspective of several fields, including sequential Bayesian optimisation (BO) of unknown functions. An example BO objective could be to stay within a computational evaluation budget while improving a GP model of the unknown function. Some recent methods are able to pose multiple-step look-ahead in GPs as a single joint optimisation problem [7]. However, we focus on physical systems such as mobile robots which are required to physically move and observe any queried location: observations cannot be made in parallel or at freely specified locations. This implies reachability, observability and cost limitations that are not usually considered when using BO.

Considering some of these aspects, recent literature [8, 3] performs non-myopic "informative path planning" for environmental monitoring. Similar to our approach, these methods perform Monte-Carlo tree search (MCTS) [9] in belief space with a GP belief. However, they assume robot actions have deterministic outcomes, and do not consider the case where the unknown environment features can affect robot transition dynamics, as we do. This recent literature uses a partially observable Markov decision process (POMDP) [10] problem formulation, and plan with MCTS trees of computationally expensive GP-represented beliefs due to their BO objective. Our Bayes-adaptive MDP (BAMDP) [11] formulation more appropriately represents the existence of fixed but *a priori* unknown environment dynamics. In the context of a goal-based planning objective, the two formulations and solution methods are equivalent, as we demonstrate in this paper. However, we use model-based BRL techniques to *root sample* the GP environment belief. This avoids computationally expensive belief updates, enabling the construction of larger MCTS trees within the same computational budget.

Monte-Carlo tree search in GP-modelled unknown environments has also been carried out in [12], where the environment features affect only transition durations in a semi-MDP. A model-based BRL method with a Gaussian process dynamical model (GPDM) belief representation was presented in [13]. Their method must make restrictive maximum likelihood transition/observation assumptions for tractability, due to their unfocused Monte-Carlo planning algorithm. Alternatively, some RL techniques such as Gaussian Process temporal difference [14] use GPs to directly model an MDP value function. We argue that GP modelling the real-world environmental phenomena, rather than the value function, lets us provide physically principled and interpretable prior knowledge.

3 Preliminaries

Markov Decision Processes. We consider stochastic shortest path (SSP) MDP problems [15], as they are well-suited for specifying single episode goal-driven missions. An SSP MDP is defined as a tuple $\mathcal{M} = \langle S, s_0, A, T, C, G \rangle$, where S is a finite set of states; $s_0 \in S$ 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 $T(s, a, s') = p(s' \mid s, a)$; $C : S \times A \to \mathbb{R}_{\geq 0}$ is a cost function; and $G \subset S$ is a set of absorbing, zero-cost goal states. A history h of an MDP $\mathcal{M}$ is a state-action sequence $s_0 a_0 s_1 a_1 \cdots a_{t-1} s_t$ such that $T(s_i, a_i, s_{i+1}) > 0$

for all $i \in \{0, \cdots, t-1\}$. We denote the set of all histories of $\mathcal{M}$ as $\mathcal{H}^{\mathcal{M}}$. A stationary, deterministic policy is a mapping $\pi : S \to A$ that defines the action to take at each state. A policy is proper in a state s if it reaches a goal state $s_g \in G$ when starting from s with probability 1. In an SSP MDP there must exist a policy that is *proper* in all states, and all improper policies must incur infinite cost. Under these assumptions, there exists a cost-optimal proper policy [16].

Bayesian RL. In BRL, an agent uses Bayesian inference to maintain a posterior distribution, or *belief*, over the true dynamics of the underlying model given some prior distribution. For an MDP, either or both of the transition function T and cost function C could be a priori unknown. We focus on the case where only T is unknown. C can be assumed to be known in our setting, as it represents the known effect of a *given instance* of environment dynamics on the robot. For example, the time cost of travelling against given water current vectors can be calculated given the value of the vectors and the vehicle's known dynamics. Given a history $h = s_0 a_0 s_1 a_1 \cdots a_{t-1} s_t$, it is possible to generate the posterior belief over the transition function T given h. This can be carried out with successive applications of Bayes' rule $p(T \mid h_i) \propto p(h_i \mid T)p(T)$ from the initial history $h_0 = s_0$ up to the full history $h_t = h$. This allows for the definition of a Bayes-adaptive MDP (BAMDP) [11], which achieves Bayes optimality by adding histories to its state representation, and encoding uncertainty over T in its transition function. Let $\mathcal{M} = \langle S, s_0, A, T, C, G \rangle$ be an MDP with a prior belief $p(T)$ over the true transition function T. The corresponding BAMDP is an MDP $\mathcal{M}^+ = \langle S^+, s_0^+, A, T^+, C^+, G^+ \rangle$, where $S^+ = S \times \mathcal{H}^{\mathcal{M}}$; $s_0^+ = (s_0, h_0)$; $C^+((s,h),a) = C(s,a)$; $G^+ = \{(s,h) \in S^+ \mid s \in G\}$; and

$$T^+((s,h),a,(s',has')) = \int_T T(s,a,s')p(T \mid h)dT. \tag{1}$$

Although the state-history pairs in S^+ are redundant because the current state can be extracted from the history, we use the (s,h) notation for clarity as in [6]. A policy in a BAMDP is a mapping $\pi : S \times \mathcal{H}^{\mathcal{M}} \to A$. The optimal policy π^* minimises the expected cumulative cost to reach G^+, given the prior over T. This policy is stationary in S^+ but is history-dependent in the original MDP. π^* considers the posterior $p(T \mid h)$ and adapts its action selection to account for the conditional distribution of T given the observed h.

Gaussian Processes. A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution [5]. A GP regression is of the form $o(s) \sim \mathcal{GP}(m(s), k(s,s'))$, giving a probability distribution over functions fully specified by the mean $m(s)$ and kernel $k(s,s')$ functions. We can let $m(s) = 0$ without loss of generality. Given a dataset of n noisy observations $\mathcal{D} = \{(s_i, o(s_i) + \epsilon_i)\}_{i=1}^n$ for locations s_i and where $\epsilon_i \sim \mathcal{N}(0, \sigma_\eta^2)$ is Gaussian observation noise, GP regression predicts unknown environment feature values at all inputs s_*. The kernel function k is parameterised by hyperparameters $\boldsymbol{\theta}$. Given hyperparameter priors $p_0(\boldsymbol{\theta})$, their values are commonly optimised by maximising the log marginal likelihood for the model given the dataset. The resulting Gaussian posterior, conditioned on the observations $\mathbf{o} = [o(s_1) + \epsilon_1, \ldots, o(s_n) + \epsilon_n]^\mathsf{T}$, is a multivariate normal $p^{\mathcal{GP}}\big(o(s_*) \mid s_*, \mathcal{D})\big) \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$, where $\boldsymbol{\mu}_* = \mathbf{K}_*^\mathsf{T}(\mathbf{K}_n + \sigma_\eta^2 \mathbf{I})^{-1}\mathbf{o}$, and $\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^\mathsf{T}(\mathbf{K}_n + \sigma_\eta^2 \mathbf{I})^{-1}\mathbf{K}_*$.

The positive semi-definite kernel matrix $\mathbf{K}_n = [k(s,s')]_{s,s' \in \mathbf{s}_n}$, $\mathbf{K}_* = [k(s,s')]_{s \in \mathbf{s}_n, s' \in \mathbf{s}_*}$, $\mathbf{K}_{**} = [k(s,s')]_{s,s' \in \mathbf{s}_*}$, and $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix. We can sample functions from the GP posterior at a finite set of m points, incurring $O((n+m)^3)$ computational cost.

4 Approach

4.1 Problem Formulation

In order to clearly separate known system transition dynamics from the unknown environment dynamics, we represent the unknown environment and its effect on the agent as an *MDP with Unknown Feature Values (U-MDP)* [4].

Let S_k be a set of state features with discrete, known values (e.g. pose of a robot in a grid map) and S_e a set of state features with unknown values in $\mathbb{R}$ (e.g. the water current vector at a pose). Let $o : S_k \to S_e$ be an *a priori* unknown mapping that specifies the values $o(s_k) \in S_e$ observed at locations $s_k \in S_k$. An SSP U-MDP is a tuple $\mathcal{M}^o = \langle S^o, s_0^o, A^o, T^o, C^o, G^o \rangle$ where: $S^o = S_k \times S_e$;

s_0^o is the initial state $s_0^o = \big(s_{k,0}, o(s_{k,0})\big)$; A^o is a finite set of actions; T^o is the U-MDP transition function $T^o : (S_k \times S_e) \times A \times S_k \to [0, 1]$. As the state of the U-MDP is uniquely defined by the value of the known state feature(s) $s_k \in S_k$, the transition function of the U-MDP only represents the change in the known state feature(s); $C^o : S^o \times A \to \mathbb{R}_{\geq 0}$ is the cost function; and $G^o \subset S_k$ is the set of goal states, defined only across known value state features as $o(s_g)$ is not known for all $s_g \in G^o$. The problem addressed in this paper is formalised as an SSP U-MDP. The objective is to find a policy that minimises the expected cost to a reach state $(s_k, o(s_k)) \in S^o$ such that $s_k \in G^o$.

4.2 From U-MDPs to GP-BAMDPs

For notational simplicity, in the following we assume a single state feature with unknown values. The approach presented below can easily be extended to cases with more than one unknown value state feature, either using a multi-output GP [17] or multiple single-output GPs. The former assumes non-independent feature dynamics, where learning about one could improve predictions of another.

To estimate the unknown mapping o, we propose that the agent maintains a GP model created by adding a new observation of o at each timestep. Specifically, after observing history $h = (s_{k,0}, s_{e,0})a_0(s_{k,1}, s_{e,1})a_1 ... a_{t-1}(s_{k,t}, s_{e,t})$, we define $\mathcal{D}_h = \{(s_{k,i}, s_{e,i}) \mid i \in \{0, \dots, t\}\}$. Then, the GP model is denoted as $\mathcal{GP}_{\mathcal{D}_h}$ and the GP posterior over $o(s_k)$ is given by $p^{\mathcal{GP}}\big(s_e \mid s_k, \mathcal{D}_h\big)$. We assume that observations of o have negligible noise, which corresponds to full observability of the current state. Finally, note that by modelling unknown environment features with a GP we are implicitly making regularity and Lipschitz continuity modelling assumptions on the environment feature functions [5].

We now formulate SSP U-MDP as a BAMDP with GP belief over the transition function.

Proposition 1. *Let $\mathcal{M}^o = \langle S^o, s_0^o, A^o, T^o, C^o, G^o \rangle$ be a U-MDP. The GP-BAMDP for $\mathcal{M}^o$ is a BAMDP $\mathcal{M}^{o+} = \langle S^{o+}, s_0^{o+}, A^o, T^{o+}, C^{o+}, G^{o+} \rangle$ where the transition function incorporates the GP posterior. Formally, for $s = (s_k, s_e)$, $s' = (s'_k, s'_e) \in S^o$; $a \in A$; and $h = (s_{k,0}, s_{e,0})a_0(s_{k,1}, s_{e,1})a_1 ... a_{t-1}(s_{k,t}, s_{e,t}) \in \mathcal{H}^{\mathcal{M}^o}$ for $(s_{k,t}, s_{e,t}) = (s_k, s_e)$:*

$$T^{o+}((s, h), a, (s', has')) = \quad T^o\big((s_k, s_e), a, s'_k\big) \cdot p^{\mathcal{GP}}\big(s'_e \mid s'_k, \mathcal{D}_h\big). \tag{2}$$

Proof. We start by noting that the integral in Equation (1) is the product of two components. The first component is a specific possible transition function. As the single unknown component of T^o is the mapping o, the value of $T^o(s, a, s')$ given knowledge of o is defined according to the U-MDP transition function, ensuring that the unknown state feature dynamics are consistent with o:

$$T^o((s_k, s_e), a, (s'_k, s'_e) \mid o) = T^o\big((s_k, s_e), a, s'_k\big)\mathbb{I}[s'_e = o(s'_k)], \tag{3}$$

where $\mathbb{I}[.]$ is the indicator function. The second component is the posterior distribution over possible transition functions given a history h, which in our case is the GP posterior $p^{\mathcal{GP}}$:

$$p(o(s'_k) \mid h) = p^{\mathcal{GP}}\big(o(s'_k) \mid s'_k, \mathcal{D}_h\big). \tag{4}$$

As the unknown state feature dynamics are deterministic given o, the integral over T from Equation (1) only has value when the indicator function in Equation (3) is 1. This leads to Equation (2), where all uncertainty in T is captured in the GP posterior over o, and T^{o+} represents the combination of the U-MDP transition function and the GP belief over the values of the unknown state features. □

4.3 Solving U-MDPs with BAMCP

GP-BAMCP. Having framed the U-MDP mission planning problem as a BAMDP, we can exploit MCTS planning frameworks that were developed in this context. Specifically, we base our algorithm on BAMCP [6]. This algorithm plans in belief space[1], but builds search trees of action-observation histories rather than of belief states. The search tree consists of alternating state and action nodes and is constructed over the course of Monte-Carlo trials starting from the root node and sampling action outcomes from a generative model. BAMCP adapts the concept of *root sampling* from POMCP [18], where for each MCTS trial a transition function T is sampled from the root belief node and used throughout the trial. Actions are chosen inside the search tree using a *tree policy*, most commonly UCT [9]. New leaf nodes' values are estimated heuristically by continuing the trial trajectory from the leaf node using a *rollout policy*.

[1]The distribution over the transition function T in the case of BAMCP.

Algorithm 1 GP-BAMCP: PLAN

1: **procedure** PLAN($\mathcal{GP}_{\mathcal{D}_h}$, history h, start state (s_k, s_e), goal state set G^o)
2: **repeat**
3: $\hat{o} \leftarrow$ sample from $\mathcal{GP}_{\mathcal{D}_h}$
4: $\hat{T}((s_k, s_e), a, (s'_k, s'_e)) \leftarrow T^o((s_k, s_e), a, (s'_k, s'_e) \mid \hat{o})$
5: SIMULATE$((s_k, s_e), h, \hat{T}, G^o)$
6: **until** TIMEOUT( )
7: **return** $\arg\min_a Q(h, a)$
8: **end procedure**

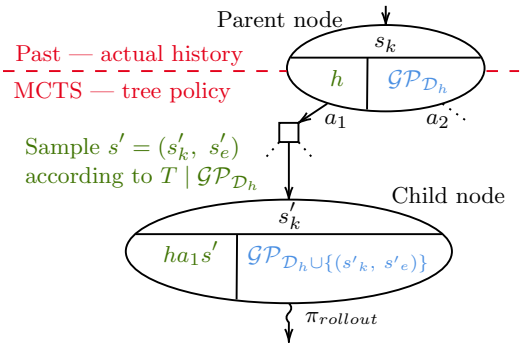


Figure 1: Example BAMDP MCTS search-tree, with search nodes as ellipses. Green shows search node generation/contents with a root sampling approach, and blue shows full belief planning.

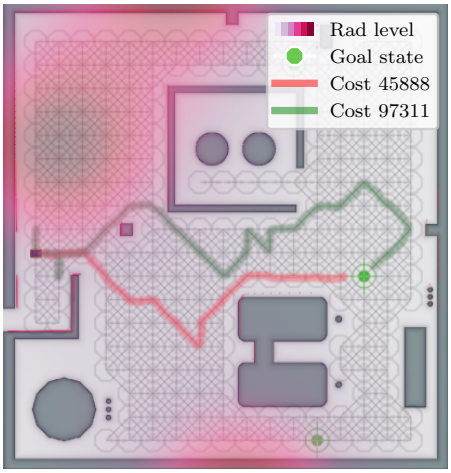
Our online MCTS planning algorithm, a modified version of BAMCP for GP beliefs, is described in Algorithm 1. To simplify the presentation, we assume that the rollout policy is able to reach the goal set with probability 1. Our algorithm replaces BAMCP's depth d and reward R parameters with the goal set G and costs C, respectively, to reflect the SSP mission setting of our work. In our case root sampling of T is performed by sampling from the current GP posterior, as described in lines 3 and 4. Concretely, we take a sample $\hat{o} \sim \mathcal{GP}_{\mathcal{D}_h}$ where $\hat{o} : S_k \to S_e$ is a possible mapping from known states to values of the unknown state features. Sampling $\hat{o}$ from the root belief node corresponds to sampling a possible environment that is consistent with the current GP environment model, i.e. trained only on real-world observations. Finally, in line 7, the agent greedily selects a single real-world action by running MCTS trials up to a computational budget.

Treatment of continuous s_e. Equation (2) represents a BAMDP with both discrete (known value) and continuous (unknown value) state features, and therefore also a combination of a discrete transition function, T^o, and a continuous transition function given by the GP belief over o. This presents challenges for MCTS methods, since the probability of transitioning to the same state twice is 0. To enable generalisation between histories and allow the search tree to reach depths greater than 1, our algorithm aggregates similar outcomes in s_e into the same child search node. Each search node has an associated "mean" value $\overline{s_e}$, where all outcomes $\|s'_e - \overline{s_e}\|_1 < \epsilon$ will be associated with that node. When two MCTS trials sample $s = (s_k, s_e)$ and $s' = (s_k, s'_e)$ from the same start state and action, if $\|s_e - \overline{s_e}\|_1 < \epsilon$ and $\|s'_e - \overline{s_e}\|_1 < \epsilon$, the two histories will be associated with the same child node. Note that this only associates histories to nodes and does not discretise the computation of reward or transition probabilities. The ϵ parameter therefore controls the search-tree branching factor, similarly to ϵ in [12] and the two branching factor parameters in progressive widening MCTS [19].

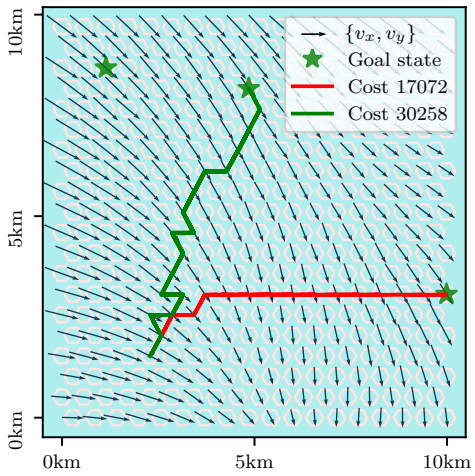
4.4 Theoretical Analysis

Equivalence to Partially Observable MDP Belief Planning Methods. Figure 1 depicts an example MCTS search-tree mid-mission. Several real actions have been taken in the environment, corresponding to the history h in the root node. When carrying out BAMCP root sampling, a new leaf node is added to the MCTS tree by appending the parent node history with a new state. This new state $s \sim T(s, a, \cdot)$ is sampled from the MDP transition function which was itself sampled from the root for this MCTS simulation, in lines 3 and 4 in Algorithm 1. This is depicted in green in Figure 1.

Several previous approaches [8, 3] that integrate GPs and MDPs for decision-making in uncertain environments use a partially observable MDP (POMDP) [10]. We call these *POMDP belief MCTS* approaches. These methods incorporate a GP environment belief into the POMDP state, and use MCTS to plan in belief space. Here, new belief state leaf nodes are generated using a *belief update* from the parent node. Belief update in this context is carried out by sampling a hypothesised data point $\tilde{o} \sim p^{\mathcal{GP}}(\cdot \mid s_k, \mathcal{D}_{parent})$ from the parent node GP belief posterior and augmenting the child belief node's GP dataset with this new point: $\mathcal{D}_{parent} \cup \{(s_k, \tilde{o})\}$. This is shown in blue in Figure 1,



(a) Example radiation domain visualisation, with a robot trajectory generated by GP-SSP-BAMCP (red) and by GP belief MCTS (green) algorithms.



(b) Example ocean currents domain visualisation, with an AUV trajectory generated by GP-SSP-BAMCP (red) and by GP belief MCTS (green) algorithms.

Figure 2: Experiment domains. Randomly selected multi-goal problem instances are shown.

where the GP in the child node can be explicitly generated by a belief update from the parent node GP. GP belief updates require adding a single new data point to the model, the complexity of which can be reduced from $O(N^3)$ to $O(N^2)$ where N is the number of data and sample points [17]. Even with this reduction, the belief update is still computationally expensive as the dataset of real and hypothesised observations grows. Furthermore, the belief updates are required sequentially, once per MCTS simulation to generate a new leaf node. This greatly slows the MCTS procedure.

In contrast, our method does not explicitly generate these hypothesised GPs and plans only using histories. The child node in Figure 1 still represents the same belief as the GP shown in blue, but only contains the history obtained using a root sampled transition function. We prove the history-GP equivalence, and hence the validity of root sampling in our setting, by showing that the probability of generating a history from the BAMDP is the same for root sampling as it is for maintaining and updating full GP belief at each belief node.

Proposition 2. *Let $\mathcal{P}_\pi^{h_t}(h_{t+\tau})$ be the probability of a history $h_{t+\tau}$ in the BAMDP, starting at history h_t under policy π, when carrying out individual GP belief updates at every stage; and $\tilde{\mathcal{P}}_\pi^{h_t}(h_{t+\tau})$ be the probability of $h_{t+\tau}$ when carrying out GP root sampling. Then, $\mathcal{P}_\pi^{h_t}(h_{t+\tau}) = \tilde{\mathcal{P}}_\pi^{h_t}(h_{t+\tau})$ for all policies π and all histories $h_{t+\tau}$.*

The proof is given in the appendix and is a direct adaptation of Lemma 1 in [20], accounting for the GP posterior over the transition function, and applies to a general stochastic policy $\pi : \mathcal{H}^{\mathcal{M}} \times A \to [0, 1]$.

Once any new real observations have been added to the root node GP, one can draw an arbitrary number of root samples to plan with at little additional computational cost. This means we can run more trials with the same computational budget, thus building a larger MCTS search tree, as we demonstrate in Section 5. Finally, we note that, due to the BO setting of [8, 3], where the objective is directly related to the uncertainty in the function being predicted by the GP, it is not enough to ensure the same distribution over histories. Thus, they must maintain full GP beliefs in their search nodes.

5 Experiments

Domains Description. We experimentally evaluate the proposed method in two simulated domains.

1) Radiation domain: a robot must navigate to a goal location (single-goal variant), or one of three goal locations (multi-goal variant), in a 20m $\times$ 20m reactor room with an unknown distribution of radiation level, while minimising its cumulative exposure. The GP model is log-warped [21] to constrain the predictions to be strictly positive and better model order-of-magnitude variation

in radiation level caused by $1/r^2$ "solid angle" radiation physics. Goal locations and radiation distributions are randomly generated and described in full in the appendix. The map (Figure 2a) is discretised into an 8-connected grid with side length 1.0m. The robot pose therefore comprises the known value U-MDP state features: S_k is a finite set of (x, y) locations $\{x, y\} \subseteq S_k$. The radiation level is a single unknown value state feature $S_e = \mathbb{R}_{\geq 0}$ where $rad\_exp \in S_e$ is the level at a location. The reactor room world is from [22] and is used with Gazebo [23] and ROS [24].

2) Ocean currents domain: an autonomous underwater vehicle (AUV) must navigate underwater to one of a set of two to three goal locations (multi-goal variant) or a single goal location (single-goal variant) across a 10km $\times$ 10km map, under the influence of currents. These are drawn from a real-world ocean current dataset and modelled online by a multi-output coregionalised GP [25]. The AUV is simulated by a kinematics, guidance, navigation and control (GNC) model of a small AUV. The robot pose comprises the known value U-MDP state features: S_k is a finite set of (x, y) locations $\{x, y\} \subseteq S_k$ on a 18×20 hexagonal grid of states, giving approximately 500m spacing between states. An example state grid with ground-truth currents is shown in Figure 2b. The unknown value state features $S_e = \mathbb{R}^2$ represent the current x and y velocities $\{v_x, v_y\} \in S_e$. The cost function encodes the expected traversal time between states given the AUV's water-relative velocity and the current vector values. Additional domain details and parameters are given in the appendix.

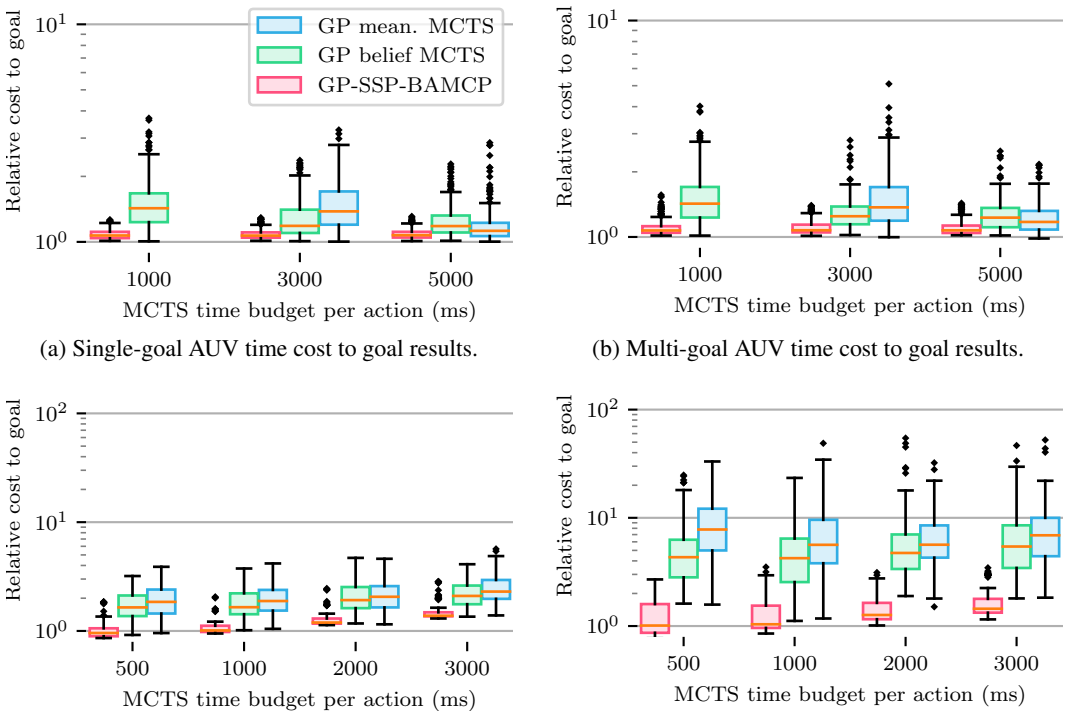
Algorithms. Our online BAMCP variant as described in Algorithm 1 is evaluated against two other GP belief planning algorithm baselines. The performance of sampling-based methods is dependent on the assigned compute budget to select each action, therefore we vary compute budget in each experiment. The *GP mean belief MCTS* algorithm represents a full GP belief planning approach that makes maximum likelihood assumptions. This is similar to [13], but with a GP rather than GPDM model and replacing the Monte Carlo action selection search with MCTS due to the complexity of the search problem. The *GP belief MCTS* algorithm represents the other full GP belief planning approaches [8, 3] which plan with GP beliefs inside the MCTS search tree, but sample a fixed environment dynamics instance for the MCTS rollout. This avoids carrying out belief updates during the rollout, leading to a speed up in MCTS trials per second. For all algorithms the rollout policy is to choose the action that minimises the travel distance from the next state to the closest goal state.

Results. The plots in Figure 3 are from multiple randomly generated problem instances: 10 for the currents domain, and 5 for the radiation domain. Each algorithm/MCTS time budget combination is given 25 repeats. Values are normalised by the expected minimum achievable cost for the corresponding randomly generated problem, calculated using an exact method and full knowledge of the environment. Due to stochasticity in the simulated robot and environment, it is possible for some runs to achieve costs below this value. Experiments are run on a 3.20GHz i7 / 64GB RAM machine.

In the ocean currents domain, Figures 3a and 3b, GP-SSP-BAMCP significantly outperforms the other algorithms. It is capable of achieving close to optimal cost with only 1 second of computation budget. Given an increased computational budget, each algorithm achieves lower cost-to-goal mean and variance. At lower computational budgets, the GP belief MCTS algorithm consistently outperforms the GP mean belief MCTS algorithm, which must carry out a belief update for each step it takes towards a goal state during a rollout. The 1000ms / GP mean belief MCTS combination is not shown, as the algorithm is not capable of building a meaningful plan within that time limit.

GP-SSP-BAMCP also significantly outperforms the baselines in the radiation domain, shown in Figures 3c and 3d. In this case, note that the increase in cost-to-goal with additional computation time is caused by the robot spending more time planning while being exposed to radiation, while the normalisation denominator is kept constant. For all methods additional computation time increases plan quality, but this can be offset by the additional stationary time spent planning. The 3000ms time budget case corresponds to the robot spending $\sim 50\%$ of its runtime planning rather than moving.

As a summary, GP-SSP-BAMCP's higher performance is due to its ability to carry out far more MCTS trials than the two full belief planning methods. On average across all sampled problem scenarios, GP-SSP-BAMCP carries out $\sim 100\times$ more MCTS trials per second than GP mean belief MCTS, and $\sim 40\times$ more than GP belief MCTS.



(a) Single-goal AUV time cost to goal results.

(b) Multi-goal AUV time cost to goal results.

(c) Single-goal radiation exposure cost to goal results.

(d) Multi-goal radiation exposure cost to goal results.

Figure 3: Single- and multi-goal experiment results. Plots consist of 25 simulated experiments for each algorithm/MCTS time budget combination in each randomly generated problem instance for that domain.

6 Conclusion

We have proposed a unified Bayesian RL framework for single-mission robot planning in GP-modelled unknown environments, and demonstrated that we are able to plan more effectively in representative real-world environments than previous approaches are able to. One potential avenue for future work is to apply progressive widening [19] or function approximation [20] techniques to our BAMCP search tree, to determine whether these produce better plans given the continuous GP-BAMDP state-space. Relaxing the assumption of negligible measurement noise would require introducing partial observability alongside transition function uncertainty. The reformulation would transform the BAMDP into a more complex Bayes-adaptive POMDP [26] with continuous GP belief. This would allow us to address settings with very high localisation uncertainty and sensor noise.

Limitations. As with any method using exact GP regression, there is a limit to the size of environment that can practically be modelled. Although root sampling limits the computational cost of GP planning, exact GP regression scales with $O((n+m)^3)$ where n is the number of data points and m is the number of sample points. In future work we aim to replace exact GP predictions with approximate GP posterior sampling [27] to reduce the GP computational burden and improve scalability.

As our evaluation is in simulation there may exist a sim-to-real transfer gap when using the method with a real robot. This concern should be partly alleviated by our use of realistic Gazebo [23] simulation and use of real-world currents data with a complex kinematic AUV simulation. The proposed method is also a higher-level planning approach, meaning that the gap should be less wide than with low-level control methods, which are more sensitive to small sim-to-real changes.

Finally, for some environments or problem settings there is less inherent value in carrying out online planning. For example, with only a single goal state and few feasible routes to that state, the best approach for the radiation domain may be to navigate as quickly as possible to that state without incurring the radiation exposure costs of stopping to plan.

Acknowledgments

This work received EPSRC funding via the "From Sensing to Collaboration" programme grant [EP/V000748/1]. Matthew Budd was supported by an Amazon Web Services Lighthouse scholarship. Paul Duckworth was supported by the Cancer Research UK Radnet Oxford Centre grant (CRUK A28736).

References

[1] M. Turchetta, F. Berkenkamp, and A. Krause. Safe exploration in finite Markov decision processes with Gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

[2] A. Wachi, Y. Sui, Y. Yue, and M. Ono. Safe exploration and optimization of constrained MDPs using Gaussian processes. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[3] 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 Robotics and Automation Letters (IEEE RA-L)*, 4(4), 2019.

[4] M. Budd, B. Lacerda, P. Duckworth, A. West, B. Lennox, and N. Hawes. Markov decision processes with unknown state feature values for safe exploration using gaussian processes. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7344–7350. IEEE, 2020.

[5] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[6] A. Guez, D. Silver, and P. Dayan. Scalable and efficient Bayes-adaptive reinforcement learning based on Monte-Carlo tree search. *Journal of Artificial Intelligence Research (JAIR)*, 48, 2013.

[7] S. Jiang, D. Jiang, M. Balandat, B. Karrer, J. Gardner, and R. Garnett. Efficient nonmyopic Bayesian optimization via one-shot multi-step trees. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[8] P. Morere, R. Marchant, and F. Ramos. Sequential Bayesian optimization as a POMDP for environment monitoring with UAVs. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.

[9] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning (ECML)*, pages 282–293. Springer, 2006.

[10] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[11] M. O. Duff. *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes.* PhD thesis, Dept. of Computer Science, University of Massachusetts Amherst, 2003.

[12] P. Duckworth, B. Lacerda, and N. Hawes. Time-bounded mission planning in time-varying domains with semi-MDPs and Gaussian processes. In *Conference on Robot Learning (CoRL)*. Journal of Machine Learning Research, 2021.

[13] P. Dallaire, C. Besse, S. Ross, and B. Chaib-draa. Bayesian reinforcement learning in continuous POMDPs with Gaussian processes. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2604–2609. IEEE, 2009.

[14] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 201–208, 2005.

[15] D. P. Bertsekas and J. N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3), 1991.

[16] Mausam and A. Kolobov. *Planning with Markov decision processes: An AI perspective*. Morgan & Claypool Publishers, 2012.

[17] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *International Conference on Information Processing in Sensor Networks (ISPN)*. IEEE, 2008.

[18] D. Silver and J. Veness. Monte-Carlo planning in large POMDPs. *Advances in Neural Information Processing Systems (NeurIPS)*, 23, 2010.

[19] A. Couëtoux, J.-B. Hoock, N. Sokolovska, O. Teytaud, and N. Bonnard. Continuous upper confidence trees. In *International Conference on Learning and Intelligent Optimization*, pages 433–445. Springer, 2011.

[20] A. Guez, N. Heess, D. Silver, and P. Dayan. Bayes-adaptive simulation-based search with value function approximation. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014.

[21] E. Snelson, Z. Ghahramani, and C. Rasmussen. Warped Gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*, 16, 2003.

[22] T. Wright, A. West, M. Licata, N. Hawes, and B. Lennox. Simulating ionising radiation in Gazebo for robotic nuclear inspection challenges. *Robotics*, 10(3):86, 2021.

[23] N. Koenig and A. Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154. IEEE, 2004.

[24] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.

[25] C. Williams, E. V. Bonilla, and K. M. Chai. Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems (NeurIPS)*, 20, 2007.

[26] S. Katt, F. A. Oliehoek, and C. Amato. Learning in POMDPs with Monte Carlo tree search. In *International Conference on Machine Learning (ICML)*, pages 1819–1827. PMLR, 2017.

[27] J. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning (ICML)*. PMLR, 2020.

Appendix

7 List of acronyms and abbreviations

AUV	Autonomous underwater vehicle
BAMCP	Bayes-adaptive Monte-Carlo planning
BAMDP	Bayes-adaptive Markov decision process
BO	Bayesian optimisation
BRL	Bayesian reinforcement learning
GNC	Guidance, navigation, control
GP	Gaussian process
GPDM	Gaussian process dynamical model
MCTS	Monte-Carlo tree search
POMDP	Partially observable Markov decision process
ROS	Robot operating system
SSP	Stochastic shortest path
U-MDP	MDP with unknown feature values

8 Proof of Proposition 2

Proof. With individual belief updates at every stage, the history density is (shortening $\mathcal{P}_\pi^{h_t}(h_{t+\tau})$ to $\mathcal{P}_\pi^{h_t}$ and $\tilde{\mathcal{P}}_\pi^{h_t}(h_{t+\tau})$ to $\tilde{\mathcal{P}}_\pi^{h_t}$):

$$\begin{aligned}
\mathcal{P}_\pi^{h_t} &= p(a_t s_{t+1} a_{t+1} ... s_{t+\tau} \mid h_t, \pi) \\
&= p(a_t \mid h_t, \pi) p(s_{t+1} \mid h_t, \pi, a_t) p(a_{t+1} \mid h_{t+1}, \pi) ... p(s_{t+\tau} \mid h_{t+\tau-1}, a_{t+\tau}, \pi) && (5) \\
&= \prod_{t \le t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \le t+\tau} p(s_{t'} \mid h_{t'-1}, a_{t'-1}) && (6) \\
&= \prod_{t \le t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \le t+\tau} \int_T T(s_{t'-1}, a_{t'-1}, s_{t'}) p(T \mid h_{t'-1}) dT, && (7)
\end{aligned}$$

where $s_t = (s_k, s_e)$.

Given the definition of T^+ in (2), and the fact that a history h_t uniquely specifies a GP $\mathcal{GP}_{\mathcal{D}_h}$ of observations up to time t:

$$\mathcal{P}_\pi^{h_t} = \prod_{t \le t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \le t+\tau} \left[T^o\big(s_{t'-1}, a, s_{k,t'}\big) p^{\mathcal{GP}}\big(s_{e,t'} \mid s_{k,t'}, \mathcal{D}_{t'-1}\big) \right]. \tag{8}$$

The GP posterior $p^{\mathcal{GP}}\big(s_{e,t'} \mid s_{k,t'}, \mathcal{D}_{t'-1}\big)$ is a multivariate normal distribution (MVN). A GP belief update with a noise-free sampled observation is performed by conditioning the posterior MVN on the sampled value (for compactness we remove s_k from the MVN probability density function $p^{\mathcal{GP}}$):

$$\begin{aligned}
\mathcal{P}_\pi^{h_t} = \prod_{t \le t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \le t+\tau} T^o\big(s_{t'-1}, a, s_{k,t'}\big) \cdot \\
\left[p^{\mathcal{GP}}\big(s_{e,t+1} \mid \mathcal{D}_t\big) \cdot \prod_{\substack{t+1 < t' \\ \le t+\tau}} p\big(s_{e,t'} \mid s_{e,t'-1}, \ldots, s_{e,t+1}\big) \right]. && (9)
\end{aligned}$$

The repeated belief update product in the square brackets in (9) can be recognised as being equivalent (via the chain rule for probability) as being equivalent to the joint distribution across all values of $s_{e,t'}$:

$$\left[\cdots \right] = p^{\mathcal{GP}}\big(s_{e,t+1}, ..., s_{e,t+\tau} \mid \mathcal{D}_t\big). \tag{10}$$

Therefore the rollout distribution is identical between individual belief updates and root sampling:

$$\mathcal{P}_{\pi}^{h_t} = \prod_{t \leq t' < t+\tau} \pi(h_{t'}, a_{t'}) \cdot \prod_{t < t' \leq t+\tau} T^{o}\big(s_{t'-1}, a, s_{k,t'}\big) \cdot p^{\mathcal{GP}}\big(s_{e,t+1}, ..., s_{e,t+\tau} \mid \mathcal{D}_t\big) = \tilde{\mathcal{P}}_{\pi}^{h_t}. \tag{11}$$

Given the rollout distribution equivalence, search tree node statistics for both methods will converge to the same values in expectation. □

9 Radiation Domain Details

9.1 Radiation simulation

Radiation is simulated using $1/r^2$ "solid angle" radiation physics. Radiation sources $\{(s_i, \mathbf{x}_i), ...\}_{i=0}^{n}$ have strength s_i and pose $\mathbf{x}_i$. Source strength is the exposure value at a distance of 1m. The radiation exposure $\lambda(\mathbf{x})$ at robot pose $\mathbf{x}$ from these radiation sources is then

$$\lambda(\mathbf{x}) = \sum_{i}^{n} \frac{s_i^{src}}{\|\mathbf{x} - \mathbf{x}_i^{src}\|^2}. \tag{12}$$

9.2 Domain Details

A problem instance consists of the following components: **a)** a randomly generated distribution of radiation sources in the environment (below), **b)** the start location in the grid map, sampled from a uniform distribution across the map, and **c)** 3 goal states, also uniformly sampled. Sampled problem instances are discarded when they result in trivial solutions (e.g. due to the start and goal locations being too close) or where one goal is significantly closer to the start location than the other sampled goals.

Random radiation fields are generated in both of the following ways:

1. Random point-source distribution: insert between 5 to 20 radiation sources (uniform random sampling), with randomly sampled z position values $z \in \{1.0, 1.5, 2.5\}$ and randomly sampled strengths $s \in \{1000, 2000, 5000, 10000\}$. x and y position values are sampled uniformly within the bounds of the map ± 2.0m.

2. Randomly generated Gaussian random field distribution: evenly cover the map with radiation sources at $z = 1.0$ and draw their log strengths from a Gaussian random field. The Gaussian random field is generated with a radial basis function kernel, using uniformly sampled lengthscale hyperparameter $l \in \{3.0, 5.0, 7.0\}$ and variance hyperparameter $\sigma \in \{60, 75, 90\}$.

9.3 Gazebo simulation

A visualisation of the reactor room world in Gazebo is shown in Figure 4. The simulated robot is a Clearpath Jackal[2], using standard Gazebo lidar and odometry sensor simulation and standard ROS components such as AMCL localisation[3].

[2] https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/

[3] http://wiki.ros.org/amcl

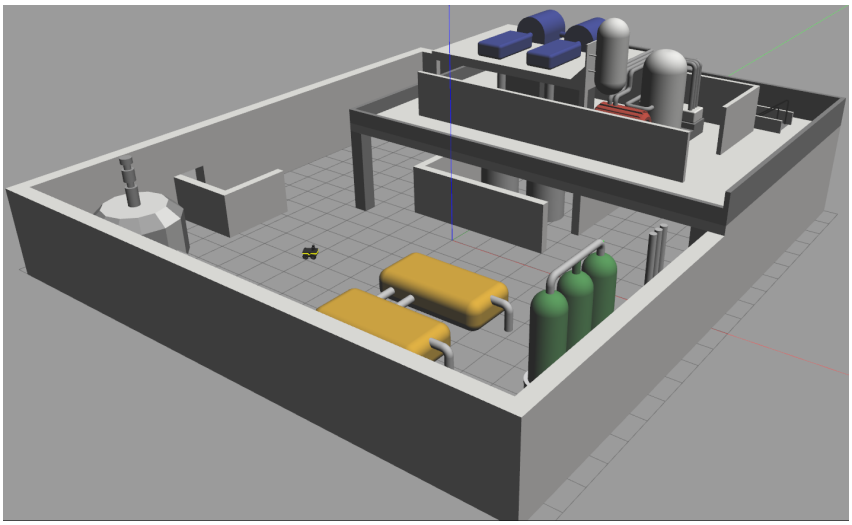


Figure 4: Visualisation of the reactor room Gazebo world in the middle of a simulated mission.

9.4 Algorithm parameters

The UCT exploration constant was dynamically set to equal the value of the decision node multiplied by 1.414. The guided rollout policy attempts to minimise the L2 distance from the goal state at the next state:

$$\pi_{rollout}(s_k) = \underset{a \in A(s_k)}{\arg\min}\, \text{dist}\,(s'_k, s_g)\, \forall s_g \in G, \tag{13}$$

where $A(s_k)$ are the enabled actions at a state and dist is a function that returns the shortest grid map path distance between two states in the grid map. The value of the ϵ parameter was 1.0 in the log GP space, meaning that MCTS search nodes covers increasingly large ranges of continuous value with increasing radiation level.

9.5 U-MDP Cost and Transition Structures

Transitions between grid map states are assumed to be deterministic as robot navigation does not fail in this easy-to-localise environment.

The state space $S^o = S_k \times S_e$ where $\{x, y\} \subseteq S_k$ and $rad\_exp \in S_e$. The set of actions $A^o = \{\text{left}, \text{left-up}, \text{left}, \text{up}, \text{right-up}, \text{right}, \text{right-down}, \text{down}, \text{left-down}\}$.

The cost structure is

$$C^o\big((s_k, s_e), a\big) = rad\_exp \cdot (\text{mcts-time} + d_a / v_{robot})\,, \tag{14}$$

where mcts-time is the MCTS time budget allocated to the algorithm, d_a is the travel distance associated with the action (diagonal actions are 1.414 times further), and v_{robot} is the manually estimated average velocity of the simulated robot, taken to be $= 0.3\text{ms}^{-1}$.

9.6 GP model

The GP is a single-output GP that models $o : \mathbb{R}^2 \rightarrow \text{Dist}(\mathbb{R})$ where the GP input is $\{x, y\} \subseteq S_k$. The GP is trained on log radiation measurements and predicts log radiation levels.

The GP kernel is a combination of a bias kernel and a radial basis function kernel:

$$k(s_k, s'_k) = \sigma^2 \exp\left(-\frac{\|s_k - s'_k\|^2}{2l^2}\right). \tag{15}$$

The radial basis function kernel hyperparameters are assigned the following uninformative Gamma distribution priors:

GP hyperparameter	Prior
Lengthscale l	$\Gamma(1, 0.5)$
Variance σ	$\Gamma(1, 4)$

This corresponds roughly to an expected lengthscale of 2m (standard deviation 2m) and expected log magnitude variance of 0.5 (standard deviation 0.5). At the beginning of each experiment the agent is provided with observations at the start state and two immediately neighbouring states as prior knowledge.

10 Underwater Currents Experiment

10.1 Domain Details

A problem instance consists of the following components: **a)** a 10km $\times$ 10km current field drawn from the real-world currents dataset in the same manner as [12], sampled from a fixed set of 12 fields where there is some variation in current across the field (rather than e.g. consistent current in one direction), **b)** the AUV start location, sampled from a uniform distribution across the field, and **c)** between 1 and 3 (with uniform probability) goal states, also uniformly sampled. Sampled problem instances are discarded when they result in trivial solutions (e.g. due to the start and goal locations being too close) or where one goal is significantly closer to the start location than the other sampled goals.

This experiment makes use of E.U. Copernicus Marine Service Information[4]. This is a dataset of north/east ocean current vectors on a 1500m spaced grid. To allow sampling of ground truth values at locations other than the grid locations, interpolation of the dataset using a spatio-temporal GP is carried out in the same manner as [12]. The currents experiment, the dataset used covers the region from approximately 47 to 62 latitude and -12 to 5 longitude. The dataset was originally collected on May 1 2020.

10.2 Kinematic GNC simulation

The vehicle's movement is determined by a kinematic calculation given of the vehicle's yaw, pitch and velocity control demands, process noise and the currents acting on the vehicle. Currents are drawn from the currents dataset at the vehicle's simulated true position. Underwater localisation is via an underwater acoustic beacon-aided extended Kalman filter. The vehicle uses this acoustic time-of-flight position feedback to navigate to the target location of the action selected by the MCTS planner. An example detailed run through the kinematic GNC simulator is shown in Figure 5.

10.3 Algorithm parameters

The UCT exploration constant was dynamically set to equal the value of the decision node multiplied by 1.414. The guided rollout policy attempts to minimise the L2 distance from the goal state at the next state:

$$\pi_{rollout}(s_k) = \underset{a \in A(s_k)}{\arg\min} \|s'_k - s_g\|_2 \forall s_g \in G, \tag{16}$$

where $A(s_k)$ are the enabled actions at a state. The value of the ϵ parameter was 0.1.

10.4 U-MDP Cost and Transition Structures

Transitions between grid map states are assumed to be deterministic as the AUV will always eventually reach a specified goal. The state space $S^o = S_k \times S_e$ where $\{x, y\} \subseteq S_k$ and $\{v_x, v_y\} \in S_e$. The set of actions $A^o = \{0°, 60°, 120°, 180°, 240°, 300°, \}$ corresponds to the direction the vehicle wishes to travel to the hex grid state in that direction. The vehicle travels against the current such that its net movement is in the action-specified direction.

The cost is the time taken to carry out the transition, given a constant vehicle speed $v = 0.6\text{m s}^{-1}$, the currents acting on the vehicle, and the requirement from the navigation controller that the net

[4] Available: https://resources.marine.copernicus.eu/?option=com_csw&view=details&product_id=NORTHWESTSHELF_ANALYSIS_FORECAST_PHY_004_013. . Accessed 2021-08.

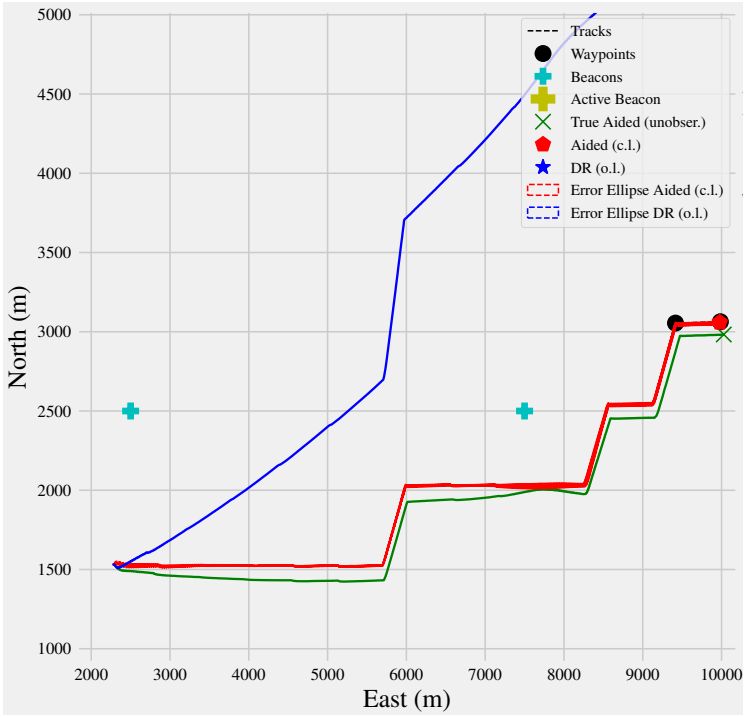


Figure 5: Example run through kinematics / GNC simulator. The green line is the vehicle's true path, the red line is the demanded path the vehicle has attempted to follow (where the demanded path is a result of an MCTS action selection at each grid MDP state), and the blue line is the "dead-reckoned/odometry-only" position estimate for the AUV. This is where the vehicle would localise to without any external EKF acoustic time-of-flight feedback.

direction of travel is in the action-specified direction. Given the chosen action this is solved to find the net velocity v_a^{eff} and the direction the vehicle must steer against the current at the state.

$$C^o\big((s_k, s_e), a\big) = \frac{d_s}{v_a^{eff}}, \tag{17}$$

where d_s is the distance between states in the hexagonal grid.

10.5 GP model

The GP is a vector-output coregionalised GP that models $o : \mathbb{R}^2 \rightarrow \text{Dist}(\mathbb{R}^2)$ where the GP input is $\{x, y\} \subseteq S_k$.

The GP kernel is the sum of a bias kernel and a radial basis function kernel. The RBF hyperparameters are assigned the following relatively broad Gamma distribution priors based on sensible current values seen in the whole currents dataset, and distribution of optimised lengthscale parameters seen when GPs are trained on a small number of random subsets of the dataset:

GP hyperparameter	Prior
Lengthscale l	$\Gamma(a = 49.0, b = 0.014)$
Variance σ	$\Gamma(a = 1.0, b = 4.0)$

This corresponds roughly to an expected lengthscale of 3500m (standard deviation 500m) and expected current magnitude variance of 0.25 (standard deviation 0.0625). At the beginning of each experiment the agent is provided with observations at the start state and two immediately neighbouring states as prior knowledge.