Time-Bounded Mission Planning in Time-Varying Domains with Semi-MDPs and Gaussian Processes

Paul Duckworth
Oxford Robotics Institute
University of Oxford
pduckworth@robots.ox.ac.uk

Bruno Lacerda
Oxford Robotics Institute
University of Oxford
bruno@robots.ox.ac.uk

Nick Hawes
Oxford Robotics Institute
University of Oxford
nickh@robots.ox.ac.uk

Abstract: Uncertain, time-varying dynamic environments are ubiquitous in real world robotics. We propose an online planning framework to address time-bounded missions under time-varying dynamics, where those dynamics affect the duration and outcome of actions. We pose such problems as semi-Markov decision processes, where actions have a duration distributed according to an *a priori* unknown time-varying function. Our approach maintains a belief over this function, and time is propagated through a discrete search tree that efficiently maintains a subset of reachable states. We show improved mission performance on a marine vehicle simulator acting under real-world spatio-temporal ocean currents, and demonstrate the ability to solve co-safe linear temporal logic problems, which are more complex than the reachability problems tackled in previous approaches.

Keywords: Representing Uncertainty, Semi-MDPs, Gaussian Processes, MCTS

1 Introduction

Uncertain, time-varying dynamics pose a challenging planning problem in robotics. Such dynamics are present in applications including marine surface vehicles planning to navigate complex ocean currents [1], and mobile robots planning service tasks through the ebb and flow of human activity [2]. In order to efficiently satisfy time-bounded goals (i.e. goals with deadlines), robots operating in such environments should exploit predictions of the environmental dynamics and make observations to reduce their uncertainty. To address this problem we introduce an online planning framework for time-bounded reachability problems in environments with non-stationary, unknown or uncertain dynamics, where those dynamics affect the duration and outcome of actions.

We formulate the above problem as a semi-Markov decision process (SMDP) [3], an extension of MDPs [4] that considers action durations. We posit that each action duration can be *a priori* unknown, continuous, and drawn from a non-stationary distribution. Thus, we introduce a *time-varying SMDP* (TV-SMDP) where a time-dependent transition model is built using Gaussian process (GP) regression [5] over latent environment variables, with observations gathered online. We tackle time-bounded reachability problems for co-safe linear temporal logic (csLTL) [6], with the objective of maximising the probability of satisfaction within a deadline. csLTL missions are a challenge in time-varying environments since they can require revisiting states, so a solution must consider that states may have different dynamics at different times.

We generate policies using Monte Carlo tree search (MCTS) [7], sampling action durations, and propagating time through the search tree. We incorporate the GP predictions of the environment into a time-dependent transition model to estimate the arrival time at future states. This eliminates problems with existing methods for time-varying MDPs that only estimate mean first passage times [1, 8]. Our search tree represents a reachable abstraction of the full state space, whilst appropriately propagating uncertainty. This allows our method to scale better than exact approaches [9, 10].

As a motivating example, consider the hydrothermal vent monitoring task shown in Figure 1 [11]. An autonomous underwater vehicle is required to observe three geological features within a time bound T. Ocean current vectors are represented by blue arrows. At time $\frac{T}{2}$ the south easterly currents switch 180° to north westerly. In order to maximise the probability of completing the mission within

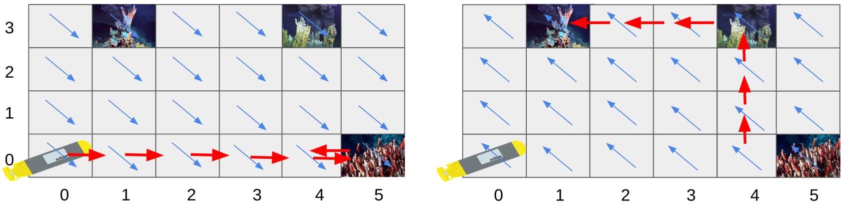

Figure 1: An underwater inspection mission under time-varying dynamics. Hydrothermal vent image credit to National Geographic.

the time-bound, the goal states must be visited in an order that exploits the environment's dynamics. Red arrows demonstrate the optimal path.

Our contributions are to: i) formulate time-dependent decision making problems into an SMDP framework where action duration is *a priori* unknown and affected by environment dynamics; ii) incorporate a time-dependent transition model that appropriately manages environment uncertainty on transitions; iii) satisfy time-bounded csLTL missions in time-varying environments. To the best of our knowledge, this is the first paper to consider temporal logic missions in time-varying domains.

2 Related Work

There are many applications for robots which are aware of partially known or time-varying environment dynamics, e.g. informative environment sampling [12, 13, 14, 15] and safe exploration [16, 17, 18, 19]. Time-varying MDPs have been previously solved by propagating the estimated time of first arrival to future states [1, 8, 10]. However, considering only the first arrival time at future states is not suitable for missions that require re-visiting states, e.g. missions specified in csLTL.

Semi-MDPs [20] (SMDPs) have previously been used to model scenarios where states have stochastic waiting times. Several objectives for SMDPs have been considered, e.g. time-bounded reachability or long-run average rewards [21]. Time-bounded csLTL has been addressed using timed MDPs [22], a model akin to SMDPs but where time is discretised in advance. None of these works considered time-varying or partially known dynamics. We demonstrate that SMDPs, augmented with a belief over a time-varying transition model, can be used for time-bounded csLTL mission planning in environments where unknown dynamics affect state transitions. Each action duration is propagated through a search tree using MCTS [7] to efficiently search the reachable state space.

Previous work has addressed planning in sequential Bayesian optimisation [15, 23] and maximum seek and sample [12] frameworks. In these works planning under an unknown reward function is framed as a partially observable MDP (POMDP). The unknown function in these works can be extended to be time-varying, but it only influences the reward function, not the transition dynamics. Other works such as Bayesian model-based reinforcement learning [24, 25, 26, 27], offer a framework where a belief and uncertainty is maintained over the MDP model itself. They show that an MDP with unknown transition dynamics can be formulated as a POMDP where the model is learned during execution. This is a very similar setting to our framework, where our observation function is represented by a Gaussian process. However, these works do not explicitly model latent state variables, nor consider continuous, stochastic and time-varying action durations.

Model-free reinforcement learning works have also considered similar problems to ours, with [28] considering optimal time-bounded reachability problems, and [29] demonstrating that latent variable models can be used to represent non-stationary environment dynamics. However, the former does not reason about latent variables and time-varying dynamics, and the latter does not consider time-bounded goals or the probability of mission satisfaction.

3 Preliminaries

Gaussian Process Regression A GP [5] is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. GPs place a Gaussian prior over the space of functions, and are popular priors for Bayesian nonparametrics. A GP is fully specified by its mean $m(s)$ and covariance functions $k(s, s')$, i.e. $f(s) \sim \mathcal{GP}(m(s), k(s, s'))$. We let $m(s) = 0$ without loss of generality. Given a dataset of N_d observations, $\mathcal{D} = \{s_i, z_i\}_{i=0}^{N_d}$ and a prior assumption of joint-Gaussianity, the joint distribution of any observed values and the function value at a new state s' under the prior can be explicitly computed. The Gaussian posterior distribution can then be obtained by exact inference:

$$P(f(s') \mid s', \mathcal{D} = \{s_i, z_i\}_{i=0}^{N_d}) = \mathcal{N}(f(s') \mid \mu(s'), \Sigma(s')), \tag{1}$$

where:

$$\begin{aligned} \mu(s') &= \mathbf{k}_*(K + \sigma^2 \mathbf{I})^{-1}\mathbf{z}, \\ \Sigma(s') &= k(s', s') - \mathbf{k}_*^T (K + \sigma^2 I)^{-1}\mathbf{k}_*, \end{aligned} \tag{2}$$

and $\mathbf{z} = [z_0, z_1, \dots, z_{N_d-1}]^T$. We use the following compact notation: K denotes the $N_d \times N_d$ matrix of covariances evaluated between all pairs of training points. Similarly, $\mathbf{k}_*$ denotes the $N_d \times 1$ vector of covariances between the test point s' and each of the training data points. Finally, $k(s', s')$ is the covariance of the test location with itself. The choice of mean and covariance functions encode prior assumptions over the function, such as smoothness or periodicity, and are parameterised by hyperparameters $\boldsymbol{\theta}$. Specifically, for perpendicular directions of ocean current vectors, we use a multi-output spatio-temporal kernel function. We describe the specific covariance functions and hyperparameter priors used in the experiments section, and optimise $\boldsymbol{\theta}$ for the marginal log-likelihood.

Time-Bounded Reachability over Semi-MDPs An SMDP is an extension to an MDP where actions have strictly positive duration distributions. An SMDP is defined as $\mathcal{M} = \langle \mathcal{S}, \overline{s}, A, \mathcal{T}, \Delta, AP, L \rangle$, where: $\mathcal{S}$ is a finite set of discrete states; $\overline{s} \in \mathcal{S}$ is the initial state; A is a finite set of actions; $\mathcal{T} : \mathcal{S} \times A \to Dist(\mathcal{S})$ is a probabilistic transition function, i.e. $\mathcal{T}(s, a)(s')$ returns the probability of arriving at state s' after taking action a in state s; $\Delta : \mathcal{S} \times A \times \mathcal{S} \to Dist(\mathbb{R}^+)$ is the action duration distribution, i.e. $\Delta(s, a, s')$ represents a strictly positive and continuous time distribution of action a starting in state s and finishing in s'; AP is a set of state atomic propositions; and $L : \mathcal{S} \to 2^{AP}$ is a labelling function, such that $p \in L(s)$ iff p is true in state s. An SMDP represents the possible evolutions of a system where, in each state s, any action a from the enabled actions $A(s) = \{a \in A \mid \mathcal{T}(s, a)(s') > 0 \text{ for some } s' \in \mathcal{S}\}$ can be selected. We define a *path* ρ over an SMDP as a sequence, i.e. $\rho = (s_0, a_1, \delta_1)(s_1, a_2, \delta_2) \, \dots$ where $s_0 = \overline{s}$, $a_{n+1} \in A(s_n)$ and $\delta_{n+1} \in \mathbb{R}^+$ is the duration of a_{n+1}. We denote the set of all paths of $\mathcal{M}$ starting from $\overline{s}$ as $Path_{\mathcal{M}}$.

Action selection can be described by a time-dependent policy $\pi : \mathcal{S} \times \mathbb{R}^+ \to A$, which maps a state and time to an action to be executed. We denote the set of all such policies as $\Pi_{\mathcal{M}}$. Given a fixed policy π all nondeterminism over action selection is resolved and a probability measure $Pr^{\pi}_{\mathcal{M}}$ over the space of all paths through $\mathcal{M}$ can be specified [30]. In this paper, we aim to synthesise policies that maximise the probability of reaching a set of goal states $\mathcal{G} \subset \mathcal{S}$ under a user-defined time bound $0 < T < \infty$. Formally, we define $reach^T_{\mathcal{G}} : Path_{\mathcal{M}} \to \{0, 1\}$ such that $reach^T_{\mathcal{G}}((s_0, a_1, \delta_1)(s_1, a_2, \delta_2) \, \dots) = 1$ if and only if there exists n such that $s_n \in \mathcal{G}$ and $\sum_{i=1}^{n} \delta_i \leq T$. Our objective is then to find $Pr^*_{\mathcal{M}}(reach^T_{\mathcal{G}}) = \sup_{\pi \in \Pi_{\mathcal{M}}} Pr^{\pi}_{\mathcal{M}}(reach^T_{\mathcal{G}})$ and a corresponding policy $\pi^* = \arg\max_{\pi \in \Pi_{\mathcal{M}}} Pr^{\pi}_{\mathcal{M}}(reach^T_{\mathcal{G}})$. Finding $Pr^{\pi}_{\mathcal{M}}(reach^T_{\mathcal{G}})$ is equivalent to integrating over the time-bound of the joint pdf of every possible action distribution choice, over all possible paths, to the goal states. Maximising this product exactly is intractable, thus we will use sampling-based methods to approximate π^*. Specifically, we propose a tree search algorithm that only expands reachable states and, in order to effectively reason about the time remaining to reach the goal, maintains a representation of elapsed time in the search nodes.

Co-Safe Linear Temporal Logic Co-safe linear temporal logic (csLTL) [6] provides a convenient way to specify missions for robots [31]. A csLTL statement ϕ over atomic propositions AP is specified using the following grammar: $\phi ::= true \mid p \mid \neg p \mid \phi \wedge \phi \mid \mathtt{X}\, \phi \mid \phi\, \mathtt{U}\, \phi$, where $p \in AP$. Operator $\mathtt{X}$ is read “next”; $\mathtt{U}$ is read “until”. We can derive useful operators, in particular $\mathtt{F}\, \phi$, read “eventually”, which requires that ϕ to be satisfied in the future. Please refer to [32] for more details.

We denote the maximum probability of satisfying ϕ in time less than or equal to T as $Pr^*_{\mathcal{M}}(\phi^{\leq T})$. It is known [6] that any csLTL formulae ϕ written over AP can be represented as a deterministic finite automaton (DFA) $\mathcal{A}_{\phi} = \langle \mathcal{X}, \overline{x}, \mathcal{X}_F, 2^{AP}, \delta_{\phi} \rangle$, where: $\mathcal{X}$ is a finite set of states, with states $x \in \mathcal{X}$ representing the current stage of satisfaction of ϕ; $\overline{x} \in \mathcal{X}$ is the initial state; $\mathcal{X}_F \subseteq \mathcal{X}$ is the set of accepting states; 2^{AP} is the alphabet; and $\delta_{\phi} : \mathcal{X} \times 2^{AP} \to \mathcal{X}$ is a transition function. We maintain the

DFA state throughout planning, which builds an abstraction of the *product* SMDP $\mathcal{M}_\phi$. The product SMDP $\mathcal{M}_\phi$ behaves like the original SMDP but is augmented with information about the current state of $\mathcal{A}_\phi$ in order to keep track of the satisfaction of ϕ; states in $\mathcal{M}_\phi$ are of the form $(s, x) \in \mathcal{S} \times \mathcal{X}$. A full definition of the product construction can be found [33]. Computing the maximum probability of satisfying ϕ in T time $Pr^*_{\mathcal{M}}(\phi^{\leq T})$ can be reduced to a time-bounded reachability problem in $\mathcal{M}_\phi$ [6, 22]. Specifically, $Pr^*_{\mathcal{M}}(\phi^{\leq T}) = Pr^*_{\mathcal{M}_\phi}(reach^T_{\mathcal{G}_\phi})$ where $\mathcal{G}_\phi = \{(s, x) \in S \times \mathcal{X} \mid x \in \mathcal{X}_F\}$.

4 Time-Varying and Uncertain Dynamics

We are interested in planning scenarios where time-varying environment dynamics affect a robot's actions and are *a priori* unknown. We define the environment as a continuous $d-$dimensional space with strictly positive time, and assume the environment dynamics can be modelled by a function that is smooth and time-varying, i.e. $f : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^m$. For time-varying dynamics that are *unknown* in advance, the model is required to maintain a belief about how the spatio-temporal environment function f affects the robot's actions and transitions. For example in ocean domains, the *a priori* unknown latent state feature could represent ocean current velocity. We employ a GP to maintain a belief distribution over the true value of f. As the robot moves through the environment, the belief is sequentially updated with observations of f at the robot's current location.

We define a time-varying SMDP with *a priori* unknown dynamics and GP belief (TV-SMDP-GP) as $\mathcal{M} = \langle \mathcal{S}_k \times \mathcal{S}_e, (\bar{s_k}, \bar{s_e}), \hat{f}, A, \mathcal{T}, \Delta, AP, L \rangle$. This extends the standard SMDP model in several ways. First, we factorise the state space, i.e. $\mathcal{S} = \mathcal{S}_k \times \mathcal{S}_e$. This state space is built from known and fully observable state features $\mathcal{S}_k \subseteq \mathbb{R}^d$, and unknown or partially observable latent state features $\mathcal{S}_e \subseteq \mathbb{R}^m$. The value of the latent features is assumed to be uniquely defined according to the unknown time-varying mapping f, i.e. states are of the form $s = (s_k, f(s_k, t))$. We assume $\mathcal{S}_k$ to be a finite abstraction of $\mathbb{R}^d$, i.e. we discretise the environment into a finite set. Furthermore, we restrict the labelling function to known state features $\mathcal{S}_k$, i.e. $L : \mathcal{S}_k \rightarrow 2^{AP}$.

Second, we extend the transition and action duration functions to consider the latent state features, and be time-dependent, as follows: $\mathcal{T} : \mathcal{S}_k \times A \times \mathbb{R}^+ \rightarrow Dist(\mathcal{S}_k)$, i.e. $\mathcal{T}(s_k, a, t)(s'_k)$ returns the probability of arriving at state s'_k after taking action a in state s_k at time t; and $\Delta :\ (\mathcal{S}_k \times \mathcal{S}_e) \times A \times (\mathcal{S}_k \times \mathcal{S}_e) \times \mathbb{R}^+ \rightarrow Dist(\mathbb{R}^+)$ is a strictly positive and continuous distribution where $\Delta(s, a, s', t)$ returns the duration distribution of action a from state $s = (s_k, s_e)$ to $s' = (s'_k, s'_e)$ starting at time t. Note that we assume the transition function $\mathcal{T}$ depends only on the known state features, whereas the action duration distribution Δ is also dependent on the time-varying latent state feature. This dependence enables rich modelling of the influence of the latent state feature. For example, in the case illustrated in Figure 1, our model specifies that the action duration depends on the ocean currents.

Finally, we introduce an approximate mapping function $\hat{f} : \mathcal{S}_k \times \mathbb{R}^+ \rightarrow Dist(\mathcal{S}_e)$ that models the belief over the unknown function f. This formulation provides a flexible function approximator for the time-varying dynamics of the unknown latent state features $\mathcal{S}_e$. We place a GP prior with spatio-temporal kernel over the belief $\hat{f}$, and condition on observations of the environment function f which are gathered at the robot's current location and incrementally added to the dataset $\mathcal{D}$ online.

Our flexible TV-SMDP-GP formalisation behaves similarly to a regular SMDP; however, the model evolves according to time-varying dynamics informed by the belief over latent state features. Thus, this formulation is similar to maintaining a belief over the transition function in a POMDP, or model-based BRL, as described in Section 2. Our formulation can be easily adapted to *known* time-varying dynamics by simply making $S = S_k$ and not using the function approximator $\hat{f}$, i.e. maintaining the time-varying distributions $\mathcal{T}$ and Δ over the fully known state space. We refer to this as TV-SMDP and evaluate it as an upper baseline in Section 6.

5 Solving Time-Varying SMDPs

Online Planning and Execution For efficiency, we are interested in online planning methods that restrict computation to reachable states and times, and are capable of utilising time-dependent transitions. Our proposed framework is presented in Algorithm 1. At each planning epoch (line 2), the robot samples the environment around its current state and updates its GP belief (line 3); then uses MCTS for τ trials to repeatedly sample paths from the current search node (defined later) to

Algorithm 1 TV-SMDP WITH GP BELIEF (TV-SMDP-GP)

```
Input: TV-SMDP $\mathcal{M}$, csLTL $\phi$, Time-bound $T$, Belief $\hat{f}$; $\#$trials $\tau$; $\#$observations $\omega$;
1: Initialise: Compute DFA $\mathcal{A}_\phi := \langle \mathcal{X}, \overline{x}, \mathcal{X}_F, 2^{AP}, \delta_\phi \rangle$; $t \leftarrow 0$; $s_k \leftarrow \bar{s_k}$; $x \leftarrow \delta_\phi(\bar{x}, L(\bar{s_k}))$
2: while $x \notin \mathcal{X}_F$ and $t < T$ do
3:     Add $\omega$ noisy observations to $\mathcal{D} \leftarrow \{s_k + \sigma_i, f(s_k + \sigma_i, t - \sigma'_i)\}_{i=0}^{\omega}$; Update belief funtion $\hat{f}$
4:     $b \leftarrow (s_k, \hat{f}(s_k, t))$; $n \leftarrow (b, x, t)$
5:     $\hat{\pi}^* \leftarrow$ Perform $\tau$ UCT trials from $n$, with goal $\{(b, x, t) \mid x \in \mathcal{X}_F$ and $t < T\}$, and extract policy
6:     Execute action $\hat{\pi}^*(n)$
7:     Update $s_k$ and $t$ according to outcome of $\hat{\pi}^*(n)$; $x \leftarrow \delta_\phi(x, L(s_k))$
8: end while
```

provide an approximate optimal policy $\hat{\pi}^*$ (line 4-5). Finally a single action is taken (line 6), and the known state feature and DFA state are updated accordingly (line 7). In our setting, the robot executes the single most sampled action from the root node, observes the environment to update its belief, and re-plans. However, a robot could follow $\hat{\pi}^*$ for multiple actions until the uncertainty on transitions grows larger than a user specified threshold.

UCT We employ upper confidence bounds applied to trees (UCT) [7], a widely used variant of MCTS. UCT incrementally builds a search tree where, for each node n, and action a, we keep an estimate $Q(n, a)$ of the probability of satisfying the time-bounded mission specification ϕ. This estimate is computed by backpropagating a reward of 1 for trials that satisfy the mission within the bound, and maintaining a sample average. A trial is a sample path through the product of the TV-SMDP-GP and the csLTL DFA, and evolves according to $\mathcal{T}$, Δ and δ_ϕ. Actions along trials are chosen according to the UCT formula, which considers the current value estimate $Q(n, a)$, and a term that favours less-explored action choices. When a trial reaches a leaf node, the *rollout* policy $\overline{\pi}$ is used to provide an initial estimate of its value.

Search Node Representation and Sampling Search nodes in the UCT tree represent reachable states. We augment them with the *time* that the node is reached, along with satisfaction information of ϕ. Formally, we define a node as $n := (b, x, t)$, where $b = (s_k, \hat{f}(s_k, t))$ represents the current known state s_k and the belief $\hat{f}$ at s_k and t; $x \in \mathcal{X}$ is the DFA state; and $t \in \mathbb{R}^+$ is the time the node is reached. Given a node n and an action a, the UCT algorithm needs to sample a successor node. A challenge for sampling in non-stationary environments is the inherent chicken-and-egg problem that arises: the time-varying dynamics observed at future states affects the action durations to reach those states. Arriving slightly earlier, or later, in-turn affects the dynamics the robot experiences, affecting future actions ad infinitum. For tractability, we sample from the distributions in a sequential fashion. First, we sample the known state feature $s'_k \sim \mathcal{T}(s_k, a, t)$. Second, the unknown state feature is estimated at s'_k using the current GP belief $\hat{f}$, i.e. $b' = (s'_k, \hat{f}(s'_k, t))$. Third, time is incremented to $t' = t + \delta$, where δ is sampled from Δ, accounting for the current belief $\hat{f}$. There are several ways to account for the belief when sampling from Δ, e.g. considering only the expected value of $\hat{f}$ does not take into account the uncertainty over the value of the unknown state feature; sampling from $\hat{f}(s_k, t)$ and $\hat{f}(s'_k, t)$ can yield an unnecessary number of search nodes; and integration is computationally expensive to be used in sampling-based search. To address these issues, we instead extend Δ to consider the mean and variance of the belief $\hat{f}(s'_k, t)$. This allows for computationally efficient sampling of durations that considers the uncertainty over the value of the unknown state features. We provide more information about the specific extension to Δ used in our experiments in the Appendix. Finally, the successor DFA state x' is defined by synchronising the labels of s'_k with the transition function of $\mathcal{A}_\phi$, i.e. $x' = \delta_\phi(x, L((s'_k)))$. This effectively builds a reachable abstraction of the product SMDP on-the-fly. The successor node can now be defined as $n' := (b', x', t')$.

Note that the same successor node will never be sampled twice, since δ is sampled from a continuous distribution. Thus, in order to maintain meaningful sample averages $Q(n, a)$, one needs to aggregate search nodes. To do so, when sampling a new node $n' := ((s'_k, \hat{f}(s'_k, t)), x', t')$ from n, we check if there already exists a child node n'' of n, such that $s''_k = s'_k$ and $|t'' - t'| < \epsilon$, where ϵ is a small positive constant used to treat close continuous values as the same outcome. If the child node n'' exists, then we continue the trial from n'' rather than creating a new search node n'. The benefit of sampling the node's time, over discretising it in advance, is that only reachable states are added to the search tree, and ϵ can be changed on-the-fly. This parameter can be considered as restricting the branching factor of the tree, where each child node is both reachable and representative of the dynamics. Progressive widening [34] is an alternative approach to do this based on the frequency of visits, and also relies on parameterisation.

6 Evaluation

We compare three models: i. an SMDP with a stationary action duration distribution Δ as a lower baseline on performance. ii. a TV-SMDP with fully known dynamics, $\mathcal{T}$ and Δ are as per an oracle. iii. a TV-SMDP-GP under unknown environment dynamics that is required to maintain a belief $\hat{f}$.

We evaluate the models by performing time-bounded csLTL missions. First, in a simulated environment with rotating Gaussian dynamics. We demonstrate csLTL specifications with one, two and three goal propositions, such as in autonomous deep-sea hydrothermal vent monitoring. Second, we use real-world ocean currents data from three different regions of the Atlantic-European NW Shelf [35]. Dataset analysis is presented in Appendix 2, along with further example images.

Experiment 1: Rotating Gaussian Dynamics We perform 150 random time-bounded csLTL missions in a 10×10 grid-world, under a multivariate Gaussian function f that is rotating counter-clockwise around a fixed point in the environment. The true action duration Δ^* between state s and s' at time t is dependent on the environment, i.e. $\Delta^*(s, a, s') = 1 + f(s', t)$ seconds, where the height of the multivariate Gaussian slows the robot's traversal actions. We generate 50 random missions by sampling a start and goal state, and defining the mission as: $\texttt{F}\, g$ (eventually satisfy g) within time T, where g is an atomic proposition labelling the sampled goal state. The timebound T is defined as a scalar multiple of the L_1 distance between the start and goal states, and is successively relaxed creating four difficulty settings. We repeat the 50 missions with one additional goal g_1, specifying the mission as: $\texttt{F}\, g \wedge \texttt{F}\, g_1$ (eventually satisfy g and g_1 in any order), within time T. Finally, we repeat the 50 missions a third time with a second additional goal g_2 and extend the mission to: $\texttt{F}\, g \wedge \texttt{F}\, g_1 \wedge \texttt{F}\, g_2$ within time T. Full experiment details are provided in Appendix 1. Images of the time-varying environment f at six timepoints are shown in Appendix 1 Figure 4.

The SMDP model is characterised by sampling each action duration from a stationary $\Delta = log\mathcal{N}(1, 0.01)$ distribution. This is akin to a discrete-time MDP, where time is discretised in advance to 1 second intervals, i.e. equal to the duration of transitioning between any two neighbouring states when not accounting for the environment dynamics. TV-SMDP has full access to query f without noise, so Δ is time-varying, and it perfectly represents the environment's influence over the transition model. TV-SMDP-GP maintains a GP belief over f, and at each planning epoch samples 10 observations around its current location with $\mathcal{U}(0, 1)$ perturbations.

Each MCTS planning epoch performs $\tau = 1000$ simulated trials, plus an additional $\tau' = 1000$ if $Q(n, a) = 0$ for all available actions at the root node. Therefore the tree size is bounded by 2000 reachable states, given that only one node is added per trial, as is standard in MCTS algorithms. If $Q(n, a) = 0$ for all actions after $\tau + \tau'$ trials, the rollout policy action is chosen. During the simulation phase, an admissible heuristic rollout policy is used based on the L_1 distance between the current state and each goal, with ties broken randomly. Finally, we backpropagate a reward whenever the DFA progresses towards the accepting state, according to the progression function defined in [31]. For example, for mission $\texttt{F}\, g \wedge \texttt{F}\, g_1$, we backpropagate a reward of $1/2$ whenever either goal is visited during the trial. This allows for improved efficiency over csLTL missions with multiple goals, by providing the search algorithm with a less sparse reward signal. Implementation details are provided in Appendix 1.

Experiment 2: Real-World Ocean Currents Dataset Due to the difficulty of performing real-world experiments in marine robotics, planning researchers typically use ocean datasets to evaluate systems on long-term spatio-temporal data. We use a publicly available ocean currents dataset from the EU project Copernicus Marine Service [35]. The dataset contains hourly ocean current forecast data over the Atlantic-European NW Shelf, at a resolution of $1,500$m. Based on an approach from [1], we interpolate between available forecasts for three separate regions using a GP trained on 12 hours of data for that region. Full details are provided in the Appendix 2.

Table 1: Rotating Gaussian dynamics: Averages over missions completed by all models.

	Num of Goals	Success Rate	Avg Planning Steps to Goal	Avg Plan Time per step	Avg Mission Time to Goal
1. SMDP	1	52.0% (23.3)	9.24 (2.7)	3.94(s) (1.1)	13.78(s) (4.2)
2. TV-SMDP		88.5% (15.3)	8.70 (2.1)	4.28(s) (1.2)	11.83(s) (3.2)
3. TV-SMDP-GP		71.0% (22.5)	8.54 (1.7)	140.09(s) (34.3)	12.69(s) (3.8)
1. SMDP	2	61.0% (18.7)	16.90 (7.8)	5.97(s) (1.5)	25.77(s) (10.2)
2. TV-SMDP		93.5% (10.5)	13.80 (4.7)	6.49(s) (1.7)	19.11(s) (6.4)
3. TV-SMDP-GP		84.0% (12.1)	12.61 (3.8)	138.23(s) (43.8)	20.34(s) (7.9)
1. SMDP	3	52.5% (13.3)	26.64 (11.2)	10.62(s) (2.5)	40.54(s) (12.9)
2. TV-SMDP		98.0% (1.6)	21.42 (8.2)	11.30(s) (2.8)	29.72(s) (9.6)
3. TV-SMDP-GP		90.0% (6.3)	16.97 (6.3)	205.84(s) (57.3)	27.10(s) (8.4)

Table 2: Real world Ocean Currents: 30 Missions over three regions of Ocean.

	Success Rate	Avg Steps to Goal	Avg Plan Time	Avg Mission Time
1. SMDP	30.2% (15.4)	7.55 (0.7)	4.94(s) (1.2)	8.89(hrs) (1.9)
2. TV-SMDP	52.7% (14.1)	7.56 (0.7)	85.52(s) (21.6)	8.83(hrs) (1.7)
3. TV-SMDP-GP	36.7% (16.7)	7.49 (0.6)	163.08(s) (40.8)	8.79(hrs) (1.8)

We perform 30 time-bounded reachability missions over three different ocean regions where the currents (u, v) affect robot traversal action duration. We sample a random start $\bar{s_k}$ and goal g within a 6×6 grid, and random start time t from the available forecasts. We compare the three models over four time-bound mission difficulties by relaxing T. We force the robot to exploit the currents by setting very short time-bounds, i.e. less than the required time to travel under its own force. Region two is shown at two timepoints in Figure 3 (left). Full experiment details are provided in Appendix 2, along with further ocean dataset images of Region three in Figure 6.

We have artificially disadvantaged the TV-SMDP-GP model in this experiment. At the start of missions the TV-SMDP-GP relies only on the modelling assumptions of the GP prior. Whereas, in a real-world robotic deployment, the belief function could be pre-trained with the most recent available daily forecasts ahead of time. In this scenario, the TV-SMDP-GP mission performance would tend towards the TV-SMDP model, and the belief would only be required to capture the inherent unforeseen dynamics of the ocean affecting the robot's dynamics.

Results & Discussion Results summarising all 150 time-bounded missions in Experiment 1 under Gaussian dynamics are provided in Table 1, and Experiment 2 under real-world ocean currents in Table 2 (std shown in parenthesis). Table results are aggregated for missions that were completed by all three models at any difficulty setting, so provide a direct comparison of the completed missions.

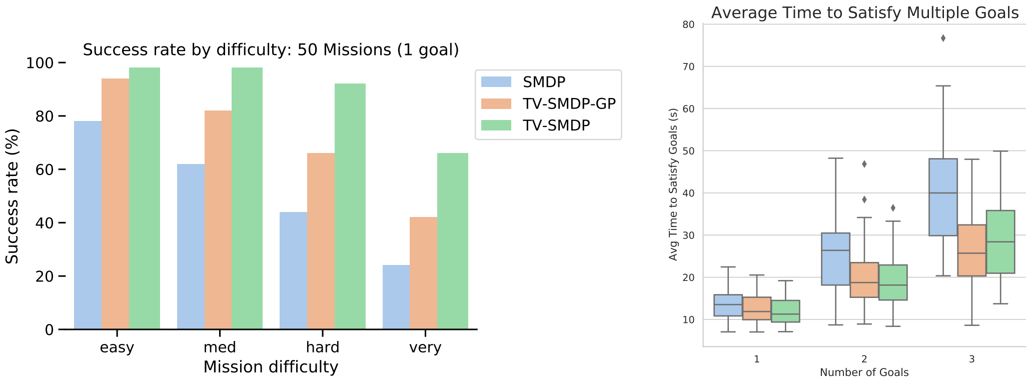

Figure 2: Left: Success-rate of 50 missions under rotating Gaussian dynamics by time-bound difficulty. csLTL missions specified with one goal state (see Appendix 1 Figure 5 for missions with additional goal states). Right: Distributions of total time to complete csLTL missions with multiple goals during Experiment 1: rotating Gaussian dynamics.

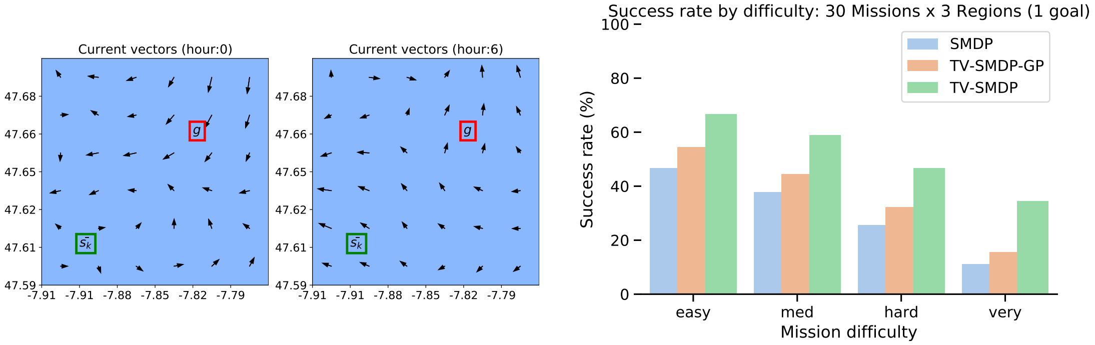

Figure 3: Left: Ocean currents region two: A portion Atlantic Ocean (47.6° to 47.7°N and $-7.9°$ to $-7.8°$E) at two timepoints, along with a mission: start $\bar{s_k}$ (green) and goal g (red). Right: Success-rate of 90 missions (30 $\times$ 3 regions) under real-world ocean dynamics by time-bound difficulty.

Figure 2 (left) presents the proportion of mission successes (binary outcome) for missions with one goal, broken down by the four mission difficulties specified by T. Either the robot successfully satisfied a mission or it failed (no variance). There is a clear trend: when tight time-bounds are specified, the TV-SMDP and TV-SMDP-GP models exploit the spatio-temporal dynamics of the domain and plan ahead in order to succeed, unlike the SMDP model which plans using a stationary Δ.

Figure 2 (right) presents the distributions of total simulated mission time for missions with increasing number of goals. We see that overall mission time increases, with higher variance, as more goals are specified which is as expected. Importantly however, we also see that the TV-SMDP and TV-SMDP-GP models require less planning steps per mission on average (Table 1), and satisfy the missions quicker than the SDMP model. However, planning wallclock time is substantially higher for models that are required to maintain a GP belief function.

Finally, in Table 2 we see that TV-SMDP and TV-SMDP-GP are able to exploit ocean currents over three different regions of the Atlantic-European ocean. By maintaining a belief, the TV-SMDP-GP model is required to make predictions outside of its dataset of observations, and as such propagates its uncertainty into planning in order to make better action choices in this online setting.

7 Conclusion

In this paper we have proposed an online planning framework to address time-bounded missions under time-varying dynamics, where those dynamics affect the duration and outcome of actions. We demonstrate how *a priori* unknown environment dynamics can be represented using a GP, and incorporated into the time-dependent Semi-MDP dynamics, and solved using MCTS. We show improved mission performance on marine vehicle data acting under real-world spatio-temporal ocean currents, and demonstrate the ability to solve csLTL missions in time-varying domains. As time-bounded reachability specifications become more difficult to satisfy, TV-SMDP and TV-SMDP-GP are shown to exploit the known, or predicted dynamics of the environment to assist completing the csLTL mission within the time-bound. To the best of our knowledge, we present the first algorithm to handle time-varying problems with unknown dynamics, where those dynamics affect robot actions, and the first algorithm capable of satisfying csLTL missions in time-varying domains.

In future work we first intend to integrate a sparse GP approximation as per [36] for scaling to more observations. Second, we intend to extend the framework to handle full LTL mission specifications, and develop a framework for persistent surveillance rather than single missions.

Acknowledgments

We thank colleagues at the Oxford Robotics Institute, specifically Charlie Street, Marc Rigter and Faaria Khan, for sharing supporting software. This work was 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] L. Liu and G. S. Sukhatme. A solution to time-varying Markov decision processes. *IEEE Robotics and Automation Letters (RAL)*, 3(3):1631–1638, 2018.

[2] J. Pulido Fentanes, B. Lacerda, T. Krajník, N. Hawes, and M. Hanheide. Now or later? Predicting and maximising success of navigation actions from long-term experience. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[3] R. A. Howard. *Dynamic Programming and Markov Processes.* MIT Press, 1960.

[4] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

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

[6] O. Kupferman and M. Vardi. Model checking of safety properties. *Formal Methods in System Design (FMSD)*, 19(3), 2001.

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

[8] J. Xu, K. Yin, and L. Liu. Reachable space characterization of Markov decision processes with time variability. In *Robotics: Science and Systems (RSS)*, 2019.

[9] J. A. Boyan and M. L. Littman. Exact solutions to time-dependent MDPs. In *Neural Information Processing Systems (NeurIPS)*, pages 1026–1032, 2001.

[10] E. Rachelson, P. Fabiani, and F. Garcia. Timdppoly: An improved method for solving time-dependent MDPs. In *21st International Conference on Tools with AI*, pages 796–799. IEEE, 2009.

[11] C. Van Dover. *The ecology of deep-sea hydrothermal vents*. Princeton University Press, 2000.

[12] G. Flaspohler, V. Preston, A. P. Michel, Y. Girdhar, and N. Roy. Information-guided robotic maximum seek-and-sample in partially observable continuous environments. *IEEE Robotics and Automation Letters (RAL)*, 4(4):3782–3789, 2019.

[13] A. Viseras, D. Shutin, and L. Merino. Online information gathering using sampling-based planners and GPs: An information theoretic approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 123–130, 2017.

[14] K.-C. Ma, L. Liu, H. K. Heidarsson, and G. S. Sukhatme. Data-driven learning and planning for environmental sampling. *Journal of Field Robotics*, 35(5):643–661, 2018.

[15] 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)*, pages 6381–6388, 2017.

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

[17] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause. Learning-based model predictive control for safe exploration. In *Conference on Decision and Control (CDC)*, pages 6059–6066. IEEE, 2018.

[18] M. Turchetta, F. Berkenkamp, and A. Krause. Safe exploration for interactive machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2887–2897, 2019.

[19] M. Budd, B. Lacerda, P. Duckworth, and N. Hawes. Markov decision processes with unknown state feature values for safe exploration using Gaussian processes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[20] S. J. Bradtke and M. O. Duff. Reinforcement learning methods for continuous-time Markov decision problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 393–400, 1995.

[21] T. Chen and J. Lu. Towards analysis of semi-Markov decision processes. In *International Conference on Artificial Intelligence and Computational Intelligence*, pages 41–48. Springer, 2010.

[22] B. Lacerda, D. Parker, and N. Hawes. Multi-objective policy generation for mobile robots under probabilistic time-bounded guarantees. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2017.

[23] R. Marchant, F. Ramos, and S. Sanner. Sequential Bayesian optimisation for spatial-temporal monitoring. In *Conference on Uncertainty in AI (UAI)*, 2014.

[24] Y. Wang, K. S. Won, D. Hsu, and W. S. Lee. Monte Carlo Bayesian reinforcement learning. *International Conference on Machine Learning (ICML)*, 2012.

[25] A. Guez, D. Silver, and P. Dayan. Efficient Bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1025–1033, 2012.

[26] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 8(5-6):359–483, 2016.

[27] G. Lee, B. Hou, A. Mandalika, J. Lee, and S. S. Srinivasa. Bayesian policy optimization for model uncertainty. In *International Conference on Learning Representations (ICLR)*, 2019.

[28] Z. Cao, H. Guo, J. Zhang, F. Oliehoek, and U. Fastenrath. Maximizing the probability of arriving on time: A practical Q-learning method. In *Thirty-First AAAI Conference on AI*, 2017.

[29] A. Xie, J. Harrison, and C. Finn. Deep reinforcement learning amidst lifelong non-stationarity. *ICML Workshop: Lifelong ML*, 2020.

[30] J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov chains: chapter of Markov random fields by David Griffeath*, volume 40. Springer Science & Business Media, 2012.

[31] B. Lacerda, F. Faruq, D. Parker, and N. Hawes. Probabilistic planning with formal performance guarantees for mobile service robots. *The International Journal of Robotics Research (IJRR)*, 38(9), 2019.

[32] A. Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13 (1):45–60, 1981.

[33] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT press, 2008.

[34] 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.

[35] Copernicus:. Europe's eyes on earth. `https://resources.marine.copernicus.eu/?option=com_csw&task=results?option=com_csw&view=details&product_id=NORTHWESTSHELF_ANALYSIS_FORECAST_PHY_004_013`, 2020. Accessed: 2020-06.

[36] M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

[37] M. Tonani, P. Sykes, R. R. King, N. McConnell, A.-C. Péquignet, E. O'Dea, J. A. Graham, J. Polton, and J. Siddorn. The impact of a new high-resolution ocean model on the Met Office North-West European Shelf forecasting system. *Ocean Science*, 15(4):1133–1158, 2019.

Appendix

1 Experiment 1: Rotating Gaussian Dynamics

Environment Dynamics: A single rotating multivariate Gaussian over a 10×10 grid-world:

$$f(s_k, t) = \mathcal{N}\left(m \otimes \begin{bmatrix} cos(t/r) + 5 \\ sin(t/r) + 5 \end{bmatrix}, \Sigma\right),$$

where $m = [3, 3]$, r=$\frac{2\pi}{50}$, $\otimes$ denotes element-wise product and $\Sigma = \text{diag}(5.1, 5.1)$.

The known state features (x, y) are 1m apart; the agent speed is 1m/s; and the robot's available actions $A(s)$ are: 1m $\times\{up, down, left, right\}$ in all but boundary states when this set is reduced.

The true and deterministic action duration Δ^* between state s and s' at time t is dependent on the spatio-temporal environment: $\Delta^*(s, a, s') = 1 + f(s', t)$ seconds, where the dynamics slows the robot's traversal.

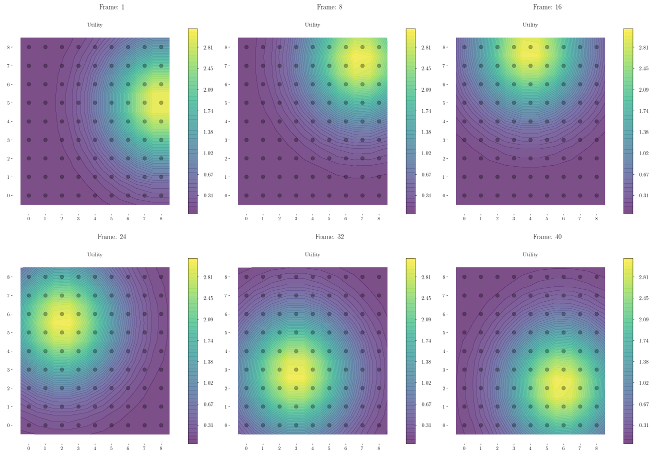

Figure 4: Left: Counter-clockwise rotating Gaussian dynamics that slow the robot's traversal actions, at six timepoints. Yellow indicates high values and slowest action durations.

Missions: We specify mission difficulty based upon the imposed time-bound between *easy* and *very* difficult, i.e. a scalar $\{2.2, 2.0, 1.8, 1.5\}\times||\bar{s_k}, g||_1$ respectively, where $|| \cdot ||_1$ refers to the L_1 distance between the start state $\bar{s_k}$ and the goal g. This distance is set to a minimum of 8 for the 50 random missions. Additional goals are added with an L_1 distance of no less than 4. Each mission is performed for each difficulty setting.

TV-SMDP-GP: Observations are sequentially added to $\mathcal{D}$:
$\{((x + 1) - 2\sigma_i, (y + 1) - 2\sigma_j, t - \sigma_k), f(x, y, t)\}_{i=0}^{\omega}$, where $\sigma_i, \sigma_j, \sigma_k \sim \mathcal{U}(0, 1)$ and $\omega = 10$.

GP Belief: The GP belief $\hat{f} : \mathbb{R}^2 \times \mathbb{R}^+ \to Dist(\mathbb{R})$ is queried for a belief distribution over s'_e for each new search node n', i.e. $\mathcal{N}(\hat{f}(s'_k, t) \mid \mu(s'_k, t), \Sigma(s'_k, t))$, where μ and Σ are specified in Equation (2). The kernel function k is a spatio-temporal kernel comprising of: $k_{\text{additive}} = k_{xy} + k_t + k_{lin}$, where: $k_{xy} = \sigma_{xy}\text{exp}\left(-\frac{((x,y)-(x',y'))^2}{2l_{xy}^2}\right)$,
$k_t = \sigma_t\text{exp}\left(-\frac{(t-t')2}{2l_t^2}\right)$, and $k_{lin} = \sigma_{lin}^2((x, y, t)(x', y', t'))$.

We place prior Gamma distributions over all the GP hyperparameters: $\boldsymbol{\theta} = (l_{xy}, \sigma_{xy}, l_t, \sigma_t, \sigma_{lin})$:

$$p(l_{xy}) = \Gamma(1, 1), p(\sigma_{xy}) = \Gamma(1, 1), p(l_t) = \Gamma(10, 1), p(\sigma_t) = \Gamma(1, 1), p(\sigma_{lin}) = \Gamma(10, 1).$$

The TV-SMDP-GP action duration distribution Δ maintains the mean and variance of the belief distribution $\hat{f}$ over latent state features (as described in Section 5. This facilitates the choice of Δ to be a strictly positive, truncated-Normal distribution centered around the GP mean and variance: $\Delta(b, a, b', t) := \mathcal{N}(\mu(\hat{f}(s'_k, t')), \Sigma(\hat{f}(s'_k, t')))$, truncated from below, by min$= 1$(s), where μ and Σ are defined in Equation (2).

MCTS Parameters: number of known states = 10×10
number of trials $\tau = 1000$
additional trials $\tau' = 1000$
max trial length $\zeta = 100$
exploration weight $p = 0.9$
time epsilon (for considering search nodes equal) ϵ : 0.5
rollout policy used: $\bar{\pi}(s_k) = \arg\min_{a \in A(s)} ||s'_k, g_i||_1 \, \forall \, g_i \in \mathcal{G}$.

Multi-goal Results:

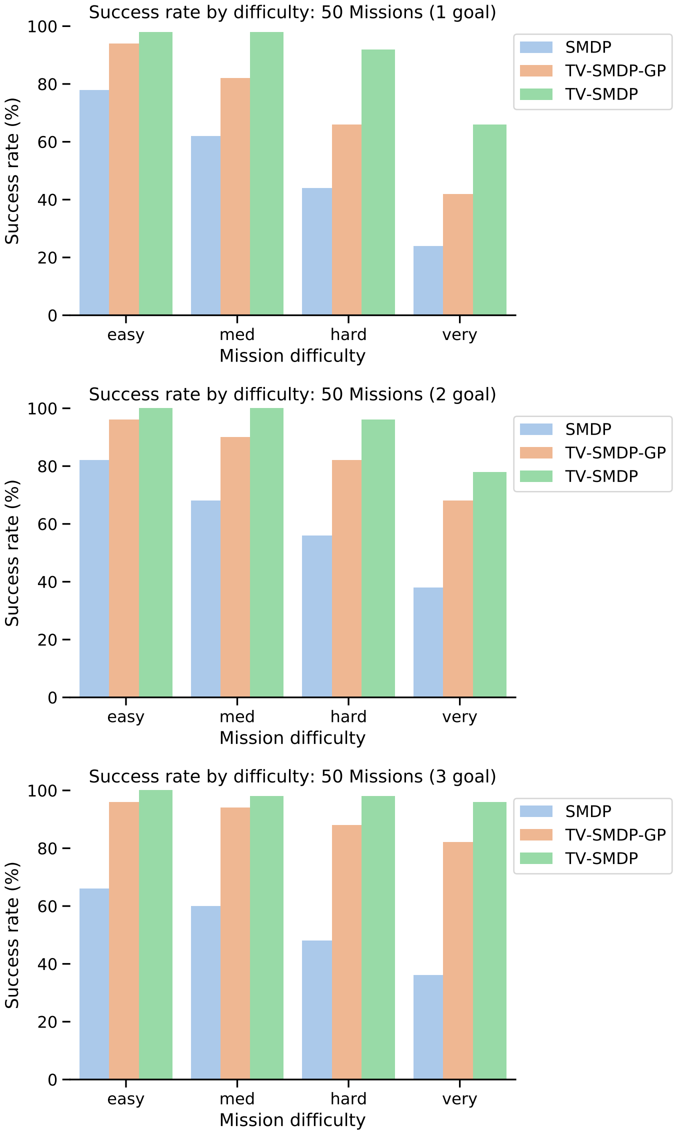

Figure 5: Success-rate of 50 missions under rotating Gaussian dynamics by time-bound difficulty. csLTL missions specified with one (top), two (middle) and three (bottom) goals.

2 Experiment 2: Real World Ocean Current Dynamics

Environment Dynamics: We accessed hourly oceanic current forecasts for three regions of the Atlantic-European NW Shelf:
Region 1: Norwegian Sea $61.1°$ to $61.2°$ latitude and $4.5°$ to $4.65°$ longitude;
Region 2: Atlantic Ocean (West of France) $47.6°$ to $47.7°$ latitude and $-7.91°$ to $-7.79°$ longitude;
Region 3: Atlantic Ocean (North of Scotland) $59.8°$ to $59.9°$ latitude and $-4.67°$ to $-4.55°$ longitude.

The dataset comprises of a grid of fixed locations 1500m apart. Based on an approach from [1], we interpolate between available hourly forecasts using a Gaussian process trained on 12 hours of available forecast data. The forecast we used was for May 1st 2020 and is available online at [35]. Further details on the forecast model in [37].

The known state features are the sensor (x, y) locations 1500m apart, arranged in a dense grid. The robot's available actions are 1500m in four cardinal directions corresponding to vehicle headings: $A(s) = \{0°, 90°, 180°, 270°\}$, in all but boundary states when this set is reduced.

The true action duration of the robot is sampled from a $log\mathcal{N}(\mathbf{v}_r(a), 0.01)$ distribution which is relative to the spatio-temporal ocean currents. We denote the velocity of the robot relative to the water when taking action a by $\mathbf{v}_r(a)$. The velocity of the ocean current at state s and time t is denoted $\mathbf{v}_c(s, t)$, which is obtained by querying the Ocean Simulator GP at the location and time.

The robot velocity is assumed constant: $\mathbf{v}_r(a)$ = $||0.5||_2$ for all actions. Therefore the expected true action duration between state s and s' at time t is: $1500/||\mathbf{v}_r(a) + \mathbf{v}_c(s, t)||_1$ seconds. Under $(u, v) = [0, 0]$ currents, the expected duration of an action would be $1500 \times (0.5)^{-\frac{1}{2}} = 3000$ seconds, or 50 minutes. A positive u current is from the west. A positive v current is from the south.

Missions: We specify mission difficulty based upon the imposed time-bound between *easy* and *very* difficult, i.e. a scalar $\{2700, 2800, 2900, 3000\} \times ||\bar{s_k}, g||_1$ seconds respectively, where $|| \cdot ||_1$ refers to the L_1 distance between start state $\bar{s_k}$ and goal g. This distance is set to a minimum of 8 for these 30 missions. Each mission is performed for each difficulty setting.

TV-SMDP-GP: Observations are sequentially added to $\mathcal{D}$:
$\{((x + 1) - 2\sigma_i, (y + 1) - 2\sigma_j, t - \sigma_k), f(x, y, t)\}_{i=0}^{\omega}$, where $\sigma_i, \sigma_j, \sigma_k \sim \mathcal{U}(0, 1)$ and $\omega = 10$.

GP Belief: The multi-output GP belief $\hat{f} : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow Dist(\mathbb{R}^2)$ is queried for a belief distribution over over s_e, the perpendicular current vector u and v, for each new search node n', i.e. $\mathcal{N}(\hat{f}(s'_k, t) \mid \mu(s'_k, t), \Sigma(s'_k, t))$, where μ and Σ are specified in Equation (2). The kernel function k is a multi-output spatio-temporal kernel comprising of: $k_{\text{additive}} = k_{xy} + k_t + k_{lin}$, where:
$k_{xy} = \sigma_{xy}\text{exp}\left(-\frac{((x,y)-(x',y'))^2}{2l_{xy}^2}\right)$,
$k_t = \sigma_t\text{exp}\left(-\frac{(t-t')2}{2l_t^2}\right)$, and $k_{lin} = \sigma_{lin}^2((x, y, t)(x', y', t'))$.

We place prior Gamma distributions over all the GP hyperparameters, $\boldsymbol{\theta} = (l_{xy}, \sigma_{xy}, l_t, \sigma_t, \sigma_{lin})$:

$$p(l_{xy}) = \Gamma(1, 1), p(\sigma_{xy}) = \Gamma(1, 1), p(l_t) = \Gamma(10, 1), p(\sigma_t) = \Gamma(1, 1), p(\sigma_{lin}) = \Gamma(10, 1),$$

The TV-SMDP-GP action duration distribution Δ maintains the mean and variance of the belief distribution $\hat{f}$ over latent state features (as described in Section 5. This facilitates the choice of Δ to be a strictly positive, truncated-Normal distribution centered around the GP mean and variance: $\Delta(b, a, b', t) := \mathcal{N}(\mu(\hat{f}(s'_k, t')), \Sigma(\hat{f}(s'_k, t')))$, truncated from below, by min$= 1000$(s) or (16 minutes), where μ and Σ are defined as per Equation (2).

MCTS Parameters: number of known states = 6 × 6
number of trials $\tau = 1500$
additional trials $\tau' = 1500$
max trial length $\zeta = 100$
exploration weight $p = 0.9$
time epsilon (for considering search nodes equal) ϵ : 10^-6
rollout policy used: $\bar{\pi}(s_k) = \arg\min_{a \in A(s)} ||s'_k, g||$

Real World Ocean Dataset: Region 3 over 15 hours

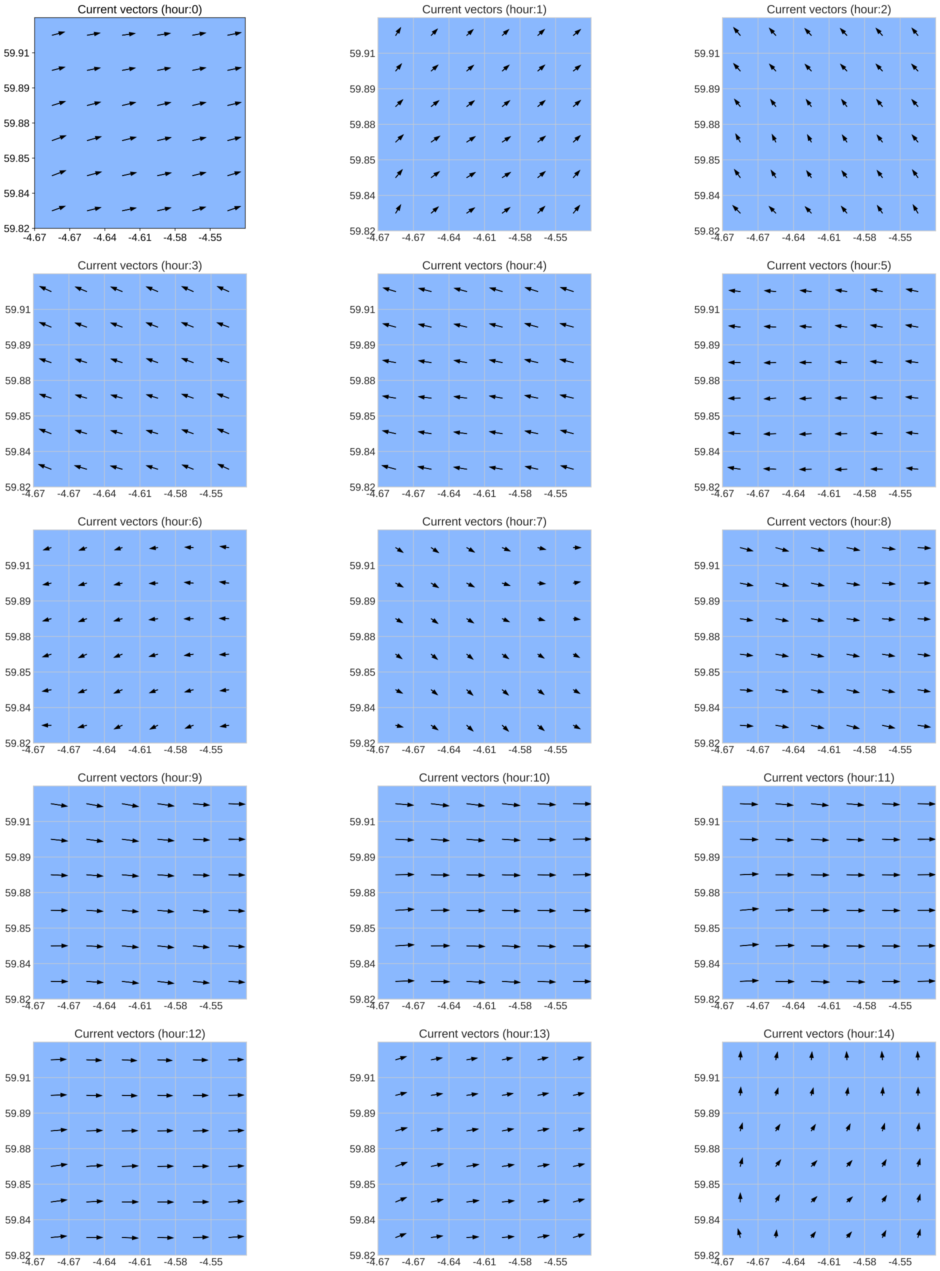

Figure 6: 15 hour images of Atlantic-European NW Shelf Dataset for Region 3: $59.8°$ to $59.9°$ latitude and $-4.67°$ to $-4.55°$ longitude.

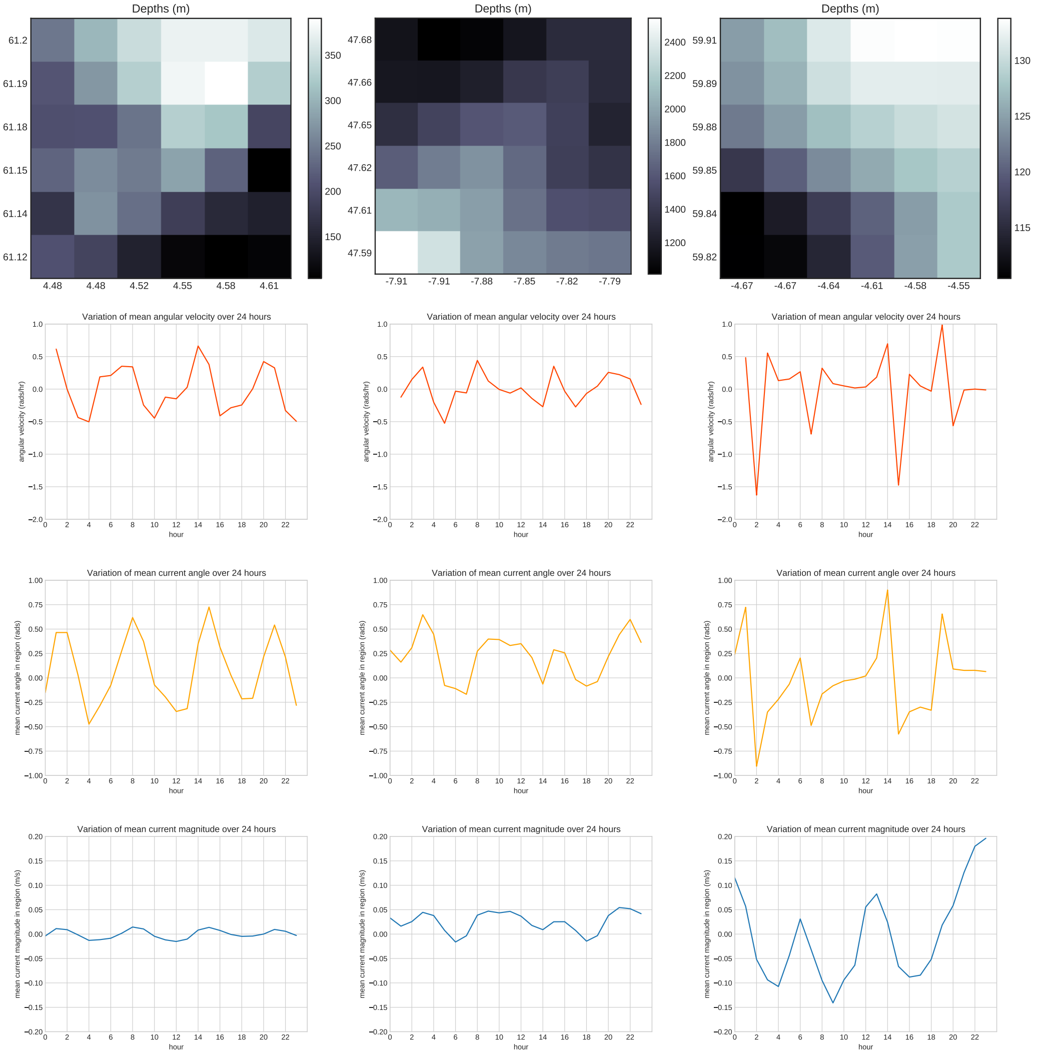

Figure 7: Atlantic-European NW Shelf Region 1, 2 and 3 from left to right. 1st row: Depth of region. 2nd row: Variation over time of average current angular velocity. 3rd row: Variation over time of average current angle. 4th row: Variation over time of average current magnitude.