Making Decisions with Spatially and Temporally Uncertain Data

Lantao Liu, Gaurav S. Sukhatme
Dept. of Computer Science
University of Southern California
{lantao.liu, gaurav}@usc.edu

Abstract—Taking advantage of known uncertainty patterns (e.g., uncertainty distributions) and strategically exploiting the uncertainty priors has been a prevalent methodology in the design of planning algorithms for autonomous systems that need to cope with various sources of uncertainty. However, in many scenarios the pattern of uncertainty is dynamic – it reveals new patterns varying with some underlying parameters. We envision that such “multi-dimensional” uncertainty dynamics represent a more general description of the uncertainty, but could introduce extra complexity in modeling and solving planning problems. Therefore, we propose a principled planning and decision-making framework that takes into account multiple dimensional uncertainty.

One relevant and concrete example is to deal with uncertainty that varies significantly both spatially and temporally. To incorporate such variability, in this paper we propose to generalize Markov Decision Processes, by incorporating the time-varying uncertainty and redesigning the fundamental value propagation mechanisms. We transform this decision-making method that considers only immediate and static uncertainty descriptions to an upgraded planning framework that is able to adapt to future time-varying uncertainty over some horizon.

I. INTRODUCTION

To illustrate the research challenges and impact of the proposed approach, consider a scenario where an underwater vehicle needs to spend a few weeks to navigate across a vast area of ocean to reach a goal location (underwater vehicles such as autonomous gliders can travel a long distance but move at speeds comparable to those of, or even slower than, typical ocean currents [63, 51]). Besides the obstacle regions, e.g., shores and shipping lanes which could result in collisions, the planning algorithms also need to take into account the ocean currents which vary both spatially and temporally. These currents are often strong enough to alter the vehicle’s motion significantly, as illustrated in Fig. 1. When planning for a long-range and long-term task, an accurate planner will require a forecast of ocean currents at the destination, as well as intermediate way-points, in order to determine suitable actions.

One drawback of state-of-the-art decision-making methodologies lies in the fact that, the basic model relies on a known uncertainty description that is “snapshot at a certain moment”[44, 40, 36]. Such frameworks may be used for online planning through “catching up” to the latest dynamics via a series of repetitive replanning processes. However, replanning strategies still do not take into account future time-varying uncertainty dynamics which can be beneficial if properly considered. In essence, we believe that the planning and decision-making community still lacks a general methodology to compute control policies that consider not only a fully known (current and past) stochastic description, but also a (possibly uncertain) prediction of future dynamics. Therefore, in this paper we investigate decision-theoretic planning under multi-dimensional uncertainty, and design an architecture with fundamentally new mechanisms that generalize existing decision-making methodologies. We validated our method in the scenario of ocean monitoring, and the results show that our approach leads to smaller travel distances and shorter travel times compared to state-of-the-art method.

II. RELATED WORK

Motion planning under uncertainty is an important functionality for autonomous robots operating in uncontrolled environments [37, 19, 60]. Many planning under uncertainty problems can be formalized in a decision-theoretic framework [6, 15] where the two main sources of uncertainty arise from imperfect sensors and actions. Comparing with deterministic path/trajecory planners (e.g., methods based on A* [17] or shortest path [64] or optimal control [5]) and many probabilistic motion planners (e.g., approaches based on RRT [24] or PRM [25]), an advantage of the decision-theoretic framework is that it exploits the stochastic structure of the world model and enforces rewards/penalties for outcomes of future actions. Markov decision processes (MDPs) have been

---

(a) Pattern of ocean currents
(b) Dec 20, 2015
(c) Dec 21, 2015
(d) Dec 22, 2015

Fig. 1. Ocean currents along the US west coast predicted by the Regional Ocean Modeling Systems (ROMS) [48].
developed as extremely useful conceptual models for understanding decision-theoretic planning [40, 4]. Building upon the MDP architecture, a partially observable MDP (POMDP) is a generalization of MDPs to situations where an agent cannot reliably identify the underlying environment state [23, 1, 45]. Such an extension dramatically increases the computational complexity, making exact solutions practically intractable as an agent could potentially need to account for all of the previous observations and actions rather than just the current state [39, 9, 32, 30]. One drawback of existing decision-theoretic methods lies in models’ reliance on known static transition dynamics and the lack of a principled methodology that considers not only a fully known stochastic description, but also a prediction of future dynamics.

Markov transition dynamics can be time-varying. Time-varying Markov models have been investigated for pattern analysis of economic growth [35, 3], which aims at understanding the dynamics of growth based on a collection of different states corresponding to different countries. However, these existing models have been constructed using Hidden Markov Models (HMMs), and assume that there is no action to control state transitions. Time-varying HMM-based models have also been employed for analyzing fiscal policies such as civilian spending and taxing [2], stock market [21], economic inflation [26], term structure of interest rates [34], as well as extreme temperature events [20]. Distinct from these methods, optimizing action policies while integrating a predictive transition model is an important characteristic, as well as a challenging objective, for robotics decision-making research.

Proximal works also include reinforcement learning (RL) [22, 55, 27]. In RL, an agent tries to maximize accumulated rewards by interacting with the environment during its life-time, where the environment is typically formulated as an MDP. RL methods differ from the classical MDPs in that they usually do not need knowledge about the MDP model and they target large MDPs where exact methods become slow or infeasible [11]. RL methods can be divided into two broad classes, model-based and model-free, which perform optimization in different fashions [12]. Model-based RL uses agent’s experience to construct an internal model of the transition dynamics and obtain immediate outcomes in the environment. Policies are then computed by searching or planning in this learned model [61]. In contrast, model-free RL uses experience to learn directly the state/action values or policies which can achieve the same optimal behavior but without estimation or use of a world model [52]. A technique related to future prediction is the temporal difference (TD) learning [54, 57]. More formally, TD learning is a technique where the learning agent learns to predict the expected value of a variable occurring at the end of a sequence of states. RL extends the TD technique by allowing the learned state-values to guide actions, and correct previous predictions based on availability of new observations [55, 7]. Additionally, in order to extend the discrete-time actions, a temporal abstraction based concept called options has been designed so that actions can be performed in a more flexible manner [53]. Nevertheless, although these learning techniques allow agents to learn rewards or transition dynamics while they are interacting with the environment, in general they neither consider time-varying transition dynamics nor have embedded predictive models into the basic algorithmic value propagation (iteration) processes.

The proposed method is very relevant to environmental monitoring scenarios. Environmental monitoring and sensing allow scientists to gain a greater understanding of the planet and its environmental processes [16]. Increasingly a variety of autonomous robotic systems including marine vehicles [18, 8, 58], aerial vehicles [62, 10, 50], and ground vehicles [59, 56, 43], are designed and deployed to measure environmental attributes related to, e.g., physical, chemical or biological parameters. Particularly, autonomous underwater gliders are advantageous in that they are capable of long-term ocean monitoring tasks over several weeks or months [46, 33, 38, 31]. Predictions of ocean currents have previously been used to improve the navigation capabilities of autonomous vehicles and improve the safety of their operation [29, 41, 49, 13]. While these prior works provide a basis for navigation, they are missing a principled analysis of the uncertainty of predictions.

Unlike typical decision-theoretic or RL based schemes employed in the spatio-temporal monitoring domains, in this work we investigate planning under multi-dimensional uncertainty dynamics and propose a novel model to address the issue. Specifically, we extend the basic decision-theoretic planning framework to a more generalized methodology by considering the stochastic transition dynamics that occur in both spatial and temporal dimensions.

III. Problem Description

Markov Decision Processes (MDPs) have served as an important basis for decision-making research in many domains. In this work, we assume that robots’ states are fully observable and use an MDP to model and analyze the uncertainty and decision structures. With appropriate adjustments, the underlying model can be extended to partially observable scenarios as long as the basic planning framework is completed and clear.

A. Preliminary: Standard Markov Decision Processes

Definition 3.1: An MDP $\mathcal{M}$ is defined by a 4-tuple $\mathcal{M} = \langle S, A, T, R \rangle$, where $S = \{s\}$ and $A = \{a\}$ represent the countable state space and action space, respectively. The stochastic transition dynamics, also known as the transition model, for an agent can be expressed as

$$T_a(s, s') = \Pr(s_{k+1} = s'|s_k = s, a_k = a)$$

which is a probability mass function that leads the agent to succeeding state $s_{k+1} = s'$ when it executes the action $a_k = a$ from state $s_k = s$. The time step $k$ is also known as the computing epoch. The forth element $R_a(s, s')$ in the tuple is a positive reward scalar for performing action $a$ on $s$ and reaching $s'$.

A control policy is a complete mapping $\pi : S \rightarrow A$ so that the agent applies the action $a_k \in A$ in state $s_k \in S$ at step $k$. 

$$T_a(s, s') = \Pr(s_{k+1} = s'|s_k = s, a_k = a)$$
If the action is independent of $k$, the policy is called stationary and $a_k$ is simply denoted by $a$.

**Note that, unlike conventional formulations, here we use $k$ instead of $t$ to index the algorithmic iterative steps/epochs, for both stationary and non-stationary models. In other words, we also regard the time-dependent non-stationary algorithm iterations as a momentary event, in order to distinguish the temporal process (evolution) which will be discussed in the remainder of this paper.**

Starting from state $s_0$ for a number of $k$ steps, let the sequence of future actions be $\{a_1, a_2, \cdots, a_k\}$ and the sequence of future states be $\{s_1, s_2, \cdots, s_k\}$; then the value (accumulated reward) for state $s_0$ can be expressed as

$$V(s_0) = \sum_{i=0}^{k} \gamma^i R_a(s_i, s_{i+1}). \quad (2)$$

If steps number $k$ is finite, $\gamma = 1$; otherwise $\gamma \in (0, 1)$ is a discount factor for discounting future rewards at a geometric rate. In this work, we are interested in stationary MDPs with infinite horizon (i.e., $k \to \infty$).

**Definition 3.2:** the Q-value of a state-action pair $(s, a)$ is defined as the the one-step look-ahead value of state $s$ if the immediate action $a$ is performed. More formally,

$$Q(s, a) = \sum_{s'\in S} T_a(s, s')(R_a(s, s') + \gamma V(s')). \quad (3)$$

The MDP problem is to find an optimal policy $\pi^*$ satisfying

$$V_{\pi^*}(s) \equiv V^*(s) = \max_{a \in A} Q(s, a), \forall s \in S. \quad (4)$$

When $\gamma < 1$, there exists a stationary policy that is optimal. In this case, $V^*$ is the unique solution to the Bellman equations:

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} T_a(s, s')(R_a(s, s') + \gamma V^*(s')). \quad (5)$$

From Eq. (5), the optimal action policy $\pi^*(s)$ can be obtained

$$\pi^*(s) = \arg \max_{a \in A} \sum_{s' \in S} T_a(s, s')(R_a(s, s') + \gamma V^*(s')). \quad (6)$$

Employing Bellman’s principle of optimality avoids enumerating solutions naively. In particular, dynamic programming based value iteration (VI) and policy iteration (PI) are the most widely used strategies for solving MDPs [40, 4].

**B. Challenge: A Two Dimensional Uncertainty Problem**

In many practical scenarios, e.g., marine vehicles in the ocean or aerial vehicles flying in the air, the transition model need not be static. Fig. 2(a) shows a spiral trajectory of an unpowered robot pushed by a spinning disturbance force field in a small local area. This scenario illustrates that the transition dynamics caused by environmental disturbances can vary over time, which consequently requires the control policy also to be a function of time in order to reject the time-varying disturbances.

In the remainder of this paper, we use a two-dimensional case to describe the essence of multi-dimensional transition dynamics.

**Definition 3.3:** The first dimension corresponds to the conventional transition model – the known/learned probability distribution that describes the agent’s uncertain outcomes. For example, in the underwater vehicle planning scenario, since the uncertain outcomes of each action are usually characterized spatially, thus, the first dimension in this example is based on spatial uncertainty.

**Definition 3.4:** The second dimension corresponds to the dynamic and uncertain change of the first dimension uncertainty. More formally, the MDP transition model is not necessarily static, and it may vary with some other parameters. In the underwater vehicle planning case, the transition model can vary with time and it can also be uncertain due to the changing ocean currents. Thus, the second dimensional dynamics here is the *time-varying transition model* (i.e., temporal dynamics of spatial dynamics).

Since a non-stationary MDP is dependent of time, we thus have first attempted to employ it to model time-varying transition dynamics. Specifically, future moments $t$ at which the transition dynamics and rewards are predicted are mapped/modelled as algorithmic epochs $k$ in a non-stationary MDP. Fig. 2(b) shows a result from field trials, from which we can observe that the policies form a few regions similar to the “optimization local minima”, which would trap the marine...
vehicle in dead-ends. With thorough analysis, we conclude with a few possible reasons:

- Although formulating the temporal dimension transition model as a time-dependent non-stationary MDP can integrate predicted future dynamics, the standard MDP solution method (e.g., value iteration) for iteratively propagating values mixed with both spatial and temporal dynamics becomes meaningless as it is hard to interpret underlying computing mechanisms. This could be the main reason for the “trapping” patterned policies depicted in Fig 2(b).

- The non-stationary property of MDPs corresponds to the finite-horizon of value propagation from a certain state. Modeling time-varying transition dynamics with a non-stationary MDP will therefore, limit the horizon of value propagation, and thus results in incomplete knowledge and impact of distant states. In other words, the non-stationary MDP used in this way will not only limit the temporal horizon, but also the spatial horizon, which suggests a highly suboptimal solution.

- The MDP is essentially a static offline method, but the future transition dynamics are changing and uncertain. A static offline model cannot effectively update and correct old predictions with new observations.

IV. TECHNICAL APPROACH

The analysis and concerns in Section III-B motivate us to design a decision-making framework that involves both spatial and temporal (more generally, multi-dimensional) transition dynamics.

A. Augment from “Spatial Only” to Both Spatial & Temporal

In order to update with temporal dynamics, existing approaches used in the MDP formulation but without modifying the MDP mechanism is to run the MDP solver at every time step after the agent reaches a new state (or whenever it is necessary, e.g., see [14]). However such an ad-hoc replanning approach is inefficient and can result in a highly sub-optimal solution as it only considers current but never future transition dynamics. Since the optimal values of all states together determine the optimal action policy \( \pi(s) \), \( \forall s \in S \), outdated values of distant states evaluated at an early period can mislead the agent towards a wrong direction from the global perspective. Imagine that the values are propagated from states far away – and equivalently the far future – but impact or even determine the agent’s immediate action at the current state and moment. However, the agent will need to spend time to reach a distant state, but the previously assumed transition dynamics may have changed before the agent arrives at that distant state. This indicates that the standard MDP model and its ad-hoc extensions are limited in their abilities in catching up to temporal dimensional transition dynamics.

Therefore, the MDP transition model \( T_a(s,s') \) needs to be a function of time, and we re-write it as \( T_a(s,s',t) \). Similarly, the reward function becomes \( R_a(s,s',t) \). Built on these changes, the value function is modified accordingly:

\[
V^*(s, t) = \max_{a \in A} \sum_{s' \in S} T_a(s, s', t) \left( R_a(s, s', t) + \gamma V^*(s', t') \right).
\]

Comparing the standard MDP (6) with Eq. (7), we see that every term of Eq. (7) is a function of time. We define this as a temporal-MDP: \( \mathcal{M}(t) = < S, A, T(t), R(t) > \).

**Remark 4.1:** A major difference between the temporal-MDP \( \mathcal{M}(t) \) and the conventional MDP \( \mathcal{M} \) lies in that, \( \mathcal{M}(t) \) is built on, and thus requires estimation of, transition time among states, which is different from MDP’s “state hopping” – an important property inherited from Markov Chains.

The transition time here denotes the travel time of a robot from a state \( s \) to another state \( s' \). For \( \mathcal{M}(t) \), standard solution methods such as value/policy iteration are not directly applicable due to the addition of temporal dimensional dynamics. Formally, although formulating the temporal dimension transition model as a time-dependent non-stationary MDP can utilize predicted dynamics, the standard iterative solution schemes (e.g., value iteration) for propagating values, which are mixed with both spatial and temporal dynamics, becomes meaningless. In other words, we can still use the old MDP framework to compute and obtain a solution, but the underlying computing mechanism cannot be interpreted cleanly as it lumps two different/dimensional transition dynamics together.

Consequently, solving \( \mathcal{M}(t) \) needs to incorporate future transition dynamics at correct moments in time. For instance, assume that at time \( t_0 \) the robot is situated at state \( s_0 \), and let \( t(s_0, s) \) be the state transition time from \( s_0 \) to an arbitrary state \( s \). Since the transition model is a function of time, when the robot eventually arrives at state \( s \), the transition dynamics that impact the robot is \( T_a(s, s', \tau) \) where \( \tau = t_0 + t(s_0, s) \) instead of some static \( T_a(s_0, s) \) – or equivalently \( T_a(s_0, s, t_0) \). Therefore, value evaluation/iteration for state \( s \) at the starting moment \( t_0 \) needs to utilize the transition model \( T_a(s, s', \tau) \) captured at \( \tau \). Fig. 3 depicts such an idea.

Note that, here we use a perfect prediction model that assumes the estimated times \( t(s_0, s) \) and \( \tau \) are accurate. This however, is unrealistic due to the agent’s stochastic behavior. This problem is addressed in the following subsections.
MDP is built on discrete state space $S$, whereas the robot motion is continuous. In continuous space, we re-define these variables:

- State $x$ is the counterpart of $s$ but in continuous state space. When $x$ coincides at $s$, we denote such state as $x(s)$ in continuous space. We can also map $x$ back to discrete space: $x \mapsto s$ if $||x - s||$ is less than space partition resolution;
- Local control reference $a(s)$ at $s$ is a vector and it is mapped from MDP’s action $a \in A$;
- Vector $d(s)$ expresses environmental/external disturbance at $s$.

Oftentimes the robot’s action $a(s)$ and external disturbance $d(s)$ are additive (e.g., forces, velocities) and produce a resultant/net vector $r(s) = a(s) + d(s)$ applied on the robot. We assume $a(s)$ and $d(s)$ contain noises subject to independent Gaussian distributions. Given a fixed time $T$, the robot's resultant state $x$ after applying $a(s)$ and being disturbed by $d(s)$ also follows a Gaussian distribution:

$$f_a(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right)$$

(8)

where $\mu$ and $\Sigma$ are the mean and covariance of $x$, respectively. It is worth mentioning that the MDP may produce multiple optimal actions (with equal optimal value) at some state. In such a case, a mixture distribution can be used,

$$f_{\{a_1, \ldots, a_k\}}(x) = \sum_{i=1}^{k} w_i f_{a_i}(x) = \frac{1}{k} \sum_{i=1}^{k} f_{a_i}(x),$$

(9)

where the weighting parameter $w_i$ for component PDFs are identical as actions have the same optimal value.

Let $\{a^*(s)\}$ be the set of optimal actions at state $s$, the transition dynamics thus can be expressed as

$$T_a(s, s', t) = \Pr(s'_t \mid s_t, \{a^*(s_t)\}, d(s_t)).$$

(10)

In practice, such discrete probability mass function is approximated by integrating Eq. (8) over discretized volumes.

### C. Transition Time Estimation

As mentioned earlier, temporal-MDP $\mathcal{M}(t)$ is parameterized with transition time, thus a good time estimation scheme is important. Specifically, from currently situated state $s_0$, the robot needs to estimate transition time to all other states. We develop an approach by taking two steps: the first step is to estimate “local” transition time from each state $s \in S$ to its one-hop succeeding states $\mathcal{N}(s) \subseteq S$ that $s$ can directly transit to, built on which the second step is to estimate the “global” transition time from robot’s current state $s_0$ to all other multi-hop states $S \setminus \mathcal{N}(s_0)$.

#### 1) Local One-Hop Transition Time Estimate:

Due to the stochastic nature of transition model, the robot situated at a state $s$ may eventually arrive at any one-hop succeeding state $s' \in \mathcal{N}(s)$. However, the estimation of transition time $t(s, s')$ is based on the assumption that the robot will reach a designated next state $s'$ with probability 1. To satisfy this assumption, we choose the action $\tilde{a}(s)$ with which the robot motion is toward $s'$. Let $\tilde{r}(s) = \tilde{a}(s) + d(s)$ denote the resultant of such selected action and environmental disturbance at $s$.

To simplify the calculation, one way is to transform the coordinate system such that the robot’s motion direction $\tilde{r}(s)$ is exactly on an arbitrary coordinate basis, built on which the multivariate PDF can be approximated to a univariate PDF by marginalization.

Fig. 4 shows an illustration with robot state $s$ located at the coordinate origin. There is an angle $\theta$ between $\tilde{r}(s)$ and a basis $x_1$. To transform the coordinate system such that $\tilde{r}(s)$ is on $x_1$, the coordinate needs to rotate with corresponding rotation matrix $R(\theta)$. More generally, with a rotation matrix $R$ and replacing $x = Rx$ in Eq. (8), we obtain a distribution in a transformed coordinate system:

$$f(x) \propto \exp\left(-\frac{1}{2}(x - R^{-1}\mu)^T R^T \Sigma^{-1} R(x - R^{-1}\mu)\right)$$

$$\propto \exp\left(-\frac{1}{2}(\tilde{x} - R^T \mu)^T (\tilde{\Sigma})^{-1}(\tilde{x} - R^T \mu)\right)$$

(11)

where $\tilde{\mu} = R^T \mu$ and $\tilde{\Sigma} = R^T \Sigma R$ after transformation.

Next we can calculate the expectation of resultant states in the robot’s motion direction. Formally, let $x_i$ denote a selected $i$-th basis (variable) in the direction of motion in the new coordinate system, and $x_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_d)$ represent all other variables. Then the conditional expectation
of \( x_i \) can be calculated by
\[
E(x_i|x_{-i} = 0) = \int x_i f(x_i|x_{-i}) I_{x_{-i} = 0}(x_{-i}) dx
\]
\[
= \int x_i f(x_i,x_{-i}) I_{x_{-i} = 0}(x_{-i}) dx
\]
\[
= \int_{0}^{\tau} x_i f(x_i,x_{-i}) I_{x_{-i} = 0}(x_{-i}) dx_{-i} dx_i
\]
where indicator variable \( I_{x_{-i} = 0}(x_{-i}) = 1 \) if \( x_{-i} = 0 \) and 0 otherwise. And \( \tau \) can be either \( +\infty \) (motion in the direction of \( x_i \)) or \( -\infty \) (motion against the direction of \( x_i \)).

Since states are obtained by applying \( r(s) \) for a fixed time \( T \), the transition time \( t(s,s') \) is approximated by
\[
t(s,s') = \frac{||s-s'||}{E(x_i|x_{-i} = 0)/T}
\]
where \( ||s-s'|| \) represents the translation (Euclidean distance) between the two states and the denominator is the estimated velocity given that the destination state is \( s' \).

2) **Global Multi-Hop Transition Time Estimate:**

Using the one-hop transition time, we can estimate the global multi-hop transition time \( t(s,s_c) \) from an arbitrary state \( s \) to a non-succeeding ending state \( s_c \in S \). The idea is to compute the first passing transition time at state \( s_c \). Note, the classic Markov Chain first-passage times method does not apply here because it only considers, and therefore only analyzes, the number/pattern of hops, instead of estimating the real non-identical travel time costs.

We formulate the problem in the fashion of Kolmogorov equations [28]. For example, from \( s_1 \), the transition time \( t(s_1,s_c) \) can be represented by an expectation \( E(t(s_1,s_{j(1)}^{(1)})) + t(s_{j(1)}^{(1)},s_c) \), where \( s_{j(1)}^{(1)} \in N(s_1) \) is \( s_1 \)'s one-hop succeeding states, and \( t(s_{j(1)}^{(1)},s_c) \) is again a multi-hop transition time from \( s_{j(1)}^{(1)} \) to ending state \( s_c \). Similarly, we can formulate transition time \( t(s_i,s_c) \) for all other \( s_i \in S \) to \( s_c \):
\[
t(s_1,s_c) = \sum_{\forall s_{j(1)}^{(1)} \in N(s_1)} T_a(s_1,s_{j(1)}^{(1)}), t(t(s_1,s_{j(1)}^{(1)}) + t(s_{j(1)}^{(1)},s_c))
\]
\[
t(s_n,s_c) = \sum_{\forall s_{j(n)}^{(n)} \in N(s_n)} T_a(s_n,s_{j(n)}^{(n)}), t(t(s_n,s_{j(n)}^{(n)}) + t(s_{j(n)}^{(n)},s_c))
\]
where \( t(s_c,s_c) = 0 \) and each \( t(s_i,s_{j(1)}^{(1)}) \) denotes the previously obtained one-hop transition time.

We have a total of \( |S| \) variables and \( |S| \) equations, which form a linear system. From the current state \( s_0 \), there are \( |S| - |N(s_0)| \) multi-hop non-succeeding states. Therefore, we need to solve \( |S| - |N(s_0)| \) linear systems in each of which we specify a different ending state \( s_c \in S \setminus N(s_0) \). In many decision-making scenarios, the number of one-hop succeeding states are limited, thus many coefficients in the linear system are zeros, which can be solved by sparse linear system solvers.

**D. Synergy of Processes from Spatial & Temporal Dimensions**

Finally, the estimated transition time need to be incorporated into the MDP's value iteration process. As described earlier, the transition dynamics \( T_a(s,s',t) \) are a function of time. Here the parameter \( t \) is exactly the transition time from the robot’s current state \( s_0 \) to a “to-be-evaluated” state \( s \), i.e.,
\[
T_a(s,s',t) = T_a(s,s',t(s_0,s)).
\]
Specifically, after each iteration of standard MDP value propagation, the action policy is updated. Based on the updated policy, the transition time estimates from current situated state to all other states \( t(s_0,s) \forall s \in S \) are calculated, which are then used to update the transition dynamics \( T_a(s,s',t(s_0,s)) \) at \( s \). The updated transition dynamics are in turn utilized for next value iteration and time estimate. The corresponding pseudo-code is described in Alg. 1.

**Algorithm 1: Value Iteration (with robot current state \( s_0 \))**

1. Initialize: \( k \leftarrow 1 \)
2. foreach \( s \in S \) do
   3. Initialize \( V_0(s) = 0, t(s_0,s) = 0 \)
   4. \( f^* \) value propagation in spatial dimension */
   5. foreach \( s \in S \) do
      6. \( \pi_k^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{N}(s)} T_a(s,s',t(s_0,s)) \cdot \left( R_a(s,s',t(s_0,s)) + \gamma V_{k-1}(s',t(s_0,s')) \right) \)
      7. update optimal action \( a^*(s) = \pi_k^*(s) \)
   8. \( f^* \) transition time estimates in temporal dimension */
   9. foreach \( s \in S \) do
      10. Estimate one-hop transition time \( t(s,s_j) \)
      11. Estimate multi-hop transition time \( t(s_0,s) \)
   12. \( k \leftarrow k + 1 \), goto Step 5.
13. Terminate if algorithm reaches given tolerance

In essence, the underlying computing mechanism can be imagined as value iterations that combine both spatial “expansion” and temporal “evolution”, which occur simultaneously in two dimensions. Specifically, the process in the spatial dimension corresponds to the standard MDP value iteration that proceeds on predicted transition dynamics; whereas the other process in the temporal dimension estimates and updates future transition dynamics based on action policies generated from the spatial process.

**E. Transition Dynamics Update and Correction**

The horizon of transition dynamics prediction can be limited. Also, estimation of transition time is uncertain and the uncertainty grows as the planning horizon increases. This implies that the proper way to reduce the propagated uncertainty is to correct the prior estimates after some period. In particular, while planning for long-range and long-term missions (e.g., in marine vehicle planning scenarios, the distance between
are available; (2) a shorter control horizon which will be updated whenever new predictions are available; (2) a shorter control horizon that is used to optimize robot’s motion. In our framework, a time step is the time window the robot transiting from one state to a succeeding state in the discrete state space, and we utilize the control horizon to “smooth” discrete actions. In greater detail, the MDP generates a set of actions, from which we map to a series of control/action references \{a_1, \cdots, a_n\} as described in Section IV-B. We observe that such an action sequence can cause drastic changes of motion when the robot switches from one state to another. Consequently, the SMPC mechanism smooths control inputs by minimizing deviations from both discrete states (trajectory way-points) and the action references subject to motion dynamics. More formally, let \(u_t\) denote the smoothed control input, then we have,

\[
\begin{align*}
\min_{\{u_{t_0}, \cdots, u_{t_{N-1}}\}} & \mathbb{E} \left( \sum_{t=t_0}^{t_{N-1}} l_t(x_t, u_t, d_t) \right) \\
\text{s.t.,} & \quad x_{t+\Delta t} = h(x_t, u_t, d_t), \\
& \quad x_{t_0} = x(s_0),
\end{align*}
\]

(16)

where \(s_0\) and \(t_0\) are the robot’s current state and time, respectively; \(N\) denotes the control horizon; \(\Delta t = t_i - t_{i-1}\) \((1 \leq i \leq N)\) represents the transition time to the expected next state, and

\[
l_t(x_t, a_t, d_t) = \| h(x_t, u_t, d_t) - x_t(s) \|^2 + \| u_t - a_t(s) \|^2
\]

(17)

measures the deviations from discrete states and action references. State \(s\) denotes the expected one-hop succeeding state resulted from current optimal action and environmental disturbance.

Since the objective of Eq. (16) is an expectation, the problem can be solved by dynamic programming as well:

\[
V(x_t) = \min_u l_t(x_t, u_t, d_t) + \mathbb{E} \left( V(x_{t+\Delta t}) \right)
\]

(18)

The control horizon \(N\) usually is a small number compared to the state size, solving Eq. (18) thus requires a small computational effort. Eq. (16) produces a series of control inputs \(u_{t_0}, \cdots, u_{t_{N-1}}\), but only the first one is implemented (i.e., used as local control reference). The process is repeated until the specified goal state is reached.

V. EXPERIMENTS

We validated our method in the scenario of ocean monitoring, where the ocean currents vary both spatially and temporally. A simulator written in C++ was built in order to test the proposed decision-making framework. The robot used in simulation is an underwater glider with a simplified kinematic model. The simulation environment was constructed as a two dimensional ocean surface, and we tessellated the environment into a grid map. We represent the center of each cell/grid as a state, where each non-boundary state has a total of nine actions, i.e., a non-boundary state can move in the directions of N, NE, E, SE, S, SW, W, NW, plus an idle action (returning to itself). Time varying ocean currents are external disturbances for the robot and are represented as a vector field. Specifically, vector \(\mathbf{u}(x_t)\) denotes the easting velocity component (along latitude axis) and vector \(\mathbf{v}(x_t)\) denotes the northing component (along longitude axis) and \(t\) represents time. The disturbance at \(x_t\) can thus be written as \(d(x_t) = \mathbf{u}(x_t) + \mathbf{v}(x_t)\). As mentioned earlier, a continuous state \(x\) can be mapped to a discrete tessellated state \(s \in S\) within tessellation resolution.

We consider two independent sources of uncertainty: (1) uncertainty of robot motion \(a(x_t) \sim \mathcal{N}(\mu_a, \Sigma_a)\), and (2) uncertainty of predicted ocean current disturbance \(d(x_t) \in \mathcal{N}(\mu_d, \Sigma_d)\). Thus the resultant vector \(r(x_t) = a(x_t) + d(x_t)\) applied on the robot is also subject to Gaussian distribution \(\mathcal{N}(\mu_a + \mu_d, \Sigma_a + \Sigma_d)\).

We first investigated the policy patterns generated from this proposed framework with two dimensional transition dynamics. While considering only robot action and its uncertainty (no external disturbance), Fig. 5(a) shows a policy map (red arrows) on a funnel-like time-surface where the color-map indicates estimated transition times from the robot state (bottom of the funnel). Fig. 5(b) and 5(c) are projected policy maps onto a 2D plane (ocean surface). A brighter region implies larger chance of being visited by the robot, and the difference of brighter regions in two figures reveals differing “magnitudes” of uncertainty (Fig. 5(b) has larger action uncertainty than Fig. 5(c)).

We tested our approach and the standard MDP with artificially created time-varying disturbances. Fig. 6 shows policies on a 10×10 grid map with a spinning vector field where each vector rotates with certain rate. Fig. 6(a) and Fig. 6(b) are policies generated from standard MDP and the presented temporal-MDP, respectively. Fig. 6(c) shows a predicted resultant vector field (i.e., \(r(s) = a(s) + d(s), \forall s \in S\)) for temporal-MDP, from which we can observe a pattern similar to the policy map in Fig. 6(a) that is produced from the standard MDP. This also indicates that, after a synergistic operation of spatial and temporal transition dynamics, the temporal-MDP “eliminates” the temporal dimensional dynamics and produces results similar to those of standard MDP with only spatial (one dimensional) transition dynamics.

We validated the method with a larger number of states and differing disturbance patterns. Fig. 7 shows a set of planning results, where the top two figures are results for a spinning vector field of disturbances, and the bottom two ones are disturbed by a vortex-like vector field with translating vortex center. Fig. 7(a) and 7(c) are trajectories from standard MDP, and Fig. 7(b) and 7(d) are results from the temporal-MDP method (here we assume the prediction horizon is sufficiently large).
Fig. 5. (a) Temporal-MDP policy (red arrows) on a funnel-like time-surface where the color-map indicates estimated transition time from the robot’s state (bottom of the funnel); (b)(c) Projected 2D policy maps with different magnitudes of action uncertainty. Red target symbol is the goal state to reach. (a) Temporal-MDP policy (red arrows) on a funnel-like time-surface where the color-map indicates estimated transition time from the robot’s state (bottom of the funnel); (b)(c) Projected 2D policy maps with different magnitudes of action uncertainty. Red target symbol is the goal state to reach.

Fig. 6. Policies under time-varying disturbances (disturbances are not shown in the figure) which is a spinning vector field $d(x, y_t) = R(\omega t)d_t$, where $R(\omega t)$ is the rotation matrix with rotating rate $\omega$ and $d$ is a constant value. (a) Policy map of standard MDP; (b) Policy map of temporal-MDP; (c) Predicted resultant vector $r(s) = a(s) + d(s)$ for temporal-MDP, which reveal similar pattern to the policy map of standard MDP.

We can observe that the trajectories of our method are much smoother and shorter.

Statistics with regard to trajectory lengths and time costs are provided in Fig. 8(a) and 8(b). These results indicate that the temporal-MDP method leads to smaller travel distances and shorter travel times. We use the Eigen iterative sparse matrix solver to compute linear systems for estimating multi-hop transition time. Fig. 8(c) shows that our method requires $\sim 15$ seconds for $\sim 1000$ states and $\sim 50$ seconds for $\sim 2000$ states (on a desktop with a 3.30GHz Intel i7 CPU).

We also tested the algorithms with real ocean current data. The simulator is able to read and process data from the Regional Ocean Model System (ROMS) which predicts/forecasts ocean currents up to 72 hours in advance. This allows us to utilize these ocean predictions to model the temporal dimensional transition dynamics. However, ROMS provides eight datasets for one day (every three hours). This means the data is not time-continuous. To address this, we use Gaussian Processes Regression (GPR) to interpolate and extrapolate the missing parts. Fig. 9 shows results from running the MDP and our method. Since the task spans multiple days, we set the prediction horizon of our method as three hours, i.e., whenever new prediction data is available. The results reveal that this presented approach saves both time and energy.

VI. CONCLUSION AND FUTURE WORK

Markov Decision Processes as state-of-the-art decision-making methods are based on static and momentary stochastic transition dynamics, which lead to sub-optimal solutions if the transition dynamics are varying with other parameters. Motivated by decision-making problems in an environmental spatio-temporal monitoring context, we presented a generalized MDP framework that takes into account a transition model that varies both spatially and temporally, and developed mechanisms to estimate newly introduced temporal parameters as well as integrate dynamic processes of different dimensions. We validated our method with various disturbances including...
those from real ocean data. The results show that our approach saves both time and energy compared to state-of-the-art methods. It is worth mentioning that, although we built a model by incorporating temporal parameters, in general the model can be constructed analogously for any other new parameters and corresponding dimension augmentations. In the future, we plan to test the method in field trials and extend the current algorithmic architecture to other scenarios (and dimensions).

REFERENCES

[1] Douglas Aberdeen and Jonathan Baxter. Scalable Internal-State Policy-Gradient Methods for POMDPs. In Proceedings of the Nineteenth International Conference on Machine Learning. ICML ’02, pages 3–10. Morgan Kaufmann Publishers Inc., 2002.

[2] Luca Agnello, Gilles Dufnott, and Ricardo Sousa. Using time-varying transition probabilities in markov switching processes to adjust us fiscal policy for asset prices. Economic Modelling, 34(C):25–36, 2013.

[3] Marco Bazzi, Francisco Blasques, Siem Jan Koopman, and Andr Lucas. Time varying transition probabilities for markov regime switching models. Tinbergen Institute Discussion Papers 14-072/III, Tinbergen Institute, 2014.

[4] Dimitri P. Bertsekas. Dynamic Programming and Optimal Control. Athena Scientific, 2nd edition, 2000. ISBN 1886529094.

[5] J. E. Bobrow. Optimal robot path planning using the minimum-time criterion. IEEE Transactions on Robotics & Automation, 4(4):443–450, 1988.

[6] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. Journal of Artificial Intelligence Research, 11:1–94, 1999.

[7] Justin Boyan. Least-squares temporal difference learning. In In Proceedings of the Sixteenth International Conference on Machine Learning, pages 49–56. Morgan Kaufmann, 1999.

[8] R. Camilli, B. Bingham, M. Jakuba, H. Singh, and J. Whelan. Integrating in-situ chemical sampling with auv control systems. In IEEE/MTS OCEANS Conf., pages 101–104, 2004.

[9] Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 963–972, 1996.

[10] I. Colomina and P. Molina. Unmanned aerial systems for photogrammetry and remote sensing: A review. Journal of Photogrammetry and Remote Sensing, 92:79 – 97, 2014.

[11] Peter Dayan and Nathaniel D. Daw. Decision theory, reinforcement learning, and the brain. Cognitive, affective & behavioral neuroscience, 8(4):429–453, 2008.

[12] Peter Dayan and Yael Niv. Reinforcement Learning: The Good, The Bad and The Ugly. Current Opinion in Neurobiology, 18(2):185–196, 2008.

[13] Arvind A. de Menezes Pereira, Jonathan Binney, Burton H. Jones, Matthew Ragan, and Gaurav S. Sukhatme. Toward risk aware mission planning for autonomous underwater vehicles. In International Conference on Intelligent Robots and Systems, pages 3147–3153, Sep 2011.

[14] Arvind A. de Menezes Pereira, Jonathan Binney, Geoffrey A. Hollinger, and Gaurav S. Sukhatme. Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. Journal of Field Robotics, 30(5):741–762, Oct 2013.

[15] Richard Dearden and Craig Boutilier. Abstraction and approximate decision-theoretic planning. Artificial Intelligence, 89(12):219 – 283, 1997.

[16] M. Dunbabin and L. Marques. Robotics for environmental monitoring: Significant advancements and applications. IEEE Robot. Autom. Mag., 19(1):24 – 39, 2012.

[17] David Ferguson, Maxim Likhachev, and Anthony (Tony) Stentz. A guide to heuristic-based path planning. In Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems. International Conference on Automated Planning and Scheduling (ICAPS), June 2005.

[18] Edward Fiorelli, Pradeep Bhatia, and Naomi Ehrich Leonard. Adaptive sampling using feedback control of an autonomous underwater glider fleet. In Proc. 13th Int. Symposium on Unmanned Untethered Submersible Tech, pages 1–6, 2003.

[19] Carlos Ernesto Guestrin. Planning Under Uncertainty in Complex Structured Environments. PhD thesis, Stanford, CA, USA, 2003. AAI3104233.

[20] Reza Hosseini, NhuD. Le, and JamesV. Zidek. Time-varying markov models for binary temperature series in agrorisk management. Journal of Agricultural, Biological, and Environmental Statistics, 17(2):283–305, 2012.

[21] Tokunaga T. Isogai A., Kanoh S. An extension of the markov-switching model with time-varying transition probabilities: Bull-bear analysis of the japanese stock market. Technical report, Hitotsubashi University, 2004.

[22] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: a survey. Journal of Artificial Intelligence Research, 4:237–285, 1996.

[23] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. Artificial Intelligence, 101:99–134, 1998.

[24] S. Karaman, M.R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime Motion Planning using the RRT”. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), May 2011.

[25] Lydia Kavraki, Petr Svestka, Jean Latombe, and Mark Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Technical report, Stanford, CA, USA, 1994.

[26] Rim Khemiri and Mohamed Sami Ben Ali. Exchange rate pass-through and inflation dynamics in tunisia: A markov-switching approach. Economics: The Open-Access, Open-Assessment E-Journal, 7(2013-43), 2013.

[27] J. Kober, J. Andrew (Drew) Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. International Journal of Robotics Research, July 2013.

[28] Andrey N. Kolmogorov. Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung. Mathematische Annalen, 104(1):415–458, 1931.

[29] Dov Kruger, Rustam Stolkin, Aaron Blum, and Joseph Briganti. Optimal auv path planning for extended missions in complex, fast-flowing estuarine environments. In ICRA, pages 4265–4270, 2007.

[30] Hanna Kurniawati, Yanzhu Du, David Hsu, and Wee Sun Lee.
Motion planning under uncertainty for robotic tasks with long time horizons. Int. J. Rob. Res., 30(3):308–323, March 2011.

N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. Journal of Field Robotics, 27(6):718–740, 2010.

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence, AAAI ’99/IAAI ’99, pages 541–548, 1999.

Travis Miles, Sang Hoon Lee, Anna Whlin, Ho Kyung Ha, Tae Wan Kim, Karen M. Assmann, and Oscar Schofield. Glider observations of the dotson ice shelf outflow, Deep Sea Research Part II: Topical Studies in Oceanography, 2015.

Alain Monfort and Fulvio Pegoraro. Switching varma term structure models - extended version. Working papers, Banque de France, 2007.

Bruno Morier and Vladimir Teles. A time-varying markov-switching model for economic growth. Textos para discusso 305, Escola de Economia de So Paulo, Getulio Vargas Foundation (Brazil), 2011.

Gergely Neu, András György, Csaba Szepesvári, and András Antos. Online markov decision processes under bandit feedback. In Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems, pages 1804–1812, 2010.

Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. Int. J. Rob. Res., 29(8):1053–1068, 2010.

D. A. Paley, F. Zhang, D. M. Fratantoni, and N. E. Leonard. Glider control for ocean sampling: The glider coordinated control system. IEEE Transaction on Control System Technology, 16(4):735–744, 2008.

Christos Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. Math. Oper. Res., 12(3):441–450, August 1987.

Martin L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., 1st edition, 1994.

Clment Pflrs, Yan Paillhas, Pedro Patrn, Yvan Petillot, Jonathan Evans, and David Lane. Path planning for autonomous underwater vehicles. IEEE Transactions on Robotics, 23(2):331–341, 2007.

S.Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. Control Engineering Practice, 11(7):733 – 764, 2003.

Matteo Reggente, Alessio Mondini, Gabriele Ferri, Barbara Mazzolai, Matteo Gabelletti, Paolo Dario, and Achim J. Lilienthal. The dustbot system: Using mobile robots to monitor pollution in pedestrian area. Chem. Eng., 23:273–278, 2010.

Sheldon M. Ross. Introduction to stochastic dynamic programming. Probability and mathematical statistics. Academic Press, New York, 1983.

Nicholas Roy. Finding Approximate POMDP solutions Through Belief Compression. Journal of Artificial Intelligence Research, 23, 2005.

Oscar Schofield, Scott Glenn, Robert Chant, Mark A. Moline, Paul Bissett, and John Wilkins. The evolution of a nearshore coastal observatory and the establishment of the new jersey shelf observing system. 44(11):52–58, 2003.

Suresh Sethi and Gerhard Sorger. A theory of rolling horizon decision making. Ann. Oper. Res., 29(1-4):387–416, 1991.

Alexander F. Shchepetkin and James C. McWilliams. The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. Ocean Modelling, 9(4):347–404, 2005.

Ryan N. Smith, Yi Chao, Peggy P. Li, David A. Caron, Burton H. Jones, and Gaurav S. Sukhatme. Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model. Int. J. Rob. Res., 29(12):1475–1497, 2010.

Ryan N. Smith, Arvind Pereira, Yi Chao, Peggy P. Li, and Gaurav S. Sukhatme. Autonomous underwater vehicle trajectory design coupled with predictive ocean models: A case study. In Proceedings of The IEEE International Conference on Robotics and Automation., 2010.

Ryan N. Smith, Mac Schwager, Stephen L. Smith, Burton H. Jones, Daniela Rus, and Gaurav S. Sukhatme. Persistent ocean monitoring with underwater gliders: Adapting sampling resolution. Journal of Field Robotics, 28(5):714 – 741, 2011.

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. Pac model-free reinforcement learning. In Proceedings of the 23rd International Conference on Machine Learning, pages 881–888, 2006.

Richard Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112:181–211, 1999.

Richard S. Sutton. Learning to predict by the methods of temporal differences. In MACHINE LEARNING, pages 9–44. Kluwer Academic Publishers, 1988.

Richard S. Sutton and Andrew G. Barto. Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

Onur Tekdas, Volkan Isler, Jong Hyun Lim, and Andreas Terzis. Using mobile robots to harvest data from sensor fields. Wireless Commun., 16(1):22–28, 2009.

Gerald Tesauru. Temporal difference learning and td-gammon. Commun. ACM, 38(3):58–68, 1995.

Pratap Tokekar, Deepak Bhadaura, Andrew Studenski, and Volkans Isler. A robotic system for monitoring carp in minnesota lakes. J. Field Robotics, 27(6):779–789, 2010.

Marco Trincavelli, Matteo Reggente, Silvia Coradeschi, Amy Loutfi, Hiroshi Ishida, and Achim J. Lilienthal. Towards environmental monitoring with mobile robots. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2210–2215, 2008.

Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. Int. J. Rob. Res., 31(11):1263–1278, September 2012.

Thomas J. Walsh, Sergiu Goschin, and Michael L. Littman. Integrating sample-based planning and model-based reinforcement learning. In AAAI, AAAI Press, 2010.

Adam C. Watts, Vincent G. Ambrosia, and Everett A. Hinkley. Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use. Remote Sensing, 4(6):1671–1692, June 2012.

Russell B. Wynn, Veerle A.I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brian J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous underwater vehicles (auvs): Their past, present and future contributions to the advancement of marine geoscience. Marine Geology, 352:451 – 468, 2014.