Weathering Ongoing Uncertainty: Learning and Planning in a Time-Varying Partially Observable Environment

Gokul Puthumanaillam, Xiangyu Liu, Negar Mehr and Melkior Ornik

Abstract— Optimal decision-making presents a significant challenge for autonomous systems operating in uncertain, stochastic and time-varying environments. Environmental variability over time can significantly impact the system's optimal decision making strategy for mission completion. To model such environments, our work combines the previous notion of Time-Varying Markov Decision Processes (TVMDP) with partial observability and introduces Time-Varying Partially Observable Markov Decision Processes (TV-POMDP). We propose a two-pronged approach to accurately estimate and plan within the TV-POMDP: 1) Memory Prioritized State Estimation (MPSE), which leverages weighted memory to provide more accurate time-varying transition estimates; and 2) an MPSE-integrated planning strategy that optimizes long-term rewards while accounting for temporal constraint. We validate the proposed framework and algorithms using simulations and hardware, with robots exploring a partially observable, time-varying environments. Our results demonstrate superior performance over standard methods, highlighting the framework's effectiveness in stochastic, uncertain, time-varying domains.
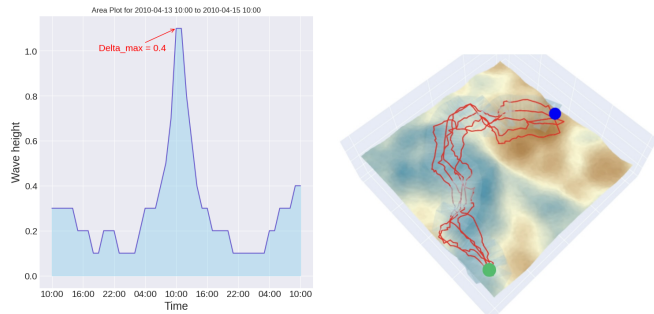
I. INTRODUCTION

Consider an autonomous marine robot deployed to traverse an unfamiliar environment over an extended period of time. Such a task is often complicated by the environmental conditions that are subject to time-varying forces such as waves, tides or thermohaline circulation [1]–[3]. Given that the environment is subject to change, it is often reasonable to assume that these changes occur at a relatively slow pace as illustrated in Fig. 1(a). These features can generate significant uncertainty whose effects on the robot's dynamics are challenging to accurately predict or model [4], [5]. Consequently, the robot's dynamics can be considered unknown, time-varying and stochastic.

While POMDPs provide a framework for sequential decision making under uncertainty, the introduction of time-variability in POMDP is nontrivial. As [6] highlights, the challenge lies in the fact that time is unidirectional and incorporating time into state space leads to an explosion in the size of the state space, dramatically increasing computational complexity. To address this, we propose *Time-Varying POMDPs* (TV-POMDPs) that represent transitions as dynamic probability functions avoiding state space explosion.

(a) Area plot depicting the variation in wave heights (disturbance) over a 48-hour period. The data is part of a dataset that includes various environmental factors [7]. The red arrow annotation indicates the maximal rate of change ($\Delta_{\max}$) in wave height.

(b) The trajectory of an Unmanned Surface Vehicle (USV) from Sec V-A attempting to navigate from a starting point (green dot) to a goal point (blue dot). The USV's route (red lines) visibly adjusts in response to the changing wave heights.

Fig. 1: Time-varying environments and their effects.

The key implication is that the robot's dynamics at any given time depends on the current state and the temporal context. The effects of an action taken in a particular state may have different outcomes depending on when it is executed. This makes it challenging to learn policies for sequential decision making and control, since past experience may become obsolete under time-variation. As shown by [6], a basic Bellman backup mechanism [8] is not sufficient. This presents a twofold challenge: accurately estimating the time-varying dynamics of the system and effectively planning the system's actions in response to this dynamic uncertainty. We propose *Memory Prioritized State Estimation* (MPSE), a strategy which hinges on the understanding that in an environment characterized by incremental change, historical data still holds considerable value to learn the transition function — there exists a substantial correlation between past observations, recent observations and the current state.

A. Related Works

Classical approaches like MDPs [9] and POMDPs [10] offer valuable foundations for modeling dynamics and decision-making under uncertainty. However, these models assume stationary transition probabilities, which are not suitable for time-varying environments.

Time-Dependent MDPs (TDMDPs) [11] encode time as a component in the state space. However, analysis [12] shows that solving TDMDPs can be computationally intractable, limiting real-world use. Semi-MDPs [13] allow only transition times to vary while keeping probabilities stationary.

ϵ-stationary MDPs [14], [15] only permit small variations in probabilities, insufficient for dynamic environments.

An approach better suited to time-varying environments is Time-Varying MDPs (TVMDPs) [6], that accommodates spatial and temporal stochasticity in transitions. However, they rely on MDPs that are known a priori. Further work [16] extends this to learn and plan in unknown TVMDPs. A key limitation is that these do not incorporate partial observability, which is ubiquitous in real-world settings.

Learning in time-varying environments poses significant challenges. Stationary RL methods [17] [18] require extensive training episodes to converge to optimal behavior. However, in time-varying settings, the environment continues to evolve during training data collection, invalidating previously learned policies [19], [20]. Online RL algorithms [21] adapt policies continuously but naively updating estimates risks decreased performance if past experiences become obsolete under time-variation [22], [23].

Retaining pertinent memories is key for estimation and planning in time-varying environments [24]. Memory-based approaches [25], [26] can outperform Monte Carlo [27] methods but resetting memory when dynamics change is unsuitable for time-variation. Selective experience replay [28] and memory attention [29] allow replaying pertinent experiences. However, these may still replay outdated experiences if temporal relevance is not considered.

Our work builds on these approaches by adaptively prioritizing observations based on temporal relevance, facilitating estimation and planning under time-varying, partial observability.

B. Contributions

In this paper, we introduce a Time-Varying Partially Observable Markov Decision Process (TV-POMDP) framework to model time-varying stochastic dynamics. We propose Memory Prioritized State Estimation (MPSE) to facilitate estimation and planning within this framework. MPSE selectively prioritizes observations based on correlation and recency. We derive a modified Maximum Likelihood Estimate using these prioritized samples to optimize state estimation. Additionally, we present a planning strategy tailored to leverage the identified time-varying transitions to improve action selection. We demonstrate our approach in real-world and simulated experiments, considering an autonomous marine vehicle traversing waypoints and a robot navigating stochastic, time-varying environments. Our results show improved state estimation accuracy and superior planning performance, highlighting the effectiveness of our proposed method.

II. PRELIMINARIES

A. Partially Observable Markov Decision Process

A Partially Observable Markov Decision Process (POMDP) [10] is an extension of the classic Markov Decision Process (MDP) that is used to model decision-making problems in which the agent cannot directly observe the underlying state of the system. It is represented by the tuple (S,A,Z,T,O,R,γ), which denotes state space, action space, observation space,

state transitions, observation functions, rewards, and the discount factor, respectively.

The agent uses a *belief state*, which is the probability distribution over the state space S and is updated as the agent interacts with the environment and receives observations. The agent's aim within a POMDP is to determine an optimal policy π^*, optimizing the expected cumulative rewards.

III. TIME-VARYING PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

POMDPs model the environment as a stochastic process for optimal decision-making, assuming stationary transition probabilities. However, real-world dynamics like those of the autonomous marine robot that was introduced earlier are often time-varying due to varying factors like winds and waves. Fig. 1(b) provides an illustration of a scenario where the robot's transition function is subject to temporal variations. To better model such scenarios, we propose an extension to the traditional POMDP framework.

Our modified POMDP retains the original state and action spaces, avoiding inflation in computational complexity. The TV-POMDP model is represented by the tuple $(S, A, Z, T_t, O, R_t, \gamma, b_0)$ capturing state, action, and observation spaces, time-varying transition and reward functions, the observation function, the discount factor, and the initial belief state at time t_0. The *time-varying transition probability*, denoted by $T_t(s,a,s')$, defines the probability of transitioning from state s to state s' under action a at time t.

Similarly, the reward function is reformulated as $R_t(s,a)$ which gives the immediate reward for taking action a in state s at time t. The value function V, traditionally representing the cumulative discounted return from belief state b, now incorporates the temporal dimension — $V_t(b)$ gives the expected future discounted reward from belief state b at time t. The value function $V_t(b)$ is given by:

$$V_t(b) = \max_{a \in A} \sum_{s \in S} b(s) \left[R_t(s,a) + \gamma \sum_{s' \in S} T_t(s',s,a) V_{t+1}(b') \right],$$
(1)

where γ is the discount factor and $V_{t+1}(b')$ is the value of belief state b' at future time $t+1$.

IV. LEARNING AND PLANNING IN A TV-POMDP

The TV-POMDP formulation captures the temporal dynamics of the environment and acknowledges that the same action taken in the same state can lead to different outcomes at different points in time, i.e., *non-repeatability* of outcomes. For instance, the tides might be stronger at certain times of the day compared to others — negating the possibility of the robot coming back to the same state and learning from multiple runs. This feature necessitates the development of an online learning approach.

However, online learning in the described context has its own set of intricacies. If the transition probabilities were independent, i.e., the observations at time $t-1$ do not impact the estimate of transition probabilities for time t, it is meaningless to attempt to learn the transition probability function. Thus

learning needs to leverage some structural characteristics. In many practical domains, there exists some innate knowledge regarding the maximal rate of environmental changes, Δ_{max},

$$|T_t(s',a|s) - T_{t-1}(s',a|s)| \leq \Delta_{\max}.$$

We leverage this underlying structure to serve as a foundation for the agent's learning strategy, guiding its process of estimating and planning in such time-varying environments. Our problem, thus, is a two-fold one:

1) Online learning: Estimate T_t in a single run by interacting with the environment, leveraging the knowledge of its maximal rate of change.
2) Optimal decision making: Using T_t, devise an optimal policy π^* that maximizes the future rewards.

A. Memory Prioritized State Estimation

In order to learn and estimate a transition probability function, we need to rely on information in form of observations from the environment. Processing all past observations is computationally expensive, so we focus on *windows* of recent observations with the premise that, compared to individual observations, these windows can provide a comprehensive view of the time-varying dynamics at play. The size of the time window is empirically chosen, and the method is generally not very sensitive to the window size as long as it is within a range that captures the temporal patterns.

1) Memory Prioritization: Not all observations carry equal significance. Some might offer better insights into the temporal patterns and transitions in the environment, while others might be relatively uninformative, imperfect or incomplete. We propose a prioritizing scheme that assigns a weight to each sample based on its potential *informativeness*. This weight is determined by a combination of factors:

1) *Autocorrelation*: Quantifies the degree to which observations are dependent on each other, measuring serial dependence. For a series of observations $z = (z_1, z_2, ..., z_n)$, the autocorrelation [30] at lag k is:

$$\rho(k) = \frac{\sum_{t=k+1}^{n}(z_t - \bar{z})(z_{t-k} - \bar{z})}{\sqrt{\sum_{t=1}^{n}(z_t - \bar{z})^2 \sum_{t=1}^{n}(z_{t-k} - \bar{z})^2}}, \quad (2)$$

We define the autocorrelation score A_s by

$$A_s(z_i) = \rho(t - t_i). \quad (3)$$

$A_s(z_i)$ captures how similar the dynamics of the environment are at time t to those at time t_i when z_i was observed. High value of $A_s(z_i)$ suggest that z_i is pertinent and can provide insight for decision-making at t.

2) *Recency*: The value of an observation often diminishes over time, captured by a recency score R_s. For an observation z_i at time t_i, R_s is formulated as

$$R_s(z_i) = \frac{1}{t - t_i + \varepsilon}, \quad (4)$$

where t is the current time and ε prevents division by zero. R_s quantifies the freshness of an observation. Higher R_s means the observation is more recent and more likely to be relevant for the current state of the system.

3) *Deviation*: Observations that differ significantly from the mean, signifying critical transitions.

$$D_s(i) = |z_i - \bar{z}_t|. \quad (5)$$

Here, D_s quantifies the extent to which an observation z_i deviates from the mean $\bar{z}_t$ within its time window. While higher D_s values can indicate both informative observations and outliers, the inclusion of autocorrelation score reduces the potential impact of outliers, allowing D_s to be less influential on the overall estimation.

Each observation z_i is assigned a combined weight ω_i, which is a weighted sum of these factors:

$$\omega_{z_i} = w_a A_s(z_i) + w_r R_s(z_i) + w_d D_s(z_i), \quad (6)$$

where w_a, w_r and w_d are the weights that represent the relative importance of autocorrelation, recency and deviation.

2) Estimation of Time-Varying Transition Probability Function: To estimate the time-varying transition probability function, we use a weighted likelihood approach informed by our prioritized memory scheme. We define a likelihood function [16], $L(T_t|o_{0:\mathcal{T}}, a_{0:\mathcal{T}}, \omega_{0:\mathcal{T}})$, that encapsulates the probability of observing a particular sequence of observations $z = (z_0, z_1, ..., z_\mathcal{T})$ within a window, given a sequence of actions $a = (a_0, a_1, ..., a_\mathcal{T})$ and the model T_t.

$$L(T_t|z_{0:\mathcal{T}}, a_{0:\mathcal{T}}, \omega_{0:\mathcal{T}}) = \prod_{t=0}^{\mathcal{T}} \left[\mathrm{Prob}_{\hat{P}}(z_t|a_t, \omega_t) \right]^{\omega_t}, \quad (7)$$

$\mathrm{Prob}_{\hat{P}}(z_t|a_t, \omega_t)$ represents the probability of observing z_t given action a_t and the model, adjusted by the weight ω_t.

To solve for T_t, we frame it as a constrained optimization problem with The objective of maximizing the log-likelihood function, subject to maximal rate of change constraint $\Delta_{\max}$. The optimization problem is then:

$$\hat{T}_t(o, a, o') = \underset{T_t}{\arg\max} \ \log L(T_t|o_{0:\mathcal{T}}, a_{0:\mathcal{T}}, \omega_{0:\mathcal{T}})$$
$$\text{subject to} \quad |T_t(o',a|o) - T_{t-1}(o',a|o)| \leq \Delta_{\max}.$$

The work in [16] shows that this is a convex problem. To solve this problem efficiently in real-time, we employ CVXOPT [31], an optimization library.

3) Policy Optimization and Planning: Our ultimate objective is to devise an optimal policy using the estimated transition function, $\hat{T}_t(s, a, s')$, that maximizes expected long-term rewards. To reflect the system's dynamic nature, we adjust the Bayesian belief state update with the estimated T_t:

$$b_{t+1}(s') = \eta \Omega(z_t|s'_t, a_t) \sum_{s \in S} \hat{T}_t(s'_t, a_t|s_t) b_t(s). \quad (8)$$

Here, $b(s')$ is the belief state at time $t+1$, $\Omega(z|s',a)$ is the likelihood of observing z after taking action a and transitioning to state s', and η is a normalizing constant to ensure a valid probability distribution. Optimal policy, $\pi(b)$, leverages expected $V_t(b)$ (Equation 1) to maximize expected rewards:

$$\pi(b) = \underset{a \in A}{\arg\max} \sum_{s' \in S} R_t(s,a) \hat{T}_t(s,a,s') b(s) + \gamma V_t(b'). \quad (9)$$

(a) Our approach (MPSE): # waypoints followed: 117

(b) DT-POMDP (classical estimation): # waypoints followed: 103

(c) TA-DESPOT: # waypoints followed: 99
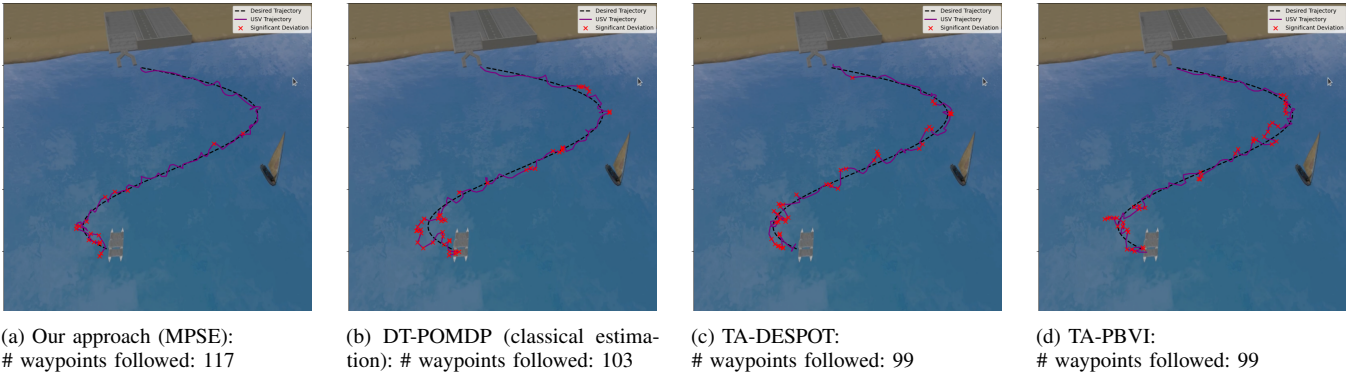
(d) TA-PBVI: # waypoints followed: 99

Fig. 2: The figures display the simulation environment and the waypoints tracked by the algorithms within a TV-POMDP (Scenario 2 with $\Delta_{max} = 0.02$). If the robot's estimate from a waypoint exceeds 3m, it is marked with **x** indicating notable deviations. The purple lines indicate the USV's trajectory, and the black lines are the desired 150-waypoint trajectory.

The problem's complexity in Equation 9 is tied to the dimensions of the state and action spaces, the real-time optimized transition function $\hat{T}_t(s,a,s')$ significantly reduces the computational burden. Commonly used algorithms such as Policy Iteration is adept at solving this problem efficiently.

V. EXPERIMENTS AND DISCUSSIONS

This section evaluates our algorithm's performance in TV-POMDPs through two cases: a simulated marine environment with an USV following waypoints, and a robot in a hardware experiment navigating the shortest path to a destination.

A. Simulated Marine Experiment

We simulate an USV navigating 150 waypoints, with the ultimate goal of reaching a final waypoint. The simulation is carried out in Gazebo [32] with ROS2 [33] as the middleware. Adopted from [34], the simulation models sophisticated water-vessel interactions through hydrodynamics plugin and utilizes the wavefield plugin [35] to reproduce effects of winds and waves.

- **Action Space:** The USV, based on [36], is modeled as a single engine vessel with action space $\mathcal{A}$ defined by:
  - Thrust $\in [0,35]N$, with discrete increments of 5N
  - Thrust Direction $\in [-90°,90°]$, with $10°$ increments.
- **Partial Observations:** The USV receives noisy position estimates as observations, with noise distributed by $\sim \mathcal{N}(0,\sigma^2)$ where $\sigma = 1.5$m was adjusted to reflect USV GPS accuracy levels [37].
- **Reward:** $\begin{cases} +10, & \text{per waypoint reached in the right} \\ & \text{order within a radius of 3m} \\ +50, & \text{upon reaching the final goal} \\ 0, & \text{otherwise} \end{cases}$
- **Model Uncertainty:** Hydrodynamic disturbances introduce time-varying transitions in simulation. Specifically, if a disturbance occurs, the trajectory deviates by an angle $\theta \sim \mathcal{N}(0,10^2), \theta \in [-10°,10°]$.

1) Baselines: The standard POMDP model does not natively handle time-varying dynamics, limiting applicability. We introduce modifications to enable comparisons.

We use classical estimation techniques with the Discrete-Time POMDP (DT-POMDP), which adds discrete time elements to create time-layered state structures. However, DT-POMDP has fixed horizons and discretization limitations. We extend Point-Based Value Iteration (PBVI) [38], a point-based POMDP planning method, to Time-Aware PBVI (TA-PBVI) by incorporating a time dimension into the belief points. We also extend Determinized Sparse Partially Observable Tree (DESPOT) [39], a belief tree planning method, to Time-Aware DESPOT (TA-DESPOT) by incorporating time steps as state variables during tree expansion to handle time.

2) Simulation results: We evaluate the performance of the algorithms in two distinct scenarios:

Scenario 1 - Constant Environmental Conditions: We employ the wavefield plugin to simulate a constant wind speed of 5m/s, thereby fixing the transition probability at 0.7 — essentially operating in a standard POMDP. For any action, the USV transitions to the expected next state with 70% probability. With 30% probability, a disturbance occurs, deviating the trajectory as described in V-A. This simplified scenario provides a performance reference before introducing time-varying disturbances in Scenario 2.

Scenario 2 - Adding time-varying disturbances: We use the hydrodynamics plugin and the wavefield plugin to introduce a time-varying transition function, $T_t = 1 - t/20$. This function simulates increasing currents over time that hinder the USV's motion, causing more stochastic transitions [2]. The USV's trajectory deviates by a probability of $t/20$ as described in V-A. The linear current growth matches real-world acceleration of currents in certain climates [40].

Discussions: As shown in the results of Scenario 1 (Table I, the baselines are well-established for solving standard POMDPs. TA-DESPOT performs best due to its exhaustive look-ahead search, while MPSE closely mirrors the performance. As shown in our simulations (Fig. 2) and the results (Table I), the time-varying dynamics in Scenario 2 present significant challenges for established baselines despite using time as a variable in their models. Unlike our approach, the baselines do not include mechanisms to prioritize historical samples alongside new data. This leads to their core
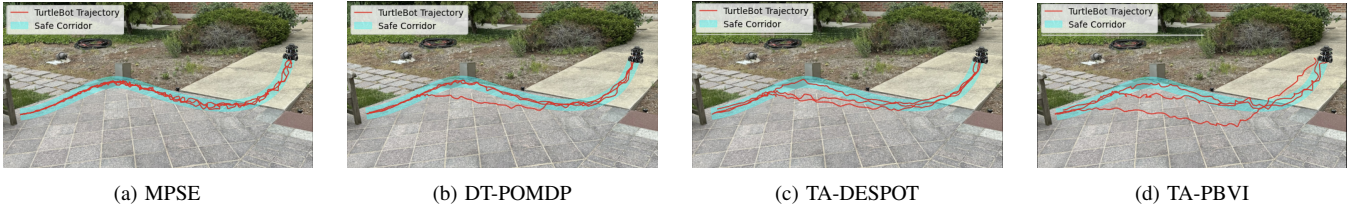
| (a) MPSE | (b) DT-POMDP | (c) TA-DESPOT | (d) TA-PBVI |

Fig. 3: The figures represent the hardware setup and trajectories for MPSE and baselines in a TV-POMDP (Scenario 3 with $\Delta_{max}=0.03$). Cyan corridor represents the map limits (H_{safe}) and red lines represent TurtleBot's trajectories. The background consists of two different terrains: the white areas depict rougher cemented terrain, while the darker areas show smoother terrain. The roughness variation between these surfaces is substantial enough to impact the TurtleBot's mobility.

	Constant Transition Probability $T_t=0.7$ for $0 \le t \le 20$		Time-Varying Transition Probability $T_t=1-t/20$ for $0 \le t \le 20$ $\Delta_{max}=0.02$	
	MAE	# waypoints followed	MAE	# waypoints followed
MPSE	0.16626	121	**0.22820**	**117**
DT-POMDP	0.19505	110	0.25921	103
TA-DESPOT	**0.14221**	**129**	0.27003	99
TA-PBVI	0.16088	123	0.29612	99

TABLE I: Comparison of mean absolute error (MAE) and the number of waypoints followed for Scenarios 1 and 2. The USV transitions to the next expected state with probability T_t and deviates with probability $1-T_t$ as described in V-A.

algorithms' inability to adapt effectively to evolving, uncertain dynamics, highlighting a critical limitation: a sole reliance on the most recent observations is not adequate for accurate planning in time-varying POMDPs. On the contrary, MPSE's selective combination of past and present information enables more effective adaptation to unknown, evolving dynamics.

Without mechanisms to prioritize samples from memory alongside new data, it becomes difficult for their core algorithms to adapt to the unknown, evolving dynamics. Fundamentally, this underscores a key limitation of relying solely on the most recent observations to accurately plan in a time-varying POMDPs. MPSE's superior performance highlights the value of this mechanism for settings with unknown, continuously changing dynamics.

B. Hardware Ground Vehicle Experiment

To further evaluate the capability of our algorithm in real-world settings, we conduct physical experiments using a TurtleBot Burger navigating a rough terrain. The robot is tasked with navigating from a designated start point to a predefined goal location in the shortest possible route while ensuring that it does not go off-road. The TurtleBot is equipped with an onboard computer, u-blox ZED F9P GPS module, and LIDAR for SLAM [41]. The agent has access to the safe coridor (H_{safe}) through an occupancy grid (H), generated using SLAM. H_{safe} defines the shortest, safest corridor to the objective. It moves at a maximum speed of

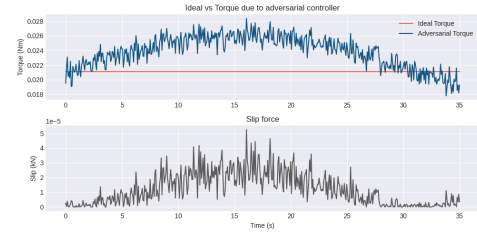0.22 m/s, receiving 10 Hz GPS observations. We leverage ROS to interface the sensors with the onboard computer.

Fig. 4: The red line represents the robot's maximum torque, while the blue line illustrates the effect of the introduced adversarial controller. The brown line depicts the resulting slippage on the robot's wheels. To empirically validate these effects on the model's transition probabilities, we conducted trials with and without the adversarial controller. By tracking the robot's GPS transitions and comparing against the ground truth without adversarial control, we tuned the model to account for the torque and slippage impact on transitions.

- **Action Space:** The robot's movements are defined by its steering angle, captured in the discretized action space:
  - Steering Angle (θ): $\theta \in [-\varphi, \varphi]$ in increments of $5°$, where $\varphi \in [-90, 90]$ is the maximum steering angle.
- **Partial Observations:** At each timestep, noisy GNSS provides the position estimate modeled as signal strength $SS_1 = f(\hat{d}_t)$, where $\hat{d}_t$ is the true position plus Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with horizontal error $\sigma = 1.5m$ [42].
- **Reward Function:** The reward $R(h)$ at any point h in the occupancy grid H is defined as follows:

$$R(h) = \begin{cases} -1, & \text{if } h \in H_{safe}, \\ -10, & \text{if } h \in H_{unsafe}, \\ +5, & \text{on reaching the objective waypoint} \end{cases}$$

- **Model Uncertainty:** To emulate time-varying environmental conditions like increasing rain, which could cause slip, we implement an adversarial controller that modulates the wheel torque — simulating the varying levels of wheel-ground friction, mimicking the effect of slip on the state transitions of the robot. Fig. 4 shows the effects of this artificially induced, time-varying slip. Consequently, the robot's trajectory deviates within an angle range $\zeta \sim \mathcal{N}(0, \sigma^2)$ where $\zeta \in [-10°, 10°]$ and $\sigma = 5°$.

	Constant Slip $T_t = 0.9$ for $0 \le t \le 20$		Time-Varying Slip $T_t = 0.9 \times e^{-0.05t}$ for $0 \le t \le 20$ and $\Delta_{max} = 0.02$		Time-Varying slip $T_t = \begin{cases} 1/(1+e^{-(t-5)}); 0 \le t \le 10 \\ e^{-(t-10)^2/2}; 10 \le t \le 20 \end{cases}$	
	MAE	**Cost incurred**	**MAE**	**Cost incurred**	**MAE**	**Cost incurred**
MPSE	0.09652	-34	**0.14201**	**-45**	**0.18195**	**-61**
DT-POMDP	0.11505	-56	0.18569	-62	0.30395	-84
TA-DESPOT	**0.07221**	**-31**	0.23723	-81	0.39528	-86
TA-PBVI	0.09988	-43	0.27453	-96	0.41741	-102

TABLE II: Comparison of total error and total cost incurred for different scenarios. The TurtleBot transitions to next expected state with the defined probability T_t and deviates with a probability $1 - T_t$ as described in V-B.
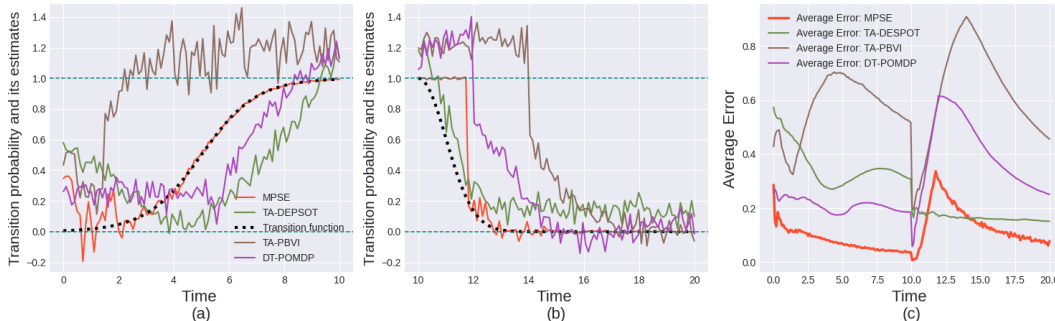

Fig. 5: Subfigures (a) and (b) display transition function changes in the first and second mission halves, respectively, while (c) shows maximal error over time. To ensure valid probabilities, the estimated transition probabilities are clipped between teal lines. The results depicted are after applying this clipping constraint. Although clipping alters individual probability values, it does not significantly impact the overall trends and comparative results. The key conclusions remain valid.

1) Hardware results: The results presented in Table II quantify the total error and the cost incurred when navigating from the start to goal in three different scenarios. Scenario setting for 1 and 2 remain consistent with the simulation settings. As shown, the results for the first two scenarios roughly align with the simulation results discussed earlier.

Scenario 3: Adaptive Transition Functions: We challenge our approach's adaptability by switching the transition functions midway through the experiment — emulating abrupt change in terrain. The robot transitions to the next expected state following $T_{t1}(t) = 1/(1+e^{-(t-5)})$ for the first 10s, after which it switches to $T_{t2}(t) = e^{-(t-10)^2/2}$ for the next 10s. Fig. 3 presents the trajectories followed in this scenario and Fig. 5 presents the estimated transition function.

Discussions: The third scenario significantly challenges all the algorithms. DT-POMDP with classical estimation struggled due to inability to swiftly adjust, a limitation in non-stationary settings [43]. TA-PBVI and TA-DESPOT were constrained by their dependence on preset beliefs/trees that are difficult to update. This intrinsic design choice means that when the environment changes unexpectedly, they have to heavily recompute or adjust these structures. This process is computationally intensive and time-consuming, leading to less agile responses. As observed in Fig. 5, TA-DESPOT responded fastest owing to its flexible tree [39]. However, its convergence lagged as optimizing the tree under altered conditions is difficult [27]. MPSE took time to balance retaining prior learning versus acquiring new samples, a

tradeoff in lifelong learning [44]. But after this adjustment period, MPSE outperformed all methods in convergence speed through continuous adaptation. By continuously balancing learning with selective forgetting, MPSE provides robust performance despite time-varying transitions.

VI. CONCLUSION

This work introduced Time-Varying Partially Observable Markov Decision Processes (TV-POMDPs) to address the challenges of decision-making in time-varying environments. To facilitate effective planning in such environments, we proposed a novel methodology called Memory Prioritized State Estimation (MPSE). Motivated by the need to manage the uncertainty arising from time-varying dynamics and partial observability, MPSE selectively prioritizes observations based on their information content, thereby enhancing the accuracy of estimations and ensuring computationally efficient planning. An associated planning strategy was also presented, tailored specifically to exploit the modeled time-varying transitions. We validated our approach through real-world and simulated experiments demonstrating the effectiveness of modeling the relevant systems as TV-POMDPs and utilizing MPSE for more accurate decision-making. Our planning strategy outperformed multiple baselines in leveraging the time-varying nature of the environment for optimized action selection. Our proposed work successfully integrates capabilities to plan in stochastic, partially observabile, and time-varying environments, overcoming limitations inherent in previous approaches.

REFERENCES

[1] J. Wolf and R. Flather, "Modelling waves and surges during the 1953 storm," *Philosophical transactions of the royal society A: mathematical, Physical and Engineering Sciences*, vol. 363, no. 1831, 2005.

[2] T. Konuk and J. Shragge, "Modeling full-wavefield time-varying sea-surface effects on seismic data: A mimetic finite-difference approach," *Geophysics*, vol. 85, no. 2, 2020.

[3] F. El-Hawary, Ed., *The ocean engineering handbook*, 1st ed. Crc Press, 2000.

[4] B. C. Shah and S. K. Gupta, "Long-distance path planning for unmanned surface vehicles in complex marine environment," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, 2020.

[5] F. Wang, Y. Bai, and L. Zhao, "Physical consistent path planning for unmanned surface vehicles under complex marine environment," *Journal of Marine Science and Engineering*, vol. 11, no. 6, 2023.

[6] L. Liu and G. S. Sukhatme, "A solution to Time-Varying Markov Decision Processes," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, 2018.

[7] B. Huang, P. W. Thorne, V. F. Banzon, T. Boyer, G. Chepurin, J. H. Lawrimore, M. J. Menne, T. M. Smith, R. S. Vose, and H.-M. Zhang, "NOAA extended reconstructed sea surface temperature (ERSST), Version 5," *NOAA National Centers for Environmental Information*, 2017.

[8] R. Bellman, "A markovian decision process," *Journal of Mathematics and Mechanics*, vol. 6, no. 5, 1957.

[9] D. P. Bertsekas, *Dynamic programming and optimal control*, 3rd ed. Athena Scientific, 2005, vol. I.

[10] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, 1998.

[11] J. Boyan and M. Littman, "Exact solutions to time-dependent MDPs," in *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[12] E. Rachelson, P. Fabiani, and F. Garcia, "TiMDPpoly: An improved method for solving time-dependent mdps," in *21st IEEE International Conference on Tools with Artificial Intelligence*, 2009.

[13] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1, 1999.

[14] Z. Kalmár, C. Szepesvári, and A. Lőrincz, "Module-based reinforcement learning: Experiments with a real robot," *Autonomous Robots*, vol. 5, 1998.

[15] I. Szita, B. Takács, and A. Lorincz, "ε–MDPs: Learning in Varying Environments." *Journal of Machine Learning Research*, vol. 3, no. 1, 2003.

[16] M. Ornik and U. Topcu, "Learning and planning for time-varying MDPs using maximum likelihood estimation," *Journal of Machine Learning Research*, vol. 22, no. 1, 2021.

[17] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[18] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12, 1999.

[19] B. C. da Silva, E. W. Basso, A. L. C. Bazzan, and P. M. Engel, "Dealing with non-stationary environments using context detection," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[20] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, 2019.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[22] S. Lange, T. Gabel, and M. Riedmiller, "Batch reinforcement learning," in *Reinforcement Learning*. Springer, 2012, vol. 12.

[23] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, 2017.

[24] S. Ramp, R. Davis, N. Leonard, I. Shulman, Y. Chao, A. Robinson, J. Marsden, P. Lermusiaux, D. Fratantoni, J. Paduan, F. Chavez, F. Bahr, S. Liang, W. Leslie, and Z. Li, "Preparing to predict: The second autonomous ocean sampling network (AOSN-II) experiment in the Monterey Bay," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 56, no. 3, 2009.

[25] L. Meng, R. Gorbet, and D. Kulić, "Memory-based deep reinforcement learning for POMDPs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[26] T. Phan, L. Belzner, M. Kiermeier, M. Friedrich, K. Schmid, and C. Linnhoff-Popien, "Memory bounded open-loop planning in large POMDPs using Thompson Sampling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019.

[27] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23, 2010.

[28] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[29] M. Etchart, P. Ladosz, and D. Mulvaney, "Spatio-temporal attention deep recurrent q-network for pomdps," in *19th EPIA Conference on Artificial Intelligence*, 2019.

[30] B. E. Huitema and J. W. McKean, "Autocorrelation estimation and inference with small samples." *Psychological Bulletin*, vol. 110, no. 2, 1991.

[31] M. S. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: A Python package for convex optimization, version 1.1.5," 2012.

[32] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[33] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, 2022.

[34] B. Bingham, C. Agüero, M. McCarrin, J. Klamo, J. Malia, K. Allen, T. Lum, M. Rawson, and R. Waqar, "Toward maritime robotic simulation in Gazebo," in *MTS/IEEE OCEANS*, 2019.

[35] J. Tessendorf *et al.*, "Simulating ocean water," *Simulating nature: realistic and interactive techniques. SIGGRAPH*, vol. 1, no. 2, 2001.

[36] O. I. Borisov, V. S. Gromov, A. A. Pyrkin, A. A. Bobtsov, and N. A. Nikolaev, "Robotic boat setup for control research and education," *IFAC-PapersOnLine*, vol. 49, 2016.

[37] F. Sotelo-Torres, L. V. Alvarez, and R. C. Roberts, "An unmanned surface vehicle (USV): Development of an autonomous boat with a sensor integration system for bathymetric surveys," *Sensors*, vol. 23, no. 9, 2023.

[38] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *18th International Joint Conference on Artificial Intelligence*, 2003.

[39] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," in *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[40] B. Margolius and M. M. O'Reilly, "The analysis of cyclic stochastic fluid flows with time-varying transition rates," *Queueing Systems: Theory and Applications*, vol. 82, no. 1–2, 2016.

[41] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, 2006.

[42] *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Artech, 2013.

[43] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online planning algorithms for POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, no. 1, 2008.

[44] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, 2019.