# Path Planning with Spatiotemporal Optimal Stopping for Stochastic Mission Monitoring

Graeme Best, *Student Member, IEEE,* Wolfram Martens, *Member, IEEE,* and Robert Fitch, *Member, IEEE*

***Abstract*—We consider an optimal stopping formulation of the mission monitoring problem, where a monitor vehicle must remain in close proximity to an autonomous robot that stochastically follows a predicted trajectory. This problem arises in a diverse range of scenarios, such as autonomous underwater vehicles supervised by surface vessels, pedestrians monitored by aerial vehicles, and animals monitored by agricultural robots. The key problem characteristics we consider are that the monitor must remain stationary while observing the robot, robot motion is modelled in general as a stochastic process, and observations are modelled as a spatial probability distribution. We propose a resolution-complete algorithm that runs in polynomial time. The algorithm is based on a sweep-plane approach and generates a motion plan that maximises the expected observation time and value. A variety of stochastic models may be used to represent the robot trajectory. We present results with data drawn from real AUV missions, a real pedestrian trajectory dataset and Monte Carlo simulations. Our results demonstrate the performance and behaviour of our algorithm, and relevance to a variety of applications.**

***Index Terms*—path planning, planning under uncertainty, optimal stopping, combinatorial optimisation, mission monitoring, marine robotics, pedestrian tracking, agricultural robotics.**

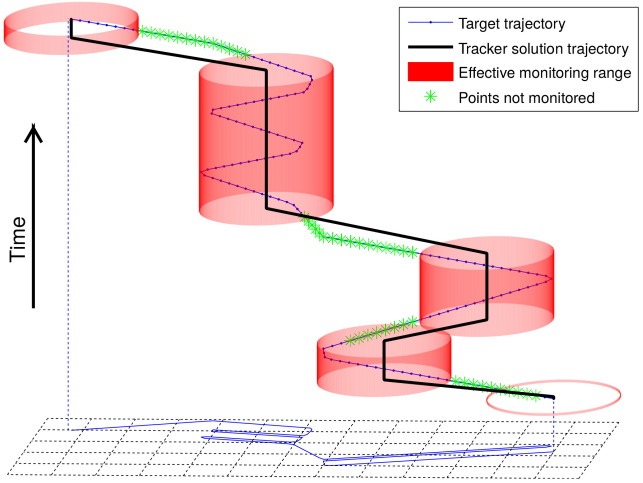


Fig. 1. Geometric interpretation of the spatiotemporal optimal stopping problem. A sample robot trajectory (an AUV mission) is shown in blue and also projected onto a plane in the two spatial dimensions. An example monitor trajectory solution is overlaid. Cylinders represent effective monitoring range (in this case a fixed-range communication model) at stopping locations. Green stars represent parts of the mission that are not monitored.

## I. Introduction

Mission monitoring is a supervisory problem where a robot or a manually driven vehicle tracks the progress of an autonomous mobile robot or other agent in performing a preplanned task. There are many examples of such tasks, including undersea surveys [1], [2], environmental monitoring [3], autonomous farming [4], [5] and planetary exploration [6]. Monitoring allows for rapid response to failures and to important information that the robot may discover during the progress of its mission [7]–[11]. Additionally, the monitoring vehicle may augment mission capabilities by providing observations from external viewpoints, such as for accurate localisation and navigation [12]–[16] or online sensor calibration [17]. The motion of the robot is typically represented by a mission plan, which may be defined probabilistically to take into account uncertain vehicle dynamics, environment models and mission objectives [18]–[21]. In some cases, the monitor vehicle must remain stationary in order to observe or communicate with the robot. The monitor vehicle must decide where to stop, and when to move to the next observation location. An illustration of the problem is shown in Fig. 1.

Classical *optimal stopping* problems [22], such as the well-known secretary problem, involve a binary choice; at each time point, the decision at hand is simply whether to stop or continue. If this choice can be repeated, the problem can be considered to be one-dimensional in the sense that it involves a choice of nonoverlapping intervals along a single dimension representing time. However, mission monitoring also involves spatial dimensions. We refer to this case as *spatiotemporal optimal stopping*. The goal of this work is to develop complete algorithms for a spatiotemporal optimal stopping problem where the motion of the target robot and the observations in general are stochastic.

Our formulation is motivated by a variety of real-world scenarios. Of particular interest is autonomous underwater vehicle (AUV) operations. Most AUVs in practice are supervised by powered surface vessels. The AUV navigates autonomously, often following a pre-planned trajectory with reasonable accuracy, but failures can occur that require human intervention. The AUV may also discover information of immediate value. Therefore, effective monitoring is relevant even if the robot is autonomous; monitoring allows operators to respond to failures and relevant information quickly.

Acoustic systems used for communication with the AUV have limited range and are unreliable, and some operators must

*This work was supported by the Australian Centre for Field Robotics; the New South Wales Government; the Australian Research Council's Discovery Project funding scheme (No. DP140104203); and the Faculty of Engineering & IT, The University of Sydney, under the Faculty Research Cluster Program.

The authors are with the Australian Centre for Field Robotics (ACFR), The University of Sydney, Sydney, Australia. {g.best,w.martens,rfitch}@acfr.usyd.edu.au

R. Fitch is also with the Centre for Autonomous Systems (CAS), University of Technology Sydney.

Manuscript received ??; revised ??.

stop and deploy this communication equipment, with engines powered down, for maximum efficiency [23]. In the simplest case, communication may be modelled deterministically with a fixed-range. However, this simple approach does not consider unpredictable hardware and environments; in practice, probabilistic models are typically required to account for physically realistic conditions [24]. An optimal stopping solution maximises the time spent communicating or observing effectively; this is achieved by choosing valuable observation locations and times, and reducing time spent unnecessarily travelling and stopping and starting the surface vessel.

Other real-world scenarios that motivate this problem arise in a wide range of robotics applications. These scenarios include: flying robots that must land during observational periods to conserve energy [25], pedestrian tracking for hands-free filming and photography [26], [27], acoustically-covert surveillance for tracking animals [28], ground-based mobile recharge stations for aerial vehicles [29], aerial robots that must be stationary to achieve accurate measurements of radio-tagged wildlife [30], and underwater robots that need to stop and surface to communicate or observe some phenomenon [31].

A key focus of this paper is the case where a mission is defined probabilistically. A probabilistic mission definition can take into account uncertainties such as unknown mission objectives, stochastic vehicle dynamics and imprecise environment models. Deterministic missions can often be generated directly from human-defined or automated plans, such as for AUV missions [32]–[36] or autonomous farming [4], [5]. In some applications, stochastic models of vehicle motion may be formulated as direct extensions of deterministic models, such as by adding model- or data-driven uncertainty to the trajectory [6], [18], [19]. In other cases, particular parameters of a mission are unknown, such as mission objectives or reactions to unforeseen events, leading to multi-modal predictions that are not direct extensions of a deterministic model [20], [21].

We propose a polynomial-time resolution-complete algorithm (complete with respect to the discretisation resolution) for the stochastic mission monitoring problem. Our algorithm generates an optimal nonoverlapping set of observation "cylinders" (in the simplest deterministic case) in the 3D configuration space consisting of two spatial dimensions and one time dimension (Fig. 1). These cylinders represent a stationary observation range and time, and are linked by a path that respects motion constraints of the monitor platform. The objective is *effective monitoring*, defined as maximising the expected overlap time between the probabilistic observation regions and the stochastic mission trajectory. Hardware-setup time penalties are naturally modelled geometrically by modifying the cylinder heights when evaluating trajectory overlap. Time and space are discretised, but fine resolution is feasible in practice. The algorithm first reduces the problem to a longest-path graph search, then passes a sweep-plane through the temporal dimension to compute a resolution-complete solution in polynomial time. We present an algorithm for the general probabilistic case as well as an elegant variant tuned for the deterministic case that runs more efficiently in practice.

In addition to analytical evaluation, we provide extensive simulation results for several example scenarios and realistic applications to demonstrate the relevance and applicability to real-world scenarios. In particular, we give implementation details and empirical analysis for applying the mission monitoring algorithm to two application case studies: 1) an AUV monitoring application using a probabilistic trajectory model with stochastic kinematics, localisation uncertainty, a closed-loop controller [37] and a realistic underwater communication model [24]; and 2) a pedestrian monitoring application using a realistic trajectory prediction model [20] and observations with occlusions in a cluttered environment. We present results with data drawn from actual AUV missions [23], a real pedestrian trajectory dataset [38] and Monte Carlo simulations of example trajectory models. The simulations illustrate the behaviour and performance of the algorithm when planning with various modelling assumptions. Overall, the experiments highlight advantages of the probabilistic formulation, demonstrate that the algorithm admits a broad range of probabilistic trajectory prediction and probabilistic observation models for practical scenarios, as well as clock-time performance that shows the solution is viable for practical use in mission monitoring.

### A. Contributions

The key contributions of this work are: 1) a spatiotemporal optimal stopping formulation of the stochastic mission monitoring problem, and 2) an efficient resolution-complete algorithm for this problem that reduces the problem to a longest-path search in a directed acyclic graph. Robot motion and observations are modelled such that they yield a probability distribution describing effective communication or observations for every point in time and space. We consider several such stochastic models in our experiments, including an AUV prediction model that extends a deterministic control policy [37] using Monte Carlo simulations, a realistic underwater communications model [24], and a multi-modal intention inference pedestrian prediction model [20].

A preliminary version of this paper appeared as [39]. This extended version additionally contains: a generalised formulation for probabilistic observation models; relaxation of the start and end constraints; expanded algorithmic details; and extensive implementation details, simulated experiments and discussion for AUV mission monitoring and pedestrian monitoring case studies.

The remainder of this paper is organised as follows. Sec. II describes related literature for mission monitoring, optimal stopping in robotics, sweep-plane algorithms, path planning with temporal constraints and coordinated-control. Sec. III formally defines the spatiotemporal optimal stopping formulation and mission monitoring objective. Sec. IV gives an overview of the algorithm, which is divided into two phases: Sec. V details the spatiotemporal graph generation phase, while Sec. VI details the sweep-plane algorithm and analysis. Sec. VII presents simulated experiments for analysing the behaviour of the algorithm and the probabilistic formulation. Sec. VIII presents two application case studies, with implementation details, extensive simulation results and empirical analysis. Finally, Sec. IX summarises the paper and future research directions.

## II. Related Work

A closely related problem is studied by Best and Anstee [23], who propose a greedy planner and a genetic algorithm for AUV mission monitoring where the AUV trajectory and communication model are assumed to be deterministic. The genetic algorithm is shown to achieve reasonable results but makes no guarantees on convergence, runtime, or optimality. In this paper, we develop the spatiotemporal optimal stopping formulation, generalise the problem to admit stochastic target robot trajectories and stochastic observation models, and present efficient algorithms with analytical guarantees.

Optimal stopping describes a class of problems that require a choice of when to take a particular action in order to maximise an expected reward [22]. Recently, Lindhé and Johansson [40] studied an optimal stopping problem for a robot that communicates with a base station while traversing a predefined path. The robot must choose stopping points that maximise communication quality while also making progress along its path. Our problem is similar, but here the stopping locations are also to be optimised since the robot is not restricted to a predefined path, and the reward received is time dependent. Further, feasible next actions depend on previous actions; a given stopping decision constrains the reachability of future stopping locations. Planning must consider the entire stopping sequence, rather than one-stop planning. Das *et al.* [41] also recently applied optimal stopping theory to a robot motion planning problem. The problem was to choose sampling locations for persistent collection of water samples by an AUV. Similar to [40], the stopping locations are restricted to points along a predefined path. Sampling is instantaneous, rather than pausing motion for a planned time interval such as in [40] and our problem. The *beachcombers' problem* [42] is a related theoretical problem where a team of robots perform coverage of a line interval. Each robot can switch between two behaviours: searching slowly while observing, or walking quickly but blindly. While this problem is also similar and has interesting theoretical properties, it does not address time-varying observations, range sensing or multi-dimensional environments.

Sweep-plane algorithms are often used for computational geometry problems such as Voronoi decomposition, intersections between line segments and unions of rectangles [43], [44]. An $\mathbb{R}^{n-1}$ hyperplane is swept monotonically through an $\mathbb{R}^n$ space, and calculations are performed at event points. Robot motion planning problems can often be formulated geometrically and solved with sweep-plane solutions [45], [46]. Our approach features a sweep-plane moving through time, where the event calculations represent optimal sub-problems and lead to an optimal global solution.

An event can be thought of as a vertex in a spatiotemporal search graph with edges linking back to previous events. This construction forms a directed acyclic graph and therefore a longest path can be computed in polynomial time [47], [48]. Bopardikar *et al.* [49] employ this approach for dynamic vehicle routing, where an agent maximises the number of space-time demands visited. Our problem again is similar, however our agent seeks to occupy a region defined probabilistically over time. The novelty of our approach in comparison lies in our proposed graph construction algorithm to maintain optimality for a complex constraint space and objective function.

Although there appears to be a dearth of work that directly extends temporal optimisation problems to consider space, there is a large body of literature that extends spatial optimisation to consider time. Prominently, vehicle routing problems (VRPs) have been studied with various time constraints, such as VRPs with time windows [50]. Similarly, graph-coverage problems have been addressed with time-constrained edge availabilities [51]. The key difference in our work is that time is considered as an objective to be maximised (for effective monitoring) rather than as a constraint.

In marine robotics, coordinated-control problems have received much attention due to the benefits realised by multi-robot systems [52]. Related problems include formation control and communication connectivity maintenance [12], [53], [54], and target following [55], [56], and these problems are generally approached using closed-loop control with a sliding time horizon. We focus on longer-term path planning with an objective characterised as optimal stopping, and therefore formulate a combinatorial optimisation solution.

## III. Problem Formulation

The problem involves two mobile agents: 1) a *target* agent which follows a probabilistic trajectory defined by a mission plan, and 2) a *tracker* agent that seeks to effectively monitor the target throughout the mission. To monitor effectively, the tracker must be within observation/communication range of the target and must be stationary. The trajectory of the tracker can therefore be characterised as a sequence of stopping waypoints in time and space. This scenario presents an optimisation problem with the target's trajectory as the independent variable, while the tracker's trajectory is optimised. In this section, we formally define the characteristics of the target and tracker trajectories, the general definition for probabilistic observation models, and the idea of effective monitoring as an optimisation objective.

### A. Target Trajectory (Independent Variable)

The trajectory of the target is described as its position as a function of time $x(t) : [0, T] \to \mathcal{X}$, where $\mathcal{X}$ is the space of all possible target locations. Time $T$ is a planning horizon (in our experiments, we set $T$ to the full duration of the mission). The mission is discretised into $N$ time steps $t_i := (i-1)\Delta_{\mathrm{t}} \in \mathcal{T}$, with $t_1 = 0$ and $t_N = T$. The trajectory is not known precisely ahead of time, and therefore the predicted position of the target at time $t_i$ is represented as a random variable $X_i$ with a known distribution $X_i \sim D_i$ and associated probability density function $\rho_i(x)$. Therefore, the predicted trajectory of the target is described as the sequence of random variables $X := (X_1, X_2, ..., X_N)$.

### B. Tracker Trajectory (Dependent Variable)

The trajectory of the tracker is described as its position as a function of time $y(t) : [0, T] \to \mathcal{Y}$, where $\mathcal{Y}$ is the space of all feasible positions of the tracker. The trajectory of the tracker is characterised as alternating between two states $\{\text{STOPPED}, \text{MOVING}\} := \mathcal{S}$, which is described by a function of time $s(t) : [0, T] \to \mathcal{S}$. The functions $y(t)$ and $s(t)$ are sampled at time steps $t_i \in \mathcal{T}$, resulting in the sequences of positions $Y = (y_1, y_2, ..., y_N)$ and states $S = (s_1, s_2, ..., s_N)$.

*1) Stationary waypoints:* The trajectory of the tracker is also described by the tuple $U = [\hat{Y}, T^{\text{a}}, T^{\text{d}}]$, where $\hat{Y} := (\hat{y}_1, \hat{y}_2, ..., \hat{y}_M)$ is a sequence of waypoint positions with sequences of associated arrival times $T^{\text{a}} := (t^{\text{a}}_1, t^{\text{a}}_2, ..., t^{\text{a}}_M)$ and departure times $T^{\text{d}} := (t^{\text{d}}_1, t^{\text{d}}_2, ..., t^{\text{d}}_M)$.

During the time interval $[t^{\text{a}}_i, t^{\text{d}}_i)$, the tracker is in the STOPPED state and is stationary at the waypoint position $\hat{y}_i \in \hat{\mathcal{Y}} \subseteq \mathcal{Y}$, where $\hat{\mathcal{Y}}$ is the set of positions where the tracker may stop. During the time interval $[t^{\text{d}}_i, t^{\text{a}}_{i+1})$, the tracker is in the MOVING state and is travelling between consecutive waypoints $\hat{y}_i, \hat{y}_{i+1}$. The sequences of arrival and departure times satisfy the constraints: $t^{\text{d}}_1 \geq 0, t^{\text{a}}_M \leq T$, and $t^{\text{a}}_i < t^{\text{d}}_i < t^{\text{a}}_{i+1}, \forall i$.

The required travel time $\delta(\hat{y}_i, \hat{y}_j) := t^{\text{a}}_j - t^{\text{d}}_i$ between two waypoints is defined by a function $\delta(\hat{y}_i, \hat{y}_j) : \hat{\mathcal{Y}} \times \hat{\mathcal{Y}} \to \mathbb{R}_{\geq 0}$. The proposed algorithm does not depend on the exact trajectory taken to achieve this travel time. We require $\delta(\hat{y}_i, \hat{y}_j) = 0$ iff $\hat{y}_i = \hat{y}_j$.

By this definition, the position as a function of time has

$$y(t) = \hat{y}_i, \ \forall t \in [t^{\text{a}}_i, t^{\text{d}}_i), \ \forall i \in \{1, 2, ..., M\},$$

and the state as a function of time is

$$s(t) = \begin{cases} \text{STOPPED} & \text{if } t \in \bigcup_{i=1}^{M} [t^{\text{a}}_i, t^{\text{d}}_i) \\ \text{MOVING} & \text{otherwise.} \end{cases}$$

*2) Start and end conditions:* The start position $\hat{y}_1$ and end position $\hat{y}_M$ of the tracker are assumed to be elements of given sets $\hat{\mathcal{Y}}_{\text{start}}$ and $\hat{\mathcal{Y}}_{\text{end}}$, respectively. These positions may be fixed, for example when the tracking vehicle is used for deploying/retrieving the target vehicle at fixed locations. Alternatively, if the sets are non-singleton then $\hat{y}_1$ and $\hat{y}_M$ are to be optimised by the proposed planner.

### C. Monitoring Effectiveness (Objective Function)

The goal of the tracker is to effectively monitor the target. At time $t_i$, the monitoring effectiveness is described by a function $f(X_i, y_i, s_i) : \mathcal{X} \times \mathcal{Y} \times \mathcal{S} \to [0, 1]$, with output ranging from 0 (not monitoring) to 1 (effectively monitoring). This function is defined as

$$f(X_i, y_i, s_i) := \begin{cases} \tilde{f}(\|X_i - y_i\|) & \text{if } s_i = \text{STOPPED} \\ 0 & \text{if } s_i = \text{MOVING}, \end{cases}$$

where $\tilde{f}(r_i) : \mathbb{R}_{\geq 0} \to [0, 1]$ is the observation model (interchangeable with communication model) and describes the observation value of monitoring the target from a distance of $r_i$. Without loss of generality, the observation value is assumed to be scaled between 0 and 1. For example, $\tilde{f}(r_i)$ may describe the probability of the tracker successfully communicating with or detecting the target, or the expected communication bandwidth. This function may be a simple binary $r$-disk model (Sec. III-D) or a more realistic observation model (Sec. VIII). For clarity, we define the observation model as translation-, orientation- and time-invariant, however the algorithm can readily be extended for more general models. (Sec. VIII-B demonstrates a translation-dependent model.)

The monitoring effectiveness objective function $F(X, Y, S)$ is defined as the expected monitoring effectiveness over the duration of the mission:

$$\begin{aligned} F(X, Y, S) &:= \mathbb{E}\left[\Delta_{\text{t}} \sum_{i=1}^{N} f(X_i, y_i, s_i)\right] \\ &= \Delta_{\text{t}} \sum_{i=1}^{N} \mathbb{E}\left[f(X_i, y_i, s_i)\right], \end{aligned}$$

which can be interpreted as the expected weighted sum of time that the tracker is STOPPED, weighted by the observation values. $F(X, Y, S)$ can be evaluated using the expected values

$$\begin{aligned} \mathbb{E}\left[f(X_i, y_i, \text{STOPPED})\right] &= \mathbb{E}\left[\tilde{f}(\|X_i - y_i\|)\right] \\ &= \int_{\mathcal{X}} \rho_i(x) \tilde{f}(\|x - y_i\|) dx, \qquad (1) \\ \mathbb{E}\left[f(X_i, y_i, \text{MOVING})\right] &= 0. \end{aligned}$$

For convenience, we also introduce notation for the monitoring effectiveness evaluated at the set of discrete timesteps $\subseteq \mathcal{T}$ that fall within a (continuous) time period of interest $\mathtt{T}$:

$$F_{\mathtt{T}} := \Delta_{\text{t}} \sum_{\eta \in \mathcal{N}} \mathbb{E}\left[f(X_\eta, y_\eta, s_\eta)\right], \ \mathcal{N} = \{\eta : t_\eta \in \mathtt{T} \cap \mathcal{T}\}.$$

### D. Deterministic Problem Instances

As a special case, we address deterministic scenarios where both: 1) the target's trajectory $X$ is deterministic (i.e., each $\rho_i(x)$ is defined as a Dirac delta function), and 2) the monitoring effectiveness function $\tilde{f}(r_i)$ is defined as the binary $r$-disk model with monitoring range $r$, i.e.,

$$\tilde{f}(r_i) := \begin{cases} 1 & \text{if } r_i \leq r \\ 0 & \text{otherwise.} \end{cases}$$

In these deterministic problem instances, the expected value $\mathbb{E}\left[f(X_i, y_i, s_i)\right] = f(X_i, y_i, s_i)$ and evaluates to 0 or 1 only.

For this special case, we also assume that the average speed of the tracker between waypoints is not less than the maximum instantaneous speed of the target $\|\dot{x}\|_{\max}$, i.e.,

$$\frac{\|\hat{y}_j - \hat{y}_i\|}{\delta(\hat{y}_i, \hat{y}_j)} \geq \|\dot{x}\|_{\max}, \ \forall \hat{y}_i, \hat{y}_j. \qquad (2)$$

### E. Problem Statement

The optimisation problem to be solved is stated as follows. *For a given target trajectory $X$, find the set of stopping waypoints $U$ with positions $\hat{Y}$, arrival times $T^a$ and departure times $T^d$, such that the travel time $\delta(\hat{y}_i, \hat{y}_j)$ constraints are met and the expected monitoring effectiveness $F(X, Y, S)$ is maximised over the mission duration.*

We address this problem for the general case, as well as for the special deterministic case defined in Sec. III-D. The proposed algorithm has improved efficiency if certain reasonable assumptions hold, which are defined later in Sec. V-A.

**Algorithm 1** Overview of trajectory planner for tracker

```
function MAIN($X$)
    $[\mathcal{V}, \mathcal{E}, \{t_i^{\text{d}}\}, \{t_j^{\text{a}}\}, \{\omega\}] \leftarrow$ GENERATEGRAPH($X$)
    $[\mathcal{V}, \mathcal{E}, \mathcal{V}_{\text{start}}, \mathcal{V}_{\text{end}}] \leftarrow$ STARTENDCOND'S($\mathcal{V}, \mathcal{E}, X$)
    $[\{\Omega\}, \{\psi\}] \leftarrow$ SWEEPPLANE($\mathcal{V}, \mathcal{E}, \{\omega\}, \{t_i^{\text{a}}\}$)
    $[U, F] \leftarrow$ BACKTRACK($\{\Omega\}, \{\psi\}, \mathcal{V}, \mathcal{E}, \mathcal{V}_{\text{start}}, \mathcal{V}_{\text{end}}$)
    return $[U = [\hat{Y}, T^{\text{a}}, T^{\text{d}}], F]$
```

## IV. Algorithm Overview

The proposed algorithm is divided into a graph generation phase and then a longest-path graph search using a sweep-plane. Pseudocode is listed in Alg. 1.

The first phase generates a search graph such that paths through this graph describe feasible solution trajectories for the tracker. The generated graph is directed acyclic, which enables the optimal solution to be found efficiently. The graph generation exploits geometric properties of the problem so that, under certain conditions, the efficiency of the algorithm is improved and optimality is maintained. This graph generation procedure is presented in Sec. V.

In the second phase, the optimal solution is found by performing a longest-path search through the spatiotemporal graph. We describe this algorithm geometrically as a sweep-plane moving through time. For general graphs, a longest-path search is NP-hard. However, since the generated graph is directed acyclic, the optimal trajectory is found in polynomial time. This sweep-plane algorithm is presented in Sec. VI.

## V. Spatiotemporal Search Graph

This section describes the process of generating the spatiotemporal search graph, as summarised in Alg. 2. Sec. V-A describes how to generate the graph vertices, where each vertex represents a time interval at a position. Sec. V-B describes how to generate edges, such that each edge represents the travel time and monitoring effectiveness for travelling between a pair of vertices. Finally, Sec. V-C discusses further adjustments to the graph so that the start and end conditions are met.

### *A. Vertices*

A set of graph vertices is generated, with each vertex $v_i \in \mathcal{V}$ representing a potential stopping location in time and space. This is achieved by selecting a discrete set of positions in space in the neighbourhood of the target's path. Time is incorporated for each position by considering all times that the tracker is expected to be effectively monitoring the target. We first introduce the spatial dimensions in Sec. V-A1, followed by the temporal dimensions in Sec. V-A2.

*1) Spatial dimensions:* The set of positions $\mathcal{P} \subseteq \hat{\mathcal{Y}}$ is generated by first discretising the space, such that $\mathcal{P} = \hat{\mathcal{Y}} \cap \mathcal{P}_1$, where $\mathcal{P}_1$ is a discrete set of positions. The algorithm is optimal with respect to the discretisation used for $\mathcal{P}_1$. The best choice for discretisation is problem specific; a uniform grid is used in all figures and most of the experiments, while Sec. VIII-A uses an adaptive-resolution grid and Sec. VIII-B uses a probabilistic roadmap (PRM).

**Algorithm 2** Generating a spatiotemporal search graph of potential waypoint positions and times

```
function GENERATEGRAPH($X$)
    Select potential stopping locations $p_i \in \mathcal{P}$
    Generate vertices $v_\eta = [p_\eta, \tau_\eta^{\text{a}}, \tau_\eta^{\text{d}}] \in \mathcal{V}$
    Find feasible edges $e_k = \langle v_i, v_j \rangle \in \mathcal{E}$
    Calculate edge times $t_i^{\text{d}}, t_j^{\text{a}}$ for each edge $e_k$
    Calculate edge weight $\omega_{i,j}$ for each edge $e_k$
    return $[\mathcal{V}, \mathcal{E}, \{t_i^{\text{d}}\}, \{t_j^{\text{a}}\}, \{\omega_{i,j}\}]$
```

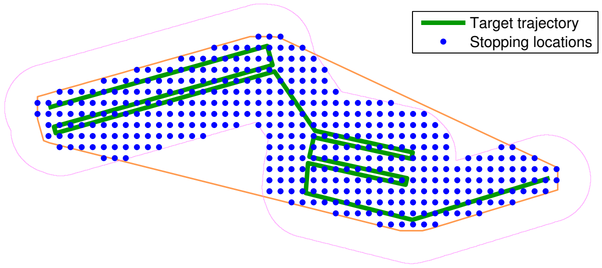


Fig. 2. Example showing possible stopping locations around a deterministic target trajectory moving through 2 spatial dimensions. Also shown are the boundaries of the $\mathcal{P}_2$ monitoring region (pink) and $\mathcal{P}_3$ convex hull (orange) described in Sec. V-A1.

The search space can be further reduced by taking the intersection with two additional sets, such that $\mathcal{P} = \hat{\mathcal{Y}} \cap \mathcal{P}_1 \cap \mathcal{P}_2 \cap \mathcal{P}_3$. The set $\mathcal{P}_2$ describes the monitoring region and $\mathcal{P}_3$ describes the convex hull of the mission. These sets are formally defined below and an example of these sets is illustrated in Fig. 2. Under reasonable conditions (defined below), optimality is still guaranteed after performing this culling. If a particular condition does not hold for a specific problem instance, then the associated set may be omitted to guarantee optimality. This additional culling of the search domain is not essential; the algorithm has the same runtime complexity, but the culling will improve the efficiency of the algorithm in practice.

The set $\mathcal{P}_2$ is the set of all points $p_i$ that have a non-zero monitoring effectiveness (i.e., are within monitoring range) for part of the target's trajectory, i.e.,

$$\exists X_\eta \in X : \mathbb{E}\left[\tilde{f}(\|X_\eta - p_i\|)\right] > 0.$$

Lemma 1 shows optimality is maintained if the $\mathcal{P}_2$ culling is used, assuming Condition 1 holds. For $\mathcal{P}_3$, first define CH as the convex hull of the set of all possible locations visited by the target and the tracker start and end sets; i.e., the convex hull of the set

$$\{x : \exists i, \rho_i(x) > 0\} \cup \hat{\mathcal{Y}}_{\text{start}} \cup \hat{\mathcal{Y}}_{\text{end}}.$$

The set $\mathcal{P}_3$ is all points that are in CH or are a distance less than the $\mathcal{P}_1$ discretisation spacing away from the boundary of CH. Lemma 2 shows optimality is maintained if the $\mathcal{P}_3$ culling is used, assuming Conditions 2, 3 and 4 hold.

**Condition 1. Triangle inequality:** The travel times satisfy the triangle inequality, i.e., $\delta(\hat{y}_a, \hat{y}_b) \leq \delta(\hat{y}_a, \hat{y}_i) + \delta(\hat{y}_i, \hat{y}_b)$.

**Condition 2. Convex hull is feasible:** All positions in CH are feasible stopping locations, i.e., CH $\subseteq \hat{\mathcal{Y}}$.

**Condition 3. Monotonically decreasing observation value:** The observation model $\tilde{f}(r_i)$ is a monotonically decreasing function of distance.

**Condition 4. Monotonically increasing travel time:** The travel time monotonically increases with distance, for a fixed start or end position; i.e., if $\|\hat{y}_i - \hat{y}_a\| \geq \|\hat{y}_i - \hat{y}_b\|$ then $\delta(\hat{y}_i, \hat{y}_a) \geq \delta(\hat{y}_i, \hat{y}_b)$ and $\delta(\hat{y}_a, \hat{y}_i) \geq \delta(\hat{y}_b, \hat{y}_i)$.

**Remark 1.** If the distributions $\rho_i(x)$ or the observation function $\tilde{f}(r_i)$ have an unbounded support, then $\mathcal{P}$ is potentially an infinite set. However, this is not an issue since the reachability pruning (later in Sec. V-C) ensures the search space is finite. For computational reasons, to further reduce the size of the graph it may be appropriate to approximate $\mathcal{P}_2$ and $\mathcal{P}_3$ using non-zero lower bounds, i.e., $\mathbb{E}\left[\tilde{f}(\|X_\eta - p_i\|)\right] > LB_1$ for $\mathcal{P}_2$ and $\rho_i(x) > LB_2$ for $\mathcal{P}_3$.

**Lemma 1. Stopping in effective monitoring region:** *If the travel times satisfy the triangle inequality (Condition 1), then an optimal solution trajectory $U$ only contains waypoints at locations $\hat{y}_i \in \hat{Y}$ which satisfy $\exists X_\eta \in X$ : $\mathbb{E}\left[\tilde{f}(\|X_\eta - \hat{y}_i\|)\right] > 0$.*

*Proof:* Define two partial solution trajectories over the time interval $[t_i^{\mathrm{a}}, t_k^{\mathrm{d}})$: 1) $\hat{Y} = (\hat{y}_i, \hat{y}_j, \hat{y}_k)$ and 2) $\hat{Y}^* = (\hat{y}_i, \hat{y}_k)$; with $\hat{y}_j$ satisfying $\nexists X_\eta \in X : \mathbb{E}\left[\tilde{f}(\|X_\eta - \hat{y}_j\|)\right] > 0$. It follows from this definition of $\hat{y}_j$ that the monitoring effectiveness $F$ while STOPPED at $\hat{y}_j$ is $F_{[t_j^{\mathrm{a}}, t_j^{\mathrm{d}})} = 0$. Combined with the MOVING time intervals, $F_{[t_i^{\mathrm{d}}, t_k^{\mathrm{a}})} = 0$, with interval length $L = t_k^{\mathrm{a}} - t_i^{\mathrm{d}}$. Therefore $F(\hat{Y}) = F_{[t_i^{\mathrm{a}}, t_i^{\mathrm{d}}) \cup [t_k^{\mathrm{a}}, t_k^{\mathrm{d}})}$. For $\hat{Y}^*$, the monitoring effectiveness while MOVING is $F_{[t_i^{\mathrm{d}*}, t_k^{\mathrm{a}*})} = 0$, with interval length $L^* = t_k^{\mathrm{a}*} - t_i^{\mathrm{d}*}$. Therefore $F(\hat{Y}^*) = F_{[t_i^{\mathrm{a}}, t_i^{\mathrm{d}*}) \cup [t_k^{\mathrm{a}*}, t_k^{\mathrm{d}})}$. Since the triangle inequality holds (Condition 1), it follows that $L \geq L^*$. Therefore $\exists \{t_i^{\mathrm{d}\prime}, t_k^{\mathrm{a}\prime}\} : (t_i^{\mathrm{d}\prime} \geq t_i^{\mathrm{d}}) \wedge (t_k^{\mathrm{a}\prime} \leq t_k^{\mathrm{a}})$, where $\{t_i^{\mathrm{d}*}, t_k^{\mathrm{a}*}\} = \{t_i^{\mathrm{d}\prime}, t_k^{\mathrm{a}\prime}\}$ is a feasible choice for departure and arrival times. The optimal choice for $\{t_i^{\mathrm{d}*}, t_k^{\mathrm{a}*}\}$ will always result in a greater or equal monitoring effectiveness than if $\{t_i^{\mathrm{d}*}, t_k^{\mathrm{a}*}\} = \{t_i^{\mathrm{d}\prime}, t_k^{\mathrm{a}\prime}\}$ were chosen, therefore:

$$\begin{aligned} F(\hat{Y}^*) &\geq F_{[t_i^{\mathrm{a}}, t_i^{\mathrm{d}\prime}) \cup [t_k^{\mathrm{a}\prime}, t_k^{\mathrm{d}})} \\ &= F_{[t_i^{\mathrm{a}}, t_i^{\mathrm{d}}) \cup [t_i^{\mathrm{d}}, t_i^{\mathrm{d}\prime}) \cup [t_k^{\mathrm{a}\prime}, t_k^{\mathrm{a}}) \cup [t_k^{\mathrm{a}}, t_k^{\mathrm{d}})} \\ &= F(\hat{Y}) + F_{[t_i^{\mathrm{d}}, t_i^{\mathrm{d}\prime}) \cup [t_k^{\mathrm{a}\prime}, t_k^{\mathrm{a}})} \\ &\geq F(\hat{Y}). \end{aligned}$$

It follows that $F(\hat{Y})$ will never decrease if $\hat{y}_j$ was removed from the sequence. This generalises to longer sequences since $F$ is additive over partial sequences; therefore an optimal sequence exists with all $\hat{y}_i$ in range of an $X_\eta$. □

**Lemma 2. Stopping in convex hull:** *If* CH $\subseteq \hat{\mathcal{Y}}$ *(Condition 2), $\tilde{f}$ is monotonically decreasing (Condition 3), and the travel time monotonically increases with distance (Condition 4), then an optimal solution trajectory $U$ only contains waypoints at locations $\hat{y}_i \in \hat{Y}$ which are in* CH.

*Proof:* Define a stopping position $\hat{y}_a \notin$ CH. By definition, there exists a half-plane $\mathcal{H}$ such that CH $\subset \mathcal{H}$, $\hat{y}_a \notin \mathcal{H}$, and $\hat{y}_a^*$ lies on the boundary of $\mathcal{H}$ where $\hat{y}_a^*$ is the closest point to $\hat{y}_a$ in CH. The line segment $\hat{y}_a$ to $\hat{y}_a^*$ is perpendicular to the boundary of $\mathcal{H}$; therefore $\hat{y}_a^*$ is closer than $\hat{y}_a$ to any point in $\mathcal{H}$, i.e.,

$$\|\hat{y}_a^* - h\| < \|\hat{y}_a - h\|, \forall h \in \mathcal{H}. \tag{3}$$

Therefore, since $X \subset \mathcal{H}$ and $\tilde{f}$ is monotonically decreasing (Condition 3):

$$\tilde{f}(\|X_i - \hat{y}_a^*\|) \geq \tilde{f}(\|X_i - \hat{y}_a\|), \forall X_i \in X.$$

It follows that the monitoring effectiveness of a solution that contains a waypoint at $\hat{y}_a^*$ will never decrease if this waypoint were moved to $\hat{y}_a$ instead. It is assumed that $\hat{y}_a^* \in \hat{\mathcal{Y}}$, which will hold if Condition 2 holds.

To be optimal, selecting $\hat{y}_a^*$ instead of $\hat{y}_a$ must also not result in a lower monitoring effectiveness at the previous and next waypoints in the sequence. Define the partial solutions $\hat{Y} = (\hat{y}_i, \hat{y}_a, \hat{y}_j)$ and $\hat{Y}^* = (\hat{y}_i, \hat{y}_a^*, \hat{y}_j)$, where $\hat{y}_i, \hat{y}_j \in$ CH $\subset \mathcal{H}$. It follows from (3) and Condition 4 that the travel times will not increase by selecting $\hat{y}_a^*$ instead of $\hat{y}_a$, i.e.,

$$\delta(\hat{y}_i, \hat{y}_a^*) \leq \delta(\hat{y}_i, \hat{y}_a) \quad \text{and} \quad \delta(\hat{y}_a^*, \hat{y}_j) \leq \delta(\hat{y}_a, \hat{y}_j).$$

Therefore the departure from $\hat{y}_i$ need not be earlier and the arrival to $\hat{y}_j$ need not be later if $\hat{y}_a^*$ is chosen instead of $\hat{y}_a$; hence the monitoring effectiveness at $\hat{y}_i$ and $\hat{y}_j$ will not decrease if $\hat{y}_a^*$ is chosen instead of $\hat{y}_a$.

It follows that $F(\hat{Y}^*) \geq F(\hat{Y})$. Given that $\hat{y}_1, \hat{y}_M \in$ CH, this generalises to longer sequences. Therefore an optimal solution trajectory has all $\hat{y}_i \in$ CH. □

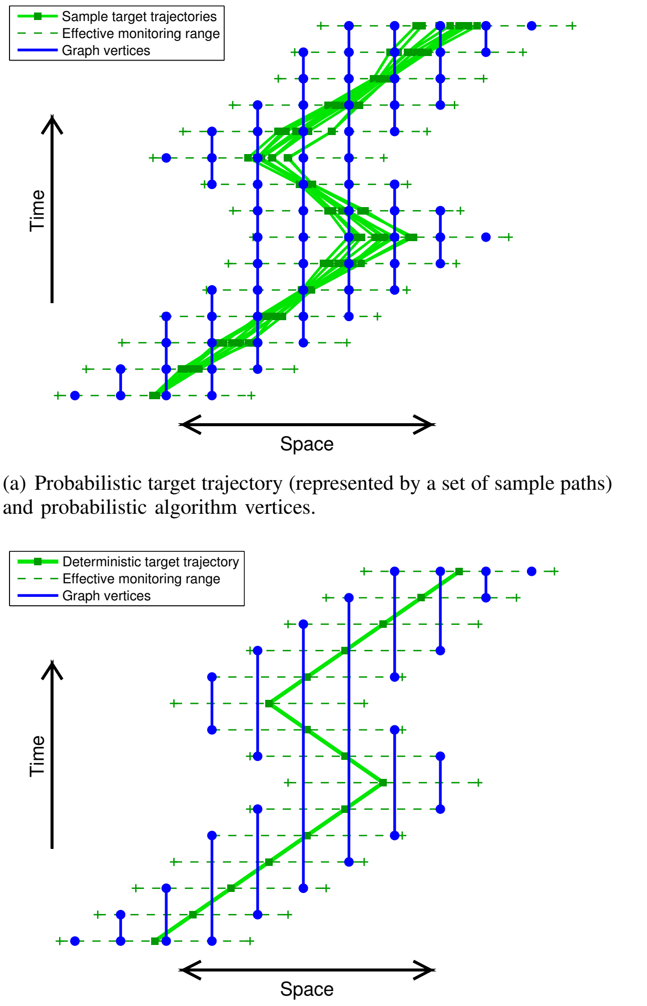
*2) Temporal dimensions:* Each vertex $v_\eta \in \mathcal{V}$ represents a position $p_\eta \in \mathcal{P}$ and a time interval $[\tau_\eta^{\mathrm{a}}, \tau_\eta^{\mathrm{d}}] \subseteq \mathcal{T}$, denoted by the tuple $v_\eta := [p_\eta, \tau_\eta^{\mathrm{a}}, \tau_\eta^{\mathrm{d}}]$. We first describe how to select $\tau_\eta^{\mathrm{a}}, \tau_\eta^{\mathrm{d}}$ for the general problem, then describe a procedure that improves the efficiency for deterministic cases. Figure 3 shows an example set of generated vertices for a) probabilistic and b) deterministic problems.

For probabilistic problem instances, each vertex has a time interval length equal to the time discretisation, i.e., $\tau_\eta^{\mathrm{d}} - \tau_\eta^{\mathrm{a}} = \Delta_{\mathrm{t}}$. For each position $p_\eta \in \mathcal{P}$, a vertex is created for each time step $\tau_\eta^{\mathrm{a}} \in \mathcal{T}$ where the target has a non-zero probability of being in range of the tracker, i.e.,

$$v_\eta = [p_\eta, \tau_\eta^{\mathrm{a}}, \tau_\eta^{\mathrm{d}} = \tau_\eta^{\mathrm{a}} + \Delta_{\mathrm{t}}] \in \mathcal{V} \text{ iff } \mathbb{E}\left[\tilde{f}(\|X(\tau_\eta^{\mathrm{a}}) - p_\eta\|)\right] > 0.$$

This definition is referred to as the probabilistic algorithm. An example of this vertex generation is illustrated in Fig. 3(a) overlaying a probabilistic target trajectory represented by a set of sample trajectories. Each vertical blue line segment is a vertex with the bottom at time $\tau_\eta^{\mathrm{a}}$ and the top at $\tau_\eta^{\mathrm{d}}$. In this example, the extreme samples are used to approximate the boundaries of the non-zero regions of the distribution.

For deterministic problem instances (defined in Sec. III-D), only a single vertex needs to be created for each contiguous subsequence of times where the target is in range of the tracker. More formally, $\mathrm{T}_i \subseteq \mathcal{T}$ denotes the set of all times where the target would be effectively monitored if the tracker were STOPPED at $p_i$ at time $t_l$, i.e., $\mathrm{T}_i := \{t_l \in \mathcal{T} : \tilde{f}(\|X_l - p_i\|) = 1\}$. Each $\mathrm{T}_i$ is then divided into non-overlapping subsequences, with each subsequence being a maximal run



(a) Probabilistic target trajectory (represented by a set of sample paths) and probabilistic algorithm vertices.

(b) Deterministic target trajectory and deterministic algorithm vertices.

Fig. 3. Examples showing graph vertices overlaying a target trajectory moving through a single spatial dimension, for the probabilistic and deterministic algorithms. Each vertical blue line segment represents a vertex in the search graph. Each vertex maps to a potential STOPPED position for the tracker, with arrival and departure times determined by the edges (Fig. 4).

of consecutive timesteps $(t_j, t_{j+1}, ..., t_{j+k}) \subseteq \mathrm{T}_i$. Each subsequence forms a new vertex $v_\eta$ in the search graph with $\tau^{\mathrm{a}}_\eta$ and $\tau^{\mathrm{d}}_\eta$ chosen as the subsequence start and end times, i.e.,

$$[p_\eta, \tau^{\mathrm{a}}_\eta, \tau^{\mathrm{d}}_\eta] = [p_i, t_j, t_{j+k} + \Delta_{\mathrm{t}}].$$

This definition is referred to as the deterministic algorithm. Fig. 3(b) illustrates that, in contrast to the probabilistic algorithm, each vertex can span multiple timesteps.

The deterministic algorithm is typically more efficient since fewer vertices are generated. This adjustment maintains optimality since, for the deterministic case, if an optimal solution path contains the position $p_i$ at time $t_i$ then it is optimal to stay at position $p_i$ for all timesteps consecutive to $t_i$ when the target is still in range. A proof of this guarantee is provided later in Lemma 3 after the edges have been introduced.

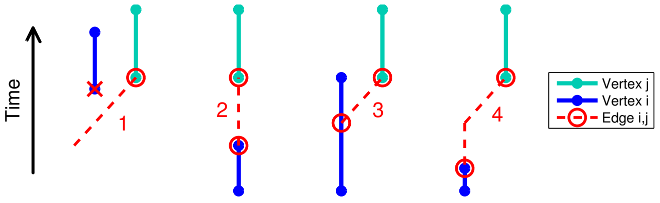


Fig. 4. Illustration of the edge categories described in Alg. 3 and Sec. V-B. From left to right: *infeasible*, *same position*, *smaller gap* and *larger gap*.

### B. Edges

A solution trajectory is represented by a path through the graph with consecutive vertices connected by directed edges $e_\eta \in \mathcal{E}$. An edge is denoted $e_\eta = \langle v_i, v_j \rangle$ and describes travelling from vertex $v_i$ at position $p_i$ to vertex $v_j$ at position $p_j$ at some time in the solution trajectory.

Each edge has an associated departure time $t^{\mathrm{d}}_{e_\eta} := t^{\mathrm{d}}_i$ and arrival time $t^{\mathrm{a}}_{e_\eta} := t^{\mathrm{a}}_j$ which describes the exact time the tracker moves from $p_i$ to $p_j$. We require $t^{\mathrm{a}}_\eta, t^{\mathrm{d}}_\eta$ satisfy

$$t^{\mathrm{a}}_\eta = \tau^{\mathrm{a}}_\eta < t^{\mathrm{d}}_\eta. \tag{4}$$

The key advantage of having a fixed arrival time $t^{\mathrm{a}}_\eta$ (4) for a vertex is that the calculations for an edge $e_\eta = \langle v_i, v_j \rangle$ do not depend on the choice of arrival time for a previous edge $e_m = \langle v_h, v_i \rangle$ or the path taken to or from an edge; therefore optimal sub-paths are additive and generally lead to globally optimal solutions. For the probabilistic algorithm, selecting $t^{\mathrm{a}}_\eta = \tau^{\mathrm{a}}_\eta$ is optimal relative to the temporal resolution since each vertex represents only stopping for a single time step at $p_\eta$. For the deterministic case, where a vertex represents a contiguous subsequence of in-range timesteps, this choice is still optimal (shown later in Lemma 3).

Each edge also has an associated weight $\omega_{i,j}$ which is defined as the monitoring effectiveness over the time interval $[\tau^{\mathrm{a}}_i, \tau^{\mathrm{a}}_j)$ if that edge were chosen, i.e.,

$$\omega_{i,j} := F_{[\tau^{\mathrm{a}}_i, \tau^{\mathrm{a}}_j)}. \tag{5}$$

Each edge is in one of four categories, which determines the edge weight and moving times. The conditions are derived directly from the geometric properties illustrated in Fig. 4. The calculations are listed in Alg. 3 and described as follows.

1) **Infeasible –** An edge is included if and only if the vertex $v_j$ is reachable from $v_i$, i.e., $\delta(p_i, p_j) \leq \tau^{\mathrm{a}}_j - \tau^{\mathrm{a}}_i$.
2) **Same Position –** The two vertices are at the same position and therefore merged into a single waypoint.
3) **Smaller Gap –** The gap between the vertices is smaller than $\delta(p_i, p_j)$; therefore there will be no time spent in the STOPPED state while not effectively monitoring.
4) **Larger Gap –** The gap is larger than $\delta(p_i, p_j)$; therefore there must be some time spent in the STOPPED state while not effectively monitoring.

**Remark 2.** The trajectory represented by an edge $\langle v_i, v_j \rangle$ may pass through another vertex $v_\eta = [p_\eta, \tau^{\mathrm{a}}_\eta, \tau^{\mathrm{d}}_\eta]$. This occurs if there exists a $v_\eta$ where $p_\eta = p_i$ and the time interval $[\tau^{\mathrm{a}}_\eta, \tau^{\mathrm{d}}_\eta]$ overlaps with $[t^{\mathrm{a}}_i, t^{\mathrm{d}}_i]$, or where $p_\eta = p_j$ and the time $[\tau^{\mathrm{a}}_\eta, \tau^{\mathrm{d}}_\eta]$ overlaps with $[t^{\mathrm{a}}_j, t^{\mathrm{d}}_j]$. For efficient computation, and

**Algorithm 3** Edge weight and time calculations for the four categories illustrated in Fig. 4 and described in Sec. V-B.

```
function EdgeCalculation($e_\eta = \langle v_i, v_j\rangle$)
    $\rho \leftarrow \mathbb{E}\left[\tilde{f}(\|X_\eta - p_i\|)\right] : t_\eta = \tau_i^{\text{a}}$
    $t_j^{\text{a}} \leftarrow \tau_j^{\text{a}}$
    if $\delta(p_i, p_j) \geq \tau_j^{\text{a}} - \tau_i^{\text{a}}$ then                ▷ infeasible
        Do not include $e_\eta$ in $\mathcal{E}$
    else if $p_i = p_j$ then                                   ▷ same position
        $t_i^{\text{d}} \leftarrow \tau_j^{\text{a}}$
        $\omega_{i,j} \leftarrow \rho \times \left(\tau_i^{\text{d}} - \tau_i^{\text{a}}\right)$
    else if $\delta(p_i, p_j) \geq \tau_j^{\text{a}} - \tau_i^{\text{d}}$ then           ▷ smaller gap
        $t_i^{\text{d}} \leftarrow \tau_j^{\text{a}} - \delta(p_i, p_j)$
        $\omega_{i,j} \leftarrow \rho \times \left(\tau_j^{\text{a}} - \tau_i^{\text{a}} - \delta(p_i, p_j)\right)$
    else                                                        ▷ larger gap
        $t_i^{\text{d}} \leftarrow \tau_j^{\text{a}} - \delta(p_i, p_j)$
        $\omega_{i,j} \leftarrow \rho \times \left(\tau_i^{\text{d}} - \tau_i^{\text{a}}\right)$
    return $[t_i^{\text{d}}, t_j^{\text{a}}, \omega_{i,j}]$                     ▷ depart, arrive, weight
```

without loss of optimality, the computations in Alg. 3 ignore the value of any $v_a$. Therefore, $\omega_{i,j}$ may underestimate $F_{[\tau_i^{\text{a}}, \tau_j^{\text{a}})}$. However, this case will be realised by a path $(\langle v_i, v_a\rangle, \langle v_a, v_j\rangle)$ where $\omega_{i,a}$ and $\omega_{a,j}$ comply with (5). Underestimating $\omega_{i,j}$ does not result in suboptimal solutions since a longest-path search will choose $(\langle v_i, v_a\rangle, \langle v_a, v_j\rangle)$ instead.

**Lemma 3. Optimal arrival time for deterministic cases:** *For deterministic problem instances (defined in Sec. III-D), if a path passes through $v_\eta$, then it is optimal for the solution trajectory to arrive at $p_\eta$ with $t_\eta^a$ chosen as $\tau_\eta^a$ (using the definitions for $t_\eta^a$ and $\tau_\eta^a$ from Secs. V-A2 and V-B).*

*Proof:* Consider the path consisting of a feasible edge $\langle v_i, v_j\rangle$, where $p_i \neq p_j$, for three cases: A - choose $t_j^{\text{a}} = t_{j_{\mathsf{A}}}^{\text{a}}$ where $\tau_j^{\text{a}} < t_{j_{\mathsf{A}}}^{\text{a}} \leq \tau_j^{\text{d}}$; B - choose $t_j^{\text{a}} = t_{j_{\mathsf{B}}}^{\text{a}}$ where $t_{j_{\mathsf{B}}}^{\text{a}} = \tau_j^{\text{a}}$; and C - choose $t_j^{\text{a}} = t_{j_{\mathsf{C}}}^{\text{a}}$ where $t_{j_{\mathsf{C}}}^{\text{a}} < \tau_j^{\text{a}}$. By this definition,

$$t_{j_{\mathsf{A}}}^{\text{a}} > t_{j_{\mathsf{B}}}^{\text{a}} = \tau_j^{\text{a}} > t_{j_{\mathsf{C}}}^{\text{a}}.$$

The following proof shows that B has a monitoring effectiveness greater than or equal to A and C.

Firstly, we show that B is a feasible choice; i.e., it does not require departing $p_i$ before the start time $\tau_i^{\text{a}}$. Consider the pair of start times $(\tau_i^{\text{a}}, \tau_j^{\text{a}})$. When the target moves in a straight line at maximum speed $\|\dot{x}\|_{\max}$ (i.e., gradient in Fig. 3(b)), the vertices will have start times with this same gradient between pairs, i.e.,

$$|\tau_j^{\text{a}} - \tau_i^{\text{a}}| = \frac{\|p_j - p_i\|}{\|\dot{x}\|_{\max}}.$$

If the target turns (e.g. upper half of Fig. 3(b)), or moves slower, this time difference must be larger; therefore generally

$$|\tau_j^{\text{a}} - \tau_i^{\text{a}}| \geq \frac{\|p_j - p_i\|}{\|\dot{x}\|_{\max}}.$$

Applying the speed assumption yields

$$|\tau_j^{\text{a}} - \tau_i^{\text{a}}| \geq |t_j^{\text{a}} - t_i^{\text{d}}| = \delta(p_i, p_j). \tag{6}$$

An exception could occur at the beginning of the mission (since $|\tau_j^{\text{a}} - \tau_i^{\text{a}}| = 0$ if $\tau_j^{\text{a}} = \tau_i^{\text{a}} = 0$); however the vertex adjustments described later in Sec. V-C ensure this will not prevent an optimal path from being chosen. From (6), it follows that if $t_j^{\text{a}} = t_{j_{\mathsf{B}}}^{\text{a}}$ then $t_i^{\text{d}} \geq \tau_i^{\text{a}}$, and therefore B is a feasible choice.

For A, the tracker departs $p_i$ at a time $\partial := t_{j_{\mathsf{A}}}^{\text{a}} - t_{j_{\mathsf{B}}}^{\text{a}}$ later than for B. Therefore B will spend $\partial$ less time at $p_i$ and $\partial$ more time at $p_j$ than A. In B, the extra time spent at $p_j$ is the interval $[t_{j_{\mathsf{B}}}^{\text{a}}, t_{j_{\mathsf{A}}}^{\text{a}})$. By definition of a vertex for the deterministic case (Sec. III-D), $F_{[t_{j_{\mathsf{B}}}^{\text{a}}, t_{j_{\mathsf{A}}}^{\text{a}})} = \partial$, which is maximal. A can not improve on this during the extra time at $p_i$, and therefore B has a greater or equal monitoring effectiveness than A. Note this assumes $\tau_i^{\text{a}} < \tau_j^{\text{a}}$; however, it follows from (6) and the triangle inequality assumption from Lemma 1 that an optimal path will not contain $\langle v_i, v_j\rangle$ if $\tau_i^{\text{a}} \geq \tau_j^{\text{a}}$, since $v_j$ would also be reachable from the vertex preceding $v_i$.

To achieve C, the tracker will spend more time at $p_j$ than for B. This extra time is before $\tau_j^{\text{a}}$, and therefore by definition of a vertex, $F_{[t_{j_{\mathsf{C}}}^{\text{a}}, \tau_j^{\text{a}})} = 0$, which is minimal; hence B has a greater or equal monitoring effectiveness than C. This shows that $t_j^{\text{a}} = \tau_j^{\text{a}}$ is optimal. Note that this assumes that there is no vertex $v_k$ where $p_k = p_j$ and $\tau_k^{\text{a}} < \tau_j^{\text{a}}$; however, if a $v_k$ exists then the planner has the option to choose the subsequence $(v_i, v_k, v_j)$ if this is feasible. For $(v_i, v_k, v_j)$, this proof also shows that $t_k^{\text{a}} = \tau_k^{\text{a}}$ is the optimal arrival time for $v_k$, and $t_j^{\text{a}}$ is irrelevant since $p_j = p_k$. □

### C. Start and End Conditions

The graph needs to be adjusted to ensure that the solution path satisfies the start and end constraints. This includes trimming or removing some of the edges, as well as defining the sets of start $v_{\text{start}} \in \mathcal{V}_{\text{start}}$ and end $v_{\text{end}} \in \mathcal{V}_{\text{end}}$ vertices.

The set of allowable start positions is calculated as $\mathcal{P}_{\text{start}} = \hat{\mathcal{Y}}_{\text{start}} \cap \mathcal{P}$. If a problem instance requires start positions that are not already in $\mathcal{P}$, then these positions should be added to $\mathcal{P}$ and corresponding vertices and edges added to the graph. For each $p_{\text{start}} \in \mathcal{P}_{\text{start}}$, if it is not in-range at time $t_1$ (i.e. $\mathbb{E}\left[\tilde{f}(\|X(t_1) - p_{\text{start}}\|)\right] = 0$), then additional vertices are generated between time $t_1$ and the first timestep where $p_{\text{start}}$ is in-range. For each $p_{\text{start}} \in \mathcal{P}_{\text{start}}$, the vertex with $\tau_i^{\text{a}} = t_1$ is included in the set of possible start vertices $v_{\text{start}} \in \mathcal{V}_{\text{start}}$.

To ensure the search always selects a path that begins at $v_{\text{start}} \in \mathcal{V}_{\text{start}}$, all other vertices $v_i$ are adjusted using the rules described in Alg. 4. In the first case, $v_i$ is reachable from at least one $p_{\text{start}} \in \mathcal{P}_{\text{start}}$ and therefore no adjustment is made. The second case removes unreachable vertices. The third case trims all $v_i$ that are reachable only at some time after $\tau_i^{\text{a}}$. The

**Algorithm 4** Vertex set adjustments for the start condition

```
for each $v_i \in \mathcal{V} \setminus \mathcal{V}_{\text{start}}$ do
    if $\exists p_{\text{start}} \in \mathcal{P}_{\text{start}} : \tau_i^{\text{a}} \geq \delta(p_{\text{start}}, p_i)$ then
        continue                                                ▷ keep $v_i$
    else if $\forall p_{\text{start}} \in \mathcal{P}_{\text{start}} : \tau_i^{\text{d}} \leq \delta(p_{\text{start}}, p_i)$ then
        $\mathcal{V} \leftarrow \mathcal{V} \setminus v_i$                                    ▷ remove $v_i$
    else
        $\tau_i^{\text{a}} \leftarrow \min_{p_{\text{start}} \in \mathcal{P}_{\text{start}}} \delta(p_{\text{start}}, p_i) + t_1$                  ▷ trim $v_i$
```

**Algorithm 5** Sweep-plane graph search: forward pass

```
1: function SweepPlane($\mathcal{V}, \mathcal{E}, \{\omega\}, \{t_i^{\rm a}\}$)
2:     $\Omega_{\rm start} \leftarrow 0,\ \forall v_{\rm start} \in \mathcal{V}_{\rm start}$
3:     for $t = t_1, t_2, ..., t_N$ do
4:         for each $v_i \in \mathcal{V} \setminus \mathcal{V}_{\rm start} : t_i^{\rm a} = t$ do
5:             $\mathcal{E}_i \leftarrow \{\epsilon : \langle v_\epsilon, v_i \rangle \in \mathcal{E}\}$          ▷ edges into $v_i$
6:             $\psi_i \leftarrow \text{argmax}_{\epsilon \in \mathcal{E}_i} [\Omega_\epsilon + \omega_{\epsilon,i}]$   ▷ optimal edge
7:             $\Omega_i \leftarrow \Omega_{\psi_i} + \omega_{\psi_i,i}$                ▷ path weight
8:     return $[\{\Omega\}, \{\psi\}]$           ▷ path weights, back-pointers
```

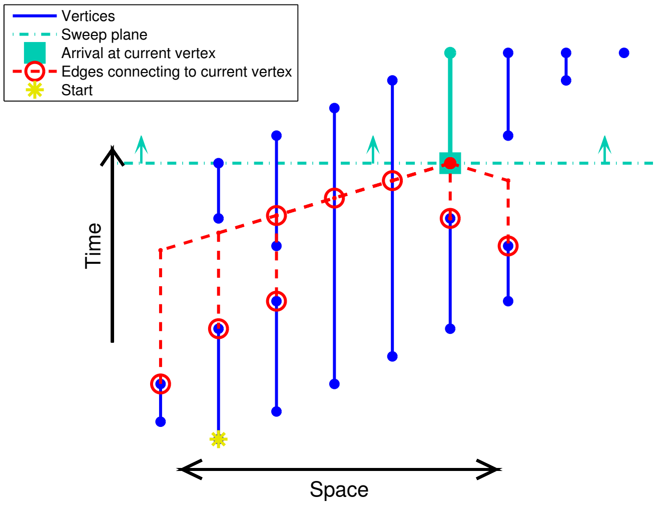


Fig. 5. Sweep-plane at a particular time instant showing all feasible edges into the current vertex, analogous to Fig. 3(b) but including the start condition adjustment (for a singleton $\mathcal{V}_{\rm start}$). Constant velocity travel time.

third case is necessary for the deterministic algorithm where each vertex may span multiple timesteps, in order to ensure the Lemma 3 result holds. For simplicity, the reachability calculations assume the triangle inequality holds for travel times (Condition 1), although the algorithm could readily be adapted for other cases.

Similarly, the set of allowable end positions is calculated as $\mathcal{P}_{\rm end} = \hat{\mathcal{Y}}_{\rm end} \cap \mathcal{P}$. For each position $p_{\rm end} \in \mathcal{P}_{\rm end}$, the vertex with the latest departure time $\tau_{\rm i}^{\rm d}$ is added to the set of possible end vertices $v_{\rm end} \in \mathcal{V}_{\rm end}$. The optimal start and end positions (for non-singleton $\mathcal{P}_{\rm start}$ or $\mathcal{P}_{\rm end}$) are found by the algorithm in the following section.

## VI. Sweep-Plane Algorithm

The optimal tracker trajectory is found by searching for the longest-path through the spatiotemporal graph. For general graphs, a longest-path search is NP-hard. However, optimal polynomial-time algorithms exist if the graph is a directed acyclic graph (DAG), since a topological ordering[1] of $\mathcal{V}$ exists. The spatiotemporal graph generation defined in Sec. V reduces the problem to a longest-path search through a DAG, and therefore, the solution can be found in $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$ time. This section describes a longest-path search, which can be visualised as a sweep-plane moving through time. The algorithm

**Algorithm 6** Backtracking to find the optimal trajectory

```
1: function BackTrack($\{\Omega\}, \{\psi\}, \mathcal{V}, \mathcal{E}, \mathcal{V}_{\rm start}, \mathcal{V}_{\rm end}$)
2:     $\rho_i \leftarrow \mathbb{E}\left[\tilde{f}(\|X_\eta - p_i\|)\right] : t_\eta = \tau^{\rm a}_{\rm end},\ \forall v_i \in \mathcal{V}_{\rm end}$
3:     $v_\eta \leftarrow \underset{v_i \in \mathcal{V}_{\rm end}}{\text{argmax}} \left[\Omega_i + \rho_i \times \left(\tau_i^{\rm d} - \tau_i^{\rm a}\right)\right]$      ▷ end vertex
4:     $F \leftarrow \Omega_\eta + \rho_\eta \times \left(\tau_\eta^{\rm d} - \tau_\eta^{\rm a}\right)$ ▷ monitoring effectiveness
5:     repeat
6:         $v_{\rm prev} \leftarrow \psi_\eta$        ▷ previous vertex along trajectory
7:         $e \leftarrow \langle v_{\rm prev}, v_\eta \rangle$                ▷ find edge into $v_\eta$
8:         $\hat{y}_i \leftarrow p_\eta$                       ▷ waypoint position
9:         $t_i^{\rm a} \leftarrow t_e^{\rm a}$                        ▷ waypoint arrive
10:        $t_{i-1}^{\rm d} \leftarrow t_e^{\rm d}$             ▷ previous waypoint depart
11:        $v_\eta \leftarrow v_{\rm prev},\ i \leftarrow i - 1$   ▷ move to previous vertex
12:    until $v_\eta \in \mathcal{V}_{\rm start}$
13:    $\hat{y}_i \leftarrow p_\eta, t_i^{\rm a} \leftarrow t_1$
14:    return $[[\{\hat{y}\}, \{t^{\rm a}\}, \{t^{\rm d}\}], F]$  ▷ trajectory, path weight
```

includes finding the optimal $v_{\rm start} \in \mathcal{V}_{\rm start}$ and $v_{\rm end} \in \mathcal{V}_{\rm end}$. We then analyse the optimality and time complexity of the full algorithm, and discuss practical considerations.

### A. Forward Pass

The longest-path search begins with a forward pass through the graph that visits the nodes in topological order. A topological ordering of the vertices can be found by visiting $v_i$ in order of ascending time $t = \tau_i^{\rm a}$. This can be thought of as a sweeping plane as illustrated in Fig. 5 and described in Alg. 5. The sweep-plane represents a plane covering $\mathcal{P}$ at a particular time $t$, and moves linearly through increasing time $\mathcal{T}$ (*line 3*). A vertex $v_i$ is explored once the sweep-plane reaches $t = \tau_i^{\rm a}$ (*line 4*). For efficient evaluation of the vertex set in *line 4*, $\mathcal{V}$ should be pre-sorted by ascending $\tau_i^{\rm a}$. When $v_i$ is explored (*line 5*), all edges $e_\epsilon$ leading in to $v_i$ are compared (*line 6*) and the optimal previous vertex with an edge into each $v_i$ is denoted $\psi_i$. The sum of weights along the optimal path leading to vertex $v_i$ through edge $e_{\psi_i}$ is calculated recursively and denoted $\Omega_i$ (*line 7*).

### B. Backtracking

Lastly, the optimal solution path is found by backtracking from a $v_{\rm end} \in \mathcal{V}_{\rm end}$ to a $v_{\rm start} \in \mathcal{V}_{\rm start}$, as described in Alg. 6. The end vertex is chosen as the vertex in $\mathcal{V}_{\rm end}$ with the highest path weight (*line 3*). The algorithm proceeds by recursively following the back-pointers $\psi$ until a $v_{\rm start} \in \mathcal{V}_{\rm start}$ is reached (*lines 5-12*). Backtracking will always lead to a $v_{\rm start} \in \mathcal{V}_{\rm start}$ due to the adjustments in Sec. V-C. The expected monitoring effectiveness is $F = \Omega_{\rm end} + \rho \times \left(\tau^{\rm d}_{\rm end} - \tau^{\rm a}_{\rm end}\right)$, since the weight of the end vertex is not accounted for by the edge weights.

### C. Analysis

*1) Graph search optimality:* For each $v_i$, the forward pass calculates the preceding vertex $\psi_i$ and the sum of edge weights $\Omega_i$ for the optimal path from any $v_{\rm start} \in \mathcal{V}_{\rm start}$ to $v_i$, if the mission were to end at time $\tau_i^{\rm a}$. The algorithm recursively solves optimal sub-problems until the optimal path through the graph is found, and therefore the full problem is solved optimally relative to the resolution of the graph.

[1] If there exists a path from vertex $v_i$ to $v_j$, then $v_i$ precedes $v_j$ in a topological sort.

*2) Time complexity:* Time complexity is analysed as follows. Let the spatial resolution be $|\mathcal{P}|$, temporal resolution $|\mathcal{T}|$, number of vertices $|\mathcal{V}|$ and number of edges $|\mathcal{E}|$. The complexity for generating the set of vertices is $\mathcal{O}(|\mathcal{V}|) = \mathcal{O}(|\mathcal{P}| \cdot |\mathcal{T}|)$ and for the edges is $\mathcal{O}(|\mathcal{E}|) = \mathcal{O}(|\mathcal{V}|^2)$. Therefore the computation time for generating the graph is $\mathcal{O}(|\mathcal{V}|^2) = \mathcal{O}(|\mathcal{P}|^2 \cdot |\mathcal{T}|^2)$. The topological sort has complexity $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$, the graph search forward pass has complexity $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$, and the backtracking has complexity $\mathcal{O}(|\mathcal{T}|)$. Therefore the computation time of the sweep-plane algorithm overall is $\mathcal{O}(|\mathcal{P}|^2 \cdot |\mathcal{T}|^2)$.

*3) Practical considerations:* In practice there is a trade-off between solution quality and runtime of the algorithm. Firstly, the algorithm requires finite spatial and temporal resolutions, since the computation requires a finite set of vertices. This is not limiting since, in practice, there is little benefit in having a resolution higher than the positioning accuracy of the tracker vehicle. Secondly, for some prediction models it may be impractical to solve the observation value integral (1) exactly. In our experiments we solve (1) using Monte Carlo integration, such that the belief of the target's trajectory is approximated by a set of particles.

## VII. Experiments

This section describes simulation experiments that illustrate the behaviour of the algorithm and advantages of the probabilistic formulation. First, we compare the current planning algorithm to previous algorithms for deterministic AUV missions [23]. Second, we give an illustrative example that demonstrates the advantage of the probabilistic formulation of the problem. Third, we evaluate planning with probabilistic predictions that model temporal uncertainty. Later in Sec. VIII, we demonstrate planning with two realistic trajectory and observation model implementations for application case studies.

### A. Missions and Parameters

The following simulations were performed using the same target trajectories and parameter values as the real AUV missions described in [23]. Two hour-long AUV missions are considered as target trajectories, named Middle Harbour (depicted in Fig. 6) and Jervis Bay. In [23], these missions were executed by a REMUS 100 AUV. Both missions alternate between densely scanning local regions of interest and moving in straight lines between these regions. Two extreme cases for the trajectory are also considered: *circular* is a circular path with radius slightly less than the monitoring range, and *linear* is a straight path.

Parameter values are as follows: $r = 200\,\mathrm{m}$ monitoring range for an $r$-disk communication model (defined in Sec. III-D), $2\,\mathrm{m/s}$ constant target speed, $25\,\mathrm{m}$ grid spacing, $\Delta_\mathrm{t} = 10\,\mathrm{s}$ time steps, travel time between tracker waypoints

$$\delta(\hat{y}_i, \hat{y}_j) = \frac{\|\hat{y}_j - \hat{y}_i\|}{\|\dot{y}\|} + T_\mathrm{pen},$$

with $\|\dot{y}\| = 5\,\mathrm{m/s}$ tracker speed and $T_\mathrm{pen} = 30\,\mathrm{s}$ constant penalty for deploying and retrieving the monitoring hardware, and fixed start and end conditions, $\hat{y}_1 = \mathbb{E}[X_1]$ and $\hat{y}_M = \mathbb{E}[X_N]$ respectively, so that the tracker is in an appropriate position to deploy and recover the target vehicle. Computation times are shown for an unoptimised MATLAB implementation, running on a single core of an Intel i7 processor.

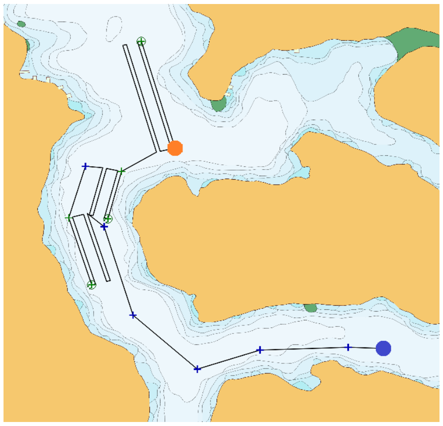


Fig. 6. Deterministic AUV mission plan from [23] used in the simulations. It is a 1 hour mission for a REMUS 100 AUV in Middle Harbour, Sydney. Starts at orange dot (near middle) and ends at blue dot (bottom right).

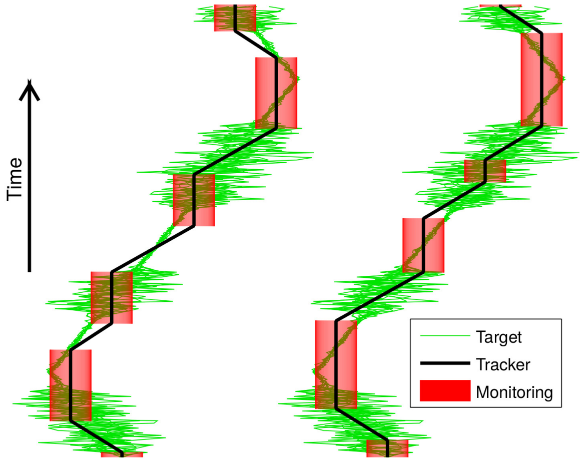
TABLE I
Simulation results for deterministic target trajectories. The two planners output identical solution trajectories.

| Mission | $F/T$ | $M$ | Deterministic $\|\mathcal{V}\|$ | Deterministic time (s) | Probabilistic $\|\mathcal{V}\|$ | Probabilistic time (s) |
|---|---|---|---|---|---|---|
| Middle Harbour | 79.5 | 8 | 2860 | 0.5 | 49483 | 101 |
| Jervis Bay | 79.2 | 7 | 3121 | 0.6 | 59146 | 144 |
| *circular* | 95.8 | 3 | 1203 | 0.3 | 27338 | 32 |
| *linear* | 52.2 | 6 | 430 | 0.2 | 8062 | 3.7 |

Columns: Monitoring effectiveness as a percentage of mission duration; Num. stopping locations $M$; Num. vertices $|\mathcal{V}|$; Computation time (s).

### B. Deterministic Target Trajectory

Table I shows simulation results for four deterministic target trajectories. The deterministic and the probabilistic algorithms output the same solution trajectories, however the deterministic algorithm had a lower computation time due to the reduced number of vertices. The *linear* trajectory resulted in the lowest monitoring effectiveness since a straight-line target trajectory means the tracker cannot cut corners to reduce travel time. The *linear* trajectory required the fewest number of vertices since the convex hull of a straight line has the smallest area. The computation time is approximately quadratic in $|\mathcal{V}|$ (regression fit with $R^2 > 0.99$), which agrees with the theoretical analysis. The algorithm shows a small improvement in the monitoring effectiveness over the greedy algorithm and genetic algorithm results reported in [23]. The key advantage of the proposed sweep-plane algorithm is guaranteed and faster runtime (the deterministic algorithm is approximately 50 times faster than the genetic algorithm), and provably optimal solutions, as well as the applicability to probabilistic scenarios as demonstrated in the following experiments.



(a) Deterministic planner (b) Probabilistic planner

Fig. 7. Comparing planning with a deterministic model to planning with a probabilistic model. Green lines are sample target trajectories drawn from the probabilistic model. Red regions represent the monitoring range around the chosen stopping locations. Probabilistic planner achieves higher monitoring effectiveness since it selects regions with low spatial uncertainty.

### C. Planning with Uncertainty

We demonstrate how planning while taking into account an accurate model for the uncertainty of the target trajectory improves the monitoring effectiveness. Figure 7 presents a target mission that alternates between sections with high spatial uncertainty and low spatial uncertainty.

Figure 7(a) shows the optimal stopping locations for the tracker if there were no uncertainty in the target trajectory. Figure 7(b) shows the solution when planning with a probability distribution $D_i$ that accurately models the uncertainty. The advantage of the probabilistic planning is that it chooses to stop at and stay longer in the regions with lower spatial uncertainty. For a Monte Carlo simulation drawing 10000 sample target trajectories, the deterministic planner has a mean monitoring effectiveness (as a percentage of mission duration) $F/T = 47.5\,\%$, while the probabilistic planner improves on this with $F/T = 54.1\,\%$. The solution path length given by the deterministic planner overestimates the expected monitoring effectiveness; conversely, the probabilistic planning accurately predicts the expected monitoring effectiveness.

### D. Probabilistic Trajectory with Temporal Uncertainty

Now we consider an example probabilistic target trajectory for an agent with uncertain speed. For a target with accurate localisation, uncertainty in position is usually due to variance in speed, rather than deviation from the path. To describe this, at time $t_i$ the target is a distance $d_i$ along the path from the start. We define $d_i$ by the recursive equation

$$d_{i+1} = d_i + \Delta_{\mathrm{t}} \dot{d}_i, \qquad (7)$$

where the speed $\dot{d}_i$ along the path at any time instance is assumed to be drawn from a Gaussian distribution that is independent of other time instances:

$$\dot{d}_i \sim \mathcal{N}\left(\|\dot{x}\|_{\mathrm{ave}}, \sigma^2/\Delta_{\mathrm{t}}\right).$$

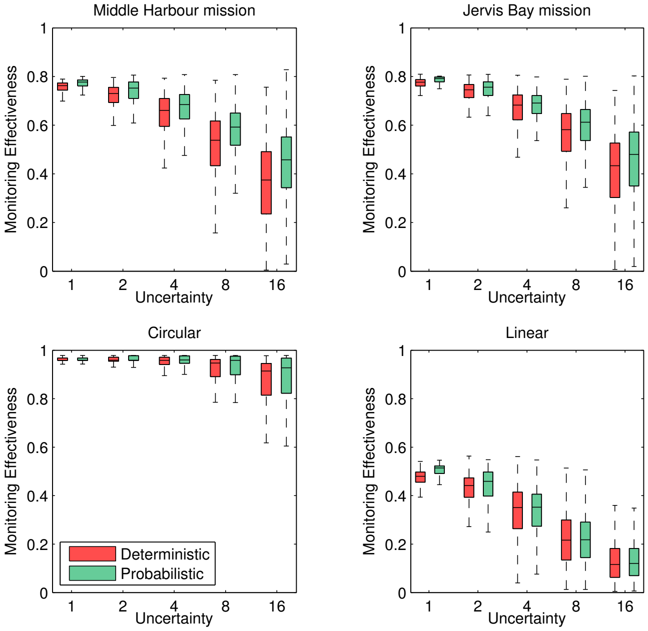


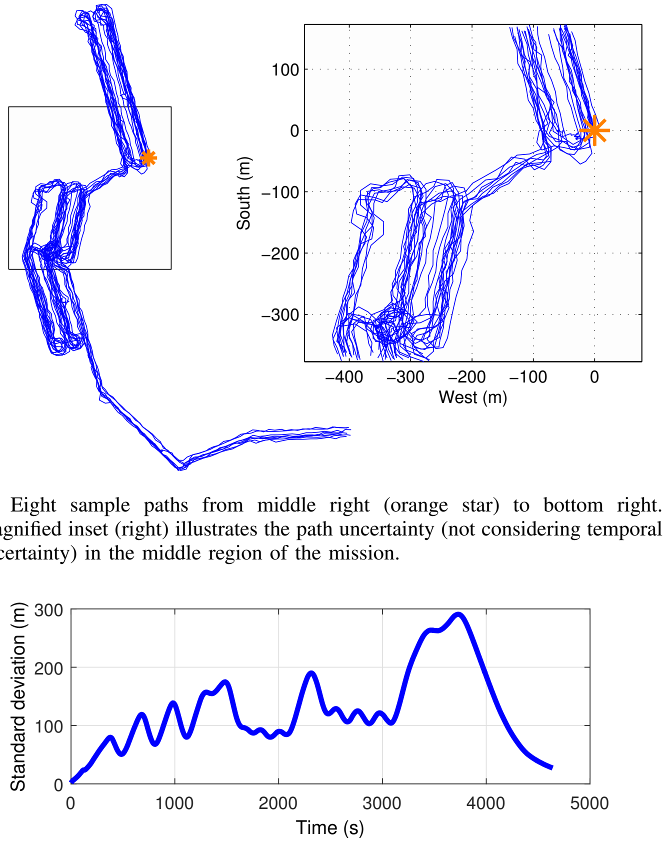
Fig. 8. Monte Carlo simulation results for a probabilistic target with uncertain speed (10000 samples); planning with (right bars) and without (left bars) taking into account the uncertainty model (8). $F/T$ on vertical axes; $\sigma_{\text{rate}}$ (uncertainty growth rate) on horizontal axes. Error bars show the sample minima, quartiles and maxima.

The general solution to (7) (distance travelled along the path), for $d_1 = 0$ with zero uncertainty, is also Gaussian, with mean and variance increasing linearly over time:

$$\mu_i = \|\dot{x}\|_{\mathrm{ave}} t_i \quad \text{and} \quad \Sigma_i = \sigma^2 t_i. \qquad (8)$$

Figure 8 shows the results of Monte Carlo simulations performed by drawing 10000 sample target trajectories directly from (7), with the objective function evaluated for the planned tracker trajectory. Planning was performed using the mean only with no uncertainty (left bars) or using the uncertainty model (8) (right bars). The horizontal axes shows varying speed uncertainty $\sigma \propto \sigma_{\text{rate}}$, where $\sigma_{\text{rate}}$ is the standard deviation of completion time in minutes for a 1 hour mission.

The monitoring effectiveness is significantly higher when planning using the uncertainty model, since the planner can choose to stop longer in regions with low spatial uncertainty. A single-tailed paired *t*-test confirms this performance improvement ($p < 0.001$) for all 20 missions except *linear* with $\sigma_{\text{rate}} \geq 4$. The probabilistic planning did not achieve significant improvements for the *linear* missions since there are no mission portions with relatively low spatial uncertainty (e.g., where the path folds back on itself). The monitoring effectiveness for all scenarios decreased as the uncertainty increased, since the probability of the target being within the 200 m communication radius at any position decreases. The monitoring effectiveness for the *circular* mission is less affected by increasing uncertainty since all possible positions for the target are within 200 m of a stopping location in the centre of the circle. The *circular* mission does not quite reach 100 % monitoring effectiveness since the tracker is required to travel from the start location and to the end location.



(a) Eight sample paths from middle right (orange star) to bottom right. Magnified inset (right) illustrates the path uncertainty (not considering temporal uncertainty) in the middle region of the mission.

(b) Spatial uncertainty over the mission duration. Measured as standard deviation of position for 1000 sample trajectories.

Fig. 9. Example predicted AUV trajectory given by the prediction model described in the case study, for the Middle Harbour mission (Fig. 6).

## VIII. Application Case Studies

In this section we present application case studies in AUV and pedestrian monitoring to demonstrate the relevance and applicability of the problem formulation and algorithm to real-world scenarios. The purpose of the case studies is to: 1) demonstrate feasibility of the problem formulation and algorithm for realistic scenarios; 2) formulate implementations of realistic probabilistic prediction and observation models; 3) detail choices made while implementing our planner; 4) present extensive simulated experiments under various modelling assumptions; and 5) evaluate and discuss the simulation results with real data and Monte Carlo simulations.

### A. AUV Mission Monitoring

Our first case study is for a mission monitoring scenario where an AUV is monitored by a surface vessel. We first formulate a realistic probabilistic model of the scenario, and then evaluate the performance of the algorithm under various modelling assumptions. The probabilistic formulation features a realistic probabilistic trajectory model (illustrated in Fig. 9) and an underwater acoustic communication model (Fig. 10). The formulation takes into account various causes of uncertainty in typical AUV missions in ocean environments, including localisation error, ocean currents, unpredictable mission pauses and unreliable communication. The formulation is motivated by our experiences with a REMUS 100 AUV in ocean environments, although is general enough to be adapted for other scenarios. The following simulation results show the advantages of taking into account these uncertainties when planning the trajectory of the monitoring surface vessel.

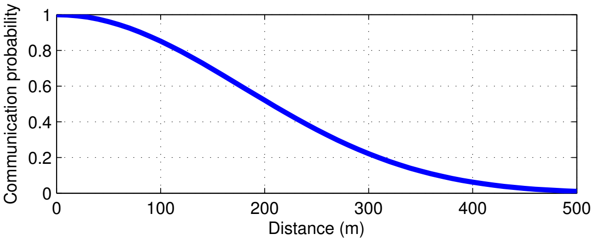


Fig. 10. Underwater acoustic communication model used in the AUV mission monitoring case study.

*1) Simulation scenarios:* We compare planning with and without a probabilistic AUV trajectory model, and with and without an acoustic communication model. The probabilistic trajectory model, detailed below, generates a set of 100 Monte Carlo sample paths to represent the belief of the AUV's trajectory. The deterministic trajectory model is generated with a single simulation that assumes zero uncertainty. The acoustic communication model, detailed below, describes the probability of successful communication for a given distance between the AUV and the surface vehicle. The planning scenarios without the communication model use an $r$-disk model with $r = 264.2\,\text{m}$, so that the two models have equal centroids. The tracker parameters are the same as in Sec. VII-A, and planning is performed using the probabilistic algorithm in all cases. The following subsections describe in detail our formulation for implementing realistic prediction and communication models.

*2) AUV prediction model implementation:* The prediction model for the AUV's motion is formulated probabilistically by adding various random disturbances to the deterministic mission plans in Sec. VII. The sequence of spatial probability distributions $(D_1, D_2, ..., D_N)$ of the AUV's position is represented as a set of Monte Carlo sample trajectories of the following model. Each sample trajectory is calculated by iteratively: 1) updating the state by sampling a stochastic kinematics model, 2) adding localisation noise, and then 3) updating the controller using either a closed-loop control policy or executing a surfacing behaviour. Fig. 9(a) shows an example set of sample paths and Fig. 9(b) shows the spatial uncertainty (caused by path uncertainty and temporal uncertainty) over the duration of the Middle Harbour mission.

The stochastic kinematics model for the AUV is described as follows. The AUV moves through $\mathbb{R}^2$ with position coordinates $(x_x, x_y)$ relative to a fixed earth frame and heading $\theta$. The standard unicycle robot kinematics model, with speed $v$ and angular velocity $\omega$, is extended to include varying ocean currents $(\dot{c}_x, \dot{c}_y)$, speed and angular velocity control uncertainty ($\varepsilon_v$ and $\varepsilon_\omega$ respectively) and a maximum angular

TABLE II
SIMULATION RESULTS WITH THE PROBABILISTIC AUV PREDICTION MODEL AND UNDERWATER ACOUSTIC COMMUNICATION MODEL.

| | Planning Scenario | | Results | | | Computation | |
|---|---|---|---|---|---|---|---|
| Mission | Predictions | Communications | $F/T$ full model | $F/T$ planner | $M$ | $\lvert\mathcal{V}\rvert$ | time (s) |
| Middle Harbour | Prob. | Acoustics | **68.05** | 67.90 | 7 | 61,265 | 177 |
| | Prob. | $r$-disk | 64.14 | 82.06 | 6 | 105,947 | 324 |
| | Det. | Acoustics | 65.13 | 73.64 | 7 | 61,052 | 131 |
| | Det. | $r$-disk | 57.78 | 87.51 | 7 | 96,350 | 267 |
| Jervis Bay | Prob. | Acoustics | **65.83** | 65.57 | 10 | 80,401 | 309 |
| | Prob. | $r$-disk | 57.19 | 81.94 | 8 | 142,692 | 603 |
| | Det. | Acoustics | 61.18 | 74.43 | 9 | 85,137 | 245 |
| | Det. | $r$-disk | 53.58 | 86.19 | 7 | 133,501 | 514 |
| *circular* | Prob. | Acoustics | **57.68** | 57.68 | 13 | 54,294 | 127 |
| | Prob. | $r$-disk | 48.23 | 90.43 | 5 | 99,569 | 280 |
| | Det. | Acoustics | 52.37 | 67.46 | 14 | 53,778 | 105 |
| | Det. | $r$-disk | 50.06 | 97.29 | 3 | 86,099 | 205 |
| *linear* | Prob. | Acoustics | **55.35** | 54.32 | 7 | 14,233 | 54 |
| | Prob. | $r$-disk | 53.20 | 60.97 | 6 | 20,061 | 40 |
| | Det. | Acoustics | 53.53 | 60.85 | 7 | 10,220 | 26 |
| | Det. | $r$-disk | 50.15 | 66.94 | 5 | 13,079 | 24 |

Columns: Monitoring effectiveness as a percentage of mission duration for probabilistic simulator and communication model; Objective function value (as a percentage of mission duration) output by planner, with respect to the planning scenario rather than the full model; Num. stopping locations $M$; Num. vertices $|\mathcal{V}|$; Computation time (s). Best results with respect to full model are in bold.

velocity control $\bar{\omega}$, giving the first-order equations:

$$\begin{aligned}\dot{x}_x &= (v+\varepsilon_v)\cos\theta + \dot{c}_x\\ \dot{x}_y &= (v+\varepsilon_v)\sin\theta + \dot{c}_y\\ \dot{\theta} &= \omega + \varepsilon_\omega,\ -\bar{\omega} \le \omega \le \bar{\omega}.\end{aligned}$$

The exact position of the AUV is not known and therefore the following controller is instead a function of the position estimate $(\tilde{x}_x, \tilde{x}_y)$. The position estimate has localisation error $(\varepsilon_x, \varepsilon_y)$, such that $(\tilde{x}_x, \tilde{x}_y) = (x_x + \varepsilon_x, x_y + \varepsilon_y)$.

We model the AUV's controller using the non-linear feedback control policy in [37, Ch. 9.3] that controls the angular velocity $\omega$. This policy has the goal of reducing the difference between the current position/heading and the projection onto the desired path of the deterministic mission plan. This policy is asymptotically stable for the standard unicycle model under standard assumptions [37], and simulations suggest it is also suitable for our extended kinematics model.

Some AUVs require pausing the mission and surfacing to, for example, receive a GPS fix or transmit data. We model unpredictable surfacing events as a stationary Poisson point process [57] with average rate $\lambda$. The length of time that the AUV surfaces for each event is drawn from a known probability distribution. The AUV is unpowered while surfacing and therefore drifts with the ocean currents, i.e., $\dot{x}_x = \dot{c}_x$, $\dot{x}_y = \dot{c}_y$ and $\dot{\theta} = 0$. Additionally, we assume that the localisation uncertainty is reset to zero while surfacing, i.e., $\varepsilon_x = \varepsilon_y = 0$.

The parameters of the model for our simulations are defined as follows. All probability distributions are Gaussian with mean $\mu$ and standard deviation $\sigma$, except where stated. The ocean currents in each axis have $\mu = 0$ m/s, $\sigma = 1$ m/s (although environment specific models [32] could be used if available), speed has $\mu = 2$ m/s, $\sigma = 0.5$ m/s, maximum angular velocity $\bar{\omega} = \pi/32$ rad/s, angular velocity error has $\mu = 0, \sigma = \pi/64$ rad/s, localisation error accumulates linearly over time (e.g. due to using dead-reckoning) according to

$$\begin{aligned}\varepsilon_x^{t+1} &= \varepsilon_x^t + \mathcal{N}\left(0, \varsigma^2/\Delta_{\mathrm{t}}\right)\\ \varepsilon_y^{t+1} &= \varepsilon_y^t + \mathcal{N}\left(0, \varsigma^2/\Delta_{\mathrm{t}}\right),\end{aligned}$$

with $\varsigma = 0.3$ m and $\varepsilon_x = \varepsilon_y = 0$ initial conditions, average surfacing rate $\lambda = 1$ /hour and a surfacing period uniformly distributed between 0 and 5 min.

*3) Acoustic communication model implementation:* Acoustic communication in ocean environments between an AUV and a surface vessel is highly unreliable. We simulate communication using a realistic underwater acoustic communication model proposed in [24]. This model defines the probability of successful communication of a packet of data as a function of distance between two locations. The model parameters are functions of various characteristics of the environment and the communication hardware, such as transmission power, frequency, water depth, wind speed and shipping noise. The resulting model for chosen parameters is shown in Fig. 10.

*4) Planner implementation:* The planner requires evaluating the expected monitoring effectiveness (1) for stopping and monitoring at each candidate stopping location and time. For this case study, we compute the integral (1) as the normalised sum of the communication probabilities (given by the communication model) at the sampled AUV positions (given by the AUV trajectory prediction samples).

The planner also requires a predefined set of candidate stopping locations, and the algorithm is optimal with respect to this set. For this case study we employ an adaptive grid spacing so that the planner focuses more attention on promising regions. This is achieved by setting the grid spacing to 25 m in regions and times where the probability of successful communication $\mathbb{E}[\tilde{f}]$ is greater than 0.6, 50 m for $\mathbb{E}[\tilde{f}] > 0.4$, 100 m for $\mathbb{E}[\tilde{f}] > 0.2$, and ignoring regions where $\mathbb{E}[\tilde{f}] \le 0.2$.

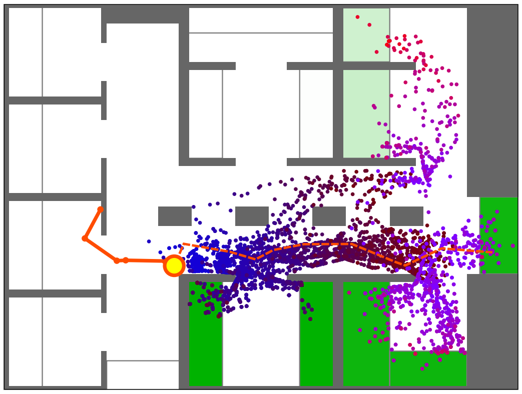


Fig. 11. The simulated office environment with an example observed trajectory (solid-orange), current position (yellow circle), 15 goal regions (green rectangles, with shade proportional to prediction probability), future trajectory prediction particles (colour changes over time), and the ground-truth future trajectory (dashed orange).

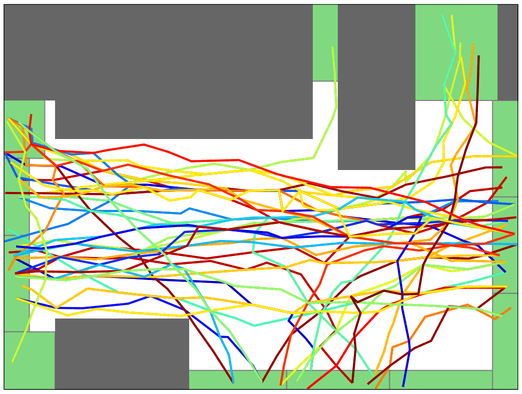


Fig. 12. The environment of the dataset [38] with 50 example pedestrian trajectories and 12 selected goal regions (green), as presented in [20].

*5) Results and discussion:* Table II shows the simulation results for the four example missions and four planning scenarios. For each planning scenario, the first results column shows the monitoring effectiveness (as a percentage of mission duration) of the planned tracker trajectory when simulating the full prediction and communication models. For all missions, planning with the full prediction and communication models (1st row) achieves the highest monitoring effectiveness, which highlights the advantage of the probabilistic formulation.

Planning with deterministic predictions and the acoustic model (3rd row) achieves a moderate improvement in monitoring effectiveness over planning with probabilistic predictions and the $r$-disk model (2nd row). This suggests the communication model is more valuable than the probabilistic predictions in this instance; however, using both is the most valuable.

The objective function computed by the planner (2nd results column), which is measured relative to the current planning scenario rather than the full model, significantly overestimates the monitoring effectiveness relative to the full probabilistic model (1st column). This overestimate is worse in the 4th rows since the deterministic planning scenario is most different to the full probabilistic model.

The number of stopping locations $M$ is similar for all planning scenarios for three of the missions. However, for the *circular* mission, planning with the acoustics model results in more than double the number of stops than the $r$-disk model. This is because the acoustics model favours being close to the AUV and therefore the planner chooses to make many short stops near the predicted AUV position. The $r$-disk model treats all samples within the communication range as equally effective, and (for the deterministic trajectory case) all samples are within communication range of a single stopping location in the centre of the circle. Although the time spent deploying and retrieving the monitoring hardware is longer for the acoustic model case, the monitoring effectiveness is improved since the chosen stopping locations and times have a higher probability of successful communication.

The computation time is approximately quadratic in the number of vertices in the search graph $|\mathcal{V}|$ (quadratic regression has $R^2 = 0.97$), which agrees with the time complexity analysis. In most cases, using deterministic predictions is slightly faster than probabilistic predictions, since it takes less time to generate the trajectory model and evaluate the observation value (1). Note that due to the filtering for the adaptive grid-spacing, the acoustics model was significantly faster than the $r$-disk model. The computation time for the fourth scenario could be reduced by using the deterministic planner instead (tested in Sec. VII-B) without compromising performance. In practical applications where computation time is more important than monitoring effectiveness, this planning scenario may be more desirable.

### B. Pedestrian Monitoring in Cluttered Environments

Our second case study is for a mission monitoring scenario where an aerial or ground vehicle monitors or aids a pedestrian or other similarly behaving agent. Similar to the first case study, we first formulate a realistic probabilistic model of the scenario, and then evaluate the performance of the algorithm under various modelling assumptions. The probabilistic formulation features a multi-modal intention-inference trajectory prediction model [20] (illustrated in Fig. 11), and a visibility-based observation model that takes into account occlusions in a cluttered environment. Motivating scenarios include filming a sporting event with a mobile camera but stationary filming locations, tracking an animal, monitoring boats in a cluttered harbour, aiding a disabled person, or monitoring other robots moving around a warehouse.

*1) Simulation scenarios:* The simulation results compare planning with and without taking into account the probabilistic prediction model, as well as with and without taking into account the occlusions in the observation model. The probabilistic predictions, detailed below, are represented by a set of 100 Monte Carlo sample paths, which are biased random walks through a probabilistic roadmap (PRM). The deterministic prediction is defined as the maximum likelihood estimate of the probabilistic model, which is the shortest path to the most likely goal region. Predictions were performed after the target moved through 5 edges of the PRM to improve the estimation precision. The tracker moves on average 5 times

TABLE III
SIMULATION RESULTS WITH THE PROBABILISTIC PEDESTRIAN PREDICTION MODEL AND $r$-DISK WITH OCCLUSIONS OBSERVATION MODEL. THE PEDESTRIAN DATASET HAD 442 PEDESTRIAN TRAJECTORIES, AND THE OFFICE ENVIRONMENT HAD 100 RANDOM TRAJECTORIES.

| | Planning Scenario | | Results | | | Computation | |
|---|---|---|---|---|---|---|---|
| Environment | Predictions | Occlusions | $F/T$ full model | $F/T$ ground truth | $M$ | $\lvert\mathcal{V}\rvert$ | time (s) |
| Pedestrian dataset | Prob. | Yes | **56.11** (6.55) | **66.74** (12.84) | 2.54 (0.56) | 6,026 (1,183) | 3.0 (0.6) |
| | Prob. | No | 55.55 (7.09) | 65.86 (13.95) | 2.50 (0.55) | 6,155 (1,197) | 1.6 (0.4) |
| | Det. | Yes | 51.14 (9.07) | 63.03 (16.22) | 2.02 (0.20) | 1,971 (497) | 0.7 (0.1) |
| | Det. | No | 50.14 (9.51) | 62.36 (16.76) | 1.99 (0.11) | 1,994 (472) | 0.6 (0.1) |
| Office environment | Prob. | Yes | **35.94** (9.45) | **64.73** (23.77) | 3.52 (1.00) | 5,554 (2,847) | 3.2 (1.5) |
| | Prob. | No | 30.09 (9.08) | 56.86 (23.88) | 2.98 (0.87) | 7,233 (3,504) | 2.2 (1.3) |
| | Det. | Yes | 29.54 (9.06) | 50.47 (26.17) | 2.09 (0.42) | 834 (411) | 0.5 (0.1) |
| | Det. | No | 25.52 (8.23) | 45.85 (23.05) | 1.80 (0.43) | 1,240 (591) | 0.5 (0.1) |

Columns: Mean monitoring effectiveness as a percentage of mission duration for probabilistic predictions and $r$-disk with occlusions observation model; Mean monitoring effectiveness as a percentage of mission duration for ground truth trajectory and $r$-disk with occlusions observation model; Num. vertices $|\mathcal{V}|$; Computation time (s). Standard deviation in parenthesis. Best results with respect to full model and ground truth in bold.

faster than the target, and both agents avoid collisions with static obstacles. The tracker starts at the same position as the target while the end position is to be optimised by the planner.

We consider two environments and ground truth trajectories: a footpath environment with a real pedestrian trajectory dataset[2] with 442 trajectories [38] (Fig. 12) that was used for validation of the prediction algorithm [20], and a more cluttered office environment with 100 random trajectories drawn from the same dynamics model (Fig. 11).

*2) Trajectory prediction model implementation:* The prediction model for the target uses the intention inference trajectory prediction model proposed in [20]. Fig. 11 shows an example environment, trajectory and prediction. In this model, the target is assumed to be driven by the high-level intention to move to an unknown goal region within a cluttered environment. Prediction is based on the observed trajectory and a static environment map. The prediction algorithm first estimates the intended goal region using a recursive Bayes' approach, and then uses the resulting probability distribution to perform Monte Carlo sampling of random walks through a PRM. Each random walk biases towards shortest paths to the estimated goal regions. The random walks are interpolated using a stochastic speed to give a set of 100 particles representing the predicted position of the target at every future timestep.

*3) Observation model implementation:* The tracker observes the target only when within an observation range ($r$-disk model) and the line-of-sight is not occluded by the static obstacles in the environment. As a comparison, we also consider an $r$-disk model that ignores occlusions. For the simulations we let $r$ be 20 % of the width of each environment.

*4) Planner implementation:* For this case study, the tracker plans a path through a probabilistic roadmap that respects static obstacles in the environment. To achieve this, the spatial discretisation set $\mathcal{P}_1$ used during the vertex generation phase of the planner is defined as the same vertices in the PRM used by the prediction algorithm. The travel time between waypoints for the tracker is proportional to the shortest path through the euclidean-distance PRM plus a constant.

Since the PRM respects static obstacles, there are regions of the environment where the tracker cannot stop. This violates the condition CH $\subseteq \hat{\mathcal{Y}}$ (Condition 2) required for Lemma 2 and therefore the convex hull culling $\mathcal{P}_3$ cannot be used. This may increase runtime but does not affect optimality.

*5) Results and discussion:* The simulation results are shown in Tab. III averaged over the set of trajectories for each environment. A single-tailed paired $t$-test supports the hypothesis that in both environments, planning with the probabilistic model and the occlusions achieves a significantly higher monitoring effectiveness than the other planning scenarios. The hypothesis achieved statistical significance ($p < 0.001$ in all cases) when measured assuming either the full probabilistic trajectory prediction (1st results column) or the ground truth trajectories (2nd results column).

Planning while taking into account occlusions achieved a significant improvement in monitoring effectiveness compared to planning while ignoring the occlusions, and therefore shows the benefit of planning with a more accurate observation model. These improvements were more significant in the office environment compared to the footpath, since the office environment is more cluttered and therefore a larger portion of the in-range region was occluded from most positions.

In the office environment, the ground truth trajectories were samples generated from the probabilistic model, and therefore the monitoring effectiveness improvements between planning with the probabilistic to the deterministic model were similar when measured relative to the full model or to the ground truth. In the footpath environment, the ground truth trajectories were taken from the real pedestrian dataset, and the probabilistic planning still had a higher monitoring effectiveness than the deterministic planning. This suggests the intention inference model was a better estimator for the pedestrian trajectories than the shortest path model, and therefore improved the planning performance. Note that the ground truth monitoring effectiveness was much higher than the full model, since the former was measured up until the time when the sample trajectory reached the goal region, while the latter was measured until the longest sample reached its goal region.

The computation times for these simulations were faster than the AUV mission monitoring experiments due to the

[2] Dataset published at: graphics.cs.ucy.ac.cy/research/downloads/crowd-data

shorter planning horizon. Planning without occlusions was faster than planning with occlusions since the line-of-sight collision checking was time consuming. The deterministic planning was significantly faster than the probabilistic planning due to requiring fewer vertices, and therefore may be more desirable in practical situations where computation time is more important than monitoring effectiveness.

## IX. Discussion and Future Work

We have proposed a spatiotemporal optimal stopping formulation and a polynomial-time sweep-plane algorithm for the stochastic mission monitoring problem. The algorithm solves the problem with a reduction to a longest-path search through a directed acyclic graph. The graph construction phase further reduces the size of the search space by exploiting geometric characteristics of the problem under reasonable assumptions. The simulation results validate the performance of our algorithm, show the value of the probabilistic formulation and describe implementations of probabilistic models for realistic applications. The algorithm admits a general class of probabilistic trajectory prediction models and probabilistic observation models, and therefore is applicable to a variety of real-world problems, such as the demonstrated AUV monitoring and pedestrian monitoring applications. Our implementation is unoptimised, but still exhibits reasonable clock-time performance ranging from milliseconds to a few minutes. Our results show the algorithm is feasible for operational use as is, and motivate important extensions as avenues of future work.

In particular, we are interested in extending the approach to allow replanning for partially known mission trajectories that are discovered over time. For simple assumptions, replanning can be performed by repeating the algorithm when new information is obtained. For more complex scenarios, where the mission discovery depends on the tracker observations, the planner may need to consider the value of obtained information when planning sequences of stopping locations and times. In this case, the problem is likely to be intractable, and therefore extensions that give approximately optimal solutions should be developed.

Additionally, it would be interesting to further generalise the observation models. An orientation-dependent observation model, such as a narrow field-of-view camera, can be accommodated by adding a tracker-orientation dimension to the search space. Time-varying observation models and dynamic communication rates [58] can readily be addressed by redefining the observation value as a function of time. Also, other definitions for monitoring effectiveness could be considered, such as worst-case as opposed to expected observation time.

Larger scale operations can feature multiple tracker and target agents, such as when multiple AUVs survey an area while being supervised by multiple surface vessels. For multi-target scenarios, a naive formulation that sums all targets' observation times is tractable, but likely to give undesirable plans that only follow a single target. An objective function that favours dividing the observation time between the targets, such as a minimax function, would be more beneficial, but is likely to be intractable and approximations should be considered. Multi-tracker scenarios can be solved by planning in the joint action space, preferably in a decentralised manner that is robust to unreliable communication between tracker vehicles [59].

## Acknowledgment

The authors would like to thank Stuart Anstee for introducing us to this problem and for helpful discussion of AUV monitoring in practice.

## References

[1] S. B. Williams, O. R. Pizarro, M. V. Jakuba, C. R. Johnson, N. S. Barrett, R. C. Babcock, G. A. Kendrick, P. D. Steinberg, A. J. Heyward, P. J. Doherty, I. Mahon, M. Johnson-Roberson, D. Steinberg, and A. Friedman, "Monitoring of benthic reference sites: Using an autonomous underwater vehicle," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 73–84, 2012.

[2] S. B. Williams, O. Pizarro, and B. Foley, "Return to Antikythera: Multi-session SLAM based AUV mapping of a first century BC wreck site," in *Proc. of FSR*, 2015.

[3] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, 2012.

[4] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *J. Field Robot.*, vol. 26, no. 8, pp. 651–668, 2009.

[5] D. Ball, P. Ross, A. English, T. Patten, B. Upcroft, R. Fitch, S. Sukkarieh, G. Wyeth, and P. Corke, "Robotics for sustainable broad-acre agriculture," in *Proc. of FSR*, 2013.

[6] T. Peynot, S.-T. Lui, R. McAllister, R. Fitch, and S. Sukkarieh, "Learned stochastic mobility prediction for planning with control uncertainty on unstructured terrain," *J. Field Robot.*, vol. 31, no. 6, pp. 969–995, 2014.

[7] C. R. German, M. V. Jakuba, J. C. Kinsey, J. Partan, S. Suman, A. Belani, and D. R. Yoerger, "A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of autonomous surface vehicles and autonomous underwater vehicles," in *Proc. of IEEE/OES AUV*, 2012.

[8] P. E. Hagen, N. Størkersen, B.-E. Marthinsen, G. Sten, and K. Vestgård, "Rapid environmental assessment with autonomous underwater vehicles – examples from HUGIN operations," *J. Marine Syst.*, vol. 69, no. 1-2, pp. 137–145, 2008.

[9] N. K. Yilmaz, C. Evangelinos, P. F. J. Lermusiaux, and N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE J. Oceanic Eng.*, vol. 33, no. 4, pp. 522–537, 2008.

[10] O. Khatib, X. Yeh, G. Brantner, B. Soe, B. Kim, S. Ganguly, H. Stuart, S. Wang, M. Cutkosky, A. Edsinger, P. Mullins, M. Barham, C. R. Voolstra, K. N. Salama, M. L'Hour, and V. Creuze, "Ocean one: A robotic avatar for oceanic discovery," *IEEE Robot. Autom. Mag.*, vol. 23, no. 4, pp. 20–29, 2016.

[11] G. Best and P. Moghadam, "An evaluation of multi-modal user interface elements for tablet-based robot teleoperation," in *Proc. of ARAA ACRA*, 2014.

[12] M. F. Fallon, G. Papadopoulos, J. J. Leonard, and N. M. Patrikalakis, "Cooperative AUV navigation using a single maneuvering surface craft," *Int. J. Robot. Res.*, vol. 29, no. 12, pp. 1461–1474, 2010.

[13] G. Heppner, A. Roennau, and R. Dillman, "Enhancing sensor capabilities of walking robots through cooperative exploration with aerial robots," *J. Autom., Mob. Robot. and Intell. Syst.*, vol. 7, no. 2, pp. 5–11, 2013.

[14] L. Klodt, S. Khodaverdian, and V. Willert, "Motion control for UAV-UGV cooperation with visibility constraint," in *Proc. of IEEE CCA*, 2015, pp. 1379–1385.

[15] M. Saska, V. Vonásek, T. Krajnik, and L. Preucil, "Coordination and navigation of heterogeneous MAV-UGV formations localized by a 'hawk-eye'-like approach under a model predictive control scheme," *Int. J. Robot. Res.*, vol. 33, no. 10, pp. 1393–1412, 2014.

[16] N. Kottege and U. R. Zimmer, "Underwater acoustic localization for small submersibles," *J. Field Robot.*, vol. 28, no. 1, pp. 40–69, 2011.

[17] D. L. Bongiorno, M. Bryson, and S. B. Williams, "Dynamic spectral-based underwater colour correction," in *Proc. of IEEE OCEANS*, 2013.

[18] K. Karydis, I. Poulakakis, J. Sun, and H. G. Tanner, "Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty," *Int. J. Robot. Res.*, vol. 34, no. 10, pp. 1278–1295, 2015.

[19] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Aggressive moving obstacle avoidance using a stochastic reachable set based potential field," in *Proc. of WAFR*, 2014.

[20] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *Proc. of IEEE/RSJ IROS*, 2015, pp. 5817–5823.

[21] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Auton. Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[22] Y. S. Chow, H. Robbins, and D. Siegmund, *Great expectations: The theory of optimal stopping*. Houghton Mifflin Boston, 1971.

[23] G. Best and S. Anstee, "Motion planning for autonomous underwater vehicle supervision," in *Proc. of ARAA ACRA*, 2014.

[24] G. A. Hollinger, S. Yerramalli, S. Singh, U. Mitra, and G. Sukhatme, "Distributed coordination and data fusion for underwater search," in *Proc. of IEEE ICRA*, 2011, pp. 349–355.

[25] R. Brockers, P. Bouffard, J. Ma, L. Matthies, and C. Tomlin, "Autonomous landing and ingress of micro-air-vehicles in urban environments based on monocular vision," *Proc. of SPIE*, vol. 8031, pp. 803 111 1–12, 2011.

[26] T. Naseer, J. Sturm, and D. Cremers, "FollowMe: Person following and gesture recognition with a quadrocopter," in *Proc. of IEEE/RSJ IROS*, 2013, pp. 624–630.

[27] W. Hönig and N. Ayanian, "Dynamic multi-target coverage with robotic cameras," in *Proc. of IEEE/RSJ IROS*, 2016, pp. 1871–1878.

[28] M. Dunbabin and A. Tews, "Toward robotic visual and acoustic stealth for outdoor dynamic target tracking," in *Proc. of ARAA ACRA*, 2012.

[29] N. Mathew, S. L. Smith, and S. L. Waslander, "A graph-based approach to multi-robot rendezvous for recharging in persistent tasks," in *Proc. of IEEE ICRA*, 2013, pp. 3497–3502.

[30] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Proc. of Robotics: Science and Systems*, 2015.

[31] C. D'Este, B. Seton, J. McCulloch, D. Smith, and C. Sharman, "Avoiding marine vehicles with passive acoustics," *J. Field Robot.*, vol. 32, no. 1, pp. 152–166, 2015.

[32] J. Witt and M. Dunbabin, "Go with the flow: Optimal AUV path planning in coastal environments," in *Proc. of ARAA ACRA*, 2008.

[33] D. Rao and S. B. Williams, "Large-scale path planning for underwater gliders in ocean currents," in *Proc. of ARAA ACRA*, 2009.

[34] J. Das, K. Rajan, S. Frolov, F. Pyy, J. Ryany, D. Caron, and G. Sukhatme, "Towards marine bloom trajectory prediction for AUV mission planning," in *Proc. of IEEE ICRA*, 2010, pp. 4784–4790.

[35] C. Fang and S. Anstee, "Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle," in *Proc. of IEEE OCEANS*, 2010.

[36] J. Faigl and G. A. Hollinger, "Unifying multi-goal path planning for autonomous data collection," in *Proc. of IEEE/RSJ IROS*, 2014, pp. 2937–2942.

[37] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of robot control*. Springer, 2012.

[38] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[39] G. Best, W. Martens, and R. Fitch, "A spatiotemporal optimal stopping problem for mission monitoring with stationary viewpoints," in *Proc. of Robotics: Science and Systems*, 2015.

[40] M. Lindhé and K. Johansson, "Exploiting multipath fading with a mobile robot," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1363–1380, 2013.

[41] J. Das, F. Py, J. B. J. Harvey, J. P. Ryan, A. Gellene, R. Graham, D. A. Caron, K. Rajan, and G. S. Sukhatme, "Data-driven robotic sampling for marine ecosystem monitoring," *Int. J. Robot. Res.*, vol. 34, no. 12, pp. 1435–1452, 2015.

[42] J. Czyzowicz, L. Gąsieniec, K. Georgiou, E. Kranakis, and F. MacQuarrie, "The beachcombers' problem: Walking and searching with mobile robots," in *Proc. of SIROCCO*, 2014.

[43] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Springer, 1985.

[44] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, *Computational geometry*. Springer, 2000.

[45] J.-C. Latombe, *Robot Motion Planning*. Kluwer, 1991.

[46] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[47] E. L. Lawler, *Combinatorial optimization: Networks and matroids*. Courier Dover Publications, 1976.

[48] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.

[49] S. D. Bopardikar, S. L. Smith, and F. Bullo, "On dynamic vehicle routing with time constraints," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1524–1532, 2014.

[50] P. Toth and D. Vigo, Eds., *The vehicle routing problem*. Philadelphia: Society for Industrial and Applied Mathematics, 2001.

[51] E. Aaron, D. Krizanc, and E. Meyerson, "Multi-robot foremost coverage of time-varying graphs," in *Algorithms for Sensor Systems*. Springer, 2015, pp. 22–38.

[52] A. P. Aguiar, J. Almeida, M. Bayat, B. Cardeira, R. Cunha, A. Hausler, P. Maurya, A. Oliveira, A. Pascoal, A. Pereira, M. Rufino, L. Sebastiao, C. Silvestre, and F. Vanni, "Cooperative control of multiple marine vehicles: Theoretical challenges and practical issues," in *Proc. of IFAC MCMC*, 2009, pp. 412–417.

[53] F. Lobo Pereira, J. Sousa, R. Gomes, and P. Calado, "A model predictive control approach to AUVs motion coordination," in *Coordination Control of Distributed Systems*. Springer, 2015, pp. 9–18.

[54] J. Almeida, C. Silvestre, and A. Pascoal, "Cooperative control of multiple surface vessels with discrete-time periodic communications," *Int. J. Robust Nonlin.*, vol. 22, no. 4, pp. 398–419, 2012.

[55] M. Bibuli, M. Caccia, L. Lapierre, and G. Bruzzone, "Guidance of unmanned surface vehicles: Experiments in vehicle following," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 92–102, 2012.

[56] P. Calado and J. Sousa, "Leader-follower control of underwater vehicles over acoustic communications," in *Proc. of IEEE OCEANS*, 2011.

[57] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes*, 2nd ed. Springer, 2003.

[58] A. Kassir, R. Fitch, and S. Sukkarieh, "Communication-aware information gathering with dynamic information flow," *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 173–200, 2015.

[59] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Decentralised Monte Carlo tree search for active perception," in *Proc. of WAFR*, 2016.

**Graeme Best** completed his B.E. in Electrical and Computer Systems Engineering and B.Sc. in Computer Science at Monash University, Melbourne, Australia, in 2014. He is currently a Ph.D. candidate at the Australian Centre for Field Robotics (ACFR) at the University of Sydney. His research interests include planning algorithms for multi-robot teams performing coordinated perception tasks.

**Wolfram Martens** received his Ph.D. in Mechanical Engineering in 2013 at Technische Universität Berlin, Germany. He is currently a research fellow at the Australian Centre for Field Robotics (ACFR) at the University of Sydney, Australia. His research interests include active perception and probabilistic methods.

**Robert Fitch** received the Ph.D. degree in computer science from Dartmouth College, Hanover, NH, in 2005. He is currently an Associate Professor in the Centre for Autonomous Systems (CAS) at the University of Technology Sydney, Australia. Previously, he was a senior research fellow at the Australian Centre for Field Robotics (ACFR) at the University of Sydney. His research interests include systems of outdoor robots and their application to key problems in agriculture and environmental monitoring.