

A Spatiotemporal Optimal Stopping Problem for Mission Monitoring with Stationary Viewpoints

Graeme Best, Wolfram Martens and Robert Fitch
Australian Centre for Field Robotics (ACFR)
The University of Sydney
New South Wales, Australia
{g.best, w.martens, rfitch}@acfr.usyd.edu.au

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 pre-planned trajectory. This problem arises when autonomous underwater vehicles are monitored by surface vessels, and in a diverse range of other scenarios. The key problem characteristics we consider are that the monitor must remain stationary while observing the robot, and that the robot motion is modelled in general as a stochastic process. We propose a resolution-complete algorithm for this problem 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. A variety of stochastic models may be used to represent the expected robot trajectory. We present results drawn from real AUV trajectories and Monte Carlo simulations that validate the correctness of our algorithm and its feasibility in practice.

I. INTRODUCTION

Mission monitoring is a supervisory problem where a robot or a manually driven vehicle tracks the progress of an autonomous mobile robot in performing a pre-planned task. There are many examples of such tasks, including undersea surveys [38, 21], monitoring our natural environment [18], autonomous farming [5] and planetary exploration [33]. Monitoring allows for rapid response to failures and to important information that the robot may discover during the progress of its mission [22, 23, 7]. Additionally, the monitoring vehicle may augment mission capabilities by providing observations from external viewpoints, such as for accurate localisation and navigation [24, 34, 4, 27]. 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.

Classical *optimal stopping* problems [12], 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 in general is stochastic.

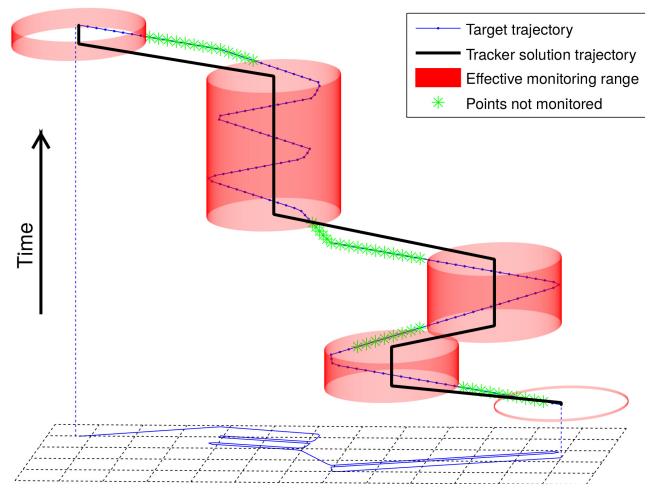


Fig. 1. Geometric interpretation of the spatiotemporal optimal stopping problem. A deterministic robot trajectory is shown in blue and also projected onto a plane in the two spatial dimensions. An example monitor trajectory solution is overlaid. Cylinders represent effective monitoring range at stopping locations. Green stars represent parts of the mission that are not monitored.

This work is motivated directly by 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 some operators must stop and deploy this communication equipment, with engines powered down, for maximum efficiency [6]. An optimal stopping solution maximises the time spent observing, and minimises time spent in stopping and starting the surface vessel which may incur a time penalty. This problem is not limited to the AUV monitoring context; other motivating examples include flying robots that must land during observational periods to conserve energy [10], acoustically-covert surveillance for tracking animals [36, 19], ground-based mobile recharge stations for aerial vehicles [32], aerial robots that must be stationary to achieve accurate measurements of radio-tagged wildlife [13], and un-

derwater robots that need to stop and surface to communicate or observe some phenomenon [16].

A related spatiotemporal optimal stopping problem for AUV monitoring has recently been studied [6]. However, AUV motion is limited to the deterministic case, and the solutions lack completeness guarantees. The contribution of this work is an efficient resolution-complete algorithm for this problem that in general models AUV motion as a random process. The stochastic model can be in any form that yields a spatial distribution at a given time point.

Our algorithm generates an optimal nonoverlapping set of “cylinders” 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 to maximise the expected overlap time between the cylinders 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 uses a sweep-plane approach to compute a resolution-complete solution in polynomial time.

In addition to analytical evaluation, we provide simulation examples drawn from actual AUV trajectories that illustrate the behaviour of the algorithms. We also validate our results by comparing expected observation time with actual observation time from Monte Carlo experiments, where the trajectory of the robot is drawn from an assumed stochastic process. The clock-time performance of our implementation shows that the solution is viable for practical use in mission monitoring.

II. RELATED WORK

A closely related problem is studied by Best and Anstee [6], who propose a greedy planner and a genetic algorithm for AUV mission monitoring where the AUV trajectory is 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 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 [12]. Recently, Lindhé and Johansson [30] study 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 agent must also choose a stopping location, 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.

Sweep-plane algorithms are often used for computational geometry problems such as Voronoi decomposition, intersections between line segments and unions of rectangles [15]. 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 [28]. 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 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 [29, 14]. Bopardikar *et al.* [9] employ this approach for dynamic vehicle routing, where an agent maximises the number of space-time demands visited. Dono [17] studies search graph culling using the convex hull of reachable points. Various models for modelling stochastic motion of a vehicle have been proposed, such as [25, 3]. 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.

In marine robotics, coordinated-control problems have received much attention due to the benefits realised by multi-robot systems [1]. Related problems include formation control and communication connectivity maintenance [31, 20, 35, 2], and target following [8, 11], 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.

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 [37]. 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.

III. PROBLEM FORMULATION

The problem involves two mobile agents: 1) a *target* which follows a probabilistic trajectory defined by a mission plan, and 2) a *tracker* that seeks to effectively monitor the target throughout the mission. To monitor effectively, the tracker must be within 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, and the tracker’s trajectory is to be optimised. In this section, we formally define the characteristics of the target and tracker trajectories, and the idea of effective monitoring as an optimisation objective.

A. Target Trajectory (Independent)

The trajectory of the target is described as its position as a function of time $x(t) : [0, T] \rightarrow \mathcal{X}$, where T is the mission duration and \mathcal{X} is the space of all possible target locations. We assume that the trajectory is not known precisely ahead of time, and therefore the predicted location of the target at time t_i is represented as a random variable X_i with some known distribution $X_i \sim D_i$. The distribution D_i has probability density function $\rho_i(x)$. A distribution is defined for every discrete time step $t_i := (i-1)\Delta_t \in \mathcal{T}$ over the duration of the mission, and therefore the predicted trajectory of the target is given by the sequence of random variables $X := (X_1, X_2, \dots, X_N)$.

This paper addresses two categories for the probability distributions D_i . For the general case, D_i may be any given probability distribution. We also make further refinements to the algorithm for the special case where the target trajectory is predicted precisely and therefore considered deterministic; i.e., each $\rho_i(x)$ is defined as a Dirac delta function.

B. Tracker Trajectory (Dependent)

The trajectory of the tracker is also described as its position as a function of time $y(t) : [0, T] \rightarrow \mathcal{Y}$, 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 $s(t) : [0, T] \rightarrow \mathcal{S}$. When in the STOPPED state, the tracker stops and remains stationary at a waypoint position $\hat{y}_i \in \hat{\mathcal{Y}} \subseteq \mathcal{Y}$. Over the course of a mission, the sequence of M waypoints is denoted $\hat{Y} := (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_M)$.

1) *Arrival and Departure Times:* The tracker remains STOPPED at each waypoint during the time interval of the waypoint arrival and departure times $[t_i^a, t_i^d]$, therefore

$$y(\tau) = \hat{y}_i, \forall \tau \in [t_i^a, t_i^d], \forall i \in \{1, 2, \dots, M\}.$$

The tracker is MOVING between consecutive waypoints $(\hat{y}_i, \hat{y}_{i+1})$ over the time interval $[t_i^d, t_{i+1}^a]$. Therefore, the state as a function of time is defined as

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

The sequences of associated arrival and departure times are denoted $T^a := (t_1^a, t_2^a, \dots, t_M^a)$ and $T^d := (t_1^d, t_2^d, \dots, t_M^d)$ respectively, and satisfy the constraints $t_i^a < t_i^d < t_{i+1}^a, \forall i$.

For convenience, we denote the trajectory of the tracker as a tuple $U = [\hat{Y}, T^a, T^d]$, which is described by a sequence of waypoint locations \hat{Y} , and associated sequences of arrival times T^a and departure times T^d .

2) *Travel time:* The required travel time $t_j^a - t_i^d$ between two waypoints is known and defined by some function $\delta(\hat{y}_i, \hat{y}_j) : \hat{\mathcal{Y}} \times \hat{\mathcal{Y}} \rightarrow \mathbb{R}_{\geq 0}$. The proposed algorithm is not dependent on the exact trajectory taken to achieve this travel time between waypoints. We require $\delta(\hat{y}_i, \hat{y}_j) = 0$ iff $\hat{y}_i = \hat{y}_j$, since \hat{y}_i and \hat{y}_j would effectively become a single waypoint.

3) *Start and End Conditions:* Practically, tracking vehicles are often also used for deployment/recovery of the target. Hence, without loss of generality, we assume the constraints:

$$\begin{cases} \hat{y}_1 = \mathbb{E}[X_1], & t_1^a = 0, \\ \hat{y}_M = \mathbb{E}[X_N], & t_M^a \leq T. \end{cases} \quad (1)$$

This assumption is not limiting. We require only that \hat{y}_1 and \hat{y}_M are known.

4) *Discretisation:* The position function $y(t)$ and state function $s(t)$ are sampled at discrete time steps $t_i \in \mathcal{T}$, resulting in the sequences of positions $Y = (y_1, y_2, \dots, y_N)$ and states $S = (s_1, s_2, \dots, s_N)$.

C. Effective Monitoring

The goal of the tracker is to effectively monitor the target. At time t_i , the monitoring effectiveness is described by a function $f(X_i, y_i, s_i) : \mathcal{X} \times \mathcal{Y} \times \mathcal{S} \rightarrow \{0, 1\}$, with 1 meaning effectively monitoring and 0 otherwise, defined as

$$f(X_i, y_i, s_i) := \begin{cases} \tilde{f}(\|X_i - y_i\|) & \text{if } s_i = \text{STOPPED} \\ 0 & \text{if } s_i = \text{MOVING}, \end{cases} \quad (2)$$

where $\tilde{f}(r_i) : \mathbb{R}_{\geq 0} \rightarrow \{0, 1\}$ is the monitoring effectiveness while STOPPED, defined as the r -disk model

$$\tilde{f}(r_i) := \begin{cases} 1 & \text{if } r_i \leq r \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The monitoring range parameter is denoted r . Other definitions for $\tilde{f}(r_i)$ may be used, such as a probabilistic communication model, but we focus on the binary case here for clarity.

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

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

which can be interpreted as the expected total amount of time that the tracker is STOPPED and in range of the target. $F(X, Y, S)$ can be evaluated using the expected values

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

Remark 1. For the special case where X_i is deterministic, $\mathbb{E} [f(X_i, y_i, s_i)] = f(X_i, y_i, s_i)$ and evaluates to 0 or 1 only.

For convenience, we also introduce notation for the monitoring effectiveness over a subset T of the mission as

$$F_{\mathsf{T}} := \Delta_t \sum_{\eta \in \mathcal{N}} \mathbb{E} [f(X_{\eta}, y_{\eta}, s_{\eta})], \quad \mathcal{N} = \{\eta : t_{\eta} \in \mathsf{T} \cap \mathcal{T}\}.$$

We also introduce \tilde{F}_{T} with the same definition but with the assumption that $s_{\eta} = \text{STOPPED}, \forall t_{\eta} \in \mathsf{T}$.

D. Problem Statement

Fig. 1, shown earlier, illustrates a geometric representation of the target and tracker moving through time and the monitoring effectiveness for a deterministic target problem specification. Using this visual model, the optimisation problem to be solved can be stated as follows.

For a given target trajectory X (blue line), select the positions \hat{Y} (coordinates of red cylinders in horizontal plane), arrival times T^a (bottoms of cylinders) and departure times T^d (tops) of a set of stopping waypoints U (red cylinders of radius r and vertical gaps $\delta(\hat{y}_i, \hat{y}_j)$), such that the sum of the expected monitoring effectiveness $F(X, Y, S)$ is maximised over the mission duration (number of green stars is minimised).

IV. GRAPH GENERATION

The proposed algorithm is divided into a graph generation phase and then a longest path graph search which utilises a sweep-plane, as outlined in Alg. 1. This section focuses on the process of generating the vertices and edges of the search graph, as summarised in Alg. 2. The result is a graph with vertices \mathcal{V} and edges \mathcal{E} , with paths through this graph describing solution trajectories for the tracker.

A. Vertices

A set of graph vertices is generated, with each vertex 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.

1) *Space*: The set of discrete space locations $\mathcal{P} \subseteq \hat{Y}$ is selected as the intersection $\hat{Y} \cap \mathcal{P}_1 \cap \mathcal{P}_2 \cap \mathcal{P}_3$, with each set described in the following points.

- i) \mathcal{P}_1 is the set of all points p_i that are within monitoring range of part of the target's trajectory, i.e., $\exists X_\eta \in X : \mathbb{E} \left[\tilde{f}(\|X_\eta - p_i\|) \right] > 0$.

Algorithm 1 Overview of trajectory planner for tracker

```

1: function MAIN( $X, r$ )
2:    $[\mathcal{V}, \mathcal{E}] \leftarrow \text{GENERATEGRAPH}(X, r)$ 
3:    $[\mathcal{V}, \mathcal{E}, v_{\text{start}}, v_{\text{end}}] \leftarrow \text{STARTENDCOND}'S(\mathcal{V}, \mathcal{E}, X)$ 
4:    $[\{\Omega\}, \{\psi\}] \leftarrow \text{SWEEPPLANE}(\mathcal{V}, \mathcal{E})$ 
5:    $[U, F] \leftarrow \text{BACKTRACKING}(\{\Omega\}, \{\psi\}, v_{\text{END}}, \mathcal{V}, \mathcal{E})$ 
6:   return  $[U = [\hat{Y}, T^a, T^d], F]$ 

```

Algorithm 2 Generating a graph of potential waypoint positions and times

```

1: function GENERATEGRAPH( $X, r$ )
2:   Select potential stopping locations  $p_i \in \mathcal{P}$ 
3:   Generate vertices  $v_\eta = [p_\eta, \tau_\eta^a, \tau_\eta^d] \in \mathcal{V}$ 
4:   Find feasible edges  $e_k = \langle v_i, v_j \rangle \in \mathcal{E}$ 
5:   Calculate edge times  $t_i^d, t_j^a$  for each edge  $e_k$ 
6:   Calculate edge weight  $\omega_{i,j}$  for each edge  $e_k$ 
7:   return  $[\mathcal{V}, \mathcal{E}]$  ▷ Vertices, Edges

```

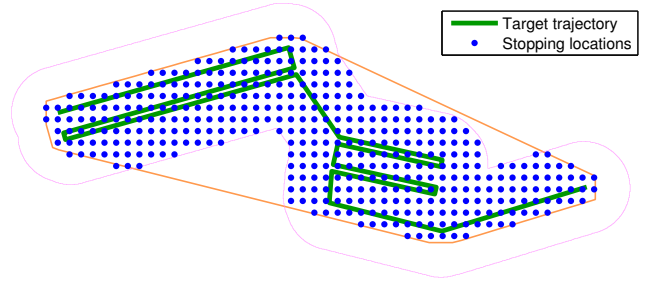


Fig. 2. Possible stopping locations around a deterministic target trajectory moving through 2 spatial dimensions. Also shown are the boundaries of the convex hull (orange) and monitoring region (pink) described in Sec. IV-A1.

- ii) \mathcal{P}_2 is the set of all points x that are within the space bounded by the convex hull of all possible locations ($\{x : \exists i, \rho_i(x) > 0\}$) visited by the target.
- iii) \mathcal{P}_3 is a discrete set of possible stopping points. All examples use a uniform grid.

An example of the resulting set of stopping locations is shown in Fig. 2. In practical circumstances, these set definitions are reasonable. Specific conditions are defined in Lemmas 1 (\mathcal{P}_1) and 2 (\mathcal{P}_2) below. If a condition is not valid for a specific problem then that set may be omitted to guarantee optimality, potentially at the cost of higher computation time.

Remark 2. If the distributions $\rho_i(x)$ are unbounded, then \mathcal{P} is potentially an infinite set. However, the reachability pruning (see later in Sec. IV-C) ensures the search space is finite. For computational reasons, to further reduce the size of the graph it may be necessary to approximate \mathcal{P}_1 and \mathcal{P}_2 using non-zero lower bounds, i.e., $\rho_i(x) > LB$.

Remark 3. Due to the \mathcal{P}_3 discretisation, the space $\text{CH}(X)$ for \mathcal{P}_2 should be expanded in all directions by a distance the size of the discretisation spacings, to avoid excluding potentially optimal waypoints near the boundary.

Lemma 1. Stopping in effective monitoring region:

An optimal solution trajectory U only contains waypoints at locations $\hat{y}_i \in \hat{Y}$ which satisfy $\exists X_\eta \in X : \mathbb{E} \left[\tilde{f}(\|X_\eta - \hat{y}_i\|) \right] > 0$. The proof requires triangle inequality to hold for $\delta(\hat{y}_i, \hat{y}_j)$, i.e., $\delta(\hat{y}_a, \hat{y}_b) \leq \delta(\hat{y}_a, \hat{y}_i) + \delta(\hat{y}_i, \hat{y}_b)$.

Proof: Define two partial solution trajectories over the time interval $[t_i^a, t_k^d]$: i) $\hat{Y} = (\hat{y}_i, \hat{y}_j, \hat{y}_k)$ and ii) $\hat{Y}^* = (\hat{y}_i, \hat{y}_k)$; with \hat{y}_j satisfying $\nexists X_\eta \in X : \mathbb{E} \left[\tilde{f}(\|X_\eta - \hat{y}_j\|) \right] > 0$. For \hat{Y} , the monitoring effectiveness F while STOPPED at \hat{y}_i is $F_{[t_j^a, t_j^d]} = 0$, due to the condition imposed on \hat{y}_j . Combined with the MOVING times, $F_{[t_i^a, t_k^d]} = 0$, with interval length $L = t_k^a - t_i^d$. Therefore $F(\hat{Y}) = \tilde{F}_{[t_i^a, t_i^d] \cup [t_k^a, t_k^d]}$. For \hat{Y}^* , the monitoring effectiveness while MOVING is $F_{[t_i^a, t_k^d]} = 0$, with interval length $L^* = t_k^a - t_i^d$. Therefore $F(\hat{Y}^*) = \tilde{F}_{[t_i^a, t_i^d] \cup [t_k^a, t_k^d]}$.

It follows from the triangle inequality assumption that $L \geq L^*$. Therefore $\exists \{t_i^{d'}, t_k^{a'}\} : (t_i^{d'} \geq t_i^d) \wedge (t_k^{a'} \leq t_k^a)$, where $\{t_i^{d'}, t_k^{a'}\} \leftarrow \{t_i^{d'}, t_k^{a'}\}$ is a feasible choice for departure and

arrival times. The optimal choice for $\{t_i^d, t_k^a\}$ will always result in a greater or equal monitoring effectiveness than if $\{t_i^d, t_k^a\} \leftarrow \{t_i^d, t_k^a\}$ were chosen, therefore:

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

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_{η} . \square

Lemma 2. Stopping in convex hull:

An optimal solution trajectory U only contains waypoints at locations $\hat{y}_i \in \hat{Y}$ which are in $\text{CH}(X)$, where $\text{CH}(X)$ is the space bounded by the convex hull of the set comprising all possible target positions X and known tracker positions \hat{y}_1, \hat{y}_M . The proof requires $\text{CH}(X) \subseteq \hat{Y}$, and that the travel time monotonically increases with distance from a fixed start or end position; i.e., $\delta(\hat{y}_i, \hat{y}_a) \geq \delta(\hat{y}_i, \hat{y}_b)$ and $\delta(\hat{y}_a, \hat{y}_i) \geq \delta(\hat{y}_b, \hat{y}_i)$, $\forall(\hat{y}_i, \hat{y}_a, \hat{y}_b) : \|\hat{y}_a - \hat{y}_i\| \geq \|\hat{y}_b - \hat{y}_i\|$.

Proof: Define a stopping position $\hat{y}_a \notin \text{CH}(X)$. By definition, there exists a half-plane \mathcal{H} such that $\text{CH}(X) \subset \mathcal{H}$, $\hat{y}_a \notin \mathcal{H}$, and \hat{y}_a^* lies on the boundary of \mathcal{H} where \hat{y}_a^* is the closest point to \hat{y}_a in $\text{CH}(X)$. 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., $\|\hat{y}_a^* - h\| < \|\hat{y}_a - h\|, \forall h \in \mathcal{H}$. Therefore, since $X_i \in \mathcal{H}$, we have $\tilde{f}(\|X_i - \hat{y}_a^*\|) \geq \tilde{f}(\|X_i - \hat{y}_a\|), \forall X_i \in X$. 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 $\hat{y}_a^* \in \hat{Y}$, which will hold if $\text{CH}(X) \subseteq \hat{Y}$.

To be optimal, selecting \hat{y}_a^* instead of \hat{y}_a must not negatively affect \tilde{F} 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 $\hat{y}_i, \hat{y}_j \in \text{CH}(X) \subset \mathcal{H}$. It follows from the monotonic assumption that the travel times must not increase by selecting \hat{y}_a^* , i.e., $\delta(\hat{y}_i, \hat{y}_a^*) \leq \delta(\hat{y}_i, \hat{y}_a)$ and $\delta(\hat{y}_a^*, \hat{y}_j) \leq \delta(\hat{y}_a, \hat{y}_j)$. 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 $\hat{y}_1, \hat{y}_M \in \text{CH}(X)$, this generalises to longer sequences. Therefore an optimal solution trajectory has all $\hat{y}_i \in \text{CH}(X)$. \square

2) *Incorporating Time:* A vertex $v_{\eta} \in \mathcal{V}$ represents the tuple $v_{\eta} := [p_{\eta}, \tau_{\eta}^a, \tau_{\eta}^d]$, where $p_{\eta} \in \mathcal{P}$ and $\tau_{\eta}^a, \tau_{\eta}^d$ are described as follows for the general and deterministic target cases.

In the general case, a new vertex is created for every time step that the target is possibly in range of the tracker if the tracker was STOPPED at position p_{η} , i.e., $\mathcal{V} := \left\{ v_{\eta} = [p_{\eta}, \tau_{\eta}^a, \tau_{\eta}^d \leftarrow \tau_{\eta}^a + \Delta_t] : \mathbb{E} \left[\tilde{f}(\|X(\tau_{\eta}^a) - p_{\eta}\|) \right] > 0 \right\}$. This definition is referred to as the generalised algorithm.

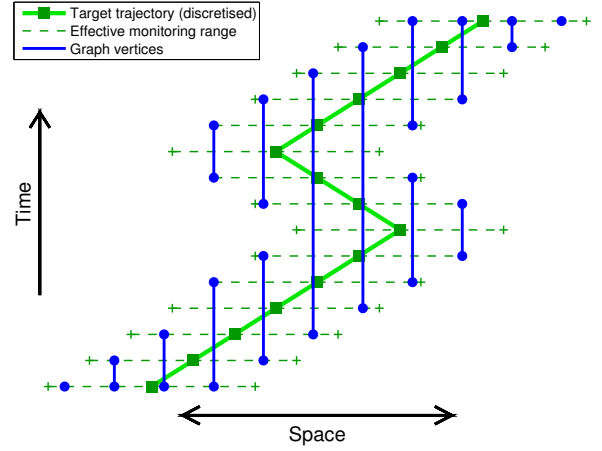


Fig. 3. Vertices overlaying an example deterministic target trajectory. Each vertical blue line segment represents a vertex in the search graph. Each vertex maps to a potential STOPPED position in the tracker trajectory, with the arrival and departure times determined by the edges (Fig. 4).

For the special case of a deterministic target trajectory, 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, $T_i \subseteq \mathcal{T}$ denotes the set of all times such that the target would be effectively monitored if the target were STOPPED at p_i at time t_i , i.e., $T_i := \{t_i \in \mathcal{T} : \tilde{f}(\|X_{t_i} - p_i\|) = 1\}$. Each T_i is then divided into multiple subsequences, with each subsequence being a complete run of consecutive timesteps $(t_j, t_{j+1}, \dots, t_{j+k}) \subseteq T_i$. Each subsequence forms a new vertex v_{η} in the search graph, such that $[p_{\eta}, \tau_{\eta}^a, \tau_{\eta}^d] \leftarrow [p_i, t_j, t_{j+k} + \Delta_t]$. This is illustrated in Fig. 3: each vertical blue line segment is a vertex with the bottom at time τ_{η}^a and the top at τ_{η}^d . This definition is referred to as the deterministic algorithm. Justification for the adjustments made in this deterministic case is provided later in Lemma 3.

B. Edges

A solution trajectory is represented by a path through the graph with consecutive vertices connected by directed edges. 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. The set of all edges included in the search graph is denoted $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$.

Each edge has an associated departure time $t_{e_{\eta}}^d := t_i^d$ and arrival time $t_{e_{\eta}}^a := t_j^a$ which describes the exact time the tracker moves from p_i to p_j . We require t_{η}^a, t_{η}^d satisfy

$$t_{\eta}^a = \tau_{\eta}^a < t_{\eta}^d. \quad (4)$$

For the general case, selecting $t_{\eta}^a \leftarrow \tau_{\eta}^a$ is optimal relative to the temporal resolution since each vertex represents only stopping for a single time step at p_{η} . 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). The key advantage of having a fixed arrival time (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

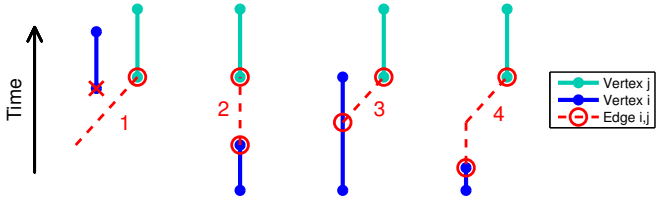


Fig. 4. The four possible edge categories described in Alg. 3 and Sec. IV-B.

Algorithm 3 Edge weight and moving time calculations for an edge. The four categories are illustrated in Fig. 4.

```

1: function EDGE_CALCULATION( $e_\eta = \langle v_i, v_j \rangle$ )
2:    $\rho \leftarrow \mathbb{E} \left[ \tilde{f}(\|X_\eta - p_i\|) \right] : t_\eta = \tau_i^a$ 
3:    $t_j^a \leftarrow \tau_j^a$ 
4:   if  $\delta(p_i, p_j) \geq \tau_j^a - \tau_i^a$  then ▷ Cat. 1
5:     Do not include  $e_\eta$  in  $\mathcal{E}$ 
6:   else if  $p_i = p_j$  then ▷ Cat. 2
7:      $t_i^d \leftarrow \tau_j^a$ 
8:      $\omega_{i,j} \leftarrow \rho \times (\tau_i^d - \tau_i^a)$ 
9:   else if  $\delta(p_i, p_j) \geq \tau_j^a - \tau_i^d$  then ▷ Cat. 3
10:     $t_i^d \leftarrow \tau_j^a - \delta(p_i, p_j)$ 
11:     $\omega_{i,j} \leftarrow \rho \times (\tau_j^a - \tau_i^a - \delta(p_i, p_j))$ 
12:   else ▷ Cat. 4
13:     $t_i^d \leftarrow \tau_j^a - \delta(p_i, p_j)$ 
14:     $\omega_{i,j} \leftarrow \rho \times (\tau_i^d - \tau_i^a)$ 
15:   return  $[t_i^d, t_j^a, \omega_{i,j}]$  ▷ Depart, Arrive, Weight

```

edge $e_m = \langle v_h, v_i \rangle$ or the path taken to or from an edge; and therefore optimal sub-paths are additive.

Each edge also has an associated weight $\omega_{i,j}$ which is defined as the amount of time spent effectively monitoring over the time interval $[\tau_i^a, \tau_j^a]$, i.e.,

$$\omega_{i,j} := F_{[\tau_i^a, \tau_j^a]}. \quad (5)$$

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., $\delta(p_i, p_j) \leq \tau_j^a - \tau_i^a$.
- 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 STOPPED while not in range.
- 4) **Larger Gap** – The gap is larger than $\delta(p_i, p_j)$; therefore there must be some time STOPPED while not in range.

Remark 4. The edge weight calculations do not strictly comply with (5). For an edge $\langle v_i, v_j \rangle$ where there exists another vertex v_a at the same position as v_i with a later time or at the same position as v_j with an earlier time; then $\omega_{i,j}$ may underestimate $F_{[\tau_i^a, \tau_j^a]}$. 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 $\omega_{i,j}$ is not a problem 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 case:

If a path passes through v_η , then it is optimal for the solution trajectory to arrive at p_η with t_η^a chosen as τ_η^a . This applies when the target trajectory is deterministic, and also assumes that the average speed of the tracker between waypoints is not less than the maximum instantaneous speed of the target, i.e., $\frac{\delta(\hat{y}_i, \hat{y}_j)}{\|\hat{y}_j - \hat{y}_i\|} \geq \|\dot{x}(t)\|_{max}, \forall \hat{y}_i, \hat{y}_j$.

Proof: Consider the path consisting of a single edge $\langle v_i, v_j \rangle$ for three cases: i) choose $t_j^a \leftarrow t_{j_1}^a$ where $\tau_j^a < t_{j_1}^a \leq \tau_j^d$; ii) choose $t_j^a \leftarrow t_{j_2}^a$ where $t_{j_2}^a = \tau_j^a$; and iii) choose $t_j^a \leftarrow t_{j_3}^a$ where $t_{j_3}^a < \tau_j^a$. The following proof shows that ii) has a monitoring effectiveness greater than or equal to i) and iii).

Firstly, consider the start times of a pair of vertices (τ_i^a, τ_j^a) . When the target moves in a straight line at a constant speed $\|\dot{x}\|$ (i.e., gradient in Fig. 3), the vertices will have start times with this same gradient between pairs, i.e., $|\tau_j^a - \tau_i^a| = \|p_j - p_i\| \cdot \|\dot{x}(t)\|$. If the target turns (e.g. upper half of Fig. 3), or moves slower, this time difference must always be larger; and therefore generally $|\tau_j^a - \tau_i^a| \geq \|p_j - p_i\| \cdot \|\dot{x}(t)\|$. Applying the speed assumption,

$$|\tau_j^a - \tau_i^a| \geq |t_j^a - \tau_i^d| = \delta(p_i, p_j). \quad (6)$$

An exception could occur at the beginning of the mission; however the adjustments in Sec. IV-C make this impossible.

From (6), it follows that if $t_j^a \leftarrow t_{j_2}^a$ then $t_i^d \geq \tau_i^d$. For ii), the tracker departs p_i at a time $\partial := t_{j_1}^a - t_{j_2}^a$ earlier than for i). Therefore ii) will spend ∂ less time at p_i and ∂ more time at p_j than i). The extra time spent at p_j is the interval $[t_{k_2}^a, t_{k_1}^a]$, and by definition of a vertex for the deterministic case (see Remark 1) $F_{[t_{k_2}^a, t_{k_1}^a]} = \partial$, which is maximal. Case i) can not improve on this during the extra time at p_j , and therefore ii) has a greater or equal monitoring effectiveness than i). The above assumes $\tau_i^a < \tau_j^a$; however, it follows from (6) that an optimal path will not contain $\langle v_i, v_j \rangle$ if $\tau_i^a > \tau_j^a$.

To achieve iii), the tracker will spend more time at v_j than for ii). This time is before τ_j^a , and therefore $F_{[t_{j_3}^a, \tau_j^a]} = 0$, which is minimal; hence ii) has a greater or equal monitoring effectiveness than iii). This shows that $t_j^a \leftarrow \tau_j^a$ is optimal. \square

C. Start and End Conditions

The second phase (Sec. V) backtracks through the graph until a start vertex v_{start} is found. To allow for this, a new position $p_{start} = \hat{y}_1$ is added to \mathcal{P} , if it is not already in the set, and vertices and edges are added as before. The start vertex v_{start} is selected as the earliest v_i at \hat{y}_1 . To ensure that backtracking always finds a path back to v_{start} , all other vertices are adjusted using the rules:

$$\begin{cases} \text{do nothing} & \text{if } \tau_i^a \geq \delta(p_{start}, p_i) \\ \text{remove } v_i & \text{else if } \tau_i^d \leq \delta(p_{start}, p_i) \\ \tau_i^a = \delta(p_{start}, p_i) & \text{otherwise.} \end{cases} \quad (7)$$

In (7), case two removes all v_i that are not reachable from v_{start} . For the Lemma 3 result to hold and therefore the t_i^a

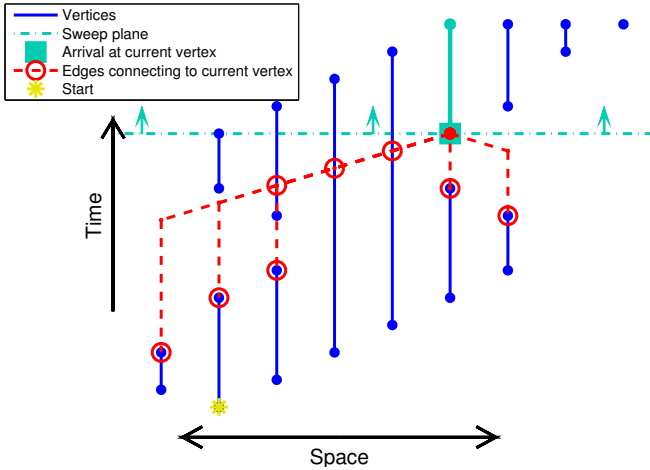


Fig. 5. Sweep plane at a particular time instant showing all feasible edges into the current vertex, analogous to Fig. 3 but including the start condition adjustment. Travel time assumes constant velocity motion.

Algorithm 4 Sweep-plane graph search: forward pass

```

1: function SWEEPPLANE( $\mathcal{V}$ ,  $\mathcal{E}$ )
2:    $\Omega_{\text{start}} \leftarrow 0$ 
3:   for  $t = t_1, t_2, \dots, t_N$  do
4:     for each  $v_i \in \mathcal{V} : \tau_i^a = t$  do
5:        $\mathcal{E}_i \leftarrow \{e_\epsilon : e_\epsilon = \langle v_\epsilon, v_i \rangle \in \mathcal{E}\} \triangleright$  Edges into  $v_i$ 
6:        $\psi_i \leftarrow \operatorname{argmax}_{e_\epsilon \in \mathcal{E}_i} [\Omega_\epsilon + \omega_{e_\epsilon, i}]$ 
7:        $\Omega_i \leftarrow \Omega_{\psi_i} + \omega_{\psi_i, i}$ 
8:   return  $[\{\Omega\}, \{\psi\}] \triangleright$  Path weights, Back-pointers
  
```

selection (4) to be optimal, the third case trims all v_i that are reachable only at some time after τ_i^a .

For the end condition, new vertices and edges are added at position $p_{\text{end}} = \hat{y}_M$. The latest v_i at \hat{y}_M is denoted v_{end} .

V. SWEEP PLANE ALGORITHM

In this section, we propose a longest-path graph search algorithm for finding the optimal tracker trajectory. The graph can be efficiently searched since it is a directed acyclic graph and therefore a topological ordering of \mathcal{V} exists.

1) *Forward Pass*: A topological ordering can be found by visiting v_i in order of ascending time $t = \tau_i^a$. This can be thought of as a sweeping plane as illustrated in Fig. 5 and described in Alg. 4. 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 $t = \tau_i^a$ (line 4). For efficient evaluation of the vertex set in line 4, \mathcal{V} should be pre-sorted by ascending τ_i^a . When v_i is explored (line 5), all edges e_ϵ leading in to v_i are compared (line 6) and the optimal previous vertex with an edge into each v_i is denoted ψ_i . The sum of weights along the optimal path leading to vertex v_i through edge e_{ψ_i} is calculated recursively and denoted Ω_i (line 7).

2) *Backtracking*: Lastly, the optimal solution path is found by backtracking from v_{end} to v_{start} by recursively following the back-pointers ψ until v_{start} is found. Backtracking will always

find v_{start} due to the adjustments in Sec. IV-C. The monitoring effectiveness is $F = \Omega_{\text{end}} + (\tau_{\text{end}}^d - \tau_{\text{end}}^a)$.

A. Analysis

For each v_i , the forward pass calculates the preceding vertex ψ_i and the sum of edge weights Ω_i for the optimal path from v_{start} to v_i , if the mission ends at time τ_i^a . The algorithm recursively solves optimal sub-problems until the full problem is solved optimally.

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 complexity for generating the graph is $\mathcal{O}(|\mathcal{P}|^2 \cdot |\mathcal{T}|^2)$. The topological sort has complexity $\mathcal{O}(|\mathcal{V}| \log |\mathcal{V}|)$ and the path search is $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$. Therefore the sweep-plane algorithm has time complexity $\mathcal{O}(|\mathcal{P}|^2 \cdot |\mathcal{T}|^2)$.

VI. EXPERIMENTS

This section describes simulation experiments in effective monitoring of an AUV by a manned surface vehicle. In this case, effective monitoring (defined formally in Sec. III-C) allows the surface vehicle to communicate with the AUV, respond to critical events, or intervene in the mission.

Simulations were performed using the same parameter values and target trajectories as the AUV missions described in [6]. Parameter values are as follows: $r = 200$ m monitoring range, 2 m/s constant target speed, 25 m grid spacing, $\Delta_t = 10$ s time steps, travel time $\delta(\hat{y}_i, \hat{y}_j) = \frac{\|\hat{y}_j - \hat{y}_i\|}{\|\dot{y}\|} + T_{\text{pen}}$, with $\|\dot{y}\| = 5$ m/s tracker speed and $T_{\text{pen}} = 30$ s for deploying and retrieving the monitoring hardware. Two hour-long AUV missions are considered as target trajectories, named Middle Harbour and Jervis Bay. The *circular* and *linear* missions are two extreme cases for the trajectory.

A. Deterministic Target Trajectory

Table I shows simulation results for four deterministic target trajectories. The deterministic and the generalised algorithms output the same solution trajectories, however the generalised algorithm had a higher computation time. The algorithm shows an improvement in the objective function over the Genetic Algorithm results reported in [6]. The key advantage of the proposed sweep plane algorithm is faster and guaranteed runtime, and provably optimal solutions.

TABLE I
SIMULATION RESULTS FOR DETERMINISTIC TARGET TRAJECTORIES. THE TWO PLANNERS OUTPUT IDENTICAL SOLUTION TRAJECTORIES.

Mission	F/T	M	Deterministic		Generalised	
			$ \mathcal{V} $	$time$	$ \mathcal{V} $	$time$
Middle Harbour	79.5	8	2860	0.5	49483	101
Jervis Bay	79.2	7	3121	0.6	59146	144
<i>circular</i>	95.8	3	1203	0.3	27338	32
<i>linear</i>	52.2	6	430	0.2	8062	3.7

Columns: Objective function F/T %; Num. stopping locations M ; Num. vertices $|\mathcal{V}|$; Computation time (s).

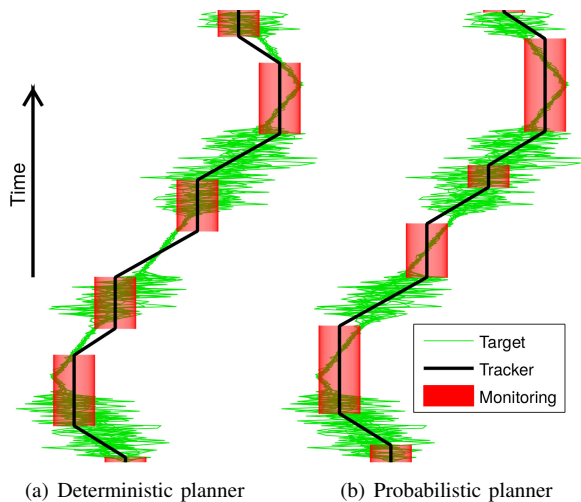


Fig. 6. Comparing planning with a deterministic model to planning with an accurate uncertainty model. Green lines are sample target trajectories drawn from the probabilistic model. Red regions represent the monitoring range around the chosen stopping locations.

B. 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. Fig. 6 presents a target mission that alternates between sections with high spatial uncertainty and low spatial uncertainty. The illustrated target trajectories are samples drawn from this uncertainty model.

Fig. 6(a) shows the optimal stopping locations for the tracker if there were no uncertainty in the target trajectory. Fig. 6(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 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 $F/T = 47.5\%$, 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.

C. Realistic Probabilistic Trajectory

Now we consider an example probability distribution definition. 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. The speed \dot{d}_i along the path at any time instance is assumed to be normally distributed $\dot{d}_i \sim \mathcal{N}(\|\dot{x}\|_{\text{ave}}, \sigma^2)$ and independent of other time instances. This gives the recursive definition for d_i as

$$d_{i+1} = d_i + \Delta_t \dot{d}_i. \quad (8)$$

The general solution to (8), for $d_1 = 0$ with zero uncertainty, gives the state estimate with mean and variance

$$\mu_i = (i-1)\|\dot{x}\|_{\text{ave}}\Delta_t \quad \text{and} \quad \Sigma_i = (i-1)\sigma^2\Delta_t^2. \quad (9)$$

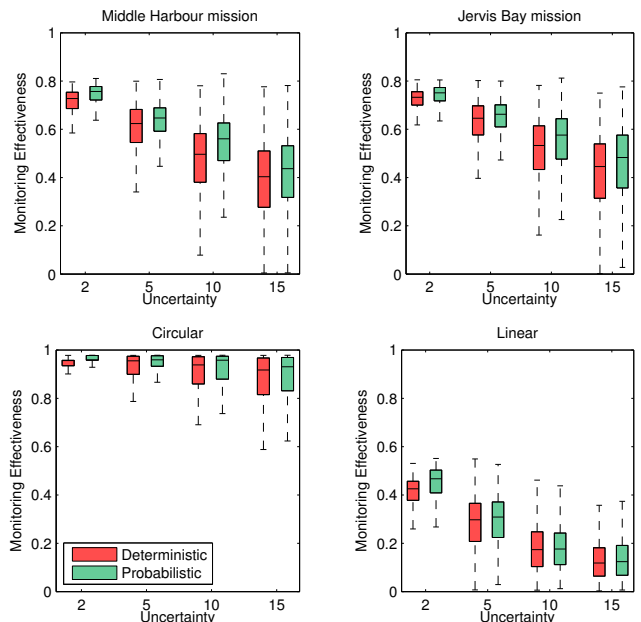


Fig. 7. Monte Carlo simulation results for a probabilistic target (10000 samples); planning with and without taking into account the uncertainty model. F/T on vertical axes; σ_{rate} (uncertainty growth rate) on horizontal axes. Error bars show the sample minima, quartiles and maxima.

Fig. 7 shows the results of Monte Carlo simulations performed by drawing 10000 sample target trajectories directly from (8) and the objective function was evaluated using the planned tracker trajectory. Planning was performed either with or without taking into account the uncertainty model (9). The horizontal axes shows varying speed uncertainty $\sigma \propto \sigma_{\text{rate}}$; such that σ_{rate} is defined as the standard deviation of completion time in minutes for a 1 hour mission.

The monitoring effectiveness is higher when planning using the uncertainty model, due to the reasons discussed in Sec. VI-B. A single-tailed t -test confirms this ($p < 0.01$) for all 16 missions except *linear* with $\sigma_{\text{rate}} = 10$.

VII. DISCUSSION AND FUTURE WORK

The results validate the performance of our algorithm and show the value of the probabilistic formulation. Our implementation is unoptimised, but still exhibits reasonable clock-time performance. Execution time ranged from milliseconds to a few minutes using a standard desktop computer. The approach is feasible for operational use as is, and can easily be used in replanning for partially known mission trajectories that are discovered over time. Our results also motivate many important avenues of future work, including generalising for probabilistic representations of the monitor's observation range and dynamic communication rates [26]. Further, an orientation-dependent monitoring model could be accommodated by adding a tracker-orientation dimension to the search space. It is also interesting to consider worst-case as opposed to expected observation time, and multi-robot cases.

ACKNOWLEDGEMENTS

This work is supported in part by the Australian Centre for Field Robotics and the New South Wales State Government.

REFERENCES

- [1] A. Pedro Aguiar, J. Almeida, M. Bayat, B. Carneira, 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*, pages 412–417, 2009.
- [2] J. Almeida, C. Silvestre, and A. Pascoal. Cooperative control of multiple surface vessels with discrete-time periodic communications. *Int. J. Robust Nonlin.*, 22(4): 398–419, 2012.
- [3] R.P. Anderson and D. Milutinovic. On the construction of minimum-time tours for a Dubins vehicle in the presence of uncertainties. *J. Dyn. Sys., Meas., Control*, 137(3): 031001 – 031001–8, 2014.
- [4] A. Bahr, J.J. Leonard, and M.F. Fallon. Cooperative localization for autonomous underwater vehicles. *Int. J. Robot. Res.*, 28(6):714–728, 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] G. Best and S. Anstee. Motion planning for autonomous underwater vehicle supervision. In *Proc. of ARAA ACRA*, 2014.
- [7] G. Best and P. Moghadam. An evaluation of multi-modal user interface elements for tablet-based robot teleoperation. In *Proc. of ARAA ACRA*, 2014.
- [8] M. Bibuli, M. Caccia, L. Lapierre, and G. Bruzzone. Guidance of unmanned surface vehicles: Experiments in vehicle following. *IEEE Robot. Autom. Mag.*, 19(3):92–102, 2012.
- [9] S.D. Bopardikar, S.L. Smith, and F. Bullo. On dynamic vehicle routing with time constraints. *IEEE Trans. Robot.*, 30(6):1524–1532, 2014.
- [10] 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*, 8031:803111 1–12, 2011.
- [11] P. Calado and J. Sousa. Leader-follower control of underwater vehicles over acoustic communications. In *Proc. of IEEE OCEANS*, 2011.
- [12] Y.S. Chow, H. Robbins, and D. Siegmund. *Great expectations: The theory of optimal stopping*. Houghton Mifflin Boston, 1971.
- [13] 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 RSS*, 2015.
- [14] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*, volume 2. MIT press Cambridge, 2001.
- [15] M. De Berg, M. Van Kreveld, M. Overmars, and O.C. Schwarzkopf. *Computational geometry*. Springer, 2000.
- [16] C. D’Este, B. Seton, J. McCulloch, D. Smith, and C. Sharman. Avoiding marine vehicles with passive acoustics. *J. Field Robot.*, 32(1):152–166, 2015.
- [17] T.F. Dono. Optimized landing of autonomous unmanned aerial vehicle swarms. Master’s thesis, Naval Postgraduate School, 2012.
- [18] M. Dunbabin and L. Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robot. Autom. Mag.*, 19(1):24–39, 2012.
- [19] M. Dunbabin and A. Tews. Toward robotic visual and acoustic stealth for outdoor dynamic target tracking. In *Proc. of ARAA ACRA*, 2012.
- [20] 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.*, 29(12): 1461–1474, 2010.
- [21] C. Fang and S. Anstee. Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle. In *Proc. of IEEE OCEANS*, 2010.
- [22] 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.
- [23] 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.*, 69(1-2):137–145, 2008.
- [24] 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.*, 7(2):5–11, 2013.
- [25] 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.*, OnlineFirst, 2015.
- [26] A. Kassir, R. Fitch, and S. Sukkarieh. Communication-aware information gathering with dynamic information flow. *Int. J. Robot. Res.*, 34(2):173–200, 2015.
- [27] N. Kottege and U.R. Zimmer. Underwater acoustic localization for small submersibles. *J. Field Robot.*, 28(1):40–69, 2011.
- [28] S.M. LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [29] E.L. Lawler. *Combinatorial optimization: Networks and matroids*. Courier Dover Publications, 1976.
- [30] M. Lindhé and K.H. Johansson. Exploiting multipath fading with a mobile robot. *Int. J. Robot. Res.*, 32(12): 1363–1380, 2013.
- [31] F. Lobo Pereira, J. Sousa, R. Gomes, and P. Calado. A model predictive control approach to AUVs motion coordination. In J.H. van Schuppen and T. Villa, editors, *Coordination Control of Distributed Systems*, pages 9–18. Springer, 2015.
- [32] 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*, pages 3497–3502, 2013.
- [33] T. Peynot, S.-T. Lui, R. McAllister, R. Fitch, and S. Sukkariéh. Learned stochastic mobility prediction for planning with control uncertainty on unstructured terrain. *J. Field Robot.*, 31(6):969–995, 2014.
- [34] M. Saska, V. Vonásek, T. Krajník, 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.*, 33(10):1393–1412, 2014.
- [35] J.M. Soares, A.P. Aguiar, A.M. Pascoal, and A. Martinoli. Joint ASV/AUV range-based formation control: Theory and experimental results. In *Proc. of IEEE ICRA*, pages 5579–5585, 2013.
- [36] A. Tews and M. Dunbabin. Acoustic masking of a stealthy outdoor robot tracking a dynamic target. In J.P. Desai, G. Dudek, O. Khatib, and V. Kumar, editors, *Experimental Robotics*, pages 775–786. Springer, 2013.
- [37] P. Toth and D. Vigo, editors. *The vehicle routing problem*. Society for Industrial and Applied Mathematics, Philadelphia, 2001.
- [38] 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.*, 19(1):73–84, 2012.