Journal of Intelligent & Robotic Systems (2022) 106:2
https://doi.org/10.1007/s10846-022-01676-3

REGULAR PAPER

Trajectory Planning and Optimization for Minimizing Uncertainty in Persistent Monitoring Applications

Michael Ostertag[1] · Nikolay Atanasov[1] · Tajana Rosing[2]

Received: 6 January 2022 / Accepted: 10 June 2022

Abstract

This paper considers persistent monitoring of environmental phenomena using unmanned aerial vehicles (UAVs). The objective is to generate periodic dynamically feasible UAV trajectories that minimize the estimation uncertainty at a set of points of interest in the environment. We develop an optimization algorithm that iterates between determining the observation periods for a set of ordered points of interest and optimizing a continuous UAV trajectory to meet the required observation periods and UAV dynamics constraints. The interest-point visitation order is determined using a Traveling Salesman Problem (TSP), followed by a greedy optimization algorithm to determine the number of observations that minimizes the maximum steady-state eigenvalue of a Kalman filter estimator. Given the interest-point observation periods and visitation order, a minimum-jerk trajectory is generated from a bi-level optimization, formulated as a convex quadratically constrained quadratic program. The resulting B-spline trajectory is guaranteed to be feasible, meeting the observation duration, maximum velocity and acceleration, region enter and exit constraints. The feasible trajectories outperform existing methods by achieving comparable observability at up to 47% higher travel speeds, resulting in lower maximum estimation uncertainty.

Keywords Informative path planning · Persistent monitoring · Aerial systems: perception and autonomy

1 Introduction

Persistent sensing and data collection is important for tracking environmental phenomena such as atmospheric pollution [1, 2] and wildfire ignition at the wildland-urban interface [3, 4]. Early detection of dangerous conditions is a critical factor in mitigating damages to ecological systems and human infrastructure [5]. Persistent monitoring methods have utilized satellite data, human-piloted aircraft, and networks of stationary sensors, optimizing their placement for a variety of resource constraints [6–9]. Satellite data is often at too low resolution (e.g. pixel widths of 1 km for MODIS and 375 m for VIIRS) or too infrequent to detect rapidly emerging trends [10–13]. Regular flights of human-piloted vehicles is useful for ongoing tracking efforts in localized regions but is too expensive for large-scale monitoring. Dense sensor deployments can monitor the environment effectively but as the size of the region grows, sensor network deployment and maintenance can quickly become cost prohibitive. Stationary sensing networks also suffer from information redundancy since many environmental phenomena are highly temporally and spatially correlated.

Recent persistent monitoring research has focused on developing motion planning and control techniques for mobile sensing robots, such as unmanned aerial vehicles (UAVs), that can monitor large environments and deliver high-frequency, high-resolution data more effectively [14–22]. Technological advancements in sensing, computation, and communication enable UAVs to supply critical real-time data for a variety of applications, depending on the equipped visual, thermal, chemical, or depth sensors.

The goal of persistent monitoring is to minimize estimation uncertainty by planning an optimal trajectory for

✉ Michael Ostertag
mosterta@ucsd.edu

Nikolay Atanasov
natanasov@eng.ucsd.edu

Tajana Rosing
tajana@ucsd.edu

[1] Department of Electrical and Computer Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla 92093, CA, USA

[2] Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla 92093, CA, USA

Published online: 16 August 2022

the sensing platforms. Probabilistic filtering techniques are employed to integrate the sensed information over time into an estimate the signal of interest. Kalman filtering is commonly used [22, 23], including generalizations such as extended and unscented Kalman filtering [24], particle filtering [25], and variational inference [26].

Researchers in persistent monitoring have used a variety of objective functions for quantifying the effectiveness of an informative trajectory. We chose to minimize the maximum eigenvalue of the steady-state covariance matrix, which intuitively represents a bound on the uncertainty of the model at any POI and is commonly referred to as the spectral radius or E-optimality [27]. Other potential objective functions include the trace of the covariance matrix, which is the sum of the uncertainty, or the determinant of the covariance matrix, which is the volume of uncertainty [28]. By optimizing for the steady-state uncertainty, the developed controller may perform suboptimally for initial cycles of a periodic traversal of a trajectory, but it can be shown that the uncertainty evolution converges at an exponential rate from any initial covariance to a cyclic pattern of values (Sec IV.A in [29]).

Our previous work [22] proposes an initial solution for a UAV flying along a prescribed sensing trajectory. The objective was to plan a velocity profile along the path that minimizes the maximum eigenvalue of the estimation covariance matrix of a Kalman filter. In contrast to other works, which only optimize to minimize the revisitation frequency [30], the method balances the number of consecutive measurements at a point with the frequency of visitation. While the generated trajectory obeyed maximum velocity constraints of the UAV, the sensing trajectory was not continuous, and resulted in paths that were infeasible to follow (Fig. 1).

Contributions We present a planner for feasible trajectories for persistent monitoring with UAVs. Modern approaches [14, 22, 23, 31] ignore the dynamics of monitoring platforms, resulting in violations of full observability assumptions during simulations with controllers that track a trajectory. In contrast, our proposed feasible Trajectory Optimization for Persistent monitoring (f-TOP) plans a trajectory, adhering to a UAV's dynamic constraints while minimizing estimation uncertainty, with the following contributions:

1. Bi-level optimization that plans a minimum-jerk, three times continuously differentiable trajectory that minimizes steady-state estimation uncertainty,
2. Greedy Knockdown Algorithm to determine the optimal number of observations of each POI to minimize worst-case bound on estimation uncertainty for periodic sampling,

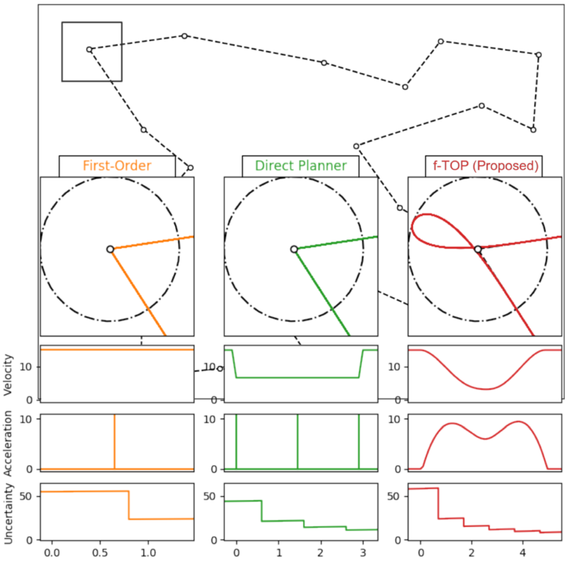

Fig. 1 Our proposed f-TOP algorithm attempts to plan trajectories that adhere to the dynamic constraints of the system, resulting in trajectories that can be tracked and remaining within the sensing distance of a POI. The First-Order and Direct Planner generate infeasible trajectories with discontinuities in the velocity profile

3. Second-order cone program (SOCP) for calculating B-spline trajectory coefficients that enforce constraints of vehicle dynamics.

The remaining paper is organized as follows. Section 2 outlines related work relevant to trajectory planning for persistent monitoring. Section 3 describes the problem formulation. Section 4 provides background information necessary for the formulation of our optimization. Section 5 describes our optimization approach for planning feasible trajectories. Section 6 compares our proposed method against baseline and existing works. The paper concludes with Section 7.

2 Related Work

Our contribution is at an intersection of several related research areas in persistent monitoring. Inspired by the periodic sensing results from the stationary sensing literature [8, 9], we extend Kalman filter convergence results to persistent monitoring scenarios. We focus on planning dynamically feasible trajectories for robotic sensing platforms that better meet observability assumptions, which are commonly assumed but often unachieveable in existing works without significant reduction in robot performance. In this section, we provide a brief overview of relevant literature in sensor scheduling, trajectory planning for persistent monitoring, and feasible trajectories.

2.1 Sensor Scheduling

In sensor scheduling, a group of stationary sensors attempts to measure and approximate a phenomenon in a resource-efficient manner by selecting a subset of sensors to sample at each time step. Initial work calculates bounds on the estimation uncertainty by forming sensor scheduling as a convex problem problem and creating heuristic and open-loop policies [6, 32]. Zhao et al. [29] advance the sensor scheduling problem for infinite time horizons, proving that for linear Gaussian processes and sensing functions the steady-state solution is independent of initial estimation covariance and the optimal estimation can be approximated arbitrarily close by a finite, periodic schedule. Greedy scheduling algorithms work well in many cases, approaching optimal results at reduced computational complexity. Jawaid and Smith [7] leverage submodularity of the estimation error to show that greedy approaches can approach optimal results, and additional efforts prove greedy algorithms can be optimal for specific cases of uncorrelated noise [8, 9].

The primary difference between persistent monitoring with robotic agents and sensor scheduling is that there is a non-negligible switching cost between targets for robotic sensing platforms, which must physically move to measure new regions. Despite the difference, we can leverage and extend sensor scheduling results [9, 29] by modeling our POIs as uncorrelated and proposing a finite, periodic schedule with an additional period where no observations occur.

2.2 Trajectory Planning for Persistent Monitoring

Instead of relying on fixed-position sensors for estimation, an alternate cost-effective approach for sensing across a large area is UAV-mounted sensors [33]. The simplest formulation of persistent monitoring a set of discrete POIs is the patrol problem [34], where every POI must be visited with the objective of minimizing time between revisits. Minimizing time between revisits results in a near-optimal strategy when all POIs have equivalent uncertainty in state transitions, but are sub-optimal when the POIs have different uncertainty characteristics.

For POIs with varying rates of uncertainty, persistent monitoring trajectories can be planned that optimize one of several different metrics of uncertainty, such as maximum eigenvalue, log determinant, or trace of an estimate covariance matrix [28]. Decomposing trajectory generation into separate path planning and velocity control problems allows for developing optimal solutions for constrained problems with reduced complexity [35]. The first approaches for planning persistent monitoring trajectories optimize wait times, visit orders, and velocity along a given 1-dimensional path.

For a given 1-dimensional path, the persistent monitoring problem can be simplified into calculating a required length of observation time at each POI prior to travelling to the next. The length of observation time is modelled based on different accumulation and clearing models of uncertainty. Smith et al. [14] represent uncertainty at POIs is modelled as a linear, continuous accumulation model with uncertainty reduction directly proportional to the time spent measuring a point, planning a periodic velocity as a composition of basis functions that minimized the maximum accumulation value. Similar works expand the concepts to different uncertainty accumulation models that clear in linear proportion to the distance of nearby robotic platforms [15] and for symmetric, non-linear accumulation models [16]. Ostertag et al. [22] plans a velocity along a prescribed trajectory by determining the optimal dwell times using a greedy application of an Kalman filter at an infinite-time horizon. Researchers have planned trajectories in two-dimensional space using the concept of infinitesimal perturbation analysis, which approximates the gradient of the cost function but with limited guarantees of optimality [18, 20].

Related to persistent monitoring is the field of informative trajectory planning where a robotic agent optimizes for information gain while travelling through an environment but does not necessarily measure POIs periodically or with full observability. A common formulation is the orienteering problem [36], whereby an agent travels along a graph and collects rewards for visiting locations, attempting to maximize the reward with a limited resource spent on each edge taken. For problems formatted on continuous planes, optimal solutions are NP-hard, but recent work in sampling-based path planning [19, 37, 38] and finite-horizon, search-based path planning [39] show promising results for exploration problems, which are similar but not immediately applicable to persistent monitoring.

2.3 Trajectory Representation for UAVs

When dealing with real UAV systems, trajectory continuity is important to consider since trajectories with abrupt changes in velocity or acceleration, such as are common in simple waypoint paths, will result in high tracking error when a UAV is operating at high velocities. Continuous trajectories are more complicated to solve, so researchers have simplified the problem into graph searches using a discrete number of fixed motion primitives [40, 41]. While these approaches are useful for planning real-time actions due to the limited number of movement patterns, they are almost always sub-optimal.

To handle the complexity of developing continuous trajectories, researchers have formulated the trajectory optimization as finding coefficients for a set of basis functions. By limiting to searching for the set of coefficients, the solution space is sufficiently limited to formulate the problem in a tractable manner. Objective functions commonly minimize the snap (4th derivative) [40, 42–45] or jerk (3rd derivative) [46, 47] of the trajectory. Quadrotors are differentially flat [48], meaning that for a selected set of trajectory parameters, typically position in three dimensions and yaw, the control inputs can be precisely calculated from the first three derivatives of the trajectory.

Initial work in developing trajectories for quadrotors uses piece-wise polynomial basis functions. The polynomial basis functions have issues with numerical stability and are difficult to fit to non-linear constraints on maximum velocity and acceleration. Mellinger et al. [42] formulate the optimization as a quadratic program optimizing for coefficients of piecewise two-time continuously differentiable polynomial basis curves, solving for an arbitrary time range and scaling to meet any required constraints. The constraints are linear, however, and checked only at discrete time instances, which results in a large number of linear constraints and no guarantee of meeting maximum velocity or acceleration constraints. Mueller et al. [40] resolves the numerical stability of polynomial basis functions by for end point conditions instead of coefficients, proving a simple mapping between end point conditions and the coefficients. Richter et al. [44] improves the speed to solve the optimization by solving an unconstrained quadratic optimization followed by a check on constraints. Polynomial bases are numerically ill-conditioned for complex paths, and as the order of the polynomial increases to meet pathing demands, properties such as curvature and the extrema of derivatives must be found using numerical root-finding methods.

To solve the shortcomings of polynomial basis functions, researchers began exploring Bezier curves and B-splines. This class of curves are defined by a set of control points, are numerically stable, and have beneficial properties of convexity and linear formulation of derivatives. The last property enables simple calculation for a bound on the maximum magnitude of lower order derivatives. B-splines have a limited support for any given point in time with the support transition in accordance to a set of coefficients called knots. The limited support enables further numerical stability and complex maneuvers. Ding et al. [49] leverage the properties of B-splines to efficiently solve for a path using a kinodynamic search and elastic optimization to fine tune control points. Zhou et al. [50] utilize a similar method, but model the path as a non-uniform B-spline with an iterative time adjustment method, adjusting the knot spacing to achieve more aggressive trajectories. Due to the convex hull property of B-splines, both methods place constraints on the control points to avoid collisions.

Due to the advantages of numerical stability, continuity, and complex pathing while meeting dynamic constraints able to meet dynamic constraints of a robotic platform, we plan our trajectories by developing optimizations for the control points of B-splines. Unlike [50], our proposed trajectory optimization minimizes the magnitude of jerk, which enables simple computation for a 4th-order B-spline.

3 Problem Formulation

Given a set of points of interest (POIs) in the environment, we focus on planning a dynamically feasible trajectory for a quadrotor robot to estimate a time-dependent value of interest at these points with minimum uncertainty. Prior to stating the problem formally, we define the environment model, quadrotor dynamics, and the sensing model. The paper follows the convention of denoting matrices by capitalized bold symbols $\boldsymbol{A}$ and vectors by lowercase bold symbols $\boldsymbol{a}$.

3.1 Environment Model

Consider a set of N POIs spread throughout an environment $\mathcal{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_N\}$ where $\mathbf{q}_i \in \mathbb{R}^3$. The objective is to track the values $\boldsymbol{x}_k = [x_1(t_k), ..., x_N(t_k)]^\intercal$ of an environmental phenomenon at the N POIs over discrete time steps $t_k, t_{k+1}, \ldots$. Time is discretized with sampling frequency f_s and associated period f_s^{-1} such that $t_{k+1} = t_k + f_s^{-1}\ \forall\ k \in \mathbb{Z}_{\geq 0}$. To simplify notation, variables that depend on time are denoted with a subscript k where appropriate, e.g. $\boldsymbol{A}_k = \boldsymbol{A}(t_k)$. We assume the values at each location are independent and time varying according to the following discretized model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{m}_k, \tag{1}$$

$$\boldsymbol{m}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{W} f_s^{-1}), \tag{2}$$

where $\boldsymbol{m}_k$ are independent increments drawn from a Gaussian distribution with mean value change $\boldsymbol{\mu}_k$ and a diagonal covariance $\boldsymbol{W} f_s^{-1}$.

3.2 Quadrotor Model

A quadrotor equipped with a sensor is attempting to estimate the phenomenon at each POI. The state of the quadrotor is defined by its orientation $\boldsymbol{R} \in SO(3)$, world-frame position $\mathbf{s} \in \mathbb{R}^3$, velocity $\dot{\mathbf{s}} \in \mathbb{R}^3$, acceleration $\ddot{\mathbf{s}} \in \mathbb{R}^3$, and jerk $\dddot{\mathbf{s}} \in \mathbb{R}^3$. The control inputs of the quadrotor are the body-frame thrust $f \in \mathbb{R}_{\geq 0}$ and rotational velocity $\boldsymbol{\omega} \in \mathbb{R}^3$, which are directly related to the rotational velocity of the four rotors [42]. The quadrotor state evolves with the following dynamics:

$$\ddot{\mathbf{s}} = \boldsymbol{R}\boldsymbol{e}_3 f / m + g\boldsymbol{e}_3, \tag{3}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\hat{\boldsymbol{\omega}}, \tag{4}$$

where m is the mass of the quadrotor, g is the acceleration due to gravity, $\boldsymbol{e}_3$ is the unit vector for z-axis in the world frame, and $\hat{\cdot} : \mathbb{R}^3 \mapsto \mathfrak{so}(3)$ denotes the hat map, which maps 3-D axis-angle vectors to a skew-symmetric matrix.

The quadrotor system is differentially flat with respect to its position $\mathbf{s}$ and yaw angle ψ states [48]. This means that, given a desired time-parametrized flat trajectory $\mathbf{s}_d$, ψ_d, we can recover the control inputs f and $\boldsymbol{\omega}$ that will track the desired trajectory:

$$f = m\|\ddot{\mathbf{s}}_d + g\boldsymbol{e}_3\|, \tag{5}$$

$$\hat{\boldsymbol{\omega}} = \boldsymbol{R}_d^\top \dot{\boldsymbol{R}}_d, \tag{6}$$

where $\boldsymbol{R}_d$ and $\dot{\boldsymbol{R}}_d$ are the desired rotation and rotational velocity. The desired orientation $\mathbf{R}_d$ is obtained from rotation about yaw ψ_d and desired acceleration $\mathbf{a}_d$ according to [42]. The desired rotation and acceleration are provided by the planned trajectory that is generated in Section 5, which the UAV is attempting to track.

The quadrotor has bounds on its velocity and acceleration due to physical constraints. The total thrust that can be generated by the motors is limited, resulting in a maximum acceleration bound. To account for the lack of air drag in the simplified dynamics model, we also introduce a maximum velocity constraint. Hence, all intended quadrotor trajectories should lie in the admissible set:

$$\mathcal{S} := \{\mathbf{s} \in C^3(\mathbb{R}, \mathbb{R}^3) \mid \|\dot{\mathbf{s}}\| \leq v_{\max}, \|\ddot{\mathbf{s}}\| \leq a_{\max}\}, \tag{7}$$

where $C^3(\mathbb{R}, \mathbb{R}^3)$ denotes the space of three-times continuously differentiable functions $\mathbb{R} \mapsto \mathbb{R}^3$, $v_{\max}$ is the maximum velocity magnitude, and $a_{\max}$ is the maximum acceleration magnitude.

3.3 Sensing Model

A sensor is mounted on the quadrotor that can sense a POI q_i if it is within a given closed, convex region B_i, which represents the union of all UAV positions such that $\mathbf{q}_i$ is in the sensor field of view. An example trajectory and observation region are shown in Fig. 2. The sensor captures observations with a sampling frequency of f_s and each measurement is corrupted by additive noise:

$$\boldsymbol{y}_k = \mathbf{H}_k \boldsymbol{x}_k + \boldsymbol{\eta}, \tag{8}$$

$$\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{V}), \tag{9}$$

$$\mathbf{H}_{i,j}(t_k) = \begin{cases} 1, & i = j \text{ and } \mathbf{s}(t_k) \in B_i, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

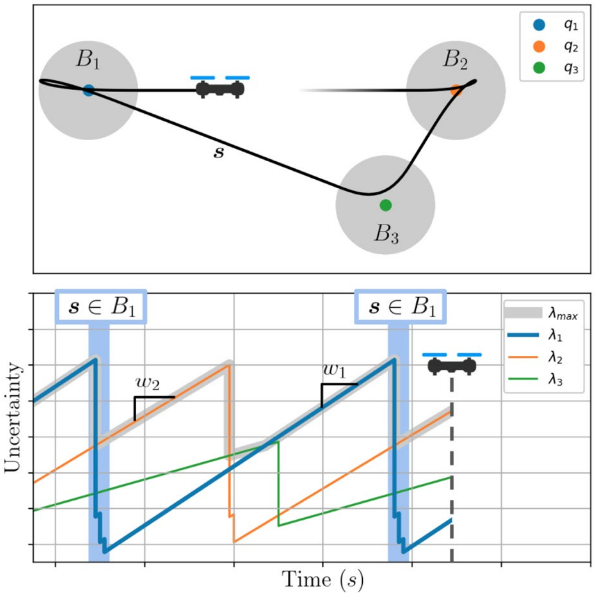

Fig. 2 Persistent monitoring example of 3 POIs and their associated observation regions. (top) The minimum-jerk trajectory must spent a different amount of time within each observation region to meet the constraints of 3 observations in B_1, 2 observations in B_2, and 1 observation in B_3. The trajectory is constrained to remain within each observation region. (bottom) Evolution of the covariance over time. When not observed, the uncertainty grows linearly in proportion to $\mathbf{W}$

where $\mathbf{H}_k$ is a binary observation matrix and $\boldsymbol{\eta}$ is additive, Gaussian observation noise with a positive semidefinite covariance matrix $\boldsymbol{V}$. The matrix $\mathbf{H}_k$ models the sensor field of view by assigning 1 to visible POIs and 0, otherwise. Equation 8 models a sensor that provides direct measurements of the features $\boldsymbol{x}_k$ at the POIs. For example, a downward-facing visible-light or thermal camera used for vegetation imaging or temperature estimation can be modeled by Eq. 8.

The true value of the environmental phenomenon is continuous in time but due to the discrete-time observations, $\boldsymbol{x}_k$ is estimated using a discrete-time Kalman filter. The filter maintains a mean estimate z_k and a covariance matrix $\boldsymbol{\Sigma}_k$, whose values are predicted using the environment model in Eq. 2 and updated using the sensor measurements and the sensing model in Eq. 8. As shown in Fig. 3, the predict stage incorporates the Gaussian noise of the environment process to obtain a predicted estimate z_k^- and associated uncertainty $\boldsymbol{\Sigma}_k^-$, and the update stage applies the information gained from an observation $\boldsymbol{y}_k$ to obtain an improved estimate z_k with uncertainty $\boldsymbol{\Sigma}_k$. The estimated value is important for data collection and is introduced for completeness, but our trajectory optimization is only concerned with minimizing the covariance of the estimation. Despite the discrete updates of the Kalman filter, the precise values of the covariance matrix can be calculated at any point t using the following:

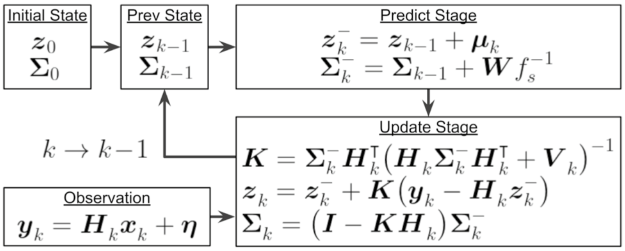
Initial State
z_0
Σ_0
Prev State
z_{k-1}
Σ_{k-1}
Predict Stage
$z_k^- = z_{k-1} + \mu_k$
$\Sigma_k^- = \Sigma_{k-1} + W f_s^{-1}$
$k \rightarrow k-1$
Update Stage
$K = \Sigma_k^- H_k^\intercal (H_k \Sigma_k^- H_k^\intercal + V_k)^{-1}$
$z_k = z_k^- + K(y_k - H_k z_k^-)$
$\Sigma_k = (I - K H_k)\Sigma_k^-$
Observation
$y_k = H_k x_k + \eta$

Fig. 3 The Riccati update procedure for a discrete Kalman filter consists of two stages: a Predict Stage that updates the mean and covariance of the estimate based on the known system model and an Update Stage that refines the prediction according to the measured value and observation noise

$$\dot{\boldsymbol{\Sigma}}(t) = \begin{cases} \boldsymbol{W} + c_k\delta(t-t_k)\,, & \mathbf{s}(t_k) \in B_i, \\ \boldsymbol{W}\,, & \text{otherwise,} \end{cases} \tag{11}$$

$$c_k = -\boldsymbol{\Sigma}_k \mathbf{H}_k^\intercal \left(\mathbf{H}_k \boldsymbol{\Sigma}_k \mathbf{H}_k^\intercal + V\right)^{-1} \mathbf{H}_k \boldsymbol{\Sigma}_k, \tag{12}$$

where c_k is the result of the Riccati Kalman filter update and $\delta(t-t_k)$ is the Dirac delta function.

Each observation reduces the estimation uncertainty of a POI by a monotonically decreasing amount. This property of diminishing returns of the Kalman filter has been exploited to optimize sensor selection and scheduling in previous works [8, 9, 51, 52]. As we describe later, the number of consecutive observations of a POI d_i will be an important optimization target due to the inherent trade off between reducing the uncertainty of individual POIs and reducing total cycle time, impacting the uncertainty of all POIs as described in Section 4.1.

3.4 Problem Statement

Our objective is to plan a trajectory for a sensing quadrotor that results in a minimum estimation uncertainty across all POIs. While researchers have used a variety of optimization goals [28], we choose to minimize the maximum eigenvalue of the steady-state covariance matrix, which bounds the estimation uncertainty at any POI and is commonly referred to as E-optimality [27].

When considering persistent sensing over an infinite time horizon, optimal trajectories have been proven to be approximately periodic [29]. Hence, we restrict our focus to the class of periodic trajectories, which maintain full observability of the POIs at every loop. An infinite time horizon is selected for the optimization goal because any initial covariance will converge to an infinite-horizon steady-state condition exponentially fast for a periodic trajectory [29], removing the influence of initial conditions. Additionally, we obtain a solution that works for any initial position along the trajectory, which is equivalent to starting the trajectory at any point within a sensing period. While suboptimal in some instances, removing the reliance on specific timing of sampling allows for a lower cycle time and guarantees a computable upper bound for the maximum estimation uncertainty.

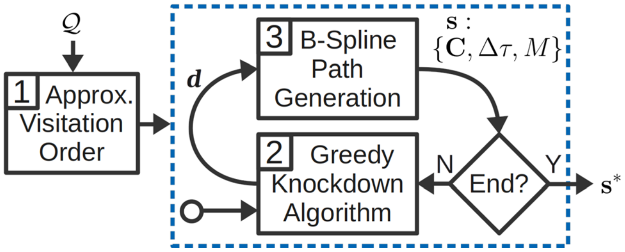
$\mathcal{Q}$
1 Approx. Visitation Order
$\boldsymbol{d}$
3 B-Spline Path Generation
$\mathbf{s}:$
$\{\mathbf{C}, \Delta\tau, M\}$
2 Greedy Knockdown Algorithm
N
End?
Y
$\mathbf{s}^*$

Fig. 4 Algorithm flow of the Feasible Trajectory Optimizer for Persistent Monitoring (f-TOP), which computes optimal observation periods and feasible trajectories in an iterative manner

Problem 1 **(Minimizing Steady-State Uncertainty subject to Trajectory Feasibility Constraints)** For a quadrotor with dynamics in Eqs. 3-6 and POIs $\mathcal{Q}$, find a periodic, time-parameterized trajectory $\mathbf{s}$ with loop period T to minimize the maximum eigenvalue $\lambda_{\max}^{\infty}$ of the steady-state Kalman filter covariance matrix $\boldsymbol{\Sigma}$ that evolves as defined in Eq. 11 at an infinite-time horizon for any initial sampling position:

$$\begin{aligned} \mathbf{s}^* \; &= \underset{T,\mathbf{s}}{\arg\min}\; \left(\lambda_{\max}^{\infty}(T,\mathbf{s})\right) \\ &\text{s.t.} \quad \mathbf{s} \in \mathcal{S} \end{aligned} \tag{13}$$

where $\mathcal{S}$ is the admissible set of trajectories defined in Eq. 7 and $\lambda_{\max}^{\infty}$ is defined as:

$$\lambda_{\max}^{\infty}(T,\mathbf{s}) = \limsup_{t\to\infty} \left[\max_{t\le\tau\le t+T} \lambda_{\max}(\tau,\mathbf{s}(\tau))\right] \tag{14}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $\boldsymbol{\Sigma}$.

Our approach to generating a feasible minimum steady-state uncertainty trajectory considers three subproblems:

1. Determine POI visitation order,
2. Calculate optimal number of consecutive observations,
3. Generate a dynamically feasible trajectory.

As outlined in Fig. 4, the POI visitation order is calculated once for a given set of POIs $\mathbf{q}$. Then, the Greedy Knockdown Algorithm calculates the optimal observation periods d that minimize the maximum estimation uncertainty. The Feasible Trajectory Optimizer for Persistent Monitoring (f-TOP) calculates minimum-jerk trajectories that meet the observation periods d and dynamic constraints of the system. If the feasible trajectory timing does not match the optimal observation periods, then the Greedy Knockdown Algorithm and proposed f-TOP algorithm are run iteratively until convergence.

4 Optimal Observation Lengths

Our goal is to minimize the estimation uncertainty of a set of POIs by determining the optimal observation length for each POI. This presents an interesting trade off; the more time spent observing any single POI, the more uncertain we become about the current value of the other POIs.

In this section, we outline how to calculate the maximum eigenvalue of the Kalman filter estimation covariance at steady state and describe the trade-off between minimizing loop time and decreasing the uncertainty for a single POI. Then, we describe an approach to calculate the optimal observation lengths for a set of POIs. First, we determine an optimal visitation order for the POIs, which is equivalent to finding the shortest cyclic path through the POIs. For calculating the optimal observation lengths for each POI, we implement the Greedy Knockdown Algorithm (GKA) from our previous work [22] with updated stopping criteria. The output of GKA is an optimal continuous observation length for each POI, which is used as a constraint when calculating feasible paths in Section 5.

4.1 Steady-State Kalman Filter Bounds

For a discrete Kalman filter, a periodic sequence of observations will result in the estimation covariance to converge exponentially fast [29] to a periodic steady-state covariance $\mathbf{\Sigma}(t)$ where for a loop with cycle time T, $\mathbf{\Sigma}(t) = \mathbf{\Sigma}(t + T)$ as $t \rightarrow \infty$. We define the steady-state value for the i-th eigenvalue immediately prior and after observation d as $\lambda_i^{(d)}$ and $\lambda_i^{-(d)}$ with the maximum value at steady-state $\lambda_i = \lambda_i^{(1)}$ for simplicity. For POIs with uncorrelated values, the eigenvalues are ordered such that λ_i corresponds to the maximum uncertainty of POI $\mathbf{q}_i$. The maximum eigenvalue of the steady-state covariance matrix is guaranteed to be bounded if every POI is measured at least once per cycle.

Lemma 1 (Bounded Uncertainty Guarantee 9) If all POIs $\mathcal{Q}$ are observed at least once during a cycle of duration T, the maximum eigenvalue of the steady-state Kalman filter covariance $\mathbf{\Sigma}$ will be bounded as $\lim_{t\rightarrow\infty} \lambda_{\max}(t) \leq b$ for some finite constant b.

During each measurement of POI $\mathbf{q}_i$, $d_i \in \mathbb{Z}_{\geq 1}$ consecutive observations are made where each observation is binary as described in Eq. 10. Additional observations result in a reduction in uncertainty such that $\lambda_i^{(d+1)} < \lambda_i^{(d)}$, but the uncertainty reduction decreases for each observation added such that $\lambda_i^{(d)} - \lambda_i^{(d+1)} < \lambda_i^{(d-1)} - \lambda_i^{(d)}$. An example of the evolution of the eigenvalues over time at steady state can be seen in Fig. 2.

For uncorrelated POIs at steady state, the reduction in uncertainty from the d_i consecutive observations is followed by a linear increase in uncertainty in proportion to $\mathbf{W}$. The lack of correlation means that each eigenvalue evolves independently with the same cycle period T. For the purpose of computing the maximum eigenvalue of the steady-state covariance, the observations of each POI can be shifted in time to align, such that the first observation of each POI occurs at the same time instance. The rearrangement of observation times can be captured by a new condensed observation matrix $\tilde{\mathbf{H}}_k$ where k is the number of discrete observations. Let $\mathbf{E}_i$ be the elementary matrix with 1 at its i-th diagonal entry and 0 everywhere else, then $\tilde{\mathbf{H}}_k$ is defined as:

$$\tilde{\mathbf{H}}_k = \begin{cases} \sum_{\{i|d_i \geq k\}} \mathbf{E}_i, & 1 \leq k < \hat{d}, \\ \mathbf{0}, & \hat{d} \leq k, \end{cases} \tag{15}$$

where $\hat{d} = \max_i d_i$. The matrix $\tilde{\mathbf{H}}_k$ is a condensed observation matrix where $\tilde{\mathbf{H}}_1$ is the identity matrix, indicating all POIs are observed at least once, and the last non-zero observation matrix $\tilde{\mathbf{H}}_{\hat{d}}$ contains 1s for the POIs that had the highest number of observations. With all observations condensed, the maximum uncertainty for each POI λ_i with the modified time alignment will occur immediately prior to the application of $\tilde{\mathbf{H}}_1$, which is described more formally in Eq. 17.

For a periodic cycle where $T \bmod f_s^{-1} = 0$, the uncertainty can be calculated precisely. Under these conditions, each observation per loop will be taken at the exact location as the previous loop. Aperiodic cycles where $T \bmod f_s^{-1} \neq 0$ can be converted to periodic by raising T to the nearest f_s^{-1} multiple. Using $\tilde{\mathbf{H}}_k$, the worst-case interobservation time is given as:

$$\hat{T} = \lceil T f_s - \hat{d} \rceil f_s^{-1}, \tag{16}$$

where $\lceil \cdot \rceil$ is the ceiling function. For high sampling frequencies, the change in cycle time will result in minimal change in overall error, but for lower sampling frequencies, which may occur with slow-responding air quality sensors or certain image processing applications, this rounding can impact the overall performance of the system.

The worst-case steady-state uncertainties $\tilde{\mathbf{\Sigma}}$ resulting from using $\tilde{\mathbf{H}}_k$ and $\hat{T}$ forms the following system of equations:

$$\begin{aligned} \tilde{\mathbf{\Sigma}}_j &= \tilde{\mathbf{\Sigma}}_i + \mathbf{W} f_s^{-1} - \tilde{\mathbf{\Sigma}}_i \tilde{\mathbf{H}}_i \left(\tilde{\mathbf{H}}_i^{\intercal} \tilde{\mathbf{\Sigma}}_i \tilde{\mathbf{H}}_i + V_i \right)^{-1} \tilde{\mathbf{H}}_i^{\intercal} \mathbf{\Sigma}_i \\ &\qquad \forall\, 1 \leq i < \tilde{d},\ j = i + 1 \\ \tilde{\mathbf{\Sigma}}_1 &= \tilde{\mathbf{\Sigma}}_{\hat{d}} + \mathbf{W} \hat{T} - \tilde{\mathbf{\Sigma}}_{\hat{d}} \tilde{\mathbf{H}}_{\hat{d}} \left(\tilde{\mathbf{H}}_{\hat{d}}^{\intercal} \tilde{\mathbf{\Sigma}}_{\hat{d}} \tilde{\mathbf{H}}_{\hat{d}} + V_{\hat{d}} \right)^{-1} \tilde{\mathbf{H}}_{\hat{d}}^{\intercal} \tilde{\mathbf{\Sigma}}_{\hat{d}} \end{aligned} \tag{17}$$

where $\tilde{\mathbf{\Sigma}}_1$ contains the maximum uncertainty for all POIs immediately prior to the first observation, such that $[\tilde{\mathbf{\Sigma}}_1]_{i,i} = \lambda_i$. For smaller values of $\hat{d}$, the equations can be quickly solved directly, but for larger values of $\hat{d}$, the problem can be solved efficiently using structure-preserving algorithms [53].

Proposition 1 (Uncertainty Bound) If all POIs $\mathcal{Q}$ are observed at least once during a cycle of duration T, the eigenvalues of the steady-state Kalman filter covariance $\mathbf{\Sigma}$ will be bounded as:

$$\lim_{t\to\infty} \lambda_{\max}(t) \leq \max_{i} \ \lambda_i \tag{18}$$

where λ_i is the i-th eigenvalue of $\tilde{\mathbf{\Sigma}}_1$ as calculated in Eq. 17 from a modified observation matrix $\tilde{\mathbf{H}}$ formed from the number of consecutive observations d_i of each POI in Eq. 15 and the worst-case interobservation time $\hat{T}$.

The uncertainty bound can be computed for a worst-case inter-observation time and number of observations for each POI. The remainder of Section 4 explores how the cycle time and observation periods are selected to minimize this bound.

4.2 Visitation Order

The cycle time T can be split into two components: T_i, the time spent observing $\mathbf{q}_i$, and $T_{i\to i+1}$, the travel time between successive observation regions B_i and B_{i+1}. In this section, we present a solution for minimizing the interobservation travel time by formulating the problem as a standard TSP.

Modern solutions to the TSP involve minimizing the total cost to visit all points and finding a Hamiltonian tour that only visits each POI once. For our application, the cost between POIs is the time required to travel between them. A state-of-the-art exact TSP solver is Concorde [54], which leverages Lin-Kernighan heuristics, branch-and-bound methods, and other upper- and lower-bound methods to converge to solutions quickly for real-world problems but still with exponential complexity for worst-case situations.

We restrict our trajectory to the class of periodic trajectories that is periodic for position and all higher order derivatives, such that:

$$\mathbf{s}(t+T) = \mathbf{s}(t), \qquad T = \sum_{i=1}^{N} \left(T_i + T_{i\to i+1} \right). \tag{19}$$

Due to being periodic, the trajectory finishes a loop with travel from the last $\mathbf{q}_N$ to the first POI $\mathbf{q}_1$.

For a given visitation order, the travel time between POIs can be fixed by selecting entrance and exit points for the observation region B_i for each POI, which separates the problem into parallel subproblems of generating a trajectory segment within each observation region and the trajectory between consecutive observation regions as illustrated in Fig. 2. Within each B_i, the trajectory segment $\mathbf{s}_i(t)$ begins at an entrance point $\mathbf{s}_i^{in} = \mathbf{s}_i(0)$ and ends at an exit point $\mathbf{s}_i^{out} = \mathbf{s}_i(T_i)$, both of which are located within the closed observation region $\mathbf{s}_i^{in}, \mathbf{s}_i^{out} \in B_i$. When the UAV is not in an observation region, the estimation uncertainty grows linearly with time.

The interpoint travel time between observation regions is minimized by minimizing the distance between consecutive, convex observation regions B_i and B_j while maximizing the trajectory velocity $\|\dot{\mathbf{s}}\| = v_{\max}$ with $\|\ddot{\mathbf{s}}\| = 0$ as follows:

$$\mathbf{s}_j^{in*}, \mathbf{s}_i^{out*} \ = \ \underset{\mathbf{s}_j^{in}\in B_j, \mathbf{s}_i^{out}\in B_i}{\arg\min} \ \frac{\|\mathbf{s}_j^{in} - \mathbf{s}_i^{out}\|}{v_{\max}}, \tag{20}$$

$$T_{i\to j} \ = \ \frac{\|\mathbf{s}_j^{in*} - \mathbf{s}_i^{out*}\|}{v_{\max}}, \tag{21}$$

where $\mathbf{s}_j^{in*}$ and $\mathbf{s}_i^{out*}$ are the optimal entrance and exit positions for observation region B_j and B_i, respectively, $T_{i\to j}$ is the interpoint travel time, and $j = i + 1 \bmod N$ closes the cycle so that the last point leads into the first point in the visitation order. The distance between regions will be unique, but in the case where the entrance and exit positions are not unique, the interpoint trajectory closest to the direct path from one point to the next is used. The interpoint travel times $T_{i\to j}$ are used in calculation of the total loop time and are input into the Greedy Knockdown Algorithm in Section 4.3.

The fixed entrance and exit positions for each observation region enable the trajectory $\mathbf{s}$ to be decomposed into trajectories that are within an observation region $\mathbf{s}_i$ and interpoint trajectories that are between observation regions $\mathbf{s}_{i\to j}$, such that:

$$\mathbf{s} = \mathbf{s}_1 \cup \mathbf{s}_{1\to 2} \cup \mathbf{s}_2 \cup \ldots \cup \mathbf{s}_N \cup \mathbf{s}_{N\to 1}. \tag{22}$$

The interpoint trajectories are fully defined as the linear interpolation between $\mathbf{s}_i^{out*}$ and $\mathbf{s}_j^{in*}$ with $\|\dot{\mathbf{s}}\| = v_{\max}$ and $\|\ddot{\mathbf{s}}\| = 0$. Continuity is maintained between the trajectory segments by constraining the trajectory within an observation region to begin at the entrance position and end at the exit position with 0 acceleration and velocities aligned with the incoming and outgoing trajectories. By constraining the trajectory at each entrance and exit position, trajectories within each observation region can be calculated independent of each other by the optimization outlined in Section 5.2.

Although we generate a visitation order by solving a TSP, the interpoint travel times and entrance and exit positions can be calculated for any arbitrary visitation order of POIs. However, the visitation order has a direct impact on the observation periods calculated in Section 4.3 and the trajectories generated in Section 5, so any change to the visitation order would require re-executing the iterative trajectory generation portion of f-TOP.

4.3 Greedy Knockdown Algorithm

Prior to generating the trajectory $\mathbf{s}_i$ through each observation region B_i, the length of time spent observing each POI must be determined. Since we constrict the trajectory to visit each POI once per cycle for one or more consecutive observations, the goal is to determine the number of consecutive observations that will minimize the worst-case steady-state uncertainty. To accomplish this, we extend the Greedy Knockdown Algorithm from our previous work [22] by updating the method with an improved stopping criterion and more efficient computation. We chose to implement a greedy algorithms because they have been shown to be near-optimal for approximate supermodular metrics [52], such as the mean-squared error and spectral norm of the estimation covariance matrix, and optimal when state and sensor noise are uncorrelated [9].

Our proposed solution GKA, shown in Algorithm 1, determines the optimal number of observations of each POI d_i from the interpoint travel times $T_{i\to j}$, the uncertainty in the environmental model $\mathbf{W}$, and the covariance of the noise in the observation model $\mathbf{V}$. The initial run of GKA leverages the assumption that each POI is observed at least once and sets $d_i = 1 \; \forall \; i$. Following runs of GKA calculate the number of initial observations from the minimum traversal time of each subregion, an output of the proposed f-TOP algorithm in Section 5.2.

Algorithm 1 Greedy knockdown algorithm.

```
Input:  $T_{i\to j}$, $\mathbf{d}_0$, $\mathbf{W}$, $\mathbf{V}$
Output: $\mathbf{d}$
 1: $\mathbf{d} = 0^{N_a \times N}$, $T = 0^{N_a \times 1}$, $\tilde{\lambda} = 0^{N_a \times N}$
 2: $\mathbf{d}[0,:] \leftarrow \mathbf{d}_0$, $a \leftarrow 1$
 3: while Stopping Criterion (Prop. 2) not met do
 4:     Form $\tilde{\mathbf{H}}$ from $\mathbf{d}[a-1,:]$ using Eq. 15
 5:     Calculate $\tilde{\mathbf{\Sigma}}$ from Eq. 17 (Section 4.1)
 6:     $\tilde{\lambda}[a,:] \leftarrow$ Eigenvalues($\tilde{\mathbf{\Sigma}}$)
 7:     $c_{\max} \leftarrow \text{argmax}_c \; \tilde{\lambda}[a,c])$
 8:     $\mathbf{d}[a,:] \leftarrow \mathbf{d}[a-1,:]$
 9:     $\mathbf{d}[a,c_{\max}] \leftarrow \mathbf{d}[a,c_{\max}] + 1$
10:     $a \leftarrow a + 1$
11: end while
12: $r_{\min} \leftarrow \text{argmin}_r \; (\max_c \tilde{\lambda}[r,c])$
13: return $\mathbf{d}[r_{\min},:]$
```

GKA is a greedy algorithm that proceeds in iterations denoted by a (Algorithm 1). The algorithm may progress for several iterations beyond the optimum, so the iteration that contains the optimal number of iterations and the final iteration may be different.

For each iteration, a bound on the maximum steady-state uncertainty for each point is calculated by forming $\tilde{\mathbf{H}}$ (Algorithm 1 Line 4) and computing the steady-state uncertainty (Algorithm 1 Line 10) using the techniques outlined in Section 4.1. The bound is the result of solving the nested Kalman filter formulation in Eq. 17 with the special observation matrix $\tilde{\mathbf{H}}$. An observation is added to the POI with the largest estimation uncertainty as denoted by the i-th diagonal element in $\tilde{\mathbf{\Sigma}}$ (Algorithm 1 Lines 6-8). The algorithm continues until termination conditions are met that guarantee the minimum upperbound on the maximum eigenvalue has been found. The last step is to check all iterations and return the optimal number of observations of each POI d_i associated with the minimum upperbound.

Each iteration of GKA adds an observation to the POI with the current maximum estimation uncertainty, and this addition has competing effects. Firstly, the additional sequential observation reduces the estimation uncertainty of the POI being observed. As the number of consecutive observations of a single POI increases, the uncertainty of that POI reduces exponentially and asymptotically to the value that a stationary sensor with the sample sampling characteristics could achieve if left at the POI. Secondly, each observation increases the overall cycle time by f_s^{-1} (i.e. $T_{a+1} = T_a + f_s^{-1}$), resulting in a linear increase in uncertainty of all other POIs. The GKA stopping criterion in Algorithm 1 Line 9 identifies the point when additional observations of a POI can no longer improve the maximum estimation uncertainty (Fig. 5).

Since the reduction in uncertainty decreases for each subsequently added observation, after some number of observations, the reduction from adding an observation will be less than the uncertainty increase when an observation is taken at another point. Once this critical point is reached, GKA is stopped after the maximum uncertainty for any two following iterations is higher than the maximum uncertainty at the latch point (Algorithm 1 Line 9). Algorithm 1 uses the array notation $\tilde{\lambda}[a,:]$ to denote the eigenvalues associated with iteration a of the algorithm generated with the modified observation matrix $\tilde{\mathbf{H}}$. For simplicity in describing the progression of the algorithm and stopping criterion, it is useful to express the eigenvalues for a specific iteration a and consecutive observation d_i in a single term $\tilde{\lambda}_i^{(d_i,a)}$. Formally, the stopping criterion is stated as follows.

Proposition 2 (GKA Stopping Criterion) For the iterative Greedy Knockdown Algorithm, the minimum steady-state uncertainty is reached at an optimal number of observations $\mathbf{d}^*$ and associated total number of observations a^*. The optimal point is reached at or before any two POIs have reached

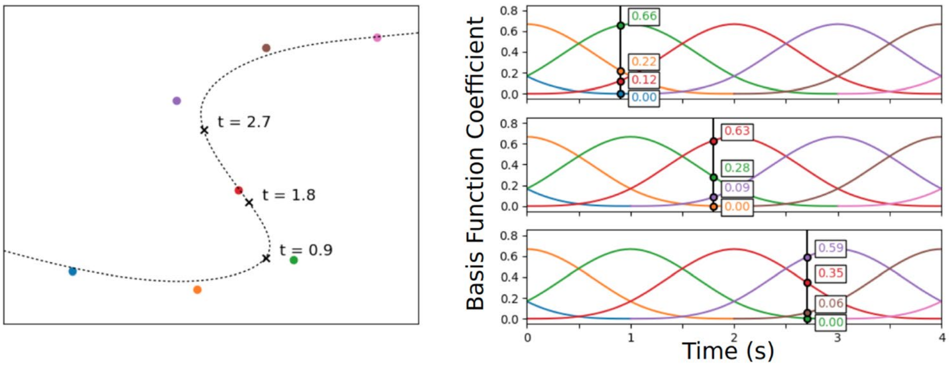

Fig. 5 (left) The control points and resulting trajectory of a B-Spline of order 4 with three different time points marked ($t = 0.9$, $t = 1.8$, and $t = 2.7$). (right) The color-coded basis functions $N_k(t)$ corresponding to each control point. Note that at any time point, the position is calculated from a weighted average of up to 4 control points and the current value of the basis function

the condition where any additional observation will result in an increase in uncertainty for either POI such that:

$$d_i^* = \arg\min_{d_i} \left[\tilde{\lambda}_i^{(d_i,a)} - \tilde{\lambda}_i^{(d_i+1,a+1)} < w_i f_s^{-1} \right] \tag{23}$$

where $\tilde{\lambda}_i^{(d_i,a)}$ is the i-th eigenvalue of the maximum covariance matrix $\tilde{\mathbf{\Sigma}}_1$ for d_i consecutive observations during round a of GKA and $w_i = [\mathbf{W}]_{i,i}$. The stopping point a^* is reached when $d_i \geq d_i^*$ for two or more i. Since the maximum eigenvalue for future observations $a > a^*$ increases monotonically, the minimum upperbound of the maximum eigenvalue has been reached or passed at iteration a^*.

Proof See Appendix A.1

The result of GKA is the number of observations required for each POI to minimize the upperbound on the steady-state uncertainty. Each of the d_i observations requires a period of f_s^{-1}, which serves as constraints for the length of the observation period. In the next section, the trajectory optimization generates a smooth trajectory that ensures that the required observation period while adhering to the dynamic constraints of the UAV.

5 Minimum Uncertainty Trajectories

From the observation lengths calculated by GKA (i.e. $d_i f_s^{-1}$) and physical constraints of the platform, we generate a feasible trajectory that meets these constraints while minimizing the average jerk. As shown in Eq. 22, the whole trajectory can be decomposed into interpoint trajectories and trajectories within an observation region $\mathbf{s}_i$. The interpoint trajectories are fully defined by a linear interpolation between consecutive observation regions with maximum velocity and no acceleration. In this section, we present a method for generating the remaining trajectories within the observation region using B-spline trajectories to ensure adherence to constraints on the dynamics of the system, the required observation length, and trajectory continuity at each transition between trajectories within and between observation regions. First, we present a brief background on B-spline trajectories and then we present our proposed f-TOP algorithm for generating feasible trajectories within each observation region.

5.1 B-Spline Trajectory

To generate $\mathcal{C}^3$ continuous trajectories that meet constraints on velocity and acceleration, we leverage a B-spline representation. B-spline curves have an attractive property that the curve and its derivatives are contained within the convex hull of their control points and offer controllable levels of continuity and numerical stability due to their limited, time-varying support.

B-splines are piecewise polynomials defined for each subregion i by a set of control points $\mathbf{C}_i$, k-th order basis functions $\boldsymbol{N}_k(t)$, and knots τ_j. The k-th order basis functions can be calculated using the Cox-de Boor algorithm as follows:

$$\mathbf{s}_i = \mathbf{C}_i \boldsymbol{N}_k(t) \tag{24}$$

$$\left[\boldsymbol{N}_1(t)\right]_j = \begin{cases} 1\,, & \tau_j \leq t < \tau_{j+1} \\ 0\,, & \text{otherwise} \end{cases} \tag{25}$$

$$\left[\boldsymbol{N}_k(t)\right]_j = \frac{t-\tau_j}{\tau_{j+k}-\tau_j}\left[\boldsymbol{N}_{k-1}(t)\right]_j + \frac{\tau_{j+k+1}-t}{\tau_{j+k+1}-\tau_{j+1}}\left[\boldsymbol{N}_{k-1}(t)\right]_{j+1}, \tag{26}$$

where $\left[\boldsymbol{N}_k(t)\right]_j$ is the j-th basis function of $\boldsymbol{N}_k(t)$.

Each basis function provides limited support over the interval $\tau_j \leq t < \tau_{j+k}$. At any given point along the curve, a B-spline function will have a maximum support of $k+1$ different basis functions, transitioning between basis functions at each knot. We focus on developing trajectories with uniform B-splines, which have a uniform spacing between all knots, such that $\Delta\tau = \tau_{j+1} - \tau_j$.

At the transition points defined by the knots τ_k, the curve has C^{k-1} continuity. The derivative of the B-spline bases results in a linear combination of lower order bases, which can be succinctly represented as:

$$\frac{d^a \mathbf{s}(t)}{dt^a} = \left(\frac{1}{\Delta\tau^a}\right)\mathbf{C}_i\mathbf{A}^{(a)}\boldsymbol{N}_{\boldsymbol{k-1}}(t), \tag{27}$$

where $\mathbf{C}_i$ is a matrix of the control points, $\mathbf{A}^{(a)}$ is a Toeplitz matrix that captures the relationship between $\mathbf{C}$ and the control points of the a-th derivative $\mathbf{C}^{(a)}$, and $\boldsymbol{N}_{\boldsymbol{k-1}}(t)$ is vector of the value of each basis function at time t. For the derivation of Eq. 27 see Appendix A.2.

Due to the fact that the basis functions sum to unity $\sum_j N_{j,k}(t) = 1, \forall t$, the value of a function defined by B-splines will remain within the convex hull created by the control points of the support. As shown in Eq. 27, the derivative In the case where a robotic platform has the same maximum allowable velocity and acceleration for the entirety of its trajectory, the constraints are guaranteed to be met if the magnitude of all control points c are less than the limiting value (i.e. $v_{\max}$ or $a_{\max}$ as defined in $\mathcal{S}$).

5.2 Trajectory Optimization with f-TOP

Trajectories that are optimized to be informative must take the dynamic constraints of the sensing platform into account. If a trajectory cannot be followed and observation locations are different from planned, the resulting uncertainty in an estimate can be arbitrarily bad. Our proposed Feasible Trajectory Optimization for Persistent Monitoring (f-TOP) takes the point order and required number of observations for each POI and generates a *feasible* trajectory that guarantees the correct number of observations are taken within each sensing region while meeting dynamic constraints of the UAV.

The proposed f-TOP optimization has a multi-objective formulation, primarily to minimize the maximum steady-state uncertainty as formulated in Eq. 13 and secondarily to provide the smoothest possible path as shown in Eq. 28a. By applying the results from Section 4.2 and the output of GKA from Section 4.3, minimizing the maximum steady-state uncertainty can be simplified to minimizing the total travel time subject to constraints on entrance and exit conditions in Eq. 28c and the minimum number of required observations in Eq. 28e for each observation region.

The secondary objective of a smooth path is formulated as a minimization of the L^2-norm of the jerk of the trajectory with a feasibility constraint that the trajectory remains within the valid state space $\mathcal{S}$ as defined in Eq. 7. By restricting the trajectory $\mathbf{s}$ to $\mathcal{S}$ in Eq. 28b, the maximum velocity and acceleration are bounded by the limits of the platform. As smoothness is a secondary objective, the coefficient ϵ can be set arbitrarily small so that it does not affect the primary goal. The full multi-objective formulation is as follows:

$$\min_{\mathbf{s}} \sum_{i=1}^{N}\left(T_i + T_{i\rightarrow i+1}\right) + \epsilon \int \|\dddot{\mathbf{s}}(t)\|^2 dt \tag{28a}$$

$$\text{s.t.} \quad \mathbf{s} \in \mathcal{S} \tag{28b}$$

$$\mathbf{s}_i^{in} = \mathbf{s}_i^{in*} \quad \mathbf{s}_i^{out} = \mathbf{s}_i^{out*} \qquad \forall\, i \tag{28c}$$

$$\mathbf{s}_i \in B_i \qquad \forall\, i \tag{28d}$$

$$T_i = \int_{\{t|\mathbf{s}(t)\in B_i\}} 1\, dt \geq d_i f_s \quad \forall\, i, 0 \leq t < T \tag{28e}$$

where the trajectory is segmented as described in Eq. 22. Note that since GKA assumes that the UAV remains within the observation region of a POI for a continuous period of time, the additional constraint of $\mathbf{s}_i \in B_i$ is applied.

The optimization and constraints from Eq. 28a can be reformulated using B-splines into a quadratically constrained quadratic program (QCQP). The primary objective is simplified by removing the interpoint travel times $T_{i\rightarrow j}$, which are constant due to the entrance and exit constraints, and the time within each observation region is updated with the number of control points and a fixed, uniform knot spacing such that $T_i = M_i\Delta\tau_i$. The secondary objective of minimum jerk is converted into a linear relationship of control points using Eq. 27, which form a convex hull of achievable values for the trajectory derivatives. For the special case of a 4th-order B-spline with $k = 4$, the objective can be simplified into a quadratic form (see Appendix A.3) as shown below:

$$\min_{\mathbf{C}_i, M_i, \Delta\tau_i} \sum_{i=1}^{N}\left(M_i\Delta\tau_i + \epsilon\mathbf{C}_i^{\intercal}\mathbf{A}^{(3)}\mathbf{A}^{(3)\intercal}\mathbf{C}_i\right) \tag{29a}$$

s.t. Eqs. (28b) and (28c)

$$\mathrm{c}_{ij} \in B_i \qquad \forall\, i, j \tag{29b}$$

$$\|\dot{\mathbf{c}}_{ij}\|_2 \leq v_{\max} \quad \|\ddot{\mathbf{c}}_{ij}\|_2 \leq a_{\max} \qquad \forall\, i, j \tag{29c}$$

$$M_i\Delta\tau_i \geq d_i/f_s \qquad \forall\, i \tag{29d}$$

where $\mathbf{C}_i \in \mathbb{R}^{M_i\times 3}$ is the matrix of all control points and $\mathbf{c}_{i,j}$ is a vector of the j-th control point for the i-th observation region, $\mathbf{A}^{(3)}$ is a third-order Toeplitz matrix capturing the relationship between control points of B-spline derivatives as in Eq. 27, the first and second order control points are computed as $\dot{\mathbf{c}}_{ij} = (\mathbf{c}_{ij} - \mathbf{c}_{i(j-1)})/\Delta\tau_i$ and $\ddot{\mathbf{c}}_{ij} = (\mathbf{c}_{ij} - 3\mathbf{c}_{i(j-1)} + 3\mathbf{c}_{i(j-2)} - \mathbf{c}_{i(j-3)})/\Delta\tau_i^2$, $i \in \{1,...,N\}$ is the POI index, and $j \in \{1,...,M_i\}$ is the interior control point index. Note that M_i only refers to the control points within the observation region. When implementing, $k-1$ additional control points must be added outside of the observation region to define the entrance and exit conditions in Eq. 28c where $\mathbf{s}_i$ is computed from control points and B-spline basis functions as described in Eq. 24.

The objective of the optimization problem in Eq. 29a has a linear and a quadratic component that defines the mean-square of the jerk with quadratic constraints on the magnitude of position, velocity, and acceleration. This form of QCQP can be efficiently solved by modern convex optimization solvers with interior-point methods [55]. While the problem is solvable, the constraints of the problem are restrictive with many values of $\Delta\tau$ resulting in infeasible conditions. The feasibility of the constraints is closely linked to the time that the robotic platform remains within the observation region $M_i\Delta\tau$, the size of the observation region B_i, and the dynamic constraints. Larger M_i values result in more flexibility in the planner in terms of the number of control points, and so long as the robotic platform can remain within the sensing region of a point, which is determined by $v_{\max}$ and $a_{\max}$, and has a sufficient number of control points, then it can meet the enter and exit constraints of the region of interest.

To better optimize the search for a feasible domain, we introduce a bi-level formulation of the optimization problem with slack variables that guide the selection of appropriate M_i and $\Delta\tau$ values. The outer problem controls the number of control points and knot spacing while the inner problem minimizes the slack of the constraints, which is added to provide direct information for how close the solver is to a feasible solution. The inner problem refits the constraints from Eq. 29a with slack variables for the velocity constraint π_v, acceleration constraint π_a, and overall slack π. The outer problem can adjust M_i and $\Delta\tau_i$ with gradient descent methods to better approximate good values for subsequent iterations. The inner problem is defined for a given observation region B_i as a second-order cone program (SOCP) as follows:

$$f(M_i, \Delta\tau_i) = \min_{\mathbf{C}_i} \Pi_i + \epsilon_\pi \pi_i^{(a)} + \epsilon \mathbf{C}_i^\intercal \mathbf{A}^{(3)} \mathbf{A}^{(3)\intercal} \mathbf{C}_i \tag{30a}$$

s.t. Eqs. (28b), (28c), and (29b)

$$(\pi_i^{(v)} v_{\max}, \dot{\mathbf{c}}_{ij}^{(x)}, \dot{\mathbf{c}}_{ij}^{(y)}, \dot{\mathbf{c}}_{ij}^{(z)}) \in \mathcal{Q}^4 \qquad \forall j \tag{30b}$$

$$(\pi_i^{(a)} a_{\max}, \ddot{\mathbf{c}}_{ij}^{(x)}, \ddot{\mathbf{c}}_{ij}^{(y)}, \ddot{\mathbf{c}}_{ij}^{(z)}) \in \mathcal{Q}^4 \qquad \forall j \tag{30c}$$

$$\Pi_i \geq \pi_i^{(v)} \quad \Pi_i \geq \pi_i^{(a)} \tag{30d}$$

where $\dot{\mathbf{c}}_{ij}^{(x)}$, $\dot{\mathbf{c}}_{ij}^{(y)}$, and $\dot{\mathbf{c}}_{ij}^{(z)}$ represent the first, second, and third entry of $\dot{\mathbf{c}}_{ij}$, respectively, π_i is an upperbound on the slack, ϵ_π is an added weight to the acceleration, and the slacked constraints are formulated as a second-order cone. An n-dimensional quadratic cone is defined as:

$$\mathcal{Q}^n = \left\{ \mathbf{x} \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + x_3^2 + \ldots + x_n^2} \right\} \tag{31}$$

and the normalized velocity and acceleration constraints are transformed from Eq. 29b to the following form:

$$\|\dot{\mathbf{c}}_{ij}\|_2 / v_{\max} \leq \pi_v \tag{32}$$

$$\|\ddot{\mathbf{c}}_{ij}\|_2 / a_{\max} \leq \pi_a \tag{33}$$

where π_v and π_a are the slack for the velocity and acceleration constraints, respectively. While it may look difficult to solve a SOCP with NM_i cone constraints, each region of interest is independent due to the entrance and exit constraints. Each independent problem results in $2M_i$ fourth-order cones and a positional constraint to remain within the convex optimization region, which is easily solvable by modern standards.

The final bi-level optimization consists of a higher level controller that sets the knot spacing and number of control points that are guided to a feasible region and a set of independent lower-level problems that depend on knot spacing and number of control points $f(M_i,\Delta\tau_i)$ that can be solved in parallel as follows:

$$\min_{M_i,\Delta\tau_i} \sum_{i=1}^{N} f(M_i, \Delta\tau_i) \tag{34a}$$

$$\text{s.t. } M_i\Delta\tau_i \geq d_i/f_s \quad \forall\, i \tag{34b}$$

where the inner problem $f(\Delta\tau_i,M_i)$ is defined in Eq. 30a.

In the inner problem objective statement in Eq. 30a, the reason for an added focus on minimizing the maximum acceleration is twofold. First, the minimum normalized slack on velocity (π_v) is 1 since the entrance and exit conditions for each observation region have the UAV travelling at $v_{\max}$. In order to further guide good solutions after the slack in velocity has been removed, an additional focus needs to be added to the acceleration slack. Secondly, the maximum acceleration along a trajectory has a large impact on the ability of a UAV to track a trajectory since maximum acceleration is directly linked to the maximum force produced by the motors, which is a limiting factor on performance. The further the acceleration is from the upper limit, the better a UAV can recover from tracking errors.

It is important to note that the minimum sensing time $d_if_s^{-1}$ may differ from the amount of time it takes to traverse the region of interest $\Delta\tau_iM_i$. When they differ (i.e. $\Delta\tau_i\neq\Delta\tau_iM_i$), the GKA from Section 4.3 is rerun with the new minimum traversal times to determine if a more optimal series of observations can be obtained. The iteration continues until convergence, which is achieved when the Greedy Knockdown Algorithm maintains its number of observations for two consecutive cycles.

6 Simulation Results

6.1 Simulation Setup

We evaluate the f-TOP algorithm against three baseline algorithms in a simulated scenario requiring monitoring a set of randomly distributed points over a 50 acre region (450-by-450 m). We assume the environmental model from Eq. 2 is zero mean and distribute the covariance growth **W** in accordance to a Chi-squared distribution. We vary the number of points in the region between 10 and 80, which can represent points that an expert selects as useful for inputting into an environmental model in the air quality example or points that have high probability of event detection from thermal or RGB camera sensors in forest fire detection scenarios.

Each POI can be measured within a 10 m circular observation region that does not overlap with the observation regions of other POIs. The circular region approximates the performance of a UAV that is mounted with a camera on a gimbal system. Despite aggressive maneuvers, the camera can remain pointed at the POI. We use a UAV model based on a commercially available Phantom 4 quadrotor, which has a mass of 1.216 kg, maximum velocity of 20 m/s, maximum acceleration of 19.2 m/s^2, and battery life of 20 min. The maximum acceleration is the maximum magnitude in the X-Y plane, which was calculated using the maximum lift capability and required lift to maintain altitude against the acceleration due to gravity. During simulation, limits were placed on each motor individually. Aggressive maneuvers at the acceleration limit can result in tracking instability when using a realistic controller.

The test configurations include the maximum velocity and acceleration and more conservative configurations in 10% steps for maximum velocities $v_{\max} \in \{10, 12, 14, 16, 18, 20\}$ m/s and two maximum acceleration configurations at 90% and 100% with $a_{\max} \in \{17.3, 19.2\}$. Each test configuration was run for 10 randomly drawn configurations for the expected battery lifetime of the platform (20 min), which resulted in a variable number of average loops, dependent on maximum velocity and number of POIs.

6.2 Comparison Algorithms

We compared f-TOP to three baseline approaches: a naive constant max velocity speed controller (Constant), a velocity controller obtained using the algorithm proposed by [14] with uncertainty decay estimates equal to a first-order approximation of a Kalman filter (Linear), and a velocity controller that leverages the GKA but without feasibility constraints, similar to [22] (GKA 2019).

The Constant method calculates the visitation order by solving a TSP as described in Section 4.2 and then traverses the POIs at maximum velocity. The method assumes linear travel between successive points and does not take observations or dynamic constraints into consideration. These assumptions lead to trajectories with instantaneous velocity changes at each POI, which results in tracking errors due to the dynamic limits of the UAV, and potentially unobserved points at higher maximum velocities or for POIs with small observation regions.

The Linear method is based on the uncertainty models used in [14], which include a linear increase in uncertainty when a POI is not being observed and a linear decrease when it is. Since the steady state uncertainty of the Kalman filter is achieved when, for each loop, the reduction in uncertainty equals the increase in uncertainty for each cycle of the loop, a first-order approximation of the reduction rate of the discrete Kalman filter is the product of the loop decrease and the sampling rate, i.e., $T\mathbf{W}f_s$ where **W** is the uncertainty growth from Eq. 2 and T is the loop time at steady state. A UAV trajectory is obtained using the velocity controller proposed by Smith et al. [14]. While Linear is an improvement over Constant, it still lacks a guarantee that all points of interest will be observed.

The GKA 2019 method calculates a POI visitation order by solving a TSP as described in Section 4.2, and optimizes the velocity profile over the resulting path using [22]. The method attempts to meet the required observation periods by reducing velocity within the regions of interest. The method plans a trajectory that minimizes an upper bound on the steady-state error, but like the Constant and Linear methods, GKA 2019 generates trajectories with instantaneous velocity changes.

6.3 Results

All results for the Constant, Linear, GKA 2019, and f-TOP trajectory planners were captured for the same POI placement and uncertainty growth rates. The minimum upper bound on the steady-state error was approximated by the maximum eigenvalue of the covariance matrix of the estimation over the last loop of each test. All uncertainty results were normalized to the bound on max uncertainty of the f-TOP algorithm.

Linear, GKA 2019, and f-TOP guarantee that each POI will be observed at least once by remaining within the observation region for each POI longer than the sampling period. The Constant planner lacks this guarantee of observability because it travels at maximum velocity. Due to the cyclic nature of the plans, not observing a POI results in an unstable condition whereby the uncertainty for the unobserved POI will grow unboundedly.

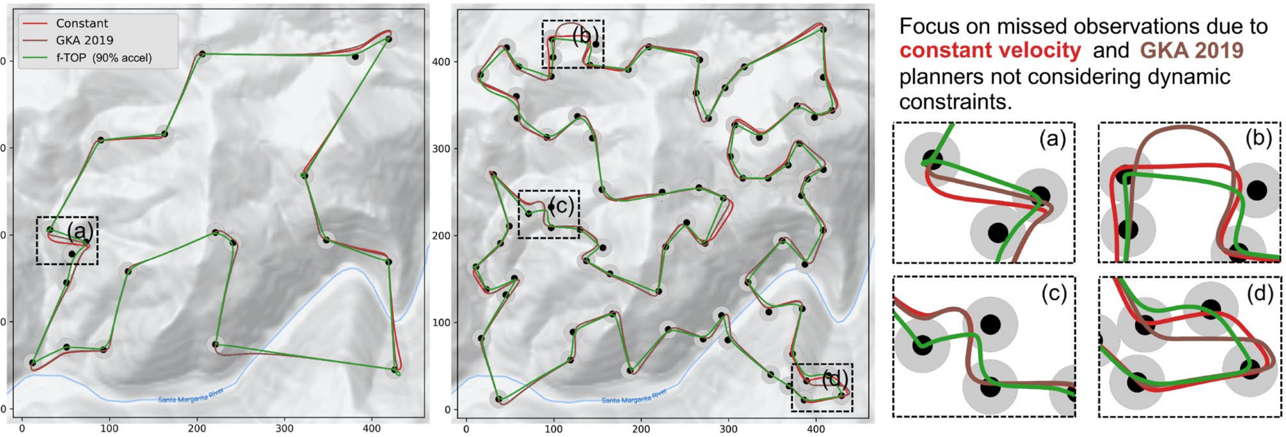

Fig. 6 Simulated trajectories for a quadrotor robot obtained from the constant velocity, GKA 2019, and proposed f-TOP methods. The POIs are shown with their associated sensing regions (light gray). Subplots (a) - (d) focus on regions that highlight the differences in the methods and how the performance of the proposed f-TOP method enables more observations

Despite that the trajectories are optimized to visit all POIs, when the plans are followed by a non-linear, geometric controller [56] that obeys the UAV dynamics, the *actual coverage* can differ from the planned coverage. The Constant, Linear, and GKA 2019 plan trajectories without incorporating acceleration limits, resulting in trajectories that cannot be followed. When POIs were sufficiently far from each other, the UAV controller converged to the planned trajectories during the straight flight segments between observation regions. As the number of POIs increased, resulting in a dense configuration that required high agility, the Constant, Linear, and GKA 2019 trajectories regularly resulted in missing observations of POIs. Examples of trajectories and differences between planned and actual coverage can be seen in Fig. 6, and a quantitative comparison between the planned and simulated trajectories is provided as the Max X-Y Error in Table 1.

Paths planned by f-TOP at the performance limits (i.e. $v_{\max} = 20$ m/s and $a_{\max} = 19.2$ m/s^2) also suffered from tracking errors. This is due to a mismatch between maximum magnitude constraints on the dynamics with which the trajectories are constructed and the actual drone model, which includes limits on the maximum force for each motor. Depending on the maneuver, the required correction to follow the desired trajectory can max out the force ability of an individual motor. For paths further within the performance limits, the proposed f-TOP algorithm generates paths that can be followed more reliably.

The actualized coverage, which is the percentage of POIs that received at least one observation per loop, can be observed in Fig. 7 and Table 1. As the maximum allowable velocity increases, the coverage of the POIs drops. For the plans that do not obey maximum acceleration constraints (i.e. Constant, Linear, GKA 2019), the UAV can only achieve 50% of the maximum velocity before reaching unstable conditions. The plans by f-TOP incorporate maximum acceleration constraints, resulting in plans that are *feasible*. Several variants of plans were generated for maximum acceleration $a_{\max}$ values of 90% and 100% of the maximum listed accelerations.

Despite the lower loop time of the the naive (Constant) and baseline (Linear and GKA 2019) methods, the maximum uncertainty, Max Covar as seen in Table 1, is substantially worse than the result of f-TOP due to the actualized coverage of the planned trajectories. By incorporating maximum acceleration constraints and planning feasible paths, the robotic platforms are able travel 47% faster while maintaining the same coverage. The higher achievable speed results in lower average loop times for a given coverage level and a significant improvement in maximum uncertainty of the measurement at each POI. Note that the Max Covar is normalized to the minimum eigenvalue from any method for a given trial, such that a value of 0.0 means that the method had the minimum uncertainty for a single trial.

The best performing method was the f-TOP algorithm with acceleration limited to 90% of the maximum value of the platform. Despite the slightly improved loop times of f-TOP with higher accelerations, the trajectories planned to the limit of the system resulted in errors due to the imprecise tracking of the controller, which updated the trajectory plan online at 100 Hz, causing compounding error when operating at the system limited in complex maneuvers. By limiting the maximum acceleration and velocity to 90% of the maximum value of the platform, the controller was better able to control the system and achieve the required observations in a minimum time.

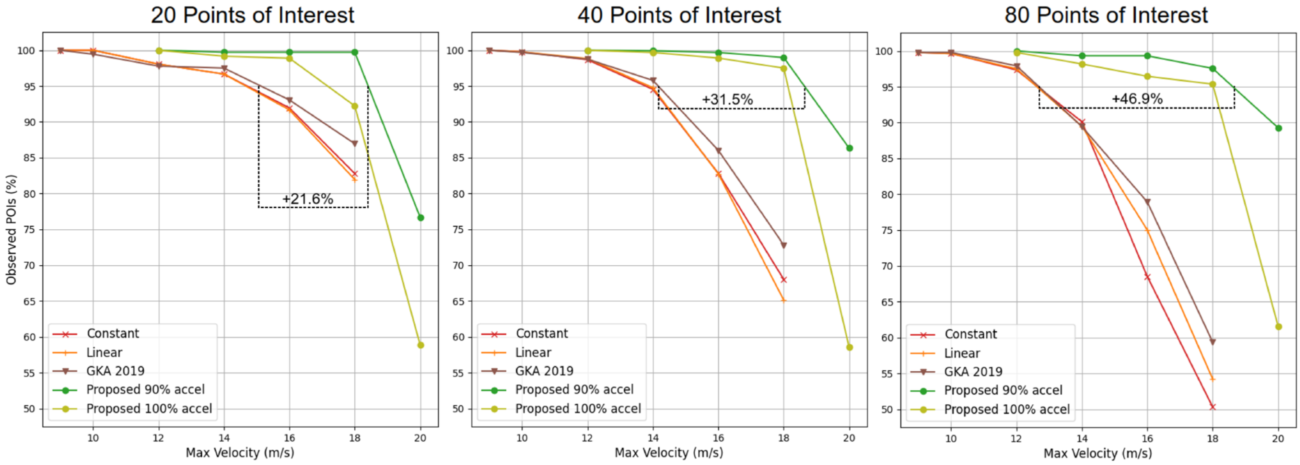

Fig. 7 Comparison of methods showing the impact of not planning with higher-order dynamic constraints. Increasing maximum velocity constraint results in naive planners creating infeasible trajectories that result in missing observations. As a region increases in numbers of POIs, the maneuvers to measure all points become more aggressive, resulting in naive methods requiring significantly lower speeds to achieve similar average observability. Results generated are average of 10 random trials for each condition

Table 1 Statistics for each simulated method for 10 randomized distributions of 80 POIs with random uncertainty growth rates. A graphical summary of important results can be seen in Fig. 7 (right). Max Covar is the maximum eigenvalue of estimation covariance normalized to the smallest maximum eigenvalue from any algorithm for the same trial. Max X-Y Error is the maximum distance between the planned and simulated trajectory for any time point in the given trial. Coverage is the percentage of POIs with at least one observation

Method	Constraints		Max Covar (%)			Max X-Y Error (m)			Coverage (%)			Loop Time (s)
	v_{max}	a_{max}	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
Constant	10	–	79.7	36.8	379.5	21.5	14.3	53.0	99.6	97.5	100.0	327.6
	12	–	169.1	20.7	449.9	29.5	21.8	52.4	97.3	90.0	100.0	273.0
	14	–	308.0	175.0	419.4	45.9	28.6	> 100	90.1	82.5	97.5	234.0
	16	–	400.0	332.6	465.4	**>100**	50.6	> 100	68.5	0.0	82.5	204.7
Linear	10	–	79.7	36.8	379.5	21.5	14.3	53.0	99.6	97.5	100.0	327.6
	12	–	164.9	20.7	449.9	29.0	21.8	52.4	97.5	90.0	100.0	273.0
	14	–	295.7	91.0	419.4	44.5	28.7	82.1	89.5	81.2	96.2	234.0
	16	–	378.8	292.7	465.4	**>100**	50.6	> 100	75.0	61.3	85.0	204.7
GKA 2019	10	–	55.7	38.7	117.8	18.8	14.4	24.5	99.8	97.5	100.0	332.6
	12	–	153.7	22.3	365.8	27.9	20.4	52.7	97.9	93.8	100.0	280.8
	14	–	304.6	116.6	442.8	44.7	26.3	91.2	89.4	80.0	96.2	242.8
	16	–	364.5	268.6	442.8	62.9	36.3	94.1	78.9	71.2	86.2	215.0
f-TOP	12	17.3	29.0	11.3	163.3	7.7	5.8	10.0	100.0	100.0	100.0	266.6
(90% accel)	14	17.3	80.7	0.0	379.5	13.0	8.8	19.9	99.3	96.2	100.0	238.2
	16	17.3	59.0	0.0	312.0	13.3	8.1	21.8	99.3	97.5	100.0	222.1
	18	17.3	131.4	0.0	352.9	23.6	10.3	73.6	97.6	93.8	100.0	222.0
f-TOP	12	19.2	51.1	0.0	377.5	10.3	6.2	16.2	99.8	98.8	100.0	264.4
(100% accel)	14	19.2	128.1	4.4	379.5	66.0	7.1	> 100	98.2	95.0	100.0	235.0
	16	19.2	154.1	3.0	328.8	19.3	12.2	25.6	96.5	92.5	100.0	216.8
	18	19.2	169.7	25.6	343.6	44.3	14.4	> 100	95.4	85.0	100.0	209.3

The bolded entries highlight test configurations where the controller was unable to follow the planned trajectory

7 Conclusion

Persistent monitoring using mobile robotic platforms offers substantial benefits over stationary sensors and low-resolution satellite imagery. When planning trajectories for UAVs, it is critical to optimize for observation periods and to constrain the trajectory to meet the dynamics constraints of the sensing platform to ensure a feasible trajectory. The f-TOP method generates optimal observation periods using the Greedy Knockdown Algorithm and forms continuous, feasible trajectories by optimizing the control points and time knots of a B-spline in a novel bi-level formulation. The resulting trajectories have guarantees on their position, velocity, and acceleration, meeting provided constraints. Simulations for randomized POI distributions show that the trajectories planned using the f-TOP framework outperform related methods, achieving the same level of observability at 47% higher travel speeds, resulting in significant performance gains.

Future work will focus on applying the f-TOP algorithm in physical experiments. For persistent monitoring in the real world, the generated trajectory must involve rendezvous with a recharging station at regular intervals, placing additional requirements for the planned trajectories. Online replanning would be needed to deal with dynamic obstacles or changing objectives. Lastly, the f-TOP algorithm could be extended to include environmental models that capture measurement correlation.

Appendix

A.1 GKA Stopping Criterion (Section 4.3)

For uncorrelated POIs, each additional observation results in a decrease in the eigenvalue of estimation uncertainty for a POI, such that $\lambda_i^{(d_i-1,a)} > \lambda_i^{(d_i,a)}$ where $\lambda_i^{(d_i,a)}$ is the eigenvalue associated with the estimation uncertainty of POI q_i for d_i observations and a total observations. An additional observation of a different point q_j results in an uncertainty increase as $\lambda_i^{(d_i,a+1)} > \lambda_i^{(d_i,a)} + w_i f_s^{-1}$ for $j \neq i$. To see this relationship, note that $\lambda_i^{(d_i,a)} = \rho(\lambda_i^{(d_i,a)}) + a w_i f_s^{-1}$ where $\rho(\cdot)$ is the reduction in uncertainty due to an added observation, a known monotonic function.

When a POI meets the stopping condition $\lambda_i^{(d_i^*,a)} - \lambda_i^{(d_i^*+1,a+1)} < w_i f_s^{-1}$, then it follows that $\lambda_i^{(d_i^*,a)} < \lambda_i^{(d_i^*+1,a+1)} + w_i f_s^{-1} < \lambda_i^{(d^*+1,a+2)}$, which states that when the stopping condition is met, the uncertainty λ_i can only increase after the set of observations at q_i and at any other POI. In GKA, an observation is added to the POI with the highest uncertainty. When any two or more POI meet the stopping condition, then adding an observation to any POI that meets the stopping criterion will result in an overall increase in uncertainty.

A.2 B-Spline Derivatives (Section 5.1)

The derivative for a B-spline is defined as:

$$\dot{s}(t) = \sum_{j=1}^{M_i} \mathbf{c}_{i,j} \frac{d\boldsymbol{N}_k(t)}{dt} \tag{35}$$

$$\left[\frac{d\boldsymbol{N}_k(t)}{dt}\right]_j = k\left(\frac{[\boldsymbol{N}_{k-1}(t)]_j}{\tau_{j+k}-\tau_j} - \frac{[\boldsymbol{N}_{k-1}(t)]_{j+1}}{\tau_{j+k+1}-\tau_{j+1}}\right) \tag{36}$$

For uniform spacing $\Delta\tau = \tau_{j+1} - \tau_j$, the derivative simplifies as:

$$\left[\frac{d\boldsymbol{N}_k(t)}{dt}\right]_j = \frac{1}{\Delta\tau}\left([\boldsymbol{N}_{k-1}(t)]_j - [\boldsymbol{N}_{k-1}(t)]_{j+1}\right) \tag{37}$$

$$\dot{s}(t) = \sum_{j=1}^{M_i} \frac{\mathbf{c}_{i,j}}{\Delta\tau}\left([\boldsymbol{N}_{k-1}(t)]_j - [\boldsymbol{N}_{k-1}(t)]_{j+1}\right) \tag{38}$$

$$= \frac{1}{\Delta\tau}\mathbf{C}_i\mathbf{A}^{(1)}\boldsymbol{N}_{k-1}(t) \tag{39}$$

where $\mathbf{A}^{(1)}$ captures the interaction between basis functions. With $\mathbf{A}^{(a)}$ capturing the linear interaction between basis functions for derivative a, further derivatives take the form:

$$\frac{d^a s(t)}{dt^a} = \left(\frac{1}{\Delta\tau^a}\right)\mathbf{C}_i\mathbf{A}^{(a)}\boldsymbol{N}_{k-1}(t) \tag{40}$$

A.3 B-Spline Objective Reformulation (Section 5.2)

The primary objective from from Eq. 28a can be simplified by noting that $T_{i\to i+1}$ is constant and will not affect the minimization goal and that $T_i = M_i\Delta\tau_i$. The secondary objective, the average jerk power, can be replaced by the linear combination of control points and bases in Eq. 27. The result is an updated cost function as follows:

$$\min_{c,\ \Delta\tau,\ M_i} \sum_{i=1}^{N}\left(M_i\Delta\tau_i + \frac{\epsilon}{\Delta\tau_i^6}\mathbf{C}_i^{\intercal}\mathbf{A}^{(3)}\int N_{k-3}N_{k-3}^{\intercal}dt\ \mathbf{A}^{(3)\intercal}\mathbf{C}_i\right) \tag{41}$$

The secondary objective can be further modified for the specific case of minimizing the square of jerk magnitude for $k = 4$. For a 4rd-order representation of position, the 3rd derivative results in a basis function of order 1, which is either 1 for $\tau_i \leq t < \tau_{i+1}$ and 0 otherwise as shown in Eq. 25. Since the limited domain of the base does not overlap with any other base, the outer product of the 0th-order basis function results in $\mathbf{I}$ and the integral results in $\Delta\tau\mathbf{I}$, where $\mathbf{I}$ is the identity matrix as seen below:

$$\int N_{k-3}N_{k-3}^{\intercal}dt = \int N_0 N_0^{\intercal}dt = \Delta\tau\mathbf{I} \tag{42}$$

Since ϵ is an arbitrarily small scale factor, we incorporate $1/M_i$ and $1/\Delta\tau^5$ into ϵ, resulting in the reformulated objective:

$$\min_{\mathbf{C}_i, M_i, \Delta\tau_i} \sum_{i=1}^{N} \left(M_i \Delta\tau_i + \epsilon \mathbf{C}_i^{\intercal} \mathbf{A}^{(3)} \mathbf{A}^{(3)\intercal} \mathbf{C}_i \right) \tag{43}$$

Author Contributions Conceptualization: Michael H. Ostertag, Nikolay Atanasov; Methodology: Michael H. Ostertag, Nikolay Atanasov; Software, investigation, and analysis: Michael H. Ostertag; Writing - original draft preparation: Michael H. Ostertag; Writing - review and editing: Michael H. Ostertag, Nikolay Atanasov, Tajana Rosing; Supervision: Nikolay Atanasov, Tajana Rosing

Funding We gratefully acknowledge support from the National Science Foundation National Robotics Initiative (CNS-1830399).

Code Availability The custom code used for generating the results found in this paper can be accessed at https://github.com/UCSD-SEELab/persistent-monitoring.

Declarations

Conflict of Interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Alvear, O., Zema, N.R., Natalizio, E., Calafate, C.T.: Using uav-based systems to monitor air pollution in areas with poor accessibility. J. Adv. Transp. 2017. https://doi.org/10.1155/2017/8204353 (2017)
2. Alvear, O., Calafate, C.T., Zema, N.R., Natalizio, E., Hernández-Orallo, E., Cano, J. -C., Manzoni, P.: A discretized approach to air pollution monitoring using uav-based sensing. Mob. Netw. Appl. **23**(6), 1693–1702 (2018). https://doi.org/10.1007/s11036-018-1065-4
3. Allison, R.S., Johnston, J.M., Craig, G., Jennings, S.: Airborne optical and thermal remote sensing for wildfire detection and monitoring. Sensors **16**(8), 1310 (2016). https://doi.org/10.3390/s16081310
4. Bushnaq, O.M., Chaaban, A., Al-Naffouri, T.Y.: The role of uav-iot networks in future wildfire detection. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2021.3077593 (2021)
5. Aslan, Y.E., Korpeoglu, I., Ulusoy, Ö.: A framework for use of wireless sensor networks in forest fire detection and monitoring. Comput. Environ. Urban. Syst. **36**(6), 614–625 (2012). https://doi.org/10.1016/j.compenvurbsys.2012.03.002
6. Le Ny, J., Feron, E., Dahleh, M.A.: Scheduling continuous-time kalman filters. IEEE Trans. Autom. Control **56**(6), 1381–1394 (2011). https://doi.org/10.1109/TAC.2010.2095970
7. Jawaid, S.T., Smith, S.L.: Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. Automatica **61**, 282–288 (2015). https://doi.org/10.1016/j.automatica.2015.08.022
8. Asghar, A.B., Jawaid, S.T., Smith, S.L.: A complete greedy algorithm for infinite-horizon sensor scheduling. Automatica **81**, 335–341 (2017). https://doi.org/10.1016/j.automatica.2017.04.018
9. Zhang, H., Ayoub, R., Sundaram, S.: Sensor selection for kalman filtering of linear dynamical systems: complexity, limitations and greedy algorithms. Automatica **78**, 202–210 (2017). https://doi.org/10.1016/j.automatica.2016.12.025
10. Justice, C., Giglio, L., Korontzi, S., Owens, J., Morisette, J., Roy, D., Descloitres, J., Alleaume, S., Petitcolin, F., Kaufman, Y.: The modis fire products. Remote Sens. Environ. **83**(1-2), 244–262 (2002). https://doi.org/10.1016/S0034-4257(02)00076-7
11. Koltunov, A., Ustin, S.L., Prins, E.M.: On timeliness and accuracy of wildfire detection by the goes wf-abba algorithm over california during the 2006 fire season. Remote Sens. Environ. **127**, 194–209 (2012). https://doi.org/10.1016/j.rse.2012.09.001
12. Schroeder, W., Oliva, P., Giglio, L., Csiszar, I.A.: The new viirs 375 m active fire detection data product: Algorithm description and initial assessment. Remote Sens. Environ. **143**, 85–96 (2014). https://doi.org/10.1016/j.rse.2013.12.008
13. Koltunov, A., Ustin, S.L., Quayle, B., Schwind, B., Ambrosia, V.G., Li, W.: The development and first validation of the goes early fire detection (goes-efd) algorithm. Remote Sens. Environ. **184**, 436–453 (2016). https://doi.org/10.1016/j.rse.2016.07.021
14. Smith, S.L., Schwager, M., Rus, D.: Persistent robotic tasks: Monitoring and sweeping in changing environments. IEEE Trans. Robot. **28**(2), 410–426 (2012). https://doi.org/10.1109/TRO.2011.2174493
15. Cassandras, C.G., Lin, X., Ding, X.: An optimal control approach to the multi-agent persistent monitoring problem. IEEE Trans. Autom. Control **58**(4), 947–961 (2013). https://doi.org/10.1109/TAC.2012.2225539
16. Song, C., Liu, L., Feng, G., Xu, S.: Optimal control for multi-agent persistent monitoring. Automatica **50**(6), 1663–1668 (2014). https://doi.org/10.1016/j.automatica.2014.04.011
17. Atanasov, N., Le Ny, J., Daniilidis, K., Pappas, G.J.: Information Acquisition with Sensing Robots: Algorithms and Error Bounds. In: Robotics and Automation (ICRA), IEEE International Conference On, Pp. 6447–6454. https://doi.org/10.1109/ICRA.2014.6907811 (2014)
18. Lin, X., Cassandras, C.G.: An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces. IEEE Trans. Autom. Control **60**(6), 1659–1664 (2015). https://doi.org/10.1109/TAC.2014.2359712
19. Lan, X., Schwager, M.: Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing

robots. IEEE Trans. Robot. **32**(5), 1230–1244 (2016). https://doi.org/10.1109/TRO.2016.2596772

20. Zhou, N., Cassandras, C.G., Yu, X., Andersson, S.B.: Optimal event-driven multi-agent persistent monitoring with graph-limited mobility. IFAC-PapersOnLine **50**(1), 2181–2186 (2017)
21. Yu, X., Andersson, S.B., Zhou, N., Cassandras, C.G.: Optimal dwell times for persistent monitoring of a finite set of targets. In: American Control Conference (ACC). https://doi.org/10.23919/ACC.2017.7963817, vol. 2017, pp 5544–5549. IEEE (2017)
22. Ostertag, M., Atanasov, N., Rosing, T.: Robust velocity control for minimum steady state uncertainty in persistent monitoring applications. In: 2019 American Control Conference (ACC). https://doi.org/10.23919/ACC.2019.8814376, pp 2501–2508. IEEE (2019)
23. Pinto, S.C., Andersson, S.B., Hendrickx, J.M., Cassandras, C.G.: Optimal periodic multi-agent persistent monitoring of a finite set of targets with uncertain states. In: 2020 American Control Conference (ACC). https://doi.org/10.23919/ACC45564.2020.9147376, pp 5207–5212. IEEE (2020)
24. Metia, S., Oduro, S., Sinha, A.: Pollutant profile estimation using unscented Kalman filter. In: Advances in Control, Signal Processing and Energy Systems. https://doi.org/10.1007/978-981-32-9346-5_2, pp 17–28. Springer (2020)
25. Garg, S., Ayanian, N.: Persistent monitoring of stochastic spatio-temporal phenomena with a small team of robots. In: Robotics: Science and Systems (RSS) Conference. https://doi.org/10.15607/RSS.2014.X.038, vol. 2014. Robotics: Science and Systems (2014)
26. Zhao, S., Ma, Y., Huang, B.: Probabilistic monitoring of sensors in state-space with variational bayesian inference. IEEE Trans. Ind. Electron. **66**(3), 2154–2163 (2018). https://doi.org/10.1109/TIE.2018.2838088
27. Kiefer, J.: Optimum experimental designs. Journal of the Royal Statistical Society: Series B (Methodological) **21**(2), 272–304 (1959). https://doi.org/10.1111/j.2517-6161.1959.tb00338.x
28. Yang, C., Kaplan, L., Blasch, E.: Performance measures of covariance and information matrices in resource management for target state estimation. IEEE Trans. Aerosp. Electron. Syst. **48**(3), 2594–2613 (2012). https://doi.org/10.1109/TAES.2012.6237611
29. Zhao, L., Zhang, W., Hu, J., Abate, A., Tomlin, C.J.: On the optimal solutions of the infinite-horizon linear sensor scheduling problem. IEEE Trans. Autom. Control **59**(10), 2825–2830 (2014). https://doi.org/10.1109/TAC.2014.2314222
30. Hari, S.K.K., Rathinam, S., Darbha, S., Kalyanam, K., Manyam, S.G., Casbeer, D.: Bounding algorithms for persistent monitoring of targets using unmanned vehicles. In: 2019 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 615–621. https://doi.org/10.1109/ICUAS.2019.8798134 (2019)
31. Hari, S.K.K., Rathinam, S., Darbha, S., Kalyanam, K., Manyam, S.G., Casbeer, D.: Persistent monitoring of dynamically changing environments using an unmanned vehicle (2018)
32. Joshi, S., Boyd, S.: Sensor selection via convex optimization. IEEE Trans. Signal Process. **57** (2), 451–462 (2008). https://doi.org/10.1109/TSP.2008.2007095
33. Tang, L., Shao, G.: Drone remote sensing for forestry research and practices. J. Forestry Res. **26**(4), 791–797 (2015). https://doi.org/10.1007/s11676-015-0088-y
34. Portugal, D., Pippin, C., Rocha, R.P., Christensen, H.: Finding optimal routes for multi-robot patrolling in generic graphs. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 363–369. https://doi.org/10.1109/IROS.2014.6942585 (2014)
35. Kant, K., Zucker, S.W.: Toward efficient trajectory planning: The path-velocity decomposition. Int. J. Robot. Res. **5**(3), 72–89 (1986). https://doi.org/10.1177/027836498600500304
36. Yu, J., Schwager, M., Rus, D.: Correlated orienteering problem and its application to persistent monitoring tasks. IEEE Trans. Robot. **32**(5), 1106–1118 (2016). https://doi.org/10.1109/TRO.2016.2593450
37. Hollinger, G.A., Sukhatme, G.S.: Sampling-based robotic information gathering algorithms. Int. J. Robot. Res. **33**(9), 1271–1287 (2014). https://doi.org/10.1177/0278364914533443
38. Ghaffari Jadidi, M., Valls Miro, J., Dissanayake, G.: Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. Int. J. Robot. Res. **38**(6), 658–685 (2019). https://doi.org/10.1177/0278364919844575
39. Atanasov, N., Le Ny, J., Daniilidis, K., Pappas, G.J.: Decentralized active information acquisition: Theory and application to multi-robot slam. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/ICRA.2015.7139863, pp 4775–4782. IEEE (2015)
40. Mueller, M.W., Hehn, M., D'Andrea, R.: A computationally efficient motion primitive for quadrocopter trajectory generation. IEEE Trans. Robot. **31**(6), 1294–1310 (2015). https://doi.org/10.1109/TRO.2015.2479878
41. Liu, S., Atanasov, N., Mohta, K., Kumar, V.: Search-based motion planning for quadrotors using linear quadratic minimum time control. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/IROS.2017.8206119, pp 2872–2879. IEEE (2017)
42. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation. https://doi.org/10.1109/ICRA.2011.5980409, pp 2520–2525. IEEE (2011)
43. Tang, M., Tong, R., Wang, Z., Manocha, D.: Fast and Exact Continuous Collision Detection with Bernstein Sign Classification. ACM Transactions on Graphics **33**(6). https://doi.org/10.1145/2661229.2661237 (2014)
44. Richter, C., Bry, A., Roy, N.: Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In: Inaba, M., Corke, P. (eds.) Robotics Research: The 16th International Symposium ISRR. Springer Tracts in Advanced Robotics. https://doi.org/10.1007/978-3-319-28872-7_37, pp 649–666. Springer (2016)
45. Tang, S., Kumar, V.: Autonomous flight. Annual Review of Control Robotics, and Autonomous Systems **1**(1), 29–52 (2018). https://doi.org/10.1146/annurev-control-060117-105149
46. Liu, S., Watterson, M., Mohta, K., Sun, K., Bhattacharya, S., Taylor, C.J., Kumar, V.: Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments. IEEE Robotics and Automation Letters **2**(3), 1688–1695 (2017). https://doi.org/10.1109/LRA.2017.2663526
47. Liu, S., Mohta, K., Atanasov, N., Kumar, V.: Search-Based motion planning for aggressive flight in SE(3). IEEE Robot. Autom. Lett. **3**(3), 2439–2446 (2018). https://doi.org/10.1109/LRA.2018.2795654
48. Van Nieuwstadt, M.J., Murray, R.M.: Real-time trajectory generation for differentially flat systems. International Journal of Robust and Nonlinear Control **8**(11), 995–1020 (1998). https://doi.org/10.1002/(SICI)1099-1239(199809)8:11%3C995::AID-RNC373%3E3.0.CO;2-W
49. Ding, W., Gao, W., Wang, K., Shen, S.: An Efficient B-spline-Based Kinodynamic Replanning Framework for Quadrotors. https://doi.org/10.1109/TRO.2019.2926390 (2019)
50. Zhou, B., Gao, F., Wang, L., Liu, C., Shen, S.: Robust and efficient quadrotor trajectory generation for fast autonomous flight. IEEE

Robotics and Automation Letters **4**(4), 3529–3536 (2019). https://doi.org/10.1109/LRA.2019.2927938

51. Tzoumas, V., Jadbabaie, A., Pappas, G.J.: Sensor placement for optimal kalman filtering: Fundamental limits, submodularity, and algorithms. In: 2016 American Control Conference (ACC). https://doi.org/10.1109/ACC.2016.7524914, pp 191–196. IEEE (2016)
52. Chamon, L.F., Pappas, G.J., Ribeiro, A.: Approximate supermodularity of kalman filter sensor selection. IEEE Trans. Autom. Control **66**(1), 49–63 (2020). https://doi.org/10.1109/TAC.2020.2973774
53. Chu, E.-W., Fan, H.-Y., Lin, W.-W.: A structure-preserving doubling algorithm for continuous-time algebraic riccati equations. Linear Algebra and Its Applications **396**, 55–80 (2005). https://doi.org/10.1016/j.laa.2004.10.010
54. Applegate, D., Bixby, R., Chvatal, V., Cook, W.: Concorde TSP solver (2006)
55. MOSEK ApS: MOSEK Optimizer API for Python 9.3.7. https://docs.mosek.com/latest/pythonapi/index.html(2019)
56. Lee, T., Leok, M., McClamroch, N.H.: Geometric tracking control of a quadrotor uav on se (3). In: 49th IEEE Conference on Decision and Control (CDC). https://doi.org/10.1109/CDC.2010.5717652, pp 5420–5425. IEEE (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Michael H. Ostertag is a PhD candidate at UC San Diego in the Department of Electrical and Computer Engineering, advised by Prof. Tajana Simunic Rosing. After receiving B.S./M.S. degrees from the Rochester Institute of Technology, he worked at BlackBox Biometrics developing technology to help detect mild traumatic brain injury in soldiers and athletes. His research interests include environmental sensing using distributed sensors and autonomous robotic platforms.

Nikolay Atanasov is an Assistant Professor of Electrical and Computer Engineering at the University of California San Diego. He obtained a Ph.D. degree in Electrical and Systems Engineering from the University of Pennsylvania in 2015. His research focuses on robotics, control theory, and machine learning. He works on probabilistic models that unify geometry and semantics in simultaneous localization and mapping (SLAM) and optimal control and reinforcement learning techniques for active perception.

Tajana Šimunić Rosing is a Professor, a holder of the Fratamico Endowed Chair, IEEE & ACM Fellow, and a director of System Energy Efficiency Lab at UCSD. Her research interests are in energy efficient computing, computer architecture, and cyber-physical systems. Prior to coming to UCSD, she was a research scientist at HP Labs while also leading research efforts at Stanford University. She finished her PhD in EE in 2001 at Stanford, concurrently with finishing her Masters in Engineering Management. Prior to her PhD, she worked as a senior design engineer at Altera, now Intel.