Robust Velocity Control for Minimum Steady State Uncertainty in Persistent Monitoring Applications

Michael Ostertag[1], Nikolay Atanasov[1], and Tajana Rosing[2]

Abstract— We present a velocity controller for persistent monitoring applications that minimizes the maximum eigenvalue of the Kalman filter covariance for any initial sensing position and any initial covariance. A set of points of interest in the environment can be measured along a closed static path by an autonomous, mobile robotic sensing platform. We model the environmental phenomenon at the points of interest as a Wiener process that is estimated by a Kalman filter. We propose a Greedy Knockdown Algorithm to determine the optimal number of observations for each point of interest per cycle and formulate the problem as a linear program with a set of robustness constraints. In simulation, the proposed controller is compared to constant velocity and existing first-order velocity controllers in the literature. The proposed method outperforms existing methods across test cases with a range of different parameters: number of points of interest, noise level of the observation model, and maximum velocity.

I. INTRODUCTION

The objective of persistent monitoring is to continually measure a dyanmic environment to provide accurate estimates. These estimates of phenomena can be used in models to predict future conditions (e.g. monitoring windspeed, temperature, and humidity to predict future weather) or predict current conditions at unobserved locations (e.g. interpolating air pollution measurements in regions between stationary measurement sites). Many environmental phenomena can be predicted using models with additive Gaussian noise or as Wiener processes. For linear observation and state transition models with Gaussian noise, the Kalman filter is an optimal solution for state estimation. Many works in informative path planning utilize Gaussian process regression, but the Kalman filter implementation provides a known framework for efficient updates from a large number of observations, each over a limited subset of states.

Traditional approaches to persistent monitoring have utilized fixed sensors to maximize gathered information, leveraging Kalman filters when estimating phenomena or processes obscured by Gaussian noise. Le Ny et al. [1] have proposed policies for open-loop periodic sensor scheduling and formulate a bound for linear Gaussian systems using continuous-time Kalman filters. Greedy algorithms work well in many instances [2], [3] and have even been shown to be optimal for specific cases of uncorrelated noise [4]. but since environmental phenomena are highly temporally and spatially correlated, static deployments provide less information than a comparable sensor on mobile robotic sensing platforms.

In recent years, researchers have focused on developing trajectories for persistent monitoring of phenomena and targets thanks to increased processing capabilities and lower costs of mobile robotic sensing platforms, which can carry sensors to measure more interesting and useful locations. Approximate controllers with bounded error can be calculated [5], but finding optimal solutions is NP-hard that requires solving for both an optimal path and velocity. Researchers have shown that decomposing trajectory generation into separate path planning and velocity control problems allows for developing optimal solutions for constrained problems with reduced complexity [6].

Smith et al. [7] proposed the first optimal velocity controller for a linear, continuous accumulation model, formatting the optimization problem as a linear program. Cassandras et al. [8] expanded the use of an optimal velocity controller to a different objective and new accumulation model that clears in linear proportion to the distance of nearby robotic platforms. The work was later expanded into two-dimensional space using the concept of infinitesimal perturbation analysis, but with limited guarantees of optimality [9], [10]. Song et al. [11] extended the results to symmetric, non-linear accumulation models and proposed a method to determine optimal initial placements.

Yu et al. [12] utilized a similar set of models but proved that optimal accumulations can be achieved by using dwell times as constraints. The dwell times are calculated as the time the sensing platform must be within the sensing region of a point of interest until its accumulation reaches 0. Along a set path, the results closely mirror those of Smith et al. [7], but for certain situations of homogeneous targets, the dwell times in combination with solutions to the traveling salesman problem can result in optimal trajectories.

Rapidly exploring random cycles [13] have been used for sampling-based path formation and minimizing infinite horizon cycle costs. The work is similar to the presented methods whereby a Kalman filter is used for estimation of a target phenomenon, and the objective is to minimize the largest eigenvalue of the covariance of the estimate.

Our work extends the existing research by developing a controller for autonomous, mobile robotic sensing platforms that results in an optimal minimum bound on the maximum eigenvalue of the Kalman filter covariance for an unknown initial sampling position. We compare our method to controllers that approximate Kalman filtering estimation

[1]The authors are with the Electrical and Computer Engineering Department, University of California, San Diego, La Jolla, CA 92093, USA {mosterta, natanasov}@ucsd.edu
[2]The author is with the Computer Science Engineering Department, University of California, San Diego, La Jolla, CA 92093, USA tajana@ucsd.edu
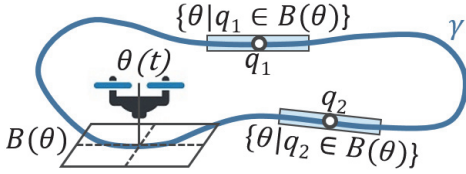
Fig. 1. Persistent monitoring example of 2 points of interest (q_1 and q_2) by a drone at position $\theta(t)$ with sensing area $B(\theta)$.

Fig. 2. Riccati update procedure for a discrete Kalman filter.

with linear accumulation models and show that our proposed approach is robust to initial sampling position.

II. PROBLEM FORMULATION

Consider a compact environment $\mathcal{E} \in \mathbb{R}^d$ that contains a set of N points of interest for an environmental phenomenon $\boldsymbol{q} = [q_1, ..., q_N]^T$ where $q_i \in \mathcal{E}$. These points of interest lie on a stationary closed curve contained within the environment $\gamma : [0, L] \to \mathbb{R}^d$, where L is the perimeter length and $\gamma(0) = \gamma(L)$. The points are sensing locations at which an autonomous robotic platform can measure values $\boldsymbol{x}$ of the phenomenon, such as the intensity of a fire at a particular location or the concentration of a pollutant.

We assume that the environmental phenomenon is time varying and the value of the phenomenon $\boldsymbol{x} \in \mathbb{R}^N$ evolves as a Wiener process. Wiener processes increase with independent, Gaussian increments, for which we define the mean and covariance with a model of the environment $\mathcal{M}(\boldsymbol{q})$. A Gaussian distribution is a natural choice for environmental models, such as those used for the spread of wildfires, since the evolution between states is typically the summation of a large number of smaller physical interactions for which detailed input data cannot reasonably be obtained. For a given timestep, τ the environmental model evolves as follows:

$$\boldsymbol{x}(t + \tau) = \boldsymbol{x}(t) + \mathcal{M}(\boldsymbol{q}) \quad \mathcal{M}(\boldsymbol{q}) \sim \mathcal{N}(\boldsymbol{\mu}, \tau \boldsymbol{W}) \quad (1)$$

where the independent increments are drawn from a Gaussian distribution with mean $\boldsymbol{\mu}$ and full-rank covariance $\tau \boldsymbol{W}$ that is directly proportional to the step size of the increment τ.

To simplify the description of the robotic platform along the path, the curve γ is parameterized by arc-length $\theta \in [0, L]$, and the position of the robotic platform at time t is defined as $\gamma(\theta(t))$ where $\theta : \mathbb{R}_{\geq 0} \to \mathcal{E}$ and $\dot{\theta}(t) = v(t)$, where $v(t)$ is the velocity at time t and $v(t) \in (v_{min}, v_{max}]$. The minimum and maximum velocities can be set to adhere to local regulations, expert advice, or physical limitations of the robotic platform.

The robotic platform has a set sampling rate f_s, and at every sampling time point, any point of interest q_i within the finite sensing footprint $B(\theta) \subset \mathcal{E}$ is sensed. A binary observation model is used with the observation $\boldsymbol{y}(t)$ corrupted by zero-mean Gaussian noise with covariance matrix $\boldsymbol{V}$ and is expressed as follows:

$$\boldsymbol{y}(t) = \boldsymbol{H}(\theta(t))\boldsymbol{x}(t) + \boldsymbol{n}_{obs} \quad \boldsymbol{n}_{obs} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}) \quad (2)$$

$$h_{i,j} = \begin{cases} 1 & i = j \text{ and } q_i \in B(\theta(t)) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$
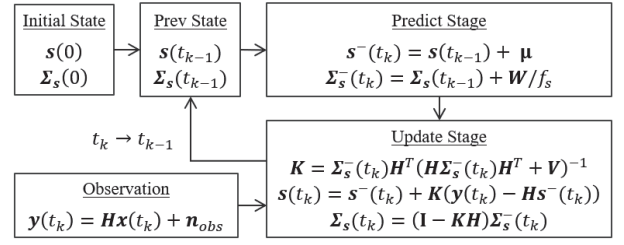
where $\boldsymbol{y}(t)$ is the observation at time t, $\boldsymbol{H}(\theta(t))$ is a binary matrix describing the link between observed and environmental states that depends on the current position of the platform with elements $h_{i,j}$, $\boldsymbol{x}$ is a vector of true values at all points of interest $\boldsymbol{q}$, and $\boldsymbol{n}_{obs}$ is the observation noise drawn from an uncorrelated zero-mean Gaussian distribution with diagonal covariance $\boldsymbol{V}$ and elements V_i along the diagonal. The notation for $\boldsymbol{H}(\theta(t))$ is simplified to $\boldsymbol{H}$ for convenience.

The robotic platform estimates the true value of the phenomenon $\boldsymbol{x}(t)$ by inputting noisy observations $\boldsymbol{y}(t)$ into a Kalman filter, which tracks estimates of the environmental state $\boldsymbol{s}(t)$ with associated covariance $\Sigma_s(t)$. The discrete Riccati equation is used to maintain the states of the Kalman filter with observations taken every $1/f_s$. The iterative flow can be seen in Fig. 2. The Riccati implementation of the Kalman filter provides value for the estimate and uncertainty immediately preceding an update ($\boldsymbol{s}^-$ and Σ_s^-) and following an update ($\boldsymbol{s}$ and Σ_s), but the precise values can be calculated at any point t using the following:

$$\dot{\Sigma}_s(t) = \begin{cases} \boldsymbol{W} + c(\boldsymbol{q}, t_k)\, \delta(t - t_k) \,, & q_i \in B(\theta(t)) \\ \boldsymbol{W} \,, & \text{otherwise} \end{cases} \quad (4)$$

$$c(\boldsymbol{q}, t) = -\Sigma_s(t)\boldsymbol{H}^T(\boldsymbol{H}\Sigma_s(t)\boldsymbol{H}^T + \boldsymbol{V})^{-1}\boldsymbol{H}\Sigma_s(t)$$

where $c(\boldsymbol{q}, t)$ is the result of the Riccati Kalman filter update, $\delta(t - \tau)$ is the Dirac delta function, and $t_k = k/f_s \ \forall k \in \mathbb{Z}_{\geq 0}$. Tracking the covariance matrix in this method is a representation of the reality of a continuous growth model and the discrete capture rate of real systems.

Problem 1 (Minimizing Steady-State Uncertainty). *For a robotic platform continuously monitoring points of interest $\boldsymbol{q}$ along a closed path γ, the objective is to find the optimal velocity controller v^* to minimize the maximum eigenvalue λ_{max}^∞ of the steady-state Kalman filter covariance matrix Σ_s at an infinite-time horizon for any initial sampling position.*

$$v^* = \operatorname*{arg\,min}_{v : \mathbb{R}_{\geq 0} \to \mathbb{R}} \left(\lambda_{max}^\infty \right) \quad (5)$$

$$s.t. \quad \dot{\theta}(t) = v(t)$$

$$v_{min} < v \leq v_{max}$$

where v^* is an optimal continuous-time velocity controller, the Kalman filter covariance evolution is defined in Eq. (4), and λ_{max}^∞ is defined as:

$$\lambda_{max}^\infty = \limsup_{t \to \infty} \left[\max_{t \leq \tau \leq t+T} \lambda_{max}(\tau) \right] \quad (6)$$

where λ_{max} is the maximum eigenvalue of Σ_s and T is the time required to complete one cycle of the closed loop γ.

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 point of interest and is commonly referred to as the spectral radius, but researchers in informative path planning and persistent monitoring have used a number of different objective functions. 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 [14].

Since persistent monitoring involves monitoring a phenomenon over long timeframes, we are concerned with the minimization of the objective function at an infinite-time horizon. At an infinite-time horizon with a periodic controller, Σ_s converges at an exponential rate from any initial covariance to a cyclic pattern of values (Sec IV.A in [15]). By optimizing for the steady state cycle of Σ_s, the developed controller may perform suboptimally for initial cycles along the path γ, but it provides two benefits: the controller is robust to any initial assumptions of the phenomenon and will quickly converge to the optimal solution.

The general optimization formulation is complex, requiring solutions for a continuous velocity controller and a hybrid Kalman filter for estimation. In the following section, we show how the general optimization can be simplified into a linear program.

III. TECHNICAL APPROACH

We transform the general optimization problem into a linear program using approximation techniques and properties of the discrete Kalman filter. We also make the assumption that the points of interest have non-overlapping sensing regions, such that if $q_i \in B(\theta(t))$ then $q_j \notin B(\theta(t))$, $\forall \, i \neq j$.

A. Velocity Controller Approximation

In order to make the optimal velocity controller v^* in Eq. (5) more tractable, we focus on a specific class of appropriate controllers termed periodic position-feedback controllers and approximate the controller using basis functions. These approximations allow the optimization problem to be formatted as a linear program instead of a search through continuous space.

Assuming that a controller exists that positions a robotic platform to capture a series of observations that result in a minimum uncertainty, it has been shown in sensor scheduling research that the optimal solution can be arbitrarily closely approximated by a controller with a periodic schedule [15]. The situation in which a robotic platform constrained to a path is simply a constrained version of the sensor scheduling problem with non-zero switching costs.

At an infinite-time horizon, a periodic controller will be able to closely approximate the optimal solution, so we constrained the solution space to periodic position-feedback

controllers, similar to previous works [7], [11]. These velocity controllers depend only on the position of the robotic platform along the closed loop γ and not on time, such that $v : [0, L] \rightarrow \mathbb{R}_{>0}$. Using a position-dependent controller results in a constant time to complete one cycle (T):

$$T = \int_0^L v^{-1}(\theta)d\theta \qquad (7)$$

A continuous function can be approximated by a series of basis functions, reducing the complexity of the optimization with a small drop in accuracy. We approximate the inverse of the velocity controller by a series of weighted basis functions, a technique that has been used in the literature [7], [11]. We chose to use a rectangular basis function that divides θ into J segments with identical arclength. A rectangular basis function allows for precise control of the velocity of a sensing platform over a region where adjusting J can tune the length of individual regions.

$$v^{-1}(\theta) = \sum_{j=0}^{J-1} \alpha_j \beta_j(\theta) \qquad (8)$$

$$\beta_j(\theta) = \begin{cases} 1 \,, & \frac{jL}{J} \geq \theta > \frac{(j+1)L}{J} \\ 0 \,, & \text{otherwise} \end{cases}$$

The functional approximation of v allows optimization over a discrete number of variables, enabling a solution from a linear program. Eq. (8) allows the cycle time T to be calculated from only the variables α_j and known values L and J as follows:

$$T = \int_0^L \sum_{j=0}^{J-1} \alpha_j \beta_j(\theta)d\theta = \frac{L}{J} \sum_{j=0}^{J-1} \alpha_j \qquad (9)$$

where α_j is the inverse velocity along the segment β_j.

B. Loop and Sampling Cycle Timing

For a periodic velocity controller, the position of the robotic platform repeats every T, such that $\theta(t + T) = \theta(t)$, but the sampling locations for consecutive loops will typically differ. Only in limited cases where T is evenly divisible by $1/f_s$ will robotic platform sample at the same locations for each iteration of the loop. In order to ensure that the point of interest has at least d_i observations per loop for *any initial position*, we introduce a robustness constraint.

Lemma 1 (Robustness Constraint). *A robotic sensing platform must be within the sensing range of point q_i for at least d_i/f_s seconds to ensure d_i observations are performed for any initial sampling position.*

$$\int_{\{\theta|q_i \in B(\theta)\}} \sum_{j=0}^{J-1} \alpha_j \, \beta_j(\theta) \, d\theta \geq \frac{d_i}{f_s} \qquad (10)$$

where $\{\theta|q_i \in B(\theta)\}$ describes the sensing region of q_i and the velocity controller is described in Eq. (8).

Proof: For any initial starting position, the robotic platform will capture an observation in the range $[\phi_1 - v_{max}/f_s, \phi_1)$ due to periodic nature of sampling where

$[\phi_i, \phi_2]$ is the min and max θ from the continuous sensing region $\{\theta | q_i \in B(\theta)\}$. For $\theta(0) \in [\phi_1 - v_{max}/f_s, \phi_1)$, the worst possible initial position would be immediately prior to the start of the sensing region at ϕ_1. In order to achieve d_i observations starting at $\phi_1 - \varepsilon$ where ε is very small, the robotic platform must spend a minimum of d_i/f_s within the sensing region as described in Eq. (10). $\qquad\square$

The Robustness Constraint is important for discrete sampling estimation using position-dependent velocity controllers because the velocity controller has no notion of where the sensors are in their sampling cycle, and unless the loop cycle time is evenly divisible by the sampling period, the robotic platform will take observations at different positions within the sampling region for each cycle. If the optimization is performed without the robustness constraint, then the robotic platform may fail to capture an observation of a point of interest, leading to unbounded uncertainty in the worst case.

Lemma 2 (Maximum Velocity). *The robotic platform should always travel at the maximum allowable velocity. When no points of interest are within the sensing region (i.e. $q_i \notin B(\theta)$), the optimal velocity is $v(\theta) = v_{max}$. When points of interest are in the sensing region (i.e. $q_i \in B(\theta)$), the optimal velocity is the maximum velocity that meets the Robustness Constraint in Lemma 1.*

Proof: The objective of the robotic platform is to minimize the maximum eigenvalue of the Kalman filter covariance matrix. The covariance matrix has a linear relationship with W such that $\Sigma_s(t + \tau) = \Sigma_s(t) + \tau W$. Since W is a covariance matrix, all eigenvalues are positive, and the linear combination increases the eigenvalues of Σ_s. The loop cycle time T is inversely proportional to the velocity. Therefore, to minimize the loop cycle time and the growth of the eigenvalues due to the addition of W, the velocity should be at the maximum permitted velocity for segment β_j. For unobserved regions $\{\theta | q_i \notin B(\theta)\} \forall\ i \in \{1, ..., N\}$, α_j should be $1/v_{max}$. For observed regions $\{\theta | q_i \in B(\theta)\}$, α_j should be the minimum allowable (i.e. maximum velocity) that adheres to Lemma 1. $\qquad\square$

On account of the different sampling position for each cycle, the loop cycle time between observations in a given range of interest $[\phi_1, \phi_2]$ may change per loop cycle. In order to calculate a bound on the maximum uncertainty, we need to use the worst-case loop time between the end of an observation series and the start of the next.

Lemma 3 (Worst-Case Interobservation Time). *For a robotic agent moving along closed path γ and sampling all points of interest within its sensing range (i.e. when $q_i \in B(\theta(t))$) with frequency f_s, the worst-case interobservation T_{wc} between observation series of d_i consecutive samples is calculated as follows:*

$$T_{wc} = (\lceil T f_s \rceil + 1 - d_i)/f_s \qquad (11)$$

where $\lceil \cdot \rceil$ is the ceiling function and d_i is the number of consecutive observations of q_i with $d_i \geq 1$.

Proof: The average number of observations per loop is calculated by $T f_s$, the product of the loop cycle time and sampling frequency. Since the number of observations per loop is an integer by definition, some loop iterations have $\lceil T f_s \rceil$ observations and others have $\lfloor T f_s \rfloor$ with the ratio between the two equal to the fractional portion of the average. The length of time spent observing q_i does not change due to the Robustness Constraint in Lemma 1.

The length of time spent in an observation series is calculated as the difference between the first and last observation. For d_i observations, this amounts to $(d_i - 1)/f_s$. The time difference between the last observation of a cycle and the first observation of the next cycle is the difference between the total cycle time $T = \lceil t f_s \rceil / f_s$ and the time spent in an observation series. The result is shown in Eq. (11). $\qquad\square$

As the sampling rate f_s increases, the difference between the average and worst-case loop cycle times decrease. In applications where the sampling rate is low, such as in air quality sampling, the relative alignment of the loop and sampling cycle can have significant implications on the steady-state uncertainty bound.

C. Steady-State Kalman Filter Bounds

The goal of the problem statement (Eq. 5) can be reformulated with the goal of finding optimal velocity controller v^* that results in a minimum bound on the maximum steady-state uncertainty λ_{max}^∞ of the Kalman filter estimate. We begin the development of a bound on the maximum steady-state uncertainty by proving the Kalman filter covariance matrix is bounded and then show how to calculate the bound directly.

The uncertainty of an estimate using a discrete Kalman filter is guaranteed to be bounded if every point of interest is measured at least once per cycle. This result is well documented within the sensor scheduling literature, and it can be extended to our persistent monitoring application.

Lemma 4 (Bounded Uncertainty Guarantee). *If all points of interest q are observed at least once during a cycle of duration T, the steady state Kalman filter estimation covariance Σ_s will be bounded (i.e. $\lim_{t \to \infty} \lambda_{max}(\Sigma_s(t)) < b$).*

Proof: The uncertainty for a given point of interest q_i will grow at a rate consistent with the noise in the environmental model W. For a single cycle, the covariance
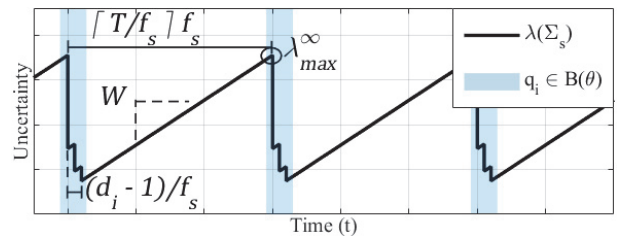

Fig. 3. Example uncertainty of Kalman filter estimate for one point with a cycle time of T, d_i observations, and covariance of the environmental process W. The shaded regions indicate when the point of interest q_i is within sensing range $B(\theta)$. The maximum steady-state value of the uncertainty λ_{max}^∞ immediately precedes the first observation.

matrix grows immediately following the last consecutive observation in a series, such that for an observation taken at t and time between observation T_o, the covariance matrix evolves as:

$$\Sigma_s(t + T) = \Sigma_s(t) + T_o\boldsymbol{W} \qquad (12)$$

Using the Courant-Fischer min-max principle [16], the max eigenvalue of the Kalman filter covariance matrix evolves as follows:

$$\begin{aligned}
\lambda_{max}\big(\Sigma_s(t+T)\big) &= \lambda_{max}\big(\Sigma_s(t) + T_o\boldsymbol{W}\big) \\
&= \max_{x \neq 0} \frac{x^T \lambda_{max}(\Sigma_s(t) + T_o\boldsymbol{W})x}{x^T x} \\
&\leq \max_{x_A \neq 0} \frac{x_A^T \lambda_{max}(\Sigma_s(t))x_A}{x_A^T x_A} + T_o \max_{x_B \neq 0} \frac{x_B \lambda_{max}(\boldsymbol{W})x_B}{x_B^T x_B} \\
&= \lambda_{max}\big(\Sigma_s(t)\big) + T_o\lambda_{max}\big(\boldsymbol{W}\big) \qquad (13)
\end{aligned}$$

where $\lambda_{max}(\cdot)$ represents the maximum eigenvalue of an input matrix. The principle holds because both Σ_s and $\boldsymbol{W}$ are positive, semi-definite matrices. The covariance matrix Σ_s grows until an observation occurs, after which, the uncertainty is reduced as seen in Eq. (4).

For binary observation matrices with diagonal covariance $\boldsymbol{V}$, the maximum eigenvalue for the observed point of interest q_i is V_i and the bound will decrease monotonically for consecutive observations of the same point of interest. Since each point is measured at least once per cycle, $\lambda_{max}\big(\Sigma_s(t)\big) \leq \max_i V_i$ for a t immediately after the observation with max V_i. The maximum amount of uncertainty gain between the last observation of a cycle and the first observation of the next cycle can be calculated by setting T_o equal to T_{wc} from Lemma 3 and d_i to $d_{min} = \min_i d_i$, such that the uncertainty bound is calculated as:

$$\lambda_{\max}(\Sigma_s) \leq \max_i \ V_i + \big((\lceil Tf_s \rceil + 1 - d_{min})/f_s\big)\lambda_{\max}(\boldsymbol{W}) \qquad (14)$$

Eq. (14) holds for any time t. □

For a periodic sequence of observations with period T_{per}, the periodic discrete Riccati formulation of the Kalman filter is below:

$$\begin{aligned}
\Sigma_s(k+1) &= \Sigma_s(k) + \boldsymbol{W}/f_s \\
&\quad - \Sigma_s(k)\boldsymbol{H}(k)\big(\boldsymbol{H}(k)^T\Sigma_s(k)\boldsymbol{H}(k) + V\big)^{-1}\boldsymbol{H}(k)^T\Sigma_s(k)
\end{aligned} \qquad (15)$$

$$\Sigma_s(k) = \Sigma_s(k + T_{per}) \qquad \boldsymbol{H}(k) = \boldsymbol{H}(k + T_{per}) \qquad (16)$$

$$\begin{aligned}
\text{where} \ \ T_{per} &= g_{min}T \\
g_{min} &= \operatorname*{argmin}_{g} \bmod(gT, 1/f_s) = 0
\end{aligned}$$

Remark (Uncorrelated Assumption). *If T is divisible by the sampling period $1/f_s$, then the resulting periodic observation matrix $\boldsymbol{H}$ can be determined relatively easily and Σ_s can be solved using methods such as the structure-preserving algorithm proposed by Chu et al. [17]. In the more common case when T is not divisible by $1/f_s$, the observations are at different locations for g_{min} loop cycles, resulting in an observation matrix that can have intercycle differences in the* numbers of samples between sensing regions for neighboring points of interest.

To handle the intercycle differences, we chose to limit the noise of our environmental model to uncorrelated models with a diagonal covariance matrix $\boldsymbol{W}$ with elements W_i and examine the worst case loop scenario. For uncorrelated noise in the observation (V_i) and environmental model (W_i), we can directly calculate the worst-case bound on the uncertainty at steady state. Determining worst-case observation patterns in correlated covariance matrices is a path of ongoing research.

For d_i consecutive observations in a cycle, the worst-case steady state uncertainty $\lambda_{i,\infty}$ (i.e. the uncertainty immediately preceding the first observation) can be solved using the following equations:

$$\begin{aligned}
\lambda_{i,\infty} &= \lambda_{i,1} - W_i/f_s - (\lambda_{i,1})^2(\lambda_{i,1} + V_i)^{-1} \\
\lambda_{i,1} &= \lambda_{i,2} - W_i/f_s - (\lambda_{i,2})^2(\lambda_{i,2} + V_i)^{-1} \\
&\ \ \ ... \\
\lambda_{i,d_i-1} &= \lambda_{i,\infty} - T_{wc}W_i - (\lambda_{i,\infty})^2(\lambda_{i,\infty} + V_i)^{-1} \quad (17)
\end{aligned}$$

where W_i and V_i are the noise in the environmental model and observation model for point q_i, respectively, and T_{wc} is the worst-case interobservation time outlined in Lemma 3.

Solving for $\lambda_{i,\infty}$ using the nesting functions grows exponentially complicated due to the number of terms, but once converted into quadratic form, the solution can be easily computed. The nesting equations can be simplified into a general form offline and then be hardcoded into a device as functions that depend on T, W_i, V_i, and d, which we express as $F_d : \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0} \ \forall \ d \in \{1, ..., \lfloor T/f_s \rfloor\}$ where $F_d(T, W_i, V_i) = \lambda_{i,\infty}$.

D. Greedy Knockdown Algorithm

For multiple points of interest around a path, the optimal controller problem becomes finding the optimal dwell time at each point of interest and moving at maximum velocity when no points are within the sensing radius. We propose to find the dwell times that result in a minimum bound of maximum uncertainty using a greedy algorithm, termed the Greedy Knockdown Algorithm. For sensor scheduling problems, greedy algorithms were proven optimal when state and sensor noise were uncorrelated [4] and periodic controllers over finite loops, such as those formed by a position-dependent velocity controller, can arbitrarily closely approximate the optimal steady state solution [15]. We assume $\boldsymbol{W}$ and $\boldsymbol{V}$ are uncorrelated, and that each point of interest is measured for only a single series of observations for a cycle.

Our proposed solution, the Greedy Knockdown Algorithm, determines the optimal number of observations for uncorrelated points of interest using the location of the points q, the uncertainty in the environmental model $\boldsymbol{W}$, the covariance of the noise in the observation model $\boldsymbol{V}$, the length of the path L, and the maximum velocity v_{max}. For each iteration within the algorithm, the maximum steady state uncertainty for all points is calculated, and an observation is added to the point with the highest uncertainty. The algorithm continues

until a desired depth N_a or the end conditions are met. The algorithm was developed to find the optimal upperbound uncertainty within the Kalman filter covariance matrix for points of interest that have non-overlapping sensing regions and are uncorrelated, but we will later show that the algorithm still performs well for overlapping cases.

Algorithm 1 Greedy Knockdown Algorithm
Input: q, W, V, L, v_{max}, N, N_a
Output: d
 1: $d = 0^{N_a \times N}$, $T = 0^{N_a \times 1}$, $\lambda_\infty = 0^{N_a \times N}$
 2: $d[0,:] = 1$
 3: **for** $a = 1$ to N_{depth} **do**
 4:    Calculate $T[a-1]$ with $d[a-1,:]$ using Eq. (18)
 5:    Calculate $\lambda_\infty[a,:]$ using Es. (17)
 6:    $c\_max = \mathrm{argmax}_c\ \lambda_\infty[a,c]$
 7:    $d[a,:] = d[a-1,:]$
 8:    $d[a,c\_max] = d[a,c\_max] + 1$
 9:    **if** End Condition Met **then**
10:       break
11:    **end if**
12: **end for**
13: $r\_min = \mathrm{argmin}_r\ (\max_c \lambda_\infty[r,c])$
14: **return** $d[r\_min,:]$

The Greedy Knockdown Algorithm is initialized by assuming each point is visited at least once per loop, so the number of consecutive measurements associated with each point of interest i (d_i) at $a = 0$ is set to 1.

For each iteration of the Greedy Knockdown Algorithm, the loop cycle time is calculated using the fact that when the robotic platform is not within measurement range of any point of interest, it is optimal for the platform to be moving at maximum velocity as outlined in Lemma 2.

The loop cycle time is calculated for each iteration of the Greedy Knockdown Algorithm using the following:

$$T_a = \int_{\{\theta|q \notin B(\theta)\}} \frac{d\theta}{V_{max}} + \sum_{i=1}^{N} \max\left(\int_{\{\theta|q_i \in B(\theta)\}} \frac{d\theta}{V_{max}}, \frac{d_i}{f_s}\right) \tag{18}$$

where T_a is the loop cycle time at algorithm depth a, the integrals are over $\{\theta|q \notin B(\theta)\}$ and $\{\theta|q_i \in B(\theta)\}$, which stand for all locations θ that are and are not in range of any point of interest q_i, respectively.

For each iteration, the maximum steady state uncertainty for each point of interest q_i is calculated using Eq. **??**. The point with the highest maximum steady state uncertainty has an additional observation added, incrementing d_i and increasing the total accumulation at all other points of interest. The result is the length of time that a robotic platform must be within range of each point of interest in order to achieve an optimal minimum bound on uncertainty for any initial starting location.

We update Problem 1 by modifying the minimium steady-state uncertainty objective with a maximum velocity objective since by Lemma 2, maximizing the velocity subject to

the Robustness Constraint in Lemma 1 minimizes the steady-state uncertainty. The Robustness Constraint utilizes the number of observations provided by the Greedy Knockdown Algorithm, resulting in the following formulation:

$$\underset{\alpha_0,...,\alpha_{J-1}}{\arg\min} \quad \sum_{j=0}^{J-1} \alpha_j \tag{19}$$

$$\text{s.t.} \quad \int_{\{q_i \in B(\theta)\}} \sum_{j=0}^{J-1} \alpha_j\,\beta_j(\theta)\,d\theta \geq \frac{d_i}{f_s}$$

$$\frac{1}{v_{min}} < \alpha_j \leq \frac{1}{v_{max}}$$

$$\forall\ i \in \{1,...,N\},\ j \in \{0,...,J-1\}$$

The minimization target is α_j, which are the weights of J known basis functions $\beta_j(\theta)$. The program is linear with the number of constraints increasing with the number of points of interest, which can be solved using any standard linear program solver. The solution to Eq. 19 results in a set $\{\alpha_0,...,\alpha_{J-1}\}$ that is used with Eq. 8 to determine the optimal speed controller to achieve the minimum bound of the maximum steady state uncertainty of all points of interest.

IV. SIMULATION RESULTS

For our tests, we used a drone model with an observation window that corresponded to a field of view of $45°$ and $30°$ along the x- and y-axis, respectively, with a constant altitude of 15 m, resulting in a sensing window $B(\theta)$ of 30 m by 17.32 m. The drone followed a simple circle path of length L, such that $\theta \in [0, L]$ with points of interest with non-overlapping sensing regions randomly drawn from a uniform distribution over $[0, L]$ for position and over $[0, 1]$ for W_i. The velocity controller was approximated using $J = 200$ rectangular basis segments.

The tests were run for 600 seconds, and the minimum upperbound on the steady state error was approximated by the maximum eigenvalue of the Kalman filter covariance matrix over the last 150 seconds of each test. For each test, the steady state eigenvalue was calculated with the same points of interest and environmental model for three controllers: constant velocity controller, first-order velocity controller, and proposed velocity controller. The constant velocity controller sets $\alpha_j = 1/v_{max}\ \forall j \in \{0,...,J-1\}$. The first-order controller approximates the discrete Kalman filter updates in the continuous framework proposed by [7] by setting the uncertainty reduction ($c(q_i)$ from [7]) to the following:

$$c(q_i) = T_0 V_i f_s \tag{20}$$

where T_0 is used as an approximation of the loop time at steady state, V_i is from the covariance of the environmental model, and f_s is the sampling rate. This value was selected because at steady state conditions, the increase in uncertainty (V_i) for each cycle of the loop should equal the reduction and the majority of reduction occurs with a single sample.

The maximum eigenvalue of the Kalman filter covariance matrix for each controller was normalized across a trial
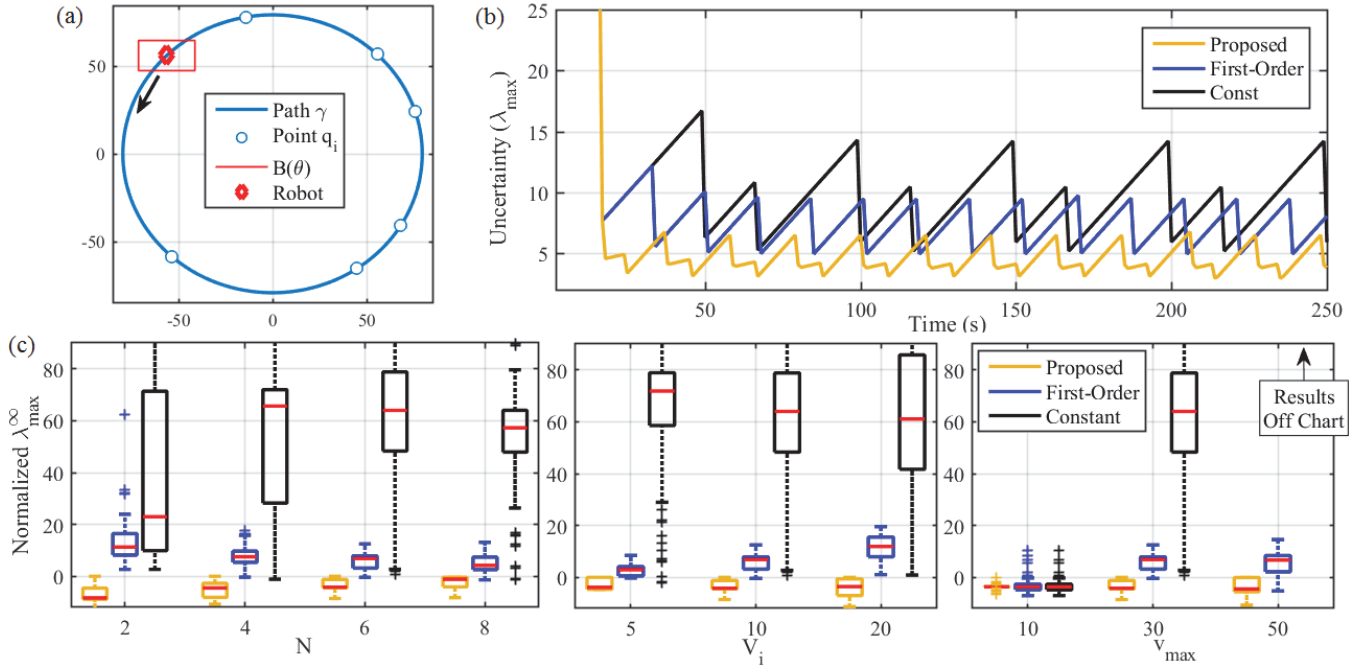
Fig. 4. (a) Example simulation setup for $N = 6$ with the robotic platform traveling in the counter-clockwise direction. (b) Example plot of maximum eigenvalues for the three controllers under test. (c) Result distributions from 1000 runs with box representing 25th and 75th percentile and red line representing median for following test configurations: (c, Left) $N \in \{2, 4, 6, 8\}$, $v_{max} = 30$ m/s, and $V_i = 10$. (c, Middle) $V_i \in \{5, 10, 20\}$, $N = 6$, and $v_{max} = 30$ m/s. (c, Right) $v_{max} \in \{10, 30, 50\}$ m/s, $N = 6$, and $V_i = 10$.

by the maximum covariance bound output by the Greedy Knockdown Algorithm.

$$\text{Normalized } \lambda_{max}^{\infty} = 100\big(\lambda_{max}^{\infty}(\Sigma_s) - \lambda_{wc}^{\infty}\big)/\lambda_{wc}^{\infty} \quad (21)$$

where λ_{wc}^{∞} is the worst-case maximum bound on the eigenvalues of the Kalman filter covariance matrix. A negative value represents maximum eigenvalues that are below the worst-case bound while a positive value represents maximum eigenvalues that are higher than the worst-case bound. Since the bound is a maximum bound on the value, we would expect all results from our proposed algorithm to be negative.

The tests were performed in Python 3.6 using the optimization package PuLP [18] on a laptop with an Intel quad-core CPU at 2.30 GHz with 8.00 GB RAM.

A. Comparison to First Order Approximation

The number of points of interest (N), the maximum velocity (v_{max}), and the noise in the observation model (n_{obs}) were varied one at a time while the other variables were held constant. Each testing configuration was run for 100 randomly drawn trials with each controller tested on each trial. With an L of 500 m and a minimum v_{max} of 10 m/s, the drone has sufficient time to complete multiple passes of the loop, approximating steady state for stable cases. A summary of the results can be seen in Fig. 4.

The first configuration used $v_{max} = 30$ m/s and $V_i = 10$ with $N \in \{2, 4, 6, 8\}$. The number of points of interest was adjusted to examine the effect of the robustness constraint and the first-order approximation of the Kalman filter. The first-order approximation works best when points of interest

only require a single observation. In the case of $N = 2$, the number of observations required for the point with the highest rate of growth is always greater than 1, though, because the optimal strategy is to repeatedly sample the point until the uncertainty of the other point surpasses the first. As the number of points increases, more points of interest can be optimally measured in only one observation and the first-order velocity controller performs better.

The next testing configuration examined the controllers under different levels of observation noise using $V_i \in \{5, 10, 20\}$, $N = 6$, and $v_{max} = 30$ m/s. For a point of reference, the maximum eigenvalue bound from the proposed method for the nominal test case ($N = 6$, $v_{max} = 30$ m/s, $n_{obs} = 10$) is 23.6 (avg) with a range of (13.4, 52.5) over 100 tests. The more noise in the observation model, the more successive observations at a single point can improve the estimate of that point. Observation models that have very low levels of noise would obtain a very good estimate from a single observation, serving as a benefit for the first-order controller.

The last testing configuration examined the controllers at different maximum velocities using $v_{max} \in \{10, 30, 50\}$ m/s, $N = 6$, and $V_i = 10$. At low maximum velocity limits, a constant velocity controller set to max velocity works comparably to other algorithms due to the relative size of sensing regions to the distance covered between each observation. Sensing regions at least as large as v_{max}/f_s ensure that at least one observation is captured; lower velocities allow multiple observations per point of interest. As the maximum velocity increases, the controllers need to ensure they observe

each point of interest in order to maintain stability in the form of a Robustness Constraint. As can be seen in Fig. 4 (Right), the Constant Velocity controller has poor performance in all high max velocity test cases.

The Proposed controller achieved a maximum eigenvalue of 4.0% below the worst-case bound on average with a range of 0.0% to 15.1% where 0.0% represents the maximum eigenvalue achieving the worst-case bound. The First-Order velocity controller achieved maximum eigenvalues in the range of -3.0% (best at $N = 6$, $V_i = 10$, $v_{max} = 10$) to 13.0% (worst at $N = 2$, $V_i = 10$, $v_{max} = 30$) above the worst-case bound with 6.0% above on average. The Constant Velocity controller performed poorly in all test cases except where $v_{max} = 10$ where the performance was comparable to the First-Order controller. In all trials, the maximum eigenvalue of the Proposed Controller was below the calculated worst-case bound.

B. Robustness Results

The proposed controller was developed to be robust to initial position where more robust controllers have lower ranges of λ^∞_{max} across different initial sampling locations. To test the robust guarantee, we compared the first-order controller to our proposed controller by examining the range of λ^∞_{max} normalized to the uncertainty bound output from the Greedy Knockdown Algorithm over 10 different initial sampling positions for 500 unique trials. Both controllers had low average ranges (proposed: 0.07%, first-order: 0.39%). The Robustness Constraint is to handle corner cases of slight misalignment, though, which can be seen by examining the worst-case range across all trials (proposed: 2.5%, first-order: 7.1%) and the number of outliers (proposed: 0.8% over 1.0%, first-order: 8.8% over 1.0%). The first-order approximation had 10 times as many outliers as the proposed algorithm. An example of the robustness to initial sampling position can be seen in Table I. The first-order approximation has two different possible max eigenvalues due to alignment of sensing regions, but the proposed algorithm has the same maximum eigenvalue regardless of initial sampling position due to the Robustness Constraint.

V. CONCLUSIONS

We have presented a method for developing a velocity controller with robust optimal bound on the maximum eigenvalue of the Kalman filter covariance with constraints from an iterative Greedy Knockdown Algorithm. The algorithm results in a velocity controller that is robust to initial sampling positions and outperforms existing periodic velocity controllers around closed loops. The optimal bound was only for uncorrelated environmental and observation models, but loose bounds and an initial framework for the correlated case

was presented and is a topic of future work. The work will also be expanded to robust open-loop velocity controllers.

VI. ACKNOWLEDGEMENTS

REFERENCES

[1] J. Le Ny, E. Feron, and M. A. Dahleh, "Scheduling continuous-time kalman filters," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1381–1394, 2011.

[2] S. T. Jawaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.

[3] A. B. Asghar, S. T. Jawaid, and S. L. Smith, "A complete greedy algorithm for infinite-horizon sensor scheduling," *Automatica*, vol. 81, pp. 335–341, 2017.

[4] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202–210, 2017.

[5] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6447–6454.

[6] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The international journal of robotics research*, vol. 5, no. 3, pp. 72–89, 1986.

[7] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.

[8] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2013.

[9] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1659–1664, 2015.

[10] N. Zhou, C. G. Cassandras, X. Yu, and S. B. Andersson, "Optimal event-driven multi-agent persistent monitoring with graph-limited mobility," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2181–2186, 2017.

[11] C. Song, L. Liu, G. Feng, and S. Xu, "Optimal control for multi-agent persistent monitoring," *Automatica*, vol. 50, no. 6, pp. 1663–1668, 2014.

[12] X. Yu, S. B. Andersson, N. Zhou, and C. G. Cassandras, "Optimal dwell times for persistent monitoring of a finite set of targets," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 5544–5549.

[13] X. Lan and M. Schwager, "Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.

[14] C. Yang, L. Kaplan, and E. Blasch, "Performance measures of covariance and information matrices in resource management for target state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2594–2613, 2012.

[15] L. Zhao, W. Zhang, J. Hu, A. Abate, and C. J. Tomlin, "On the optimal solutions of the infinite-horizon linear sensor scheduling problem." *IEEE Trans. Automat. Contr.*, vol. 59, no. 10, pp. 2825–2830, 2014.

[16] D. Hilbert, *Methods of mathematical physics*. CUP Archive, 1955.

[17] E.-W. Chu, H.-Y. Fan, and W.-W. Lin, "A structure-preserving doubling algorithm for continuous-time algebraic riccati equations," *Linear algebra and its applications*, vol. 396, pp. 55–80, 2005.

[18] S. Mitchell, M. OSullivan, and I. Dunning, "Pulp: a linear programming toolkit for python," *The University of Auckland, Auckland, New Zealand, http://www. optimization-online. org/DB_FILE/2011/09/3178. pdf*, 2011.

TABLE I
TRIAL 5 OF THE ROBUSTNESS TEST WITH VARYING INITIAL POSITIONS.

Initial Position	0	6	12	18	24
First-Order	22.9	23.9	23.9	22.9	22.9
Proposed	22.3	22.3	22.3	22.3	22.3