JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 43, No. 4, April 2020

Mutual Information Based Sensor Tasking with Applications to Space Situational Awareness

Nagavenkat Adurthi*
University of Alabama in Huntsville, Huntsville, Alabama 35899
Puneet Singla†
Pennsylvania State University, University Park, Pennsylvania 16802
and
Manoranjan Majji‡
Texas A&M University, College Station, Texas 77843

https://doi.org/10.2514/1.G004399

This paper focuses on the problem of managing or tasking a network of sensors to accurately track a number of objects while using information theoretic sensor performance metrics. The mathematical formulation of optimally tasking a group of sensors using the mutual information as a sensor utility measure is discussed along with the relative merits of maximizing mutual information. The resulting sensor-tasking optimization problem is shown to be combinatorial in nature, for which the computational complexity increases with an increase in the number of objects as well as the number of sensors. Depending upon the number of objects and available sensors, appropriate suboptimal approximations are presented to alleviate the computational complexity of the tasking problem. The submodular property of the mutual information measure is used to provide guarantees on the optimality of different approximations. Numerical simulations involving tracking ground objects with moving unmanned aerial vehicles and tracking resident space objects with ground-based sensors are considered to show the efficacy of the developed methods.

I. Introduction

IN TYPICAL filtering problems, the sensors are considered to be passive while the process is dynamic. But, in many applications, the measurements made by physical sensors are often influenced by ambient conditions, environmental constraints, operating costs, and constraints. Furthermore, most sensors (such as radars, telescopes, range finders, and cameras) are directional in nature. Some sensors, such as temperature and pressure, are also predominantly local in nature and effective to within a small neighborhood of their location. When the dynamic process leaves the field of view (FOV) of the sensor, it becomes ineffective in observing the process. Many of the limited FOV sensors, which are directional- and/or location-based sensors, can often be reoriented to constantly keep the desired objects within their field of view for longer periods of time, thereby significantly improving the tracking performance. For example, in Ref. [1], the authors studied the effectiveness of optimally locating mobile atmospheric sensors in comparison to stationary sensors for tracking the dispersion of a volcanic plume. Here, the mobile sensors are actively positioned to track the evolving volcanic plume cloud, and thereby deliver better measurements for the filter. In space object tracking applications, where directional sensors such as cameras and telescopes are often used, the sensor becomes ineffective in tracking the satellites that leave its FOV and has to be reoriented using the knowledge in the motion of the satellites. In Ref. [2], the authors proposed an optimization problem to compute the right ascension and declination angles for telescopes to efficiently track geosynchronous space objects. In Ref. [3], the authors described the problem of tracking geosynchronous satellites by reorienting the fixed ground based sensors. It was observed that an optimal orienting strategy outperforms simple heuristic scans.

The location, pointing direction, and other operational parameters of the sensors are a function of the state of the unresolved objects that is to be tracked or observed. Dynamical system models, historical state estimates, and measurements of the process can be used to predict the state of the object over time, along with the corresponding state uncertainties. Using these predicted states and their uncertainties, one is then able to optimize sensor parameters (e.g., orientation and position) well ahead in time to make "better" observations. This is because better measurements lead to better estimates of the underlying process. This process can be efficiently automated to track multiple objects and is the very essence of sensor tasking considered in this paper. In particular, by exploiting statistical measures of information theoretic concepts in conjunction with high-fidelity models of the physical and sensor systems, the dynamic optimization of the sensor management process can have robust and physically meaningful solutions.

A basic difficulty in optimizing sensor configurations is the problem of assessing the sensor performance that will result from the sensor decisions before obtaining the sensor measurements. One of the widely used sensor information measure is the Fisher information matrix (FIM) ([4] p. 326). The FIM is conventionally used for its ease in computation in conjunction with the Kalman filter and the extended Kalman filter. The FIM in conjunction with the extended Kalman filter (EKF) was used for sensor tasking applied to satellite tracking [5]. Here, the problem was framed as a binary integer optimization problem that can be solved very efficiently. In Ref. [6], the authors provided an interesting development and computation of information gain for tasking multiple observations; here, the information gain is computed as the mean of the Kullback–Leibler (KL) divergence measure. A similar expression using the concept of mutual information is used in this paper. In addition to divergence measures for information, the authors of Ref. [7] provided an interesting application of space object tracking and custody maintenance using the Dempster–Shafer theory for sensor tasking.

Related to optimal sensor tasking is the problem of optimal sensor placement/selection, which is widely used in target tracking problems [8,9]. For instance, in Ref. [8], optimal sensor placement and motion coordination of the mobile sensor networks are used to address the target tracking problem. This is achieved by maximizing

Presented as Paper 2014-4168 at the 2014 AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA, January 4–7, 2014; received 20 February 2019; revision received 3 August 2019; accepted for publication 4 December 2019; published online 26 February 2020.

*Assistant Professor, Mechanical and Aerospace Engineering.
†Associate Professor, Aerospace Engineering.
‡Assistant Professor, Aerospace Engineering.

the Fisher information matrix. Tharmarasa et al. [9] studied the problem of selecting a small subset of the available sensors in a large network of sensors in order to track multiple targets. A search scheme based on a combination of optimization methods and the FIM is used to perform this task. Similarly, in the recent work of Refs. [10,11], the authors proposed a clustering scheme for sensor networks to improve the localization and tracking efficiency in the multitarget tracking problem. In Ref. [12], under the assumption of Gaussian processes for temperature sensor placement, the authors showed mutual information to be a better choice than entropy for sensor placements.

Single-time-step sensor-tasking methods have been widely studied in the literature [5,13] because they are computationally attractive, whereas multi-time-step sensor-tasking approaches are often computationally prohibitive (Ref. [13] p. 855). In this respect, the main objective of this work is to realize an information theoretic framework to task multiple sensors in order to accurately track multiple objects over multiple time steps. In this work, it has been assumed that the sensors are capable of target resolution and perfect association. Hence, the problem of data association is considered to be solved before sensor tasking. This assumption has been made to address the challenging problem of sensor tasking in isolation, without considering the effects of coupling it with the artifacts of a data association process. With a perfect data association assumption, a statistically consistent receding-horizon-based computational approach is presented to evaluate the joint mutual information for all objects, sensors, and time steps in the given horizon, building upon our work in this area [14–17]. In Ref. [17], the single-time-step sensor-tasking problem is studied using a suboptimal mutual information cost as a sensor utility function. In Ref. [14], a formulation of the problem as a mixed-integer nonlinear programming problem was presented. This paper presented a more comprehensive treatment of the sensor-tasking problem that subsumed the aforementioned formulations as special cases with extensive details in our solution. In particular, the relative merits of using mutual information for sensor tasking over the Fisher information have been demonstrated. Given that the resulting tasking problem leads to a nonlinear NP-hard integer programming problem, different suboptimal approximations to the ideal sensor-tasking problem are presented to alleviate the computational complexity of the original problem. The computational complexity of these approximations is discussed with respect to the number of target objects and available sensors. The mutual information-based sensor performance measure is shown to be submodular and nondecreasing in nature, and these properties are used to provide optimality bounds on the greedy algorithms (see the works of Nemhauser et al. [18] and Chekuri and Pal [19]). Finally, computationally attractive sequential methods are developed to further reduce the computational burden associated with greedy algorithms. The resulting sensor-tasking framework provides different alternatives to solve the multistep sensor-tasking problem, depending upon the number of target objects and sensors available, and is applicable to various ground-, air-, or space-based target tracking problems. Numerical simulations involving the tracking of multiple moving targets [unmanned aerial vehicles (UAVs)] with fixed sensors and tracking multiple space objects with a number of ground stations are considered to validate the developed framework.

The paper is organized as follows: In Sec. II, the formal problem of sensor tasking is introduced and the ideal optimization problem is formalized using a generic sensor performance measure. In Sec. III, the various sensor performance measures are discussed with specific emphasis on the Fisher information matrix and mutual information. Furthermore, computational approaches are developed to evaluate these measures. In Sec. IV, the exact optimization problem from Sec. II is reformulated by introducing decision variables into the mutual information cost, and approximate methods are discussed to solve this optimization problem. In Sec. V, we show that the mutual information computed in Sec. III is submodular and nondecreasing. We then proceed to leverage important results for maximizing submodular functions to find the sensor task that maximizes mutual information cost. Finally, in Sec. VI, numerical simulations are used to illustrate the approaches discussed in this paper.

II. Problem Formulation

Let us consider the problem of tracking N_o number of objects with N_s configurable sensors over a period of N_T time steps. The process dynamics are described as

$$\boldsymbol{x}_{k+1}^{(i)} = \boldsymbol{f}_k^{(i)}\left(\boldsymbol{x}_k^{(i)}\right) + G_k^{(i)}\omega_k^{(i)}, \qquad i = \{1, 2, \ldots, N_o\} \tag{1}$$

where $\boldsymbol{x}_{k+1}^{(i)} \in \mathbb{R}^{n_i}$. The objects can be dynamic or stationary and can have different process models $\boldsymbol{f}_k^{(i)}$ that best describe the evolution of their states. The objects are tracked using observations from an N_s number of sensors with the following measurement models:

$$\boldsymbol{y}_{k+1}^{(i,j)} = \boldsymbol{h}_{k+1}^{(j)}\left(\boldsymbol{x}_{k+1}^{(i)}, \boldsymbol{\theta}_{k+1}^{(j)}\right) + \nu_{k+1}^{(j)}, \qquad i = \{1, 2, \ldots, N_o\}, \quad j = \{1, 2, \ldots, N_s\} \tag{2}$$

where $\boldsymbol{y}_{k+1}^{(i,j)} \in \mathbb{R}^{m_j}$ and denotes the measurement from the ith object made by the jth sensor at time step $k+1$. The sensor model $\boldsymbol{h}_{k+1}^{(j)}$ can represent a wide variety of sensors, ranging from a simple pinhole approximation for a camera or more intricate models for optical or electromagnetic sensors like Radio Detection and Ranging (RADARS). The process noise sequence $\omega_k^{(i)} \sim \mathcal{N}\left(\omega_k^{(i)}: 0, Q_k^{(i)}\right)$ and measurement noise sequence $\nu_{k+1}^{(j)} \sim \mathcal{N}\left(\nu_{k+1}^{(j)}: 0, R_{k+1}^{(j)}\right)$ are assumed to be mutually independent. Here, the notation $\mathcal{N}(\boldsymbol{x}: \boldsymbol{\mu}, P)$ defines a Gaussian probability density function (PDF) for random vector $\boldsymbol{x}$, with mean $\boldsymbol{\mu}$ and covariance matrix P. $\boldsymbol{\theta}_{k+1}^{(j)} \in \mathbb{R}^p$ represents the configuration parameters of the sensor j at time step $k+1$. When the process and sensor models are linear, the system equations in Eqs. (1) and (2) are represented as $\boldsymbol{f}^{(i)}\left(\boldsymbol{x}_k^{(i)}\right) \equiv A_k^{(i)}\boldsymbol{x}_k^{(i)}$ and

$$\boldsymbol{h}_{k+1}^{(j)}\left(\boldsymbol{x}_{k+1}^{(i)}, \boldsymbol{\theta}_{k+1}^{(j)}\right) \equiv H_{k+1}^{(i,j)}\left(\boldsymbol{\theta}_{k+1}^{(j)}\right)\boldsymbol{x}_{k+1}^{(i)}$$

When the extended Kalman filter is used, the nonlinear models in Eqs. (1) and (2) are linearized about the most recent estimate. To this end, we denote the linearized models as

$$A_{k/k}^{(i)} = \left.\frac{\partial \boldsymbol{f}_i\left(\boldsymbol{x}_k^{(i)}\right)}{\partial \boldsymbol{x}_k^{(i)}}\right|_{\boldsymbol{x}_k^{(i)} = \hat{\boldsymbol{x}}_{k/k}^{(i)}}$$

and

$$H_{k+1/k}^{(i,j)}\left(\boldsymbol{\theta}_{k+1}^{(j)}\right) = \left.\frac{\partial \boldsymbol{h}_j\left(\boldsymbol{x}_{k+1}^{(i)}, \boldsymbol{\theta}_{k+1}^{(j)}\right)}{\partial \boldsymbol{x}_{k+1}^{(i)}}\right|_{\boldsymbol{x}_{k+1}^{(i)} = \hat{\boldsymbol{x}}_{k+1/k}^{(i)}}$$

where $\hat{\boldsymbol{x}}_{k/k}^{(i)}$ is the *a posteriori* estimate at time k using measurements up to time k. Similarly, $\hat{\boldsymbol{x}}_{k+1/k}^{(i)}$ is the *a priori* estimate at time $k+1$ using measurements up to time k. The sensors are subjected to constraints (such as limited FOV) at each time step k represented as

$$\mathcal{C}_k^s\left(\boldsymbol{\theta}_k^{(1)}, \boldsymbol{\theta}_k^{(2)}, \ldots, \boldsymbol{\theta}_k^{(N_s)}\right) \leq 0 \tag{3}$$

The inequality in Eq. (3) represents the constraints on the sensor configurations (modes or parameters). The initial conditions for the ith object is modeled by a known PDF, which is generally assumed to be Gaussian with a known mean $\hat{\boldsymbol{x}}_{0/0}^{(i)}$ and covariance $P_{0/0}^{(i)}$. The object states at the kth time step are collectively represented by the vector $\boldsymbol{X}_k = \left[\boldsymbol{x}_k^{(1)}; \boldsymbol{x}_k^{(2)}; \ldots; \boldsymbol{x}_k^{(N_o)}\right]$, whereas the sensor measurements at the same time step are represented as $\boldsymbol{Y}_k = \left[\boldsymbol{y}_k^{(:,1)}; \boldsymbol{y}_k^{(:,2)}; \ldots; \boldsymbol{y}_k^{(:,N_s)}\right]$ with $\boldsymbol{y}_k^{(:,j)} = \left[\boldsymbol{y}_k^{(1,j)}; \boldsymbol{y}_k^{(2,j)}; \ldots; \boldsymbol{y}_k^{(N_o,j)}\right]$, where $\boldsymbol{y}_k^{(i,j)}$ is the measurement output when the jth sensor observes the ith object at time step k. Here, we use the notation $[\boldsymbol{x}; \boldsymbol{y}]$ to represent a vector constructed by vertically concatenating the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. The colon ":" in the superscript or subscript denotes the entire set.

The state for all objects up to time steps k, is represented as $\boldsymbol{X}_{(1:k)}^{(1:N_o)} = \left[\boldsymbol{x}_{(1:k)}^{(1)}; \boldsymbol{x}_{(1:k)}^{(2)}; \ldots, \boldsymbol{x}_{(1:k)}^{(N_o)}\right]$, where $\boldsymbol{x}_{(1:k)}^{(i)} = \left[\boldsymbol{x}_1^{(i)}; \boldsymbol{x}_2^{(i)}; \ldots, \boldsymbol{x}_k^{(i)}\right]$. This grouping process assumes that the problem of associating measurement to objects is already carried out. Similarly the measurements are denoted as $\boldsymbol{Y}_{(1:k)}^{(:,1:N_s)} = \left[\boldsymbol{y}_{(1:k)}^{(:,1)}; \boldsymbol{y}_{(1:k)}^{(:,2)}; \ldots; \boldsymbol{y}_{(1:k)}^{(:,N_s)}\right]$. Here, $\boldsymbol{y}_{(1:k)}^{(:,j)}$ represents all the measurements from the jth sensor up to time k. Furthermore, the measurement can be grouped with respect to the ith object and will be denoted as $\boldsymbol{Y}_{(1:k)}^{(i,:)} = \left[\boldsymbol{y}_1^{(i,:)}; \boldsymbol{y}_2^{(i,:)}; \ldots; \boldsymbol{y}_k^{(i,:)}\right]$, where $\boldsymbol{y}_k^{(i,:)} = \left[\boldsymbol{y}_k^{(i,1)}; \boldsymbol{y}_k^{(i,2)}; \ldots; \boldsymbol{y}_k^{(i,N_s)}\right]$. Additionally, we use the general notation $\boldsymbol{X}_{(B)}^{(A)}$ to represent the vector of target states with indices in set A and time steps in set B. Lastly, $\hat{\boldsymbol{x}}_{k_p/k_q}^{(i)}$ denotes the estimate of the ith object state at time k_p, which is conditioned on the measurements up to time k_q.

To optimize the sensor parameters, one has to determine the performance or effectiveness of the sensors in reducing the uncertainty associated with a group of objects over a finite time horizon N_T, even before the actual measurements are acquired. The optimal sensor-tasking problem assumes the following as input: the models for the objects and sensor, as well as their uncertainties, constraints, and initial conditions. It is then required to generate the optimal configurations for all sensors and for all time steps. To this end, the relevant optimization problem can be informally posed as

$$\max_{\left\{\boldsymbol{\theta}_k^{(j)}\right\}} \; J\left(\boldsymbol{X}_{(1:T)}, \boldsymbol{Y}_{(1:T)}\right) \tag{4}$$

$$\text{subject to:} \begin{cases} \boldsymbol{x}_{k+1}^{(i)} = f_i\left(\boldsymbol{x}_k^{(i)}\right) + G_k^{(i)}\omega_k^{(i)}, \hat{\boldsymbol{x}}_{0/0}^{(i)}, P_{0/0}^{(i)}, & i = 1, 2, \ldots, N_o \\ \boldsymbol{y}_{k+1}^{(i,j)} = h_j\left(\boldsymbol{x}_{k+1}^{(i)}, \boldsymbol{\theta}_{k+1}^{(j)}\right) + \nu_{k+1}^{(j)}, & j = 1, 2, \ldots, N_s \\ \mathcal{C}_k^s\left(\boldsymbol{\theta}_k^{(1)}, \boldsymbol{\theta}_k^{(2)}, \ldots, \boldsymbol{\theta}_k^{(N_s)}\right) \leq 0 & k = 1, 2, \ldots, N_T, N_T \text{ final time step} \end{cases} \tag{5}$$

We refer to this notional optimization problem as the ideal problem. The chosen sensor performance J in Eq. (4) is a function of all the states of the objects and possible measurements. It is to be noted that both the states and measurements are random variables with uncertainties determined by the state dynamics, measurement models, and initial conditions; whereas J is a deterministic value for a given set of sensor parameters $\left\{\boldsymbol{\theta}_k^{(j)}\right\}$ ($j = 1, 2, \ldots, N_s$ and $k = 1, 2, \ldots, N_T$). In the next section, the specific sensor performance cost used is described in detail; subsequently, the corresponding computational aspects are also developed. The sensor performance cost in the optimization problem of Eq. (4) is highly influenced by the filtering method used. In this paper, we focus on sigma-point-based nonlinear filters for their efficiency [20].

III. Sensor Performance

Appropriate performance measures need to be realized in order to achieve meaningful tasking operations. In general, the dynamical system models, measurement models, and filtering/estimation algorithms influence the choice of the information measure. Also, the choice of the appropriate information measure (or sensor performance) is often problem dependent. If one agrees that the PDF represents the state uncertainty, then probabilistic or information theoretic measures can be used to assess sensor performance. In this respect, various sensor performance measures or information measures such as the Fisher information matrix ([21] p. 39), the Kullback–Leibler divergence [4,22], entropy ([4] p. 224), the Rényi information divergence [23–25], and mutual information (MI) ([4] p. 231) have been used. Among all these measures, the FIM has been widely used to compute the sensor performance because the inverse of the FIM provides the lower bound on the state error covariance matrix according to the Cramer–Rao lower bound (CRLB). Assuming the CRLB inequality to be satisfied, the FIM can be easily evaluated for linear system models (such as in a Kalman filter [26]) or problems in which the linearized models work reasonably well (as in the extended Kalman filter ([27] p. 195)). Although the CRLB is achieved for linear systems, there are no guarantees that it can be achieved for general nonlinear systems. Furthermore, the FIM is an approximation of the information and can be derived from the Taylor series expansion of the Kullback–Leibler divergence measure (see Appendix A and Ref. ([21] p. 39) for more details) about the estimate. As discussed in Appendix A, the FIM defines the local curvature, and hence provides a local measure of the information at a given estimate. On the other hand, the mutual information provides a consistent information measure to measure sensor performance and, unlike the FIM, it is a function of the entire probability density function rather than an estimate. To motivate the advantages of using the mutual information measure, let us consider the problem of localizing two objects with three sensors. The sensors are directional and have the constraint that they cannot observe more than one object at the same time. But, a single object can have multiple sensors observing it. In other words, the sensor parameters $\boldsymbol{\theta}$ in the sensor model [Eq. (2)] would represent the direction (or angle in this planar example). With this constraint and the limited number of objects, the possible directions (or configurations) become a finite set. The object uncertainties are modeled as a Gaussian PDF with means and covariance matrices: $\mu^{(1)} = [3;5]^T$, $\mu^{(2)} = [7;5]^T$, $P^{(1)} = [9,5;5,9]$, and $P^{(2)} = [25,-10;-10,9]$. The three sensors are located at $s_1 = [0;0]^T$, $s_2 = [5;0]^T$, and $s_3 = [10;0]^T$ The model for the jth sensor when observing the ith object, with state $x^{(i)} = [x^{(i)}(1), x^{(i)}(2)]$, is given by

$$y^{(i,j)} = \left[\sqrt{(xv(1) - s_j(1))^2 + \left(x^{(i)}(2) - s_j(2)\right)^2}; \; \operatorname{atan}\left(\frac{x^{(i)}(2) - s_j(2)}{x^{(i)}(1) - s_j(1)}\right)\right]^T + \omega_j$$

The Gaussian noise ω_j has zero mean and covariance R_j as

$$R_1 = R_3 = \operatorname{diag}\left(\left[10^2, \left(\frac{10\pi}{180}\right)^2\right]\right)$$

and

$$R_2 = \operatorname{diag}\left(\left[5^2, \left(\frac{5\pi}{180}\right)^2\right]\right)$$

The sensor-tasking problem has a total of eight possibilities. The cost can simply be evaluated for all these possible cases, and the best sensor pairing is chosen. To make the description simple, the decision variables $v^{(i,j)}$ are introduced to enumerate the eight possibilities. The decision variable $v^{(i,j)} \in \{0, 1\}$ represents the tasking of the jth sensor to the ith object when $v^{(i,j)} = 1$, and it is zero otherwise. The constraint that one sensor can observe only one object is transformed into the constraint

$$\sum_{i=1}^{2} v^{(i,j)} = 1$$

Table 1 Exhaustive search for sensor tasking using MI and FIM

Configuration	$v^{(1,1)}$	$v^{(1,2)}$	$v^{(1,3)}$	$v^{(2,1)}$	$v^{(2,2)}$	$v^{(2,3)}$	$\log\left(\frac{\lvert P_x\rvert\lvert P_y\rvert}{\lvert P\rvert}\right)$	$\lvert\text{FIM}\rvert$
C_1	1	0	0	0	1	1	**1.748**	0.4548
C_2	1	1	0	0	0	1	1.5324	0.45811
C_3	0	0	0	1	1	1	1.3824	0.0949
C_4	1	1	1	0	0	0	1.3001	0.042179
C_5	0	0	1	1	1	0	1.2643	0.26176
C_6	1	0	1	0	1	0	1.2234	**0.82134**
C_7	0	1	1	1	0	0	1.1579	0.19131
C_8	0	1	0	1	0	1	0.9491	0.75882

Note that the constraint

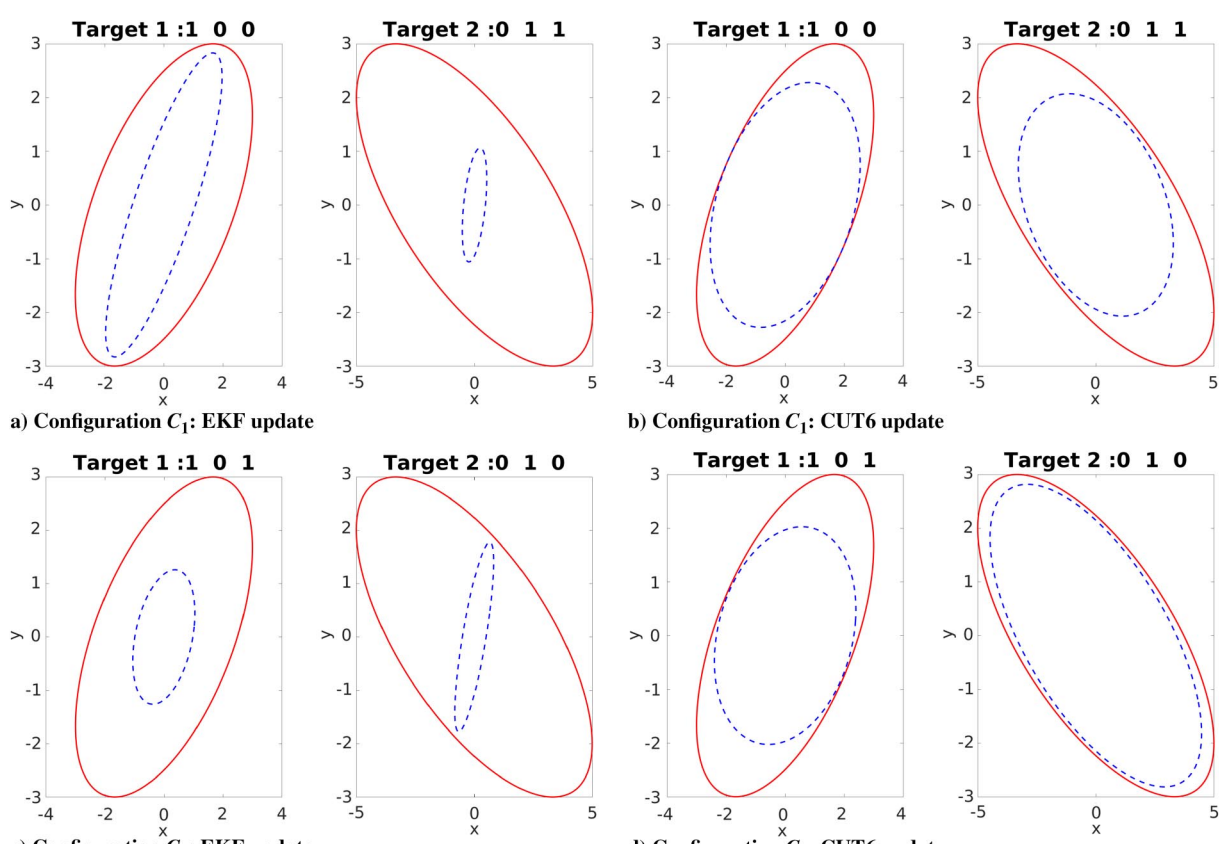
$$\sum_{i=1}^{2} v^{(i,j)} \leq 1$$

can be used to include the case where the jth sensor is apparently not used.

Table 1 shows the exhaustive search of the decision variables that satisfy the constraints of the sensor. Both the MI and the FIM are evaluated for comparison. The FIM is computed using the EKF (as in Refs. [5,28]) update equations, and the covariance in the MI is computed using the Conjugate Unscented Transform (CUT) points of 6th order, denoted as CUT6 points [20]. The expression used to compute the MI is shown in the header of Table 1 as

$$\log\left(\frac{|P_x||P_y|}{|P|}\right)$$

Here, P_x is the covariance of $x = [x_1; x_2]^T$, P_y is the covariance of y (selected measurement variables stacked as a column vector), and P is the joint covariance of x and y. Further details for the MI are described in Sec. III.A. This example is just to illustrate the conventional sensor tasking using the FIM with the EKF in comparison to the MI computed by sigma points. It is observed from Table 1 that the maximum of the MI and the FIM results in different configurations of C_1 and C_6, respectively. To further investigate these configurations, the *a priori* and *a posteriori* covariance ellipses for both targets are shown in Fig. 1 for configurations C_1 and C_6. In Fig. 1a, the EKF tends to overestimate the *a posteriori* covariance for target 2, whereas the CUT6 more properly captures the *a posteriori* covariance (this has also been verified by using a Gauss–Hermite quadrature with seven points in each dimension to compute the *a posteriori* covariance). Similarly, Figs. 1c and 1d show the *a priori* and *a posteriori* covariance ellipses for both the targets when the maximum FIM configuration of C_6 is selected. Here, the EKF tends to overestimate the *a posteriori* covariances for both the targets, and hence overestimates the information in the system. It is evident that the estimates in Fig. 1b are better than Fig. 1d. The mutual information in C_1 computed by a higher-order cubature provides a better and consistent measure for sensor selection. Hence, it is advantageous to choose a



a) Configuration C_1: EKF update

b) Configuration C_1: CUT6 update

c) Configuration C_6: EKF update

d) Configuration C_6: CUT6 update

Fig. 1 CUT6 vs EKF: reduction in covariance for cases C_1 and C_6.

consistent nonlinear filter and information measure that can result in efficient sensor configurations and consistent estimates.

In this respect, we use CUT-based sigma-point filters to accurately evaluate mutual information while assuming the *a posteriori* density function to be Gaussian. It should be mentioned that the higher-order methods provide more accurate estimates of the covariance matrix for the *a posteriori* Gaussian density function as compared to the EKF or the unscented Kalman filter [20,29]. In the following, the computational aspects of computing the MI are discussed. For ease of notation, the indices of the objects and measurements are dropped. The information measures are first discussed for a single generic object and a measurement vector over a period of one time step. These are then extended to the case of multiple measurements and multiple time steps.

A. Mutual Information

The mutual information at time step $k+1$ for state $\boldsymbol{x}_{k+1}$ and measurement $\boldsymbol{y}_{k+1}$ random variables is given by ([4] p. 231)

$$I(\boldsymbol{x}_{k+1};\boldsymbol{y}_{k+1}) = \iint p(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1}) \log\left\{\frac{p(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1})}{p(\boldsymbol{x}_{k+1})p(\boldsymbol{y}_{k+1})}\right\} \mathrm{d}\boldsymbol{x}_{k+1}\,\mathrm{d}\boldsymbol{y}_{k+1} \tag{6}$$

Furthermore, by using the conditional probability rule ([30] p. 98)

$$p(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1}) = p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{k+1})p(\boldsymbol{y}_{k+1})$$

the MI becomes [31]

$$I(\boldsymbol{x}_{k+1};\boldsymbol{y}_{k+1}) = E_{\boldsymbol{y}_{k+1}}[D_{\mathrm{KL}}(p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{k+1})||p(\boldsymbol{x}_{k+1}))]$$

Here, $D_{\mathrm{KL}}(p||q)$ is the Kullback–Leibler divergence from PDF q to PDF p. Note that the random vectors $\boldsymbol{x}_{k+1}$ and $\boldsymbol{y}_{k+1}$ can be any random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ for which the PDFs $p(\boldsymbol{x},\boldsymbol{y})$, $p(\boldsymbol{x})$, $p(\boldsymbol{y})$, $p(\boldsymbol{x}|\boldsymbol{y})$, and $p(\boldsymbol{y}|\boldsymbol{x})$ exist. Unless otherwise stated, the logarithm is taken with base e. Alternatively, by using the conditional probability rule ([30] p. 98)

$$p(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1}) = p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})p(\boldsymbol{x}_{k+1})$$

the MI can also be derived as

$$I(\boldsymbol{x}_{k+1};\boldsymbol{y}_{k+1}) = E_{\boldsymbol{x}_{k+1}}[D_{\mathrm{KL}}(p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})||p(\boldsymbol{y}_{k+1}))]$$

where mutual information is defined as the reduction in uncertainty ([4] p. 231). Hence, maximizing the MI will reduce the uncertainty in the system. Consequently, the MI can be interpreted as the expected Kullback–Leibler divergence of the *a priori* PDF $p(\boldsymbol{x})$ from the *a posteriori* PDF $p(\boldsymbol{x}||\boldsymbol{y})$. Hence, more the *a posteriori* PDF is different from the *a priori* PDF, more is the information. In general, the integrals in the MI have to be evaluated numerically, for example, using Monte Carlo methods or quadrature methods.

For a linear system with Gaussian uncertainties, the MI takes a simple analytical expression because the state PDF remains Gaussian for all time steps. Let the joint Gaussian PDF for $[\boldsymbol{x}_{k+1};\boldsymbol{y}_{k+1}]^T$ be given as

$$p(\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1}|I_k) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_{k+1}\\ \boldsymbol{y}_{k+1}\end{bmatrix}:\begin{bmatrix}\hat{\boldsymbol{x}}_{k+1/k}\\ \hat{\boldsymbol{y}}_{k+1}\end{bmatrix},\Sigma_{k+1/k}\right),$$
$$\Sigma_{k+1/k} = \begin{bmatrix}P_{k+1/k} & \Gamma_{k+1/k}\\ \Gamma^T_{k+1/k} & \Xi_{k+1/k}\end{bmatrix} \tag{7}$$

with

$$\hat{\boldsymbol{x}}_{k+1/k} = E[\boldsymbol{x}_{k+1}], \quad \hat{\boldsymbol{y}}_{k+1} = E[\boldsymbol{y}_{k+1}]$$
$$P_{k+1/k} = E[(\boldsymbol{x}_{k+1}-\hat{\boldsymbol{x}}_{k+1/k})(\boldsymbol{x}_{k+1}-\hat{\boldsymbol{x}}_{k+1/k})^T]$$
$$\Gamma_{k+1/k} = E[(\boldsymbol{x}_{k+1}-\hat{\boldsymbol{x}}_{k+1/k})(\boldsymbol{y}_{k+1}-\hat{\boldsymbol{y}}_{k+1})^T]$$

and

$$\Xi_{k+1/k} = E[(\boldsymbol{y}_{k+1}-\hat{\boldsymbol{y}}_{k+1})(\boldsymbol{y}_{k+1}-\hat{\boldsymbol{y}}_{k+1})^T]$$

Here, the expectations are with respect to the *a priori* PDF

$$p(\boldsymbol{x}_{k+1}|I_k) = \mathcal{N}(\boldsymbol{x}_{k+1}:\hat{\boldsymbol{x}}_{k+1/k}, P_{k+1/k})$$

Here, I_k is the set of measurements up to time k. When the process and measurement models are nonlinear, the expectation integrals can be computed numerically using quadratures as described in Ref. [20]. $\Gamma_{k+1/k}$ is the cross covariance of the state and measurement random vectors, $\Xi_{k+1/k}$ is the covariance for the measurement random vector, and $P_{k+1/k}$ is the *a priori* state covariance. The mean and covariance for the *a posteriori* PDF

$$p(\boldsymbol{x}_{k+1}|I_k,\boldsymbol{y}_{k+1}) = \mathcal{N}(\boldsymbol{x}_{k+1}:\hat{\boldsymbol{x}}_{k+1/k+1}, P_{k+1/k+1})$$

are given by the Kalman filter update equations or, equivalently, by computing the conditional Gaussian PDF [32] in Eq. (7) as

$$\hat{\boldsymbol{x}}_{k+1/k+1} = \hat{\boldsymbol{x}}_{k+1/k} + K_{k+1}(\boldsymbol{y}_{k+1}-\hat{\boldsymbol{y}}_{k+1})$$
$$P_{k+1/k+1} = P_{k+1/k} - K_{k+1}\Gamma^T_{k+1/k}$$

and $K_{k+1} = \Gamma_{k+1/k}\Xi^{-1}_{k+1/k}$. The measurement covariance $\Xi_{k+1/k}$ and cross covariance $\Gamma_{k+1/k}$ are implicitly a function of the sensor parameters because

$$\Xi_{k+1/k} = H_{k+1/k}P_{k+1/k}H^T_{k+1/k} + R_{k+1}$$

and $\Gamma_{k+1/k} = P_{k+1/k}H^T_{k+1/k}$ for a linear measurement model $H_{k+1} \equiv H_{k+1}(\boldsymbol{\theta}_{k+1})$, which is a function of the sensor configurations. The MI

$$I(\boldsymbol{x}_{k+1};\boldsymbol{y}_{k+1}) = \frac{1}{2}\log\left(\frac{|P_{k+1/k}||\Xi_{k+1/k}|}{|\Sigma_{k+1/k}|}\right)$$

is then simplified as

$$I(\boldsymbol{x}_{k+1};\boldsymbol{y}_{k+1}) = \frac{1}{2}\log\left(\frac{|P_{k+1/k}|}{|P_{k+1/k}-\Gamma_{k+1/k}\Xi^{-1}_{k+1/k}\Gamma^T_{k+1/k}|}\right)$$
$$= \frac{1}{2}\log\left(\frac{|\Xi_{k+1/k}|}{|\Xi_{k+1/k}-\Gamma^T_{k+1/k}P^{-1}_{k+1/k}\Gamma_{k+1/k}|}\right) \tag{8}$$

where the last two expressions of Eq. (8) are derived using the determinant of the joint covariance as

$$|\Sigma_{k+1/k}| = |P_{k+1/k}-\Gamma_{k+1/k}\Xi^{-1}_{k+1/k}\Gamma^T_{k+1/k}||\Xi_{k+1/k}|$$

or (Ref. [33] fact 8.13.36)

$$|\Sigma_{k+1/k}| = |\Xi_{k+1/k}-\Gamma^T_{k+1/k}P^{-1}_{k+1/k}\Gamma||P_{k+1/k}|$$

This expression is similar to the mean of the Kullback–Leibler divergence measure as developed in Ref. [6]. The MI is a measure of the reduction of uncertainty, i.e., the reduction from $|P_{k+1/k}|$ to $|P_{k+1/k}-\Gamma_{k+1/k}\Xi^{-1}_{k+1/k}\Gamma^T_{k+1/k}|$ or from $|\Xi_{k+1/k}|$ to $|\Xi_{k+1/k}-\Gamma^T_{k+1/k}P^{-1}\Gamma_{k+1/k}|$. The former notion is the reduction in the covariance of the state PDF, whereas the latter is the reduction in the covariance of the measurement, and thereby maximizing the likelihood of the measurement. This development of mutual information for linear system models provides valuable insights into computing the mutual information for nonlinear system models. If the approximate joint Gaussian PDF of states and measurements can be computed efficiently, then the

equation in Eq. (8) provides analytical expressions to compute the MI. With this motivation, we propose to use high-order sigma points [20,34–36] to approximate the joint PDF as Gaussian. Note that, in general, the MI in Eq. (8) can contain any state and measurement variables over multiple time steps and objects. In the next section, we provide numerical methods to compute the joint mutual information for multiple objects, measurements, and time steps.

B. Computing Joint Mutual Information over Multiple Time Steps

Optimizing sensor parameters well ahead of time requires the joint MI of all objects and sensors over multiple time steps. Assuming a Gaussian approximation to this joint PDF of object states and sensor measurements, it suffices to only compute the joint mean and covariance. Given the state estimates $\hat{\boldsymbol{x}}_{0/0}^{(i)}$ and covariance $P_{0/0}^{(i)}$ for each object at time step $k = 0$, the objective is to compute the joint mutual information using all the object states and the measurements up to the final time step N_T. Let

$$p\left(\boldsymbol{X}_{(0:N_T)}^{1:N_o}, \boldsymbol{Y}_{(0:N_T)}^{(1:N_o,1:N_s)}\right)$$

represent the joint Gaussian PDF of all the objects, over all time steps, and all the corresponding measurements. Often, the objects can be assumed independent when the objects can be perfectly resolved from measurements and are not controlled with respect to each other. This assumption reduces the joint MI

$$I\left(\boldsymbol{X}_{(0:N_T)}^{1:N_o}; \boldsymbol{Y}_{(0:N_T)}^{(1:N_o,1:N_s)}\right)$$

as the sum of the MI of individual objects:

$$I\left(\boldsymbol{X}_{(0:N_T)}^{1:N_o}; \boldsymbol{Y}_{(0:N_T)}^{(1:N_o,1:N_s)}\right) = \sum_{i=1}^{N_o} I\left(\boldsymbol{X}_{(0:N_T)}^{(i)}; \boldsymbol{Y}_{(0:N_T)}^{(i,1:N_s)}\right) \tag{9}$$

The problem then reduces to computing $p\left(\boldsymbol{X}_{(0:N_T)}^{(i)}, \boldsymbol{Y}_{(0:N_T)}^{(i,1:N_s)}\right)$ for each object individually, which in turn reduces to computing the joint mean and covariance of each object with its corresponding measurements over all time steps. As the computational procedure is similar to all objects, the superscript i is dropped and the process is described for a generic object. We first describe mutual information for a linear system over all time steps $k = 0, 1, \ldots, N_T$ and measurement variables $j = 1, 2, \ldots, N_s$. Given the PDF for the initial state as $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{x}_0 : \boldsymbol{x}_{0/0}, P_{0/0})$, the joint PDF $p\left(\boldsymbol{X}_{(0:N_T)}, \boldsymbol{Y}_{(0:N_T)}\right)$ is computed by the joint transition matrix F as

$$\left[\boldsymbol{X}_{(0:N_T)}; \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)}\right]^T = F\left[\boldsymbol{x}_0; \boldsymbol{W}_{(0:N_T-1)}; \boldsymbol{V}_{(1:N_T)}^{(1:N_s)}\right]^T$$

where the matrix F is derived by recursively applying the dynamical system matrices $A_k^{(i)}$ and $H_{k+1}^{(i,j)}$ for $k = 0, 1, \ldots, N_T$. The initial condition $\boldsymbol{x}_0$, the process noise sequence $\boldsymbol{W}_{(0:N_T-1)} = [\omega_0, \omega_1, \ldots, \omega_{N_T-1}]$, and the measurement noise sequence

$$\boldsymbol{V}_{(1:N_T)}^{1:N_s} = \left[\nu_{(1:N_T)}^{(1)}, \nu_{(1:N_T)}^{(2)}, \ldots, \nu_{(1:N_T)}^{(N_s)}\right]$$

were assumed to be mutually independent. Hence, the PDF $p\left(\boldsymbol{x}_0, \boldsymbol{W}_{0:N_T-1}, \boldsymbol{V}_{(1:N_T)}^{1:N_s}\right)$ is completely determined as the product of the PDFs for each random variable. Under the linear map F, the joint PDF $p\left(\boldsymbol{X}_{(0:N_T)}, \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)}\right)$ is Gaussian and is directly computed from $p\left(\boldsymbol{x}_0, \boldsymbol{W}_{(0:N_T-1)}, \boldsymbol{V}_{(1:N_T)}^{1:N_s}\right)$ using the transformation of probabilities ([37] p. 34). Alternatively, the mean and covariance matrix of $\left[\boldsymbol{X}_{(0:N_T)}; \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)}\right]^T$ are computed as

$$\begin{aligned}
\boldsymbol{\mu} &= E\begin{bmatrix} \boldsymbol{X}_{(0:N_T)} \\ \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)} \end{bmatrix} = F\begin{bmatrix} \hat{\boldsymbol{x}}_{0/0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \\
\Sigma &= E\left(\begin{bmatrix} \boldsymbol{X}_{(0:N_T)} \\ \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)} \end{bmatrix} - \boldsymbol{\mu}\right)\left(\begin{bmatrix} \boldsymbol{X}_{(0:N_T)} \\ \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)} \end{bmatrix} - \boldsymbol{\mu}\right)^T \\
&= F\begin{bmatrix} P_{0/0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & Q_{(0:N_T-1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & R_{(1:N_T)}^{(1:N_s)} \end{bmatrix} F^T
\end{aligned} \tag{10}$$

where $Q_{(0:N_T-1)}$ is the block diagonal matrix with the matrices $[Q_0, Q_1, \ldots, Q_{N_T-1}]$ along the main diagonal. Similarly, $R_{(1:N_T)}^{(1:N_s)}$ is the block diagonal matrix of all the measurement noise covariances from time step 1 to N_T. Similar to the linear system, for a nonlinear system as in the Eqs. (1) and (2), the system of equations is expressed as

$$\left[\boldsymbol{X}_{(0:N_T)}; \boldsymbol{Y}_{(1:T)}^{(1:N_s)}\right]^T = \mathcal{F}\left(\boldsymbol{x}_0, \boldsymbol{W}_{(0:N_T-1)}, \boldsymbol{V}_{(1:N_T)}^{(1:N_s)}\right)$$

Given the initial state and noise variables, the nonlinear map $\mathcal{F}(.)$ generates all the state and measurement variables. The nonlinear map is similar to the linear map, except that $\mathcal{F}$ is now a nonlinear implicit function of the initial state and noise variables. The mean and covariance are then computed numerically using sigma points as

$$\begin{aligned}
\boldsymbol{\mu} &= E\begin{bmatrix} \boldsymbol{X}_{(0:N_T)} \\ \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)} \end{bmatrix} = E\left[\mathcal{F}\left(\boldsymbol{x}_0, \boldsymbol{W}_{(0:N_T-1)}, \boldsymbol{V}_{(1:N_T)}^{(1:N_s)}\right)\right] \\
&= \sum_{q=1}^{N} w_q \mathcal{F}\left(\boldsymbol{x}_0^{(q)}, \boldsymbol{W}_{(0:N_T-1)}^{(q)}, \boldsymbol{V}_{(1:N_T)}^{(1:N_s),q}\right)
\end{aligned} \tag{11}$$

$$\begin{aligned}
\Sigma &= E\left(\begin{bmatrix} \boldsymbol{X}_{(0:N_T)} \\ \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)} \end{bmatrix} - \boldsymbol{\mu}\right)\left(\begin{bmatrix} \boldsymbol{X}_{(0:N_T)} \\ \boldsymbol{Y}_{(1:N_T)}^{(1:N_s)} \end{bmatrix} - \boldsymbol{\mu}\right)^T \\
&= \sum_{q=1}^{N} w_q \left(\mathcal{F}\left(\boldsymbol{x}_0^{(q)}, \boldsymbol{W}_{(0:N_T-1)}^{(q)}, \boldsymbol{V}_{(1:N_T)}^{(1:N_s),q}\right) - \boldsymbol{\mu}\right) \\
&\quad \times \left(\mathcal{F}\left(\boldsymbol{x}_0^{(q)}, \boldsymbol{W}_{(0:N_T-1)}^{(q)}, \boldsymbol{V}_{(1:N_T)}^{(1:N_s),q}\right) - \boldsymbol{\mu}\right)^T
\end{aligned} \tag{12}$$

where the N sigma points or quadrature points

$$\left\{\left(\boldsymbol{x}_0^{(q)}, \boldsymbol{W}_{(0:N_T-1)}^{(q)}, \boldsymbol{V}_{(1:N_T)}^{(1:N_s),q}\right), w_q\right\}$$

are generated from the distribution

$$\mathcal{N}\left(\left[\boldsymbol{x}_0; \boldsymbol{W}_{(0:N_T-1)}; \boldsymbol{V}_{(1:N_T)}^{(1:N_s)}\right]^T : [\hat{\boldsymbol{x}}_{0/0}; \boldsymbol{0}; \boldsymbol{0}]^T, \right.$$
$$\left. \operatorname{diag}\left(P_{0/0}, Q_{(1:N_T-1)}, R_{(1:N_T)}^{(1:N_s)}\right)\right)$$

In the absence of process noise, or when the process noise can be considered to be small (as is often the case for space object motion), the computational complexity is reduced significantly. The uncertainty in the system is only due to initial conditions and measurement noise. Hence, the number of sigma points required is also reduced. Using the index i for a generic object, the joint covariance matrix Σ over all measurements and time steps in Eq. (12) is expressed as $\underset{(0:N_T)}{\overset{(i,1:N_s)}{\Sigma}}$ which can further be partitioned to separate the state and measurement subblock matrices as

$$\underset{(0:N_T)}{\overset{(i,1:N_s)}{\Sigma}} = \begin{bmatrix} P^{(i)}_{(0:N_T)} & \Gamma^{(i,1:N_s)}_{(0:N_T)} \\ \Gamma^{(i,1:N_s)^T}_{(0:N_T)} & \Xi^{(i,1:N_s)}_{(0:N_T)} \end{bmatrix} \tag{13}$$

The mutual information in Eq. (8) for the ith object is then given as

$$I\left(\boldsymbol{X}^{(i)}_{(0:N_T)}; \boldsymbol{Y}^{(i,1:N_s)}_{(0:N_T)}\right) = \frac{1}{2}\log\left\{\frac{\left|P^{(i)}_{(0:N_T)}\right|}{\left|P^{(i)_{(0:N_T)}} - \Gamma^{(i,1:N_s)}_{(0:N_T)}\left(\Xi^{(i,1:N_s)}_{(0:N_T)}\right)^{-1}\left(\Gamma^{(i,1:N_s)}_{(0:N_T)}\right)^T\right|}\right\} \tag{14}$$

$$= \frac{1}{2}\log\left\{\frac{\left|\Xi^{(i,1:N_s)}_{(0:N_T)}\right|}{\left|\Xi^{(i,1:N_s)}_{(0:N_T)} - \left(\Gamma^{(i,1:N_s)}_{(0:N_T)}\right)^T\left(P^{(i,1:N_s)}_{(0:N_T)}\right)^{-1}\Gamma^{(i,1:N_s)}_{(0:N_T)}\right|}\right\} \tag{15}$$

The measurement covariance $\Xi^{(i,1:N_s)}_{(0:N_T)}$ and the cross covariance $\Gamma^{(i,1:N_s)}_{(0:N_T)}$ are functions of the sensor configurations. In this case, the MI in Eq. (15) is easily computed for a subset of sensors and time steps by only using the relevant rows and columns in the joint covariance matrix in Eq. (13). This helps to perform a straightforward exhaustive search to find the best subset of sensors to maximize the MI over time. This problem of optimizing sensor configurations is described in further detail in the next section. It should be noticed that the accurate computation of the MI requires the accurate computation of the mean and covariance for nonlinear models. In this respect, we prefer to use higher-order cubature/quadrature methods [20].

IV. Sensor Tasking

Embedded in expressions (14) and (15) are the sensor configuration parameters. These sensor configuration parameters $\boldsymbol{\theta}^{(j)}_k \in \mathbb{R}^p$ for $k = 1, 2, \ldots, N_T$ in the measurement model [Eq. (2)] influence the measurements and the sensor noise, and hence influence the mutual information. The objective of the sensor-tasking process is to then find the best combination of sensor parameters to maximize the information in the system. These configuration parameters can be binary, integer, discrete (such as a simple on/off or other operating modes), and even continuous (such as orientation, position). In this paper, we choose to express all the configuration parameters as discrete. All the continuous sensor parameters are appropriately discretized over their respective domains. For a given sensor, once all the configuration parameters are uniformly expressed as discrete values, we construct the tensor product of the parameter values to enumerate all possible configurations for the sensor. We denote this set as $\mathcal{S}^{(k)}_j$ for the jth sensor at time step k. The sensor is then a constraint to operate in any of these configurations $\boldsymbol{\theta}^{(j)}_k \in \mathcal{S}^{(k)}_j$. Furthermore, we denote the index for the set $\mathcal{S}^{(k)}_j$ as $s_{(j,k)}$. The complete set of sensor configurations is denoted as $\mathcal{E} = \cup_k \left\{\cup_j \mathcal{S}^k_j\right\}$ for $j = 1, 2, \ldots, N_s$ and $k = 1, 2, \ldots, N_T$. We assume that sets $\mathcal{S}^{(k)}_j$ and $\mathcal{E}$ are ordered with respect to their indices.

Every combination of sensor parameter states $\boldsymbol{\theta}^{(j)}_k$ uniquely configures the jth sensor at time step k. To this end, we consider this configuration to be a new sensor, $\boldsymbol{h}^{(j)}\left(\boldsymbol{x}^{(i)}_k, \boldsymbol{\theta}^{(j)}_k\right) \equiv \boldsymbol{h}^{s_{(j,k)}}\left(\boldsymbol{x}^{(i)}_k\right)$ in Eq. (2), when $\boldsymbol{\theta}^{(j)}_k$ takes the value in $\mathcal{S}^{(k)}_j$ corresponding to the index $s_{(j,k)}$. For simpler notation purposes, we also denote the sensor models as $\boldsymbol{h}^{j'}\left(\boldsymbol{x}^{(i)}_k\right)$, where j' is the index for the set $\mathcal{E}$, with $j' = 1, 2, \ldots, N_{s'}$ and $N_{s'} = |\mathcal{E}|$. As $\mathcal{E}$ is the set of all configurations for all sensors over all time steps, we consider the index of $\mathcal{E}$ to be incremented first along sensors and then time steps. The problem is then to select an appropriate set of sensors in set $\mathcal{E}$, which is subject to user defined constraints. Alternatively, one can use binary variables to select/deselect a sensor. Now, the optimization problem of maximizing the information or sensor utility cost becomes a nonlinear binary-integer programming problem. Hence, integer nonlinear programming solvers [38] are required to maximize the information measure. These solvers can be computationally expensive and sensitive to initial conditions, resulting in solutions that are often local maxima. We first develop the exact sensor optimization problem in terms of a nonlinear binary-integer program for completeness and to illustrate the sensor-tasking problem. As the exact optimal solution is often intractable, we then proceed to develop approximate methods to alleviate the computational complexity and generate near-real-time solutions. Furthermore, in Sec. V, we show that the MI is submodular, and we leverage greedy heuristics with known suboptimal bounds.

A. Maximizing Mutual Information

In this section, the MI is transcribed into an objective function for optimization using binary integer variables. These variables select or deselect sensors in set $\mathcal{E}$. The MI cost function in Eq. (9) [and Eq. (14)] is modified to include the effect of the decision variables $v^{(i,j')}_k$ as

$$\max_{v^{(i,j')}_k}: \sum_{i=1}^{N_o} I\left(\boldsymbol{X}^{(i)}_{(0:N_T)}; v^{(i,1:N'_s)}_{(0:N_T)}\boldsymbol{Y}^{(i,1:N'_s)}_{(0:N_T)}\right) \tag{16}$$

$$\text{subject to: } A\boldsymbol{v} \leq b, \quad C\boldsymbol{v} = d \tag{17}$$

$$v^{(i,j')}_k \in \{0, 1\} \tag{18}$$

$$1 \leq i \leq N_o, \quad 1 \leq j' \leq N'_s, \quad 0 \leq k \leq N_T \tag{19}$$

where

$$v^{(i,1:N'_s)}_{(0:N_T)} = \left\{v^{(i,j')}_k | 1 \leq j' \leq N'_s, 0 \leq k \leq N_T\right\}$$

is the collection of all decision variables for object i over all time steps, and the vector $\boldsymbol{v} = \left[\ldots, v^{(i,j')}_k, \ldots\right]^T$. $v^{(i,j')}_k$ is introduced to selectively remove or add the corresponding sensor for an object at a particular time as

$$v^{(i,1:N'_s)}_{(0:N_T)}\boldsymbol{Y}^{(i,1:N'_s)}_{(0:N_T)} = \Bigg[\underbrace{v^{(i,1)}_0\boldsymbol{y}^{(i,1)}_0, v^{(i,2)}_0\boldsymbol{y}^{(i,2)}_0, \ldots, v^{(i,N'_s)}_0\boldsymbol{y}^{(i,N'_s)}_0}_{k=0}, \ldots, \underbrace{v^{(i,1)}_T\boldsymbol{y}^{(i,1)}_T, v^{(i,2)}_T\boldsymbol{y}^{(i,2)}_T, \ldots, v^{(i,N'_s)}_T\boldsymbol{y}^{(i,N'_s)}_T}_{k=N_T}\Bigg] \tag{20}$$

The objective function I in Eq. (16) is to be interpreted as the information in all the object states and measurements over the time period of zero to N_T. The binary variables $\boldsymbol{v} = \left\{v^{(i,j')}_k\right\}$ are used to include or remove sensors from the vector [Eq. (20)], and hence from computation of the mutual information. When $v^{(i,j')}_k = 0$, then object i is not observed with sensor j' at time k; i.e., the measurement variable $y^{(i,j')}_k$ is removed, and it has no effect on the mutual information cost. Constraint matrices A, b, C, and d are used to model operational constraints and budgets. For example, operational constraints can include switching off sensors when visibility or observability is low, and operation budgets can include the cost of using a sensor at different times. Note that, in many problems, the variable $v^{(i,j')}_k$ may not be independent. For example, when a sensor has multiple operating modes, only one mode/configuration can be used at a single time step. These constraints can be enforced by using matrices C and d. Further details and applications are described in Sec. V.

The effect of removing sensors, when computing the mutual information, translates to removing the corresponding rows and columns of the sensor in the joint covariance matrix. Alternatively, the decision variables $\boldsymbol{v}$ can also be introduced directly into the measurement covariance and cross-covariance matrices of Eq. (13) in such a way so as to negate the effect of the sensor in computing the MI. Specifically, $\Xi^{(i)}$ and $\Gamma^{(i)}$ of Eqs. (13) and (14) are partitioned as

$$\Gamma^{(i)} = \left[\Gamma_1^{(i)} \,\middle|\, \Gamma_2^{(i)} \,\middle|\, \ldots \,\middle|\, \Gamma_k^{(i)} \,\middle|\, \ldots \,\middle|\, \Gamma_T^{(i)} \right] \tag{21}$$

$$\Gamma_k^{(i)} = \left[\begin{array}{cccc} v_k^{(i,1)}\Gamma_{(1,k)}^{(i,1)} & v_k^{(i,2)}\Gamma_{(1,k)}^{(i,2)} & \cdots & v_k^{(i,N_s')}\Gamma_{(1,k)}^{(i,N_s')} \\ \hline v_k^{(i,1)}\Gamma_{(2,k)}^{(i,1)} & v_k^{(i,2)}\Gamma_{(2,k)}^{(i,2)} & \cdots & v_k^{(i,N_s')}\Gamma_{(2,k)}^{(i,N_s')} \\ \hline \vdots & \vdots & \cdots & \vdots \\ \hline v_k^{(i,1)}\Gamma_{(N_T,k)}^{(i,1)} & v_k^{(i,2)}\Gamma_{(N_T,k)}^{(i,2)} & \cdots & v_k^{(i,N_s')}\Gamma_{(N_T,k)}^{(i,N_s')} \end{array} \right] \tag{22}$$

and

$$\Xi^{(i)} \left[\Xi_{(1)}^{(i)} \,\middle|\, \Xi_{(2)}^{(i)} \,\middle|\, \ldots \,\middle|\, \Xi_{(k)}^{(i)} \,\middle|\, \ldots \,\middle|\, \Xi_{(N_T)}^{(i)} \right] \tag{23}$$

$$\Xi_{(k)}^{(i)} = \left[\begin{array}{cccc} v_1^{(i,1)} v_k^{(i,1)} \Xi_{(1,k)}^{(1,1),i} & v_1^{(i,1)} v_k^{(i,2)} \Xi_{(1,k)}^{(1,2),i} & \cdots & v_1^{(i,1)} v_k^{(i,N_s')} \Xi_{(1,k)}^{(1,N_s'),i} \\ \hline v_1^{(i,2)} v_k^{(i,1)} \Xi_{(1,k)}^{(2,1),i} & v_1^{(i,2)} v_k^{(i,2)} \Xi_{(1,k)}^{(2,2),i} & \cdots & v_1^{(i,2)} v_k^{(i,N_s')} \Xi_{(1,k)}^{(2,N_s'),i} \\ \hline \vdots & \vdots & \cdots & \vdots \\ \hline v_T^{(i,N_s')} v_k^{(i,1)} \Xi_{(N_T,k)}^{(N_s',1),i} & v_T^{(N_s',i)} v_k^{(i,2)} \Xi_{(N_T,k)}^{(N_s',2),i} & \cdots & v_T^{(i,N_s')} v_k^{(i,N_s')} \Xi_{(N_T,k)}^{(N_s',N_s'),i} \end{array} \right] \tag{24}$$

where $\Gamma_{(k_i,k_{j'})}^{i,j'}$ is the cross covariance of the ith object state at time step k_i with the measurement output of the j'th sensor at time step $k_{j'}$. Similarly, $\Xi_{(k_1,k_2)}^{(j_1',j_2'),i}$ is the covariance of the j_1'th sensor measurement (of the ith object) at time step k_1 with the j_2'th sensor measurement (of the ith object) at time step k_2. Note that the coefficient of $\Xi_{(k_1,k_2)}^{(j_1',j_2'),i}$ is one when $j_1' = j_2'$ and $k_1 = k_2$, and it is $v_{k_1}^{(i,j_1')} v_{k_2}^{(i,j_2')}$ otherwise. This is important so as to not make the determinant zero when a sensor is removed, as well as to consistently compute the right MI cost when the sensor is selected or deselected. On substituting Eqs. (22) and (24) into the expression for the MI in Eq. (15) and then expanding the determinant, the MI is transformed into a function of the decision variables $\boldsymbol{v}$ of the form

$$I_{(\mathrm{Det})}: \max_{\boldsymbol{v}} \sum_{i=1}^{N_o} \left\{ \frac{|\Xi^{(i)}|}{|\Xi^{(i)} - (\Gamma^{(i)})^T (P^{(i)})^{-1} \Gamma^{(i)}|} \right\} = \sum_{i=1}^{N_o} \frac{p_i(\boldsymbol{v})}{q_i(\boldsymbol{v})} = \frac{p(\boldsymbol{v})}{q(\boldsymbol{v})} \tag{25}$$

$$\text{subject to: } Av \le b, \quad Cv = d \tag{26}$$

$$v_{(k)}^{(i,j')} - (v_{(k)}^{(i,j')})^2 = 0 \tag{27}$$

$$1 \le i \le N_o \quad 1 \le j' \le N_s \quad 0 \le k \le N_T \tag{28}$$

where the logarithm in Eq. (15) has been dropped without loss of optimality. The constraint $v_{(k)}^{(i,j')} - \left(v_{(k)}^{(i,j')}\right)^2 = 0$ is used to enforce the binary constraint for the decision variables. Note that $p_i(\boldsymbol{v})$, $q_i(\boldsymbol{v})$, $p(\boldsymbol{v})$, and $q(\boldsymbol{v})$ are polynomials in the variables $\boldsymbol{v}$. Also, $|P^{(i)}|$ is a constant with respect to the decision variables $\boldsymbol{v}$. This form of optimization can be solved by special numerical solvers such as GloptiPoly [39], which considers a series of semi-definite programming (SDP) relaxations to obtain the global optimum solution. But, such relaxations for higher-order polynomials lead to increased computational complexity for each SDP relaxation. Alternatively, polynomial solvers such as Bertini™ can be used [40]. In most cases, the determinant in Eq. (25) exponentially increases the order of the polynomials. Even though there might be a few binary variables, the degree of the polynomials can be large when the number of states in the process model or measurement model are high. Hence, these numerical methods are only effective for low-dimensional systems to find the global maximum solution. In this regard, the exact optimization problem in Eq. (25) is only used to illustrate the ideal sensor-tasking optimization problem and to help develop and motivate greedy heuristics in Sec. V that can achieve suboptimal solutions to this ideal sensor-tasking problem.

B. Sequential in Time

In this sequential strategy, the optimization problem is solved only for the variables at the current time step. The optimization problem at the subsequent time step considers all the optimized variables up until the previous time step as fixed. The process continues up to the final time step N_T. The greedy time strategy only optimizes the information for a specific time step, and hence the number of configurations are reduced to $2^{N_o \times N_s}$ for a single time step. The computational cost is proportional to $N_T \times 2^{N_o \times N_s}$ for all time steps. The sequential in time optimization problem at time step k can be framed as

$$\max_{\{v_k^{i,j'}\}} I(\boldsymbol{X}_{(0:N_T)}: v_{(0:k)} \boldsymbol{Y}_{(0:k)} | \{v_0^{i,j'}\}^*, \{v_1^{i,j'}\}^*, \ldots, \{v_{k-1}^{i,j'}\}^*, \{v_{k+1}^{i,j'}\} = 0, \ldots, \{v_{N_T}^{i,j'}\} = 0) \tag{29}$$

$$\text{subject to: } A_k v_k \le b_k, \quad C_k v_k = d_k$$
$$v_{(k)}^{(i,j')} \in \{0,1\}, \quad 1 \le i \le N_o \quad 1 \le j' \le N_s' \tag{30}$$

where $\{v_0^{i,j'}\}^*, \{v_1^{i,j'}\}^*, \ldots, \{v_{k-1}^{i,j'}\}^*$ are the fixed decision variables at time steps $0, 1, \ldots, k-1$, respectively. As the sequential in time strategy progresses forward in time, the variables

$\{v_{k+1}^{i,j'}\}, \{v_{k+2}^{i,j'}\}, \ldots, \{v_{N_T}^{i,j'}\}$ are set to zero for all $1 \leq i \leq N_o$ and $1 \leq j \leq N_s'$. The conditioning in Eq. (29) is to be interpreted as known or constant values while computing the joint mutual information. In the case of a constraint in which a sensor can only observe a single target at any particular time, the possible configurations reduce to $N_T \times N_o^{N_s'}$. Note that the constraint matrices A_k, C_k, b_k, and d_k in Eq. (30) are constraints for a given time step k, and are not the general constraint matrices A, V, b, and d in Eq. (26).

C. Sequential in Sensors

Assuming that the tasking has been done up to $j-1$ sensors, the optimization for the jth sensor is framed as

$$\max_{\{v_{0:N_T}^{(:,j')}\}} I\Big(\boldsymbol{X}_{(1:N_T)}: v_{(0:N_T)}^{(:,j')}\boldsymbol{Y}_{(0:N_T)}^{(:,j')} \Big| \Big\{v_{0:N_T}^{(:,1)}\Big\}^*, \Big\{v_{0:N_T}^{(:,2)}\Big\}^*, \ldots, \Big\{v_{0:N_T}^{(:,j'-1)}\Big\}^*, \Big\{v_{0:N_T}^{(:,j'+1)}\Big\} = 0, \ldots, \Big\{v_{0:N_T}^{(:,N_s')}\Big\} = 0\Big) \tag{31}$$

$$\text{subject to: } A_{j'} v_{0:N_T}^{(:,j')} \leq b_{j'}, \quad C_{j'} v_{0:N_T}^{(:,j')} = d_{j'}$$
$$v_k^{(i,j')} \in \{0,1\}, \quad 1 \leq i \leq N_o, \quad 0 \leq k \leq N_T \tag{32}$$

$$\Big\{v_{0:N_T}^{(:,j')}\Big\} = \Big[\Big\{v_0^{(1,j')}, v_0^{(2,j')}, \ldots, v_0^{(N_o,j')}\Big\}, \cdots, \Big\{v_{N_T}^{(1,j')}, v_{N_T}^{(2,j')}, \ldots, v_{N_T}^{(N_o,j')}\Big\}\Big] \tag{33}$$

The MI cost in Eq. (31) is computed by considering all the previous sensor assignments $\Big\{v_{0:N_T}^{(:,1)}\Big\}^*, \Big\{v_{0:N_T}^{(:,2)}\Big\}^*, \ldots, \Big\{v_{0:N_T}^{(:,j'-1)}\Big\}^*$ as knowns/constants and having the remaining variable set as $v_k^{(i,u)} = 0$ $\forall\ j'+1 \leq u \leq N_s$, $1 \leq i \leq N_o$, and $1 \leq k \leq N_T$. Equivalently, the corresponding columns in the joint covariance matrix can be deleted in Eqs. (22) and (24) when evaluating the MI. It can be seen that the optimization at each iteration has only $N_o \times N_T$ variables. The total number of configurations possible are $N_s \times 2^{N_o \times N_T}$. But, with the constraint that a sensor can observe only one target at a particular time, the number of configurations reduces to $N_s \times N_o^{N_T}$.

D. Sequential in Objects

Yet another approach is to be sequential in objects. The objects can be ordered in decreasing order of their state uncertainties or by a user-defined priority order. The sensor-tasking problem is then solved for each object in the priority order while considering the tasking solutions of the previous objects as fixed. An additional penalty cost is often necessary to avoid the case where all the sensor resources are used up in the tasking problems of the first few objects in the priority list. The optimization problem for the ith object is given as

$$\max_{v_{(0:N_T)}^{(i,:)}} I\Big(\boldsymbol{X}_{(0:N_T)}^{(i)}: v_{(0:N_T)}^{(i,:)}\boldsymbol{Y}_{(0:N_T)}^{(i,:)} \Big| \Big\{v_{(0:N_T)}^{(1,:)}\Big\}^*, \Big\{v_{(0:N_T)}^{(2,:)}\Big\}^*, \ldots, \Big\{v_{(0:N_T)}^{(i-1,:)}\Big\}^*, \Big\{v_{(0:N_T)}^{(i+1,:)}\Big\} = 0, \ldots, \Big\{v_{(0:N_T)}^{(N_o,:)}\Big\} = 0\Big) - \lambda^T \phi\Big(v_{(0:N_T)}^{(i,:)}\Big) \tag{34}$$

$$\text{subject to: } A_i v_{0:N_T}^{(i,:)} \leq b_i, \quad C_i v_{0:N_T}^{(i,:)} = d_i$$
$$v_k^{(i,j')} \in \{0,1\}, \quad 1 \leq j' \leq N_s', \quad 0 \leq k \leq N_T \tag{35}$$

$$\Big\{v_{0:N_T}^{(i,:)}\Big\} = \Big[\Big\{v_0^{(i,1)}, v_0^{(i,2)}, \ldots, v_0^{(i,N_s')}\Big\}, \ldots, \Big\{v_{N_T}^{(i,1)}, v_{N_T}^{(i,2)}, \ldots, v_{N_T}^{(i,N_s)}\Big\}\Big] \tag{36}$$

The penalty in the information cost $\phi\Big(v_{(0:N_T)}^{(i,:)}\Big)$ is introduced to minimize the number of sensors allocated to the ith object, hence sparing some sensors for the next object. The penalty assumed can be taken as

$$\phi\Big(v_{(0:N_T)}^{(i)}\Big) = \left[\sum_{j'=1}^{N_s'} v_0^{(i,j')}, \sum_{j'=1}^{N_s} v_1^{(i,j')}, \ldots, \sum_{j'=1}^{N_s'} v_{N_T}^{(i,j')}\right]^T$$

to represent the number of sensors used in each time step. Note that $\lambda \in \mathbb{R}^{N_T}$ is the desired scaling vector that can be used to relatively weight the number of sensors used at different time steps. Alternatively, another approach would be to minimize the number of sensors used while maintaining the information above a user-defined threshold:

$$\min: \sum_{j'=1}^{N_s'} v_{(0:N_T)}^{(i)} \tag{37}$$

$$\text{subject to: } I\Big(\boldsymbol{X}_{(0:N_T)}^{(i)}: v_{(0:N_T)}^{(i)}\boldsymbol{Y}_{(0:N_T)}^{(i)} | v_{(0:N_T)}^{(i-1)}, v_{(0:N_T)}^{(i-2)}, \ldots, v_{(0:N_T)}^{(1)}\Big) \geq I_{\epsilon_i} \tag{38}$$

where I_{ϵ_i} is the information threshold used to specify the minimum amount of information to be gained for each object. A lower threshold would lead to fewer sensors tasked to object i, hence making sensors available for the remaining objects. The number of possible configurations for the sequential in objects approach is $N_o \times 2^{N_s' \times N_T}$.

Often in these sequential strategies, an exhaustive search might be computationally attractive because computing the MI over the search space of decision variables can be effectively parallelized with ease. In very large-scale problems, these strategies can be combined to make the sensor-tasking problem more tractable. It is emphasized that the choice of approximation is often problem dependent.

E. Mutually Independent Measurements

To further ease computational complexity, the sensor measurement states are assumed to be mutually independent at the current time step. This assumption is generally not valid because the measurement states are correlated through the object states. Nevertheless, for large-scale problems, the resultant cost function is very attractive because it can be solved efficiently using linear binary integer programming solvers that use branch-and-bound methods. With a single time step moving horizon (i.e., $N_K = 1$), the optimization problem becomes

$$\max_{\{v_k^{i,j'}\}} \sum_{i=1}^{N_o} \sum_{j'=1}^{N_s} v_{(k)}^{(i,j')} I\Big(\boldsymbol{x}_k^{(i)}: \boldsymbol{y}_{(k)}^{(i,j')}\Big) \tag{39}$$

where

$$I\Big(\boldsymbol{x}_k^{(i)}: v_{(k)}^{(i,j')}\boldsymbol{y}_{(k)}^{(i,j')}\Big) = v_{(k)}^{(i,j')} I\Big(\boldsymbol{x}_k^{(i)}: \boldsymbol{y}_{(k)}^{(i,j')}\Big)$$

because, when $v_k^{(i,j')} = 0$, the information becomes

$$I\Big(\boldsymbol{x}_k^{(i)}: v_k^{(i,j')}\boldsymbol{y}_{(k)}^{(i,j')}\Big) = I\Big(\boldsymbol{x}_k^{(i)}: \varnothing\Big) = 0$$

When solving the optimization in Eq. (39) at time step k with $N_K = 1$, the estimates $\hat{\boldsymbol{x}}_{k-1/k-1}$ and $P_{k-1/k-1}$ of all the objects at time $k-1$ are known. In Ref. [5], a similar binary integer programming was solved using the FIM as the sensor performance cost. Once the optimization problem is solved for time step k, the sensors are configured and wait for measurements at time k. The estimates for the objects that were tasked to be observed are updated at time k with the newly available measurements. For objects with no measurements, the *a priori* estimates simply become the *a posteriori* estimates,

and a new optimization problem is solved for time step $k+1$. Note that all the measurements from all the sensors over time steps are correlated through the state dynamics for an object. Although this assumption disregards this correlation, it provides a computationally attractive approach. Note that problem (39), when arranged in table, is similar to the information table in Refs. [3,5]. In Ref. [3], the authors described a sensor-tasking problem by first constructing the information table over multiple time steps. The auction algorithm was then used to select the elements of this information table. It is to be noted that, in an ideal sensor-tasking problem, the information table over time cannot be constructed before the sensor task. This is because the information over time is dependent on the sensor task, and hence is a nonlinear problem. But, as in Refs. [5,3], to make the optimization problem linear, the sensor tasking and information cost are decoupled. In this paper, we choose to generate a suboptimal solution to the exact nonlinear optimization problem. We describe the greedy search approach in the next section.

V. Greedy Search

The optimal sensor-tasking problem is known to be NP hard and is essentially a combinatorial problem. This optimization problem has $N_o \times N_s' \times N_T$ variables, and hence $2^{N_o \times N_s' \times N_T}$ possible configurations. This is computationally prohibitive for practical applications. In this section, we address this computational complexity by leveraging the submodular property of the MI to design greedy approaches. We first start by providing a brief review of submodular functions, and then we show that the MI in Eq. (14) [and Eq. (15)] is in fact submodular and nondecreasing. We then state the suboptimal bounds for the greedy approaches used to maximize submodular functions.

Definition 1, submodular functions (Nemhauser et al. [18]): Given a finite set $E = \{s_1, s_2, \ldots, s_N\}$, a real-valued function $f(S)$ for $S \subset E$ is submodular if it satisfies any one of the following equivalent conditions:

1) $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subset E$.

2) $\Delta_e(A) \geq \Delta_e(B)$ for all $A \subset B \subset E$ and $e \in E - B$ where, for any S, $\Delta_e(S) = f(S \cup \{e\}) - f(S)$.

Definition 2, nondecreasing functions (Nemhauser et al. [18]): A real-valued function $f(S)$ for $S \subset E$ is nondecreasing if it satisfies any one of the following equivalent conditions:

1) $f(B) \geq f(A)$ for all $A \subset B \subset E$.

2) $\Delta_e(S) \geq 0$ for all $S \subset E$ and $e \in E - S$ where, for any S, $\Delta_e(S) = f(S \cup \{e\}) - f(S)$.

Submodular functions are especially interesting because they behave as concave functions. Together with the nondecreasing property, these functions can be maximized by simple greedy search approaches. In this regard, we first show that the MI, in Eq. (14), is submodular and nondecreasing. We define $f_{\mathrm{MI}}^{(i)}(A) = I\left(\boldsymbol{X}_{(0:N_T)}^{(i)}; \boldsymbol{Y}_A^{(i)}\right)$ for a single object, where $A \subset \mathcal{E}$ is the set of active or selected sensor configurations (or, equivalently, the set of indices $s_{j,k}$). Because every combination of sensor configurations is considered a new sensor, selecting a configuration corresponds to selecting the corresponding sensor in set $\mathcal{E}$. When a sensor configuration is selected (i.e., $s_{j,k} \in A$), we set the corresponding binary variables $v_k^{(i,j')} = 1$ when the mean of the ith object (predicted state) at time k is in the FOV of the jth sensor with configuration $j' \equiv s_{j,k}$, and it is $v_k^{(i,j')} = 0$ otherwise. By $j' \equiv s_{j,k}$, we simply refer to the process of taking the corresponding sensor in $\mathcal{E}$ with index j'. We then compute the mutual information in expression (16) [or Eq. (25)] for active configuration set A. Also, we take $f_{\mathrm{MI}}^{(i)}(\varnothing) = 0$ because, when no sensors are active, the corresponding mutual information is taken to be zero. In Lemmas 1 and 2, we first show that this function $f_{\mathrm{MI}}^{(i)}(A)$ is a submodular nondecreasing function. We use the following identities (from Ref. [4]) for mutual information, given any three random variables $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$:

$$I(\boldsymbol{X};\boldsymbol{Y}) = H(\boldsymbol{X}) - H(\boldsymbol{X}|\boldsymbol{Y}) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X})$$

$$H(\boldsymbol{Y},\boldsymbol{X}) = H(\boldsymbol{Y}|\boldsymbol{X}) + H(\boldsymbol{X}) = H(\boldsymbol{X}|\boldsymbol{Y}) + H(\boldsymbol{Y})$$

$H(\boldsymbol{X}) \geq 0$, $H(\boldsymbol{X}) \geq H(\boldsymbol{X}|\boldsymbol{Y})$, and finally $H(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{Z}) = H(\boldsymbol{Y}|\boldsymbol{Z})$ when $\boldsymbol{Y}$ is independent of $\boldsymbol{X}$ when conditioned on $\boldsymbol{Z}$. Here, $H(\boldsymbol{X})$ is the entropy (Shannon entropy) ([4] p. 224). For the following two lemmas, we drop the superscript indicating the object index and the subscript indicating the time index for the sake of readability. Note that $f(A)$ should be understood as the mutual information for a single object using the sensors in set A.

Lemma 1: $f_{\mathrm{MI}}(A)$ is nondecreasing.

Proof: Let B be A set of sensors (indices) such that $A \cap B = \varnothing$, and LET $\boldsymbol{Y}_B$ be the corresponding measurement variables. As the random vectors $\boldsymbol{x}$ in Eq. (1) form a Markov process, the random variables in $\boldsymbol{Y}_B$ (measurement variables) are mutually independent when conditioned on $\boldsymbol{X}$ (state variables):

$$\begin{aligned} f_{\mathrm{MI}}(A \cup B) &= I(\boldsymbol{X};\boldsymbol{Y}_A,\boldsymbol{Y}_B) = H(\boldsymbol{Y}_A,\boldsymbol{Y}_B) - H(\boldsymbol{Y}_A,\boldsymbol{Y}_B|\boldsymbol{X}) \\ &= H(\boldsymbol{Y}_A) + H(\boldsymbol{Y}_B|\boldsymbol{Y}_A) - H(\boldsymbol{Y}_A|\boldsymbol{X}) - H(\boldsymbol{Y}_B|\boldsymbol{X}) \end{aligned} \tag{40}$$

$$\begin{aligned} &= H(\boldsymbol{Y}_A) - H(\boldsymbol{Y}_A|\boldsymbol{X}) + H(\boldsymbol{Y}_B|\boldsymbol{Y}_A) - H(\boldsymbol{Y}_B|\boldsymbol{X}) \geq H(\boldsymbol{Y}_A) \\ &\quad - H(\boldsymbol{Y}_A|\boldsymbol{X}) = I(\boldsymbol{X};\boldsymbol{Y}_A) = f_{\mathrm{MI}}(A) \end{aligned} \tag{41}$$

The last inequality is given by considering that $\boldsymbol{Y}_B$ is conditionally independent of $\boldsymbol{Y}_A$, given $\boldsymbol{X}$ as

$$H(\boldsymbol{Y}_B|\boldsymbol{Y}_A) - H(\boldsymbol{Y}_B|\boldsymbol{X}) = H(\boldsymbol{Y}_B|\boldsymbol{Y}_A) - H(\boldsymbol{Y}_B|\boldsymbol{Y}_A,\boldsymbol{X}) \geq 0$$

Here, the inequality is obtained by the identity that conditioning always reduces entropy ([4] p. 237). Now, by taking $C = A \cup B$, we have $f_{\mathrm{MI}}(C) \geq f_{\mathrm{MI}}(A)$, where $A \subset C$. Hence, by Definition 2, f_{MI} is nondecreasing. □

Lemma 1 is intuitive because adding more measurements can never decrease the information. This is a simple and yet important observation for maximizing the MI cost. In the next lemma, we show that the MI cost is submodular. Submodular functions are functions that exhibit the property of diminishing returns; i.e., it is better (more gain in information) to add a sensor to a smaller set of sensors than to a larger set. This property is often used in devising greedy heuristics for submodular functions.

Lemma 2: $f_{\mathrm{MI}}(A)$ is submodular.

Proof: Because the expressions in Eqs. (14) and (15) are equivalent, we use the expression in Eq. (15) for the MI to show that it is submodular. First, from the process and measurement models in Eqs. (1) and (2), we observe that the measurement variables $\boldsymbol{y}$ are mutually independent when conditioned on all the state variables $\boldsymbol{x}$ up to time N_T; i.e.,

$$\begin{aligned} &p(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T|\boldsymbol{X}_{(0:N_T)}) \\ &\quad = p(\boldsymbol{y}_1|\boldsymbol{X}_{(0:N_T)})p(\boldsymbol{y}_2|\boldsymbol{X}_{(0:N_T)}) \ldots p(\boldsymbol{y}_T|\boldsymbol{X}_{(0:N_T)}) \end{aligned}$$

Hence, the corresponding covariance for a set of measurements $\boldsymbol{Y}_A$ conditioned on $\boldsymbol{X}$, for PDF $p(\boldsymbol{Y}_A|\boldsymbol{X})$, is given as

$$R_A = \mathrm{blkdiag}(R_{a_1}, R_{a_2}, \ldots, R_{a_{|A|}})$$

where $a_1, a_2, \ldots, a_{|A|}$ are the indices in set A, R_a is the noise covariance for the sensor with index a in set $\mathcal{E}$, and the function blkdiag concatenates the block matrices along the main diagonal of a matrix. Alternatively, the joint noise covariance matrix for a set of sensors with indices in set A is just the concatenation of their corresponding noise covariance matrices along the main diagonal. We observe that, for a given joint PDF,

$$p(\boldsymbol{X},\boldsymbol{Y}_A) = \mathcal{N}\left([\boldsymbol{X};\boldsymbol{Y}_A]^T : [\hat{\boldsymbol{X}};\hat{\boldsymbol{Y}}_A]^T, \begin{bmatrix} P & \Gamma_A \\ \Gamma_A^T & \Xi_A \end{bmatrix}\right)$$

the covariance for the conditional PDF $p(\boldsymbol{X}|\boldsymbol{Y}_A)$ is given as $P - \Gamma_A \Xi_A^{-1} \Gamma_A^T$ and the covariance for the conditional PDF $p(\boldsymbol{Y}_A|\boldsymbol{X})$ is given as $\Xi_A - \Gamma_A^T P^{-1} \Gamma_A$ (Ref. [32] fact 8.1.3). Using this identity, the denominator in Eq. (15) when using $\boldsymbol{Y}_A$

measurements is nothing but the covariance of the likelihood PDF $p(\boldsymbol{Y}_A|\boldsymbol{X})$ with covariance R_A. The determinant of this covariance is simply given as

$$|R_A| = \prod_{a\in A}|R_a|$$

To show that $f(A)$ is submodular, we need to show that $\Delta_e(A) \geq \Delta_e(B)$ for all $A \subset B \subset \mathcal{E}$ and $e \in \mathcal{E} - B$. From Definition 1, we have $\Delta_e(A) = f(A \cup \{e\}) - f(A)$. Using the expression in Eq. (15), the numerator for $f(A \cup \{e\})$ is the measurement covariance computed by taking all the sensors in $A \cup \{e\}$; and the denominator is the constructed by concatenating the corresponding noise covariances:

$$\begin{aligned}\Delta_e(A) &= \frac{1}{2}\log\frac{\left|\begin{bmatrix}\Sigma_A & \Sigma_{Ae}\\ \Sigma_{Ae}^T & \Sigma_e\end{bmatrix}\right|}{|R_A||R_e|} - \frac{1}{2}\log\frac{|\Sigma_A|}{|R_A|}\\ &= \log\frac{|\Sigma_e - \Sigma_{Ae}^T\Sigma_S^{-1}\Sigma_{Ae}|}{|R_e|}\end{aligned} \tag{42}$$

$$\begin{aligned}\Delta_e(B) &= \frac{1}{2}\log\frac{\left|\begin{bmatrix}\Sigma_B & \Sigma_{Be}\\ \Sigma_{Be}^T & \Sigma_e\end{bmatrix}\right|}{|R_B||R_e|} - \frac{1}{2}\log\frac{|\Sigma_B|}{|R_B|}\\ &= \log\frac{|\Sigma_e - \Sigma_{Be}^T\Sigma_B^{-1}\Sigma_{Be}|}{|R_e|}\end{aligned} \tag{43}$$

where the identity for the determinant for the block matrix from Remark 2 (Appendix B) is used: Σ_A is the measurement covariance computed for sensors in set A, and Σ_e is the sensor measurement covariance computed for sensor e: Σ_{Ae} is the cross-diagonal block matrix for the joint measurement covariance for sensors in $A \cup \{e\}$. Matrices Σ_B and Σ_{Be} are similarly defined. It then suffices to show that

$$\Sigma_{Ae}^T\Sigma_S^{-1}\Sigma_{Ae} \leq \Sigma_{Be}^T\Sigma_B^{-1}\Sigma_{Be}$$

As $A \subset B$, we take B as the union of disjoint sets as $B = C \cup A$; we can then partition Σ_B and Σ_{Be} as

$$\Sigma_B = \begin{bmatrix}\Sigma_C & \Sigma_{CA}\\ \Sigma_{CA}^T & \Sigma_A\end{bmatrix}, \qquad \Sigma_{Be} = \begin{bmatrix}\Sigma_{Ce}\\ \Sigma_{Ae}\end{bmatrix} \tag{44}$$

Here, Σ_{CA} is the cross-diagonal block matrix of Σ_B

Now, consider

$$\begin{aligned}&\Sigma_{Be}^T\Sigma_T^{-1}\Sigma_{Be} - \Sigma_{Ae}^T\Sigma_A^{-1}\Sigma_{Ae}\\ &\quad= \begin{bmatrix}\Sigma_{Ce}^T & \Sigma_{Ae}^T\end{bmatrix}\begin{bmatrix}\Sigma_C & \Sigma_{CA}\\ \Sigma_{CA}^T & \Sigma_A\end{bmatrix}^{-1}\begin{bmatrix}\Sigma_{Ce}\\ \Sigma_{Ae}\end{bmatrix}\\ &\qquad- \begin{bmatrix}\Sigma_{Ce}^T & \Sigma_{Ae}^T\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & \Sigma_A^{-1}\end{bmatrix}\begin{bmatrix}\Sigma_{Ce}\\ \Sigma_{Ae}\end{bmatrix}\end{aligned} \tag{45}$$

$$= \begin{bmatrix}W_1^{-1} & -W_1^{-1}\Sigma_{CA}\Sigma_A^{-1}\\ -\Sigma_A^{-1}\Sigma_{CA}^T W_1^{-1} & \Sigma_A^{-1}\Sigma_{CA}^T W_1^{-1}\Sigma_{CA}\Sigma_A^{-1}\end{bmatrix} = \begin{bmatrix}W_1^{-1} & K\\ K & K^T W_1 K\end{bmatrix} \tag{46}$$

where Remark 3 (Appendix B) was used in the final expression with

$$K = -W_1^{-1}\Sigma_{CA}\Sigma_A^{-1}$$

Using the Schur compliment (Ref. [32] fact 9.1.5), we can observe that

$$\begin{bmatrix}W_1^{-1} & K\\ K & K^T W_1 K\end{bmatrix}$$

is positive semidefinite. Hence, we have

$$\Sigma_{Be}^T\Sigma_B^{-1}\Sigma_{Be} - \Sigma_{Ae}^T\Sigma_A^{-1}\Sigma_{Ae} \geq 0$$

or

$$\Sigma_{Ae}^T\Sigma_A^{-1}\Sigma_{Ae} \leq \Sigma_{Be}^T\Sigma_B^{-1}\Sigma_{Be}$$

This leads to

$$\Sigma_e - \Sigma_{Ae}^T\Sigma_A^{-1}\Sigma_{Ae} \geq \Sigma_e - \Sigma_{Be}^T\Sigma_B^{-1}\Sigma_{Be}$$

Using the identity for positive definite matrices P and Q, if $0 \leq P \leq Q$, then $|P| \leq |Q|$ (Ref. [33] corollary 8.4.10); it follows that

$$|\Sigma_e - \Sigma_{Ae}^T\Sigma_A^{-1}\Sigma_{Ae}| \geq |\Sigma_e - \Sigma_{Be}^T\Sigma_B^{-1}\Sigma_{Be}|$$

and hence $\Delta_e(A) \geq \Delta_e(B)$.

Alternatively, in the case when sets A and B are disjoint ($A \cap B = \varnothing$), the proof is simpler. Here, we need to show $f_{\mathrm{MI}}(A) + f_{\mathrm{MI}}(B) \geq f_{\mathrm{MI}}(A \cup B)$:

$$f_{\mathrm{MI}}(A \cup B) = I(\boldsymbol{X}; \boldsymbol{Y}_{A\cup B}) = \frac{1}{2}\log\left\{\frac{|\Sigma_{A\cup B}|}{|R_{A\cup B}|}\right\} = \frac{1}{2}\log\left\{\frac{|\Sigma_{A\cup B}|}{|R_A||R_B|}\right\} \tag{47}$$

where $|R_{A\cup B}| = |R_A||R_B|$ because $R_{A\cup B}$ is a block matrix with zero offdiagonal matrices. Express $\Sigma_{A\cup B}$ as a block matrix

$$\Sigma_{A\cup B} = \begin{bmatrix}\Sigma_A & \Sigma_{AB}\\ \Sigma_{AB}^T & \Sigma_B\end{bmatrix}$$

where Σ_{AB} is the cross-diagonal subblock matrix. Using Fisher's inequality (Ref. [41] theorem 7.8.5) for the determinant of a positive definite matrix, we have

$$|\Sigma_{A\cup B}| \leq |\Sigma_A||\Sigma_B|$$

The expression in Eq. (47) becomes

$$\begin{aligned}f_{\mathrm{MI}}(A \cup B) &= \frac{1}{2}\log\left\{\frac{|\Sigma_{A\cup B}|}{|R_A||R_B|}\right\} \leq \frac{1}{2}\log\left\{\frac{|\Sigma_A||\Sigma_B|}{|R_A||R_B|}\right\}\\ &= \frac{1}{2}\log\left\{\frac{|\Sigma_A|}{|R_A|}\right\} + \frac{1}{2}\log\left\{\frac{|\Sigma_B|}{|R_B|}\right\} = f_{\mathrm{MI}}(A) + f_{\mathrm{MI}}(B)\end{aligned} \tag{48}$$

□

We have seen that the MI for each object is submodular and nondecreasing. It is then required to show that the total MI for independent objects in expression (9) is also submodular and nondecreasing. This is shown in the next proposition.

Proposition 1: Assume the objects to be independent; then, the total MI for all objects is submodular and nondecreasing.

Proof: As the MI for a set of independent objects is the sum of the MI for each object, we represent the total MI as

$$f_{\mathrm{MI}}(A) = \sum_{i=1}^{N_o} f_{\mathrm{MI}}^{(i)}(A)$$

Because each $f_{\mathrm{MI}}^{(i)}(A)$ is submodular and nondecreasing, we only need to show that the sum of the submodular functions is submodular and nondecreasing. We show this result only for the first two objects because this result can then be applied repeatedly to all the objects. Let $f_{\mathrm{MI}}(A) = f_{\mathrm{MI}}^{(1)}(A) + f_{\mathrm{MI}}^{(2)}(A)$; then,

$$\begin{aligned} f_{\mathrm{MI}}(A) + f_{\mathrm{MI}}(B) &= f_{\mathrm{MI}}^{(1)}(A) + f_{\mathrm{MI}}^{(2)}(A) + f_{\mathrm{MI}}^{(1)}(B) + f_{\mathrm{MI}}^{(2)}(B) \\ &= (f_{\mathrm{MI}}^{(1)}(A) + f_{\mathrm{MI}}^{(1)}(B)) + (f_{\mathrm{MI}}^{(2)}(A) + f_{\mathrm{MI}}^{(2)}(B)) \\ &\geq (f_{\mathrm{MI}}^{(1)}(A \cup B) + f_{\mathrm{MI}}^{(1)}(A \cap B)) \\ &\quad + (f_{\mathrm{MI}}^{(2)}(A \cup B) + f_{\mathrm{MI}}^{(2)}(A \cap B)) \\ &= (f_{\mathrm{MI}}^{(1)}(A \cup B) + f_{\mathrm{MI}}^{(2)}(A \cup B)) \\ &\quad + (f_{\mathrm{MI}}^{(1)}(A \cap B) + f_{\mathrm{MI}}^{(2)}(A \cap B)) \\ &= f_{\mathrm{MI}}(A \cup B) + f_{\mathrm{MI}}(A \cap B) \end{aligned}$$

Hence, $f_{\mathrm{MI}}(A)$ is a submodular set function. The nondecreasing property of $f_{\mathrm{MI}}(A) = f_{\mathrm{MI}}^{(1)}(A) + f_{\mathrm{MI}}^{(2)}(A)$ can also be shown similarly. □

Now that we have shown the MI to be a submodular set function and nondecreasing, we can leverage interesting results for maximizing such a function using greedy approaches that are known to achieve near-optimal solutions [18,42,43] and that are within a constant factor of the optimal solution. This greedy selection can significantly reduce the computational costs. We first review these results from the literature and adapt these approaches to devise suboptimal algorithms for tasking sensors. The greedy heuristic considered is [18]

$$\mathcal{P}^G = \mathcal{P}^G \cup \arg\max_{e \in \mathcal{E} - \mathcal{P}^G} \{\Delta_e(\mathcal{P}^G)\} \tag{49}$$

where $\mathcal{P}^G$ is the solution generated by the greedy heuristic in Eq. (49). The greedy approach here can start from $\mathcal{P}^G = \varnothing$.

Theorem 1 (Nemhauser et al. [18]): For the optimization problem,

$$\max_A \{f(A) : |A| \leq n\}$$

where f is submodular, nonnegative, and nondecreasing; and $f(\varnothing) = 0$. The greedy heuristic algorithm in Eq. (49) achieves a solution $\mathcal{P}^G$ such that

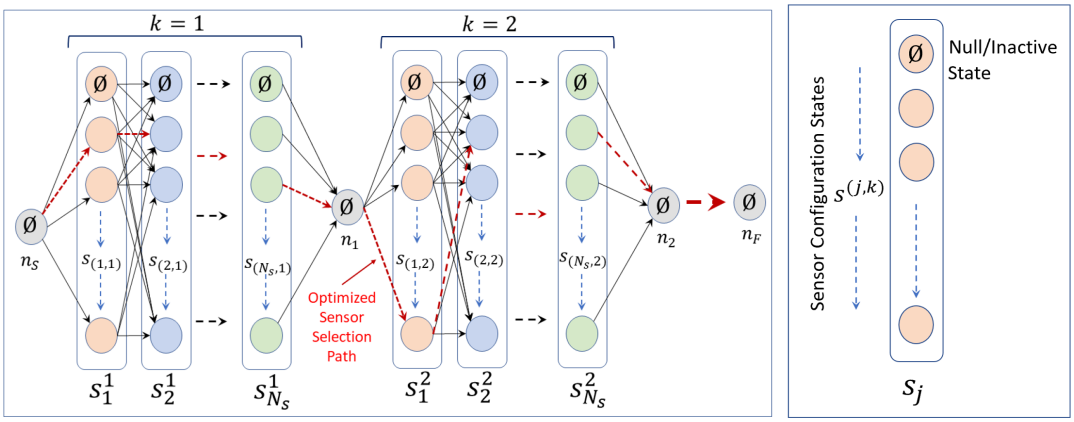
$$\frac{f(\mathcal{P}^G)}{f(\mathcal{P}^*)} \geq \left(1 - \left(\frac{n-1}{n}\right)^n\right) \geq 1 - \frac{1}{e}$$

where $\mathcal{P}^*$ is the optimal solution, and n is the cardinality constraint for sets $\mathcal{P}^*$ and $\mathcal{P}^G$.

This theorem is an important result for which we need to only select K sensors. By using the greedy heuristic in Eq. (49), the solution obtained is close to the optimal solution and within a bound of $1 - (1/e)$. In Ref. [18], the authors also developed an optimality bound with matroid constraints on the selected set $\mathcal{P}^G$. Unfortunately, for a general set constraints as in Eq. (25), it is difficult to obtain optimality bounds. In this work, we consider a special case where, for every sensor, only one configuration can be active at any given time step. This is typically the case in many sensor-tasking problems. For example, only one orientation or location for the sensor is possible at any given time step.

A. Sensor Selection with Single Configuration Constraint

The specific problem we consider in this paper is about selecting one configuration for each sensor at each time step. As discussed in Sec. IV, each operating mode/configuration for a sensor is considered to be a new sensor. The problem is then to just select or make active one sensor among this group of sensors. We first represent the sensor selection problem as a directed acyclic graph (DAG), as shown in Fig. 2a. The nodes n_S and n_F represent the start and end of the DAG. It can be observed in Fig. 2a that the directed paths start from nodes n_S to n_F. Figure 2b shows the sensor configuration block $S_j^{(k)}$ for a single sensor j at time k, and $s_{(j,k)}$ is the index corresponding to the node in the column. A null configuration is also included as the top node for each sensor block. This node helps in switching off the sensor. The sensor configuration blocks from nodes n_S to n_1 represent all the sensor configurations for all sensors at a time step of $k = 1$. The nodes in one sensor configuration block are connected to all the nodes in the next sensor block. These connections form the directed edges for the DAG. Every walk $\mathcal{P}$ from n_S to n_F will select exactly one configuration for every time step and every sensor. The objective is to find the optimal path or walk $\mathcal{P}^*$ along the graph from starting node n_S to the final node n_F that maximizes the MI. The nodes $n_S, n_1, n_2, \ldots, n_T, n_F$ represent empty nodes with no sensors. Hence, traversing them has no effect on the total MI. These nodes are just used to demarcate the sensor configurations for each time step. Note that the MI function is not additive along the nodes of the graph and has to be evaluated for a given path $\mathcal{P}$ as $f_{\mathrm{MI}}(\mathcal{P})$. Here, the weight of each edge is taken as one. In a more general setting, the weight of the edges can indicate the cost of using a specific configuration for each sensor. We now can leverage the greedy approach for a directed graph from the following work from Ref. [19], where the problem is referred as the submodular orientation problem (SOP). A polynomial time algorithm for the SOP is given in Algorithm 1.



a) Sensor selection DAG b) Sensor states

Fig. 2 Directed graph sensor selection.

Algorithm 1: $RG(a, b, B, \mathcal{T}, N)$ [19]

Result: $\mathcal{P}^G$
if $L(a, b) > B$, return infeasible;
$\mathcal{P}^G \leftarrow \{a, b\}$;
if $N = 0$, return $\mathcal{P}^G$;
$p = \mathcal{P}_1^G \cup \mathcal{P}_2^G$, $m = \Delta_p(\mathcal{T})$;
for each $e \in \mathcal{E}$, **do**
  for $1 \leq B_1 \leq B$, do
    $\mathcal{P}_1^G \leftarrow RG(a, e, B_1, \mathcal{T}, N-1)$;
    if $\mathcal{P}_1^G$ is infeasible, **then**
      continue;
    else
      $\mathcal{P}_2^G \leftarrow RG(e, b, B - B_1, \mathcal{T} \cup \mathcal{P}_1^G, N-1)$;
    end
    if $\mathcal{P}_2^G$ is infeasible continue;
    $p = \mathcal{P}_1^G \cup \mathcal{P}_2^G$;
    if $\Delta_p(\mathcal{T}) > m$, **then**
      $\mathcal{P}^G \leftarrow \mathcal{P}_1^G \cup \mathcal{P}_2^G$;
      $m \leftarrow \Delta_p(\mathcal{T})$
    end
  end
end
return $\mathcal{P}^G$.

$RG(a, b, B, \mathcal{T}, N)$ is a recursive greedy algorithm and achieves

$$f(\mathcal{P}^G) \geq \frac{f(\mathcal{P}^*)}{[1 + \log_2(n)]}$$

approximation, where $\mathcal{P}^*$ is the optimal path and n is the number of nodes in path $\mathcal{P}^*$. Procedure RG of Algorithm 1 finds the suboptimal greedy path from node a to node b for the DAG. Argument B is the budget, and $\mathcal{T}$ is the path traversed so far. Here, it is required to have the recursion depth of $N \geq [1 + \log_2(n)]$. As we take each edge with a weight of one, the budget B is the length of the path from n_S to n_F in Fig. 2a; i.e., $n = N_S \times T$. The length of the arc $L(a, b)$ from node a to node b is the minimum number of steps to walk along the graph from a to b. If there is no directed walk in the graph between two nodes, the length is taken as infinity (or infeasible). The algorithm essentially tries to approximately guess the midnode and the associated budget for the optimal path $\mathcal{P}^*$, and then it recursively finds the left and right paths to this midnode in a greedy manner. The algorithm is initiated as $RG(n_S, n_F, n, \emptyset, N)$. In Ref. [19], the authors further improved the algorithm using a binary search for the budget and achieved an $\mathcal{O}(\log_2(K))$ approximation. It can be observed that the graph structure considered in this paper, as shown in Fig. 2a, has no cycles; hence, the runtime is typically much faster in real-time implementation. As we are required to generate sensor tasks or schedule in near-real time, we extend the sequential approaches described in Secs. IV.B and IV.C that, in turn, can be solved by the RG algorithm by simply modifying the cost function.

1. *Sequential in Time*

Similar to the sequential in time strategy in Sec. IV.B, the optimization problem is solved sequentially for each time step while holding the sensor tasks in previous time steps as constants. The DAG for the kth time step is illustrated in Fig. 3a. The objective is to find a path $\mathcal{P}$ in this graph from n_S to n_F so as to maximize the modified MI cost

$$\mathcal{P}_k^G = \arg\max_{\mathcal{P}} f_{\mathrm{MI}}^{(1:k-1)}(\mathcal{P}) = I(\boldsymbol{X}; \{\mathcal{P} \cup \mathcal{P}_{1:k-1}^G\})$$

where the fixed set of sensor configurations up to the $k-1$th time step is $\mathcal{P}_{1:k-1}^G = \mathcal{P}_1^G \cup \mathcal{P}_2^G \cup \ldots \cup \mathcal{P}_{k-1}^G$.

2. *Sequential in Sensors*

Similar to the greedy in time strategy, the greedy in sensors approach tasks each sensor individually. When the number of sensors are higher than the number of time steps, it might then be desirable to iterate through the sensors in a greedy way rather than in time. Assuming that the sensors are reindexed according to a user-defined priority order (for example, according to the number of objects in the FOV), the sensor-tasking problem begins by sequentially optimizing each sensor over a period of time while accounting for the all the previous sensor schedules. The case for the jth sensor is depicted in Fig. 3b. The optimization problem is

$$\mathcal{P}_j^G = \arg\max_{\mathcal{P}} f_{\mathrm{MI}}^{(1:j-1)}(\mathcal{P})$$

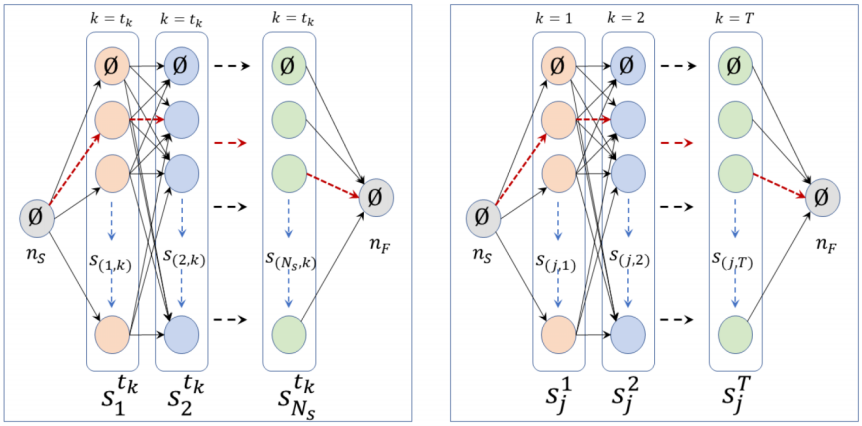
where the objective function is

$$f_{\mathrm{MI}}^{(1:j-1)}(\mathcal{P}) = I(\boldsymbol{X}; \{\mathcal{P} \cup \mathcal{P}_{1:j-1}^G\})$$

where the fixed set of sensor configurations up to sensor j is $\mathcal{P}_{1:j-1}^G = \mathcal{P}_1^G \cup \mathcal{P}_2^G \cup \ldots \cup \mathcal{P}_{j-1}^G$.

B. Receding Horizon

With motivation from the practical advantages of model predictive control [44], the sensor-tasking problem can also be solved in a receding-horizon way. In this approximation, the decision variables are only solved for a time window of $[k, k + N_K]$, where N_K is the user-defined number of time steps. Once solved for $[k, k + N_K]$, the sensor schedule is applied for the time interval, the measurements collected are processed, and a new optimization problem is solved for



a) Sequential in time **b) Sequential in sensors**

Fig. 3 Tasking graph for the sequential selection.

the next time interval. Such an approximation can considerably reduce the dimension of the optimization problem. As noted in Refs. [3,13], multistep sensor tasking is beneficial but can also be disadvantageous when the object process models are not well known or when the targets are maneuvering. Hence, we use a receding-horizon approach to task sensors over a shorter time window and then re-solve the problem at the end of the window. The measurements collected during the time window are fully processed before the new sensor-tasking problem is solved for the next time window.

VI. Numerical Simulations

In this section, we use two applications to illustrate the joint multistep mutual information-based sensor-tasking approaches proposed in the previous section paper. The first application is about the two-dimensional (2-D) tracking of unmanned aerial vehicles using simple range-bearing sensors with limited FOVs. The second example is a three-dimensional (3-D) problem of tracking multiple satellites with ground-based sensors with limited FOVs. In both of these examples, the limited FOV sensors can actively be reoriented to improve the tracking performance.

A. Example 1: UAV Tracking

Figure 4 depicts the scenario where 20 UAVs are tracked by two range-bearing sensors in a 2-D planar region. This example can also depict the tracking of ground-based robots using a camera or laser range finder. The sensors considered have a limited FOV (modeled as a triangle in 2-D) given by vertex angles $\pi/9$ and $(pi/18)$. There is no limit on the depth or range of the sensor's FOV, even though Fig. 4 shows a closed triangular FOV only for illustrative purposes. The sensors are located at $[-20, 30]$ km and $[30, 20]$ km, respectively. The first sensor noise covariance is $R_1 = \mathrm{diag}([(100\ \mathrm{m})^2, (2\pi/180\ \mathrm{rad})^2])$, whereas the second sensor has a noise covariance of $R_2 = \mathrm{diag}([(50\ \mathrm{m})^2, (0.5\pi/180\ \mathrm{rad})^2])$. Although they have limited FOVs, these sensors can be actively reoriented to better track the targets. The total simulation duration is 1500 s with a 5 s time step. The UAVs (or ground robots) perform a turning motion with an unknown turn rate. The kinematics of the turning motion is modeled by the set of nonlinear equations called coordinated turn (CT) equations, which are typically modeled as a Dubins car, as described in Ref. ([45] p. 467). The CT model is characterized by constant speed and a constant turn rate. The turn rate Ω is usually unknown, and is hence appended to the state vector, making the model nonlinear. The system equations for the CT model and the corresponding process noise, where the state vector is $\boldsymbol{x} = [x, y, v_x, v_z, \Omega]^T$, are

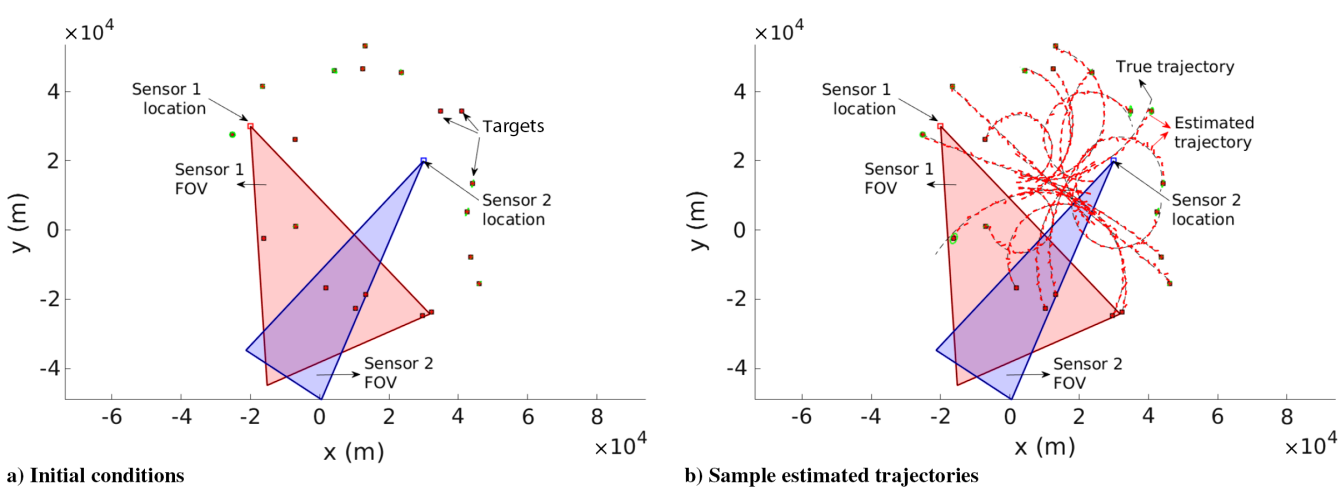
$$\boldsymbol{x}_k = \begin{bmatrix} 1 & \frac{\sin(\Omega\Delta t)}{\Omega} & 0 & -\frac{1-\cos(\Omega\Delta t)}{\Omega} & 0 \\ 0 & \cos(\Omega\Delta t) & 0 & -\sin(\Omega\Delta t) & 0 \\ 0 & \frac{1-\cos(\Omega\Delta t)}{\Omega} & 1 & \frac{\sin(\Omega\Delta t)}{\Omega} & 0 \\ 0 & \sin(\Omega\Delta t) & 0 & \cos(\Omega\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{k-1} + \omega_{k-1},$$

$$Q_{k-1} = 0.16 \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t & 0 \\ 0 & 0 & 0 & 0 & \frac{0.01}{0.16}\Delta t \end{bmatrix} \tag{50}$$

where Δt is the time-step duration. The true initial locations for the targets are randomly generated in the rectangular region given by diagonal vertices at [1000 m, 1000 m] and [25,000 m, 25,000 m]. The corresponding velocities and turn rates are generated in the interval of $[125, 175]$ m/s and $[-1, 1](\pi/180)$ rad/s, respectively. The initial covariance for the filters is taken as

$$P_0 = \mathrm{diag}([(10\ \mathrm{m})^2, (10\ \mathrm{m})^2, (1\ \mathrm{m/s})^2, (1\ \mathrm{m/s})^2, (0.2\ \mathrm{rad/s})^2])$$

The initial mean for the filters is randomly generated from a Gaussian PDF using the true initial condition as the mean and P_0 as the covariance. This process tries to model the scenario in which there is one ground truth and the analyst initializes the filters from historical data or recent measurements that are corrupted with noise. All the filters use the Conjugate Unscented Trasnforms points of 8th order, denoted as CUT8 [20]. The orientation angle for each sensor is discretized using 20 uniformly spaced nodes on $[-\pi, \pi]$. Every orientation of a sensor can be considered to be a new sensor. The objective is then to select the appropriate set of sensors over time. Because only one orientation can be active at any time instant, we use



a) Initial conditions

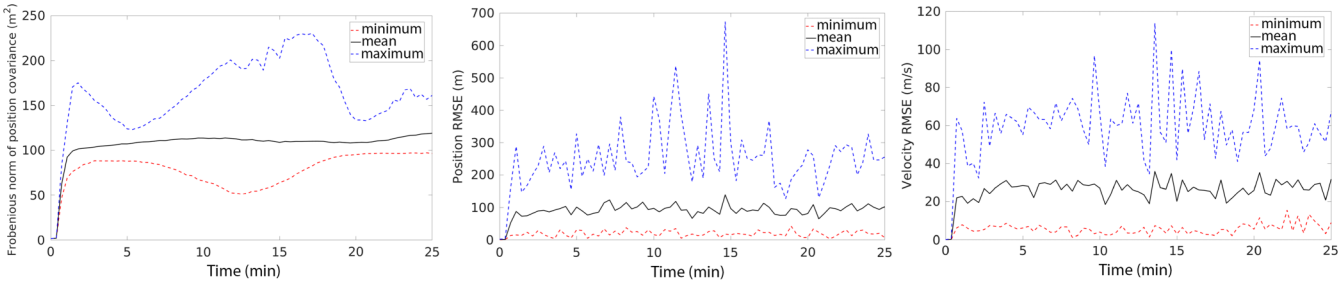
b) Sample estimated trajectories

Fig. 4 Example 1: UAV tracking.

the greedy algorithms developed in Sec. V to pick the suboptimal set of sensors.

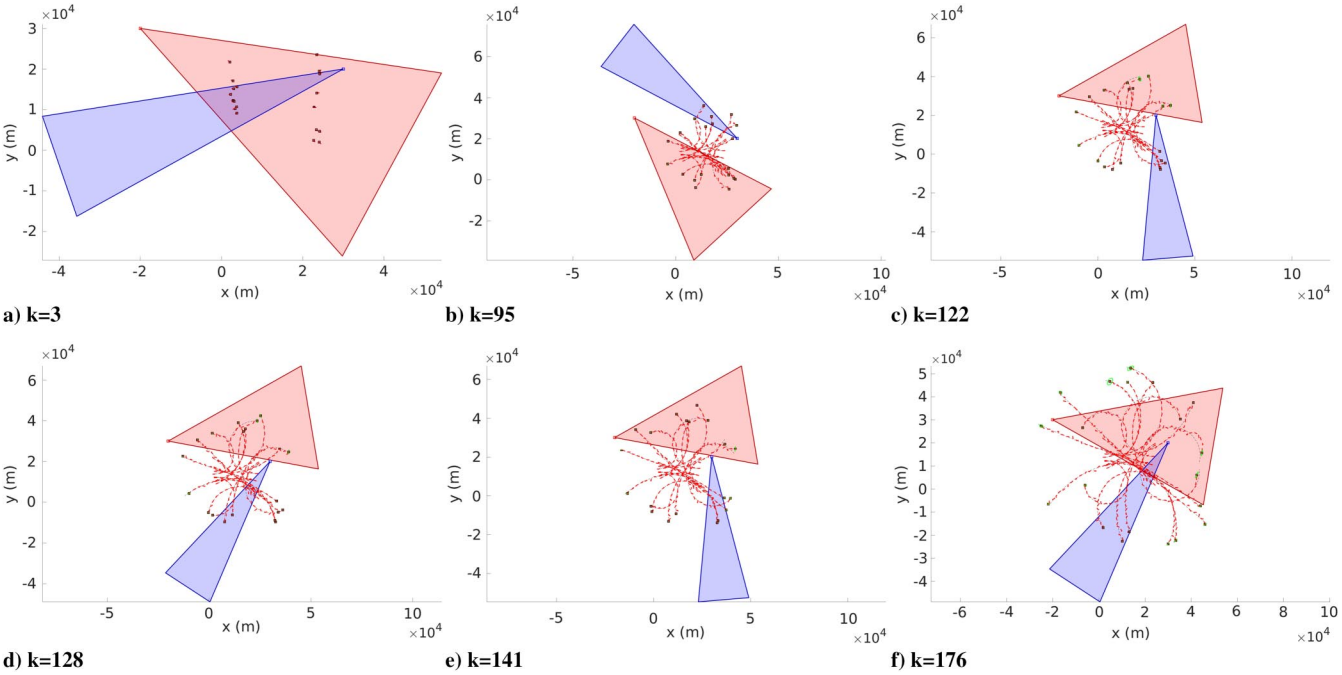
We compare all the sequential algorithms developed in Sec. V. In addition, we also simulate a simple raster scan, where the orientation of sensor linearly traces the nodes with each time step. This tries to model a simple rotating sensor. Furthermore, we also compare our proposed approaches with the approach described in Ref. [46], where the authors proposed to first precompute the information table and then, using an auction algorithm, to select the sensor tasks. Each row of the information table corresponds to a target, and the columns correspond to time steps. The information for each target and for each time is computed by orienting the sensor toward the predicted location of the target. In this way, the entire information table can be constructed. The auction algorithm is simple to implement and proceeds by picking the maximum entries in the table, with the constraint that only one entry per column is selected. This is because the sensor can only have one orientation at any given time step. We label this algorithm as ITAUC in the following simulations and use the mutual information to compute the entries in the information table by selecting the orientation of the sensor closest to the unit vector from the sensor to the target's predicted position. Note that we can also use a binary integer programming solver to select entries in the table that satisfies the constraints that only one entry in each column is selected. In Ref. [5], the authors generated a similar information table for a single–time-step sensor-tasking problem and used a binary integer programming solver to maximize the information. We first show the results of running the filters without considering any FOV constraints in Fig. 5. In these figures, all sensors can see all objects at all time steps. This simulation is only provided as a basis for comparison to the results when FOV constraints are applied and sensors orientations are optimized.

Figure 6 shows snapshots of the UAV tracking simulations at random time steps for the sequential in time greedy solution. It can be seen that the sensor-tasking approach optimizes the orientation of the sensors to track all the objects. In Fig. 7, the sensor tasking is performed over a receding horizon of 15 time steps, as described in Sec. V.A. Note that, in all the simulations, the same ground truth is used. Furthermore, the same measurements are used if the object is within the FOV of the sensor. This provides a means to compare the performance of all the sensor-tasking schemes. The scales of the errors are high in Figs. 5 and 7 because the sensor noise used in the simulation is high. The maximum values in Fig. 7 are high because some objects are occasionally lost when the predicted motion is far away from the true motion. It can be observed that a simple raster scan has poor performance. The sequential in time and sequential in sensors methods, as described in Sec. V, have a better mean performance than the ITAUC. This is because the algorithms in Sec. V compute the joint mutual information over all sensors, objects, and time steps for the sensor configurations. It should be noted that simple heuristics based on precomputing the information tables as in Refs. [5,46] and posing a linear programming problem provide no guarantee that the joint information is maximized over time because, in general, the information is not additive over sensors and time steps. The joint information (MI or KL) is strongly coupled to the sensor-tasking decision variables, and hence cannot be computed before fixing a sensor task/schedule.



a) Frobenius covariance position b) RMSE position c) RMSE velocity

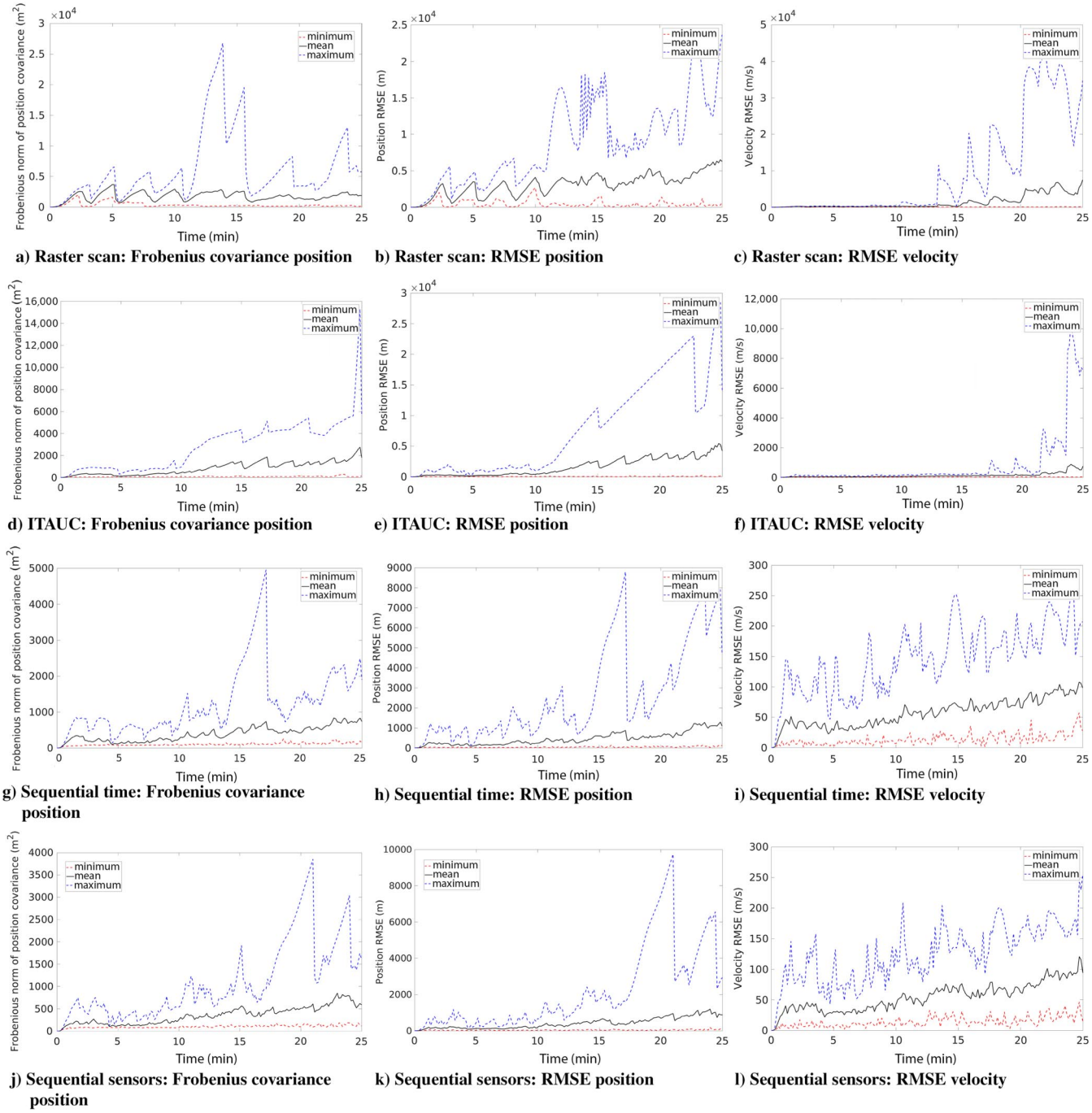
Fig. 5 Example 1: estimation with no FOV for UAV. (RMSE denotes rms error.)



a) k=3 b) k=95 c) k=122

d) k=128 e) k=141 f) k=176

Fig. 6 Example 1: UAV tracking snapshots.



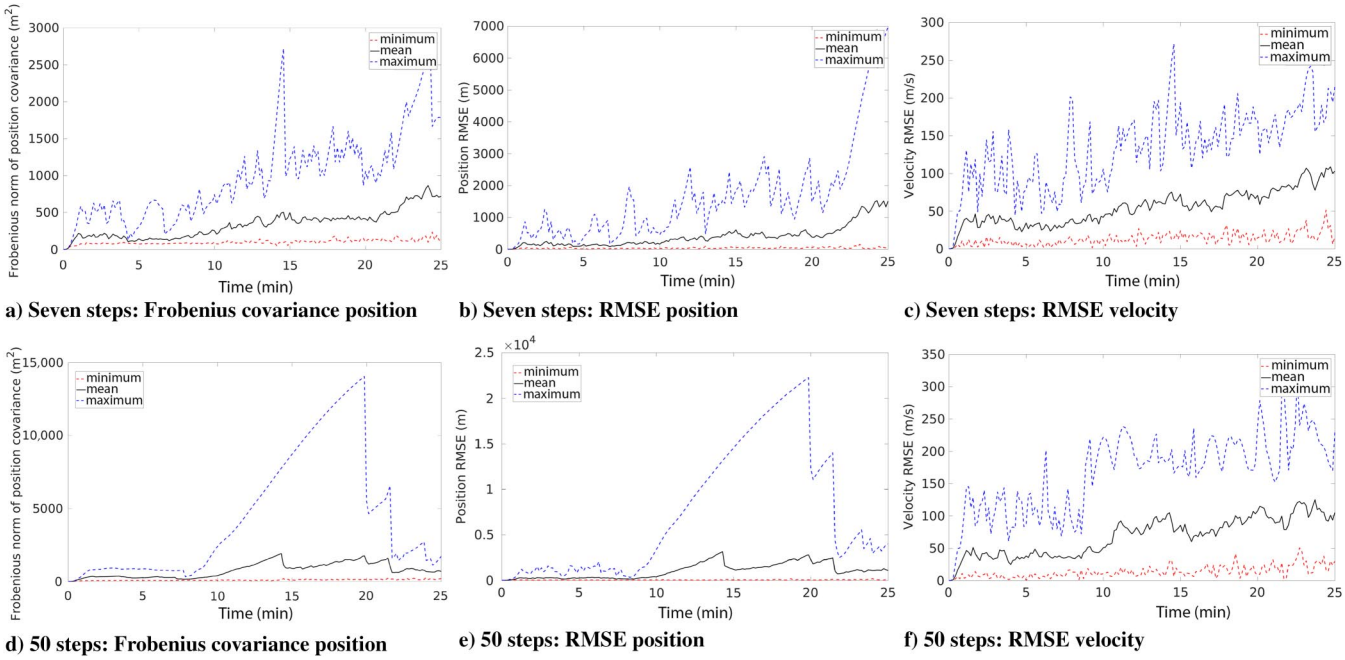
a) Raster scan: Frobenius covariance position
b) Raster scan: RMSE position
c) Raster scan: RMSE velocity
d) ITAUC: Frobenius covariance position
e) ITAUC: RMSE position
f) ITAUC: RMSE velocity
g) Sequential time: Frobenius covariance position
h) Sequential time: RMSE position
i) Sequential time: RMSE velocity
j) Sequential sensors: Frobenius covariance position
k) Sequential sensors: RMSE position
l) Sequential sensors: RMSE velocity

Fig. 7 Example 1: Sensor tasking for UAV.

As noted in Refs. [13,46], longer time periods for solving the sensor tasks can lead to degraded performance. This is because the sensor-tasking methods only use prior states and measurements to predict the motion of the objects, and then they optimize the sensors for this predicted motion. For longer time durations, the accumulated errors in prior data, unmmodeled dynamics and model parameters can lead to large deviations of the predicted states from the true states. Figure 8 describes this scenario, in which it can be observed that a shorter time horizon of seven steps does better than the results for the corresponding sequential in time sensor tasking with 15 time-step windows in Fig. 7. Furthermore, in Fig. 8, by taking a larger receding time horizon of 50 time steps, the performance becomes worse because, once the estimates are farther from the truth, they are essentially lost. In such cases, it is often required to use a raster scan to find the objects again.

B. Example 2: Satellite Tracking Using Greedy Algorithms

In this example, 50 satellites are tracked using two ground-based sensors. We compare the sequential in time approach using the greedy recursive algorithm in Sec. V along with a simple raster scan and the ITAUC algorithm. The schematics for two 3-D sensors is depicted in Fig. 9. Here, each sensor has a maximum FOV denoted by the field of regard (FOR), which represents the limits on the orientation of the sensor. The sensor also has a FOV that is much smaller than the FOR. This FOV can be oriented anywhere within the FOR. Only objects within the FOV are observable by the sensor. Hence, the FOV of the sensor has to be actively oriented within the FOR. To this end, the FOR is partitioned into cells such that the FOVs overlap when oriented toward the centers of adjacent cells. The extent of overlap can be controlled by the resolution of the partition. Although the FOR is represented as a rectangular region in Fig. 9, in the simulations,



a) Seven steps: Frobenius covariance position
b) Seven steps: RMSE position
c) Seven steps: RMSE velocity
d) 50 steps: Frobenius covariance position
e) 50 steps: RMSE position
f) 50 steps: RMSE velocity

Fig. 8 Example 1: sequential in time, comparing receding-horizon time windows.

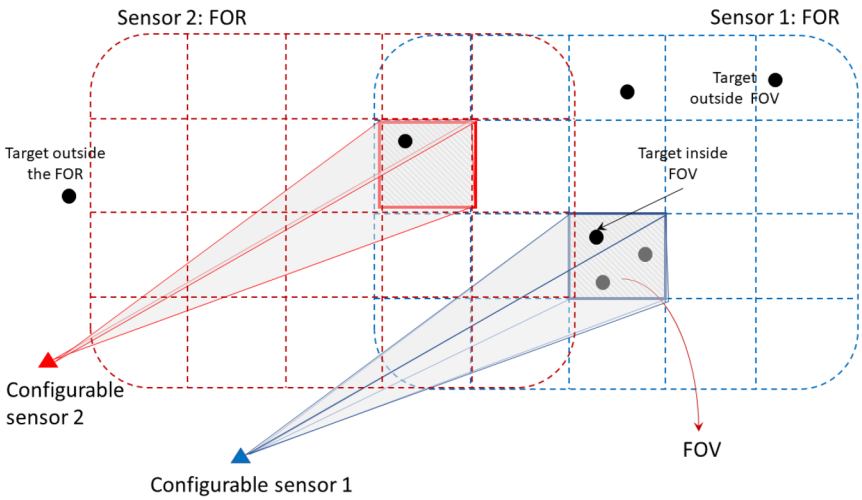


Fig. 9 Configurable 3-D sensor, FOR, and FOV.

we use a conical region for the FOR and the FOV for ease of implementation.

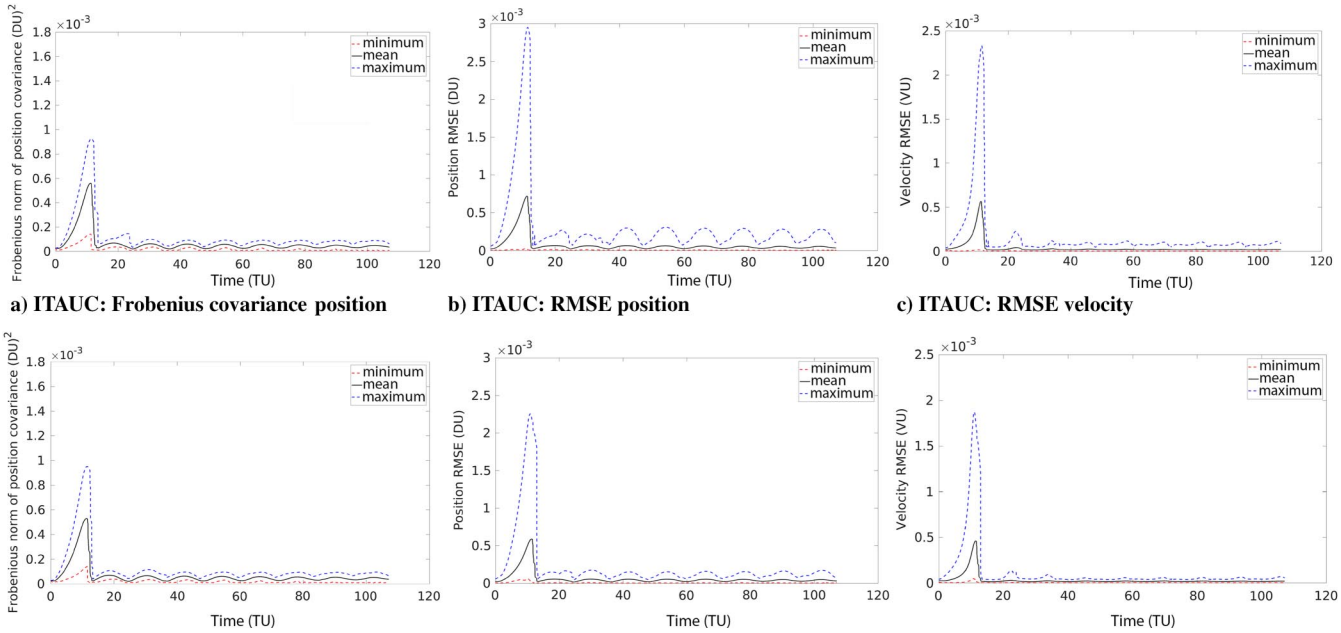
The objects are randomly generated from a uniform distribution over the Keplerian orbital elements $a \in [8{,}000, 10{,}000]$ km, $e \in [0.3, 0.7]$, $i \in [1.3, 1.7]$ rad, $\omega \in [0, 1.22]$ rad, $\Omega \in [0, 0.1]$ rad, and $M_0 \in [0, 0.52]$ rad. The dynamics of each space object is modeled by the two-body dynamics with a J2 perturbation ([47] p. 593). Furthermore, the perigee of every orbit is checked to see if it greater than the radius of the Earth. The initial covariance for each filter is taken as

$$P_{0/0} = \mathrm{diag}(0.01\ \mathrm{km}^2, 0.01\ \mathrm{km}^2, 0.01\ \mathrm{km}^2, 1e^{-8}\ \mathrm{km}^2/\mathrm{s}^2, 1e^{-8}\ \mathrm{km}^2/\mathrm{s}^2, 1e^{-8}\ \mathrm{km}^2/\mathrm{s}^2)$$

and the filters are initialized in a similar manner, as in example 1. The three hypothetical sensors chosen are located on the surface of the Earth at elevation and azimuth angles of $[1.3, 2.3]$ rad and $[1.65, 0.26]$ rad, respectively, measured from an Earth-centric and Earth-fixed inertial reference frame. The conical FORs have half-angles of $\pi/4$ and $\pi/3$, respectively. In these simulations, the Earth is considered to be stationary. The rotation of the Earth can be incorporated into the sensor-tasking process because the rotation of the Earth can be considered deterministic, and hence the positions of the sensors are known at all time instants. When the sensor moves in a deterministic motion, the sensor model in Eq. (2) will be time varying with respect to the sensor's location. The sensors in this example only measure the local range, elevation, and azimuth angles of the objects (within the FOV) with respect to a reference frame centered at the sensor's location and parallel to the Earth-centric and Earth-fixed frame. All the sensors have a noise covariance of

$$R = \mathrm{diag}([(100\ \mathrm{m})^2(0.05\pi/180\ \mathrm{rad})^2, (0.05\pi/180\ \mathrm{rad})^2])$$

The half-angles for the conic FOV of the sensors are taken as 0.2315 and 0.3876 rad, respectively. The orientations of the sensor mounts are discretized on a uniform grid within the FOR using an 11-by-11 grid. The total simulation time is 24 h with a time step of 10 s. The receding-horizon tasking window is taken as 50 time steps. A new sensor-tasking problem, for the next window, is solved at the end of the current tasking window, when all the measurements are processed. Two algorithms are compared: the ITAUC algorithm as described in example 1, and the sequential in time



a) ITAUC: Frobenius covariance position
b) ITAUC: RMSE position
c) ITAUC: RMSE velocity
d) Sequential time: Frobenius covariance position
e) Sequential time: RMSE position
f) Sequential time: RMSE velocity

Fig. 10 Example 2: comparison of sensor-tasking approaches for satellite tracking (DU: Distance units, TU: Time units).

greedy algorithm as described in Sec. V. The results are shown in Fig. 10. It can be observed that the sequential in time greedy approach tends to do better than the ITAUC approach. This can be attributed to the joint computation of the MI in the sequential in time approach. The joint mutual information helps in coordinating the orientations of the sensors so as to track all objects while taking into account the previous sensor tasks.

C. Example 3: Satellite Tracking Using Exhaustive Search

In this example, a satellite tracking scenario (similar to example 2) is used to illustrate the approaches in Secs. IV.B–IV.D. Here, we use an exhaustive search instead of a nonlinear binary integer solver because it lends itself to parallel processing. Ground-based sensors with limited FORs are used to observe the satellites. The limited number of sensors are dynamically tasked to track multiple satellites orbiting the earth. Efficient tasking helps in making optimal use of all the sensors to track all the objects. The ground-based sensors measure the local radial distance, elevation, and azimuth angles of each space object. The Earth is considered to be fixed while the satellites orbit the Earth. This assumption is only made for ease of illustration and is not a limitation for the approaches described in this paper. The approaches developed in this work can easily incorporate moving sensors (for example, sensors on a rotating Earth or in orbit). In such cases, the sensor model in Eq. (2) becomes time varying with respect to the sensor's position. The sensor information computations in Sec. III.B would directly use the dynamic time varying models (time varying because the sensor's position is a known time-varying function) of the sensors to compute the joint mutual information over time. The two randomly selected sensors on the surface of the Earth are located at elevation and azimuth angles of $[1.74, 0.13]$ rad and $[1.04, 1.30]$ rad. Each sensor has a limited FOR modeled by a cone with half–angles of $\pi/3$ and $\pi/3$. The orientation of the FOR is radially outward from the center of the Earth. Within this FOR, the sensors have a FOV with a half-angle of 0.3 rad that can be oriented to observe space objects. The sensor is constrained such that it can only observe a single space object at a given time instant and has to be reoriented within the FOR to observe another object; this model for a limited field-of-view sensor was inspired from the work in Ref. [5]. This sensor model with such constraints helps define the pointing direction of the sensor toward the predicted location of the target. Note that this constraint is only used during the sensor-tasking optimization process to select the best orientations of the sensors. Once the sensor task is solved and applied, this constraint is disregarded and any object within the FOV of the sensor generates a valid measurement for the filters to process. For example, a telescope with a camera system has to be oriented in the direction of the objects that are to be observed.

Once a sensor is tasked to observe a specific object at a future time instant, it is assumed that the sensor can be quickly reconfigured into a direction that is oriented toward the predicted mean of the object's position at this time instant. The constraint of observing only one object at any particular instant of time is modeled as [5]

$$\sum_{i=1}^{N_t} v_k^{(i,j)} = 1 \quad \begin{cases} j = 1,2,\ldots,N_s \\ k = 1,2,\ldots,N_T \end{cases} \tag{51}$$

This constraint allows multiple sensors to observe the same satellite. When there are multiple overlapping FOVs of different sensors, some space objects can be observed by multiple sensors. The sensor-tasking problem then becomes interesting in assigning sensors to these objects over a period of time. In the following examples, the filters and tasking procedures are simulated over 24 h with a measurement time interval of 5 min. The 15 orbits are randomly generated from a uniform distribution over the classical Keplerian elements as $a \in [8075.7, 8135.7]$ km, $e \in [0.0636, 0.3364]$, $i \in [0.85, 1.7]$ rad, $\Omega = 0$ rad, $\omega = 0$ rad, and $M_0 \in [0, (\pi/20)]$ rad. The orbits are selected such that they intersect the conic FOR of the sensors, and their perigee is greater than the radius of the Earth. The initial means for each filter are randomly selected from a Gaussian distribution with the mean as the true initial condition of the satellite and covariance

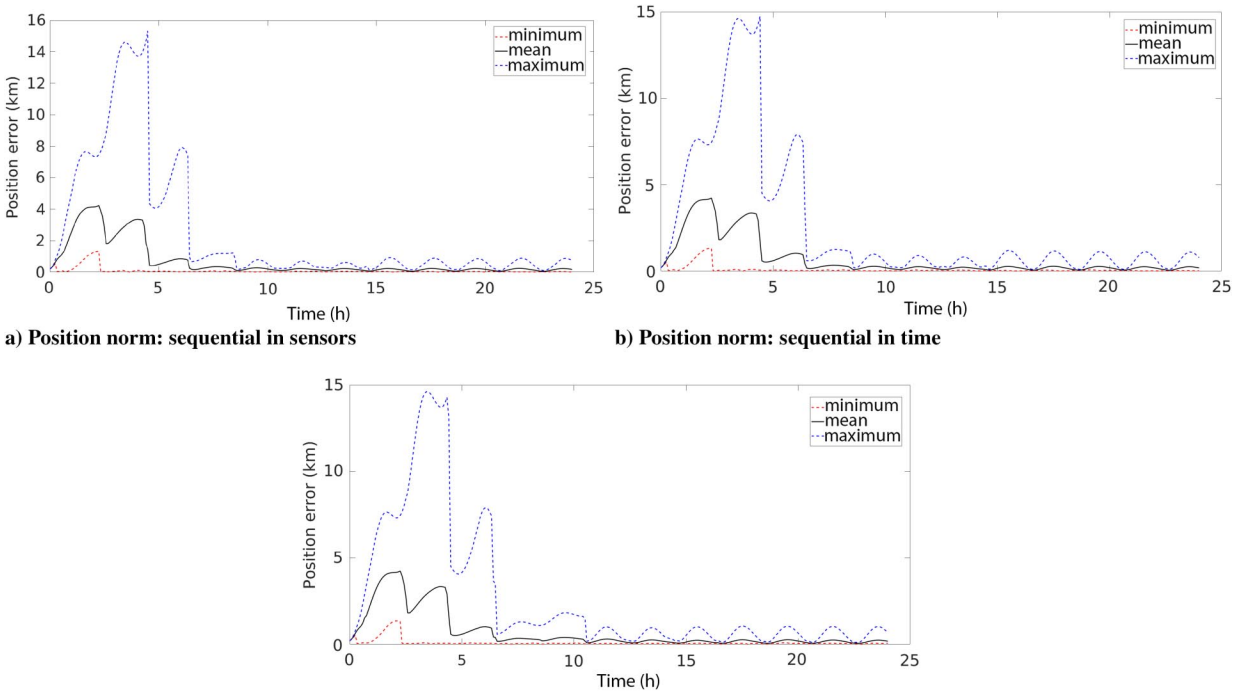
$$P_{0/0} = \mathrm{diag}(0.01\ \mathrm{km}^2, 0.01\ \mathrm{km}^2, 0.01\ \mathrm{km}^2, 1e^{-8}\ \mathrm{km}^2/\mathrm{s}^2, 1e^{-8}\ \mathrm{km}^2/\mathrm{s}^2, 1e^{-8}\ \mathrm{km}^2/\mathrm{s}^2)$$

Each filter also has the same initial covariance of $P_{0/0}$. A large measurement noise for the sensors is considered as $R = \mathrm{diag}((0.1\ \mathrm{km})^2, (2\ \mathrm{deg})^2, (2\ \mathrm{deg})^2)$.

The FORs of the sensors overlap, and some satellites can be observed by both of the sensors. A moving horizon window of five time steps (or 25 min) is considered to optimize the sensor schedule. The initial conditions of the satellites are such that they more or less

simultaneously enter the field of view of the sensors. This makes the sensor-tasking procedure challenging because the sensors have to be scheduled in a way to reduce the uncertainty in the entire system. The measurements for the filters are generated from the true satellite orbits when they are within the FOV of the sensors.

The three exhaustive suboptimal search methods of sequential in sensors, sequential in time, and sequential in objects are individually used to task the sensors. The same set of satellites, filter conditions, and measurements is used for all the methods considered in this example. For the sequential in objects case, the objects are ordered using the trace of their covariance; for the sequential in sensors case, the sensors are ordered by the number of objects in the FOV. Figure 11 shows the 2-norm of the position error in the satellites. The maximum, mean, and minimum errors among the satellites are plotted as a function of time. It can be seen that all the methods have similar tracking performances. Similarly, Fig. 12 shows the minimum, mean, and maximum of the Frobenius norm of the covariance matrices for all the satellites at every time step. The errors grow initially and then reduce after 6 h into the simulation, when multiple sensor measurements are tasked for each satellite over multiple revolutions.

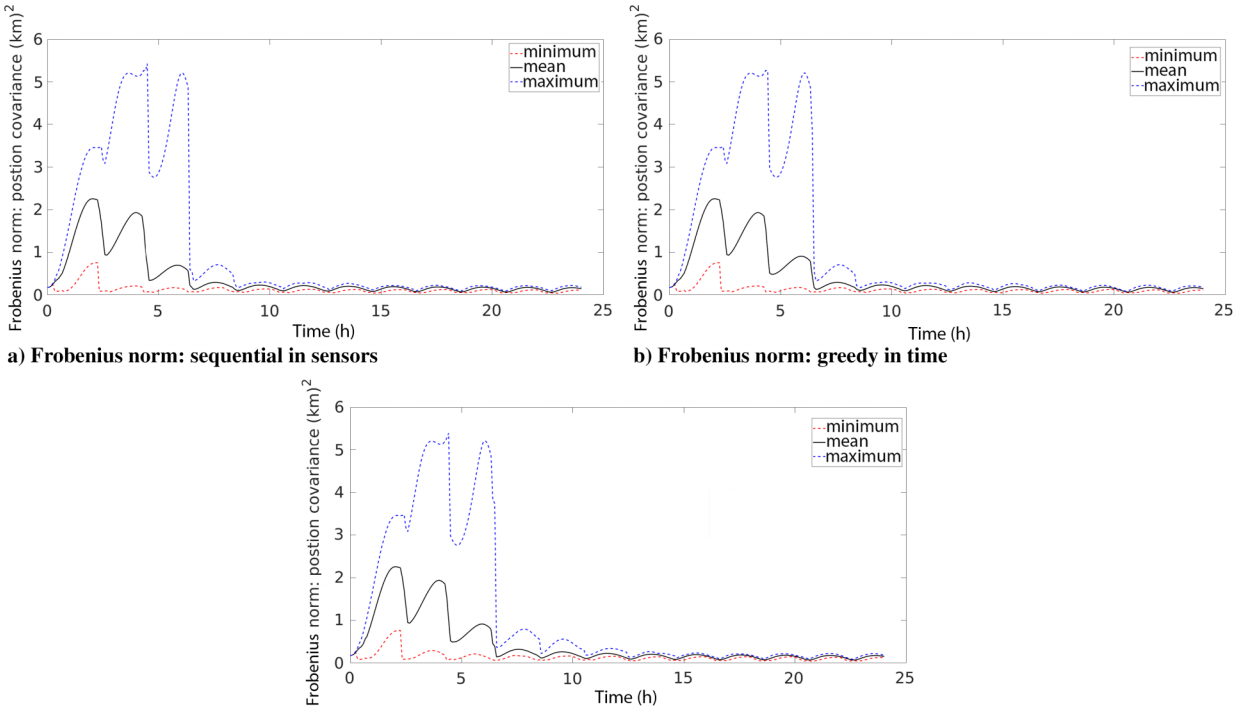


a) Position norm: sequential in sensors

b) Position norm: sequential in time

c) Position norm: sequential in targets

Fig. 11 Example 3, case 1: 2-norm position error (in kilometers) vs time (in hours).



a) Frobenius norm: sequential in sensors

b) Frobenius norm: greedy in time

c) Frobenius norm: greedy in targets

Fig. 12 Example 1: Frobenius norm of position covariance matrix.

The measurements for each satellite are only available when they are inside the FOR of the sensors and have to complete one full revolution about Earth to intersect the FOR of the sensors again. The sensor-tasking routine accounts for the constraint that only one object can be observed by the sensor at any given time step. Hence, there can be stretches of time where some satellites do not have observations, leading to a growth in their uncertainty. It can be observed in Fig. 11 that, after 10 h, the errors are kept low and the sensor-tasking procedure appropriately distributes the observations for all the objects.

On close observation of Figs. 11 and 12 in the time interval of 6 to 11 h, it can be seen that the sequential in sensors and sequential in time methods are slightly better that the sequential in objects method. This is due to the penalty of limiting the number of sensor observations in the method for sequential in objects. Although the tracking performance is similar for all the sequential methods, the computational costs vary. The ideal exhaustive search problem would have around $1.4272e + 45$ configurations. But, with the constraint in Eq. (51) for feasible assignments, the exhaustive search for the sequential in sensors approach has a total of $N_s \times N_o^{N_T} = 1518750$ possibilities; whereas the sequential in time and the sequential in objects approaches have computational costs of $N_T \times N_o^{N_s} = 1125$ and $N_o \times 2^{N_s N_T} = 15360$, respectively. Although these lead to a large space for an exhaustive search, it should be noted that most of the search space is infeasible because the sensors have limited fields of view. All the tasking events that contain an assignment of a sensor to an object outside its FOR can be simply skipped, hence reducing the search space significantly. The sequential in sensors method takes an average of 47 s to compute the sensor schedule over the entire time window of 25 min, whereas the sequential in time and the sequential in objects methods take less than 5 s. As an exhaustive search naturally lends itself for parallel computations, executing the exhaustive search over the entire feasible tasking search space can be greatly increased by using multiple threads or cores that are ubiquitous on modern processors. All the simulations are done in MATLAB for this paper. A C/C++ implementation will significantly increase the speed of the exhaustive search.

Figure 13 shows a heat map of the trace of the covariance matrix for each satellite in the left column. Each row corresponds to an object, and the x axis corresponds to time. A similar conclusion can be made that all the sequential methods have similar performances in reducing the uncertainty. The intensity of the color map indicates the magnitude of the trace. The right column in Fig. 13 shows the sensor-tasking schedule for each object along the y axis. A dark vertical line or mark indicates that a sensor has been assigned to the object. It can be observed that there are empty vertical bands of no sensors being tasked. This is because the objects almost move in and out of the FOR as a group, leading to intervals of time with no objects in the FOR. The objects have to roughly make a revolution before entering the FOR.

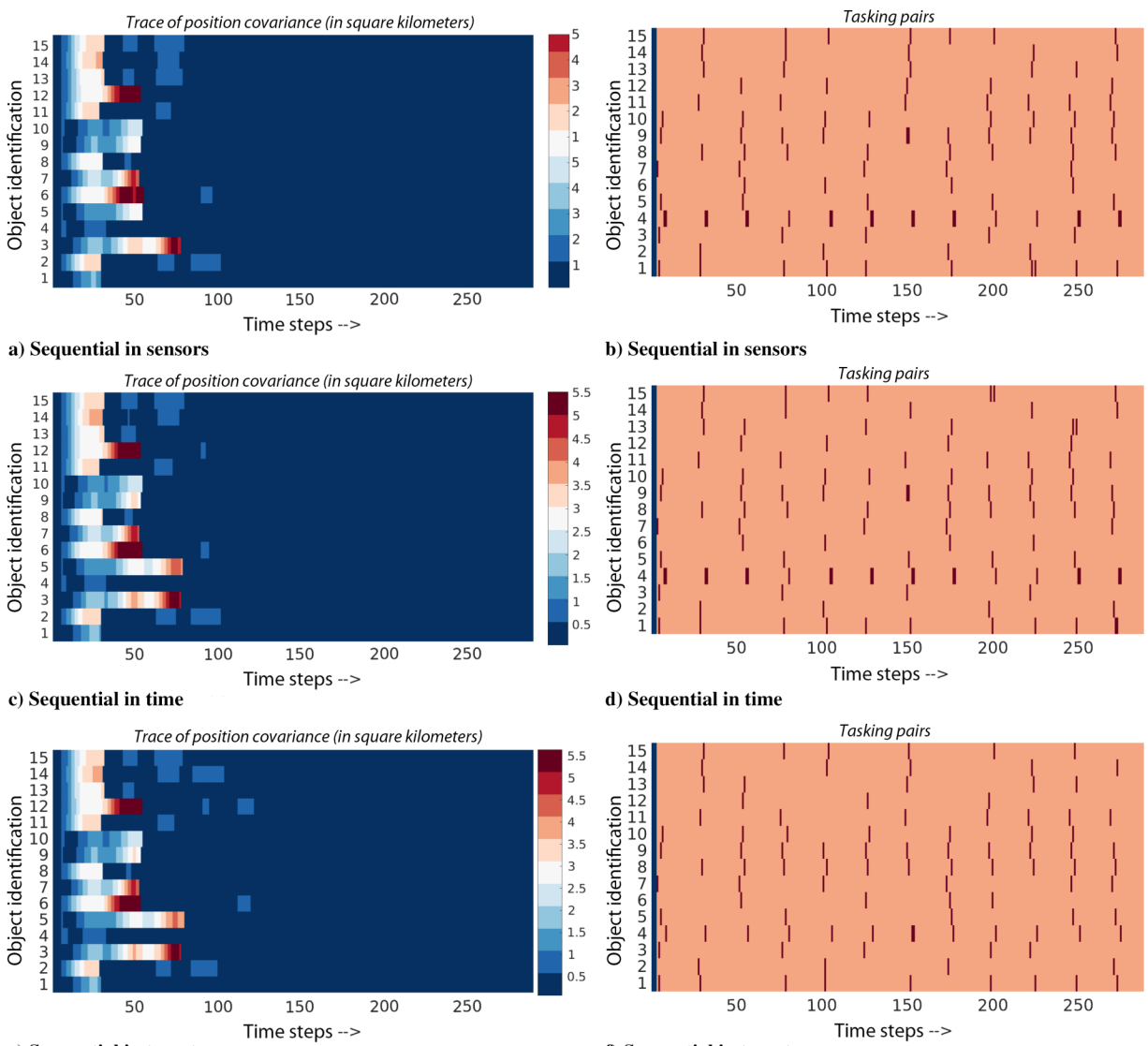


Fig. 13 Example 3, case 1: heat map for trace of position covariance matrix.

VII. Conclusions

The problem of tracking multiple objects using a limited number of sensors with operational constraints was considered. The multi-step joint mutual information was used over multiple objects, sensors, and time steps as a measure of sensor performance. This joint mutual information was maximized to find the optimal sensor schedule or configuration. It was observed that the ideal problem of maximizing the joint mutual information cost is nonlinear and nonconvex, as well as often combinatorial in nature, and thus computationally challenging and intractable. To address this challenge, it was first shown that the mutual information cost considered in this paper is submodular and nondecreasing, and thus lends itself to efficient greedy approaches that can achieve suboptimal solutions with a constant bound to the optimal solution. Furthermore, methods that were sequential in time, sequential in sensors, and sequential in objects were proposed in a moving horizon approach to further reduce the computational complexity. The numerical simulations of tracking multiple UAVs and satellites with limited ground-based sensors illustrated the performances of the various approaches proposed in this paper. In future work, the combined problem of sensor tasking and data association will be pursued.

Appendix A: Fisher Information Matrix as a Measure of Local Information

Consider the likelihood PDF $p(\boldsymbol{y}|\boldsymbol{x})$, where $\boldsymbol{y}$ is the vector of measurements and $\boldsymbol{x}$ is the state being estimated from these measurements. Then, the Kullback–Leibler divergence between the likelihood PDF $p(\boldsymbol{y}|\boldsymbol{x})$ computed at $\boldsymbol{x}$ and the likelihood PDF $p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)$ computed at an infinitesimal change in the estimate is given as ([21] p. 39)

$$\begin{aligned} D_{\mathrm{KL}}(p(\boldsymbol{y}|\boldsymbol{x})||p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)) &= \int p(\boldsymbol{y}|\boldsymbol{x}) \log\left\{\frac{p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)}\right\} \\ \mathrm{d}\boldsymbol{y} &= \int p(\boldsymbol{y}|\boldsymbol{x})(\log\{p(\boldsymbol{y}|\boldsymbol{x})\} \\ &\quad - \log\{p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)\})\,\mathrm{d}\boldsymbol{y} \end{aligned} \tag{A1}$$

The Taylor series expansion of $\log\{p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)\}$ about $\boldsymbol{x}$, up to the second order, is given as

$$\begin{aligned} \log\{p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)\} &\approx \log\{p\} + \frac{1}{p}\sum_{i=1}^{n}\frac{\partial p}{\partial x_i}\Delta x_i \\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[-\frac{1}{p^2}\frac{\partial p}{\partial x_i}\frac{\partial p}{\partial x_j} + \frac{1}{p}\frac{\partial^2 p}{\partial x_j \partial x_i}\right]\Delta x_i \Delta x_j \end{aligned} \tag{A2}$$

where $p \equiv p(\boldsymbol{y}|\boldsymbol{x})$ and is a PDF in the random variable $\boldsymbol{y}$. The partial derivatives $\partial p/\partial x_i$ and $\partial^2 p/\partial x_j \partial x_i$ are evaluated at the given(or known) estimate $\boldsymbol{x}$. Substituting the Taylor series expansion of Eq. (A2) into the KL divergence expression of Eq. (A1) gives the simplified KL divergence as

$$\begin{aligned} &D_{\mathrm{KL}}(p(\boldsymbol{y}|\boldsymbol{x})||p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)) \\ &\approx -\int p(\boldsymbol{y}|\boldsymbol{x})\left(\frac{1}{p}\sum_{i=1}^{n}\frac{\partial p}{\partial x_i}\Delta x_i \right. \\ &\quad \left. + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[\frac{1}{p^2}\frac{\partial p}{\partial x_i}\frac{\partial p}{\partial x_j} + \frac{1}{p}\frac{\partial^2 p}{\partial x_j \partial x_i}\right]\Delta x_i \Delta x_j\right)\mathrm{d}\boldsymbol{y} \end{aligned} \tag{A3}$$

$$\begin{aligned} &= -\sum_{i=1}^{n}\int \frac{\partial p}{\partial x_i}\,\mathrm{d}\boldsymbol{y}\Delta x_i \\ &\quad + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\int p(\boldsymbol{y}|\boldsymbol{x})\left[\frac{1}{p^2}\frac{\partial p}{\partial x_i}\frac{\partial p}{\partial x_j}\right]\mathrm{d}\boldsymbol{y}\Delta x_i \Delta x_j \\ &\quad - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\int \frac{\partial^2 p}{\partial x_j \partial x_i}\,\mathrm{d}\boldsymbol{y}\Delta x_i \Delta x_j \end{aligned} \tag{A4}$$

Under smooth regularity conditions for the likelihood PDF $p(\boldsymbol{y}|\boldsymbol{x})$ and all of its first two partial derivatives with respect to $\boldsymbol{x}$, the integral and derivative operators can be interchanged such that

$$\int \frac{\partial^2 p}{\partial x_j \partial x_i}\,\mathrm{d}\boldsymbol{y} = \frac{\partial^2}{\partial x_j \partial x_i}\left(\int p(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{y}\right) = \frac{\partial^2}{\partial x_j \partial x_i}(1) = 0$$

and

$$\int \frac{\partial p}{\partial x_i}\,\mathrm{d}\boldsymbol{y} = \frac{\partial}{\partial x_i}\int p(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{y} = 0$$

The KL divergence then becomes

$$\begin{aligned} &D_{\mathrm{KL}}(p(\boldsymbol{y}|\boldsymbol{x})||p(\boldsymbol{y}|\boldsymbol{x}+\boldsymbol{\Delta}x)) \\ &\approx \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\int p(\boldsymbol{y}|\boldsymbol{x})\left[\frac{1}{p^2}\frac{\partial p}{\partial x_i}\frac{\partial p}{\partial x_j}\right]\mathrm{d}\boldsymbol{y}\Delta x_i \Delta x_j \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\underbrace{\int p(\boldsymbol{y}|\boldsymbol{x})\left[\frac{1}{p^2}\frac{\partial p}{\partial x_i}\frac{\partial p}{\partial x_j}\right]\mathrm{d}\boldsymbol{y}}_{\mathrm{FIM}_{ij}}\Delta x_i \Delta x_j \end{aligned}$$

where the ith row and j^{th} column of the FIM are given as

$$\begin{aligned} \mathrm{FIM}_{ij} &= \int p(\boldsymbol{y}|\boldsymbol{x})\left[\frac{1}{p^2}\frac{\partial p}{\partial x_i}\frac{\partial p}{\partial x_j}\right] \\ \mathrm{d}\boldsymbol{y} &= \int p(\boldsymbol{y}|\boldsymbol{x})\left[\frac{\partial \log(p)}{\partial x_i}\frac{\partial \log(p)}{\partial x_j}\right] \\ \mathrm{d}\boldsymbol{y} &= E_{\boldsymbol{y}}\left[\frac{\partial \log(p)}{\partial x_i}\frac{\partial \log(p)}{\partial x_j}|\boldsymbol{x}\right] \end{aligned} \tag{A5}$$

It can be observed that the FIM is the Hessian of the KL divergence evaluated at the estimate $\boldsymbol{x}$. Hence, the FIM is a measure of the local curvature at the estimate.

Appendix B: Useful Matrix Identities

Remark 1 (Ref. [32] fact 3.2.2) Woodbury identity:

$$(A + CBC^T)^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^T A^{-1} C)^{-1} C^T A^{-1} \tag{B1}$$

Remark 2 (Ref. [32] fact 9.1.2):

$$\begin{aligned} \left|\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\right| &= |A_{11}| \cdot |A_{22} - A_{21}A_{11}^{-1}A_{12}| \\ &= |A_{22}| \cdot |A_{11} - A_{12}A_{22}^{-1}A_{21}| \end{aligned} \tag{B2}$$

Remark 3 (Ref. [32] fact 9.1.3):

$$\begin{aligned} &\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}W_2^{-1}A_{21}A_{11}^{-1} & -W_1^{-1}A_{12}A_{22}^{-1} \\ -A_{22}^{-1}A_{21}W_1^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}W_1^{-1}A_{12}A_{22}^{-1} \end{bmatrix} \end{aligned} \tag{B3}$$

with $W_1 = A_{11} - A_{12}A_{22}^{-1}A_{21}$ and $W_2 = A_{22} - A_{21}A_{11}^{-1}A_{12}$. Furthermore, using Remark 1, we have $W_1^{-1} = A_{11}^{-1} + A_{11}^{-1}A_{12}W_2^{-1}A_{21}A_{11}^{-1}$.

Acknowledgments

This material is based upon the work jointly supported by the U.S. Air Force Office of Scientific Research (FA9550-15-1-0313 and FA9550-17-1-0088).

References

[1] Madankan, R., Singla, P., and Singh, T., "Parameter Estimation of Atmospheric Release Incidents Using Maximal Information Collection," *Dynamic Data-Driven Environmental Systems Science*, edited by S. Ravela, and A. Sandu, Springer International, Switzerland, Cham, 2015, pp. 310–321.

[2] Frueh, C., Fiedler, H., and Herzog, J., "Heuristic and Optimized Sensor Tasking Observation Strategies with Exemplification for Geosynchronous Objects," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 5, 2018, pp. 1036–1048. https://doi.org/10.2514/1.G003123

[3] Gehly, S., Jones, B., and Axelrad, P., "Sensor Allocation for Tracking Geosynchronous Space Objects," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 1, 2018, pp. 149–163. https://doi.org/10.2514/1.G000421

[4] Cover, T. M., and Thomas, J. A., *Elements of Information Theory*, Wiley Series in Telecommunications and Signal Processing, Wiley-Interscience, New York, 2006.

[5] Williams, P. S., Spencer, D. B., and Erwin, R. S., "Coupling of Estimation and Sensor Tasking Applied to Satellite Tracking," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 993–1007. https://doi.org/10.2514/1.59361

[6] Gualdoni, M. J., and DeMars, K. J., "The Information Content of Data Arcs for Multi-Step Sensor Tasking," AIAA Paper 2018-0727, 2018. https://doi.org/10.2514/6.2018-0727

[7] Jaunzemis, A. D., Holzinger, M. J., and Luu, K. K., "Sensor Tasking for Spacecraft Custody Maintenance and Anomaly Detection Using Evidential Reasoning," *Journal of Aerospace Information Systems*, Vol. 15, No. 3, 2018, pp. 131–156. https://doi.org/10.2514/1.I010584

[8] Martnez, S., and Bullo, F., "Optimal Sensor Placement and Motion Coordination for Target Tracking," *Automatica*, Vol. 42, No. 4, 2006, pp. 661–668. https://doi.org/10.1016/j.automatica.2005.12.018

[9] Tharmarasa, R., Kirubarajan, T., and Hernandez, M. L., "Large-Scale Optimal Sensor Array Management for Multitarget Tracking," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 37, No. 5, 2007, pp. 803–814. https://doi.org/10.1109/TSMCC.2007.901003

[10] Soderlund, A. A., and Kumar, M., "Optimization of Multitarget Tracking Within a Sensor Network via Information-Guided Clustering," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 2, 2019, pp. 317–334. https://doi.org/10.2514/1.G003656

[11] Soderlund, A., Kumar, M., and Kim, D., "Rapid Clustering for Optimal Sensor Selection in Heterogeneous Wireless Sensor Networks," *2018 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2018-1135, 2018.

[12] Guestrin, C., Krause, A., and Singh, A. P., "Near-Optimal Sensor Placements in Gaussian Processes," *Proceedings of the 22nd International Conference on Machine Learning*, ACM Press, New York, 2005, pp. 265–272.

[13] Mahler, R. P., *Advances in Statistical Multisource-Multitarget Information Fusion*, Artech House, Norwood, MA, 2014.

[14] Adurthi, N., Singla, P., and Majji, M., "Conjugate Unscented Transform Based Approach for Dynamic Sensor Tasking and Space Situational Awareness," *2015 American Control Conference (ACC)*, Chicago, IL, 2015, pp. 5218–5223. https://doi.org/10.1109/ACC.2015.7172154

[15] Adurthi, N., Singla, P., and Singh, T., "Optimal Information Collection for Nonlinear Systems—An Application to Multiple Target Tracking and Localization," *2013 American Control Conference*, Washington, D.C., 2013, pp. 3864–3869. https://doi.org/10.1109/ACC.2013.6580429

[16] Adurthi, N., and Singla, P., "Information Driven Optimal Sensor Control for Efficient Target Localization and Tracking," *2014 American Control Conference*, Portland, OR, 2014, pp. 610–615. https://doi.org/10.1109/ACC.2014.6859310

[17] Adurthi, N., Singla, P., and Majji, M., "Conjugate Unscented Transformation Based Orbital State Estimation and Sensor Tasking for Efficient Space Surveillance," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2014-4168, 2014.

[18] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L., "An Analysis of Approximations for Maximizing Submodular Set Functions—I," *Mathematical Programming*, Vol. 14, No. 1, 1978, pp. 265–294. https://doi.org/10.1007/BF01588971

[19] Chekuri, C., and Pal, M., "A Recursive Greedy Algorithm for Walks in Directed Graphs," *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, IEEE Publ., Piscataway, NJ, 2005, pp. 245–253. https://doi.org/10.1109/SFCS.2005.9

[20] Adurthi, N., Singla, P., and Singh, T., "Conjugate Unscented Transformation: Applications to Estimation and Control," *Journal of Dynamics Systems, Measurement, and Control*, Vol. 140, No. 3, 2017, Paper 030907.

[21] Kullback, S., *Information Theory and Statistics*, Courier Corp., Chelmsford, MA, 1997.

[22] Kullback, S., and Leibler, R. A., "On Information and Sufficiency," *Annals of Mathematical Statistics*, Vol. 22, No. 1, 1951, pp. 79–86. https://doi.org/10.1214/aoms/1177729694

[23] Kreucher, C., Kastella, K., and Hero, A. O., III, "Information Based Sensor Management for Multitarget Tracking," *Proceedings of SPIE*, Vol. 5204, 2003, Paper 481.

[24] Kreucher, C., Hero, A. O., and Kastella, K., "A Comparison of Task Driven and Information Driven Sensor Management for Target Tracking," *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05*, IEEE Publ., Piscataway, NJ, 2005, pp. 4004–4009.

[25] Kreucher, C., Hero, A. O., and Kastella, K., "A Comparison of Task Driven and Information Driven Sensor Management for Target Tracking," *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2005, pp. 4004–4009. https://doi.org/10.1109/CDC.2005.1582788

[26] Kalman, R., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45. https://doi.org/10.1115/1.3662552

[27] Anderson, B. D. O., and Moore, J. B., *Optimal Filtering*, Prentice–Hall, Englewood Cliffs, NJ, 1979.

[28] Ponda, S., Kolacinski, R. M., and Frazzoli, E., "Trajectory Optimization for Target Localization Using Small Unmanned Aerial Vehicles," *AIAA Guidance, Navigation, and Control Conference (GNC)*, AIAA Paper 2009-6015, 2009.

[29] Adurthi, N., and Singla, P., "A Conjugate Unscented Transformation Based Approach for Accurate Conjunction Analysis," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 9, Sept. 2015, pp. 1642–1658. https://doi.org/10.2514/1.G001027

[30] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw–Hill, New York, 1984.

[31] Adurthi, N., "Conjugate Unscented Transform Based Methods for Uncertainty Quantification, Nonlinear Filtering, Optimal Feedback Control and Dynamic Sensing," Ph.D. Dissertation, State Univ. of New York at Buffalo, Buffalo, NY, May 2016.

[32] Petersen, K. B., and Pedersen, M. S., "The Matrix Cookbook, version 20121115," Technical Univ. Denmark, TR-3274, Kongens Lyngby, Denmark, Nov. 2012, http://matrixcookbook.com/.

[33] Bernstein, D. S., *Matrix Mathematics: Theory, Facts, and Formulas*, Princeton Univ. Press, Princeton, NJ, 2009.

[34] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, 2000, pp. 477–482. https://doi.org/10.1109/9.847726

[35] Arasaratnam, I., and Haykin, S., "Cubature Kalman Filter," *IEEE Transactions on Automatic Control*, Vol. 54, No. 6, 2009, pp. 1254–1269. https://doi.org/10.1109/TAC.2009.2019800

[36] Ito, K., and Xiong, K., "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, Vol. 45, No. 5, 2000, pp. 910–927. https://doi.org/10.1109/9.855552

[37] Jazwinski, A. H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.

[38] Burer, S., and Letchford, A. N., "Non-Convex Mixed-Integer Nonlinear Programming: A Survey," *Surveys in Operations Research and Management Science*, Vol. 17, No. 2, 2012, pp. 97–106. https://doi.org/10.1016/j.sorms.2012.08.001

[39] Henrion, D., and Lasserre, J.-B., "GloptiPoly: Global Optimization Over Polynomials with Matlab and SeDuMi," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 29, No. 2, 2003, pp. 165–194. https://doi.org/10.1145/779359

[40] Bates, D. J., Hauenstein, J. D., Sommese, A. J., and Wampler, C. W., "Bertini: Software for Numerical Algebraic Geometry," 2006. https://doi.org/10.7274/R0H41PB5

[41] Horn, R. A., and Johnson, C. R., *Matrix Analysis*, 2nd ed., Cambridge Univ. Press, New York, 2012. https://doi.org/10.1017/9781139020411

[42] Krause, A., Singh, A., and Guestrin, C., "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies," *Journal of Machine Learning Research*, Vol. 9, Feb. 2008, pp. 235–284.

[43] Krause, A., Guestrin, C., Gupta, A., and Kleinberg, J., "Near-Optimal Sensor Placements: Maximizing Information While Minimizing Communication Cost," *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, ACM Press, New York, 2006, pp. 2–10.

[44] Rawlings, J., Meadows, E., and Muske, K., "Nonlinear Model Predictive Control: A Tutorial and Survey," *IFAC Proceedings Volumes*, Vol. 27, No. 2, 1994, pp. 185–197. https://doi.org/10.1016/S1474-6670(17)48151-1

[45] Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R., *Estimation with Applications to Tracking and Navigation*, Wiley, New York, 2002.

[46] Gehly, S., Jones, B., and Axelrad, P., "Sensor Allocation for Tracking Geosynchronous Space Objects," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 1, 2018, pp. 149–163. https://doi.org/10.2514/1.G000421

[47] Vallado, D. A., and McClain, W. D., *Fundamentals of Astrodynamics and Applications*, Space Technology Library, Microcosm Press, Portland, OR, 2001.