

A Unified Approach to Optimally Solving Sensor Scheduling and Sensor Selection Problems in Kalman Filtering

Shamak Dutta, Nils Wilde, and Stephen L. Smith

Abstract—We consider a general form of the sensor scheduling problem for state estimation of linear dynamical systems, which involves selecting sensors that minimize the trace of the Kalman filter error covariance (weighted by a positive semidefinite matrix) subject to polyhedral constraints. This general form captures several well-studied problems including sensor placement, sensor scheduling with budget constraints, and Linear Quadratic Gaussian (LQG) control and sensing co-design. We present a mixed integer optimization approach that is derived by exploiting the optimality of the Kalman filter. While existing work has focused on approximate methods to specific problem variants, our work provides a unified approach to computing optimal solutions to the general version of sensor scheduling. In simulation, we show this approach finds optimal solutions for systems with 30 to 50 states in seconds.

I. INTRODUCTION

In large scale dynamical systems, it is practically infeasible to measure *all* output variables due to communication or energy constraints. Thus, practical solutions often have to rely on only accessing a subset of the data. Optimizing the subset of output variables to measure in order to best perform state estimation is commonly known as the *sensor scheduling problem*. This finds applications in multi-robot environmental monitoring [1]–[3], minimal controllability problems [4]–[6], control and sensing co-design [7]–[9], among others.

Sensor scheduling considers a stochastic linear system evolving in discrete time, which can be partially observed using sensors. Due to resource constraints (e.g., energy constraints), at each time step only a subset of sensors can be turned on. By limiting the number of active sensors, the sensor battery life is prolonged, allowing for long-term usage. The objective is to determine a sensor schedule, i.e., a subset of sensors to be activated at each step, that minimizes the trace of the Kalman filter weighted error covariance while satisfying the resource constraints. The weight is provided by a positive semidefinite matrix which can encode several objectives including final state error and average error. When the chosen sensor set is not allowed to change over time, the problem is known as the *sensor selection* problem. The sensor scheduling problem and its variants are challenging; existing approaches have provided approximate solutions via greedy algorithms [7], [10]–[18], convex relaxations [19]–[27], and randomized algorithms [28]–[30]. However, in this

paper we strive to compute *optimal* solutions leveraging advances in mixed integer optimization.

While approximation algorithms have yielded useful insights into specific variants of the problem, an approach that computes optimal solutions to the general version has been elusive. The main difficulty lies in the analysis of the Kalman filter recursive equations in terms of the selected sensor subsets. However, the combinatorial nature of the problem suggests encoding the problem as an integer program.

Despite integer programming being NP-complete [31] and generally being considered intractable beyond specific cases, the mixed integer optimization community has made tremendous theoretical and practical advances in last few decades with *machine-independent* speedup factors of upto 580,000 [32, Section 2.1]. One of the benefits is that modern solvers are anytime i.e., if terminated early, approximate solutions with suboptimality certificates are provided to the user. It is relatively straightforward to set up most combinatorial problems as integer programs by modelling the objects of interest as binary decision variables. However, the structure of the resulting objective may not be amenable to optimization. It is well known that the formulation plays an important role in solving integer programs [33, Section 1.2]. This is because the relaxation is used extensively to provide lower bounds in the branch and bound procedure in mixed integer optimization. Thus, if the relaxation has good structure (linearity, convexity) and is tight, one can find optimal solutions relatively quickly by pruning many parts of the search tree. The question we wish to answer is whether one can leverage the structure of the sensor scheduling problem to formulate mixed integer programs.

Contributions: Our main contribution is the formulation of the general sensor scheduling problem as a mixed integer program with a convex quadratic objective and linear constraints (Theorem 1). This enables us to leverage modern solvers such as GUROBI [34] or CPLEX [35]. We accomplish this using the optimality property of the Kalman filter for the trace of a weighted error covariance. This enables us to optimize over the class of linear filters, resulting in a convex minimization problem. This circumvents the need to work with the recursive form of the filter and the associated non-linear algebraic Riccati equations, which can be difficult to model. In simulations, our approach computes optimal solutions to sensor selection and scheduling problems on systems with 35 to 50 states in seconds.

Related Work: Sensor scheduling has been studied in various forms in the literature. The difference in formulations generally boils down to the choice of objective function,

This research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Nutrien Ltd.

S. Dutta and S. L. Smith are with the Department of Electrical and Computer Engineering, University of Waterloo, Canada {stephen.smith, shamak.dutta}@uwaterloo.ca. N. Wilde is with the Cognitive Robotics Department, Delft University of Technology, Netherlands (N.Wilde@tudelft.nl).

defined in terms of the Kalman filter error covariance. The work in [11] characterizes the submodularity of commonly used functions such as the trace, maximum eigenvalue, and log determinant. This paper considers the trace as it is a direct measure of the estimation error which we seek to minimize.

The trace is not a submodular function and does not benefit from the constant factor approximation guarantees provided by the greedy algorithm. The typical way to address this is to use surrogate objectives such as the log determinant, which is submodular and can be efficiently approximately maximized. However, [17, Remark 1] notes that it is a poor proxy for minimizing the trace. There is a large body of work tackling the log determinant/maximum eigenvalue objective [10]–[13], [19], [21], [30] and various other controllability metrics [23], [26], [28]. However, these algorithms do not optimally solve trace minimization over a finite horizon.

The trace minimization of the *steady state* error covariance has been considered [14], [16], [36], [37]. It was established to be NP-hard in [14], which also demonstrated that greedy algorithms perform well in practice but without guarantees. In fact, there is no polynomial time constant factor approximation algorithm for this problem unless $P = NP$ [16].

On a *finite horizon*, greedy algorithms are a popular tool for minimizing the trace of the error covariance due to its computational efficiency and empirical success. Recent work has analyzed their performance through the lens of approximate submodularity [17], weak submodularity [18], [29], and identified strict conditions for submodularity [15]. However, [17, Section VI] notes that the performance bounds tend to be quite loose in practice and greedy algorithms perform near-optimally on small scale instances. The work in this paper computes optimal solutions to instances not previously considered which may help in a more thorough empirical study of the greedy algorithm.

There has been limited work in computing the optimal solution to the sensor scheduling problem. Convex relaxations have provided useful insights [20], [23], [24], [26], [27] for approximate solutions. The work in [38] characterizes certain properties of the optimal solution and uses it to construct a suboptimal solution via tree pruning. The work closest to ours is [22], which formulates a quadratic program with ℓ_0 constraints. The authors solve a relaxed version of the problem and construct an approximate solution through iterative reweighting. As a baseline comparison, we convert the ℓ_0 constraint to an indicator constraint to form a MIQP, which we discuss in detail in Section IV-A.

Outline: We begin with a brief introduction to linear estimation in Section II. This contains results about the optimality of linear estimators which apply to Kalman filtering. Section III contains the problem formulation as well as a few examples of the problem types it captures. Our mixed integer reformulation is presented in Section IV. We investigate its performance through numerical simulations in Section V and conclude with some directions for future work in Section VI. Due to space limitations, the proofs of all results are omitted and can be found in the preprint [39].

II. PRELIMINARIES

Notation: Let $[n]$ denote the set of positive integers ranging from 1 to n . For a vector $v \in \mathbb{R}^n$, define $\text{supp}(v) := \{i \in [n] : v_i \neq 0\}$. Further, for any $S \subseteq [n]$, let $v_S \in \mathbb{R}^{|S|}$ be the vector with entries v_i for $i \in S$. The set of $n \times n$ positive-definite matrices is denoted by \mathbb{S}_{++}^n . For a matrix $X \in \mathbb{R}^{m \times n}$, denote the Frobenius norm by $\|X\|_F := (\text{trace}(X'X))^{1/2}$. For two random vectors x, y with dimensions n and m respectively, denote the matrix of covariances between x and y by $\Sigma_{xy} := \mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])'] \in \mathbb{R}^{n \times m}$.

A. Linear Estimation

Let x and y be jointly distributed random vectors (of size n and m respectively) with zero mean and covariance given by

$$\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \in \mathbb{R}^{(n+m) \times (n+m)}. \quad (1)$$

The following characterizes the optimal linear estimator.

Definition 1 (Theorem 2.1, [40]). The optimal linear estimator of x given y is defined as

$$\hat{x} := K_* y, \quad (2)$$

where the optimal coefficient matrix $K_* \in \mathbb{R}^{n \times m}$ is the solution that minimizes the expected squared error

$$K_* := \arg \min_K \mathbb{E} \left[(x - Ky)' (x - Ky) \right] = \Sigma_{xy} \Sigma_{yy}^{-1}. \quad (3)$$

The resulting estimation error is

$$\mathbb{E} \left[(x - K_* y)' (x - K_* y) \right] = \text{trace} \left(\Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} \right). \quad (4)$$

The optimal linear estimator also satisfies a stronger property, given in the following lemma. This is a relatively straightforward extension of the previous result.

Lemma 1. Given a positive semi-definite matrix $M = Q'Q \in \mathbb{S}_+^n$, the coefficient matrix K_* in (3) is also minimizes the M -weighted expected square error:

$$K_* = \arg \min_{K \in \mathbb{R}^{n \times m}} \mathbb{E} \left[(x - Ky)' M (x - Ky) \right]. \quad (5)$$

III. PROBLEM FORMULATION

Our setup closely follows [22] which captures a variety of interesting sensor scheduling problems.

Dynamical System: Consider the following linear system:

$$\begin{aligned} x_{k+1} &= Ax_k + w_k, \\ y_k &= Cx_k + v_k, \end{aligned} \quad (6)$$

where the process noise w_k , measurement noise v_k , and initial state x_1 are zero mean uncorrelated random variables. For all $k \geq 1$, we assume the covariance of w_k and v_k are given by $W \in \mathbb{S}_{++}^n$ and $V \in \mathbb{S}_{++}^m$ respectively and the *a priori* covariance of the initial state x_1 is given by $\Sigma_{1|0} \in \mathbb{S}_{++}^n$. Further, we assume the state $x_k \in \mathbb{R}^n$ and the vector resulting from all m sensor measurements is $y_k \in \mathbb{R}^m$.

Sensor Schedule: For a given time horizon T , the total number of measurements is mT since each step yields m sensor readings. For $i \in [mT]$, consider binary indicator variables $\gamma_i \in \{0, 1\}$ such that if sensor $j \in [m]$ is on at time step $k \in [T]$, then $\gamma_{m(k-1)+j} = 1$. The binary vector $\gamma = [\gamma_1, \dots, \gamma_{mT}] \in \{0, 1\}^{mT}$ is called a *sensor schedule*.

Kalman filter: The Kalman filter (KF) is usually described recursively in terms of the estimates from the previous time step. However, viewing it directly as a linear estimator over the set of sensor measurements will be more useful for our purposes. Let $\gamma \in \{0, 1\}^{mT}$ be a sensor schedule over a horizon T with $S_\gamma := \text{supp}(\gamma)$. For each $k \in [T]$, define

$$\begin{aligned} Y_k &:= [y'_1, \dots, y'_k]' \in \mathbb{R}^{mk} \text{ (measurements until time } k) \\ \hat{x}_{k|k}^\gamma &:= K_{*,k}^\gamma Y_{k,\gamma} \in \mathbb{R}^n \text{ (KF estimate using selected measurements)} \\ e_k^\gamma &:= x_k - \hat{x}_{k|k}^\gamma \in \mathbb{R}^n \text{ (KF estimation error)} \\ \Sigma_{k|k}^\gamma &:= \Sigma e_k^\gamma e_k^{\gamma'} \in \mathbb{S}_+^n \text{ (KF posterior error covariance)} \end{aligned} \quad (7)$$

Note that $Y_{k,\gamma} \in \mathbb{R}^{|S_\gamma|}$ only uses the measurement variables selected by the sensor schedule γ . Further, $K_{*,k}^\gamma \in \mathbb{R}^{n \times |S_\gamma|}$ contains the optimal Kalman filter coefficients for estimating the state x_k using measurement variables $Y_{k,\gamma}$. This is different from the gain matrix used in the recursive form of the filter. Each row $i \in [n]$ of $K_{*,k}^\gamma$ contains the coefficients of the optimal linear combination of the measurement variables used in estimating the i^{th} entry of the state x_k .

In summary, given the schedule γ , the Kalman filter computes the optimal linear estimate $\hat{x}_{k|k}^\gamma$ of the state x_k and provides the *a posteriori* error covariance estimate $\Sigma_{k|k}^\gamma$. Our goal is to compute the schedule γ that minimizes a function related to $\Sigma_{k|k}^\gamma$, subject to resource constraints on γ .

Problem 1 (Sensor Scheduling). Given a time horizon $T > 0$, cost matrices $Q_k \in \mathbb{R}^{n \times n}$ with $M_k := Q_k' Q_k \in \mathbb{S}_+^n$, constraint matrix $H \in \mathbb{R}^{h \times mT}$, and budget $b \in \mathbb{R}^h$, compute the sensor schedule γ that solves

$$\begin{aligned} &\text{minimize}_{\gamma \in \{0,1\}^{mT}} \sum_{k=1}^T \text{trace} \left(Q_k \Sigma_{k|k}^\gamma Q_k' \right) \\ &\text{subject to } H\gamma \leq b. \end{aligned} \quad (8)$$

The objective and constraints capture several interesting problems. For example, consider the following objectives.

- 1) *Total Error:* $Q_k = \theta_k \mathbb{I}_n, k \in [T]$, where $\theta_k \in \mathbb{R}_{\geq 0}$.
- 2) *Final State Error:* $Q_1 = \dots = Q_{T-1} = 0, Q_T = \mathbb{I}_n$.
- 3) *LQG Sensing Design:* It is known that the optimal strategy for LQG control is to provide state estimates via Kalman filtering and then perform LQR control using the state estimate in place of the true state. A similar principle holds in LQG control and sensing co-design [7, Theorem 1] where one designs a sensor selection policy as well as the optimal controller. The result states the budgeted sensor selection problem can be decoupled from the control design where the sensor schedule is computed to optimize $\text{trace}(\Theta \Sigma_{k|k}^\gamma)$, and Θ is computed from the LQG system parameters.

Further, the polyhedral constraints in $H\gamma \leq b$ are general and capture different types of resource constraints.

- 1) *Sensor Selection:* One commits to a sensor subset of size p for all steps: $\sum_{i=1}^m \gamma_i = p$ and $\gamma_i = \gamma_{i+jm}$, for $i \in [m], j \in [T-1]$ effectively using only m variables.
- 2) *Sensor Scheduling:* One selects a sensor subset for each time step $k \in [T]$ of size p : $\sum_{i=1}^m \gamma_{(k-1)m+i} = p$.
- 3) *Energy Constraints:* If sensor i incurs an energy cost α_i each time it is switched on and has an energy budget β_i , then the constraint can be represented as $\sum_{k=1}^T \gamma_{(k-1)m+i} \leq \beta_i / \alpha_i$.

Various problems in the literature are modeled as combinations of these objectives and constraints. For example, objective 1) and constraint 1) model the *sensor selection* problem [11], [12], [15], [17], [18], [20], [24], [29]. Combining the accumulated error in objective 2) and constraint 2) is sensor scheduling with budget constraints [27], [38].

IV. MIXED INTEGER PROGRAM FOR SENSOR SCHEDULING

This section describes our MIQP formulation. As discussed in the introduction, one requires good formulations in mixed integer programming to compute optimal solutions relatively quickly [33]. For example, if the objective is convex (when the binary variables are relaxed), one can quickly solve the ensuing convex optimization problem to get a lower bound on the optimal solution to the mixed integer program. This helps in eliminating parts of the search tree by comparing lower bounds (by solving the relaxed problem) and upper bounds given by feasible solutions. In the context of sensor scheduling, it is not immediately obvious how one would model the objective in (8) as a convex function.

The convex MIQP formulation is presented in Section IV-A with Theorem 1 giving the main result. Note that the results in Section IV-A depend on the covariance matrices between the states and observations in system (6), which are fully specified in Section IV-B.

A. Binary Convex Reformulation

We wish to rewrite the objective in (8) purely in terms of continuous variables. By exploiting the optimality of the Kalman filter, we reformulate the summands in (8) as convex minimization problems where the continuous variables are the coefficients of linear filters. The key idea is to optimize over the class of linear filters using *all* sensor measurement variables with the constraint that a measurement variable i can only be used if its corresponding entry in the schedule satisfies $\gamma_i = 1$. This is established in Lemma 2.

Lemma 2. Given a schedule $\gamma \in \{0, 1\}^{mT}$ and a time step $k \in [T]$, each summand in (8) can be represented as the following optimization problem

$$\begin{aligned} \text{trace} \left(Q_k \Sigma_{k|k}^\gamma Q_k' \right) &= \min \left\{ c_k(K) : [K]_{ij}(1 - \gamma_j) = 0, \right. \\ &\quad \left. i \in [n], j \in [mk], \right. \\ &\quad \left. K \in \mathbb{R}^{n \times mk} \right\}, \end{aligned} \quad (9)$$

where $c_k : \mathbb{R}^{n \times mk} \rightarrow \mathbb{R}_{\geq 0}$ is the expected squared error of a linear filter specified by coefficients K and is defined as

$$\begin{aligned} c_k(K) &:= \mathbb{E} \left[(x_k - KY_k)' M_k (x_k - KY_k) \right] \\ &= \text{trace} \left(M_k \left(K \Sigma_{Y_k Y_k} K' - 2 \Sigma_{x_k Y_k} K' + \Sigma_{x_k x_k} \right) \right). \end{aligned} \quad (10)$$

Note that we delay the definition of the covariance matrices appearing in c_k until Section IV-B since it is not crucial to understanding the formulated MIQP. The interested reader can refer directly to Equation (16) for the exact definition of the covariance matrices. Next, we show that c_k is convex. The proof is omitted and can be found in [39].

Lemma 3 (Convexity). The function $c_k : \mathbb{R}^{n \times mk} \rightarrow \mathbb{R}_{\geq 0}$ defined in Lemma 2 is convex.

Finally, we establish that the sensor scheduling problem (8) can be represented as a mixed integer convex quadratic program, which can be tackled by solvers like GUROBI [34].

Theorem 1 (MIQP for Sensor Scheduling). The sensor scheduling problem (8) can be formulated as the following mixed-integer convex optimization problem

$$\begin{aligned} \min_{\substack{\gamma \in \{0,1\}^{mT} \\ K_k \in \mathbb{R}^{n \times mk}}} & \sum_{k=1}^T c_k(K_k) \\ \text{subject to} & H\gamma \leq b, \\ & [K_k]_{ij} (1 - \gamma_j) = 0, i \in [n], j \in [mk], k \in [T] \end{aligned} \quad (11)$$

where the functions $c_k(\cdot)$ are given by Lemma 2.

The optimization problem (11) is a MIQP of the sensor scheduling problem where the decision variables are the filter coefficients K_k and binary sensor schedule γ . The last constraint allows a filter coefficient to be used only if the corresponding sensor variable is turned on. This constraint can be implemented as a SOS Type-1 constraint in GUROBI or CPLEX. It should be noted that the number of binary variables is mT and the number of continuous variables is $\sum_{k=1}^T nmk = nm \frac{T(T+1)}{2}$. For high-dimensional systems and long horizons, this can quickly become intractable. It would be desirable to have a minimal formulation purely in terms of binary variables γ , which we leave to future work.

Remark (Comparison with [22]). At first glance, Theorem 1 seems similar to the quadratic program in [22]. However, the key difference is the relationship between the continuous and binary variables. In Theorem 1, there is one indicator constraint for each entry of the matrix K_k which is easy to implement in modern solvers and also leads to fast solve times as evidenced in Section V. In [22], the constraint is more involved. It has the form $g(X) \leq \gamma_i$, where γ_i is a binary decision variable and g is the ℓ_0 norm of the sum of ℓ_1 norms of the columns of the matrix of continuous variables X . This constraint cannot be directly implemented in modern mixed integer solvers. One has to setup indicator variables to model the relationship between γ_i and the ℓ_0

norm. Unfortunately, this does not yield fast solve times as shown in Section V. It should be noted that [22] did not set out to solve an integer program; it relaxed the formulation to compute approximate solutions using an iterative weighting technique which yielded good solutions in simulations.

B. Covariance Matrices

The last piece is to specify the covariance matrices used in the definition of the functions c_k in (10). First, we require a representation of state x_j in terms of the initial state x_1 .

Observation 1 (State Representation). For any $j \in \mathbb{N}$,

$$x_j = A^{j-1} x_1 + \sum_{i=1}^{j-1} A^{j-i-1} w_i, \quad (12)$$

Observation 2 (Zero-mean States). For any $j \in \mathbb{N}$, $\mathbb{E}(x_j) = 0$. This follows from the combination of Observation 1 and the fact that the initial state x_0 and process noise w_t are zero-mean random variables.

Using these two observations, we will establish the structure of the covariance between any two states x_i and x_j .

Observation 3 (Covariance between States). For $j \leq i$, the covariance between states x_i and x_j is given by

$$\begin{aligned} \Sigma_{x_i x_j} &= \mathbb{E} \left[\left(A^{i-1} x_1 + \sum_{t=1}^{i-1} A^{i-t-1} w_t \right) \right. \\ &\quad \left. \left(A^{j-1} x_1 + \sum_{t=1}^{j-1} A^{j-t-1} w_t \right)' \right] \in \mathbb{R}^{n \times n} \\ &= \mathbb{E} \left(A^{i-1} x_1 x_1' A^{j-1'} + \sum_{t=1}^{j-1} A^{i-t-1} w_t w_t' A^{j-t-1'} \right) \\ &= A^{i-1} \Sigma_{1|0} A^{j-1'} + \sum_{t=1}^{j-1} A^{i-t-1} W (A^{j-t-1})'. \end{aligned} \quad (13)$$

For $j > i$, $\Sigma_{x_i x_j} = \Sigma'_{x_j x_i}$.

Since the measurement is a linear combination of the state (6), we can determine the covariance between any state and measurement variable and also between measurement variables. This is described in the following observations.

Observation 4 (Covariance between State and Measurement). For $j \leq t$, the matrix of covariances between state $x_t \in \mathbb{R}^n$ and all sensor observation $y_j \in \mathbb{R}^m$ is given by

$$\begin{aligned} \Sigma_{x_t y_j} &= \mathbb{E} \left(x_t (C x_j + v_j)' \right) \\ &= \Sigma_{x_t x_j} C' \in \mathbb{R}^{n \times m} \end{aligned} \quad (14)$$

Observation 5 (Covariance between Measurements). The covariance between measurements y_i and y_j is given by

$$\begin{aligned} \Sigma_{y_i y_j} &= \mathbb{E} \left((C x_i + v_i) (C x_j + v_j)' \right) \\ &= C \Sigma_{x_i x_j} C' + \mathbb{E}(v_i v_j') \in \mathbb{R}^{m \times m} \end{aligned} \quad (15)$$

If $i \neq j$ then $\mathbb{E}(v_i v_j') = 0$ (Kalman filter assumptions) and is equal to $V \in \mathbb{S}_{++}^m$ (noise covariance) otherwise.

We now establish the form of the matrices in Theorem 1 whose entries are given by the observations above.

$$\begin{aligned} \Sigma_{x_k Y_k} &= \begin{bmatrix} \Sigma_{x_k y_1} & \cdots & \Sigma_{x_k, y_k} \end{bmatrix} \in \mathbb{R}^{n \times mk}, \\ \Sigma_{Y_k Y_k} &= \begin{bmatrix} \Sigma_{y_1 y_1} & \cdots & \Sigma_{y_1 y_k} \\ \vdots & \ddots & \vdots \\ \Sigma_{y_k y_1} & \cdots & \Sigma_{y_k y_k} \end{bmatrix} \in \mathbb{S}_{++}^{mk}. \end{aligned} \quad (16)$$

By substituting (16) into (10), the MIQP provided in Theorem 1 is now fully specified. This forms our approach to solving the general sensor scheduling problem.

V. EVALUATION

To evaluate our proposed MIQP, we consider two variants involving budget constraints: sensor selection (Section V-A) and sensor scheduling (Section V-B). For both problems, we compare against

- *Mo et al.* [22]: We convert the quadratic program using an ℓ_0 constraint to a MIQP using an indicator constraint (see Section IV-A for a detailed discussion).
- Greedy algorithm: We keep selecting sensors one at a time until the budget constraint is met. In each step, the sensor yielding the highest reduction in error is chosen. For more details, see [11], [17].

We now describe the system parameters. The process matrix $A \in \mathbb{R}^{n \times n}$ is drawn from a uniform distribution on $[0, 1]$ and normalized so that the magnitude of the largest eigenvalue is 0.5. The measurement matrix $C \in \mathbb{R}^{m \times n}$ is drawn from a uniform distribution on $[0, 1]$. The measurement noise matrix is constructed as $W = LL' \in \mathbb{S}_+^n$ where $L \in \mathbb{R}^{n \times n}$ is drawn from a uniform distribution on $[0, 1]$. Finally, the noise matrix is $V = \sigma^2 \mathbb{I}_m$ where $\sigma^2 = 0.01$ and the initial state covariance is $\Sigma_{1|0} = \mathbb{I}_n$. We run each algorithm on 20 trials with each trial drawing a new set of system parameters. The experiments are implemented in GUROBI [34] on an AMD Ryzen 7 2700 8-core processor with 16 GB of RAM.

For both selection and scheduling, the goal is to minimize the final state estimation error. Thus, the cost matrices $Q_k \in \mathbb{R}^{n \times n}$ are defined as $Q_k = 0$ for $k \in [T - 1]$ and $Q_T = \mathbb{I}_n$. The definitions of the constraint matrix and vector are given in Section III. We now proceed to discuss the results.

A. Sensor Selection with Budget Constraints

The first question we seek to answer is how the runtime of the MIQP scales as the system size increases. We set a timeout of 120 seconds for each algorithm. The results of this experiment are shown in Figure 1. We see that the approach of [22] does not scale favourably and it times out on systems with $n > 30$. On the other hand, the proposed MIQP solves these instances in under a second which is on average, roughly 50 times faster for a system with 25 states. However, the proposed MIQP also shows an increase in runtime as the system size increases albeit at a much slower rate. In fact, until systems of size $n = 35$, the runtime is comparable with greedy. The runtime of greedy is extremely efficient solving all instances in under a second. Further, the approach of [22] exhibits high variance in its runtime which is in contrast to

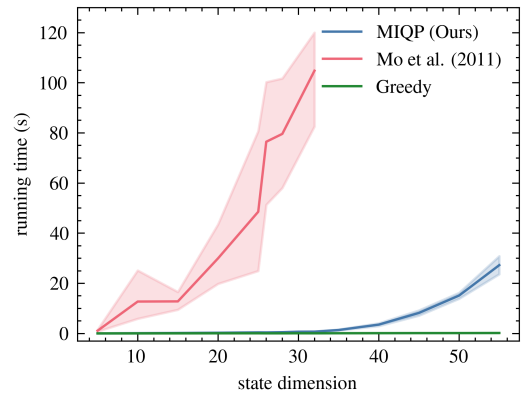


Fig. 1. **Sensor Selection:** Runtime comparison of our MIQP, [22], and the greedy algorithm. The parameters are $T = 3$ with $m = 10$ sensors and a budget of $p = 5$. The x -axis indicates the state dimension while the y -axis is the running time measured in seconds.

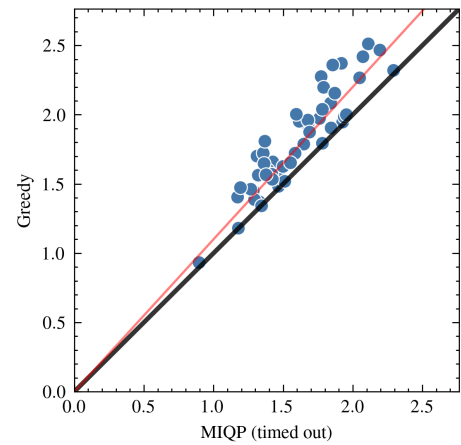


Fig. 2. **Sensor Selection:** Solution quality comparison of our MIQP (timed out after 10 seconds) and the greedy algorithm. The x and y coordinate denote the MIQP (timed out) solution and the greedy solution respectively.

the proposed MIQP and the greedy algorithm which exhibit low runtime variability. In summary, the MIQP has a better runtime when compared to [22] with timings up to 40 times faster while providing *optimal* solutions.

We also investigate the quality of the solution returned by the MIQP when given a time out of 10 seconds. In this situation, it would be apt to compare it to greedy since both return approximate solutions. The setup for this experiment involves fixing the state dimension $n = 10$, horizon $T = 3$, number of sensors $m = 25$, and budget $p = 5$. We select a time out of 10 seconds as it is practical assumption on the runtime for an algorithm. We draw 50 random instances of the system parameters and run both algorithms to get their solutions. We observed that the MIQP timed out on *all* drawn instances. For each instance i , we plot (x_i, y_i) where x_i is the objective value of the solution returned by the MIQP and y_i is the objective value of the greedy solution on instance i . We plot these points in Figure 2. We see that all points lie above the black line $y = x$ indicating that even in this challenging setting, the MIQP outperforms greedy on all instances even

though it times out. The red line is the function $y = 1.1x$ and thus points that lie above it are instances where the greedy obtains solutions that are at least 10% worse than the MIQP solution. We recorded the mean error difference to be 13%. To summarize, these results suggest the 10 second timed out MIQP returns solutions of quality better or equal to that of the greedy algorithm. The added benefit is that the MIQP gives a certificate of suboptimality even though it times out whereas the greedy algorithm, in general, cannot.

B. Sensor Scheduling with Budget Constraints

We now study the runtime of each approach for sensor scheduling problems. The results are shown in Figure 3. Note that sensor scheduling is a more challenging problem than sensor selection as one has to select a sensor subset for each time step. We see the approach of [22] can only tackle systems of size up to $n = 14$ before timing out (set to 100 seconds). In contrast, our MIQP solves the same instances in under a second giving improvements of up to 40 to 60 times. The greedy remains extremely efficient and provides solutions in under a second for all instances. The summary here is that our MIQP can tackle systems of size up to $n = 35$ while providing optimal solutions before starting to display an increase in runtime, demonstrating its effectiveness.

Our final experiment compares the solution quality of the MIQP, with a 10 second time out, to the greedy algorithm. This follows the same setup used in the second experiment for sensor selection (Section V-A) which generates 50 random instances. In this setting, the MIQP was observed to time out on *all* drawn instances. The results are shown in Figure 4. The observations are similar to that in sensor selection. We note the MIQP always returns solutions whose quality is better or equal to that of the greedy algorithm. In addition, the points above the red line are the instances where the greedy solution is at least 10% worse. When compared to the results in sensor selection (Figure 2), the number of points above the red line is greater in Figure 4. This indicates the performance difference between greedy and the MIQP is larger in sensor scheduling which is also evidenced by the average error difference recorded to be 19%.

VI. CONCLUSIONS

We studied the generalized version of the sensor scheduling problem capturing problems such as sensor placement, scheduling, and LQG sensing design. Our approach was rooted in mixed integer optimization. We formulated a mixed integer quadratic program by exploiting the optimality of the Kalman filter. In simulations, we showed the effectiveness of the approach in computing optimal solutions to systems with 30 to 50 states. In addition, the solver also returned better quality solutions over the popular greedy algorithm when constrained to time out within a few seconds. Looking forward, we believe a minimal formulation that solely contains integer variables would be desirable. This is because the number of continuous variables in the MIQP scales quadratically with the time horizon whereas the number of integer variables only depends on the number of sensors.

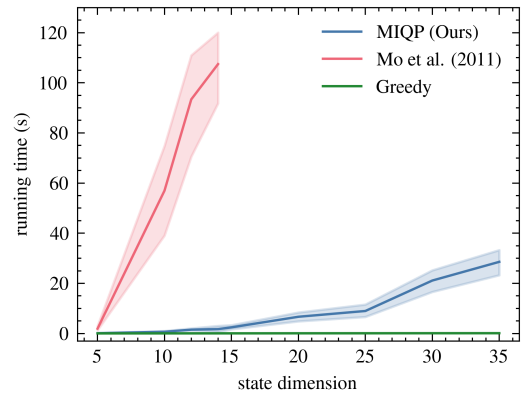


Fig. 3. **Sensor Scheduling:** Runtime comparison of different algorithms on varying system sizes. The system parameters are $T = 3$, $m = 10$, and budget $p = 5$. The runtime is measured in seconds indicated on the y -axis.

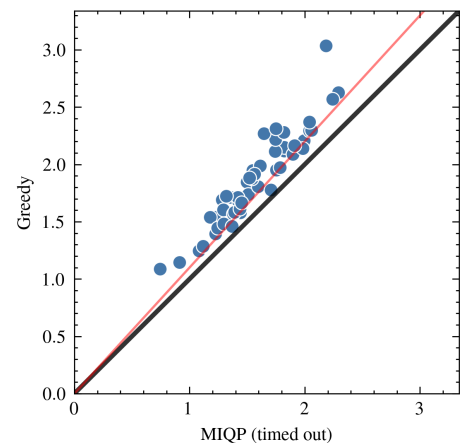


Fig. 4. **Sensor Scheduling:** Comparison of solutions the MIQP (timed out after 10 seconds) and the greedy algorithm. The points above the black line indicate MIQP obtaining better solutions than greedy. The greedy solution of points above the red line are at least 10% worse than the MIQP solution.

Further, it would also be interesting to study the structure of the relaxed problem and identify any avenues for the development of approximation algorithms.

REFERENCES

- [1] J. Le Ny and G. J. Pappas, “On Trajectory Optimization for Active Sensing in Gaussian Process Models,” in *Conference on Decision and Control (CDC)*. IEEE, 2009, pp. 6286–6292.
- [2] X. Lan and M. Schwager, “Rapidly Exploring Random Cycles: Persistent Estimation of Spatiotemporal Fields with Multiple Sensing Robots,” *IEEE Transactions on Robotics*, pp. 1230–1244, 2016.
- [3] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, “A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers,” *IEEE Transactions on Robotics*, 2023.
- [4] A. Olshevsky, “Minimal Controllability Problems,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.
- [5] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, “Minimal Actuator Placement with Bounds on Control Effort,” *IEEE Transactions on Control of Network Systems*, pp. 67–78, 2015.
- [6] T. H. Summers, F. L. Cortesi, and J. Lygeros, “On Submodularity and Controllability in Complex Dynamical Networks,” *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, 2015.
- [7] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, “LQG Control and Sensing Co-Design,” *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1468–1483, 2020.

- [8] G. Zardini, A. Censi, and E. Frazzoli, "Co-design of Autonomous Systems: From Hardware Selection to Control synthesis," in *European Control Conference (ECC)*. IEEE, 2021, pp. 682–689.
- [9] M. Siami and A. Jadbabaie, "A Separation Theorem for Joint Sensor and Actuator Scheduling with Guaranteed Performance Bounds," *Automatica*, vol. 119, p. 109054, 2020.
- [10] Shamaiah, Manohar and Banerjee, Siddhartha and Vikalo, Haris, "Greedy Sensor Selection: Leveraging Submodularity," in *Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 2572–2577.
- [11] S. T. Jawaid and S. L. Smith, "Submodularity and Greedy Algorithms in Sensor Scheduling for Linear Dynamical Systems," *Automatica*, vol. 61, pp. 282–288, 2015.
- [12] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Sensor Placement for Optimal Kalman Filtering: Fundamental Limits, Submodularity, and Algorithms," in *American Control Conference (ACC)*. IEEE, 2016.
- [13] —, "Near-optimal Sensor Scheduling for Batch State Estimation: Complexity, Algorithms, and Limits," in *Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2695–2702.
- [14] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor Selection for Kalman Filtering of Linear Dynamical Systems: Complexity, Limitations and Greedy Algorithms," *Automatica*, vol. 78, pp. 202–210, 2017.
- [15] P. Singh, M. Chen, L. Carlone, S. Karaman, E. Frazzoli, and D. Hsu, "Supermodular Mean Squared Error Minimization for Sensor Scheduling in Optimal Kalman Filtering," in *American Control Conference (ACC)*. IEEE, 2017, pp. 5787–5794.
- [16] L. Ye, N. Woodford, S. Roy, and S. Sundaram, "On the Complexity and Approximability of Optimal Sensor Selection and Attack for Kalman Filtering," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2146–2161, 2020.
- [17] L. F. Chamon, G. J. Pappas, and A. Ribeiro, "Approximate Supermodularity of Kalman Filter Sensor Selection," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 49–63, 2020.
- [18] A. Kohara, K. Okano, K. Hirata, and Y. Nakamura, "Sensor Placement Minimizing the State Estimation Mean Square Error: Performance Guarantees of Greedy Solutions," in *Conference on Decision and Control*. IEEE, 2020, pp. 1706–1711.
- [19] S. Joshi and S. Boyd, "Sensor Selection via Convex Optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2008.
- [20] J. E. Weimer, B. Sinopoli, and B. H. Krogh, "A Relaxation Approach to Dynamic Sensor Selection in Large-Scale Wireless Networks," in *International Conference on Distributed Computing Systems Workshops*. IEEE, 2008, pp. 501–506.
- [21] N. Moshtagh, L. Chen, and R. Mehra, "Optimal Measurement Selection for Any-Time Kalman Filtering with Processing Constraints," in *Conference on Decision and Control*. IEEE, 2009, pp. 5074–5079.
- [22] Y. Mo, R. Ambrosino, and B. Sinopoli, "Sensor Selection Strategies for State Estimation in Energy Constrained Wireless Sensor Networks," *Automatica*, vol. 47, no. 7, pp. 1330–1338, 2011.
- [23] Y. Wang, M. Sznajder, and F. Dabbene, "A Convex Optimization Approach to Worst-Case Optimal Sensor Selection," in *Conference on Decision and Control*. IEEE, 2013, pp. 6353–6358.
- [24] A. Carmi and P. Gurfil, "Sensor Selection via Compressed Sensing," *Automatica*, vol. 49, no. 11, pp. 3304–3314, 2013.
- [25] U. Münz, M. Pfister, and P. Wolfsum, "Sensor and Actuator Placement for Linear Systems Based on H_2 and H_∞ Optimization," *IEEE Transactions on Automatic Control*, pp. 2984–2989, 2014.
- [26] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo, "An ADMM Algorithm for Optimal Sensor and Actuator Selection," in *Conference on Decision and Control*. IEEE, 2014, pp. 4039–4044.
- [27] D. Maity, D. Hartman, and J. S. Baras, "Sensor Scheduling for Linear Systems: A Covariance Tracking Approach," *Automatica*, vol. 136, p. 110078, 2022.
- [28] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and Randomized Actuator Scheduling with Guaranteed Performance Bounds," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1686–1701, 2020.
- [29] A. Hashemi, M. Ghasemi, H. Vikalo, and U. Topcu, "Randomized Greedy Sensor Selection: Leveraging Weak Submodularity," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 199–212, 2020.
- [30] C. I. Calle and S. D. Bopardikar, "Probabilistic Performance Bounds for Randomized Sensor Selection in Kalman Filtering," in *American Control Conference (ACC)*. IEEE, 2021, pp. 4395–4400.
- [31] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*. McGraw-Hill Higher Education New York, 2008.
- [32] D. Bertsimas, A. King, and R. Mazumder, "Best Subset Selection via a Modern Optimization Lens," *The Annals of Statistics*, vol. 44, no. 2, pp. 813–852, 2016.
- [33] O. Günlük and J. Linderoth, "Perspective Reformulation and Applications," in *Mixed Integer Nonlinear Programming*. Springer, 2011, pp. 61–89.
- [34] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: <https://www.gurobi.com>
- [35] I. I. Cplex, "V12. 1: User's Manual for CPLEX," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.
- [36] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a Stochastic Sensor Selection Algorithm with Applications in Sensor Scheduling and Sensor Coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.
- [37] L. Zhao, W. Zhang, J. Hu, A. Abate, and C. J. Tomlin, "On the Optimal Solutions of the Infinite-Horizon Linear Sensor Scheduling Problem," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2825–2830, 2014.
- [38] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On Efficient Sensor Scheduling for Linear Dynamical Systems," *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.
- [39] S. Dutta, N. Wilde, and S. L. Smith, "A Unified Approach to Optimally Solving Sensor Scheduling and Sensor Selection Problems in Kalman Filtering," *arXiv preprint arXiv:2304.02692*, 2023.
- [40] Anderson, Brian DO and Moore, John B, *Optimal Filtering*. Courier Corporation, 2012.