



Brief paper

Sensor scheduling for linear systems: A covariance tracking approach[☆]Dipankar Maity^{a,*}, David Hartman^b, John S. Baras^b^a Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, NC, 28223, USA^b Department of Electrical and Computer Engineering, University of Maryland at College Park, MD, 20742, USA

ARTICLE INFO

Article history:

Received 26 July 2020

Received in revised form 11 July 2021

Accepted 17 October 2021

Available online xxx

Keywords:

Kalman filter

Sensor design

Semidefinite programming

Sensor scheduling

ABSTRACT

We consider the classical sensor scheduling problem for linear systems where only one sensor is activated at each time. We show that the sensor scheduling problem has a close relation to the *sensor design* problem and the solution of a sensor schedule problem can be extracted from an equivalent sensor design problem. We propose a convex relaxation to the sensor design problem and a reference covariance trajectory is obtained from solving the relaxed sensor design problem. Afterwards, a *covariance tracking* algorithm is designed to obtain an approximate solution to the sensor scheduling problem using the reference covariance trajectory obtained from the sensor design problem. While the sensor scheduling problem is NP-hard, the proposed framework circumvents this computational complexity by decomposing this problem into a convex sensor design problem and a covariance tracking problem. We provide theoretical justification and a sub-optimality bound for the proposed method using dynamic programming. The proposed method is validated over several experiments portraying the efficacy of the framework.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

1.1. Motivation and prior work

Advancements in network control systems, distributed systems, and the development of multi-agent autonomous systems for surveillance require the development of efficient algorithms allocating resources to manage the sensory data originating from a large number of sensors observing different parts of a single or distributed system, see for example, [Evans et al. \(2005\)](#), [Gupta et al. \(2006\)](#), [Maity and Baras \(2015\)](#), [Williams \(2007\)](#) and [Zhang and Hristu-Varsakelis \(2006\)](#). These problems have a long history starting with [\(Meier et al., 1967\)](#) and [Athans \(1972\)](#), and they include static sensor scheduling problems as well as trajectory optimization scenarios for mobile sensors, e.g., [Williams \(2007\)](#). In the sensor scheduling problem, we aim to minimize an error criteria (e.g., the mean square error) where the error is dependent on sensor measurements over a fixed time horizon. We are constrained by the number of sensors that can be activated at each time. This problem has many applications including estimation of spatial phenomenon in [Nowak et al. \(2004\)](#), target tracking in [Masazade et al. \(2012\)](#), robot navigation in [Vitus \(2011\)](#).

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Luca Schenato under the direction of Editor Christos G. Cassandras.

* Corresponding author.

E-mail addresses: dmaity@uncc.edu (D. Maity), dhartma2@umd.edu (D. Hartman), baras@umd.edu (J.S. Baras).

Different methods have been applied to solve these problems. For example, in [Nowak et al. \(2004\)](#) the scheduler is found by employing hierarchical sensor networks to trade-off the mean square error and communication cost. Whereas, a sparsity promoting penalty function is added to the objective function to help find a scheduler in [Masazade et al. \(2012\)](#), and, in [Vitus \(2011\)](#), a scheduler is found by solving a novel incremental optimization problem.

The optimization problems solving for optimal sensor schedules are generally mixed-integer nonlinear programs, and thus, quickly become intractable. Often these optimization problems do not possess any amenable structure that could be exploited to reduce their computational complexities. Owing to this difficulty, a whole array of approximation attempts have been proposed to solve these problems. The work of [Meier et al. \(1967\)](#) proposed a solution that checks all possible sensor schedules, whereas, [Vitus et al. \(2012\)](#) devised a solution that prunes the exponentially sized search tree to reduce the search space at the expense of added computation due to pruning. [Joshi and Boyd \(2008\)](#) relaxed the problem into a convex optimization problem, a heuristic that often works well in practice. [He and Chong \(2004\)](#) modeled the sensor scheduling problem as a partially observable Markov decision process (POMDP) and proposed approximate solutions to solve this POMDP. A stochastic optimization based solution for an infinite-horizon steady state problem is addressed by [Gupta et al. \(2006\)](#). Greedy solutions to the sensor scheduling problem have been proposed as well, such as [Wang et al. \(2004\)](#). [Chhetri](#)

et al. (2007) propose a greedy solution that also includes integer programming. Some of these greedy solutions leverage ideas from submodularity, e.g., Tzoumas et al. (2016). These approaches also include optimizing a slightly different objective function (e.g., convexification of the objective). Additionally, Liu et al. (2014) has proposed an optimal sensor schedule by restricting the scheduling in the class of periodic functions. The existence of periodic sensor scheduling has been proven in Orihuela et al. (2014) where a sub-optimal one-step ahead strategy is thoroughly studied as a possible example of observation scheme.

1.2. Contribution and outline of the paper

We revisit the sensor scheduling problem of linear Gaussian systems and recast it as a sensor design problem, which, to the best of our knowledge, has not been explored in the past research. While in a sensor scheduling problem we search for the optimal schedule for a given set of sensors, in a sensor design problem (details are provided in Section 3.1) we design the optimal sensors. These two classes of problems are treated differently as their primary objectives are different. However, sensor scheduling problems have a close connection to sensor design problems and we show that the former can be expressed as a special case of the latter. We further demonstrate that a class of sensor design problems can be solved by convex optimization. Using this derived connection between sensor scheduling problems and sensor design problems, an approximate solution of a scheduling problem to select one sensor at any time is constructed. While some of the prior works e.g., Joshi and Boyd (2008) and Tzoumas et al. (2016) modify the cost function of a scheduling problem to make it convex, we show that the sensor design problem is already a convex optimization problem when the sensor parameters are restricted to a convex set. In this approach, we thus avoid the need to solve mixed integer programming problems which are inherent to sensor scheduling problems.

The contributions of this work are as follows: First, we show that there is a one-to-one connection between sensor scheduling problems and sensor designing problems. Second, we show that sensor designing problems are indeed convex when the optimization parameters lie within convex sets. Third, from the equivalent design problem of a scheduling problem, we obtain a *reference covariance trajectory* that is used by our tracking algorithm (Algorithm 1) to find a sensor schedule. Fourth, we use an approximate dynamic programming based argument to provide guarantees and a sub-optimality bound of our algorithm. Finally, through numerical experiments, we demonstrate the efficacy of our proposed method over some existing methods, e.g., Gupta et al. (2006) and Tzoumas et al. (2016).

The rest of the paper is structured as follows: In Section 2, the problem is formulated as a nonlinear integer program. In Section 3.1, we propose a solution to this problem where we construct a sensor design problem, and then in Section 3.2, we provide a covariance-tracking algorithm to obtain the solution of the original sensor scheduling problem from the solution of the sensor design problem. We analyze the performance of our algorithm in Section 3.3 using dynamic programming based arguments. Numerical analysis on the performance of our approach is provided in Section 4. Finally, we conclude this article in Section 5.

2. Problem formulation

We consider a linear system of the form

$$X_{t+1} = A_t X_t + W_t, \quad (1)$$

where $X_t \in \mathbb{R}^n$, $X_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ is the initial state, $\{W_t\}_{t=0}^T$ is an i.i.d sequence of Gaussian random variable with $W_0 \sim \mathcal{N}(0, \mathcal{W})$,

and W_t is independent of X_0 (denoted as $W_t \perp\!\!\!\perp X_0$) for all t . The dynamical system (1) is equipped with N sensors which are described by

$$Y_t^i = C_t^i X_t + V_t^i, \quad i \in \mathbb{N} \triangleq \{1, \dots, N\}, \quad (2)$$

where $\{V_t^i\}_{t=0}^T$ is an i.i.d sequence of Gaussian random variables with $V_0^i \sim \mathcal{N}(0, \mathcal{V}^i)$. Furthermore, for all t, s and $i \neq j \in \mathbb{N}$, V_t^i and V_s^j are independent, i.e., $V_t^i \perp\!\!\!\perp V_s^j$, and also $V_t^i \perp\!\!\!\perp X_0$, $V_t^i \perp\!\!\!\perp W_s$.

Only one out of the N sensors are used at any given time to obtain the measurements, which are then used in estimating the state of the system (1). Let $\sigma : [0, T] \rightarrow \mathbb{N}$ be a sensor schedule function such that $\sigma(t) = i$ denotes that the i th sensor is used at time t to obtain the measurement Y_t^i corresponding to sensor i . Thus, the received measurements up until time t can be represented as $\mathcal{Y}_t(\sigma) \triangleq \{Y_0^{\sigma(0)}, \dots, Y_t^{\sigma(t)}\}$. For a given schedule σ , the estimation error and the estimation error covariance at time t are defined as $e_t(\sigma) \triangleq X_t - \mathbb{E}[X_t | \mathcal{Y}_t(\sigma)]$ and $P_t(\sigma) \triangleq \mathbb{E}[e_t(\sigma)e_t(\sigma)^T]$, respectively. The objective is to find a sensor schedule σ to minimize a cumulative expected quadratic error $\sum_{t=0}^T \mathbb{E}[e_t(\sigma)^T e_t(\sigma)]$ over a finite horizon $[0, T]$, that is, to minimize $\sum_{t=0}^T \text{tr}(P_t(\sigma))$.

We define the following two matrix valued functions to maintain brevity in the subsequent analysis.

$$g_t(i, M) = M - M C_t^i (C_t^i M C_t^{iT} + \mathcal{V}^i)^{-1} C_t^i M, \quad (3a)$$

$$h_t(M) = A_{t-1} M A_{t-1}^T + \mathcal{W}. \quad (3b)$$

From Kalman filtering theory, we obtain that, for all t ,

$$P_t(\sigma) = g_t(\sigma(t), P_{t-1}(\sigma)), \quad (4a)$$

$$P_{t|t-1}(\sigma) = h_t(P_{t-1}(\sigma)), \quad P_{0|0} = \Sigma_0. \quad (4b)$$

The optimal sensor scheduling problem is as follows.

Problem 1 (Sensor Scheduling Problem). Given a system (1)–(2), find a schedule $\sigma : [0, T] \rightarrow \mathbb{N}$ that solves the following optimization problem:

$$\min \sum_{t=0}^T \text{tr}(P_t(\sigma))$$

subject to $P_t(\sigma) = g_t(\sigma(t), P_{t-1}(\sigma))$,

$$P_{t|t-1}(\sigma) = h_t(P_{t-1}(\sigma)), \quad P_{0|0} = \Sigma_0,$$

with variables $\sigma, P_t, P_{t|t-1}$.

Problem 1 is combinatorial in nature due to the discrete mapping of the scheduling function $\sigma(\cdot)$, and generally, it is NP-hard, see e.g., Tzoumas et al. (2016). Majority of the prior works rely on integer programming or relaxations to solve Problem 1. In this work, we propose an efficient sub-optimal solution that studies the problem from the perspective of sensor-design rather than sensor scheduling.

3. Optimal sensor schedule

In order to solve Problem 1, we will construct a computationally inexpensive simplified problem that provides a sub-optimal solution to Problem 1. In this section, instead of looking for a schedule $\sigma(\cdot)$, we focus on a seemingly different problem that seeks to design sensors to be used over the horizon $[0, T]$.

Before starting our discussion on the sensor design problem and its relation with the scheduling problem of Problem 1, let us define two matrices Q_t and $Q_{t|t-1}$ (information matrices for Kalman filtering) as follows

$$Q_t(\sigma) \triangleq P_t^{-1}(\sigma), \quad Q_{t|t-1}(\sigma) \triangleq P_{t|t-1}^{-1}(\sigma).$$

Therefore, from (3a) and (4a), we may write:

$$Q_t^{-1}(\sigma) = Q_{t|t-1}^{-1}(\sigma) - Q_{t|t-1}^{-1}(\sigma)C_t^\top(\sigma)S_t^{-1}(\sigma)C_t(\sigma)Q_{t|t-1}^{-1}(\sigma),$$

where $S_t(\sigma) \triangleq C_t(\sigma)P_{t|t-1}(\sigma)C_t(\sigma)^\top + \mathcal{V}_t(\sigma)$ and $C_t(\sigma) \triangleq C_t^{\sigma(t)}$, $\mathcal{V}_t(\sigma) \triangleq \mathcal{V}^{\sigma(t)}$. Using Woodbury equality¹, one may equivalently write

$$Q_t(\sigma) = Q_{t|t-1}(\sigma) + C_t^\top(\sigma)\mathcal{V}_t^{-1}(\sigma)C_t(\sigma). \quad (5)$$

Using the new variables $Q_t(\sigma)$ and $Q_{t|t-1}(\sigma)$, Problem 1 is rewritten as Problem 2.

Problem 2. Given a system (1)–(2), find a schedule $\sigma : [0, T] \rightarrow \mathbb{N}$ that solves the following problem:

$$\min \sum_{t=0}^T \text{tr}(P_t(\sigma))$$

$$\begin{aligned} \text{subject to } P_t(\sigma) &= Q_t^{-1}(\sigma), \quad P_{t|t-1}(\sigma) = h(P_{t-1}(\sigma)), \\ Q_t(\sigma) &= Q_{t|t-1}(\sigma) + C_t^\top(\sigma)\mathcal{V}_t^{-1}(\sigma)C_t(\sigma), \\ Q_{t|t-1}(\sigma) &= P_{t|t-1}^{-1}(\sigma), \quad Q_{0|-1} = \Sigma_0^{-1}, \end{aligned}$$

with variables $\sigma, P_t, P_{t|t-1}, Q_t, Q_{t|t-1}$.

Although, the formulations in Problem 1 and Problem 2 may appear different as their constraints are different, one can verify that these two problems are equivalent, and thus, by solving one, we can recover the solution for the other. Next, we show the advantage of using Problem 2 over Problem 1 for solving the sensor scheduling problem by discussing a sensor design problem.

3.1. Sensor design problem

In this section, we focus on solving a sensor design problem which is closely related to the sensor scheduling problem and provides a reasonable heuristic for solving the latter problem. To that end, a related sensor design problem is presented in Problem 3.

Problem 3. Given a system (1), and the sets C_t and V_t , design a linear sensor $Y_t = C_t X_t + V_t$ where $V_t \sim \mathcal{N}(0, \mathcal{V}_t)$, to solve the following problem:

$$\min \sum_{t=0}^T \text{tr}(P_t)$$

$$\begin{aligned} \text{subject to } P_t &= Q_t^{-1}, \quad Q_t = Q_{t|t-1} + R_t, \\ Q_{t|t-1} &= (h_t(P_{t-1}))^{-1}, \quad Q_{0|-1} = \Sigma_0^{-1}, \\ R_t &= C_t^\top \mathcal{V}_t^{-1} C_t, \quad C_t \in \mathcal{C}_t, \quad \mathcal{V}_t \in \mathcal{V}_t, \end{aligned}$$

with variables $P_t, Q_t, Q_{t|t-1}, R_t, C_t, \mathcal{V}_t$.

Remark 1. By replacing the constraints $C_t \in \mathcal{C}_t, \mathcal{V}_t \in \mathcal{V}_t$ with $(C_t, \mathcal{V}_t) \in \{(C_t^i, \mathcal{V}_t^i)\}_{i \in \mathbb{N}}$ in Problem 3, we recover Problem 2.

We may further relax the constraints in Problem 3 to their equivalent linear matrix inequalities (LMIs) to obtain a relaxed problem. By using $P_t \succeq Q_t^{-1}$ and $Q_{t|t-1} \preceq (h_t(P_{t-1}))^{-1}$, and after some simplifications via using Schur Complement, we obtain the following relaxed problem.

¹ $(A + UCV)^{-1} \triangleq A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$.

Problem 4. For the given dynamical system (1), design a sensor of the form $Y_t = C_t X_t + V_t$, where $V_t \sim \mathcal{N}(0, \mathcal{V}_t)$, to solve the following problem:

$$\begin{aligned} \min \sum_{t=0}^T \text{tr}(P_t) \\ \text{subject to } Q_t &= Q_{t|t-1} + R_t, \\ \begin{bmatrix} P_t & I \\ I & Q_t \end{bmatrix} &\succeq 0, \\ \begin{bmatrix} \mathcal{W}^{-1} - Q_{t|t-1} & \mathcal{W}^{-1}A_{t-1} \\ A_{t-1}^\top \mathcal{W}^{-1} & Q_{t-1} + A_{t-1}^\top \mathcal{W}^{-1}A_{t-1} \end{bmatrix} &\succeq 0, \\ R_t \in \mathbb{R}_t, \quad Q_{0|-1} &= \Sigma_0^{-1}, \end{aligned}$$

with variables $P_t, Q_t, Q_{t|t-1}, R_t$, where $\mathbb{R}_t = \{C_t^\top \mathcal{V}_t^{-1} C_t \mid C_t \in \mathcal{C}_t, \mathcal{V}_t \in \mathcal{V}_t\}$ is a given set of positive semidefinite matrices which is constructed from the sets \mathcal{C}_t and \mathcal{V}_t .

Note that the constraint $R_t \in \mathbb{R}_t$ in Problem 4 is an equivalent representation of the constraints $R_t = C_t^\top \mathcal{V}_t^{-1} C_t, C_t \in \mathcal{C}_t, \mathcal{V}_t \in \mathcal{V}_t$ in Problem 3. Once $\{R_t\}_{t=0}^T$ is found by solving Problem 4, $\{C_t, \mathcal{V}_t\}_{t=0}^T$ can be found by solving

$$C_t^\top \mathcal{V}_t^{-1} C_t = R_t,$$

for all $t = 0, \dots, T$. While Problem 4 is a relaxation of Problem 3, the following theorem states that an optimal solution of Problem 4 is also an optimal solution for Problem 3.

Theorem 1. An optimal solution of the relaxed problem (Problem 4) is also an optimal solution of the original problem (Problem 3), and vice-versa.

Proof. Firstly, due to the relaxations, any feasible solution of Problem 3 is a feasible solution for Problem 4, and hence the optimal solution of Problem 3 is a feasible solution for Problem 4. The theorem is proved once we show that for every feasible solution of Problem 4 there exists a feasible solution for Problem 3 that produces the same, if not a lesser, objective value. In order to prove this, let us consider $\{P_t, Q_t, Q_{t|t-1}\}$ to denote a feasible solution of Problem 4 and let us construct a new tuple $\{\bar{P}_t, \bar{Q}_t, \bar{Q}_{t|t-1}\}$, for all t , as follows:

$$\begin{aligned} \bar{Q}_{0|-1} &= Q_{0|-1} \quad \bar{R}_t = Q_t - Q_{t|t-1}, \quad \bar{P}_t = \bar{Q}_t^{-1}, \\ \bar{Q}_t &= \bar{Q}_{t|t-1} + \bar{R}_t, \quad \bar{Q}_{t+1|t} = (h_{t+1}(\bar{P}_t))^{-1}. \end{aligned} \quad (6)$$

Based on this construction of $\{\bar{P}_t, \bar{Q}_t, \bar{Q}_{t|t-1}\}$, one can verify using mathematical induction that $\bar{P}_t \preceq P_t, \bar{Q}_t \succeq Q_t$ and $\bar{Q}_{t|t-1} \succeq Q_{t|t-1}$ for all t . Matrix \bar{R}_t , as defined in (6), satisfies $\bar{R}_t \in \mathbb{R}_t$. Therefore, based on (6), one can conclude that $\{\bar{P}_t, \bar{Q}_t, \bar{Q}_{t|t-1}\}$ is a feasible solution of Problem 3 since they satisfy all the constraints of Problem 3. Furthermore, since $\bar{P}_t \preceq P_t$, it then follows that $\sum_{t=0}^T \text{tr}(\bar{P}_t) \leq \sum_{t=0}^T \text{tr}(P_t)$. Thus, an optimal solution of Problem 4 is a feasible solution of Problem 3, and vice-versa. This completes the proof. \square

Due to Theorem 1, the LMI-based relaxations introduced in Problem 4 do not affect the optimality since an optimal solution of the relaxed problem is also optimal for the original problem. This is a key advantage of this approach over existing methods. It is noteworthy that the LMI-based relaxations retain the optimality of a sensor-design problem. Furthermore, Problem 4 is a mixed integer semidefinite program and one could use efficient numerical techniques, e.g., Gally et al. (2018) to solve it directly.

Note that Problem 4 is convex when \mathbb{R}_t is a convex set for all t . Moreover, if \mathbb{R}_t is a convex hull of a set of ℓ matrices $\{R_t^1, \dots, R_t^\ell\}$ for all t , then we can replace the constraint $R_t \in \mathbb{R}_t$ with the constraints $R_t = \sum_{i=1}^\ell \theta_i^t R_t^i, \theta_i^t \in [0, 1]$ and $\sum_{i=1}^\ell \theta_i^t = 1$. In this case, Problem 4 can be further simplified to Problem 5.

Problem 5. For the given dynamical system (1), design a sensor of the form $Y_t = C_t X_t + V_t$, where $V_t \sim \mathcal{N}(0, \mathcal{V}_t)$, to solve the following problem:

$$\min \sum_{t=0}^T \text{tr}(P_t)$$

subject to $Q_t = Q_{t|t-1} + \sum_{i=1}^{\ell} \theta_t^i R_t^i, \quad Q_{0|-1} = \Sigma_0^{-1}$

$$\begin{bmatrix} P_t & I \\ I & Q_t \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} \mathcal{W}^{-1} - Q_{t|t-1} & \mathcal{W}^{-1} A_{t-1} \\ A_{t-1}^T \mathcal{W}^{-1} & Q_{t-1} + A_{t-1}^T \mathcal{W}^{-1} A_{t-1} \end{bmatrix} \succeq 0,$$

$$\sum_{i=1}^{\ell} \theta_t^i = 1, \quad 0 \leq \theta_t^i \leq 1.$$

with variables $\theta_t, P_t, Q_t, Q_{t|t-1}$.

At this point, we are ready to address the sensor scheduling problem and its connection to Problems 3–5.

3.2. Sensor scheduling problem

The sensor scheduling problem can be viewed as a sensor design problem if we restrict the design variables (C_t, \mathcal{V}_t) to be one of the elements in the set $\{(C_t^i, \mathcal{V}_t^i)\}_{i \in \mathbb{N}}$ as mentioned in Remark 1. Equivalently, if we restrict $R_t = \{R_t^i\}_{i \in \mathbb{N}}$ in Problem 4, where $R_t^i = C_t^{iT} \mathcal{V}_t^{i-1} C_t^i$, then we recover a solution to Problem 1. However, solving Problem 4 with the non-convex constraint $R_t \in \{R_t^i\}_{i \in \mathbb{N}}$ is computationally expensive despite the availability of efficient numerical techniques, e.g., Gally et al. (2018). Therefore, we relax the constraint $R_t \in \{R_t^i\}_{i \in \mathbb{N}}$ as $R_t \in \text{co}(\{R_t^i\}_{i \in \mathbb{N}})$ where $\text{co}(\cdot)$ denotes the convex hull operation. With this convex hull relaxation approach, the relaxed sensor scheduling problem becomes exactly the same as Problem 5.

By solving a relaxation of Problem 1, as presented in Problem 5, one obtains the variables $\{\{\theta_t^i\}_{i \in \mathbb{N}}\}_{t=0}^T$, or equivalently $R_t^o = \sum_{i=1}^{\ell} \theta_t^i R_t^i$ and the associated P_t^o, Q_t^o and $Q_{t|t-1}^o$. If, $\theta_t^i \in \{0, 1\}$ for all i and t , then this relaxed optimal solution $\{\{\theta_t^i\}_{i \in \mathbb{N}}\}_{t=0}^T$ is an optimal schedule for Problem 1. However, in general, the obtained θ_t^i are not binary-valued, and hence the solution of Problem 5 may not readily be useful as a solution to Problem 1. Therefore, we propose a tracking-based algorithm to use the solution of Problem 5 as a guide to construct a sub-optimal solution to Problem 1.

Algorithm 1 Covariance Tracking Algorithm

Input $\leftarrow \{P_t^o\}_{t=0}^T, P_{0|-1} = \Sigma_0,$

for $t = 0 : T$

$$M_t(i) \leftarrow g_t(i, P_{t|t-1}), \quad i \in \mathbb{N},$$

$$\sigma(t) \leftarrow \text{argmin}_i \|P_t^o - M_t(i)\|_F$$

$$P_t \leftarrow g_t(\sigma(t), P_{t|t-1}),$$

$$P_{t+1|t} \leftarrow h_t(P_t),$$

end

Output $\leftarrow \sigma.$

Algorithm 1 takes the solution $\{P_t^o\}_{t=0}^T$ obtained from solving Problem 5 as an initial guess, and initiates $P_{0|-1}$ at Σ_0 as required by (4b). The notation $\|\cdot\|_F$ in Algorithm 1 represents the Frobenius norm. The algorithm produces a covariance trajectory $\{P_t\}_{t=0}^T$ that is close to the reference trajectory $\{P_t^o\}_{t=0}^T$ in Frobenius norm.

The reasoning behind the construction of Algorithm 1 is to keep the error covariance $P_t(\sigma)$ close to P_t^o , since P_t^o is the best covariance one could possibly obtain given the set of sensors. The

algorithm is reminiscent of a trajectory-tracking problem where P_t^o serves as the reference trajectory. In the following we formally provide technical justifications of using such a heuristic method and its merits using dynamic programming based arguments.

3.3. Dynamic programming and optimality guarantees

Let us denote the value function associated with Problem 1 to be U_t , which is given as follows

$$U_t(P) = \min_{P_{t|t-1}=P, \{\sigma(k)\}_{k=t}^T} \sum_{k=t}^T \text{tr}(P_k(\sigma)), \quad (7)$$

where P is a positive semidefinite matrix. Similarly, we denote the value function associated with the SDP relaxation of Problem 1 (equivalent to Problem 5 with $\ell = N$ and $R_t^i = C_t^{iT} \mathcal{V}_t^{i-1} C_t^i$) by U_t^o :

$$U_t^o(P) = \min_{P_{t|t-1}=P, \{\{\theta_k^i\}_{i \in \mathbb{N}}\}_{k=t}^T} \sum_{k=t}^T \text{tr}(P_k(\sigma)). \quad (8)$$

The difference between U_t and U_t^o is that the feasible choice of a sensor at time t for U_t has to be one of the $\{R_t^i\}_{i \in \mathbb{N}}$ (or equivalently $\{C_t^i, \mathcal{V}_t^i\}_{i \in \mathbb{N}}$), whereas the feasible choice of a sensor for U_t^o is any of the sensors that lie within the convex hull of $\{R_t^i\}_{i \in \mathbb{N}}$. Therefore, $U_t^o(P) \leq U_t(P)$ for all symmetric $P \succeq 0$. In what follows, we will suppress the $P_{t|t-1} = P$ constraint in the definitions of the value function to maintain notational brevity.

From dynamic programming, one may write

$$\begin{aligned} U_t(P) &= \min_{\sigma(t)} (\text{tr}(P_t(\sigma)) + U_{t+1}(P_{t+1|t}(\sigma))) \\ &= \min_{\sigma(t)} \left(\text{tr}(g_t(\sigma(t), P)) + U_{t+1}(h_{t+1}(g_t(\sigma(t), P))) \right), \\ U_T(P) &= \min_{\sigma} \text{tr}(g_T(\sigma, P)). \end{aligned}$$

In the following we will exploit some of the properties of U_t and the solutions obtained from solving the SDP relaxation (P_t^o, Q_t^o and $P_{t|t-1}^o$) to solve for an approximate value function associated with (7).

Before proceeding, let us present some useful properties of the map $g_t(\cdot, \cdot)$ defined in (3) which will assist us in our subsequent analyses. With a slight variation to Lemma 1-e from Sinopoli et al. (2004), one can prove that, for any fixed $i \in \mathbb{N}$, $g_t(i, M)$ is concave in M . Furthermore, we can characterize the derivative of the function $g_t(i, \cdot)$ by the following lemma.

Lemma 2 (Vitus et al., 2012). For each $i \in \mathbb{N}$ and for any positive semi-definite matrices M, L , it follows that

$$\frac{dg_t(i, M + \epsilon L)}{d\epsilon} \Big|_{\epsilon=0} = H_t(i, M) L H_t(i, M)^T, \quad (9)$$

where $H_t(i, M) = (I - M C_t^{iT} (C_t^i M C_t^{iT} + \mathcal{V}_t^i)^{-1} C_t^i)$.

Proof. Let us define $\tilde{g}_t(i, M) = (C_t^i M C_t^{iT} + \mathcal{V}_t^i)^{-1}$, and therefore,

$$\frac{d\tilde{g}_t(i, M + \epsilon L)}{d\epsilon} = -\tilde{g}_t(i, M + \epsilon L) C_t^i L C_t^{iT} \tilde{g}_t(i, M + \epsilon L).$$

Using (3a) and after some simplifications, we obtain

$$\frac{dg_t(i, M + \epsilon L)}{d\epsilon} \Big|_{\epsilon=0} = H_t(i, M) L H_t(i, M)^T. \quad \square$$

The following proposition shows that the value function U_t^o is locally Lipschitz, which is an important component in constructing Algorithm 1.

Proposition 3. For any two symmetric positive semidefinite matrices P and Q with bounded Frobenius norms, and for all $t = 0, \dots, T$, there exists a constant $K > 0$, such that

$$\|U_t^o(P) - U_t^o(Q)\| \leq K \|P - Q\|_F. \quad (10)$$

Proof. We prove this in an inductive way. Let us first consider $t = T$, and hence,

$$\begin{aligned} U_T^o(P) - U_T^o(Q) &= \min_{\{\theta^t\}_{i \in N}} \text{tr}(g_T(\theta, P)) - \min_{\{\theta^t\}_{i \in N}} \text{tr}(g_T(\theta, Q)) \\ &\stackrel{(a)}{\leq} \text{tr}(g_T(\theta^*, P) - g_T(\theta^*, Q)) \\ &\stackrel{(b)}{\leq} \text{tr}(H_T(\theta^*, Q)(P - Q)H_T^T(\theta^*, Q)) \\ &\leq \|P - Q\|_F \|H_T^T(\theta^*, Q)H_T(\theta^*, Q)\|_F \end{aligned}$$

where $\theta^* = [\theta^{1*}, \dots, \theta^{N*}]$ in (a) is a minimizer of $\text{tr}(g_T(\theta, Q))$, and (b) follows from the concavity property of the function $g_T(\theta, \cdot)$ along with Lemma 2. From the expression of $H_T(\theta^*, Q)$ in Lemma 2, along with the fact that Q has a bounded Frobenius norm, one can verify that there exists a finite $K > 0$ such that $\|H_T^T(\theta^*, Q)H_T(\theta^*, Q)\|_F \leq K$. Therefore,

$$U_T^o(P) - U_T^o(Q) \leq K \|P - Q\|_F.$$

The inductive hypothesis can be proven in a similar way which we omit due to page limitation. \square

The following proposition states that an upper bound on U_t is found from U_t^o .

Proposition 4. For any time t and $P \geq 0$ with bounded Frobenius norm, there exists a finite $\alpha > 0$ such that $U_t(P) \leq U_t^o(P) + \alpha$.

Based on these propositions, we are now ready to perform an approximate dynamic programming with the value function $U_t(P)$ to design a sub-optimal solution as follows. Recall that the value function $U_t(P)$ satisfies

$$U_t(P) = \min_{\sigma(t)} (\text{tr}(P_t(\sigma)) + U_{t+1}(P_{t+1|t}(\sigma))),$$

which can be re-written as

$$\begin{aligned} U_t(P) &\leq \alpha + \min_{\sigma(t)} (\text{tr}(P_t(\sigma)) + U_{t+1}^o(P_{t+1|t}(\sigma))), \\ &\leq \alpha + U_t^o(P_{t|t-1}^o) \\ &\quad + \min_{\sigma(t)} (\text{tr}(P_t(\sigma)) + U_{t+1}^o(P_{t+1|t}(\sigma)) - U_t^o(P_{t|t-1}^o)), \end{aligned}$$

where $P_{t|t-1}^o$ is obtained from the SDP relaxation. More specifically, by solving the SDP relaxation one obtains $\{P_t^o, Q_{t|t-1}^o, Q_t^o\}_{t=0}^T$ and from these one can construct $P_{t|t-1}^o = Q_{t|t-1}^{o-1}$. Thus, we have that $U_t^o(P_{t|t-1}^o) = \text{tr}(P_t^o) + U_{t+1}^o(P_{t+1|t}^o)$, and therefore,

$$\begin{aligned} U_t(P) &\leq \alpha + U_t^o(P_{t|t-1}^o) \\ &\quad + \min_{\sigma} (\text{tr}(P_t(\sigma) - P_t^o) + U_{t+1}^o(P_{t+1|t}(\sigma)) - U_{t+1}^o(P_{t+1|t}^o)) \\ &\leq \alpha + U_t^o(P_{t|t-1}^o) \\ &\quad + \min_{\sigma} (\text{tr}(P_t(\sigma) - P_t^o) + K \|P_{t+1|t}(\sigma) - P_{t+1|t}^o\|_F) \\ &\leq \alpha + U_t^o(P_{t|t-1}^o) \\ &\quad + \min_{\sigma} (\text{tr}(P_t(\sigma) - P_t^o) + K \|A_t\|^2 \|P_t(\sigma) - P_t^o\|_F) \\ &\leq \alpha + U_t^o(P_{t|t-1}^o) + K_1 \min_{\sigma} \|P_t(\sigma) - P_t^o\|_F, \end{aligned} \quad (11)$$

where $K_1 = K \|A_t\|^2 + \sqrt{n}$ and we have used Proposition 3 and the fact that for any $X \in \mathbb{R}^{n \times n}$, $\text{tr}(X) \leq \sqrt{n} \|X\|_F$.

Thus, performing the optimization $\min_{\sigma} \|P_t(\sigma) - P_t^o\|_F$ essentially minimizes an upper bound of the value function U_t ,

or equivalently, an upper bound of $\sum_{t=0}^T \text{tr}(P_t)$. Therefore, in essence, Algorithm 1 performs an approximate dynamic programming type optimization by minimizing an upper bound of the value function U_t . The following lemma provides a sub-optimality bound of Algorithm 1.

Lemma 5. Let σ, σ^* and θ^* denote the schedule obtained from Algorithm 1, the true optimal schedule of Problem 1, and the solution to Problem 5, respectively. Then,

$$\sum_{t=0}^T \text{tr}(P_t(\sigma)) \leq \sum_{t=0}^T \text{tr}(P_t(\sigma^*)) + \epsilon \quad (12)$$

where $\epsilon \triangleq \sqrt{n} \sum_{t=0}^T \frac{1-\lambda^{T+1-t}}{1-\lambda} \beta_t + \sqrt{n} \sum_{t=0}^T \|P_t(\theta^*) - P_t(\sigma^*)\|_F$, $\beta_t \triangleq \|g_t(\sigma^*(t), P_{t|t-1}(\theta^*)) - P_t(\theta^*)\|_F$ and $\lambda \triangleq \max_t \|A_{t-1}H(\sigma^*(t), P_{t|t-1}(\theta^*))\|^2$.

Proof. The proof of this lemma is omitted due to page limitation and can be found in Maity et al. (2021). \square

From the definition of β_t , we notice that ϵ captures the covariance mismatch between the schedules θ^* and σ^* , and consequently, portraying the effects of the relaxed sensor design problem on the overall optimality of the approach. If the optimal covariance $\{P_t(\sigma^*)\}_{t=0}^T$ is ‘‘close’’ to the covariance from the relaxed optimization $\{P_t(\theta^*)\}_{t=0}^T$ then the suboptimality bound decreases, which is expected. Furthermore, the bound ϵ depends on the system dimension n and degrades with the system’s dimension.

4. Numerical evaluations

We empirically evaluate the performance of our algorithm by applying it on a wide range of (randomly generated) scenarios and comparing it to (suitable modifications of) Tzoumas et al. (2016) and Gupta et al. (2006) and a random search algorithm. The random search algorithm randomly generates 2000 schedules and reports the best among these schedules. It is noteworthy that the algorithm proposed by Gupta et al. (2006) is for an infinite horizon steady-state estimation problem and here we adopt this algorithm to our finite horizon problems.

We conduct several experiments by varying the dimensions of the system. Four sets of experiments were conducted for four different dimensions of $A \in \mathbb{R}^{n \times n}$ for $n = 4, 6, 8$, and 10. For each dimension n , we generate thirty scenarios by randomly generating thirty A matrices with eigenvalues in the range $[1, 1.5]$. For each scenario, we consider four randomly chosen sensors with different numbers of dimensions (i.e., different number of rows for matrices C_i). Noise W_t is chosen to have zero mean and unit variance, i.e., $W_t \sim \mathcal{N}(0, I)$. The measurement noises are $V_t^i \sim \mathcal{N}(0, V^i)$ where V^i is a diagonal covariance matrix whose diagonal elements are chosen randomly between 0 and 1. We consider a time horizon of $T = 100$.

In Fig. 1 (left), we illustrate the performance of our algorithm compared to the others. The x -axis represents the dimension of the system and the y -axis represents the cost averaged over the randomly generated scenarios. As can be seen, our algorithm performs better than the other algorithms in terms of the cost $\sum_{t=0}^T \text{tr}(P_t)$. We also compare the run-time of these algorithms in Fig. 1 (right). While the run-times for random search and Gupta et al. (2006) are slightly smaller than ours, their performances are significantly worse than ours. On the other hand, while the performance of Tzoumas et al. (2016) is comparable to ours, their run-time is significantly higher compared to ours. In this way the proposed method provides a balanced trade-off between the computation-time and the objective value.

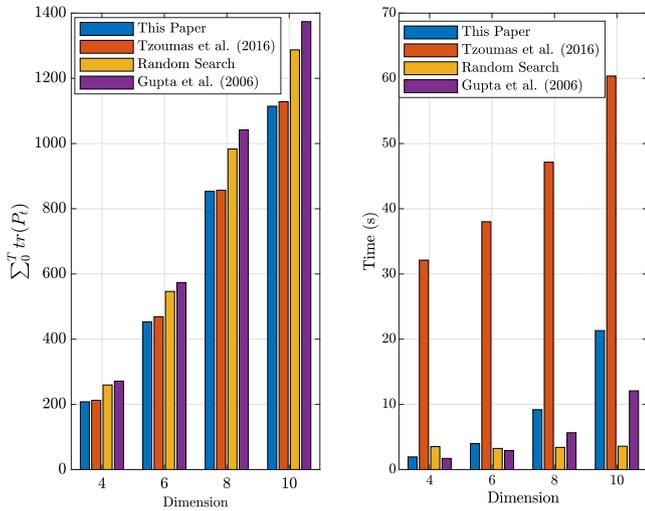


Fig. 1. Plot of the cost $\sum_{t=0}^T \text{tr}(P_t(\sigma))$ and computation times for our algorithm, Tzoumas et al. (2016), Gupta et al. (2006), and a random schedule algorithm.

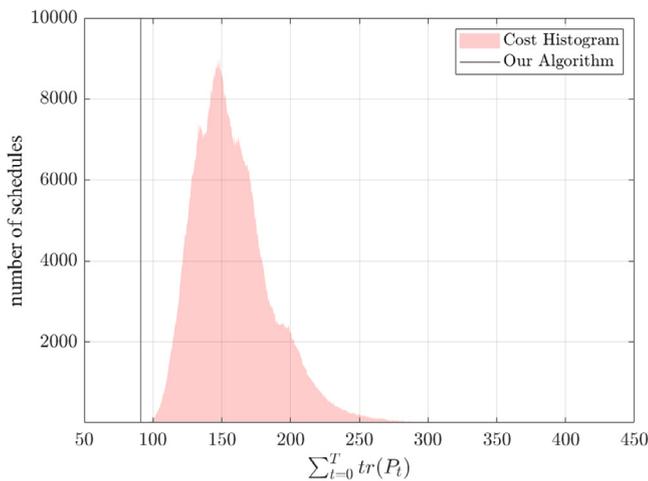


Fig. 2. Plot of total costs for all permutation of schedules. The gray line corresponds to the cost by using the schedule resulting from our tracking algorithm.

It is worth mentioning that during our study of sensor scheduling problems, we have noticed that the optimal schedule converge to a periodic scheduling scheme after a disproportionately shorter transient period. Similar observation has also been witnessed in other works, e.g., [Jawaid and Smith \(2015\)](#) and the optimality of such periodic behaviors is proven in [Orihuela et al. \(2014\)](#). The periodicity of the optimal policy can be further exploited in our framework to generate optimal scheduling for infinite horizon problems and to further reduce the computational complexity for finite horizon problems.

4.1. Near-optimal performance

While in the last set of experiments we illustrated that our algorithm outperforms the other algorithms, in this section we quantitatively demonstrate how close to the true optimum our solution can be. To compute the optimal schedule, we need to resort to exhaustive searches, and hence, we reduce the time horizon to $T = 9$ in order to maintain tractability of the exhaustive search methods. We randomly pick one of the thirty 10-dimensional scenarios that were used in the previous experiment. We have four available sensors to be scheduled for the

interval $[0, 9]$, and hence, the total number of possible schedules are 4^{10} . We compare the performances of all permutations of schedules (4^{10} of them) to the performance resulting from our algorithm by plotting a histogram in [Fig. 2](#). In [Fig. 2](#), the x-axis represents the cost ($\sum_{t=0}^T \text{tr}(P_t)$) and the y-axis represents how many schedules (out of the 4^{10} possible ones) can achieve that cost.² Such a histogram represents how likely it is to find a random schedule that will produce a given cost. For this example, our algorithm finds the optimal schedule, whereas the other algorithms fail to find the optimal solution.

5. Conclusion

In this paper, we reformulated the sensor scheduling problem as a sensor design problem whose convex relaxation is solved by a semidefinite programming approach. While such a relaxation does not readily produce a solution to the scheduling problem, we presented a *covariance-tracking* algorithm to construct a sensor schedule from the solution of the sensor design problem. The foundation of our algorithm is justified by using an approximate dynamic programming based argument where we show that the tracking based algorithm indeed minimizes an upper bound of the optimal cost (value function). A sub-optimality bound of our proposed algorithm is also derived and discussed. Performance of our algorithm is demonstrated on several examples and compared with several existing methods to show the merit of our framework.

References

- Athans, M. (1972). On the determination of optimal costly measurement strategies for linear stochastic systems. *Automatica*, 8(4), 397–412.
- Chhetri, A. S., Morrell, D., & Papandreou-Suppappola, A. (2007). On the use of binary programming for sensor scheduling. *IEEE Transactions on Signal Processing*, 55(6), 2826–2839.
- Evans, R., Krishnamurthy, V., Nair, G., & Sciacca, L. (2005). Networked sensor management and data rate control for tracking maneuvering targets. *IEEE Transactions on Signal Processing*, 53(6), 1979–1991.
- Gally, T., Pfetsch, M. E., & Ulbrich, S. (2018). A framework for solving mixed-integer semidefinite programs. *Optimization Methods & Software*, 33(3), 594–632.
- Gupta, V., Chung, T. H., Hassibi, B., & Murray, R. M. (2006). On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2), 251–260.
- He, Y., & Chong, K. (2004). Sensor scheduling for target tracking in sensor networks. In *43rd IEEE conference on decision and control* (pp. 743–748). Atlantis, Bahamas.
- Jawaid, S. T., & Smith, S. L. (2015). Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. *Automatica*, 61, 282–288.
- Joshi, S., & Boyd, S. (2008). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2), 451–462.
- Liu, S., Fardad, M., Masazade, E., & Varshney, P. K. (2014). Optimal periodic sensor scheduling in networks of dynamical systems. *IEEE Transactions on Signal Processing*, 62(12), 3055–3068.
- Maity, D., & Baras, J. S. (2015). Dynamic, optimal sensor scheduling and value of information. In *2015 18th international conference on information fusion (fusion)* (pp. 239–244). IEEE.
- Maity, D., Hartman, D., & Baras, J. S. (2021). Sensor scheduling for linear systems: A covariance tracking approach. ArXiv Eprints.
- Masazade, E., Fardad, M., & Varshney, P. K. (2012). Sparsity-promoting extended kalman filtering for target tracking in wireless sensor networks. *IEEE Signal Processing Letters*, 19(12), 845–848.
- Meier, L., Peschon, J., & Dressler, R. (1967). Optimal control of measurement subsystems. *IEEE Transactions on Automatic Control*, 12(5), 528–536.
- Nowak, R., Mitra, U., & Willett, R. (2004). Estimating inhomogeneous fields using wireless sensor networks. *IEEE Transactions on Selected Areas in Communications*, 22(6), 999–1006.
- Orihuela, L., Barreiro, A., Gómez-Estern, F., & Rubio, F. R. (2014). Periodicity of Kalman-based scheduled filters. *Automatica*, 50(10), 2672–2676.
- Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M. I., & Sastry, S. S. (2004). Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9), 1453–1464.

² To be precise, we divided the x-axis into intervals of 0.5 (e.g., $[90, 90.5]$ etc.), and evaluated how many schedules produce a cost within each interval.

- Tzoumas, V., Jadbabaie, A., & Pappas, G. J. (2016). Near-optimal sensor scheduling for batch state estimation: Complexity, algorithms, and limits. In *55th IEEE conference on decision and control* (pp. 2695–2702). Las Vegas, USA.
- Vitus, M. (2011). Sensor placement for improved robotic navigation. *Robotics: Science and Systems VI*, 217.
- Vitus, M. P., Zhang, W., Abate, A., Hu, J., & Tomlin, C. J. (2012). On efficient sensor scheduling for linear dynamical systems. *Automatica*, 48(10), 2482–2493.
- Wang, H., Yao, K., Pottie, G., & Estrin, D. (2004). Entropy-based sensor selection heuristic for target localization. In *Proceedings of the 3rd international symposium on information processing in sensor networks* (pp. 36–45). ACM.
- Williams, J. L. (2007). *Information theoretic sensor management* (Ph.D. thesis), Massachusetts Institute of Technology.
- Zhang, L., & Hristu-Varsakelis, D. (2006). Communication and control co-design for networked control systems. *Automatica*, 42(6), 953–958.



Dipankar Maity (Senior Member, IEEE) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, India, in 2013, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA in 2018.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC, USA. He was a Postdoctoral Fellow with the Georgia Institute of Technology, Atlanta, GA, USA. During his Ph.D., he

was a Visiting Scholar at the Technical University of Munich (TUM) and at the KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include temporal logic-based controller synthesis, control under communication constraints, intermittent-feedback control, multi-agent systems, stochastic games, and integration of these ideas in the context of cyber-physical systems.



David Hartman received the B.S. degree in electrical and computer science from UC Berkeley, in 2014. He is currently a Ph.D. student in electrical and computer engineering at the University of Maryland, College Park, MD under the supervision of Professor John S. Baras. His research interests include sensor scheduling, risk-sensitive control, and reinforcement learning. He has interned at various companies including Nokia Bell Labs and GE Global Research.



John S. Baras received the Diploma in Electrical and Mechanical Engineering from the National Technical University of Athens, Athens, Greece, in 1970, and the M.S. and Ph.D. degrees in Applied Mathematics from Harvard University, Cambridge, MA, USA, in 1971 and 1973, respectively. Since 1973, he has been with the Department of Electrical and Computer Engineering, University of Maryland at College Park, MD, USA, where he is currently a Distinguished University Professor. He is also a faculty member of the Applied Mathematics, Statistics and Scientific Computation Program, and Affiliate Professor in the Fischell Department of Bioengineering, the Department of Mechanical Engineering, and the Department of Decision, Operations and Information Technologies, Robert H. Smith School of Business. Since 2013, he has been Guest Professor at the School of Electrical Engineering of the Royal Institute of Technology (KTH), Sweden. From 1985 to 1991, he was the Founding Director of the Institute for Systems Research (ISR) (one of the first six National Science Foundation Engineering Research Centers). In 1990, he was appointed to the endowed Lockheed Martin Chair in Systems Engineering. Since 1992, he has been the Director of the Maryland Center for Hybrid Networks (HYNET), which he co-founded.

He is an IEEE Life Fellow, SIAM Fellow, AAAS Fellow, NAI Fellow, IFAC Fellow, AMS Fellow, AIAA Associate Fellow, Member of the National Academy of Inventors (NAI) and a Foreign Member of the Royal Swedish Academy of Engineering Sciences (IVA). Major honors and awards include the 1980 George Axelby Award from the IEEE Control Systems Society, the 2006 Leonard Abraham Prize from the IEEE Communications Society, the 2014 Tage Erlander Guest Professorship from the Swedish Research Council, and a three year (2014–2017) Senior Hans Fischer Fellowship from the Institute for Advanced Study of the Technical University of Munich, Germany. In 2016 he was inducted in the University of Maryland A. J. Clark School of Engineering Innovation Hall of Fame. He was awarded the 2017 Institute for Electrical and Electronics Engineers (IEEE) Simon Ramo Medal, the 2017 American Automatic Control Council (AACC) Richard E. Bellman Control Heritage Award, and the 2018 American Institute for Aeronautics and Astronautics (AIAA) Aerospace Communications Award. In 2018 he was awarded a Doctorate Honoris Causa by the National Technical University of Athens, Greece.

He has educated 87 doctoral students, 116 MS students and has mentored 55 postdoctoral fellows. He has given many plenary and keynote addresses in major international conferences worldwide. He has been awarded eighteen patents and has been honored worldwide with many awards as innovator and leader of economic development.

He has educated 87 doctoral students, 116 MS students and has mentored 55 postdoctoral fellows. He has given many plenary and keynote addresses in major international conferences worldwide. He has been awarded eighteen patents and has been honored worldwide with many awards as innovator and leader of economic development.