On Efficient Sensor Scheduling for Linear Dynamical Systems

Michael P. Vitus[† a], Wei Zhang[† b], Alessandro Abate[c], Jianghai Hu[d], Claire J. Tomlin[e]

[a] *Department of Aeronautics and Astronautics, Stanford University, CA 94305, USA*

[b] *Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210, USA*

[c] *DCSC, Delft University of Technology, Delft, Netherlands*

[d] *Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA*

[e] *Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94708, USA*

Abstract

Consider a set of sensors estimating the state of a process in which only one of these sensors can operate at each time-step due to constraints on the overall system. The problem addressed here is to choose which sensor should operate at each time-step to minimize a weighted function of the error covariances of the state estimations. This work investigates the development of tractable algorithms to solve for the optimal and suboptimal sensor schedules. A condition on the non-optimality of an initialization of the schedule is developed. Using this condition, both an optimal and a suboptimal algorithm are devised to prune the search tree of all possible sensor schedules. The suboptimal algorithm trades off the quality of the solution and the complexity of the problem through a tuning parameter. The performance of the suboptimal algorithm is also investigated and an analytical error bound is provided. Numerical simulations are conducted to demonstrate the performance of the proposed algorithms, and the application of the algorithms in active robotic mapping is explored.

Key words: Sensor scheduling, Kalman Filter, Sensor Selection, Multi-Sensor Estimation, Networked Control Systems

1 Introduction

Recent work in estimation theory has dealt with various topics ranging from sensor fusion from multiple sources, coverage control in wireless sensor networks, estimation with intermittent or delayed observations, data association for tracking multiple targets, scheduling of sensors measurements, and many more. This work focuses on the problem of *sensor scheduling* which consists of selecting one (or multiple) sensors out of a number of available sensors at each time-step to minimize a function of the estimation error over a finite time-horizon. Additionally, this work can be extended to the problem of optimal positioning of sensors or trajectory planning for mobile sensors. Other possible applications of this work include management of sensor networks, energy efficient control of buildings, and state estimation with sensor constraints.

With the advances of sensor networks and the improvement of unmanned systems for reconnaissance and surveillance missions, the environment is being inundated with sensor networks monitoring external processes [2,19,10,17]. In these networks, a sensor scheduling policy might be desired due to constraints on the communication bandwidth or power requirements that limit the number of nodes that can operate at each time-step. Another application of sensor scheduling is in energy efficient control of buildings through participatory sensing. By leveraging the occupants' smartphones to localize the users in the building and to infer their destinations [33] an energy saving policy can be enabled by adjusting the lights, the computer power settings, or the temperature set points. The building occupants could be localized by using triangulation of the smartphone's Wi-Fi signal [29]; however, this would quickly drain the battery. Sensor scheduling has the potential to

Email addresses: vitus@stanford.edu (Michael P. Vitus[†]), zhang@ece.osu.edu (Wei Zhang[†]), a.abate@tudelft.nl (Alessandro Abate), jianghai@purdue.edu (Jianghai Hu), tomlin@eecs.berkeley.edu (Claire J. Tomlin).

[1] [†]These authors contributed equally to this work.

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

decrease the power consumption by determining when to use the Wi-Fi triangulation to accurately locate the user, or otherwise integrate the smartphone's inertial measurements to provide a rough estimate. In addition to conserving power, sensors may interfere with one another, as with sonar range-finding sensors, and thus may not operate at the same time. Another application is in terrain relative navigation [20] for underwater vehicles which correlates depth readings from sonar measurements with a pre-existing map to localize the vehicle. Typically there are several sonar devices onboard the vehicle with different measurement patterns as well as different noise characteristics. Given the interference between the sonar sensors, only one device may be operated at once and therefore the objective would be to manage the schedule of sonar devices to better localize the vehicle.

The sensor scheduling problem has been studied extensively in the past. In a seminal work, Meier et al. [21] proposed a solution to the discrete-time scheduling problem through the use of dynamic programming which enumerates all possible sensor schedules. The combinatorial complexity makes this method intractable for long schedule horizons. A local gradient method was also proposed which is more likely to be computationally feasible, but only provides a suboptimal solution. Given the inherent computational complexity in solving the sensor scheduling problem, the research community has concentrated on developing computationally efficient algorithms.

In [1], a relaxed dynamic programming procedure is applied to obtain a suboptimal strategy that is bounded by a pre-specified distance to optimality, but is only applicable for an objective function that minimizes the final step estimation error. A convex optimization procedure was developed in [15] as a heuristic to solve the problem of selecting k sensors from a set of m. Although no optimality guarantees can be provided for the solution, numerical experiments suggest that it performs well. In [11] and [23], heuristics such as a sliding window or a greedy selection policy were employed in order to prune the search tree. Another approach [12] proposed to switch sensors randomly according to a probability distribution to obtain the best upper bound on the expected steady-state performance. Savkin et al. [25] considered the problem of robust sensor scheduling in which the noise and uncertainty models were unknown deterministic functions. Their solution was given in terms of a solution to a Riccati differential equation. Rezaeian [24] formulated the sensor scheduling problem as a partially observable Markov decision process (POMDP) and minimized the estimation entropy to obtain the optimal observability. A sensor scheduling algorithm trading off the performance and sensor usage costs was devised in [13], and was also formulated as a POMDP and solved via an approximation process.

Sensor selection for target tracking has also been extensively studied. Isler et al. [14] proposed a sensor selection approximation algorithm to minimize the estimation error of a target that is guaranteed to be within a factor of 2 of the smallest error. An entropy-based heuristic algorithm proposed in [30] greedily selects the next sensor that provides the greatest reduction in entropy at the next time-step. Ertin et al. [8] proposed a greedy algorithm by choosing the sensor which maximizes the mutual information at the next time-step. The target tracking problem has also been formulated as a POMDP [13] and solved through a Monte Carlo method using a sampling-based Q-value approximation for computing the cost of a sensor schedule. Another Monte Carlo method is proposed in [6] which chooses the sensor to minimize the predicted mean-square error of the target state estimate. Two greedy based sensor scheduling algorithms were developed in [7]; they formulated the problem as a mixed integer optimization program and solved it through a branch and bound technique.

As compared to previous works, the main distinction of this paper is the development of several efficient scheduling algorithms that drastically reduce the computational complexity while also providing an analytical bound for the solution quality. This is accomplished by leveraging the recent results of optimal control for switched systems which can be thought of as the dual of the sensor scheduling problem. A switched system consists of a family of subsystems, each with specific dynamics, and allows for controlled switching between the different subsystems. The analysis and design of controllers for hybrid systems has received a large amount of attention from the research community [3,28,4,18,5,31,32]. Zhang et al. [31,32] proposed a method based on dynamic programming to solve for the optimal discrete mode sequence and continuous input for the discrete-time linear quadratic regulation problem for switched linear systems. They proposed several efficient and computationally tractable algorithms for obtaining the optimal and bounded suboptimal solution through effective pruning of the search tree, which grows exponentially with the horizon length.

This work presents three main contributions, arising out of the insights from the control of switched systems in [31,32], to reduce the computational complexity of the sensor scheduling problem. First, the properties of the estimation process are analyzed to develop a condition on the non optimality of the initialization of a sensor schedule. Second, based on the previous condition, two efficient pruning techniques are developed which provide optimal and suboptimal solutions. These algorithms can significantly reduce the computational complexity and thus enable the solution of larger systems with longer scheduling horizons. The suboptimal algorithm includes a tuning parameter which trades off the quality of the solution with the complexity of the problem, for small and large values respectively. Third, an analytical bound

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

on the quality of the solution from the suboptimal algorithm that provides insight into the performance of the algorithm is presented. Specifically, as the tuning parameter decreases the suboptimal solution approaches the optimal solution asymptotically, and as the tuning parameter increases the error only grows linearly. The properties of the optimal and suboptimal algorithms are demonstrated through several numerical examples. Although the paper focuses on choosing only one sensor, the proposed algorithms also apply to the case of selecting multiple sensors at each time-step at the cost of increased complexity.

The paper is structured as follows. Section 2 describes the standard sensor scheduling problem formulation. Then, several properties of the objective function are explored in Section 3. In Section 4, a pruning algorithm is proposed which provides the optimal solution. In Section 5, the optimal algorithm is generalized to provide a suboptimal solution while reducing the computational complexity, and the error of this suboptimal solution is bounded. Numerical examples on the performance of the suboptimal solutions are presented in Section 6 and an application in active robotic mapping is presented in Section 7. The paper concludes with directions of future work.

2 Problem Formulation

Consider the following linear stochastic system defined by

$$x\left(k+1\right)=Ax\left(k\right)+w\left(k\right),\,\forall k\in T_N, \tag{1}$$

where $x\left(k\right)\in\mathbb{R}^n$ is the state of the system, $w\left(k\right)\in\mathbb{R}^n$ is the process noise and $T_N=\{0,\ldots,N-1\}$ is the time horizon. The initial state, $x(0)$, is assumed to be of a zero mean Gaussian distribution with covariance Σ_0 i.e., $x(0)\sim\mathcal{N}\left(0,\Sigma_0\right)$. At each time-step, only one sensor is allowed to operate from a set of M sensors. The measurement of the i^{th} sensor is,

$$y_i\left(k\right)=C_ix\left(k\right)+v_i\left(k\right),\,\forall k\in T_N, \tag{2}$$

where $y_i(k)\in\mathbb{R}^p$ and $v_i(k)\in\mathbb{R}^p$ are the measurement output and noise of the i^{th} sensor at time k, respectively. The process and measurement noise have zero mean Gaussian distributions, $w\left(k\right)\sim\mathcal{N}\left(0,\Sigma_w\right)$, $v_i\left(k\right)\sim\mathcal{N}\left(0,\Sigma_{v_i}\right)$, $\forall i\in\mathbb{M}$, where $\mathbb{M}\triangleq\{1,\ldots,M\}$ is the set of M sensors. The process noise, measurement noise and initial state are assumed to be mutually independent. Let λ_w^- be the smallest eigenvalue of Σ_w and assume that $\lambda_w^->0$. Denote by $\mathbb{M}^t$ the set of all ordered sequences of sensor schedules of length t where $t\leq N$. An element $\sigma=\{\sigma_0,\sigma_1,\ldots,\sigma_{t-1}\}\in\mathbb{M}^t$ is called a (t-horizon) *sensor schedule*. Under a given sensor schedule σ, the measurement sequence is

$$y(k)=y_{\sigma_k}(k)=C_{\sigma_k}x(k)+v_{\sigma_k}(k),\forall k\in\{0,\ldots,t-1\}.$$

For each $k\leq t$ with $t\leq N$ and each $\sigma\in\mathbb{M}^t$, let $\hat{\Sigma}_k^\sigma$ be the predictor covariance matrix of the optimal estimate of $x(k)$ given the measurements $\{y(0),\ldots,y(k-1)\}$. By a standard result of linear estimation theory, the Kalman filter is the minimum mean square error estimator, and the predictor covariance of the system state estimate evolves according to the Riccati recursion [16]

$$\begin{aligned}\hat{\Sigma}_{k+1}^\sigma=&A\hat{\Sigma}_k^\sigma A^{\mathrm{T}}+\Sigma_w-\\&A\hat{\Sigma}_k^\sigma C_{\sigma_k}^{\mathrm{T}}\left(C_{\sigma_k}\hat{\Sigma}_k^\sigma C_{\sigma_k}^{\mathrm{T}}+\Sigma_{v_{\sigma_k}}\right)^{-1}C_{\sigma_k}\hat{\Sigma}_k^\sigma A^{\mathrm{T}}\end{aligned} \tag{3}$$

with initial condition $\hat{\Sigma}_0^\sigma=\Sigma_0$ and $k\leq t$. Let $\mathbb{R}_+$ and $\mathbb{Z}_+$ be the set of nonnegative real numbers and integers, respectively. Define $J\left(\sigma\right):\mathbb{M}^t\rightarrow\mathbb{R}_+$ as the accrued estimation error under schedule σ, i.e.,

$$J_t(\sigma)=\sum_{k=1}^{t}\mathrm{tr}\left(\hat{\Sigma}_k^\sigma\right). \tag{4}$$

The sensor scheduling problem is formulated as the following discrete optimization problem:

$$\underset{\sigma\in\mathbb{M}^N}{\text{minimize}}\;J_N\left(\sigma\right), \tag{5}$$

and its optimal value is denoted by V_N.

3 Characterization of the Objective Function

The main challenge in solving Problem (5) lies in the exponential growth of the discrete set $\mathbb{M}^N$ with respect to the horizon length N. This exponential growth requires careful development of computationally-tractable solutions, which are derived from the properties developed in this section.

Let $\mathcal{A}$ denote the *positive semidefinite cone*, which is the set of all symmetric positive semidefinite matrices. A *Riccati Mapping* $\rho_i:\mathcal{A}\rightarrow\mathcal{A}$ is defined that maps the current covariance matrix, $\hat{\Sigma}_k$, under a new measurement from sensor $i\in\mathbb{M}$ to the next covariance matrix,

$$\begin{aligned}\rho_i(\hat{\Sigma}_k)=&A\hat{\Sigma}_kA^{\mathrm{T}}-\\&A\hat{\Sigma}_kC_i^{\mathrm{T}}\left(C_i\hat{\Sigma}_kC_i^{\mathrm{T}}+\Sigma_{v_i}\right)^{-1}C_i\hat{\Sigma}_kA^{\mathrm{T}}+\Sigma_w.\end{aligned} \tag{6}$$

A k-horizon Riccati mapping, $\phi_k^\sigma:\mathcal{A}\rightarrow\mathcal{A}$ is similarly defined that maps the covariance matrix at time 0, Σ_0, to the covariance matrix at time-step k, using the first k elements of the sensor schedule σ:

$$\phi_k^\sigma\left(\Sigma_0\right)=\rho_{\sigma_{k-1}}\left(\ldots\rho_{\sigma_1}\left(\rho_{\sigma_0}\left(\Sigma_0\right)\right)\right). \tag{7}$$

An example of the search tree, for two sensors, corresponding to the problem defined in Eqn. (5) is shown

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

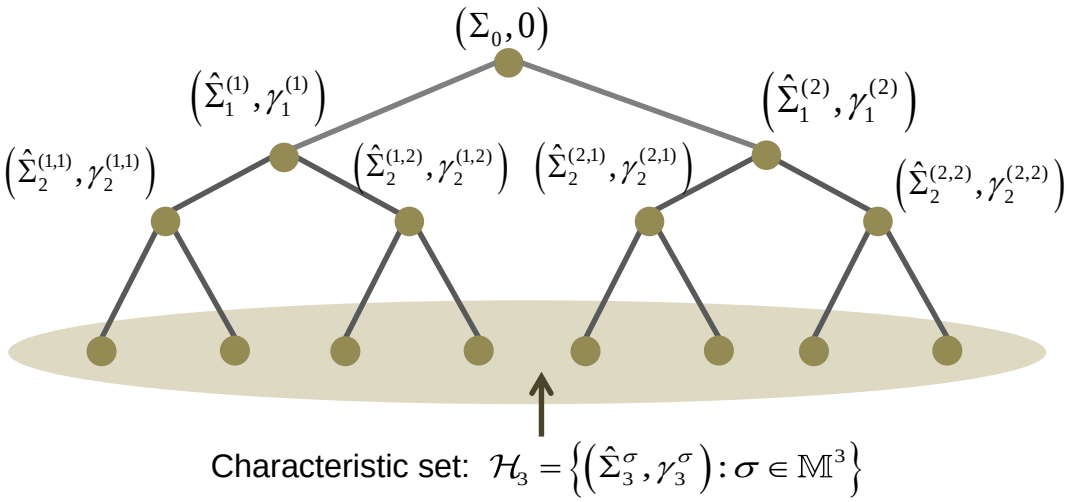


Fig. 1. The search tree and characteristic set for the sensor scheduling problem for an example with two sensors. This tree is the enumeration of all possible sensor schedules with the corresponding covariance and running cost at each time-step. The superscript for each covariance matrix, e.g. $\hat{\Sigma}^{(1,2)}$, denotes the sensor schedule used to obtain that estimate of the state.

in Figure 1. Each node on the k^{th} level of the tree corresponds to a k-horizon sensor schedule $\sigma \in \mathbb{M}^k$ and is represented by the so-called *characteristic pair* $(\Sigma_k^\sigma, \gamma_k^\sigma)$ that consists of the covariance matrix Σ_k^σ and the accrued cost $\gamma_k^\sigma = J_k(\sigma)$ for the schedule σ. These pairs can be computed iteratively using the Riccati mapping. For example, the pair $(\Sigma_1^{(1)}, \gamma_1^{(1)})$ in Figure 1 can be obtained as $\Sigma_1^{(1)} = \rho_1(\Sigma_0)$, and $\gamma_1^{(1)} = \mathrm{tr}\left(\Sigma_1^{(1)}\right)$, which can in turn be used to compute the pair to its right as $\Sigma_2^{(1,2)} = \rho_2\left(\Sigma_1^{(1)}\right)$ and $\gamma_2^{(1,2)} = \gamma_1^{(1)} + \mathrm{tr}\left(\Sigma_2^{(1,2)}\right)$. Clearly, if two nodes have the same characteristic pair, then they will have the same sets of descendants. The set of all the characteristic pairs at time-step k is called the *(k-horizon) characteristic set.*

Definition 1 (Characteristic Set) *The sequence of sets* $\{\mathcal{H}_k\}_{k=0}^N$ *generated recursively by* $\mathcal{H}_{k+1} = h_{\mathbb{M}}(\mathcal{H}_k)$ *with initial condition* $\mathcal{H}_0 = \{(\Sigma_0, 0)\}$ *is called the* characteristic set *associated with Problem* (5)*. Here the mapping* $h_{\mathbb{M}}(\cdot)$ *is called the* characteristic set mapping*, which is defined by:*

$$h_{\mathbb{M}}(\mathcal{H}) = \{(\rho_i(\Sigma), \gamma + \mathrm{tr}(\rho_i(\Sigma))) : \forall i \in \mathbb{M},\ \forall (\Sigma, \gamma) \in \mathcal{H}\}.$$

The characteristic sets grow exponentially in size from the singleton $\{(\Sigma_0, 0)\}$ to the set $\mathcal{H}_N$ consisting of up to M^N pairs, each comprising a positive semidefinite matrix and an accrued cost. Let $\mathcal{H}_k(i) = (\Sigma_k(i), \gamma_k(i))$ be the i^{th} element of the set $\mathcal{H}_k$. For any subset $\hat{\mathcal{H}}_k \subset \mathcal{H}_k$, the set of schedules corresponding to $\hat{\mathcal{H}}_k$ is defined by,

$$\mathbb{M}(\hat{\mathcal{H}}_k) = \{\sigma \in \mathbb{M}^k : (\hat{\Sigma}_k^\sigma, \gamma_k^\sigma) \in \hat{\mathcal{H}}_k\}. \qquad (8)$$

The main idea of the proposed solution method is motivated by the following properties of the Riccati mapping.

Theorem 1 *For any* $i \in \mathbb{M}$ *and any* $\Sigma_1, \Sigma_2 \in \mathcal{A}$*,*
*(i) [**Monotonicity**] If* $\Sigma_1 \preceq \Sigma_2$*, then* $\rho_i(\Sigma_1) \preceq \rho_i(\Sigma_2)$*;*
*(ii) [**Concavity**]* $\rho_i(c\Sigma_1 + (1-c)\Sigma_2) \succeq c\rho_i(\Sigma_1) + (1-c)\rho_i(\Sigma_2)$*,* $\forall c \in [0,1]$*.*

Remark 1 *The monotonicity property is a well-known result and its proof is provided in [16]. The concavity property is an immediate consequence of Lemma 1-(e) in [26].*

Thus, systems starting with a larger initial covariance, in the positive semidefinite sense, will yield larger covariances at all future time-steps. This result is important because it provides insight on how to reduce the complexity of the scheduling problem.

Theorem 1 can be repeatedly applied to obtain the following corollary.

Corollary 1 *Let* $\sigma \in \mathbb{M}^N$ *and* $\Sigma_1, \Sigma_2 \in \mathcal{A}$*, then* $\forall k \in [0, N]$*,* $\forall c \in [0,1]$*,*
(i) If $\Sigma_1 \preceq \Sigma_2$*, then* $\phi_k^\sigma(\Sigma_1) \preceq \phi_k^\sigma(\Sigma_2)$*;*
(ii) $\phi_k^\sigma(c\Sigma_1 + (1-c)\Sigma_2) \succeq c\phi_k^\sigma(\Sigma_1) + (1-c)\phi_k^\sigma(\Sigma_2)$*.*

4 Computation of the Optimal Sensor Schedule

To enable the study on larger systems with longer scheduling horizons, it is necessary to prune branches from the exponential search tree that are redundant and thus will not lead to the optimal solution. The proposed algorithm uses the properties of the Riccati mapping to obtain an easy-to-check condition to prune branches of the search tree without discarding the optimal schedule.

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

For example from Corollary 1, if two nodes $(\hat{\Sigma}_2^{(1,2)}, \gamma_2^{(1,2)})$ and $(\hat{\Sigma}_2^{(2,1)}, \gamma_2^{(2,1)})$ in Figure 1 satisfy the condition

$$\hat{\Sigma}_2^{(2,1)} \preceq \hat{\Sigma}_2^{(1,2)} \text{ and } \gamma_2^{(2,1)} \leq \gamma_2^{(1,2)},$$

then by the monotonicity of the Riccati mapping, all the descendants of the node $(\hat{\Sigma}_2^{(1,2)}, \gamma_2^{(1,2)})$ in the search tree will have a larger cost than the corresponding descendants of the node $(\hat{\Sigma}_2^{(2,1)}, \gamma_2^{(2,1)})$. Hence, the exploration of the branches under $(\hat{\Sigma}_2^{(1,2)}, \gamma_2^{(1,2)})$ can be avoided, or equivalently the pair $(\hat{\Sigma}_2^{(1,2)}, \gamma_2^{(1,2)})$ can be pruned from the characteristic set $\mathcal{H}_2$. Such a pair will be called *redundant*. By further considering the concavity of the Riccati mapping, other redundant pairs can be identified and pruned from the search tree. The following definition provides a condition to characterize redundant pairs.

Definition 2 (Algebraic Redundancy) *A pair $(\Sigma, \gamma) \in \mathcal{H}$, where $\mathcal{H}$ is a characteristic set (Def. 1), is called algebraically redundant with respect to $\mathcal{H} \backslash \{(\Sigma, \gamma)\}$, if there exist nonnegative constants $\{\alpha_i\}_{i=1}^{l-1}$ such that*

$$\sum_{i=1}^{l-1} \alpha_i = 1, \text{ and } \begin{bmatrix} \Sigma & 0 \\ 0 & \gamma \end{bmatrix} \succeq \sum_{i=1}^{l-1} \alpha_i \begin{bmatrix} \Sigma(i) & 0 \\ 0 & \gamma(i) \end{bmatrix}$$

where $l = |\mathcal{H}|$ and $\{(\Sigma(i), \gamma(i))\}_{i=1}^{l-1}$ is an enumeration of $\mathcal{H} \setminus \{(\Sigma, \gamma)\}$.

Using Corollary 1, one can show that the branches corresponding to the redundant pairs can be pruned without discarding the optimal solution of the sensor scheduling problem.

Theorem 2 *If the pair $(\Sigma, \gamma) \in \mathcal{H}_t$ is algebraically redundant, then the pair and all of its descendants can be pruned without eliminating the optimal solution from the search tree.*

PROOF. Let (Σ, γ) be an algebraic redundant pair satisfying the condition in Definition 2 with some constants $\{\alpha_i\}_{i=1}^{l-1}$. It suffices to show that there exists a pair $\left(\tilde{\Sigma}, \tilde{\gamma}\right) \in \mathcal{H}_t \setminus (\Sigma, \gamma)$ such that $\forall \sigma^{N-t} \in \mathbb{M}^{N-t}$,

$$\gamma + \sum_{k=1}^{N-t} \mathrm{tr}(\phi_k^{\sigma^{N-t}}(\Sigma)) \geq \tilde{\gamma} + \sum_{k=1}^{N-t} \mathrm{tr}(\phi_k^{\sigma^{N-t}}(\tilde{\Sigma})).$$

From the monotonicity and concavity of $\phi_k^{\sigma^r}$,

$$\gamma + \sum_{k=s}^{N} \mathrm{tr}(\phi_{k-t}^{\sigma^r}(\Sigma)) \geq \sum_{i=1}^{l-1} \alpha_i \left[\gamma(i) + \sum_{k=s}^{N} \mathrm{tr}(\phi_{k-t}^{\sigma^r}(\Sigma(i))) \right]$$

where $r = N - t$, $s = t + 1$ and $l = |\mathcal{H}_t|$. Finally, the convex combination of the scalar variables indexed by i is lower bounded by the smallest one, i.e.,

$$\gamma + \sum_{k=s}^{N} \mathrm{tr}(\phi_{k-t}^{\sigma^r}(\Sigma)) \geq \gamma(i^*) + \sum_{k=s}^{N} \mathrm{tr}(\phi_{k-t}^{\sigma^r}(\Sigma(i^*)))$$

where $i^* = \arg\min_{i \in [0, l-1]} \gamma(i) + \sum_{k=s}^{N} \mathrm{tr}(\phi_{k-t}^{\sigma^r}(\Sigma(i)))$. Therefore the branch defined by (Σ, γ) and its descendants can be eliminated because it will not contain the optimal solution. □

Definition 3 (Equivalent Subset) *Let the set $\bar{\mathcal{H}} = \left\{\left(\bar{\Sigma}(i), \bar{\gamma}(i)\right)\right\}_{i=1}^{|\bar{\mathcal{H}}|}$ be called an equivalent subset of $\mathcal{H} = \{(\Sigma(i), \gamma(i))\}_{i=1}^{|\mathcal{H}|}$, if the set $\bar{\mathcal{H}} \subset \mathcal{H}$ contains a schedule that leads to the global optimal solution, i.e.*

$$\min_{i \leq |\bar{\mathcal{H}}|} \bar{\gamma}(i) = \min_{i \leq |\mathcal{H}|} \gamma(i).$$

According to Theorem 2 and Definition 3, an equivalent subset of a characteristic set $\mathcal{H}$ can be obtained by removing all the redundant pairs in $\mathcal{H}$ as illustrated in Algorithm 1. The first step is to sort the set in ascending order based upon the current cost of the branches, which is a reasonable heuristic for obtaining the minimum size of the equivalent subset. The equivalent subset is then initialized to the current best branch. Next, each entry in $\mathcal{H}$ is tested with the current equivalent subset, $\mathcal{H}^{(i-1)}$, to determine if it can be eliminated. If not, then it is appended to the current subset. The equivalent subset returned by this algorithm is denoted by $ES(\mathcal{H})$.

Algorithm 1 $[ES\,(\mathcal{H})]$

1: Sort $\mathcal{H}$ in ascending order such that $\gamma(i) \leq \gamma(i+1)$, $\forall i \in \{1, \ldots, |\mathcal{H}| - 1\}$
2: $\mathcal{H}^{(1)} = \{(\mathcal{H}(1))\}$
3: **for** $i = 2,\ldots,|\mathcal{H}|$ **do**
4: **if** $\mathcal{H}(i)$ satisfies Definition 2 with respect to $\mathcal{H}^{(i-1)}$ **then**
5: $\mathcal{H}^{(i)} = \mathcal{H}^{(i-1)}$
6: **else**
7: $\mathcal{H}^{(i)} = \mathcal{H}^{(i-1)} \cup \mathcal{H}(i)$
8: **end if**
9: **end for**

An efficient method for computing the optimal sensor schedule which uses the proposed pruning technique is stated in Algorithm 2. The procedure first initializes the characteristic set to the pair composed of the *a priori* covariance of the initial state and initial cost. Then, for each time-step it computes the characteristic set mapping and calculates the equivalent subset with Algorithm 1. Once the tree is fully built, the optimal sensor schedule is determined.

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

Algorithm 2 Sensor Scheduling for a Finite Horizon

1: $\mathcal{H}_0 = \{(\Sigma_0, 0)\}$
2: **for** $k = 1,\ldots,N$ **do**
3: $\quad \mathcal{H}_k = h_{\mathbb{M}}(\mathcal{H}_{k-1})$
4: $\quad$ Perform $ES(\mathcal{H}_k)$
5: **end for**
6: $\sigma^* = \arg\min_{\sigma \in \mathbb{M}(\mathcal{H}_N)} \gamma_N^\sigma$

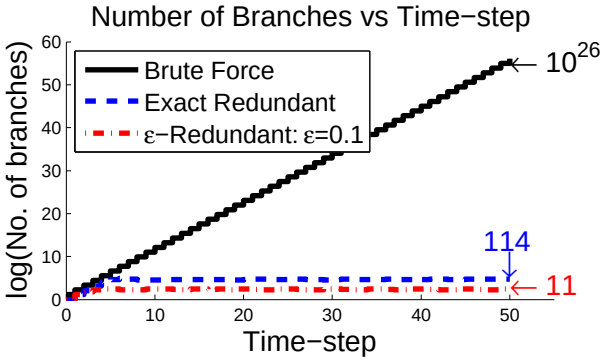


Fig. 2. Comparison of the number of branches at each time-step for the brute force enumeration, the exact redundant elimination and the numerical redundant algorithm (See Section 5) with $\epsilon = 0.1$.

By using the proposed pruning technique, the complexity of the problem could be drastically reduced as displayed in the following example. Consider a system defined by Eqns. (1), (2) with three sensors

$$A = \begin{bmatrix} 0.9 & -0.15 \\ 0.1 & 1.8 \end{bmatrix}, \quad \Sigma_w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$C_1 = \begin{bmatrix} 1.0 & 0.0 \end{bmatrix}, \quad \Sigma_{v_1} = 0.1,$$
$$C_2 = \begin{bmatrix} 0.0 & 1.0 \end{bmatrix}, \quad \Sigma_{v_2} = 0.3,$$
$$C_3 = \begin{bmatrix} 0.25 & -0.75 \end{bmatrix}, \quad \Sigma_{v_3} = 0.2,$$

and a schedule horizon of $N = 50$. Figure 2 compares the number of branches required for the brute force enumeration versus Algorithm 2 which also provides the optimal solution but prunes redundant branches. At the final time-step, there are 10^{26} branches in the whole search tree, but only 114 branches are required for Algorithm 2.

Even though the optimal solution prunes a large number of branches, the growth of the search tree may still become prohibitive for some problems. Therefore, an approximate solution may be desired.

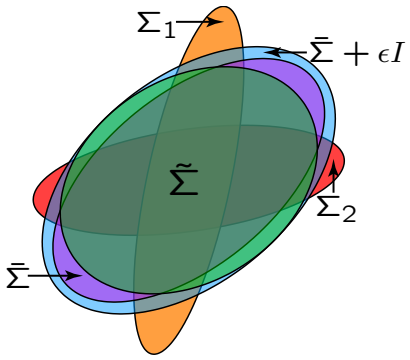


Fig. 3. Example covariances that demonstrate the concept of ϵ-redundancy. One possible convex combination of Σ_1 and Σ_2 is represented by $\tilde{\Sigma}$ and let $\tilde{\gamma}$ be the same convex combination of their costs. Σ_1 and Σ_2 cannot strictly eliminate $\bar{\Sigma}$ because it does not contain $\tilde{\Sigma}$. However, if the condition is relaxed by ϵ, then $\bar{\Sigma} + \epsilon I$ does contain $\tilde{\Sigma}$; consequently, if $\bar{\gamma} + \epsilon \geq \tilde{\gamma}$ then the branch can be eliminated.

5 Suboptimal Scheduling

5.1 Suboptimal Scheduling Algorithm

To further reduce the complexity, the algebraic redundancy concept can be generalized to allow for numerical error. Similar to Definition 2, the following definition provides a condition for testing the ϵ-redundancy of a matrix.

Definition 4 (ϵ-Redundancy) *A pair $(\Sigma, \gamma) \in \mathcal{H}$ is called ϵ-redundant with respect to $\mathcal{H} \setminus \{(\Sigma, \gamma)\}$, if there exist nonnegative constants $\{\alpha_i\}_{i=1}^{l-1}$ such that*

$$\sum_{i=1}^{l-1} \alpha_i = 1, \quad \begin{bmatrix} \Sigma + \epsilon I & 0 \\ 0 & \gamma + \epsilon \end{bmatrix} \succeq \sum_{i=1}^{l-1} \alpha_i \begin{bmatrix} \Sigma(i) & 0 \\ 0 & \gamma(i) \end{bmatrix}$$

where $l = |\mathcal{H}|$ and $\{(\Sigma(i), \gamma(i))\}_{i=1}^{l-1}$ is an enumeration of $\mathcal{H} \setminus \{(\Sigma, \gamma)\}$.

Figure 3 illustrates the premise behind ϵ-redundancy with respect to Definition 4. Let $\tilde{\Sigma}$ represent one possible convex combination of Σ_1 and Σ_2. In this example, Σ_1 and Σ_2 cannot strictly eliminate $\bar{\Sigma}$ because $\bar{\Sigma}$ does not contain $\tilde{\Sigma}$. However, if the condition is relaxed for some $\epsilon > 0$, then $\bar{\Sigma} + \epsilon I$ contains $\tilde{\Sigma}$. Consequently, for that ϵ if $\bar{\gamma} + \epsilon$ is greater than the same convex combination of γ_1 and γ_2 then $\bar{\Sigma}$ can be eliminated. By using the ϵ-approximation it further reduces the number of branches in the search tree and the complexity of the problem. Consequently, this may enable the solution of problems that are otherwise intractable for the optimal algorithm.

Similar to Algorithm 1, the convex condition in Definition 4 can be used to identify and prune ϵ-redundant matrices of a characteristic set. Denote by $ES_\epsilon(\mathcal{H})$ the set

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

of the remaining pairs after removing all the ϵ-redundant pairs in $\mathcal{H}$ that satisfy the conditions given in Definition 4. To determine the ϵ-approximate solution of the sensor scheduling problem, Algorithm 2 can be modified by substituting $ES_\epsilon(\cdot)$ for $ES(\cdot)$. The modified algorithm will be referred to as the *suboptimal scheduling algorithm (or $ALGO_\epsilon$)* in the rest of this paper. The essential part of this algorithm is to compute the so-called ϵ-relaxed characteristic sets $\{\mathcal{H}_k^\epsilon\}_{k=0}^N$

$$\mathcal{H}_k^\epsilon = ES_\epsilon(h_{\mathbb{M}}\left(\mathcal{H}_{k-1}^\epsilon\right)), \text{with } \mathcal{H}_0^\epsilon = \{(\Sigma_0, 0)\}. \quad (9)$$

The set $\mathcal{H}_N^\epsilon$ typically contains fewer pairs than $\mathcal{H}_N$ and is easier to compute. To simplify the computation, the schedule that minimizes $J_N(\sigma)$ among all the schedules in $\mathbb{M}\left(\mathcal{H}_N^\epsilon\right)$ can be used as an alternative to the optimal schedule. Denote by $\sigma^{\epsilon,N}$ the suboptimal schedule computed by $ALGO_\epsilon$, namely,

$$\sigma^{\epsilon,N} = \underset{\sigma\in\mathbb{M}\left(\mathcal{H}_N^\epsilon\right)}{\arg\min}\ J_N(\sigma).$$

An example of the complexity for the suboptimal algorithm is shown in Figure 2 for an $\epsilon = 0.1$. By eliminating ϵ-redundant pairs, the complexity is reduced from 114 for the optimal algorithm to 11. While the suboptimal algorithm drastically reduces the computational complexity, it might sacrifice the quality of the solution. Consequently, an upper bound on the distance from the optimal solution is needed.

5.2 Performance Analysis

For each $k \leq N$, define the *(k-horizon) relaxed value function* V_k^ϵ as

$$V_k^\epsilon = \min_{\sigma\in\mathbb{M}(\mathcal{H}_k^\epsilon)} J_k(\sigma) = \min_{(\Sigma,\gamma)\in\mathcal{H}_k^\epsilon} \gamma. \quad (10)$$

Under this notation, the cost associated with the schedule returned by the suboptimal algorithm is V_N^ϵ. The goal of this section is to derive an upper bound for the average-per-stage error, namely $\frac{1}{N}(V_N^\epsilon - V_N)$, incurred by the relaxation procedure of Eqn. (9).

5.2.1 Perturbation Analysis of the Riccati Mapping

For each $i \in \mathbb{M}$ and $\Sigma \in \mathcal{A}$, define

$$\bar{A}_i(\Sigma) \triangleq A - AK_i(\Sigma)C_i, \quad (11)$$

where $K_i(\Sigma)$ is the Kalman gain associated with sensor i and matrix Σ defined as

$$K_i(\Sigma) = \Sigma C_i^{\mathrm{T}}(C_i\Sigma C_i^{\mathrm{T}} + \Sigma_{v_i})^{-1}.$$

To develop an analytical expression for the bound of the average-per-stage error, the effect of a perturbation of the initial covariance on all future covariances must be determined. To this end, the directional derivative of the Riccati mapping is first characterized.

Lemma 1 *For each $i \in \mathbb{M}$ and any $\Sigma, Q \in \mathcal{A}$,*

$$\left.\frac{d\rho_i(\Sigma+\epsilon Q)}{d\epsilon}\right|_{\epsilon=0} = \bar{A}_i(\Sigma)Q\bar{A}_i(\Sigma)^T,$$

where $\bar{A}_i(\Sigma)$ is defined in Eqn. (11).

PROOF. Let $i \in \mathbb{M}$, $\Sigma \in \mathcal{A}$ and $Q \in \mathcal{A}$ be arbitrary but fixed. Define $f(\epsilon) = C_i(\Sigma+\epsilon Q)C_i^{\mathrm{T}} + \Sigma_{v_i}$. It can be easily shown that,

$$\frac{df^{-1}(\epsilon)}{d\epsilon} = -f^{-1}(\epsilon)C_iQC_i^{\mathrm{T}}f^{-1}(\epsilon).$$

Taking the derivative of $\rho_i(\Sigma+\epsilon Q)$ with respect to ϵ and letting $\epsilon = 0$ yields

$$\begin{aligned}\left.\frac{d\rho_i(\Sigma+\epsilon Q)}{d\epsilon}\right|_{\epsilon=0} &= A\Sigma C_i^{\mathrm{T}}f^{-1}(0)C_iQC_i^{\mathrm{T}}f^{-1}(0)C_i\Sigma A^{\mathrm{T}} \\ &+AQA^{\mathrm{T}} - AQC_i^{\mathrm{T}}f^{-1}(0)C_i\Sigma A^{\mathrm{T}} - A\Sigma C_i^{\mathrm{T}}f^{-1}(0)C_iQA^{\mathrm{T}} \\ &= A\left[(I-\Sigma C_i^{\mathrm{T}}f^{-1}(0)C_i)Q(I-C_i^{\mathrm{T}}f^{-1}(0)C_i\Sigma)\right]A^{\mathrm{T}}.\end{aligned}$$

Noting that $f^{-1}(0) = \left(C_i\Sigma C_i^{\mathrm{T}} + \Sigma_{v_i}\right)^{-1}$ and by the definition of $\bar{A}_i(\Sigma)$, the desired result is obtained. □

Theorem 3 *For any $\Sigma, Q \in \mathcal{A}$, $i \in \mathbb{M}$ and $\epsilon \in \mathbb{R}_+$, the following holds:*

$$\rho_i(\Sigma+\epsilon Q) \preceq \rho_i(\Sigma) + \left(\bar{A}_i(\Sigma)Q\bar{A}_i(\Sigma)^T\right)\epsilon. \quad (12)$$

PROOF. By the concavity of the Riccati mapping (Theorem 1), it can be easily verified that the mapping $\mu_{i,\Sigma,Q} : \mathbb{R}_+ \to \mathcal{A}$ defined by $\mu_{i,\Sigma,Q}(\epsilon) = \rho_i(\Sigma+\epsilon Q)$, $\forall \epsilon \in \mathbb{R}_+$, is also concave in ϵ for any $i \in \mathbb{M}$. Thus $\mu_{i,\Sigma,Q}(\epsilon)$ can be upper bounded by an affine function of ϵ, namely, $\mu_{i,\Sigma,Q}(0) + \mu'_{i,\Sigma,Q}(0)\epsilon$, which together with Lemma 1 yields the desired inequality. □

For any $k = 0, \ldots, N$, $\Sigma \in \mathcal{A}$ and $\sigma \in \mathbb{M}^N$, denote by $g_k^\sigma(\Sigma; Q)$ the directional derivative of the k-step Riccati mapping ϕ_k^σ at Σ along an arbitrary direction $Q \in \mathcal{A}$, i.e.,

$$g_k^\sigma(\Sigma; Q) \triangleq \left.\frac{d\phi_k^\sigma(\Sigma+\epsilon Q)}{d\epsilon}\right|_{\epsilon=0}. \quad (13)$$

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

The following lemma describes how to analytically compute g_k^σ, which will be useful in determining a bound for the error incurred using the suboptimal algorithm.

Lemma 2 *For any $\Sigma, Q \in \mathcal{A}$ and $\sigma \in \mathbb{M}^N$, it holds that $g_0^\sigma(\Sigma; Q) = Q$ and*

$$g_k^\sigma(\Sigma; Q) = \prod_{t=k-1}^{0} \left(\bar{A}_{\sigma(t)}(\phi_t^\sigma(\Sigma))\right) Q \prod_{t=0}^{k-1} \left(\bar{A}_{\sigma(t)}(\phi_t^\sigma(\Sigma))\right)^T,$$

for $k = 1, \ldots, N$.

PROOF. For simplicity, let $\hat{A}_t = \bar{A}_{\sigma(t)}(\phi_t^\sigma(\Sigma))$. The case $k = 0$ follows directly from the fact that

$$\phi_0^\sigma(\Sigma + \epsilon Q) = \Sigma + \epsilon Q.$$

Suppose that the result holds for a general $k \leq N - 1$,

$$\phi_k^\sigma(\Sigma + \epsilon I) = \phi_k^\sigma(\Sigma) + \left[\prod_{t=k-1}^{0} \hat{A}_t Q \prod_{t=0}^{k-1} \hat{A}_t^{\mathrm{T}}\right] \epsilon + o(\epsilon),$$

where $o(\epsilon)$ satisfies $o(\epsilon)/\epsilon \to 0$ as $\epsilon \to 0$. Now, it has to be shown that this is also true for $k + 1$. Notice that,

$$\begin{aligned}\phi_{k+1}^\sigma(\Sigma + \epsilon I) &= \rho_{\sigma(k)}(\phi_k^\sigma(\Sigma + \epsilon I)) \\ &= \rho_{\sigma(k)}\left(\phi_k^\sigma(\Sigma) + \left[\prod_{t=k-1}^{0} \hat{A}_t Q \prod_{t=0}^{k-1} \hat{A}_t^{\mathrm{T}}\right] \epsilon + o(\epsilon)\right).\end{aligned}$$

Applying Lemma 1 to the right-hand side will yield the desired result. □

Since the function $g_k^\sigma(\Sigma; Q)$ is a directional derivative, it must be linear in the perturbation direction Q.

Lemma 3 *For any $\Sigma, Q_1, Q_2 \in \mathcal{A}$, $a, b \in \mathbb{R}$, $k \leq N$ and $\sigma \in \mathbb{M}^N$, the following holds:*

$$g_k^\sigma(\Sigma; aQ_1 + bQ_2) = a g_k^\sigma(\Sigma; Q_1) + b g_k^\sigma(\Sigma; Q_2).$$

Similar to Eqn. (12), an affine upper bound for $\phi_k^\sigma(\Sigma + \epsilon I)$ can be obtained using Lemma 2.

Theorem 4 *For any $\Sigma \in \mathcal{A}$, $\epsilon \in \mathbb{R}_+$, $Q \in \mathcal{A}$ and $k = 0, \ldots, N$, the k-step effect of a perturbation, $\Sigma + \epsilon Q$, can be upper bounded by $\phi_k^\sigma(\Sigma + \epsilon Q) \preceq \phi_k^\sigma(\Sigma) + g_k^\sigma(\Sigma; Q)\epsilon$.*

The function $g_k^\sigma(\Sigma; Q)$ quantifies how a perturbation error incurred at some generic time t will affect the error covariance matrix k iterations later, provided that no further perturbation is applied after step t. The following theorem establishes conditions under which the error term $g_k^\sigma(\Sigma)$ decays exponentially as k increases.

Theorem 5 *Fix arbitrary $\Sigma \in \mathcal{A}$, $N \in \mathbb{Z}_+$, and $\sigma \in \mathbb{M}^N$. If there exists a constant $\beta < \infty$ such that $\Sigma_k^\sigma(\Sigma) \preceq \beta I_n$ for all $k \leq N$, then*

$$\mathrm{tr}(g_k^\sigma(\Sigma)) \leq \frac{n\beta}{\lambda_w^-}\eta^k,\ k = 0, \ldots, N,$$

where

$$\eta = \frac{1}{1 + \alpha\lambda_w^-} < 1 \quad \text{and} \quad \alpha = \frac{\lambda_w^-}{\|A\|^2\beta^2 + \lambda_w^-\beta}. \quad (14)$$

Here, $\|A\|$ denotes the Euclidean norm of A.

PROOF. See Appendix.

The above theorem reveals an important property of the k-horizon Riccati mapping $\phi_k^\sigma : \mathcal{A} \to \mathcal{A}$, namely, boundedness of the trajectory implies an exponential disturbance attenuation. This property plays a crucial role in deriving the error bound for the proposed suboptimal algorithm.

5.2.2 Error Bound

Denote by $\sigma^* = \{\sigma^*(0), \ldots, \sigma^*(N-1)\}$ the optimal N-horizon sensor schedule, i.e.,

$$\sigma^* = \arg\min_{\sigma \in \mathbb{M}^N} J_N(\sigma).$$

For each $j = 0, \ldots, N-1$, let σ_j^* be the sensor schedule obtained by removing the first j steps from σ^*, i.e.,

$$\sigma_j^* = \{\sigma^*(j), \ldots, \sigma^*(N-1)\}.$$

Let $\{\hat{\Sigma}_k^*\}_{k=0}^N$ be the optimal covariance trajectory, namely, $\hat{\Sigma}_k^* = \hat{\Sigma}_k^{\sigma^*}$, for $k = 0, \ldots, N$, and let γ_k^* be the accrued cost of the first k steps of the optimal covariance trajectory, i.e., $\gamma_k^* = \sum_{t=1}^k \mathrm{tr}(\hat{\Sigma}_t^*)$, $k = 1, \ldots, N$. Denote by β^* the peak estimation error along the optimal covariance trajectory, namely,

$$\beta^* \triangleq \max_{k=1,\ldots,N} \|\hat{\Sigma}_k^*\|. \quad (15)$$

The following lemma is used to derive the desired error bound.

Lemma 4 *For any $k = 1, \ldots, N$ and $\epsilon \in \mathbb{R}_+$, there always exists a pair $(\Sigma_k, \gamma_k) \in \mathcal{H}_k^\epsilon$ satisfying the following inequalities:*

$$\begin{cases} \Sigma_k \preceq \hat{\Sigma}_k^* + \left[I_n + \sum_{j=1}^{k-1} g_{k-j}^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right]\epsilon \\ \gamma_k \leq \gamma_k^* + \left[k + \sum_{j=1}^{k-1}\sum_{l=1}^{k-j} \mathrm{tr}(g_l^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n))\right]\epsilon. \end{cases} \quad (16)$$

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

PROOF. For $k = 1$, Eqn. (9) leads to

$$\mathcal{H}_1^\epsilon = ES_\epsilon(h_{\mathbb{M}}(\mathcal{H}_0)) = ES_\epsilon(\mathcal{H}_1).$$

Thus, the desired inequalities follow directly from Definition 4.

Suppose that the results hold for a general $k \leq N-1$ and let $(\Sigma_k, \gamma_k) \in \mathcal{H}_k^\epsilon$ be the pair satisfying Eqn. (16). It suffices to show that there also exists a pair $(\Sigma_{k+1}, \gamma_{k+1}) \in \mathcal{H}_{k+1}^\epsilon$ satisfying Eqn. (16) for $k+1$. Define

$$\tilde{\Sigma}_{k+1} = \rho_{\sigma^*(k)}(\Sigma_k), \text{ and } \tilde{\gamma}_{k+1} = \gamma_k + \mathrm{tr}\left(\tilde{\Sigma}_{k+1}\right).$$

Clearly, the pair $(\tilde{\Sigma}_{k+1}, \tilde{\gamma}_{k+1}) \in h_{\mathbb{M}}(\mathcal{H}_k^\epsilon)$, but may not be in $\mathcal{H}_{k+1}^\epsilon$ after applying ES_ϵ. Nevertheless, according to Definition 4, there must exist a pair $(\Sigma_{k+1}, \gamma_{k+1}) \in \mathcal{H}_{k+1}^\epsilon$ such that

$$\begin{cases} \Sigma_{k+1} \preceq \tilde{\Sigma}_{k+1} + \epsilon I_n; \\ \gamma_{k+1} \leq \tilde{\gamma}_{k+1} + \epsilon. \end{cases} \tag{17}$$

According to the induction hypothesis, Eqn. (16), Theorem 4, and Lemma 3,

$$\begin{aligned} \tilde{\Sigma}_{k+1} \preceq\ & \rho_{\sigma^*(k)}\left(\hat{\Sigma}_k^*\right) + \\ & \epsilon g_1^{\sigma^*(k)}\left(\hat{\Sigma}_k^*; I_n + \textstyle\sum_{j=1}^{k-1} g_{k-j}^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right) \\ =\ & \hat{\Sigma}_{k+1}^* + \epsilon g_1^{\sigma^*(k)}\left(\hat{\Sigma}_k^*; I_n\right) \\ & + \epsilon \textstyle\sum_{j=1}^{k-1} g_1^{\sigma^*(k)}\left(\hat{\Sigma}_k^*; g_{k-j}^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right) \end{aligned}$$

Using Lemma 2, it can be easily verified that for each $j = 1, \ldots, k-1$,

$$g_1^{\sigma^*(k)}\left(\hat{\Sigma}_k^*; g_{k-j}^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right) = g_{k+1-j}^{\sigma_j^*}\left(\hat{\Sigma}_j^*; I_n\right).$$

Therefore,

$$\tilde{\Sigma}_{k+1} \preceq \hat{\Sigma}_{k+1}^* + \left[\sum_{j=1}^{k} g_{k+1-j}^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right]\epsilon. \tag{18}$$

By the induction hypothesis and Eqn (16),

$$\begin{aligned} \tilde{\gamma}_{k+1} &= \gamma_k + \mathrm{tr}\left(\tilde{\Sigma}_{k+1}\right) \\ &\leq \gamma_k + \mathrm{tr}(\hat{\Sigma}_{k+1}^*) + \epsilon \textstyle\sum_{j=1}^{k} \mathrm{tr}\left(g_{k+1-j}^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right) \\ &\leq \gamma_k^* + \mathrm{tr}(\hat{\Sigma}_{k+1}^*) + k\epsilon \\ &\quad + \epsilon \textstyle\sum_{j=1}^{k-1}\sum_{l=1}^{k-j} \mathrm{tr}(g_l^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)) \\ &\quad + \epsilon \textstyle\sum_{j=1}^{k} \mathrm{tr}\left(g_{k+1-j}^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right) \\ &\leq \gamma_{k+1}^* + k\epsilon + \epsilon \textstyle\sum_{j=1}^{k}\sum_{l=1}^{k+1-j} \mathrm{tr}(g_l^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)). \end{aligned}$$

Combining equations (17), (18) and (19) yields the desired inequality in equation (16). □

Theorem 6 *Let η^* be the constant defined by Eqn. (14) with β^* in place of β. Then*

$$\frac{1}{N}\left(V_N^\epsilon - V_N\right) \leq \left(\frac{n\beta^*\eta^*}{\lambda_w^-(1-\eta^*)} + 1\right)\epsilon.$$

PROOF. It follows immediately from Lemma 4 that

$$V_N^\epsilon \leq V_N + \epsilon\left[\sum_{j=1}^{N-1}\sum_{l=1}^{N-j} \mathrm{tr}\left(g_l^{\sigma_j^*}(\hat{\Sigma}_j^*; I_n)\right)\right].$$

This together with Theorem 5 and the definition of β^* yields the desired inequality. □

The error bound derived above depends on the peak estimation error β^* of the optimal schedule. Although the exact value of β^* is not available, there are many ways to upper bound β^*. For example, $\beta^* \leq J_N(\sigma^{\epsilon,N})$, due to the following simple fact.

$$\beta^* < \sum_{t=1}^{N} \mathrm{tr}\left(\hat{\Sigma}_t^{\sigma^*}\right) \leq \sum_{t=1}^{N} \mathrm{tr}\left(\hat{\Sigma}_t^{\sigma}\right), \forall \sigma \in \mathbb{M}^N.$$

Notice that regardless of the way of estimating β^*, the upper bound given in Theorem 6 could be conservative. The particular value of the bound may not be of crucial importance, however its analytical form reveals several important properties of the proposed suboptimal algorithm. First of all, the bound clearly indicates that as the relaxation parameter $\epsilon \to 0$, the performance of the suboptimal algorithm $ALGO_\epsilon$ approaches the optimal one asymptotically with no performance gap. In addition, the error in general grows only linearly as ϵ increases. These appealing properties provide theoretical justifications for the suboptimal algorithm developed in this section.

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

6 Numerical Examples

6.1 3D Process with Four Sensors

Consider a sensor scheduling problem with horizon length $N = 50$ and the following system matrices.

$$A = \begin{bmatrix} -0.6 & 0.8 & 0.5 \\ -0.1 & 1.5 & -1.1 \\ 1.1 & 0.4 & -0.2 \end{bmatrix}, \quad \Sigma_w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$C_1 = \begin{bmatrix} 0.75 & -0.2 & -0.65 \end{bmatrix}, \quad \Sigma_{v_1} = 0.53,$$
$$C_2 = \begin{bmatrix} 0.35 & 0.85 & 0.35 \end{bmatrix}, \quad \Sigma_{v_2} = 0.8,$$
$$C_3 = \begin{bmatrix} 0.2 & -0.65 & 1.25 \end{bmatrix}, \quad \Sigma_{v_3} = 0.2,$$
$$C_4 = \begin{bmatrix} 0.7 & 0.5 & 0.5 \end{bmatrix}, \quad \Sigma_{v_4} = 0.5.$$

The brute force search for this problem would require to explore about 10^{30} branches, which is not numerically tractable. However, the problem can be efficiently solved using the proposed suboptimal scheduling algorithm $ALGO_\epsilon$. The algorithm is tested with initial covariance $\Sigma_0 = I_3$ under 4 different relaxation parameters $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$. For all these relaxation parameters, the resulting suboptimal sensor schedule is the same with the same cost function value 850.57. The obtained suboptimal schedule is shown in Figure 4(a). It is interesting to note that the sensor schedule is periodic for the non-transient portion of the schedule, with a repeating sequence of $\{4, 1, 4, 2, 1, 2, 3\}$. Counter-intuitively, even though sensor 2 has a significantly larger sensor noise, it is used more than sensor 3 which has the smallest noise. One reason for this is because sensor 2 provides the most direct information about the 2nd dimension of the state. In addition, Figure 4(b) shows the number of branches in the search tree per time-step for $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$, which saturates around $166, 43, 25$ and 18, respectively. It can also be seen that the number of branches in the search tree typically saturates around a smaller number for a larger ϵ.

6.2 Examples With Randomly Generated Matrices

To further demonstrates its performance, the suboptimal algorithm $ALGO_\epsilon$ is tested using 100 random instances of the sensor scheduling problem with $M = 3$ sensors and state dimension $n = 4$. A relatively small horizon length $N = 14$ is chosen for which the brute force search approach can be carried out to compute the ground truth to examine the cost performance of $ALGO_\epsilon$. In generating the random systems, the pair (A, C_i), $\forall i \in \mathbb{M}$, was restricted to be unobservable, with the exception that if all the sensors were used at once then the system would be fully observable. The rationale for this restriction was to coerce the optimal solution to switch between sensors instead of only using one sensor for the entire time horizon. The noise for each sensor, Σ_{v_i}, was randomly chosen from a uniform distribution between $(0, 1)$, the initial covariance was $\Sigma_0 = 0.1I_4$ and the process noise was $\Sigma_w = I_4$.

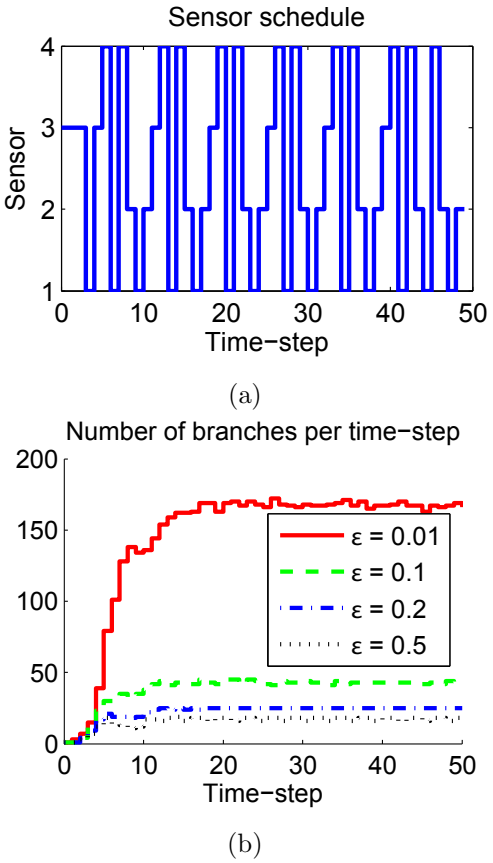
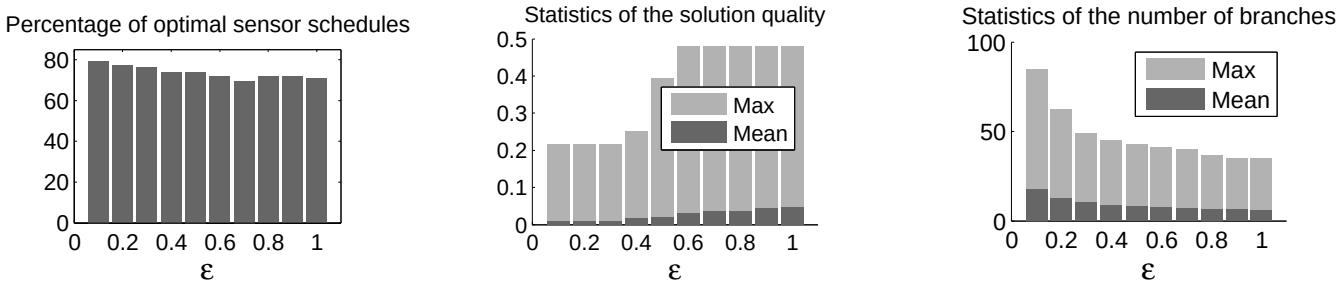


Fig. 4. Results for $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$. (a) Suboptimal sensor schedule. (b) Number of matrices per time-step.

For each problem, both the optimal solution and the suboptimal solutions under 10 different relaxation values $\epsilon = \{0.1, 0.2, \ldots, 1.0\}$ are calculated. Figure 5(a) displays the percentage of the solutions that is optimal for each ϵ. As ϵ is increased there is a slow decrease in the number that is optimal. Figure 5(b) displays the mean and maximum percentage of the final cost over the optimal solution for each ϵ. For all ϵ, the solution is well within 0.5% of the optimal objective function value for most of the instances and is closer to optimal as ϵ is decreased. Figure 5(c) shows the number of branches in the search tree at the final time-step. As ϵ increases, fewer branches are needed to represent the search tree, and even an $\epsilon = 0.1$ requires on average four orders of magnitude fewer branches than brute force enumeration. The figure suggests that the general trend for both the mean and maximum values is an exponential decay as ϵ increases.

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST



(a) (b) (c)

Fig. 5. Performance of the suboptimal algorithm for different ϵ. (a) The percentage of solutions for the suboptimal algorithm that is the optimal solution. (b) Mean and maximum relative error, in percentage, between the suboptimal and optimal solution for each ϵ. (c) The mean and maximum number of branches in the search tree at the final time-step for each ϵ.

7 Application in Active Mapping

One motivating application of this sensor scheduling framework is in simultaneous localization and mapping (SLAM) which is a fundamental task in robotics [27]. SLAM is concerned with constructing a globally consistent model of the environment and consists of a vehicle navigating through and sensing an unknown environment. Examples of typical sensor readings used are monocular or stereo camera images and 2D/3D point clouds. By combining the robot's odometry measurements and the environment measurements a globally consistent map of the environment can be generated by solving a nonlinear estimation problem. Typically, most approaches are only considered with the estimation process and the robot is manually guided through the unknown environment. These approaches, however, neglect how the control inputs and/or path for the robot affects the quality of the map, even though it can have a dramatic effect. In this example, the design of the control inputs to minimize the total estimation error is investigated. Previously, Ny et al. [22] proposed a simple suboptimal greedy solution technique which employed forward value iteration and a myopic heuristic pruning algorithm. The method proposed in [22] can be considered an instance of the proposed algorithm in this work with an $\epsilon = \infty$.

This example illustrates the active mapping problem in Figure 6, which is concerned with planning trajectories for the robot through the environment to acquire the best map. To investigate how the navigation of the robot affects the quality of the map, the environment is assumed to be composed of M distinct features with known locations. These features have an unknown quantity associated with them, for example air quality, and the robot can acquire a noisy measurement of it when in sensing range. The measurements of the features are corrupted by additive Gaussian noise and the noise characteristics depend upon the state of the vehicle, i.e. the further the vehicle is away from the feature the noisier the measurement is. Therefore, different trajectories

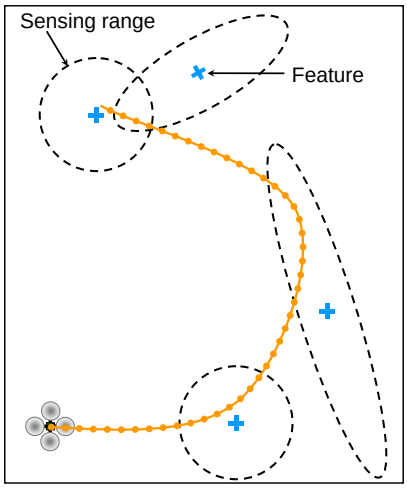


Fig. 6. An illustration of the active mapping problem. There are features scattered throughout the environment with known locations and the robot can receive noisy measurements of them when it is within sensing range.

through the environment will result in different measurement sequences resulting in varying map qualities. Finding the best trajectory through the environment can be transformed to a sensor scheduling problem because each location in the environment has different sensing characteristics. The active mapping problem can be formulated as the following optimization program [22],

$$\begin{aligned} \underset{u_k, \forall k \in [0, N-1]}{\text{minimize}} \quad & \operatorname{tr}\left(\hat{\Sigma}_N\right) \\ \text{s.t.} \quad & s_{k+1} = f\left(s_k, u_k\right) \\ & \hat{\Sigma}_{k+1} = A\hat{\Sigma}_k A^{\mathrm{T}} + \Sigma_w - \\ & \qquad Q(s_{k+1}) R(s_{k+1})^{-1} Q^{\mathrm{T}}(s_{k+1}) \end{aligned}$$

where s_k is the state of the vehicle at time-step k, $f(\cdot)$ is the dynamics of the vehicle, $C(s)$ is defined as the sensor measurement characteristics at the vehicle state s, $Q(\cdot)$ is defined as $Q(s_{k+1}) = A\hat{\Sigma}_k C^{\mathrm{T}}(s_{k+1})$, $R(\cdot)$ is defined as $R(s_{k+1}) = C(s_{k+1})\hat{\Sigma}_k C^{\mathrm{T}}(s_{k+1}) + \Sigma_v(s_{k+1})$,

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

and $\Sigma_v(s)$ is the measurement noise of the sensor at the vehicle state s. The optimization variables are the control inputs, u_k, at each time-step.

In previous sections it was assumed that at each time-step all sensors were available to all branches but in this example that is not the case. Only nodes corresponding to the same state of the vehicle have the same sensors available. Even though the sensors are now state dependent, the proposed pruning algorithm still applies. Using similar notation as introduced in the previous sections, a node is re-defined to include the states of the vehicle, i.e. (Σ, γ, s). Similarly, the Riccati mapping is now defined as,

$$\rho_u(\hat{\Sigma}_k, s_k) = A\hat{\Sigma}_k A^{\mathrm{T}} + \Sigma_w - Q(f(s_k,u))R(f(s_k,u))^{-1}Q^{\mathrm{T}}(f(s_k,u)),$$

and the characteristic sets are now generated via,

$$\mathcal{H}_{k+1} = h_{\mathbb{U}}(\mathcal{H}_k) \text{ from } \mathcal{H}_0 = \{(\Sigma_0, 0, s_0)\} \text{ with}$$

$$h_{\mathbb{U}}(\mathcal{H}) = \{(\rho_u(\Sigma, s), \mathrm{tr}(\rho_u(\Sigma, s)), f(s,u)) : \forall(\Sigma, \gamma, s) \in \mathcal{H}, \forall u \in \mathbb{U}(s)\}.$$

where $\mathbb{U}$ is the set of all inputs and $\mathbb{U}(s)$ is the set of feasible inputs at state s. Let $\mathcal{H}(s)$ be all of the sets in $\mathcal{H}$ with state s. Now the active mapping problem is formulated as a sensor scheduling problem in Algorithm 3. The solution procedure is the same as before except for

Algorithm 3 Active Mapping Algorithm

1: $\mathcal{H}_0 = \{\Sigma_0, 0, s_0\}$
2: **for** $k = 1,\ldots,N$ **do**
3: $\quad\bar{\mathcal{H}} = h_{\mathbb{U}}(\mathcal{H}_{k-1})$
4: $\quad\mathcal{H}_k = \emptyset$
5: $\quad$**for all** s **do**
6: $\quad\quad\mathcal{H}_k = \mathcal{H}_k \cup ES_\epsilon\left(\bar{\mathcal{H}}(s)\right)$
7: $\quad$**end for**
8: **end for**
9: $u^* = \underset{j\in\{1,\ldots,|\mathcal{H}_N|\}}{\arg\min}\ \gamma_N(j)$

lines $5-7$ which applies the pruning algorithm for all sets at the same state. The pruning algorithm can no longer be applied to all the sets in $\mathcal{H}_k$ because not every path will have the same set of sensors available to it over the remaining time-steps. Consequently, the pruning algorithm can only be applied to the sets with the same state.

The results from the proposed algorithm are illustrated through numerical simulations. The environment is represented as a grid world and the vehicle can move to its neighboring positions on the grid. At each time-step, the vehicle can take a noisy measurement of the feature if its location is within the sensing range of that feature. The sensor noise model is assumed to be an affine function of the distance from the feature. In the examples presented, the planning horizon is $N = 50$. The pruning algorithm was implemented in C++ using the semidefinite programming algorithm SDPA [9].

Figure 7 compares the greedy heuristic method and the numerically redundant pruning algorithm with an $\epsilon = 0.1$. The objective function obtained for the greedy method is 0.88 and the redundant pruning method is 0.69. The objective function for the simple greedy, myopic policy is over 27% larger than the solution with $\epsilon = 0.1$. Figures (a)-(b) show the output of the greedy heuristic method and the numerically redundant pruning algorithm, respectively. The feature locations are indicated by the crosses and the maximum sensing range is shown by the circles. The solid line is the trajectory of the vehicle through the environment. Given the myopic nature of the greedy policy, it navigates the vehicle along the area in which the sensing regions overlap; in contrast, the solution for the numerically redundant method deviates away from this area and navigates closer to the features which reduces the overall uncertainty. Figures (c)-(d) show the number of branches at each time-step for each method. For $\epsilon = 0.1$, the number of branches increases exponentially until time-step 35 and then decreases.

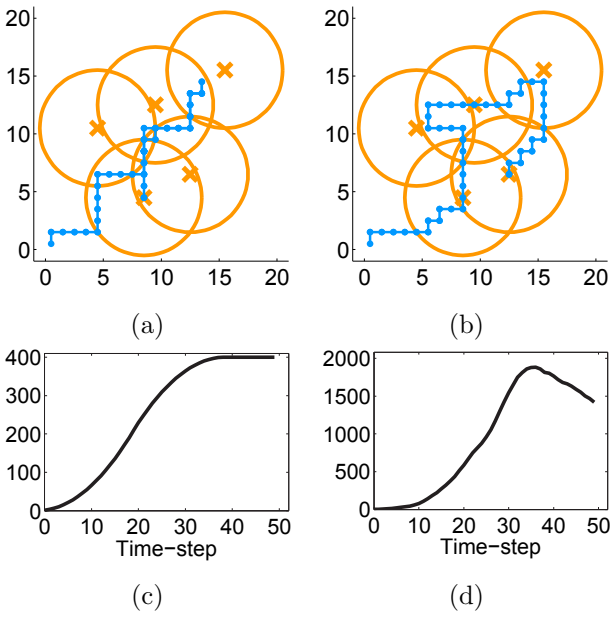


Fig. 7. (a)-(b) The solution from the greedy heuristic method and the numerically redundant pruning method with an $\epsilon = 0.1$, respectively. The feature locations are indicated by the crosses and their sensing radii are shown by the circles. The solid line is the solution of the vehicle's path through the environment. (c)-(d) Number of branches at each time-step for greedy heuristic and the numerically redundant pruning methods, respectively.

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

8 Conclusions

This work has studied the sensor scheduling problem by deriving a condition for which an initial schedule is not part of the optimal solution. Using this condition, two algorithms were devised, which provide the optimal and suboptimal solutions, to prune the search tree to enable the solution of larger systems and longer time horizons. The algorithms trade off the quality of the solution and the complexity of the problem. A bound on the quality of the solution from the suboptimal algorithm was also provided.

While the problem of choosing only one sensor at each time-step is presented, the algorithms developed are also applicable to the case of selecting multiple sensors at each time-step. In this case, the problem's complexity may prevent solving for the optimal solution. However, the suboptimal algorithm can address this by allowing for more error in the solution of the problem which will reduce the complexity of the problem.

There are several interesting areas of future work that the authors wish to explore. First, it has been previously noticed that the sensor schedules tend to be periodic for the non-transient portion of the schedule. The authors would like to analyze this behavior to determine conditions for the periodicity and a bound for the objective function if the periodic schedule were used. This might allow for early termination of the algorithm if the best schedule so far were periodic. Second, the authors want to extend these methods to consider the case in which the sensors depend on the state of the system. Lastly, another extension of interest is to modify the objective function to include a direct measure of the power consumption of each sensor.

References

[1] P. Alriksson and A. Rantzer. Sub-optimal sensor scheduling with error bounds. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[2] S. Arai, Y. Iwatani, and K. Hashimoto. Optimal sensor scheduling of sensors in a sensor network for mobile robot navigation. In *Proceedings of the American Control Conference*, New York City, NY, July 2007.

[3] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, March 1999.

[4] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41:1709–1721, October 2005.

[5] M. S. Branicky and G. Zhang. Solving hybrid control problems: level sets and behavioral programming. In *Proceedings of the American Control Conference*, Chicago, Illinois, June 2000.

[6] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola. Scheduling multiple sensors using particle filters in target tracking. In *IEEE Workshop on Statistical Signal Processing*, pages 549 – 552, September 2003.

[7] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola. On the use of binary programming for sensor scheduling. *IEEE Transactions on Signal Processing*, 55(6):2826 –2839, June 2007.

[8] E. Ertin, J. W. Fisher, and L. C. Potter. Maximum mutual information principle for dynamic sensor query problems. In *Proceedings of the 2nd international conference on information processing in sensor networks*, pages 405–416, Berlin, Heidelberg, 2003. Springer-Verlag.

[9] K. Fujisawa, M. Kojima, K. Nakata, and M. Yamashita. SDPA (SemiDefinite Programming Algorithm) user's manual — version 6.2.0. Technical Report B-308, Dept. Math. and Comp. Sciences, Tokyo Institute of Technology, December 1995.

[10] S. Gao, C. T. Vu, and Y. Li. Sensor scheduling for k-coverage in wireless sensor networks. In *Mobile Ad-hoc and Sensor Networks*, volume 4325 of *Lecture Notes in Computer Science*, pages 268–280. Springer, 2006.

[11] V. Gupta, T. Chung, B. Hassibi, and R. M. Murray. Sensor scheduling algorithms requiring limited computation. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii – 825–8 vol.3, May 2004.

[12] V. Gupta, T. Chung, B. Hassibi, and R. M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251 – 260, February 2006.

[13] Y. He and K. P. Chong. Sensor Scheduling for Target Tracking in Sensor Networks. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[14] V. Isler and R. Bajcsy. The sensor selection problem for bounded uncertainty sensing models. *IEEE Transactions on Automation Science and Engineering*, 3(4):372 –381, October 2006.

[15] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, February 2009.

[16] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1986.

[17] Y. Li, L. W. Krakow, E. K. P. Chong, and K. N. Groom. Approximate stochastic dynamic programming for sensor scheduling to track multiple targets. *Digital Signal Processing*, 19(6):978–989, 2009.

[18] B. Lincoln and A. Rantzer. Relaxed optimal control of piecewise linear systems. In *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*, Brittany, France, June 2003.

[19] S. Martinez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[20] D. K. Meduna, S. L. Rock, and R. McEwen. Low-cost terrain relative navigation for long-range AUVs. In *OCEANS 2008: Oceans, Poles and Climate: Technological Challenges*, Quebec City, Quebec, September 2008.

[21] L. Meier III, J. Peschon, and R. M. Dressler. Optimal control of measurement subsystems. *IEEE Transactions on Automatic Control*, 12(5):528–536, October 1967.

[22] J. L. Ny and G. Pappas. On trajectory optimization for active sensing in Gaussian process models. In *Proceedings*

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

of IEEE International Conference on Decision and Control, Shanghai, China, December 2009.

[23] Y. Oshman. Optimal sensor selection strategy for discrete-time state estimators. *IEEE Transactions on Aerospace and Electronic Systems*, 30(2):307 –314, April 1994.

[24] M. Rezaeian. Sensor scheduling for optimal observability using estimation entropy. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 307 –312, march 2007.

[25] A. V. Savkin, R. J. Evans, and E. Skafidas. The problem of optimal robust sensor scheduling. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 4, pages 3791–3796, December 2000.

[26] B. Sinopoli, M. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, September 2004.

[27] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[28] C. J. Tomlin, J. Lygeros, and S. S. Sastry. A game theoretic approach to controller design for hybrid systems. *In Proceedings of IEEE*, 88(7):949–969, July 2000.

[29] A. Varshavsky, M. Y. Chen, E. de Lara, J. Froehlich, D. Haehnel, J. Hightower, A. LaMarca, F. Potter, T. Sohn, K. Tang, and I. Smith. Are GSM phones the solution for localization? In *Proceedings of IEEE Mobile Computing Systems and Applications*, pages 34–42, August 2006.

[30] H. Wang, K. Yao, G. Pottie, and D. Estrin. Entropy-based sensor selection heuristic for target localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 36–45, New York, NY, USA, 2004. ACM.

[31] W. Zhang, A. Abate, and J. Hu. Efficient suboptimal solutions of switched LQR problems. In *Proceedings of the American Control Conference*, St. Louis, MO, June 2009.

[32] W. Zhang and J. Hu. On the value functions of the optimal quadratic regulation problem for discrete-time switched linear systems. In *Proceedings of IEEE International Conference on Decision and Control*, Cancun, Mexico, December 2008.

[33] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proceedings IROS 2009*, October 2009.

A Proof of Theorem 5

Lemma 5 (Schur Complement Lemma) *Suppose that Z_1, Z_2 and Z_3 are respectively, $n_1 \times n_1$, $n_1 \times n_2$ and $n_2 \times n_2$ dimensional matrices and that both Z_1 and Z_3 are nonsingular. Define*

$$Z = \begin{bmatrix} Z_1 & Z_2 \\ Z_2^T & Z_3 \end{bmatrix}, \; S_1 \triangleq Z_1 - Z_2 Z_3^{-1} Z_2^T$$
$$\text{and } S_2 \triangleq Z_3 - Z_2^T Z_1^{-1} Z_2.$$

Then $Z \succ 0 \Leftrightarrow S_1 \succ 0$ and $Z \succ 0 \Leftrightarrow S_2 \succ 0$.

For any $\Sigma \in \mathcal{A}$, denote by $K_i(\Sigma)$ the Kalman gain associated with sensor $i \in \mathbb{M}$ and covariance matrix Σ, which is given by:

$$K_i(\Sigma) = \Sigma C_i^T (C_i \Sigma C_i^T + \Sigma_i^v)^{-1}. \tag{A.1}$$

For each $i \in \mathbb{M}$ and $\Sigma \in \mathcal{A}$, define

$$\bar{A}_i(Q) \triangleq A - A K_i(Q) C_i. \tag{A.2}$$

Now fix arbitrary $\Sigma \in \mathcal{A}$, $N \in \mathbb{Z}_+$ and $\sigma \in \mathbb{M}^N$. Let β be a constant satisfying the condition stated in Theorem 5. For simplicity, let $\hat{\Sigma}_t := \Sigma_t^\sigma(\Sigma)$ and $\hat{A}_t := A - A K_{\sigma_t}(\hat{\Sigma}_t) C_{\sigma_t}$. Notice that $\hat{\Sigma}_t \succeq \Sigma_w \succ 0$ for all $t \geq 1$. Define

$$\hat{Q}_t = \hat{\Sigma}_t^{-1}, \; \forall t = 1, \ldots, N. \tag{A.3}$$

Lemma 6 *For each $t = 1, \ldots, N-1$,*

$$\hat{Q}_t - \hat{A}_t^T \hat{Q}_{t+1} \hat{A}_t \succeq \alpha I_n, \forall t = 1, \ldots, N-1,$$

where α is defined in (14).

PROOF. By the hypothesis in Theorem 5 $\lambda_w^- I_n \preceq \hat{\Sigma}_t \preceq \beta I_n$, for all $t = 1, \ldots, N$, which implies

$$\frac{1}{\beta} I_n \preceq \hat{Q}_t \preceq \frac{1}{\lambda_w^-} I_n, \; \forall t = 1, \ldots, N. \tag{A.4}$$

Therefore, $\alpha I_n \preceq \alpha\beta \hat{Q}_t$ and thus $\hat{Q}_t - \alpha I_n \succeq (1 - \alpha\beta)\hat{Q}_t$. Since $\alpha\beta < 1$, it follows that

$$\begin{aligned} &\hat{A}_t \left(\hat{Q}_t - \alpha I_n \right)^{-1} \hat{A}_t^T \preceq (1 - \alpha\beta)^{-1} \hat{A}_t \hat{Q}_t^{-1} \hat{A}_t^T \\ =&\hat{A}_t \hat{Q}_t^{-1} \hat{A}_t^T + \left(\frac{1}{1 - \alpha\beta} - 1 \right) \hat{A}_t \hat{Q}_t^{-1} \hat{A}_t^T \\ \preceq&\hat{A}_t \hat{Q}_t^{-1} \hat{A}_t^T + \frac{\alpha\beta^2 \|A\|^2}{1 - \alpha\beta} I_n \\ =&\hat{A}_t \hat{Q}_t^{-1} \hat{A}_t^T + \lambda_w^- I_n. \end{aligned} \tag{A.5}$$

Furthermore, equation (3) leads to

$$\hat{\Sigma}_{t+1} - \hat{A}_t \hat{\Sigma}_t \hat{A}_t^T \succeq \lambda_w^- I_n,$$

which in turn implies

$$\hat{Q}_{t+1}^{-1} - \hat{A}_t \hat{Q}_t^{-1} \hat{A}_t^T \succeq \lambda_w^- I_n.$$

This together with (A.5) yields

$$\hat{Q}_{t+1}^{-1} - \hat{A}_t (\hat{Q}_t - \alpha I_n)^{-1} \hat{A}_t^T \succeq \hat{Q}_{t+1}^{-1} - \hat{A}_t \hat{Q}_t^{-1} \hat{A}_t^T - \lambda_w^- I_n \succeq 0.$$

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST

By Lemma 5, this indicates that

$$\begin{bmatrix} \hat{Q}_{t+1}^{-1} & \hat{A}_t \\ \hat{A}_t^T & \hat{Q}_t - \alpha I_n \end{bmatrix} \succeq 0.$$

Using Lemma 5 again yields $\hat{Q}_t - \alpha I_n - \hat{A}_t^T \hat{Q}_{t+1} \hat{A}_t \succeq 0$. □

PROOF. [**Proof of Theorem 5**] Let $\hat{\Sigma}_t$, $\hat{A}_t$, $\hat{Q}_t$ be the same as defined in the proof of Lemma 6. For each $l = 1, \ldots, n$, let $\xi^{(l)}(t)$ be the solution of the following linear time-varying system:

$$\xi^{(l)}(t+1) = \hat{A}_t \xi^{(l)}(t),\ t = 0, \ldots, N-1, \text{ with } \xi(0) = u^{(l)}.$$

where $u^{(l)}$ denotes the standard unit vector in $\mathbb{R}^n$ with value 1 at the l^{th} position and zeros elsewhere. It can be easily verified that

$$\mathrm{tr}(g_t^\sigma) = \mathrm{tr}\left(\sum_{l=1}^{n} \left(\xi^{(l)}(t)\right) \left(\xi^{(l)}(t)\right)^T \right) = \sum_{l=1}^{n} \|\xi^{(l)}(t)\|^2.$$

For each $l = 1, \ldots, n$, consider the Lyapunov function defined by:

$$L_t^{(l)} \triangleq \xi^{(l)}(t)^T \hat{Q}_t \xi^{(l)}(t), \text{ for } t = 0, \ldots, N.$$

By Lemma 6, for each $t = 0, \ldots, N-1$, it follows that

$$\begin{aligned} L_t^{(l)} - L_{t+1}^{(l)} &= \left(\xi^{(l)}(t)\right)^T \left(\hat{Q}_t - \hat{A}_t^T \hat{Q}_{t+1} \hat{A}_t\right) \left(\xi^{(l)}(t)\right) \\ &\geq \alpha \|\xi^{(l)}(t)\|^2 \geq \alpha \lambda_w^- L_t^{(l)}. \end{aligned}$$

This also implies that $L_t^{(l)}$ is non-increasing. Thus, $L_t^{(l)} - L_{t+1}^{(l)} \geq \alpha \lambda_w^- L_{t+1}^{(l)}$. Hence,

$$L_{t+1}^{(l)} \leq \frac{1}{1 + \alpha \lambda_w^-} L_t^{(l)} \leq \eta^{t+1} L_0^{(l)}.$$

This together with (A.4) implies that for $t = 0, \ldots, N$,

$$\|\xi^{(l)}(t)\|^2 \leq \beta L_t^{(l)} \leq \beta \eta^t L_0^{(l)} \leq \frac{\beta}{\lambda_w^-} \eta^t \|u^{(l)}\|^2 = \frac{\beta}{\lambda_w^-} \eta^t.$$

Therefore, $\mathrm{tr}(g_t^\sigma) = \sum_{l=1}^{n} \|\xi^{(l)}(t)\|^2 \leq \frac{n\beta}{\lambda_w^-} \eta^t$. □

Preprint submitted to Automatica
Received November 3, 2011 15:10:44 PST