On Efficient Sensor Scheduling for Linear Dynamical Systems

Michael P. Vitus, Wei Zhang, Alessandro Abate, Jianghai Hu and Claire J. Tomlin

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 covariance of the state estimation at each time-step. This work investigates the development of tractable algorithms to solve for the optimal and suboptimal sensor schedule. First, a condition on the non-optimality of an initialization of the schedule is presented. Second, using this condition, both an optimal and a suboptimal algorithm are devised to prune the search tree of all possible sensor schedules. This pruning enables the solution of larger systems and longer time horizons than with enumeration alone. The suboptimal algorithm trades off the quality of the solution and the complexity of the problem through a tuning parameter. Third, a hierarchical algorithm is formulated to decrease the computation time of the suboptimal algorithm by using results from a low complexity solution to further prune the tree. Numerical simulations are performed to demonstrate the performance of the proposed algorithms.**

I. Introduction

The problem of *sensor scheduling* is to select one out of multiple available sensors at each time-step to minimize a weighted sum of all the estimation errors over a certain time horizon. Sensor scheduling is an essential technology for applications that have constraints in which only a subset of the sensors can operate at each time-step. An example of such a system is a wireless sensor network comprised of multiple nodes monitoring an external process. The nodes perform some local processing of the data which is then transmitted to a central aggregation process. Constraints on the network's communication bandwidth might not allow all of the nodes to communicate at each time-step. Also, each node may only have a limited amount of power and therefore it should be turned off to conserve power when its measurement is not required. Consequently, the objective is to manage the schedule of nodes' measurements. Sensor scheduling can also be used to handle sensors which interfere with one another, as with sonar range-finding sensors, and thus cannot operate at the same time.

Meier et al. [1] 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. In [2], a relaxed dynamic programming procedure is applied to obtain a suboptimal strategy, which is bounded by a pre-specified distance to optimality, that minimizes the trace of the final state estimate covariance. A convex optimization procedure was developed in [3] 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 [4], a sliding window and thresholding algorithm were applied in order to prune the search tree; a suboptimal solution with no bound on the optimality was obtained. A method that switches sensors randomly according to a probability distribution to obtain the best upper bound on the expected steady-state performance was developed in [5].

The sensor scheduling problem can also be thought of as the dual of an optimal control problem for switched systems, which form a class of hybrid systems. A switched system consists of a family of subsystems, each with specified dynamics, and allows for 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 [6], [7], [8], [9], [10], [11], [12]. Specifically, Zhang et al. [11], [12] 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 four main contributions that arise out of the insights from the control of switched systems in [11], [12]. First, a condition is presented that expresses when the initialization of a sensor schedule is not optimal. Second, based on the previous condition, two efficient pruning techniques are developed which provide optimal and suboptimal solutions. These algorithms can significantly reduce the computation complexity and thus enable the solution of larger systems with longer scheduling horizon than through brute force enumeration. 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 *a posteriori* bound on the quality of the solution from the suboptimal solution is presented. Fourth, a hierarchical algorithm is formulated which reduces the complexity of the problem while maintaining the quality of the solution.

The paper proceeds as follows. Section II describes the

Michael P. Vitus is with the Department of Aeronautics and Astronautics, Stanford University, CA 94305. Email: vitus@stanford.edu

Alessandro Abate is with the Department of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Netherlands. Email: a.abate@tudelft.nl

Wei Zhang and Jianghai Hu are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906. Email:{zhang70, jianghai}@purdue.edu

Claire J. Tomlin is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. Email: tomlin@eecs.berkeley.edu

standard sensor scheduling problem formulation. Then, several properties of the objective function are explored and a theorem which is useful for pruning branches in the search tree is presented in Section III. In Section IV, a description of tractable algorithms for determining the optimal and suboptimal solutions is provided, and the performance of both algorithms is explored through random simulations. In Section V, the hierarchical method is formulated. The paper concludes with directions of future work.

II. 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 horizon. The initial state, $x(0)$, is assumed to be 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 dynamics 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 also assumed to be mutually independent. Denote by $\mathbb{M}^t$ the set of all ordered sequences of sensor schedules of length t where $t\leq N$. An element $\sigma^t=\{\sigma_0^t,\sigma_1^t,\ldots,\sigma_{t-1}^t\}\in\mathbb{M}^t$ is called a (t-horizon) *sensor schedule*. Under a given sensor schedule σ^t, the measurement sequence is,

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

For each $k\leq t$ with $t\leq N$ and each $\sigma^t\in\mathbb{M}^t$, let $\hat{\Sigma}_k^{\sigma^t}$ be the 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 covariance of the system state estimate evolves according to the Riccati recursion,

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

with initial condition $\hat{\Sigma}_0=\Sigma_0$ and $k\leq t$. Define $V\left(\sigma^t\right):\mathbb{M}^t\to\mathbb{R}_+$ as the accrued cost of the weighted trace of the estimation error covariance matrix at each time-step,

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

where q_k is a non-negative scalar weighting factor for each time-step. The use of this objective function also enables the representation of the estimation error at the final time-step by setting $q_k=0,\ \forall k\neq N$. Formally, the objective is,

$$\underset{\sigma^N\in\mathbb{M}^N}{\text{minimize}}\ V\left(\sigma^N\right). \tag{5}$$

III. Properties of the Objective Function

Let $\mathcal{A}$ denote the *positive semidefinite cone*, which is the set of all symmetric positive semidefinite matrices. Similar to [12], a *Riccati Mapping*, $\rho_i:\mathcal{A}\to\mathcal{A}$ can be defined, which maps the current covariance matrix, $\hat{\Sigma}_k$, under the measurement from sensor $i\in\mathbb{M}$ to the next covariance matrix of the estimate,

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

A k-horizon Riccati mapping, $\phi_k^{\sigma^t}:\mathcal{A}\to\mathcal{A}$ can similarly be defined, which maps the covariance matrix, Σ_0, under the first k elements of the sensor schedule σ^t,

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

Definition 1 (Switched Riccati Mapping): *The mapping $\rho_{\mathbb{M}}:2^{\mathcal{A}}\to 2^{\mathcal{A}}$, defined by,*

$$\rho_{\mathbb{M}}\left(\mathcal{S}\right)=\left\{\rho_i\left(\hat{\Sigma}\right):\forall i\in\mathbb{M},\ \forall\hat{\Sigma}\in\mathcal{S}\right\},\mathcal{S}\in 2^{\mathcal{A}}$$

is called the switched Riccati mapping.

The switched Riccati mapping maps the set of positive semidefinite matrices to another set of positive semidefinite matrices by mapping each matrix in $\mathcal{S}$ through each possible sensor measurement.

Definition 2 (Characteristic Sets): *Let $\{\mathcal{H}_k\}_{k=0}^N$ be defined as the characteristic sets as they completely characterize the objective function. Each set is of the form $(\Sigma,\gamma)\in\mathcal{A}\times\mathbb{R}_+$ and is generated recursively by:*

$$\mathcal{H}_{k+1}=h_{\mathbb{M}}\left(\mathcal{H}_k\right)\ \textit{from}\ \mathcal{H}_0=\left\{\left(\Sigma_0,\mathrm{tr}\left(\Sigma_0\right)\right)\right\}\ \textit{with}$$

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

Let $h_{\mathbb{M}}(\cdot)$ be referred to as the characteristic set mapping. The characteristic sets grow exponentially in size from the singleton set $\{(\Sigma_0,\mathrm{tr}(\Sigma_0))\}$ to the set $\mathcal{H}_N$ consisting of up to M^N pairs of a positive semidefinite matrix and an accrued cost. These sets express the covariance of the estimate and the objective cost at every time-step under every possible sensor schedule.

Let $\mathcal{H}_k(i)$ be the i^{th} element of the set $\mathcal{H}_k$, $\Sigma_k(i)$ and $\gamma_k(i)$ be the covariance matrix and objective cost, respectively, corresponding to the i^{th} element of the set $\mathcal{H}_k$, $\kappa(\Sigma_k(i))\in\mathbb{M}^k$ be the ordered sensor schedule corresponding to the covariance estimate of the state $\Sigma_k(i)$ and κ^* be the optimal sensor schedule for the problem. Figure 1 depicts the search tree for an example with two sensors. The tree grows exponentially with each time-step, requiring careful development of computationally-tractable solutions.

The main idea of the subsequent solution methods 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 $\rho_i\left(\Sigma_1\right)\preceq\rho_i\left(\Sigma_2\right)$;*
*(ii) [**Concavity**] $\rho_i\left(c\Sigma_1+(1-c)\Sigma_2\right)\succeq c\rho_i\left(\Sigma_1\right)+(1-c)\rho_i\left(\Sigma_2\right)$, $\forall c\in[0,1]$.*

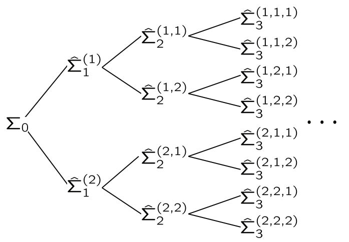

Σ_0
$\hat{\Sigma}_1^{(1)}$
$\hat{\Sigma}_1^{(2)}$
$\hat{\Sigma}_2^{(1,1)}$
$\hat{\Sigma}_2^{(1,2)}$
$\hat{\Sigma}_2^{(2,1)}$
$\hat{\Sigma}_2^{(2,2)}$
$\hat{\Sigma}_3^{(1,1,1)}$
$\hat{\Sigma}_3^{(1,1,2)}$
$\hat{\Sigma}_3^{(1,2,1)}$
$\hat{\Sigma}_3^{(1,2,2)}$
$\hat{\Sigma}_3^{(2,1,1)}$
$\hat{\Sigma}_3^{(2,1,2)}$
$\hat{\Sigma}_3^{(2,2,1)}$
$\hat{\Sigma}_3^{(2,2,2)}$
$\cdots$

Fig. 1. The search tree for the sensor scheduling problem for an example with two sensors. This tree is the enumeration of all possible sensor schedules and the covariance of the estimate at each time-step. The superscript for each covariance matrix, $\hat{\Sigma}$, is the sensor schedule used to obtain that estimate of the state.

Remark 1: *The monotonicity property is a well-known result and its proof is provided in [13]. The concavity property is an immediate consequence of Lemma 1-(e) in [14].*

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 into how to reduce the complexity of the scheduling problem.

Theorem 1 can be repeatedly applied to result in the following corollary.

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

Definition 3 (Algebraic Redundancy): *A pair $(\Sigma, \gamma) \in \mathcal{H}$ is called algebraically redundant with respect to $\mathcal{H} \setminus \{(\Sigma, \gamma)\}$, if there exist nonnegative constants $\{\alpha_i\}_{i=1}^{l-1}$ such that*

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

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

Using the results from Corollary 1 and Definition 3, the following theorem provides a condition which characterizes the branches that can be pruned without eliminating the optimal solution of the sensor scheduling problem.

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

Proof: It suffices to show that there exists a pair $\left(\tilde{\Sigma}, \tilde{\gamma}\right) \in \mathcal{H}_t \setminus (\Sigma, \gamma)$ such that $\forall \sigma^{N-t} \in \mathbb{M}^{N-t}$,

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

Let $\{\alpha_i\}_{i=1}^{l-1}$ be the constants satisfying Definition 3. From the monotonicity and concavity of $\phi_k^{\sigma^r}$,

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

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

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

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

IV. Description of Algorithms

Using the properties of the objective function and search tree, two efficient algorithms will be developed. The first method will employ pruning of the search tree via Theorem 2 to provide the optimal solution. The second method will reduce the complexity of the problem even more but will only yield a suboptimal solution.

A. Optimal Solution

The method of enumerating all possible sensor schedules is only tractable for relatively short time horizons, but through efficient pruning of the search tree, larger time horizon solutions are possible. Theorem 2 can be used to define a condition that characterizes the redundancy of a branch with respect to other branches. Consequently, an efficient algorithm can be developed which uses Theorem 2.

Let an equivalent subset of the search tree be defined as one that still contains the optimal sensor schedule. In computing the characteristic sets, Definition 3 can be applied to calculate an equivalent subset of $\mathcal{H}_k$, $\forall k \in \{1, \ldots, N\}$, which is outlined 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 initialized to the current best branch. Next, each entry in $\mathcal{H}_k$ is tested with the current equivalent subset, $\mathcal{H}_k^{(i-1)}$, to determine if it can be eliminated. If not, then it is appended to the current subset.

Algorithm 1 Computation of the Equivalent Subsets

```
sort $\mathcal{H}_k$ in ascending order such that $\gamma(i) \leq \gamma(i+1)$,
  $\forall i \in \{1, \ldots, |\mathcal{H}_k| - 1\}$.
$\mathcal{H}_k^{(i)} = \{\mathcal{H}_k(1)\}$
for $i$ = 2,. . .,$|\mathcal{H}_k|$ do
    if $\mathcal{H}_k(i)$ satisfies Definition 3 with $\mathcal{H}_k^{(i-1)}$ then
        $\mathcal{H}_k^{(i)} = \mathcal{H}_k^{(i-1)}$
    else
        $\mathcal{H}_k^{(i)} = \mathcal{H}_k^{(i-1)} \cup \mathcal{H}_k(i)$
    end if
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 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.

Algorithm 2 Sensor Scheduling for a Finite Horizon

```
$\mathcal{H}_0 = \{(\Sigma_0, \mathrm{tr}(\Sigma_0))\}$
for $k = 1, \ldots, N$ do
    $\mathcal{H}_k = h_{\mathbb{M}}(\mathcal{H}_{k-1})$
    Perform Algorithm 1 with $\mathcal{H}_k$
end for
$\kappa^* = \underset{j \in \{1,\ldots,|\mathcal{H}_N|\}}{\arg\min} V(\kappa(\mathcal{H}_N(j)))$
```

The complexity of the algorithm will be demonstrated through an example. Consider the system defined by Eqns. (1), (2) with three sensors and a schedule horizon of $N = 50$ with $q_k = 1, \forall k$. Figure 2 compares the number of branches required to obtain the optimal solution to the sensor scheduling problem for the brute force enumeration of all the branches 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 tree, but only 124 are required for the pruning algorithm.

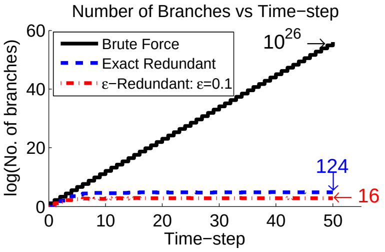


Fig. 2. Comparison of the number of branches at each time-step for the brute force enumeration, the exact redundant elimination and the numerical redundant algorithm with $\epsilon = 0.1$.

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.

B. Suboptimal Solution

The following section describes an algorithm which approximates the search tree by pruning branches which are numerically redundant. Similar to Definition 3, the following definition provides a condition for testing the ϵ-redundancy of a matrix.

Definition 4 (ϵ-Redundant): *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. In this example, Σ_1 and Σ_2 cannot strictly eliminate $\bar{\Sigma}$ but for some ϵ, one possible convex combination of Σ_1 and Σ_2, represented by $\tilde{\Sigma}$, is completely contained within $\bar{\Sigma} + \epsilon I$. Consequently, for that ϵ if $\bar{\gamma}$ is greater than the same convex combination of γ_1 and γ_2 then $\bar{\Sigma}$ can be eliminated. Using the ϵ-approximation 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.

From Theorem 2, only the solution from the relaxed branch $\bar{\Sigma} + \epsilon I_n$ can be guaranteed to be worse than the remaining solutions. Therefore, the optimal solution can no longer be guaranteed, but Theorem 3 provides an *a posteriori* bound on the quality of the solution.

To efficiently compute the ϵ-equivalent subset, Algorithm 1 can be modified to use Definition 4 instead of Definition 3; the result is denoted $Algo_{NR}(\epsilon)$. Finally, to determine the ϵ-approximate solution of the sensor scheduling problem, Algorithm 2 can be modified by substituting $Algo_{NR}(\epsilon)$ for Algorithm 1; the result is referred to as $Algo_S(\epsilon)$.

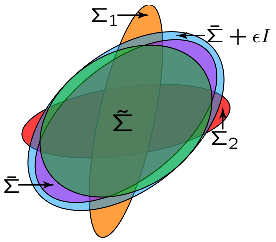


Fig. 3. Example covariances that demonstrate the concept of ϵ-redundancy.

As compared with the algebraic redundancy, the ϵ-redundancy concept can typically eliminate many more branches of the search tree. Thus, $Algo_S(\epsilon)$ can usually be carried out rather efficiently. Before further illustrating the simplicity of $Algo_S(\epsilon)$ through numerical examples, an upper bound for the objective function corresponding to the schedule returned by $Algo_S(\epsilon)$ over the optimal solution is provided.

Theorem 3: *Let* κ^* *be the optimal* N*-horizon sensor schedule and* κ^ϵ *be the* N*-horizon schedule returned by* $Algo_S(\epsilon)$ *for some* $\epsilon > 0$*. Then there exists a finite constant* β *such that* $\frac{1}{N}\left(V(\kappa^\epsilon) - V(\kappa^*)\right) \leq \beta \cdot \epsilon$

Proof: Due to the limited space, the reader is referred to [15] for the detailed proof. ■

Remark 2: *The above theorem guarantees that by choosing* ϵ *small enough, the average-per-stage cost can be made arbitrarily close to the optimal one.*

Figure 2 compares the number of branches for the optimal and suboptimal algorithm with $\epsilon = 0.1$. The suboptimal algorithm significantly reduced the complexity of the problem requiring only 16 branches compared with 124 for the optimal algorithm. Another example of the ϵ-approximate solution is illustrated for the system defined by Eqns. (1), (2) with

$$A = \begin{bmatrix} -0.6 & 0.8 & 0.5 \\ -0.1 & 1.5 & -1.1 \\ 1.1 & 0.4 & -0.2 \end{bmatrix}, \qquad \Sigma_w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

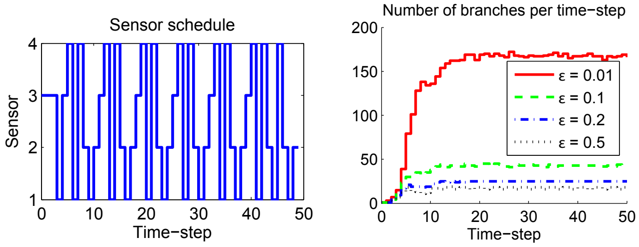
$$C_1 = \begin{bmatrix} 0.75 & -0.2 & -0.65 \end{bmatrix}, \qquad \Sigma_{v_1} = 0.53,$$

$$C_2 = \begin{bmatrix} 0.35 & 0.85 & 0.35 \end{bmatrix}, \qquad \Sigma_{v_2} = 0.8,$$

$$C_3 = \begin{bmatrix} 0.2 & -0.65 & 1.25 \end{bmatrix}, \qquad \Sigma_{v_3} = 0.2,$$

$$C_4 = \begin{bmatrix} 0.7 & 0.5 & 0.5 \end{bmatrix}, \qquad \Sigma_{v_4} = 0.5$$

and $q_k = 1$, $\forall k$. In this example, the horizon length is $N = 50$ which results in 10^{30} branches for the naive brute force search. Figure 4(a) shows the resulting sensor schedule for the suboptimal algorithm. This solution is the same for $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$ with objective function value of 850.57. It is interesting to note that the sensor schedule is periodic for the non-transient portion of the schedule, with a repeating sequence of $\{4, 1, 4, 2, 1, 2, 3\}$. 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 converge to $166, 43, 25$ and 18, respectively. Typically, the number of branches in the search tree converges to a small number for large ϵ.



(a) (b)

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

Table I compares the performance of different ϵ for 200 random cases in which there are $M = 3$ sensors and $n = 4$ states with a planning horizon of length $N = 50$. Each row represents the percentage of cases in which for that ϵ it obtained a strictly larger objective cost than another ϵ. For example, in 41.5% of the cases $Algo_S(5.0)$ performed worse than $Algo_S(0.1)$, and for 0.5% of the cases $Algo_S(0.1)$ performed worse than $Algo_S(0.5)$. In general, smaller ϵ will generate a better sensor schedule with a smaller objective function than larger ϵ, but in rare cases larger ϵ can obtain a better solution. Consider the following example, which refers to Figure 1, to explain this uncommon occurrence. Let the branch $\{1, 2, 1\}$ be the optimal solution in this problem. For $\epsilon = 0.1$, at time-step 2 the branch $\{2, 1\}$ will eliminate $\{1, 2\}$. However for $\epsilon = 1.0$, at time-step 1 the branch $\{1\}$ will eliminate $\{2\}$ since it has a larger approximation, and therefore $\{1, 2\}$ will not be eliminated at the next time-step. Consequently, the larger ϵ will yield a better solution than the smaller ϵ.

TABLE I

EACH ROW REPRESENTS THE PERCENTAGE OF CASES IN WHICH, FOR THE GIVEN ϵ, A STRICTLY LARGER OBJECTIVE COST WAS OBTAINED THAN FOR THE CORRESPONDING ϵ IN THE COLUMN.

ϵ	0.1	0.2	0.5	1.0	2.0	3.0	5.0
0.1	-	0.5	0.5	0.5	0.0	0.0	0.0
0.2	1.1	-	0.5	0.5	0.5	0.5	0.0
0.5	5.3	4.3	-	1.1	1.6	1.1	0.5
1.0	11.7	10.6	8.5	-	0.5	1.1	1.1
2.0	26.6	26.1	25.0	18.6	-	2.7	2.1
3.0	31.9	31.4	30.3	26.1	16.0	-	4.8
5.0	41.5	41.5	40.4	36.7	28.2	22.3	-

C. Performance

To characterize the performance of the suboptimal algorithm, 100 random instances were performed with $M = 3$ sensors, state dimension $n = 4$, $q_k = 1$, $\forall k \in \{1, \ldots, N\}$, and a horizon of length $N = 14$. 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 are used at once then the system is 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. For each problem, both the optimal solution and the suboptimal solutions over $\epsilon = \{0.1, 0.2, \ldots, 1.0\}$ were calculated. Figure 5(a) displays the percentage of the solutions that is optimal for each ϵ. As ϵ is increased there is a slow decay 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. As illustrated in the figure, the general trend for both the mean and maximum values is an exponential decay as ϵ increases.

V. HIERARCHICAL ALGORITHM

For larger ϵ, fewer matrices are needed to characterize the objective function and therefore the sensor schedule can be quickly computed. Another important aspect of the ϵ-approximate algorithm is that in general, the smaller the ϵ the closer the solution is to optimal since more branches are being explored. Therefore, it would be desirable to be able to combine the benefits from both the larger and smaller ϵ.

To this end, a hierarchical algorithm can be devised which uses the objective function, $\bar{V}$, acquired from the larger ϵ as an upper bound to prune branches when computing the solution with smaller ϵ. The upper bound can be used in two different ways to prune branches for the smaller ϵ. A branch can be pruned if the current value of the objective function is larger than the upper bound or if a lower bound on the minimal future objective cost along that branch is larger than the upper bound. A lower bound for the future

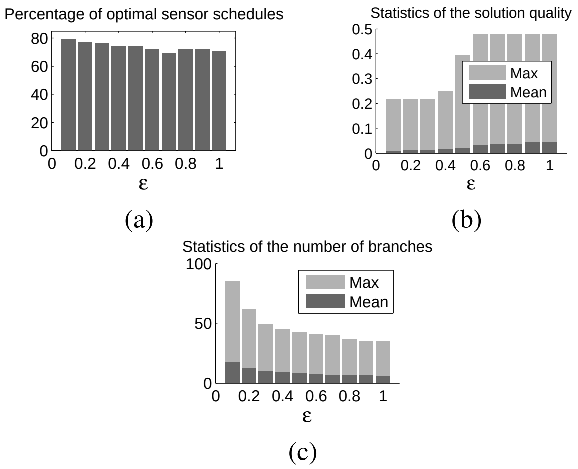


Fig. 5. Performance of the suboptimal algorithm for different ϵ. (a) The percentage of solutions for the suboptimal algorithm that is the optimal solution. (b) Mean and maximum relative error, in percentage, between the suboptimal and optimal solution for each ϵ. (c) The mean and maximum number of branches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

cost can be obtained through the use of *all* the sensors to determine an estimate of the state. This is indeed a lower bound because the Kalman filter is the minimum mean square estimator for a linear, Gaussian system and by removing sensor measurements the estimate cannot be improved.

Figure 6 shows the performance of the hierarchical method for 50 difficult[1] random cases with state dimension $n = 4$, $M = 3$ sensors, $q_k = 1 \ \forall k \in \{1, \dots, N\}$ and a horizon of $N = 50$. As before, the pair (A, C_i), $\forall i \in \mathbb{M}$, was restricted to be unobservable, but if all the sensors are used at once then the system is fully observable. For the hierarchical algorithm, an $\epsilon = 1.0$, which requires on average only 33 branches, is used as the bounding solution to prune the search tree for $\epsilon = 0.1$. The average number of branches required for the hierarchical method for $\epsilon = 0.1$ is 28 compared with 139 for the non-hierarchical method. Also note that the algorithm is only able to prune branches after time-step 7 because the lower bound on the objective function is not a good representation of the cost of the branch in the early stages of building the tree.

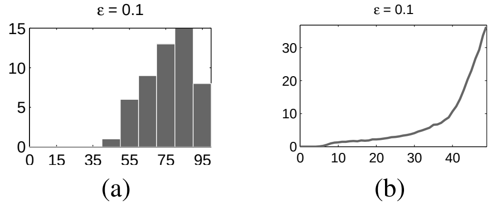


Fig. 6. Comparison of the performance of the hierarchical method. (a) Histogram showing the percentage of the original number of branches pruned by the hierarchical method for $\epsilon = 0.1$. (b) Plot of the average number of branches pruned at each time-step for $\epsilon = 0.1$.

VI. Conclusions

To solve the sensor scheduling problem, a condition on when an initial schedule is not part of the optimal was developed. 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.

There are several interesting areas of future work that the authors wish to explore. First, it has been 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. Second, the authors want to extend these methods to consider the case in which the sensors depend on the state of the system. Lastly, the authors wish to apply these algorithms to actual systems performing tasks such as multi-agent perception and environment discovery.

[1] Typically, if the larger ϵ needed a significant number of branches to represent the search tree, then the sensor scheduling problem was deemed difficult. A difficult case was required to have at least 15 branches at the final time-step for $\epsilon = 1.0$ to represent the search tree.

References

[1] L. Meier III, J. Peschon, and R. M. Dressler, "Optimal control of measurement subsystems," *IEEE Transactions on Automatic Control*, vol. 12, pp. 528–536, October 1967.

[2] 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.

[3] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *To appear, IEEE Transactions on Signal Processing*, 2008.

[4] 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*, 2004.

[5] 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*, 2004.

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

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

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

[9] 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*, June 2003.

[10] M. Branicky and G. Zhang, "Solving hybrid control problems: Level sets and behavioral programming," in *In Proc. American Contr. Conf.*, June 2000.

[11] 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.

[12] 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.

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

[14] B. Sinopoli, M. Schenato, M. Franceschetti, K. Poola, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1453–1464, September 2004.

[15] W. Zhang, J. Hu, and M. Vitus, "An error bound for the sensor scheduling problem," Tech. Rep. TR ECE 09-09, Electrical and Computer Engineering, Purdue University, 2009.