Effective Sensor Scheduling Schemes Employing Feedback in the Communication Loop

Ling Shi*, Michael Epstein*, Bruno Sinopoli† and Richard.M.Murray*

***Abstract*— In this paper, we consider a state estimation problem over a bandwidth limited network. A sensor network consisting of N sensors is used to observe the states of M plants, but only $p \leq N$ sensors can transmit their measurements to a centralized estimator at each time. Therefore a suitable scheme that schedules the proper sensors to access the network at each time so that the total estimation error is minimized is required. We propose four different sensor scheduling schemes. The static and stochastic schemes assume no feedback from the estimator to the scheduler, while the two dynamic schemes, Maximum Error First (MEF) and Maximum Deduction First (MDF) assume such feedback is available. We compare the four schemes via some examples and show MEF and MDF schemes are better than the static and stochastic schemes, which demonstrates that feedback can play an important role in this remote state estimation problem. We also show that MDF is better than MEF as MDF considers the total estimation error while MEF considers the individual estimation error.**

I. Introduction

Advances in fabrication technology and computer architecture have led to the rapid growth of computation capabilities while simultaneously decreasing chip size and power consumption. The latter gave birth to the fast developing field of sensor networks which have gained great attention in recent years [1], [2]. Many control applications now take advantage of the sensor networks and the loops are closed via the network [3]. This type of control system is called a networked control system (NCS). NCS provides many advantages which classical control system does not have, for example, reducing the system wiring, making the system easy to operate and maintain and increasing system agility. Despite the many advantages NCS has brought, finite bandwidth, network induced delays and possibly data packet drops severely degrade the system performance and may even cause system instability [4].

In the past decade, researchers have studied different networked control problems, mostly analyzing how the network in the closed loop affects the system performance and designing controllers that consider this effect to optimize the system performance. For example, in [5], the authors studied how the packet drops in the network affect the state estimation and they provided upper and lower bounds on the critical packet arrival rate below which the estimation error diverges. In [6], the authors studied the same estimation problem and gave similar results when only partial observation is received. Nilsson [7] studied how stochastic delays affect the control performance.

The effect of finite bandwidth of the network on the control performance was studied by Wong and Brockett in [8], [9] and further pursued by [10], [11] where the authors provided the minimum data rate that the network has to provide in order to have stable state estimate and closed loop stability.

When bandwidth is limited so that a single network is shared by many users, effective access control or network access scheduling schemes are required. Walsh and et.al studied the problem of when to schedule which plant to access the network so that all plants remain stable in [12] and [13]. They proposed a protocol MEF-TOD (Max Error First Try Once Discard) and showed that under certain conditions global exponential stability can be achieved. Tiwari and et.al studied the sensor scheduling problem in [14] where a single sensor has to determine which one of the two plants it needs to observe at each time step so as to minimize the estimation error. In [15], the authors considered a different scheduling problem where there is only one plant but with multiple sensors. They proposed a stochastic sensor scheduling scheme and provided the optimal probability distribution over the sensors to be selected.

In this paper, we study a networked state estimation problem. A sensor network consisting of N sensors is used to observe the states of M plants, but only $p \leq N$ sensors can transmit their measurements to the estimator at each time. The estimator is a Kalman filter. Since the estimator is usually attached to the controller which has enough power capability, feedback from the estimator to the network scheduler is then possible. We attempt to explore how much this feedback can improve the estimator performance. Four sensor scheduling schemes are proposed for this purpose. The static and stochastic schemes assume no feedback while the two dynamic schemes, Maximum Error First (MEF) and Maximum Deduction First (MDF) assume such feedback is available.

The idea of MEF is similar to the MEF-TOD protocol in [12], however we face a more complicated issue where there are two levels of scheduling. The first one is to schedule which p sensors to access the network. The second one is to schedule which plants those p sensors observe. We show that MEF and MDF are both better than the static and stochastic schemes, which suggests that feedback can play an important role for this estimation problem. We further show that MDF is better than MEF. It turns out MDF is even better than the

∗: Control and Dynamical Systems, California Institute of Technology; Pasadena, CA 91106. Email: {shiling, epstein, murray}@cds.caltech.edu
†: Electrical Engineering and Computer Science, University of California, Berkeley; Berkeley, CA 94720. Email: {sinopoli}@eecs.berkeley.edu
Tel: (626) 395-2313, Fax: (626) 796-8914.
Work supported in part by AFOSR grant FA9550-04-1-0169

locally optimal solution using a combinatorial approach as shown in Section VI.

We also attempt to include the plug-and-play feature so that those schemes will be suitable for general ad hoc sensor networks where the sensors available to use are in general not fixed. Ad hoc networks have attracted much attention in recent years due to their flexibility and easy operation [16], [17], [18]. Developing efficient algorithms and methods that are suitable for ad hoc networks is becoming increasingly important. We show that the MEF and MDF schemes are well suitable for such a situation.

The rest of the paper is organized as follows. In Section II, the mathematical model of the problem is given. In Section III through V we propose four schemes for scheduling the access of the sensors and compare their performances in Section VI. Conclusions and future work are given at the end.

II. Problem Set Up

Consider a sensor network consisting of N sensors which can observe the states of M different plants (Figure 1).

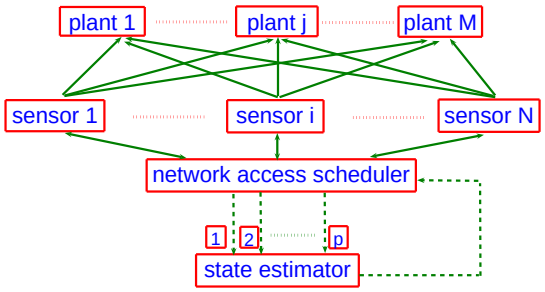

Fig. 1. System block diagram

Plant j has the following dynamics

$$x_{k+1}^j = A^j x_k^j + w_k^j, \tag{1}$$

where $j = 1, \cdots, M$. When sensor i observes plant j, it returns

$$y_k^{ij} = C^{ij} x_k^j + v_k^{ij}, \tag{2}$$

where $i = 1, \cdots, N$.

In Equ (1) and (2), $x_k^j \in \mathbb{R}^{n_j}$ is the state vector, $y_k^{ij} \in \mathbb{R}^{m_{ij}}$ is the observation vector, w_k^j and v_k^{ij} are process and measurement noises which are assumed to be white, Gaussian and zero mean with covariance matrices $Q^j \geq 0$ and $R^{ij} > 0$ respectively. The sensors send their measurements to a state estimator via a bandwidth limited network so that only $p \leq N$ sensors are allowed to access the network at each time step.

From now on denote the sensor space as $\mathbf{S} = \{\mathbf{S}_1, \cdots, \mathbf{S}_N\}$ and the plant space as $\mathbf{P} = \{\mathbf{P}_1, \cdots, \mathbf{P}_M\}$, where $\mathbf{S}_i$ and $\mathbf{P}_j$ are individual sensor and plant. Further denote $\mathbf{\Sigma}_k^j$ as the set of sensors that all observe $\mathbf{P}_j$ at time k and denote $\mathcal{Y}_k^j$, $\mathcal{C}_k^j$ and $\mathcal{R}_k^j$ as the single representation of all sensors in $\mathbf{\Sigma}_k^j$. For example, if $\mathbf{\Sigma}_k^j = \{\mathbf{S}_1, \cdots, \mathbf{S}_p\}$, then $\mathcal{Y}_k^j = [y_k^{1j}; \cdots; y_k^{pj}]$, $\mathcal{C}_k^j = [C^{1j}; \cdots; C^{pj}]$ and $\mathcal{R}_k^j$=**diag**$(R^{1j}, \cdots, R^{pj})$.

The state estimator is a Kalman filter. Denote $\hat{x}_k^j$ as the estimated state of $\mathbf{P}_j$ at time k given all previous measurements. Also denote P_k^j as the a priori estimation error covariance (simply write as error covariance later). Then $\hat{x}_k^j$ and P_k^j evolve as

$$\hat{x}_{k+1}^j = A^j \hat{x}_k^j + K_k^j(\mathcal{Y}_k^j - \mathcal{C}_k^j \hat{x}_k^j) \tag{3}$$

$$K_k^j = A^j P_k^j (\mathcal{C}_k^j)^T [\mathcal{C}_k^j P_k^j (\mathcal{C}_k^j)^T + \mathcal{R}_k^j]^{-1} \tag{4}$$

$$\begin{aligned} P_{k+1}^j &= A^j P_k^j (A^j)^T + Q^j \\ &\quad - A^j P_k^j (\mathcal{C}_k^j)^T [\mathcal{C}_k^j P_k^j (\mathcal{C}_k^j)^T + \mathcal{R}_k^j]^{-1} \mathcal{C}_k^j P_k^j (A^j)^T. \end{aligned} \tag{5}$$

Notice that if there is no sensor observing $\mathbf{P}_j$ at time k, $\mathbf{\Sigma}_k^j$ is a null set and therefore $\mathcal{C}_k^j = 0$ and $\mathcal{R}_k^j = 0$. As a result, Equ (3)-(5) reduce to

$$\hat{x}_{k+1}^j = A^j \hat{x}_k^j, \tag{6}$$

$$P_{k+1}^j = A^j P_k^j (A^j)^T + Q^j. \tag{7}$$

The problem we attempt to solve is to come up with a sensor scheduling scheme so that

$$\sum_{j=1}^{M} \mathrm{Tr}(P_k^j)$$

is minimized at each k. Notice that the minimization is only taken at the current time step while many previous works in literature try to minimize the steady state estimation error. The reason we choose this cost function is that in general ad hoc networks, N is a varying number as existing sensors can quit due to the power drainage or malfunctioning and new sensors can join. Consequently the sensors available at one time maybe quite different than at another time step. It therefore does not make sense to consider the long term behavior of the estimation error. This can be thought as a best effort minimization problem.

In the following sections, we present four scheduling schemes and compare their performances through some examples.

III. Static Scheduling

Static sensor scheduling schemes are the simplest among all schemes which only require little computation and are very easy to implement, for example, in a network using token rings or polling. Most static scheduling schemes are not suitable for general ad hoc networks as they assume a fixed set of sensors. There are many static schemes and we present one possible scheme in Algorithm I.

Algorithm I is periodic and it takes $\mathrm{LCM}(M, N)$ cycles to repeat, where $\mathrm{LCM}(M, N)$ denotes the least common multiple of M and N. Apparently, Algorithm I guarantees

TABLE I

ALGORITHM I: STATIC SCHEME

1) $k = 1, i = 1$.

2) $\mathbf{S}_{i \mod N}, \cdots, \mathbf{S}_{i+p-1 \mod N}$ are selected to access the network.

3) For $j = 0, \cdots, p-1$, $\mathbf{S}_{i+j \mod N}$ observes $\mathbf{P}_{i+j \mod M}$.

4) $k = k+1, i = i+p$. Go to Step 2.

TABLE II

ALGORITHM II: STOCHASTIC SCHEME

1) $k = 1$.

2) Select p sensors out of N sensors according to some distribution π.

3) If $\mathbf{S}_i$ is selected, it observes $\mathbf{P}_j$ with probability χ_{ij}.

4) $k = k+1$. Go to Step 2.

fair use of the sensors for each plant. However in many situations, the plants may have different dynamics and some are even unstable. It is then natural to schedule more sensors to observe the more unstable plants and hence help to control the unstable process. This is also reflected through the cost $\sum_{i=1}^{M} \mathrm{Tr}(P_k^i)$ that we try to minimize. Algorithm I has not considered this term at all. Most good scheduling schemes are very unfair, *i.e.*, allocating more resources (sensors in this case) to the plant who has the highest priority according to some cost functions. The dynamic scheduling schemes in Section V provide such examples.

Assume the sensors available are fixed over time. Then given the explicit parameters of the plants and sensors, we can provide conditions for the convergence of the upper bound of $\sum_{j=1}^{M} \mathrm{Tr}(P_k^j)$ when using Algorithm I. For simplicity, we consider a special case with $M = 2, N = 3$ and $p = 1$. The result is readily extendable to other cases. Let us define the following functions for any $X \geq 0$.

$$\begin{aligned} g_{ij}(X) &= A^j X (A^j)^T + Q^j \\ &\quad - A^j X (C^{ij})^T [C^{ij} X (C^{ij})^T + R^{ij}]^{-1} C^{ij} X (A^j)^T \\ h_j(X) &= A^j X (A^j)^T + Q^j, \end{aligned} \tag{8}$$

where $i = 1, 2, 3$ and $j = 1, 2$. Then for $k = 6z, z = 0, 1, \cdots$, the error covariances for the two plants evolve as

$$\begin{aligned} P_{k+6}^1 &= h_1 g_{21} h_1 g_{31} h_1 g_{11}(P_k^1), \\ P_{k+6}^2 &= g_{32} h_2 g_{12} h_2 g_{22} h_2 (P_k^1). \end{aligned}$$

Lemma 1: Let $P_\infty^j, j = 1, 2$ satisfy

$$P_\infty^1 = h_1 g_{21} h_1 g_{31} h_1 g_{11}(P_\infty^1), \tag{9}$$

$$P_\infty^2 = g_{32} h_2 g_{12} h_2 g_{22} h_2 (P_\infty^2). \tag{10}$$

Then $\sum_{j=1}^{2} \mathrm{Tr}(P_k^j)$ is bounded at each time step if and only if Equ (9) and (10) have bounded solutions.

Proof: We omit the proof as it is straightforward to show. ■

IV. STOCHASTIC SCHEDULING

The stochastic scheduling scheme is similar to the static scheme except that the sensor selection process is random rather than determined a priori. Both schemes assume no feedback from the estimator to the scheduler. The details of the stochastic scheme are presented Algorithm II.

In Algorithm II, π is the distribution over the subsets of $\mathbf{S}$ which consists of p sensors exactly. $\sum_{j=1}^{M} \chi_{ij} = 1$ for each $i = 1, \cdots, N$. Algorithm II also assumes the available sensors are fixed, therefore we are able to give conditions on the convergence of the upper bound of $E[\sum_{j=1}^{2} \mathrm{Tr}(P_k^j)]$ which is similar to that in [15]. Due to the limited space here, we omit writing out the convergence conditions. The advantage of this algorithm is that by properly choosing the distributions π and χ according to the plant and sensor parameters, the upper bound of $E[\sum_{j=1}^{2} \mathrm{Tr}(P_k^j)]$ can be minimized. The disadvantage is that it has not considered $\sum_{i=1}^{M} \mathrm{Tr}(P_k^i)$ either. As we show in the next two sections, by allowing feedback from the estimator to the scheduler and through minimizing $\sum_{i=1}^{M} \mathrm{Tr}(P_k^i)$, the estimator performance is greatly enhanced.

V. DYNAMIC SCHEDULING

In this section, we propose two dynamic sensor scheduling schemes which use feedback from the estimator to the scheduler. In essence we attempt to solve the following problem.

Problem 2 (Dynamic Sensor Scheduling):
Assume the estimator has enough power to broadcast its information to the network scheduler. Use this feedback to determine at each time which p sensors access the network and which plants those sensors observe so as to minimize $\sum_{i=1}^{M} \mathrm{Tr}(P_k^i)$.

We first present a locally optimal solution using a combinatorial approach and show that this locally optimal solution is intractable due to its high computational complexity. By locally optimal, we mean the minimization is taken at each time step rather than over a time horizon. After that we propose two locally suboptimal but tractable solutions. It turns out from the examples that one of the locally suboptimal solution is even better than the locally optimal solution in the long run. This shows that local optimality does not necessarily imply global optimality.

A. Locally Optimal Solution

There are $\mathbf{C}_N^p = \frac{N!}{p!(N-p)!}$ ways to select p out of N sensors, and each selected sensor can observe any one of the M plants. As a result, in total there are $\mathbf{C}_N^p M^p$ ways to determine which p sensors access the network and their associated plants to observe. Denoting ω as one such way and Ω as the space consisting of all such ways, then Ω has

TABLE III

ALGORITHM III: DYNAMIC SCHEDULING - MAX ERROR FIRST

1) $k = 1$.

2) $t = 1$. $\mathbf{S}$ has $N - t + 1$ sensors. $\Sigma_k^j = \emptyset, j = 1, \cdots, M$.

3) • For each $\mathbf{P}_j, j = 1, \cdots, M$
- Compute the open loop errors according to Equ (7).
• Store the trace of those errors in a buffer $\mathcal{B}$.
• Sort $\mathcal{B}$ in descending order.

4) • Let $\mathbf{P}_t$ be the plant having the first value in $\mathcal{B}$.
• For each $\mathbf{S}_i$ in $\mathbf{S}$, $i = 1, \cdots, N - t + 1$,
- Compute E_{it} according to Equ (11) or (12).
• Let $\mathbf{S}_{q(t)}$ be the one that minimizes the trace of E_{it}
• Replace the first value in $\mathcal{B}$ by the trace of $E_{q(t)t}$.

5) • Sort $\mathcal{B}$.
• Record $C^{q(t)t}$ into an array $\mathcal{A}$.
• Move $\mathbf{S}_{q(t)}$ from $\mathbf{S}$ to Σ_k^j.

6) $t = t + 1$. If $t \neq p + 1$, goto Step 4.

7) $k = k + 1$. Goto Step 2.

TABLE IV

ALGORITHM IV: DYNAMIC SCHEDULING - MAX DEDUCTION FIRST

1) $k = 1$.

2) $t = 1$. $\mathbf{S}$ has $N - t + 1$ sensors. $\Sigma_k^j = \emptyset, j = 1, \cdots, M$.

3) For each $\mathbf{P}_j, j = 1, \cdots, M$
- Compute Δ_k^{ij} according to Equ (13) or (14) for each $\mathbf{S}_i$ from $\mathbf{S}$, $i = 1, \cdots, N - t + 1$.

4) For each $\mathbf{P}_j, j = 1, \cdots, M$
- Store the trace of Δ_k^{ij} in a buffer $\mathcal{B}_j$, $i = 1, \cdots, N - t + 1$.
- Sort $\mathcal{B}_j$ in descending order.
- Store $\mathcal{B}_j[1]$ in $\mathcal{D}$.

5) • Sort $\mathcal{D}$. Let $\mathbf{P}_t$ have the first value in $\mathcal{D}$.
• Let $\mathbf{S}_{q(t)}$ returns the first value in $\mathcal{B}_t$.
• Record $C^{q(t)t}$ into an array $\mathcal{A}$.
• Move $\mathbf{S}_{q(t)}$ from $\mathbf{S}$ to Σ_k^j.

6) $t = t + 1$. If $t \neq p + 1$, goto Step 3.

7) $k = k + 1$. Goto Step 2.

cardinality $\mathbf{C}_N^p M^p$. Problem 2 can be cast as

$$\min_{\omega \in \Omega} \sum_{i=1}^{M} \mathrm{Tr}(P_k^i).$$

Clearly as N becomes large, the above minimization problem is intractable as it takes $O(\mathbf{C}_N^p M^{p+1})$ times to obtain. Therefore we seek tractable solutions but possibly at the price of losing local optimality. We propose two locally suboptimal solutions to Problem 2 in the next two sections. The advantages of these solutions are that they only take polynomial time in M, N, p and are suitable for general ad hoc networks where N might be varying.

B. Dynamic Scheduling: Max Error First

We propose in this section a locally suboptimal solution to Problem 2 and call it Dynamic Scheduling with Maximum Error First (MEF). The idea is that the plant having the largest open loop error has the highest priority to use the sensors. Once a sensor is selected by this plant, it has access to the network. Recall that we have defined $\mathbf{\Sigma}_k^j$ as the set of sensors observing $\mathbf{P}_j$ at time k. Let $\mathbf{S}_i$ be one sensor from $\mathbf{S}$ which consists of all the available sensors that $\mathbf{P}_j$ can use. If $\mathbf{\Sigma}_k^j$ is empty, define

$$\begin{aligned} E_{ij} &= A^j P_k^j (A^j)^T + Q^j \\ &- A^j P_k^j (C^{ij})^T [C^{ij} P_k^j (C^{ij})^T + R^{ij}]^{-1} C^{ij} P_k^j (A^j)^T \end{aligned} \quad (11)$$

where C^{ij}, R^{ij} are parameters for $\mathbf{S}_i$. Otherwise add $\mathbf{S}_i$ to $\mathbf{\Sigma}_k^j$ and define

$$\begin{aligned} E_{ij} &= A^j P_k^j (A^j)^T + Q^j \\ &- A^j P_k^j (\mathcal{C}_k^j)^T [\mathcal{C}_k^j P_k^j (\mathcal{C}_k^j)^T + \mathcal{R}_k^j]^{-1} \mathcal{C}_k^j P_k^j (A^j)^T \end{aligned} \quad (12)$$

where $\mathcal{C}_k^j$ and $\mathcal{R}_k^j$ are defined in Section II. With these notations the MEF scheme is presented in Algorithm III.

Clearly at each time step the array $\mathcal{A}$ tells which p sensors will access the network and which plants they observe. Step 3 in Algorithm III takes $O(M \log M)$ times to sort M elements. Step 4 takes $O(N - t + 1)$ times. Before step 4, $\mathcal{B}$ is already sorted and step 4 only changes the first value in $\mathcal{B}$, hence it takes only $O(\log M)$ times to resort $\mathcal{B}$ in step 5. Since $t = 1, \cdots, p$, in total it takes $O(M \log M + pN + p \log M)$ times to execute the algorithm for each k.

With this algorithm, it is possible to schedule multiple sensors observing the same plant which may be very unstable compared with other plants. Therefore it is not a fair scheme to all the plants. As we have discussed in Section III, good scheduling schemes tend to be very unfair.

C. Dynamic Scheduling: Max Deduction First

We present another locally suboptimal solution here and call it Dynamic Scheduling with Maximum Deduction First (MDF). The difference between this and MEF scheme is the different priorities of plants using the sensors. Again let $\mathbf{S}_i$ be one sensor from $\mathbf{S}$ which consists of all the available sensors that $\mathbf{P}_j$ can use. If $\mathbf{\Sigma}_k^j$ is empty, define

$$\Delta_k^{ij} = A^j P_k^j (C^{ij})^T [C^{ij} P_k^j (C^{ij})^T + R^{ij}]^{-1} C^{ij} P_k^j (A^j)^T. \quad (13)$$

Otherwise add $\mathbf{S}_i$ to $\mathbf{\Sigma}_k^j$ and call this new set $\mathbf{\Sigma}_k^{j'}$ with new parameters $\mathcal{C}_k^{j'} = [\mathcal{C}_k^j; C^{ij}]$ and $\mathcal{R}_k^{j'} = \mathbf{diag}(\mathcal{R}_k^j, R^{ij})$. Define

$$\begin{aligned} \Delta_{k-}^{ij} &= A^j P_k^j (\mathcal{C}_k^j)^T [\mathcal{C}_k^j P_k^j (\mathcal{C}_k^j)^T + \mathcal{R}_k^j]^{-1} \mathcal{C}_k^j P_k^j (A^j)^T. \\ \Delta_{k+}^{ij} &= A^j P_k^j (\mathcal{C}_k^{j'})^T [\mathcal{C}_k^{j'} P_k^j (\mathcal{C}_k^{j'})^T + \mathcal{R}_k^{j'}]^{-1} \mathcal{C}_k^{j'} P_k^j (A^j)^T. \\ \Delta_k^{ij} &= \Delta_{k+}^{ij} - \Delta_{k-}^{ij}. \end{aligned} \quad (14)$$

With these notations, the MDF scheme is presented in Algorithm IV.

Similar to Algorithm III, array $\mathcal{A}$ gives a locally suboptimal solution to Problem 2. It is also easy to show that step 3 in Algorithm IV takes $O(MN)$ times. Step 4 takes $O(MN \log N)$ times because it needs to sort N elements M times. Step 5 takes $O(M \log M)$ times to sort M elements. All the other steps takes constant time. Since $t = 1, \cdots, p$, in total it takes $O(pMN + pMN \log N + pM \log M)$ times to execute the algorithm for each k. This algorithm is also possible to schedule multiple sensors for the same plant and is therefore not a fair scheme either.

Compare with the $O(\mathbf{C}_N^p M^{p+1})$ time complexity, these two schemes provide unbeatable computational advantage, though Algorithm IV is slightly worse than Algorithm III. As we see from the examples in the next section, Algorithm IV gives better performance than Algorithm III. The reason is that Algorithm IV gives the highest priority to the plant who reduces most of the total estimation error while Algorithm III gives the highest priority to the plant having the largest open loop error, *i.e.*, it only considers the individual estimation error.

These two schemes are also compatible with varying N, *i.e.*, in a sensor plug-and-play situation where existing sensors can quit and new sensors can join at anytime, as long as the network scheduler keeps track of all the existing and new sensors. This can be done using some group communication protocols, for example see [19].

However, unlike the static or stochastic scheduling schemes in the previous sections, where we can give convergence conditions on the upper bound on the error covariance (or in expected sense), it is quite difficult to give conditions for convergence for these two schemes. The reason is that the sensors to be chosen and the plants to be observed are dynamically changing at all times. There are no clear static or statistical patterns of the selected sensors and their associated plants that we can use. We leave this part as future work that needs to be further pursued. Intuitively, the two schemes make the best effort in minimizing the cost function and their actual performances will be better than the static or stochastic schemes as demonstrated through the examples in the next section.

VI. Examples

We compare the different schemes through some examples in this section.

Example 3: Here we consider a simple example which is taken from [15] with slight modification of the measurement noise covariances. In this case, $M = 1, N = 2, p = 1$. The plant and sensors parameters are given as follows.

$$A = \begin{bmatrix} 1 & 0 & 0.2 & 0 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.0004 & 0.0001 & 0.0040 & 0.0010 \\ 0.0001 & 0.0004 & 0.0010 & 0.0040 \\ 0.0040 & 0.0010 & 0.0400 & 0.0100 \\ 0.0010 & 0.0040 & 0.0100 & 0.0400 \end{bmatrix}$$

$$C^{11} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, C^{21} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R^{11} = \begin{bmatrix} 2.4 & 0 \\ 0 & 1.0 \end{bmatrix}, R^{21} = \begin{bmatrix} 1.8 & 0 \\ 0 & 0.1 \end{bmatrix}$$

We compare the four different schemes here and the results are shown in Figure 2. We also plot the selected sensors for the stochastic and dynamic scheduling schemes in Figure 3.

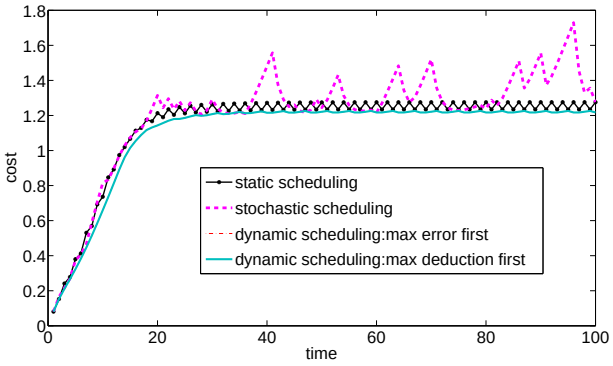

Fig. 2. Comparison of different scheduling schemes

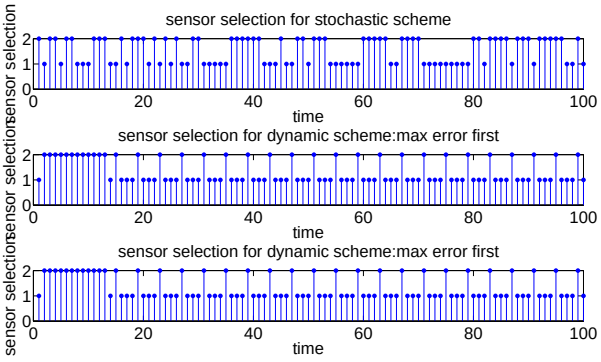

Fig. 3. Sensor selection based on stochastic or dynamic schemes

The two schemes from dynamic scheduling turn out to be the same for this case. This is because $M = 1$ and there is no other plant competing to use the sensors. It is also easy to see the local optimal solution using the combinatorial approach also overlaps with these two schemes and therefore the result is not shown in the figures. For this example, the static scheme is better than the stochastic scheme. In general

this may not be true as we choose an arbitrary distribution π and χ for this particular example. As shown in the next example where we optimize the distribution, *i.e.*, put more weight for those sensors having smaller noise covariances, the static scheme is worse than the stochastic scheme.

Example 4: We consider a more interesting example where six sensors are available to observe three plants, but only two sensors are allowed to access the network each time, that is $M = 3, N = 6$ and $p = 2$. To save space, we only list the parameters for plant two and sensor two respectively. The parameters of plant one and three are variations of those for plant two.

$$
\begin{aligned}
A^2 &= \begin{bmatrix} 1 & 0 & 0.15 & 0 \\ 0 & 1 & 0 & 0.15 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
Q^2 &= \begin{bmatrix} 0.0003 & 0.0000 & 0.0034 & 0.0006 \\ 0.0000 & 0.0003 & 0.0006 & 0.0034 \\ 0.0034 & 0.0006 & 0.0450 & 0.0079 \\ 0.0006 & 0.0034 & 0.0079 & 0.0450 \end{bmatrix} \\
C^{21} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, C^{22} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \\
C^{23} &= \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\
R^{21} &= \begin{bmatrix} 1.5 & 0 \\ 0 & 0.1 \end{bmatrix}, R^{22} = \begin{bmatrix} 2.0 & 0 \\ 0 & 0.8 \end{bmatrix} \\
R^{23} &= \begin{bmatrix} 3.0 & 0 \\ 0 & 0.1 \end{bmatrix}.
\end{aligned}
$$

We compare the four schemes and plot the results in Figure 4 and 6. Figure 5 and 7 are the zoomed in versions respectively for close comparison.

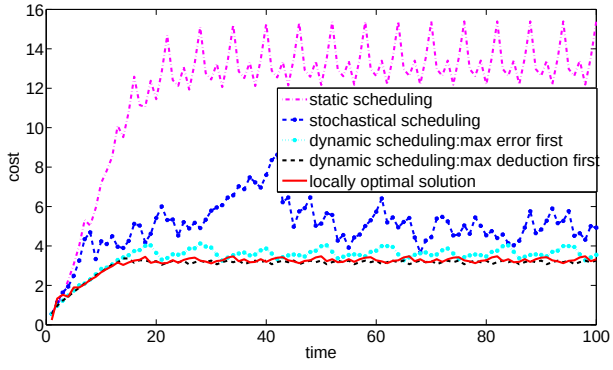

Fig. 4. Comparison of Different Scheduling Schemes

Clearly the two dynamic schemes are much better than the static and stochastic schemes. The MDF and MEF schemes are about 4 times better than the static scheme and 2 times better than the optimized stochastic scheme.

The MDF scheme is also observed to be better than the MEF scheme as we have discussed before. Since the number of M, N, p are not too big, for comparison purpose, we

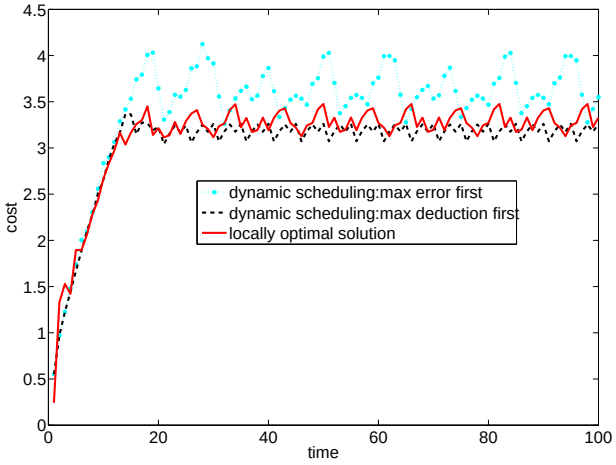

Fig. 5. Close Comparison

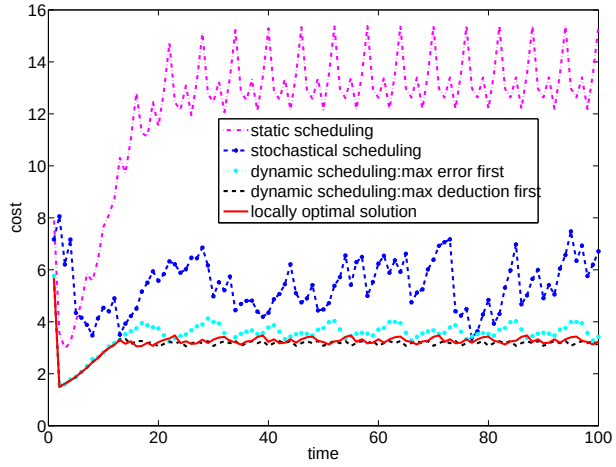

Fig. 6. Comparison of Different Scheduling Schemes

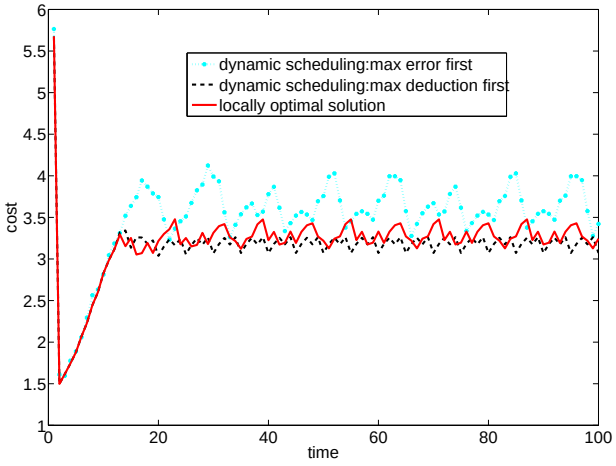

Fig. 7. Close Comparison

also plot the locally optimal solution in the figure. With little surprise, MDF is even better than the locally optimal solution for most of the time. Again as we have discussed before, local optimality does not necessarily imply global optimality. Therefore other local schemes, even though not locally optimal, might have better performances.

VII. Conclusions and Future Work

We have considered a remote centralized state estimation problem in this paper. Only a subset of the available sensors are able to send their measurement to an estimator due to the finite bandwidth of the network. In order to minimize the total estimation error at each time step, we have proposed four different sensor scheduling schemes and showed the dynamic schemes with feedback from the estimator to the scheduler outperform those without feedback.

The analysis of the dynamic schemes is quite difficult as there are no static or stochastic patterns of the selected sensors and their associated plants. Hence it is not straightforward to show the convergence of the total estimation error as we have done for the static and stochastic schemes. In the future work, we will look at this issue and pursue the convergence conditions for the two dynamic schemes.

Another interesting problem is to determine the minimum number p such that the total estimation error is bounded. It is clear that if p is sufficiently small and M, N are sufficiently large, the total estimation error quickly diverges. It will be of great interest to obtain a closed form relationship between these numbers and the plants and the sensors parameters, which shows the fundamental tradeoff between the accuracy of the estimation and the resources available.

Finally, it would be interesting and natural to close the loop via the network which is the dual of the estimation problem we have considered.

References

[1] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks." Proceedings of the 13th Mediterranean Conference on Control and Automation, 2005, pp. 719 – 724.

[2] M. Aboelaze and F. Aloul, "Current and future trends in sensor networks: a survey." Second IFIP International Conference on Wireless and Optical Communications Networks, March 2005, pp. 551 – 555.

[3] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S.S.Sastry, "Distributed control applications within sensor networks," vol. 91. Proceedings of the IEEE, Aug 2003, pp. 1235 – 1246.

[4] W. Zhang, "Stability analysis of networked control systems," Ph.D. dissertation, Case Western Reserve University, 2001.

[5] B.Sinopoli, L.Schenato, M.Franceschetti, K.Poolla, M.Jordan, and S.Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[6] X. Liu and A. Goldsmith, "Kalman filtering with partial observation losses." IEEE Control and Decision, 2004.

[7] J.Nilsson, "Real time control systems with delays," Ph.D. dissertation, Lund Institute of Technology, Lund, Sweden, 1998.

[8] W.S.Wong and R.W.Brockett, "Systems with finite communication bandwidth-part i: State estimation problems," *IEEE Trans. Automat. Contr.*, vol. 42, Sept 1997.

[9] ——, "Systems with finite communication bandwidth-part ii: Stabilization with limited information feedback," *IEEE Trans. Automat. Contr.*, vol. 44, May 1999.

[10] S. C. Tatikonda, "Control under communication constraints," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[11] A. Sahai, "Anytime information theory," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[12] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Transactions on Control Systems Magazine*, vol. 21, pp. 57 – 65, Feb 2001.

[13] G. C. Walsh, H. Ye, and L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 438 – 446, May 2002.

[14] A. Tiwari, M. Jun, D. E. Jeffcoat, and R. M. Murray, "Analysis of dynamic sensor coverage problem using kalman filters for estimation." Proceedings of the 16th IFAC World Congress, 2005.

[15] V. Gupta, T. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and dynamic sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[16] I. Chlamtac, M. Conti, and J. Liu, "Mobile ad hoc networking: Imperatives and challenges," vol. 1, no. 1. Ad Hoc Networks, July 2003.

[17] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," vol. 43. IEEE Transactions on Communications Magazine, March 2005, pp. 123 – 131.

[18] C.-Y. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," vol. 91. Proceedings of the IEEE, Aug 2003, pp. 1247 – 1256.

[19] I. Beier and H. Konig, "A protocol supporting distributed group and qos management." IEEE Conference on Protocols for Multimedia Systems - Multimedia Networking, Nov 1997, pp. 213 – 222.