Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Fast and Optimal Sensor Scheduling for Networked Sensor Systems

Shogo Arai, Yasushi Iwatani, and Koichi Hashimoto

Abstract— **This paper addresses a sensor scheduling problem for a class of networked sensor systems whose sensors are spatially distributed and measurements are influenced by state dependent noise. Sensor scheduling is required to achieve power saving since each sensor operates with a battery power source. The scheduling problem is formulated as a model predictive control problem with single sensor measurement per time. It is assumed that all sensors have state dependent noise and have the same characteristics, which follows from the properties of networked sensor systems. We propose a fast and optimal sensor scheduling algorithm for a class of networked sensor systems. Computation time of the proposed algorithm is proportional to the number of sensors and does not depend on the prediction horizon. In addition, we provide a fast sensor scheduling algorithm for a general class of systems by using a linear approximation of the sensor model.**

I. Introduction

A networked sensor system is a collection of spatially distributed sensors that are networked. Applications of networked sensor systems include habitat monitoring, animal tracking, forest-fire detection, precision farming, and disaster relief applications [11], [13]. In recent years, networked sensor systems have been implemented in control systems such as robot control systems [2], [8] and target tracking systems [14]. Sensors in a networked sensor system are usually connected wirelessly, and each sensor operates with a battery power source. It is therefore desired for each sensor to prolong the battery life, or equivalently, to achieve power saving [7]. For power saving, the wireless communication of sensors should be restricted, since it takes much power. One of approaches to solve the problem is to select available sensors dynamically. This process is called sensor scheduling.

One of major problems on sensor scheduling is to reduce computation time, since the number of possible sensor sequences increases exponentially with the number of the sensors. In particular, a predictive control method [3], a branch and bound method [5], and a sub-optimal method based on relaxed dynamic programming [1] have been proposed for sensor scheduling. In addition, a sensor scheduling strategy for continuous-time systems has been provided in [9]. These approaches assume that sensors have different characteristics, that is, each sensor observes a different measurement or the covariance matrices for the sensor model are different from each other.

Unfortunately, the existing approaches can not be applied to sensor scheduling for networked sensor systems for the following two reasons. First, a networked sensor system usually consists of a few types of sensors [13]. In other words, many sensors in a sensor network have the same characteristics, while the existing approaches assume that sensors have different characteristics as mentioned before. Thus the existing approaches can not provide a reasonable solution for the sensor scheduling problems for networked sensor systems. Second, sensors in a networked sensor system are spatially distributed. Each measurement noise may depend on the position of a target relative to the position of the sensor. In particular, measurements taken by cameras or radar sensors are influenced by state dependent noise [4], [10], [12]. The existing works do not provide any optimal sensor scheduling algorithm for systems with state dependent noise.

This paper addresses a sensor scheduling problem for a class of systems whose measurements are influenced by state dependent noise. It is assumed that all the sensors have the same characteristic, which follows from the properties of networked sensor systems. The sensor scheduling problem is formulated as a model predictive control problem with single sensor measurement per time. A fast and optimal sensor scheduling algorithm which minimizes a quadratic cost function at each time is proposed. The proposed algorithm is optimal for a class of networked sensor systems. The computation time of the proposed algorithm is proportional to the number of the sensors and does not depend on the prediction horizon. In addition, we provide a fast sensor scheduling algorithm for a general class of systems by using a linear approximation of the sensor model.

We use the following notation. For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\mathrm{cs}(\boldsymbol{A})$ stands for

$$\begin{bmatrix} A_{11} & A_{21} & \cdots & A_{m1} & A_{12} & A_{22} & \cdots & A_{mn} \end{bmatrix}^\top,$$

where $A_{j\ell}$ is the (j,ℓ)-th element of $\boldsymbol{A}$. For matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A}\otimes\boldsymbol{B}$ means their Kronecker product. The Kronecker delta is denoted by $\delta_{\ell m}$. The expectation operator is denoted by $\mathrm{E}[\cdot]$.

II. Sensor scheduling problem

A. System description

This paper considers a class of networked sensor systems as illustrated in Fig. 1. The system has N sensors, and the sensors are labeled from 1 to N. It is assumed that only one sensor is available at each time to achieve power saving.

Let us now describe details of the system model.

The controlled object is represented as a discrete-time linear time-invariant system

$$\boldsymbol{x}_p(k+1) = \boldsymbol{A}_p\boldsymbol{x}_p(k) + \boldsymbol{B}_p\boldsymbol{u}(k) + \boldsymbol{w}(k), \qquad (1)$$

S. Arai, Y. Iwatani, and K. Hashimoto are with the Department of System Information Sciences, Tohoku University, Aramaki Aza Aoba 6-6-01 Sendai Japan (e-mail: arai@ic.is.tohoku.ac.jp; iwatani@ic.is.tohoku.ac.jp; koichi@ic.is.tohoku.ac.jp).

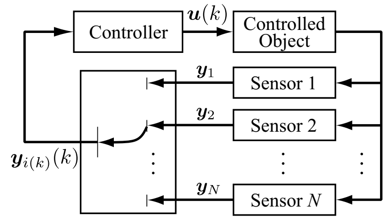

Fig. 1. A block diagram of a networked sensor system.

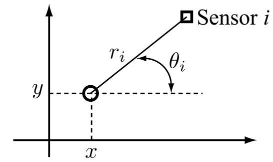

Fig. 2. Definitions of θ_i and r_i.

where $\boldsymbol{x}_p(k) \in \mathbb{R}^{n_p}$ is the state vector, $\boldsymbol{u}(k) \in \mathbb{R}^r$ the control input, and $\boldsymbol{w}(k)$ the process noise. The noise $\boldsymbol{w}(k)$ is white, Gaussian and zero mean with a covariance matrix $\boldsymbol{W}$. The time index k is sometimes omitted to simplify notation. The initial state $\boldsymbol{x}_p(0)$ is a random variable whose expectation value and covariance matrix are known constants.

The controller is given by

$$\boldsymbol{x}_c(k+1) = \boldsymbol{A}_c\boldsymbol{x}_c(k) + \boldsymbol{B}_{1c}\boldsymbol{y}_{i(k)}(k) + \boldsymbol{B}_{2c}\boldsymbol{u}(k), \quad (2)$$

$$\boldsymbol{u}(k) = \boldsymbol{C}_c\boldsymbol{x}_c(k) + \boldsymbol{D}_c\boldsymbol{y}_{i(k)}(k), \quad (3)$$

where $\boldsymbol{x}_c(k) \in \mathbb{R}^{n_c}$ is the state of the controller, $\boldsymbol{y}_{i(k)}(k) \in \mathbb{R}^p$ the measurement taken by sensor $i(k)$, and $i(k)$ the label of the sensor selected at time k. Note that the goal of this paper is to develop a fast and optimal sensor scheduling algorithm for a given controller, not to design the controller.

The sensor model is of the form:

$$\boldsymbol{y}_i(k) = \boldsymbol{C}\boldsymbol{x}_p(k) + \sum_{\ell=1}^{q} \boldsymbol{d}_{i\ell}(\boldsymbol{x}(k))v_{i\ell}(k), \quad (4)$$

where $\boldsymbol{x}$ is defined by

$$\boldsymbol{x}(k) = \begin{bmatrix}\boldsymbol{x}_p^\top(k) & \boldsymbol{x}_c^\top(k)\end{bmatrix}^\top, \quad (5)$$

and $\boldsymbol{v}_i(k) = [v_{i1}(k), \ldots, v_{iq}(k)]^\top$ is white, Gaussian and zero mean with a covariance matrix $\boldsymbol{V}$. It is assumed that $\boldsymbol{v}_i(k)$, $\boldsymbol{w}(k)$ and $\boldsymbol{x}_p(0)$ are mutually independent. Clearly $\boldsymbol{d}_{i\ell}(\boldsymbol{x}(k))$ is independent of $\boldsymbol{v}_i(k)$. The matrices $\boldsymbol{C}$ and $\boldsymbol{V}$ are independent of sensor selection, which implies that all the sensors have the same characteristics. Note that a state-dependent sensor scheduling algorithm is required, since $\boldsymbol{y}_i(k)$ is influenced by state dependent noise. The matrix function $\boldsymbol{d}_{i\ell}(\boldsymbol{x})$ is a function of $\boldsymbol{x}$ not of only $\boldsymbol{x}_p$. This helps to develop a camera model as shown in Example 2.

We will show several sensors whose mathematical models can be written by (4).

Example 1: Consider radar sensors that measure the position of a target in the (x, y) plane. Standard models of radar sensors are given by

$$\boldsymbol{y}_i = \begin{bmatrix}x\\ y\end{bmatrix} + \begin{bmatrix}\cos\theta_i & -\sin\theta_i\\ \sin\theta_i & \cos\theta_i\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & r_i\end{bmatrix}\boldsymbol{v}_i \quad (6)$$

[12], or

$$\boldsymbol{y}_i = \begin{bmatrix}x\\ y\end{bmatrix} + a(r_i)\begin{bmatrix}\cos\theta_i & -\sin\theta_i\\ \sin\theta_i & \cos\theta_i\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & b\end{bmatrix}\boldsymbol{v}_i \quad (7)$$

[4], where $\boldsymbol{y}_i$ is the measurement, θ_i the angle between the x axis and the vector joining sensor i to the target, r_i the distance from sensor i to the target, $a(r_i)$ a function such that $a^2(r_i)$ is a quadratic function of r_i, b a constant, and $\boldsymbol{v}_i$ a white, Gaussian and zero mean noise (see Fig. 2). It is clear that (6) and (7) are described by (4). In addition, networked sensor systems consisting of radar sensors whose models are described by (6) or (7) satisfy the following two: (1) All the sensors have the same characteristics, (2) The measurement noise depends on the state of the system.

Example 2: Let a camera and a target be set at (p_x, p_y) and (x, y), respectively. The optical axis of the camera is directed parallel to the x axis. The target is projected onto the image plane at $\bar{Y} := Y + \bar{v}$ due to quantization errors or calibration errors [10], where

$$Y = \frac{f}{x - p_x}(y - p_y) \quad (8)$$

and $\bar{v}$ is the white and Gaussian noise (see the left of Fig. 3). The target position y is estimated from $\bar{Y}$ by

$$\bar{y} = p_y + \frac{x - p_x}{f}\bar{Y}, \quad (9)$$

or

$$\bar{y} = p_y + \frac{x_c - p_x}{f}\bar{Y}, \quad (10)$$

when (2) is used as an observer, where $\bar{y}$ is an estimate of y, f the focal length of the camera, and x_c an estimate of x from the observer. Equation (9) is not valid when x is unknown, but (10) is always available. Replacing x with x_c in (8) and substituting it into (10), we have the camera model of the form

$$\bar{y} = y + \frac{x_c - p_x}{f}\bar{v}. \quad (11)$$

We here use two cameras to obtain two dimensional information, since it is difficult to get the depth information in the camera system. Let cameras ℓ and m be set at

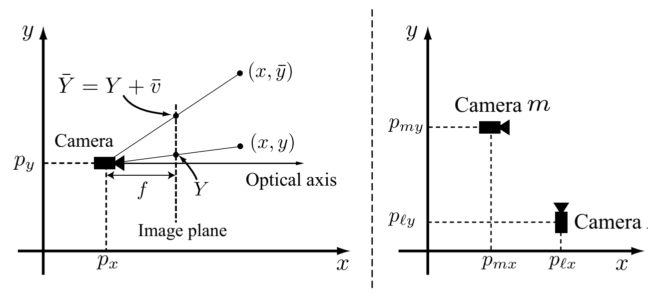

Fig. 3. Left: The pinhole camera model. Right: Camera location.

$(p_{\ell x},\ p_{\ell y})$ and $(p_{mx},\ p_{my})$, respectively, as illustrated in the right of Fig. 3. The optical axes of cameras ℓ and m are parallel to the y axis and the x axis, respectively. The combination of cameras ℓ and m is labeled by i. Then the camera model is represented by

$$\boldsymbol{y}_i = \begin{bmatrix} x \\ y \end{bmatrix} + \frac{1}{f} \begin{bmatrix} y_c - p_{\ell y} & 0 \\ 0 & x_c - p_{mx} \end{bmatrix} \boldsymbol{v}, \tag{12}$$

where (x_c, y_c) is an estimate of (x, y). Equation (12) is represented by (4), since $\boldsymbol{d}_{i\ell}$ is a function of $\boldsymbol{x}$.

Example 3: The model (4) includes a class of sensors with stochastic parametric uncertainties [6]. For example, a sensor model of the form

$$\boldsymbol{y}_i(k) = (\boldsymbol{C} + \Delta\boldsymbol{C}_i)\boldsymbol{x}_p(k) \tag{13}$$

is considered in [6], where

$$\Delta\boldsymbol{C}_i = \sum_{j=1}^{p} \bar{\boldsymbol{C}}_{ij} v_{ij}(k), \tag{14}$$

and v_{ij} is a white, Gaussian and zero mean noise with $\mathrm{E}[v_{ij}(k)v_{i\ell}(\tau)] = \delta_{j\ell}\delta_{k\tau}$. Equation (13) can be represented by (4).

B. Sensor scheduling problem

The closed loop system described by (1)–(4) is of the form:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\sum_{\ell=1}^{q} \boldsymbol{d}_{i\ell}(\boldsymbol{x}) v_{i\ell}(k) + \begin{bmatrix} \boldsymbol{w}(k) \\ \boldsymbol{0} \end{bmatrix}, \tag{15}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_p + \boldsymbol{B}_p\boldsymbol{D}_c\boldsymbol{C} & \boldsymbol{B}_p\boldsymbol{C}_c \\ \boldsymbol{B}_{1c}\boldsymbol{C} + \boldsymbol{B}_{2c}\boldsymbol{D}_c\boldsymbol{C} & \boldsymbol{A}_c + \boldsymbol{B}_{2c}\boldsymbol{C}_c \end{bmatrix}, \tag{16}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_p\boldsymbol{D}_c \\ \boldsymbol{B}_{1c} + \boldsymbol{B}_{2c}\boldsymbol{D}_c \end{bmatrix}. \tag{17}$$

We also define $n := n_p + n_c$ and $\boldsymbol{x}_0 := \mathrm{E}[\boldsymbol{x}(0)]$.

This paper considers the following problem.

Problem 1: Let a positive integer T and positive definite symmetric matrices $\boldsymbol{Q}_p \in \mathbb{R}^{n_p \times n_p}$, $\boldsymbol{R} \in \mathbb{R}^{r \times r}$ and $\boldsymbol{\Pi} \in \mathbb{R}^{n_p \times n_p}$ be given. A cost function is defined by

$$J = \mathrm{E}\Big[\sum_{k=0}^{T} \{\boldsymbol{x}_p^\top(k)\boldsymbol{Q}_p\boldsymbol{x}_p(k) + \boldsymbol{u}^\top(k)\boldsymbol{R}\boldsymbol{u}(k)\} + \boldsymbol{x}_p^\top(T+1)\boldsymbol{\Pi}\boldsymbol{x}_p(T+1)\Big]. \tag{18}$$

Find

$$\{i^*(0),\ \cdots,\ i^*(T)\} = \arg\min_{i(0),\cdots,i(T)} J(T) \tag{19}$$

for (15).

It is assumed in this paper that model predictive control is implemented. Problem 1 is solved at each time. Thus a fast algorithm for solving Problem 1 is desired. One of the most primitive methods to solve Problem 1 is as follows: Calculate values of the cost function for all possible sensor sequences from time 0 to T and compare these values. This is called the *exhaustive search method* in this paper. Clearly the number of comparisons is N^{T+1} for the exhaustive search method. Thus the exhaustive search method is not suitable for model predictive control from the point of view of computation time as will shown in Examples 4 and 5.

Note that $\mathrm{E}[\boldsymbol{x}(k)]$ is required to derive the optimal sensor scheduling at each time. Therefore we have to estimate $\mathrm{E}[\boldsymbol{x}_p(k)]$. We use (2) as an observer for estimation in numerical examples presented in this paper.

III. Fast and optimal sensor scheduling

This section proposes a fast and optimal sensor scheduling algorithm. It is assumed throughout this section that there exist constant matrices $\boldsymbol{S}_\ell \in \mathbb{R}^{p \times n}$ and $\boldsymbol{s}_{i\ell} \in \mathbb{R}^p$ such that

$$\boldsymbol{d}_{i\ell}(\boldsymbol{x})\boldsymbol{d}_{im}^\top(\boldsymbol{x}) = (\boldsymbol{S}_\ell\boldsymbol{x} + \boldsymbol{s}_{i\ell})(\boldsymbol{S}_m\boldsymbol{x} + \boldsymbol{s}_{im})^\top \tag{20}$$

for all $i \in \{1, 2, \cdots, N\}$ and $\ell, m \in \{1, 2, \cdots, q\}$. The assumption (20) implies the following two:

1) The variance of the measurement noise can be represented by a quadratic function of $\boldsymbol{x}$.
2) The coefficient matrices $\boldsymbol{S}_\ell$ in (20) are independent of the sensor selection.

The sensor models (12) and (13) satisfy (20). Note that (20) does not imply that

$$\boldsymbol{d}_{i\ell} = \boldsymbol{S}_\ell\boldsymbol{x} + \boldsymbol{s}_{i\ell}. \tag{21}$$

For example, the radar model (7) with $a \equiv 1$ and $b = 1$ satisfies (20) but not (21).

We first derive a time evolution equation for

$$\boldsymbol{X}(k) := \mathrm{E}[\boldsymbol{x}(k)\boldsymbol{x}^\top(k)]. \tag{22}$$

Lemma 1: It is assumed that (20) holds. When sensors $i(0),\ i(1),\ \cdots,\ i(k)$ are selected,

$$\begin{aligned} \boldsymbol{X}(k+1) =& \boldsymbol{A}\boldsymbol{X}(k)\boldsymbol{A}^\top \\ &+ \sum_{\ell=1}^{q}\sum_{m=1}^{q} V_{\ell m}\boldsymbol{B}\boldsymbol{S}_\ell\boldsymbol{X}(k)\boldsymbol{S}_m^\top\boldsymbol{B}^\top \\ &+ \boldsymbol{\Psi}_i(k, \boldsymbol{x}_0) \end{aligned} \tag{23}$$

holds, where $V_{\ell m}$ is the (ℓ, m)-th element of $\boldsymbol{V}$, and

$$\begin{aligned} \boldsymbol{\Psi}_i(k) =& \sum_{\ell=1}^{q}\sum_{m=1}^{q} V_{\ell m}\boldsymbol{B}(\boldsymbol{S}_\ell\boldsymbol{A}^k\boldsymbol{x}_0\boldsymbol{s}_{im}^\top \\ &+ \boldsymbol{s}_{i\ell}(\boldsymbol{S}_m\boldsymbol{A}^k\boldsymbol{x}_0)^\top + \boldsymbol{s}_{i\ell}\boldsymbol{s}_{im}^\top)\boldsymbol{B}^\top + \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \end{aligned}$$

Proof: The solution of (15) is given by

$$\boldsymbol{x}(k) = \boldsymbol{A}^k\boldsymbol{x}(0) + \sum_{m=0}^{k-1} \boldsymbol{A}^{k-1-m}\boldsymbol{f}_{i(m)}(m, \boldsymbol{x}),$$

where

$$\boldsymbol{f}_{i(m)}(m, \boldsymbol{x}) = \boldsymbol{B}\sum_{\ell=1}^{q} \boldsymbol{d}_{i\ell}(\boldsymbol{x}) v_{i\ell}(m) + \begin{bmatrix} \boldsymbol{w}(m) \\ \boldsymbol{0} \end{bmatrix}.$$

It is straightforward to verify that

$$\begin{aligned}\boldsymbol{X}(k) =& \mathrm{E}\Big[\boldsymbol{A}^k\boldsymbol{x}(0)\boldsymbol{x}^\top(0)(\boldsymbol{A}^k)^\top \\ &+\sum_{j=0}^{k-1}\boldsymbol{f}_i(j,\boldsymbol{x})\boldsymbol{x}^\top(0)(\boldsymbol{A}^k)^\top \\ &+\sum_{j=0}^{k-1}\boldsymbol{A}^k\boldsymbol{x}(0)\boldsymbol{f}_i^\top(j,\boldsymbol{x}) \\ &+\sum_{j=0}^{k-1}\sum_{m=0}^{k-1}\boldsymbol{f}_i(j,\boldsymbol{x})\boldsymbol{f}_i(m,\boldsymbol{x})^\top\Big]. \end{aligned} \tag{24}$$

The second and the third terms of (24) vanish since $\boldsymbol{x}(0)$, $\boldsymbol{w}(k)$ and $\boldsymbol{v}_i(k)$ are mutually independent. We obtain (23) from (20) and $\mathrm{E}[\boldsymbol{x}(k)] = \boldsymbol{A}^k\boldsymbol{x}_0$. ■

In Lemma 1, $\boldsymbol{\Psi}_i$ is affine in $\boldsymbol{x}_0$. An affine function of $\boldsymbol{x}_0$ for the cost function at each time is denoted by $\boldsymbol{\Phi}_i$ as derived in the following lemma.

Lemma 2: If (20) is satisfied, then

$$\begin{aligned}&\mathrm{E}\Big[\boldsymbol{x}_p^\top(k)\boldsymbol{Q}_p\boldsymbol{x}_p(k)+\boldsymbol{u}^\top(k)\boldsymbol{R}\boldsymbol{u}(k)\Big] \\ &\qquad = \mathrm{tr}[\boldsymbol{Q}\boldsymbol{X}(k)]+\mathrm{tr}[\boldsymbol{\Phi}_i(k)]\end{aligned} \tag{25}$$

holds, where

$$\begin{aligned}\boldsymbol{Q} =& \begin{bmatrix}\boldsymbol{Q}_p & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} + \begin{bmatrix}\boldsymbol{C}^\top\boldsymbol{D}_c^\top \\ \boldsymbol{C}_c^\top\end{bmatrix}\boldsymbol{R}\begin{bmatrix}\boldsymbol{D}_c\boldsymbol{C} & \boldsymbol{C}_c\end{bmatrix} \\ &+\sum_{\ell=1}^{q}\sum_{m=1}^{q}V_{\ell m}\boldsymbol{S}_m^\top\boldsymbol{D}_c^\top\boldsymbol{R}\boldsymbol{D}_c\boldsymbol{S}_\ell, \\ \boldsymbol{\Phi}_i(k) =& \boldsymbol{D}_c^\top\boldsymbol{R}\boldsymbol{D}_c\sum_{\ell=1}^{q}\sum_{m=1}^{q}V_{\ell m}\{\boldsymbol{S}_\ell\boldsymbol{A}^k\boldsymbol{x}_0\boldsymbol{s}_{im}^\top \\ &+\boldsymbol{s}_{i\ell}(\boldsymbol{S}_m\boldsymbol{A}^k\boldsymbol{x}_0)^\top+\boldsymbol{s}_{i\ell}\boldsymbol{s}_{im}^\top\}.\end{aligned} \tag{26}$$

Proof: Substituting (2), (3), and (4) into (18) yields (25), since $\boldsymbol{x}(k)$ and $\boldsymbol{v}_i(k)$ are mutually independent. ■

The following theorem plays an important role to derive a fast and optimal sensor scheduling algorithm.

Theorem 1: If (20) holds, $i^*(k)$ in (19) is obtained from

$$i^*(k) = \arg\min_{i(k)}\ \mathrm{tr}[\boldsymbol{P}(k)\boldsymbol{\Psi}_i(k)]+\mathrm{tr}[\boldsymbol{\Phi}_i(k)], \tag{27}$$

where

$$\boldsymbol{P}(T+1) = \begin{bmatrix}\boldsymbol{\Pi} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}, \tag{28}$$

$$\begin{aligned}\boldsymbol{P}(k) =& \boldsymbol{Q}+\boldsymbol{A}^\top\boldsymbol{P}(k+1)\boldsymbol{A} \\ &+\sum_{\ell=1}^{q}\sum_{m=1}^{q}V_{\ell m}\boldsymbol{S}_m^\top\boldsymbol{B}^\top\boldsymbol{P}(k+1)\boldsymbol{B}\boldsymbol{S}_\ell.\end{aligned} \tag{29}$$

Proof: We first define

$$\begin{aligned}\bar{J}(m) =& \mathrm{E}\Big[\sum_{k=m}^{T}\big(\boldsymbol{x}_p^\top(k)\boldsymbol{Q}_p\boldsymbol{x}_p(k)+\boldsymbol{u}^\top(k)\boldsymbol{R}\boldsymbol{u}(k)\big) \\ &+\boldsymbol{x}_p^\top(T+1)\boldsymbol{\Pi}\boldsymbol{x}_p(T+1)\Big]\end{aligned} \tag{30}$$

for $m \in \{0, 1, \cdots, T\}$. By using the principle of mathematical induction, we will prove that

$$\begin{aligned}\bar{J}(m) =& \mathrm{tr}[\boldsymbol{P}(m)\boldsymbol{X}(m)+\boldsymbol{P}(m)\boldsymbol{\Psi}_i(m)] \\ &+\mathrm{tr}[\boldsymbol{\Phi}_i(m)+h(i(m+1),\ldots,i(T))]\end{aligned} \tag{31}$$

holds, where $h(i(m+1),\ldots,i(T))$ is a function which is independent of sensor selection at time m.

It is straightforward from (23) and (25) to obtain

$$\bar{J}(T) = \mathrm{tr}[\boldsymbol{P}(T)\boldsymbol{X}(T)+\boldsymbol{P}(T)\boldsymbol{\Psi}_i(T)]+\mathrm{tr}[\boldsymbol{\Phi}_i(T)].$$

Therefore (31) holds for $m = T$.

If (31) holds for $m = \ell$, we have

$$\begin{aligned}\bar{J}(\ell-1) =& \mathrm{tr}[\boldsymbol{P}(\ell-1)\boldsymbol{X}(\ell-1) \\ &+\boldsymbol{P}(\ell-1)\boldsymbol{\Psi}_i(\ell-1)+\boldsymbol{P}(\ell)\boldsymbol{\Psi}_i(\ell)] \\ &+\mathrm{tr}[\boldsymbol{\Phi}_i(\ell-1)+\boldsymbol{\Phi}_i(\ell) \\ &+h(i(\ell+1),\ldots,i(T))].\end{aligned} \tag{32}$$

The third term of (32) does not depend on $i(\ell-1)$, since $\boldsymbol{P}(\ell)$ is independent of the sensor scheduling for all k. Hence (31) holds for $m = \ell-1$.

We have (27) form the principle of optimality. ■

We are now ready to provide a fast sensor scheduling algorithm by using Theorem 1. The optimal sensor selection $i^*(0)$ is computed at each time as follows. Note that $\boldsymbol{P}(0)$ is a solution of Riccati equation (29) with the initial value of (28), and it can be computed in advance, since (29) is independent of sensor selection. In (27), $\boldsymbol{\Psi}_i(0)$ and $\boldsymbol{\Phi}_i(0)$ are calculated when $\boldsymbol{x}_0$ is given. Consequently $i^*(0)$ is found by comparing possible N values. Computation time of deriving $i^*(0)$ is proportional to N and is independent of T, while the number of all possible sensor sequences from time 0 to T is N^{T+1}. Thus the proposed algorithm is fast even for large-scale networked sensor systems.

It is straightforward to obtain the following two corollaries of Theorem 1.

Corollary 1: If (20) is satisfied, there exist convex polyhedra $\mathbb{S}_1$, …, $\mathbb{S}_N$ which partition $\mathbb{R}^n$ such that

$$i^*(0), \quad \text{for } \boldsymbol{x}_0 \in \mathbb{S}_{i^*(0)}. \tag{33}$$

Corollary 2: Suppose that (20) holds, and all the eigenvalues of $\boldsymbol{A}$ and

$$\boldsymbol{H} = \boldsymbol{A}^\top\otimes\boldsymbol{A}^\top+\sum_{\ell=1}^{q}\sum_{m=1}^{q}V_{\ell m}(\boldsymbol{S}_m^\top\boldsymbol{B}^\top)\otimes(\boldsymbol{S}_\ell^\top\boldsymbol{B}^\top)$$

lie within the unit circle. Then there exists $\boldsymbol{P}_\infty$ such that

$$\begin{aligned}\boldsymbol{P}_\infty =& \boldsymbol{Q}+\boldsymbol{A}^\top\boldsymbol{P}_\infty\boldsymbol{A} \\ &+\sum_{\ell=1}^{q}\sum_{m=1}^{q}V_{\ell m}\boldsymbol{S}_m^\top\boldsymbol{B}^\top\boldsymbol{P}_\infty\boldsymbol{B}\boldsymbol{S}_\ell.\end{aligned} \tag{34}$$

Moreover, (27) holds for the infinite-time quadratic cost function of the form

$$J_\infty = \lim_{T\to\infty}\frac{1}{T}J(T), \tag{35}$$

when $\boldsymbol{P}(k)$ in (27) is replaced with $\boldsymbol{P}_\infty$.

Proof: The column expansion of (29) has the form:

$$\mathrm{cs}(\boldsymbol{P}(k)) = \boldsymbol{H}\mathrm{cs}(\boldsymbol{P}(k+1)) + \mathrm{cs}(\boldsymbol{Q}). \tag{36}$$

Since $\boldsymbol{H}$ is stable, there exists $\boldsymbol{P}_\infty$ satisfying (34).

It is straightforward to verify that

$$\mathrm{tr}[\boldsymbol{P}_\infty(\boldsymbol{X}(k+1) - \boldsymbol{X}(k))] = \lambda_i(k) - \hat{J}(k) \tag{37}$$

holds from (23), (25), and (34), where

$$\lambda_i(k) = \mathrm{tr}[\boldsymbol{P}_\infty \boldsymbol{\Psi}_i(k)] + \mathrm{tr}[\boldsymbol{\Phi}_i(k)].$$

Furthermore we have

$$J_\infty = \lim_{T\to\infty} \frac{\mathrm{tr}[\boldsymbol{P}_\infty(\boldsymbol{X}(0) - \boldsymbol{X}(T))]}{T} + \sum_{k=0}^{T} \frac{\lambda_i(k)}{T} \tag{38}$$

from (25) and (37). The first term of (38) converges to 0, since $\boldsymbol{A}$ and $\boldsymbol{H}$ are stable. There exists a positive number m such that $|\lambda_i(k)| < m$ satisfies for all $k \in [0, 1, \cdots)$ and $i \in \{1, 2, \cdots, N\}$. Thus the second term of (38) converge for an arbitrarily given sensor sequence. Hence the corollary can be proven from

$$\min_{i(0), i(1), \cdots} J_\infty = \lim_{T\to\infty} \min_{i(0), \cdots, i(T)} \sum_{k=0}^{T} \frac{\lambda_i(k)}{T}. \tag{39}$$

■

Example 4: Consider a vehicle that travels on the two-dimensional plane. The position of the vehicle is denoted by (x, y). The state equation of the vehicle in continuous time is given by

$$\dot{\bar{\boldsymbol{x}}}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{x}}_p + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \bar{\boldsymbol{u}} + \bar{\boldsymbol{w}}, \tag{40}$$

where $\bar{\boldsymbol{x}}_p := [x\ y\ \dot{x}\ \dot{y}]^\top \in \mathbb{R}^4$ and $\bar{\boldsymbol{u}} \in \mathbb{R}^2$. The covariance matrix of $\bar{\boldsymbol{w}}$ is set to $0.001\boldsymbol{I}$. The state equation (40) is discretized with a sampling period 0.1. An observer (2) and a controller (3) are implemented such that the poles of (1) are set to $0.91 \pm 0.055\mathrm{j}$, $0.92 \pm 0.030\mathrm{j}$, $0.86 \pm 0.091\mathrm{j}$ and $0.86 \pm 0.091\mathrm{j}$, where j denotes the imaginary unit.

Four cameras are set at $(0.3, -0.3)$, $(-0.3, 0.3)$, $(0.3, 1)$ and $(1, 0.3)$ as illustrated in Fig. 4. The optical axes of cameras 1 and 3 are parallel to the y axis, and 2 and 4 are parallel to the x axis. The labels 1, 2, 3 and 4 are attached to camera combinations of $(1, 2)$, $(2, 3)$, $(3, 4)$ and $(4, 1)$, respectively. The sensor model (12) is used for $i \in \{1, 2, \cdots, 4\}$, $\ell \in \{1, 3\}$ and $m \in \{2, 4\}$.

Parameters are set to

$$\boldsymbol{V} = 1.0 \times 10^{-8}\boldsymbol{I}, \quad f = 0.01,$$
$$\boldsymbol{Q}_p = \boldsymbol{I}, \quad \boldsymbol{R} = \boldsymbol{I}, \quad \boldsymbol{\Pi} = \boldsymbol{I}, \quad T = 4.$$

The left of Fig. 4 shows sample paths of the vehicle for

$$\boldsymbol{x}_p(0) = \begin{bmatrix} 0.61 & 0.61 & 0.01 & 0.01 \end{bmatrix}^\top,$$
$$\boldsymbol{x}_c(0) = \begin{bmatrix} 0.6 & 0.6 & 0 & 0 \end{bmatrix}^\top,$$

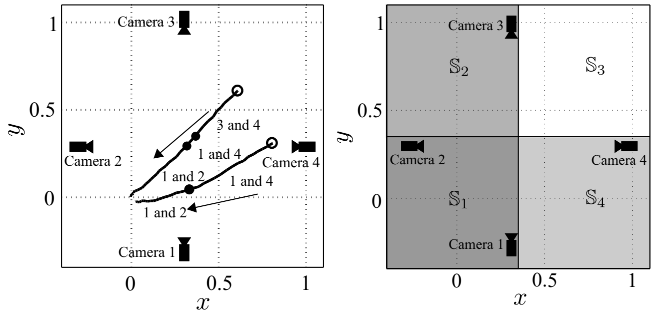

Fig. 4. Left: Sample paths of the vehicle. The symbol • represents points where the selected cameras are changed. The numbers under the sample paths mean the indexes of the selected cameras. The symbol ○ stands for the initial positions of the vehicle. Right: The convex polyhedra $\mathbb{S}_1$, $\mathbb{S}_2$, $\mathbb{S}_3$, and $\mathbb{S}_4$ in Example 4.

and

$$\boldsymbol{x}_p(0) = \begin{bmatrix} 0.81 & 0.31 & 0.01 & 0.01 \end{bmatrix}^\top,$$
$$\boldsymbol{x}_c(0) = \begin{bmatrix} 0.8 & 0.3 & 0 & 0 \end{bmatrix}^\top.$$

The sensor scheduling was performed for $\mathrm{E}[\boldsymbol{x}_p(k)] = \boldsymbol{x}_c(k)$. The proposed algorithm and exhaustive search method give the same sensor scheduling and the same sample path when the noise sequences are same and the exhaustive search method uses Lemmas 1 and 2 to calculate the values of the cost function.

The proposed method and the exhaustive search method require, respectively, 4 comparisons and 1024 comparisons to derive $i^*(0)$ at each time, where recall that the numbers of comparisons are equal to N and N^{T+1}, respectively. The proposed method takes 0.84 [msec.] at each time on average, while the exhaustive search method takes 310 [msec.]. Computation time is less than the sampling time for the proposed method, but not for the exhaustive search method. The programs ran in MATLAB 7.1 on a PentiumD 3.2 GHz PC with 2 GB of RAM.

The right of Fig. 4 illustrates the convex polyhedra $\mathbb{S}_1$, $\mathbb{S}_2$, $\mathbb{S}_3$, and $\mathbb{S}_4$ in (33). Each switched line for two cameras that faces each other is the equidistant line. This is reasonable, since the dynamics are decoupled.

IV. FAST SENSOR SCHEDULING BASED ON A LINEAR APPROXIMATION

We proposed the fast and optimal sensor scheduling algorithm in the previous section when (20) holds. The sensor models (12) and (13) satisfies (20), but it does not hold for (6) or (7). This section is devoted to the general model and provides a fast sensor scheduling algorithm based on a linear approximation even when (20) is not true.

The following corollary can be proven in a similar way to Theorem 1.

Corollary 3: Suppose that there exist constant matrices $\boldsymbol{S}_{i\ell} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{s}_{i\ell} \in \mathbb{R}^p$ such that

$$\begin{aligned} \boldsymbol{d}_{i\ell}(\boldsymbol{x})\boldsymbol{d}_{im}^\top(\boldsymbol{x}) = \boldsymbol{s}_{i\ell}\boldsymbol{s}_{im}^\top + \boldsymbol{S}_{i\ell}(\boldsymbol{x} - \boldsymbol{x}_0)\boldsymbol{s}_{im}^\top \\ + \boldsymbol{s}_{i\ell}(\boldsymbol{x} - \boldsymbol{x}_0)^\top \boldsymbol{S}_{im}^\top \end{aligned} \tag{41}$$

holds for all $i \in \{1, 2, \cdots, N\}$ and $\ell, m \in \{1, 2, \cdots, q\}$. Then (27) in Theorem 1 is also true when $\boldsymbol{\Phi}_i$, $\boldsymbol{\Psi}_i$, and $\boldsymbol{P}$ are replaced with

$$\begin{aligned}\boldsymbol{\Phi}_i(k) =& \boldsymbol{D}_c^\top \boldsymbol{R} \boldsymbol{D}_c \sum_{\ell=1}^{q} \sum_{m=1}^{q} V_{\ell m} \{\boldsymbol{S}_{i\ell}(\boldsymbol{A}^k - \boldsymbol{I})\boldsymbol{x}_0 \boldsymbol{s}_{im}^\top \\ &+ \boldsymbol{s}_{i\ell}(\boldsymbol{S}_{im}(\boldsymbol{A}^k - \boldsymbol{I})\boldsymbol{x}_0)^\top + \boldsymbol{s}_{i\ell}\boldsymbol{s}_{im}^\top\}, \qquad (42)\\ \boldsymbol{\Psi}_i(k) =& \sum_{\ell=1}^{q} \sum_{m=1}^{q} V_{\ell m} \boldsymbol{B} \{\boldsymbol{S}_{i\ell}(\boldsymbol{A}^k \boldsymbol{x}_0 - \boldsymbol{I})\boldsymbol{s}_{im}^\top \\ &+ \boldsymbol{s}_{i\ell}(\boldsymbol{S}_{im}(\boldsymbol{A}^k - \boldsymbol{I})\boldsymbol{x}_0)^\top + \boldsymbol{s}_{i\ell}\boldsymbol{s}_{im}^\top\}\boldsymbol{B}^\top \\ &+ \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \qquad (43)\\ \boldsymbol{P}(k) =& \boldsymbol{Q} + \boldsymbol{A}^\top \boldsymbol{P}(k+1)\boldsymbol{A}. \qquad (44)\end{aligned}$$

Let us now propose a fast sensor scheduling for general systems. Suppose that $\boldsymbol{d}_{i\ell}(\boldsymbol{x})$ is differentiable. Then we obtain (41) with

$$\boldsymbol{s}_{i\ell} = \boldsymbol{d}_{i\ell}(\boldsymbol{x}_0), \qquad (45)$$

$$\boldsymbol{S}_{i\ell} = \left.\frac{\partial \boldsymbol{d}_{i\ell}}{\partial \boldsymbol{x}}^\top\right|_{\boldsymbol{x}=\boldsymbol{x}_0} \qquad (46)$$

from a linear approximation of $\boldsymbol{d}_{i\ell}(\boldsymbol{x})\boldsymbol{d}_{im}^\top(\boldsymbol{x})$ around $\boldsymbol{x}_0$. Corollary 3 gives a suboptimal solution $i^*(0)$. Repeating from definitions of (45) and (46) to obtaining $i^*(0)$, we have a fast sensor scheduling.

Example 5: Consider the vehicle shown in Example 4 again. Forty nine radar sensors are set at

$$(0, 0),\ (0, 0.25), \cdots, (0, 1.5),\ (0.25, 0), \cdots, (1.5, 1.5).$$

The sensor model is given by (6). The covariance matrix of $\boldsymbol{v}$ is given by $\mathrm{diag}(0.002, 0.01)$. The other parameters are the same as Example 4. Fig. 5 shows a sample path of the vehicle for

$$\begin{aligned}\boldsymbol{x}_p(0) &= \begin{bmatrix}1.21 & 1.21 & 0.01 & 0.01\end{bmatrix}^\top, \\ \boldsymbol{x}_c(0) &= \begin{bmatrix}1.2 & 1.2 & 0 & 0\end{bmatrix}^\top.\end{aligned}$$

The sensor scheduling was performed for $\mathrm{E}[\boldsymbol{x}_p(0)] = \boldsymbol{x}_c(0)$.

The numbers of comparisons for the proposed method and the exhaustive search method to derive $i^*(0)$ are 49 and more than 2.8×10^8, respectively, where recall that they are equal to N and N^{T+1}, respectively. The proposed method takes 18 [msec.] to get $i^*(0)$ on average, though the exhaustive search method can not be used due to lack of memory where Monte Carlo method is implemented in the exhaustive search method to compute the values of the cost function. The proposed method is useful for large-scale networked sensor systems.

V. Conclusion

In this paper, a sensor scheduling problem for the class of systems whose measurements are influenced by state dependent noise was addressed. It is assumed that all sensors have state dependent noise and have the same characteristics, which follows from the properties of networked sensor systems. The sensor scheduling problem was formulated as a model predictive control problem with single sensor measurement per time. We proposed a fast and optimal sensor scheduling algorithm for a class of networked sensor systems. Computation time of the proposed algorithm is proportional to the number of sensors and does not depend on the prediction horizon. In addition, a fast sensor scheduling algorithm for a general class of systems was provided by using a linear approximation of the sensor model. The numerical examples showed that the proposed method provide reasonable solutions and is useful for large-scale networked sensor systems.

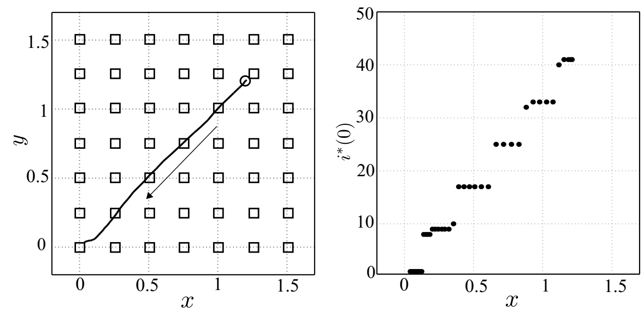

Fig. 5. Left: Solid line; a sample path of the vehicle. □; the positions of the radar sensors. ○; the initial position. Right: The sensor scheduling.

References

[1] P. Alriksson and A. Rantzer: Sub-Optimal Sensor Scheduling with Error Bounds; *Proc. 16th IFAC World Congress* (2005)

[2] M. A. Batalin, G. S. Sukhatme, and M. Hattig: Mobile Robot Navigation Using a Sensor Network; *Proc. of IEEE Conf. on Robotics and Automation* (2004)

[3] T. H. Chung, V. Gupta, B. Hassibi, J. W. Burdick, and R. M. Murray: Scheduling for Distributed Sensor Networks with Single Sensor Measurement per Time Step; *Proc. of IEEE Conf. on Robotics and Automation* (2004)

[4] T. H. Chung, V. Gupta, B. Hassibi, J. W. Burdick, and R. M. Murray: On a Decentralized Active Sensing Strategy Using Mobile Sensor Platforms in a Network; *Proc. of IEEE Conf. on Decision and Control* (2004)

[5] Z. G. Feng, K. L. Teo, and Y. Zaho: Branch and Bound Method for Sensor Scheduling in Discrete Time; *Journal of Industrial and Management Optimization*, Vol. 1, No. 4, pp. 499–512 (2005)

[6] L. E. Ghaoui: State-Feedback Control of Systems with Multiplicative Noise via Linear Matrix Inequalities; *Systems and Control Letters*, Vol. 24, pp. 223–228 (1995)

[7] S. Kagami, and M. Ishikawa: A Sensor Selection Method Considering Communication Delays; *Proc. of IEEE. Conf. on Robotics and Automation* (2004)

[8] K. Kotay, R. Peterson, and D. Rus: Experiments with Robots and Sensor Networks for Mapping and Navigation; *Proc. of Intl. Conf. on Field and Service Robotics* (2005)

[9] H. W. J. Lee, K. L. Teo, and A. E. B. Lim: Sensor Scheduling in Continuous Time; *Automatica*, Vol. 37, pp. 2017–2023 (2001)

[10] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry: *An Invitation to 3-D Vision*; Springer (2004)

[11] H. Morikawa: Ubiquitous Sensor Networks; *Proc. of US-Japan Workshop on Sensors, Smart Structures and Mechatronic Systems* (2005)

[12] K. V. Ramachandra: *Kalman Filtering Techniques for Radar Tracking*; Marcel Dekker (2000)

[13] I. Stojmenović: *Sensor Networks*; Wiley InterScience (2005)

[14] F. Zaho, J. Shin, and J. Reich: Information-Driven Dynamic Sensor Collaboration for Tracking Applications; *IEEE Signal Processing Mag.*, Vol. 19, No. 2, pp. 61–72 (2002)