2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Fast Sensor Scheduling for Spatially Distributed Heterogeneous Sensors

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. A networked sensor system usually consists of a large number of sensors, but the sensors can be classified into a few different types. We therefore introduce a concept of *sensor types* in the sensor model to provide a fast and optimal sensor scheduling algorithm for a class of networked sensor systems, where the sensor scheduling problem is formulated as a model predictive control problem. The computation time of the proposed algorithm increases exponentially with the number of the sensor types, while that of standard algorithms is exponential in the number of the sensors. In addition, we propose a fast sensor scheduling algorithm for a general class of networked sensor 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 [1], [2]. In recent years, networked sensor systems have been implemented in control systems such as robot control systems [3], [4] and target tracking systems [5]. Sensors in a networked sensor system are connected wirelessly, and each sensor operates with a battery power source. It is therefore required for each sensor to prolong the battery life, or equivalently, to achieve power saving [6]. One approach to meeting this requirement is to restrict the number of available sensors at each time and select a set of measuring sensors dynamically. This process is called sensor scheduling.

The major problem in 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 [7], a branch and bound method [8], and a sub-optimal method based on relaxed dynamic programming [9] have been proposed for sensor scheduling. In addition, a sensor scheduling strategy for continuous-time systems has been provided in [10]. These approaches assume that sensor characteristics are different from each other, that is, each sensor observes a different state or the covariance matrices for the sensor model are different from each other.

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).

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 [2]. In other words, many sensors in a networked sensor system 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. The measurement noise of each sensor may depend on the position of a measured object relative to the position of the sensor. In particular, measurements taken by cameras or radar sensors are influenced by state dependent noise [11], [12], [13]. The existing works do not provide any optimal sensor scheduling algorithm for systems with state dependent noise.

To solve the problems, we have proposed a fast and optimal sensor scheduling algorithm for networked sensor systems whose measurements are influenced by state dependent noise [14]. The sensor scheduling problem is formulated as a model predictive control problem with single sensor measurement per time. The scheduling algorithm minimizes a given quadratic cost function at each time. Its computation time is proportional to the number of the sensors, and it does not depend on the prediction horizon.

This paper presents a fast sensor scheduling algorithm for a class of networked sensor systems with heterogeneous sensors, since the previous scheduling algorithm proposed in [14] is valid only for a class of networked sensor systems whose all sensors have the same characteristics. To this end, a concept of *sensor types* is introduced in the sensor model. The key idea to obtain a fast sensor scheduling algorithm is to separate the original sensor scheduling problem into two scheduling problems: scheduling of sensor types and scheduling of sensors in a given sensor type sequence. The later is solved by using a fast algorithm that is similar to the algorithm proposed in [14]. The fast scheduling algorithm proposed in this paper is optimal for a class of networked sensor systems. Its computation time increases exponentially with the number of sensor types, while that of standard methods is exponential in the number of sensors. The proposed algorithm is efficient, since a networked sensor system usually consists of a few sensor types as mentioned before. In addition, we provide a fast sensor scheduling algorithm for a general class of networked sensor systems by using a linear approximation of the sensor model.

We use the following notation. The expectation operator is denoted by $\mathrm{E}[\cdot]$. The Kronecker delta is denoted by $\delta_{\ell m}$.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

II. Sensor scheduling problem

A. System description

This paper considers a class of systems as illustrated in Fig. 1. The system has N sensors indexed from 1 to N. For simplicity it is assumed, until Section IV, that only one sensor is available at each time to achieve power saving. Systems that can use multiple sensors simultaneously are discussed in Section V.

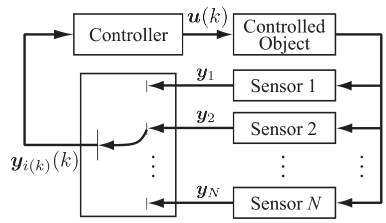

Fig. 1. A block diagram of a networked sensor system.

Let us now describe details of the system model.

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

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

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 sensor model is of the form:

$$\begin{aligned}\boldsymbol{y}_{i(k)}(k) =& \boldsymbol{C}_{\sigma(i(k))} \boldsymbol{x}_p(k) \\ &+ \sum_{\ell=1}^{q_{\sigma(i(k))}} \boldsymbol{d}_{i(k)\ell}(\boldsymbol{x}(k)) v_{i(k)\ell}(k)\end{aligned} \tag{2}$$

where $\boldsymbol{y}_{i(k)}(k) \in \mathbb{R}^{p_{\sigma(i)}}$ is the measurement taken by sensor $i(k)$, $i(k)$ the index of the selected sensor at time k, $\boldsymbol{d}_{i\ell}$ a vector function that is differentiable with respect to $\boldsymbol{x}$. The output $\boldsymbol{y}_{i(k)}(k)$ is simply written by $\boldsymbol{y}_i(k)$. The vector $\boldsymbol{x}$ is defined by

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

where $\boldsymbol{x}_c$ is the state vector of the controller which will be defined later. 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 1. The noise

$$\boldsymbol{v}_i(k) = \begin{bmatrix} v_{i1}(k) & v_{i2}(k) & \ldots & v_{iq}(k) \end{bmatrix}^\top \in \mathbb{R}^{q_{\sigma(i)}}$$

is white, Gaussian and zero mean with a covariance

$$\mathrm{E}[\boldsymbol{v}_i(k)\boldsymbol{v}_i^\top(\tau)] = \boldsymbol{V}_{\sigma(i)} \delta_{k\tau}.$$

It is assumed that $\boldsymbol{v}_i(k)$, $\boldsymbol{w}(k)$ and $\boldsymbol{x}_p(0)$ are mutually independent. It is clear that $\boldsymbol{d}_{i\ell}(\boldsymbol{x}(k))$ is independent of $\boldsymbol{v}_i(k)$. The function σ is a surjection from $\{1, 2, \ldots, N\}$ onto $\{1, 2, \ldots, M\}$ that satisfies

$$\boldsymbol{C}_{\sigma(i)} = \boldsymbol{C}_{\sigma(\ell)} \text{ and } \boldsymbol{V}_{\sigma(i)} = \boldsymbol{V}_{\sigma(\ell)} \Rightarrow \sigma(i) = \sigma(\ell),$$

where $\sigma(i)$ is sometimes simply written by σ. Each value of σ represents a *sensor type*. The number of sensor types is always less than or equal to the number of sensors, that is, $M \leq N$. A networked sensor system usually consists of a few sensors types [2]. Thus M is far less than N for a class of networked sensor systems. The concept of sensor types is effective for our scheduling algorithm proposed later in this paper, since the computation time of the proposed algorithm increases exponentially with M while that of the standard scheduling algorithms is exponential in N.

Finally, the controller is given by

$$\boldsymbol{x}_c(k+1) = \boldsymbol{A}_{c\sigma} \boldsymbol{x}_c(k) + \boldsymbol{B}_{1c\sigma} \boldsymbol{y}_i(k) + \boldsymbol{B}_{2c\sigma} \boldsymbol{u}(k), \tag{3}$$

$$\boldsymbol{u}(k) = \boldsymbol{C}_{c\sigma} \boldsymbol{x}_c(k) + \boldsymbol{D}_{c\sigma} \boldsymbol{y}_i(k) \tag{4}$$

where $\boldsymbol{x}_c(k) \in \mathbb{R}^{n_c}$ is the state of the controller. The system matrices in (3) and (4) are defined for each sensor type σ. 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.

Our previous result [14] provides a fast sensor scheduling algorithm for systems with a single sensor type, i.e. systems with $M = 1$. This paper focuses on systems with multiple sensor types.

Example 1: Consider four radar sensors and four cameras that measure the position of a target in the (x, y) plane as illustrated in Fig. 2.

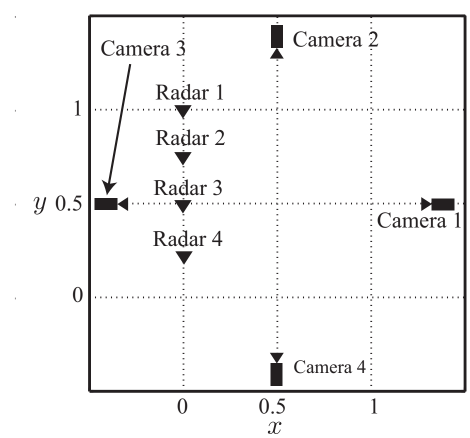

Fig. 2. Cameras and radar sensors in the field

The radar sensors are indexed from 1 to 4, and they are modeled by

$$\boldsymbol{y}_i = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & r_i \end{bmatrix} \boldsymbol{v}_i \tag{5}$$

[12] 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, and $\boldsymbol{v}_i$ a white, Gaussian and zero mean noise (see Fig. 3). It is clear that (5) is described by (2).

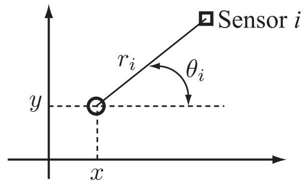

Fig. 3. Definitions of θ_i and r_i.

The cameras are also indexed from 1 to 4. The optical axes of cameras 1 and 3 are parallel to the x axis, and those of cameras 2 and 4 are parallel to the y axis. The indexes 5, 6, 7, and 8 are attached to camera combinations of (1,2), (2,3), (3,4) and (4,1). The cameras are modeled by

$$\boldsymbol{y}_i = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{y_c - p_{my}}{f} \\ 0 \end{bmatrix} v_{i1} + \begin{bmatrix} 0 \\ \frac{x_c - p_{\ell x}}{f} \end{bmatrix} v_{i2} \tag{6}$$

[14], where f is the focal length, (x_c, y_c) an estimate of (x, y), and v_{i1} and v_{i2} measurement noises for cameras m and ℓ, respectively. Equation (6) is represented by (2), since $\boldsymbol{d}_{i\ell}$ is a function of $\boldsymbol{x}$.

It is assumed that all the radar sensors have the same covariance matrix $\boldsymbol{V}_{\mathrm{r}}$ and all the cameras have the same covariance matrix $\boldsymbol{V}_{\mathrm{c}}$. Then the system has

$$\sigma(i) = \begin{cases} 1, & \text{for } i = 1, 2, 3, 4, \\ 2, & \text{for } i = 5, 6, 7, 8. \end{cases} \tag{7}$$

The number of sensors is $N = 8$, and the number of sensor types is $M = 2$.

B. Sensor scheduling problem

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

$$\boldsymbol{x}(k+1) = \boldsymbol{A}_\sigma \boldsymbol{x}(k) + \boldsymbol{B}_\sigma \sum_{\ell=1}^{q} \boldsymbol{d}_{i\ell}(\boldsymbol{x}) v_{i\ell}(k) + \begin{bmatrix} \boldsymbol{w}(k) \\ \boldsymbol{0} \end{bmatrix}, \tag{8}$$

where

$$\boldsymbol{A}_\sigma = \begin{bmatrix} \boldsymbol{A}_p + \boldsymbol{B}_p \boldsymbol{D}_c \boldsymbol{C}_\sigma & \boldsymbol{B}_p \boldsymbol{C}_{c\sigma} \\ \boldsymbol{B}_{1c\sigma} \boldsymbol{C}_\sigma + \boldsymbol{B}_{2c\sigma} \boldsymbol{D}_{c\sigma} \boldsymbol{C}_\sigma & \boldsymbol{A}_{c\sigma} + \boldsymbol{B}_{2c\sigma} \boldsymbol{C}_{c\sigma} \end{bmatrix}, \tag{9}$$

$$\boldsymbol{B}_\sigma = \begin{bmatrix} \boldsymbol{B}_p \boldsymbol{D}_{c\sigma} \\ \boldsymbol{B}_{1c\sigma} + \boldsymbol{B}_{2c\sigma} \boldsymbol{D}_{c\sigma} \end{bmatrix}. \tag{10}$$

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

$$J = \mathrm{E}\Big[\sum_{k=0}^{T} \{ \boldsymbol{x}_p^\top(k) \boldsymbol{Q}_p \boldsymbol{x}_p(k) + \boldsymbol{u}^\top(k) \boldsymbol{R} \boldsymbol{u}(k) \} + \boldsymbol{x}_p^\top(T+1) \boldsymbol{\Pi} \boldsymbol{x}_p(T+1) \Big]. \tag{11}$$

Find

$$\{ i^*(0), \ \cdots, \ i^*(T) \} = \arg \min_{i(0), \cdots, i(T)} J \tag{12}$$

for given $\boldsymbol{x}_0$.

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. It is clear that the exhaustive search method requires N^{T+1} comparisons to determine the optimal sensor sequence. Thus the exhaustive search method is not suitable for model predictive control from the point of view of computation time as will be shown in Section IV.

A state-dependent sensor scheduling algorithm is necessary, since $\boldsymbol{y}_i(k)$ is influenced by state dependent noise in (2). Note that $\mathrm{E}[\boldsymbol{x}(k)]$ is required to derive the optimal sensor sequence at time k. Therefore we have to estimate $\mathrm{E}[\boldsymbol{x}_p(k)]$ at each time. We use (3) as an observer for estimation in numerical examples presented in this paper.

III. Fast sensor scheduling

The key idea to obtain a fast sensor scheduling algorithm is to separate the original sensor scheduling problem into two scheduling problems: scheduling of sensor types and scheduling of sensors in a given sensor type sequence. The later is formulated as follows.

Problem 2: Let a sensor type sequence from time 0 to T be given and the given sensor type at time k be denoted by σ_k. Then find

$$\{ i_{\mathrm{s}}^*(0), \ \cdots, \ i_{\mathrm{s}}^*(T) \} = \arg \min_{i(0) \in \mathbb{I}(0), \cdots, i(T) \in \mathbb{I}(T)} J \tag{13}$$

where

$$\mathbb{I}(k) = \{ i \mid \sigma(i) = \sigma_k \}. \tag{14}$$

A. Optimal sensor scheduling

The following lemma provides a fast and optimal sensor scheduling for Problem 2.

Lemma 1: If there exist constant matrices $\boldsymbol{S}_{\sigma\ell} \in \mathbb{R}^{p \times n}$ and $\boldsymbol{s}_{i\ell} \in \mathbb{R}^p$ such that

$$\boldsymbol{d}_{i\ell}(\boldsymbol{x}) \boldsymbol{d}_{im}^\top(\boldsymbol{x}) = (\boldsymbol{S}_{\sigma\ell} \boldsymbol{x} + \boldsymbol{s}_{i\ell})(\boldsymbol{S}_{\sigma m} \boldsymbol{x} + \boldsymbol{s}_{im})^\top, \quad \forall i \in \{1, 2, \cdots, N\}, \quad \forall \ell, m \in \{1, 2, \cdots, q_\sigma\} \tag{15}$$

then the optimal sensor $i_{\mathrm{s}}^*(k)$ defined by (13) satisfies

$$i_{\mathrm{s}}^*(k) = \arg \min_{i(k) \in \mathbb{I}(k)} \mathrm{tr}[\boldsymbol{P}(k) \boldsymbol{\Psi}_i(k)] + \mathrm{tr}[\boldsymbol{\Phi}_i(k)] \tag{16}$$

where

$$\boldsymbol{P}(T+1) = \begin{bmatrix} \boldsymbol{\Pi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{17}$$

$$\boldsymbol{P}(k) = \boldsymbol{Q}(k) + {\boldsymbol{A}_{\sigma_k}}^\top \boldsymbol{P}(k+1) \boldsymbol{A}_{\sigma_k} + \sum_{\ell=1}^{q_{\sigma_k}} \sum_{m=1}^{q_{\sigma_k}} V_{\sigma_k \ell m} \boldsymbol{S}_{\sigma_k m}^\top \boldsymbol{B}_{\sigma_k}^\top \boldsymbol{P}(k+1) \boldsymbol{B}_{\sigma_k} \boldsymbol{S}_{\sigma_k \ell}, \tag{18}$$

$$\boldsymbol{\Psi}_i(k) = \sum_{\ell=1}^{q_{\sigma_k}} \sum_{m=1}^{q_{\sigma_k}} V_{\sigma_k \ell m} \boldsymbol{B}_{\sigma_k} (\boldsymbol{S}_{\sigma_k \ell} \boldsymbol{\mathcal{A}}(k) \boldsymbol{x}_0 \boldsymbol{s}_{im}^\top + \boldsymbol{s}_{i\ell} (\boldsymbol{S}_{\sigma_k m} \boldsymbol{\mathcal{A}}(k) \boldsymbol{x}_0)^\top + \boldsymbol{s}_{i\ell} \boldsymbol{s}_{im}^\top) \boldsymbol{B}_{\sigma_k}^\top + \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{19}$$

$$\boldsymbol{\Phi}_i(k) = \boldsymbol{D}_{c\sigma_k}^\top \boldsymbol{R} \boldsymbol{D}_{c\sigma_k} \sum_{\ell=1}^{q_{\sigma_k}} \sum_{m=1}^{q_{\sigma_k}} V_{\sigma_k \ell m} \{ \boldsymbol{S}_{\sigma_k \ell} \boldsymbol{\mathcal{A}}(k) \boldsymbol{x}_0 \boldsymbol{s}_{im}^\top + \boldsymbol{s}_{i\ell} (\boldsymbol{S}_{\sigma_k m} \boldsymbol{\mathcal{A}}(k) \boldsymbol{x}_0)^\top + \boldsymbol{s}_{i\ell} \boldsymbol{s}_{im}^\top \}, \tag{20}$$

$$\boldsymbol{Q}(k) = \begin{bmatrix} \boldsymbol{Q}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_{\sigma_k}^\top \boldsymbol{D}_{c\sigma_k}^\top \\ \boldsymbol{C}_{c\sigma_k}^\top \end{bmatrix} \boldsymbol{R} \begin{bmatrix} \boldsymbol{C}_{\sigma_k}^\top \boldsymbol{D}_{c\sigma_k}^\top \\ \boldsymbol{C}_{c\sigma_k}^\top \end{bmatrix}^\top + \sum_{\ell=1}^{q_{\sigma_k}} \sum_{m=1}^{q_{\sigma_k}} V_{\sigma_k \ell m} \boldsymbol{S}_{\sigma_k m}^\top \boldsymbol{D}_{c\sigma_k}^\top \boldsymbol{R} \boldsymbol{D}_{c\sigma_k} \boldsymbol{S}_{\sigma_k \ell}, \tag{21}$$

$$\boldsymbol{\mathcal{A}}(k) = \boldsymbol{A}_{\sigma_{k-1}} \cdots \boldsymbol{A}_{\sigma_1} \boldsymbol{A}_{\sigma_0}, \tag{22}$$

and $V_{\sigma_k \ell m}$ is the (ℓ, m) elements of $\boldsymbol{V}_{\sigma_k}$.

Proof: The proof can be done in the same way as Theorem 1 in [14]. ■

Lemma 1 gives a fast method to solve Problem 2. Actually, $\boldsymbol{\mathcal{A}}(k)$ and $\boldsymbol{P}(k)$ can be computed before sensor scheduling, since the sensor type sequence is given. Thus, $i_s^*(k)$ in (13) is obtained from (16) by comparing possible $\mu(\sigma_k)$ values at each time, where $\mu(\sigma)$ denotes the number of sensors that belong to σ.

Applying Lemma 1 to the set of all possible sensor type sequences yields a solution of Problem 1 as follows.

1) $\mathbb{O} \leftarrow \{1, 2, \cdots, M\}^{T+1}$.
2) Repeat the following operations a), b) and c) from $m = 1$ to M^{T+1}.
   a) Choose a sensor type sequence $O_m \in \mathbb{O}$.
   b) Derive the optimal sensor sequence for O_m by using Lemma 1, and obtain the corresponding cost J_m.
   c) $\mathbb{O} \leftarrow \mathbb{O}/O_m$.
3) $m^* = \arg\ \min_m J_m$.
4) Obtain $\{i_s^*(0),\ \cdots,\ i_s^*(T)\}$ for O_{m^*} as the optimal sensor sequence for Problem 1.

Let us now evaluate the computational cost of the proposed algorithm. Let σ_{mk} denote the k-th element of O_m. For each m, $\sum_{k=0}^{T} \mu(\sigma_{mk})$ values of the cost function are computed to obtain J_m. Hence the number of comparisons required in Step 2 is given by

$$\sum_{m=1}^{M^{T+1}} \sum_{k=0}^{T} \mu(\sigma_{mk}) = \sum_{k=0}^{T} \sum_{m=1}^{M^{T+1}} \mu(\sigma_{mk}) = (T+1) N M^T. \tag{23}$$

The last equality follows from the fact that each sensor type appears M^T times in all possible sensor type sequences. Furthermore, Step 3 compares M^{T+1} values to obtain m^*. We therefore have the following theorem

Theorem 1: The computational cost of the proposed algorithm is given by $O(TNM^T)$.

Recall that the exhaustive search method requires N^{T+1} sensor sequences to determine the optimal sensor sequence, and its computational cost increases exponentially with N. Thus the proposed algorithm is effective for networked sensor systems with heterogeneous sensors when the number of sensor types, M, is far less than the number of sensors, N.

B. Fast sensor scheduling based on a linear approximation

We proposed the fast and optimal sensor scheduling algorithm in the previous section when (15) holds. The measurement model (5) does not satisfy (15), while (15) holds for (6). This section is devoted to a generalization and provides a fast sensor scheduling algorithm based on a linear approximation that is valid even when (15) is not true. To this end, the following corollary is established.

Corollary 1: 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

$$\boldsymbol{d}_{i\ell}(\boldsymbol{x}) \boldsymbol{d}_{im}^\top(\boldsymbol{x}) = \boldsymbol{s}_{i\ell} \boldsymbol{s}_{im}^\top + \boldsymbol{S}_{i\ell}(\boldsymbol{x}(k) - \boldsymbol{x}_0) \boldsymbol{s}_{im}^\top + \boldsymbol{s}_{i\ell} (\boldsymbol{x}(k) - \boldsymbol{x}_0)^\top \boldsymbol{S}_{im}^\top, \quad \forall i \in \{1, 2, \cdots, N\}, \quad \forall \ell, m \in \{1, 2, \cdots, q_\sigma\} \tag{24}$$

holds. Then (16) in Lemma 1 is also true when $\boldsymbol{\Phi}_i$, $\boldsymbol{\Psi}_i$, and $\boldsymbol{P}$ are replaced with

$$\boldsymbol{P}(k) = \boldsymbol{Q}(k) + \boldsymbol{A}_{\sigma_k}^\top \boldsymbol{P}(k+1) \boldsymbol{A}_{\sigma_k}, \tag{25}$$

$$\boldsymbol{\Phi}_i(k) = \boldsymbol{D}_{c\sigma_k}^\top \boldsymbol{R} \boldsymbol{D}_{c\sigma_k} \sum_{\ell=1}^{q_{\sigma_k}} \sum_{m=1}^{q_{\sigma_k}} V_{\sigma_k \ell m} \times \big\{ \boldsymbol{S}_{i\ell} (\boldsymbol{\mathcal{A}}(k) - \boldsymbol{I}) \boldsymbol{x}_0 \boldsymbol{s}_{im}^\top + \boldsymbol{s}_{i\ell} (\boldsymbol{S}_{im} (\boldsymbol{\mathcal{A}}(k) - \boldsymbol{I}) \boldsymbol{x}_0)^\top + \boldsymbol{s}_{i\ell} \boldsymbol{s}_{im}^\top \big\}, \tag{26}$$

$$\boldsymbol{\Psi}_i(k) = \sum_{\ell=1}^{q_{\sigma_k}} \sum_{m=1}^{q_{\sigma_k}} V_{\sigma_k \ell m} \boldsymbol{B}_{\sigma_k} \{ \boldsymbol{S}_{i\ell} (\boldsymbol{\mathcal{A}}(k) \boldsymbol{x}_0 - \boldsymbol{I}) \boldsymbol{s}_{im}^\top + \boldsymbol{s}_{i\ell} (\boldsymbol{S}_{im} (\boldsymbol{\mathcal{A}}(k) - \boldsymbol{I}) \boldsymbol{x}_0)^\top + \boldsymbol{s}_{i\ell} \boldsymbol{s}_{im}^\top \} \boldsymbol{B}_{\sigma_k}^\top + \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{27}$$

Proof: The corollary can be proven in a similar way to Lemma 1. ■

Let us now propose a fast sensor scheduling for general systems. Suppose that $\boldsymbol{d}_{i\ell}(\boldsymbol{x})$ is differentiable. Then we obtain (24) with

$$\boldsymbol{s}_{i\ell} = \boldsymbol{d}_{i\ell}(\boldsymbol{x}_0), \tag{28}$$

$$\boldsymbol{S}_{i\ell} = \left. \frac{\partial \boldsymbol{d}_{i\ell}}{\partial \boldsymbol{x}}^\top \right|_{\boldsymbol{x}=\boldsymbol{x}_0} \tag{29}$$

from a linear approximation of $\boldsymbol{d}_{i\ell}(\boldsymbol{x}) \boldsymbol{d}_{im}^\top(\boldsymbol{x})$ around $\boldsymbol{x}_0$. Corollary 1 with the sensor scheduling algorithm proposed in the previous section gives a suboptimal sensor sequence. Repeating from definitions of (28) and (29) to obtaining a suboptimal sensor sequence, we have a fast sensor scheduling algorithm.

IV. NUMERICAL EXAMPLE

Consider again the networked sensor system shown in Example 1. The goal here is to control a vehicle that travels on the two-dimensional plane using the networked sensor system. Let $(x,\ y)$ be the position of the vehicle. The state equation of the vehicle in continuous time is given by

$$\dot{\bar{\boldsymbol{x}}}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \bar{\boldsymbol{x}}_p + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \bar{\boldsymbol{u}} + \bar{\boldsymbol{w}}, \qquad (30)$$

where $\bar{\boldsymbol{x}} := [x\ y\ \dot{x}\ \dot{y}]^\top \in \mathbb{R}^4$ 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 (30) is discretized with a sampling period 0.1. An observer (3) and a controller (4) are implemented such that the poles of (8) 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.

Radar sensors 1, 2, 3, and 4 are set at

$$(0, 1),\ (0, 0.75),\ (0, 0.5),\ (0, 0.25),$$

and cameras 1, 2, 3, and 4 are set at

$$(1.3, 0.5),\ (0.5, 1.3),\ (-0.3, 0.5),\ (0.5, -0.3),$$

respectively. Parameters are set to

$$\begin{aligned} &\boldsymbol{V}_{\mathrm{r}} = \mathrm{diag}(0.002,\ 0.01), \\ &\boldsymbol{V}_{\mathrm{c}} = \mathrm{diag}(0.001,\ 0.001), \\ &\boldsymbol{Q}_p = \boldsymbol{I}, \quad \boldsymbol{R} = \boldsymbol{I}, \quad \boldsymbol{\Pi} = \boldsymbol{I}, \quad T = 3, \quad f = 0.01 \end{aligned}$$

Fig. 4 illustrates sample paths of the vehicle for

$$\begin{aligned} \boldsymbol{x}_p(0) &= \begin{bmatrix} 1.01 & 1.01 & 0.01 & 0.01 \end{bmatrix}^\top, \\ \boldsymbol{x}_c(0) &= \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^\top. \end{aligned}$$

In the proposed method, the radar sensor model (5) is linearized by (24), so that the proposed algorithm can be applied. The same noise sequence was used for the both cases.

Fig. 5 shows the obtained sensor sequences. The selected sensor of the proposed method at time 8 is different from that of the exhaustive search method. The exhaustive search method is better than the proposed method from the point of view of the cost, since the proposed method is based on a linear approximation. However, the difference between the obtained costs is small. In fact, the difference is less than 0.5 % at time 8.

On the other hand, the proposed method has less computation time than the exhaustive search method. The proposed method takes 5.3×10^{-2} [s] at each time on average, and the computation time in each step is within the sampling period. Meanwhile, the exhaustive search method takes 9.2×10^4 [s] where Monte Carlo method is used to calculate the values of the cost function at each time. The result confirms the effectiveness of the proposed sensor scheduling algorithm. The programs ran in MATLAB 7.1 on a PentiumD 3.2 GHz PC with 2 GB of RAM.

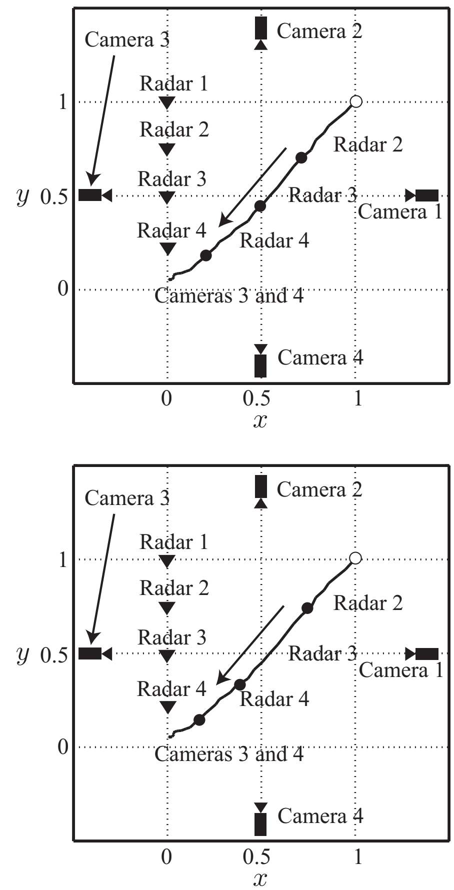

Fig. 4. Sample paths of the vehicle. The selected sensors are described with the paths. ○: the initial positions. ●: switching points. Top: proposed method. Bottom: exhaustive search method.

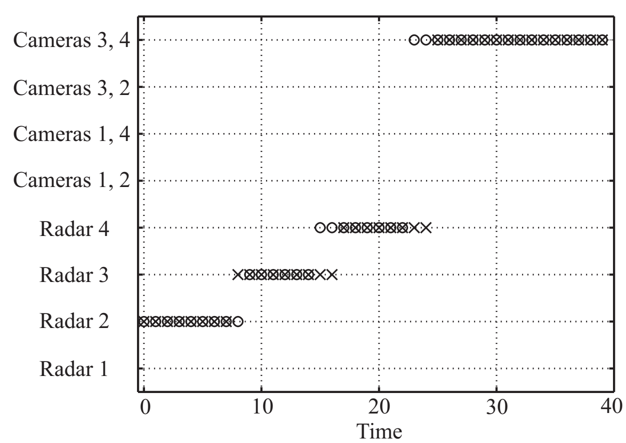

Fig. 5. Sequences of the selected sensors. ○: proposed method. ×: exhaustive search method.

V. Sensor scheduling for systems that can use multiple sensors simultaneously

In the previous sections, this paper assumed that one sensor is available at each time. The proposed algorithms can be applied to systems that can use multiple sensors at each time, when all possible sensor combinations are indexed. This section shows the proposed method has also less computational complexity than the exhaustive search method for systems that use L sensors simultaneously.

The number of selections of L sensors from N sensors without repetition is ${}_NC_L$. The number of selections of L sensor types from M sensor types with repetition is ${}_{M+L-1}C_L$. Thus the computational cost of the proposed method is given by

$$O(({}_NC_L)^{T+1}), \tag{31}$$

and that of the exhaustive search method is represented by

$$O(T({}_NC_L)({}_{M+L-1}C_L)^T). \tag{32}$$

It is straightforward to verify that

$${}_NC_L >_{M+L-1} C_L \tag{33}$$

is equivalent to

$$N+1 > M+L. \tag{34}$$

This implies that the proposed method is effective when the sum of the numbers of sensor types M and simultaneously available sensors L is far less than the number of sensors N.

VI. Conclusion

In this paper, a sensor scheduling problem for the class of systems whose measurements are influenced by state dependent noise was addressed. A concept of *sensor types* in the sensor model was introduced, and it has an important advantage to making a fast sensor scheduling algorithm. The sensor scheduling problem was formulated as a model predictive control problem, and we proposed a fast sensor scheduling algorithm for a class of networked sensor systems with a few sensor types. The computation time of the proposed algorithm increases exponentially with the number of sensor types, while the computation time of a primitive algorithm is exponential in the number of sensors. The computation time of the proposed algorithm may be further reduced when a branch and bound method is implemented. A numerical example demonstrated that the proposed algorithm is effective for networked sensor systems with heterogeneous sensors.

References

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

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

[3] 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)

[4] 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)

[5] 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)

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

[7] 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)

[8] 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)

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

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

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

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

[13] K. Umeda, J. Ota, and H. Kimura: Fusion of Multiple Ultrasonic Sensor Data and Imagery Data for Measuring Moving Obstacle's Motion; *Proc. of Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems* (1996)

[14] S. Arai, Y. Iwatani, and K. Hashimoto: Fast and Optimal Sensor Scheduling for Networked Sensor Systems; *Proc. of IEEE Conf. on Decision and Control* (2008)