

A Sensor Selection Method Considering Communication Delays

Shingo Kagami and Masatoshi Ishikawa
Department of Information Physics and Computing,
University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Email: {Shingo_Kagami,Masatoshi_Ishikawa}@ipc.i.u-tokyo.ac.jp

Abstract—A sensor selection method for distributed real-time sensing is proposed where the effect of communication delays is taken into account. First, a multi-sensor fusion algorithm that allows communication delays is proposed. This method takes advantage of the information form of Kalman filtering in order to maintain scalability with the number of sensors. Based on this algorithm, an algorithm is developed that calculates mutual information between an observation and the state to be estimated, which provides a criterion for sensor selection. Simulated results of a target-tracking task show its effectiveness.

I. INTRODUCTION

Rapid advances of networking technologies, such as wireless communication and mobile ad hoc networking, and device integration technologies, such as VLSI and MEMS, are enabling vast numbers of small sensors to be spread over the real world and networked with each other [1].

In particular, efforts aiming at open and flexible sensor networks that can adapt to dynamically changing environments and objectives are increasingly gaining attention, rather than closed and fixed sensor networks that have been constructed in application-specific ways. They include many challenges from diversified points of view such as development of sensor nodes, communication facilities, network architectures, software architectures and algorithms [2], [3], [4].

To effectively utilize these sensor networks where an enormous number of various sensors will be included, we will also need innovations in sensory information processing. One of important issues is how to map a sensing task into the network of distributed sensors, that is, how to determine a set of sensors to be activated and integrate the information obtained by them. It must be optimized considering various kinds of conditions such as computational complexity, network bandwidth, communication delays, and power dissipation. Activating all of the available sensors might be undesirable for networks with band-limited communication media or battery-constrained sensor nodes; it is important to evaluate each sensor and accordingly specify a set of sensors to be activated.

This paper proposes a multi-sensor fusion method suited for integrating data from a huge number of sensors, and a method of evaluating each sensor to specify a set of sensors to be activated, where the effect of communication delays between sensors and the user is taken into account. When a state of interest is to be estimated in real time, a sensor with a too long

delay is almost useless no matter how accurate the sensor is. On the other hand, a sensor that suffers from too much noise is also ineffective no matter how short its delay is. Appropriate handling of this tradeoff is required.

In the section II, the proposed sensor fusion method based on the Kalman filter is described. In the section III, the sensor selection method that resolves the above stated tradeoff is described. Experimental results are shown in the section IV.

II. DELAY-TOLERANT KALMAN FILTERING

A. Related Work and Strategy

Kalman filtering in a situation where communication delays exist in delivery of sensory information is defined as a problem to find the linear minimum variance estimate given all the observations that arrived at the user (or filtering processor) up to the current time, bearing in mind that any observations that have not arrived are not available for estimation.

The usual Kalman filter has a recursive form, that is, the optimal estimate is obtained by fusing the observation with the predicted state from the optimal estimate at the previous time step. It assumes no delay in sensory information delivery.

Delays can be coped with by applying only time-update (predict) operations after the time of observation of which information arrived most recently, provided that the observations arrive in order. When information of an earlier observation arrives after arrival of a later observation, the belated information cannot be fused and is simply wasted.

It is easily found that we can permit time delays of $L - 1$ time steps at most if we calculate the optimal estimate starting from the estimate at L steps behind, instead of starting from 1 step behind.

However, performing this in a simple way will cost huge computation time and memory amount, since we have to store all the observations from all the sensors until they get L steps old, and execute the observation update procedure for each stored observation for each time step. It is thus not scalable with respect to the number of sensors.

Maeyama et al. proposed a retroactive sensor data fusion algorithm based on the Kalman filter, where sensory data with delays are coped with [5]. It rather focused on the scalability with respect to the maximum length of delays, L in our notation, while we prefer the scalability with respect to the number of sensors.

Durrant-Whyte et al. proposed decentralized Kalman filtering for data fusion in sensor networks [6], [7], [8]. This series of work is based on Hashemipour et al.'s parallel Kalman filtering [9], and can be regarded as its decentralized extension where the existence of the central processor is removed.

Decentralization by Rao et al. [7] assumed network topologies where all the nodes are fully connected to every other node. Grime and Durrant-Whyte [8] extended this architecture to support tree-connected network topologies by introduction of an additional filter associated with each communication link.

All of these architectures of the parallel Kalman filter assumed that every communication link has no, or sufficiently short, time delays. Although Grime and Durrant-Whyte's method [8] is also one of those kinds, it is notable in that it did not assume fully connected topologies and thereby sensory information is allowed to be delivered with delays by passing through several nodes. In that sense, it achieves optimal filtering in a situation with communication delays.

However, their method handles communication delays with a combination of filtering at every node and filtering at every communication link. Thereby sensory information cannot pass through any nodes that do not implement the filters. This means that their method can be applied only to closed and application-specific networks. In TCP/IP networks for example, intermediate nodes just relay packets and no filtering operations can be embedded in them. In wireless communication, even one-hop delay cannot be neglected. In a heterogeneous networking environment, we cannot assume all of the nodes to be under our control.

This section describes an algorithm for the Kalman filter that achieves optimal filtering in a situation with communication delays, scalability with the number of sensors, and allowing implementation at only end nodes.

B. Algorithm

We assume the dynamics of the state of interest \mathbf{x} and the observation \mathbf{y} are described as

$$\mathbf{x}(k+1) = A(k)\mathbf{x}(k) + B(k)\mathbf{u}(k) \quad (1)$$

$$\mathbf{y}(k) = C(k)\mathbf{x}(k) + \mathbf{w}(k) \quad (2)$$

$$\begin{aligned} E[\mathbf{u}(k)] &= \bar{\mathbf{u}}(k), \quad E[\mathbf{w}(k)] = \bar{\mathbf{w}}(k) \\ E[(\mathbf{u}(k) - \bar{\mathbf{u}}(k))(\mathbf{u}(l) - \bar{\mathbf{u}}(l))^T] &= \delta_{kl}U(k) \\ E[(\mathbf{w}(k) - \bar{\mathbf{w}}(k))(\mathbf{w}(l) - \bar{\mathbf{w}}(l))^T] &= \delta_{kl}W(k) \\ E[(\mathbf{u}(k) - \bar{\mathbf{u}}(k))(\mathbf{w}(l) - \bar{\mathbf{w}}(l))^T] &= O \end{aligned}$$

where $\mathbf{x}(k)$ is the state of interest at time instant k , and $\mathbf{y}(k)$ is the observation at time k . The vectors $\mathbf{u}(k)$ and $\mathbf{w}(k)$ are the process noise and the observation noise at time k , respectively. $E[\cdot]$ denotes expectation and δ_{ij} is the Kronecker delta.

It is a well-known result of the Kalman filter theory that the optimal estimate $\hat{\mathbf{x}}(k | k-1)$ of the state at time k given the

observations up to time $k-1$ is obtained by

$$\hat{\mathbf{x}}(k | k-1) = A(k-1)\hat{\mathbf{x}}(k-1 | k-1) + B(k-1)\bar{\mathbf{u}}(k-1) \quad (3)$$

$$P(k | k-1) = A(k-1)P(k-1 | k-1)A^T(k-1) + B(k-1)U(k-1)B^T(k-1) \quad (4)$$

and the optimal estimate $\hat{\mathbf{x}}(k | k)$ at time k given the observations up to time k by

$$P(k | k) = \{P^{-1}(k | k-1) + C^T(k)W^{-1}(k)C(k)\}^{-1} \quad (5)$$

$$\hat{\mathbf{x}}(k | k) = \hat{\mathbf{x}}(k | k-1) + P(k | k)C^T(k)W^{-1}(k)\{\mathbf{y}(k) - (C(k)\hat{\mathbf{x}}(k | k-1) + \bar{\mathbf{w}}(k))\} \quad (6)$$

where $P(i | j)$ denotes the estimate error covariance at time i given the observations up to time j , that is,

$$P(i | j) \equiv E[(\hat{\mathbf{x}}(i | j) - \mathbf{x}(i))(\hat{\mathbf{x}}(i | j) - \mathbf{x}(i))^T]. \quad (7)$$

These procedures are sometimes referred to as time update and observation update, respectively.

When the observation system is comprised of m sensors and observations by different sensors are mutually uncorrelated, the above observation equation can be partitioned into

$$\mathbf{y}_i(k) = C_i(k)\mathbf{x}(k) + \mathbf{w}_i(k) \quad (i = 1, 2, \dots, m)$$

$$\mathbf{y}(k) = [\mathbf{y}_1^T(k), \mathbf{y}_2^T(k), \dots, \mathbf{y}_m^T(k)]^T$$

$$C(k) = [C_1^T(k), C_2^T(k), \dots, C_m^T(k)]^T$$

$$\mathbf{w}(k) = [\mathbf{w}_1^T(k), \mathbf{w}_2^T(k), \dots, \mathbf{w}_m^T(k)]^T$$

$$W(k) = \text{blockdiag}\{W_1(k), W_2(k), \dots, W_m(k)\}.$$

This partitioned observation enables the observation update of Eqs. (5) and (6) to be transformed into

$$P^{-1}(k | k) = P^{-1}(k | k-1) + \sum_{i=1}^m C_i^T(k)W_i^{-1}(k)C_i(k) \quad (8)$$

$$P^{-1}(k | k)\hat{\mathbf{x}}(k | k) = P^{-1}(k | k-1)\hat{\mathbf{x}}(k | k-1) + \sum_{i=1}^m C_i^T(k)W_i^{-1}(k)\{\mathbf{y}_i(k) - \bar{\mathbf{w}}_i(k)\}. \quad (9)$$

This is based on a form of the Kalman filter that is expressed in terms of the inverse of the error covariance rather than the error covariance itself. This form is known as the information form, or the information filter [10]. It is utilized in many implementations of the parallel Kalman filter, and proved to be well suited for multi-sensor situations [6].

Equations (8) and (9) show that only the summations of the observation data are needed for one-step observation update. As an implementation, we thus need to prepare L information buffers that correspond to the past L time instants, and to add received sensory information to the corresponding buffers.

Based on the above discussion, we construct an algorithm of Kalman filtering that permit communication delays as follows. The algorithm is depicted in Fig. 1.

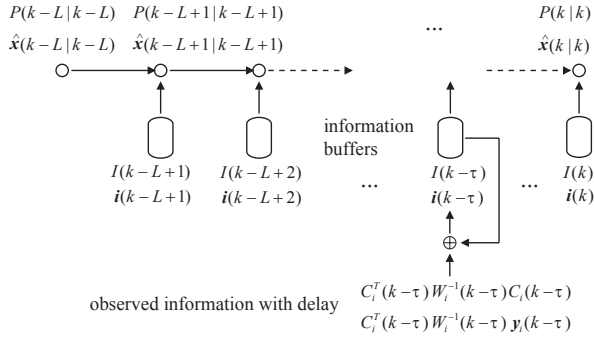


Fig. 1. Operation of the Delay-Tolerant Kalman filter.

Algorithm 1: Each sensor transmits the information state $C_i^T(k)W_i^{-1}(k)\{\mathbf{y}_i(k) - \bar{\mathbf{w}}_i(k)\}$ and the error information $C_i^T(k)W_i^{-1}(k)C_i(k)$ to the user with the time stamp k .

The user has the information state buffers $\mathbf{i}(j)$ and the error information buffers $I(j)$ ($j = k - L + 1, \dots, k$ for each) to store the received corresponding sensory information.

At each time instant, the user executes the following procedures:

- 1) Receive information $C_i^T(j)W_i^{-1}(j)\{\mathbf{y}_i(j) - \bar{\mathbf{w}}_i(j)\}$ and $C_i^T(j)W_i^{-1}(j)C_i(j)$ from a sensor i , and add them to the buffers corresponding to the time stamp j :

$$\begin{aligned} \mathbf{i}(j) &:= \mathbf{i}(j) + C_i^T(j)W_i^{-1}(j)\{\mathbf{y}_i(j) - \bar{\mathbf{w}}_i(j)\} \\ I(j) &:= I(j) + C_i^T(j)W_i^{-1}(j)C_i(j) \end{aligned}$$

Repeat this for all the sensory data that arrive at the user at this time instant, except for the case $j < k - L + 1$ where the sensory data are discarded. It should be noted that this procedure just executes addition, and thus the iteration with respect to the number of arriving sensory data will not cause severe increase of computation time.

- 2) Perform time-update and observation-update procedures starting from $\hat{\mathbf{x}}(k-L|k-L)$ to $\hat{\mathbf{x}}(k|k)$. Specifically,

$$\begin{aligned} &\text{for } j = k - L + 1, k - L + 2, \dots, k \text{ begin} \\ P(j|j-1) &:= A(j-1)P(j-1|j-1)A^T(j-1) \\ &\quad + B(j-1)U(j-1)B^T(j-1) \\ \hat{\mathbf{x}}(j|j-1) &:= A(j-1)\hat{\mathbf{x}}(j-1|j-1) \\ &\quad + B(j-1)\bar{\mathbf{u}}(j-1) \\ P(j|j) &:= \{P^{-1}(j|j-1) + I(j)\}^{-1} \\ \hat{\mathbf{x}}(j|j) &:= P(j|j)\{P^{-1}(j|j-1)\hat{\mathbf{x}}(j-1|j-1) \\ &\quad + \mathbf{i}(j)\} \\ &\text{end} \end{aligned}$$

is executed. Along with this calculation, store $\hat{\mathbf{x}}(k-L+1|k-L+1)$ and $P(k-L+1|k-L+1)$ for the next time instant.

- 3) Release the memory spaces for the buffers $\mathbf{i}(k-L+1)$ and $I(k-L+1)$. Allocate new memory spaces for $\mathbf{i}(k+1)$ and $I(k+1)$, and initialize them with a zero vector $\mathbf{0}$ and a zero matrix O .

□

In the rest of this paper, this algorithm is called DTKF (Delay-Tolerant Kalman Filter) for short.

III. SENSOR SELECTION METHOD CONSIDERING DELAYS

This section describes a method of evaluating sensory information that arrives with time delay based on the framework presented in the previous section.

A. Strategy

On sensor selection, it is obviously desirable to select a sensor that brings about much information about the state of interest and reduces uncertainty associated with it. Mutual information has thus been widely used to evaluate sensors or to optimize sensing behaviors. We shall derive a simple algorithm based on DTKF to compute mutual information for a situation with delays.

In the strict sense, mutual information must be evaluated for any feasible combinations and for any feasible time series of sensor selection. It is, however, not realistic for real-time processing. In addition, evaluating the whole sensing time series requires discussion on stationary processes, and cannot be applied to dynamically varying situations.

For these reasons, we shall discuss how much information is brought about to the current-time estimate when an observation by a sensor arrives with a delay, and investigate the performance of the sensor selection based on this approximation.

B. Algorithm

It is well known that the mutual information between the observation and the estimate of the state can be computed from the estimate error covariance when the process noise and the observation noise are Gaussian [11]. The mutual information for the case with communication delays can be derived just in a similar way to the case without delays [12] as follows.

Let \mathbf{y} be the observation that we want to evaluate, and \mathcal{Y} be the set of all the observations the user has received except \mathbf{y} . The mutual information between \mathbf{y} and the estimate of the state $\mathbf{x}_{\mathcal{Y}}$ given all the observations except \mathbf{y} is, from its definition, given by

$$\begin{aligned} I_{\text{mutual}}(\mathbf{x}_{\mathcal{Y}}; \mathbf{y}) &= I_{\text{mutual}}(\mathbf{y}; \mathbf{x}_{\mathcal{Y}}) \\ &= \iint p(\mathbf{y})p(\mathbf{x} | \mathcal{Y}, \mathbf{y}) \log \frac{p(\mathbf{x} | \mathcal{Y}, \mathbf{y})}{p(\mathbf{x} | \mathcal{Y})} d\mathbf{x}d\mathbf{y} \\ &= E \left[\log \frac{p(\mathbf{x} | \mathcal{Y}, \mathbf{y})}{p(\mathbf{x} | \mathcal{Y})} \right] \end{aligned} \quad (10)$$

where $p(\cdot)$ and $p(\cdot | \cdot)$ denote probability density and conditional probability density, respectively.

The conditional probability density functions $p(\mathbf{x} | \mathcal{Y})$ and $p(\mathbf{x} | \mathcal{Y}, \mathbf{y})$ are expressed as

$$\begin{aligned} p(\mathbf{x} | \mathcal{Y}) &= \frac{\exp \left\{ -\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{\mathcal{Y}})^T P_{\mathcal{Y}}^{-1}(k)(\mathbf{x} - \hat{\mathbf{x}}_{\mathcal{Y}}) \right\}}{\sqrt{(2\pi)^n |P_{\mathcal{Y}}(k)|}} \\ p(\mathbf{x} | \mathcal{Y}, \mathbf{y}) &= \frac{\exp \left\{ -\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{\mathcal{Y}, \mathbf{y}})^T P_{\mathcal{Y}, \mathbf{y}}^{-1}(k)(\mathbf{x} - \hat{\mathbf{x}}_{\mathcal{Y}, \mathbf{y}}) \right\}}{\sqrt{(2\pi)^n |P_{\mathcal{Y}, \mathbf{y}}(k)|}}. \end{aligned}$$

We can see that $\hat{\mathbf{x}}_{\mathcal{Y}}(k)$ is the estimate obtained by DTKF given all the information except \mathbf{y} , $\hat{\mathbf{x}}_{\mathcal{Y}, \mathbf{y}}(k)$ is the one given all the information including \mathbf{y} , and $P_{\mathcal{Y}}(k)$ and $P_{\mathcal{Y}, \mathbf{y}}(k)$ are

the estimation error covariances for those cases. Substituting these density functions into Eq. (10) gives

$$I_{\text{mutual}}(\mathbf{x}_y; \mathbf{y}) = \frac{1}{2} \log |P_y(k) P_{y,y}^{-1}(k)|. \quad (11)$$

Equation (11) provides means to calculate the mutual information between sensory information from a sensor i and the estimate during the execution of DTKF. Here, we assume that the communication delay τ_i and the error information $C_i^T(k - \tau_i)W_i^{-1}(k - \tau_i)C_i(k - \tau_i)$ associated with the sensor i are known.

When a user executes the DTKF algorithm at time k , a sensor i is either selected to be used for the sensor fusion or not selected. If selected, the error covariance $P(k | k)$ obtained by DTKF is based on the sensory data including one from the sensor i . Hence this gives $P_{y,y}(k)$. Removing the contribution of the sensor i from this gives $P_y(k)$, and this removal is easily realized by subtracting, since the contribution has been simply added to the error information buffer.

On the other hand, if the sensor i is not selected, the error covariance $P(k | k)$ obtained by DTKF gives $P_y(k)$. To obtain $P_{y,y}(k)$, we just need to add the contribution of the sensor i to the error information buffer.

By this means, we can calculate Eq. (11) for both of the cases. The algorithm can be described as follows.

Algorithm 2: Let S be the set of sensor indices that are selected currently. After the error covariance $P(k | k)$ is obtained by using DTKF, execute the following procedure:

```

if  $i \in S$  then
   $P_{y,y}(k) := P(k | k)$ 
   $\alpha := -1$ 
else
   $P_y(k) := P(k | k)$ 
   $\alpha := 1$ 
end
for  $j = k - L + 1, k - L + 2, \dots, k$  begin
   $P(j | j - 1) := A(j - 1)P(j - 1 | j - 1)A^T(j - 1)$ 
   $\quad + B(j - 1)U(j - 1)B^T(j - 1)$ 
  if  $j = k - \tau_i$  then
     $P(j | j) := \{P^{-1}(j | j - 1) + I(j)$ 
     $\quad + \alpha C_i^T(j)W_i^{-1}(j)C_i(j)\}^{-1}$ 
  else
     $P(j | j) := \{P^{-1}(j | j - 1) + I(j)\}^{-1}$ 
  end
end
if  $i \in S$  then
   $P_y(k) := P(k | k)$ 
else
   $P_{y,y}(k) := P(k | k)$ 
end

```

At this point, both of $P_{y,y}(k)$ and $P_y(k)$ are obtained. Calculate the mutual information according to Eq. (11). \square

This is an algorithm that evaluates just one sensor. Consequently the computation time needed to evaluate multiple sensors linearly depends on the number of the sensors. But this

is not a crucial disadvantage because evaluation of sensors, or sensor selection, is not needed for every step unlike the filtering algorithm.

It should also be noted that the assumption that the delay τ_i and the error information $C_i^T(k - \tau_i)W_i^{-1}(k - \tau_i)C_i(k - \tau_i)$ are known is not needed for DTKF, but needed only for the mutual information calculation. If these are unknown on sensor selection, it will degrade only the accuracy of selection and the result of DTKF is guaranteed to agree exactly with the optimal estimate by the selected sensors.

IV. EXPERIMENTAL RESULTS

We evaluated the proposed algorithms by numerical experiments. A target tracking task by high-speed vision sensors (e.g. [13]) was modeled as follows: The state vector represents the position and velocity of the target $\mathbf{x} \equiv (x, y, z, \dot{x}, \dot{y}, \dot{z})^T$, and its dynamics is given by

$$\begin{aligned} A(k) &= \begin{pmatrix} I_3 & \Delta t I_3 \\ O_3 & 0.99 I_3 \end{pmatrix}, \quad B(k) = I_6, \\ U(k) &= \sigma_u^2 \text{diag}\{0, 0, 0, 1, 1, 1\} \end{aligned}$$

where $\Delta t = 0.001$ [s].

The sensors are assumed to be orthogonal projection cameras that output the target positions in the image-plane coordinates. Each camera is directed randomly, and has its individual observation noise covariance, that is,

$$\begin{aligned} C_i &= (I_2 \quad \mathbf{0}_2) R(\theta_i, \phi_i, \gamma_i) (I_3 \quad O_3) \\ W_i &= \sigma_i^2 I_2 \end{aligned}$$

where $R(\theta, \phi, \gamma)$ is a rotation matrix, and $\theta_i, \phi_i, \gamma_i$ and σ_i are randomly determined for each sensor i .

Each sensor has an individual and time-variant communication delay. The delay when a sensor i transmits its observation is modeled as a white Gaussian random sequence with a distribution $N(\bar{d}_i, \sigma_d^2)$, where \bar{d}_i is the mean delay determined for each i , and σ_d^2 is a constant common for all the sensors.

Firstly, performance of DTKF was evaluated in comparison with the conventional Kalman filter where belated sensory data are dropped. Four different parameter setups were examined. An experiment for each setup contained 50 trials. In each trial, target trajectory and sensor configuration were generated randomly, and the conventional Kalman filter and DTKF were executed to estimate this trajectory for 100 time steps.

For all the setups, we used the parameters in Table I. The parameter σ_i^2 were set in different manners for the four setups:

- Setup 1 Generated to distribute uniformly in interval $(0, 1.0^2)$
- Setup 2 Generated to distribute uniformly in interval $(0, 5.0^2)$
- Setup 3 $\sigma_i^2 = 0.1 \times 1.0^2$ if $\bar{d}_i > L/2$; otherwise $\sigma_i^2 = 0.8 \times 1.0^2$
- Setup 4 $\sigma_i^2 = 0.1 \times 5.0^2$ if $\bar{d}_i > L/2$; otherwise $\sigma_i^2 = 0.8 \times 5.0^2$

The setup 1 and 2 were to simulate situations where the sensors are randomly configured, whereas the setup 3 and 4 were to simulate situations where the tradeoff between the delays and the sensing accuracy is arbitrarily elicited.

Figure 2 shows the mean square errors (MSE) over the 50 trials for the four setups. Indices of the lateral axis represent

TABLE I
PARAMETER SETUP IN DTKF EVALUATION.

parameter	value
# of sensors m	20
maximum delay L	10
σ_d^2	2.0^2
σ_u^2	1.0^2
d_i	uniformly distributed in $(1, L)$

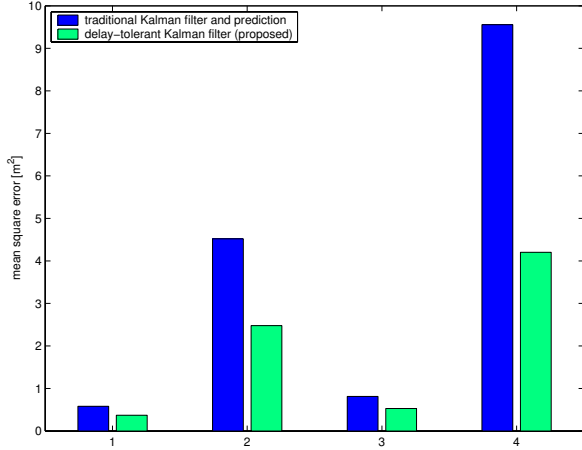


Fig. 2. Performance evaluation of the proposed delay-tolerant Kalman filter.

the four setups. The left side of a group of two bars shows the MSE by the conventional Kalman filter, and the right side shows the MSE by the proposed method. It is clearly shown that the proposed method provides better performance. In particular, the setup 4 where the observation noise is large and arbitrarily given shows significant improvement, presumably because the conventional filter dropped much valuable information from distant but accurate sensors.

Secondly, the proposed sensor selection method was evaluated in comparison with other three criteria for selection: random, mutual information, and length of mean delay. At every fifth time step, the four criteria including the proposed one were computed, and a prescribed number of sensors were selected according to each criterion. Sensory information from the selected sensors was fused by the above proposed DTKF at every time step. We examined four parameter setups shown in Table II. For each setup, 50 trials of 100-step estimation based on the four selection criteria were performed.

Figure 3 shows the MSE over the 50 trials for the four parameter setups. Indices of the lateral axis represent the four setups as in the case with Fig. 2. A group of four bars consists of, starting from left to right, the MSE by random selection, by the mutual information only, by the length of mean delay, and by the proposed method.

As a whole, the proposed method offers good, or at least not bad performance. When the observation noise is large or the number of selected sensor is small in particular, the proposed method is clearly advantageous.

TABLE II
PARAMETER SETUPS IN THE EVALUATION OF SENSOR SELECTION.

setup	1	2	3	4
# of sensors m	20			
# of selected sensors	5			3
maximum delay L	10			
σ_i^2 (interval in which σ_i^2 is uniformly distributed)	$(0, 1.0^2)$	$(0, 5.0^2)$	$(0, 0.2^2)$	$(0, 1.0^2)$
σ_u^2	1.0^2			
σ_d^2	2.0^2			
d_i	uniformly distributed in $(1, L)$			

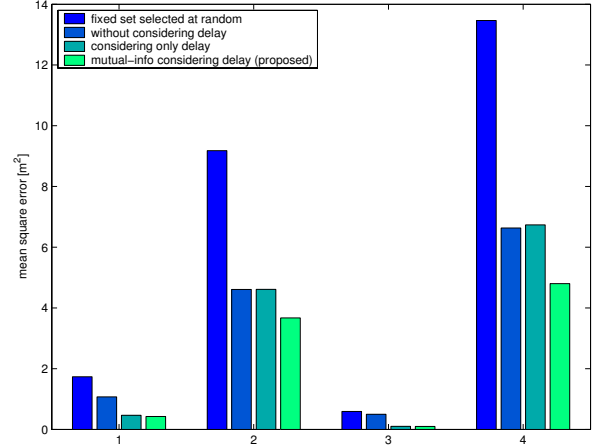


Fig. 3. Performance evaluation of the proposed sensor selection method.

On the other hand, when the observation noise is not so large, the method based simply on the mean delay also provides comparable performance. Let us see closely some samples from the trials to find the reason for this.

Figure 4 shows the profiles of the sensors and the time series of selection by the proposed method in a sample trial among the 50 trials for the setup No. 3, where observation noise is small. The profiles on the left side show the mean delay (blue) and the normalized amount of observation noise (green) of each sensor; the selection time series on the right side shows the set of sensors selected at each time step with the orders of priority for selection. It is observed from this figure that the sensors with small mean delays were preferentially selected. This means the selection by the proposed method is equivalent of the selection based on the mean delay for this particular setup. This behavior can be justified considering that contribution by selecting an accurate sensor is relatively small when observation noise is averagely small.

By contrast, when observation noise is large, a sensor with a short delay is not always attractive and the propose method takes into account both of communication delays and accuracy of sensing as shown in Fig. 5. These results show that the proposed method achieves adaptability to various circumstances.

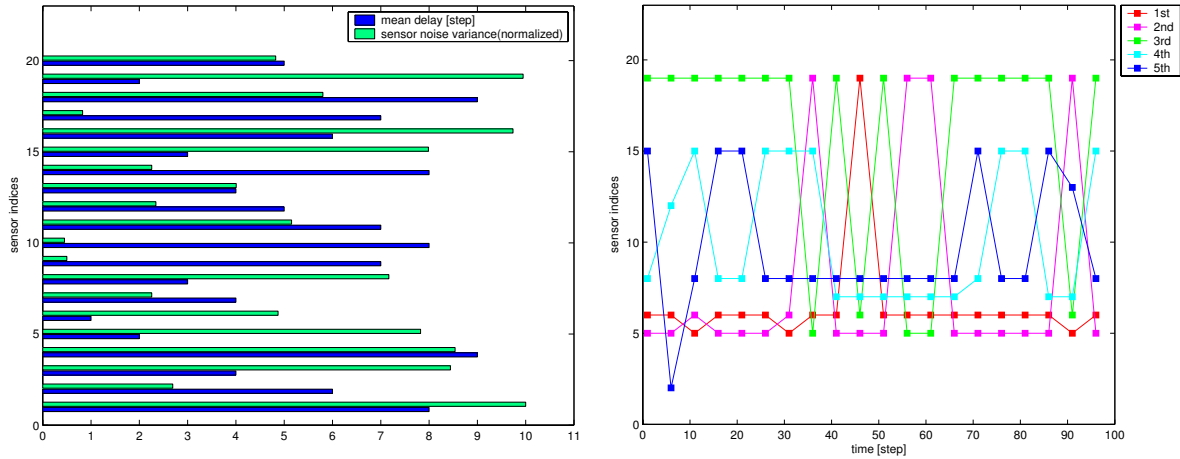


Fig. 4. Profiles of the sensors and the sequence of selection; a sample among the trials for the setup No. 3 (small observation noise).

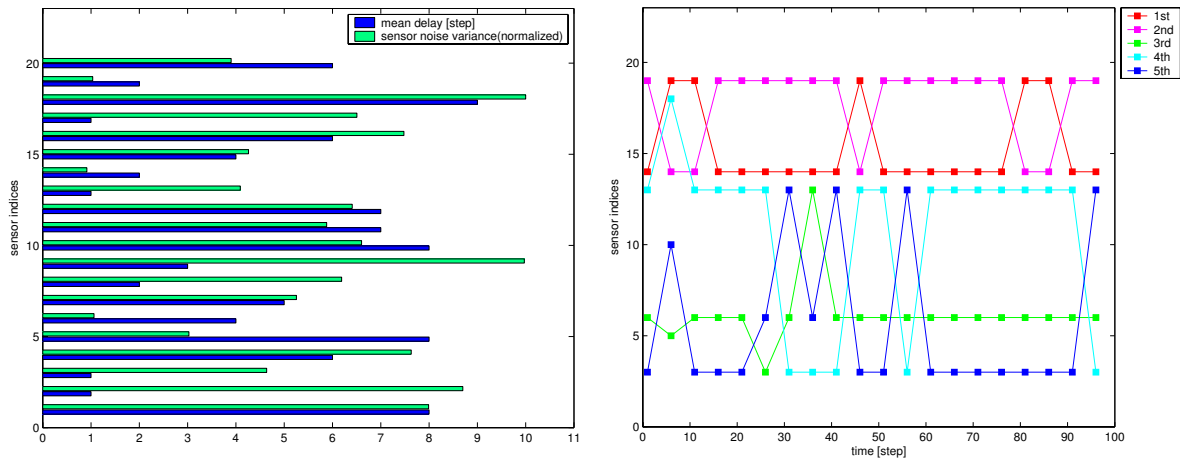


Fig. 5. Profiles of the sensors and the sequence of selection; a sample among the trials for the setup No. 2 (large observation noise).

V. CONCLUSION

In this paper, we proposed a sensor fusion algorithm based on the Kalman filter that permits communication delays and a sensor selection algorithm built on it. Future work will involve decentralizing the algorithms for scalable implementation, and application to real tasks.

REFERENCES

- [1] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.
- [2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *ACM/IEEE Int. Conf. Mobile Computing and Networking*, 1999, pp. 263–270.
- [3] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for "Smart Dust";" in *ACM/IEEE Int. Conf. Mobile Computing and Networking*, 1999, pp. 271–278.
- [4] G. Asada, M. Dong, T. S. Lin, F. Newberg, G. Pottie, W. J. Kaiser, and H. O. Marcq, "Wireless integrated network sensors: Low power systems on a chip," in *European Solid State Circuits Conference*, 1998.
- [5] S. Maeyama, A. Ohya, and S. Yuta, "Non-stop outdoor navigation of a mobile robot – retroactive positioning data fusion with a time consuming sensor system –," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1995, pp. 130–135.
- [6] H. Durrant-Whyte and M. Stevens, "Data fusion in decentralised sensing networks," in *ARO Workshop on Intelligent Systems*, 2000.
- [7] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Int. J. Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.
- [8] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, 1994.
- [9] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Trans. Automatic Control*, vol. 33, no. 1, pp. 88–94, 1988.
- [10] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs: Prentice-Hall, 1979.
- [11] S. Omatu, Y. Tomita, and T. Soeda, "An alternative expression of the mutual information for Gaussian processes," *IEEE Trans. Information Theory*, vol. 22, pp. 593–595, 1976.
- [12] S. Arimoto, *Kalman Filter*. Tokyo: Sangyo Tosho, 1977, (in Japanese).
- [13] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida, "A digital vision chip specialized for high-speed target tracking," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 191–199, 2003.