

*Full Paper*

## **Networked Estimation with an Area-Triggered Transmission Method**

**Vinh Hao Nguyen and Young Soo Suh \***

Department of Electrical Engineering, University of Ulsan, Namgu, Ulsan 680-749, Korea

E-mail: vinhhao@hcmut.edu.vn. E-mail: suh@ieee.org

\* Author to whom correspondence should be addressed. Email: suh@ieee.org

*Received: 18 January 2008 / Accepted: 7 February 2008 / Published: 15 February 2008*

---

**Abstract:** This paper is concerned with the networked estimation problem in which sensor data are transmitted over the network. In the event-driven sampling scheme known as level-crossing or send-on-delta, sensor data are transmitted to the estimator node if the difference between the current sensor value and the last transmitted one is greater than a given threshold. The event-driven sampling generally requires less transmission than the time-driven one. However, the transmission rate of the send-on-delta method becomes large when the sensor noise is large since sensor data variation becomes large due to the sensor noise. Motivated by this issue, we propose another event-driven sampling method called area-triggered in which sensor data are sent only when the integral of differences between the current sensor value and the last transmitted one is greater than a given threshold. Through theoretical analysis and simulation results, we show that in the certain cases the proposed method not only reduces data transmission rate but also improves estimation performance in comparison with the conventional event-driven method.

**Keywords:** Networked estimation, event-driven, level-crossing, send-on-delta.

---

### **1. Introduction**

The traditional way to transmit sensor data in networked control systems is to sample sensor signals equidistant in time. This approach is widely used since analysis and design are simple and inherited from well-established control system theories. If the network speed is high and the traffic is sparse, the periodic sampling approach has many merits because network-induced delay is small and can be

ignored. But when the network bandwidth is limited due to executing tasks of several nodes, time delay becomes large and randomly varying. To avoid these problems, the sensor data transmission rate should be reduced.

Recent works have discussed event-driven alternatives to traditional time-triggered sampling scheme. It has been shown to be more efficient than time-triggered one in some situations, especially in network bandwidth improvement. For instance in [1], the authors provided a comparison of time-triggered impulse control and level-triggered one where the level-triggered scheme gave lower average error for the same average rate of impulse. In [2, 3], an adjustable deadband was defined on each node to reduce network traffic. The node does not broadcast a new message if its signal is within the deadband. A level-crossing sampling scheme in association with an 1-bit coding and decoding strategy to reduce data size was introduced in [4]. This coding sampling scheme becomes highly efficient under data-rate constraints since the nodes only transmit one bit, 0 or 1, per sample. In [5], an optimal level-triggered sampling design problem was proposed in order to minimize the distortion of a filter over a finite horizon. A similar sampling scheme named send-on-delta (SOD) transmission was explored in [6-8]. By adjusting the threshold value at each sensor node, data transmission rate is reduced in order for the network bandwidth to increase and can be used for other traffic. Although the event-driven schemes mentioned above have several names, the basic principle of sampling is the same. That is, in framework of networked control system, a sensor node transmits data only if the difference  $\Delta$  between the current sensor value and the last transmitted one is greater than a given threshold. Hereafter, for ease in representation we refer to them as SOD sampling.

One major issue of SOD sampling method as pointed out in [9] is that it does not detect signal oscillations or steady-state error if the difference  $\Delta$  remains within the threshold range during a long time. Furthermore, in highly noisy systems, the sensor value could exceed the threshold not because the true value exceeds the threshold but because sensor noise exceeds the threshold. In that case, there could be many unnecessary sensor data transmissions.

Motivated by the perspective results of the event-based integral sampling in [9] and the networked estimation problem using SOD transmission method in [7], we propose a novel sampling scheme called area-triggered or send-on-area (SOA) in which sensor data are sent only when the integral of  $\Delta$  is greater than a given threshold. The proposed SOA sampling scheme is slightly different from [9] to improve bandwidth network in case of large noise. Then, a networked estimator based on Kalman filter is formulated to estimate states of the system periodically even when the sensor nodes do not transmit data. Through theoretical analysis and simulation results, we show that in the certain cases the proposed method gives better estimation performance than the SOD method in [7].

## 2. Area-triggered sampling scheme

Consider a networked control system described by the linear continuous-time model:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w(t) \\ y(t) &= Cx(t) + v(t)\end{aligned}\tag{1}$$

where  $x \in R^n$  is the state of the plant,  $u$  is the deterministic input signal,  $y \in R^p$  is the measurement output which is sent to the estimator node by the sensor nodes.  $w(t)$  is the process noise with

covariance  $Q$ , and  $v(t)$  is the measurement noise with covariance  $R$ . We assume that  $w(t)$  and  $v(t)$  are uncorrelated, zero mean white Gaussian random processes.

The area-triggered sampling scheme illustrated in Fig.1b is stated as follows:

- Let  $y_{last,i}$  ( $1 \leq i \leq p$ ) be the last transmitted value of the  $i$ -th sensor output at instant  $t_{last,i}$ . The area formed by the current output  $y_i(t)$  and  $y_{last,i}$  along time:

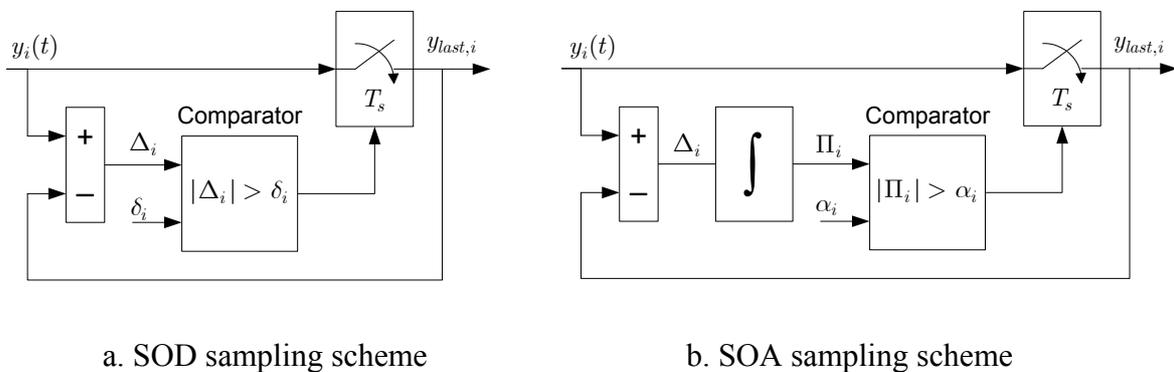
$$\Pi_i(t, t_{last,i}) \triangleq \int_{t_{last,i}}^t (y_i(t) - y_{last,i}) dt. \quad (2)$$

- A new sensor value will be sent to the estimator node if the following condition is satisfied:

$$|\Pi_i(t, t_{last,i})| > \alpha_i \quad (3)$$

where  $\alpha_i$  is the given threshold value at the  $i$ -th sensor node.

Note that in the SOD sampling scheme [7] shown in Fig.1a, a new sensor value will be sent if  $|\Delta_i(t, t_{last,i})| > \delta_i$ , where  $\delta_i$  is the given threshold value.



**Figure 1.** SOD and SOA sampling schemes

### 2.1. Effect of noise on sensor data transmission rate

In this section, we assume  $u(t) = 0$  to investigate effect of noise on the system. The difference between the current sensor value and the last transmitted one is defined by:

$$\begin{aligned} \Delta_i(t, t_{last,i}) &\triangleq y_i(t) - y_{last,i} \\ &= C_i [\Phi(t, t_{last,i}) - I] x(t_{last,i}) + C_i \int_{t_{last,i}}^t \Phi(t, r) w(r) dr + v_i(t) - v_i(t_{last,i}) \end{aligned} \quad (4)$$

where  $\Phi(t, t_{last,i}) = \exp(A(t - t_{last,i}))$ . From (2) we have:

$$\Pi_i(t, t_{last,i}) = \int_{t_{last,i}}^t \Delta_i(t, t_{last,i}) dr. \quad (5)$$

Inserting (4) into (5), we obtain (6):

$$\begin{aligned}
 \Pi_i(t, t_{last,i}) &= \int_{t_{last,i}}^t \Delta_i(t, t_{last,i}) dr \\
 &= \int_{t_{last,i}}^t C_i [\Phi(t, t_{last,i}) - I] x(t_{last,i}) dr \\
 &\quad + \int_{t_{last,i}}^t C_i \int_{t_{last,i}}^u \Phi(t, r) w(r) dr du + \int_{t_{last,i}}^t [v_i(t) - v_i(t_{last,i})] dr \\
 &= \int_{t_{last,i}}^t C_i [\Phi(t, t_{last,i}) - I] x(t_{last,i}) dr \\
 &\quad + \int_{t_{last,i}}^t C_i \left[ \int_r^t \Phi(u, r) du \right] w(r) dr + \int_{t_{last,i}}^t v_i(t) dr - \int_{t_{last,i}}^t v_i(t_{last,i}) dr.
 \end{aligned} \tag{6}$$

Assuming  $x(t_{last,i}) = 0$ , we can derive the variance of  $\Pi_i(t, t_{last,i})$  as follows

$$\begin{aligned}
 E\{\Pi_i \Pi_i'\} &= C_i \int_{t_{last,i}}^t \theta(t, r) w(r) w'(r) \theta'(t, r) dr C_i' + \int_{t_{last,i}}^t 2R(i, i) dr \\
 &= C_i \int_{t_{last,i}}^t \theta(t, r) Q \theta'(t, r) dr C_i' + 2R(i, i) (t - t_{last,i})
 \end{aligned} \tag{7}$$

where  $R(i, i)$  is the  $(i, i)$ -th element of  $R$  and  $\theta(t, r)$  is defined by

$$\theta(t, r) \triangleq \int_r^t \Phi(u, r) du.$$

Note that the variance of  $\Delta_i(t, t_{last,i})$  in the SOD sampling scheme [7] is given by

$$E\{\Delta_i \Delta_i'\} = C_i \int_{t_{last,i}}^t \Phi(t, r) Q \Phi'(t, r) dr C_i' + 2R(i, i). \tag{8}$$

From (7) and (8) we see that both  $E\{\Pi_i \Pi_i'\}$  and  $E\{\Delta_i \Delta_i'\}$  consists of two terms. The first term depends on the process noise  $w(t)$  and the second one depends on the measurement noise  $v(t)$ . Therefore, we believe  $E\{\Pi_i \Pi_i'\}$  (in the case of SOA) plays the same role as  $E\{\Delta_i \Delta_i'\}$  (in the case of SOD) in investigation of data transmission rate. This assumption will be verified in the next section 2.3. In the analysis in [7], we know that data transmission rate is proportional to  $E\{\Delta_i \Delta_i'\}$ . In the highly noisy systems, large  $R$  value makes  $E\{\Delta_i \Delta_i'\}$  in (8) large. It leads to increase data transmission rate when applying the SOD sampling scheme. But with the SOA sampling scheme, we can constrain the effect of  $R$  on  $E\{\Pi_i \Pi_i'\}$  in (7) by lowering the factor  $(t - t_{last,i})$ . This situation can be achieved by choosing the  $\alpha_i$  threshold value such that  $(t - t_{last,i}) \ll 1s$ . In other words, if the time interval of two consecutive sensor data packets is less than one second, then the SOA method can be applied. This is not a so strict condition in networked control systems.

### 2.2. $\Pi_i$ computation and SOA sampling in discrete time at the sensor nodes

Let  $T$  be sampling time of the signals  $y_i(t)$ ,  $y_{k,i}$  ( $k = 1, 2, 3, \dots$ ) be the  $k$ -th sampled value  $y_i(kT)$  from time the  $i$ -th sensor node transmits  $y_{last,i}$ . If the sensor output has no noise,  $y_{k,i}$  lies on the

smooth curve  $C_i x(t)$ . But under the effect of measurement noise,  $y_{k,i}$  will be in the vicinity of  $C_i x(t)$  as illustrated in Fig.2. Supposing that condition (3) is satisfied at  $t = t_{3,i}$ ,  $\Pi_i(t, t_{last,i})$  shown approximately by the slashed area is calculated:

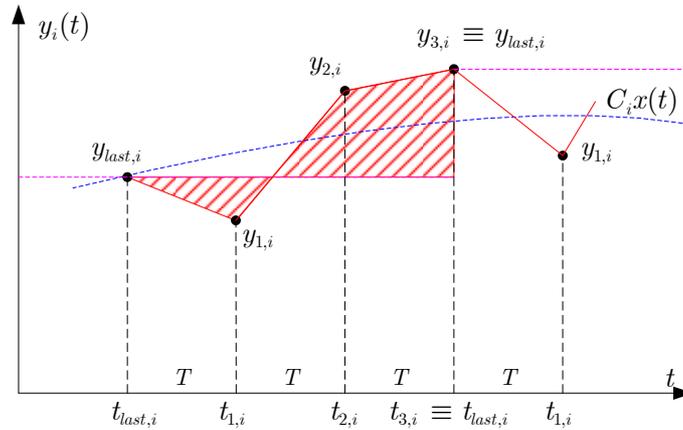


Figure 2. Sensor output with noise in discrete time.

$$\begin{aligned} \Pi_i &= \int_{t_{last,i}}^{t_{1,i}} (y_i(t) - y_{last,i})dt + \int_{t_{1,i}}^{t_{2,i}} (y_i(t) - y_{last,i})dt + \int_{t_{2,i}}^{t_{3,i}} (y_i(t) - y_{last,i})dt \\ &\approx (y_{1,i} - y_{last,i})T/2 + (y_{1,i} + y_{2,i} - 2y_{last,i})T/2 + (y_{2,i} + y_{3,i} - 2y_{last,i})T/2 \end{aligned} \quad (9)$$

The principle of computing the slashed area in (9) is to divide it into several small parts. Each part is a trapezoid with the height  $T$  and two sides  $(y_{k,i} - y_{last,i})$ ,  $(y_{k+1,i} - y_{last,i})$  except for the first part which is a triangle. However, using (9) is inconvenient if condition (3) is not satisfied for a long time interval because the sensor node has to spend much memory on storing  $y_{k,i}$ . An algorithm to calculate  $\Pi_i$  and transmit sensor data in which we only use 3 memory units to store  $(\Pi_i, y_{last,i}, y_{0,i})$  at every period  $T$  is proposed as follows:

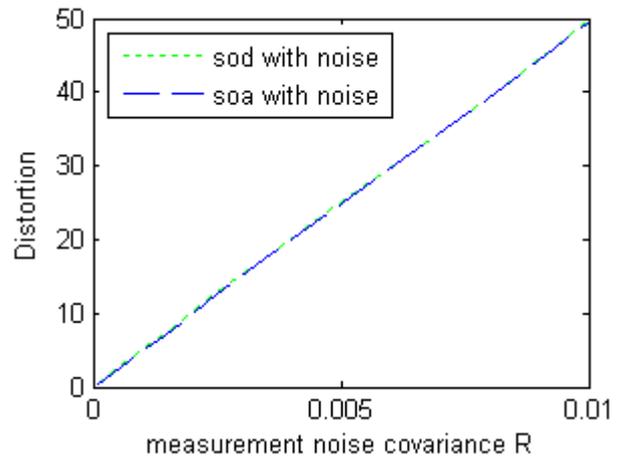
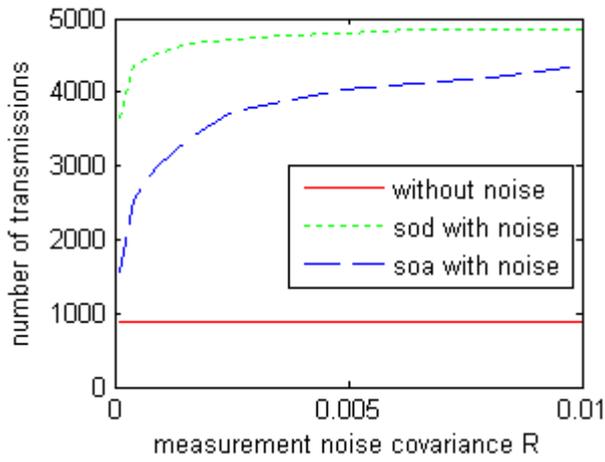
$$\begin{aligned} &\Pi_i = 0; k = 0; y_{0,i} = y_{last,i}; \\ &\text{while } |\Pi_i| < \alpha_i \\ &\quad k = k + 1 \\ &\quad \Pi_i = \Pi_i + (y_{k,i} + y_{0,i} - 2y_{last,i})T/2 \\ &\quad y_{0,i} = y_{k,i} \\ &\text{end while} \\ &y_{last,i} = y_{k,i}; \text{ Transmit } y_{last,i} \end{aligned} \quad (10)$$

### 2.3. Effect of noise on signal distortion

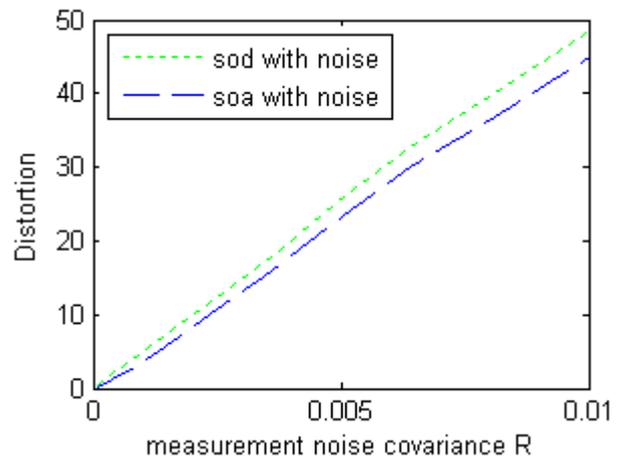
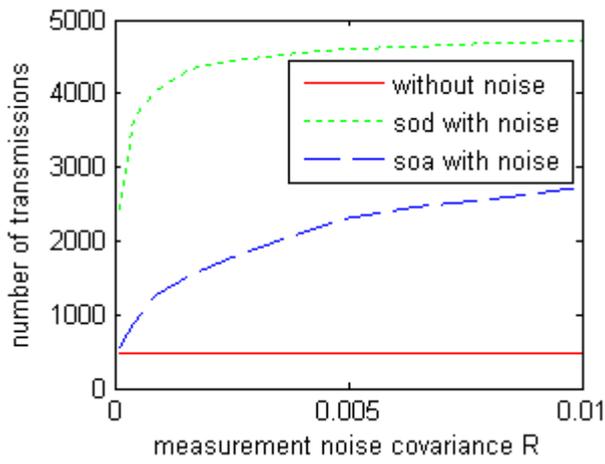
We define a performance index called squared error distortion  $D_i$  for each sensor node:

$$D_i \triangleq \sum_{k=1}^N (y_{r,i}(kT) - y_{last,i}(kT))^2 \quad (11)$$

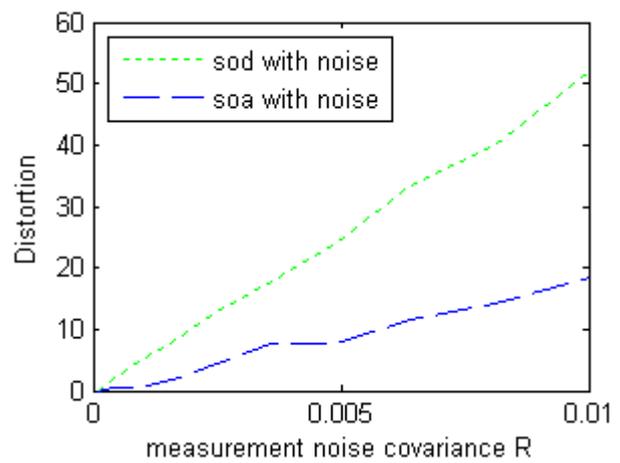
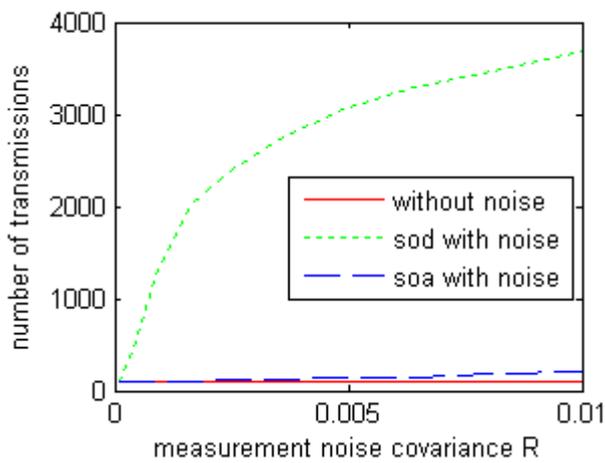
where  $y_{r,i}(kT) = C_i x(kT)$  is the true output value of the  $i$ -th sensor without measurement noise and  $y_{last,i}(kT)$  is the  $i$ -th sensor value received at the estimator node.



a. Case 1:  $\delta = 0.005$ ,  $\alpha = 0.000125$

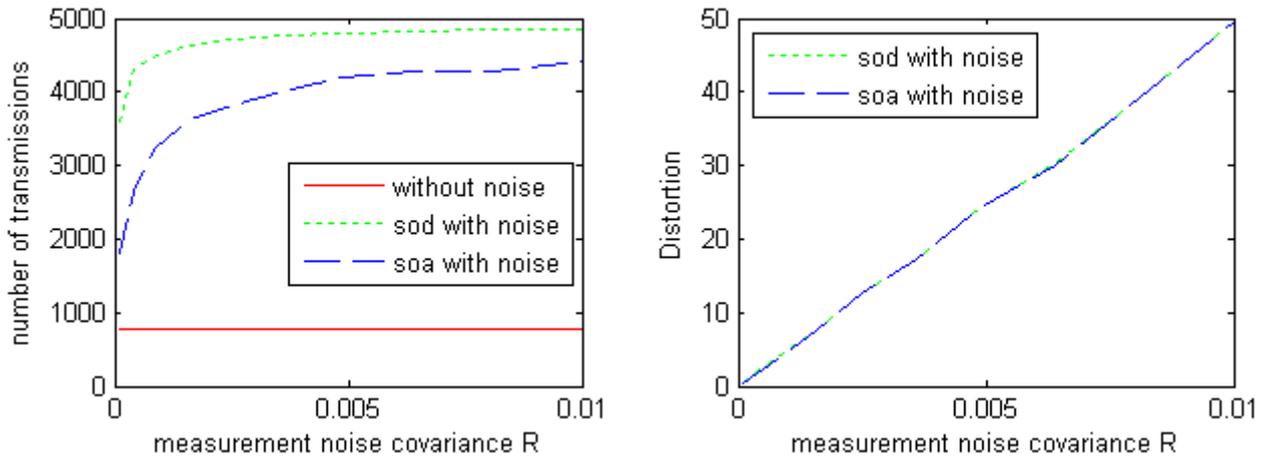


b. Case 2:  $\delta = 0.01$ ,  $\alpha = 0.0005$

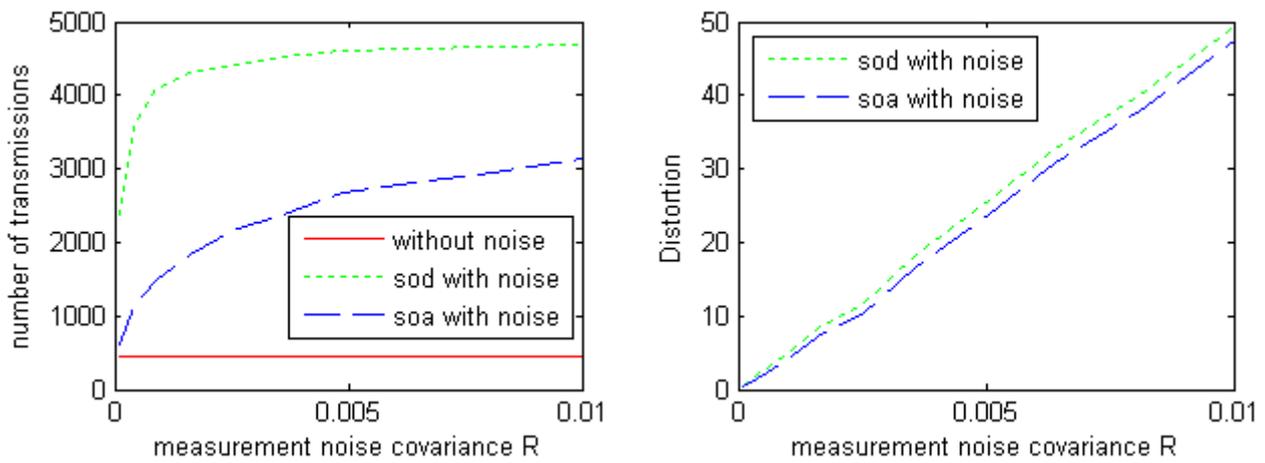


c. Case 3:  $\delta = 0.05$ ,  $\alpha = 0.0125$

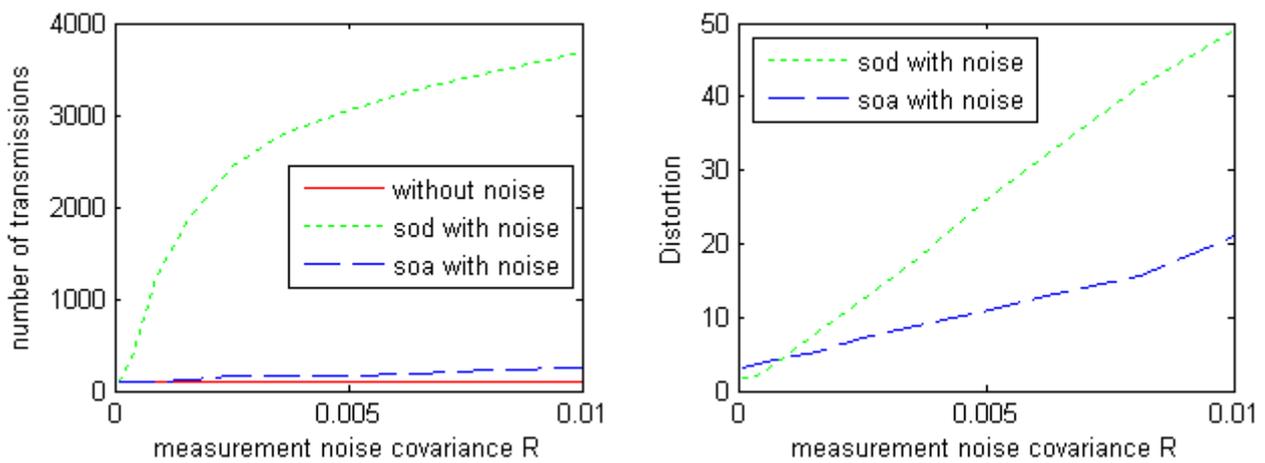
**Figure 3.** Effect of  $R$  on data transmission rate and distortion for  $y(t) = 0.1t + v(t)$ .



a. Case 1:  $\delta = 0.005$ ,  $\alpha = 0.00011$



b. Case 2:  $\delta = 0.01$ ,  $\alpha = 0.00039$



c. Case 3:  $\delta = 0.05$ ,  $\alpha = 0.00870$

**Figure 4.** Effect of  $R$  on data transmission rate and distortion for  $y(t) = 5(1 - e^{-0.1t}) + v(t)$ .

It is very difficult to derive an explicit expression of signal distortion in two methods SOD and SOA by theoretical analysis because it depends not only on the given threshold but also the system model and noise. Thus, in this section we evaluate the effect of measurement noise on distortion by considering an example instead.

Consider two sensor outputs  $y(t) = 0.1t + v(t)$  and  $y(t) = 5(1 - e^{-0.1t}) + v(t)$ , where  $R$  is varying. We will see how data transmission rate and distortion vary as  $R$  varies in both methods SOD and SOA. The evaluation is implemented with  $T = 0.01s$  in 50 seconds. The threshold values  $\delta$  and  $\alpha$  are given in 3 cases:

- Case 1.  $\delta = 0.005$ : small threshold
- Case 2.  $\delta = 0.01$ : medium threshold
- Case 3.  $\delta = 0.05$ : large threshold

In the above example,  $\alpha$  is chosen according to  $\delta$  such that number of data transmissions in two methods is identical as  $v(t) = 0$  (without noise). From the results in the Fig. 3 and Fig. 4, the effect of  $R$  on data transmission rate and distortion is summarized as follows:

- When the threshold value is small as in case 1, distortion of SOA is equivalent to that of SOD but data transmission rate of SOA is smaller than that of SOD. In this case, using SOA will reduce data transmission rate.
- When the threshold value increases as in case 2, distortion of SOA is slightly smaller than that of SOD and data transmission rate of SOA is much smaller than that of SOD. In other words, we get benefit of data transmission rate reduction and a little distortion reduction from SOA method.
- When the threshold value is large as in case 3, SOA reduces not only data transmission rate but also distortion. The larger  $R$  value is, the more reduction is. This result is remarkable! It means that in the SOA method, signal distortion is not degraded along with the reduction of data transmission rate even when system noise is large. Therefore, we hope that estimation performance will be significantly improved when applying a filter.

### 3. State estimation with SOA Transmission method

The networked estimation problem applying SOA transmission method is depicted as follows:

1. Measurement outputs  $y_i$  ( $1 \leq i \leq p$ ) are sampled at the period  $T$  but their data are only transmitted to the estimator node when (3) is satisfied.
2. For simplicity in the problem formulation, transmission delay from the sensor nodes to the estimator node is ignored.
3. The estimator node estimates states of the plant regularly at the period  $T$  regardless of whether or not sensor data arrive. If there is no  $i$ -th sensor data received for  $t > t_{last,i}$ , the estimator node considers that the measurement value of the  $i$ -th sensor output  $y_i(t)$  is still equal to  $y_{last,i}$  but the measurement noise increases from  $v_i(t)$  to  $v_{n,i}(t) = v_i(t) + \Delta_i(t, t_{last,i})$ .

To formulate a state estimation problem, the bound of  $\Delta_i(t, t_{last,i})$  needs to be determined. In the next section, we will compute the bound of  $\Delta_i(t, t_{last,i})$  and then a modified Kalman filter is applied for state estimation.

### 3.1. Bound of $\Delta_i(t, t_{last,i})$ :

At time  $t = t_{1,i}$ , if the estimator node does not receive  $i$ -th sensor data as illustrated in Fig.2, from (3) and (9) we know that:

$$|\Pi_i(t_{1,i}, t_{last,i})| = |(y_{1,i} - y_{last,i})T/2| < \alpha_i, \quad (12)$$

therefore

$$-2\alpha_i/T < (y_{1,i} - y_{last,i}) < 2\alpha_i/T. \quad (13)$$

At time  $t = t_{2,i}$ , if the estimator node does not receive  $i$ -th sensor data yet then:

$$|(y_{1,i} - y_{last,i})T/2 + (y_{1,i} + y_{2,i} - 2y_{last,i})T/2| < \alpha_i, \quad (14)$$

or

$$-2\alpha_i/T - 2(y_{1,i} - y_{last,i}) < (y_{2,i} - y_{last,i}) < 2\alpha_i/T - 2(y_{1,i} - y_{last,i}). \quad (15)$$

Inserting (13) into (15) we obtain:

$$-6\alpha_i/T < (y_{2,i} - y_{last,i}) < 6\alpha_i/T. \quad (16)$$

Computing similarly for instants  $t = t_{k,i}$  ( $k = 3, 4, \dots$ ), we always have the following inequality:

$$-6\alpha_i/T < (y_{k,i} - y_{last,i}) < 6\alpha_i/T. \quad (17)$$

In other words, the bound of  $\Delta_i(t, t_{last,i})$  is given by

$$|\Delta_i(t, t_{last,i})| < 6\alpha_i/T. \quad (18)$$

Note that  $\Delta_i(t, t_{last,i}) = 0$  at time the estimator node receives  $i$ -th sensor data. Then, the process is repeated as (12)-(17) as long as the estimator node does not receive data from the  $i$ -th sensor node yet.

### 3.2. State estimation

Assuming that  $\Delta_i(t, t_{last,i})$  has a uniform distribution with (18), variance of  $\Delta_i(t, t_{last,i})$  will be  $(6\alpha_i/T)^2/3$ . Thus, if there is no  $i$ -th sensor data received for  $t > t_{last,i}$ , variance of measurement noise is increased from  $R(i, i)$  to  $R(i, i) + (6\alpha_i/T)^2/3$ .

A modified Kalman filter for state estimation  $\hat{x}_k$  at step  $k$ , where there is a change in the measurement update part of the discrete Kalman filter algorithm [10], is given as in the Fig.5. We use the discretized system model sampled at period  $T$ :

$$A_d = e^{AT}, \quad B_d = \int_0^T e^{Ar} B dr,$$

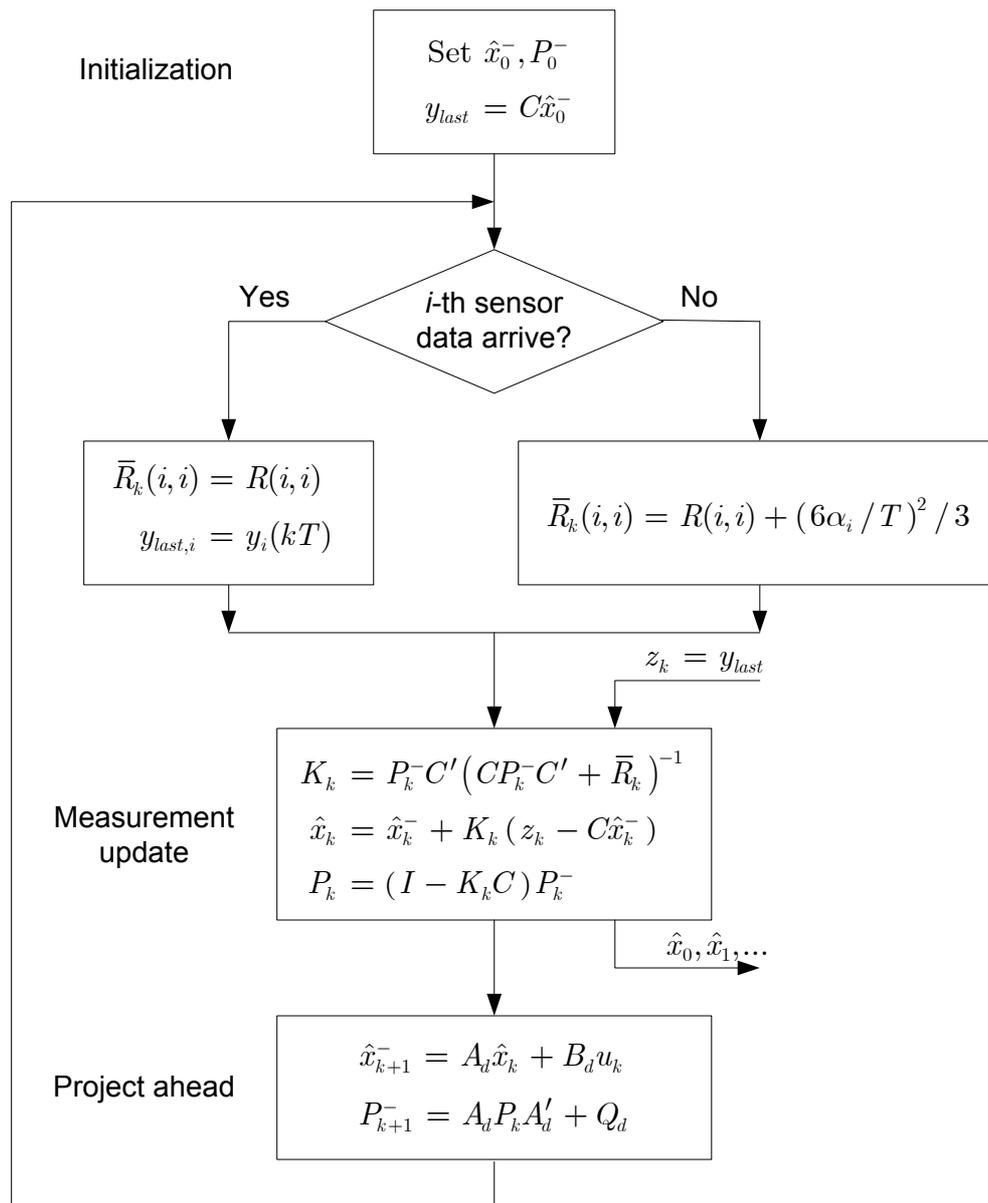
$Q_d$  is the process noise covariance of the discretized system:

$$Q_d = \int_0^T e^{Ar} Q e^{A'r} dr,$$

and  $y_{last}$  is the vector of  $p$  last received sensor values:

$$y_{last} = [y_{last,1} \quad y_{last,2} \quad \dots \quad y_{last,p}]'$$

In the modified Kalman filter in Fig.5, the states of the plant are estimated regularly at every period  $T$  regardless of whether or not sensor data arrive. If  $i$ -th sensor data arrive,  $\Delta_i(t, t_{last,i}) = 0$ , the modified Kalman filter acts like the conventional Kalman filter. Otherwise, it uses  $y_{last,i}$  as the measurement value and  $\bar{R}(i, i) = R(i, i) + (6\alpha_i/T)^2/3$  as measurement noise covariance for state estimation.



**Figure 5.** Structure of the modified Kalman filter.

#### 4. Simulation

To verify the proposed filter, we consider an example of the second-order system with step input where the output is sampled by the SOD and SOA methods:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -1/a & -b/a \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ M/a \end{bmatrix} u(t) + w(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) + v(t)$$

$$Q = 0.01, R = 0.01, T = 10ms$$

The system parameters are given in the following two cases for performance evaluation:

- Case 1. (underdamped system)  $M = 30$ ,  $a = 5$ ,  $b = 1$ .
- Case 2. (undamped system)  $M = 30$ ,  $a = 5$ ,  $b = 0$ .

The simulation process is implemented for 50 seconds. In each case, we use two methods (SOD and SOA) for performance comparison. Estimation performance is evaluated by the average distortion:

$$\bar{D}_i = \frac{1}{N} \sum_{k=1}^N e_{k,i}^2 = \frac{1}{N} \sum_{k=1}^N (x_{k,i} - \hat{x}_{k,i})^2 \quad (19)$$

where  $e_i$  ( $i = 1, 2$ ) is the estimation error,  $x_i$  is the reference state, and  $N = 5000$ .

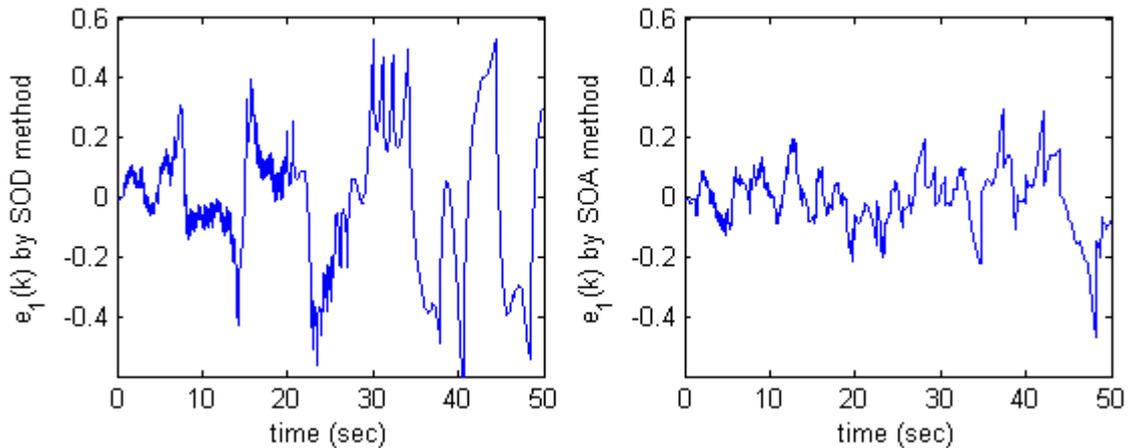
The simulation results with different threshold values for the two cases are shown in Table 1 and Table 2, where  $\alpha$  value is chosen according to  $\delta$  value such that  $n$  (number of sensor data transmissions) is identical in the two methods. In both cases, we see that when  $\delta$  is small (i.e.  $\delta = 0.1, 0.3$ ), estimation performance of SOD and SOA is almost the same. Rigorously speaking, SOD is slightly better than SOA. However, when  $\delta$  is increasing, SOA method shows to be outperform significantly. For example as  $\delta = 0.9$ , SOA outperforms SOD by the  $\bar{D}_1$  reduction of 5 times (i.e. 0.0116 vs. 0.0536 in case 1, and 0.0017 vs. 0.0081 in case 2). As illustrated in Fig. 6 and Fig. 7, the estimation error of SOA is much smaller than that of SOD.

**Table 1.** Estimation performance of 2 methods with different threshold values in case 1

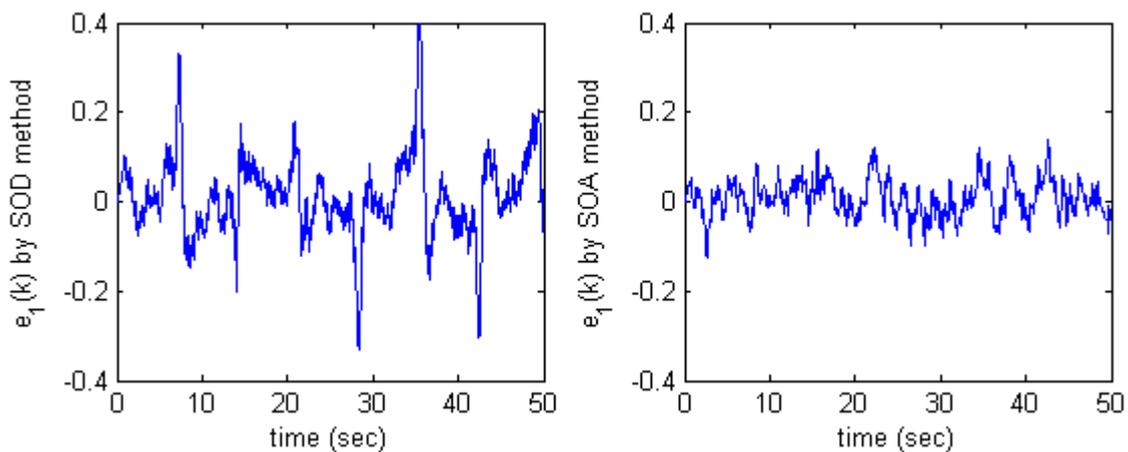
$\delta$	0.1	0.3	0.5	0.7	0.9
$\alpha$	0.0006	0.0081	0.0302	0.0534	0.0881
$n$	2465	490	240	175	140
$\bar{D}_1$ by SOD	6.08e-4	0.0026	0.0101	0.0268	0.0536
$\bar{D}_1$ by SOA	6.40e-4	0.0030	0.0061	0.0103	0.0116
$\bar{D}_2$ by SOD	0.0080	0.0106	0.0176	0.0205	0.0313
$\bar{D}_2$ by SOA	0.0082	0.0110	0.0139	0.0144	0.0148

**Table 2.** Estimation performance of 2 methods with different threshold values in case 2

$\delta$	0.1	0.3	0.5	0.7	0.9
$\alpha$	0.0005	0.0038	0.0094	0.0181	0.0303
$n$	3347	1601	1094	822	658
$\bar{D}_1$ by SOD	3.49e-4	8.94e-4	0.0022	0.0043	0.0081
$\bar{D}_1$ by SOA	3.65e-4	8.32e-4	0.0015	0.0016	0.0017
$\bar{D}_2$ by SOD	0.0052	0.0067	0.0087	0.0109	0.0131
$\bar{D}_2$ by SOA	0.0053	0.0070	0.0079	0.0080	0.0080



**Figure 6.** Estimation error as  $\delta = 0.9$ ,  $\alpha = 0.0881$  in case 1



**Figure 7.** Estimation error as  $\delta = 0.9$ ,  $\alpha = 0.0303$  in case 2.

Through simulation results, we see that the SOA method provides better estimation performance than the SOD method. The key reason is that in the SOD method, noise-containing sensor data not only make transmission rate increase but also degrade estimation performance. Whereas, thanks to the integral block which acts as a noise filter in the SOA sampling scheme, the sensor 1 node does not send highly noisy data, but it transmits data only when the output indeed changes value. For that reason, the transmitted sensor data in the SOA method is more reliable. This helps estimation performance better.

## 5. Conclusion

In this paper, the state estimation problem with SOA transmission method over network has been considered. We have shown that when increasing the threshold value to improve bandwidth network, the SOA method gives much better estimation performance than the SOD method. This is very useful in the realistic applications where sensor data transmission rate needs to be lowered due to joining of many sensor nodes or for power saving in wireless networks. The SOA method has been also proven to be more efficient than the SOD method in the highly noisy systems.

## Acknowledgements

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R01-2006-000-11334-0). The authors also would like to thank Ministry of Commerce, Industry and Energy and Ulsan Metropolitan City which partly supported this research through the Network-based Automation Research Center (NARC) at University of Ulsan.

## References and Notes

1. Astrom, K.J.; Bernhardsson, B.M. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings 41st Conference on Decision and Control*; Las Vegas, U.S.A., 2002; pp 2011–2016.
2. Otanez, P.G.; Moyne, J.R.; Tilbury, D.M. Using deadbands to reduce communications in networked control systems. In *Proceedings of the American Control Conference*; Anchorage, U.S.A., 2002; pp 3015-3020.
3. Sandra, H.; Peter, H.; Eckehard, S.; Martin, B. Network Traffic Reduction in Haptic Telepresence Systems by Deadband Control. In *Proceedings IFAC World Congress*; International Federation of Automatic Control: Prague, Czech Republic, 2005.
4. Kofman, E.; Braslavsky, J.H. Level Crossing Sampling in Feedback Stabilization under Data-Rate Constraints. In *Proceedings 45th Conference on Decision and Control*; San Diego, U.S.A., 2006; pp 4423-4428.
5. Rabi, M.; Moustakides, G.V.; Baras, J.S. Multiple Sampling for Estimation on a Finite Horizon. In *Proceedings 45th Conference on Decision and Control*; San Diego, U.S.A., 2006; pp 1351-1357.
6. Mikowicz, M. Send-On-Delta Concept: An Event-Based Data Reporting Strategy. *Sensors* **2006**, *65*, 49-63.
7. Suh, Y.S.; Nguyen, V.H.; Ro, Y.S. Modified Kalman filter for networked monitoring systems employing a send-on-delta method. *Automatica* **2007**, *43*(2), 332-338.
8. Suh, Y.S. Send-On-Delta sensor data transmission with a linear predictor. *Sensors* **2007**, *7*, 537-547.
9. Mikowicz, M. Asymptotic Effectiveness of the Event-Based Sampling According to the Integral Criterion. *Sensors* **2007**, *7*, 16-37.
10. Brown, R.G.; and Hwang, P.Y.C. *Introduction to Random Signals and Applied Kalman Filtering*; John Wiley & Sons: New York, 1997.
11. Astrom, K. J. *Introduction to Stochastic Control Theory*; Academic Press: New York, 1970.