

Full Paper

Send-On-Delta Sensor Data Transmission With A Linear Predictor

Young Soo Suh

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

E-mail: yssuh@uou2.ulsan.ac.kr

Received: 3 April 2007 / Accepted: 24 April 2007 / Published: 26 April 2007

Abstract: This paper is concerned with sensor data transmission strategy. The main focus of the paper is how to reduce the number of sensor data transmission while maintaining the difference between the estimated sensor value and the real sensor value. The proposed method could be used in sensor networks and networked control systems, where number of transmission should be minimal. A linear predictor is used to predict sensor values and sensor data are transmitted if the difference between the predicted sensor value and the real sensor value exceeds the specified limit. An analytic upper bound of the mean rate of messages is provided. Through simulation, it is shown that the number of transmission could be significantly reduced compared with the periodic sampling and the conventional send-on-delta method.

Keywords: data acquisition, signal sampling, linear prediction

1. Introduction

Periodic sampling of sensor data is used in many applications since it is easy to implement and there are well-established system theory for periodically sampled signals [1].

However, there are applications, where periodic sampling is not desirable. In networked control systems [2, 3], several sensors are connected through a serial network. Sometimes only limited bandwidth is allowed for sensor data transmission; for example, the network physical bandwidth may be limited or most of a network traffic could be used by other control/monitoring traffic. In this case, periodic sampling is not suitable since transmission rate of the period sampling is generally high.

Another example is a wireless sensor networks [4], where a sensor node is battery-powered and the

battery power should be saved for longer operation time. As experimented in [5], wireless transmission consumes significantly more power compared with internal computation. Thus to reduce the battery consumption, a sampling method, which requires less transmission, is desirable.

As a transmission frequency reducing method, an event-driven sampling is a gaining popularity. In an event-driven sampling, a sampling is taken if an event happens: for example, an event could be a sensor value change exceeding the specified limit. This event-driven sampling has many names: magnitude-driven sampling [6] a deadzone method [7], a send-on-delta method [8, 9] and Lebesgue sampling [10].

Despite many names, the basic principle is the same. If we explain in the framework of a networked monitoring system, a sensor data is transmitted when the difference between last transmitted value and the current value is larger than the specified limit. When a signal change is relatively small, number of sensor data transmission is significantly less than that of the periodic sampling method. This event-driven sampling is used in many intelligent sensors and also a standard feature in Lon Network [11], which is mainly used in the Building Automation.

In this paper, we propose a transmission algorithm for intelligent sensors, which further reduces the number of transmission. In the algorithm, a sensor value is transmitted if difference between the current value and the predicted value is larger than the specified limit. The predicted value is computed using the past values. In fact, the send-on-delta method could be considered as a special case of the proposed algorithm in the sense that the predicted value is just the last transmitted value. In the send-on-delta method, the number of transmission is small when the signal does not change; that is when the first derivative is almost constant.

In the proposed algorithm, the number of transmission is reduced when the higher order (2nd and 3rd) derivative is almost constant. It is shown that the proposed algorithm reduces the number of transmission compared with the conventional send-on-delta methods.

The paper is organized as follows. In Section 2, we propose a new algorithm which combines send-on-delta concept with a linear predictor. In Section 3, we investigate the mean rate of messages for the proposed algorithm. We provide an explicit upper bound for the mean rate of messages. In Section 4, simulations are done for four signals so that we can compare the periodic sampling method, the send-on-delta method, and the proposed method. Conclusion is given in Section 5.

2. Send-on-delta transmission with a linear predictor

The conventional send-on-delta method [8, 9] is illustrated in Fig. 1. The sensor value is transmitted when the difference between the current value $x(t)$ and the last transmitted value $x_{last}(t)$ is larger than specified δ . In this method, the error $(x(t) - x_{last}(t))$ in the monitoring station (where sensor data are received) is always smaller than δ assuming there is no delay and packet loss during transmission.

We propose a new algorithm (see Fig. 2), which combines the send-on-delta concept and a linear predictor. After the sensor value transmission, a linear predictor computes the future sensor value \hat{x} based on the past values. If the difference between the current value and the predicted value is larger than δ , the sensor value is transmitted. Note that \hat{x} is computed both in the sensor node and monitoring station. Also note that the error $(x(t) - \hat{x}(t))$ in the monitoring station is always smaller than δ , which is the same estimation performance with the conventional send-on-delta method. However the number of

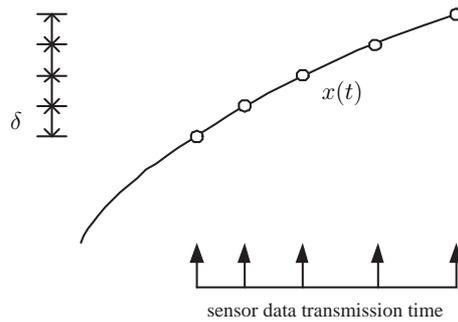


Figure 1. Conventional send-on-delta transmission

transmission is usually smaller than that of the conventional send-on-delta method.

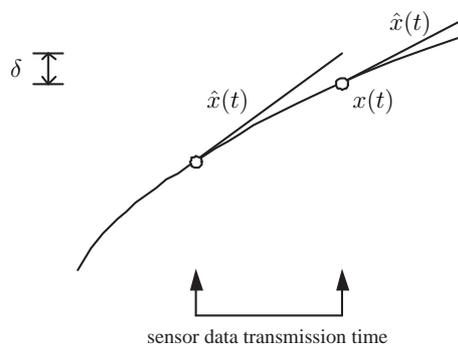


Figure 2. Send-on-delta transmission with a linear predictor

2.1. New send-on-delta algorithm with a linear predictor

The proposed algorithm is given in Fig. 3 and Fig. 4. In the sensor node, it is assumed that the sensor node is sampled with the period T . A discrete-time signal x_k is defined by $x_k = x(kT)$. Even if the sensor value is sampled with the period T , the actual transmission rate is not T since not all sampled sensor data are transmitted.

In the sensor node, $\hat{x}_k = f(\hat{x}_{k-1}, \dots, \hat{x}_{k-M-1})$ indicates a linear predictor, where M denotes the memory length. If $M = 1$, \hat{x}_k is computed based on \hat{x}_{k-1} and \hat{x}_{k-2} . How to choose a linear predictor $f(\cdot)$ is discussed later. Note that we use $f(\hat{x}_{k-1}, \dots, \hat{x}_{k-M-1})$ instead of $f(x_{k-1}, \dots, x_{k-M-1})$. If we used $f(x_{k-1}, \dots, x_{k-M-1})$, we would have obtained more accurate \hat{x}_k ; however in that case we have to transmit x_k to the monitoring station at every T seconds. Thus we cannot reduce the number of transmission.

If the difference between the current value x_k and the predicted value \hat{x}_k is larger than δ , we transmit x_k, \dots, x_{k-M} instead of just transmitting x_k . Why we have to transmit x_k, \dots, x_{k-M} is discussed in Section 4. In the conventional send-on-delta method, we only need to transmit x_k . In this sense, the transmission data size is larger than that of the conventional send-on-delta method. Despite this, we will see later in Section 4 that the overall network overhead of the proposed method is significantly smaller than the conventional send-on-delta method.

In the monitoring station, when sensor values are not received, we predict the current sensor value using a linear predictor $\hat{x}_k = f(\cdot)$. From the transmitter algorithm, it is guaranteed that the error between x_k and \hat{x}_k is smaller than δ .

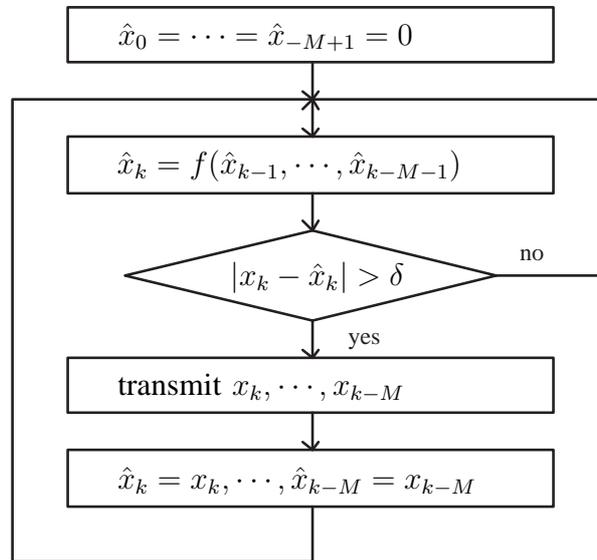


Figure 3. Transmitter Algorithm in Sensor Nodes ($x_k \triangleq x(kT)$)

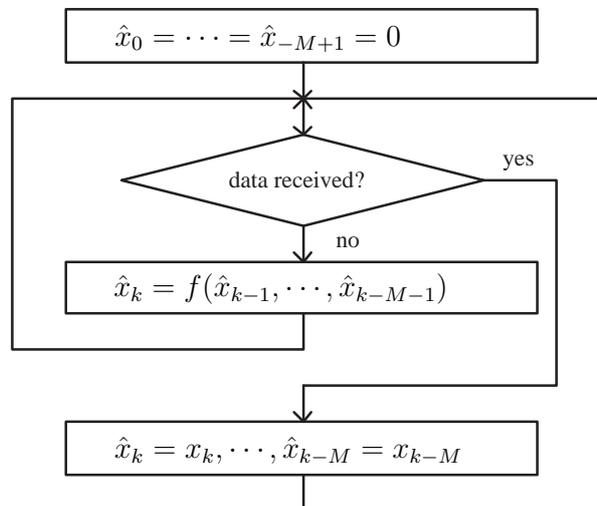


Figure 4. Receiver Algorithm in the Monitoring Station

2.2. Linear predictor

If a signal is a random process with a known distribution, we can obtain an optimal predictor [12]. In this paper, we assume that we have no information about the signal which we measure.

To derive a predictor equation, consider the Taylor expansion of a signal:

$$x(kT) = x((k-1)T) + \dot{x}((k-1)T)T + \ddot{x}((k-1)T)\frac{T^2}{2} + \dots \quad (1)$$

If we take two terms from the right hand sides of (1) and apply Euler approximation ($\dot{x}(kT) \approx (x(kT) - x((k-1)T))/T$), we obtain the following:

$$\begin{aligned} x(kT) &\approx x((k-1)T) + \dot{x}((k-1)T)T \\ &\approx x((k-1)T) + \frac{x((k-1)T) - x((k-2)T)}{T}T \\ &= 2x((k-1)T) - x((k-2)T). \end{aligned} \quad (2)$$

This is a first order predictor for $x_k = x(kT)$; this corresponds to $M = 1$ in the algorithm of Fig. 3 and Fig. 4. Similarly, we can obtain a second order predictor for $x_k = x(kT)$.

$$\begin{aligned} x(kT) &\approx x((k-1)T) + \dot{x}((k-1)T)T + \ddot{x}((k-1)T)\frac{T^2}{2} \\ &\approx 2.5x((k-1)T) - 2x((k-2)T) + 0.5x((k-3)T), \end{aligned} \quad (3)$$

where we used the following approximation

$$\begin{aligned} \ddot{x}((k-1)T) &\approx \frac{\dot{x}((k-1)T) - \dot{x}((k-2)T)}{T} \\ &\approx \frac{x((k-1)T) - x((k-2)T)}{T^2} - \frac{x((k-2)T) - x((k-3)T)}{T^2}. \end{aligned}$$

We could go further to obtain higher order predictors. However, as we will see in Section 4, the use of higher of predictors does not reduce the number of transmission rate. Thus in this paper, we will mainly use the first order predictor (2) as a linear predictor $f(\cdot)$.

3. Mean rate of messages analysis

In this paper, we compute the mean rate of messages (λ). The mean rate of messages is defined as the mean number of transmission per a unit of time. We will show that the mean rate of messages (λ) is dependent on δ and $\bar{\ddot{x}}$, which is the time average of $\ddot{x}(t)$.

To derive the mean rate of messages, we use the famous mean-value-theorem [13].

Theorem 1 (Mean-Value Theorem) Assume that x has a derivative at each point of an open interval (a, b) , and assume also that x is a continuous function at both endpoints a and b . Then there is a point c in (a, b) such that

$$\dot{x}(c) = \frac{x(b) - x(a)}{b - a}.$$

From Theorem 1, there exists α_k ($kT < \alpha_k < (k+1)T$) satisfying

$$\dot{x}(\alpha_k) = \frac{x((k+1)T) - x(kT)}{T}. \quad (4)$$

An example is given in Fig. 5. It is possible that α_k is not unique; in that case, we can choose any α_k .

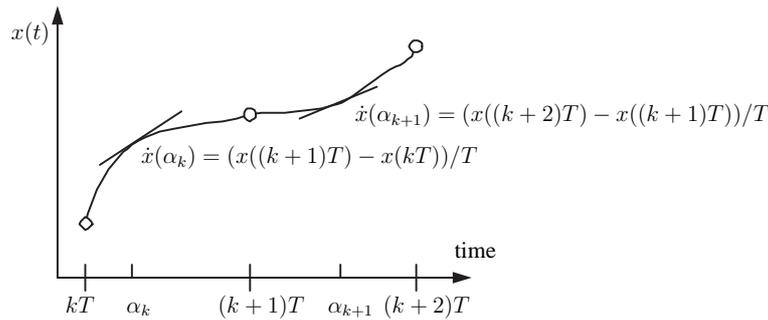


Figure 5. Mean-value-theorem and α_k definition

Assume that the sensor data is transmitted at kT ; recall that x_k and x_{k-1} are transmitted for the first order predictor. Thus it is satisfied that $\hat{x}_k = x_k$ and $\hat{x}_{k-1} = x_{k-1}$.

First we investigate $E\{|x_{k+N} - \hat{x}_{k+N}|\}$ so that we can see how often $|x_{k+N} - \hat{x}_{k+N}|$ exceeds δ . From the definition of α_k in (4), we have

$$x((k + 1)T) = x(kT) + T\dot{x}(\alpha_k). \tag{5}$$

Inserting the following relationship into (5)

$$\begin{aligned} \dot{x}(\alpha_k) &= \dot{x}(\alpha_{k-1}) + \int_{\alpha_{k-1}}^{\alpha_k} \ddot{x}(r) dr \\ &= \frac{x(kT) - x((k-1)T)}{T} + \int_{\alpha_{k-1}}^{\alpha_k} \ddot{x}(r) dr, \end{aligned} \tag{6}$$

we obtain

$$\begin{aligned} x((k + 1)T) &= 2x(kT) - x((k - 1)T) + T \int_{\alpha_{k-1}}^{\alpha_k} \ddot{x}(r) dr \\ &= 2\hat{x}(kT) - \hat{x}((k - 1)T) + T \int_{\alpha_{k-1}}^{\alpha_k} \ddot{x}(r) dr \\ &= \hat{x}((k + 1)T) + T \int_{\alpha_{k-1}}^{\alpha_k} \ddot{x}(r) dr. \end{aligned} \tag{7}$$

In the above equation, we have used the fact that $\hat{x}_k = x_k$ and $\hat{x}_{k-1} = x_{k-1}$. If we proceed to the next time step, we obtain

$$\begin{aligned} x((k + 2)T) &= x((k + 1)T) + T\dot{x}(\alpha_{k+1}) \\ &= x((k + 1)T) + T \left(\dot{x}(\alpha_k) + \int_{\alpha_k}^{\alpha_{k+1}} \ddot{x}(r) dr \right) \\ &= 2x((k + 1)T) - x(kT) + T \int_{\alpha_k}^{\alpha_{k+1}} \ddot{x}(r) dr \\ &= 2 \left(\hat{x}((k + 1)T) + T \int_{\alpha_{k-1}}^{\alpha_k} \ddot{x}(r) dr \right) - \hat{x}(kT) + T \int_{\alpha_k}^{\alpha_{k+1}} \ddot{x}(r) dr. \\ &= \hat{x}((k + 2)T) + 2T \int_{\alpha_{k-1}}^{\alpha_k} \ddot{x}(r) dr + T \int_{\alpha_k}^{\alpha_{k+1}} \ddot{x}(r) dr. \end{aligned} \tag{8}$$

We used (7) in the fourth row derivation.

Continuing the process, we can obtain a general form as follows:

$$x((k + N)T) = \hat{x}((k + N)T) + T \sum_{i=1}^N \left((N - i + 1) \int_{\alpha_{k+i-2}}^{\alpha_{k+i-1}} \ddot{x}(r) dr \right). \tag{9}$$

Note that the time average of $\alpha_k - \alpha_{k-1}$ is T , which can be verified from the following relationship:

$$\begin{aligned} \lim_{L \rightarrow \infty} \frac{1}{L} \sum_{i=1}^L (\alpha_i - \alpha_{i-1}) &= \lim_{L \rightarrow \infty} \frac{1}{L} ((L - 1)T + (T - \alpha_0) + (\alpha_L - (L - 1)T)) \\ &= T. \end{aligned}$$

Noting that

$$E\left\{\int_{\alpha_{k+i-2}}^{\alpha_{k+i-1}} \ddot{x}(r) dr\right\} = T\bar{\ddot{x}}(t)$$

we have the average value of $|x((k+N)T) - \hat{x}((k+N)T)|$ as follows

$$E\{|x((k+N)T) - \hat{x}((k+N)T)|\} = \left|T^2 \sum_{i=1}^N (N-i+1)\bar{\ddot{x}}\right| \leq T^2|\bar{\ddot{x}}| \sum_{i=1}^N (N-i+1) = T^2|\bar{\ddot{x}}| \sum_{i=1}^N i \quad (10)$$

where $|\bar{\ddot{x}}|$ is the time average of $|\ddot{x}(t)|$.

Let N_{max} be the maximum value satisfying (T is given)

$$T^2|\bar{\ddot{x}}| \sum_{i=1}^{N_{max}} i < \delta. \quad (11)$$

Then λ , the mean rate of messages, satisfies the following:

$$\lambda \leq \frac{1}{N_{max}T}. \quad (12)$$

For three different algorithms (the periodic sampling, the conventional send-on-delta method, the proposed method), mean rates of messages are compared in Table 1. For all three algorithms, the error in the remote station is less than δ ; that is, the estimation performance is the same. Thus smaller λ implies that the estimation performance can be achieved with less frequent transmission of sensor data.

Table 1. Mean rate of messages comparison for 3 different algorithms

algorithm	λ
periodic sampling [8]	$\lambda = \frac{\max \dot{x}(t)}{\delta}$
conventional send-on-delta method [8]	$\frac{\bar{\dot{x}}}{\delta} - 2\nu \leq \lambda \leq \frac{\bar{\dot{x}}}{\delta}$
proposed method	$\lambda \leq \frac{1}{N_{max}T}$

ν : average number of signal peaks (maxima and minima) in a time unit [8]

N_{max} is given in (11).

We can see that in Table 1 that λ of the proposed method depends on $\bar{\ddot{x}}$ while λ is dependent $\max \dot{x}$ (periodic sampling) and $\bar{\dot{x}}$ (send-on-delta method). For slowly varying signals, we have the following relationships in general:

$$\bar{\ddot{x}} \ll \bar{\dot{x}} \ll \max \dot{x}.$$

Thus we can expect that the proposed method has the smallest number of transmission. This expectation is verified in the next section.

4. Simulation

To verify the proposed algorithm, we applied the proposed algorithm to four control signals, which are typical control system responses. The signals are given in Table 2, where one is a step response of a

Table 2. Four signals for the number of transmission test

type of signal	signal
step response of the first-order system	$x_1(t) = 1 - e^{-t}$
step response of the second-order underdamped system	$x_2(t) = 1 - (1 + t)e^{-t}$
step response of the second-order underdamped system	$x_3(t) = 1 + \frac{T_1}{T_2 - T_1}e^{-t/T_1} + \frac{T_2}{T_2 - T_1}e^{-t/T_2}$
step response of the second-order undamped system	$x_4(t) = 1 - \cos(\omega_n t)$

$$(T_1 = 1, T_2 = 5/7, \omega_n = \pi/10)$$

first-order system and the others are step responses of second-order systems. These signals are used in [8] and we used the same signals to compare effectiveness of three algorithms.

Mean rates of messages (λ) in the case of $\delta = 0.02$ are given in Table 3, where λ is computed using equations in Table 1. For the send-on-delta method and the proposed method, upper bounds of λ are used. Note that the estimation performance for three algorithms are the same in the sense that the worst case error is δ . Thus small λ means that we can achieve the same estimation performance with small number of sensor data transmission.

Table 3. mean rate of messages (λ) comparison with $\delta = 0.02$

signal	periodic sampling λ	send-on-delta method upper bound of λ	proposed method (1st order) upper bound of λ
$x_1(t)$	50.00	9.96	2.22
$x_2(t)$	18.39	9.57	1.88
$x_3(t)$	350.00	59.97	5.88
$x_4(t)$	15.70	9.99	1.25

As can be seen in Table 3, λ of the send-on-delta method is significantly smaller than that of the periodic sampling. And λ of the proposed method is significantly smaller than that of the send-on-delta method. Thus we could expect that the number of sensor data transmission is the smallest if we use the proposed method.

Actual number of sensor transmission is given in Table 4 for the time interval $[0, 5]$ seconds. We chose the final time 5 seconds since the settling times of x_1 , x_2 , and x_3 are all 5 second. For the oscillating signal x_4 , 5 second corresponds to a quarter cycle.

As expected from Table 3, the proposed method has the smallest number of transmission. We used both the first-order predictor and the second-order predictor. Interestingly, there is almost no reduction of transmission number when we use the second-order predictor. For x_1 , x_2 , and x_4 , the numbers of transmission are the same. There is just one transmission reduction for x_3 . Thus we believe that the first-order predictor is good enough for most signals.

To see how the send-on-delta method and the proposed method work, x_1 plots are given in Fig. 6. The left plot is the result of the send-on-delta method and the right plot is the result of the proposed method.

Table 4. Number of sensor data transmission ($0 \leq t \leq 5, \delta = 0.02$)

signal	periodic sampling	send-on-delta method	proposed method (1st order)	proposed method (2nd order)
$x_1(t)$	250	44	11	11
$x_2(t)$	919	46	9	9
$x_3(t)$	1750	175	25	24
$x_4(t)$	78	48	7	7

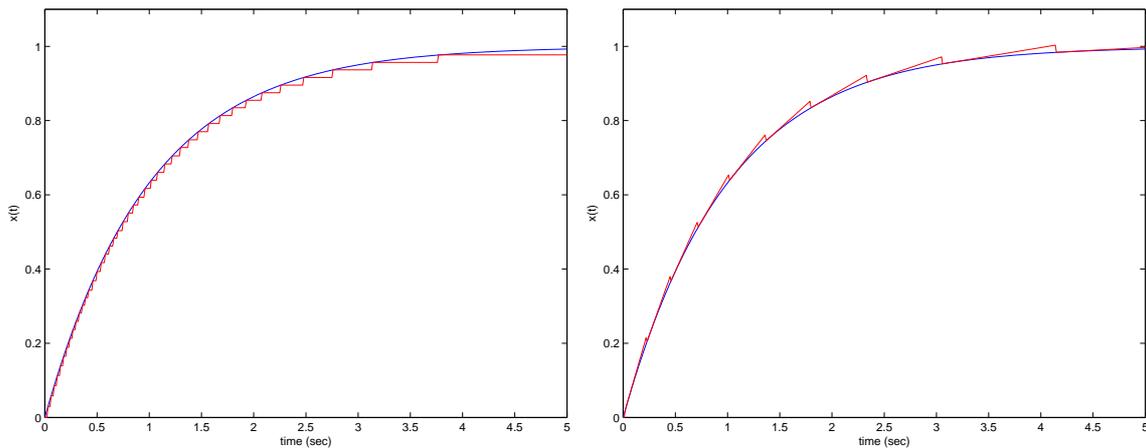


Figure 6. $x_1(t)$ plot with the send-on-delta method (left) and the proposed method (right)

Note that in the proposed method, we have to transmit x_k and x_{k-1} , while only x_k is transmitted in the periodic sampling and the send-on-delta method. The impact of this increase on the overall network transmission is small since a packet-based transmission is used in most networks. For example, in CAN 2.0A network [14], the packet overhead is at least 65 bits and in ZigBee [15], it is at least 120 bits.

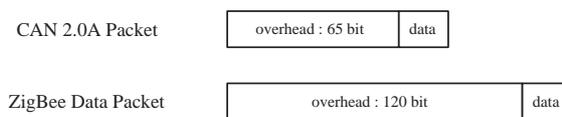


Figure 7. Packet Overhead for CAN 2.0A and ZigBee

Assuming x_k is encoded in 8 bits, total numbers of transmitted bits can be computed as in Table 5.

Using the formulas in Table 5, we can compute the total number of transmitted bits, which is given in Table 6. We can verify that the proposed method transmits much smaller number of bits despite the fact that both x_k and x_{k-1} should be transmitted.

Finally we note that it is a bad idea transmitting only x_k instead of transmitting x_k and x_{k-1} in the proposed algorithm. In most cases, a predictor output would be oscillating and thus the number of transmission becomes very large. An example is given in Fig. 8. We can see the oscillation phenomenon and the number of transmission is 250 while only 11 for the proposed method.

Table 5. Total number of transmitted bits computation formula

	send-on-delta method	proposed method
CAN 2.0A	$65 + 8 \times \#$	$65 + 2 \times 8 \times \#$
ZigBee	$120 + 8 \times \#$	$120 + 2 \times 8 \times \#$

: number of sensor data transmission

Table 6. Total number of transmitted bits

signal	CAN 2.0A		ZigBee	
	send-on-delta	proposed	send-on-delta	proposed
x_1	417	241	472	296
x_2	433	209	488	264
x_3	1465	465	1520	520
x_4	449	177	504	232

5. Conclusion

In this paper, we have proposed a new sensor data transmission algorithm, which reduces the number of transmission. A linear predictor is used to predict the sensor value and the sensor data are transmitted if the difference between the predicted sensor data value and the current sensor value exceeds specified δ . With little increase in the computational burden in the sensor nodes, we can significantly reduce the number of transmission. We believe the proposed algorithm is useful for sensor networks (for battery power saving) and networked control systems (when a bandwidth for sensor data is limited).

Acknowledgements

This work was supported by grant No. R01-2006-000-11334-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

References

1. Oppenheim, A. V.; Schaffer, R. W.; Buck, J. R. *Discrete-Time Signal Processing*. Prentice Hall, 2nd edition edition, **1999**.
2. Walsh, G. C.; Ye, H. Scheduling of networked control systems. *IEEE Control Systems Magazine* **2001**, *21*(1), 57–65.
3. Yang, T. C. Networked control system: a brief survey. *IEE Proc.-Control Theory Appl.* **2006**, *153*(4), 403–412.
4. Zhao, F.; Guibas, L. *Wireless Sensor Networks*. Elsevier, **2004**.
5. Feeney, L. M.; Nilsson, M. Investigating the energy consumption of a wireless network interface in ad ad hoc networking environment. In *Proceedings of IEEE Infocom 2001*, pages 1548–1557,

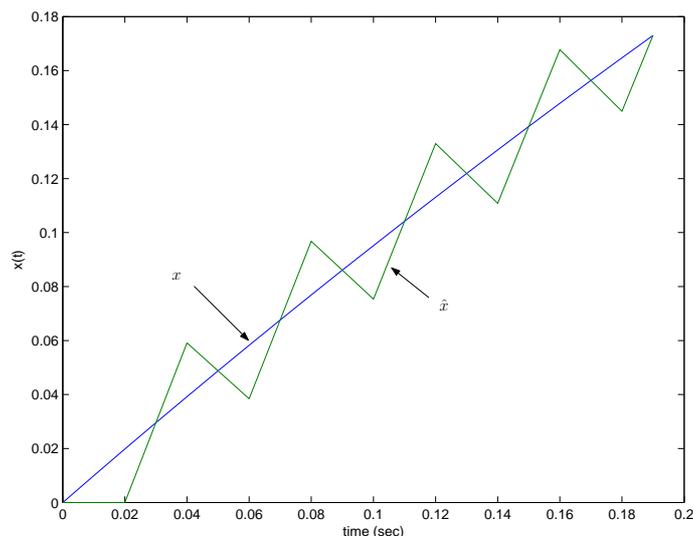


Figure 8. Oscillation problem when only x_k is transmitted ($x_1(t)$ with $\delta = 0.02$)

Anchorage, U.S.A., **2001**.

6. Miskowicz, M. Analytical approximation of the uniform magnitude-driven sampling effectiveness. In *Proc. of IEEE International Symposium on Industrial Electronics*, pages 407–410, Ajaccio, France, **2004**.
7. 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*, pages 3015–3020, Anchorage, U.S.A., **2002**.
8. Miskowicz, M. Send-on-delta concept: An event-based data reporting strategy. *Sensors* **2006**, *65*, 49–63.
9. 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.
10. Astrom, K. J.; Bernhardsson, B. M. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 2011–2016, Las Vegas, U.S.A., **2002**.
11. Plonnigs, J.; Neugebauer, M.; Kabitzsch, K. A traffic model for networked devices in the building automation. In *2004 IEEE Internatinoal Workshop on Factory Communication Systems*, pages 137–145, **2004**.
12. Gersho, A.; Gray, R. M. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, **1992**.
13. Apostol, T. M. *Mathematical Analysis*. Addison-Wesley Publishing Company, **1974**.
14. Lawrenz, W. *CAN System Engineering : From Theory to Pratical Applications*. Springer Verlag, New York, **1997**.
15. Zigbee-Alliance. *Network Specification, Version 1*. **2004**.