Sensors **2009**, *9*, 5919-5932; doi:10.3390/s90805919

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Inertial and Magnetic Sensor Data Compression Considering the Estimation Error

Young Soo Suh

Department of Electrical Engineering, University of Ulsan, Namgu, Ulsan, Korea;
E-Mail: yssuh@ulsan.ac.kr; Tel. +82-52-259-2196; Fax: +82-52-259-1686

Received: 22 June 2009; in revised form: 15 July 2009 / Accepted: 17 July 2009 / Published: 24 July 2009

Abstract: This paper presents a compression method for inertial and magnetic sensor data, where the compressed data are used to estimate some states. When sensor data are bounded, the proposed compression method guarantees that the compression error is smaller than a prescribed bound. The manner in which this error bound affects the bit rate and the estimation error is investigated. Through the simulation, it is shown that the estimation error is improved by 18.81% over a test set of 12 cases compared with a filter that does not use the compression error bound.

Keywords: compression; estimation; inertial sensor; magnetic sensor

---

1. Introduction

Largely because of the MEMS technology, inertial sensors (accelerometers and gyroscopes) are becoming smaller and cheaper [1], which makes it possible to use inertial sensors in many applications. Inertial sensors are used in motion trackers [2], personal navigation systems [3] and remote control systems [4].

In some applications such as body motion trackers (for example, the product 'Moven' by XSENS), inertial sensors are used to track body movement. As the number of inertial sensors increases, the size of sensor data increases accordingly. The sensor data is transmitted to the microprocessor board through wired or wireless communication channels. In a wireless communication channel, the transmission speed is relatively low compared with a wired communication channel. The size of the sensor data needs to be reduced if it exceeds the capacity of the communication channel. For example, the transmission rate of

raw sensor data for one MTx (commercial inertial and magnetic sensor) could be as high as 72 Kbps: 9 sensor outputs (3 accelerometers, 3 gyroscopes, and 3 magnetic sensors) $\times$ 500 Hz (maximum sampling rate) $\times$ 16 bits (16 bit A/D conversion for each sensor). If four MTx are used, the size of the sensor data exceeds the capacity of Zigbee (maximum rate is 250 Kbps [5]). More applications are expected as networked control and monitoring systems are becoming more popular [6].

One way to reduce the size of the sensor data is to compress the sensor data before transmission and decompress the received data in the microprocessor board. Data compression has been extensively studied in many areas [7]. In applications such as body motion trackers, real-time compression is preferable, in order to avoid delayed sensor data transmission and consequently the delay in motion estimation. One of the most popular real-time compression methods is ADPCM (Adaptive Differential Pulse Coded Modulation) [8], which is optimized for voice data. In [9], a simplified ADPCM method is used for inertial sensor data compression, where the maximum error (the difference between the original data and the compressed-and-then-decompressed data) is only relatively bounded (e.g., 1% of the sensor data).

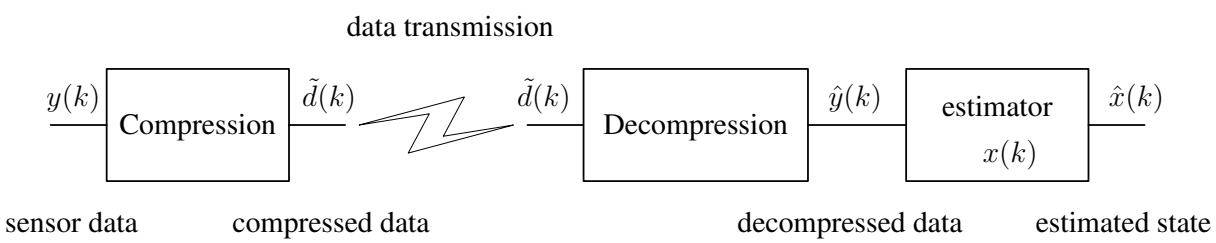
The performance indices of data compression are the bit rate and the quality of compressed data. In voice data compression, the quality of compressed data is evaluated by listening to the compressed-and-then-decompressed voice data. This rather subjective evaluation makes sense since the final destination of compressed data is the human ear. On the other hand, the final destination of compressed inertial sensor data is usually a filter (such as a Kalman filter), where orientation is estimated. Thus the quality of compression should be judged by how the compression affects the estimation error.

In this paper, a modified ADPCM method is proposed, where the absolute maximum error bound is explicitly given. Also, we investigate how this error affects the estimation error. A part of this paper was presented in [10].

2. Inertial and Magnetic Sensor Data Compression and Estimation

The overall process of compression and estimation is given in Figure 1, where k is a discrete time index. The objective is to estimate some states $x(k)$ (attitude, heading, position, etc.) using inertial sensor data $y(k)$ at a limited data transmission rate. The inertial sensor data $y(k)$ is compressed into $\tilde{d}(k)$ and transmitted to the microprocessor board. The compressed data is decompressed into $\hat{y}(k)$ and the state $x(k)$ is estimated using a filter.

Figure 1. Overview of inertial and magnetic sensor data compression and estimation.



Since the objective is to find a good estimator of $x(k)$, the quality of compression is considered good if the estimation error $x(k) - \hat{x}(k)$ is small. The quality of the compression algorithm is evaluated using

the following estimation error covariance:

$$P_{error} = \mathrm{E}\{(x - \hat{x}(\hat{y}))'(x - \hat{x}(\hat{y}))\} \tag{1}$$

where $\hat{x}(\hat{y})$ is an estimator when $\hat{y}$ is used as an output.

Note that P_{error} depends on the filter algorithm used to compute $\hat{x}(\hat{y})$ in addition to the compression algorithm.

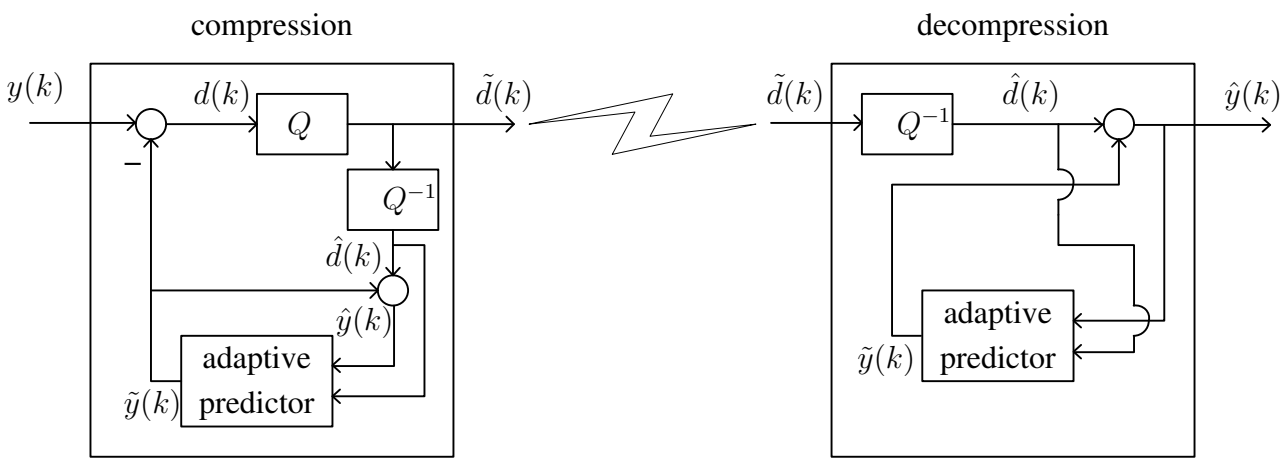
The ideal compression algorithm minimizes both P_{error} and the bit rate. However, usually if P_{error} is small, the bit rate tends to be large. In Section 4, we propose a compression algorithm where the maximum compression error is bounded. The maximum compression error bound plays a role of design parameter to adjust P_{error} and the bit rate.

3. Modified ADPCM Algorithm

The ADPCM block scheme is given in Figure 2. We assume that $y(k)$ is the output of n_y bit uniform quantizer, where $y(k)$ satisfies

$$|y(k)| \leq y_{max} \tag{2}$$

Figure 2. Encoder and decoder block scheme.



Let the quantization size δ of $y(k)$ be defined by

$$\delta = \frac{y_{max}}{2^{n_y - 1}} \tag{3}$$

If there is more than one sensor, we need one encoder for each sensor.

The sensor data $y(k)$ is compared with the predictor output $\tilde{y}(k)$. The difference $d(k)$ is coded into $\tilde{d}(k)$ and this $\tilde{d}(k)$ is transmitted to the estimator board. In the standard ADPCM, $\tilde{d}(k)$ is a quantization index $i(k)$. In this paper, $\tilde{d}(k)$ consists of one bit mode information $m(k)$ and a quantization index $i(k)$:

$$\tilde{d}(k) = \begin{bmatrix} m(k) \\ i(k) \end{bmatrix} \tag{4}$$

In the decoder, the decompressed data is $\hat{y}(k) = \tilde{y}(k) + \hat{d}(k)$. The predictor output $\tilde{y}(k)$ can be computed from $\hat{d}(k)$ and thus does not need to be transmitted.

The adaptive predictor uses the same pole-zero configuration as that in CCITT G.726 ADPCM, which is an ADPCM speech compressor/decompressor protocol proposed in 1990 [11] :

$$\tilde{y}(k) = \sum_{i=1}^{2} a_i(k-i)\hat{y}(k-i) + \sum_{i=1}^{6} b_i(k-1)\hat{d}(k-i) \tag{5}$$

From the assumption (2), if $\tilde{y}(k) > y_{max}$ from (5), we set $\tilde{y}(k) = y_{max}$. Similarly, if $\tilde{y}(k) < -y_{max}$, we set $\tilde{y}(k) = -y_{max}$.

The adaptive algorithm in the G.726 protocol is used to adjust a_i and b_i and the detail is given in [11]; the tone and transition detector part was omitted since the part is only for voice data.

The compression error $e_c(k)$ is the difference between the original signal $y(k)$ and the decompressed signal $\hat{y}(k)$:

$$\begin{aligned} e_c(k) &= y(k) - \hat{y}(k) \\ &= (\tilde{y}(k) + d(k)) - (\tilde{y}(k) + \hat{d}(k)) \\ &= d(k) - \hat{d}(k) \end{aligned} \tag{6}$$

Standard ADPCM algorithms [8] will be modified so that the maximum error is bounded as follows:

$$|e_c(k)| \leq e_{max} \tag{7}$$

Now $\tilde{d}(k)$ coding is explained. The mode bit $m(k)$ in $\tilde{d}(k)$ is used to ensure (7). If the compression error $e_c(k)$ of a standard ADPCM method satisfies (7), then the mode is 0 (i.e., $m(k) = 0$). As will be seen in Section 3.1, this is true if $|d(k)| < 2^{y_s(k)}\delta$, where $y_s(k)$ is an adaptive scaling factor. On the other hand, if the compression error $e_c(k)$ of a standard ADPCM method does not satisfy (7), then the mode is 1 (i.e., $m(k) = 1$) and a special uniform quantizer is used as in Section 3.2. Thus the mode $m(k)$ is given by

$$m(k) = \begin{cases} 0, & \text{if } \frac{|d(k)|}{2^{y_s(k)}\delta} \leq 1 \\ 1, & \text{otherwise} \end{cases} \tag{8}$$

where $y_s(k)$ is an adaptive scaling factor.

The quantization index $i(k)$ is defined differently when $m(k) = 0$ and when $m(k) = 1$.

3.1. Quantization index when $m(k) = 0$

If $m(k) = 0$, a signal $d(k)$ is quantized with n_d bits with a logarithm quantizer with an adaptive scaling factor $y_s(k)$, where the quantized index $i(k)$ ($1 \leq |i| \leq 2^{n_d-1}$) satisfies

$$f_{i-1} < \frac{|d(k)|}{2^{y_s(k)}\delta} \leq f_i \tag{9}$$

The sign of the index $i(k)$ is the same as that of $d(k)$. If $d(k) = 0$, then $i = 1$.

Coefficients f_i in (9) are computed from μ law [8] so that f_i satisfies the following:

$$\frac{\log_e(1 + \mu|f_i|)}{\log_e(1 + \mu)} = \frac{i}{2^{n_d} - 1}$$

Table 1. f_i values for $n_d = 5$.

f_0	f_1	f_2	f_3	f_4
0	0.0237	0.0503	0.0799	0.1131
f_5	f_6	f_7	f_8	f_9
0.1501	0.1915	0.2379	0.2899	0.3479
f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
0.4129	0.4855	0.5667	0.6575	0.7593
f_{15}	f_{16}			
0.8729	1.0000			

In this paper, $\mu = 5$ is used and f_i values for the case of $n_d = 5$ is given in Table 1:

The scaling adaptation factor $y_s(k)$ is computed similarly with the standard ADPCM algorithm except that $y_s(k)$ is bounded as follows:

$$3 \leq y_s(k) \leq \bar{y}_s \tag{10}$$

We note that $\bar{y}_s$ is chosen so that (7) is satisfied. First we are going to derive the upper bound of $e_c(k)$ when $\bar{y}_s$ is given.

The decompressed signal $\hat{d}(k)$ is computed as follows:

$$\hat{d}(k) = \text{sgn}(i(k))2^{y_s(k)}\delta\frac{f_{i-1} + f_i}{2}$$

where

$$\text{sgn}(\alpha) = \begin{cases} 1, & \alpha > 0 \\ 0, & \alpha = 0 \\ -1, & \alpha < 0 \end{cases}$$

The error $e_c(k)$ is then

$$\begin{aligned} |e_c(k)| &= |d(k) - \hat{d}(k)| \\ &\leq 2^{y_s(k)}\delta\frac{f_i - f_{i-1}}{2} \end{aligned} \tag{11}$$

From the fact that f_i is monotonically increasing and (10), we have

$$\begin{aligned} |e_c(k)| &\leq 2^{y_s(k)}\delta\frac{f_{2^{n_d}-1} - f_{2^{n_d}-1-1}}{2} \\ &\leq 2^{\bar{y}_s}\delta\frac{f_{2^{n_d}-1} - f_{2^{n_d}-1-1}}{2} \end{aligned}$$

Given e_{max}, to satisfy (7), $\bar{y}_s$ should satisfy the following

$$2^{\bar{y}_s}\delta\frac{f_{2^{n_d}-1} - f_{2^{n_d}-1-1}}{2} \leq e_{max} \tag{12}$$

Thus if $\bar{y}_s$ is chosen to satisfy (12), the quantization error is always smaller than e_{max} when $m(k) = 0$. We also note that in addition to the global bound e_{max}, if index $i(k)$ is known, we have a less conservative bound given in (11):

$$\bar{e}_c(k) = 2^{y_s(k)}\delta\frac{f_i - f_{i-1}}{2} \tag{13}$$

This bound will be used later in the estimation problem.

3.2. *Quantization index when* $m(k) = 1$

From (8), $m(k) = 1$ if $|d(k)| > 2^{y_s(k)}\delta$. If $m(k) = 1$, then the logarithm quantizer used cannot guarantee the maximum error (7). Noting that $d(k) = y(k) - \tilde{y}(k)$, we can see that the mode is 1 if the difference between the output $y(k)$ and the predicted value $\tilde{y}(k)$ is large, which happens when the signal change is not smooth but instead rather abrupt.

To ensure the maximum error condition (7), we introduce a uniform quantizer when the mode is 1. An example is given in Figure 3, where $y(k)$ is outside the $[\tilde{y}(k) - \delta 2^{y_s(k)}, \tilde{y}(k) + \delta 2^{y_s(k)}]$ interval and the mode is 1. The upper and lower intervals of the mode 1 interval (the mode 0 interval is the shaded area) are quantized with a uniform quantizer (quantization size is $2e_{max}$). The formal definition of the index in mode 1 is given as follows. Let u_{length} and l_{length} be defined by

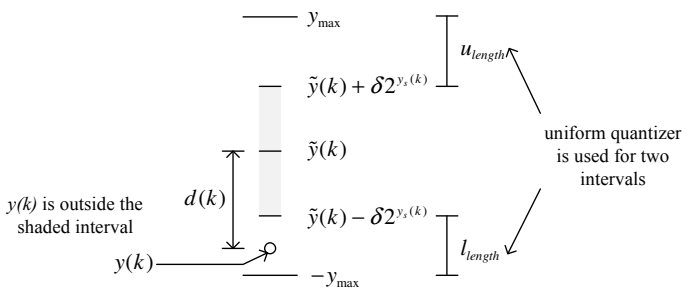
$$\begin{aligned} u_{length} &= y_{max} - (\tilde{y}(k) + \delta 2^{y_s(k)}) \\ l_{length} &= (\tilde{y}(k) - \delta 2^{y_s(k)}) + y_{max} \end{aligned}$$

Let u_{level} (the number of quantization levels for the upper interval u_{length}) be defined by

$$u_{level} = \begin{cases} 0, & \text{if } u_{length} \leq 0 \\ \text{ceil}\left(\frac{u_{length}}{2e_{max}}\right), & \text{otherwise} \end{cases} \quad where$$

ceil(α) is the smallest integer no smaller than α.

Figure 3. If $|y(k) - \tilde{y}(k)| > 2^{y_s(k)}\delta$, then $m(k) = 1$.



The index $i(k)$ is given by

$$i(k) = \begin{cases} \text{floor}\left(\frac{y(k) - \tilde{y}(k) + \delta 2^{y_s(k)}}{2e_{max}}\right) & \text{if } y(k) > \tilde{y}(k) + \delta 2^{y_s(k)} \\ u_{level} + \text{floor}\left(\frac{\tilde{y} - \delta 2^{y_s(k)} - y(k)}{2e_{max}}\right) & \text{if } y(k) < \tilde{y}(k) - \delta 2^{y_s(k)} \end{cases} \tag{14}$$

where floor(α) is the largest integer no larger than α.

The decomposed signal $\hat{d}(k)$ is computed as follows.

$$\hat{d}(k) = \begin{cases} \max\{y_{max}, \tilde{y}(k) + \delta 2^{y_s(k)} + e_{max} + i(k)2e_{max}\}, & \text{if } i < u_{level} \\ \min\{-y_{max}, \tilde{y}(k) - \delta 2^{y_s(k)} - e_{max} - (i(k) - u_{level})2e_{max}\}, & \text{otherwise.} \end{cases}$$

Since a uniform quantizer is used, the compression error $e_c(k)$ in mode 1 is bounded by

$$|e_c(k)| \leq \bar{e}_c(k) = e_{max} \tag{15}$$

Note that the number of bits for the index $i(k)$ is given by

$$n_i(k) = \text{ceil}\left(\log_2\left(\text{ceil}\left(\frac{u_{length}}{2e_m}\right) + \text{ceil}\left(\frac{l_{length}}{2e_m}\right)\right)\right)$$

When $m(k) = 1$, $n_i(k)$ changes depending on $\tilde{y}(k)$ and $y(k)$. Note that when $m(k) = 0$, n_d (the number of bits for $i(k)$) is constant. Also note that n_i can be computed in the estimation board and thus does not need to be transmitted.

4. Kalman Filter Compensating the Compression Error

In Section 3, a compression method is proposed, where the maximum compression error is e_{max}. Also if $m(k)$ and $i(k)$ are known, bounds of the compression error are given by (13) and (15). In this section, we use this information in a Kalman filter.

We assume that $y(k)$ is generated by a linear system

$$\begin{aligned} x(k+1) &= Ax(k) + w(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{16}$$

where $x \in R^n$ is the state, $y \in R^p$ is the output, and $w(k)$ and $v(k)$ are uncorrelated, zero-mean, white Gaussian noises that satisfy

$$\mathrm{E}\{w(k)w'(k)\} = Q, \;\; \mathrm{E}\{v(k)v'(k)\} = R$$

In the standard Kalman filter, $x(k)$ is estimated using $y(k)$. If $y(k)$ is compressed, $\hat{y}(k) = y(k) - e_c(k)$ is used instead. From (13) and (15), $\bar{e}_c(k)$ can be computed, which is used to reduce the estimation error. If we assume $e_{c,i}(k)$ (i-th element of $e_c(k)$) has a uniform distribution, $\mathrm{E}\{e_{c,i}^2(k)\} = \frac{1}{3}\bar{e}_{c,i}^2(k)$. By treating the compression error $e_c(k)$ as measurement noise in $y(k)$, the following model can be used for an estimator.

$$\begin{aligned} x(k+1) &= Ax(k) + w(k) \\ \hat{y}(k) &= Cx(k) + \bar{v}(k) \end{aligned} \tag{17}$$

where

$$\bar{R}(k) = \mathrm{E}\{\bar{v}(k)\bar{v}'(k)\} = R + \frac{1}{3}\begin{bmatrix} \bar{e}_{c,1}^2(k) & & \\ & \ddots & \\ & & \bar{e}_{c,p}^2(k) \end{bmatrix} \tag{18}$$

Since the compression error is compensated in the estimation algorithm, we can expect a smaller estimation error, which is verified through the simulations in Section 5. When a Kalman filter is used for (17), the estimation error covariance $P(k) = \mathrm{E}\{(x(k) - \hat{x}(k))(x(k) - \hat{x}(k))'\}$ can be computed from a Riccati equation [12].

$$P(k+1) = AP(k)A' + Q - AP(k)C'(CP(k)C' + \bar{R}(k))^{-1}CP(k)A' \tag{19}$$

Since $\bar{R}(k)$ depends on $\hat{y}(k)$, we cannot compute $P(k)$ before simulation. To evaluate the estimation error covariance without simulation, we use an upper bound of $\bar{R}(k)$:

$$\bar{R}(k) \leq R + \frac{1}{3}\begin{bmatrix} e_{max,1}^2 & & \\ & \ddots & \\ & & e_{max,p}^2 \end{bmatrix} = \bar{R}_{max}$$

Using this $\bar{R}_{max}$ in place of $\bar{R}(k)$, we can find a steady-state solution to (19).

$$\bar{P} = A\bar{P}A' + Q - A\bar{P}C'(C\bar{P}C' + \bar{R}_{max})^{-1}C\bar{P}A' \tag{20}$$

$\bar{P}$ can be considered as an upper bound of $P(k)$ in (19). From $\bar{P}$, we can see how $e_{max,i}$ affects the estimation error.

A similar idea is used in [13], where a networked estimation problem is considered.

5. Simulation

We compared three data sets using the proposed compression algorithm. Original data is 1600 bits/s for each sensor : 16 bit A/D converted data (i.e., $n_y = 16$) with the sampling rate being 100 Hz. We used $n_b = 5$: that is, the number of bits for the quantization index when $m(k) = 0$ is 5. e_{max} for each sensor is chosen so that $e_{max} = 300\delta$, where δ is different for each type of sensors. $\bar{y}_s = 12$ is found to satisfy (12).

Bit rates for the three compressed data sets are given in Table 2. All three data sets are obtained using XSENS MTx (3 accelerometer, 3 gyroscopes, and 3 magnetic sensors). Holding MTx with a hand, we moved MTx slowly (data set 1) and fast (data set 2). Data set 3 is obtained from a personal navigation system, where MTx is attached on the shoe of a pedestrian [3].

The bit rates of data set 3 is the largest because the change of data is the most abrupt. In particular, when the shoe contacts the floor, there is a large change in the accelerometer and gyroscope outputs, and consequently the compression algorithm becomes $m(k) = 1$ more often.

Table 2. Bit rates for 3 inertial and magnetic sensor data sets.

	accelerometers	gyroscopes	magnetic sensors
data set 1	643.2	641.6	617.6
	648.0	632.0	622.4
	656.0	628.8	614.4
data set 2	659.2	660.8	617.6
	662.4	640.0	622.4
	668.8	638.4	619.2
data set 3	771,2	715.2	638.4
	798.4	696.0	633.6
	694.4	686.4	627.2

To see how the compression error affects the estimation error, a simple one dimensional attitude estimation problem is considered. An attitude (θ) is estimated using two outputs: y_i is an inclinometer output and y_g is a gyroscope output, where

$$\begin{aligned} y_i &= \theta + v_i \\ y_g &= \dot{\theta} + v_g \end{aligned} \tag{21}$$

where v_i and v_g are measurement noises and $q_i = \mathrm{E}\{v_i^2\} = 0.13 \times 10^{-1}$ and $q_g = \mathrm{E}\{v_g^2\} = 0.78 \times 10^{-5}$.

An indirect Kalman filter ([12]) is used to estimate θ using y_i and y_g. In the indirect Kalman filter, the gyroscope output is integrated to compute θ_i, i.e.,

$$\dot{\theta}_i = y_g$$

Defining θ_δ as

$$\theta_\delta = \theta_i - \theta$$

we obtain

$$\dot{\theta}_\delta = v_g \tag{22}$$

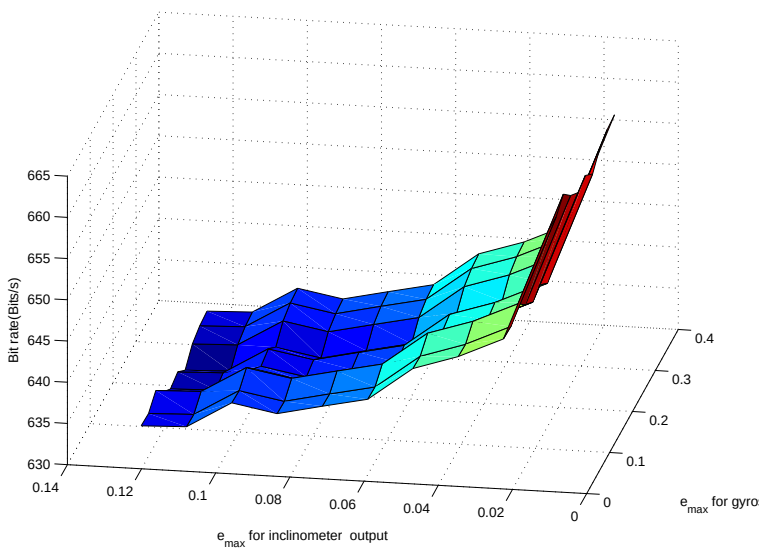
In the indirect Kalman filter, θ_δ is first estimated and $\hat{\theta}$ is obtained indirectly from $\hat{\theta} = \theta_i - \hat{\theta}_\delta$.

By discretizing (22) with the sampling period T ($T = 0.01sec$), we have

$$\begin{aligned} \theta_\delta(k+1) &= \theta_\delta(k) + \sqrt{T} v_g \\ \theta_i - y_a &= \theta_\delta(k) - v_i \end{aligned} \tag{23}$$

Now the proposed compression algorithm is used to compress y_i and y_g. The maximum compression errors $e_{max,i}$ (e_{max} for y_i) and $e_{max,g}$ (e_{max} for y_g) affect both the bit rate and the estimation error. If e_{max} is small, the chance that the mode becomes 1 increases. Thus the bit rate becomes large. How the bit rate changes with the changing e_{max} is given in Figure 4. The data y_i and y_g are generated using Matlab.

Figure 4. Relationship between e_{max} and the bit rate.



From Figure 4, we can see that the bit rate of the inclinometer outputs increases rapidly as $e_{max,i}$ is decreased. On the other hand, the bit rate of the gyroscope outputs does not change much as $e_{max,g}$ is decreased. The bit rate depends on how often the mode becomes 1: note that n_i (the number of bits needed for the quantized data when the mode is 1) is generally larger than n_d (the number of bits needed when the mode is 0). If the original signal is sufficiently smooth, $d(k)$ is small since the predicted value

$\tilde{y}(k)$ is very close to $y(k)$. Thus even if we decrease e_{max}, $d(k)$ still satisfies $|d(k)| \leq 2^{y_s(k)}\delta$ condition in (8).

The y_i and y_g signals and the compressed signals are given in Figures 5 and 6, where $e_{max,i} = 0.2876$ and $e_{max,g} = 0.0122$. We can see that the gyroscope output is relatively smooth compared with the inclinometer output. Thus the bit rate of the gyroscope output is relatively insensitive to the changes in $e_{max,g}$.

Figure 5. Inclinometer output, decompressed data, and error.

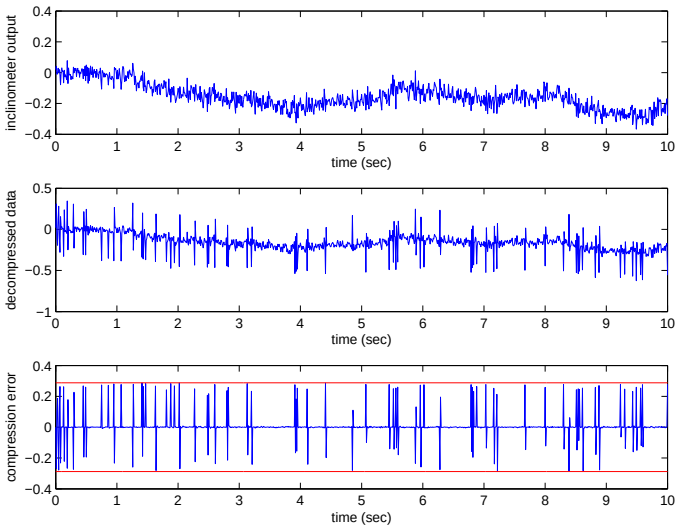
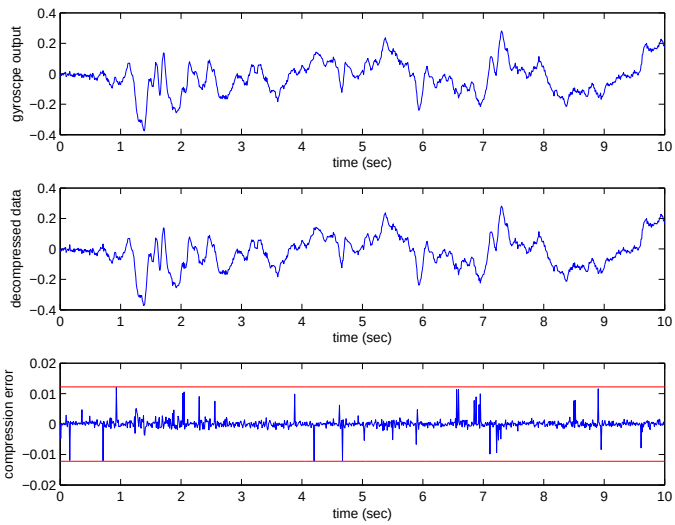


Figure 6. Gyroscope output, decompressed data, and error.



The effects of changes in e_{max} on the estimation error are given in Figure 7, where the estimation error is predicted using (20). Actual estimation error from simulation is given in Figure 8, where $\sqrt{\sum(\theta(k) - \hat{\theta}(k))^2}$ is computed.

Figure 7. Predicted estimation error ($\bar{P}$) of the proposed filter.

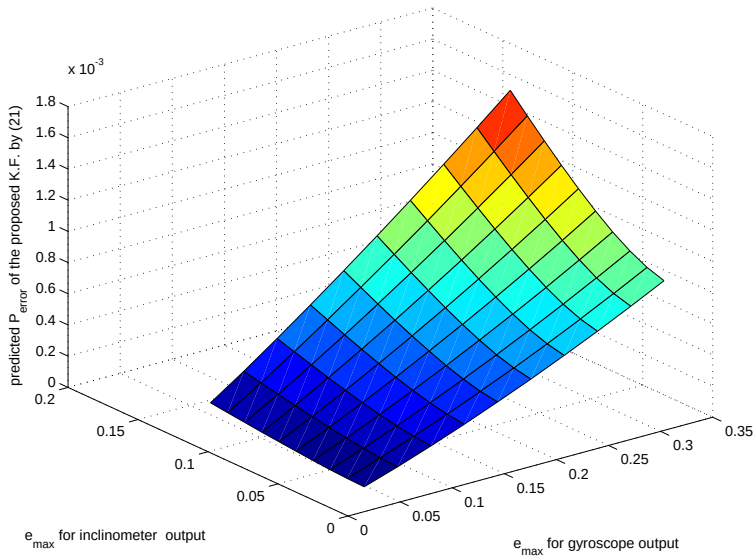
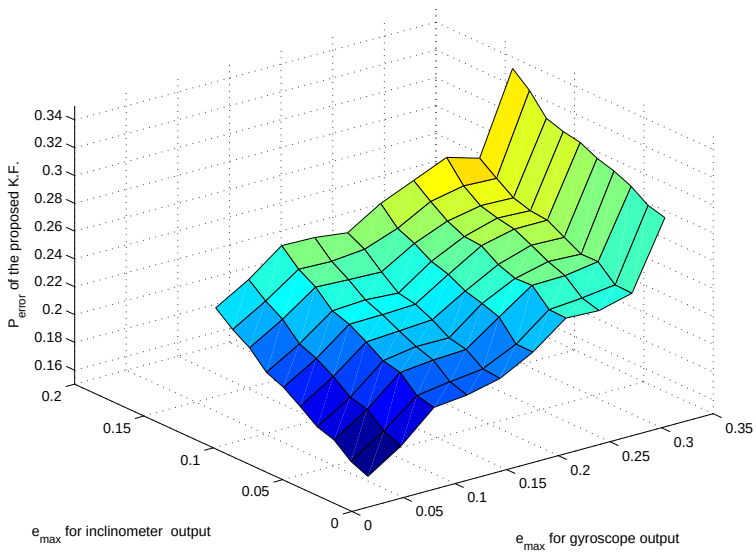


Figure 8. Estimation error ($\sqrt{\sum(\theta(k) - \hat{\theta}(k))^2}$) of the proposed filter.



The relationship between bit rates and estimation error is presented in Figure 9, where data are from Figures 4, 7 and 8. In the left graph, the points 1–6 have similar bit rates but different P_{error}. Thus in the following simulation, we chose the point 1, which corresponds to $e_{max,i} = 0.2876$ and $e_{max,g} = 0.0122$. We compared three different filters: (a) a standard Kalman filter using y_i and y_g; (b) a Kalman filter using $\hat{y}_i$ and $\hat{y}_g$ with compression error compensation (proposed in Section 4); (c) a Kalman filter using $\hat{y}_i$ and $\hat{y}_g$ without compression error compensation. We randomly generated 12 data sets and the results are given in Table 3.

Figure 9. Bit rates and P_{error}: predicted value by (21) and the experiment result.

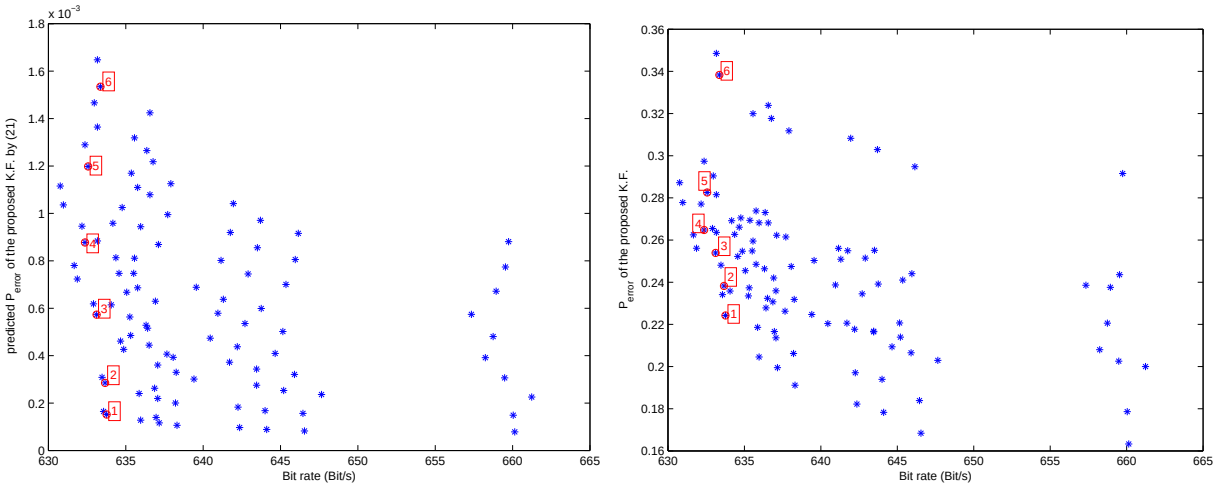


Table 3. Bit rates and estimation error of 3 filters: (a) a standard Kalman filter with uncompressed data, (b) the proposed method with $\hat{y}_i$ and $\hat{y}_g$, and (c) a Kalman filter with $\hat{y}_i$ and $\hat{y}_g$.

experiment	Bit rate		P_{error}			% improvement
	$\hat{y}_g$	$\hat{y}_i$	(a)	(b)	(c)	((c) - (b)) / (c)
1	684.8	620.3	0.318	0.437	0.452	3.32
2	654.4	623.3	0.173	0.361	0.492	26.62
3	640.0	621.3	0.138	0.345	0.402	14.19
4	645.7	622.2	0.139	0.312	0.455	31.39
5	631.2	625.7	0.161	0.382	0.447	14.50
6	633.6	621.6	0.121	0.310	0.422	26.55
7	692.7	692.7	0.308	0.427	0.445	3.96
8	640.8	622.2	0.120	0.280	0.317	11.77
9	634.4	618.6	0.134	0.295	0.537	45.15
10	696.1	618.8	0.423	0.487	0.503	3.14
11	638.4	632.9	0.191	0.394	0.464	14.99
12	638.4	621.5	0.136	0.292	0.417	30.13
average	652.5	628.4	0.197	0.360	0.446	18.81

It is not surprising that the P_{error} of the standard Kalman filter is the smallest because the original data y_i and y_g are used for measurements. We can also see that the P_{error} of the proposed filter is smaller than that of the filter (c). On average, the estimation error of the proposed filter is smaller by 18.81% compared with that of the filter (c). Note that the proposed filter (b) and the filter (c) use the same decompressed data $\hat{y}_i$ and $\hat{y}_g$ for measurements. However, in the proposed filter, the compression error information (13)

and (15) are explicitly used in (18), whereas they are ignored in the filter (c). In summary, the estimation error reduction was possible because of two facts: (1) the proposed compression method provides the compression error bound (13) and (15), and (2) the proposed filter algorithm explicitly uses the error compression bound.

6. Conclusion

In this paper, we have proposed a compression method for inertial and magnetic sensor data. The proposed compression method guarantees that the compression error is bounded by a prescribed e_{max} value. Smaller e_{max} value usually increases the bit rate and reduces the estimation error of the filter when the decompressed data is used. Thus e_{max} plays the role of a trade-off parameter between the bit rate and the estimation error. Also we have seen that by using a bound on compression error, the estimation error can be reduced.

Acknowledgments

This work was supported by the Korea Research Foundation(KRF) grant funded by the Korea government(MEST) (No. 2009-0074773).

References and Notes

1. Barbour, N.; Schmidt, G. Inertial sensor technology trends. *IEEE Sens. J.* **2001**, *1*, 332-339.
2. Welch, G.; Foxlin, E. Motion tracking: no silver bullet, but a respectable arsenal. *IEEE Comput. Graph. Appl.* **2002**, *22*, 24-38.
3. Foxlin, E. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Comput. Graph. Appl.* **2005**, *25*, 38-46.
4. Suh, Y.S.; Park, S.K.; Kim, D.; Jo, K. Remote control of a moving robot using the virtual link. In *Proceedings of IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007; pp. 2343-2348.
5. Zigbee-Alliance. *Network Specification, Version 1*; 2004.
6. Choi, D.H.; Kim, D.S. Wireless fieldbus for networked control systems using lr-wpan. *Int. J. Contr. Autom. Syst.* **2008**, *6*, 119-125.
7. Gersho, A.; Gray, R.M. *Vector Quantization and Signal Compression*; Kluwer Academic Publishers: Norwell, MA, USA, 1992.
8. Jayant, N.; Noll, P. *Digital Coding of Waveforms: Principles and Applications to Speech and Video*; Prentice-Hall: New Jersey, USA, 1984.
9. Cheng, L.; Hailes, S.; Cheng, Z.; Fan, F.Y.; Hang, D.; Yang, Y. Compressing inertial motion data in wireless sensing systems - an initial experiment. In *Proceedings of the 5th International Workshop on Wearable and Implantable Body Sensor Networks*, Hong Kong, China, 2008.
10. Suh, Y.S. Compression of inertial and magnetic sensor data for network transmission. In *Proceedings of 8th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems*, Ansan, Korea, 2009; pp. 226-229.

11. ITU. *Recommendation G.726 : 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation*; ITU: Geneva, Switzerland, 1990.
12. Brown, R.G.; Hwang, P.Y.C. *Introduction to Random Signals and Applied Kalman Filtering*; John Wiley & Sons: New York, NY, USA, 1997.
13. 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*, 332-338.