Compression of Inertial and Magnetic Sensor Data for Network Transmission

Young Soo Suh *

* *Dept. Electrical Eng., Univ. of Ulsan, Ulsan 680-749, Korea (e-mail: suh@ieee.org)*

Abstract: This paper presents a compression method for inertial and magnetic sensor data, where 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 prespecified bound. How this error bound affects the compression ratio and the estimation performance is investigated. Using the compression error bound information in the filter algorithm, the estimation performance is improved.

Keywords: compression, estimation, inertial sensor, magnetic sensor

1. INTRODUCTION

Due to mainly MEMS technology, inertial sensors (accelerometers and gyroscopes) are becoming smaller and cheaper, which made it possible that inertial sensors are used in many applications (see Barbour and Schmidt (2001)). Inertial sensors are used for motion trackers in Welch and Foxlin (2002), personal navigation systems in Foxlin (2005), and remote control systems in Suh et al. (2007a).

In some applications such as body motion trackers (for example, Moven by a company XSENS), many 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 are transmitted to the microprocessor board through wired or wireless communication channels. If the size of sensor data exceeds capacity of the communication channel, the size of sensor data needs to be reduced.

One way to reduce the size of sensor data is compressing the sensor data before transmission and decompressing the received data in the microprocessor board. In applications such as body motion trackers, real-time compression method is preferable; otherwise sensor data transmission is delayed and motion estimation is delayed. One of most popular real-time compression method is ADPCM (Adaptive Differential Predictive Control Method), which is optimized for voice data (see Jayant and Noll (1984)). In Cheng et al. (2008), 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).

When we compare data compression methods, performance indices are a compression rate and a 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 human ears. On the other hand, the final destination of compressed inertial sensor data is usually a filter (such as Kalman filter), where orientation is estimated. Thus the quality of compression should be judged by how the compression affects the estimation performance.

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 performance.

2. INERTIAL AND MAGNETIC SENSOR DATA COMPRESSION AND ESTIMATION

The overall process of compression and estimation is given in Fig. 1, where k is a discrete time index. The objective is to estimate some states $x(k)$ (attitude, heading, position, and etc) using inertial sensor data $y(k)$ while a data transmission rate is limited. 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.

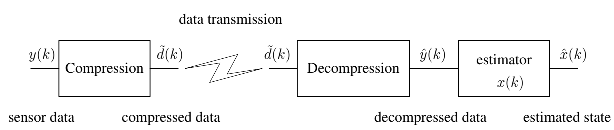

Fig. 1. Overview of inertial and magnetic sensor data compression and estimation

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

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

* The authors would like to thank Ministry of Knowledge Economy and Ulsan Metropolitan City which supported this research through the Network-based Automation Research Center (NARC) at University of Ulsan

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

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

The compression ratio is defined as follows:

$$P_{compression} = \frac{\sum_{k=1}^{N} \text{number of bits expressing } \tilde{d}(k)}{\sum_{k=1}^{N} \text{number of bits expressing } y(k)} \quad (2)$$

where N is the number of total data. Note that small $P_{compression}$ means that compression efficiency is good.

3. MODIFIED ADPCM ALGORITHM

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

$$|y(k)| \leq y_{max}. \quad (3)$$

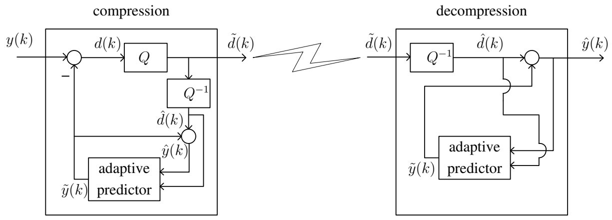

Fig. 2. Encoder and decoder block schematic

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

$$\delta = \frac{y_{max}}{2^{n_y - 1}}. \quad (4)$$

If there are more than one sensor, we need an 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 $i(k)$:

$$\tilde{d}(k) = \begin{bmatrix} m(k) \\ i(k) \end{bmatrix}. \quad (5)$$

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 need not be transmitted.

The adaptive predictor uses the same pole-zero configuration as that in CCITT G.726 ADPCM.

The adaptive algorithm in G.726 is used to adjust a_i and b_i and the detail is given in ITU (1990); the tone and transition detector part was omitted since the part is only for voice data.

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

$$e_c(k) = y(k) - \hat{y}(k) \quad (6)$$

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

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

Now $\tilde{d}(k)$ coding is explained. The mode bit $m(k)$ in $\tilde{d}(k)$ is used to ensure (7) and $m(k) = 0$ if $d(k)$ is not large. More specifically, $m(k)$ is determined as follows:

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

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

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

The sign of index $i(k)$ is the same as that of $d(k)$. If $d(k) = 0$, then $i = 1$. Coefficients f_i in (9) is computed from μ law (see Jayant and Noll (1984)).

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

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

We note that $\bar{y}_s$ is chosen so that (7) is satisfied. To do that, the upper bound of $e_c(k)$ is computed when $\bar{y}_s$ is given.

The error $e_c(k)$ is then

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

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

$$2^{\bar{y}_s}\delta \frac{f_{2^{n_d-1}} - f_{2^{n_d-1}-1}}{2} \leq e_{max}. \quad (12)$$

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):

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

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 in the mode 1 cannot guarantee the maximum error (7). Noting that $d(k) =$