

## DYNAMIC SENSOR BIAS CORRECTION FOR ATTITUDE ESTIMATION USING UNSCENTED KALMAN FILTER IN AUTONOMOUS VEHICLE

MIKIO BANDO, YUKIHIRO KAWAMATA AND TOSHIYUKI AOKI

Department of Smart Systems Research  
Hitachi Research Laboratory, Hitachi Ltd.  
1-1, Omika-cho, Hitachi-shi, Ibaraki-ken, Japan  
mikio.bando.bt@hitachi.com

Received March 2011; revised September 2011

**ABSTRACT.** *This paper describes a method for estimating sensor biases by using a low-dimensional Unscented Kalman Filter (UKF) to maintain the positional estimation accuracy of an autonomous vehicle (AV). It is difficult to estimate attitude accurately in a blind situation (such as with no GPS satellites and no landmarks), because of sensor bias. We developed a dead reckoning system for an embedded system using the UKF. The UKF has high computational effort, so, we decreased the number of dimensions in the UKF by excluding sensor biases term. On the presumption that AV drives steadily, we derived equations for the relationship between the averages of angular acceleration and gyro bias, and corrected the sensor output. Instead of using high-dimensional UKF, we corrected sensor biases by using these equations. This method quickly and accurately estimated attitude.*

**Keywords:** Sensor bias estimation, Unscented Kalman filter, Localization

1. **Introduction.** The development of autonomous vehicles (AVs) has made great progress recently. For example, the Defense Advanced Research Projects Agency (DARPA) in the U.S.A organized Grand Challenge [1] and Urban Challenge [2], which are long distance races in which autonomously controlled vehicles participate. Some companies are expected to put these technologies into practical use in the transport or factory.

AVs are classified broadly into two categories. One is a guided AV which follows a pre-established route, and the other is a self-directed AV which does not. Almost all self-directed AVs have an environmental recognition function for noticing pedestrians or obstacles near the vehicle, a control planning function for avoiding these pedestrians and obstacles, and other functions [3]. To use these functions, the AV always estimates its position. This is because localization function is one of the most important functions to realize autonomous running.

However, developing features that can recognize the environment or calculate localization for AVs is expensive. Practically applying the autonomous technology will be an important factor in reducing these costs. There are some studies on localization using low-cost sensors [4,5], but we know of no studies on accurate localization at low calculation cost.

We are going to devise localization logic of low calculation cost with the goal of installing the localization logic in embedded equipment such as the in-vehicle controller.

This paper describes results of a low calculation cost method for maintaining localization accuracy for practical application of the autonomous technology.

## 2. Localization Model.

**2.1. Dead reckoning model.** To estimate vehicle position, this study used dead reckoning model as shown in Figure 1.

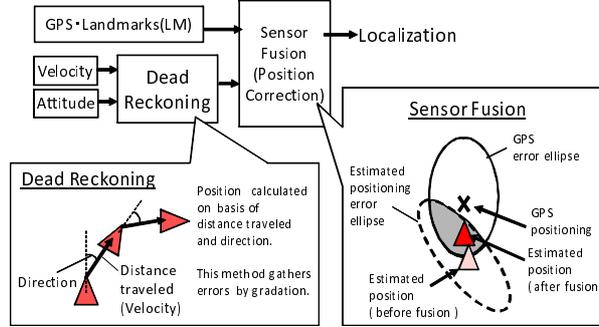


FIGURE 1. Dead reckoning model

The dead reckoning method estimates the vehicle's position on the basis of distance traveled (velocity) and the direction. The dead reckoning model influences the non-holonomic constraint of 4-wheel vehicle; thus the vehicle cannot move sideways.

Let the position of a vehicle be  $(x(t), y(t), z(t))$  at the step  $t$  and let its velocity and angular velocity be  $v(t)$  and  $\omega(t)$ , respectively. Then, during the time increment  $\delta t$  second, the variation of the vehicle position,  $(\delta x(t), \delta y(t), \delta z(t))$ , is given by

$$\delta x(t) = v(t) \cos \theta_y(t) \cos \theta_p(t) \delta t \quad (1)$$

$$\delta y(t) = v(t) \sin \theta_y(t) \cos \theta_p(t) \delta t \quad (2)$$

$$\delta z(t) = v(t) \sin \theta_p(t) \delta t \quad (3)$$

where  $\theta_y(t)$ ,  $\theta_p(t)$  are the yaw and pitch angles at step  $t$ . Thus, the vehicle position  $(x(t+1), y(t+1), z(t+1))$  at the next step  $t+1$  is estimated in the following manner:

$$\begin{pmatrix} x(t+1) \\ y(t+1) \\ z(t+1) \end{pmatrix} = \begin{pmatrix} x(t) + \delta x(t) \\ y(t) + \delta y(t) \\ z(t) + \delta z(t) \end{pmatrix} \quad (4)$$

In this way, we can calculate vehicle position if we know the vehicle velocity and attitude.

**2.2. Attitude calculation model.** The dead reckoning model needs vehicle velocity and attitude. The velocity can often be observed directly with measurement devices. However, in a lot of cases, all parts of the vehicle attitude (yaw, roll, pitch) cannot always be observed directly. Therefore, we calculate the vehicle attitude using the following method.

$$\phi(t) = \sqrt{\omega_r(t)^2 + \omega_p(t)^2 + \omega_y(t)^2} \quad (5)$$

$$\boldsymbol{\lambda}(t) = \begin{pmatrix} \lambda_1(t) \\ \lambda_2(t) \\ \lambda_3(t) \end{pmatrix} = \begin{pmatrix} \frac{\omega_r(t)}{\phi(t)} \\ \frac{\omega_p(t)}{\phi(t)} \\ \frac{\omega_y(t)}{\phi(t)} \end{pmatrix} \quad (6)$$

$$\Delta \mathbf{C}_{EV}(t) = \mathbf{I} \cos \phi(t) + \tilde{\lambda}(t) \sin \phi(t) + \lambda(t)\lambda(t)^T(1 - \cos \phi(t)) \tag{7}$$

$$\tilde{\lambda}(t) = \begin{pmatrix} 0 & -\lambda_3(t) & \lambda_2(t) \\ \lambda_3(t) & 0 & -\lambda_1(t) \\ -\lambda_2(t) & \lambda_1(t) & 0 \end{pmatrix}$$

$$\mathbf{C}_{EV}(t) = \Delta \mathbf{C}_{EV}(t)\mathbf{C}_{EV}(t - 1) \tag{8}$$

$$\boldsymbol{\theta}(t) = \begin{pmatrix} \theta_y(t) \\ \theta_r(t) \\ \theta_p(t) \end{pmatrix} = \mathbf{C}_{EV}(t)\boldsymbol{\theta}(0) \tag{9}$$

where  $t$  is an observing step,  $\theta_y(t), \theta_r(t), \theta_p(t)$  is the attitude,  $\boldsymbol{\theta}(0)$  is initial attitude vector,  $\omega_y(t), \omega_r(t), \omega_p(t)$  is the anglar velocity (yaw, roll, pitch),  $\mathbf{C}_{EV}$  is a rotation matrix from the topocentric tangent plane coordinate system to vehicle coordinate system, and  $\Delta \mathbf{C}_{EV}$  is a tiny rotation matrix.

### 3. Attitude Calculation Modeling Using UKF.

**3.1. Attitude calculation without gyro sensor.** When the vehicle attitude is calculated in an actual situation, dead reckoning positioning accumulates errors because of gyro sensor errors. Therefore, we consider how to eliminate the positioning errors using another sensor, for example, accelerometer and GPS.

When accelerometer is used, pitch angle and roll angle are calculated using the following method expressing gravity term:

$$\mathbf{G}(t) = \boldsymbol{\alpha}(t) + \mathbf{C}_{EV}^T(t)\mathbf{g} + \epsilon_G(t), \tag{10}$$

where  $\mathbf{G}$  is output value of accelerometer,  $\boldsymbol{\alpha}$  is acceleration of the vehicle, and  $\mathbf{g}$  is acceleration of gravity.

A yaw angle can be directly observed by the GPS direction term. Thus we can observe all components of vehicle attitude (yaw, roll, pitch), and occasionally eliminate the accumulating attitude errors.

**3.2. Unscented Kalman filtering (UKF) modeling.** The vehicle attitude is calculated by the UKF which has robust nonlinear output of sensors, because gyro sensor and accelerometer are predicted to output strongly nonlinear responses in outdoor areas where AVs are driving around.

The UKF algorithm uses arbitrary nonlinear function. The extended Kalman filter uses Jacobian matrix, while the UKF does the “sigma points”. The UKF algorithm will be stated below [6].

The state vector  $\mathbf{x}(t)$  and observed vector  $\mathbf{y}(t)$  at time  $t$  are defined by

$$\mathbf{x}(t) = [\boldsymbol{\theta}(t), \boldsymbol{\omega}(t), \boldsymbol{\alpha}(t), \dot{\boldsymbol{\omega}}(t), \dot{\boldsymbol{\alpha}}(t), \mathbf{b}_\omega, \mathbf{b}_\alpha] \tag{11}$$

$$\mathbf{y}(t) = [\boldsymbol{\Theta}(t), \boldsymbol{\Omega}(t), \mathbf{V}(t), \mathbf{G}(t)] \tag{12}$$

where  $\boldsymbol{\theta}$  is the attitude vector,  $\boldsymbol{\omega}$  is the angular velocity vector,  $\mathbf{b}_\omega$  is the bias term vector of gyro sensor,  $\mathbf{b}_\alpha$  is the bias term vector of accelerometer.  $\dot{\boldsymbol{\omega}}(t), \dot{\boldsymbol{\alpha}}(t)$  are able to calculate from thire time constant [7,8].  $\boldsymbol{\Theta}$  is the attitude calculating from Equation (10) and GPS direction, and  $\boldsymbol{\Omega}$  is the output value vector of gyro sensor.

Given the state vector at the step  $t - 1$ , we calculate the prediction of sigma points as follows:

$$\begin{aligned} \tilde{\chi}_0(t|t-1) &= f(\hat{\mathbf{x}}(t-1|t-1), t-1) \\ \tilde{\chi}(t|t-1) &= [\tilde{\chi}_0(t|t-1), \tilde{Q}(t|t-1), \tilde{R}(t|t-1)] \\ \tilde{y}_i &= h(\tilde{\chi}_i(t|t-1), t-1), \quad i = 0, 1, \dots, 2n \\ t &= t_0 + 1, t_0 + 2, \dots \end{aligned}$$

where

$$\begin{aligned} \tilde{Q}(t|t-1) &= [\tilde{q}_1(t|t-1), \tilde{q}_2(t|t-1), \dots, \tilde{q}_i(t|t-1)] \\ \tilde{R}(t|t-1) &= [\tilde{r}_1(t|t-1), \tilde{r}_2(t|t-1), \dots, \tilde{r}_i(t|t-1)] \\ \tilde{q}_i(t|t-1) &= \tilde{\chi}_0(t|t-1) + \gamma(\sqrt{G W G^T}_i) \\ \tilde{r}_i(t|t-1) &= \tilde{\chi}_0(t|t-1) - \gamma(\sqrt{G W G^T}_i) \\ G &= G(\hat{\mathbf{x}}(t-1|t-1), t-1) \\ W &= W(t-1) \\ i &= 1, 2, \dots, 2n \\ \hat{\mathbf{x}}(t|t) &= \hat{\mathbf{x}}(t|t-1) + K(t)e(t) \tag{13} \end{aligned}$$

$$e(t) = y(t) - \hat{y}(t|t-1) \tag{14}$$

$$K(t) = L(t|t-1)H^{-1}(t|t-1) \tag{15}$$

$$\hat{\mathbf{x}}(t|t-1) = \sum_{i=0}^{2n} w_i \tilde{\chi}_i(t|t-1) \tag{16}$$

$$\hat{y}(t|t-1) = \sum_{i=0}^{2n} w_i \tilde{y}_i(t|t-1) \tag{17}$$

$$G = G(\hat{\mathbf{x}}(t-1|t-1), t-1)$$

$$W = W(t-1)$$

$$\begin{aligned} M(t|t-1) &= \sum_{i=0}^{2n} w_i [\tilde{\chi}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)] \\ &\quad [\tilde{\chi}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)]^T + G W G^T \tag{18} \end{aligned}$$

$$\begin{aligned} L(t|t-1) &= \sum_{i=0}^{2n} w_i [\tilde{\chi}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)] \\ &\quad [\tilde{y}_i(t|t-1) - \hat{y}(t|t-1)]^T \tag{19} \end{aligned}$$

$$\begin{aligned} H(t|t-1) &= \sum_{i=0}^{2n} w_i [\tilde{y}_i(t|t-1) - \hat{y}(t|t-1)] \\ &\quad [\tilde{y}_i(t|t-1) - \hat{y}(t|t-1)]^T + V(t) \tag{20} \end{aligned}$$

$$t = t_0 + 1, t_0 + 2, \dots$$

$$\hat{\mathbf{x}}(t_0|t_0) = E[x(t_0)] \tag{21}$$

The UKF has a higher computation load than the extended Kalman filtering, instead of a higher nonlinear tracking capability [6].

The UKF has  $O(N^3)$  computational effort in the number of states  $N$  [6,7]. CPU power of our embedded system assumes 500MHz, and attitude estimation function can use less than 0.01% of the CPU power. In the case of our situation  $N = 21$ , the computational effort is too great for using in the embedded system. Figure 2 shows calculation components of the positioning system.

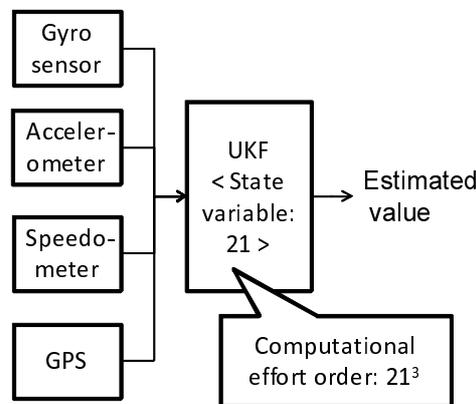


FIGURE 2. Calculation components of positioning system

Thus, we propose a method to decentralize calculation of state variables to lessen the computational effort like a bank of Kalman filters [11]. The state variables (11) are classified into two groups: one largely alters in value from moment to moment, while the other slightly alters. The latter is sensor bias terms (6 variables), so they can be calculated by linear theories that need only low computational effort. Figure 3 shows our proposed component architecture.

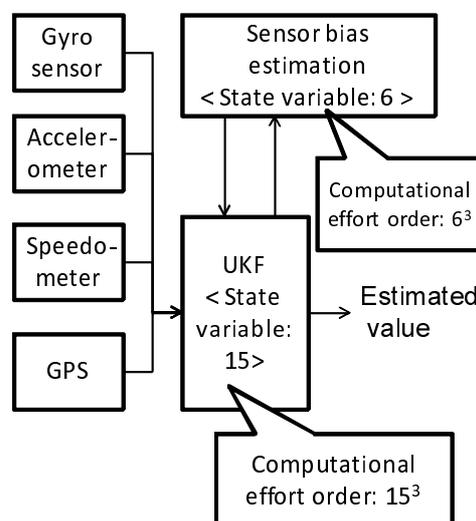


FIGURE 3. Decentralized positioning system

The proposed component architecture, which has a calculation number of times  $15^3$  order, decreases computational effort around 65 % compared with centralized architecture, which has a calculation number of times  $21^3$ , like in Figure 2.

Thus, we have to design the UKF without sensor bias term, but there are no studies on decentralized sensor bias architecture. Many studies set up bias equations as a component of the Kalman filtering [9-12]. Therefore, we developed the expression for the relationship between the gyro bias and the other estimated variables of the UKF.

#### 4. Bias Estimating Principle.

**4.1. Theory of bias estimation.** The gyro sensor bias is calculated by angular acceleration. We developed the following expressions for the relationship between the gyro sensor bias and the angular acceleration:

$$\omega_g(t) = \omega(t) + b_\omega(t) + \epsilon(t) \quad (22)$$

$$\omega_g(t) - \omega(t-1) = \omega(t) - \omega(t-1) + b_\omega(t) + \epsilon(t) \quad (23)$$

$$= \delta\omega(t-1) + b_\omega(t) + \epsilon(t) \quad (24)$$

where  $t$  is step,  $\omega_g$  is observed angular velocity of gyro sensor,  $\omega$  is estimated angular velocity,  $\delta\omega$  is variation of the angular velocity,  $b_\omega$  is bias of gyro sensor, and  $\epsilon$  is noise. Also  $\delta\omega$ ,  $b_\omega$  and  $\epsilon$  are all independent.

Taking the average of both sides of (24), we have

$$\begin{aligned} E[\omega_g(t) - \omega(t-1)] &= E[\delta\omega(t) + b_\omega(t) + \epsilon(t)] \\ &= E[\delta\omega(t)] + E[b_\omega(t)] + E[\epsilon(t)] \end{aligned} \quad (25)$$

$$\begin{aligned} E[\delta\omega(t)] &= \frac{1}{n\delta t} \sum_{i=0}^{t-1} \delta\omega(i) \\ &= \frac{(\omega(1) - \omega(0)) + (\omega(2) - \omega(1)) + \dots + (\omega(t) - \omega(t-1))}{n\delta t} \\ &= \frac{\omega(t) - \omega(0)}{n\delta t} \end{aligned} \quad (26)$$

When the average noise is assumed to be 0, and the steadily moving vehicle in finite time ( $\omega(t) = \omega(0)$ ) observes the average angular acceleration to be 0, Equation (25) changes as shown below.

$$E[\omega_g(t) - \omega(t-1)] = E[b_\omega] \quad (27)$$

$$\begin{aligned} b_\omega(t) &= nE[b_\omega(t)] - \sum_{i=0}^{n-1} b_\omega(i) \\ &= nE[\omega_g(t) - \omega(t-1)] - \sum_{i=0}^{n-1} b_\omega(i) \end{aligned} \quad (28)$$

We can calculate the gyro bias from Equation (28), and these equations require less computational effort than estimating all state variables at once.

**4.2. Result of bias estimating simulations.** To examine the above decentralizing method, we simulated bias estimating capability. We collected sensor data from a moving dump truck used as an experimental vehicle to verify our methods. The vehicle had the following devices: GPS, gyro, and acceleration and speed sensors (see Figure 4).

The GPS was used to obtain a reference position. The vehicle traveled on a sand experimental field, in a circle course of around 2 km for 6 minutes. An overview of the experimental field is shown in Figure 5.

We gathered simulation data, which were added arbitrary bias to yaw angular velocity of gyro sensor output. Using these data, estimating state variables by the UKF simulation indicated the following results.

Figure 6 shows the estimated angular acceleration (yaw). It is difficult to distinguish between the sensor bias and the angular acceleration. However, the average angular acceleration converges on the gyro bias value (see Figure 7).

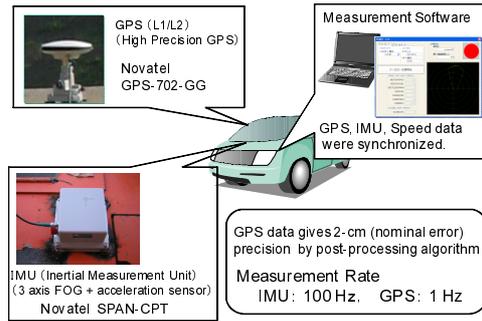


FIGURE 4. Devices in experimental vehicle

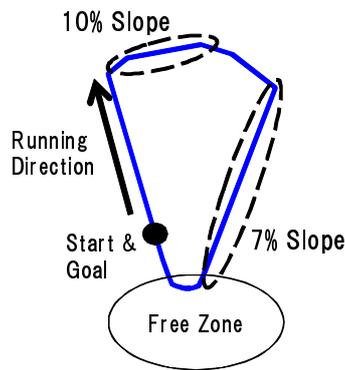


FIGURE 5. Overview of experimental field

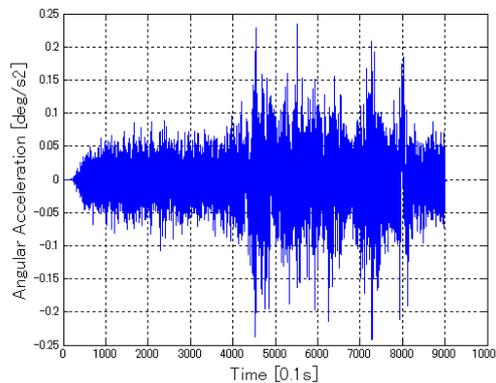


FIGURE 6. Estimated angular acceleration (gyro sensor has 0.01 deg/s yaw rate bias)

We assume that if the gyro bias can estimate Equation (28), the average estimated angular acceleration value is sequentially fed back to the raw gyro output value. Figure 8 shows the result of corrected bias of gyro output.

Correcting the gyro output value by estimated bias is noisy. To smooth the corrected gyro output, we apply the Kalman filtering to the estimated average angular acceleration. Figure 9 shows the results of smoothing average angular acceleration feedback.

The gyro output value is converges clearly to zero, because the gyro output is corrected bias.

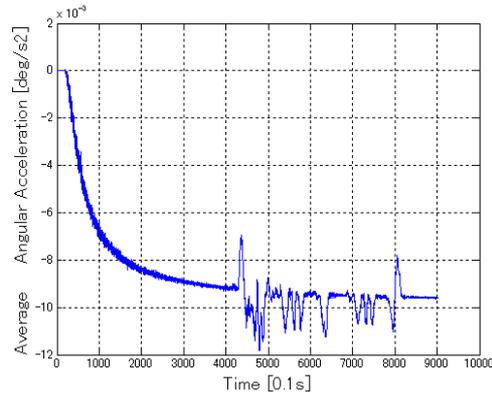


FIGURE 7. Average estimated angular acceleration (gyro sensor has 0.01 deg/s yaw rate bias)

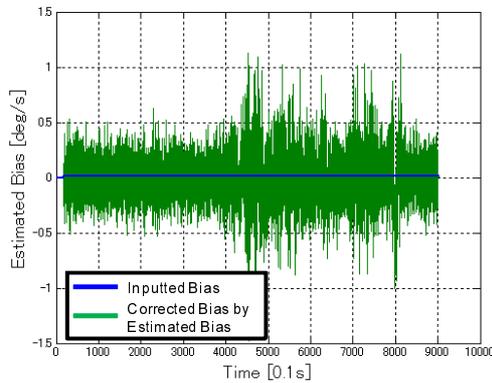


FIGURE 8. Corrected bias of gyro sensor under fed back average of estimated angular acceleration (gyro sensor has 0.01 deg/s yaw rate bias)

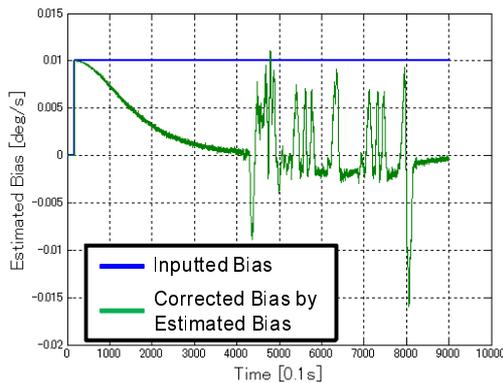


FIGURE 9. Estimated bias of gyro sensor under fed back Kalman filtering estimated angular acceleration (gyro sensor has 0.01 deg/s yaw rate bias)

Also, we simulated a low-cost gyro sensor that had 3 deg/s bias. Figure 10 shows the corrected gyro sensor output under 3 deg/s bias. These results indicate that our bias estimation method can correspond to not only small but also large bias.

Next, we checked our method for time varying bias adaptivity. We simulated the same way but added bias was increased  $10^{-3}$  per second. Figure 11 compares raw gyro sensor output and output of adding bias.

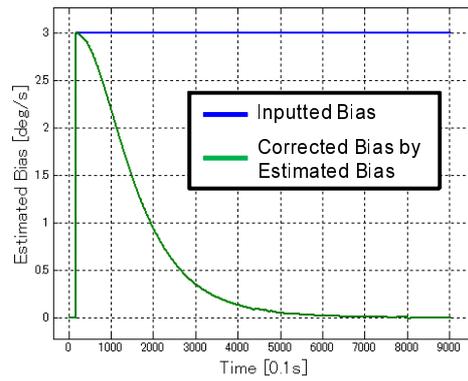


FIGURE 10. Estimated bias of gyro sensor under fed back Kalman filtering estimated angular acceleration (gyro sensor has 3.00 deg/s yaw rate bias)

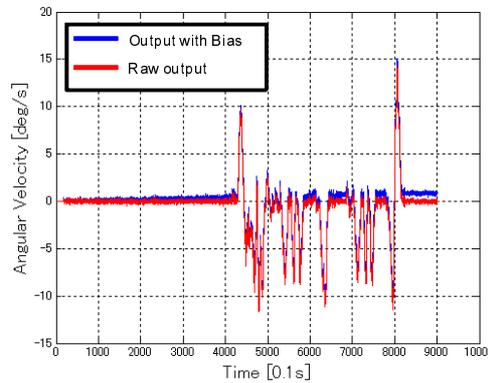


FIGURE 11. Estimated angular velocity (gyro sensor has gyrating yaw rate bias)

Let us estimate bias in the same way. Figure 12 shows the estimated angular acceleration.

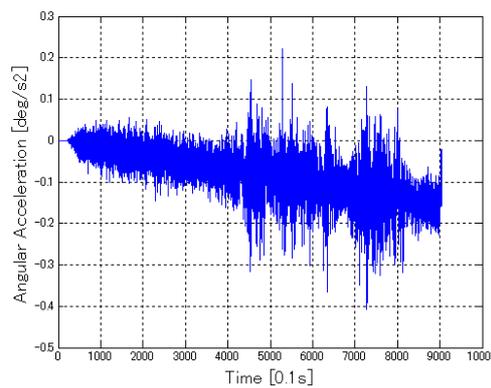


FIGURE 12. Estimated angular acceleration (gyro sensor has gyrating yaw rate bias)

As bias increased, the estimated angular acceleration decreased. This result indicates that the proposed method can estimate the gyro bias in the same way. Figure 13 shows the results of estimating gyro bias when the Kalman filtering average estimated angular acceleration is fed back to raw gyro output.

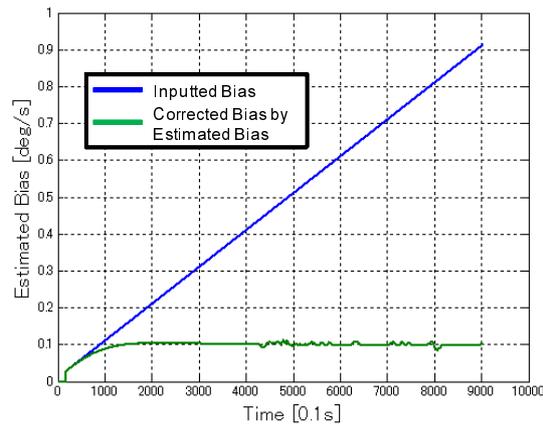


FIGURE 13. Estimated bias of gyro sensor under fed back Kalman filtering estimated angular acceleration (gyro sensor has gyrating yaw rate bias)

A measurable stationary error finally converged in Figure 13 because of large bias fluctuation. Next, we simulated small bias fluctuation  $10^{-5}$  deg/s per seconds. Figure 14 shows the results of estimating gyro bias when the Kalman filtering average estimated angular acceleration is fed back to raw gyro output.

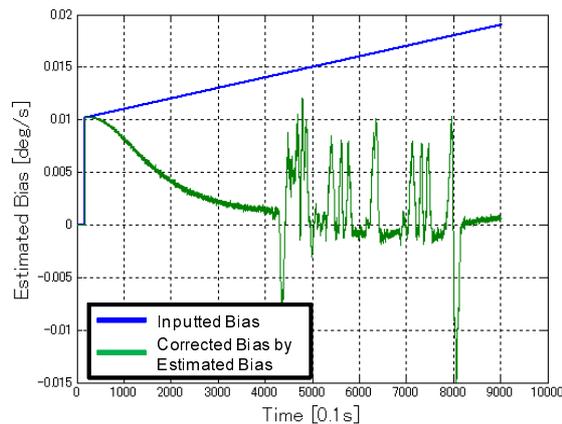


FIGURE 14. Estimated bias of gyro sensor under fed back Kalman filtering estimated angular acceleration (gyro sensor has little gyrating yaw rate bias)

Figure 13 compares some stationary errors. This simulation results indicate our method is efficient when bias fluctuation is small.

At last, we compared the estimated performance of proposed decentralized method with conventional method. Figure 15 shows estimated bias of proposed method and conventional method.

And the performance of estimated bias is shown in Table 1. The convergence rate is RMS value of estimated bias, the low value means “convergence is fast”. The average of calculation time were obtained under the condition “Intel Core i7 CPU 3.07 GHz” and “1.96 GB main RAM”.

As for the convergence, the convergence level does not have inferiority than the conventional method although it is slightly late. We expect that these values can be improved by tuning more. In addition, 20% of calculation time is improved using our method. Because

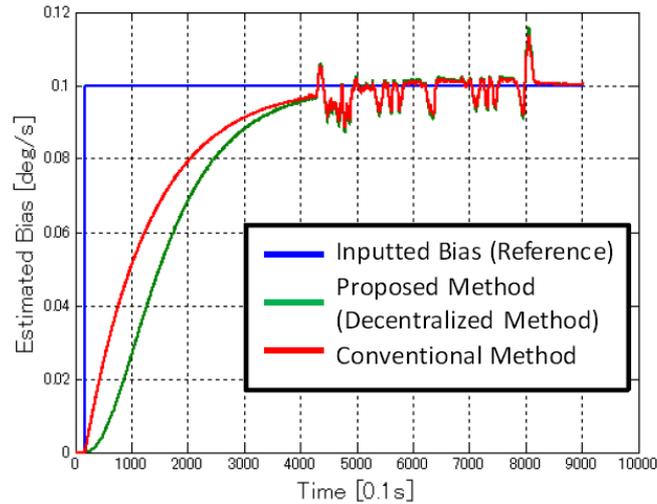


FIGURE 15. Comparison between decentralized method and conventional method

TABLE 1. The error of estimated bias

	Reference	Conventional method	Proposal method
The convergence rate (RMS)	0	0.1217	0.1392
The last convergence level [deg/s]	0.1	0.1003	0.1004
The average of calculation time [ms]	–	7.5600	5.9645

it is included in the treatment except attitude estimation and the bias estimation, it is thought that the rate of improvement decreased comparing with theoretical value.

Comparing with the conventional method, our proposal method is not inferior of bias estimation performance, but the calculation cost is lower.

**5. Experimental Result.** Next, we checked our sensor bias estimation advantage by comparing positioning errors when our method was not applied.

Using the same sensor data, we estimated the vehicle attitude and position through off-line processing. The vehicle attitude is estimated by the UKF, and the vehicle position by dead reckoning.

Figure 16(a) shows the result of positioning without sensor bias estimation. In this case, the Root Mean Square (RMS) value of dead reckoning error is 9.9 m. Figure 16(b) shows the result of positioning with sensor bias corrected using our method. The RMS value is 5.5 m.

The positioning accuracy was increased by 44% compared with that for non-corrected sensor biases. We found that the low-dimensional UKF with corrected sensor biases estimated attitude and position more accurately.

**6. Conclusions.** In this paper, we discussed about how to suppress the computational effort while maintaining positioning accuracy for application to AV. We proposed a decentralized positioning system that estimates sensor bias in addition to the UKF that estimates AV attitude maintain positioning accuracy and lessen computational load.

We still have unsolved problems, such as how to decrease stationary errors when the bias fluctuation is large, and how to suppress estimating bias fluctuation when AV moves unsteadily.

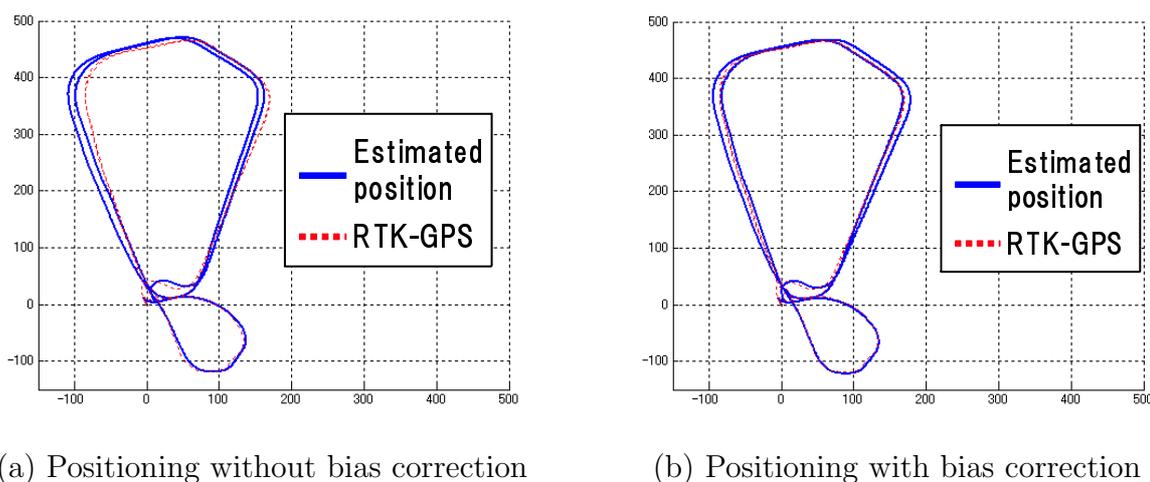


FIGURE 16. Positioning result

**Acknowledgment.** The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

#### REFERENCES

- [1] <http://www.darpa.mil/grandchallenge05/index.html>.
- [2] <http://www.darpa.mil/grandchallenge/index.asp>.
- [3] M. Tanaka, Reformation of particle filters in simultaneous localization and mapping problems, *International Journal of Innovative Computing, Information and Control*, vol.5, no.1, pp.119-128, 2009.
- [4] H. Rehbinder and X. Hu, Drift-free attitude estimation for accelerated rigid bodies, *Automatica*, vol.40, pp.653-659, 2004.
- [5] A. E. Hadri and A. Benallegue, Sliding mode observer to estimate both the attitude and the gyro-bias by using low-cost sensors, *Proc. of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2867-2872, 2009.
- [6] J. J. LaViola Jr., A comparison of unscented and extended Kalman filtering for estimating quaternion motion, *Proc. of 2003 American Control Conference*, 2003.
- [7] T. Aoki, Y. Shimagaki, T. Ikki, M. Tanikawara, S. Sugimoto, Y. Kubo and K. Fujimoto, Cycle slip detection in kinematic GPS with a jerk model for land vehicles, *International Journal of Innovative Computing, Information and Control*, vol.5, no.1, pp.153-166, 2009.
- [8] T. Aoki and S. Sugimoto, Dynamical models for automobile movements, *International Journal of Innovative Computing, Information and Control*, vol.6, no.1, pp.3-14, 2010.
- [9] S. A. Holmes, G. Klein and D. W. Murray, An  $O(N^2)$  square root unscented Kalman filter for visual simultaneous localization and mapping, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.31, no.7, 2009.
- [10] M. S. Pierre and D. Gingras, Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system, *Proc. of IEEE Intelligent Vehicles Symposium*, 2004.
- [11] W. Xue, Y.-Q. Guo and X.-D. Zhang, Application of a bank of Kalman filters and a robust Kalman filter for aircraft engine sensor/actuator fault diagnosis, *International Journal of Innovative Computing, Information and Control*, vol.4, no.12, pp.3161-3168, 2008.
- [12] N. Metni, J. M. Pfimlin and T. Hamel, Attitude and gyro bias estimation for a flying UAV, *Proc. of the 2005 IEEE International Conference on Intelligent Robots and Systems*, 2005.
- [13] C. Fan and Z. You, Highly efficient sigma point filter for spacecraft attitude and rate estimation, *Mathematical Problems in Engineering*, vol.2009, 2009.
- [14] P. D. Burns and W. D. Blair, Sensor bias estimation from measurements of known trajectories, *Proc. of the 37th Southeastern Symposium on System Theory*, pp.373-377, 2005.
- [15] X. Suo, L. Chen and A. Sheng, Asynchronous multi-sensor bias estimation with sensor location uncertainty, *Proc. of 2009 Chinese Control and Decision Conference*, pp.4317-4322, 2009.