Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100205

System Modeling and Non-Linear Estimation Performance Comparison of Monocular Vision based Integrated Navigation System

Dae Hee Won[1], *Sukchang Yun*[2], *Young Jae Lee*[2] *and Sangkyung Sung*[2,*]

[1] Department of Aerospace Engineering Sciences, University of Colorado at Boulder, Boulder, Colorado, USA
[2] Department of Aerospace Information Engineering, Konkuk University, Seoul, Korea

Received: 5 Aug. 2015, Revised: 23 Oct. 2015, Accepted: 24 Oct. 2015
Published online: 1 Mar. 2016

Abstract: This paper presents and compares the performance of nonlinear estimation filters for the inertial SLAM (Simultaneous Localization and Mapping) integrated navigation system including the extended Kalman filter, the unscented Kalman filter and the particle filter. A computer simulation is conducted to analyze the navigation accuracy as well as the capability of real-time implementation by individual filter using a monocular vision based navigation model. The detail model for the linear filter design and the initial delayed localization of the target features were investigated. Simulation results show that the unscented Kalman filter has better performance in perspective of both the navigation performance and the feasibility of real-time implementation.

Keywords: Nonlinear estimation, extended Kalman filter, vision based navigation, inertial SLAM, target features

1 Introduction

Due to the development of a high performance processor and a small-sized vision sensor, active researches have been conducted in the field of vision based navigation. Especially, in mobile robot application, the SLAM (Simultaneous Localization and Mapping), which is a technology that generates a map and identifies its position when the surroundings remain unknown, is widely adapted with the advances on these processors and sensors [1,2]. Focused on the fact that it does not require any prior information on the surroundings and robot position, so the SLAM technology can be utilized as a navigation method for the unmanned aerial vehicle (UAV) that performs surveillance missions for a long period of time in a vast range of areas.

Monocular vision system provides only relative angular measurements between the sensor and the object in image. Even though it does not give range measurements, the system is simple, low-cost, and compact compared with other systems that contain range sensors such as sonar sensor, lidar, and laser range finder. To take these advantages, this paper uses monocular vision system. As the observation model of the monocular vision sensor employs a series of coordinate transformations and trigonometric functions, it inherently incorporates nonlinear model equations. Thus nonlinear estimation is adapted for implementing the monocular vision aided integrated navigation system.

Typically, the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and particle filter (PF) are the most common filters for nonlinear estimation. Especially, there are various cases that use the UKF for a nonlinear and non-Gaussian noise system. Huang [3] adapted the UKF as a navigation filter to study the observability in a planar environment. Wang [4] applied it to feature points tracking problem by applying the unscented particle filter for the integration. Bailey [1] employed the Rao-Blackwellized particle filter (RBPF) to cope with the nonlinearity. Besides, Xianzhong [5] introduced the UKF to further enhance Bailey's RBPF performance. These previous researches show that UKF has been widely used for vision based navigation system since it reflects the nonlinear characteristics sufficiently. However, these researches are not suitable for a 3D navigation, because they considered only planar cases. On the other hands, Kim and Sukkarieh [6,7] constructed a

[*] Corresponding author e-mail: sksung@konkuk.ac.kr

vision based SLAM in 3D environment by using EKF, followed by Chun [8], where the particle filter is employed. Generally, the particle filter decreases estimation error and is suitable for various noise types; however, it may be inappropriate for a real-time implementation since it needs large computational power compared with other nonlinear filters [9].

In this paper, based on the previous works, a complete model derivation and performance comparison between nonlinear filters are studied for the monocular vision based integrated navigation system. Specifically, to characterize the EKF application on the integrated navigation system, a strict mathematical development on linearization is presented for the strong nonlinear observation model. Also the detailed description on the delayed initialization for feature point initial positioning is provided. Based on the mathematical model, the estimation result of EKF is analyzed and the performance is compared with other nonlinear filters. For the performance comparison, this paper also investigates the application of UKF to the given navigation system model. Assumed a high nonlinearity in the observation model, non-Gaussian and non-additive noise model, this paper addresses navigation performances from three nonlinear filters and illustrates computation burdens quantitatively. In section 2, a vision aided integrated navigation system and delayed initialization is introduced. Then nonlinear estimation filers are shown in section 3 with the linearization process. In section 4, simulations are conducted with performance analysis. Finally, a conclusion is made in section 5.

2 Vision Based Navigation System

It is essential to extract easy-to-track feature points from image information obtained from the vision sensor and to track these feature points within the images from consecutive image sequences. Algorithms in [10,11] are generally used for extracting and tracking feature points. Normally, feature point is fixed within local frame and the movements of feature points are correlated only with changes in the attitude and position of the vision sensor. If the vision sensor is fixed on the vehicle and its installed location is known, the movements of feature points can be used reversely in estimating the position and attitude of the vehicle [1]. Figure 1 shows a structure of integrated navigation system. In the implementation, the inertial sensors provide angular velocity and acceleration such that navigation solutions are preliminarily computed. Then, the computed solution is compensated by the tracked feature point data from vision sensor. As vision tracking rate is relatively low, an indirect filter structure is adapted for the system [2,8,12].

2.1 Process Model

The process model consists of the vehicle dynamic model and the feature point model. A linearized low grade inertial navigation system (INS) model is used as the dynamic model for vehicle, while the feature point position is assumed to be a fixed point and its error is therefore modeled as a random constant. A state vector is set to vehicle position (p^n), velocity (v^n), attitude (Ψ^n) and feature point position (m^n) in navigation frame. In here, the superscript n and k denote the navigation frame and the time index. A general low-grade inertial navigation model is given as

$$p^n_{k+1} = p^n_k + v^n_k \cdot \Delta t \tag{1}$$

$$v^n_{k+1} = v^n_k + f^n_k \cdot \Delta t \tag{2}$$

$$\psi^n_{k+1} = \psi^n_k + \omega^n_k \cdot \Delta t. \tag{3}$$

where f^n, ω^n and Δt are specific force, angular velocity and time interval, respectively.

For an indirect filter structure, the error state and its model should be defined. Equation (4) denotes the system model of the vision and INS integrated navigation system [13,14],

$$x_{k+1} = Ax_k + Bw_k \tag{4}$$

where w_k is the process noise, and A and B is defined as

$$A = \begin{bmatrix} 0 & I_{3\times3} & 0 & 0 & & 0 \\ 0 & 0 & [f_b\times] & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & & 0 \\ & \vdots & & & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 \\ C^n_b & 0 \\ 0 & -C^n_b \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}.$$

In here, $I_{3\times3}$ and f_b are 3-dimensional identity matrix and force in the body frame. Note that in this, state vector is defined as error state in (5).

$$x = \left[\, \delta p^n \ \delta v^n \ \delta\psi^n \ \delta m^n_1 \cdots \delta m^n_N \,\right]^T \tag{5}$$

A rotation matrix from the body frame to the navigation frame is denoted by C^n_b. For the feature point, the time update relationship in (6) holds.

$$\delta m^n_i(k+1) = \delta m^n_i(k) \tag{6}$$

2.2 Observation Model

The bearing and elevation of the feature points are utilized to configure an observation model, and these values are determined by both the attitude of the vehicle

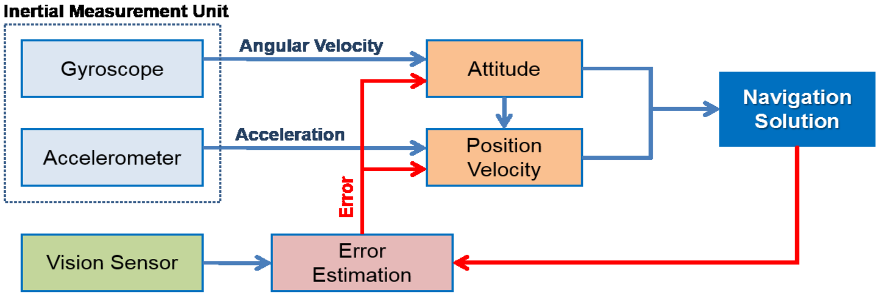

Fig. 1: An indirect estimation filter is constructed to generate the navigation solution. Error state is updated through the vision aided navigation data and combined through master INS algorithm with low update rate relatively.

and the relative position between the vehicle and feature points.

$$z_k = \begin{bmatrix} \varphi_k \\ \vartheta_k \end{bmatrix} = H\left(x_k, \eta_k\right) \qquad (7)$$

where φ_k is bearing angle at k-th epoch, θ_k is elevation angle at k-th epoch, and η_k is a measurement noise. Equation (7) is calculated on image frame (denoted by superscript s), thus a coordinate transformation in (8) is required to compute a solution in the navigation frame.

$$m^s = \begin{bmatrix} x^s \\ y^s \\ z^s \end{bmatrix} = C_b^s\left(C_n^b\left(m^n - p_v^n\right) - p_s\right) \qquad (8)$$

where p_s is a lever arm vector, C_b^n is direction cosine matrix from body frame to navigation frame, and p_v^n is vehicle position vector in navigation frame, and $M^{(\cdot)}$ is feature point position vector in each frame.

Observations are obtained as the following equation of (9), which is obtained by using the geometric relation in the image plane [8,15]. Due to the trigonometric form, the equation is highly nonlinear, thus conventional linear estimation is not suitable to estimate the state vector.

$$z_k = \begin{bmatrix} \tan^{-1}\left(\frac{y^s}{x^s}\right) \\ \tan^{-1}\left(\frac{z^s}{\sqrt{(x^s)^2+(y^s)^2}}\right) \end{bmatrix} + \eta_k \qquad (9)$$

Generally, it is difficult to compute the feature point position using monocular vision. To resolve the problem of distance calculation, this paper adopts the *delayed initialization method* [16], which finds out an intersection point of two line-of-sight vector (LOS) between time intervals. Using this triangulation method enables to compute a 3 dimensional feature point position from two vehicle positions; however, it does not guarantee to fix the feature point all the time since the measurement may not be observable in such case when the vision sensor moves to the optical axis direction. Figure 2 shows basic concept of this method, in which the intersection point is computed by two different vehicle positions, y_0^n and y_t^n and two LOS vectors, u_0^n and u_t^n.

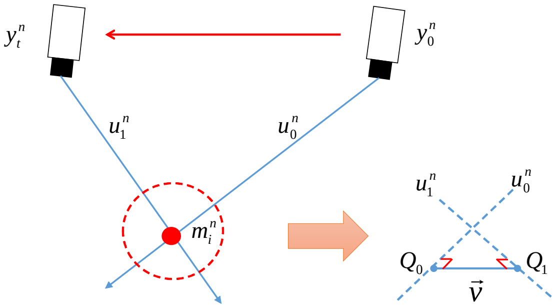

Fig. 2: Conceptual diagram for the feature point position on delayed initialization method. Through the triangulation of LOS vectors, initial feature point position is estimated. The average of two closest points regards as the position.

$$\begin{aligned} m_i^n &= G\left(\bullet\right) = \tfrac{1}{2}\left(y_0^n + y_t^n + p_0 u_0^n + p_t u_t^n\right) \\ &= \tfrac{1}{2}\left(Q_0 + Q_t\right) \end{aligned} \qquad (10)$$

$$y^n = P^n + C_b^n P_{sb}^b \qquad (11)$$

$$u^n = C_b^n C_s^b \bar{P}_{ms}^s \qquad (12)$$

$$\bar{P}_{ms}^s = \begin{bmatrix} \cos\left(\varphi\right)\cos\left(\vartheta\right) \\ \sin\left(\varphi\right)\cos\left(\vartheta\right) \\ \sin\left(\vartheta\right) \end{bmatrix} \qquad (13)$$

$$Q_0 = y_0^n + p_0 u_0^n, \quad Q_t = y_t^n + p_t u_t^n \qquad (14)$$

$$u_0^n \cdot \mathbf{v} = 0, \quad u_t^n \cdot \mathbf{v} = 0 \qquad (15)$$

$$\mathbf{v} = Q_0 - Q_t \qquad (16)$$

In many practical cases, two LOS vectors may not meet at one point as there exist various error sources such as image distortion or sensor miss-alignment, and etc.

Therefore, it finds the closest points Q_0 and Q_t which is orthogonal to $\mathbf{v}$. Then the average of two closest points in Equation (17) and (18) regards as the feature point position.

$$p_0 = \frac{\left(\left(y_t^n - y_0^n\right) \times u_t^n\right) \cdot \left(u_0^n \times u_t^n\right)}{\left|u_0^n \times u_t^n\right|^2} \tag{17}$$

$$p_t = \frac{\left(\left(y_0^n - y_t^n\right) \times u_0^n\right) \cdot \left(u_t^n \times u_0^n\right)}{\left|u_t^n \times u_0^n\right|^2} \tag{18}$$

3 Estimation Filter Modeling and Implementation

3.1 Extended Kalman Filter (EKF)

Basic structure of EKF is similar to conventional Kalman filter; however, it uses a partially linearized model so it can cope with non-linear model. Even though it has simple structure and low computation load, it cannot guarantee filter stability under highly non-linear model and non-Gaussian noise because a basis of EKF is linear model [9,17]. In every time step, linearization is conducted at nominal point. Equation (19) shows linear observation model of EKF.

$$z_k = H\left(x_k, u_k\right) \approx \nabla H_k x_k + u_k \tag{19}$$

$$\nabla H_k = \begin{bmatrix} \frac{\varphi_k}{\partial p^n} & \frac{\varphi_k}{\partial v^n} & \frac{\varphi_k}{\partial \psi^n} \\ \frac{\vartheta_k}{\partial p^n} & \frac{\vartheta_k}{\partial v^n} & \frac{\vartheta_k}{\partial \psi^n} \end{bmatrix} = \frac{\partial z_k}{\partial M_k} \frac{\partial M_k}{x_k} \tag{20}$$

where the partial derivative is given by

$$\frac{\partial z_k}{\partial M_k^s} = \begin{bmatrix} \frac{-(y^s)^2}{(x^s)^2+(y^s)^2} & \frac{-x^s z^s}{\left((x^s)^2+(y^s)^2+(z^s)^2\right)\sqrt{(x^s)^2+(y^s)^2}} \\ \frac{x^s}{(x^s)^2+(y^s)^2} & \frac{-y^s z^s}{\left((x^s)^2+(y^s)^2+(z^s)^2\right)\sqrt{(x^s)^2+(y^s)^2}} \\ 0 & \frac{\sqrt{(x^s)^2+(y^s)^2}}{(x^s)^2+(y^s)^2+(z^s)^2} \end{bmatrix}^T \tag{21}$$

and

$$\frac{\partial M_k^s}{\partial x_k} = \frac{\partial\left(C_b^s\left(C_n^b\left(M_k^n - p_v^n\right) - p_s\right)\right)}{\partial x_k} \tag{22}$$

Delayed initialization is essential for monocular vision system; however, it should take complex linearization process. Next equations are used for covariance update by using linear model, when there is additional feature point.

$$P_{Aug} = \begin{bmatrix} I & 0 \\ \nabla G_X & \nabla G_Z \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & R_{2\times 2} \end{bmatrix} \begin{bmatrix} I & 0 \\ \nabla G_X & \nabla G_Z \end{bmatrix}^T \tag{23}$$

$$\nabla G_X = \begin{bmatrix} \frac{\partial G}{\partial Pos} & \frac{\partial G}{\partial Vel} & \frac{\partial G}{\partial Att} & \frac{\partial G}{\partial FP} \end{bmatrix} \tag{24}$$

$$\nabla G_Z = \begin{bmatrix} \frac{\partial G}{\partial \varphi} & \frac{\partial G}{\partial \vartheta} \end{bmatrix} \tag{25}$$

where each partial derivative is represented by the following equation,

$$\frac{\partial G}{\partial Pos} = \frac{1}{2}\left(\begin{array}{l} I + \frac{\left(u_0^n \times u_t^n\right)}{\left|u_0^n \times u_t^n\right|^2} \begin{bmatrix} 0 & u_{tz} & -u_{ty} \\ -u_{tz} & 0 & u_{tx} \\ u_{ty} & -u_{tx} & 0 \end{bmatrix} u_0^n \\ -\frac{\left(u_t^n \times u_0^n\right)}{\left|u_t^n \times u_0^n\right|^2} \begin{bmatrix} 0 & u_{0z} & -u_{0y} \\ -u_{0z} & 0 & u_{0x} \\ u_{0y} & -u_{0x} & 0 \end{bmatrix} u_t^n \end{array} \right) \tag{26}$$

$$\frac{\partial G}{\partial Vel} = 0 \tag{27}$$

$$\frac{\partial G}{\partial Att} = \frac{1}{2}\left(\frac{\partial p_0}{\partial u_t}\frac{\partial u_t}{\partial Att}u_0^n + \frac{\partial p_t u_t^n}{\partial u_t^n}\frac{\partial u_t^n}{\partial Att}\right) \tag{28}$$

$$\frac{\partial G}{\partial FP} = 0 \tag{29}$$

$$\nabla G_Z = \tfrac{1}{2}\left(\begin{array}{l} \frac{\partial p_0}{\partial u_t} C_s^n \begin{bmatrix} -\sin(\varphi)\cos(\vartheta) & -\cos(\varphi)\sin(\vartheta) \\ \cos(\varphi)\cos(\vartheta) & -\sin(\varphi)\sin(\vartheta) \\ 0 & \cos(\vartheta) \end{bmatrix} u_0^n \\ +\frac{\partial p_t u_t^n}{\partial u_t} C_s^n \begin{bmatrix} -\sin(\varphi)\cos(\vartheta) & -\cos(\varphi)\sin(\vartheta) \\ \cos(\varphi)\cos(\vartheta) & -\sin(\varphi)\sin(\vartheta) \\ 0 & \cos(\vartheta) \end{bmatrix} \end{array} \right). \tag{30}$$

3.2 Unscented Kalman Filter (UKF)

EKF has several limitations; first, it cannot guarantee convergence because it neglects the high order term in the linearization. Also the estimate can diverge when the nominal point is not close to the estimated state vector. UKF uses nonlinear equation without linearization, and several sigma points instead of mean and covariance. Multiple sigma points describe mean and covariance, then they are used to estimate the state vector, we call it Unscented Transform (UT) [18,19,20]. Equations from (31) to (33) show the sigma points and respective weights. The basic algorithm of UKF has similar two phase processes as in the EKF (Prediction-Update) case. However, state vector is estimated by several sigma points and their weights.

$$\chi_0 = \bar{x},\ W_0 = k/\left(n+k\right)a \tag{31}$$

$$\chi_i = \bar{x} + \left(\sqrt{\left(n+k\right)P_{xx}}\right)_i, W_i = 1/2\left(n+k\right) \tag{32}$$

$$\chi_{i+n} = \bar{x} - \left(\sqrt{\left(n+k\right)P_{xx}}\right)_i, W_{i+n} = 1/2\left(n+k\right) \tag{33}$$

where χ is sigma point, $k \in R$ is a design parameter, $\bar{x}$ is the mean of state vector, P_{xx} is covariance of x, and n is size of state vector.

3.3 Particle Filter (PF)

In general, the PF generates distributed particles as state vector candidates based on a priori probability, and then estimates state vector based on particle weights. Through an individual calculation on a large number of distributed particles, the nonlinear mapping and noise distribution can be effectively reflected during state estimation process [9,21]. As the number of particles is increased, the estimation accuracy is increased and computation load is also increased. Even though it is a powerful tool for a nonlinear and non-Gaussian model, it contains a trade-off feature between estimation accuracy and computational load for effective estimation. The filter basically contains the weight calculation step as in (34), computed from the measurement and a priori estimates. Then based on weights, particles with low weight value is eliminated and multiple particles are reproduced in accordance with the weight value. In the filter implementation, the re-sampling step is also included to prevent the particle degeneracy problem and to keep the particle diversity.

$$w_k(i) = \frac{p\left(z_k|x_{k|k-1}(i)\right)}{\sum_{j=1}^{N} p\left(z_k|x_{k|k-1}(j)\right)} \tag{34}$$

4 Simulation and Analysis

Simulation is conducted to compare and analyze the estimation performance in vision aided navigation system. Considering the flight settings of an unmanned aerial vehicle (UAV), the simulation environment assumes a hexahedron as the trajectory space for the initial feature point distribution. Figure 3 shows the area where 40 feature points are randomly distributed. The trajectory is a combination of straight flight, circular turning, accelerating and elevating sections. With an initial velocity set to 10 m/s, the simulated vehicle flies along the designed path for 57.68 seconds in total [8]. The inertial sensor used in the simulation assumes a micro electro mechanical system (MEMS) grade inertial measurement unit (IMU). A vision sensor is assumed to be installed with a 45 degrees tilt angle from local horizontal plane. The maximum number of available feature points is set to seven. The specification of sensors used in the simulation is shown in Table 1.

Table 1: Sensor specification used for simulation.

Sensor	Parameter	Value (1σ)
Inertial sensor	Gyroscope bias instability	1 deg/sec
	Accelerometer bias instability	1 m/sec^2
Vision sensor	Field of view	50 deg
	Bearing estimation error	0.5 deg

During the simulation, the position, velocity and attitude are assumed to be estimated with an update rate of 100 Hz. The update rate for feature points is set to 10 Hz. In particle filter implementation, the number of particles is set to 200 per feature point, so the maximum number of particles is 1400 at 7 feature points.

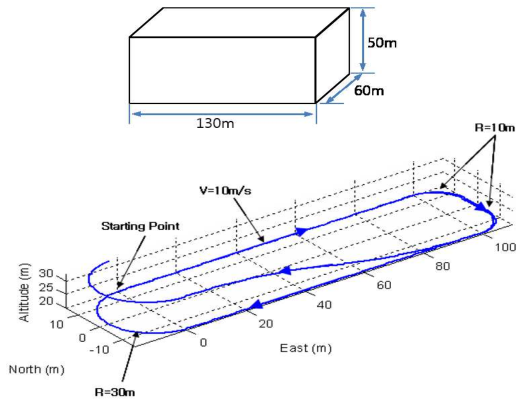

Fig. 3: Space of flight area where the feature points are randomly distributed during simulation demonstration (top) and flight trajectory (bottom).

Figure 4 indicates the true trajectory and estimated position by INS only navigation. 'INS Only' trajectory has accumulated errors which are left uncompensated and diverge. By adapting the vision based navigation as mentioned in section 2, the divergences can be greatly degraded. Figure 5 shows the results of three estimation filters. In all cases, the estimation accuracy is much improved in comparison with 'INS Only' case in Figure 4, however the estimation curve with respect time has mutually different phase for 3 filters. Figure 6 illustrates the position estimation error from the adapted filters. The PF has the best estimation accuracy among three methods as it can better reflect a full nonlinear model with a sufficiently large number of particles than the sigma points in UKF. UKF also assumes a nonlinear model, yet the number of sigma points is limited in proportional of the state dimension. The result of EKF shows the worst accuracy owing to the rough approximation in linear model.

Additionally, a random walk noise distribution is assumed in the observation model to analyze the effect of non-Gaussian noise on filter characteristics. Figure 7 presents the true and 'INS Only' estimation result while Figure 8 and Figure 9 show the estimated trajectory and error on nonlinear filters under random walk noise. PF still has the best performance as the filter process is the least affected even if the Gaussian assumption is applied. 'INS Only' result has lager error than Figure 4 due to random walk. As there is no absolute information for error correction, all estimation filters tends to increase estimation errors as time elapses; yet the drift curve differs according to the filter type. Further simulation result reveals that the PF is most robust in error drift suppression.

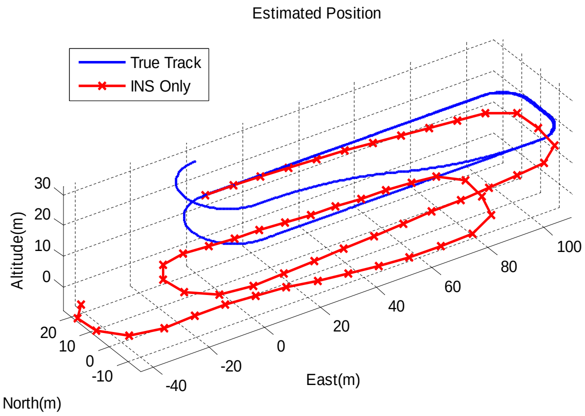

Fig. 4: Estimation result using INS only case. It is compared with true track of flight.

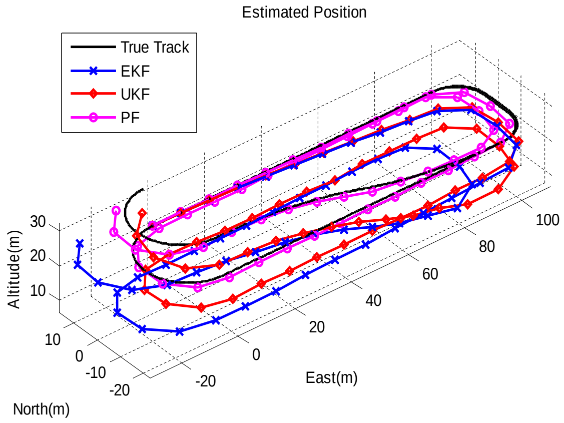

Fig. 5: Estimation result obtained from EKF, UKF, and particle filter. Results are compared with the true trajectory

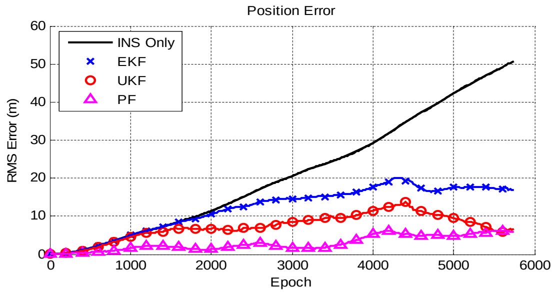

Fig. 6: Estimation error profile during flight intervals. Pure INS based estimation diverges with time elapse.

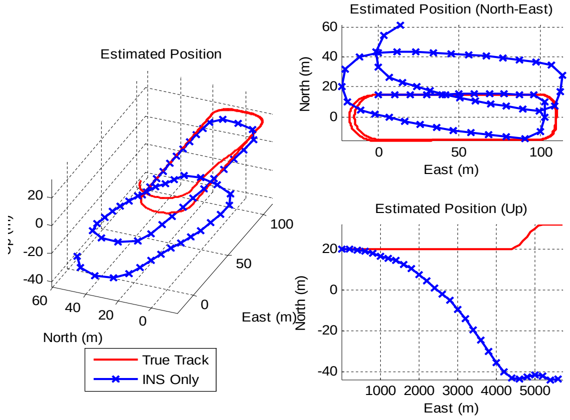

Fig. 7: Estimation result of pure INS case when the noise is applied as random walk signal: (left) 3 dimensional results, (right-top) horizontal plane and (right-bottom) vertical plane.

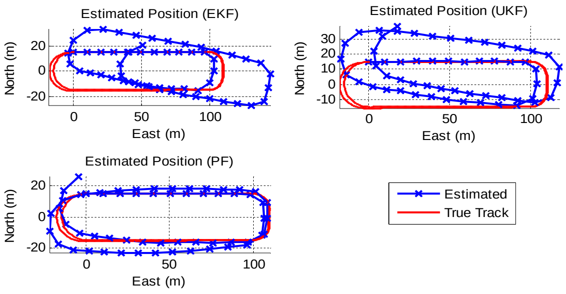

Fig. 8: Estimation result of nonlinear filters when the noise is applied as random walk signal: (left-top) EKF, (left-bottom) PF and (right) UKF.

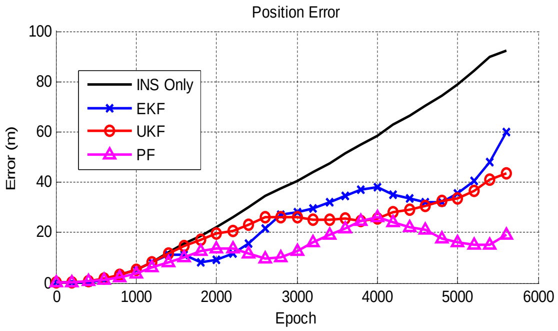

Fig. 9: Error profile of the estimated position when the noise is applied as random walk signal.

Table 2: Computation time (Intel Core 2 Duo 3.00 GHz, MATLAB R2008a).

Filter type	EKF	UKF	PF
Computation time	7 sec	15 sec	72 sec

Table 2 shows computation time for each filter. As expected, the computational burden of PF is heavier than others, despite advantage in estimation accuracy. Note that the flight time is about 58 sec, thus the estimation using PF (72 sec) cannot meet the requirement of real-time operation. Meanwhile, UKF uses only 20% of the time spent by the PF and can satisfy real-time estimation performance.

5 Conclusion

This paper analyzed performance of the vision-aided integrated navigation system via various nonlinear estimation filters. To compute navigation solution under nonlinear and non-Gaussian noise model, three nonlinear filters including the EKF, UKF and PF are adapted and position accuracy along with computational load is analyzed via computer simulation.

In viewpoint of the accuracy, PF has the best performance due to enough particles to reflect nonlinear property of model and non-Gaussian noise distribution, which, in turn, suffers from heavy computational burden. EKF typically generates fastest estimation result with the lightest computation burden, yet the estimation accuracy is the worst due to strong nonlinearity in the model. Meanwhile, UKF compromises the accuracy and the computation time. By adapting a limited number of well chosen deterministic sigma points and passing the covariance through the unscented transform, it is revealed to provide comparable accuracy with that of PF under non-Gaussian noise environment for the vision-aided integrated navigation problem. Also when considering the feasibility in real-time implementation, the computational burden of UKF is fairly acceptable.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF-2013R1A1A1006401).

References

[1] H. Durrant-Whyte, T, Bailey, Simultaneous Localization and Mapping-Tutorial: Part I, IEEE Robotics & Automation Magazine **13**, 99-110 (2006).

[2] J. Kim. Autonomous Navigation for Airborne Application, University of Sydney PH.D Thesis (2004).

[3] G.P. Huang, A.I. Mourikis, S.I. Roumeliotis, On the Complexity and Consistency of UKF-based SLAM, Proc. of IEEE International Conference on Robotics and Automation, Kobe, Japan, 4401-4408 (2009).

[4] X. Wang, H. Zhang, A UPF-UKF Framework for SLAM, Proc. of IEEE International Conference on Robotics and Automation, Roma, Italy, 1664-1669 (2007).

[5] C. Xianzhong, An Adaptive UKF-Based Particle Filter for placeMobile Robot SLAM, Proc. of International Conference on Artificial Intelligence, Hainan, China, 167-170 (2009).

[6] J. Kim, S. Sukkari, SLAM aided GPS/INS Navigation in GPS Denied and Unknown Environments, Proc. of International Symposium on GNSS/GPS, Sydney, Australia (2004).

[7] J. Kim, S. Sukkari, Autonomous Airborne Navigation in Unknown Terrain Environments, IEEE Transaction on Aerospace and Electronic Systems **40**, 1031-1045 (2004).

[8] S. Chun, Performance Improvement of INS/GPS Using Multiple Vision Sensors, Konkuk University PH.D Thesis (2007).

[9] B. Ristic, Beyond the Kalman Filter, Artech House (2004).

[10] G. Wang, H. Zhang, Good Image Feature for Bearing-only SLAM, Proc. Of IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2576-2581 (2006).

[11] C. Tomasi, T. Kanade, Detection and Tracking of Point Features, Technical Report CMU-CS-91-132 (1991).

[12] D.H. Titterton, Strapdown Inertial Navigation Technology, Peter Peregrinus Ltd. (1997).

[13] M. Bryson, S. Sukkarieh, Bearing-Only SLAM for an Airborne Vehicle, Proc. Of Australasian Conference on Robotics and Automation, Brisbane, Australia (2007).

[14] T. Vidal-Calleja, M. Bryson, S. Sukkarieh, On the Observability of Bearing-only SLAM, Proc. Of IEEE International Conference on Robotics and Automation, Roma, Italy, 4114-4119 (2007).

[15] D.H. Won, S. Chun, S. Sung, T. Kang and Y.J. Lee, Improving Mobile Robot Navigation Performance using Vision based SLAM and Distributed Filters, Proc. Of International Conference on Control, Automation and Systems 2008, Seoul, Korea, 186-191 (2008).

[16] T. Bailey, Constrained Initialization for bearing-only SLAM, Proc. Of IEEE International Conference on Robotics and Automation, Taipei, Taiwan (2003).

[17] G. Welch, G. Bishop, An Introduction to the Kalman Filter, UNC-Chapel Hill, TR 95-041 (2006).

[18] S.J. Julier, J.K. Uhlmann, New extension of the Kalman filter to nonlinear systems, Proc. SPIE, **3068**, 182-193 (1997).

[19] E.A. Wan, R. Merwe, The Unscented Kalman Filter for Nonlinear Estimation, Proc. Of Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000, Alberta, Canada, 153-158 (2000).

[20] K. Xiong, H.Y. Zhang, C.W. Chan, Performance evaluation of UKF-based nonlinear filtering, Automatica **42**, 261-270 (2006).

[21] D.H. Won, S. Chun, S. Sung, Y.J. Lee, J. Cho, J. Joo, J. Park, INS/vSLAM system using distributed particle filter, International Journal of Control, Automation and System **8**, 1232-1240 (2010).

Dae Hee Won received his B.S., M.S. and Ph.D. degrees in Aerospace Information Engineering from Konkuk University, Korea, in 2006, 2008 and 2012, respectively. Currently, he is a research associate in the Department of Aerospace Engineering Sciences at University of Colorado at Boulder since September 2013. His research interests include multiple sensory integration for navigation, precise GPS positioning, and nonlinear estimation.

Sukchang Yun is a Ph.D. candidate in the Department of Aerospace Information Engineering at Konkuk University, Korea. He received the M.S degree in Aerospace Engineering from Konkuk University in 2009. His research interests include INS/GPS integrated navigation, INS-based self-contained integrated navigation system and their H/W implementations.

Young Jae Lee is a Professor in the Department of Aerospace Information Engineering at Konkuk University, Korea. He received his Ph.D. degree in Aerospace Engineering from the University of Texas at Austin in 1990. His research interests include integrity monitoring of GNSS signal, GBAS, RTK, attitude determination, orbit determination, and GNSS-related engineering problems.

Sangkyung Sung received his B.S. and Ph.D. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 1996 and 2003, respectively. From March 1996 to February 2003, he worked for the Automatic Control Research Center in Seoul National University. Currently, he is an Associate Professor of the Department of Aerospace Information Engineering, Konkuk University. His research interests include IT fusion avionics, inertial sensors, integrated navigation, and application to unmanned systems.