**Journal of Applied Sciences, Engineering and Technology for Development**
***JASETD, Volume 2, Issue 2, Sept. 2017***

**Research Article** **ISSN 2309-0936**

# COMPARISON OF GPS SMOOTHENING METHODS BETWEEN EXTENDED KALMAN FILTER AND PARTICLE FILTER FOR UAV

**MATSUSHITA[1], K., ITO[1], S., KATO[1], S., NGETHA[2], H., & SASAKI[1*], M.**

[1]Department of Mechanical Engineering, Gifu University, 1-1 Yanagido, Gifu, 501-1193 Japan.
[2]Department of Electrical & Electronic Engineering, Dedan Kimathi University of Technology, P.O. Box 657-10100 Nyeri, Kenya.

*Received January 2017; Revised July 2017; Accepted Sept. 2017*

**ABSTRACT**

*This paper presents a method for smoothing GPS data from a UAV using Extended Kalman filtering and particle filtering for navigation or position control. A key requirement for navigation and control of any autonomous flying or moving robot is availability of a robust attitude estimate. Consider a dynamic system such as a moving robot. The unknown parameters, e.g., the coordinates and the velocity, form the state vector. This time dependent vector may be predicted for any instant time by means of system equations. The predicted values can be improved or updated by observations containing information on some components of the state vector. The whole procedure is known as Kalman filtering. On the other hand, the particle filtering algorithm is to perform a recursive Bayesian filter by Monte Carlo simulations. The key is to represent the required posterior density function by a set of random samples, which is called particles with associated weights, and to compute estimates based on these samples as well as weights. We compare the two GPS smoothening methods: Extended Kalman Filter and Particle Filter for mobile robots applications. Validity of the smoothing methods is verified from the numerical simulation and the experiments. The numerical simulation and experimental results show the good GPS data smoothing performance using Extended Kalman filtering and particle filtering.*

***Keywords*:** GPS, Extended Kalman filter, data smoothing and estimation navigation, UAV, Mobile robots.

*Correspondence to: **Minoru Sasaki**, Department of Mechanical Engineering, Gifu University, 1-1 Yanagido, Gifu, 501-1193 Japan. Email: sasaki@gifu-u.ac.jp

## INTRODUCTION

This paper presents a method for smoothing data from a UAV using Extended Kalman filter and Particle filter for navigation or position control. A key requirement for navigation and control of any autonomous flying or moving vehicle is availability of a robust attitude estimate. Consider a dynamic system such as a moving vehicle. The unknown parameters, e.g., the coordinates and the velocity, form the state vector. This time dependent vector may be predicted for any instant $t$ by means of system equations. The predicted values can be improved or updated by observations containing information on some components of the state vector. The whole procedure is known as Kalman filtering (Lewis et al., 2007; Xu, 2003; Rigatos, 2011). It corresponds to sequential adjustments in the static case. Consequently, optimal estimations of the unknowns on the basis of all observations up to the epoch *t* are obtained. The Global Positioning System was conceived as a ranging system from known positions of satellites in space to unknown positions on land, sea, in air and space. Effectively, the satellite signal is continually marked with its transmission time to that when received, the signal transit period can be measured with a synchronized receiver. Apart from point positioning, the determination of a vehicle's instantaneous position and velocity, and the precise coordination of time were original objectives of the GPS. The most widely used algorithm for multisensory fusion is the Extended Kalman Filter (EKF); however this is based on linearization of the system dynamics, which results in a suboptimal application of the recursive estimation of the standard Kalman Filter. Moreover, the EKF follows the assumption of Gaussian process/measurement noise which does not always hold. These can seriously affect the performance of the state estimation and even lead to divergence. Consequently, the performance of a control loop that uses an EKF-based estimate of the system's state vector can, in some cases, be unsatisfactory. To overcome the EKF flaws, a different approach to state estimation of nonlinear dynamical systems is proposed such as Particle Filter. The Particle Filter (PF) is a non-parametric state estimator which unlike the EKF does not make any assumption on the probability density function of the measurements. The concept of particle filtering comes from Monte-Carlo methods. The Particle Filter has improved performance over the established nonlinear filtering approaches (e.g. the EKF), since it can provide optimal estimation in nonlinear non-Gaussian state-space models. Particle filters can approximate the system's state sufficiently when the number of particles (estimations of the state vectors which evolve in parallel) is large. The PF also avoids the calculations associated with the Jacobians which appear in the EKF equations. The main stages of the PF are prediction (time update), correction (measurement update) and resampling for substituting the unsuccessful state vector estimates with those particles that have better approximated the real state vector. Comparing EKF and PF methods, the latter require more sample points to approximate the state distribution. However, the PF is a nonparametric filter which can be applied to any kind of state distribution, while the EKF state estimators are still based on the assumption of a Gaussian process and measurement noise. Validity of the smoothing methods is verified from the numerical simulations and the experiments. The numerical simulation and experimental results show the good GPS data smoothing performance. We compare between the two GPS smoothening methods: Extended

Kalman Filter and Particle Filter for mobile robot applications. Validity of the smoothing methods is verified from the numerical simulations and the experiments. The numerical simulation and experimental results show the good GPS data smoothing performance using Extended Kalman filtering and particle filtering.

**SYSTEM FORMULATION**

It corresponds to the sequential adjustment in the static case. Consequently, optimal estimations of the unknowns on the basis of all observations up to the epoch *t* are obtained. The time dependent state vector *X[t]* comprising the unknown parameters of the dynamic system may be modeled by a system of differential equations of the first order as

$$X[t] = [x(t), y(t), z(t), v_x(t), v_y(t), v_z(t), a_x(t), a_y(t), a_z(t)]^T \tag{1}$$

$$x(t+1) = x(t) + v_x \Delta T + a_x \Delta T^2/2 \tag{2}$$

$$y(t+1) = y(t) + v_y \Delta T + a_y \Delta T^2/2 \tag{3}$$

$$z(t+1) = z(t) + v_z \Delta T + a_z \Delta T^2/2 \tag{4}$$

$$v_x(t+1) = v_x(t) + a_x(t)\Delta T \tag{5}$$

$$v_y(t+1) = v_y(t) + a_y(t)\Delta T \tag{6}$$

$$v_z(t+1) = v_z(t) + a_z(t)\Delta T \tag{7}$$

Consider the following dynamical system:

$$X[t+1] = F[t]X[t] + G[t]W[t] \tag{8}$$

And the observation equation:

$$Y[t] = H[t]X[t] + V[t] \tag{9}$$

where,

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 & \frac{\Delta T^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & \frac{\Delta T^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & \frac{\Delta T^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$\mathrm{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

Where *W[t]* is the noise input and *V[t]* is the measurement noise. We shall adopt the following assumptions for the process and measurement noises.

$$\mathrm{E}\{W[t]\} = 0, \ \mathrm{E}\{V[t]\} = 0 \tag{13}$$

$$\mathrm{E}\left\{\begin{bmatrix} W[t] \\ V[t] \end{bmatrix} \begin{bmatrix} W^T[t] & V^{\mathrm{T}}[t] \end{bmatrix}\right\} = \begin{bmatrix} \mathrm{Q[t]} & 0 \\ 0 & R[t] \end{bmatrix} \delta[t] \tag{14}$$

$$\mathrm{E}\{W[t] \quad X^T[s]\} = 0, \quad \mathrm{E}\{\upsilon[t] \quad X^{\mathrm{T}}[s]\} = 0 \tag{15}$$

Where $\delta_{ts}$ is the Kronecker delta

$$\delta_{ts} = \begin{cases} \delta_{ts} = 1 \ (t = s) \\ \delta_{ts} = 0 \ (t \neq s) \end{cases} \tag{16}$$

The filter equation is given by

$$\hat{X}[t+1 \mid t] = F[t \mid t]\hat{X}[t \mid t] \tag{17}$$

$$\hat{X}[t \mid t] = \hat{X}[t] + \mathrm{K}[\mathrm{t}]\{\mathrm{Y}[\mathrm{t}]\text{-}\mathrm{H}[\mathrm{t}]\hat{X}[\mathrm{t} \mid \mathrm{t}-1]\} \tag{18}$$

where the Kalman gain is

$$K[t] = \hat{P}[t \mid t-1]H^{T}[t]\left\{H[t]\hat{P}[t \mid t-1]H^{T}[t] + R[t]\right\}^{-1} \tag{19}$$

$$R[t] = I_{3\times3} \tag{20}$$

The estimated covariance matrix is

$$\hat{P}[t+1 \mid t] = F[t]\hat{P}[t \mid t]F^{T}[t] + \frac{{\delta_w}^2}{{\delta_v}^2}\Lambda \tag{21}$$

where

$$\hat{P}[t \mid t] = \hat{P}[t \mid t-1] - K[t]H[t]\hat{P}[t \mid t-1] \tag{22}$$

and

$$\Lambda = GG^{T} = diag\{0,0,0,0,0,0,1,1,1\}$$

$$I_{3\times3} = diag\{1,1,1\} \tag{23}$$

**KALMAN FILTERING ALGORITHM**

In the discrete-time case, the state space model of the system is assumed to be expressed as:

$$x(k+1) = Ax(k) + bv(k) \tag{24}$$

$$y(k) = C^{T}x(k) + w(k) \tag{25}$$

where $A, C$ is n × n matrix, $v(k)$ is the normality white noise with mean 0 and variance $R(k)$, $w(k)$ is the system noise and observation noise with mean 0 and variance $Q(k)$. It is assumed that the process noise and the measurement noise $v(k), w(k)$ are uncorrelated each other. Now the problem is to estimate the state $x(k)$ based on the sequence of output measurement $y(1), y(2), \ldots, y(k)$.The advance estimate of state $x(k)$ at $k$ is

$$\hat{x}^{-}(k) \quad \left(= \hat{x}(k|k-1)\right) \tag{26}$$

The posteriori estimate of state $x(k)$ at $k$ using the data from at $k-1$ is

$$\hat{x}(k) \quad (= \hat{x}(k|k)) \tag{27}$$

Kalman Filter (KF) has prediction step and updating step. The KF recursion is as follows:

1) Prediction step.

Compute the estimate of state $\hat{x}^{-}(k)$ at $k$ from one sampling before posteriori estimate of the state $\hat{x}(k-1)$.

$$\hat{x}^{-}(k) = A\hat{x}(k-1) \tag{28}$$

Compute the prediction error covariance $\Sigma^{-}(k)$ from one sampling before error covariance$\Sigma(k-1)$ and measurement noise covariance $Q(k)$

$$\Sigma^{-}(k) = Q(k) + A\Sigma(k-1)A^{T} \tag{29}$$

2) Updated step
Compute measurement error $\tilde{y}(k)$ from measurement sensor value and estimate value,

$$\tilde{y}(k) = y(k) - C\hat{x}^{-}(k) \tag{30}$$

measurement error covariance $S(k)$.

$$S(k) = C\Sigma^{-}(k)C^{T} + R(k) \tag{31}$$

Kalman gain $K(k)$,

$$K(k) = \Sigma^{-}C^{T}S^{-1} \tag{32}$$

posteriori estimate of the state $\hat{x}(k)$,

$$\hat{x}(k) = \hat{x}^{-}(k) + Ky(k) \tag{33}$$

and noise covariance $\Sigma(k)$.

$$\Sigma(\mathrm{k}) = \Sigma^{-}(k) - KC\Sigma^{-}(k) \tag{34}$$

## PARTICLE FILTERING ALGORITHM

The estimation examples so far have assumed that the error in sensors such odometry and landmark range and bearing have a Gaussian probability density function. In practice we might find that a sensor has a one sided distribution or a multimodal distribution with several peaks. The functions we used in Kalman filter such as eq. (24)-(25) are strongly non-linear which means that sensor noise with a Gaussian distribution would not result in a Gaussian error distribution on the value of the function. The probability density function associated with a robot's configuration could have multiple peaks to reflect several hypotheses that equally well explain the data from the sensors. Particle Filtering (PF) is a method for state estimation that is not dependent on the probability density function of the measurements. The Monte-Carlo estimator makes no assumptions about the distribution of errors. It can also handle multiple hypotheses for the state of the system. We maintain many different versions of the vehicle's or UAV's state vector. When a new measurement is available we score how well each version of the state explains the data. We keep the best fitting states and randomly perturb them to form a new generation of states. Collectively these many possible states and their scores approximate a probability density function for the state we are trying to estimate. Apply the state update to each particle moving each particle according to the measured odometry. If $f(\cdot), g(\cdot)$ are linear and random vectors $v, w$ are Gaussian random variables, they are same as KF.

$$\mathrm{x_t} = f(x_{t-1}) + v_t \tag{35}$$

$$\mathrm{y_t} = g(x_t) + w_t \tag{36}$$

We make an observation $y_t$ of feature $t$ which has, according to the map, coordinate $x_t$. For each particle we compute the innovation. Eq. (35) shows the error between the predicted and actual landmark observation. Select the particles that best explain the observation, a process known as resampling. A common scheme is to randomly select particles according to their weight. Particles with a large weight would correspond to a large fraction of the vertical span of the cumulative histogram and therefore be more likely to be chosen. The result would have the same number of particles. Some would have been copied multiple times, others not at all. Introduction to particle filtering theory and practice with positioning applications has been widely discussed (Gordon, 1993; Doucet, 2000, Doucet, 2001). The particle filtering algorithm is to perform a recursive Bayesian filter by Monte Carlo simulations. The key is to represent the required posterior density function by a set of random samples, which is called particles with associated weights, and to compute estimates based on these samples as well as weights. Its goal is to compute filtered estimates of $x_{0:t}$ taking into account all available measurement up to time $t$, $z_{1:t}$.

In practice, the solution is to recursively obtain a posterior probability density function $p(x_0{:}t|z_{1:t})$ of states at time $t$ given all available measurements. Particle filter represents the posterior probability density function by a set of random samples with associated weight as follows:

$$\mathrm{p}(x_t|z_{1x}) \approx \sum_{i=1}^{N_p} \omega_t^i \cdot \delta(x_t - x_t^i) (37)$$

where each particle with index $I$ has a state $x_t^i$; $\delta(x)$ is Dirac delta function; $\omega_t^i$ is associated weight with $x_t^i$; $N_p$ stands for the particle number; $z_{1:t}$ denotes the measurements accumulated up to $t$. The weight is always positive, $\omega_t^i > 0$, and sum over all weights is equal to 1. If computation load can bear, $N$p is expected large enough. The state of each particle is drawn randomly from the importance sampling distribution, and one choice for the distribution is a prior probability distribution. The state can therefore be represented as

$x_t^i \sim p(x_t | x_{t-1} = x_{t-1}^i) = p(x_t | x_{t-i}^i)$(38)

Where$p(x_t | x_{t-i}^i)$denotes the state transitionprobabilities.

The sequential weight updated at each step can be calculated from

$\omega_t^i = \omega_{t-1}^i \cdot p(z_t | x_t^i)$(39)

where the initial weight is set as $\omega_0^i = 1/N_p$.

(i) Decompose: to perform multi-scale decomposition for original data, and find out a prior distribution. Original data is a certain solution with respect to GPS data processing scheme, which is recognized as a compound signal composed of multipath bias, receiver noise and environmental noise.

(ii) Initialize: to draw new particle$x_0^i$in terms of aprior distribution $p(x_0)$.

(iii) Predict and update: to calculate state $x_t^i$ and to update the weightfor each sampling state $x_{t-1}^i$.

(iv) Denoise and draw new particle: Firstly, denoising with threshold is performed. Then, new particle is to be drawn according to importance sampling. Finally, the new weight is calculated and normalized to sum to unit.

(v) Resample and compute effective number of particles: In general, the threshold *N*th is specified 2/3*N*p. The degree of degeneracy can be assessed by the effective number of particles approximately. Resampling is to be applied below a threshold.

(vi) State output:

(vii) Update with time and draw new particle.

(viii) Set *k* to *k+1*, and then go to step 3).

**EXPERIMENTS AND DISCUSSIONS**

To evaluate the performance of proposed estimation algorithms, an experimental study is described.Table1 shows the specification of controller named Ardupilot Mega 2.6 with gyroscopes, accelerometer, barometer and geomagnetic sensor for positioning experiment. Table 2 shows the specification of GPS system with 5 Hz sampling rate. Figure 1 and 2 show Ardupilot Mega 2.6and GPS module. The sampling interval of GPS receivers is 5 s. The GPS resolution of positioning accuracy is 2.5m. Here, to evaluate proposed algorithm, the differential solution of GPS station was taken as study object. Figures 3 and 4 show the effect of number of satellites in the observation error. Figure 3 is the error histogram of the fixed one point position data in case of the satellite signal number is 5 and Figure 4 is the error histogram of the fixed one point position data in case of the satellite signal number is 10. The attitude data in both cases has no white noise but non-normal distribution. XY position data in case of 5 satellites signals has non-normal distribution however in case of 10 satellites signals has normal distribution. When increasing number of satellite signals probability function of observation error approaches to normal distribution. Figures 5 and 6 show the GPS positioning data using EFK and PF with less than 7 satellites signals. Figure 5 shows XY plane estimated positioning data and Figure 6 shows Z direction estimated positioning data. We held the GPS sensor, walked straight as shown in Figure 7 and estimate the self-position. Table 3 shows the estimated results between using EKF and PF. XY estimated positioning results with KF are better than ones with PF. On the other hand, the estimated altitude positioning results with PF are better than ones with KF. The PF's average error is smaller than the KF's average error. Figures 8 and 9 show the GPS positioning data using EFK and PF with more than 10 satellites signals. Figure 8 shows XY plane estimated positioning data and Figure 9 shows Z direction estimated positioning data. We also held the GPS sensor, walked straight and estimate the self-position. Table 4 shows the estimated results between using EKF and PF. XY estimated positioning results with KF are better than ones with PF. On the other hand, the estimated altitude positioning results with PF are better than ones with KF. However if the number of satellites is increasing, the estimated positioning error has not so much difference between with KF and with PF.

Table 1. Specification of ArdupilotMega 2.6.

| | |
|---|---|
| Computer | ATMEGA2560 |
| Gyro | MPU-6000 |
| Resolution of accelerometer | 0.0001g |
| Resolution of gyro | 0.0305deg/sec |
| Barometer | MS5611 |
| Resolution of barometer | 0.012hPa |
| Geomagnetic sensor | HMC5883L |
| Resolution of geomagnetic sensor | 0.92mG(1～2deg) |
| Sourse voltage | DC5V |
| Dimension | 150 mmX120 mmX20 mm |
| Weight | 71g |

Table 2. Specification of GPS module.

| | |
|---|---|
| Positioning engine | U-blox Co. Ltd LEA-6H |
| Sampling rate | 4Hz |
| Warmup time | 26sec |
| Position accuracy | 2.5m(CEP) |
| SBAS position accuracy | 2.0m(CEP) |
| Source voltage | DC2.7〜3.6V |
| Dimension | φ55 mmX 10mm |
| Weight | 35g |

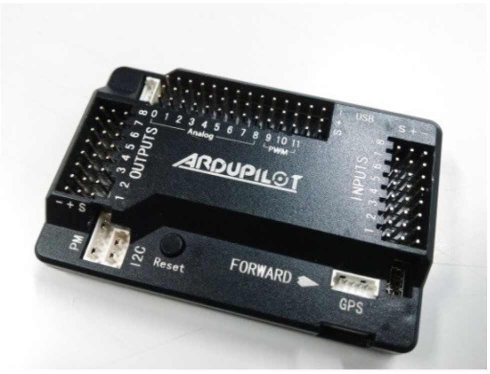


Figure 2  ArduPilotMega2.6.


Figure 1 GPS module.

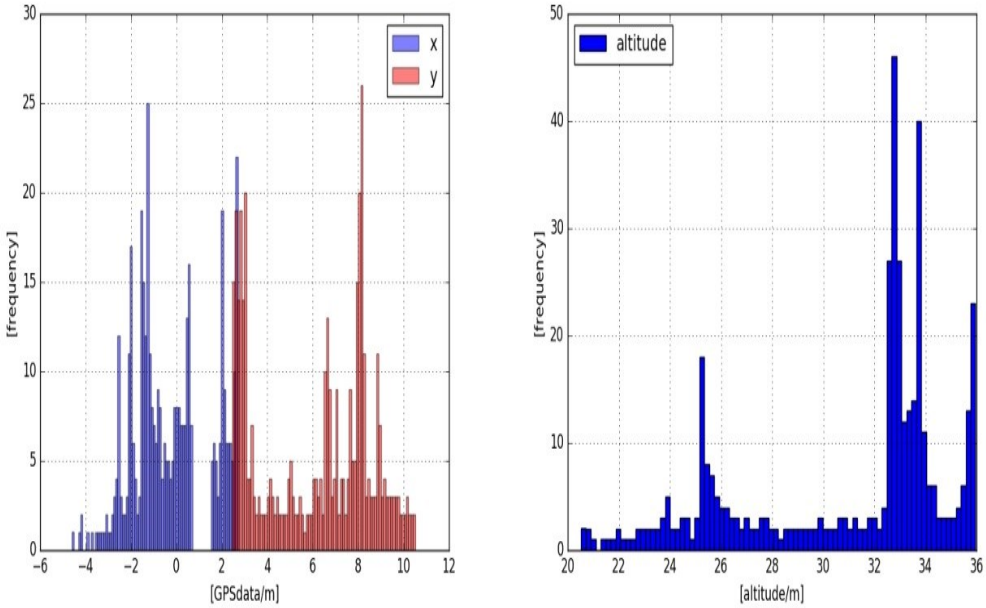


Figure 3 The histogram of noise when receiving 5 artificial satellites signals.(Left : Coordinate, Right : Altitude).

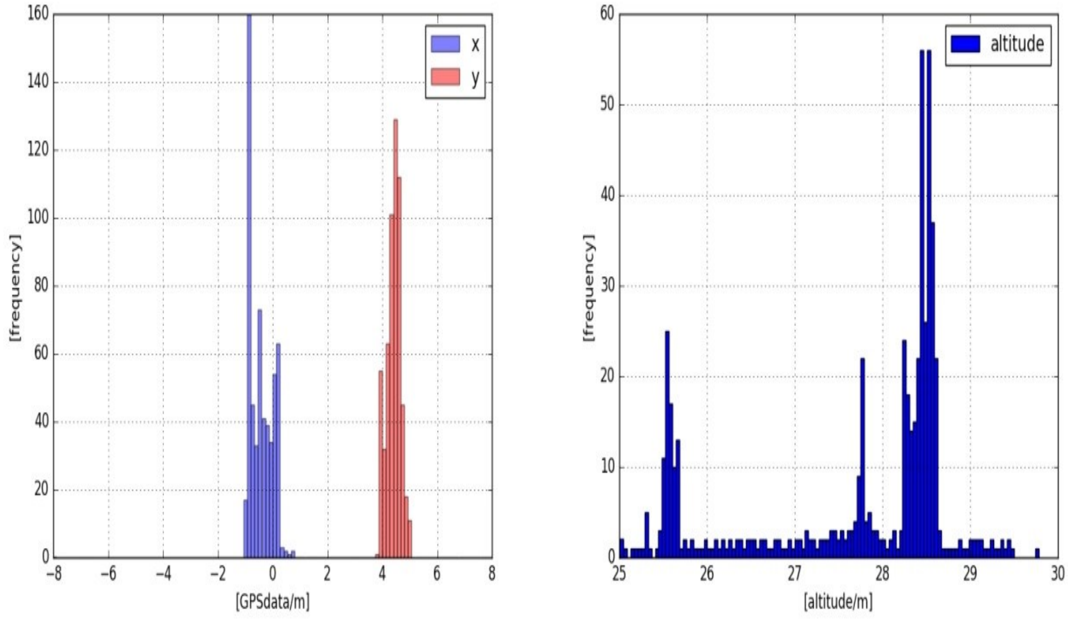


Figure 4 The histogram of noise when receiving10 artificial satellites signal. (Left: Coordinate, Right: Altitude)

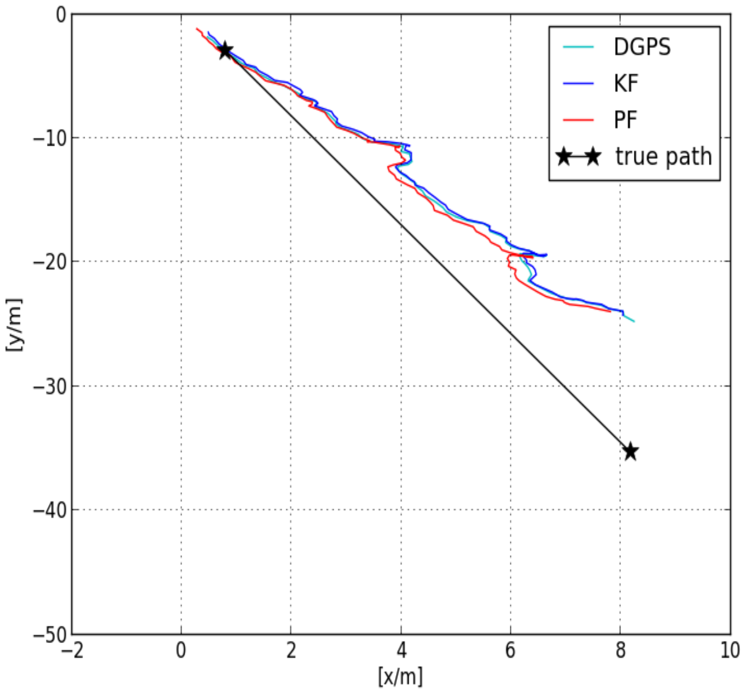


Figure 5 The estimation results of plane position when receiving less than 7 artificial satellites signals.

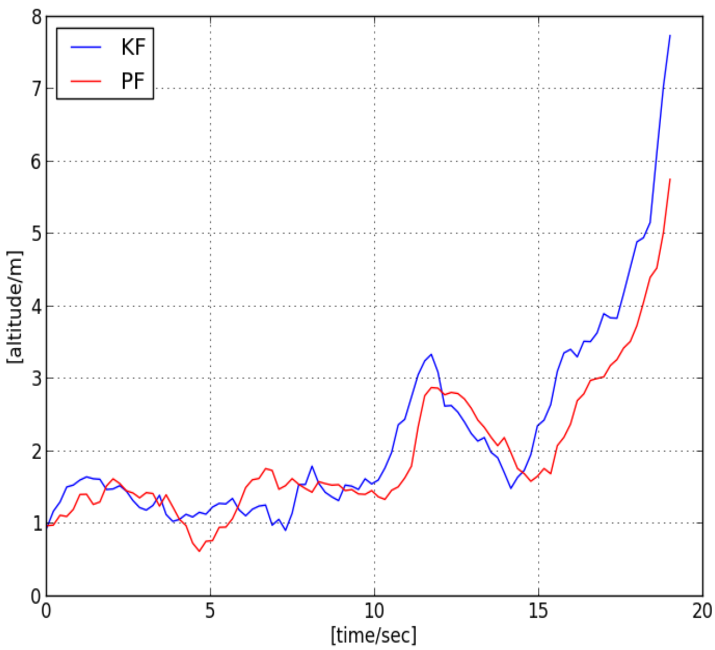


Figure 6 The estimation results of altitude position when receiving less than 7 artificial satellites.

Table 3 The estimated positioning results when receiving less than 7 artificial satellites.

| | |
|---|---|
| *KF error average [m]* | *1.629* |
| *PF error average [m]* | *1.113* |
| *KF maximum error [m]* | *5.733* |
| *PF maximum error [m]* | *4.681* |
| *KF error average of x coordination [m]* | *1.329* |
| *PF error average of x coordination [m]* | *1.601* |
| *KF error average of y coordination [m]* | *10.899* |
| *PF error average of y coordination [m]* | *11.087* |
| *KF altitude error average [m]* | *2.195* |
| *PF altitude error average [m]* | *1.949* |

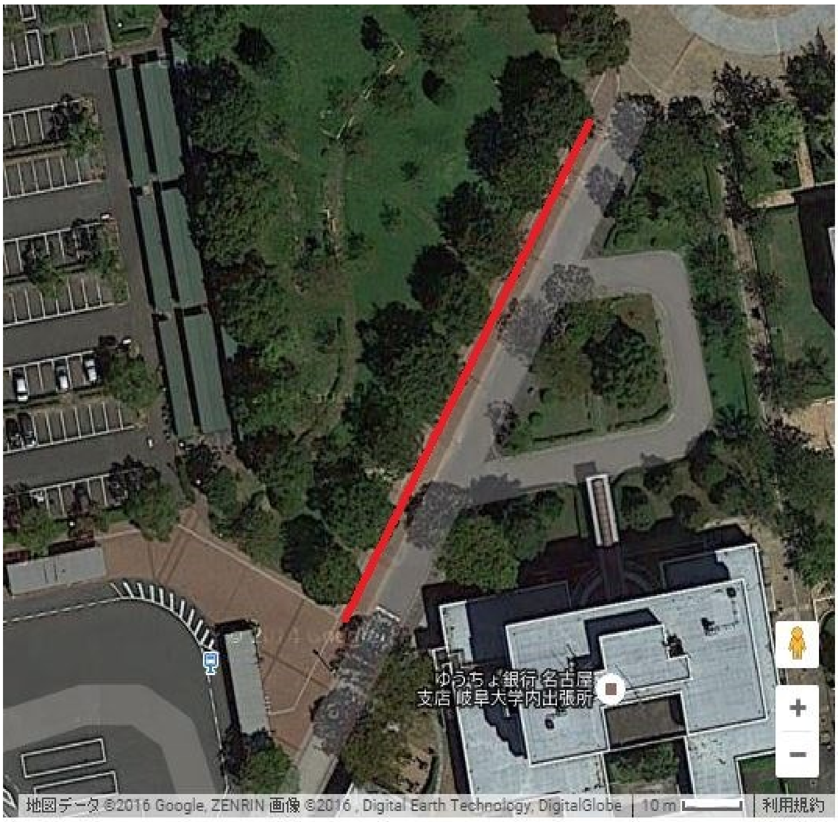


Figure 7: Experimental site.

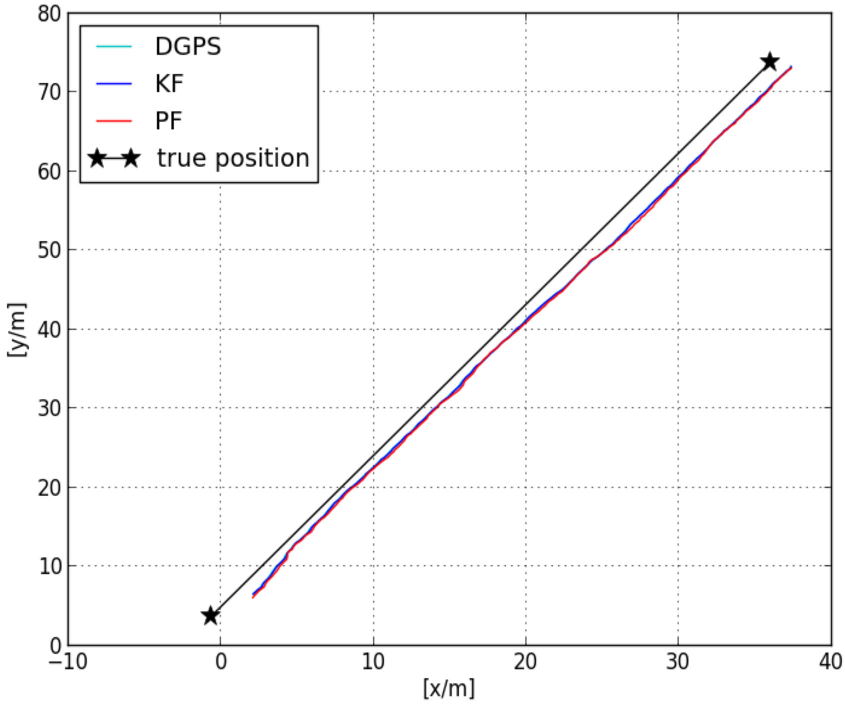


Figure 8 The estimation results of plane when altitude position when receiving more than 10 artificial satellites.

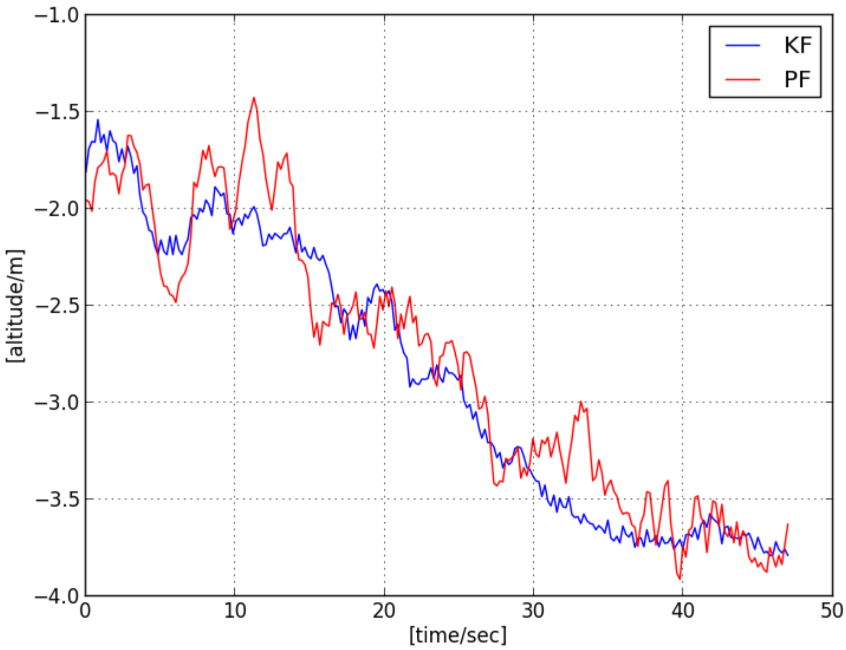


Figure 9 The estimation results of plane when altitude position when receiving more than 10 artificial satellites.

Table 4 The filtering results when receiving more than 10 artificial satellites.

| | |
|---|---|
| KF error average [m] | 5.365 |
| PF error average [m] | 5.720 |
| KF maximum error [m] | 11.372 |
| PF maximum error [m] | 11.524 |
| KF error average of x coordination [m] | 1.077 |
| PF error average of x coordination [m] | 1.072 |
| KF error average of y coordination [m] | 5.099 |
| PF error average of y coordination [m] | 5.519 |
| KF altitude error average [m] | 2.914 |
| PF altitude error average [m] | 2.827 |

## CONCLUSION

This paper presents a method for smoothing GPS data from a UAV using Extended Kalman filtering and particle filtering for navigation or position control. The Extended Kalman Filter is a widely used estimation method in autonomous navigation systems. However, it is characterized by cumulative errors due to performing an approximate linearization of the system's dynamics. The Particle Filter makes no assumptions on the forms of the state vector and measurement probability densities. In the particle filter a set of weighted particles (state vector estimates evolving in parallel) is used to approximate the posterior distribution of the state vector. To succeed the convergence of the algorithm, at each iteration resampling takes place through which particles with low weights are substituted by particles of high weights. Experiments were used to evaluate the accuracy of estimated GPS data which are based on estimation of the UAV's state vector with Extended Kalman Filter and Particle Filter methods. It was shown that the KF is a reliable and computationally efficient approach to state estimation-based control, while Particle Filtering is well-suited to accommodate non-Gaussian measurements. PF provide reliable solutions to nonlinear estimation and control problems, with the Particle Filter to require less a-priori knowledge about the statistical characteristics of the measurements and of the system's state variables. This experiments show if the number of satellites is small such as less than 7, Particle Filter shows small observation error. However, if the number of satellites is larger than 10, there is not so much difference between with EKF and with PF.

**REFERENCES**

Doucet, A., Godsill, S., Andrieu, C. (2001). Sequential Monte Carloin practice. New York: Springer-Verlag.

Doucet, A., de Freitas, N., Gordon, N. S. (2000). On sequential Monte Carlosampling methods for Bayesian filtering. *Statistics and Computing*, 10: 197−208.

Kamen, E.W., Su. J. K., (1999). *Introduction to Optimal Estimation.* Springer.

Lewis, L. Xie, L., & Popa, D. (2007). *Optimal and Robust Estimation with an Introduction to Stochastic Control Theory.* CRC Press.

Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE, Aerospace and Electronic Systems Magazine*, 25(7): 53−82.

Rigatos, G., & Zhang, Q. (2001). *Fuzzy Model Validation Using the Local Statistical Approach.* Rennes, France: Publication Interne IRISA No 1417.

Gordon, N., Salmond, J.D., & Smith A. F., (1993). *Novel approach to nonlinear/non-Gaussian Bayesian state estimation.* 140(2): 107−113.: IEEE Proceedings F, Radar and Signal Processing.

Bassevile, M., & Nikiforov, I. (1993). *Detection of Abrupt Changes: Theory and Applications.* Prentice-Hall.

Rigatos, G. G., (2011). Modelling and Control for Intelligent Industrial Systems. In *Adaptive Algorithms in Robotics and Industrial Engineering.* Springer.

Xu, G. (2003). GPS Theory, Algorithms and Applications. In S. Elliott, *Signal Processing for Active Control.* Academic Press, London: Springer.