A Fault Detection and Exclusion Algorithm using Particle Filters for non-Gaussian GNSS Measurement Noise

*Youngsun Yun[1], Doyoon Kim[2], Changdon Kee[3]

Mechanical and Aerospace Engineering, Seoul Nat'l Univ.
[1] (E-mail: zoro@snu.ac.kr), [2] (E-mail: einheit@snu.ac.kr), [3] (E-mail: kee@snu.ac.kr)

Abstract

Safety-critical navigation systems have to provide 'reliable' position solutions, i.e., they must detect and exclude measurement or system faults and estimate the uncertainty of the solution. To obtain more accurate and reliable navigation systems, various filtering methods have been employed to reduce measurement noise level, or integrate sensors, such as global navigation satellite system/inertial navigation system (GNSS/INS) integration. Recently, particle filters have attracted attention, because they can deal with nonlinear/non-Gaussian systems. In most GNSS applications, the GNSS measurement noise is assumed to follow a Gaussian distribution, but this is not true. Therefore, we have proposed a fault detection and exclusion method using particle filters assuming non-Gaussian measurement noise. The performance of our method was contrasted with that of conventional Kalman filter methods with an assumed Gaussian noise. Since the Kalman filters presume that measurement noise follows a Gaussian distribution, they used an overbounded standard deviation to represent the measurement noise distribution, and since the overbound standard deviations were too conservative compared to the actual distributions, this degraded the integrity-monitoring performance of the filters. A simulation was performed to show the improvement in performance of our proposed particle filter method by not using the sigma overbounding. The results show that our method could detect smaller measurement biases and reduced the protection level by 30% versus the Kalman filter method based on an overbound sigma, which motivates us to use an actual noise model instead of the overbounding or improve the overbounding methods.

Keywords: FDE, Integrity, Particle Filter, Sigma Overbounding.

1. Introduction

For safety-critical applications of global navigation satellite systems (GNSSs), such as aircraft and missile navigation systems, it is important to be able to detect and exclude faults that could cause risks to the accuracy and integrity, so that the navigation system can operate continuously without any degradation in performance. For high-accuracy systems, the fault detection and exclusion (FDE) function needs to be able to detect and exclude smaller biases. More independent GNSSs (e.g., the GALILEO and Global Navigation Satellite System (GLONASS) satellites) are becoming more widely available, and therefore we need to take into account simultaneous faults of multiple satellites. It is difficult to detect small errors and simultaneous multiple faults using conventional snapshot receiver autonomous integrity monitoring (RAIM) algorithms, and therefore various filtering methods have been studied for reducing the measurement noise level or integrating a GNSS with other sensors so that the navigation system can estimate its position more accurately and reliably. Because filters reduce the noise level of measurements using previous information, they can provide a better FDE performance than snapshot algorithms can. However, because most filters, such as Kalman filters, presume that the measurement noise and disturbance follow a Gaussian distribution, their performance can degrade if this assumption is not correct. In addition, when the system is nonlinear, a linearized or extended Kalman filter is used, and in this case the nonlinearity affects the estimation and FDE performance. Because GNSS measurement noise does not follow a Gaussian distribution perfectly, the Kalman filter approach has to use an inaccurate noise model that may cause performance degradation.

To address these problems, we have proposed a new fault detection and exclusion algorithm using particle filters. Particle filters have been researched over the last few years [1][2] as an alternative for solving nonlinear/non-Gaussian problems. Particle filters are based on the Monte Carlo approach, which monitors sample points, referred to as 'particles', and describes the posterior distribution of the states. The proposed algorithm estimates a distribution of a measurement residual from the posterior density and detects exceedingly large residuals to satisfy a false alarm rate. This algorithm can detect faults based on an accurate estimation of the posterior distribution, and it can detect and exclude faults almost simultaneously, so that the system can exclude a fault measurement easily. With a non-Gaussian GNSS measurement noise, a particle filter can estimate the distribution of the state more accurately than a Kalman filter can, and therefore it has a better FDE performance. In addition, if the system is also highly nonlinear, then the performance will also be better. Particle filters for nonlinear systems have undergone much research. However, the advantage of these filters with a non-Gaussian noise distribution has not been investigated. Therefore, our work focused on the effect of a non-Gaussian noise distribution on the FDE performance, which will be assessed by the simulations.

Section 2 introduces the non-Gaussian GNSS measurement model employed for our simulations, and a sigma overbounding method applied to the Kalman filter approach. Section 3 briefly discusses the employed particle filter algorithm. Section 4 discusses our proposed fault detection and exclusion algorithms based on the above particle filter, and an associated protection level calculation method. The results of the simulations are summarized and discussed in Section 5, and finally Section 6 concludes the paper with a summary and suggestions for future work.

2. Non-Gaussian GNSS Measurement Noise

This section describes a true noise model and the overbounding method used in the simulations, which are based on the work of Shively and Braff [3].

2.1 GNSS Pseudorange Noise Model

In most GNSS applications, including the Space-Based Augmentation System (SBAS) and the Ground-Based Augmentation System (GBAS), the pseudorange measurement noise is assumed to follow a Gaussian distribution. However, this is not true. In general, the core part of the noise distribution can be characterized well using a Gaussian distribution, but the tail part of the distribution is heavier than in a Gaussian distribution. The heavier tail is due to ground-reflected multipaths or to systematic receiver/antenna errors [4]. Our simulation used a Gaussian core-Laplacian (GL) tail probability density function (PDF) for the receiver noise distribution for a heavy-tailed true noise model, which will be applied to our particle filter method. Shively and Braff [3], used the GL model to represent the actual Local Area Augmentation System (LAAS) pseudorange noise data collected at the ground facility.

$$p(n) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{n^2}{2\sigma^2}\right), & |n| \le n_{tr} \\ p_{tr} \times \exp\left(-\frac{p_{tr}}{c_{tr}}|n - n_{tr}|\right), & |n| > n_{tr} \end{cases} \quad (1)$$

where n is the pseudorange noise, σ is the standard deviation of the Gaussian core density, and $n_{tr} = 2.58\sigma$ is the transition point from the Gaussian to the Laplacian distribution. The other constants are defined as follows

$$p_{tr} = p_{Gaussain}(n_{tr}) \times PUF, \quad c_{tr} = \int_{n_{tr}}^{\infty} p_{Gaussain}(n)dn \times TUF \quad (2)$$

The probability density uncertainty factor (PUF) and the tail uncertainty factor (TUF) provide confidence in consideration of the volume of data. The simulation sets PUF = 0.52 and TUF = 1 used in our model follow the convoluted three-receiver noise model shown in the Shively and Braff's paper. The narrower curve shown in Figure 1 represents the PDF.

2.2 Sigma Overbounding

Since measurement noise does not follow a Gaussian distribution, sigma overbounding methods are used to simplify implementation and to prevent integrity risks [3][5][6]. Overbounding is a procedure used to find the inflation factors (INF), which are the ratio of the overbounded and the observed sigmas

$$\sigma_{over} = INF \cdot \sigma \cdot \quad (3)$$

The inflation factor is determined using Equation (4)

$$INF = \frac{M}{5.81}, \quad \int_{M}^{\infty} p_{true}(x)dx = \int_{5.81}^{\infty} p_{N(0,1)}(x)dx = 3.12 \times 10^{-9} \quad (4)$$

where $p_{true}(x)$ is the observed true PDF, which was taken to represent the GL distribution in our work, and $P_{N(0,1)}$ is the Gaussian PDF, which has a zero mean and a standard deviation of unity. The denominator value of 5.81 is a multiplier used for LAAS users to calculate a defined fault-free protection level [7]. This method provides users with an overbound broadcast sigma that they can use to estimate the fault-free protection levels to bound position errors with the required probability. Figure 2 shows a pictorial representation of this method, which shows that the one-sided tail probabilities of the Gaussian and the GL distributions have the same observed standard deviation. Using the above method, the value of the INF was determined to be INF = 2.07. Figure 1 shows the PDF values of the true noise model (GL) and the overbounded Gaussian (OG) model. The OG model had a broader distribution, so that the tail was conservative enough to bound the heavy-tailed GL model.

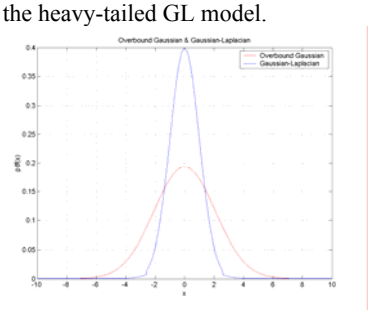

Figure 1. The Gaussian–Laplacian model and the Gaussian overbound model.

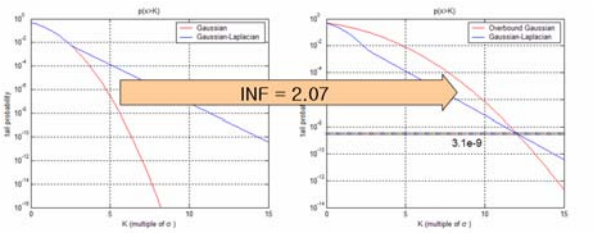

Figure 2. The set inflation factor.

3. PARTICLE FILTERS

Particle filters can deal with nonlinear/non-Gaussian dynamic systems as well as linear/Gaussian systems. They use a sequential Monte Carlo approach, and are based on a sequential importance sampling (SIS) procedure. We have used the sampling importance resampling (SIR) algorithm for the simulations discussed in this paper. This section briefly introduces the basic SIR algorithm to provide a background to our proposed FDE algorithm. A more detailed derivation of this filter is described in the literature [8].

3.1 Sequential Importance Sampling (SIS)

First, we will define the state–space equations for a dynamic system as follows

$$\mathbf{x}_k = \mathrm{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \ \mathbf{z}_k = \mathrm{h}_k(\mathbf{x}_k, \mathbf{n}_k) \quad (5)$$

where $\mathbf{x}_k$ is a state vector at the k-th epoch, $\mathbf{x}_{k-1}$ is a state vector of the previous epoch, $\mathbf{v}_{k-1}$ is a process noise vector at the previous epoch, f_k is a time propagation function of the state and the process noise, $\mathbf{z}_k$ is a measurement vector at the k-th epoch, $\mathbf{n}_k$ is a measurement noise vector at the k-th epoch, and h_k is a measurement function of the state and the measurement noise.

In the above system, the functions 'f_k' and 'h_k' need not be linear, and the noise vectors, '$\mathbf{v}_{k-1}$' and '$\mathbf{n}_k$', can follow an arbitrary probability distribution, not necessarily a Gaussian distribution, whose probability density function must be known to implement the filter. The expectation and the variance of the states are obtained from the posterior probability density distribution (PDF), which is expressed using a mass function of random samples and associated weights, as shown in Equation (6). The random samples are known as 'particles'

$$\hat{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \sum_{i=1}^{N_S} \tilde{\omega}_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \quad (6)$$

where $\mathbf{z}_{1:k} = \{\mathbf{z}_i, i = 1,...,k\}$, $\tilde{\omega}_k^{(i)}$ is the normalized weight of the (i)-th particle at epoch k, $\delta(\cdot)$ is the Dirac delta function, $\mathbf{x}_k^{(i)}$ is the (i)-the particle at epoch k, and N_S is the number of particles.

The particles are derived from the prior distribution, and the associated weights are calculated sequentially using Equation (8)

$$\mathbf{x}_k^{(i)} = \mathrm{f}(\mathbf{x}_{k-1}^{(i)}, \mathbf{v}_{k-1}), \quad \mathbf{x}_{k-1}^{(i)} \sim \hat{p}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \quad (7)$$

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}. \quad (8)$$

The conditional probabilities, $p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$ and $p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$ which are known as the 'transition prior density' and 'likelihood density', respectively, are determined by the system definition. The importance density, $q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$, which is not defined by the problem, is a design parameter chosen by the designer. After determining the importance density, we can calculate the current weight of the i-th particle and the sum of the weights of all the particles is then normalized to unity

$$\tilde{\omega}_k^{(i)} = \omega_k^{(i)} / \sum_{j=1}^{N_S} \omega_k^{(j)}. \quad (9)$$

We now need to choose the importance density to complete the filter. Theoretically, the optimum importance density is given by

$$q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)_{optimal} = p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) \quad (10)$$

as shown by Doucet et al. [9].

However, because it is not possible to know the exact PDF of Equation (10), various choices have been proposed [10][11]. The above procedure is known as sequential importance sampling (SIS). A popular choice of the importance density in the SIS algorithms is the transition prior density

$$q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) = p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}). \quad (11)$$

This method is very simple and easy to implement. However, it still has the degeneracy problem mentioned above [2]. The degeneracy problem can be mitigated by using a more refined importance density and a 'resampling (or selection)' procedure.

3.2 Resampling

The SIR filter resamples the particles when the degeneracy is too severe. A proper measure of the degeneracy is the effective sample size, N_{eff}, which is calculated from the normalized weights using equation (12). The value of $\hat{N}_{eff}$ is always smaller than the number of particles: the smaller the value of $\hat{N}_{eff}$ is, the more severe the degeneracy is. The SIR filter performs resampling when the value of $\hat{N}_{eff}$ is smaller than a given threshold. The 'resampling' is a particle rearrangement procedure to discard any particles that have low weights and to multiply particles that have large weights to have the same weights as in Equation (13).

$$\hat{N}_{eff} = 1/\sum_{i=1}^{N_S} (\tilde{\omega}_k^{(i)})^2 \le N_S. \quad (12)$$

$$\{x_{1:k}^{(i)}, \tilde{\omega}_k^{(i)}\} \rightarrow \{x_{1:k}^{(j)}, N_S^{-1}\}. \quad (13)$$

4. Integrity Monitoring

GNSS-based navigation systems must have an integrity monitoring system that contains two main functions: (1) the detection and exclusion of satellite faults and (2) an estimation of the uncertainty of the position solutions. First, the system calculates decision variables using a specific algorithm, and then it compares these to thresholds that have been set to satisfy the integrity requirements for the desired operation. If any decision variable exceeds the threshold, then the system concludes that the corresponding measurement has a fault, and excludes the detected fault. If such an exclusion is not possible, then the navigation system does not trust the current position solution and uses other navigation solutions. Second, because the system cannot know the true position estimation errors, it estimates the uncertainty of the solution, which is known as the protection level (PL). The system compares the value of the PL to a predefined alert limit (AL) for an expected operation to decide whether or not the navigation system will use the solution. We implemented four monitoring algorithms in this work: the snapshot χ^2 method (SSX2), the Kalman filter χ^2 method (KFX2), the Kalman filter individual residual (KFIR) method, and the particle filter individual residual (PFIR) method. Our proposed algorithms for the two integrity monitor functions based on particle filters, which are based on the PFIR method, are described in the following Subsections 4.1 and 4.2, and the other methods will be treated briefly.

4.1 Fault Detection and Exclusion Algorithm

There are several algorithms that can be used to detect and exclude faults using filtering methods. Our proposed algorithm is a residual method that uses the posterior density of a state, which is estimated from the particle filters and the measurement noise distribution. Residual is defined as the difference between an expected measurement and an actual measurement, which will be described in detail in the following subsections. Residual methods using the snapshot algorithm (SSX2) and the Kalman filter (KFX2) calculate the decision variables using the squared sum of the residuals, which is known as the weighted squared sum of the error (WSSE), and this follows a χ^2 distribution whose degrees of freedom are determined by the number of measurements as in Equation (14)

$$WSSE = \frac{\mathbf{r}^T \mathbf{r}}{\mathbf{C}_r} \sim \begin{cases} \chi^2(0, N_M - 4), & \text{SSX2} \\ \chi^2(0, N_M), & \text{KFX2} \end{cases} \quad (14)$$

where $\mathbf{r}$ is the residual vector, $\mathbf{C}_r$ is the covariance matrix or the residual vector, and N_M is the number of measurements.

This type of algorithm will not be described here, but can be found in literature [12][13]. The χ^2 test has low detection sensitivity to faults affecting only a small number of the components of the residuals, and this drawback can be avoided by testing the components individually. Individual tests can detect multiple faults simultaneously and also any system dynamics failures or unexpected severe disturbances. The Kalman filter or the particle filter approach can examine the residuals individually, not the combined residual, because it has previously estimated position. In the Kalman filter approach, KFIR, the algorithm compares a measurement residual to the multiple of the residual standard deviation. Since our proposed method, PFIR, assumes a non-Gaussian noise, it cannot use the residual variance. Instead, it estimates the residual distribution, and then decides whether the residual is located in a 'normal' region or not. These individual residual algorithms have to estimate the residual distribution and determine the individual detection threshold for each residual.

(1) Calculation of the residuals and the distribution

The expected measurement was calculated from the

propagation of the previous epoch's state vector as

$$\tilde{\mathbf{x}}_k = E[\mathrm{f}_k(\hat{\mathbf{x}}_{k-1}, \mathbf{v}_{k-1})] = \mathrm{f}_k(\hat{\mathbf{x}}_{k-1}, \mathbf{0}) . \quad (15)$$

The expected measurement is a projection of the propagated state from the position domain to the measurement domain. Under normal conditions, the expectation distributions lie close to each other, but when there is a measurement fault, they are separated by a distance. In our work the distance between the two distributions is referred to as the residual

$$\mathbf{r}_k = \mathrm{h}_k(\tilde{\mathbf{x}}_k, 0) - \mathbf{z}_k . \quad (16)$$

The residual distribution for a measurement is estimated as follows

$$\mathrm{p}_{R_k^j}(r_k^j) = \int_{-\infty}^{\infty} \mathrm{p}_{\tilde{\mathbf{x}}_k}(\tilde{\mathbf{x}}_k') \mathrm{p}_Z(z_k^j) d\tilde{\mathbf{x}}_k' \quad (17)$$
$$\approx \sum_{i=1}^{N_S} \hat{\mathrm{p}}_{\hat{\mathbf{x}}_{k-1}}(\hat{\mathbf{x}}_{k-1}^{(i)}) \mathrm{p}_N(\mathrm{h}_k^j(\tilde{\mathbf{x}}_k^{(i)}) - z_k^j - r_k^j)$$

where $\hat{\mathrm{p}}_{\hat{X}_{k-1}}(\cdot)$ is estimated by the particle filters using Equation (6), and $\mathrm{p}_N(\cdot)$ is the noise distribution expressed by Equation (1).

A sample of a residual distribution under nominal conditions is shown in Figure 3. The expectation is near zero, and the large residuals have low probabilities. When the residual is in the region between the two thresholds, known as the 'normal region', the system declares that there is no fault in the measurement. Otherwise, it determines the measurement as having a fault, and excludes it for continuous operation.

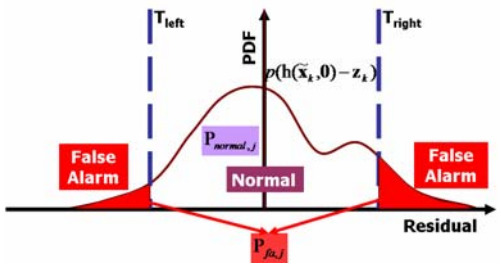

Figure 3. PDF of a residual and thresholds.

In real implementation, the expectation of the PDF is located at the calculated residual and not zero. Therefore we calculate a one-sided tail probability below or above zero as a decision variable using Equation (18) and compare it to the individual false alarm rate that will be defined in the next subsection for each measurement

$$pr_k^j = \begin{cases} \int_{-\infty}^{0} \mathrm{p}_{R_k^j}(r_k^{j\prime}) dr_k^{j\prime}, & r_k^j \geq 0 \\ \int_{0}^{\infty} \mathrm{p}_{R_k^j}(r_k^{j\prime}) dr_k^{j\prime}, & r_k^j < 0 \end{cases} . \quad (18)$$

Since the Kalman filter individual residual method assumes a Gaussian noise, the PDFs are all Gaussian, and the residual distribution can be expressed simply by a standard deviation so that the thresholds are determined as the multiple of the standard deviations.

(2) Set thresholds

The threshold is set to satisfy a specified false alarm rate when a χ^2 method based on the Gaussian distribution, SSX2 or KFX2, is employed

$$\int_{T_{\chi^2}}^{\infty} \mathrm{p}_{\chi^2}(x) dx = P_{FA} . \quad (19)$$

However, the individual residual test requires a threshold for each measurement, and therefore an individual false alarm rate, P_{fa}, needs to be determined first. Because all the measurements are independent, the individual false alarm rate is approximated by a value of $1/N_M$ of the total false alarm rate as follows.

$$P_{fa} = 1 - (1 - P_{FA})^{\frac{1}{N_M}} \approx 1 - (1 - \frac{P_{FA}}{N_M}) = \frac{P_{FA}}{N_M} . \quad (20)$$

where N_M is the number of measurements. Because we assumed the measurement noise followed a symmetrical distribution in Section 2, the particle filter algorithm declares a fault present if one-sided tail probability of a residual pr_k^j exceeded a value of $\frac{P_{FA}}{2N_M}$.

The threshold for the Kalman filter approach is determined by the following equation.

$$\int_{T_{right}}^{\infty} \mathrm{p}_R(r) dr + \int_{-\infty}^{T_{left}} \mathrm{p}_R(r) dr = P_{fa,j} = P_{fa} . \quad (21)$$

The value of T_{right} is equal to T_{left} because of the symmetry and they are determined as the multiple of sigma from the Gaussian cumulative density function and the value of P_{fa}.

4.2 Protection Level

LAAS users must calculate both fault-free and faulted protection levels (XPL_0 and XPL_1, respectively). The faulted protection levels indicate the position error protection assuming that a fault measurement exists. LAAS assumes that a fault is induced from the reference receiver errors. However, we assumed that the fault includes all possible errors contained in the measurements that are used for position estimation. The formulas used to calculate the protection levels based on a snapshot method assuming a Gaussian noise were derived as follows. The Kalman filter and the particle filter methods can be used to obtain the protection levels using a similar procedure, which will be discussed later. In GNSSs, the measurement equation can be expressed using a linear function

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (22)$$

where $\mathbf{z}$ is a measurement vector, $\mathbf{H}$ is an observation matrix, $\mathbf{x}$ is a state vector, and $\mathbf{n}$ is a noise vector. The state vector is estimated using the weighted least squares method and the estimation error can be obtained from Equation (23)

$$\delta\hat{\mathbf{x}} = \mathbf{H}_W^+ \delta\mathbf{z}, \quad \mathbf{H}_W^+ = (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{W}\mathbf{H}^{\mathrm{T}} \quad (23)$$

$$\mathbf{W} = (diag\left[\sigma_{n_1}^2 \quad \cdots \quad \sigma_{n_j}^2 \quad \cdots \quad \sigma_{n_{N_M}}^2\right])^{-1} \quad (24)$$

where $\delta\hat{\mathbf{x}}$ is the estimation error, $\delta\mathbf{z}$ is the measurement error, and $\mathbf{W}$ is the weighting matrix.

When a fault, T, occurs in a measurement, j, then the estimation error is expressed by a deterministic and a statistic part, as follows

$$\delta\hat{\mathbf{x}}^j = \begin{bmatrix} \delta\hat{x}_e^j \\ \delta\hat{x}_n^j \\ \delta\hat{x}_u^j \\ \delta\hat{x}_t^j \end{bmatrix} = \mathbf{H}_W^+ \left(\begin{bmatrix} 0 \\ \vdots \\ T_j \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} n_1 \\ \vdots \\ n_j \\ \vdots \\ n_{N_M} \end{bmatrix} \right) = \underbrace{\begin{bmatrix} \mathbf{H}_W^+(1,j) \\ \mathbf{H}_W^+(2,j) \\ \mathbf{H}_W^+(3,j) \\ \mathbf{H}_W^+(4,j) \end{bmatrix} \cdot T_j}_{\text{Deterministic}} + \underbrace{\mathbf{H}_W^+ \begin{bmatrix} n_1 \\ \vdots \\ n_j \\ \vdots \\ n_{N_M} \end{bmatrix}}_{\text{Statistic}} . \quad (25)$$

The expectation of the estimation error is a function of the fault, T

$$\mathrm{E}[\delta\hat{\mathbf{x}}^j] = \left[\mathbf{H}_W^+(1,j) \quad \mathbf{H}_W^+(2,j) \quad \mathbf{H}_W^+(3,j) \quad \mathbf{H}_W^+(4,j)\right]^T \cdot T_j \tag{26}$$

and the covariance is determined by the standard deviations of the measurement noise

$$\mathrm{COV}[\delta\hat{\mathbf{x}}^j] = \mathbf{H}_W^+ \mathbf{W}^{-1} (\mathbf{H}_W^+)^T = (\mathbf{H}^{\mathbf{T}}\mathbf{W}\mathbf{H})^{-1} \cdot \tag{27}$$

Finally, the protection levels are calculated using the following equations [7]. The term XPL_1^j denotes the protection level assuming that the j-th measurement has a fault. The letters 'H' and 'V' substituted for 'X' denote 'horizontal' and 'vertical', respectively.

$$\mathrm{HPL}_0 = k_{ffmd}\sigma_h, \quad \mathrm{HPL}_1^j = k_{md}\sigma_h + \left|T_j\sqrt{(\mathbf{H}_{1,j}^+)^2 + (\mathbf{H}_{2,j}^+)^2}\right| \tag{28}$$

$$\mathrm{VPL}_0 = k_{ffmd}\sigma_v, \quad \mathrm{VPL}_1^j = k_{md}\sigma_v + \left|T_j \cdot \mathbf{H}_{3,j}^+\right| \tag{29}$$

where $\sigma_h^2 = (\mathbf{H}^{\mathbf{T}}\mathbf{W}\mathbf{H})_{1,1}^{-1} + (\mathbf{H}^{\mathbf{T}}\mathbf{W}\mathbf{H})_{2,2}^{-1}$, $\sigma_v^2 = (\mathbf{H}^{\mathbf{T}}\mathbf{W}\mathbf{H})_{3,3}^{-1}$, $k_{ffmd} = 5.810$ and $k_{md} = 2.898$.

The standard deviation multipliers, k_{ffmd} and k_{md}, are defined in 'LAAS MASPS' [7] for three 'Performance Type 1' reference receivers. The value of the fault, T, is equal to the threshold for each FDE algorithm because it is the maximum fault that the system cannot detect. The filtering methods can estimate the protection levels with the above procedure, however, the values of the $k\sigma$ and T parts are determined differently for each algorithm. The Kalman filter method can obtain the standard deviations, σ_h and σ_v, from the covariance matrix, P, and T_j is calculated from the residual standard deviation and the individual false alarm rate. The particle filter methods can estimate them from the estimated posterior density and the residual density, respectively.

5. Simulation

A simulation was performed to assess our proposed FDE algorithm based on particle filters. To investigate the advantage of the non-Gaussian assumption, conventional methods using the Gaussian assumption were also implemented in parallel.

5.1 Simulation Setup

The propagation model and the measurement equation were defined using linear functions, as defined by the equation listed below, because we only focused on the effect of non-Gaussian noise and not on any nonlinearity. The states used were a user position in East-North-Up (ENU) coordinates and a receiver clock bias. The user was assumed to be static and 24 GPS satellite constellation was generated for each epoch for 100 epochs. The pseudorange measurements were generated from the constellation and the user's position at a frequency of 1 Hz.

$$\mathbf{x}_k = \mathrm{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{30}$$

$$\mathbf{z}_k = \mathrm{h}_k(\mathbf{x}_k, \mathbf{n}_k) = \mathbf{H}_k\mathbf{x}_k + \mathbf{n}_k \tag{31}$$

where $\mathbf{x}_k = \begin{bmatrix} x_e & x_n & x_u & x_t \end{bmatrix}^T$, $\mathbf{F} = \mathbf{I}(4\times 4)$, $\mathbf{z}_k = \left(e_u^1 \cdot R^1 - \rho_u^1, \quad e_u^2 \cdot R^2 - \rho_u^2, \quad \cdots, \quad e_u^{n_M} \cdot R^{n_M} - \rho_u^{n_M}\right)_k^T$, $\mathbf{H}_k = \begin{pmatrix} e_u^1 & e_u^2 & \cdots & e_u^{n_M} \\ -1 & -1 & \cdots & -1 \end{pmatrix}_k^T$, I is the identity matrix, e_u^i is the line of sight vector from the user to the i-th satellite, R^i is the i-th satellite's position, and ρ_u^i is the pseudorange measurement of the i-th satellite at the user. The measurement noise is derived from the assumed distribution shown in Equation (1).

The overbounded standard deviation for each pseudorange was determined using the wide area augmentation system (WAAS) model, which was a function of the elevation angle of each satellite [13]

$$\sigma_{over,i}^2 = \sigma_{UDRE,i}^2 + F^2(El_i)\sigma_{UIVE,i}^2 + \sigma_{SNR,i}^2 + \frac{\sigma_{m45}^2}{\tan^2 El_i} + \frac{\sigma_{trv}^2}{\sin^2 El_i} \tag{32}$$

where $F(\cdot)$ is the obliquity factor, El_i is the i-th satellite elevation angle, $\sigma_{\mathrm{UDRE,i}} = 0.5\mathrm{m}$, $\sigma_{\mathrm{UIVE,i}} = 0.5\mathrm{m}$, $\sigma_{\mathrm{SNR,i}} = 0.22\mathrm{m}$, $\sigma_{\mathrm{m45}} = 0.22\mathrm{m}$, $\sigma_{\mathrm{trv}} = 0.15\mathrm{m}$.

The simulator injects biases into the pseudorange of a satellite as follows

$$b_k = \begin{cases} 8\mathrm{m}, & \mathrm{k} = 20 \sim 40\,\mathrm{sec} \\ 7\mathrm{m}, & \mathrm{k} = 60 \sim 80\,\mathrm{sec} \\ 0\mathrm{m}, & \text{otherwise} \end{cases}. \tag{33}$$

Four FDE algorithms described in Section 4 were used to detect and exclude the fault and to estimate the protection levels. Subsection 5.2 discusses the results from the position filters and the integrity monitor algorithms and Subsection 5.3 discusses the position estimation and the protection levels that provided the final navigation information for the user.

5.2 Measurement Domain Result

Figure 4 shows the decision variables (WSSE) and the thresholds of the SSX2 and the KFX2 algorithms. When a decision variable exceeded the threshold, the system declared a fault. The SSX2 method could not detect any faults due to the high noise level. The KFX2 method detected the 8 m bias initially, but it failed to detect thereafter. When a filtering method fails to detect and exclude a bias, the remaining bias propagates through the filter so that it becomes more difficult to detect afterwards.

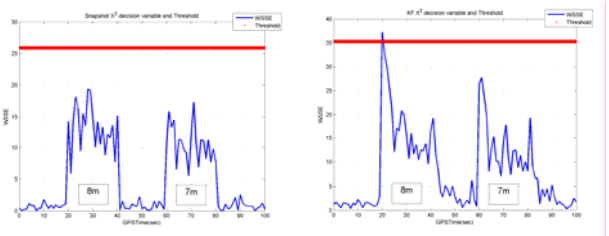

Figure 4. WSSE and threshold of SSX2 and KFX2.

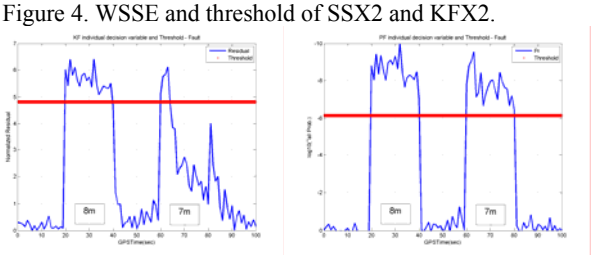

Figure 5. Residual and threshold of KFIR and PFIR (fault measurement).

The individual residual methods (KFIR and PFIR) had as many decision variables as the number of measurements at an epoch. The left-hand side of Figure 5 shows the normalized residual of the fault measurement and the threshold value for the KFIR method, and the right-hand side of Figure 5 depicts them of PFIR method. The KFIR method detected an 8 m bias from 20 to 39 s, and failed to detect a fault at an epoch of 40 s. It did not detect a 7 m bias after 64 s. The undetected bias remained in the filtering process, and led to a large position error that propagated into subsequent process, which led the monitoring system not to detect the fault after the missed detection and the other nominal residuals to be noisy. The PFIR method detected the faults at all epochs at which the system has a fault measurement.

5.3 Position Domain Result

Figure 6 shows the vertical position errors (left-hand side) and the vertical protection levels (right-hand side) for each algorithm after the FDE procedure. Since the SSX2 method (green) and the KFX2 method (black) did not detect faults well, they exhibited large position errors induced by the remaining faults. The KFIR method (red) excluded most of the 8 m bias and provided more accurate solutions from 20 to 39 s. However, the missed detection of the 8 m bias caused a large error after the epoch of 60 s. The PFIR method (blue) detected and excluded all the biases and provided the most accurate position solutions. Table 1 shows the 95% level errors for each method. As for the protection levels, the HPL_0 and VPL_0 values are not shown here, because they were always smaller than the HPL_1 and VPL_1 values, respectively, in our simulation. The data show that the SSX2 method provided the highest protection level, and the other filtering methods had lower protection level. The KFX2 method provided a slightly higher protection level than the KFIR method, which means that the KFIR method could detect smaller biases. The PFIR protection level was the lowest and this shows that the algorithm outperformed the other algorithms in terms of its FDE ability. Table 2 shows the mean of the protection levels under a fault-free condition. The data in Table 2 show that the PFIR method reduces the PLs by about 30% of the values obtained using the KFIR method. Since the data in Figure 6 show the PL values after exclusion of the detected fault measurement, they have higher values in the fault section, because of the reduced number of visible satellites. The results imply that the PFIR method can provide higher availability than the other methods can.

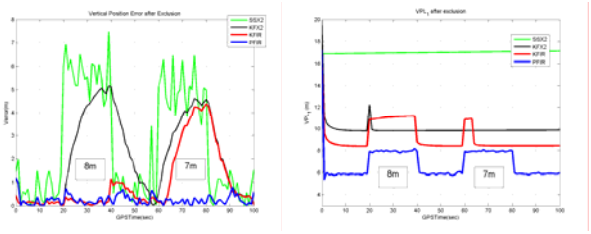

Figure 6. The vertical position errors and protection levels after exclusion.

Table 1. The 95% level position errors for each algorithm.

	SSX2	KFX2	KFIR	PFIR
Horizontal Error (m, 95%)	3.93	3.61	3.17	0.66
Vertical Error (m, 95%)	6.29	4.83	4.19	0.67

Table 2. Protection levels under fault-free conditions.

	SSX2	KFX2	KFIR	PFIR
HPL_1 (m) ($HPL_1/HPL_{1,KFIR}$, %)	9.23 (163.36)	6.46 (114.34)	5.65 (100)	3.81 (67.43)
VPL_1 (m) ($VPL_1/VPL_{1,KFIR}$, %)	22.21 (263.46)	9.87 (117.08)	8.43 (100)	5.92 (70.23)

6. Conclusion

In this work, we have developed a fault detection and exclusion algorithm that tests measurement residuals individually using particle filters. We have described our algorithm and the non-Gaussian heavy-tailed measurement model used in the simulations. The simulation results show that our proposed algorithm detected smaller measurement biases and generated smaller protection levels by about 30%, which implies that systems with the FDE algorithm can provide better integrity monitoring performance than other methods, which assume an overbounded Gaussian measurement. The data show that the conservative sigma overbounding methods degrade the integrity monitoring performance. Inversely, using a true noise model or more accurate overbounding methods, we can provide a more available navigation system without losing any integrity or continuity. Although our proposed algorithm requires a high computational capacity due to the large particle set used, it can be used in more sophisticated systems or for development and performance assessment of sigma overbounding methods in conventional systems.

In the future, more computationally efficient algorithms need to be developed for wide usage of our method. In addition, because our algorithm tests the residuals individually, it can be used to detect simultaneous multiple faults, which is another interesting topic for future research.

Acknowledgement

This research was supported in part by the Institute of Advanced Machinery and Design at Seoul National University, BK-21, and the Institute of Advanced Aerospace Technology.

Reference

1. N. J. Gordon, D. J. Salmond and A. F. M Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, Vo. 140, 1993, pp. 107–113.
2. A. Doucet, J. F. G. de Freitas and N. J. Gordon, *Sequential Monte Carlo methods in practice*, Springer-Verlag, 2001.
3. C. A. Shively and R. Braff, *An overbound concept for pseudorange error from the LAAS ground facility, (MP00W0000138)*, The MITRE Corporation, 2000.
4. I. Sayim, B. Pervan, S. Pullen and P. Enge, "Experimental and theoretical results on the LAAS sigma overbound," *Proceedings of the ION GPS 200*2, Sept. 2002.
5. B. DeCleene, "Defining pseudorange integrity," *Proceedings of the ION GPS 200*0, Sept. 2000.
6. J. Rife, S. Pullen and B. Pervan, "Core overbounding and its implications for LAAS integrity," *Proceedings of the ION GNSS 2004*, Sept. 2004.
7. RTCA, *Minimum Aviation System Performance Standards for the Local Area Augmentation System (LAAS)*, RTCA/DO-245, 1998.
8. M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, Vo. 50, 2002, pp. 174–188.
9. A. Doucet, N. J. Gordon and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *Technical Report CUED/F-INFENG/TR 359*, Cambridge University Engineering Department, 1999.
10. A. Kong, J. S. Liu and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, Vo. 89, 1994, pp. 278–288.
11. J. S. Liu and R. Chen, "Blind deconvolution via sequential imputations," *Journal of the American Statistical Association*, Vo. 90, 1995, pp. 567–576.
12. B. W. Parkinson and P. Axelrad, "Autonomous GPS integrity monitoring using the pseudorange residual," *Navigation: Journal of The Institute of Navigation*, Vo. 35, 1988, pp. 255–274.
13. T. Walter and P. Enge, "Weighted RAIM for precision approach," *Proceedings of the ION GPS 1995*, Sept. 1995.